

UNIVERSIDAD CATÓLICA DE CUENCA

UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

FACULTAD DE INFORMÁTICA Y CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA EN SISTEMAS DE LA INFORMACIÓN

DESARROLLO DE UN SISTEMA WEB PARA LA GESTIÓN DE TALENTO HUMANO EN LA COAC GAÑANSOL UTILIZANDO TÉCNICAS DE VIBE CODING

TRABAJO DE INVESTIGACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS DE LA INFORMACIÓN

AUTOR: JONNATHAN GUSTAVO PAUCAR PAUCAR.

DIRECTORA: ING. FERNANDA MICHAELLE NARVÁEZ OCHOA, MGS.

CAÑAR-ECUADOR

2026

DIOS, PATRIA, CULTURA Y DESARROLLO



UNIVERSIDADCATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

FACULTAD DE INFORMÁTICA Y CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA EN SISTEMAS DE LA INFORMACIÓN

DESARROLLO DE UN SISTEMA WEB PARA LA GESTIÓN DE TALENTO HUMANO EN LA COAC GAÑANSOL UTILIZANDO TÉCNICAS DE VIBE CODING

TRABAJO DE INVESTIGACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS DE LA INFORMACIÓN

AUTOR: JONNATHAN GUSTAVO PAUCAR PAUCAR.

DIRECTORA: ING. FERNANDA MICHAELLE NARVÁEZ OCHOA, MGS.

CAÑAR- ECUADOR

2026

DIOS, PATRIA, CULTURA Y DESARROLLO

DECLARATORIA DE AUTORÍA Y RESPONSABILIDAD

Jonnathan Gustavo Paucar Paucar portador(a) de la cédula de ciudadanía N.º **0302546098**, Declaro ser el autor@ de la obra: **"Desarrollo de un sistema web para la gestión de talento humano en la COAC Gañansol utilizando técnicas de Vibe coding"**, sobre la cual me hago responsable sobre las opiniones, versiones e ideas expresadas. Declaro que la misma ha sido elaborada respetando los derechos de propiedad intelectual de terceros y eximo a la Universidad Católica de Cuenca sobre cualquier reclamación que pudiera existir al respecto. Declaro finalmente que mi obra ha sido realizada cumpliendo con todos los requisitos legales, éticos y bioéticos de investigación, que la misma no incumple con la normativa nacional e internacional en el área específica de investigación, sobre la que también me responsabilizo y eximo a la Universidad Católica de Cuenca de toda reclamación al respecto.

Cañar,07 de octubre de 2026



……

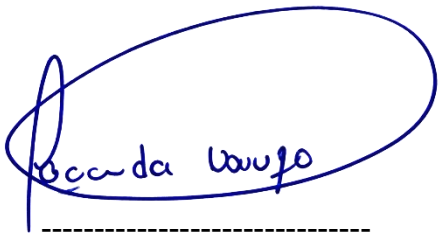
Jonnathan Gustavo Paucar Paucar

C.I. 0302546098

CERTIFICADO DEL TUTOR

Certifico que el presente trabajo denominado **"Desarrollo de un sistema web para la gestión de talento humano en la COAC Gañansol utilizando técnicas de Vibe coding"**, realizado por **Jonnathan Gustavo Paucar Paucar**, con documento de identidad No. **0302546098**, previo a la obtención del título profesional de Ingeniería en Sistemas de la Información, ha sido asesorado, supervisado y desarrollado bajo mi tutoría en todo su proceso, cumpliendo con la reglamentación pertinente que exige la Universidad Católica de Cuenca y los requisitos que determina la investigación científica.

Cañar, 07 de octubre de 2026



Ing. Fernanda Michaelle Narváez Ochoa

DEDICATORIA

Dedico este trabajo de investigación a mi familia, por su apoyo, comprensión y confianza durante mi formación universitaria. Su compañía y sus palabras de ánimo me impulsaron a continuar y a superar las dificultades que se presentaron en este camino. Este logro representa también el cariño y el esfuerzo de quienes estuvieron a mi lado en cada etapa.

A mis padres, por su amor, sus sacrificios y por enseñarme con su ejemplo el valor del esfuerzo, la responsabilidad y la perseverancia. Les dedico esta meta como una muestra de mi profunda gratitud por las oportunidades que me brindaron y por la confianza que depositaron en mí. Su apoyo me dio fuerzas para seguir adelante, incluso en los momentos de incertidumbre.

A mi hermana, por sus palabras de aliento, que me motivaron a seguir adelante durante mi formación universitaria. Te dedico este logro con mucho cariño y gratitud por animarme a alcanzar esta meta.

Finalmente, dedico este trabajo a la Universidad Católica de Cuenca, Campus Cañar, por ser parte fundamental de mi formación académica y profesional, y a quienes me acompañaron durante este proceso. A ustedes, que compartieron mis esfuerzos, celebraron mis avances y me animaron a no rendirme, les dedico esta meta alcanzada con cariño y gratitud.

AGRADECIMIENTO

Expreso mi sincero agradecimiento a quienes me acompañaron durante mi formación profesional y la culminación de este trabajo de investigación en Ingeniería en Sistemas de Información.

A mis padres, por su amor, sacrificio y apoyo incondicional, y por enseñarme el valor de la responsabilidad y la perseverancia. A mi hermana, por sus palabras de aliento, y a mi sobrino, cuya alegría y cariño me motivaron a seguir adelante.

A mi abuelo, que descansa en el cielo, por sus enseñanzas y valores, que continúan guiando mis pasos. A mi abuela, por su cariño, consejos y apoyo constante.

A mi tutora de investigación, por su orientación, paciencia y conocimientos, fundamentales para el desarrollo de esta investigación.

Finalmente, agradezco a la Universidad Católica de Cuenca, Campus Cañar, y a los docentes de la carrera de Ingeniería en Sistemas de Información, por contribuir a mi crecimiento profesional y personal.

RESUMEN

El presente trabajo de investigación aborda la ausencia de un sistema informático integrado para la gestión de Talento Humano en la Cooperativa de Ahorro y Crédito Gañansol, donde la información del personal se administraba mediante herramientas ofimáticas, registros físicos y exportaciones manuales del sistema biométrico, lo que generaba dispersión de datos, errores y demoras en la obtención de información. El objetivo fue desarrollar un sistema web para la gestión de talento humano utilizando técnicas de Vibe Coding y herramientas de desarrollo web, con el fin de automatizar y optimizar los procesos administrativos del personal. La investigación adoptó un enfoque mixto con predominio cualitativo y un nivel descriptivo; la información se obtuvo mediante reuniones de trabajo con el personal de Talento Humano y Tecnologías de la Información y el análisis documental de los formularios institucionales. El desarrollo se organizó con el marco de trabajo Scrum en seis sprints de cuatro semanas, empleando Python, Django, MySQL y herramientas de inteligencia artificial generativa como ChatGPT, Gemini y GitHub Copilot, bajo revisión humana obligatoria del código generado. Como resultado se definieron 23 requerimientos funcionales, 21 historias de usuario y 24 casos de uso, y se ejecutaron 21 casos de prueba funcional, todos con resultado «Cumple». Se concluye que el sistema centraliza en una sola plataforma los procesos de reclutamiento, inducción, personal, vacaciones, asistencia, desempeño y nómina, integra la autenticación con Active Directory y constituye una versión funcional lista para su adaptación e implementación institucional.

Palabras clave: gestión de talento humano, sistema web, Django, Scrum, Vibe Coding, Active Directory.

ABSTRACT

This thesis addresses the lack of an integrated information system for human resources management at the Gañansol Savings and Credit Cooperative, where personnel information was managed using office software, physical records, and manual exports from the biometric system—resulting in data dispersion, errors, and delays in retrieving information. This study aimed to develop a web-based system for human resources management using Vibe Coding techniques and web development tools, with the aim of automating and optimizing personnel administrative processes. The research adopted a mixed-methods approach with a qualitative focus and a descriptive scope; information was obtained through working meetings with Human Resources and Information Technology staff and through a documentary analysis of institutional forms. The development was organized using the Scrum framework into six four-week sprints, utilizing Python, Django, MySQL, and generative AI tools such as ChatGPT, Gemini, and GitHub Copilot, with mandatory human review of the generated code. As a result, 23 functional requirements, 21 user stories, and 24 use cases were defined, and 21 functional test cases were executed, all with a "Pass" result. It is concluded that the system centralizes the processes of recruitment, onboarding, human resources, vacation, attendance, performance, and payroll on a single platform; integrates authentication with Active Directory; and constitutes a functional version ready for institutional adaptation and implementation.

***Keywords*:** human talent management, web system, Django, Scrum, Vibe Coding, Active Directory

ÍNDICE

DECLARATORIA DE AUTORÍA Y RESPONSABILIDAD I
CERTIFICADO DEL TUTOR II
DEDICATORIA III
AGRADECIMIENTO IV
RESUMEN V
ABSTRACT VI
INTRODUCCIÓN XV
CAPÍTULO I 1
1.1 Planteamiento del problema 1
1.1.1 Formulación del problema 3
1.2 Antecedentes de la Investigación 4
1.3 Justificación de la investigación 5
1.4 Objetivos 6
1.4.1 Objetivo General 6
1.4.2 Objetivos Específicos 6
1.5 Limitaciones 7
1.6 Delimitaciones 7
CAPÍTULO II 8
2.1 Sistemas web 8
2.1.1 Tipos de sistemas web 8
2.1.2 Sistemas web administrativos 9
2.1.3 Sistemas web aplicados a la gestión organizacional 10
2.2 Sistemas de gestión de talento humano 10
2.2.1 Gestión de talento humano 11

2.2.2 Procesos de talento humano 12
2.2.3 Sistemas web aplicados al talento humano 12
2.3 Arquitectura y componentes de un sistema web 13
2.3.1 Frontend 13
2.3.2 Backend 14
2.3.3 Seguridad y autenticación 14
2.3.4 Integración con servicios externos 15
2.3.5 Escalabilidad del sistema 16
2.4 Tecnologías para el desarrollo web 16
2.4.1 Tecnologías frontend 17
2.4.1.1 HTML 17
2.4.1.2 CSS 17
2.4.1.3 JavaScript 17
2.4.1.4 TypeScript 18
2.4.1.5 React.js 18
2.4.2 Tecnologías backend 19
2.4.2.1 Python 19
2.4.2.2 Java 19
2.4.2.3 PHP 20
2.4.2.4 Django 20
2.4.2.5 Flask 21
2.4.2.6 FastAPI 21
2.4.3 Comparación de tecnologías frontend y backend 22
2.4.4 Bases de datos 25
2.4.5 MySQL 25
2.4.6 PostgreSQL 26
2.4.7 MongoDB 26
2.4.8 MariaDB 27

2.4.9 Comparación de sistemas gestores de bases de datos 27
2.5 Metodologías de desarrollo de software 29
2.5.1 Metodologías tradicionales de desarrollo 30
2.5.1.1 Metodología en cascada 30
2.5.1.2 Metodología espiral 30
2.5.1.3 Proceso Unificado Racional 31
2.5.1.4 Metodología de prototipado 31
2.5.2 Metodologías ágiles 32
2.5.2.1 Extreme Programming 32
2.5.2.2 Kanban 32
2.5.2.3 Scrum 33
2.5.3 Comparación de metodologías de desarrollo de software 34
2.6 Vibe Coding 36
2.6.1 Aplicación del Vibe Coding en ingeniería de software 37
2.6.2 Implicaciones técnicas en el desarrollo web 38
2.6.3 Impacto en la calidad y seguridad del software 39
2.6.4 Supervisión humana y validación del código generado 40
2.6.5 Integración de inteligencia artificial en metodologías ágiles 40
CAPITULO III 41
3.1 Enfoque de la investigación 41
3.2 Nivel de la investigación 42
3.3 Población y muestra 43
3.4 Técnicas e instrumentos de recolección de información 44
3.4.1 Reuniones de trabajo y levantamiento de requerimientos 45
3.4.2 Análisis documental 46
3.4.3 Validación de los instrumentos 46
3.5 Tratamiento y análisis de la información 48

3.6 Proceso de desarrollo mediante Scrum y Vibe Coding 48
3.6.1 Roles de Scrum 48
3.6.2 Artefactos de Scrum 49
3.6.3 Eventos y seguimiento de los sprints 49
3.6.4 Aplicación de Vibe Coding durante los sprints 50
3.6.5 Base tecnológica de partida y licenciamiento 51
3.7 Validación funcional del sistema 52
3.8 Consideraciones éticas 53
3.9 Resultados 53
3.9.1 Resultados del levantamiento de requerimientos 54
3.9.2 Análisis general 57
CAPÍTULO IV 58
4.1 Título de la Propuesta 58
4.2 Presentación 58
4.3 Justificación 59
4.4 Descripción de la propuesta 60
4.5 Desarrollo de la propuesta 62
4.5.1 Organización del desarrollo mediante Scrum 62
4.5.2 Definición del Product Backlog 64
4.5.2.1 Requerimientos no funcionales 68
4.5.2.2 Casos de uso del sistema 70
4.5.2.3 Matriz de trazabilidad de requisitos 73
4.5.2.4 Especificación textual de los casos de uso 78
4.5.3 Arquitectura tecnológica del sistema 87
4.5.3.1 Capa de presentación 87
4.5.3.2 Capa de aplicación 88
4.5.3.3 Capa de datos 88

4.5.3.4 Seguridad y control de acceso 89
4.5.3.5 Integración con Active Directory (AD) 90
4.5.3.6 Módulo de asistencia y actividades vinculado al control biométrico 91
4.5.3.7 Diseño de la base de datos 91
4.5.3.8 Diagrama de arquitectura y componentes del sistema 93
4.5.4 Herramientas utilizadas 95
4.5.4.1 Justificación de las decisiones de diseño 96
4.5.4.2 Flujo de funcionamiento del sistema 99
4.5.5 Desarrollo de los sprints 104
4.5.5.1 Sprint 1: Configuración, autenticación y estructura inicial 104
4.5.5.2 Sprint 2: Reclutamiento, onboarding y gestión de empleados 105
4.5.5.3 Sprint 3: Asistencia, permisos y vacaciones 106
4.5.5.4 Sprint 4: Evaluaciones, novedades de nómina y liquidaciones 107
4.5.5.5 Sprint 5: Activos, pasantías y reportes 107
4.5.5.6 Sprint 6: Integración, pruebas y consolidación 108
4.5.6 Aplicación de técnicas de Vibe Coding 109
4.5.6.1 Modelos de inteligencia artificial utilizados y evaluación de su desempeño 110
4.5.7 Validación funcional del sistema 114
4.5.8 Retroalimentación y ajustes realizados 117
4.5.9 Producto final 118
4.5.10 Resultados de la propuesta 121
4.6 Evidencias de la interfaz y funcionalidades del sistema desarrollado 121
4.6.1 Panel principal (Dashboard) del sistema 122
4.6.2 Módulo de Reclutamiento — Ficha de entrevista (FO-TH-015) 123
4.6.3 Módulo de Onboarding — Inducción y reinducción (FO-TH-004) 124
4.6.4 Módulo de Empleado — Base de datos de personal (FO-TH-001) 126
4.6.5 Módulo de Licencia — Control de vacaciones (FO-TH-008) 127
4.6.6 Módulo de Asistencia — Informe de actividades virtual (FO-TH-016) y datos biométricos 127

4.6.7 Módulo de Desempeño — Evaluación por competencias (FO-TH-007) 128
4.6.8 Módulo de Nómina — Solicitud de acumulación o mensualización de décimos (FO-TH-003)129
4.6.9 Resumen de operaciones CRUD verificadas por módulo..... 130
4.7 Evidencias de la integración con Active Directory..... 133
4.8 Discusión..... 136
CONCLUSIONES 139
RECOMENDACIONES..... 140
BIBLIOGRAFÍA 141
1. Tecnologías de la Información (TI) 156
5. *Sistema Web* 158
7. *Operaciones CRUD* 159
8. *Metodologías de Desarrollo de Software* 159
9. *Metodología en Cascada*..... 159
10. *Metodología Espiral*..... 159
11. *RUP (Proceso Unificado Racional)*..... 160
12. *Metodología de Prototipo*..... 160
13. *Metodologías Ágiles*..... 160
14. *Extreme Programming (XP)* 160
15. *Scrum*..... 160
16. *Kanban* 161
17. *Sistema de Gestión de Talento Humano*..... 161
Referencias..... 169
ANEXOS 173
Pasos para iniciar sesión 185

Acceso y control 187
Datos personales 187
Ubicación y contacto..... 187
Información complementaria..... 188
¿Qué es Horilla?..... 228
Alcance de este manual..... 228
Pasos para asignar permisos desde la interfaz 234
Alternativa desde el panel de administración de Django..... 234
Comandos de diagnóstico útiles..... 241

ÍNDICE DE TABLAS

Tabla 1. Comparación de tecnologías frontend y backend 22
Tabla 2. Comparación de sistemas gestores de bases de datos 27
Tabla 3. Comparación de metodologías de desarrollo de software 34
Tabla 4. Resultados del levantamiento de requerimientos para el sistema web de gestión de Talento Humano 54
Tabla 5. Planificación general de los sprints 63
Tabla 7. Requerimientos funcionales del sistema 65
Tabla 8. Requerimientos no funcionales del sistema 68
Tabla 9. Matriz de trazabilidad entre requerimientos, historias de usuario, casos de uso, funcionalidades y pruebas 73
Tabla 10. Herramientas utilizadas en el desarrollo 95
Tabla 11. Correspondencia entre las comparaciones del Capítulo II y las decisiones de diseño 96
Tabla 12. Aplicación de Vibe Coding durante el desarrollo 109
Tabla 13. Modelos de inteligencia artificial empleados y su configuración 111
Tabla 14. Indicadores utilizados para evaluar el apoyo de las herramientas generativas 112
Tabla 15. Casos de prueba funcional y resultados 114
Tabla 16. Principales observaciones y ajustes realizados 118
Tabla 17. Entregables finales de la propuesta 119

INTRODUCCIÓN

Las cooperativas de ahorro y crédito forman parte esencial del Sector Financiero Popular y Solidario del Ecuador, por lo que la eficiencia de sus procesos internos incide directamente en la calidad de los servicios que ofrecen a sus socios. Dentro de estas organizaciones, el área de Talento Humano administra información sensible y de uso permanente, como datos personales, contratos, cargos, asistencia, vacaciones, evaluaciones de desempeño y procesos de nómina. Cuando esta información se gestiona de forma manual y dispersa, se incrementan la carga operativa, el riesgo de errores y el tiempo necesario para obtener datos confiables que respalden la toma de decisiones.

El tema investigado es importante porque la COAC Gañansol no dispone de un sistema informático integrado para la gestión de Talento Humano. La información del personal se maneja mediante herramientas ofimáticas y registros documentales físicos, y los datos del sistema biométrico deben exportarse manualmente a hojas de cálculo para su control. Esta situación limita la centralización, trazabilidad e integridad de la información, por lo que resulta pertinente desarrollar una solución tecnológica que automatice estos procesos y fortalezca la gestión administrativa de la cooperativa. Además, el trabajo aporta una experiencia aplicada sobre el uso de técnicas de Vibe Coding, es decir, el desarrollo de software asistido por inteligencia artificial generativa bajo supervisión humana.

El objetivo de la investigación fue desarrollar un sistema web para la gestión de talento humano en la COAC Gañansol utilizando técnicas de Vibe Coding y herramientas de desarrollo web, para automatizar y optimizar los procesos administrativos del personal. Para alcanzarlo se plantearon tres objetivos específicos: analizar los fundamentos teóricos del sistema, evaluar las metodologías de desarrollo y las técnicas de Vibe Coding aplicables, y desarrollar el sistema considerando el levantamiento de requerimientos, el modelado, la construcción y las pruebas en entorno local.

En cuanto a la forma en que se realizó la investigación, se adoptó un enfoque mixto con predominio cualitativo y un nivel descriptivo. El componente cualitativo se desarrolló mediante reuniones de trabajo con el personal de Talento Humano y Tecnologías de la Información y el análisis documental de los formularios y registros institucionales, lo que permitió identificar los requerimientos funcionales y no funcionales del sistema. El componente cuantitativo se incorporó en la validación funcional, a través de casos de prueba ejecutados por módulo. El desarrollo se organizó con el marco de trabajo Scrum en seis sprints de cuatro semanas, utilizando Python, Django y MySQL, y aplicando técnicas de Vibe Coding con apoyo de ChatGPT, Gemini y GitHub Copilot, con revisión humana obligatoria del código generado.

El alcance del trabajo comprende el análisis, diseño, construcción y validación funcional de un sistema web que integra los módulos de reclutamiento, inducción y reinducción, base de datos del personal, control de vacaciones, asistencia, evaluación del desempeño y nómina, así como la autenticación con Active Directory. La validación se efectuó en un entorno local de pruebas, por lo que no incluye la implementación definitiva en la infraestructura de producción de la cooperativa.

El documento se organiza en cuatro capítulos. El Capítulo I presenta el planteamiento del problema, los antecedentes, la justificación, los objetivos, las limitaciones y las delimitaciones. El Capítulo II desarrolla el marco teórico sobre sistemas web, gestión de talento humano, tecnologías de desarrollo, metodologías de software y Vibe Coding. El Capítulo III describe el marco metodológico y los resultados del levantamiento de requerimientos. Finalmente, el Capítulo IV expone la propuesta, el desarrollo de los sprints, la validación funcional y las evidencias del sistema, seguido de las conclusiones y recomendaciones.

CAPÍTULO I

1.1 Planteamiento del problema

Las organizaciones financieras, como las cooperativas de ahorro y crédito, requieren una gestión eficiente del talento humano que permita administrar adecuadamente la información del personal, optimizar los procesos administrativos y fortalecer la toma de decisiones institucionales. En la actualidad, la transformación digital ha impulsado la adopción de sistemas de información capaces de integrar y automatizar procesos relacionados con la administración del personal, tales como el registro de empleados, control de asistencia, gestión de contratos, evaluación del desempeño y generación de reportes. En este contexto, el uso de soluciones tecnológicas resulta fundamental para centralizar la información, garantizar su integridad, reducir la dependencia de procedimientos manuales y facilitar el acceso oportuno a datos relevantes para la gestión organizacional.

La importancia de fortalecer los procesos internos de las cooperativas también se evidencia en el avance de la transformación digital del sector. Según la Superintendencia de Economía Popular y Solidaria (SEPS), el canal Oficina redujo su participación en las transacciones del Sector Financiero Popular y Solidario del 65,9 % en 2019 al 49,9 % en 2024, mientras que los canales digitales, como la banca celular e internet, mantienen un crecimiento superior al 30 % anual. Estos datos reflejan una tendencia sostenida hacia la digitalización, lo que hace necesario que las cooperativas modernicen no solo los servicios dirigidos a los socios, sino también los procesos administrativos internos, entre ellos la gestión del talento humano (Superintendencia de Economía Popular y Solidaria, 2025).

En contraste con este escenario de transformación digital y con la necesidad de centralizar la información institucional, la COAC Gañansol no dispone actualmente de un sistema informático integrado para la gestión de los procesos de Talento Humano. En consecuencia, la información relacionada con el personal, incluidos los datos personales, contratos, cargos, departamentos y evaluaciones de desempeño, se administra mediante herramientas ofimáticas y registros documentales físicos. Esta situación limita la centralización, trazabilidad e integridad de los datos, generando dispersión de la información y aumentando el esfuerzo requerido para gestionar procesos administrativos importantes. Como consecuencia, la búsqueda, actualización y validación de información se vuelve más lenta y propensa a errores, afectando la eficiencia operativa del área de Talento Humano.

Por otra parte, en lo que respecta al control de asistencia, el sistema biométrico no se encuentra conectado a ningún sistema de gestión, por lo que los datos deben exportarse de forma manual a hojas de cálculo para su control y análisis. Esto convierte el procedimiento en una tarea tediosa y repetitiva, que puede generar discrepancias en los registros, pérdida de información y problemas en la validación de los datos. Como consecuencia, se dificulta el seguimiento adecuado del personal, el control de su comportamiento laboral y la confiabilidad de la información gestionada. De igual manera, los procesos de contratación, la administración de cargos y departamentos y la evaluación del desempeño no se gestionan de forma centralizada, lo que incrementa la dispersión de la información y el esfuerzo operativo. Estas limitaciones, sumadas a los tiempos prolongados que requiere la obtención de datos, afectan la eficiencia de la gestión cooperativa y la toma de decisiones institucionales.

1.1.1 Formulación del problema

A pesar del avance de las herramientas digitales en las organizaciones financieras, muchas cooperativas aún presentan limitaciones en la automatización de sus procesos internos, especialmente en áreas administrativas como Talento Humano. En la COAC Gañansol, la gestión de la información del personal se realiza mediante herramientas ofimáticas y registros documentales físicos, lo que dificulta la centralización, actualización, trazabilidad y control adecuado de los datos relacionados con empleados, cargos, contratos, asistencia, permisos, desempeño y demás procesos administrativos.

Asimismo, al no contar con un sistema para la gestión de talento humano, la cooperativa mantiene procesos manuales que incrementan la carga operativa del personal responsable, generan mayor probabilidad de errores y dificultan la obtención de información oportuna para la toma de decisiones. Esta situación también afecta el control de asistencia, debido a que la información debe ser procesada manualmente, lo cual limita la validación, organización y análisis eficiente de los datos laborales.

Frente a esta situación, se evidencia la necesidad de contar con una solución tecnológica que permita mejorar la gestión de la información del talento humano, optimizar los procesos administrativos y fortalecer la disponibilidad de datos confiables para la toma de decisiones institucionales. Por ello, surge la siguiente pregunta de investigación:

¿De qué manera el desarrollo de un sistema web para la gestión de talento humano puede contribuir a optimizar los procesos administrativos del personal en la COAC Gañansol, considerando las limitaciones actuales en la centralización, trazabilidad y disponibilidad de la información?

1.2 Antecedentes de la Investigación

Gaibor (2025) señala que la implementación de sistemas de información para la gestión de recursos humanos permite reducir tiempos de procesamiento, minimizar errores y optimizar procesos administrativos relacionados con el personal. Este aporte resulta relevante para la presente investigación, debido a que evidencia la importancia de contar con herramientas tecnológicas que permitan centralizar la información, mejorar la organización de los datos y apoyar la toma de decisiones en el área de Talento Humano.

Ncube et al. (2025), en el estudio titulado “The impact of artificial intelligence on human resource management practices: An investigation”, analizan el impacto de la inteligencia artificial en la gestión de recursos humanos. Los autores concluyen que la inteligencia artificial contribuye a mejorar la eficiencia organizacional mediante la automatización de tareas, el análisis de información y el apoyo a la toma de decisiones. Este antecedente sustenta la importancia de incorporar herramientas tecnológicas que apoyen la gestión del talento humano y fortalezcan los procesos administrativos.

En el contexto nacional, Tomalá Lino (2025), establece que la automatización de procesos administrativos permite mejorar la trazabilidad, consistencia e integridad de la información, además de reducir la intervención manual en actividades repetitivas. Este aporte se relaciona con la presente investigación, ya que respalda la necesidad de optimizar los procesos de gestión del talento humano mediante soluciones digitales que permitan organizar y controlar la información institucional.

Mixi et al. (2023) destacan la importancia de las bases de datos como componente fundamental para el almacenamiento, procesamiento y análisis de información dentro de los sistemas informáticos. Este antecedente aporta al desarrollo de la investigación, debido a que

sustenta la necesidad de contar con una base de datos centralizada que permita gestionar información relacionada con el personal, mejorar la disponibilidad de los datos y facilitar la generación de reportes para la toma de decisiones.

Cueva (2023) fundamenta la gestión del talento humano como un conjunto de procesos estratégicos relacionados con la planificación, selección, desarrollo y evaluación del personal. Estos elementos permiten comprender la importancia de administrar adecuadamente la información laboral y fortalecer los procesos internos de una organización. Por ello, este antecedente contribuye a sustentar la necesidad de desarrollar un sistema web orientado a la gestión integral del talento humano en la COAC Gañansol.

Guerra (2026) evidencia que los sistemas de control de asistencia contribuyen a mejorar la precisión de los registros, reducir errores administrativos y fortalecer el seguimiento del personal. Este aporte resulta pertinente para la investigación, debido a que permite comprender la importancia de contar con información confiable y organizada para apoyar la gestión administrativa del talento humano.

1.3 Justificación de la investigación

El presente proyecto surge de la necesidad de mejorar la gestión del talento humano en la COAC Gañansol, debido a que varios procesos administrativos relacionados con la información del personal se desarrollan mediante herramientas ofimáticas y registros documentales físicos. Esta situación genera dispersión de datos, inconsistencias, riesgo de pérdida de información y limitaciones para acceder de manera oportuna a datos confiables. Por ello, resulta necesario proponer una solución que contribuya a centralizar la información, optimizar los procesos internos y fortalecer la eficiencia administrativa de la institución.

Desde el punto de vista académico y científico, la investigación es relevante porque permite aplicar conocimientos relacionados con el desarrollo de sistemas de información, automatización de procesos y gestión organizacional en un contexto real. Además, aporta al análisis de cómo las soluciones tecnológicas pueden mejorar la administración del talento humano en instituciones financieras cooperativas, donde la disponibilidad, trazabilidad y confiabilidad de la información son aspectos fundamentales para la toma de decisiones.

En este contexto, el desarrollo de un sistema web para la gestión del talento humano permitirá mejorar el registro, organización, consulta y control de la información del personal dentro de la COAC Gañansol. Asimismo, contribuirá a reducir la carga operativa, disminuir errores en el manejo de datos y facilitar la generación de reportes e indicadores para el seguimiento administrativo. De esta manera, el proyecto representa un aporte práctico para la cooperativa y una experiencia académica significativa en el diseño de soluciones informáticas orientadas a resolver necesidades institucionales reales.

1.4 Objetivos

1.4.1 Objetivo General

Desarrollar un sistema web para la gestión de talento humano en la COAC Gañansol utilizando técnicas de Vibe coding y herramientas de desarrollo web, para automatizar y optimizar los procesos administrativos del personal.

1.4.2 Objetivos Específicos

- Analizar los fundamentos teóricos del sistema web de gestión de talento humano, mediante la revisión de literatura científica en TI, ingeniería de software y Vibe coding, para sustentar el desarrollo del sistema en la COAC Gañansol.

- Evaluar las metodologías y enfoques de desarrollo aplicables al sistema web y técnicas de Vibe Coding asistidas por inteligencia artificial, mediante la identificación de sus fases, roles, artefactos y lineamientos de aplicación, para estructurar el proceso de desarrollo del sistema.

- Desarrollar el sistema web de gestión de talento humano, considerando el levantamiento de requerimientos, modelado, construcción y pruebas en entorno local, para automatizar y optimizar los procesos administrativos del personal en la COAC Gañansol.

1.5 Limitaciones

- El tiempo disponible para el desarrollo del trabajo puede limitar la profundidad del análisis, construcción y validación funcional del sistema web.
- El acceso a la información institucional de la COAC Gañansol estará sujeto a la autorización y disponibilidad del área de Talento Humano y Tecnologías de la Información.
- La validación del sistema se realizará en un entorno de pruebas, por lo que no se contempla su implementación definitiva en producción dentro de la infraestructura tecnológica institucional.
- La integración con herramientas o servicios externos se considerará únicamente a nivel lógico o funcional, sin realizar configuraciones reales en sistemas institucionales externos.

1.6 Delimitaciones

- La investigación se desarrollará en la COAC Gañansol, enfocándose en el área de Talento Humano
- El estudio se centrará en los procesos administrativos relacionados con la gestión de la información del personal de la cooperativa.
- La investigación abarcará el diseño, desarrollo y validación funcional de un sistema web para la gestión de talento humano en un entorno de pruebas.
- El alcance del trabajo llegará hasta la comprobación funcional del sistema, sin contemplar la puesta en producción definitiva ni la integración completa con la infraestructura tecnológica institucional.

CAPÍTULO II

2. MARCO TEÓRICO

2.1 Sistemas web

Un sistema web es una aplicación de software que funciona mediante tecnologías de internet y permite a los usuarios acceder a información, servicios y funcionalidades a través de un navegador o cliente web. Estos sistemas se caracterizan por no requerir instalación directa en cada equipo, ya que su acceso se realiza mediante una red, lo que facilita su disponibilidad, actualización y administración centralizada. Ehsan et al. (2022) señalan que los sistemas web modernos pueden integrarse con otros servicios digitales mediante interfaces de programación de aplicaciones, conocidas como API, lo que permite ampliar sus funcionalidades y facilitar la comunicación entre plataformas.

Los sistemas web actuales no solo cumplen funciones informativas, sino que también permiten registrar datos, automatizar procesos, administrar usuarios, generar reportes y conectar servicios externos. Por esta razón, su diseño debe considerar atributos técnicos como seguridad, interoperabilidad, escalabilidad, rendimiento y mantenibilidad. Bushong et al. (2021) destacan que los sistemas organizados mediante componentes o servicios facilitan la evolución del sistema, aunque también requieren mayor control sobre la comunicación entre sus partes.

2.1.1 Tipos de sistemas web

Los sistemas web pueden clasificarse de acuerdo con su finalidad, nivel de interacción y función dentro de una organización. Entre los principales tipos se encuentran los sistemas informativos, transaccionales, administrativos e integrados. Los sistemas informativos se

orientan a la publicación y consulta de contenido; los transaccionales permiten registrar, consultar y actualizar información; los administrativos apoyan la gestión interna de procesos; y los integrados permiten la conexión entre diferentes servicios, bases de datos o plataformas digitales. Ehsan et al. (2022) describen las API REST como el mecanismo estándar para la integración entre aplicaciones y plataformas, que es el utilizado por los sistemas integrados para comunicarse con otros servicios.

Esta clasificación permite diferenciar entre sistemas de baja interacción, como páginas informativas, y sistemas de mayor complejidad, como plataformas administrativas o integradas. Los sistemas más complejos requieren mecanismos de autenticación, autorización, validación de entradas, conexión con bases de datos y protección de la información. Rennhard et al. (2022) indican que los sistemas web de mayor complejidad deben considerar controles de seguridad adecuados para evitar accesos no autorizados, errores de validación o exposición indebida de datos.

2.1.2 Sistemas web administrativos

Los sistemas web administrativos son aplicaciones orientadas a apoyar la gestión interna de una organización mediante el registro, procesamiento, consulta y control de información. Este tipo de sistema permite digitalizar actividades que antes se realizaban mediante documentos físicos, hojas de cálculo o registros dispersos, contribuyendo a mejorar la eficiencia operativa y la disponibilidad de los datos. Bushong et al. (2021) señalan que los sistemas web administrativos permiten digitalizar actividades, centralizar información, reducir errores y facilitar el control de procesos organizacionales.

Desde el punto de vista técnico, los sistemas web administrativos requieren una estructura que permita separar la presentación visual, la lógica de funcionamiento y el

almacenamiento de datos. Esta organización facilita el mantenimiento del sistema, permite incorporar mejoras de forma progresiva y favorece la adaptación a nuevas necesidades institucionales. Bucko et al. (2023) explican que los sistemas web que manejan información restringida deben implementar mecanismos de autenticación, autorización y control de acceso para proteger los datos y limitar las acciones de cada usuario.

2.1.3 Sistemas web aplicados a la gestión organizacional

Los sistemas web aplicados a la gestión organizacional permiten integrar procesos internos en una plataforma digital accesible desde diferentes dispositivos conectados a una red. Estas soluciones facilitan la coordinación de actividades, el registro de información, la automatización de tareas y la generación de reportes que apoyan la toma de decisiones. Bushong et al. (2021) mencionan que los sistemas web aplicados a la gestión organizacional facilitan la coordinación de actividades, el registro de información, la automatización de tareas y la generación de reportes.

La gestión organizacional mediante sistemas web también requiere interoperabilidad, es decir, la capacidad de conectar la aplicación con otros sistemas, servicios o fuentes de datos. Esta característica permite que la información circule de manera más ordenada entre diferentes componentes tecnológicos y reduce la duplicidad de registros. Ehsan et al. (2022) señalan que las API REST permiten que diferentes aplicaciones intercambien información mediante solicitudes estructuradas, aunque su implementación requiere pruebas, validación de entradas, control de autenticación y revisión de seguridad.

2.2 Sistemas de gestión de talento humano

Priyashantha et al. (2023) explican que los sistemas de gestión de talento humano son soluciones tecnológicas orientadas a administrar información y procesos relacionados con el

personal de una organización. Estas plataformas pueden apoyar actividades como reclutamiento, selección, contratación, inducción, registro de empleados, asistencia, permisos, capacitación, evaluación del desempeño y gestión documental. En la literatura académica, este tipo de soluciones se relaciona con el concepto de electronic Human Resource Management o e-HRM, entendido como el uso de tecnologías web para apoyar funciones de recursos humanos.

Alshammari et al. (2024) señalan que la gestión digital del talento humano permite mejorar la disponibilidad de la información, reducir la carga administrativa y facilitar el análisis de datos relacionados con los colaboradores. Además, estas tecnologías permiten transformar las prácticas tradicionales de recursos humanos mediante plataformas que integran información, automatizan actividades y favorecen decisiones organizacionales basadas en datos.

2.2.1 Gestión de talento humano

Priyashantha et al. (2023) señalan que la gestión de talento humano comprende el conjunto de procesos orientados a atraer, organizar, desarrollar, evaluar y retener al personal dentro de una organización. Esta gestión incluye actividades relacionadas con competencias, desempeño, capacitación, bienestar laboral y alineación del personal con los objetivos institucionales. En entornos digitales, la gestión de talento humano se apoya cada vez más en sistemas de información que permiten automatizar procesos y mejorar la disponibilidad de datos laborales.

Alshammari et al. (2024) explican que la transformación digital ha influido directamente en la gestión de talento humano mediante la incorporación de plataformas web, analítica de datos, automatización y tecnologías emergentes. Estas herramientas contribuyen a mejorar la eficiencia de los procesos, fortalecer la experiencia del empleado y facilitar la toma de decisiones en las organizaciones.

2.2.2 Procesos de talento humano

Priyashantha et al. (2023) señalan que los procesos de talento humano son actividades organizadas que permiten administrar el ciclo laboral de los colaboradores dentro de una institución. Entre estos procesos se encuentran el reclutamiento, selección, contratación, inducción, capacitación, control de asistencia, gestión de permisos, evaluación del desempeño y administración de información laboral. Estos procesos requieren información confiable, actualizada y disponible para que la organización pueda gestionar adecuadamente a su personal.

Rodríguez et al. (2024) señalan que la digitalización de los procesos de talento humano permite reducir errores, mejorar la trazabilidad y facilitar el acceso a datos relevantes para la gestión interna. Además, los sistemas e-HRM contribuyen a transformar las prácticas tradicionales de recursos humanos mediante plataformas que integran información y automatizan actividades administrativas.

2.2.3 Sistemas web aplicados al talento humano

Abuhantash (2023) señala que los sistemas web aplicados al talento humano son plataformas digitales diseñadas para gestionar de manera integrada la información y los procesos relacionados con los colaboradores de una organización. Estas herramientas permiten centralizar datos, automatizar actividades administrativas, gestionar permisos, evaluaciones y generar reportes, contribuyendo a una administración más eficiente y organizada de los recursos humanos.

Rennhard et al. (2022) explican que los sistemas web que administran información personal y laboral requieren mecanismos de seguridad y control de acceso para proteger los datos frente a accesos no autorizados. Por ello, los sistemas web de talento humano deben considerar

autenticación, autorización, roles de usuario, validación de entradas y protección de la base de datos.

2.3 Arquitectura y componentes de un sistema web

La arquitectura de un sistema web se refiere a la forma en que se organizan sus componentes principales para permitir el funcionamiento adecuado de una aplicación. Esta arquitectura define cómo se relacionan la interfaz de usuario, la lógica de procesamiento, la base de datos, los servicios externos y los mecanismos de seguridad. Putra et al. (2025) explican que la arquitectura de un sistema web define la estructura y organización de sus componentes, estableciendo la forma en que interactúan el frontend, el backend y la base de datos.

Una arquitectura web adecuada permite que el sistema sea más ordenado, mantenible y adaptable a los cambios. Cuando los componentes se encuentran correctamente separados, es más fácil realizar actualizaciones, corregir errores, incorporar nuevas funcionalidades y mejorar el rendimiento de la aplicación. Putra et al. (2025) señalan que las arquitecturas orientadas a API permiten construir aplicaciones escalables, mantenibles y adaptables a los requerimientos organizacionales.

2.3.1 Frontend

El frontend corresponde a la capa visual e interactiva de una aplicación web con la que el usuario se relaciona directamente mediante el navegador. Esta parte del sistema se encarga de presentar formularios, botones, menús, tablas, paneles, mensajes y demás elementos gráficos que permiten ingresar, consultar y visualizar información. Además, puede realizar validaciones iniciales y enviar las solicitudes del usuario al backend para su procesamiento. Pérez y Quispe (2021) explican que el frontend funciona como un medio de comunicación entre el usuario y la aplicación, al integrar la presentación visual, la interacción y la validación de datos del lado del

cliente. En un sistema web de gestión de talento humano, el frontend permite acceder a los módulos de empleados, asistencia, permisos, vacaciones, evaluaciones de desempeño, novedades de nómina y reportes administrativos.

2.3.2 Backend

El backend es la capa interna del sistema web encargada de procesar las solicitudes realizadas desde el frontend. Esta capa ejecuta la lógica de negocio, valida datos, gestiona usuarios, aplica reglas internas y se comunica con la base de datos para registrar, consultar o actualizar información. Azimov (2021) señala que el backend procesa la lógica de negocio, valida datos, gestiona usuarios, aplica reglas del sistema y permite la comunicación con la base de datos.

El backend cumple una función fundamental dentro de una aplicación web, debido a que controla el funcionamiento interno del sistema y garantiza que las operaciones se ejecuten de manera correcta. Por ello, su diseño debe considerar aspectos como seguridad, rendimiento, integración, organización del código y facilidad de mantenimiento. Torres y Hora (2026) explican que el backend debe considerar mantenibilidad, seguridad, rendimiento e integración con otros servicios.

2.3.3 Seguridad y autenticación

La seguridad en un sistema web comprende el conjunto de medidas orientadas a proteger la información, los usuarios y las funcionalidades de una aplicación frente a accesos no autorizados, errores o vulnerabilidades. Este aspecto debe considerarse desde el diseño del sistema, debido a que las aplicaciones web suelen manejar información sensible y requieren mecanismos de protección adecuados. Rennhard et al. (2022) señalan que la seguridad en

sistemas web debe considerar autenticación, autorización, validación de entradas, cifrado, control de sesiones, pruebas de seguridad y protección frente a vulnerabilidades.

La autenticación y la autorización son elementos importantes dentro de la seguridad de un sistema web. La autenticación permite comprobar la identidad del usuario que ingresa al sistema, mientras que la autorización determina qué acciones puede realizar de acuerdo con su perfil o nivel de acceso. Bucko et al. (2023) explican que la autenticación permite verificar la identidad de un usuario, mientras que la autorización determina qué acciones puede realizar dentro del sistema.

2.3.4 Integración con servicios externos

La integración con servicios externos permite que un sistema web se comunique con otras aplicaciones, plataformas o fuentes de datos. Esta integración facilita el intercambio de información y permite ampliar las funcionalidades del sistema sin que todos los procesos dependan únicamente de la aplicación principal. Ehsan et al. (2022) señalan que las API REST permiten intercambiar información entre aplicaciones mediante solicitudes estructuradas.

Para que la integración con servicios externos sea adecuada, es necesario considerar aspectos como seguridad, validación de datos, control de errores, autenticación y rendimiento. Una integración mal diseñada puede generar fallos de comunicación, pérdida de información o vulnerabilidades en el sistema. Ehsan et al. (2022) indican que la integración de servicios externos requiere evaluar entradas, respuestas, autenticación, rendimiento y comportamiento ante errores antes de integrar servicios en ambientes de producción.

2.3.5 Escalabilidad del sistema

La escalabilidad es la capacidad que tiene un sistema para mantener un funcionamiento adecuado cuando aumenta la cantidad de usuarios, datos o solicitudes. En los sistemas web, este aspecto es importante porque permite que la aplicación pueda crecer de acuerdo con las necesidades de la organización sin afectar significativamente su rendimiento. Assunção et al. (2023) señalan que la escalabilidad permite que un sistema mantenga niveles adecuados de rendimiento y disponibilidad cuando aumenta la carga de trabajo, el volumen de datos o la cantidad de usuarios.

Un sistema escalable debe diseñarse con componentes organizados, flexibles y fáciles de modificar. Esto permite incorporar nuevas funciones, mejorar procesos existentes y responder a mayores demandas de uso. Assunção et al. (2023) explican que las arquitecturas modulares contribuyen a mejorar la mantenibilidad y flexibilidad de las aplicaciones.

2.4 Tecnologías para el desarrollo web

Las tecnologías para el desarrollo web permiten construir aplicaciones accesibles mediante navegadores, integrando componentes visuales, lógica de procesamiento, servicios de comunicación y almacenamiento de datos. En una arquitectura web moderna, es común separar el frontend, el backend y la base de datos, con el fin de organizar mejor las responsabilidades del sistema. Bombini et al. (2022) evidencian que, en las aplicaciones web, el frontend suele implementarse con HTML, CSS y JavaScript, mientras que el backend se encarga de procesar la lógica y exponer servicios mediante API o mecanismos de comunicación web.

2.4.1 Tecnologías frontend

2.4.1.1 HTML

Howard et al. (2025) señalan que HTML es un lenguaje de marcado utilizado para estructurar el contenido de las páginas web. Mediante etiquetas, permite organizar elementos como títulos, párrafos, imágenes, enlaces, tablas, formularios, botones y campos de entrada. En un sistema web de gestión de talento humano, HTML facilita la construcción de formularios para registrar empleados, consultar información del personal, presentar reportes y acceder a los diferentes módulos administrativos.

2.4.1.2 CSS

Howard et al. (2025) indican que CSS es una tecnología empleada para definir la presentación visual de las páginas web. Permite establecer colores, márgenes, tamaños, tipografías, alineaciones y la distribución de los elementos que conforman una interfaz. En un sistema de gestión de talento humano, CSS contribuye a presentar formularios, tablas, paneles y menús de manera ordenada, uniforme y adaptable a diferentes dispositivos, mejorando la experiencia de los usuarios.

2.4.1.3 JavaScript

Howard et al. (2025) explican que JavaScript es un lenguaje de programación utilizado para incorporar interactividad y comportamiento dinámico en las aplicaciones web. Permite gestionar eventos, validar formularios, actualizar información y controlar las acciones realizadas por el usuario dentro del navegador. En un sistema de gestión de talento humano, JavaScript puede utilizarse para validar datos de empleados, filtrar registros, mostrar notificaciones, realizar cálculos y actualizar información sin recargar completamente la página.

2.4.1.4 TypeScript

TypeScript es un lenguaje de programación basado en JavaScript que incorpora tipado estático para definir los tipos de variables, funciones, propiedades y objetos. Esta característica permite detectar errores durante el desarrollo, mejorar la organización del código y facilitar el mantenimiento de aplicaciones web con numerosos componentes. Rosa et al. (2024) utilizaron TypeScript junto con React.js para desarrollar el frontend de una aplicación web interactiva, destacando que la definición de tipos contribuyó a detectar errores y mejorar la calidad del código. En un sistema web de gestión de talento humano, TypeScript puede utilizarse para desarrollar formularios, paneles, validaciones y componentes relacionados con empleados, permisos, vacaciones, asistencia y evaluaciones de desempeño.

2.4.1.5 React.js

React.js es una biblioteca de JavaScript utilizada para construir interfaces de usuario mediante componentes reutilizables. Permite dividir una aplicación en elementos independientes, como formularios, tablas, menús, paneles y ventanas, facilitando la actualización dinámica de la información sin recargar completamente la página. Dimopoulos et al. (2024) emplearon React.js en el frontend de una plataforma web interactiva y resaltaron su organización basada en componentes y el uso del DOM virtual para actualizar eficientemente los elementos de la interfaz. En un sistema de gestión de talento humano, React.js puede facilitar la creación de paneles administrativos, formularios de empleados, tablas de asistencia, solicitudes de permisos y visualización de reportes.

2.4.2 Tecnologías backend

2.4.2.1 Python

Azimov (2021) señala que Python es un lenguaje de programación de alto nivel utilizado en distintos ámbitos del desarrollo de software, incluyendo aplicaciones web, automatización, análisis de datos e inteligencia artificial. En el desarrollo web, Python puede utilizarse en la capa backend para implementar la lógica de negocio, procesar información, manejar solicitudes y conectarse con bases de datos. Su sintaxis clara y su ecosistema de librerías lo convierten en una alternativa ampliamente utilizada para construir aplicaciones escalables y mantenibles.

Torres et al. (2026) explican que, en sistemas web, Python puede trabajar junto con frameworks que simplifican el desarrollo de aplicaciones, permitiendo organizar rutas, modelos, vistas, formularios, autenticación y acceso a datos. Esta característica facilita la construcción de aplicaciones estructuradas, donde el backend se comunica con el frontend y con la base de datos para responder a las acciones del usuario.

2.4.2.2 Java

Java es un lenguaje de programación orientado a objetos utilizado para el desarrollo de aplicaciones multiplataforma, debido a que sus programas pueden ejecutarse en diferentes sistemas operativos mediante la máquina virtual de Java (JVM). Se caracteriza por proporcionar seguridad, modularidad, reutilización de código y facilidad de mantenimiento en sistemas empresariales que requieren escalabilidad y procesamiento eficiente. Según Truszkowski y Pańczyk (2022), Java junto con la plataforma J2EE fue aplicada en el desarrollo de un sistema de gestión de recursos humanos, permitiendo la implementación de módulos para la administración del personal, estructura organizacional, salarios, capacitación, evaluación del desempeño y control de usuarios. Esta aplicación demuestra que Java es una tecnología adecuada para

construir sistemas empresariales orientados a la gestión de información, automatización de procesos y administración de diferentes componentes dentro de una arquitectura organizada.

2.4.2.3 PHP

PHP es un lenguaje de programación utilizado principalmente para el desarrollo de aplicaciones web del lado del servidor, permitiendo la generación de contenido dinámico, procesamiento de solicitudes y conexión con bases de datos. Su facilidad de implementación y compatibilidad con diferentes tecnologías lo convierten en una alternativa ampliamente utilizada para la construcción de sistemas web. Según Truszkowski y Pańczyk (2022), PHP fue analizado como una tecnología empleada en el desarrollo de aplicaciones web, destacando sus características de implementación y rendimiento frente a otros lenguajes de programación como Java. En este sentido, PHP puede ser aplicado en sistemas de talento humano para gestionar información de empleados, procesar solicitudes, validar datos y administrar procesos mediante aplicaciones web conectadas a bases de datos.

2.4.2.4 Django

Torres et al. (2026) señalan que Django es un framework web basado en Python que permite desarrollar aplicaciones de manera estructurada, segura y rápida. Este framework proporciona componentes integrados para el manejo de rutas, modelos, plantillas, formularios, autenticación, sesiones y administración de datos. Además, ofrece aplicaciones reutilizables y herramientas que apoyan el desarrollo de sistemas web con mayor productividad y organización.

Karlsson et al. (2024) explican que Django se relaciona con buenas prácticas de desarrollo web porque permite organizar el código mediante una arquitectura basada en componentes, facilitando la separación entre la lógica, la presentación y el acceso a datos.

También puede integrarse con herramientas para construir API REST, gestionar autenticación, manejar seguridad y conectar aplicaciones con servicios externos.

2.4.2.5 Flask

Flask es un microframework backend basado en Python que proporciona los componentes esenciales para el desarrollo de aplicaciones y servicios web, permitiendo definir rutas, gestionar solicitudes HTTP, generar respuestas, utilizar plantillas y establecer conexiones con bases de datos. Su estructura ligera y flexible facilita la incorporación de extensiones según las necesidades específicas de cada proyecto. Grijalva et al. (2024) emplearon Flask en el desarrollo de una aplicación web orientada al procesamiento de información mediante una interfaz de usuario, donde los datos ingresados eran gestionados en el servidor y posteriormente presentados como resultados. En un sistema de gestión de talento humano, Flask puede utilizarse para implementar servicios relacionados con la consulta y administración de empleados, validación de información, generación de reportes e integración con otros sistemas institucionales.

2.4.2.6 FastAPI

FastAPI es un framework backend basado en Python orientado al desarrollo de interfaces de programación de aplicaciones (API), que permite construir servicios web mediante validación automática de datos utilizando anotaciones de tipos, soporte para operaciones asíncronas y generación automática de documentación mediante estándares como OpenAPI y JSON Schema. Schuster et al. (2023) emplearon FastAPI en el desarrollo de la arquitectura backend de la herramienta web Cortado, donde fue utilizado para integrar las funcionalidades implementadas en Python con el frontend mediante servicios web. En un sistema de gestión de talento humano, FastAPI puede utilizarse para desarrollar servicios relacionados con la administración de

empleados, autenticación de usuarios, control de asistencia, gestión de permisos y vacaciones, evaluación del desempeño y generación de reportes, facilitando la comunicación entre la interfaz de usuario y la lógica del sistema.

2.4.3 Comparación de tecnologías frontend y backend

Con el propósito de determinar las tecnologías más adecuadas para el desarrollo del sistema web de gestión de talento humano, se compararon las principales alternativas frontend y backend. Para ello, se consideraron criterios relacionados con la flexibilidad ante cambios, documentación, incorporación de observaciones del usuario, gestión del riesgo, tiempo de entrega, complejidad, tamaño del proyecto e iteratividad del desarrollo.

Tabla 1
Comparación de tecnologías frontend y backend

Capa y tecnología	Flexibilidad ante cambios	Documentación requerida	Incorporación de observaciones del usuario	Gestión del riesgo	Tiempo de entrega	Complejidad de implementación	Adaptabilidad al tamaño del proyecto	Iteratividad e incrementalidad
Frontend: HTML, CSS y JavaScript	Alta; permite modificar directamente la estructura, los estilos y las validaciones	Media; requiere documentar archivos, estilos y funciones	Alta; los cambios visuales pueden aplicarse rápidamente según las observaciones	Media-alta; utiliza tecnologías estandarizadas y ampliamente	Corto para interfaces administrativas	Baja-media	Alta para proyectos pequeños y medianos; adecuada para sistemas administrativos	Alta; permite desarrollar y mejorar las interfaces por módulos

				compatibles				
Frontend: TypeScript y React.js	Alta; la organización por componentes facilita los cambios	Alta; requiere documentar componentes, estados, propiedades y dependencias	Alta; facilita modificar componentes de acuerdo con la retroalimentación	Alta; TypeScript reduce errores mediante el tipado estático	Medio; requiere configuración inicial y manejo de dependencias	Media-alta	Alta para proyectos medianos y grandes con interfaces complejas	Alta; permite incorporar componentes y funciones progresivamente
Backend: Python con Django	Alta; su estructura modular permite modificar o agregar aplicaciones	Media; utiliza convenciones definidas para modelos, vistas, rutas y formularios	Alta; facilita incorporar nuevos requerimientos mediante módulos	Alta; incluye autenticación, permisos, validaciones y protecciones de seguridad	Corto-medio; dispone de componentes incorporados	Media	Alta para proyectos medianos y grandes con múltiples módulos	Alta; permite desarrollar aplicaciones y funcionalidades de forma progresiva
Backend: Python con Flask	Alta; permite definir libremente la	Alta; la arquitectura y extensiones	Alta; los servicios pueden modificarse según las	Media; varias funciones de seguridad	Corto en prototipos, pero aumenta en	Baja inicialmente y alta cuando el	Alta para proyectos pequeños; media para	Alta; facilita construir funcionalidades independientes

	estructura del sistema	deben documentarse detalladamente	necesidades identificadas	ad requieren extensiones y configuración	sistemas completos	sistema crece	sistemas grandes	
Backend: Python con FastAPI	Alta; permite agregar o modificar servicios API	Media; genera documentación automática de los servicios	Alta; las funcionalidades pueden ajustarse mediante nuevas rutas y servicios	Media-alta; incluye validación automática de datos, pero requiere configurar autenticación	Corto para API; medio para aplicaciones administrativas completas	Media	Alta para servicios, API y arquitecturas distribuidas	Alta; permite incorporar servicios de forma incremental

Nota. Autoría propia.

De acuerdo con la comparación, se utilizaron HTML, CSS y JavaScript para el frontend, debido a su integración directa, menor complejidad y facilidad para desarrollar formularios, tablas, paneles y validaciones. Para el backend se emplearon Python y Django, ya que proporcionan una estructura modular e incorporan herramientas para gestionar usuarios, permisos, formularios, modelos y acceso a datos. Esta combinación permitió desarrollar

progresivamente los módulos del sistema web de gestión de talento humano y adaptar sus funcionalidades a los requerimientos institucionales.

2.4.4 Bases de datos

La base de datos es el componente encargado de almacenar, organizar y relacionar la información que utiliza un sistema web. En una aplicación administrativa, este componente permite conservar registros de usuarios, procesos, configuraciones, reportes y demás datos necesarios para el funcionamiento del sistema. Khan et al. (2023) explican que la base de datos permite almacenar, organizar y relacionar la información utilizada por un sistema web.

El diseño de la base de datos debe garantizar que la información sea consistente, segura, disponible y fácil de consultar. Para ello, es necesario definir correctamente la estructura de los datos, las relaciones entre tablas, los tipos de información y los mecanismos de acceso. Khan et al. (2023) señalan que la selección del tipo de base de datos depende de la estructura de la información, el volumen de datos, la escalabilidad requerida y el nivel de consistencia esperado.

2.4.5 MySQL

Khan et al. (2023) señalan que MySQL es un sistema gestor de bases de datos relacional utilizado para almacenar y administrar información estructurada mediante tablas, relaciones, claves primarias y claves foráneas. En los sistemas web, permite organizar los registros, mantener la consistencia de la información y ejecutar consultas mediante el lenguaje SQL. En un sistema de gestión de talento humano, MySQL facilita el almacenamiento y la relación de datos correspondientes a empleados, cargos, departamentos, contratos, asistencia, permisos, vacaciones, evaluaciones de desempeño, novedades de nómina y demás procesos administrativos. De esta manera, contribuye a centralizar la información, reducir la duplicidad de registros y agilizar la consulta y actualización de los datos del personal.

2.4.6 PostgreSQL

PostgreSQL es un sistema gestor de bases de datos objeto-relacional de código abierto que permite almacenar y administrar información estructurada mediante tablas, relaciones, claves primarias, claves foráneas y restricciones de integridad. Utiliza el lenguaje SQL y ofrece funciones orientadas al manejo seguro de transacciones, consultas complejas y grandes volúmenes de información. Pipino et al. (2024) desarrollaron una arquitectura de microservicios para apoyar la digitalización de los procesos de recursos humanos utilizando PostgreSQL como tecnología de almacenamiento. El sistema permitió gestionar el proceso de contratación, registrar información personal y competencias de los empleados, administrar ofertas laborales y controlar actividades asignadas al personal. En un sistema de gestión de talento humano, PostgreSQL puede contribuir a centralizar los datos de los trabajadores, mantener la integridad de los registros y facilitar la consulta de información relacionada con contratos, cargos, competencias y actividades laborales.

2.4.7 MongoDB

MongoDB es un sistema gestor de bases de datos NoSQL orientado a documentos, en el cual la información se almacena mediante colecciones y documentos con una estructura similar al formato JSON. Su esquema flexible permite que los registros incorporen campos diferentes y se adapten a cambios en la estructura de los datos sin modificar un conjunto rígido de tablas. Zhao et al. (2024) desarrollaron una herramienta para identificar y proteger información personal almacenada en bases de datos NoSQL, la cual fue evaluada mediante una base correspondiente a un sistema de información de recursos humanos implementado con MongoDB. Los autores analizaron documentos que contenían información de empresas, gerentes y empleados, así como las relaciones existentes entre estos registros. En un sistema de gestión de talento humano,

MongoDB puede facilitar el almacenamiento de formularios, documentos, observaciones, perfiles, evidencias y otros datos semiestructurados relacionados con el personal.

2.4.8 MariaDB

MariaDB es un sistema gestor de bases de datos relacional de código abierto que permite organizar la información mediante tablas, relaciones, claves primarias y claves foráneas. Utiliza el lenguaje SQL y proporciona mecanismos para mantener la integridad, consistencia y seguridad de los registros almacenados. Baharuddin y Abdullah (2026) desarrollaron un sistema web de gestión de empleados utilizando PHP, HTML, CSS, JavaScript y MariaDB. La base de datos fue empleada para administrar el registro de empleados, la asistencia, la nómina, las solicitudes de permisos y los registros de auditoría. Además, el sistema incorporó control de acceso basado en roles para diferenciar las funciones de administradores, personal de recursos humanos y empleados. En un sistema de gestión de talento humano, MariaDB puede contribuir a centralizar la información del personal, reducir la duplicidad de registros y agilizar la consulta y actualización de datos administrativos.

2.4.9 Comparación de sistemas gestores de bases de datos

Con el propósito de identificar el sistema gestor de bases de datos más adecuado para la aplicación web de gestión de talento humano, se compararon MySQL, PostgreSQL, MongoDB y MariaDB. Para ello, se consideraron criterios relacionados con el modelo de almacenamiento, integridad de los datos, seguridad, escalabilidad, complejidad de implementación, compatibilidad con Django y adecuación a la información administrativa del personal.

Tabla 2
Comparación de sistemas gestores de bases de datos

Indicador	MySQL	PostgreSQL	MongoDB	MariaDB

Modelo de datos	Relacional	Objeto-relacional	NoSQL orientado a documentos	Relacional
Organización de la información	Tablas y relaciones	Tablas, relaciones y tipos avanzados	Colecciones y documentos JSON	Tablas y relaciones
Integridad referencial	Alta, mediante claves primarias y foráneas	Alta, con restricciones avanzadas	Depende del diseño de la aplicación	Alta, mediante claves primarias y foráneas
Consistencia de los datos	Alta	Alta	Flexible según la configuración	Alta
Manejo de transacciones	Sí	Sí, con funciones avanzadas	Sí	Sí
Flexibilidad del esquema	Media	Media-alta	Alta	Media
Información estructurada	Alta adecuación	Alta adecuación	Media	Alta adecuación
Información semiestructurada	Media, mediante JSON	Alta, mediante JSON y JSONB	Alta	Media, mediante JSON
Consultas y relaciones complejas	Alta	Muy alta	Requiere agregaciones y diseño documental	Alta
Compatibilidad con Django	Directa y ampliamente utilizada	Directa	Requiere conectores o herramientas adicionales	Compatible mediante el backend de MySQL

Seguridad y control de acceso	Incluye usuarios, roles y privilegios	Incluye roles y controles avanzados	Incluye autenticación y autorización	Incluye usuarios, roles y privilegios
Complejidad de implementación	Media-baja	Media-alta	Media	Media-baja
Facilidad de administración	Alta	Media	Media	Alta
Adaptabilidad a sistemas administrativos	Alta	Alta	Media	Alta
Adecuación para talento humano	Alta	Alta	Media para información estructurada	Alta

Nota. Autoría propia

De acuerdo con la comparación realizada, MySQL resultó adecuado para la aplicación web de gestión de talento humano, debido a que permite organizar información estructurada mediante tablas y relaciones, mantener la integridad de los registros y conectarse directamente con Django. PostgreSQL ofrece funciones avanzadas para consultas complejas; MongoDB presenta mayor flexibilidad para datos semiestructurados; y MariaDB posee características similares a MySQL. Sin embargo, MySQL permitió centralizar y relacionar eficientemente la información de empleados, cargos, contratos, asistencia, permisos, vacaciones, evaluaciones y demás procesos administrativos del sistema.

2.5 Metodologías de desarrollo de software

Kirpitsas y Pachidis (2022) señalan que las metodologías de desarrollo de software son enfoques organizados que permiten planificar, construir, probar y mantener sistemas

informáticos. Estas metodologías definen fases, roles, actividades, entregables y mecanismos de control para guiar el trabajo del equipo de desarrollo. Además, en la literatura reciente se reconoce la existencia de metodologías tradicionales, ágiles e híbridas, las cuales se diferencian por su nivel de planificación, flexibilidad, documentación y participación del usuario.

2.5.1 Metodologías tradicionales de desarrollo

Leong et al. (2023) señalan que las metodologías tradicionales de desarrollo de software se caracterizan por seguir una planificación estructurada, con fases definidas y una secuencia relativamente estable de actividades. En este grupo se encuentran modelos como cascada, espiral, RUP y prototipado. Estas metodologías suelen dar importancia a la documentación, la planificación previa y el control formal del avance del proyecto.

2.5.1.1 Metodología en cascada

La metodología en cascada es un modelo tradicional de desarrollo de software que organiza el proyecto mediante una secuencia ordenada de fases, en la que cada etapa debe completarse antes de iniciar la siguiente. Leong et al. (2023) señalan que este enfoque favorece una planificación estructurada, una documentación clara y un mayor control de las actividades y entregables del proyecto; sin embargo, presenta limitaciones cuando los requerimientos cambian durante el desarrollo, debido a que regresar a fases anteriores puede aumentar el tiempo y los costos. De acuerdo con los autores, la metodología en cascada comprende las etapas de análisis de requisitos, diseño del sistema, implementación, pruebas, despliegue y mantenimiento.

2.5.1.2 Metodología espiral

La metodología espiral es un modelo de desarrollo de software que organiza el proyecto en ciclos repetitivos, permitiendo la planificación, construcción y evaluación progresiva del sistema. Alazzawi et al. (2023) señalan que esta metodología combina elementos del desarrollo

secuencial con ciclos iterativos orientados a la gestión de riesgos. En cada ciclo se analizan los objetivos, las alternativas disponibles, los posibles riesgos, el desarrollo de la solución y su posterior evaluación. Este enfoque resulta especialmente útil en proyectos complejos donde es necesario revisar continuamente los riesgos antes de avanzar hacia nuevas fases del producto.

2.5.1.3 Proceso Unificado Racional

El Proceso Unificado Racional, conocido como RUP, es una metodología de desarrollo de software iterativa e incremental que organiza el proyecto mediante fases definidas, actividades, roles y productos de trabajo. Alazzawi et al. (2023) señalan que este enfoque permite gestionar los requerimientos, controlar los riesgos, definir la arquitectura y validar progresivamente el sistema durante su construcción. De acuerdo con los autores, RUP comprende cuatro fases principales: inicio, en la que se establecen el alcance y los objetivos del proyecto; elaboración, orientada al análisis de requerimientos y al diseño de la arquitectura; construcción, destinada al desarrollo y las pruebas del software; y transición, enfocada en la entrega, implementación y adaptación del sistema a los usuarios.

2.5.1.4 Metodología de prototipado

La metodología de prototipado es un enfoque de desarrollo de software que se basa en la creación de versiones preliminares del sistema con el propósito de comprender mejor los requerimientos y validar las funcionalidades antes de construir la solución definitiva. Alazzawi et al. (2023) señalan que esta metodología permite a los usuarios visualizar el funcionamiento del sistema desde etapas tempranas, identificar errores, proporcionar retroalimentación y proponer mejoras cuando las necesidades aún no están completamente definidas. Su proceso comprende la identificación de requerimientos, el diseño rápido, la construcción del prototipo, la evaluación

por parte de los usuarios, el refinamiento progresivo y, finalmente, el desarrollo de la versión final del sistema, contribuyendo a reducir errores y mejorar la calidad del producto desarrollado.

2.5.2 Metodologías ágiles

Las metodologías ágiles son enfoques de desarrollo de software orientados a la entrega progresiva de funcionalidades, la colaboración entre los involucrados y la adaptación continua a los cambios en los requerimientos. Saeedi y Visvizi (2021) señalan que estas metodologías surgieron como respuesta a las limitaciones de los modelos tradicionales, especialmente en proyectos donde las necesidades pueden modificarse durante el desarrollo. Su aplicación se basa generalmente en la planificación iterativa, la priorización de requerimientos, la construcción de incrementos funcionales, la revisión frecuente de resultados y la incorporación continua de retroalimentación. Entre los enfoques ágiles más conocidos se encuentran Scrum, Extreme Programming, Kanban, Crystal y Feature Driven Development.

2.5.2.1 Extreme Programming

Extreme Programming, conocido como XP, es una metodología ágil orientada a mejorar la calidad técnica del software y responder rápidamente a los cambios mediante prácticas de desarrollo continuo. Ghimire et al. (2022) señalan que XP busca mantener un producto funcional durante todo el proceso, promoviendo la colaboración del equipo, la retroalimentación frecuente y la mejora constante del código. Su aplicación comprende actividades como la planificación de pequeñas entregas, el diseño simple, la programación en pares, la integración continua, las pruebas frecuentes, la refactorización y la entrega progresiva de software funcional.

2.5.2.2 Kanban

Kanban es un enfoque ágil de gestión de proyectos que utiliza tableros visuales para organizar, controlar y dar seguimiento a las tareas durante el desarrollo de un sistema. Ozkan et

al. (2022) señalan que esta metodología permite visualizar las actividades pendientes, en proceso y finalizadas, con el propósito de mejorar el flujo de trabajo, limitar la cantidad de tareas ejecutadas simultáneamente y fomentar la mejora continua. Su funcionamiento comprende la visualización de las actividades, la definición de límites de trabajo en proceso, el seguimiento constante del flujo de tareas, la identificación de obstáculos, la entrega continua de resultados y la implementación de mejoras progresivas, sin depender necesariamente de ciclos de trabajo cerrados como los sprints.

2.5.2.3 Scrum

Scrum es un marco de trabajo ágil utilizado para la gestión y desarrollo de proyectos de software, basado en la colaboración de equipos multidisciplinarios y la entrega continua de valor mediante incrementos funcionales del producto. Hron y Obwegeser (2022) explican que esta metodología permite adaptarse rápidamente a los cambios y mejorar progresivamente el producto a través de ciclos cortos de trabajo denominados sprints. Cada sprint tiene una duración definida, generalmente entre una y cuatro semanas, durante la cual el equipo planifica, desarrolla, prueba y entrega un conjunto de funcionalidades previamente priorizadas. Asimismo, Alami y Krancher (2022) señalan que Scrum contribuye a la calidad del software mediante la transparencia, la inspección continua, la adaptación y la retroalimentación frecuente. Su funcionamiento comprende la gestión del Product Backlog, la planificación del sprint, la definición del Sprint Backlog, el desarrollo del incremento del producto, las reuniones diarias de seguimiento (Daily Scrum), la revisión de los resultados obtenidos (Sprint Review) y la retrospectiva (Sprint Retrospective), donde se identifican oportunidades de mejora para el siguiente ciclo de trabajo.

2.5.3 Comparación de metodologías de desarrollo de software

Una vez analizadas las metodologías tradicionales y ágiles, se realizó una comparación con el propósito de identificar el enfoque que presenta mayor correspondencia con las características del proyecto. Para ello, se consideraron criterios relacionados con la flexibilidad ante cambios, documentación, participación de los usuarios, gestión del riesgo, tiempo de entrega, complejidad de implementación, adaptabilidad al tamaño del proyecto e iteratividad.

Tabla 3

Comparación de metodologías de desarrollo de software

Criterio	Cascada	Espiral	RUP	Prototipado	XP	Kanban	Scrum
Flexibilidad ante cambios	Baja, porque sigue fases secuenciales	Alta, permite ajustes en cada ciclo	Media-alta, por su desarrollo iterativo	Alta, permite modificar el prototipo	Alta, responde rápidamente a los cambios	Alta, permite reorganizar las tareas	Alta, permite ajustar el backlog entre sprints
Nivel de documentación	Alto	Alto	Alto	Medio	Bajo-medio	Bajo-medio	Medio
Participación del usuario	Baja, principalmente al inicio y al final	Media, mediante evaluaciones periódicas	Media-alta	Alta, por la validación de prototipos	Alta, requiere retroalimentación frecuente	Media-alta	Alta, mediante revisiones de sprint
Gestión del riesgo	Baja	Muy alta, constituye su enfoque principal	Alta	Media	Media	Media	Alta, mediante revisiones e

							inspección frecuente
Tiempo de entrega de resultados	Largo, porque el producto se entrega al finalizar las fases	Medio-largo	Medio	Corto-medio	Corto, mediante entregas pequeñas	Continuo	Corto, mediante incrementos por sprint
Complejidad de implementación	Baja-media	Alta	Alta	Media	Media-alta	Baja-media	Media
Adaptabilidad a proyectos pequeños	Media	Baja	Baja-media	Alta	Alta	Alta	Alta
Adaptabilidad a proyectos medianos	Media	Alta	Alta	Alta	Alta	Alta	Alta
Adaptabilidad a proyectos grandes	Alta cuando los requisitos son estables	Alta	Alta	Media	Media	Media-alta	Alta, con una adecuada organización del equipo
Iteratividad e incrementalidad	Baja	Alta	Alta	Alta	Alta	Continua	Alta
Organización del trabajo	Fases consecutivas	Ciclos orientados al riesgo	Fases e iteraciones	Versiones preliminares sucesivas	Prácticas técnicas y entregas pequeñas	Flujo visual continuo	Sprints, backlog, revisión y

			controladas				retrospectiva
Adecuación al proyecto de talento humano	Media-baja	Media	Media-alta	Alta	Media-alta	Alta	Muy alta

Nota. Autoría propia.

De acuerdo con la comparación realizada, Scrum presenta una mayor adecuación para el desarrollo del sistema web de gestión de talento humano, debido a que permite organizar el trabajo mediante sprints, adaptar los requerimientos, entregar funcionalidades de manera incremental y validar periódicamente los resultados con los responsables institucionales. A diferencia de los modelos secuenciales, Scrum facilita la incorporación de observaciones durante el desarrollo y permite mejorar progresivamente los módulos de empleados, asistencia, permisos, vacaciones, evaluaciones, novedades de nómina, liquidaciones, activos y reportes. Por estas razones, se utilizará Scrum como marco de trabajo para organizar el desarrollo del sistema.

2.6 Vibe Coding

El vibe coding constituye una práctica emergente de desarrollo de software asistida por inteligencia artificial generativa, en la cual el desarrollador formula requerimientos, problemas o instrucciones mediante lenguaje natural para obtener sugerencias, explicaciones, modificaciones o fragmentos de código. Su fundamento se relaciona con la incorporación de modelos de lenguaje de gran escala en las distintas actividades de la ingeniería de software. Fan et al. (2023) señalan que estos modelos pueden apoyar tareas de análisis, codificación, pruebas, depuración, documentación, refactorización y mantenimiento, lo que amplía las posibilidades de automatización dentro del ciclo de vida del software.

Sin embargo, el vibe coding no debe comprenderse únicamente como un mecanismo para producir código con mayor rapidez. Desde una perspectiva de ingeniería de software, constituye una forma de interacción entre el desarrollador y una herramienta de inteligencia artificial generativa que requiere conocimientos técnicos, capacidad de análisis y criterios de validación. Hou et al. (2024) señalan que las soluciones generadas mediante inteligencia artificial pueden presentar errores lógicos, prácticas inadecuadas o vulnerabilidades, por lo que deben ser revisadas, probadas y adaptadas al contexto técnico antes de incorporarse a una aplicación. En consecuencia, su utilización en el desarrollo del sistema web debe asumirse como una práctica de apoyo y no como un reemplazo del razonamiento, la responsabilidad ni la toma de decisiones del equipo de desarrollo.

2.6.1 Aplicación del Vibe Coding en ingeniería de software

La aplicación del vibe coding en ingeniería de software comprende diferentes actividades del ciclo de desarrollo, desde la interpretación inicial de los requerimientos hasta la construcción, validación y mantenimiento de la solución. Los modelos de lenguaje pueden emplearse para proponer estructuras de clases, generar funciones, explicar fragmentos de código, identificar posibles errores, elaborar casos de prueba y apoyar la documentación técnica. Fan et al. (2023) sostienen que estas herramientas poseen potencial para intervenir en múltiples etapas del desarrollo, especialmente en actividades repetitivas o que requieren la exploración rápida de alternativas técnicas.

Su utilidad no se limita a la generación automática de código. También puede contribuir al análisis de requerimientos mediante la transformación de descripciones funcionales en propuestas técnicas preliminares, historias de usuario, criterios de aceptación o estructuras de datos. De esta manera, el desarrollador puede evaluar diferentes posibilidades antes de

seleccionar la alternativa que mejor se adapte a las características del sistema. No obstante, las propuestas generadas deben contrastarse con los requerimientos funcionales, las reglas del negocio y la arquitectura definida para el proyecto, debido a que los resultados producidos por los modelos de lenguaje requieren interpretación y adaptación técnica antes de ser incorporados al software (Hou, y otros, 2024).

En el contexto de la presente investigación, el vibe coding puede apoyar la construcción del sistema web mediante la generación de componentes y soluciones específicas. Sin embargo, su uso debe mantenerse alineado con la arquitectura y los objetivos del proyecto. Hou et al. (2024) señalan que el código generado por IA puede contener errores o vulnerabilidades, por lo que debe ser revisado, probado y adaptado antes de su incorporación. Por ello, esta práctica debe utilizarse como apoyo y no como sustituto del criterio técnico del desarrollador.

2.6.2 Implicaciones técnicas en el desarrollo web

En el desarrollo web, el vibe coding puede emplearse en la construcción de componentes de interfaz, formularios, rutas, controladores, modelos de datos, consultas, validaciones y mecanismos de autenticación. Fan et al. (2023) señalan que las herramientas de inteligencia artificial también pueden utilizarse para interpretar mensajes de error, proponer correcciones y generar documentación sobre el funcionamiento de los módulos, lo que permite agilizar determinadas tareas operativas dentro del proceso de desarrollo.

No obstante, una respuesta generada por inteligencia artificial puede ser sintácticamente correcta y, al mismo tiempo, resultar incompatible con la arquitectura, las versiones de las tecnologías utilizadas o las reglas funcionales de la aplicación. Hou et al. (2024) advierten que estas herramientas pueden producir funciones duplicadas, incorporar dependencias innecesarias o

generar código descontextualizado cuando no reciben información suficiente sobre el entorno y los requerimientos del sistema.

Por ello, el código generado debe respetar criterios de nomenclatura, modularidad, legibilidad, reutilización y mantenibilidad. Alenezi y Akour (2025)señalan que la supervisión humana y la validación técnica son necesarias para garantizar la calidad y seguridad del software generado con apoyo de inteligencia artificial. En el sistema web propuesto, esta práctica puede emplearse para apoyar tareas específicas; sin embargo, las decisiones relacionadas con la arquitectura, el tratamiento de los datos y la seguridad deben permanecer bajo la responsabilidad del desarrollador.

2.6.3 Impacto en la calidad y seguridad del software

El impacto del vibe coding en la calidad del software depende de la forma en que se utilicen y validen sus resultados. La inteligencia artificial puede mejorar la productividad, proponer refactorizaciones, identificar errores y generar pruebas iniciales; sin embargo, Tosi (2024) advierte que la rapidez en la producción de código no garantiza su corrección, confiabilidad ni integración adecuada con los demás módulos.

La seguridad representa un aspecto crítico, debido a que el código generado puede omitir validaciones, implementar controles de acceso insuficientes, gestionar incorrectamente credenciales o construir consultas vulnerables. Estas deficiencias pueden introducir riesgos en aplicaciones que procesan información de usuarios o se conectan con bases de datos y servicios externos Negri-Ribalta et al. (2024).

Perry et al. (2023) señalan que algunos usuarios de asistentes de inteligencia artificial pueden producir código menos seguro y, al mismo tiempo, confiar excesivamente en el resultado

generado. En consecuencia, deben aplicarse pruebas de autenticación, revisión de permisos, validación de entradas y análisis del manejo de errores, evitando también compartir contraseñas, tokens o datos sensibles con herramientas externas.

2.6.4 Supervisión humana y validación del código generado

La supervisión humana es fundamental en el uso del vibe coding, debido a que la responsabilidad sobre el funcionamiento del sistema permanece en el desarrollador. Aunque una herramienta generativa puede producir código, explicar errores o sugerir correcciones, Tosi (2024) sostiene que la intervención de expertos sigue siendo necesaria para verificar la corrección funcional y la calidad del resultado.

La validación debe incluir la revisión de la sintaxis, la lógica, las dependencias, la compatibilidad con la arquitectura y el cumplimiento de los estándares de programación. Posteriormente, el código debe probarse en un entorno controlado, considerando tanto los escenarios esperados como las entradas inválidas, los errores de conexión y los intentos de acceso no autorizado Negri-Ribalta et al. (2024).

Asimismo, los cambios realizados con apoyo de inteligencia artificial deben registrarse mediante mecanismos de control de versiones y trazabilidad. En la presente investigación, el código solo podrá incorporarse al sistema después de comprobar el cumplimiento de los requerimientos, la compatibilidad con la arquitectura y la aplicación de los controles de seguridad establecidos Alenezi (2025).

2.6.5 Integración de inteligencia artificial en metodologías ágiles

La integración del vibe coding en las metodologías ágiles puede fortalecer el desarrollo iterativo e incremental del software, al apoyar actividades como la organización del backlog, la

descomposición de historias de usuario, la estimación del esfuerzo y la asignación de tareas. Hoda et al. (2023) proponen el enfoque de “agilidad aumentada”, en el cual la inteligencia artificial complementa las prácticas ágiles sin reemplazar la intervención ni la capacidad de decisión del equipo de desarrollo.

Durante cada iteración, las herramientas generativas pueden utilizarse para analizar requerimientos, producir componentes preliminares, elaborar documentación, apoyar las pruebas y proponer mejoras. Sin embargo, su incorporación debe ajustarse a las rutinas, necesidades y restricciones de la organización, debido a que los beneficios de productividad dependen de una integración adecuada dentro del proceso de ingeniería de software Banh et al. (2025).

Diebold (2025) sostiene que la inteligencia artificial generativa puede transformar las funciones del propietario del producto, el desarrollador y el Scrum Master, al automatizar tareas repetitivas y permitir una mayor atención a las actividades estratégicas y creativas. Por ello, en cada iteración, el código generado debe revisarse de acuerdo con los criterios de aceptación, las pruebas funcionales y la retroalimentación de los usuarios, manteniendo la colaboración humana y el control de calidad como elementos centrales del proceso ágil.

CAPITULO **III**

3. MARCO METODOLÓGICO

3.1 Enfoque de la investigación

La investigación adoptó un enfoque mixto con predominio cualitativo. El componente cualitativo permitió comprender los procesos administrativos de Talento Humano a partir de las reuniones de trabajo con los informantes institucionales y del análisis de documentos, formularios y registros utilizados por la COAC Gañansol. Esta información permitió identificar

necesidades, reglas de negocio, campos, validaciones y requerimientos funcionales y no funcionales del sistema web.

El componente cuantitativo se incorporó durante la validación funcional del sistema mediante pruebas realizadas en un entorno local. En estas pruebas se verificó el cumplimiento de las funcionalidades desarrolladas, se registraron los resultados obtenidos y se identificaron los errores e incidencias presentados en cada módulo. Esta información permitió comprobar de manera objetiva el funcionamiento del sistema y determinar los ajustes necesarios.

La integración de ambos componentes permitió contrastar las necesidades expresadas por el personal de Talento Humano y Tecnologías de la Información con los resultados obtenidos durante las pruebas funcionales. Por tanto, el enfoque mixto resultó pertinente para comprender la problemática institucional y verificar progresivamente el funcionamiento de la solución tecnológica desarrollada.

3.2 Nivel de la investigación

La investigación corresponde a un nivel descriptivo, puesto que se orientará a observar, analizar y describir la situación actual de los procesos de gestión del talento humano en la COAC Gañansol, sin manipular las condiciones en las que estos se desarrollan. La descripción abarcará las actividades administrativas realizadas mediante herramientas ofimáticas y registros documentales físicos, las limitaciones existentes en la centralización, trazabilidad y disponibilidad de la información, y las dificultades derivadas del procesamiento manual de los datos de asistencia.

Este nivel descriptivo permitirá caracterizar las necesidades, restricciones y prioridades institucionales, información que constituirá la base para la construcción y validación progresiva del sistema web, en coherencia con el problema de investigación y con los objetivos planteados.

3.3 Población y muestra

La población de estudio estuvo constituida por el personal vinculado con los procesos administrativos y tecnológicos de la gestión de Talento Humano en la COAC Gañansol, es decir, por quienes intervienen en la definición, ejecución, supervisión o soporte de dichos procesos. De acuerdo con la estructura institucional de la cooperativa, estas responsabilidades recaen en dos cargos: la jefatura de Talento Humano y la jefatura de Tecnologías de la Información.

La muestra estuvo conformada por esos dos participantes (n = 2), seleccionados mediante un muestreo no probabilístico intencional por criterio, también denominado muestreo de informantes clave. El criterio de inclusión fue ejercer la responsabilidad directa sobre los procesos de Talento Humano o sobre la infraestructura tecnológica que los soporta; el criterio de exclusión fue no participar en la definición, aprobación o soporte de dichos procesos.

Dado que la cooperativa cuenta únicamente con un responsable por cada una de estas dos áreas, la muestra corresponde a la totalidad de los cargos que cumplen el criterio de inclusión. En consecuencia, no se trata de una muestra por conveniencia en sentido estricto, sino de un censo de los informantes clave de la población definida, por lo que no fue necesario calcular un tamaño muestral estadístico.

Este tamaño resultó suficiente para los fines del estudio por tres razones. En primer lugar, la investigación es de carácter cualitativo aplicado y su propósito fue obtener información especializada sobre procesos institucionales, no estimar parámetros ni contrastar hipótesis. En

segundo lugar, la unidad de análisis fueron los procesos de Talento Humano y sus documentos de respaldo, y no las percepciones individuales del personal, por lo que la representatividad se sustenta en el acceso privilegiado de los informantes a dicha información. En tercer lugar, la información se saturó en las reuniones realizadas, ya que en la segunda reunión no surgieron procesos, campos ni reglas de negocio distintos de los identificados previamente, y los requerimientos adicionales se originaron en las revisiones de los sprints y no en nuevos informantes.

Por tanto, los resultados no pretenden generalizarse estadísticamente al conjunto del personal de la COAC Gañansol ni a otras cooperativas, sino describir y automatizar los procesos de Talento Humano de la institución objeto de estudio. Esta limitación se declara de forma explícita en el apartado de limitaciones de la investigación.

3.4 Técnicas e instrumentos de recolección de información

La recolección de información se apoyó en dos técnicas principales: las reuniones de trabajo para el levantamiento de requerimientos y el análisis documental de los archivos institucionales. Estas técnicas permitieron conocer los procesos administrativos vigentes, identificar las actividades susceptibles de automatización y establecer los requerimientos funcionales y los módulos del sistema web.

Las reuniones de trabajo se desarrollaron como sesiones abiertas de levantamiento de requerimientos, sin un cuestionario estructurado y sin levantamiento de actas formales, por tratarse de un trabajo técnico realizado directamente con los dos responsables institucionales. El instrumento de registro de esta técnica fue el listado de requerimientos elaborado durante las sesiones, cuyo producto consolidado corresponde a los requerimientos funcionales y no

funcionales presentados en las Tablas 7 y 8 del Capítulo IV y a la matriz de trazabilidad de la Tabla 9.

Para el análisis documental se empleó una matriz elaborada por el autor, que consta en el Anexo A y que organiza cada archivo institucional según su código, denominación, tipo, elementos identificados y módulo del sistema al que dio origen. Los archivos analizados fueron los formatos codificados FO-TH-001 a FO-TH-017 entregados por la cooperativa, cuya entrega se evidencia en el Anexo B.

3.4.1 Reuniones de trabajo y levantamiento de requerimientos

Las reuniones de trabajo se realizaron con el jefe de Talento Humano y el jefe de Tecnologías de la información, con el propósito de recopilar los requerimientos necesarios para el desarrollo del sistema. En una primera reunión se identificaron los procesos institucionales de la gestión de talento humano, las necesidades de los usuarios, los módulos prioritarios y las funcionalidades esperadas.

En ambas sesiones el diálogo se organizó en torno a cuatro temas: los procesos actuales del área, los documentos y formatos utilizados, las dificultades identificadas y las funcionalidades esperadas del sistema. Los hallazgos y acuerdos se anotaron durante las sesiones y se trasladaron de forma directa al listado de requerimientos, que posteriormente se depuró y se estructuró en el Product Backlog del proyecto.

Posteriormente, se realizó una segunda reunión enfocada en la revisión documental de la información proporcionada por la institución, con el fin de analizar la estructura de los registros, los campos requeridos, las validaciones y los procesos que deberán implementarse en el sistema. Esta revisión servirá como base para el desarrollo de la plataforma.

Asimismo, en cada iteración de desarrollo se presentaron los avances del sistema para recibir observaciones y realizar los ajustes correspondientes. La información obtenida se registró y organizó como requerimientos funcionales y no funcionales, los cuales se incorporaron al Product Backlog del proyecto.

3.4.2 Análisis documental

El análisis documental se aplicará a los registros, formatos y documentos institucionales proporcionados por el jefe de la Gestión de Talento Humano. Cada documento será revisado para identificar su estructura, los campos utilizados, los datos obligatorios, las validaciones requeridas y las relaciones existentes entre diferentes procesos de Talento Humano.

Este análisis permitirá transformar la información contenida en los documentos institucionales en modelos de datos, formularios digitales y módulos del sistema web, garantizando que la solución responda a la forma real en que la cooperativa gestiona la información.

Para sistematizar esta revisión se empleó la matriz de análisis documental del Anexo A, en la cual cada uno de los diecisiete formatos institucionales (FO-TH-001 a FO-TH-017) se registró con su código, denominación, tipo de archivo, elementos identificados y módulo del sistema al que dio origen. La evidencia de la entrega de estos archivos por parte de la cooperativa se presenta en el Anexo B.

3.4.3 Validación de los instrumentos

Los instrumentos utilizados no corresponden a escalas ni a cuestionarios estandarizados, sino a herramientas de registro propias de un levantamiento de requerimientos de software: el listado de requerimientos derivado de las reuniones y la matriz de análisis documental. Por esta

razón no se aplicaron procedimientos de validación por juicio de expertos externos ni pruebas de confiabilidad estadística, ya que no se midieron variables ni se generaron puntuaciones susceptibles de análisis psicométrico.

En coherencia con el enfoque mixto con predominio cualitativo, la validez de la información se aseguró mediante cuatro procedimientos. El primero fue la validación de contenido por los propios participantes: al término de cada reunión, los requerimientos registrados se leyeron y confirmaron con el jefe de Talento Humano y el jefe de Tecnologías de la Información, quienes corrigieron y aprobaron su redacción antes de incorporarlos al Product Backlog.

El segundo procedimiento fue la correspondencia documental, dado que la matriz de análisis se construyó directamente sobre los campos, validaciones y reglas contenidos en los formatos oficiales de la cooperativa, lo que evitó introducir categorías ajenas a la institución. El tercero fue la triangulación de fuentes, mediante el contraste entre lo expresado en las reuniones, lo registrado en los documentos institucionales y lo verificado en las pruebas funcionales. El cuarto fue la validación progresiva en las revisiones de cada sprint, en las cuales los responsables institucionales comprobaron que las funcionalidades implementadas correspondieran a los requerimientos previamente registrados.

Se reconoce como limitación que los instrumentos no fueron revisados por un tercero externo a la cooperativa ni sometidos a una prueba piloto, y que las reuniones no contaron con un guion estructurado ni con actas formales, lo cual se justifica por el carácter aplicado del estudio, por el número reducido de informantes clave y porque la verificación final de los requerimientos se realizó sobre el sistema en funcionamiento y no sobre el instrumento.

3.5 Tratamiento y análisis de la información

La información obtenida durante las reuniones de trabajo y el análisis documental será organizada y procesada de acuerdo con los procesos de la Gestión de Talento Humano identificados. Los requerimientos serán clasificados según sus prioridades e incorporados al Product Backlog de Scrum, instrumento que orienta la planificación de los sprints de desarrollo.

Los campos, validaciones y relaciones identificados en las hojas de cálculo y formularios servirán como base para diseñar la estructura de la base de datos y construir los módulos del sistema. Posteriormente, los avances serán revisados con el personal responsable de la cooperativa, con el fin de realizar correcciones y comprobar que las funcionalidades respondan a las necesidades institucionales.

3.6 Proceso de desarrollo mediante Scrum y Vibe Coding

El desarrollo del sistema web se organizó mediante el marco de trabajo ágil Scrum, complementado con técnicas de Vibe Coding y herramientas de inteligencia artificial generativa. La aplicación de Scrum permitió distribuir el trabajo en iteraciones, priorizar los requerimientos, presentar avances funcionales y realizar ajustes progresivos de acuerdo con las observaciones de los responsables institucionales. El proyecto se desarrolló mediante seis sprints de cuatro semanas de duración cada uno, distribuidos a lo largo de los seis meses establecidos para la ejecución de la propuesta.

3.6.1 Roles de Scrum

Para la aplicación de Scrum se establecieron los roles de Product Owner, Scrum Master y desarrollador. La jefa de Talento Humano asumió el rol de Product Owner, debido a que proporcionó los requerimientos institucionales, priorizó las necesidades y verificó que las funcionalidades desarrolladas respondieran a los procesos administrativos del área.

El jefe de Tecnologías de la Información asumió el rol de Scrum Master, facilitó la aplicación de Scrum, acompañó el seguimiento de los sprints y apoyó la resolución de impedimentos técnicos. El autor se desempeñó como desarrollador y fue responsable del análisis, diseño, programación, integración, corrección de errores, pruebas funcionales y documentación del sistema. El personal de Talento Humano participó como fuente de información y usuario validador de los incrementos, sin asumir funciones de desarrollo.

3.6.2 Artefactos de Scrum

El Product Backlog se elaboró a partir de los requerimientos obtenidos durante las reuniones de trabajo y el análisis de los documentos institucionales. En este artefacto se registraron y priorizaron las funcionalidades relacionadas con la gestión de empleados, asistencia, permisos, vacaciones, evaluaciones de desempeño, novedades de nómina, liquidaciones, activos, pasantías y reportes administrativos.

Al inicio de cada sprint, los elementos prioritarios del Product Backlog se seleccionaron y organizaron en el Sprint Backlog, donde se establecieron las tareas técnicas, las funcionalidades por desarrollar, las correcciones pendientes y los criterios de comprobación. Al finalizar cada iteración se obtuvo un incremento funcional, compuesto por módulos que fueron ejecutados, revisados y comprobados en el entorno local.

3.6.3 Eventos y seguimiento de los sprints

La planificación del sprint se realizó al inicio de cada iteración, mediante la selección de los requerimientos que serían desarrollados, la definición de las tareas técnicas y el establecimiento de los resultados esperados. El seguimiento se efectuó mediante la revisión periódica del Sprint Backlog, el control de las actividades pendientes, la comprobación de las funcionalidades desarrolladas y el registro de los errores detectados. Debido a que el desarrollo

estuvo a cargo principalmente de un solo autor, este seguimiento se adaptó a revisiones individuales del avance.

Al finalizar cada iteración se realizó la revisión del sprint, en la cual se presentaron los incrementos funcionales a los responsables de Talento Humano y Tecnologías de la Información. Las observaciones recibidas permitieron corregir campos, formularios, cálculos, permisos, opciones de navegación y reglas de funcionamiento. Posteriormente, se efectuó la retrospectiva para analizar las dificultades, las soluciones aplicadas y las mejoras necesarias para el siguiente sprint.

3.6.4 Aplicación de Vibe Coding durante los sprints

Las técnicas de Vibe Coding se utilizaron durante la construcción de los módulos como apoyo para la generación y optimización de código, la interpretación de errores, la creación de consultas, la corrección de funcionalidades y la elaboración de documentación técnica. Para estas actividades se emplearon herramientas de inteligencia artificial generativa como ChatGPT, Gemini y GitHub Copilot.

El código generado o modificado mediante estas herramientas fue revisado, adaptado y comprobado por el autor antes de incorporarlo al sistema. La validación se realizó en el entorno local, considerando los requerimientos institucionales, la arquitectura del sistema, las reglas de negocio y los controles de seguridad. De esta manera, la inteligencia artificial se empleó como herramienta de apoyo, mientras que la responsabilidad sobre las decisiones técnicas y la calidad del código permaneció bajo supervisión humana.

3.6.5 Base tecnológica de partida y licenciamiento

El sistema no se construyó desde cero. Como base tecnológica de partida se utilizó Horilla, un sistema de gestión de talento humano de software libre desarrollado en Python y Django, distribuido bajo la licencia LGPL-2.1. De esta plataforma se reutilizaron la estructura base del proyecto Django, su configuración inicial, la organización en aplicaciones, el esquema de autenticación y los componentes comunes de la interfaz y de la administración del sistema.

Los módulos de gestión de Talento Humano que constituyen el objeto de esta propuesta no formaban parte de la plataforma base y fueron desarrollados por el autor, puesto que responden a los formatos y procedimientos propios de la cooperativa. Este desarrollo comprendió los modelos, vistas, plantillas y validaciones de los diecisiete formatos institucionales codificados FO-TH-001 a FO-TH-017, con sus campos, reglas de cálculo y flujos de aprobación; la generación e importación de los documentos institucionales en formato Excel y Word; la configuración de la autenticación contra el Active Directory de la cooperativa; la definición de los perfiles y permisos institucionales; y la ejecución de las pruebas funcionales descritas en el apartado 3.7.

El uso de esta base se declara de forma expresa por dos razones. La primera es de honestidad académica, ya que permite distinguir con claridad entre los componentes preexistentes y el aporte del autor. La segunda es de cumplimiento legal, puesto que la licencia LGPL-2.1 exige conservar los avisos de autoría y de licencia del software original y poner a disposición el código fuente de los componentes cubiertos que hayan sido modificados. En consecuencia, el código entregado a la institución incorpora los archivos de licencia y de atribución correspondientes, en los que se identifica el proyecto original, su licencia y los componentes modificados durante el desarrollo.

3.7 Validación funcional del sistema

La validación del sistema se realizó de manera progresiva mediante las revisiones de los sprints, en las cuales los responsables institucionales comprobaron que las funcionalidades desarrolladas correspondieran a los requerimientos establecidos. De forma complementaria, se ejecutaron pruebas funcionales sobre cada módulo en el entorno local del sistema web, considerando aspectos técnicos, funcionales y de seguridad, como la autenticación de usuarios, la validación de datos de entrada, los cálculos implementados, la navegación y el control de acceso a la información del personal.

En correspondencia con el enfoque mixto con predominio cualitativo, la calidad de la información se aseguró mediante el contraste entre los requerimientos expresados durante las reuniones de trabajo, el contenido de los documentos institucionales y los resultados obtenidos en las pruebas funcionales. Las observaciones realizadas por los responsables de Talento Humano y Tecnologías de la Información permitieron interpretar cualitativamente el funcionamiento del sistema, mientras que la verificación del cumplimiento de las funcionalidades permitió disponer de evidencias comprobables para realizar los ajustes necesarios.

Cabe precisar que esta validación se aplicó al sistema desarrollado y no a los instrumentos de recolección, cuya validez se abordó en el apartado 3.4.3. Las revisiones de sprint operaron, además, como mecanismo de verificación de la información levantada, puesto que cada funcionalidad presentada a los responsables institucionales permitió confirmar o corregir los requerimientos registrados durante las reuniones de trabajo y el análisis documental.

3.8 Consideraciones éticas

La investigación se desarrolló con la autorización de la COAC Gañansol y con la colaboración institucional necesaria para el levantamiento de información relacionada con los procesos del área de Talento Humano. Esta autorización estuvo respaldada por el convenio existente entre la Universidad Católica de Cuenca y la cooperativa, así como por el oficio dirigido al gerente de la institución, mediante el cual se solicitó formalmente su participación en el desarrollo del proyecto.

Para la construcción del sistema, la cooperativa proporcionó documentación institucional, registros administrativos, formularios y reportes relacionados con los procesos de Talento Humano. Esta información se utilizó exclusivamente con fines académicos y tecnológicos, garantizando su confidencialidad y evitando la divulgación de datos personales, financieros o institucionales. Su manejo se limitó a las actividades necesarias para analizar los procesos, definir los requerimientos y desarrollar las funcionalidades del sistema.

Asimismo, las herramientas de inteligencia artificial empleadas durante el desarrollo se utilizaron de manera responsable, sin incorporar información sensible o confidencial de la institución. El código y los contenidos generados mediante estas herramientas fueron revisados, adaptados y comprobados antes de integrarse al sistema, manteniendo la responsabilidad humana sobre las decisiones técnicas y la protección de la información.

3.9 Resultados

En esta sección se presentan los resultados obtenidos mediante las reuniones de trabajo realizadas con los responsables de Tecnologías de la Información y Talento Humano, así como a través del análisis de la documentación, los formularios, los registros administrativos y los formatos proporcionados por la institución. El levantamiento de información permitió conocer la

manera en que se gestionaban los procesos relacionados con el personal, identificar las principales dificultades operativas, definir los requerimientos funcionales y no funcionales y establecer los módulos necesarios para el desarrollo del sistema web. Estos resultados sirvieron como base para organizar las actividades mediante Scrum y construir progresivamente las funcionalidades en un entorno local de pruebas.

3.9.1 Resultados del levantamiento de requerimientos

Tabla 4
Resultados del levantamiento de requerimientos para el sistema web de gestión de Talento Humano

N.º	Aspecto analizado	Resultado obtenido
1	Autorización y colaboración institucional	La institución autorizó el desarrollo del proyecto y facilitó la participación del personal de Tecnologías de la Información y Talento Humano. Esta colaboración permitió conocer los procesos institucionales y obtener los requerimientos necesarios para el desarrollo del sistema web.
2	Situación de los procesos de Talento Humano	Se determinó que varios procesos administrativos se gestionaban mediante documentos, formularios y registros digitales independientes. Esta forma de trabajo dificultaba la centralización de la información, incrementaba el tiempo requerido para realizar las actividades y podía generar duplicidad o inconsistencias en los registros.

3	Información proporcionada	La institución proporcionó documentación administrativa, formularios, registros del personal y formatos utilizados en los procesos de Talento Humano. Esta información permitió identificar los campos requeridos, las reglas de cada procedimiento y la estructura que debía considerarse en el sistema.
4	Principales dificultades identificadas	Se identificaron dificultades relacionadas con la búsqueda y actualización de información, el seguimiento de trámites, la elaboración manual de reportes y la dispersión de los registros. También se evidenció la necesidad de mejorar el control de permisos, vacaciones, evaluaciones, novedades y demás actividades relacionadas con la administración del personal.
5	Procesos susceptibles de automatización	Se estableció que los procesos de registro de empleados, control de asistencia, permisos, vacaciones, evaluaciones de desempeño, novedades de nómina, liquidaciones pasantías y generación de reportes podían gestionarse mediante un sistema web.
6	Módulos requeridos	Se definieron módulos de reclutamiento, incorporación, pasantías, empleados y usuarios, gestión documental, asistencia y actividades laborales, permisos y vacaciones, nómina y liquidaciones, desempeño, capacitación, reportes y configuración de roles y permisos. Cargos y

		contratos se reservaron para futuras implementaciones; el control de activos no forma parte del alcance corregido.
7	Usuarios y niveles de acceso	Se identificó la necesidad de establecer perfiles de usuario de acuerdo con las responsabilidades institucionales. Los permisos del sistema debían limitar el acceso a la información y a las acciones disponibles según el rol asignado.
8	Requerimientos funcionales	Se establecieron requerimientos relacionados con el registro, consulta, actualización y seguimiento de información; la validación de datos; la gestión de formularios; la generación de reportes; la búsqueda mediante diferentes criterios y la conservación del historial de los procesos realizados.
9	Requerimientos no funcionales	Se determinó que el sistema debía presentar una interfaz comprensible, mantener una estructura organizada, controlar el acceso mediante autenticación, proteger la información institucional y permitir futuras modificaciones o incorporaciones de funcionalidades.
10	Aplicación de Scrum y Vibe Coding	Los requerimientos fueron organizados y priorizados para su desarrollo mediante sprints. Las técnicas de Vibe Coding se utilizaron como apoyo en la generación y revisión de código, la corrección de errores, la adaptación de funcionalidades y la documentación del

		sistema. El contenido generado con inteligencia artificial fue revisado antes de su incorporación.
11	Validación de avances	Durante el desarrollo se presentaron versiones progresivas del sistema a los responsables institucionales. La retroalimentación obtenida permitió corregir campos, ajustar formularios, modificar cálculos, reorganizar opciones de navegación y adaptar las funcionalidades a los procedimientos identificados.
12	Pruebas funcionales	Las pruebas realizadas en el entorno local permitieron comprobar el registro, consulta, modificación y visualización de la información. También se verificaron las validaciones de los formularios, los cálculos, la navegación entre módulos, los permisos de usuario y la generación de reportes.

Nota. Autoría propia.

3.9.2 Análisis general

El levantamiento de requerimientos evidenció la necesidad de disponer de un sistema web que centralizara y organizara la información del área de Talento Humano. El análisis de los documentos institucionales permitió definir los módulos, campos, perfiles de usuario, validaciones y funcionalidades necesarias para gestionar procesos relacionados con empleados, asistencia, permisos, vacaciones, evaluaciones, nómina, liquidaciones, activos, pasantías y reportes. La participación de los responsables de Tecnologías de la Información y Talento Humano permitió revisar y ajustar progresivamente el sistema de acuerdo con las necesidades

identificadas, mientras que la aplicación de Scrum y el apoyo de técnicas de Vibe Coding facilitaron la organización de las actividades, la corrección de errores y la mejora de las funcionalidades desarrolladas.

Asimismo, las pruebas funcionales realizadas en el entorno local permitieron comprobar el funcionamiento de los módulos, las validaciones, los cálculos, la navegación y los controles de acceso, evidenciando que el sistema facilitó la centralización y consulta de la información y mejoró el seguimiento de los procesos administrativos de Talento Humano.

CAPÍTULO IV

4. PROPUESTA

4.1 Título de la Propuesta

"DESARROLLO DE UN SISTEMA WEB PARA LA GESTIÓN DE TALENTO HUMANO EN LA COAC GAÑANSOL MEDIANTE SCRUM Y TÉCNICAS DE VIBE CODING"

4.2 Presentación

En el presente capítulo se describe la propuesta tecnológica desarrollada para mejorar la gestión de los procesos administrativos de Talento Humano en la Cooperativa de Ahorro y Crédito Gañansol. La solución consistió en la construcción de un sistema web que permitió centralizar, registrar, consultar, actualizar y controlar la información relacionada con empleados, asistencia, permisos, vacaciones, evaluaciones de desempeño, novedades de nómina, liquidaciones, pasantías y reportes administrativos.

El desarrollo del sistema se organizó mediante el marco de trabajo ágil Scrum, distribuyendo las actividades en seis sprints de cuatro semanas cada uno. Esta organización

permitió construir progresivamente las funcionalidades, presentar avances a los responsables institucionales y efectuar correcciones de acuerdo con las observaciones recibidas durante las revisiones de cada sprint.

La propuesta se desarrolló mediante Python 3.11 y el framework Django 4.2 (LTS) para la implementación de la lógica del sistema; HTML5, CSS3 y JavaScript (ES6) para la construcción de las interfaces del cliente; y MySQL 8.0 como sistema gestor de base de datos relacional para el almacenamiento estructurado de la información. Adicionalmente, se emplearon utilidades del ecosistema de Django como el ORM integrado, el sistema de migraciones, el motor de plantillas y el módulo de autenticación. También se utilizaron técnicas de Vibe Coding mediante herramientas de inteligencia artificial generativa, como ChatGPT, Gemini y GitHub Copilot, para apoyar la interpretación de errores, generación de componentes, revisión de código, elaboración de consultas, documentación y mejora de determinadas funcionalidades.

El sistema fue ejecutado y validado en un entorno local de pruebas. Por tanto, el alcance de la propuesta comprendió el análisis de requerimientos, diseño, desarrollo, integración y comprobación funcional de los módulos, sin incluir su implementación definitiva en la infraestructura tecnológica de la cooperativa.

4.3 Justificación

La gestión administrativa del personal en instituciones cooperativas requiere centralización, trazabilidad y control de acceso a la información. Actualmente, la COAC Gañansol administra los procesos de Talento Humano mediante hojas de cálculo, formularios físicos y registros digitales dispersos, lo que ocasiona duplicidad, inconsistencias, demoras y riesgo de pérdida de datos. La propuesta se orientó a:

- Centralizar en una sola plataforma los procesos de empleados, asistencia, permisos, vacaciones, evaluaciones, nómina, liquidaciones y pasantías.
- Reducir el tiempo de búsqueda y actualización de la información del personal.
- Mejorar la trazabilidad de los registros mediante auditoría por usuario, rol y fecha.
- Proteger la información laboral aplicando autenticación, roles y permisos.
- Estandarizar los procedimientos administrativos y disminuir la dependencia de trámites manuales.
- Generar reportes con criterios de búsqueda para el seguimiento institucional.

4.4 Descripción de la propuesta

La propuesta consistió en el desarrollo de un sistema web administrativo para gestionar la información del talento humano de la COAC Gañansol. El sistema fue diseñado para funcionar mediante un navegador web y proporcionar acceso a sus funcionalidades de acuerdo con el perfil asignado a cada usuario.

La solución incorporó un panel principal desde el cual se accede a los diferentes módulos administrativos. Cada módulo permitió realizar operaciones de registro, consulta, modificación, seguimiento y generación de información, de acuerdo con las reglas y requerimientos identificados durante las reuniones de trabajo y el análisis documental.

Los módulos desarrollados fueron los siguientes:

- Gestión de reclutamiento y entrevistas. Incluye el FO-TH-015, Ficha de entrevista.
- Gestión de incorporación, inducción y reinducción. Incluye el FO-TH-004, Formato de inducción y reinducción.
- Gestión de pasantías. Incluye el FO-TH-014, Control de pasantías.
- Gestión de empleados, usuarios y cuentas institucionales. Comprende el FO-TH-001, Base de datos del personal; el FO-TH-002, Creación y administración de usuarios; y la

integración con Active Directory, mediante la cual se visualizan y vinculan las cuentas institucionales con los empleados y usuarios del sistema.

- Gestión documental. Incluye el FO-TH-005, Acta de entrega y recepción de puesto. La gestión de cargos y contratos queda contemplada para futuras implementaciones del sistema.
- Gestión de asistencia y actividades laborales. Comprende la gestión de registros de asistencia y del formulario FO-TH-016. En cuanto a los dispositivos biométricos, el alcance del proyecto se limitó a la verificación de conectividad e integración preliminar para futuras funcionalidades de administración de usuarios y sincronización de asistencia.
- Gestión de permisos y vacaciones. Incluye el FO-TH-008, Control de vacaciones; el FO-TH-010, Solicitud de permiso; el FO-TH-011, Solicitud de vacaciones; y el FO-TH-013, Liquidación de vacaciones.
- Gestión de nómina y liquidaciones. Comprende el FO-TH-003, Acumulación y mensualización de décimos; el FO-TH-009, Reporte de novedades de Talento Humano; y el FO-TH-012, Liquidación de haberes.
- Evaluación del desempeño laboral. Incluye el FO-TH-006, Evaluación en período de prueba, y una implementación inicial del FO-TH-007, Matrices de evaluación del desempeño, limitada a las funcionalidades básicas contempladas en el alcance del proyecto. Las funcionalidades adicionales del FO-TH-007 quedan previstas para futuras implementaciones.
- Gestión de capacitación. Incluye el FO-TH-017, Plan anual de capacitación.
- Generación de reportes administrativos. Permite obtener documentos y reportes relacionados con los diferentes procesos de Talento Humano.

- Configuración del sistema**.** Comprende la asignación de roles y permisos a los usuarios del sistema. Además, se mantiene disponible la opción de configuración del servidor de correo para una futura configuración y utilización, sin que esta funcionalidad haya sido objeto de pruebas dentro del alcance del proyecto.

4.5 Desarrollo de la propuesta

El desarrollo del sistema se realizó durante seis meses, organizados en seis sprints de cuatro semanas. Cada sprint comprendió las ceremonias Scrum de planificación (Sprint Planning), reunión diaria (Daily Scrum), revisión (Sprint Review) y retrospectiva (Sprint Retrospective), además de actividades de análisis, diseño, programación, pruebas y corrección. Al finalizar cada iteración se obtuvo un incremento funcional que fue presentado a los responsables de Talento Humano y Tecnologías de la Información.

4.5.1 Organización del desarrollo mediante Scrum

Para el desarrollo del proyecto se definieron de forma consistente los roles de Product Owner, Scrum Master y desarrollador. La jefa de Talento Humano desempeñó el rol de Product Owner, proporcionó y priorizó los requerimientos y validó los incrementos. El jefe de Tecnologías de la Información asumió el rol de Scrum Master, facilitó el marco de trabajo Scrum, acompañó el seguimiento de los sprints y apoyó la resolución de impedimentos técnicos.

El desarrollador fue responsable del análisis, diseño, programación, integración, pruebas y documentación del sistema. Los asistentes de Talento Humano participaron como usuarios validadores y aportaron conocimientos sobre los procesos institucionales durante la revisión de los módulos; por tanto, no se les asignaron responsabilidades de programación ni el rol de Scrum Master.

La planificación general de los sprints se presenta en la siguiente tabla.

Tabla 5
Planificación general de los sprints

Sprint	Duración	Funcionalidades principales	Incremento obtenido
Sprint 1	Cuatro semanas	Configuración del proyecto; FO-TH-001, base de datos del personal; FO-TH-002, usuarios; autenticación, roles, permisos y estructura inicial de la base de datos	Base funcional y segura del sistema
Sprint 2	Cuatro semanas	Reclutamiento y FO-TH-015; incorporación y FO-TH-004; empleados y gestión documental del FO-TH-005. Cargos y contratos: futuras implementaciones.	Módulos de incorporación y administración central del personal
Sprint 3	Cuatro semanas	Gestión de asistencia; FO-TH-008, FO-TH-010, FO-TH-011, FO-TH-013 y FO-TH-016; prueba de conectividad con dispositivos biométricos	Módulos para la gestión de asistencia, permisos, vacaciones y actividades virtuales, junto con la validación de conectividad con dispositivos biométricos
Sprint 4	Cuatro semanas	FO-TH-003, FO-TH-006, implementación inicial del FO-TH-007, FO-TH-009 y FO-TH-012	Evaluación FO-TH-006 e implementación básica inicial del FO-TH-007; nómina y liquidaciones. Las funciones adicionales del FO-TH-007 quedan para futuras implementaciones.
Sprint 5	Cuatro semanas	FO-TH-014, pasantías; FO-TH-017, capacitación; y reportes administrativos	Módulos complementarios y generación de información

Sprint 6	Cuatro semanas	Visualización y vinculación de cuentas de Active Directory; roles, permisos, corrección de errores, pruebas y documentación. Configuración de correo disponible, sin pruebas dentro del proyecto.	Versión funcional consolidada y documentada en entorno local

Nota. Autoría propia.

4.5.2 Definición del Product Backlog

El Product Backlog se elaboró con base en las necesidades identificadas durante las reuniones de trabajo y la revisión de los documentos institucionales. Las 21 historias de usuario representan los módulos efectivamente desarrollados: reclutamiento y entrevistas; incorporación, inducción y reinducción; empleados, usuarios y cuentas institucionales; cargos, contratos y entrega de puestos; asistencia y actividades bajo modalidad virtual; permisos y vacaciones; nómina y liquidaciones; evaluación del desempeño; pasantías; capacitación; activos; reportes; configuración general; dispositivos biométricos; y Active Directory. Cada historia identifica la funcionalidad, el actor, el criterio de aceptación y los formatos FO-TH relacionados. La prioridad se asignó conjuntamente con la Product Owner mediante cuatro criterios cualitativos: impacto en la continuidad de los procesos de Talento Humano, dependencia técnica respecto de otras funcionalidades, necesidad de control o seguridad y urgencia institucional. Se asignó prioridad alta a las funciones indispensables o requeridas como base de otros módulos; media-alta a los procesos importantes que dependían de la estructura central; media a los módulos complementarios que podían implementarse después de las funciones críticas; y baja a las mejoras que podían aplazarse sin afectar el incremento comprometido. No se incorporaron

elementos de prioridad baja en los seis sprints, porque quedaron fuera del alcance de la versión funcional. Requerimientos funcionales

Los requerimientos funcionales definieron las operaciones que debía realizar el sistema para responder a los procesos administrativos de Talento Humano. Su redacción mantiene correspondencia con las historias de usuario, los módulos implementados y los formatos institucionales FO-TH-001 a FO-TH-017.

Tabla 7
Requerimientos funcionales del sistema

ID	Requerimiento funcional
RF01	El sistema permitirá autenticar a los usuarios mediante credenciales de acceso.
RF02	El sistema permitirá crear, consultar, modificar y desactivar usuarios mediante el FO-TH-002, Creación y administración de usuarios.
RF03	El sistema permitirá asignar roles y permisos de acuerdo con las responsabilidades institucionales.
RF04	El sistema permitirá registrar, consultar y actualizar la información personal y laboral de los empleados mediante el FO-TH-001, Base de datos del personal.
RF05	La gestión de cargos queda prevista para futuras implementaciones y no forma parte del alcance desarrollado.
RF06	El sistema permitirá gestionar registros de asistencia. La integración biométrica se limita a la verificación de conectividad e integración preliminar; la administración de usuarios del dispositivo y la sincronización de asistencia quedan para futuras funcionalidades.

RF07	El sistema permitirá registrar, consultar, aprobar o rechazar solicitudes de permiso mediante el FO-TH-010, Solicitud de permiso.
RF08	El sistema permitirá controlar saldos y períodos mediante el FO-TH-008 y registrar, consultar, aprobar o rechazar solicitudes mediante el FO-TH-011, Solicitud de vacaciones.
RF09	El sistema permitirá gestionar la evaluación en período de prueba FO-TH-006 y las funcionalidades básicas de la implementación inicial del FO-TH-007. Las funcionalidades adicionales de las matrices de desempeño quedan previstas para futuras implementaciones.
RF10	El sistema permitirá gestionar la acumulación o mensualización de décimos y las novedades de Talento Humano mediante los FO-TH-003 y FO-TH-009.
RF11	El sistema permitirá registrar, calcular, consultar y generar liquidaciones de haberes y vacaciones mediante los FO-TH-012 y FO-TH-013.
RF12	Fuera del alcance corregido: el control de activos institucionales no se incluye entre los módulos desarrollados. RF12 se conserva únicamente para mantener la referencia de la planificación inicial.
RF13	El sistema permitirá registrar, consultar y actualizar pasantías mediante el FO-TH-014, Control de pasantías.
RF14	El sistema permitirá buscar y filtrar información mediante diferentes criterios.
RF15	El sistema permitirá generar reportes administrativos de los procesos gestionados.
RF16	El sistema conservará la información y el historial necesario para el seguimiento de los procesos.

RF17	El sistema permitirá crear, consultar, editar, eliminar y generar fichas de entrevista mediante el FO-TH-015, Ficha de entrevista, en el módulo de Reclutamiento.
RF18	El sistema permitirá registrar, consultar, editar, eliminar e importar documentos mediante el FO-TH-004, Formato de inducción y reinducción, en el módulo de Onboarding.
RF19	El sistema permitirá visualizar las cuentas institucionales de Active Directory y vincularlas con los empleados y usuarios del sistema, de acuerdo con los permisos asignados.
RF20	El sistema permitirá registrar, consultar, editar, eliminar, importar y generar informes de actividades bajo modalidad virtual mediante el formato FO-TH-016, de acuerdo con los permisos asignados.
RF21	Requerimiento futuro: la gestión de contratos vinculados con los empleados queda contemplada para futuras implementaciones del sistema.
RF22	El sistema permitirá registrar, consultar, editar, importar y generar actas de entrega y recepción de puesto mediante el FO-TH-005, como parte de la gestión documental; no incluye la gestión de cargos y contratos.
RF23	El sistema permitirá registrar, consultar, editar y dar seguimiento al plan anual de capacitación mediante el formato FO-TH-017.

Nota. Autoría propia. Se conservan los identificadores de la planificación inicial para mantener la trazabilidad. RF05 y RF21 corresponden a futuras implementaciones; RF12 queda fuera del alcance corregido. RF06 y RF09 se limitan a la conectividad biométrica preliminar y a las funciones básicas del FO-TH-007, respectivamente. La opción de correo está disponible, sin configuración ni pruebas dentro del proyecto.

Para conservar la correspondencia con los casos de uso, el RF07 se desagregó en UC12, Registrar solicitud de permiso, y UC13, Aprobar o rechazar permiso. De igual manera, el RF08

se representó mediante UC14, Registrar solicitud de vacaciones, y UC15, Aprobar o rechazar vacaciones. El RF14 corresponde a una funcionalidad transversal de búsqueda y filtrado, aplicada en los listados y consultas de empleados, asistencia, permisos, vacaciones, evaluaciones, nómina, liquidaciones, activos, pasantías y reportes; por esta razón, no pertenece exclusivamente a un solo módulo.

4.5.2.1 Requerimientos no funcionales

Los requerimientos no funcionales establecieron condiciones observables y verificables aplicables transversalmente a todos los módulos y formatos del sistema. Por esta razón, no se definió un requisito no funcional diferente para cada FO-TH; en su lugar, se especificaron condiciones comunes de seguridad, control de acceso, usabilidad, integridad, trazabilidad, protección de credenciales, estabilidad de ejecución y compatibilidad. Su cumplimiento puede comprobarse mediante pruebas de acceso, validación de formularios, revisión de datos y ejecución en los navegadores del entorno de prueba.

Tabla 8
Requerimientos no funcionales del sistema

ID	Requerimiento no funcional verificable
RNF01	El 100 % de las pantallas principales deberá mantener navegación, etiquetas y acciones consistentes; los campos obligatorios deberán identificarse visualmente y las vistas no deberán presentar desbordamiento horizontal entre 1366 × 768 y 1920 × 1080 píxeles.
RNF02	El sistema deberá exigir autenticación para las funciones restringidas y denegar o redirigir todos los intentos de acceso directo a rutas para las que el usuario no posea el permiso requerido.
RNF03	Todos los formularios deberán validar en el servidor los campos obligatorios, tipos, formatos, rangos y relaciones definidos; ante un dato inválido, el registro no se almacenará y se identificará el campo que debe corregirse.

ID	Requerimiento no funcional verificable
RNF04	La base de datos deberá mantener claves primarias, claves foráneas, restricciones de unicidad y transacciones en las operaciones que afecten varios registros, de modo que no se generen relaciones huérfanas ni duplicados definidos por las reglas del negocio.
RNF05	Los errores de validación, autorización o conexión deberán mostrar un mensaje contextual que indique la operación no completada y la acción de corrección, sin exponer trazas técnicas, contraseñas ni parámetros de conexión.
RNF06	Las operaciones de creación, modificación, aprobación, rechazo y eliminación deberán conservar, cuando corresponda, el usuario responsable, la fecha y el estado del registro, de manera que los cambios puedan identificarse durante las pruebas y consultas administrativas.
RNF07	Las credenciales y los parámetros sensibles deberán mantenerse fuera de los archivos versionados, no deberán mostrarse en mensajes de error ni registrarse en texto legible y solo podrán ser utilizados por los componentes autorizados del sistema.
RNF08	En el entorno local de prueba, la autenticación, la consulta de listados y las operaciones CRUD comunes deberán ejecutarse de forma continua con el conjunto de datos de prueba, sin bloqueos de la interfaz ni interrupciones de la sesión. No se establecen umbrales de tiempo de respuesta, debido a que en esta versión del sistema no se realizaron mediciones formales de rendimiento.
RNF09	Las contraseñas locales deberán almacenarse mediante PBKDF2 con SHA-256; todas las solicitudes de modificación deberán validar el token CSRF y el 100 % de las pruebas de rutas sin sesión o permiso deberá terminar en rechazo o redirección segura.
RNF10	Las funciones principales deberán ejecutarse sin pérdida de contenido ni acciones en las versiones de Google Chrome y Microsoft Edge instaladas en el entorno de prueba, con una resolución mínima de 1366 × 768 píxeles.

Nota. Autoría propia.

4.5.2.2 Casos de uso del sistema

Los casos de uso se definieron después de establecer los requerimientos funcionales RF01-RF23 y las historias de usuario HU01-HU21 del Product Backlog. Cada caso representa un objetivo verificable y mantiene trazabilidad con el requerimiento que lo origina, la historia de usuario, la funcionalidad implementada, el sprint de desarrollo y el caso de prueba correspondiente.

Se identificaron cinco actores humanos: Empleado, responsable de Talento Humano, Administrador del sistema, jefe departamental y Gerencia. También intervienen dos actores externos: el servidor de Active Directory, que valida y administra las credenciales institucionales, y el reloj biométrico, que proporciona los registros de marcación. Los candidatos y pasantes no se modelaron como actores directos porque sus datos son registrados por el responsable de Talento Humano.

Para asignar una granularidad adecuada se separaron los procesos que poseen objetivos, actores o resultados diferentes: permisos y vacaciones; novedades de nómina y liquidaciones; activos y pasantías. Asimismo, Reclutamiento y Onboarding se incorporaron como casos independientes porque corresponden a módulos implementados con flujos propios.

El catálogo resultante comprende 24 casos de uso: autenticación (UC01); usuarios, roles y permisos (UC02); departamentos y cargos (UC03); empleados (UC04); contratos (UC05); reclutamiento (UC06); onboarding (UC07); importación biométrica (UC08); asistencia manual (UC09); consulta de asistencia (UC10); actividades virtuales (UC11); solicitud y resolución de permisos (UC12-UC13); solicitud y resolución de vacaciones (UC14-UC15); evaluación de desempeño (UC16); novedades de nómina (UC17); liquidaciones (UC18); activos (UC19);

pasantías (UC20); reportes (UC21); cuentas de Active Directory (UC22); actas de entrega y recepción del puesto (UC23); y plan anual de capacitación (UC24).

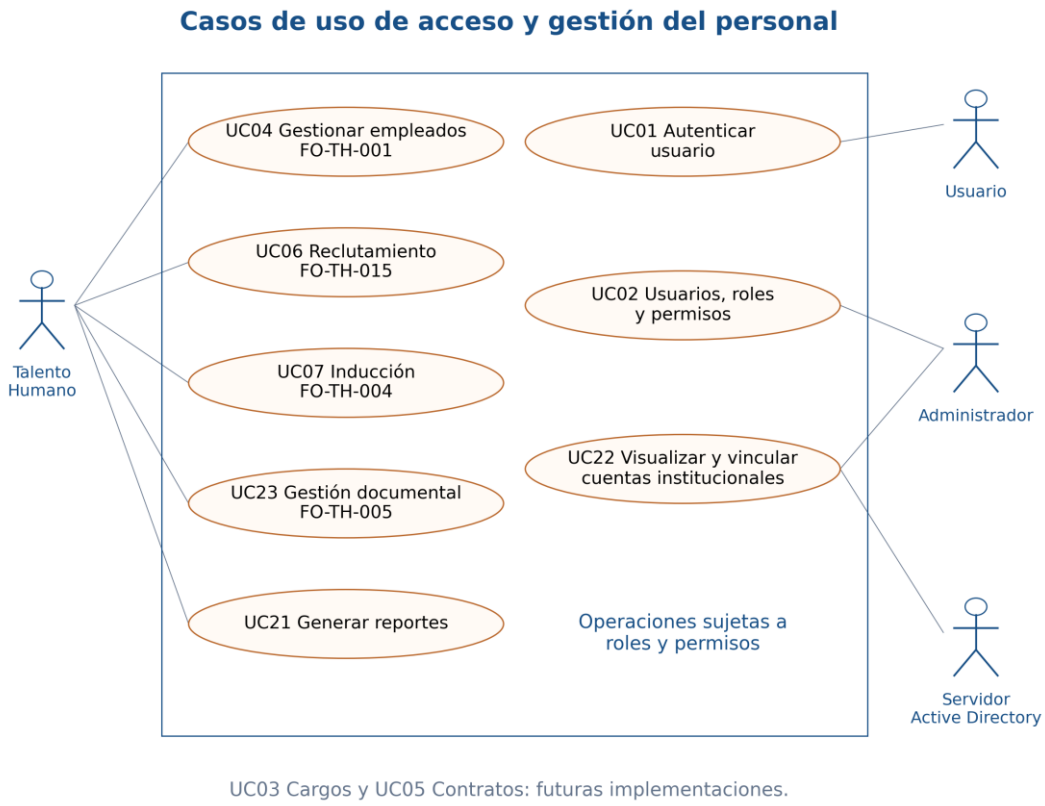


Figura 1. Casos de uso de acceso, empleados, reclutamiento, incorporación, gestión documental y reportes.

Nota. Autoría propia. Cargos y contratos se reservan para futuras implementaciones. La integración con Active Directory se representa mediante la visualización y vinculación de cuentas institucionales.

La Figura 2 presenta por separado los procesos operativos de asistencia y administración del personal. Las relaciones «include» se utilizaron únicamente cuando un caso requiere ejecutar obligatoriamente una función reutilizable, como validar el archivo biométrico, asociar la marcación con el empleado o calcular el saldo de vacaciones. Las aprobaciones se modelaron

como casos independientes, y no como «extend», porque interviene otro actor y se obtiene un resultado administrativo propio.

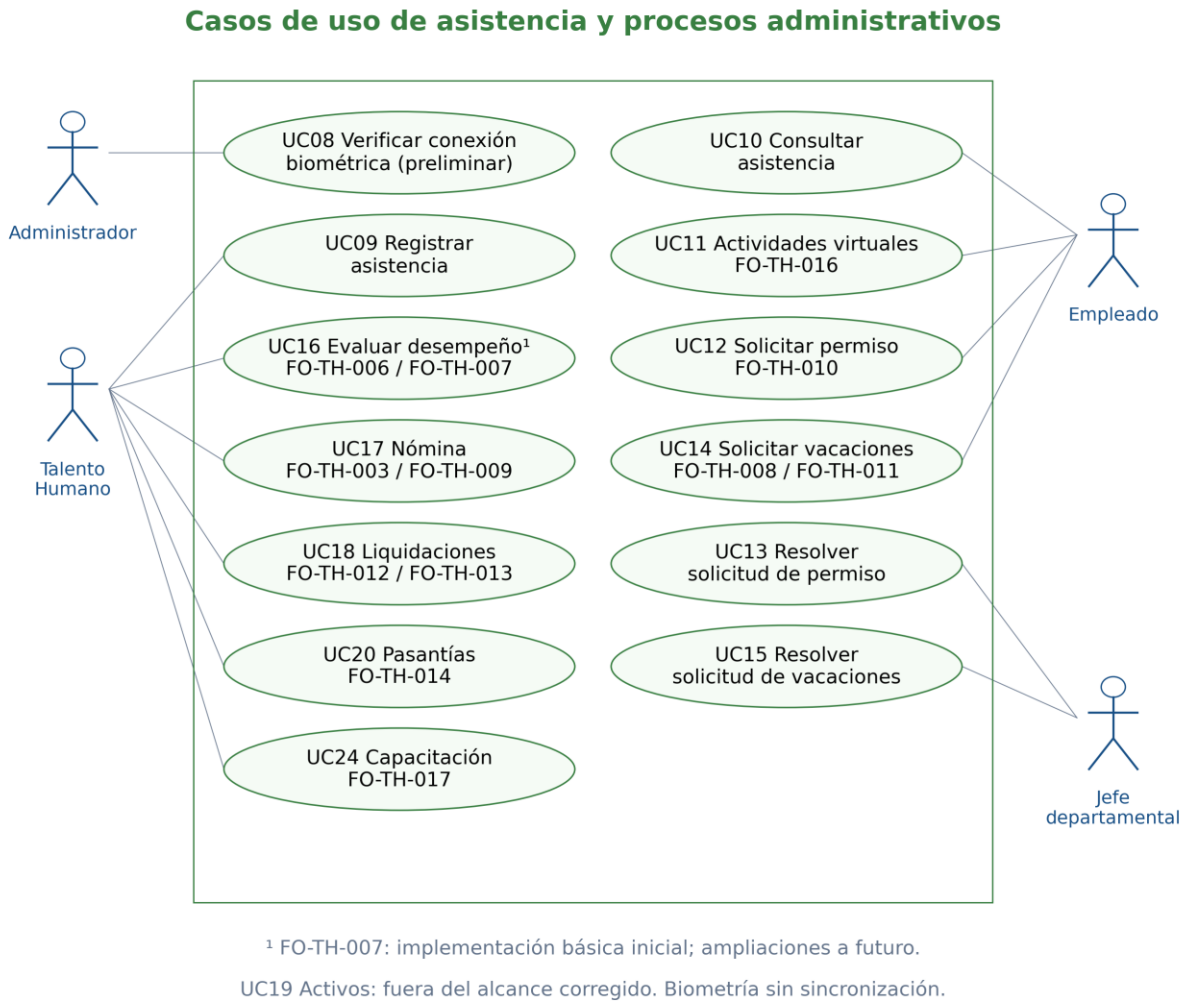


Figura 2. Casos de uso de asistencia y procesos administrativos según el alcance corregido.

Nota. Autoría propia. La biometría se limita a conectividad e integración preliminar; las funciones adicionales del FO-TH-007 quedan a futuro. El control de activos no forma parte del alcance corregido.

4.5.2.3 Matriz de trazabilidad de requisitos

Con el fin de garantizar la trazabilidad de extremo a extremo, se vinculó cada requerimiento funcional con la historia de usuario que expresa la necesidad, el caso de uso que modela la interacción, la funcionalidad y el módulo implementados, el sprint de desarrollo y el caso de prueba utilizado para su comprobación. Esta relación permite recorrer el proyecto desde la necesidad identificada hasta el resultado de validación y facilita detectar funcionalidades sin respaldo en el análisis o el diseño.

Tabla 9

Matriz de trazabilidad entre requerimientos, historias de usuario, casos de uso, funcionalidades y pruebas

RF	HU	Caso de uso	Funcionalidad / módulo / sprint	Caso de prueba y resultado
RF01	HU01	UC01 Autenticar usuario	Autenticación / usuarios / Sprint 1	CP01: credenciales válidas crean la sesión; credenciales incorrectas o cuenta inactiva se rechazan. Cumple.
RF02-RF03	HU02, HU15	UC02 Gestionar usuarios, roles y permisos	Usuarios, roles y permisos FO-TH-002 / usuarios / Sprint 1	CP02: crea, modifica o desactiva cuentas con autorización y bloquea la ruta sin permiso. Cumple.
RF05	HU03	UC03 Administración de cargos (futuro)	Fuera del alcance implementado	CP04: no se considera ejecutado dentro del alcance corregido.

RF	HU	Caso de uso	Funcionalidad / módulo / sprint	Caso de prueba y resultado
RF04	HU04	UC04 Gestionar empleados	Base de datos del personal FO-TH-001 / empleados / Sprint 2	CP03: crea, consulta, actualiza y filtra; informa campos obligatorios o identificadores duplicados. Cumple.
RF21	HU05	UC05 Gestionar contratos (futuro)	Futuras implementaciones	CP05: no se considera ejecutado; gestión de contratos pendiente.
RF17	HU16	UC06 Gestionar reclutamiento	Ficha FO-TH-015 / recruitment / Sprint 2	CP16: ejecuta CRUD, calcula el puntaje y genera el documento; impide guardar datos incompletos. Cumple.
RF18	HU17	UC07 Gestionar onboarding	Inducción FO-TH-004 / onboarding / Sprint 2	CP17: ejecuta CRUD e importa el documento institucional; reporta formato o datos inválidos. Cumple.
RF06	HU06	UC08 Verificar conectividad biométrica	Conectividad e integración preliminar / asistencia / Sprint 3	CP06: verificación de conectividad. No acredita importación ni sincronización de asistencia o administración de usuarios del dispositivo.
RF06	HU06	UC09 Registrar asistencia manual	Registro manual / asistencia / Sprint 3	CP06: gestión de registros de asistencia dentro del alcance definido; sin atribuir importación ni sincronización biométrica.

RF	HU	Caso de uso	Funcionalidad / módulo / sprint	Caso de prueba y resultado
RF06, RF14	HU06, HU14	UC10 Consultar asistencia	Consulta y filtros / asistencia / Sprint 3	CP06: consulta de los registros de asistencia del sistema, según los permisos asignados; sin atribuir sincronización biométrica.
RF20	HU19	UC11 Registrar informes de actividades virtuales	FO-TH-016 / asistencia / Sprint 3	CP19: valida el informe y permite consultar, editar, eliminar, importar o generar según permisos. Cumple.
RF07	HU07	UC12 Registrar solicitud de permiso	Solicitud de permiso FO-TH-010 / permisos / Sprint 3	CP07: registra la solicitud y rechaza fechas traslapadas o campos incompletos. Cumple.
RF07	HU07	UC13 Aprobar o rechazar permiso	Resolución / permisos / Sprint 3	CP07: actualiza estado y observación; impide resolver sin autorización. Cumple.
RF08	HU08	UC14 Registrar solicitud de vacaciones	Control y solicitud de vacaciones FO-TH-008 y FO-TH-011 / vacaciones / Sprint 3	CP08: calcula días, registra la solicitud y rechaza saldo insuficiente o traslapes. Cumple.
RF08	HU08	UC15 Aprobar o rechazar vacaciones	Resolución y actualización del saldo FO-TH-008 y FO-TH-011 / vacaciones / Sprint 3	CP08: actualiza estado y saldo; bloquea una resolución no autorizada. Cumple.

RF	HU	Caso de uso	Funcionalidad / módulo / sprint	Caso de prueba y resultado
RF09	HU09	UC16 Evaluar desempeño	FO-TH-006 y funcionalidades básicas iniciales del FO-TH-007 / desempeño / Sprint 4	CP09: verificación del FO-TH-006 y de las funciones básicas disponibles del FO-TH-007. No acredita las funcionalidades adicionales previstas a futuro.
RF10	HU10	UC17 Registrar novedades de nómina	Décimos y novedades FO-TH-003 y FO-TH-009 / payroll / Sprint 4	CP10: registra la novedad y rechaza duplicados o periodos inválidos. Cumple.
RF11	HU11	UC18 Gestionar liquidaciones	Liquidación de haberes y vacaciones FO-TH-012 y FO-TH-013 / Sprint 4	CP11: calcula y guarda; impide valores, periodos o datos incompletos. Cumple.
RF12	HU12	UC19 Controlar activos (fuera del alcance)	No incluido en los módulos desarrollados del alcance corregido	CP12: no se considera ejecutado dentro del alcance corregido.
RF13	HU13	UC20 Gestionar pasantías	Control de pasantías FO-TH-014 / Sprint 5	CP13: registra periodo, responsable y actividades; rechaza fechas inconsistentes. Cumple.
RF14-RF16	HU14	UC21 Generar reportes	Filtros y reportes / Sprints 5-6	CP14-CP15: aplica filtros, presenta información y conserva la navegación; informa ausencia de datos. Cumple.

RF	HU	Caso de uso	Funcionalidad / módulo / sprint	Caso de prueba y resultado
RF19, RF03	HU18, HU02	UC22 Visualizar y vincular cuentas de Active Directory	Cuentas institucionales y vinculación con empleados y usuarios / Sprints 1 y 6	CP18: visualización y vinculación de cuentas institucionales según permisos. Sin atribuir operaciones adicionales sobre el directorio.
RF22	HU20	UC23 Gestionar acta de entrega y recepción	FO-TH-005 / gestión documental / Sprint 2	CP20: ejecuta CRUD, importa o genera el acta y valida los campos obligatorios. Cumple.
RF23	HU21	UC24 Gestionar plan anual de capacitación	FO-TH-017 / capacitación / Sprint 5	CP21: registra actividades, periodos, responsables y participantes, y rechaza datos inconsistentes. Cumple.

Nota. Autoría propia. Los identificadores RF, HU, UC y CP se conservan como referencias de la planificación inicial. Su presencia no implica implementación o prueba: cada fila distingue el alcance desarrollado, preliminar, futuro o excluido. CP04, CP05 y CP12 no se contabilizan como pruebas ejecutadas del alcance corregido.

La matriz evidencia que los 23 requerimientos funcionales están cubiertos por las 21 historias de usuario, los 24 casos de uso, las funcionalidades implementadas y los 21 casos de prueba. También confirma la representación explícita de Reclutamiento, Onboarding, actas de entrega y recepción, informes de actividades virtuales, capacitación y Active Directory, así como la separación de estructura organizacional, contratos, permisos, vacaciones, nómina, liquidaciones, activos y pasantías. Los requerimientos no funcionales se verifican

transversalmente mediante autenticación, permisos por rol, validaciones, integridad, trazabilidad, seguridad de credenciales, tiempos de respuesta, navegación y compatibilidad en el entorno local.

4.5.2.4 Especificación textual de los casos de uso

A continuación, se especifican los casos de uso con sus actores, objetivo, precondiciones, flujo principal, alternativas, postcondiciones y reglas de negocio. Esta descripción complementa los diagramas y la matriz de trazabilidad.

UC01. Autenticar usuario

Actor(es): Usuario institucional. **Objetivo:** Permitir el ingreso seguro al sistema. **Precondiciones:** La cuenta debe existir y estar activa en el sistema o en Active Directory. **Flujo principal:** El usuario ingresa sus credenciales; el sistema las valida localmente o mediante LDAP; si son correctas, crea la sesión y presenta las opciones autorizadas. **Flujos alternativos o excepciones:** Si las credenciales son incorrectas, la cuenta está inactiva o LDAP no responde, se rechaza el acceso y se muestra un mensaje controlado. **Postcondiciones:** La sesión queda iniciada o el acceso queda denegado sin exponer credenciales. **Reglas de negocio:** Solo las cuentas activas pueden ingresar; los menús y rutas dependen de los permisos asignados.

UC02. Gestionar usuarios, roles y permisos

Actor(es): Administrador del sistema. **Objetivo:** Administrar las cuentas locales y sus autorizaciones. **Precondiciones:** El administrador debe estar autenticado y poseer permisos de administración. **Flujo principal:** Consulta las cuentas, crea o modifica un usuario, asigna roles y permisos y guarda los cambios. **Flujos alternativos o excepciones:** Se rechazan usuarios duplicados, datos inválidos o acciones ejecutadas sin autorización. **Postcondiciones:** La cuenta y

sus permisos quedan actualizados y disponibles para el control de acceso. **Reglas de negocio:** Se aplica el principio de mínimo privilegio y se conserva la trazabilidad de la operación.

UC03. Administrar departamentos y cargos

Actor(es): Responsable de Talento Humano. **Objetivo:** Mantener actualizada la estructura organizacional. **Precondiciones:** El actor debe estar autenticado y autorizado. **Flujo principal:** Registra o modifica un departamento o cargo y establece su relación organizacional. **Flujos alternativos o excepciones:** El sistema rechaza duplicados, relaciones inexistentes o campos obligatorios incompletos. **Postcondiciones:** La estructura queda disponible para relacionarla con empleados y contratos. **Reglas de negocio:** Un cargo debe pertenecer a una estructura válida y no puede eliminarse si mantiene relaciones protegidas.

UC04. Gestionar empleados

Actor(es): Responsable de Talento Humano. **Objetivo:** Centralizar la información personal y laboral del empleado. **Precondiciones:** Deben existir los catálogos organizacionales requeridos. **Flujo principal:** El actor registra, consulta o actualiza la ficha; opcionalmente la vincula con una cuenta institucional de Active Directory. **Flujos alternativos o excepciones:** Se informa cuando el documento está duplicado, faltan datos obligatorios o la cuenta ya está vinculada. **Postcondiciones:** La ficha queda almacenada y disponible para los demás procesos. **Reglas de negocio:** El documento del empleado es único y la relación con una cuenta de usuario es opcional y uno a uno.

UC05. Gestionar contratos

Actor(es): Responsable de Talento Humano. **Objetivo:** Conservar el historial contractual del personal. **Precondiciones:** El empleado y los datos organizacionales deben existir. **Flujo principal:** El actor registra el contrato, define fechas, cargo, modalidad y estado y lo vincula con

el empleado. **Flujos alternativos o excepciones:** Se rechazan fechas incoherentes, empleados inexistentes o información obligatoria incompleta. **Postcondiciones:** El contrato queda incorporado al historial laboral. **Reglas de negocio:** Todo contrato pertenece a un empleado y debe respetar las reglas de fechas y vigencia.

UC06. Gestionar reclutamiento

Actor(es): Responsable de Talento Humano. **Objetivo:** Registrar y evaluar candidatos mediante el FO-TH-015. **Precondiciones:** El actor debe estar autenticado y autorizado; el candidato todavía no requiere una ficha de empleado. **Flujo principal:** Registra los datos de entrevista, completa los criterios, calcula el puntaje y consulta o genera el documento. **Flujos alternativos o excepciones:** Se impide guardar información incompleta o puntuaciones fuera del rango definido. **Postcondiciones:** La ficha de entrevista queda almacenada con su resultado. **Reglas de negocio:** El candidato es un registro administrado por Talento Humano y no un usuario directo del sistema.

UC07. Gestionar onboarding

Actor(es): Responsable de Talento Humano. **Objetivo:** Documentar la inducción o reinducción mediante el FO-TH-004. **Precondiciones:** Debe existir el empleado o la información institucional necesaria. **Flujo principal:** El actor registra o importa el documento, completa temas, responsables, fechas y evidencias y guarda el registro. **Flujos alternativos o excepciones:** Se rechazan archivos incompatibles, datos obligatorios incompletos o relaciones inexistentes. **Postcondiciones:** La inducción queda vinculada con el empleado y disponible para consulta. **Reglas de negocio:** Cada registro debe indicar tipo, fecha, responsables y estado.

UC08. Importar marcaciones biométricas

Actor(es): Responsable de Talento Humano o Administrador autorizado. **Objetivo:** Incorporar al sistema las marcaciones exportadas por el dispositivo biométrico. **Precondiciones:** El archivo debe estar disponible y los códigos biométricos deben poder relacionarse con empleados. **Flujo principal:** El actor selecciona el archivo; el sistema valida su estructura, asocia las filas y presenta un resumen de procesados, omitidos e inconsistentes. **Flujos alternativos o excepciones:** Las filas duplicadas, mal formadas o sin empleado relacionado se omiten y se reportan. **Postcondiciones:** Las marcaciones válidas quedan almacenadas sin duplicidad. **Reglas de negocio:** La importación no modifica el dispositivo y conserva evidencia del resultado procesado.

UC09. Registrar asistencia manual

Actor(es): Responsable de Talento Humano autorizado. **Objetivo:** Registrar una marcación justificada cuando proceda. **Precondiciones:** El empleado debe existir y el actor debe poseer el permiso correspondiente. **Flujo principal:** Selecciona al empleado, fecha, hora, tipo y justificación y confirma el registro. **Flujos alternativos o excepciones:** Se rechazan duplicados, fechas inválidas o campos incompletos. **Postcondiciones:** La marcación manual queda registrada e identificada. **Reglas de negocio:** La creación manual requiere autorización y debe diferenciarse de una marcación importada.

UC10. Consultar asistencia

Actor(es): Empleado o Responsable de Talento Humano. **Objetivo:** Consultar las marcaciones según el alcance autorizado. **Precondiciones:** El actor debe estar autenticado y deben existir registros de asistencia. **Flujo principal:** Selecciona fechas, empleado, estado u otros filtros y el sistema presenta las coincidencias. **Flujos alternativos o excepciones:** Si no

existen datos, se informa sin generar un error; un empleado no puede consultar registros ajenos. **Postcondiciones:** La información se muestra sin modificar los datos. **Reglas de negocio:** El alcance de la consulta depende del rol; RF14 se aplica como función transversal de filtrado.

UC11. Registrar informe de actividades virtuales

Actor(es): Empleado. **Objetivo:** Documentar la jornada realizada bajo modalidad virtual mediante el FO-TH-016. **Precondiciones:** El empleado debe estar autenticado y tener acceso al formato. **Flujo principal:** Registra fecha, actividades y evidencias; guarda, consulta, edita o genera su informe. **Flujos alternativos o excepciones:** El sistema rechaza campos obligatorios incompletos, formatos inválidos o acciones fuera del permiso asignado. **Postcondiciones:** El informe queda relacionado con el empleado y disponible para revisión autorizada. **Reglas de negocio:** El empleado gestiona sus propios informes; Talento Humano únicamente los consulta o administra según permisos.

UC12. Registrar solicitud de permiso

Actor(es): Empleado. **Objetivo:** Solicitar un permiso laboral mediante el FO-TH-010. **Precondiciones:** El empleado debe estar autenticado y activo. **Flujo principal:** Registra el motivo, las fechas, el tiempo solicitado y los respaldos y envía la solicitud. **Flujos alternativos o excepciones:** Se rechazan traslapes, rangos inválidos o datos incompletos. **Postcondiciones:** La solicitud queda en estado pendiente. **Reglas de negocio:** El solicitante no puede aprobar su propia solicitud y las fechas deben ser coherentes.

UC13. Aprobar o rechazar permiso

Actor(es): Jefe departamental o Responsable de Talento Humano autorizado. **Objetivo:** Resolver una solicitud de permiso pendiente. **Precondiciones:** La solicitud debe existir y el actor debe tener competencia para resolverla. **Flujo principal:** Consulta la solicitud, registra la

decisión y una observación y confirma la resolución. **Flujos alternativos o excepciones:** Se bloquea la resolución duplicada o ejecutada por un usuario no autorizado. **Postcondiciones:** La solicitud queda aprobada o rechazada y se actualiza su seguimiento. **Reglas de negocio:** Toda resolución debe identificar al responsable, la fecha y el estado resultante.

UC14. Registrar solicitud de vacaciones

Actor(es): Empleado. **Objetivo:** Solicitar vacaciones mediante el FO-TH-011. **Precondiciones:** Debe existir un saldo disponible en FO-TH-008. **Flujo principal:** El empleado selecciona el periodo; el sistema calcula días laborables, valida el saldo y registra la solicitud. **Flujos alternativos o excepciones:** Se rechazan traslapes, saldo insuficiente, fechas inválidas o periodos incompletos. **Postcondiciones:** La solicitud queda pendiente sin descontar definitivamente el saldo. **Reglas de negocio:** Los fines de semana y las reglas institucionales se consideran en el cálculo aplicable.

UC15. Aprobar o rechazar vacaciones

Actor(es): Jefe departamental o Responsable de Talento Humano autorizado. **Objetivo:** Resolver la solicitud y actualizar el control de vacaciones. **Precondiciones:** La solicitud debe estar pendiente y conservar saldo suficiente. **Flujo principal:** El actor revisa fechas y saldo, registra la decisión y confirma. **Flujos alternativos o excepciones:** Se bloquea una aprobación no autorizada, duplicada o con saldo insuficiente. **Postcondiciones:** Si se aprueba, se actualizan el estado y el saldo; si se rechaza, el saldo permanece disponible. **Reglas de negocio:** La resolución debe conservar responsable, fecha y observación.

UC16. Evaluar desempeño

Actor(es): Jefe departamental o Responsable de Talento Humano. **Objetivo:** Registrar evaluaciones de periodo de prueba y desempeño mediante FO-TH-006 y FO-TH-007.

Precondiciones: El empleado, el periodo y los criterios deben existir. **Flujo principal:** El actor selecciona al empleado, completa criterios y puntuaciones, registra observaciones y confirma el resultado. **Flujos alternativos o excepciones:** Se rechazan criterios incompletos, puntuaciones fuera de rango o periodos inválidos. **Postcondiciones:** La evaluación y su detalle quedan almacenados y relacionados con el empleado. **Reglas de negocio:** El total se calcula con los criterios definidos y no puede exceder la escala establecida.

UC17. Registrar novedades de nómina

Actor(es): Responsable de Talento Humano. **Objetivo:** Gestionar acumulación de décimos y novedades mediante FO-TH-003 y FO-TH-009. **Precondiciones:** El empleado y el periodo deben existir. **Flujo principal:** El actor registra el tipo de novedad, periodo, valores y observaciones y confirma. **Flujos alternativos o excepciones:** Se rechazan duplicados, valores inválidos o periodos inconsistentes. **Postcondiciones:** La novedad queda vinculada con el empleado y el periodo. **Reglas de negocio:** Cada registro debe mantener una clave o combinación única conforme a la regla del proceso.

UC18. Gestionar liquidaciones

Actor(es): Responsable de Talento Humano. **Objetivo:** Registrar liquidaciones de haberes y vacaciones mediante FO-TH-012 y FO-TH-013. **Precondiciones:** Deben existir el empleado, su historial laboral y los periodos requeridos. **Flujo principal:** El actor completa datos de salida, periodos y valores; el sistema realiza los cálculos configurados y guarda. **Flujos alternativos o excepciones:** Se rechazan valores no válidos, periodos duplicados o información obligatoria incompleta. **Postcondiciones:** La liquidación queda almacenada y disponible para generar el documento. **Reglas de negocio:** Los valores deben corresponder al empleado y a periodos no duplicados.

UC19. Controlar activos

Actor(es): Responsable autorizado de activos o Talento Humano. **Objetivo:** Registrar la asignación y devolución de bienes institucionales. **Precondiciones:** El activo y el empleado deben existir y el bien debe estar disponible. **Flujo principal:** Selecciona el bien y el empleado, registra la entrega y posteriormente la devolución cuando corresponda. **Flujos alternativos o excepciones:** Se bloquea la asignación de un bien no disponible o una devolución inexistente. **Postcondiciones:** El estado y el responsable del activo quedan actualizados. **Reglas de negocio:** Un activo no puede mantener dos asignaciones activas simultáneas.

UC20. Gestionar pasantías

Actor(es): Responsable de Talento Humano. **Objetivo:** Administrar el FO-TH-014 y la información de los pasantes. **Precondiciones:** Deben existir el departamento y el responsable institucional. **Flujo principal:** Registra datos del pasante, periodo, responsable y actividades; consulta o actualiza el control. **Flujos alternativos o excepciones:** Se rechazan fechas incoherentes, departamentos inexistentes o datos incompletos. **Postcondiciones:** El registro de pasantía queda relacionado con la unidad responsable. **Reglas de negocio:** El pasante es información administrada por Talento Humano y no un usuario directo del sistema.

UC21. Generar reportes

Actor(es): Responsable de Talento Humano, Administrador o Gerencia autorizada. **Objetivo:** Obtener información consolidada para seguimiento administrativo. **Precondiciones:** El actor debe estar autenticado y tener permiso sobre el módulo consultado. **Flujo principal:** Selecciona el módulo y los filtros, solicita el reporte y el sistema presenta o genera la información. **Flujos alternativos o excepciones:** Si no existen coincidencias, se informa; los datos no autorizados no se incluyen. **Postcondiciones:** El reporte se muestra o genera sin alterar

los registros de origen. **Reglas de negocio:** Los filtros de RF14 y el alcance por rol se aplican transversalmente.

UC22. Administrar cuentas de Active Directory

Actor(es): Administrador del sistema del área de Tecnologías de la Información. **Objetivo:** Consultar y administrar las cuentas institucionales mediante LDAP. **Precondiciones:** El servidor debe estar configurado y el actor debe poseer permisos administrativos. **Flujo principal:** Consulta, crea, modifica, desactiva, restablece contraseñas o asigna grupos y vincula la cuenta con la ficha del empleado. **Flujos alternativos o excepciones:** Se informa de forma controlada la falta de conexión, permisos insuficientes, cuentas duplicadas o atributos inválidos. **Postcondiciones:** La operación autorizada queda aplicada en el directorio y reflejada en la relación con el empleado. **Reglas de negocio:** Las credenciales no se muestran ni se almacenan en texto legible; la vinculación usuario-empleado es uno a uno.

UC23. Gestionar acta de entrega y recepción

Actor(es): Responsable de Talento Humano. **Objetivo:** Documentar la transferencia del puesto mediante el FO-TH-005. **Precondiciones:** El empleado, el puesto y los responsables deben existir. **Flujo principal:** Registra motivo, bienes, archivos físicos o digitales, responsables y observaciones; importa o genera el acta. **Flujos alternativos o excepciones:** Se rechazan documentos incompatibles, datos obligatorios incompletos o relaciones inexistentes. **Postcondiciones:** El acta queda almacenada y vinculada con el empleado y el proceso de salida o cambio. **Reglas de negocio:** La entrega debe identificar a quien entrega, quien recibe, la fecha y los elementos transferidos.

UC24. Gestionar plan anual de capacitación

Actor(es): Responsable de Talento Humano. **Objetivo:** Planificar y controlar actividades formativas mediante el FO-TH-017. **Precondiciones:** Deben existir el periodo, las unidades y los responsables correspondientes. **Flujo principal:** Registra necesidades, temas, fechas, responsables, participantes y estado; consulta o actualiza el plan. **Flujos alternativos o excepciones:** Se rechazan fechas incoherentes, datos obligatorios incompletos o relaciones inexistentes. **Postcondiciones:** La actividad queda incorporada al plan anual y disponible para seguimiento. **Reglas de negocio:** Cada actividad debe pertenecer a un periodo y unidad responsable y conservar su estado de ejecución.

4.5.3 Arquitectura tecnológica del sistema

El sistema se construyó mediante una arquitectura organizada en tres capas principales: presentación, lógica de aplicación y almacenamiento de datos.

4.5.3.1 Capa de presentación

La capa de presentación correspondió a las interfaces con las que interactúa el usuario y se ejecutó del lado del cliente. Se implementó mediante HTML5 para la estructura semántica de los documentos, CSS3 para la definición de estilos y comportamiento responsivo, y JavaScript (ES6) para la interacción dinámica. Estas tecnologías se integraron con el motor de plantillas de Django (Django Template Language) para la renderización del lado del servidor, lo que permitió inyectar en las vistas HTML datos provenientes del contexto de cada solicitud.

Esta capa permitió presentar la información de manera organizada y facilitar el acceso a los módulos. También incorporó validaciones iniciales y elementos dinámicos para mejorar la interacción del usuario.

4.5.3.2 Capa de aplicación

La lógica de negocio se desarrolló mediante Python 3.11 y Django 4.2. Esta capa procesó las solicitudes HTTP recibidas por el servidor WSGI, aplicó las reglas de negocio, validó los datos ingresados, administró los permisos de acceso y se comunicó con la base de datos a través del ORM (Object-Relational Mapping) de Django. La organización interna del código se estructuró en aplicaciones (apps) independientes por dominio funcional (usuarios, empleados, asistencia, permisos, vacaciones, evaluaciones, nómina, liquidaciones, activos, pasantías y reportes), lo que favoreció la mantenibilidad y la separación de responsabilidades.

Como decisión de diseño, las nuevas funcionalidades se organizaron como aplicaciones Django independientes, con rutas, modelos, formularios, vistas y plantillas propios. Esta estructura permitió incorporar o modificar un dominio funcional sin alterar innecesariamente los módulos no relacionados y mantuvo el código conforme al patrón Modelo-Vista-Plantilla y a las convenciones del framework.

Django permitió estructurar el sistema mediante el patrón Modelo-Vista-Plantilla. Los modelos representaron las entidades almacenadas; las vistas procesaron las solicitudes y operaciones; y las plantillas presentaron la información al usuario.

4.5.3.3 Capa de datos

La capa de datos se implementó mediante MySQL 8.0 con el motor de almacenamiento InnoDB, que provee soporte transaccional ACID y restricciones de integridad referencial mediante llaves foráneas. La definición del esquema se gestionó mediante el sistema de migraciones de Django, lo que permitió versionar los cambios estructurales de la base de datos y aplicarlos de manera controlada. En ella se almacenó la información correspondiente a usuarios,

empleados, departamentos, cargos, contratos, asistencia, permisos, vacaciones, evaluaciones, nómina, liquidaciones, activos, pasantías y demás registros administrativos.

Las relaciones entre las tablas permitieron vincular la información del empleado con los diferentes procesos en los que participaba, reduciendo la duplicidad y facilitando las consultas.

4.5.3.4 Seguridad y control de acceso

La seguridad se implementó mediante los mecanismos nativos de Django: autenticación de usuarios con almacenamiento de contraseñas mediante el algoritmo PBKDF2 con SHA-256, gestión de sesiones firmadas del lado del servidor, protección contra ataques CSRF mediante tokens del middleware correspondiente, escape automático del motor de plantillas para mitigar inyección XSS, y uso del ORM para prevenir inyección SQL. El control de acceso a las vistas se aplicó mediante los decoradores login_required y permission_required, así como los mixins LoginRequiredMixin y PermissionRequiredMixin en las vistas basadas en clases. Cada usuario pudo acceder únicamente a las funcionalidades autorizadas de acuerdo con su rol y los permisos asignados.

Adicionalmente, los datos ingresados fueron validados en dos niveles antes de su almacenamiento: validación del lado del cliente mediante atributos HTML5 y JavaScript, y validación del lado del servidor a través de los formularios y ModelForms de Django, evitando así registros incompletos, formatos incorrectos y operaciones no autorizadas.

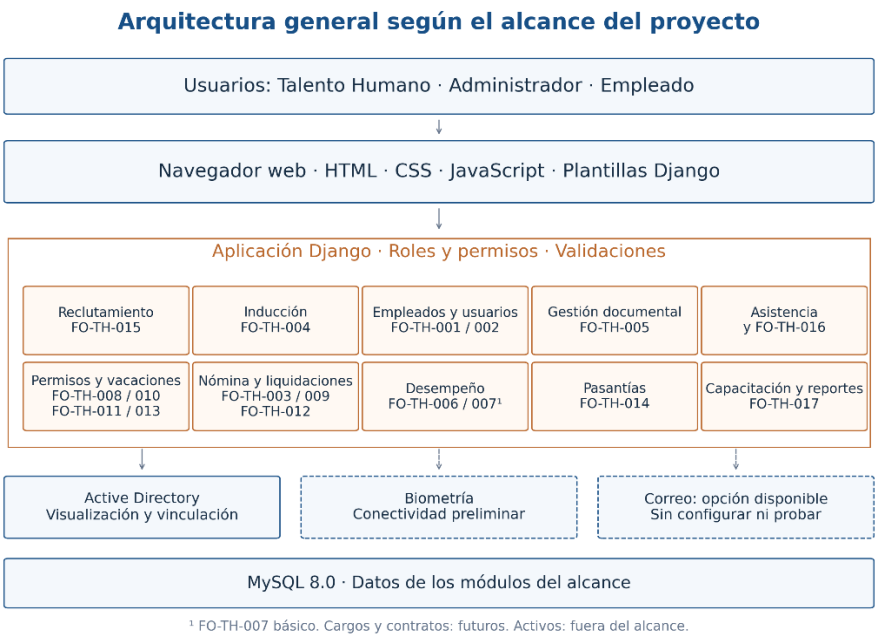


Figura 3. Arquitectura general del sistema web de gestión de Talento Humano.

Nota. Autoría propia. Las líneas discontinuas identifican integración preliminar u opciones futuras. El correo está disponible como opción de configuración, sin pruebas; cargos y contratos quedan para futuras implementaciones.

4.5.3.5 Integración con Active Directory (AD)

Además de la autenticación propia de Django, el sistema incorporó la conexión con el servidor de Active Directory (AD) de la cooperativa mediante el protocolo LDAP (Lightweight Directory Access Protocol). Esta integración permitió que el ingreso al sistema pudiera validarse contra las credenciales institucionales administradas centralmente por el departamento de Tecnologías de la Información, evitando la duplicación de usuarios y contraseñas entre el directorio institucional y la aplicación web.

A partir de esta conexión, se habilitó un módulo administrativo que permite realizar operaciones CRUD (crear, consultar, editar y eliminar) sobre las cuentas de Active Directory directamente desde la interfaz web, sin requerir el acceso directo a las herramientas de

administración del directorio. Entre las operaciones disponibles se encuentran la creación de nuevos usuarios institucionales, la edición de atributos (nombres, correo, cargo, agencia), la desactivación o eliminación de cuentas del personal que finaliza su relación laboral, el restablecimiento de contraseñas y la asignación de grupos o unidades organizativas. Cada operación quedó sujeta a los permisos y al rol del usuario administrador que la ejecuta.

Esta integración contribuyó a centralizar la administración de usuarios institucionales, reducir la carga operativa del área de Tecnologías de la Información y mantener consistencia entre la información del personal registrada en el sistema de Talento Humano y las cuentas de acceso a los recursos de red de la cooperativa.

4.5.3.6 Módulo de asistencia y actividades vinculado al control biométrico

El sistema incorporó un módulo destinado a organizar la información de asistencia proveniente del reloj biométrico institucional. Debido a que el dispositivo biométrico no expone una conexión directa en tiempo real, el módulo permite importar mediante archivos Excel los registros exportados desde el equipo, procesarlos y asociarlos automáticamente con la ficha de cada empleado. De esta manera, la información biométrica dejó de gestionarse mediante hojas de cálculo dispersas y pasó a integrarse dentro del sistema institucional, junto con los informes de actividades bajo modalidad virtual (formato FO-TH-016), permitiendo su consulta, filtrado y seguimiento desde un solo lugar.

4.5.3.7 Diseño de la base de datos

El diseño de la persistencia se elaboró a partir de los requerimientos funcionales y de los formatos institucionales de Talento Humano. Se empleó un modelo relacional normalizado para reducir la redundancia, conservar la integridad referencial y facilitar las consultas

administrativas. La entidad employee_employee constituye el eje del modelo porque vincula la ficha del colaborador con la mayoría de los módulos FO-TH mediante claves foráneas.

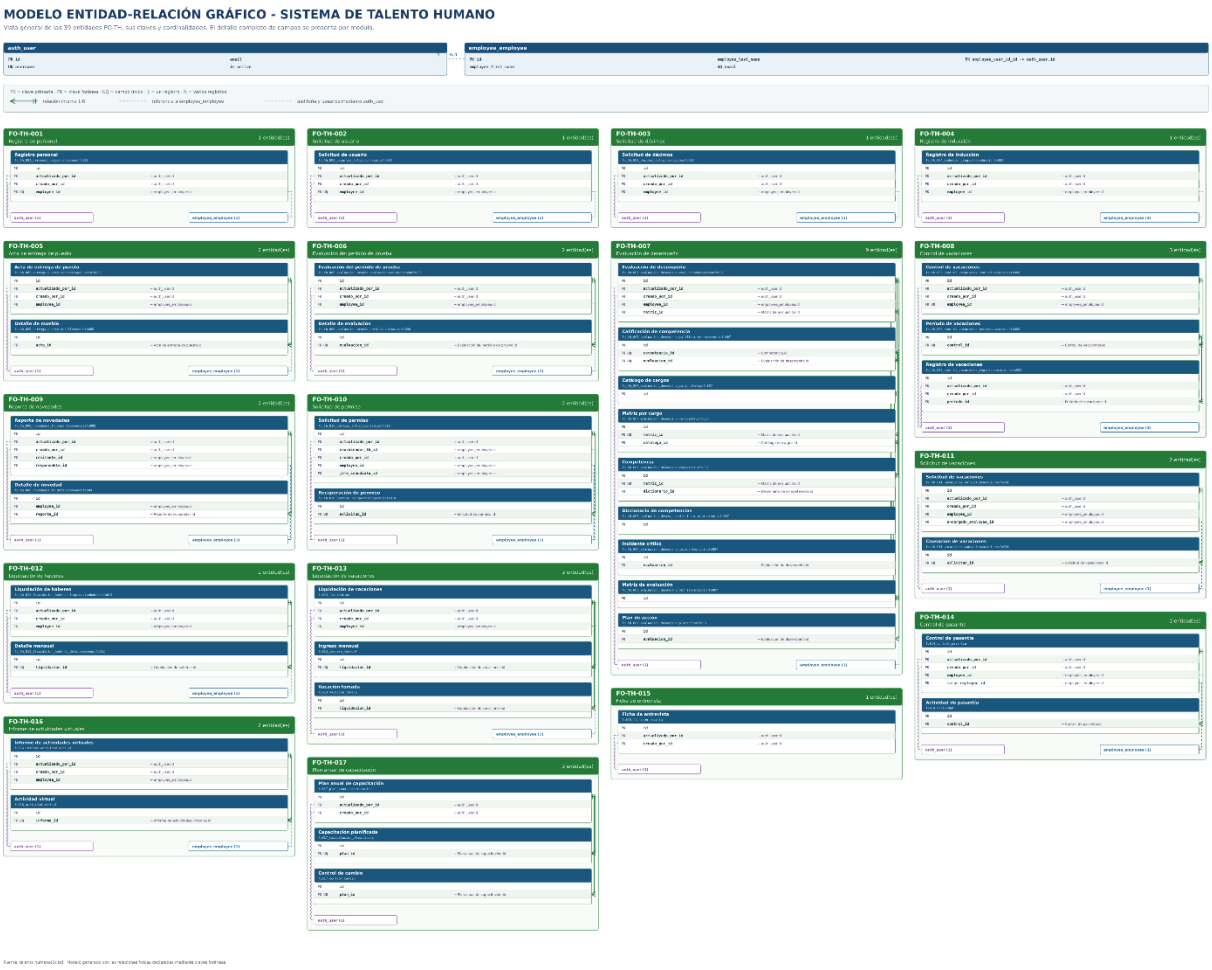


Figura 4. Modelo entidad-relación simplificado de la base de datos del sistema.

Nota. Autoría propia. Se conserva el modelo de datos como referencia estructural. La presencia de entidades de cargos, contratos o activos no acredita su implementación funcional en el proyecto. FO-TH-007 tiene alcance básico inicial; la integración biométrica es preliminar.

Cada entidad posee una clave primaria (PK) y las relaciones se implementan mediante claves foráneas (FK). El usuario se relaciona de forma opcional uno a uno con el empleado; los departamentos y cargos mantienen relaciones uno a muchos con los empleados; y cada empleado

puede poseer múltiples contratos, solicitudes, evaluaciones, registros de asistencia, liquidaciones y demás documentos institucionales.

La Figura 4 representa las entidades principales, sus claves y las cardinalidades relevantes. En azul se muestran las entidades base relacionadas con autenticación, empleado, departamento, cargo y contrato; en verde se agrupan las entidades funcionales vinculadas con empleados; y en naranja se presentan los registros previos a la contratación o relacionados principalmente con departamentos. Las etiquetas PK y FK permiten reconocer las claves primarias y foráneas, mientras que las relaciones 1:N y 0..1:1 muestran la correspondencia entre los registros.

Las entidades de pasantías y planificación de capacitación se relacionan principalmente con el departamento, mientras que la ficha de entrevista se vincula con el usuario responsable de su creación y conserva los datos del candidato antes de que exista una ficha de empleado. Las tablas de detalle, como los componentes de una evaluación, los muebles entregados o los periodos de vacaciones, mantienen relaciones 1:N con su respectiva entidad maestra.

Para conservar la legibilidad, el gráfico omite las tablas técnicas generadas por Django y los atributos secundarios; el esquema completo permanece definido en los modelos y en las migraciones del proyecto. De esta manera, la representación gráfica complementa la descripción textual y permite identificar las entidades, claves foráneas y relaciones que sustentan los requerimientos funcionales.

4.5.3.8 Diagrama de arquitectura y componentes del sistema

La organización general de la solución se representa en el siguiente diagrama de componentes, en el que se identifican las tres capas descritas anteriormente, los elementos que

las integran y los servicios externos con los que el sistema interactúa. El diagrama distingue el flujo de una petición HTTP desde el navegador hasta la base de datos, la validación de credenciales contra el servidor de Active Directory mediante el protocolo LDAP y la incorporación de los registros provenientes del reloj biométrico.

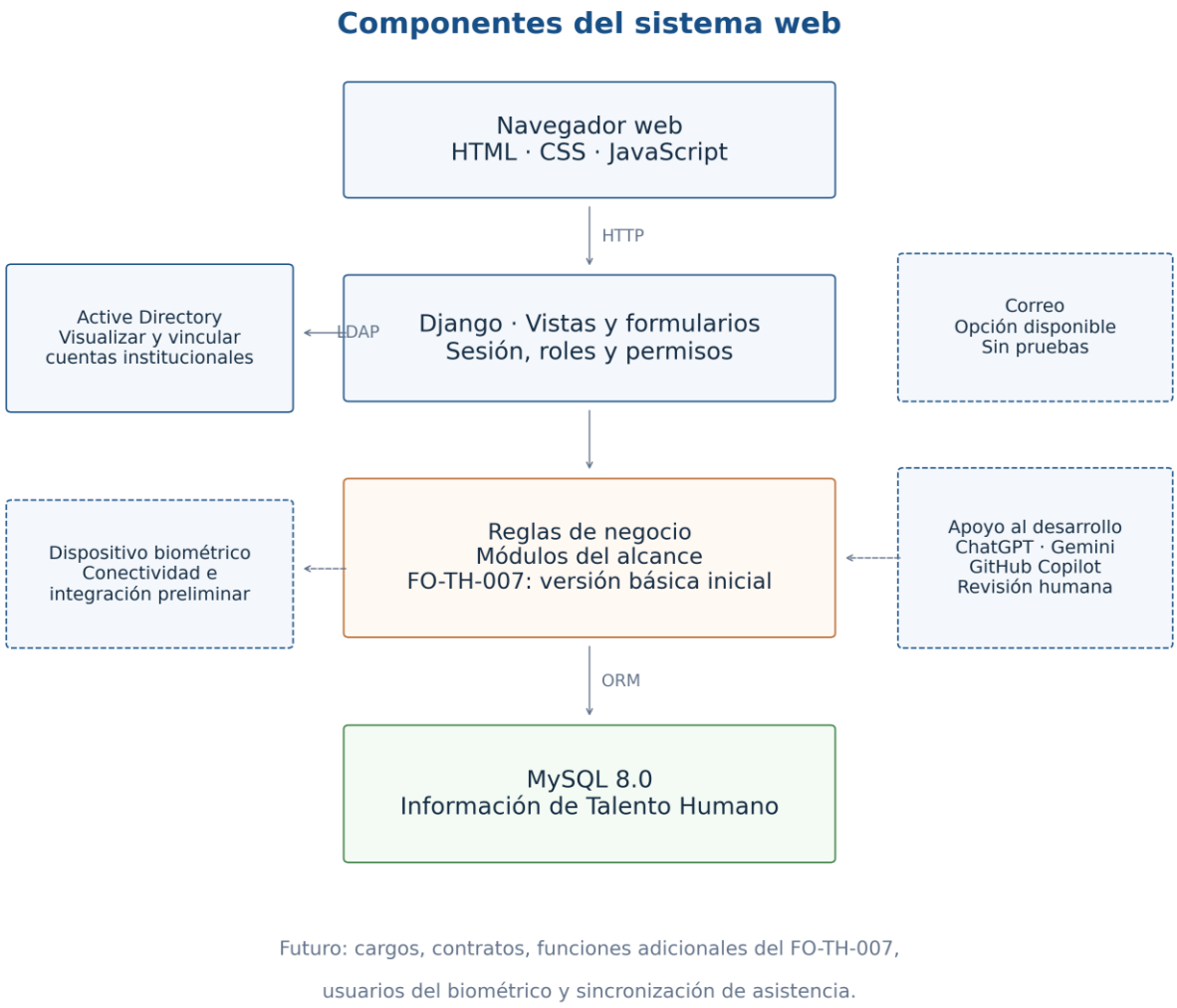


Figura 5. Diagrama de arquitectura y componentes del sistema web de gestión de Talento Humano.

Nota. Autoría propia. La conexión biométrica corresponde únicamente a verificación de conectividad e integración preliminar. No representa sincronización de asistencia ni administración de usuarios del dispositivo. El correo queda sin pruebas y las herramientas de IA se utilizaron durante el desarrollo.

Es importante precisar que las herramientas de inteligencia artificial generativa empleadas bajo el enfoque de Vibe Coding se representan con línea discontinua porque intervinieron únicamente durante la fase de construcción, como apoyo al desarrollador. El sistema en ejecución no realiza llamadas a servicios de inteligencia artificial ni depende de ellos para su funcionamiento, lo que evita dependencias externas, costos de consumo por solicitud y exposición de información institucional a servicios de terceros.

4.5.4 Herramientas utilizadas

Las herramientas seleccionadas respondieron a las características del sistema, la arquitectura definida y las necesidades identificadas.

Tabla 10
Herramientas utilizadas en el desarrollo

Herramienta	Aplicación en el proyecto
Python	Implementación de la lógica de programación del sistema
Django	Desarrollo del backend, modelos, vistas, formularios, rutas, autenticación y permisos
HTML	Estructuración de las interfaces y formularios
CSS	Diseño visual y organización de los componentes
JavaScript	Validaciones, interactividad y comportamiento dinámico
MySQL	Almacenamiento y administración de la información
Navegador web	Ejecución y comprobación de las funcionalidades
ChatGPT	Apoyo para analizar errores, proponer código y elaborar documentación
Gemini	Apoyo para revisar alternativas técnicas y corregir determinados componentes
GitHub Copilot	Asistencia durante la escritura y organización del código

Nota. Autoría propia.

4.5.4.1 Justificación de las decisiones de diseño

La selección tecnológica se fundamentó en el análisis desarrollado en el Capítulo II y se organizó por categorías equivalentes para evitar comparar herramientas de distinta naturaleza. Primero se contrastaron los lenguajes backend Python, Java y PHP; luego los frameworks backend Django, Flask y FastAPI; posteriormente las alternativas de arquitectura de presentación, es decir, renderizado del lado del servidor con Django Template Language, HTML, CSS y JavaScript frente a una aplicación cliente basada en React.js y TypeScript; y, finalmente, los sistemas gestores de bases de datos. En esta última categoría se compararon directamente los gestores relacionales MySQL, PostgreSQL y MariaDB, mientras que MongoDB se mantuvo como alternativa NoSQL de naturaleza distinta. La tabla siguiente relaciona cada decisión del Capítulo IV con los apartados que sustentan su selección.

Cabe precisar que estas decisiones estuvieron condicionadas por la base tecnológica de partida. Al construirse el sistema sobre Horilla, plataforma de software libre desarrollada en Python y Django y licenciada bajo LGPL-2.1, las comparaciones del Capítulo II confirmaron la pertinencia de mantener ese entorno tecnológico en lugar de trasladar la solución a otra plataforma, según se declara en el apartado 3.6.5.

Tabla 11

Correspondencia entre las comparaciones del Capítulo II y las decisiones de diseño

Categoría y decisión adoptada	Alternativas equivalentes analizadas	Justificación técnica	Referencia en el Capítulo II
Lenguaje backend: Python 3.11	Java y PHP	Python se integra de forma nativa con Django y con Horilla, la base tecnológica de software libre sobre la que se construyó el sistema (véase el apartado 3.6.5). Su sintaxis y ecosistema facilitaron el desarrollo de reglas de negocio, automatización e integración con MySQL y LDAP sin trasladar el sistema a otra plataforma.	2.4.2.1-2.4.2.3
Framework backend: Django 4.2 LTS	Flask y FastAPI	Django incorporó ORM, migraciones, formularios, autenticación, permisos, sesiones y panel administrativo. Flask exige seleccionar e integrar extensiones adicionales; FastAPI está orientado principalmente a API y habría requerido construir por separado la capa administrativa.	2.4.2.4-2.4.3; Tabla 1
Arquitectura de presentación: renderizado del servidor con Django Template Language, HTML5, CSS3 y JavaScript	Aplicación cliente por componentes con React.js y TypeScript	El renderizado del servidor cubrió los formularios, listados y reportes administrativos sin mantener una aplicación cliente ni un proceso de compilación independientes. React y TypeScript resultan pertinentes cuando la interfaz requiere mayor complejidad de estado en el cliente.	2.4.1.1-2.4.3; Tabla 1

Categoría y decisión adoptada	Alternativas equivalentes analizadas	Justificación técnica	Referencia en el Capítulo II
SGBD relacional: MySQL 8.0 con InnoDB	PostgreSQL y MariaDB; MongoDB se analizó por separado como NoSQL	MySQL proporcionó transacciones ACID, claves foráneas y compatibilidad directa con el ORM de Django para información administrativa estructurada. MongoDB no se trató como equivalente relacional porque utiliza un modelo documental y delega parte de la integridad referencial a la aplicación.	2.4.4-2.4.9; Tabla 2
Metodología de desarrollo: Scrum	Cascada, Espiral, RUP, Prototipado, XP y Kanban	La priorización cambiante, las entregas revisables y la participación continua de los responsables institucionales justificaron un desarrollo iterativo e incremental.	2.5.3; Tabla 3
Autenticación institucional: LDAP / Active Directory	Autenticación exclusivamente local de Django	La integración centralizó las cuentas institucionales y permitió conservar las políticas de acceso administradas por Tecnologías de la Información.	4.5.3.5
Integración biométrica: conectividad e integración preliminar	Sincronización de asistencia y administración de usuarios del dispositivo, previstas a futuro	El alcance se limitó a verificar la conectividad y preparar la integración. No se acredita una importación o sincronización operativa de marcaciones.	4.5.3.6
Apoyo al desarrollo: ChatGPT, Gemini y GitHub Copilot	Desarrollo sin asistencia generativa	Las herramientas apoyaron el análisis y la generación preliminar; toda propuesta fue revisada, adaptada y probada por el desarrollador antes de incorporarse.	2.6-2.6.4

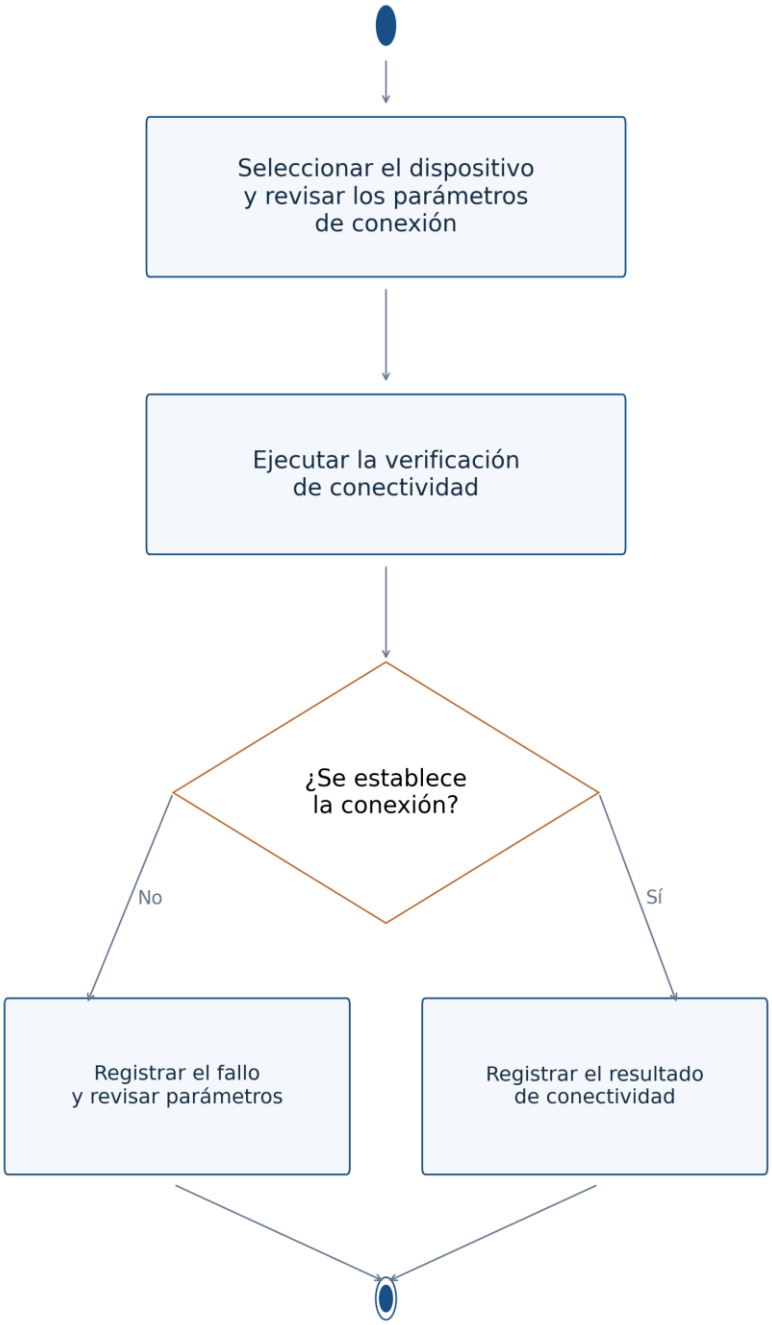
Nota. Autoría propia.

4.5.4.2 Flujo de funcionamiento del sistema

Para complementar la descripción textual de los módulos, los procesos más relevantes del sistema se representaron mediante diagramas de actividades y diagramas de secuencia. Los primeros muestran la lógica del proceso, sus puntos de decisión y los caminos alternativos ante datos inválidos; los segundos describen el intercambio de mensajes entre la interfaz, la capa de aplicación, los servicios externos y la base de datos.

El primer diagrama corresponde a la importación de los registros de marcación provenientes del reloj biométrico, proceso que sustituyó el manejo disperso de hojas de cálculo. El flujo contempla la validación del formato del archivo, la asociación de cada marcación con la ficha del empleado a partir de su código biométrico, el control de registros duplicados y la presentación de un resumen con los registros procesados, omitidos e inconsistentes.

Verificación de conectividad biométrica



Alcance: integración preliminar.
Usuarios del dispositivo y sincronización de asistencia: funcionalidades futuras.

Figura 6. Diagrama de actividades de verificación de conectividad biométrica e integración preliminar (UC08).

Nota. Autoría propia. Flujo limitado a la verificación de conectividad e integración preliminar. La administración de usuarios del dispositivo y la sincronización de asistencia se contemplan como funcionalidades futuras.

El segundo diagrama representa la gestión de permisos y vacaciones, proceso que involucra a más de un actor y que exige controlar el saldo de días disponibles. En él se observan la validación de fechas y traslapes, la transición de estados de la solicitud, la intervención del jefe departamental en la aprobación y la actualización automática del saldo y del módulo de asistencia cuando la solicitud es aprobada.

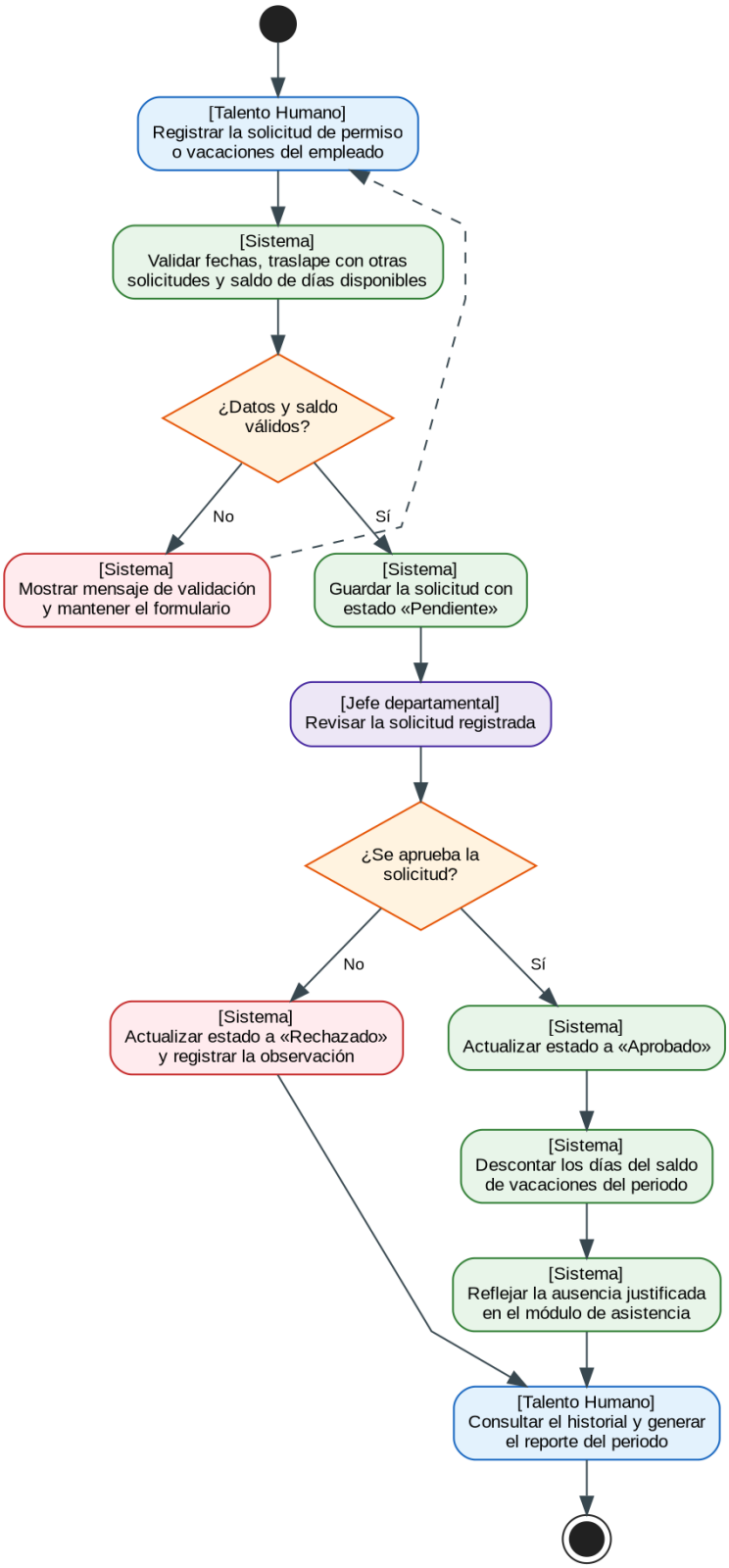

[Talento Humano]
Registrar la solicitud de permiso o vacaciones del empleado
[Sistema]
Validar fechas, traslape con otras solicitudes y saldo de días disponibles
¿Datos y saldo válidos?
No
Sí
[Sistema]
Mostrar mensaje de validación y mantener el formulario
[Sistema]
Guardar la solicitud con estado «Pendiente»
[Jefe departamental]
Revisar la solicitud registrada
¿Se aprueba la solicitud?
No
Sí
[Sistema]
Actualizar estado a «Rechazado» y registrar la observación
[Sistema]
Actualizar estado a «Aprobado»
[Sistema]
Descontar los días del saldo de vacaciones del periodo
[Sistema]
Reflejar la ausencia justificada en el módulo de asistencia
[Talento Humano]
Consultar el historial y generar el reporte del periodo

Figura 7. Diagrama de actividades del proceso de solicitud y aprobación de permisos y vacaciones (UC12-UC15).

Nota. Autoría propia.

En cuanto a los diagramas de secuencia, el primero describe la autenticación de un usuario y la integración con el servidor de Active Directory. Se aprecia la verificación del token CSRF, la validación de las credenciales institucionales mediante el protocolo LDAP, la recuperación del usuario local con su rol y sus permisos, y la creación de la sesión firmada que determina los módulos visibles.

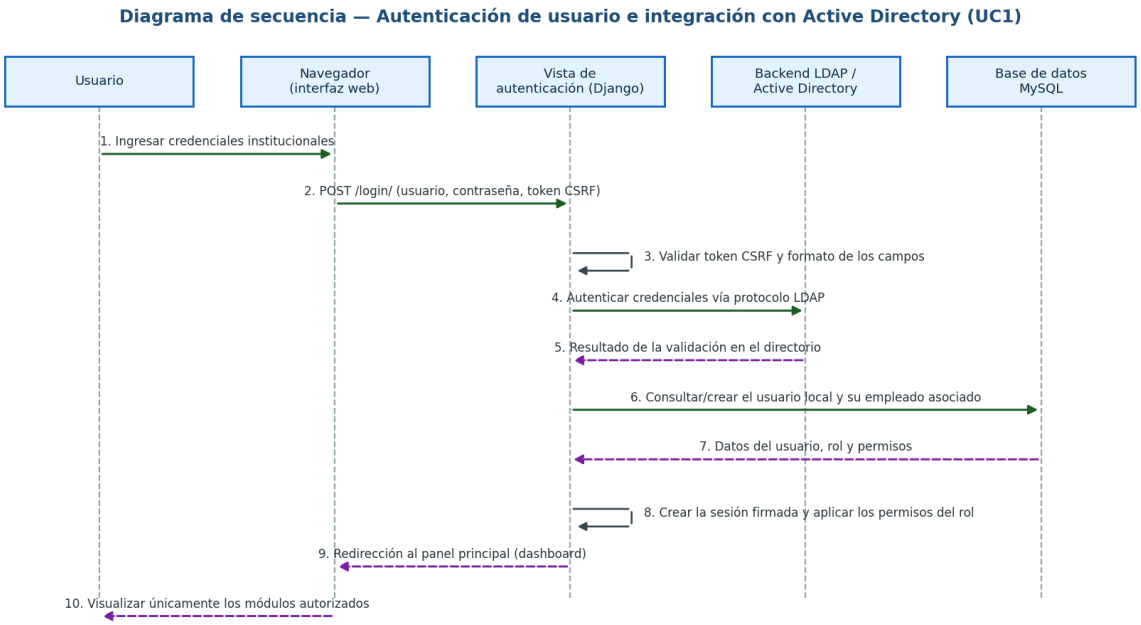


Figura 8. Diagrama de secuencia del proceso de autenticación e integración con Active Directory (UC01).

Nota. Autoría propia.

El segundo diagrama de secuencia corresponde al registro de un nuevo empleado y resulta representativo de las operaciones CRUD implementadas en los demás módulos, ya que refleja el patrón Modelo-Vista-Plantilla: la vista verifica la sesión y los permisos, el ModelForm

valida los datos ingresados y el ORM ejecuta la operación sobre la base de datos dentro de una transacción.

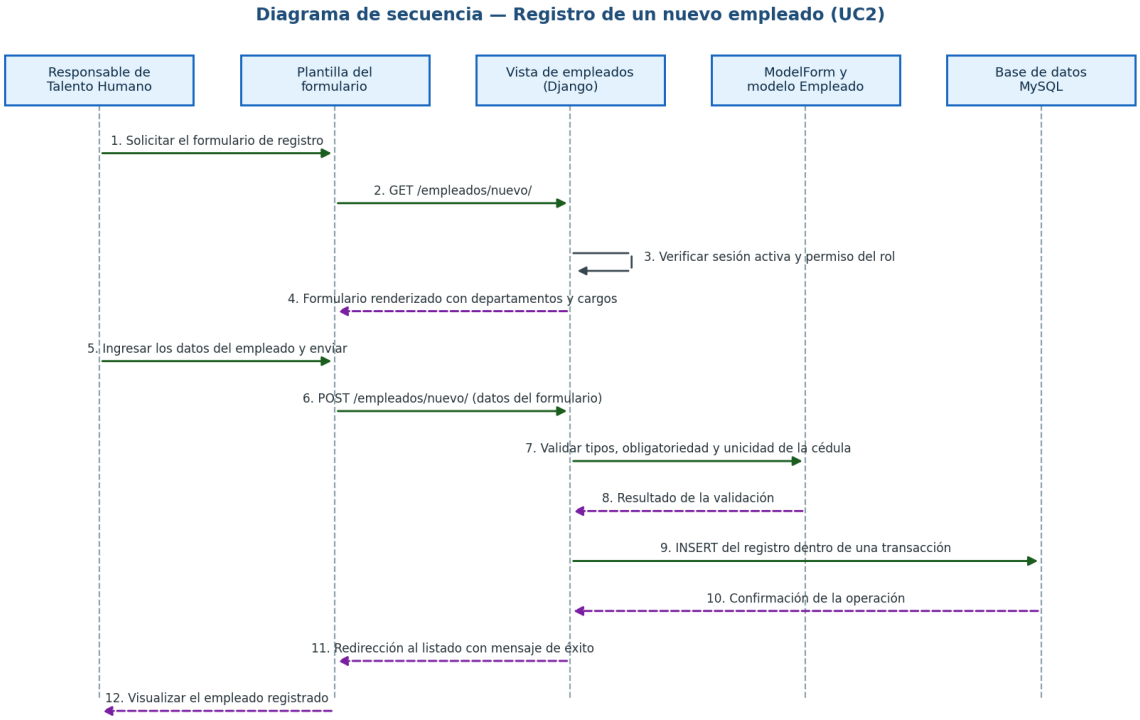


Figura 9. Diagrama de secuencia del proceso de registro de un empleado (UC04).

Nota. Autoría propia.

4.5.5 Desarrollo de los sprints

4.5.5.1 Sprint 1: Configuración, autenticación y estructura inicial

El primer sprint se orientó a establecer la base tecnológica del sistema. Se configuró el proyecto en Django 4.2 dentro de un entorno virtual de Python 3.11, se estableció la conexión con MySQL 8.0 mediante el conector mysqlclient, se definieron las variables de entorno para gestionar credenciales sensibles fuera del código fuente y se creó la estructura inicial de las aplicaciones. Se definieron los modelos necesarios para usuarios, roles y permisos, aprovechando el modelo de usuario personalizado (AbstractUser) y el sistema de grupos y permisos de Django.

Las tablas correspondientes se generaron mediante el sistema de migraciones (makemigrations y migrate).

Durante esta iteración se desarrolló el inicio de sesión, el cierre de sesión y las restricciones iniciales de acceso. Asimismo, se construyó el panel principal y la estructura de navegación para facilitar el acceso a los módulos. En esta misma etapa se inició la configuración de la conexión con el servidor de Active Directory de la cooperativa mediante LDAP, sentando la base para la autenticación institucional y para el módulo de administración de cuentas de AD que se consolidó en sprints posteriores.

Las pruebas permitieron comprobar el ingreso mediante credenciales válidas, el rechazo de credenciales incorrectas y la restricción de opciones según los permisos asignados.

El incremento obtenido fue una estructura funcional que permitió autenticar usuarios, administrar accesos y disponer de una base para la incorporación de los demás módulos.

4.5.5.2 Sprint 2: Reclutamiento, onboarding y gestión de empleados

En el segundo sprint se desarrollaron los componentes relacionados con la incorporación y la información principal del personal. Se implementaron la ficha de entrevista del módulo de Reclutamiento (FO-TH-015), los registros de inducción y reinducción del módulo de Onboarding (FO-TH-004), y los formularios para administrar empleados, departamentos, cargos, contratos y datos laborales.

El módulo de Reclutamiento permitió crear, evaluar, consultar, editar y eliminar fichas de entrevista, además de generar el formato institucional. El módulo de Onboarding permitió registrar y dar seguimiento a la inducción y reinducción, incluida la importación del documento

institucional. El módulo de empleados concentró la información personal, de contacto, institucional y laboral, con opciones de consulta, modificación y filtrado.

Las relaciones entre empleados, departamentos, cargos y contratos organizaron la información conforme a la estructura institucional. Durante las revisiones también se ajustaron los criterios de evaluación de las entrevistas, los campos de inducción, los datos obligatorios, las etiquetas y las opciones de búsqueda de acuerdo con las observaciones de los responsables institucionales.

El incremento obtenido integró el reclutamiento, el onboarding y la administración central del personal, desde la ficha de entrevista hasta el registro de la información laboral.

4.5.5.3 Sprint 3: Asistencia, permisos y vacaciones

El tercer sprint se enfocó en los procesos relacionados con el tiempo laboral y las ausencias. Se desarrollaron funcionalidades para registrar, organizar y consultar la información de asistencia, así como para controlar permisos y vacaciones.

El módulo de asistencia permitió visualizar los registros correspondientes a cada empleado, aplicar filtros y revisar la información necesaria para el seguimiento administrativo. Debido a que el reloj biométrico institucional no expone una conexión directa en tiempo real, se implementó una funcionalidad de importación de los registros exportados desde el dispositivo mediante archivos Excel, los cuales el sistema procesó y asoció automáticamente con la ficha de cada empleado dentro del entorno local de pruebas. Adicionalmente, se incorporó el módulo de informes de actividades bajo modalidad virtual (FO-TH-016), que complementó el control de asistencia del personal que labora fuera de las instalaciones.

El módulo de permisos permitió registrar solicitudes, fechas, motivos, estados y observaciones. El módulo de vacaciones permitió controlar los periodos solicitados, las fechas correspondientes y el estado de las solicitudes.

Durante las revisiones se corrigieron validaciones de fechas, campos obligatorios, estados de aprobación y opciones de navegación. El incremento obtenido permitió organizar en una misma plataforma los registros de asistencia, permisos y vacaciones.

4.5.5.4 Sprint 4: Evaluaciones, novedades de nómina y liquidaciones

En el cuarto sprint se desarrollaron las funcionalidades relacionadas con la evaluación y los procesos económicos asociados al personal. El módulo de evaluaciones permitió registrar los criterios, resultados, observaciones y periodos correspondientes a la evaluación del desempeño.

El módulo de novedades de nómina permitió organizar los registros administrativos que podían influir en el procesamiento interno de la información laboral. Por su parte, el módulo de liquidaciones permitió almacenar los datos relacionados con la finalización de la relación laboral y los valores considerados dentro del proceso.

Durante esta iteración se revisaron los cálculos, formatos numéricos, campos requeridos y relaciones con los empleados. También se mejoró la presentación de la información para facilitar su consulta.

El incremento obtenido integró los procesos de evaluación, novedades de nómina y liquidaciones con la información genel de los empleados.

4.5.5.5 Sprint 5: Activos, pasantías y reportes

En el quinto sprint se desarrollaron los módulos complementarios relacionados con activos institucionales, pasantías y generación de reportes.

El módulo de activos permitió registrar los bienes asignados al personal, su descripción, estado, fecha de asignación y demás información necesaria para su seguimiento. El módulo de pasantías permitió administrar los datos de los pasantes, periodos, responsables y actividades relacionadas con su participación institucional.

También se incorporaron reportes para consultar de forma organizada la información almacenada en los diferentes módulos. Los reportes se estructuraron a partir de filtros y criterios de búsqueda, permitiendo obtener información relacionada con empleados, asistencia, permisos, vacaciones y demás procesos administrativos.

Las observaciones recibidas permitieron reorganizar determinadas opciones, corregir campos y mejorar la presentación de los reportes.

4.5.5.6 Sprint 6: Integración, pruebas y consolidación

El sexto sprint se destinó a integrar los módulos, comprobar la navegación general y corregir los errores identificados. Se revisaron las relaciones entre los datos, los permisos asignados, las validaciones, los cálculos, los formularios y la presentación de la información.

También se efectuaron pruebas con datos válidos e inválidos para verificar el comportamiento de los formularios. Se revisó que los usuarios sin autorización no pudieran ingresar a determinadas funcionalidades y que los registros se almacenaran correctamente.

Las observaciones de los responsables de Talento Humano y Tecnologías de la Información permitieron realizar los ajustes finales. Posteriormente, se organizó la documentación técnica y funcional del sistema.

El incremento final correspondió a una versión integrada y comprobada del sistema web, disponible para su ejecución en un entorno local de pruebas.

4.5.6 Aplicación de técnicas de Vibe Coding

Las técnicas de Vibe Coding se emplearon como apoyo durante los diferentes sprints, siguiendo un protocolo de revisión humana obligatoria sobre toda salida generada por los modelos. Su utilización permitió formular instrucciones en lenguaje natural para analizar mensajes de excepción, explorar alternativas de implementación, generar estructuras preliminares de vistas, modelos y formularios, y mejorar componentes específicos del proyecto. Las herramientas utilizadas fueron ChatGPT, Gemini y GitHub Copilot, configuradas exclusivamente para tareas de programación sobre fragmentos de código no sensibles.

Las herramientas generativas no incorporaron automáticamente sus respuestas al sistema. Cada propuesta atravesó un ciclo de tres etapas: (i) revisión de coherencia con la arquitectura del proyecto, (ii) adaptación manual al estilo de código, convenciones de nombres y estructura de las apps de Django ya existentes, y (iii) verificación mediante ejecución en el entorno local y pruebas manuales con datos válidos e inválidos. Esta supervisión permitió comprobar su compatibilidad con Django 4.2, MySQL 8.0, las reglas de negocio y los requerimientos institucionales, evitando la incorporación de código no verificado.

Tabla 12
Aplicación de Vibe Coding durante el desarrollo

Actividad	Apoyo de inteligencia artificial	Control aplicado
Análisis de errores	Interpretación de mensajes y propuesta de posibles causas	Revisión del código y reproducción del error
Generación de código	Elaboración preliminar de funciones, formularios o validaciones	Adaptación a la estructura real del proyecto
Consultas de datos	Propuesta de consultas y relaciones entre modelos	Comprobación con la estructura de la base de datos

Interfaz de usuario	Sugerencias para organizar formularios y componentes	Revisión de usabilidad y correspondencia institucional
Pruebas	Propuesta de escenarios válidos e inválidos	Ejecución manual en el entorno local
Documentación	Apoyo para describir componentes y procesos	Revisión y corrección por parte del desarrollador
Seguridad	Identificación de posibles validaciones y restricciones	Comprobación de permisos y accesos
Refactorización	Sugerencias para organizar o simplificar código	Pruebas posteriores para evitar regresiones

Nota. Autoría propia.

Durante el empleo de estas herramientas no se compartieron datos personales, contraseñas, credenciales ni documentos confidenciales de la cooperativa. La responsabilidad sobre la selección, modificación, integración y funcionamiento del código permaneció en el desarrollador.

4.5.6.1 Modelos de inteligencia artificial utilizados y evaluación de su desempeño

Conviene precisar el alcance de la inteligencia artificial dentro del proyecto. El sistema desarrollado no incorpora modelos de inteligencia artificial en tiempo de ejecución ni realiza tareas de predicción o clasificación sobre la información del personal. Los modelos se utilizaron exclusivamente como herramientas de apoyo al desarrollador durante la fase de construcción, conforme al enfoque de Vibe Coding descrito en el apartado 2.6. Por consiguiente, no se ejecutó un proceso de entrenamiento ni de ajuste fino de modelos propios: se emplearon modelos preentrenados de propósito general accesibles mediante sus interfaces comerciales, con la configuración que se detalla a continuación.

Tabla 13
Modelos de inteligencia artificial empleados y su configuración

Herramienta	Modelo y versión	Configuración empleada	Tipo de tarea asistida	Datos proporcionados
ChatGPT (OpenAI)	GPT-4o / GPT-4.1	Interfaz web conversacional, configuración por defecto del proveedor. Se emplearon conversaciones separadas por dominio funcional para mantener el contexto acotado.	Análisis de mensajes de excepción, propuesta de consultas con el ORM, generación preliminar de vistas y formularios, redacción técnica.	Fragmentos de código no sensibles, mensajes de error y descripciones funcionales redactadas por el desarrollador.
Gemini (Google)	Gemini 1.5 Pro	Interfaz web, configuración por defecto. Utilizado como segunda opinión para contrastar las propuestas obtenidas.	Revisión de alternativas técnicas, comparación de enfoques de implementación y corrección de componentes específicos.	Fragmentos de código no sensibles y descripciones de requerimientos.
GitHub Copilot	Copilot (modelo GPT-4o) para Visual Studio Code	Extensión del entorno de desarrollo, sugerencias en línea. Se desactivó el envío de fragmentos en los archivos de	Autocompletado de funciones, generación de estructuras repetitivas de modelos,	Archivos del propio proyecto, excluyendo los que contienen parámetros de conexión.

	configuración con credenciales.	formularios y plantillas.

Nota. Autoría propia.

En cuanto a los parámetros considerados, la interacción con los modelos se estructuró mediante instrucciones que incluían de manera sistemática: (a) la versión del framework y del gestor de base de datos utilizados, (b) el fragmento de código o el mensaje de error correspondiente, (c) la regla de negocio institucional aplicable y (d) el resultado esperado. Se restringió la extensión de las respuestas a componentes específicos, evitando solicitar la generación de módulos completos, con el fin de mantener la revisión manual dentro de límites verificables.

Dado que no se entrenaron modelos propios, no corresponde aplicar métricas de aprendizaje automático como exactitud, precisión o exhaustividad. En su lugar, el desempeño del apoyo generativo se evaluó mediante indicadores relacionados con la utilidad y la seguridad del código incorporado al proyecto, registrados por el desarrollador durante los seis sprints.

Tabla 14

Indicadores utilizados para evaluar el apoyo de las herramientas generativas

Indicador	**Definición**	**Resultado observado**
Tasa de aceptación	Proporción de propuestas incorporadas al proyecto tras la revisión humana, respecto del total de propuestas solicitadas.	Aproximadamente 6 de cada 10 propuestas fueron incorporadas, en todos los casos con adaptaciones.
Nivel de adaptación requerido	Grado de modificación aplicado antes de integrar el código (nula, parcial o total).	La mayor parte requirió adaptación parcial a las convenciones de nombres y a la

		estructura de las apps del proyecto.
Resolución de errores	Proporción de mensajes de excepción cuya causa fue correctamente identificada con apoyo del modelo.	Resultó especialmente eficaz en errores de migraciones, consultas del ORM y validación de formularios.
Corrección funcional posterior	Superación de las pruebas funcionales del módulo tras integrar el código asistido.	Ningún componente se incorporó sin superar la prueba correspondiente del apartado 4.5.7.
Regresiones detectadas	Fallos introducidos en funcionalidades previamente comprobadas tras una refactorización asistida.	Las regresiones identificadas se corrigieron dentro del mismo sprint antes de cerrar la iteración.
Seguridad de la información	Verificación de que no se compartieron datos personales, credenciales ni documentos institucionales.	Se cumplió en la totalidad de las interacciones registradas.

Nota. Los valores corresponden al registro llevado por el desarrollador durante los seis sprints y tienen carácter descriptivo. Autoría propia.

Las limitaciones observadas coinciden con las señaladas en el apartado 2.6.3: los modelos propusieron en ocasiones funciones inexistentes en la versión de Django empleada, soluciones que omitían la validación del lado del servidor o consultas poco eficientes sobre conjuntos de datos amplios. Estas situaciones confirmaron la necesidad del protocolo de revisión humana obligatoria y sustentan la recomendación de documentar los prompts efectivos como base de conocimiento para mantenimientos posteriores.

4.5.7 Validación funcional del sistema

La validación se realizó progresivamente durante las revisiones de los sprints y mediante los casos de prueba funcional CP01-CP21 ejecutados en el entorno local. Se aplicaron pruebas de caja negra basadas en los criterios de aceptación definidos para cada historia de usuario y pruebas de caja blanca sobre las funciones críticas del backend. Para cada módulo se comprobaron las operaciones CRUD, las validaciones de entrada, el flujo de navegación, los permisos por rol y el almacenamiento íntegro de la información. Un caso se consideró cumplido cuando el escenario válido produjo el resultado esperado y el escenario alternativo fue controlado sin almacenar información inconsistente ni generar excepciones no gestionadas.

Tabla 15
Casos de prueba funcional y resultados

Caso de prueba / módulo	Comprobación realizada	Estado
CP01 - Autenticación	Ingreso con credenciales válidas; rechazo de credenciales incorrectas y cuentas inactivas; creación controlada de la sesión.	Cumple
CP02 - Usuarios y permisos	Creación, modificación y desactivación; asignación de permisos; bloqueo del acceso directo sin autorización.	Cumple
CP03 - Empleados	Registro, consulta, actualización y filtrado; control de campos obligatorios e identificadores duplicados.	Cumple
CP04 - Cargos	Gestión contemplada para futuras implementaciones. Se conserva el identificador de la planificación inicial.	Futuro
CP05 - Contratos	Gestión contemplada para futuras implementaciones.	Futuro

Caso de prueba / módulo	Comprobación realizada	Estado
CP06 - Asistencia y conectividad biométrica	Gestión de registros de asistencia y verificación de conectividad e integración preliminar. La administración de usuarios del dispositivo y la sincronización de asistencia quedan a futuro.	Acotado
CP07 - Permisos	Registro, validación de fechas y traslapes, aprobación o rechazo según permisos y seguimiento del estado.	Cumple
CP08 - Vacaciones	Cálculo de días, validación de saldo y periodos, aprobación o rechazo y actualización del estado.	Cumple
CP09 - Evaluaciones	FO-TH-006 y funcionalidades básicas de la implementación inicial del FO-TH-007. No comprende las funcionalidades adicionales previstas a futuro.	Acotado
CP10 - Novedades de nómina	Registro y relación con empleado y periodo; rechazo de duplicados o periodos inválidos.	Cumple
CP11 - Liquidaciones	Registro, cálculo y comprobación de empleado, periodos y valores obligatorios.	Cumple
CP12 - Activos	El control de activos institucionales no integra los módulos desarrollados del alcance corregido.	Excluido
CP13 - Pasantías	Registro y seguimiento de periodo, responsable y actividades; validación de fechas.	Cumple
CP14 - Reportes	Aplicación de filtros y generación de información; aviso cuando no existen coincidencias.	Cumple
CP15 - Navegación y compatibilidad	Acceso entre módulos, visualización sin desbordes y funcionamiento en Chrome y Edge del entorno local.	Cumple
CP16 - Reclutamiento FO-TH-015	CRUD de fichas, cálculo de puntaje, validación de campos y generación del documento institucional.	Cumple

Caso de prueba / módulo	Comprobación realizada	Estado
CP17 - Onboarding FO-TH-004	CRUD de inducción o reinducción e importación del documento Word; rechazo del formato inválido.	Cumple
CP18 - Active Directory	Visualización y vinculación de cuentas institucionales con empleados y usuarios, según los permisos asignados.	Acotado
CP19 - Actividades virtuales FO-TH-016	Registro, consulta, edición, eliminación, importación y generación del informe; validación de campos y permisos.	Cumple
CP20 - Acta de entrega FO-TH-005	Registro, consulta, edición, importación y generación del acta; validación de responsables, bienes, archivos y campos obligatorios.	Cumple
CP21 - Plan de capacitación FO-TH-017	Registro y seguimiento de actividades, periodos, responsables y participantes; rechazo de fechas o datos inconsistentes.	Cumple

Nota. Autoría propia. Se conserva la numeración de los casos de prueba para trazabilidad. Cargos y contratos son futuros; activos queda fuera del alcance. Los resultados de biometría, FO-TH-007 y Active Directory se restringen a las funcionalidades descritas. El servidor de correo no fue objeto de pruebas.

Las pruebas permitieron identificar errores relacionados con campos, permisos, fechas, cálculos y navegación. Estos inconvenientes fueron corregidos durante los sprints antes de consolidar la versión funcional.

Respecto del respaldo de los casos de prueba, se aclara que la ejecución de los veintiún casos abarcó más de trece módulos y, en cada uno de ellos, las cuatro operaciones del ciclo CRUD junto con sus escenarios alternativos, lo que supera las doscientas ejecuciones individuales. Incorporar una captura por cada operación extendería el documento de forma desproporcionada y resultaría redundante, ya que todos los módulos comparten la misma

estructura de vistas, plantillas y validaciones del framework Django, por lo que el comportamiento observado en un módulo es representativo del resto.

Por esta razón, la evidencia se presenta de forma representativa. Las Figuras 10 a 21 del presente capítulo documentan las pantallas principales de los módulos desarrollados, y en el Anexo C se incorpora la secuencia completa del ciclo CRUD aplicada al módulo de reclutamiento FO-TH-015, que muestra el registro de una ficha de entrevista, su almacenamiento, la consulta en el listado, la edición del registro, la descarga del documento institucional y la disponibilidad de la acción de eliminación. Esta secuencia corresponde al procedimiento aplicado de manera sucesiva en los demás módulos evaluados.

En el caso del CP18, la integración con Active Directory se comprueba en las Figuras 20 y 21, que evidencian la correspondencia entre la cuenta administrada desde el sistema web y su existencia en la consola del directorio institucional, lo que constituye la prueba directa de que la operación ejecutada en la aplicación se refleja en el servidor de Active Directory.

Los resultados consignados en la Tabla 15 corresponden, por tanto, a la ejecución registrada por el autor en el entorno local durante las revisiones de cada sprint, y las evidencias gráficas incluidas en el capítulo y en los anexos respaldan el procedimiento aplicado.

4.5.8 Retroalimentación y ajustes realizados

Las revisiones efectuadas con los responsables institucionales permitieron contrastar las funcionalidades construidas con los procedimientos de Talento Humano. Las observaciones fueron registradas e incorporadas al Sprint Backlog correspondiente.

Tabla 16
Principales observaciones y ajustes realizados

Aspecto revisado	Observación identificada	Ajuste realizado
Formularios	Existencia de campos faltantes o poco comprensibles	Incorporación y modificación de campos y etiquetas
Validaciones	Ingreso de fechas o datos inconsistentes	Implementación de reglas y mensajes de validación
Navegación	Opciones difíciles de localizar	Reorganización de menús y accesos
Permisos	Acceso a determinadas funciones sin la restricción esperada	Ajuste de roles y permisos
Cálculos	Diferencias en determinados resultados	Revisión y corrección de la lógica aplicada
Reportes	Necesidad de aplicar criterios de búsqueda	Incorporación de filtros y organización de resultados
Presentación	Tablas o formularios con exceso de información	Mejora de la distribución visual
Seguimiento	Necesidad de visualizar estados de los procesos	Incorporación de estados y observaciones

Nota. Autoría propia.

La retroalimentación permitió mejorar progresivamente el sistema y reducir las diferencias entre los requerimientos iniciales y las funcionalidades desarrolladas.

4.5.9 Producto final

El producto final correspondió a una versión funcional del sistema web de gestión de Talento Humano, ejecutada y validada en un entorno local mediante el servidor de desarrollo de Django (runserver) con MySQL 8.0 como motor de base de datos. La solución integró los módulos definidos en el Product Backlog y permitió comprobar las operaciones principales relacionadas con el registro, consulta, actualización, seguimiento y generación de información.

La estructura del proyecto quedó documentada mediante un archivo requirements.txt con las dependencias del entorno virtual y un archivo README con las instrucciones de despliegue.

El sistema consolidó en una plataforma los procesos de reclutamiento, onboarding, empleados, estructura organizacional, contratos, asistencia, permisos, vacaciones, evaluaciones, nómina, liquidaciones, activos, pasantías y reportes. También incorporó autenticación, roles, permisos, validaciones e integración administrativa con Active Directory para controlar el acceso y proteger la información.

Debido a las delimitaciones de la investigación, el producto final no fue implementado definitivamente en producción dentro de la infraestructura tecnológica institucional. No obstante, dentro del entorno local de pruebas sí se comprobó la conexión con el servidor de Active Directory mediante LDAP para la autenticación y administración de cuentas, así como la importación de los registros exportados desde el reloj biométrico. La estructura desarrollada permitió dejar preparada la solución para una futura revisión, adaptación y puesta en producción por parte de la institución.

Tabla 17
Entregables finales de la propuesta

Entregable	Descripción
Sistema web funcional	Aplicación desarrollada mediante Django y ejecutada en un entorno local
Base de datos	Estructura relacional para almacenar la información de Talento Humano
Módulo de usuarios	Gestión de usuarios FO-TH-002, asignación de roles y permisos y control de acceso. Opción de configuración del servidor de correo disponible para uso futuro, sin pruebas dentro del proyecto.
Módulo de empleados	Administración de información personal y laboral

Módulos administrativos	Asistencia y FO-TH-016; conectividad biométrica e integración preliminar; permisos y vacaciones; FO-TH-006 y funciones básicas iniciales del FO-TH-007; nómina y liquidaciones.
Módulos complementarios	Pasantías FO-TH-014 y plan anual de capacitación FO-TH-017. El control de activos institucionales no forma parte del alcance corregido.
Reportes	Consultas y presentación organizada de información
Validaciones	Controles de datos, fechas, campos obligatorios y permisos
Manual del programador	Estructura, configuración, dependencias, base de datos y componentes técnicos del sistema
Manual de usuario	Orientaciones para acceder y utilizar las funcionalidades principales
Evidencias de pruebas	Registros y capturas de las funciones contempladas en el alcance corregido. No incluyen pruebas de cargos, contratos, activos ni correo; la biometría y el FO-TH-007 se limitan a lo descrito.
Código fuente	Archivos correspondientes a la solución desarrollada
Reclutamiento y onboarding	FO-TH-015, ficha de entrevista; FO-TH-004, inducción y reinducción; y sus funciones de importación o generación documental
Integración con Active Directory	Visualización de cuentas institucionales y vinculación con los empleados y usuarios del sistema.
Formatos institucionales adicionales	Formatos FO-TH-001 a FO-TH-017 según el alcance definido. FO-TH-005 corresponde a gestión documental; FO-TH-007 tiene implementación inicial básica. Cargos y contratos quedan para futuras implementaciones.
Declaración de licenciamiento y atribución	Archivos LICENSE y NOTICE incluidos en el código fuente, con la identificación del proyecto Horilla, su licencia LGPL-2.1 y el detalle de los componentes reutilizados y modificados

Nota. Autoría propia.

4.5.10 Resultados de la propuesta

El desarrollo de la propuesta permitió disponer de una herramienta tecnológica para centralizar los principales procesos administrativos del área de Talento Humano. La organización de la información en una base de datos facilitó la relación entre empleados, cargos, departamentos, contratos, asistencia, permisos, vacaciones y demás registros institucionales.

La aplicación de Scrum permitió dividir el trabajo en incrementos funcionales y realizar ajustes durante el desarrollo. Las revisiones de cada sprint facilitaron la corrección de campos, validaciones, cálculos, permisos, reportes y opciones de navegación.

Las técnicas de Vibe Coding apoyaron la generación, revisión y mejora de determinados componentes, especialmente durante la interpretación de errores, elaboración de funciones y documentación. Sin embargo, todas las propuestas generadas fueron revisadas y comprobadas por el desarrollador antes de su incorporación.

Finalmente, las pruebas funcionales evidenciaron que el sistema permitió registrar, consultar, modificar y visualizar la información de los módulos desarrollados. También se comprobó el funcionamiento de las validaciones, la navegación, los cálculos y los controles de acceso. Por tanto, la propuesta cumplió con el objetivo de desarrollar y validar funcionalmente un sistema web orientado a optimizar la gestión administrativa del personal de la COAC Gañansol.

4.6 Evidencias de la interfaz y funcionalidades del sistema desarrollado

Con el propósito de complementar la descripción funcional presentada en los apartados anteriores, a continuación, se muestran capturas representativas del sistema web construido sobre el framework Django y ejecutado en el entorno local de pruebas. Estas evidencias permiten

observar los módulos desarrollados, mientras que la comprobación de las operaciones CRUD, las validaciones y las funciones de importación o exportación se resume en los apartados 4.5.7 y 4.6.9.

4.6.1 Panel principal (Dashboard) del sistema

La Figura 10 demuestra la funcionalidad de navegación e indicadores del panel principal. Satisface RF03 y RNF01 al mostrar únicamente los accesos autorizados en una interfaz organizada. El actor que utiliza esta vista es el Responsable de Talento Humano o el Administrador, según sus permisos. El resultado es la presentación de los módulos habilitados y de los indicadores generales para iniciar la gestión.

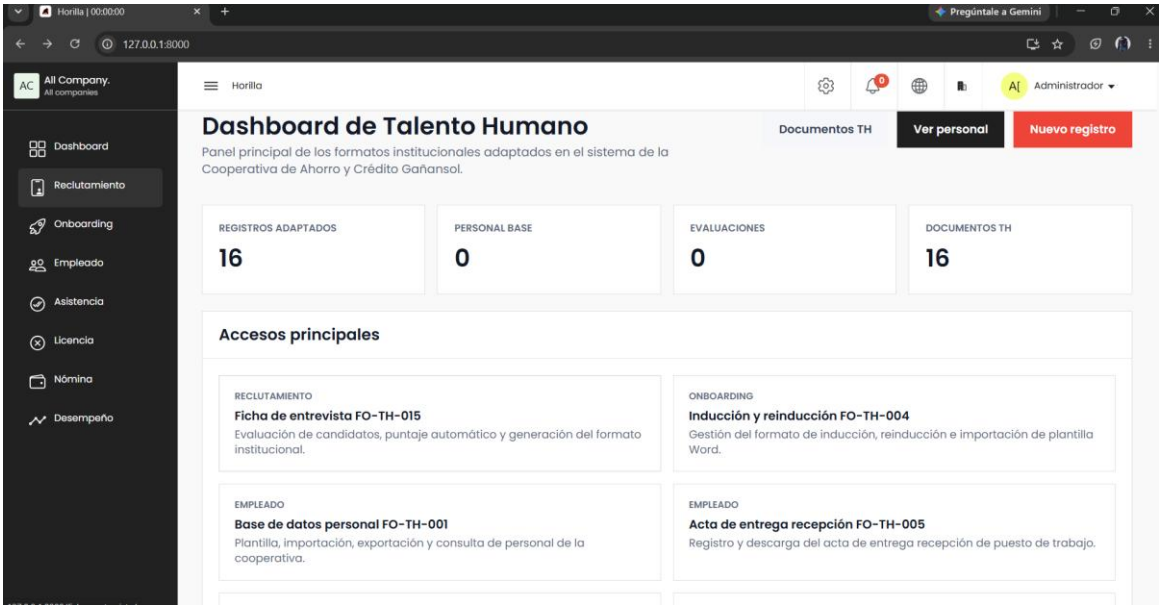


Figura 10. Panel principal (Dashboard) de gestión de Talento Humano.

Nota. Autoría propia.

La Figura 11 demuestra la funcionalidad de acceso directo a los formatos institucionales FO-TH. Satisface RF04 y RF06-RF23 al ofrecer una navegación central hacia los módulos

implementados y respeta RF03 mediante la visibilidad por permisos. El actor principal es el Responsable de Talento Humano. El resultado es la disponibilidad ordenada de las opciones autorizadas para registrar, consultar o generar cada formato.

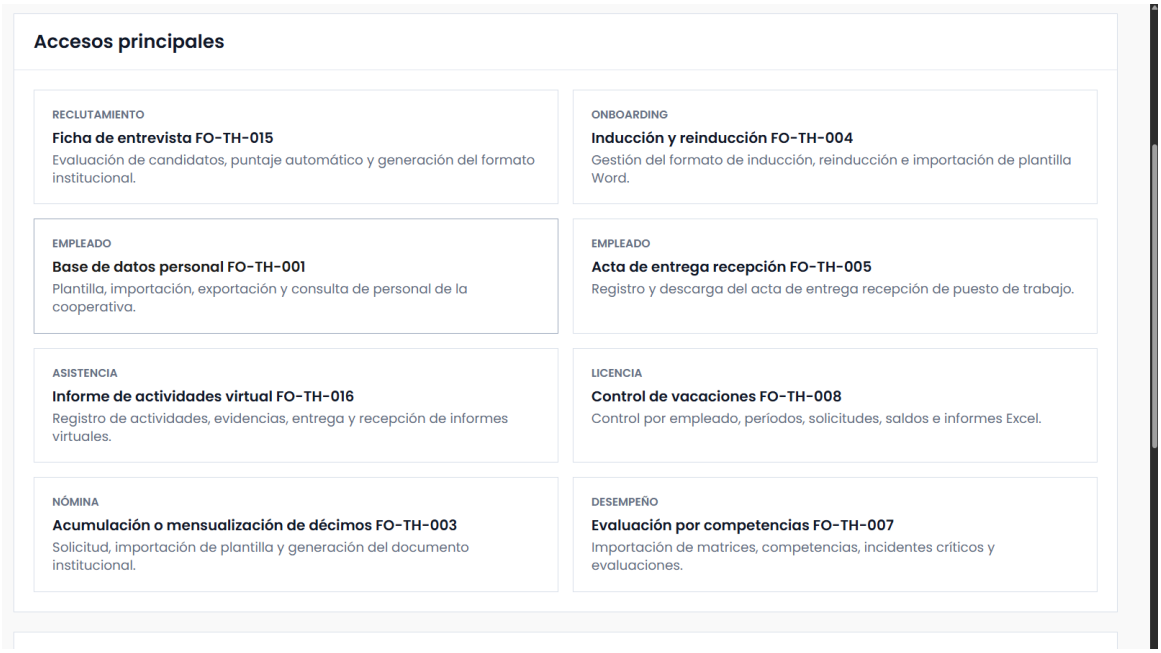


Figura 11. Accesos principales a los formatos institucionales FO-TH.

Nota. Autoría propia. La captura muestra accesos disponibles; su presencia no acredita la implementación completa de todos los módulos. El FO-TH-007 se limita a funciones básicas iniciales.

4.6.2 Módulo de Reclutamiento — Ficha de entrevista (FO-TH-015)

La Figura 12 demuestra la funcionalidad de creación, consulta, edición y eliminación de fichas de entrevista, junto con el cálculo automático del puntaje y la generación del formato institucional. Satisface RF17. El actor principal es el responsable de Talento Humano. El resultado es una ficha FO-TH-015 registrada, evaluada y disponible para consulta o descarga en Word.

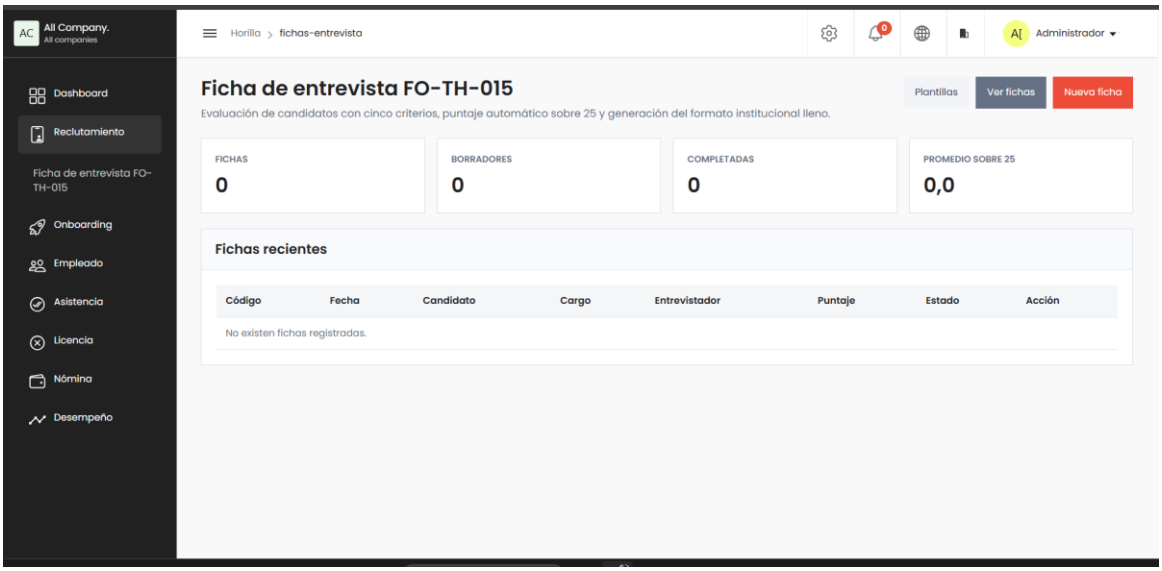


Figura 12. Ficha de entrevista FO-TH-015 del módulo de Reclutamiento.

Nota. Autoría propia.

4.6.3 Módulo de Onboarding — Inducción y reinducción (FO-TH-004)

La Figura 13 demuestra la funcionalidad de administración de registros de inducción y reinducción. Satisface RF18. El actor principal es el responsable de Talento Humano. El resultado es un listado institucional con registros consultables, editables y eliminables, junto con su estado y datos de seguimiento.

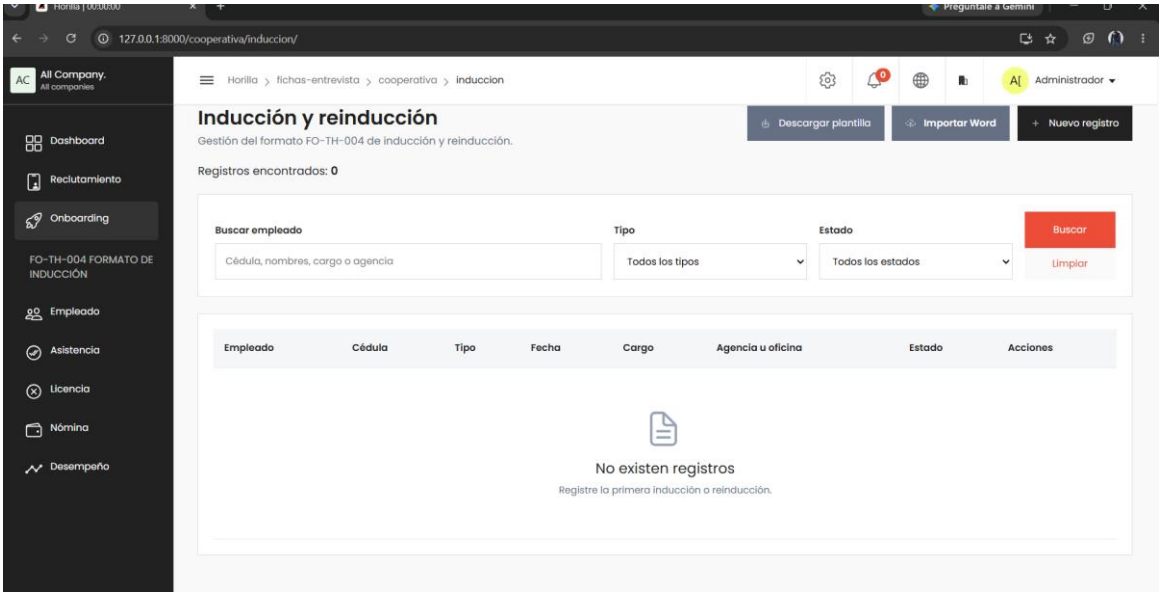


Figura 13. Listado de registros de inducción y reinducción FO-TH-004.

Nota. Autoría propia.

La Figura 14 demuestra la funcionalidad de navegación, búsqueda y filtrado dentro del módulo de Onboarding. Satisface RF14 y RF18. El actor principal es el Responsable de Talento Humano. El resultado es la localización de registros de inducción o reinducción por cédula, nombre, cargo, agencia, tipo y estado.

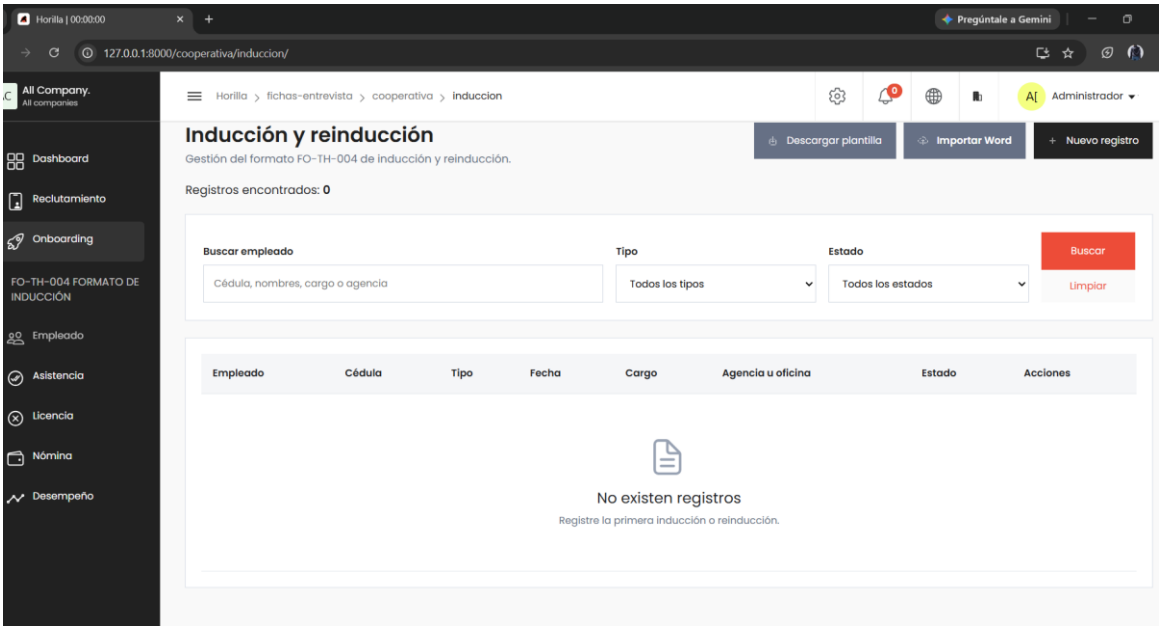


Figura 14. Navegación y búsqueda dentro del módulo de inducción y reinducción.

Nota. Autoría propia.

4.6.4 Módulo de Empleado — Base de datos de personal (FO-TH-001)

La Figura 15 demuestra la funcionalidad de registro, consulta, edición, filtrado, importación y exportación de la base de personal. Satisface RF04 y RF14. El actor principal es el Responsable de Talento Humano. El resultado es una ficha centralizada por colaborador, relacionada con su información personal y laboral.

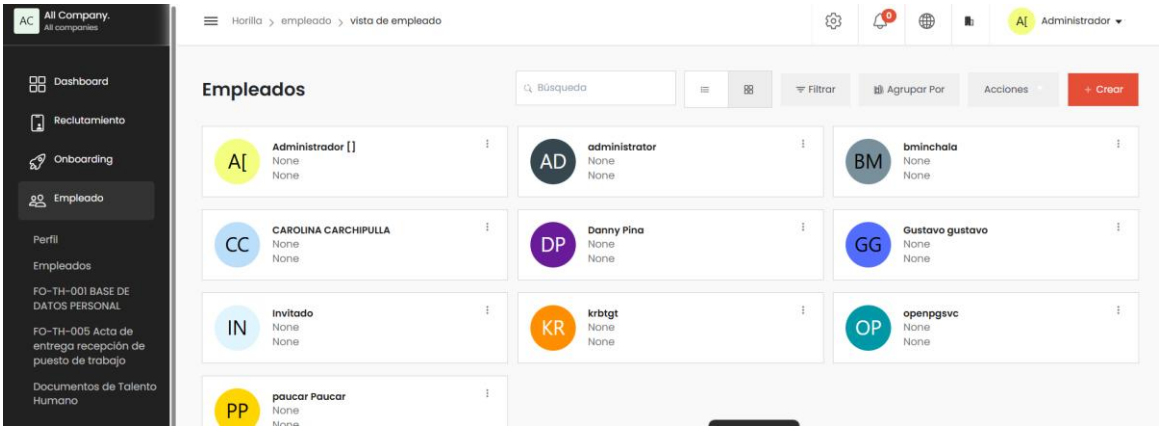


Figura 15. Listado de empleados registrados en el sistema (FO-TH-001).

Nota. Autoría propia.

4.6.5 Módulo de Licencia — Control de vacaciones (FO-TH-008)

La Figura 16 demuestra la funcionalidad de control de periodos, solicitudes y saldos de vacaciones por empleado. Satisface RF08. Los actores que intervienen son el Empleado, el Jefe departamental y el Responsable de Talento Humano. El resultado es una solicitud registrada con los días calculados, el saldo actualizado y el estado de aprobación correspondiente.

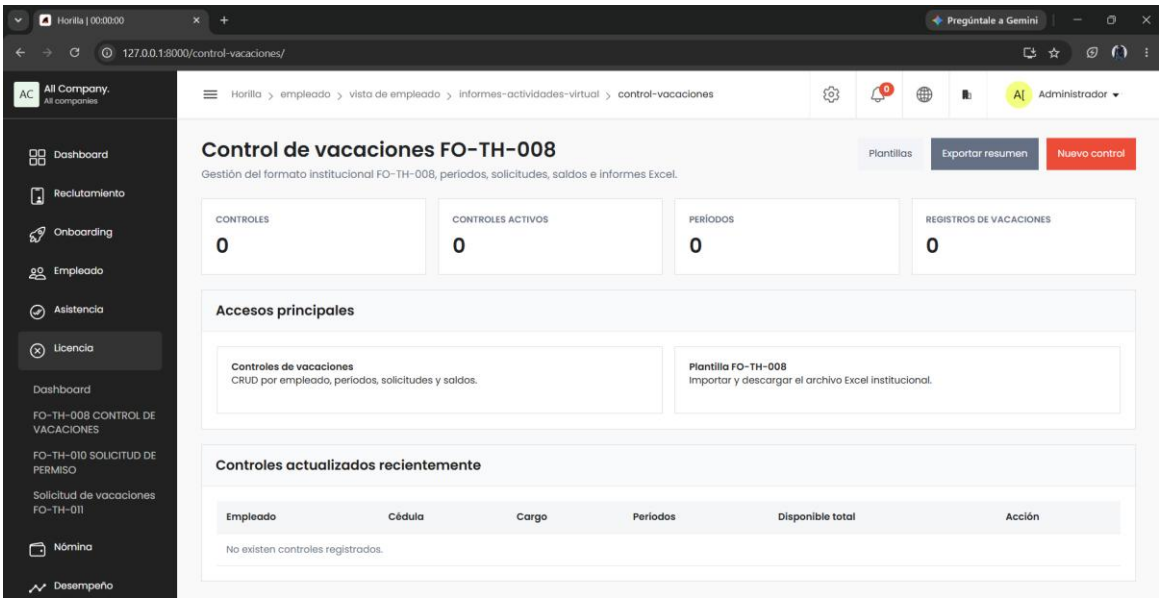


Figura 16. Control de vacaciones FO-TH-008 por empleado, períodos y saldos.

Nota. Autoría propia.

4.6.6 Módulo de Asistencia — Informe de actividades virtual (FO-TH-016) y datos biométricos

La Figura 17 demuestra la funcionalidad de registro, envío, recepción y seguimiento del informe de actividades bajo modalidad virtual, complementada con la información biométrica. Satisface RF06. Los actores principales son el Empleado y el Responsable de Talento Humano. El resultado es un informe FO-TH-016 almacenado con sus actividades, evidencias y estado de revisión.

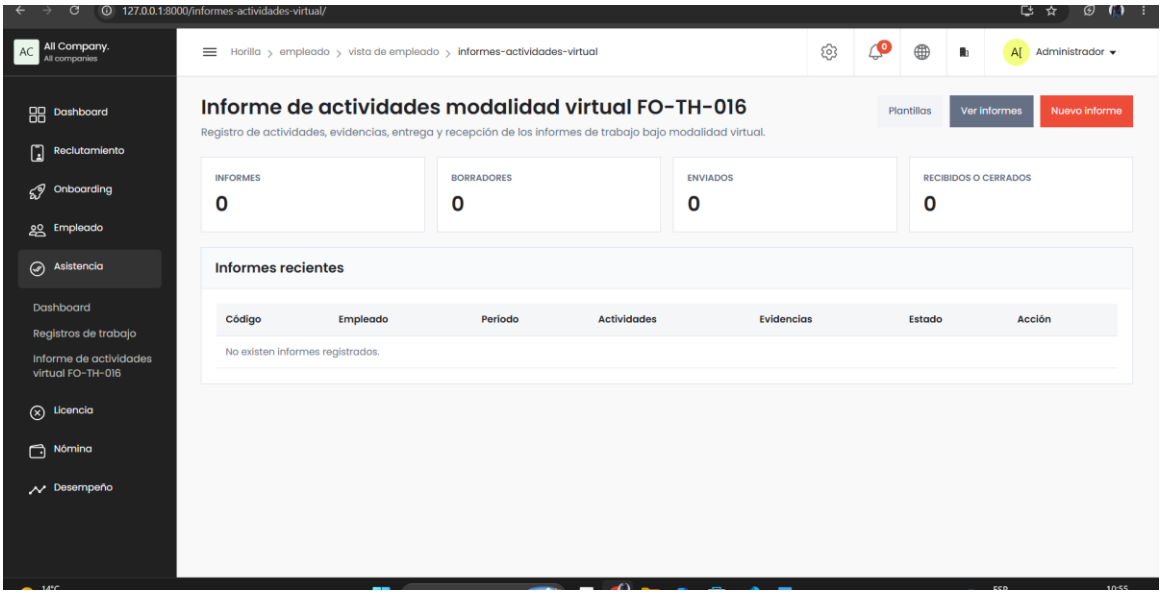


Figura 17. Informe de actividades bajo modalidad virtual FO-TH-016.

Nota. Autoría propia. Captura del FO-TH-016; no constituye evidencia de importación o sincronización biométrica. La integración del dispositivo se limita a conectividad e integración preliminar.

4.6.7 Módulo de Desempeño — Evaluación por competencias (FO-TH-007)

La Figura 18 demuestra la funcionalidad de importación de la matriz institucional, vinculación del personal y registro de evaluaciones por competencias. Satisface RF09. Los actores principales son el Jefe departamental y el Responsable de Talento Humano. El resultado es una evaluación asociada al empleado, con competencias, calificaciones y resumen consultable.

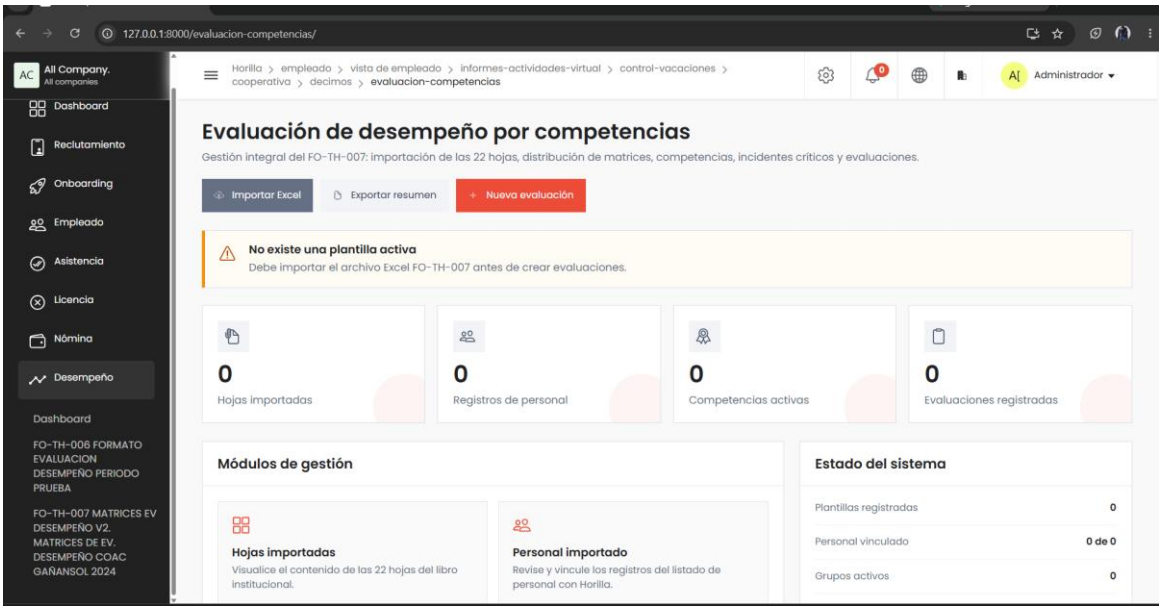


Figura 18. Implementación inicial de funciones básicas del FO-TH-007.

Nota. Autoría propia. Implementación inicial del FO-TH-007, limitada a sus funcionalidades básicas. Los botones visibles no acreditan la implementación o validación de todas las opciones; las funciones adicionales quedan a futuro.

4.6.8 Módulo de Nómina — Solicitud de acumulación o mensualización de décimos (FO-TH-003)

La Figura 19 demuestra la funcionalidad de registro, edición, eliminación, descarga e importación de solicitudes de acumulación o mensualización de décimos. Satisface RF10. Los actores son el Empleado solicitante y el Responsable de Talento Humano. El resultado es una solicitud FO-TH-003 vinculada al empleado y al año correspondiente, disponible en el formato Word institucional.

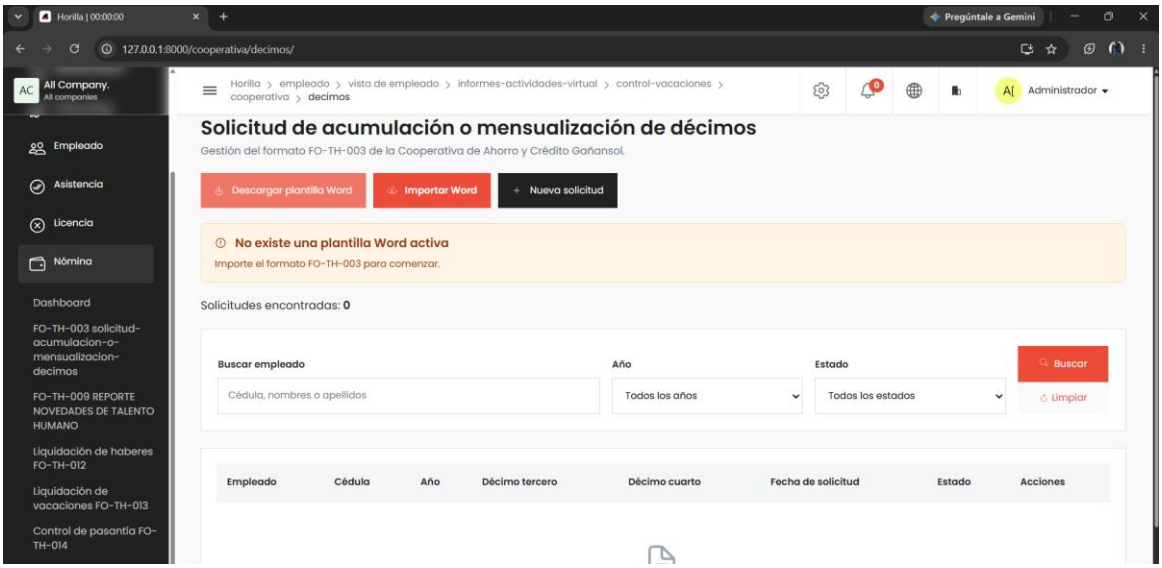


Figura 19. Solicitud de acumulación o mensualización de décimos FO-TH-003.

Nota. Autoría propia.

4.6.9 Resumen de operaciones CRUD verificadas por módulo

En todos los módulos presentados se comprobó de forma manual el ciclo completo de operaciones sobre los registros: creación mediante formularios institucionales, consulta con filtros de búsqueda, edición de los datos existentes y eliminación o desactivación de registros, respetando en cada caso los permisos asignados al rol del usuario. La siguiente lista resume el alcance CRUD verificado por módulo:

Reclutamiento (FO-TH-015): creación, edición, eliminación y consulta de fichas de entrevista; generación de plantilla institucional.

Onboarding (FO-TH-004): creación, edición, eliminación y consulta de registros de inducción/reinducción; descarga e importación de plantilla Word.

Empleado (FO-TH-001): creación, edición, eliminación, filtrado y agrupación de registros de personal; importación y exportación de la base de datos.

Licencia (FO-TH-008): creación, edición y eliminación de controles de vacaciones por periodo; exportación de resumen e importación de plantilla Excel.

Asistencia (FO-TH-016): creación, edición, envío y cierre de informes de actividades virtuales; importación de registros biométricos.

Desempeño (FO-TH-007): importación de matrices Excel, vinculación de personal, creación y consulta de evaluaciones por competencia.

Nómina (FO-TH-003): creación, edición y eliminación de solicitudes de décimos; descarga e importación de plantilla Word.

Active Directory: creación, edición, eliminación y restablecimiento de contraseña de cuentas institucionales, descrito en el apartado 4.5.3.5.

Departamentos, cargos y contratos: creación, edición, eliminación y consulta de departamentos, cargos y contratos vinculados a la ficha del empleado; validación de dependencias antes de eliminar un registro relacionado.

Permisos (FO-TH-010): creación, edición, eliminación y consulta de solicitudes de permiso; control de fechas, motivos y autorización según el rol.

Vacaciones (FO-TH-011 y FO-TH-013): creación, edición, eliminación y consulta de solicitudes y liquidaciones; cálculo de días disponibles y control de traslapes con los saldos del FO-TH-008.

Evaluación en período de prueba (FO-TH-006): creación, calificación, edición, eliminación y consulta de evaluaciones; descarga del formato institucional sobre 100 puntos.

Evaluación del desempeño (FO-TH-007): consulta del listado de personal para evaluación, que integra la hoja institucional, los empleados registrados y las cuentas de Active Directory sin duplicar personas.

Asistencia y dispositivos biométricos: creación, edición, eliminación y prueba de conexión de los dispositivos registrados; importación de las marcaciones al módulo de asistencia.

Nómina y liquidaciones (FO-TH-009 y FO-TH-012): creación, edición, aprobación, anulación, eliminación y consulta de liquidaciones de haberes y novedades; exportación del resumen e importación de plantilla.

Pasantías (FO-TH-014): creación, edición, eliminación y consulta de controles de pasantía con institución formadora, tutor, jornadas y actividades; importación y descarga del formato institucional.

Capacitación (FO-TH-017): creación, edición, eliminación y consulta de planes anuales, con registro de versiones, responsables, aprobaciones y control de cambios.

Activos, documentos y reportes institucionales: registro, asignación y consulta de activos entregados al personal; carga, renombrado, descarga y eliminación de documentos institucionales, y generación de los reportes disponibles en cada módulo.

La verificación de estas operaciones siguió el mismo procedimiento en todos los módulos, debido a que el sistema fue construido sobre una estructura común de vistas, formularios y plantillas del framework Django, en la cual cambian los campos y las reglas de negocio, pero no el flujo de creación, consulta, edición y eliminación. Registrar una captura por cada operación de cada módulo implicaría más de doscientas imágenes que repetirían la misma secuencia de pantallas, lo que extendería el documento sin aportar información nueva y resultaría

redundante. Por esta razón, la evidencia gráfica se presenta de forma representativa y no exhaustiva.

El respaldo visual se organiza en tres niveles. En el cuerpo del capítulo, las Figuras 10 a 21 documentan las pantallas principales de los módulos desarrollados y la integración con Active Directory. En el Anexo C, las Figuras A4 a A11 muestran el ciclo CRUD completo aplicado al módulo de reclutamiento FO-TH-015, desde el registro y su confirmación hasta la consulta en el listado, la edición, la descarga del documento institucional y la disponibilidad de la acción de eliminación. En el Anexo D, las Figuras A12 a A20 presentan los módulos que no cuentan con figura en este capítulo: control de vacaciones FO-TH-008, dispositivos biométricos, informes de modalidad virtual FO-TH-016, control de pasantía FO-TH-014, liquidación de haberes FO-TH-012, plan anual de capacitación FO-TH-017, evaluación en período de prueba FO-TH-006, listado de personal para evaluación y gestión de documentos institucionales.

En cada una de estas pantallas se observan los registros almacenados y los botones de consulta, descarga, edición y eliminación habilitados según el rol del usuario, lo que evidencia que el ciclo CRUD se encuentra implementado y operativo en los módulos correspondientes. Las operaciones que no cuentan con una imagen específica siguen el mismo flujo ya ilustrado y están consignadas en los casos de prueba de la Tabla 15 y en la matriz de trazabilidad de la Tabla 9, de modo que el resumen de este apartado queda respaldado en su totalidad por la evidencia presentada en el capítulo y en los anexos.

4.7 Evidencias de la integración con Active Directory

En este apartado se presentan las evidencias correspondientes a la configuración del servidor de Active Directory de la cooperativa y a su relación con el sistema web. Las capturas

respaldan la integración descrita en el apartado 4.5.3.5 y corresponden al entorno local de pruebas.

Las evidencias permiten observar la consola de administración del directorio y la correspondencia entre las cuentas institucionales y los registros de empleados gestionados en el sistema web. La validación funcional de las operaciones administrativas se realizó de acuerdo con los permisos asignados al usuario responsable.

La Figura 20 demuestra la funcionalidad de consulta y administración de cuentas, grupos y unidades organizativas del directorio institucional. Satisface RF19 y RF03. El actor principal es el Administrador del sistema del área de Tecnologías de la Información. El resultado es la visualización de la estructura del directorio y la ejecución controlada de operaciones sobre las cuentas autorizadas.

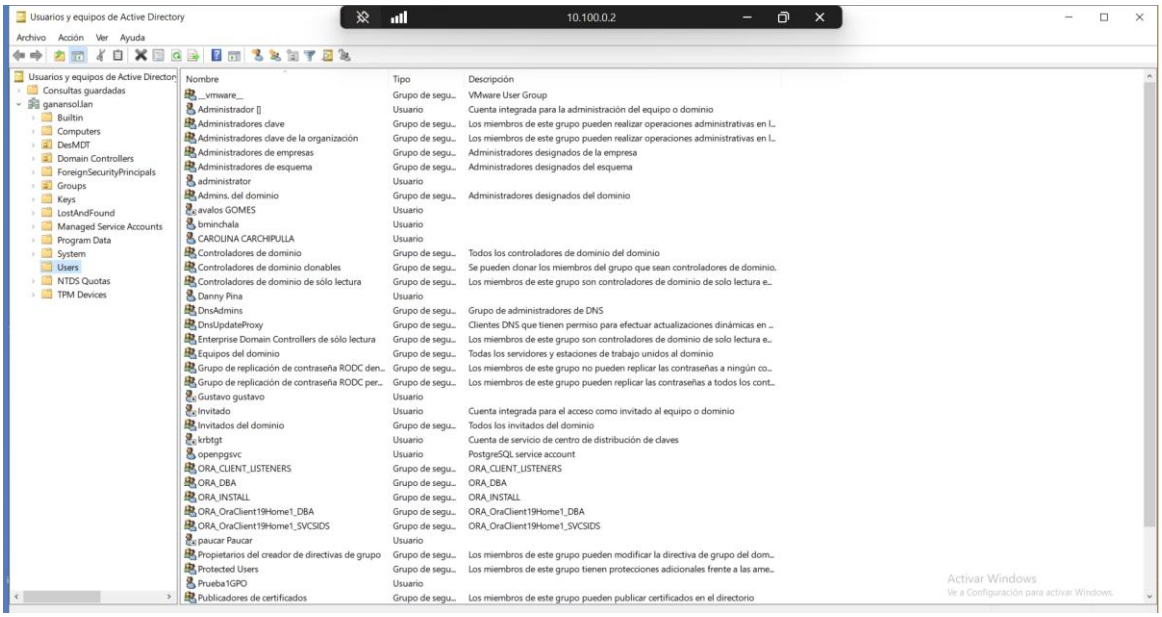


Figura 20. Consola de administración del servidor de Active Directory.

Nota. Autoría propia. La consola pertenece al servidor de Active Directory. La captura muestra las cuentas institucionales, pero no acredita operaciones CRUD ejecutadas desde el sistema web; el alcance comprende su visualización y vinculación.

La Figura 21 demuestra la funcionalidad de correspondencia entre las cuentas institucionales y las fichas de empleados. Satisface RF04 y RF19. Los actores que verifican esta información son el Administrador del sistema y el Responsable de Talento Humano. El resultado es la identificación de la relación entre el usuario del directorio y el empleado registrado, lo que permite comprobar la consistencia de los datos.

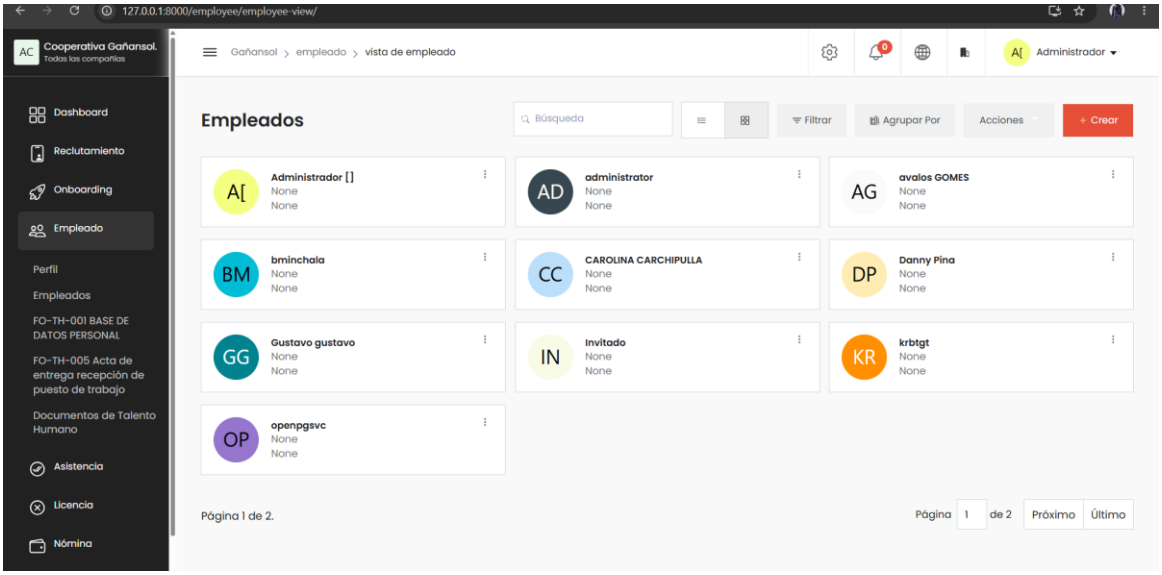


Figura 21. Registros de empleados relacionados con las cuentas institucionales.

Nota. Autoría propia.

4.8 Discusión

Los resultados obtenidos en el desarrollo y la validación del sistema web se contrastan a continuación con los antecedentes revisados, con el fin de identificar coincidencias, diferencias y consecuencias prácticas del trabajo realizado.

En relación con la centralización de la información, los resultados coinciden con lo señalado por Gaibor (2025) y por Tomalá Lino (2025), quienes sostienen que los sistemas de información aplicados a la gestión del personal reducen tiempos de procesamiento, disminuyen errores y mejoran la trazabilidad de los datos. En la COAC Gañansol, los diecisiete formatos institucionales que se administraban en hojas de cálculo y registros físicos quedaron integrados en una sola plataforma con módulos interrelacionados, lo que eliminó la duplicación de registros y la exportación manual de las marcaciones del biométrico. Existe, sin embargo, una diferencia metodológica relevante: en aquellos estudios la mejora se sustenta en indicadores medidos antes y después de la implementación, mientras que en este trabajo la cooperativa no contaba con un sistema previo que sirviera de línea base, por lo que la mejora se evidencia a través de la verificación funcional de los veintiún casos de prueba y no mediante indicadores cuantitativos de desempeño.

Respecto del aporte de la inteligencia artificial, Ncube et al. (2025) concluyen que su incorporación mejora la eficiencia organizacional mediante la automatización de tareas, el análisis de información y el apoyo a la toma de decisiones en la gestión de recursos humanos. El presente trabajo coincide en el efecto de automatización alcanzado, pero difiere en el lugar que ocupa la inteligencia artificial: en el estudio citado forma parte del producto que asiste al área de recursos humanos, mientras que aquí se empleó en el proceso de construcción del software, mediante técnicas de Vibe Coding, y el sistema entregado automatiza tareas administrativas sin

incorporar componentes inteligentes. Esta diferencia delimita el alcance de los resultados y abre una línea de trabajo futura orientada a incorporar analítica o apoyo a la decisión dentro de la propia aplicación.

Los indicadores de la Tabla 14 permiten contrastar el uso del Vibe Coding con la literatura especializada. La tasa de aceptación observada, cercana a seis de cada diez propuestas y siempre con adaptaciones, concuerda con lo advertido por Hou et al. (2024), quienes señalan que las soluciones generadas mediante inteligencia artificial pueden presentar errores lógicos, prácticas inadecuadas o vulnerabilidades y deben revisarse, probarse y adaptarse antes de incorporarse a una aplicación. De igual manera, el resultado respalda a Tosi (2024) cuando sostiene que la rapidez en la producción de código no garantiza su corrección ni su integración con los demás módulos: en este proyecto ningún componente asistido se integró sin superar previamente la prueba funcional correspondiente.

La consecuencia teórica de este hallazgo es que el aporte de las herramientas generativas se expresa como una reducción del esfuerzo de escritura inicial y como apoyo en la identificación de errores, y no como una sustitución del criterio técnico del desarrollador. La consecuencia práctica es que la supervisión humana no constituye una recomendación opcional, sino una condición del proceso: cuatro de cada diez propuestas fueron descartadas y la totalidad de las incorporadas requirió ajustes a las convenciones de nombres y a la estructura de las aplicaciones del proyecto.

En cuanto a la pregunta de investigación planteada, los resultados permiten afirmar que el desarrollo del sistema web contribuyó a optimizar los procesos administrativos del personal, puesto que centralizó la información, estableció controles de acceso por rol, automatizó cálculos y validaciones que antes se realizaban manualmente y habilitó la generación de los documentos

institucionales desde la propia plataforma. Estos resultados deben interpretarse considerando que la validación se efectuó en un entorno local de pruebas y con la participación de dos informantes clave, por lo que su alcance es institucional y no generalizable a otras cooperativas.

De lo anterior se derivan tres líneas de trabajo futuro: medir los tiempos de ejecución y la satisfacción de los usuarios una vez implantado el sistema en producción, evaluar su efecto sobre la carga operativa del área mediante indicadores comparables con los antecedentes citados, e incorporar funciones analíticas que apoyen la toma de decisiones del área de Talento Humano.

CONCLUSIONES

El análisis de los fundamentos teóricos estableció la base conceptual del sistema y aportó los criterios con los que se seleccionaron el lenguaje, el framework, el motor de base de datos y el patrón arquitectónico. Estas decisiones, sintetizadas en la Tabla 11, justificaron la arquitectura basada en Python, Django 4.2, MySQL 8.0 y el patrón Modelo-Vista-Plantilla.

La evaluación de las metodologías de desarrollo permitió estructurar el proceso de construcción mediante Scrum, cuya organización en seis sprints facilitó obtener incrementos revisables e incorporar oportunamente las observaciones institucionales. El Vibe Coding resultó un apoyo útil para interpretar errores, generar componentes preliminares y documentar, siempre bajo revisión humana obligatoria.

El desarrollo del sistema web se completó conforme a las fases de levantamiento de requerimientos, modelado, construcción y pruebas en entorno local. Se definieron 23 requerimientos funcionales, 21 historias de usuario y 24 casos de uso, y se ejecutaron 21 casos de prueba con resultado satisfactorio, incluida la integración con Active Directory, lo que permitió obtener una versión funcional lista para su adaptación e implementación.

Conforme se argumentó en la Discusión, estos resultados coinciden con Gaibor (2025) y Tomalá Lino (2025) respecto de los beneficios de centralizar la información del personal, difieren de Ncube et al. (2025) en el lugar que ocupa la inteligencia artificial, empleada aquí en el proceso de construcción y no en el producto, y confirman lo advertido por Hou et al. (2024) y Tosi (2024), pues ningún componente asistido se integró sin superar su prueba funcional.

El trabajo presenta limitaciones que acotan sus resultados: la información se levantó con dos informantes clave, los instrumentos no fueron validados por un tercero externo, no se

midieron formalmente los tiempos de respuesta exigidos por el RNF08, la validación se realizó en un entorno local y la solución se construyó sobre la base tecnológica de Horilla indicada en la Tabla 11. El trabajo futuro debería orientarse a implantar el sistema en producción, medir tiempos y satisfacción de los usuarios e incorporar funciones analíticas de apoyo a la decisión.

RECOMENDACIONES

Realizar el despliegue del sistema en la infraestructura tecnológica institucional siguiendo un plan formal de puesta en producción que contemple servidor web (Gunicorn o uWSGI), servidor de aplicaciones (Nginx), respaldo periódico de la base de datos y monitoreo básico de disponibilidad.

Evaluar la conexión directa en tiempo real con el reloj biométrico y consolidar en producción la integración con el servidor de Active Directory ya comprobada en el entorno local, con el fin de eliminar por completo la carga manual de archivos y reducir el margen de error humano.

Complementar las pruebas funcionales con pruebas automatizadas mediante el framework de testing de Django (unittest o pytest-django), estableciendo una cobertura mínima sobre las vistas críticas y las reglas de cálculo de nómina y liquidaciones.

Formalizar las políticas de acceso, uso y respaldo de la información personal del sistema en coherencia con la normativa ecuatoriana de protección de datos, definiendo periodos de retención y responsables por área.

Capacitar al personal de Talento Humano y a los administradores del sistema mediante un plan que incluya el manual de usuario final, sesiones prácticas por módulo y un canal formal para reportar incidencias.

Continuar aplicando Vibe Coding en fases posteriores manteniendo el protocolo de revisión humana, y documentar los prompts que resultaron efectivos para constituir una base de conocimiento reutilizable en futuros mantenimientos.

BIBLIOGRAFÍA

González, C. A., & Larios , D. A. (2021). *Desarrollo de un sistema web transaccional de facturación para la "Ferretería El Progreso", ubicado en la ciudad de Masatepe, departamento de Masaya en el segundo semestre de 2020.* Masatepe Nicaragua.: Universidad Nacional Autónoma de Nicaragua. Obtenido de https://repositorio.unan.edu.ni/id/eprint/16554/1/Gon%202021.pdf

Mayorga , L. C., Bermeo, O. X., Riccardi, G. A., & Guevara , V. I. (2022). Sistema web para los procesos administrativos y de producción en viveros del Cantón Milagro. *revistaingenieria*, 200–213. doi:10.33996/revistaingenieria.v6i16.100

Abuhantash, A. (2023). The Impact of Human Resource Information Systems on Organizational Performance: A Systematic Literature Review. *European Journal of Business and Management Research*, 239-247. Obtenido de https://eu-opensci.org/index.php/ejbmr/article/view/1992

Aguilera, J., Martínez, J., Olmos, S., Padilla, J., Sámano, V., & Bravo, M. (2022). Development of a Web Application for the Detection of Coronary Artery Calcium from Computed Tomography. *Applied Sciences*, 12281. Obtenido de https://doi.org/10.3390/app122312281

Alami, A., & Krancher, O. (2022). How Scrum Adds Value to Achieving Software Quality? *Journal of Empirical Software Engineering*, 1–68.

Alazzawi, A., Yas, Q., & Rahmatullah, B. (2023). A Comprehensive Review of Software Development Life Cycle Methodologies: Pros, Cons, and Future Directions. *International Journal of Computing Sciences and Management (IJCSM)*, 1-19. Obtenido de https://doi.org/10.52866/ijcsm.2023.04.04.014

Alenezi, M., & Akour, M. (2025). AI-Driven Innovations in Software Engineering: A Review of Current Practices and Future Directions. *Applied Sciences*, 1344. Obtenido de https://doi.org/10.3390/app15031344

Alshammari, A., Alqahtani, M., & Alharbi, F. (2024). Exploring the Impact of Electronic Human Resource Management (E-HRM) on Organizational Performance: A Systematic Literature Review. *Discover Sustainability*, 212. Obtenido de https://doi.org/10.1007/s43621-024-00212-7

Araujo, C., & Román, E. B. (2022). *Diseño e implementación de un sistema web para manejo y seguimiento de proyectos en el Comité de Ética de Desarrollo e Investigación en Seres Humanos (CEISH UCE) de la Universidad Central del Ecuador.* Quito: Universidad Central del Ecuador. Obtenido de

https://www.dspace.uce.edu.ec/server/api/core/bitstreams/edbea1ab-0578-4e1f-8b88-301c6b194a57/content

Assunção, W., Krüger, J., Mosser, S., & Selaoui, S. (2023). How Do Microservices Evolve? An Empirical Analysis of Changes in Open-Source Microservice Repositories. *The Journal of Systems and Software*, 111738. Obtenido de https://pdf.sciencedirectassets.com/271629/1-s2.0-S0164121223X00077/1-s2.0-S0164121223001838/main.pdf?X-Amz-Security-Token=IQoJb3JpZ2luX2VjEAoaCXVzLWVhc3QtMSJGMEQCICC%2FJdGUkAElUrMautJoWN6wbTUoL%2F3AAUC0KuZrqmdUAiA%2FZNXvXKL1spg%2Bv1Si%2FbI8FWc2S9uH9Cjqbc

Assunção, W., Krüger, J., Mosser, S., & Selaoui, S. (2023). The Journal of Systems and Software. 4–8. Obtenido de https://www.sciencedirect.com/journal/journal-of-systems-and-software

Avelino, A. J., & Montoya, L. R. (2023). *Aplicación web y móvil para la gestión de nómina del área de talento humano en la empresa "Consultax Legal S.A.* Guayaquil: Universidad Agraria del Ecuador. Obtenido de https://cia.uagraria.edu.ec/Archivos/AVELINO%20QUIMI%20ANTHONY%20JOEL.pdf

Azimov, A. (2021). *Design and Implementation of web application by using Python Django Framework.* Prague: Czech University of Life Sciences Prague. Obtenido de https://theses.cz/id/noj5xp/zaverecna_prace.pdf?lang=en

Baharuddin, A., & Abdullah, Z. (2026). Employee Management System for SMK Ilhami Aikmel with Role-Based Access Control (RBAC). *Applied Information Technology and Computer Science (AITCS)*, 733–752. Obtenido de https://doi.org/10.30880/aitcs.2025.06.02.040

Banh, L., Holldack, F., & Strobel, G. (2025). Copiloting the Future: How Generative AI Transforms Software Engineering. *Information and Software Technology*, 107751. Obtenido de https://doi.org/10.1016/j.infsof.2025.107751

Ben Kora, H., & Manita, M. (2024). Arquitectura web front-end moderna con React.js y Next.js. *University of Zawia Journal of Engineering Science and Technology (UZJEST)*, 1-13. Obtenido de https://journals.zu.edu.ly/index.php/UZJEST/article/view/55/684

Bohórquez , M. T. (2024). *Sistema web para el registro y control de asistencia.* Quito: Pontificia Universidad Católica. Obtenido de https://repositorio.puce.edu.ec/server/api/core/bitstreams/dfb22783-ac2d-4df1-ab1e-a0ca7aaee346/content

Bombini, A., Alkhansa, A., Cappelli, L., Felicetti, A., Giacomini, F., & Costantini, A. (2022). A Cloud-Native Web Application for Assisted Metadata Generation and Retrieval: THESPIAN-NER. *Applied Sciences*. Obtenido de https://doi.org/10.3390/app122412910

Bucko, A., Vishi, K., Krasniqi, B., & Rexha, B. (2023). Enhancing JWT Authentication and Authorization in Web Applications Based on User Behavior History. *Computers*, 78. Obtenido de https://www.mdpi.com/2073-431X/12/4/78

Bucko, A., Vishi, K., Krasniqi, B., & Rexha, B. (2023). Enhancing JWT Authentication and Authorization in Web Applications Based on User Behavior History. *Computers*, 78. Obtenido de https://doi.org/10.3390/computers12040078

Bushong, V., Abdelfattah, A., Maruf, A., Das, D., Lehman, A., aroszewski , E., . . . Bures, M. (2021). A Systematic Literature Review of Mobile Application Testing Methods and

Tools. *Applied Sciences*, 7856. Obtenido de https://www.mdpi.com/2076-3417/11/17/7856

Chamba Pinta, K. S. (2026). *Aplicación Web con análisis predictivo para la gestión de pedidos en la ferretería Ferritools's.* Universidad Técnica de Ambato. Obtenido de https://repositorio.uta.edu.ec/server/api/core/bitstreams/d29f41ac-3c3e-4faf-aa84-a4ffaa2a2331/content

Córdova Poveda, J. R. (2024). *Aplicación web utilizando react y nodejs para la gestión y seguimiento académico en el Departamento de Consejería Estudiantil de la Unidad Educativa Bolívar.* Ambato: Universidad Tecnica de Ambato. Obtenido de https://repositorio.uta.edu.ec/server/api/core/bitstreams/6d6204ee-646a-4440-aee9-b63ee0e4dd77/content

Córdova Poveda, J. R. (2024). *Aplicación web utilizando react y nodejs para la gestión y seguimiento académico en el Departamento de Consejería Estudiantil de la Unidad Educativa Bolívar.* Ambato: Universidad Tecnica de Ambato. Obtenido de https://repositorio.uta.edu.ec/server/api/core/bitstreams/6d6204ee-646a-4440-aee9-b63ee0e4dd77/content

Cueva Jara, C. G. (2023). *La Gestión del Talento Humano y el Desempeño Laboral del Hospital.* UNIVERSIDAD NACIONAL DE CHIMBORAZO, Ribamba. Obtenido de http://dspace.unach.edu.ec/bitstream/51000/10717/1/Cueva%20Jara%2c%20C.%282023%29%20La%20Gesti%c3%b3n%20del%20Talento%20Humano%20y%20el%20Desempe%c3%b1o%20Laboral%20del%20Hospital%20de%20Especialidades%20San%20Juan%20HOSPIESAJ.S.A.%20de%20Riobamba..pdf

Diebold, P. (2025). From Backlogs to Bots: Generative AI's Impact on Agile Role Evolution. *Journal of Software: Evolution and Process*, 37. Obtenido de https://doi.org/10.1002/smr.2740

Dimopoulos, A., Chryssinas, G., Mavroforaki, D., Gan, T. H., & Chatzakos, P. (2024). An Interactive Web-Based Platform for Support Generation and Optimisation for Metal Laser Powder Bed Fusion. *Materials*, 1639. Obtenido de https://doi.org/10.3390/ma17071639

Ehsan, A., Abuhaliqa, M. A., Catal, C., & Mishra, D. (2022). RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions. *Applied Sciences*, 12. Obtenido de https://www.mdpi.com/2076-3417/12/9/4369

Ehsani, S. (2023). *Development of a web application for data analysis.* Politecnico di Torino. Obtenido de https://theses.cz/id/noj5xp/zaverecna_prace.pdf?lang=en

Escobar Pullutasig, J. D. (2025). *Aplicación web utilizando .net core para la gestión de la información de empleados en la empresa Ciudad del Auto Cia Ltda.* Ambato: Universidad Tecnica de Ambato. Obtenido de https://repositorio.uta.edu.ec/items/f034264a-a1cd-4694-abd9-f3056d8c338a

Esquive, W., & López , G. (2021). Paralelismos entre bases de datos relacionales y no relacionales (un enfoque en seguridad. *ReCIBE. Revista electrónica de Computación, Informática, Biomédica y Electrónica*, 1–16. Obtenido de https://www.redalyc.org/articulo.oa?id=512269058002

Fan, A., Gokkaya, B., Harman, M., Lyubarskiy, M., Sengupta, S., Yoo, S., & Zhang, J. (2023). Large Language Models for Software Engineering: Survey and Open Problems. (págs. 31–53).

Melbourne, Australia: IEEE. Obtenido de https://doi.org/10.1109/ICSE-FoSE59343.2023.00008

Fan, A., Gokkaya, B., Harman, M., Lyubarskiy, M., Sengupta, S., Yoo, S., & Zhang, J. (2024). Large Language Models for Software Engineering: Survey and Open Problems. 31–53. Obtenido de https://www.semanticscholar.org/reader/4fbd174e80502f8f3c4b9f48054872b028e2445a

Flores LLangari, F. S. (2026). *Aplicativo multiplataforma basado en arquitectura MVC orientado a la gestión de la información del "Club deportivo especializado formativo Atahualpa" de la ciudad de Ambato.* Universidad Técnica de Ambato. Obtenido de https://repositorio.uta.edu.ec/server/api/core/bitstreams/f89b02d2-413b-479e-bd71-949c9d320f34/content

Gaibor Villacís, C. A. (2025). *Aplicación web con chatbot para la digitalización de la gestión de tiempo y asistencia laboral en talento humano de la empresa pública Empresa Mercado Mayorista Ambato.* Ambato: UNIVERSIDAD TÉCNICA DE AMBATO. Obtenido de https://repositorio.uta.edu.ec/server/api/core/bitstreams/5f073383-aaeb-4ad4-b36a-09d19feb2470/content

Gallador, A. A., & González, G. S. (2022). *Aplicativo web y app para la gestión de recursos humanos, control de marcaciones y bitácora digital en S.I.P Security.* Guayaquil: Universidad Agraria del Ecuador. Obtenido de https://cia.uagraria.edu.ec/Archivos/GALLARDO%20VIVERO%20ABEL%20ANDRES.pdfÑ

Ghimire, D., & Charters, S. (2022). The Impact of Agile Development Practices on Project Outcomes. *Software*, 265–275. Obtenido de https://doi.org/10.3390/software1030012

Gómez Sánchez, M. D., Herrera Herrera, D. H., Pavón Domínguez , C. E., & Jacqueline Vanessa, Q. T. (2024). *Diseño de un modelo de detección de datos duplicados mediante procesamiento del lenguaje natural para optimizar la eficiencia en la gestión de datos de un laboratorio farmacéutico.* Quito: Universidad: Universidad Internacional del Ecuador. Obtenido de https://repositorio.uide.edu.ec/handle/37000/6514

Grijalva, I., Adams, H. B., Clark, N., & McCornack, B. (2024). Detecting and counting sorghum aphid alates using smart computer vision models. *Ecological Informatics*, 102540. Obtenido de linkinghub.elsevier.com/retrieve/pii/S1574954124000827

Guerra Jordán, M. D. (2026). *Sistema de control y gestión remota para el registro de asistencia basado en biometría portátil para la empresa Megaeléctric en sus obras temporales.* Ambato: Universidad Técnica de Ambato. Obtenido de https://repositorio.uta.edu.ec/server/api/core/bitstreams/e24bedd5-5c46-41bf-a021-238f4bc13e92/content

Hemdev, N. (2025). Vibe Coding: A Mixed-Methods Case Study on Conversational AI Programming and Application Development. *International Journal on Science and Technology (IJSAT)*, 1–19. Obtenido de https://www.ijsat.org/papers/2025/3/7765.pdf

Heredia, S. R., Narváez , E. Y., & Narváez , M. E. (2024). Sistema web de gestión administrativa para PYMES utilizando la tecnología Mean Stack. *revistas.upse*, 57–67. Obtenido de https://www.revistas.upse.edu.ec/index.php/rctu/article/view/833/658

Hoda, R., Dam, H., Tantithamthavorn, C., Thongtanunam, P., & Storey, M.-A. (2023). Augmented Agile: Human-Centered AI-Assisted Software Management. *IEEE Software, 40*(4), 106-109. doi:10.1109/MS.2023.3268725

Hou, X., Zhao, Y., Liu, Y., Yang, Z., Wang, K., Li, L., . . . Wang, H. (2024). Large Language Models for Software Engineering: A Systematic Literature Review. *ACM Transactions on Software Engineering and Methodology*, 1–79. Obtenido de https://dl.acm.org/doi/epdf/10.1145/3695988

Howard, J. S., Kole, E. F., Okanlawon, S., & Yadav, K. (2025). Advanced website development using HTML CSS and JavaScript. *Innovation and Integrative Research Center Journal, 3*, 955–964. Obtenido de https://iircj.org/wp-content/uploads/99-Advanced-Website-development-using-HTML-CSS-and-JavaScript.pdf

Hron, M., & Obwegeser, N. (2022). Why and How is Scrum Being Adapted in Practice: A Systematic Review. *Journal of Systems and Software*, 111110. Obtenido de https://pure.au.dk/ws/files/326714625/Hron_2022_Why_and_how_is_scrum_adapted_in_practice.pdf

Intriago, A. J. (2023). *Desarrollo de una interfaz web de una herramienta de análisis de sentimientos para tweets (con Twitter).* Madrid: Universidad Politécnica. Obtenido de https://oa.upm.es/75388/1/TFG_ANTHONY_JHOAN_INTRIAGO_CAMACHO.pdf

Juma , A. P., Pusda , A. R., & Varona, F. (2025). Gestión del talento humano: Aplicativo web para el análisis del desempeño laboral. *Revista Científica Dejando Huellas*, 63–76. Obtenido de https://liceoaduanero.edu.ec/revista/index.php/ojs/article/view/12/10

Juma, P. A., Pusda , R. A., & Alvarez , F. (2025). Gestión del talento humano: Aplicativo web para el análisis del desempeño laboral. *Revista Científica Dejando Huellas*, 63–76. doi:https://doi.org/10.65100/recidh/12

Jurado Acurio, B. S. (2024). *Metodología híbrida para estandarizar el desarrollo de aplicaciones web, aplicando Normas ISO/IEC 25000 en la Empresa Eléctrica Ambato S.A. (EEASA).* Ambato: Universida Tecnica de Ambato. Obtenido de https://repositorio.uta.edu.ec/server/api/core/bitstreams/47ab8db2-f6da-4dd4-b1a7-7e4283c47286/content

Karlsson, S., Jongeling, R., Čaušević, A., & Sundmark, D. (2024). Exploring Behaviours of RESTful APIs in an Industrial Setting. *Software Quality Journal*, 1287–1324. Obtenido de https://doi.org/10.1007/s11219-024-09686-0

Khan, W., Kumar, T., Zhang, C., Raj, K., Roy, A., & Luo, B. (2023). SQL and NoSQL Database Software Architecture Performance Analysis and Assessments A Systematic Literature Review. *Big Data and Cognitive Computing*, 97. Obtenido de https://www.mdpi.com/2504-2289/7/2/97

Kirpitsas, I., & Pachidis, T. (2022). Evolution towards Hybrid Software Development Methods and Information Systems Audit Challenges. 316–363. Obtenido de https://doi.org/10.3390/software1030015

Kirpitsas, I., & Pachidis, T. (2022). Evolution towards Hybrid Software Development Methods and Information Systems Audit Challenges. *Software*, 316–363. Obtenido de https://doi.org/10.3390/software1030015

Korzec, A. (2022). *Model-driven Development of Front-Ends for CRUD Applications.* Berlin: Freie Universit¨at Berlin. Obtenido de https://www.inf.fu-berlin.de/inst/ag-se/theses/Korzec22-model-driven-development-of-front-ends.pdf

Kuraś, P., Strzałka, D., Kowal, B., Organiściak, P., Demidowski, K., & Vanivska, V. (2024). REDUCE—A Tool Supporting Inconsistencies Reduction in the Decision-Making Process. *Applied Sciences*, 11465. Obtenido de https://www.mdpi.com/2076-3417/14/23/11465

Kuraś, P., Strzałka, D., Kowal, B., Organiściak, P., Demidowski, K., & Vanivska, V. (2024). REDUCE—A Tool Supporting Inconsistencies Reduction in the Decision-Making Process. *Applied Sciences*, 11465. Obtenido de https://www.mdpi.com/2076-3417/14/23/11465

Leong, J., Yee, K., Baitsegi, O., Palanisamy, L., & Ramasamy, R. (2023). Hybrid Project Management between Traditional Software Development Lifecycle and Agile Based Product Development for Future Sustainability. *Sustainability*, 1121. Obtenido de https://doi.org/10.3390/su15021121

Mayorga, L. C., Riccardi, G. A., Bermeo , O. X., & Guevara, V. I. (2022). Sistema web para los procesos administrativos y de producción en viveros del Cantón Milagro. *Revista de Investigación Ingeniería y sus Alcances*, 200–213. Obtenido de https://www.redalyc.org/pdf/7457/745780966002.pdf

MESKE, C., HERMANNS, T., WEIDEN, E. V., LOSER, K.-U., & THORSTEN, B. (2025). Vibe Coding as a Reconfiguration of Intent Mediation in Software Development: Definition, Implications, and Research Agenda. *IEEE Access*, 213242–213259. Obtenido de https://arxiv.org/pdf/2507.21928

Mixi Joselyne, S. T., Navas Espin, W. R., & Francisco Nicolás, R. V. (2023). *Diseño de un modelo predictivo de aprobación de materias mediante la analítica de datos para la planificación y proyección de cursos por periodo académico en la Carrera de Telemática - Universidad de Guayaquil.* Quito: Universidad: Universidad Internacional del Ecuador. Obtenido de https://repositorio.uide.edu.ec/bitstream/37000/6515/1/UIDE-Q-TMSIG-2023-13.pdf

Narváez, D. J. (Quito). *Desarrollo De Un Sistema Web Informativo De Trámites Académicos Y Administrativos Para La ESFOT.* Quito: Escuela Politécnica Nacional. Obtenido de https://bibdigital.epn.edu.ec/bitstream/15000/22115/1/CD%2011616.pdf

Nazate , L. C. (2023). *Desarrollo de módulo web de administración de negocios y productos para el ecommerce de la "Asociación de Vendedoras de Comida San Miguel de Yahuarcocha" utilizando el framework Laravel.* Ibarra: Universidad Técnica del Norte. Obtenido de https://repositorio.utn.edu.ec/bitstream/123456789/14658/2/04%20ISC%20682%20TESIS.pdf

Ncube, T. R., Sishi, K. K., & Skinner, J. P. (2025). The impact of artificial intelligence on human resource management practices: An investigation. *SA Journal of Human Resource Management*, 1-11. doi:10.4102/sajhrm.v23i0.2960

Negri, C., Geraud, R., Sergeeva, A., & Lenzini, G. (2024). A systematic literature review on the impact of AI models on the security of code generation. *Frontiers in Big Data*, 1386720. Obtenido de https://doi.org/10.3389/fdata.2024.1386720

Ozkan, c., Bal, S., Gurgen, T., & Şahin, M. (2022). Scrum, Kanban or a Mix of Both? A Systematic Literature Review. *Proceedings of the 17th Conference on Computer Science and*

Intelligence Systems (págs. 883–893). Varsovia, Polonia: Polish Information Processing Society. Obtenido de http://dx.doi.org/10.15439/2022F143

Pérez , S. G., Quispe, J. R., & Mullicundo, F. F. (Edits.). (2021). Herramientas y tecnologías para el desarrollo web desde el FrontEnd al BackEnd. *XXIII Workshop de Investigadores en Ciencias de la Computación* (págs. 347–350). Chilecito, La Rioja, Argentina: Universidad Nacional de Jujuy. Obtenido de https://sedici.unlp.edu.ar/bitstream/handle/10915/120476/Ponencia.pdf-PDFA.pdf?sequence=1&isAllowed=y

Perry, N., Srivastava, M., Kumar, D., & Boneh, D. (2023). Do Users Write More Insecure Code with AI Assistants? *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS '23)* (págs. 2785–2799). Copenhagen: ACM (Association for Computing Machinery). Obtenido de https://dl.acm.org/doi/10.1145/3576915.3623157

Pipino, C., Rocco, G., Pagano, C., & Cipriano, A. (2024). An Innovative Blockchain-Based System for Human Resources Digitalization with Traceable Relationship Management. (págs. 1–11). Les Ulis, Francia: EDP Sciences. doi:10.1051/itmconf/20246204004

Priyashantha, K., & Chandradasa, A. (2023). Electronic Human Resource Management (e-HRM) Adoption: A Systematic Literature Review. *Sri Lanka Journal of Social Sciences*, 19-37. Obtenido de https://doi.org/10.4038/sljss.v46i1.8573

Putra, P. E., Efendi, R. W., Tamam, A. B., & Pramadi, W. A. (2025). Tren y práctica Terbaik dalam Pengembangan Web Berbasis API: Kajian Literatur terhadap Framework Laravel y React. *Infomatek*, 5–9. Obtenido de https://doi.org/10.23969/infomatek.v27i1.25122

Ramírez, W. E. (2023). Análisis de la gestión del talento humano en el contexto empresarial actual: una revisión bibliográfica. *INNOVA Research Journal*, 83–106. Obtenido de https://revistas.uide.edu.ec/index.php/innova/article/view/2234/2076

Rennhard, M., Kushnir, M., Favre, O., Esposito, D., & Zahnd, V. (2022). Automating the Detection of Access Control Vulnerabilities in Web Applications. *SN Computer Science*, 376. doi:10.1007/s42979-022-01271-1

Rennhard, M., Kushnir, M., Favre, O., Esposito, D., & Zahnd, V. (2022). Automating the Detection of Access Control Vulnerabilities in Web Applications. *SN Computer Science*, 376. Obtenido de https://doi.org/10.1007/s42979-022-01271-1

Rodríguez, M. F., Pérez, J. C., & Gómez, A. (2024). Transformación digital y gestión del talento humano: una revisión sistemática de la literatura. *Revista Científica de Administración y Gestión*. Obtenido de https://portal.amelica.org/ameli/journal/793/7934890004/

Roosevelt Manlio, M. C. (2023). *Implementación de un Sistema Biométrico con huellas dactilares para el control de asistencia del personal docente y administrativo en la unidad educativa fiscal puerto cayo.* manabí: Universidad Estatal del Sur de Manabí. Obtenido de https://repositorio.unesum.edu.ec/bitstream/53000/4779/1/Menendez%20Cevallos%20Roosevelt%20Manlio.pdf

Rosa, G., Afonso, J., Gaspar, P. D., Soares, V., & Caldeira, J. (2024). The Development of a Prototype Solution for Detecting Wear and Tear in Pedestrian Crossings. *Applied Sciences*, 6462. Obtenido de https://doi.org/10.3390/app14156462

Saeedi, K., & Visvizi, A. (2021). Software Development Methodologies, HEIs, and the Digital Economy. *Education Sciences*, 73. Obtenido de https://doi.org/10.3390/educsci11020073

Salazar, A. J., & Cercado, A. S. (2021). *Implementación de aplicación web de ventas para la distribuidora de equipos informáticos Computer Anexos.* GUAYAQUIL: Universidad Politécnica Salesiana. Obtenido de https://dspace.ups.edu.ec/bitstream/123456789/21373/1/UPS-GT003510.pdf

Sangucho, J. P., Ojeda, J. L., Giron, J. P., Bonilla , D. S., & Jara, S. (2024). Análisis de Frameworks Frontend para Aplicar UX/UI en el Desarrollo Web: Una Revisión Sistemática. *Ciencia Latina Revista Científica Multidisciplinar*, 844. Obtenido de https://ciencialatina.org/index.php/cienciala/article/view/11290/16544

Schuster, D., van Zelst, S., & van der Aalst, W. (2023). Cortado: A dedicated process mining tool for interactive process discovery. *SoftwareX*, 101373. Obtenido de https://www.sciencedirect.com/science/article/pii/S2352711023000699?via%3Dihub

Siarry, P., Cheung, S., Jabbar, M., & Li, X. (2026). Proceedings of the 2024 International Conference on Wireless Communications, Networking and Applications. *2024 International Conference on Wireless Communications, Networking and Applications (WCNA2024)* (págs. 1–13). Singapore: Springer Nature Singapore. Obtenido de https://link.springer.com/book/10.1007/978-981-95-6946-5

Silverio Torres, V. j. (2021). *Desarrollo de una aplicación móvil para la promoción de eventos para el departamento de actividades complementarias de la Pontificia Universidad Católica del Ecuador (PUCE). Caso de estudio: proyecto cultural Voces desde la Mitad del Mundo.* Quito: PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR. Obtenido de https://repositorio.puce.edu.ec/items/3e036c13-6f19-4803-81da-cde18d22c19b

Superintendencia de Economía Popular y Solidaria. (2025). *Informe sectorial de riesgos de lavado de activos y otros delitos en el SFPS.* Quito: Superintendencia de Economía Popular y Solidaria. Obtenido de https://www.seps.gob.ec/wp-content/uploads/Informe-Sectorial-de-LA-FD-Resumen-Ejecutivo-2024-SEPS.pdf

Tomalá Lino, K. J. (2025). *Gestión administrativa mediante blockchain para el programa de talento humano de Posgrado UPSE, Cantón La Libertad, año 2025.* La Libertad: Universidad Estatal Península de Santa Elena. Obtenido de https://repositorio.upse.edu.ec/server/api/core/bitstreams/3e1cbfa4-95c6-499d-91e4-44bfbdba4bcb/content

Torres, J. E. (2021). *Implementación de un sistema web para la gestión de la empresa de confitería y artículos plásticos "Don Chuta".* Milagro: Universidad Agraria del Ecuador. Obtenido de https://cia.uagraria.edu.ec/Archivos/TORRES%20SANCHEZ%20JONATHAN%20ELVIS.pdf

Torres, R., & Hora, A. (2026). Assessing Reusable Web Applications: The Django Ecosystem Case. *Software Quality Journal*, 1–35. Obtenido de https://doi.org/10.1007/s11219-026-09745-8

Tosi, D. (2024). Studying the Quality of Source Code Generated by Different AI Generative Engines: An Empirical Evaluation. *Future Internet*, 188. Obtenido de https://www.mdpi.com/1999-5903/16/6/188

Truszkowski, K., & Pańczyk, M. (2022). Comparative analysis of Web application development on Java and PHP. *Journal of Computer Sciences Institute*, 379-383. Obtenido de https://doi.org/10.35784/jcsi.3050

Vivanco Herrera, D. L. (2023). *Análisis del sistema de gestión de talento humano para la empresa florícola Puliza Flowers, Cayambe, provincia de Pichincha.* Ibarra: PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR. Obtenido de https://repositorio.puce.edu.ec/server/api/core/bitstreams/711f35b5-f0f3-42f8-8c3d-5ed6cb1c50bf/content

Zhao, Y., Li, Z., & Lv, S. (2024). Enhancing AI System Privacy: An Automatic Tool for Achieving GDPR Compliance in NoSQL Databases. *Computers, Materials & Continua*, 217–234. Obtenido de https://doi.org/10.32604/cmc.2024.052310

Anexo: Formato del Anteproyecto.

A. TÍTULO
Desarrollo de un sistema web para la gestión de talento humano en la COAC Gañansol utilizando técnicas de Vibe coding

B. DOMINIO, LÍNEA Y ÁMBITOS DE INVESTIGACIÓN			
Tecnologías de Información y Comunicación	**Ciencias exactas, naturales y tecnológicas**	Inteligencia de Negocios	
		Sistemas de Información	
		Gobierno y administración de tecnologías de información	
		Auditoría Informática	
		Seguridad Informática	
		Redes y comunicación	
		Arquitectura de Hardware	
		Arquitectura de desarrollo de software	x
		Ingeniería de Software	
		Gestión y gobierno de proyectos de tecnología informática	

		Ingeniería de requerimientos	
		Algoritmos y programación	
		Ciencias exactas y naturales (Matemáticas, Física, Química, Biología, etc.)	
		Modelaje y simulación	

C. PLANTEAMIENTO DEL PROBLEMA

Las organizaciones financieras, como las cooperativas de ahorro y crédito, requieren una gestión eficiente del talento humano que permita administrar adecuadamente la información del personal, optimizar los procesos administrativos y fortalecer la toma de decisiones institucionales. En la actualidad, la transformación digital ha impulsado la adopción de sistemas de información capaces de integrar y automatizar procesos relacionados con la administración del personal, tales como el registro de empleados, control de asistencia, gestión de contratos, evaluación del desempeño y generación de reportes. En este contexto, el uso de soluciones tecnológicas resulta fundamental para centralizar la información, garantizar su integridad, reducir la dependencia de procedimientos manuales y facilitar el acceso oportuno a datos relevantes para la gestión organizacional.

En contraste con este escenario de transformación digital y con la necesidad de centralizar la información institucional, la COAC Gañansol no dispone actualmente de un sistema informático integrado para la gestión de los procesos de Talento Humano. En consecuencia, la información relacionada con el personal, incluidos los datos personales, contratos, cargos, departamentos y evaluaciones de desempeño, se administra mediante herramientas ofimáticas y registros documentales físicos. Esta situación limita la centralización, trazabilidad e integridad de los datos, generando dispersión de la información y aumentando el esfuerzo requerido para gestionar procesos administrativos importantes. Como consecuencia, la búsqueda, actualización y validación de información

se vuelve más lenta y propensa a errores, afectando la eficiencia operativa del área de Talento Humano.

Por otra parte, en lo que respecta al control de asistencia, el sistema biométrico no se encuentra conectado a ningún sistema de gestión, por lo que los datos deben exportarse de forma manual a hojas de cálculo para su control y análisis. Esto convierte el procedimiento en una tarea tediosa y repetitiva, que puede generar discrepancias en los registros, pérdida de información y problemas en la validación de los datos. Como consecuencia, se dificulta el seguimiento adecuado del personal, el control de su comportamiento laboral y la confiabilidad de la información gestionada. De igual manera, los procesos de contratación, la administración de cargos y departamentos y la evaluación del desempeño no se gestionan de forma centralizada, lo que incrementa la dispersión de la información y el esfuerzo operativo. Estas limitaciones, sumadas a los tiempos prolongados que requiere la obtención de datos, afectan la eficiencia de la gestión cooperativa y la toma de decisiones institucionales.

D. OBJETIVO GENERAL

Desarrollar un sistema web para la gestión de talento humano en la COAC Gañansol utilizando técnicas de Vibe coding y herramientas de desarrollo web, para automatizar y optimizar los procesos administrativos del personal.

E. OBJETIVOS ESPECÍFICOS

1. Analizar los fundamentos teóricos del sistema web de gestión de talento humano, mediante la revisión de literatura científica en TI, ingeniería de

software y Vibe coding, para sustentar el desarrollo del sistema en la COAC Gañansol.

2. Evaluar las metodologías y enfoques de desarrollo aplicables al sistema web y técnicas de Vibe Coding asistidas por inteligencia artificial, mediante la identificación de sus fases, roles, artefactos y lineamientos de aplicación, para estructurar el proceso de desarrollo del sistema.

3. Desarrollar el sistema web de gestión de talento humano, considerando el levantamiento de requerimientos, modelado, construcción y pruebas en entorno local, para automatizar y optimizar los procesos administrativos del personal en la COAC Gañansol

F. JUSTIFICACIÓN

El presente proyecto surge de la necesidad de mejorar la gestión del talento humano en la COAC Gañansol, debido a que varios procesos administrativos relacionados con la información del personal se desarrollan mediante herramientas ofimáticas y registros documentales físicos. Esta situación genera dispersión de datos, inconsistencias, riesgo de pérdida de información y limitaciones para acceder de manera oportuna a datos confiables. Por ello, resulta necesario proponer una solución que contribuya a centralizar la información, optimizar los procesos internos y fortalecer la eficiencia administrativa de la institución.

Desde el punto de vista académico y científico, la investigación es relevante porque permite aplicar conocimientos relacionados con el desarrollo de sistemas de información, automatización de procesos y gestión organizacional en un contexto real. Además, aporta al análisis de cómo las soluciones tecnológicas pueden mejorar la administración del talento humano en instituciones financieras

cooperativas, donde la disponibilidad, trazabilidad y confiabilidad de la información son aspectos fundamentales para la toma de decisiones.

En este contexto, el desarrollo de un sistema web para la gestión del talento humano permitirá mejorar el registro, organización, consulta y control de la información del personal dentro de la COAC Gañansol. Asimismo, contribuirá a reducir la carga operativa, disminuir errores en el manejo de datos y facilitar la generación de reportes e indicadores para el seguimiento administrativo. De esta manera, el proyecto representa un aporte práctico para la cooperativa y una experiencia académica significativa en el diseño de soluciones informáticas orientadas a resolver necesidades institucionales reales.

G. ALCANCE

El presente proyecto consiste en el desarrollo de un sistema web para la gestión de talento humano en la COAC Gañansol, orientado a la automatización, centralización e integración de los procesos administrativos del personal mediante la aplicación de técnicas de Vibe Coding. La solución se ejecutará a nivel local, en un entorno de desarrollo y pruebas, con el propósito de verificar el funcionamiento de los módulos, validar el registro y almacenamiento de la información, realizar pruebas de operaciones CRUD y comprobar la integración lógica de los procesos antes de una posible implementación futura. El sistema será accesible desde un navegador web y estará respaldado por una base de datos que garantice el almacenamiento estructurado, seguro y centralizado de la información. El sistema estará conformado por módulos funcionales como Dashboard, Reclutamiento, Onboarding, Empleados, Asistencia, Licencias, Nómina, Desempeño y Configuración, en los cuales se implementarán operaciones CRUD para la gestión integral del talento humano. El módulo de empleados permitirá administrar la información personal y laboral, incluyendo la asignación de cargos, departamentos y contratos, mientras que el módulo de asistencia permitirá registrar y gestionar atrasos, faltas, permisos y ausencias del personal.

Asimismo, el sistema incorporará la integración lógica con dispositivos biométricos para la importación, validación y almacenamiento de los registros de entrada y salida del personal, garantizando la confiabilidad de los datos. De igual manera, se incluirá la gestión de usuarios integrada de manera lógica con servicios de directorio externo como Active Directory, permitiendo considerar la sincronización de usuarios y la creación, actualización y eliminación de registros entre ambos entornos. Finalmente, el alcance del proyecto se limita al diseño, desarrollo y validación funcional del sistema web en un entorno local de pruebas, incluyendo su arquitectura, base de datos y módulos principales, sin contemplar

el despliegue en producción, la implementación física de dispositivos biométricos ni la configuración real del servidor de Active Directory.

H. CONCEPTOS RELACIONADOS

En el contexto actual de transformación digital, las organizaciones tienen que llevar a cabo soluciones tecnológicas que les permitan optimizar los procesos administrativos y mejorar la gestión de la información. En este sentido, la creación del sistema web para la gestión del talento humano se ha convertido en una solución efectiva para la automatización, centralizar e integración de la información del personal dentro de una organización. Este tipo de propuestas se basa en varias áreas del conocimiento, como son las Tecnologías de la Información, a la Ingeniería de Software, así como a las metodologías de desarrollo de software, con el fin de proporcionar las bases adecuadas para la creación de sistemas eficientes, seguros y flexibles a las necesidades organizacionales.

1. Tecnologías de la Información (TI)

Las Tecnologías de la Información (TI) comprenden el conjunto de recursos, herramientas y sistemas que permiten el tratamiento, almacenamiento, procesamiento y transmisión de la información dentro de una organización, incluyendo hardware, software, redes de comunicación y bases de datos. Estas tecnologías facilitan la automatización de procesos, reduciendo errores, optimizando tiempos y mejorando la eficiencia operativa, permitiendo además centralizar la información y garantizar su disponibilidad, seguridad e integridad, lo que favorece el acceso oportuno a los datos por parte de los distintos niveles de la organización. Asimismo, las TI contribuyen significativamente a la toma de decisiones, ya que proporcionan información estructurada que puede ser analizada para generar reportes e indicadores que apoyan la planificación estratégica y la mejora continua (Chamba Pinta, 2026)

2. **Ingeniería de Software**

La Ingeniería de Software es la disciplina encargada de diseñar, desarrollar y mantener sistemas informáticos mediante la aplicación de metodologías, técnicas

y herramientas que permiten construir soluciones eficientes, seguras y alineadas a los requerimientos del usuario. Esta disciplina establece principios de desarrollo que buscan garantizar la calidad del software, su correcta funcionalidad y su adaptación a las necesidades del entorno organizacional **Fuente especificada no válida.**.

En el ámbito organizacional, permite construir sistemas de calidad que cumplen con características como mantenibilidad, escalabilidad y eficiencia, apoyándose en un ciclo de vida estructurado que incluye etapas como análisis, diseño, desarrollo y pruebas. Además, contribuye a la mejora continua del software, asegurando su adaptación a cambios y su correcto funcionamiento en distintos contextos (Flores LLangari, 2026).

3. **Desarrollo de Aplicaciones Web**

El desarrollo de aplicaciones web es el proceso mediante el cual se diseñan, construyen e implementan sistemas informáticos que funcionan a través de navegadores web. Este proceso comprende diversas etapas, como el análisis de requerimientos, diseño, programación, pruebas e implementación, con el objetivo de cumplir con funcionalidades específicas. Estas aplicaciones permiten a los usuarios interactuar con la información de forma dinámica y acceder a los sistemas desde distintos dispositivos (Córdova Poveda, Aplicación web utilizando react y nodejs para la gestión y seguimiento académico en el Departamento de Consejería Estudiantil de la Unidad Educativa Bolívar, 2024).

4. **Vibe coding**

El Vibe coding es una técnica de desarrollo de software asistida por inteligencia artificial, en la cual el desarrollador expresa los requerimientos o funcionalidades mediante instrucciones en lenguaje natural, y una herramienta de IA apoya en la generación, revisión, corrección, documentación y optimización del código. Este enfoque permite acelerar la construcción de prototipos y módulos funcionales, ya que la inteligencia artificial actúa como un asistente de programación que sugiere estructuras de código, ayuda a identificar errores y facilita la mejora progresiva

del sistema. Sin embargo, el uso de Vibe coding requiere supervisión humana, revisión técnica y pruebas constantes para garantizar la calidad, seguridad y correcto funcionamiento del software generado. IBM señala que el Vibe coding consiste en solicitar a herramientas de IA la generación de código en lugar de escribirlo completamente de forma manual, transformando instrucciones en lenguaje común en código ejecutable.

5. ***Sistema Web***

Un sistema web es una aplicación informática que funciona sobre un entorno web y permite gestionar información mediante una interfaz accesible desde Internet o redes internas. Este tipo de sistemas se caracteriza por operar a través de navegadores web, lo que facilita su uso sin necesidad de instalar software adicional en los dispositivos de los usuarios.

Los sistemas web están compuestos por diferentes elementos, como servidores, bases de datos, interfaces de usuario y servicios de comunicación, los cuales trabajan de manera integrada para garantizar el almacenamiento, procesamiento y acceso a la información en tiempo real. Gracias a esta estructura, permiten la centralización de datos, lo que contribuye a mantener la información organizada, actualizada y disponible para los distintos niveles de una organización (Córdova Poveda, Aplicación web utilizando react y nodejs para la gestión y seguimiento académico en el Departamento de Consejería Estudiantil de la Unidad Educativa Bolívar, 2024).

6. **Base de Datos**

Una base de datos es un sistema organizado de almacenamiento de información que permite gestionar grandes volúmenes de datos de forma estructurada, segura y eficiente. Está compuesta por un conjunto de datos relacionados que pueden ser almacenados, consultados y manipulados mediante sistemas de gestión de

bases de datos (DBMS), los cuales permiten administrar el acceso, la integridad y la seguridad de la información.

En el contexto de los sistemas web, las bases de datos cumplen un rol fundamental, ya que permiten almacenar y gestionar información relevante como datos de empleados, registros de asistencia, contratos y demás información necesaria para el funcionamiento del sistema, facilitando su acceso y procesamiento en tiempo real (Gómez Sánchez, Herrera Herrera, Pavón Domínguez , & Jacqueline Vanessa, 2024)

7. ***Operaciones CRUD***

Las operaciones CRUD (Crear, Leer, Actualizar y Eliminar) son las funciones básicas que permiten gestionar la información dentro de un sistema. Estas operaciones son esenciales en el desarrollo del sistem01a web, ya que permiten la administración de datos como empleados, asistencias, permisos y otros registros.

8. ***Metodologías de Desarrollo de Software***

Las metodologías de desarrollo de software son enfoques estructurados que permiten organizar, planificar y controlar el proceso de creación de sistemas informáticos. Estas metodologías establecen etapas, roles y técnicas que facilitan el desarrollo eficiente del software, garantizando su calidad y cumplimiento de los requerimientos del usuario (Jurado Acurio, 2024).

9. ***Metodología en Cascada***

La metodología en cascada es un modelo de desarrollo secuencial en el que cada fase del proyecto debe completarse antes de iniciar la siguiente. Este enfoque incluye etapas como análisis, diseño, implementación, pruebas y mantenimiento. Su principal ventaja es la claridad en la planificación, aunque presenta limitaciones cuando existen cambios en los requerimientos (Jurado Acurio, 2024).

10. ***Metodología Espiral***

La metodología espiral combina elementos del modelo secuencial con la gestión de riesgos. Se basa en ciclos iterativos en los que se identifican, analizan y mitigan riesgos antes de avanzar a la siguiente fase. Este enfoque permite mayor flexibilidad y control en proyectos complejos (Jurado Acurio, 2024).

11. ***RUP (Proceso Unificado Racional)***

El Proceso Unificado Racional (RUP) es una metodología iterativa que organiza el desarrollo del software en fases como inicio, elaboración, construcción y transición. Se enfoca en buenas prácticas de ingeniería de software, promoviendo la documentación, el análisis de requerimientos y la calidad del sistema (Jurado Acurio, 2024)

12. ***Metodología de Prototipo***

El prototipado consiste en la creación de versiones preliminares del sistema con el fin de validar requerimientos y obtener retroalimentación de los usuarios. Este enfoque permite mejorar el diseño del sistema antes de su implementación final, reduciendo errores y mejorando la aceptación del producto (Jurado Acurio, 2024).

13. ***Metodologías Ágiles***

Las metodologías ágiles surgen como respuesta a las limitaciones de los modelos tradicionales, proponiendo un enfoque flexible basado en el desarrollo iterativo e incremental. Estas metodologías priorizan la colaboración, la adaptación al cambio y la entrega continua de valor al usuario (Jurado Acurio, 2024).

14. ***Extreme Programming (XP)***

Extreme Programming (XP) es una metodología ágil enfocada en mejorar la calidad del software mediante prácticas como la programación en pares, pruebas constantes e integración continua. Su objetivo es responder rápidamente a los cambios y garantizar un producto funcional en cada iteración (Jurado Acurio, 2024).

15. ***Scrum***

Scrum es una metodología ágil que organiza el desarrollo en ciclos cortos denominados sprints, en los cuales se entregan funcionalidades de manera progresiva. Se basa en roles definidos como Product Owner, Scrum Master y equipo de desarrollo, promoviendo la colaboración y la mejora continua (Jurado Acurio, 2024) .

16.*Kanban*

Kanban es una metodología ágil que utiliza tableros visuales para gestionar el flujo de trabajo. Permite visualizar las tareas, limitar el trabajo en proceso y optimizar la productividad del equipo mediante la mejora continua (Jurado Acurio, 2024).

17. ***Sistema de Gestión de Talento Humano***

Un sistema de gestión de talento humano es una solución informática que permite administrar de manera integral los procesos relacionados con el personal dentro de una organización, tales como el registro de empleados, control de asistencia, gestión de contratos, evaluación del desempeño y generación de reportes. Estos sistemas facilitan la automatización, centralización y análisis de la información, contribuyendo a una mejor toma de decisiones.

Además, la adecuada gestión del talento humano es fundamental dentro de las organizaciones, ya que involucra aspectos normativos, derechos laborales y estrategias orientadas al desarrollo del personal, lo cual garantiza el cumplimiento de obligaciones y mejora el desempeño organizacional (Vivanco Herrera, 2023).

18. **Gestión de Talento Humano**

La gestión de talento humano es el conjunto de procesos orientados a la administración del personal dentro de una organización, incluyendo actividades como el reclutamiento, selección, contratación, capacitación, evaluación del desempeño y control de asistencia. Su adecuada gestión permite mejorar la productividad, el rendimiento laboral y apoyar la toma de decisiones estratégicas,

contribuyendo al cumplimiento de los objetivos organizacionales (Escobar Pullutasig, 2025).

I. TRABAJOS RELACIONADOS

Según Gaibor (2025) y diversos estudios sobre Sistemas de Información de Recursos Humanos (HRIS), la implementación de sistemas web apoyados en inteligencia artificial permite reducir tiempos de procesamiento, minimizar errores y optimizar la gestión de procesos como asistencia, nómina y control de personal. Este aporte resulta fundamental para la presente tesis, ya que sustenta el desarrollo de un sistema web para la gestión de talento humano en la COAC Gañansol, orientado a centralizar la información, automatizar procesos mediante herramientas como chatbots y generar reportes en tiempo real que faciliten la toma de decisiones. Asimismo, el uso de tecnologías web y metodologías ágiles garantiza mayor accesibilidad, flexibilidad y adaptación a los requerimientos organizacionales, contribuyendo a mejorar la eficiencia operativa y el enfoque estratégico del área de talento humano.

(Ncube, Sishi, & Skinner, 2025), en el estudio titulado “The impact of artificial intelligence on human resource management practices: An investigation”, se analizan el impacto de la inteligencia artificial en la gestión de recursos humanos. Los autores concluyen que la IA mejora la eficiencia mediante la automatización de tareas, el análisis de datos y la toma de decisiones basada en información. Este aporte sustenta el uso de herramientas de inteligencia artificial en el desarrollo del sistema propuesto, especialmente mediante técnicas de Vibe Coding para optimizar procesos administrativos y la gestión del talento humano.

En el contexto nacional (Tomalá Lino, 2025), establece que la automatización, mediante tecnologías como smart contracts y soluciones digitales, permite la ejecución autónoma de tareas administrativas bajo reglas definidas, optimizando la trazabilidad, consistencia e integridad de la información, además de reducir la intervención manual en procesos repetitivos; este enfoque constituye una base técnica para el diseño de flujos de trabajo automatizados dentro del sistema web propuesto, especialmente en módulos como asistencia, contratos y gestión de permisos. En este mismo contexto, la incorporación de tecnologías biométricas fortalece la precisión y confiabilidad en el registro de asistencia, lo cual se traduce en la integración lógica de dispositivos biométricos con la plataforma web, permitiendo la captura, validación y procesamiento automatizado de datos. De forma complementaria, Mixi et al. (2023) destacan la relevancia de las bases de datos como componente estructural para el almacenamiento, procesamiento y análisis de información, lo que sustenta la implementación de una base de datos centralizada en el sistema propuesto, orientada a soportar operaciones CRUD, generación de reportes e indicadores, y facilitar la toma de decisiones basada en datos dentro del entorno organizacional.

El estudio de Cueva (2023) fundamenta conceptualmente la gestión del talento humano como un sistema de procesos estratégicos interrelacionados que inciden directamente en el desempeño laboral y la eficiencia organizacional, estableciendo dimensiones clave como planificación, selección, desarrollo y evaluación del personal. Estos elementos permiten definir los requerimientos funcionales y la arquitectura lógica del sistema propuesto, orientando la estructuración de módulos para la gestión integral del talento humano. Asimismo, proporciona criterios para la incorporación de mecanismos de evaluación del desempeño y generación de indicadores, facilitando la implementación de funcionalidades orientadas al análisis de datos y soporte a la toma de decisiones dentro de la plataforma web.

De igual forma Guerra (2026) fundamenta la presente tesis al evidenciar que la implementación de sistemas biométricos para el control de asistencia mejora la precisión de los registros, reduce errores administrativos y permite el monitoreo en tiempo real del personal. Este planteamiento sustenta la incorporación de un módulo de control de asistencia dentro del sistema web propuesto, basado en la integración lógica de dispositivos biométricos para automatizar la captura y procesamiento de datos. Asimismo, orienta el diseño de funcionalidades enfocadas en garantizar la integridad, disponibilidad y análisis de la información.

J. METODOLOGÍA

La presente investigación adopta un enfoque mixto, combinando el análisis cualitativo de los procesos actuales de gestión de talento humano en la COAC Gañansol con una evaluación cuantitativa del funcionamiento del sistema web desarrollado. El nivel de investigación es descriptivo-aplicado, ya que se busca describir la situación actual de los procesos administrativos del personal y desarrollar una solución tecnológica que permita automatizar y optimizar la gestión de empleados, asistencia, permisos, nómina, desempeño y demás procesos relacionados.

Para el desarrollo del sistema web se aplicará la metodología ágil Scrum, la cual permitirá organizar el trabajo de manera iterativa e incremental mediante sprints. Esta metodología facilitará la planificación, levantamiento de requerimientos, diseño, construcción, pruebas y validación progresiva de los módulos funcionales del sistema, permitiendo verificar su funcionamiento en un entorno local de desarrollo y pruebas. Asimismo, se emplearán técnicas de Vibe Coding como apoyo al desarrollo asistido por inteligencia artificial, facilitando la generación, revisión, corrección, documentación y optimización del código bajo supervisión técnica del desarrollador.

K. CRONOGRAMA DE ACTIVIDADES								
N°	ACTIVIDAD	MES						
		I	II	III	IV	V	VI	MEDIOS DE VERIFICACIÓN
1	Revisión de literatura científica	x	X					Fichas bibliográficas
2	Análisis de metodologías ágiles y técnicas de Vibe Coding	X						Documento de análisis y matriz comparativa
3	Levantamiento de requerimientos y elaboración del Product Backlog	x	x					Documento de requerimientos y Product Backlog
4	Diseño del sistema y planificación de sprints	x	x	x	x	x		Diagramas, modelos y Sprint Backlog
5	Desarrollo del sistema web mediante sprints	x	x	x	x	x	x	Aplicación web funcional en entorno local
6	Pruebas y validación del sistema en entorno local	x	x	x	x	x	x	Informe de pruebas y validación

7	Documentación y manual de usuario					x	x	Manual de usuario y documentación del sistema

L. DECLARACIÓN FINAL
Los abajo firmantes declaramos bajo juramento que el proyecto descrito en este documento no ha sido presentado a otra institución nacional o internacional para su financiamiento, no causa perjuicio al ambiente, es de nuestra autoría y no transgrede norma ética alguna.

M. PARTICIPANTES	
DIRECTOR:	Ing. Fernanda Michaelle Narváez Ochoa
ESTUDIANTE 1	Jonnathan Gustavo Paucar Paucar

N. FIRMAS DE RESPONSABILIDAD	
Lugar:	Cañar
Fecha:	19 de abril del 2026

Firmas:

Nombre: Ing. Fernanda Michaelle Narváez Ochoa	Nombre: Jonnathan Gustavo Paucar Paucar
CC:	C.C.:0302546098
Director del Proyecto	**Estudiante / Egresado**

O. APROBACIÓN
Firmas:

Nombre: CC: **Primer Par Revisor**	Nombre: C.C.: **Segundo Par Revisor**

P. REFERENCIAS

Referencias

Chamba Pinta, K. S. (2026). *Aplicación Web con análisis predictivo para la gestión de pedidos en la ferretería Ferritools's.* Universidad Técnica de Ambato. Obtenido de https://repositorio.uta.edu.ec/server/api/core/bitstreams/d29f41ac-3c3e-4faf-aa84-a4ffaa2a2331/content

Córdova Poveda, J. R. (2024). *Aplicación web utilizando react y nodejs para la gestión y seguimiento académico en el Departamento de Consejería Estudiantil de la Unidad Educativa Bolívar.* Ambato: Universidad Tecnica de Ambato. Obtenido de https://repositorio.uta.edu.ec/server/api/core/bitstreams/6d6204ee-646a-4440-aee9-b63ee0e4dd77/content

Córdova Poveda, J. R. (2024). *Aplicación web utilizando react y nodejs para la gestión y seguimiento académico en el Departamento de Consejería Estudiantil de la Unidad Educativa Bolívar.* Ambato: Universidad Tecnica de Ambato. Obtenido de https://repositorio.uta.edu.ec/server/api/core/bitstreams/6d6204ee-646a-4440-aee9-b63ee0e4dd77/content

Cueva Jara, C. G. (2023). *La Gestión del Talento Humano y el Desempeño Laboral del Hospital.* UNIVERSIDAD NACIONAL DE CHIMBORAZO, Ribamba. Obtenido de http://dspace.unach.edu.ec/bitstream/51000/10717/1/Cueva%20Jara%2c%20C.%282023%29%20La%20Gesti%c3%b3n%20del%20Talento%20Humano%20y%20el%20Desempe%c3%b1o%20Laboral%20del%20Hospital%20de%20Especialidades%20San%20Juan%20HOSPIESAJ.S.A.%20de%20Riobamba..pdf

Escobar Pullutasig, J. D. (2025). *Aplicación web utilizando .net core para la gestión de la información de empleados en la empresa Ciudad del Auto Cia Ltda.* Ambato: Universidad Tecnica de Ambato. Obtenido de https://repositorio.uta.edu.ec/items/f034264a-a1cd-4694-abd9-f3056d8c338a

Flores LLangari, F. S. (2026). *Aplicativo multiplataforma basado en arquitectura MVC orientado a la gestión de la información del "Club deportivo especializado formativo Atahualpa" de la ciudad de Ambato.* Universidad Técnica de Ambato. Obtenido de https://repositorio.uta.edu.ec/server/api/core/bitstreams/f89b02d2-413b-479e-bd71-949c9d320f34/content

Gaibor Villacís, C. A. (2025). *Aplicación web con chatbot para la digitalización de la gestión de tiempo y asistencia laboral en talento humano de la empresa pública Empresa Mercado Mayorista Ambato.* Ambato: UNIVERSIDAD TÉCNICA DE AMBATO. Obtenido de https://repositorio.uta.edu.ec/server/api/core/bitstreams/5f073383-aaeb-4ad4-b36a-09d19feb2470/content

Gómez Sánchez, M. D., Herrera Herrera, D. H., Pavón Domínguez , C. E., & Jacqueline Vanessa, Q. T. (2024). *Diseño de un modelo de detección de datos duplicados mediante procesamiento del lenguaje natural para optimizar la eficiencia en la gestión de datos de un laboratorio farmacéutico.* Quito: Universidad: Universidad Internacional del Ecuador. Obtenido de https://repositorio.uide.edu.ec/handle/37000/6514

Guerra Jordán, M. D. (2026). *Sistema de control y gestión remota para el registro de asistencia basado en biometría portátil para la empresa Megaeléctric en sus obras temporales.* Ambato: Universidad Técnica de Ambato. Obtenido de https://repositorio.uta.edu.ec/server/api/core/bitstreams/e24bedd5-5c46-41bf-a021-238f4bc13e92/content

Jurado Acurio, B. S. (2024). *Metodología híbrida para estandarizar el desarrollo de aplicaciones web, aplicando Normas ISO/IEC 25000 en la Empresa Eléctrica Ambato S.A. (EEASA).* Ambato: Universida Tecnica de Ambato. Obtenido de https://repositorio.uta.edu.ec/server/api/core/bitstreams/47ab8db2-f6da-4dd4-b1a7-7e4283c47286/content

Mixi Joselyne, S. T., Navas Espin, W. R., & Francisco Nicolás, R. V. (2023). *Diseño de un modelo predictivo de aprobación de materias mediante la analítica de datos*

para la planificación y proyección de cursos por periodo académico en la Carrera de Telemática - Universidad de Guayaquil. Quito: Universidad: Universidad Internacional del Ecuador. Obtenido de https://repositorio.uide.edu.ec/bitstream/37000/6515/1/UIDE-Q-TMSIG-2023-13.pdf

Ncube, T. R., Sishi, K. K., & Skinner, J. P. (2025). The impact of artificial intelligence on human resource management practices: An investigation. *SA Journal of Human Resource Management*, 1-11. doi:10.4102/sajhrm.v23i0.2960

Roosevelt Manlio, M. C. (2023). *Implementación de un Sistema Biométrico con huellas dactilares para el control de asistencia del personal docente y administrativo en la unidad educativa fiscal puerto cayo.* manabí: Universidad Estatal del Sur de Manabí. Obtenido de https://repositorio.unesum.edu.ec/bitstream/53000/4779/1/Menendez%20Cevallos%20Roosevelt%20Manlio.pdf

Silverio Torres, V. j. (2021). *Desarrollo de una aplicación móvil para la promoción de eventos para el departamento de actividades complementarias de la Pontificia Universidad Católica del Ecuador (PUCE). Caso de estudio: proyecto cultural Voces desde la Mitad del Mundo.* Quito: PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR. Obtenido de https://repositorio.puce.edu.ec/items/3e036c13-6f19-4803-81da-cde18d22c19b

Tomalá Lino, K. J. (2025). *Gestión administrativa mediante blockchain para el programa de talento humano de Posgrado UPSE, Cantón La Libertad, año 2025.* La Libertad: Universidad Estatal Península de Santa Elena. Obtenido de https://repositorio.upse.edu.ec/server/api/core/bitstreams/3e1cbfa4-95c6-499d-91e4-44bfbdba4bcb/content

Vivanco Herrera, D. L. (2023). *Análisis del sistema de gestión de talento humano para la empresa florícola Puliza Flowers, Cayambe, provincia de Pichincha.* Ibarra: PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR. Obtenido de

https://repositorio.puce.edu.ec/server/api/core/bitstreams/711f35b5-f0f3-42f8-8c3d-5ed6cb1c50bf/content

ANEXOS

Anexo A. Matriz de análisis documental de los formatos institucionales

Código	Documento institucional	Tipo de archivo	Elementos identificados	Módulo del sistema
FO-TH-001	Base de datos del personal	Hoja de cálculo	Datos personales, contractuales y laborales del empleado; campos obligatorios de identificación	Empleado
FO-TH-002	Creación y administración de usuarios	Hoja de cálculo	Perfiles, niveles de acceso y autorizaciones por área	Usuarios y seguridad
FO-TH-003	Solicitud de acumulación o mensualización de décimos	Documento de texto	Identificación del empleado, período, opción seleccionada y firmas de autorización	Nómina
FO-TH-004	Formato de inducción y reinducción	Documento de texto	Temas de inducción, responsables, fechas y constancia de cumplimiento	Onboarding
FO-TH-005	Acta de entrega y recepción del puesto	Documento de texto	Bienes, archivos físicos y digitales, responsables entrante y saliente	Gestión documental
FO-TH-006	Evaluación de desempeño en período de prueba	Hoja de cálculo	Factores evaluados, escalas de calificación y resultado final	Desempeño
FO-TH-007	Matrices de evaluación del desempeño por competencias	Hoja de cálculo	Listado de personal, cargos, distribución de matrices, incidentes críticos, competencias comportamentales y técnicas con ponderaciones y diccionarios de competencias	Desempeño básico inicial
FO-TH-008	Control de vacaciones	Hoja de cálculo	Períodos, días generados, días tomados y saldos por empleado	Licencias
FO-TH-009	Reporte de novedades de Talento Humano	Documento de texto	Novedades del período que afectan el rol de pagos	Nómina
FO-TH-010	Solicitud de permiso	Hoja de cálculo	Tipo de permiso, fechas, motivo y autorización	Licencias
FO-TH-011	Solicitud de vacaciones	Hoja de cálculo	Período solicitado, días, reemplazo y aprobación	Licencias
FO-TH-012	Liquidación de haberes acumulados o mensualizados	Hoja de cálculo	Rubros, cálculos y totales de la liquidación	Nómina
FO-TH-013	Liquidación de vacaciones	Hoja de cálculo	Días pendientes, valores calculados y período liquidado	Licencias
FO-TH-014	Control de pasantías	Hoja de cálculo	Datos del pasante, institución educativa, fechas y actividades	Pasantías
FO-TH-015	Ficha de la entrevista	Hoja de cálculo	Datos del postulante, vacante, criterios de evaluación y resultado	Reclutamiento
FO-TH-016	Informe de actividades bajo modalidad virtual	Documento de texto	Actividades realizadas, fechas y horas reportadas	Asistencia
FO-TH-017	Plan anual de capacitación	Hoja de cálculo	Temas, participantes, cronograma y responsables	Capacitación

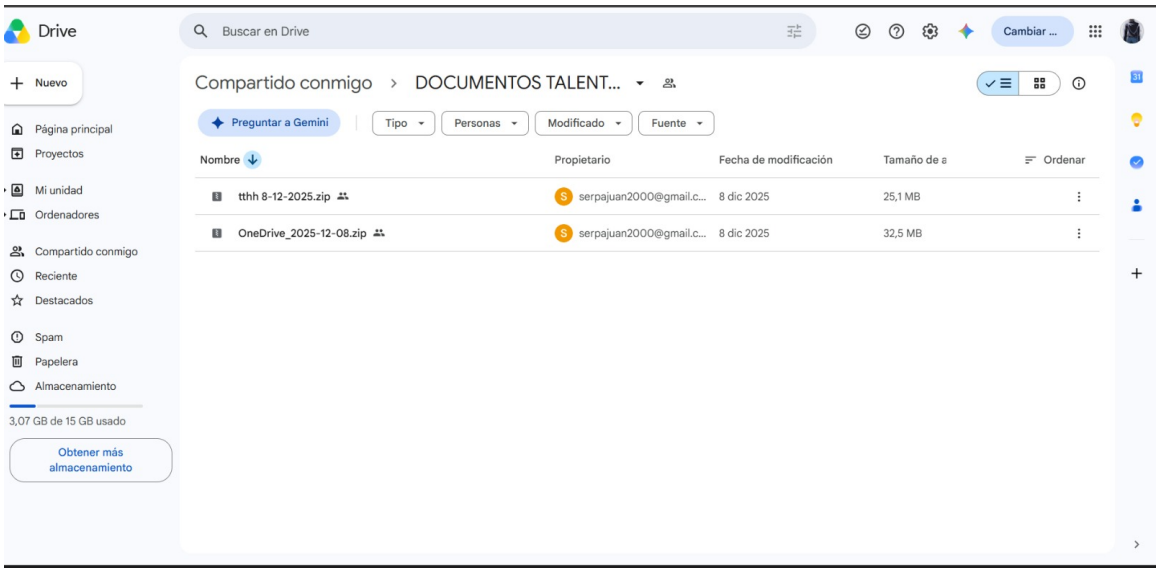
Nota. Autoría propia, elaborada a partir de los formatos entregados por la COAC Gañansol.

Anexo B. Evidencia de la documentación institucional entregada

Capturas de pantalla que respaldan la entrega de los archivos institucionales analizados en el Anexo A.

Figura A1

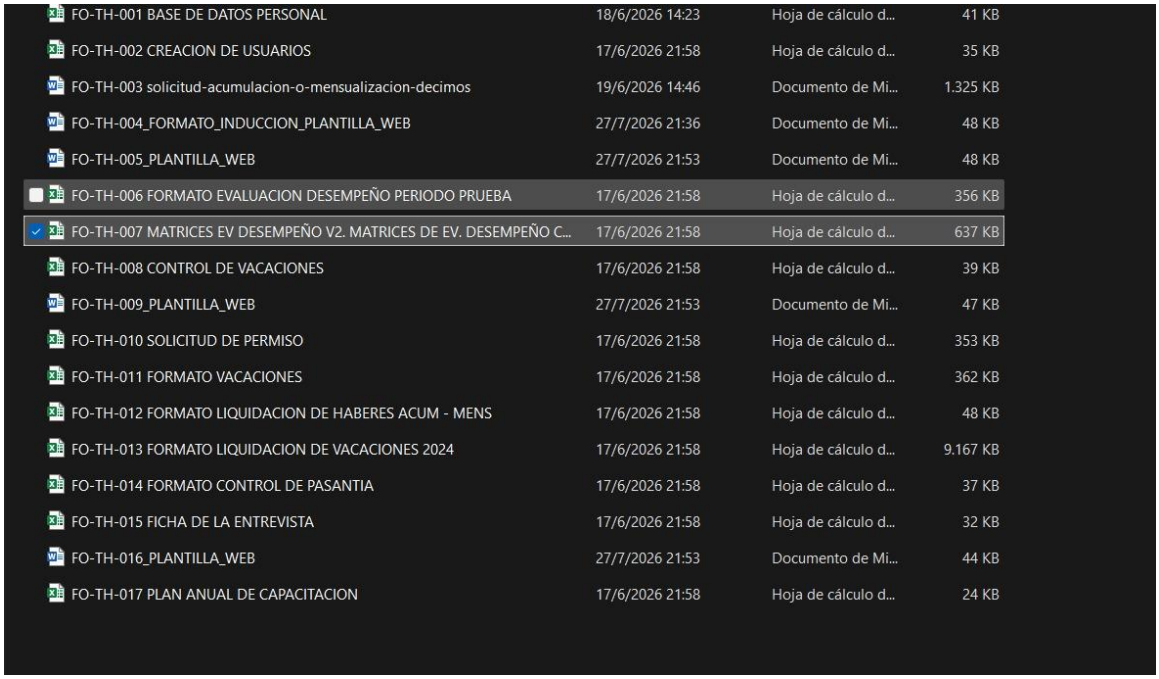
Carpeta compartida por la cooperativa con la documentación de Talento Humano



Nota. Captura proporcionada por la COAC Gañansol.

Figura A2

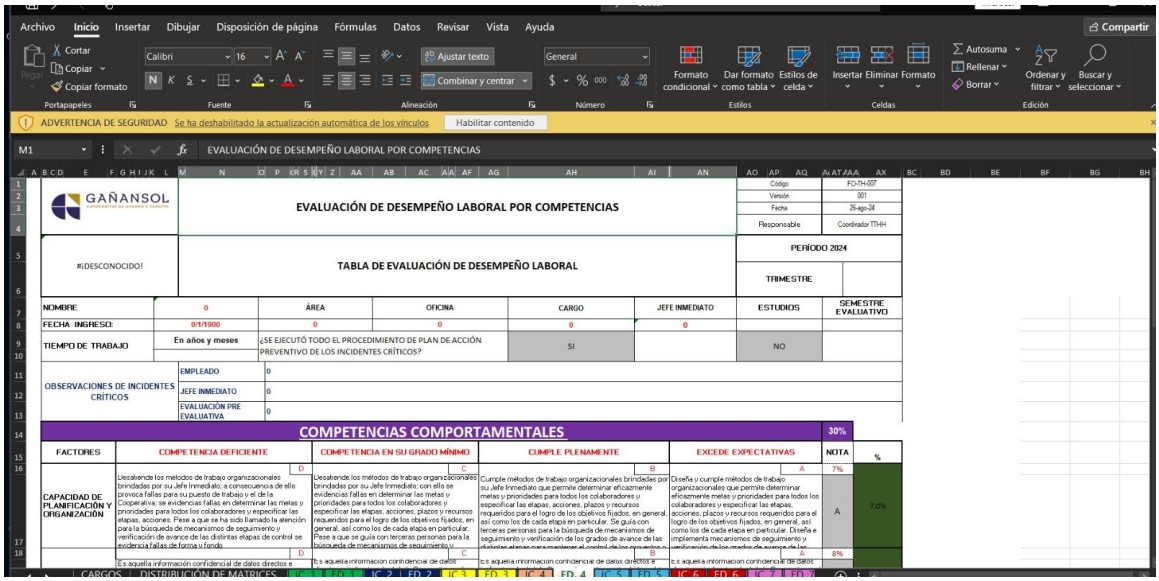
Formatos institucionales codificados FO-TH-001 a FO-TH-017 entregados para el análisis documental



Nota. Captura proporcionada por la COAC Gañansol.

Figura A3

Estructura interna del formato FO-TH-007, Matrices de evaluación del desempeño



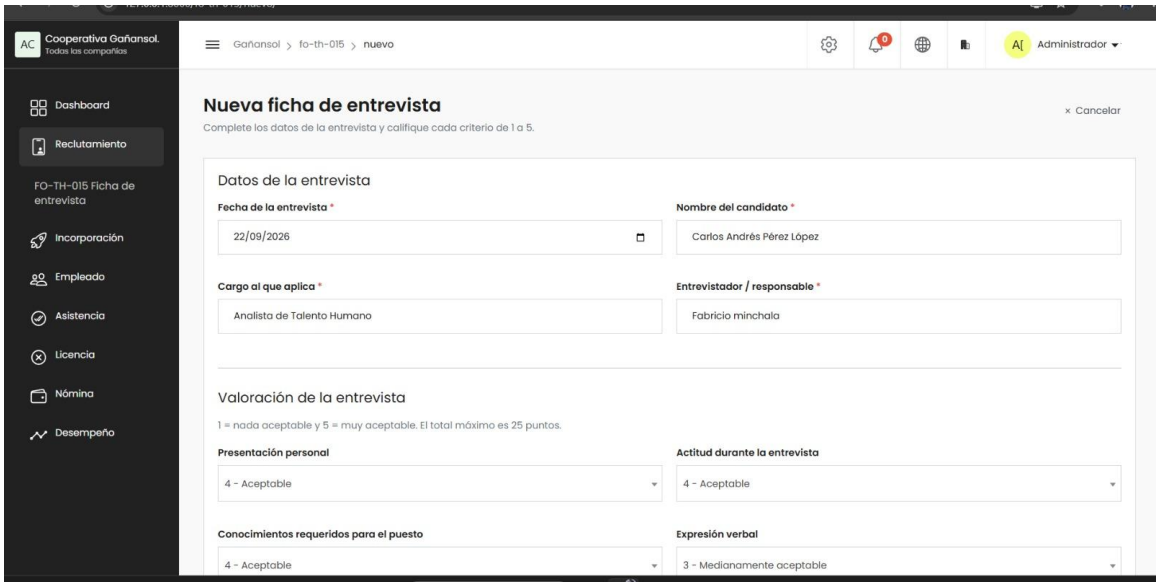
Nota. Captura proporcionada por la COAC Gañansol.

Anexo C. Evidencia complementaria de las pruebas funcionales: ciclo CRUD del módulo FO-TH-015

Las siguientes capturas corresponden a la ejecución completa del ciclo CRUD en el módulo de reclutamiento y respaldan el caso de prueba CP16 de la Tabla 15. Se presentan como evidencia representativa del procedimiento aplicado de forma sucesiva en los demás módulos del sistema, los cuales comparten la misma estructura de vistas, validaciones y plantillas.

Figura A4

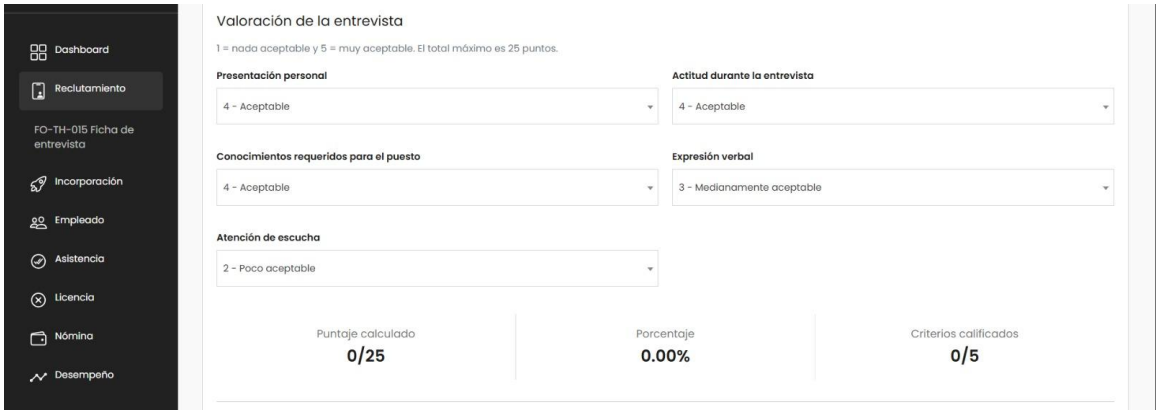
Formulario de registro de una nueva ficha de entrevista FO-TH-015 (operación de creación)



Nota. Autoría propia. Captura del entorno local de pruebas.

Figura A5

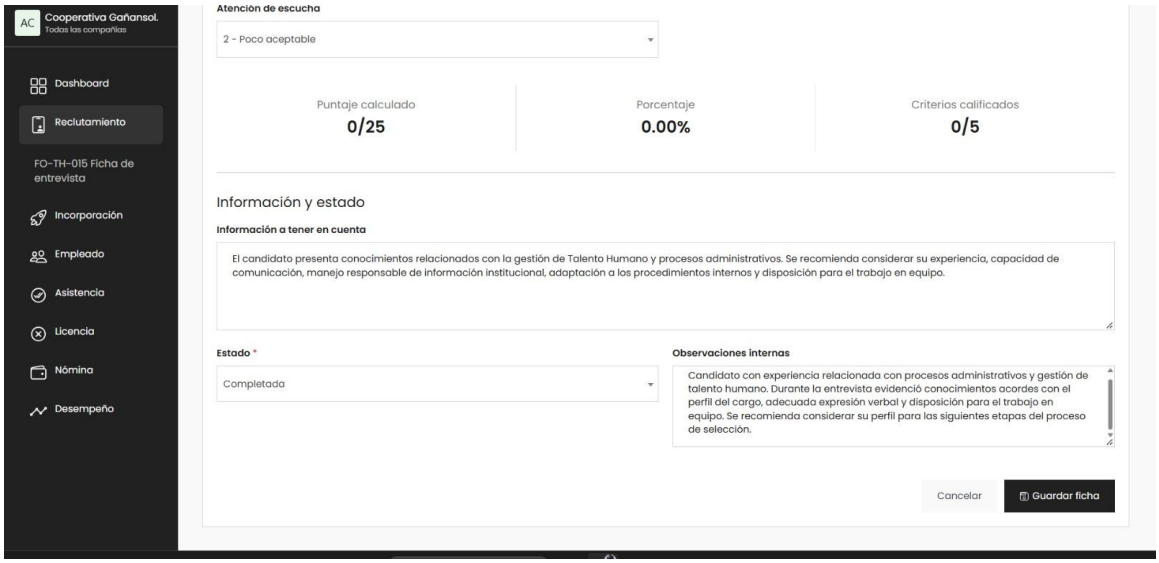
Valoración de la entrevista con los criterios institucionales y el cálculo automático del puntaje



Nota. Autoría propia. Captura del entorno local de pruebas.

Figura A6

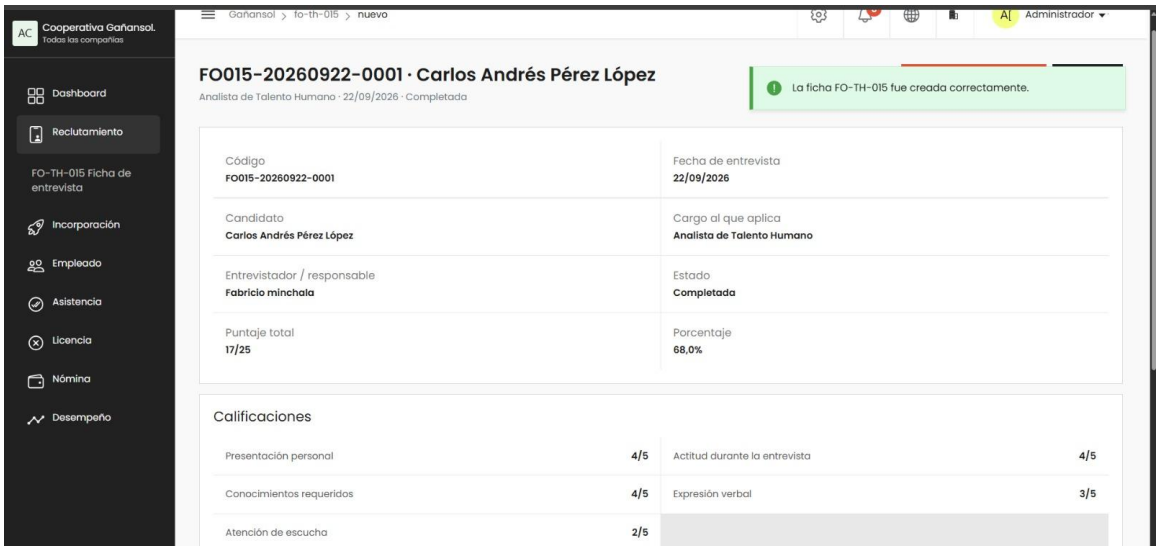
Sección de información, estado y observaciones previa al guardado del registro



Nota. Autoría propia. Captura del entorno local de pruebas.

Figura A7

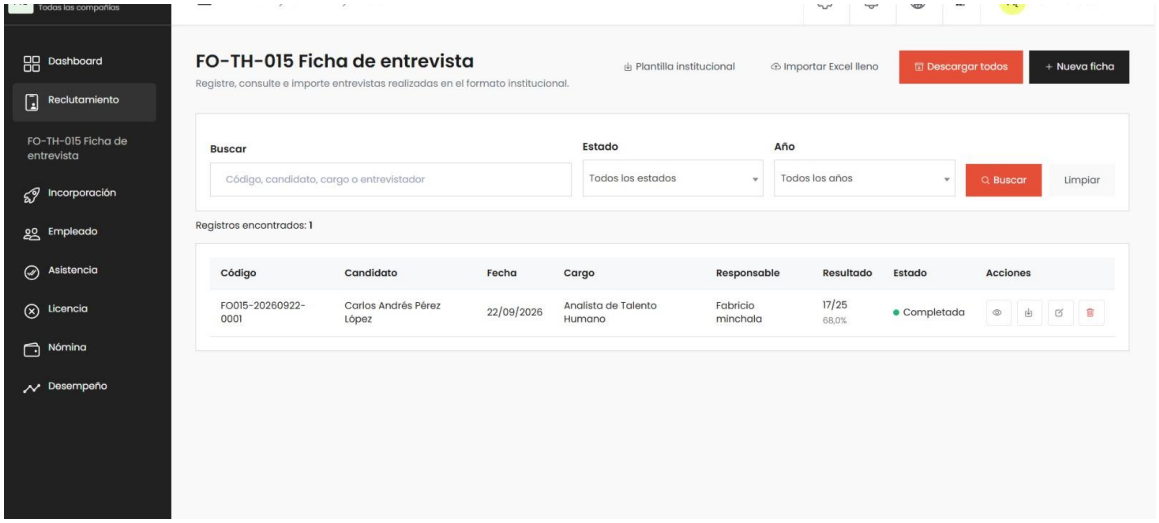
Confirmación de la creación del registro FO015-20260922-0001 y detalle de las calificaciones



Nota. Autoría propia. Captura del entorno local de pruebas.

Figura A8

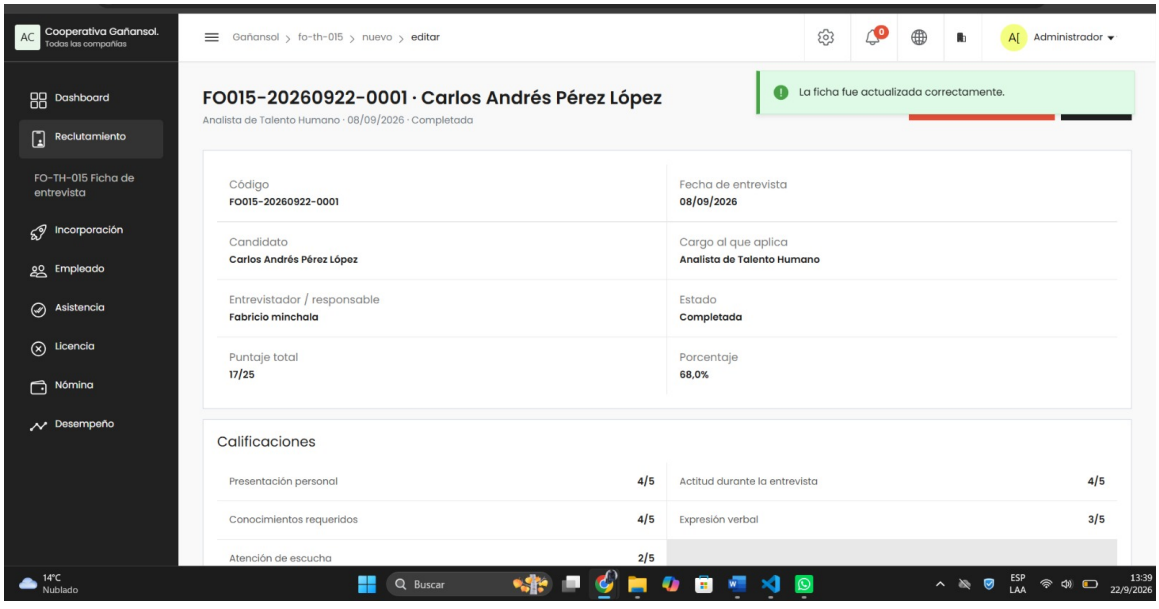
Listado de fichas registradas con las acciones de consulta, descarga, edición y eliminación (operación de lectura)



Nota. Autoría propia. Captura del entorno local de pruebas.

Figura A9

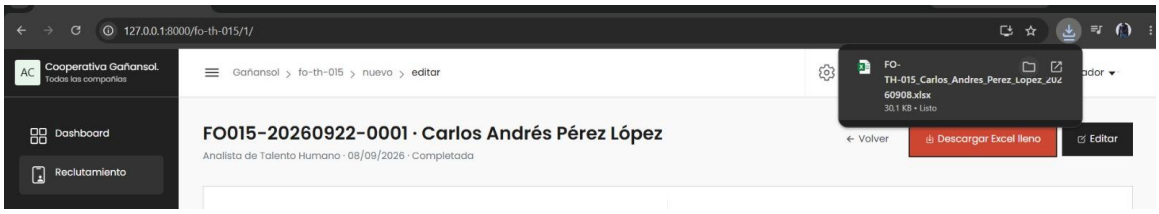
Confirmación de la actualización del registro tras modificar la fecha de la entrevista (operación de edición)



Nota. Autoría propia. Captura del entorno local de pruebas.

Figura A10

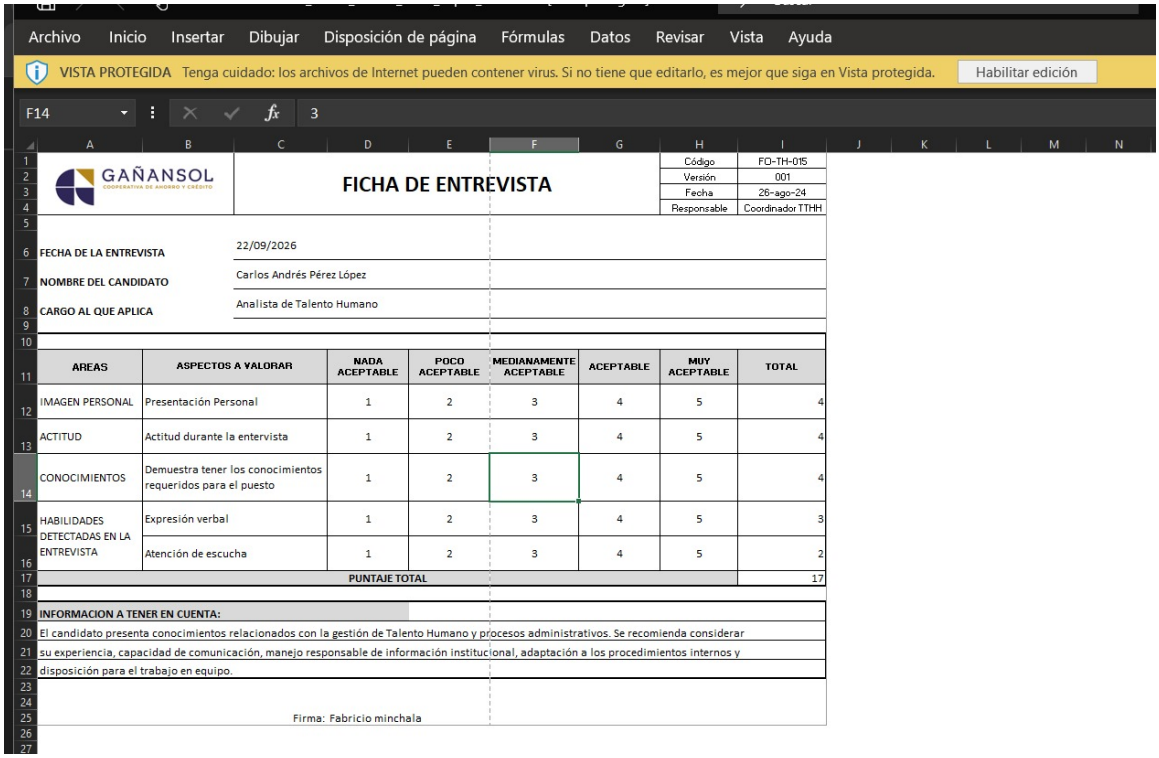
Descarga del archivo Excel generado por el sistema a partir del registro almacenado



Nota. Autoría propia. Captura del entorno local de pruebas.

Figura A11

Documento institucional FO-TH-015 generado por el sistema con los datos registrados y el puntaje total



Nota. Autoría propia. Captura del entorno local de pruebas.

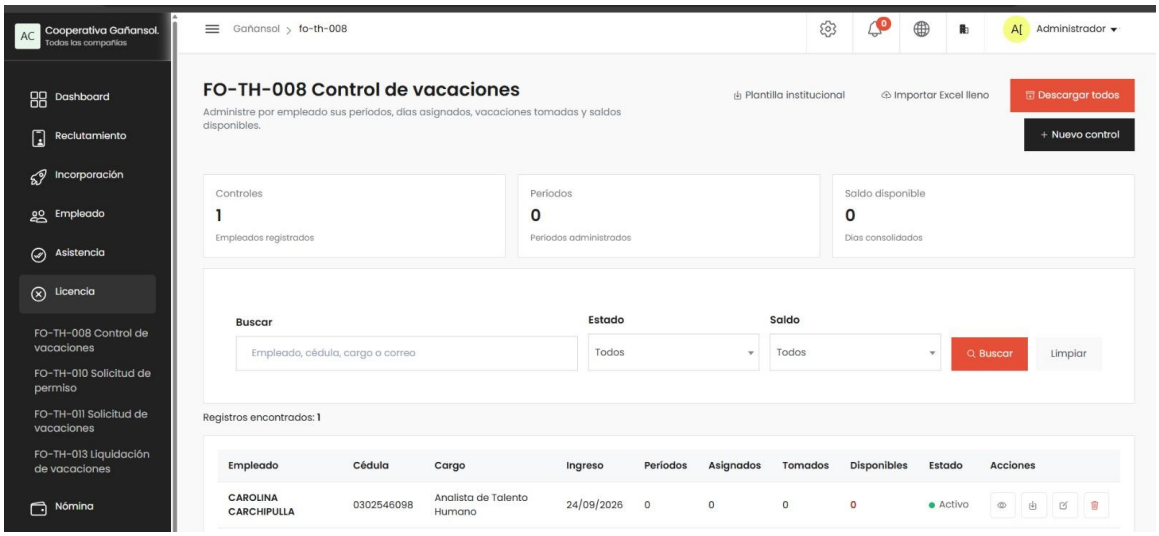
Anexo D. Evidencia de los módulos verificados sin figura en el Capítulo IV

Las siguientes capturas complementan el apartado 4.6.9 y corresponden a los módulos cuyas operaciones fueron verificadas durante las pruebas funcionales, pero que no cuentan con una

figura en el cuerpo del capítulo. Se presentan las pantallas principales de cada módulo, en las cuales se observan los registros almacenados y las acciones de consulta, descarga, edición y eliminación disponibles para el usuario autorizado. No se incluyen capturas de cada operación individual porque todos los módulos comparten el mismo flujo de trabajo, de modo que las pantallas mostradas son representativas del procedimiento aplicado en los demás casos de prueba de la Tabla 15.

Figura A12

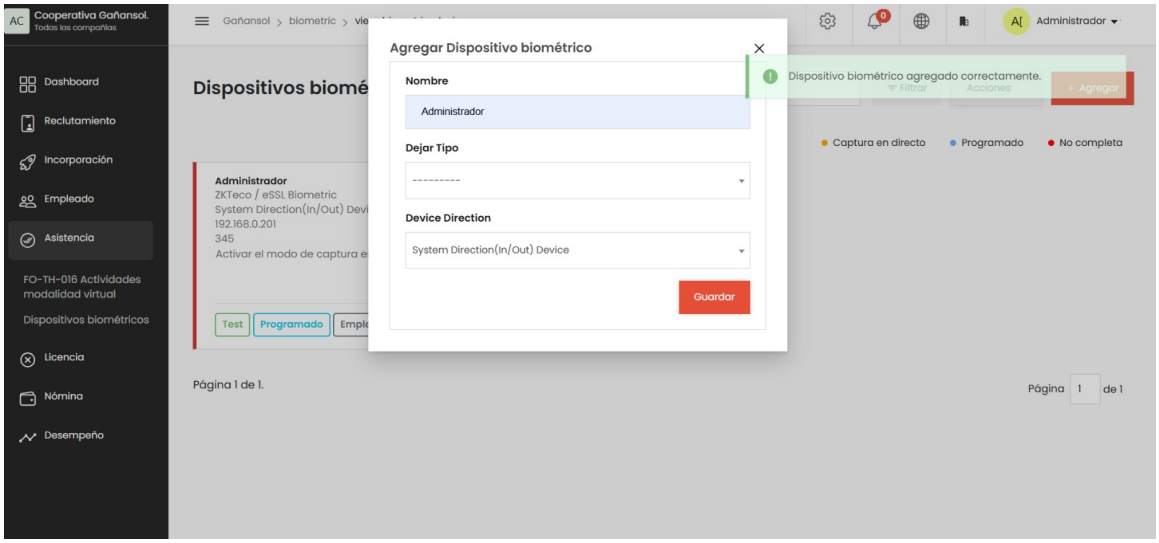
Módulo de Licencia: control de vacaciones FO-TH-008 con períodos, días asignados, tomados y saldos disponibles



Nota. Autoría propia. Captura del entorno local de pruebas.

Figura A13

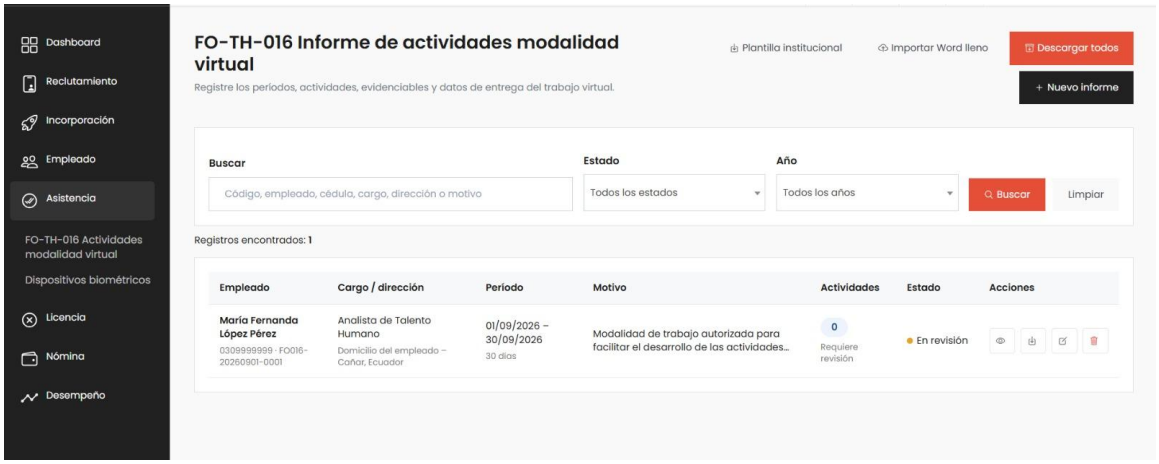
Módulo de Asistencia: configuración preliminar de dispositivos biométricos



Nota. Autoría propia. Captura de configuración preliminar del dispositivo en el entorno local. El alcance se limitó a verificar conectividad e integración preliminar; no acredita administración de usuarios ni sincronización de asistencia.

Figura A14

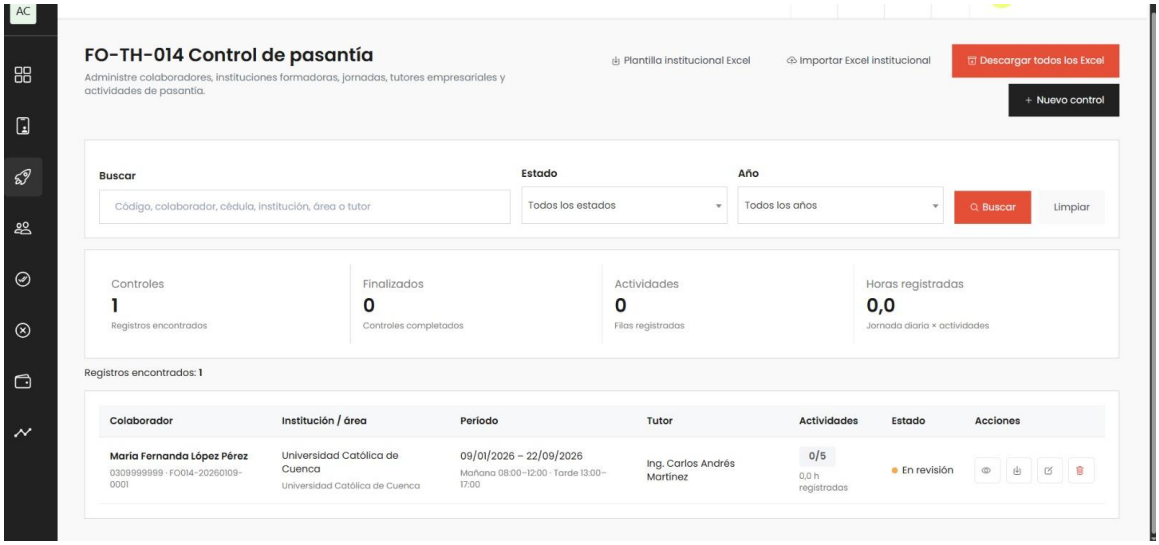
Módulo de Asistencia: informes de actividades bajo modalidad virtual FO-TH-016 con sus acciones de consulta, descarga, edición y eliminación



Nota. Autoría propia. Captura del entorno local de pruebas.

Figura A15

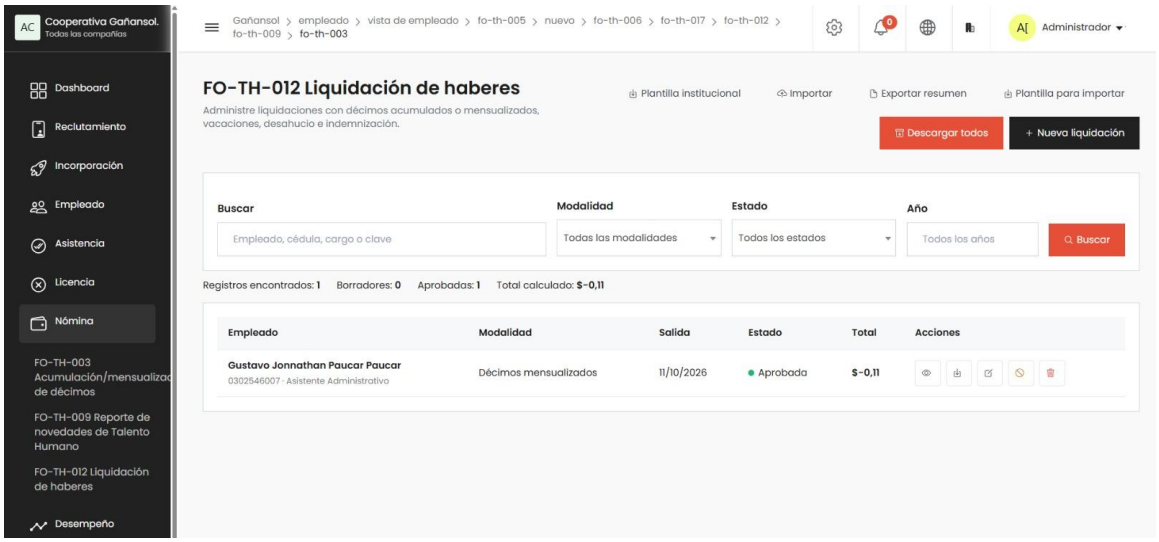
Módulo de Pasantías: control de pasantía FO-TH-014 con institución formadora, tutor, jornadas y actividades registradas



Nota. Autoría propia. Captura del entorno local de pruebas.

Figura A16

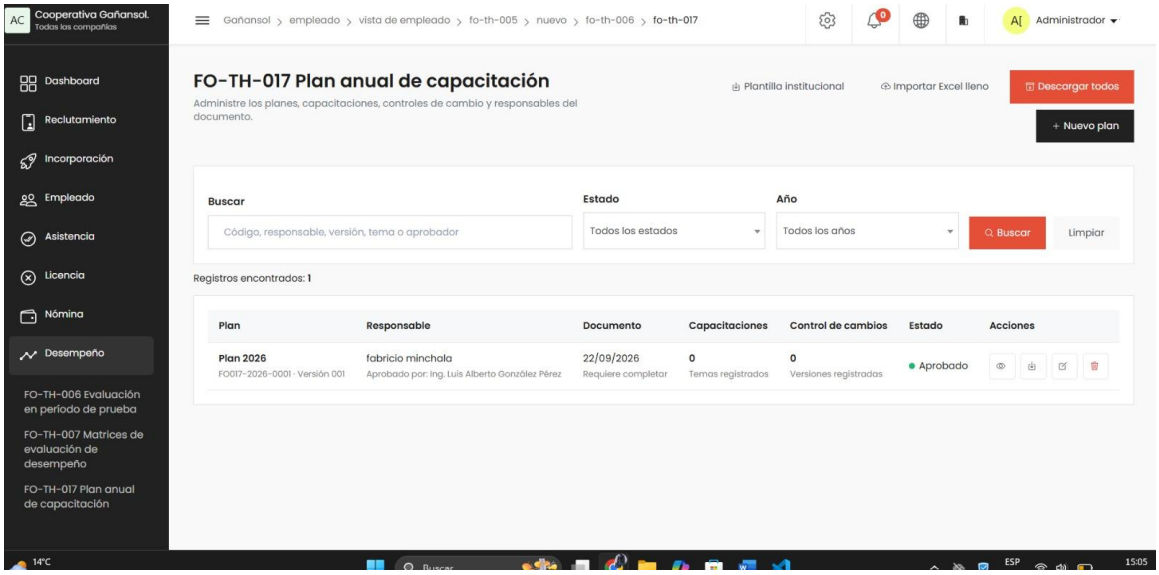
Módulo de Nómina: liquidación de haberes FO-TH-012 con modalidad, estado de aprobación y acciones disponibles



Nota. Autoría propia. Captura del entorno local de pruebas.

Figura A17

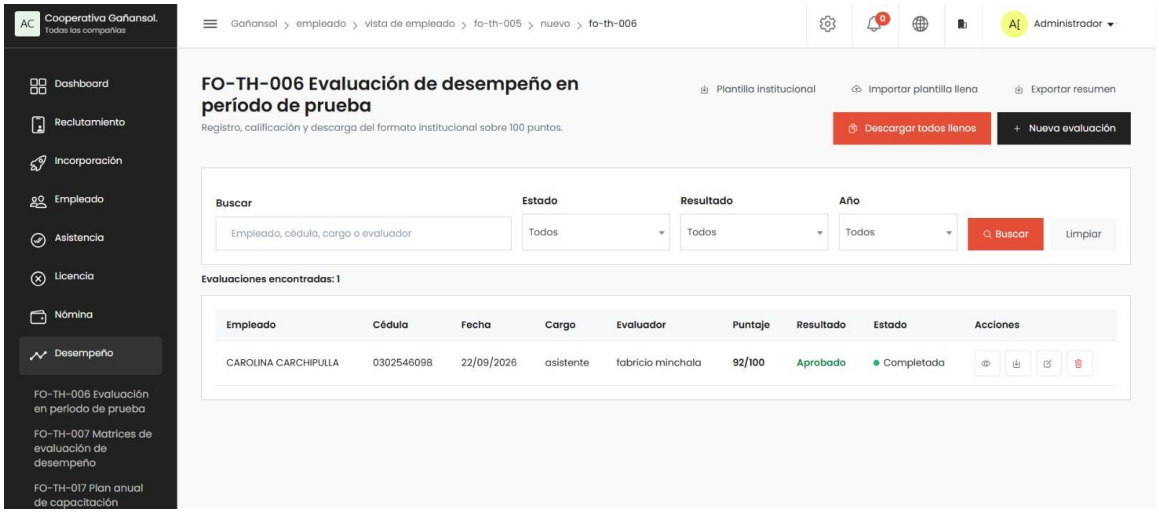
Módulo de Capacitación: plan anual FO-TH-017 con responsable, versión, aprobación y control de cambios



Nota. Autoría propia. Captura del entorno local de pruebas.

Figura A18

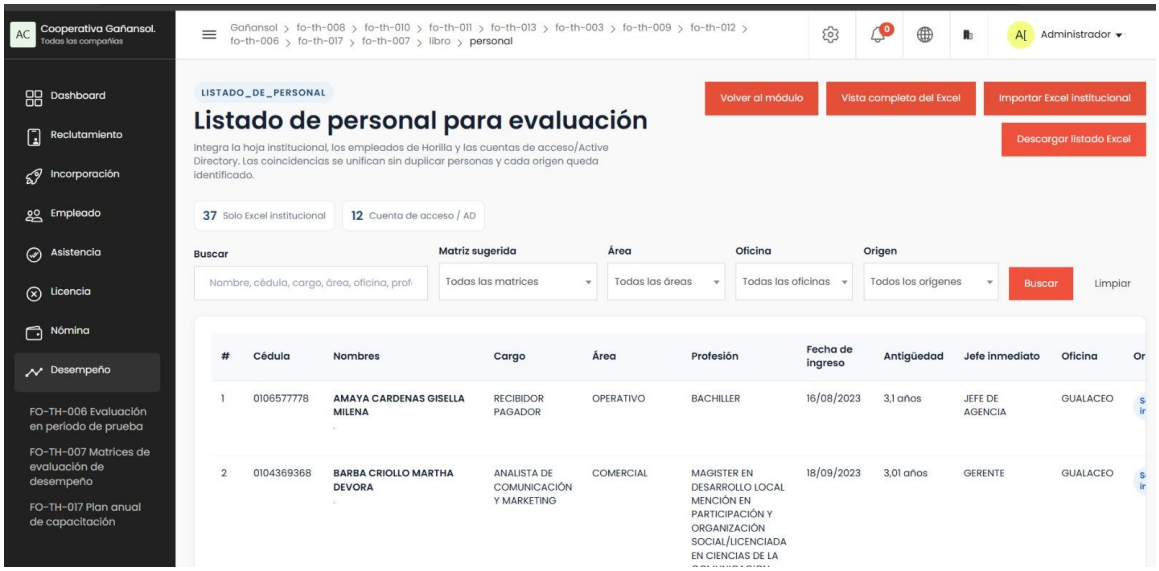
Módulo de Desempeño: evaluación en período de prueba FO-TH-006 con puntaje, resultado y estado del registro



Nota. Autoría propia. Captura del entorno local de pruebas.

Figura A19

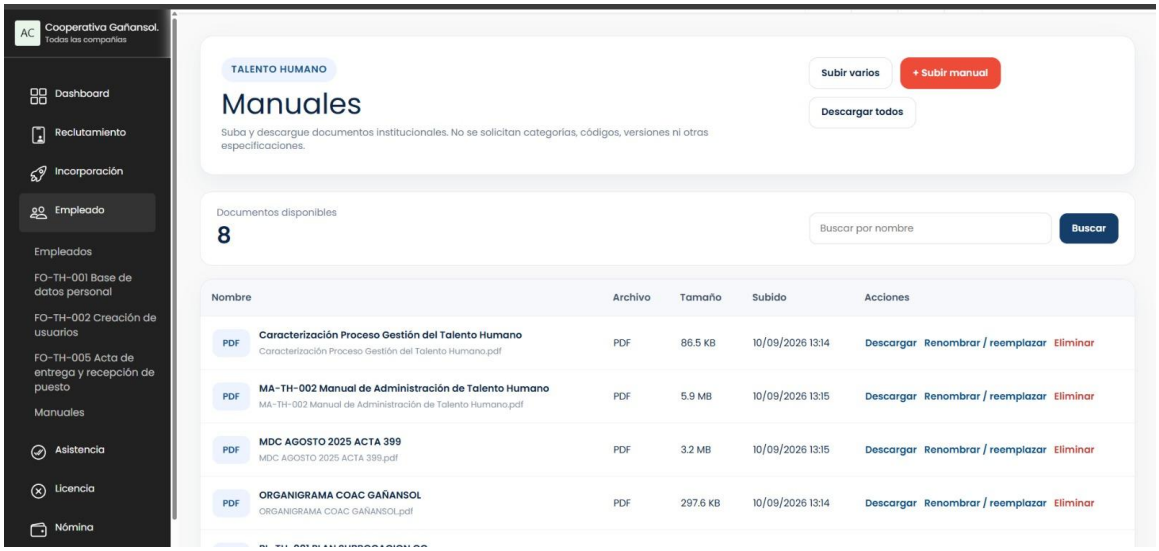
Implementación inicial del FO-TH-007: listado de personal para evaluación y vinculación con cuentas institucionales



Nota. Autoría propia. Evidencia de una funcionalidad básica de la implementación inicial del FO-TH-007. Las funcionalidades adicionales quedan previstas para futuras implementaciones.

Figura A20

Módulo de Empleado: gestión de documentos institucionales con carga, descarga, renombrado y eliminación de archivos



Nota. Autoría propia. Captura del entorno local de pruebas.

ANEXO E. MANUAL DE USUARIO

Guía integrada de módulos, pantallas y procedimientos

1. Inicio de sesión

El acceso al Sistema de Talento Humano de la Cooperativa Gañansol se realiza a través de una pantalla de inicio de sesión segura, en la que cada usuario ingresa con sus credenciales institucionales.

Pasos para iniciar sesión

1. Abrir la dirección web del sistema en el navegador.
2. Ingresar el "Usuario o correo electrónico" asignado por el administrador.
3. Digitar la "Contraseña" en el campo correspondiente (puede mostrarse u ocultarse con el ícono del ojo).
4. Presionar el botón "Inicio de sesión seguro".

Nota: Si el usuario no puede ingresar a su cuenta, puede utilizar la opción "Recuperar contraseña" ubicada debajo del botón de inicio de sesión.

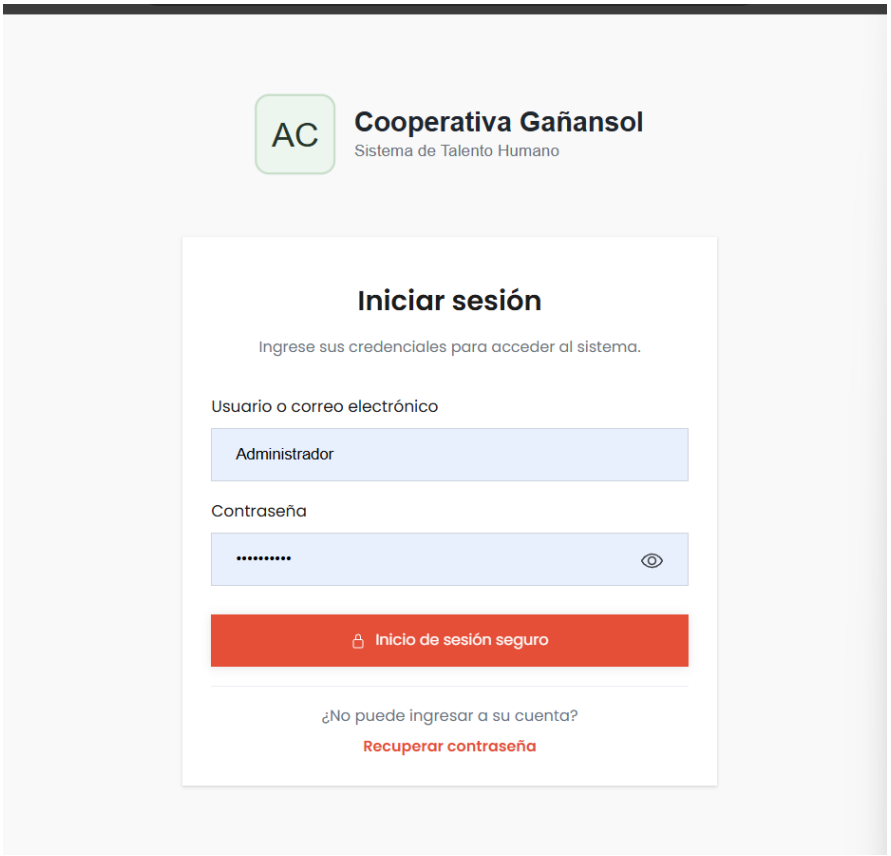


Figura 1. Pantalla de inicio de sesión del Sistema de Talento Humano

2. Pantalla principal y menú de navegación

Una vez autenticado, el sistema muestra en la parte izquierda un menú lateral con los módulos disponibles. El acceso a cada módulo depende de los permisos asignados al usuario que inició sesión.

El menú principal está organizado de la siguiente manera:

- Dashboard: panel general del sistema.
- Reclutamiento: FO-TH-015 Ficha de entrevista.
- Incorporación: FO-TH-004 Formato de inducción/reinducción, FO-TH-014 Control de pasantía.
- Empleado: Empleados, FO-TH-001 Base de datos personal, FO-TH-002 Creación de usuarios, FO-TH-005 Acta de entrega y recepción de puesto, Manuales.
- Asistencia: FO-TH-016 Informe de actividades modalidad virtual, Dispositivos biométricos.
- Licencia: FO-TH-008 Control de vacaciones, FO-TH-010 Solicitud de permiso, FO-TH-011 Solicitud de vacaciones, FO-TH-013 Liquidación de vacaciones.
- Nómina: FO-TH-003 Acumulación/mensualización de décimos, FO-TH-009 Reporte de novedades de Talento Humano, FO-TH-012 Liquidación de haberes.
- Desempeño: FO-TH-006 Evaluación de desempeño en período de prueba, FO-TH-007 Matrices de evaluación de desempeño, FO-TH-017 Plan anual de capacitación.

Nota: En la parte superior derecha de cada pantalla se ubican los accesos a configuración, notificaciones, idioma y el menú del usuario conectado (donde también se puede cerrar sesión).

3. Módulo Empleado

3.1 Listado de empleados

El apartado "Empleados" muestra en formato de tarjetas (o lista) a todos los colaboradores registrados en el sistema, permitiendo buscar, filtrar, agrupar y crear nuevos registros.

1. Utilice el campo "Buscar" para localizar registros por nombre o coincidencia.
2. Presione "+ Crear" para abrir el formulario de registro.
3. Complete los campos solicitados y guarde la información.
4. En la columna "Acciones" puede ver, editar, descargar o eliminar cada registro individualmente.

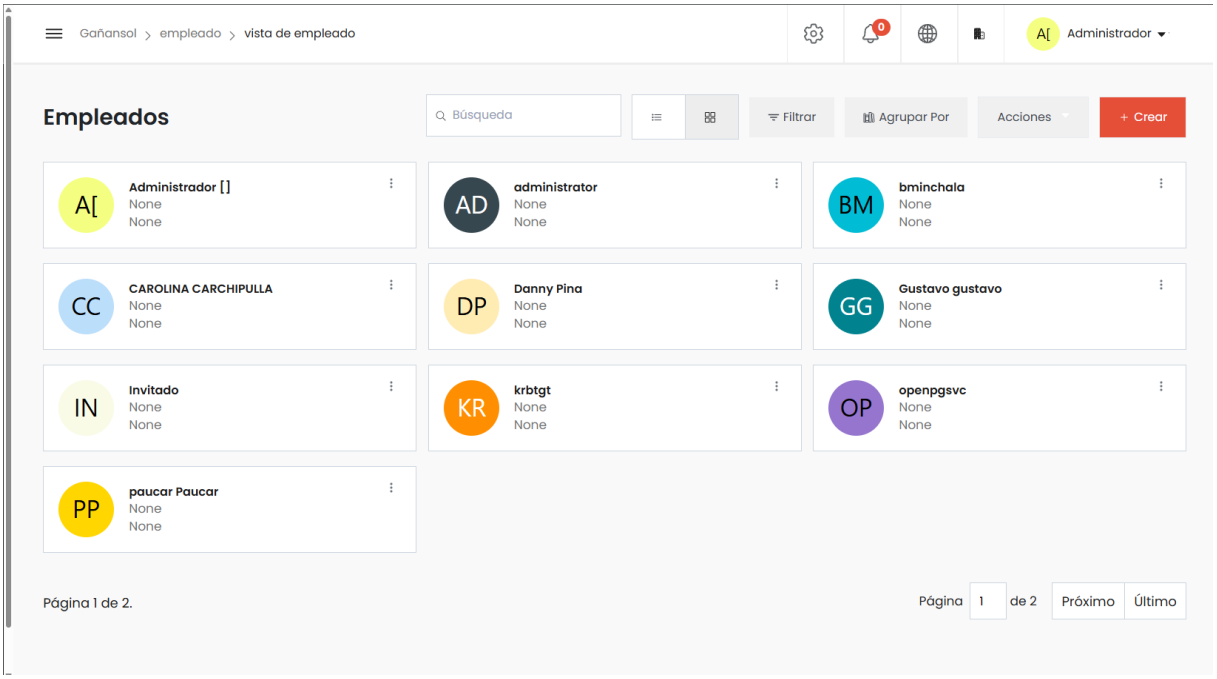


Figura 2. Listado de empleados registrados en el sistema.

3.2 Registro de un nuevo empleado

Al crear o editar un empleado se despliega el formulario "Registro de empleado", donde se debe completar la información organizada por secciones.

Acceso y control

- Insignia ID (código de identificación, ej. "PEP0002").
- Nombre distinguido de AD (Active Directory).
- Última conexión y estado "¿Está en línea?".

Datos personales

- Primer nombre y apellido.
- Correo electrónico y teléfono (obligatorios).
- Fecha de nacimiento, género, estado civil y número de hijos.

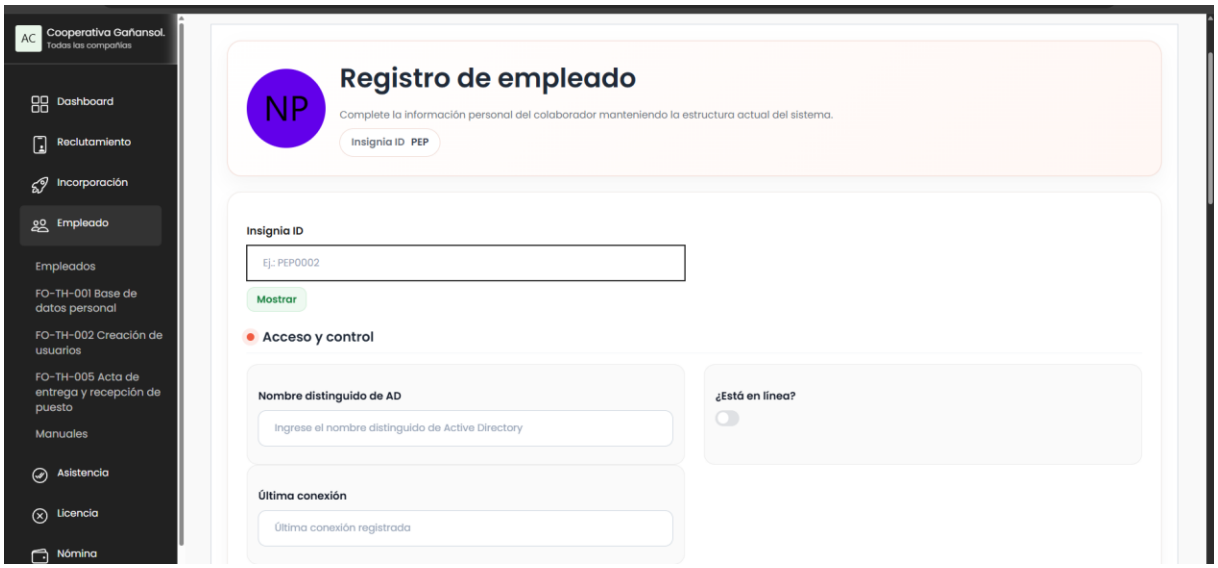
Ubicación y contacto

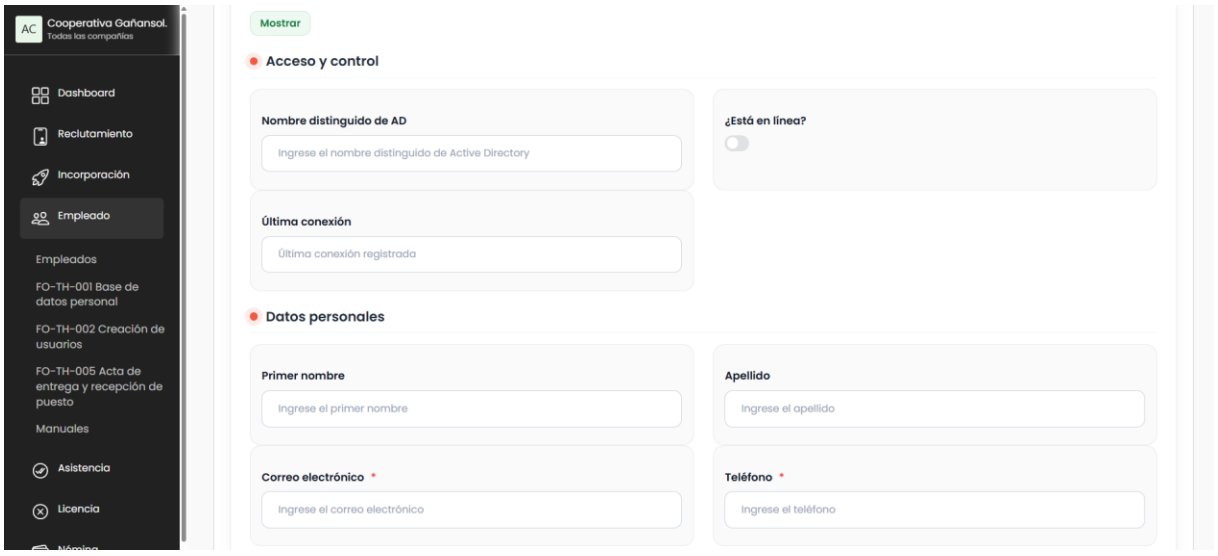
- Dirección, país, provincia o estado, ciudad y código postal.

Información complementaria

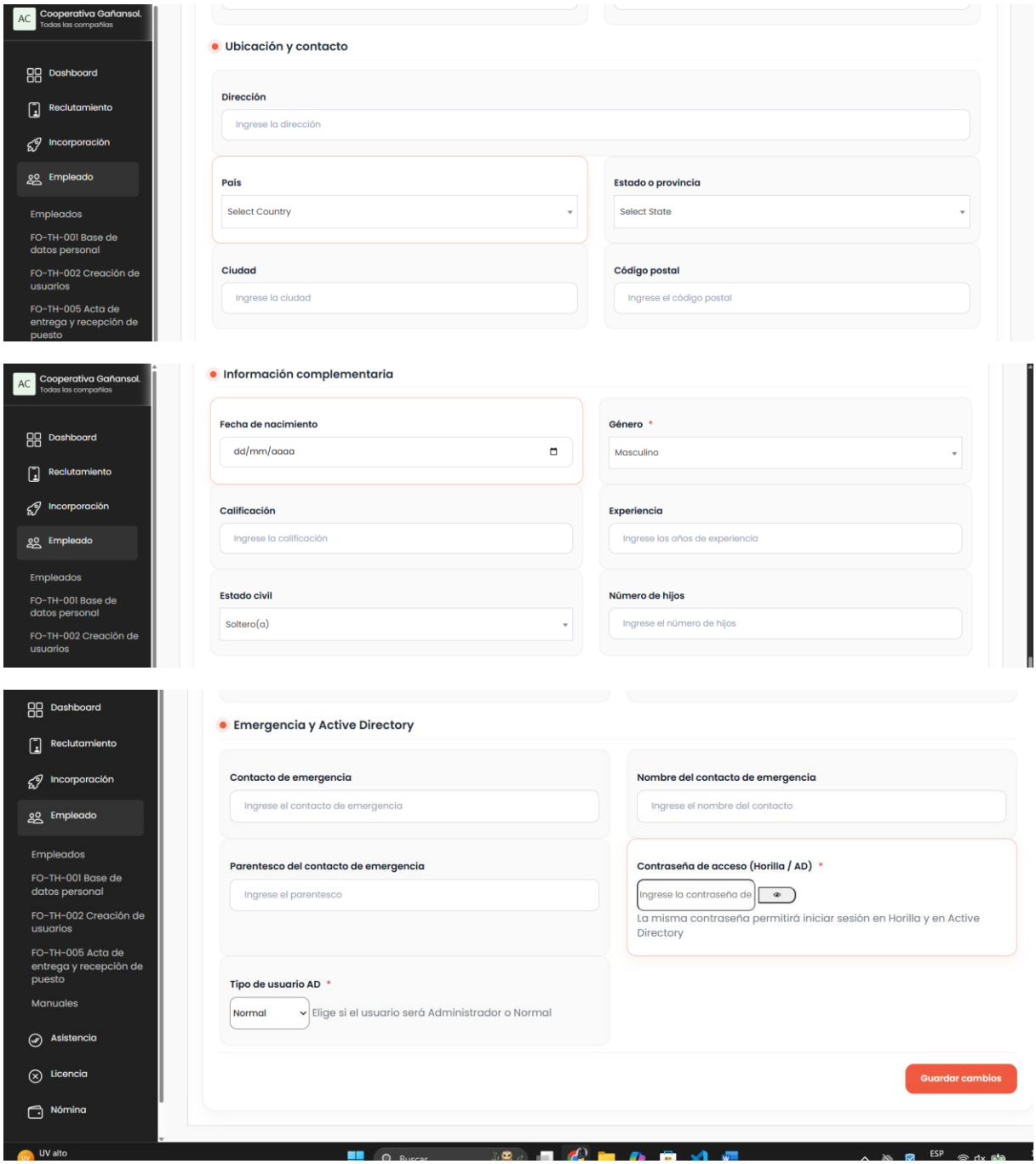
- Calificación y experiencia.

5. Una vez completados todos los campos, presionar "Guardar cambios" para registrar al colaborador.









Figuras 3–8. Formulario de registro de empleado, con las secciones de acceso y control, datos personales, ubicación e información complementaria.

4. FO-TH-001 Base de datos personal

Este módulo administra la información principal de los colaboradores registrados en el sistema (submenú del apartado Empleado).

1. Utilice el campo "Buscar" para localizar registros por cédula, número, nombre, correo o teléfono.
2. Aplique los filtros disponibles (Agencia, Estado) y presione "Buscar" para refinar el listado.
3. Presione "+ Nuevo registro" para abrir el formulario de registro.
4. Complete los campos solicitados y guarde la información.
5. Use las opciones "Exportar Excel" / "Exportar Word" o "Descargar todos" para obtener respaldos del listado.
6. En la columna "Acciones" puede ver, editar, descargar o eliminar cada registro individualmente.

El listado muestra: N.º, Cédula, Nombres, Apellidos, Correo, Teléfono, Agencia y Estado del empleado. También se puede descargar la "Plantilla" o "Importar" registros de forma masiva.

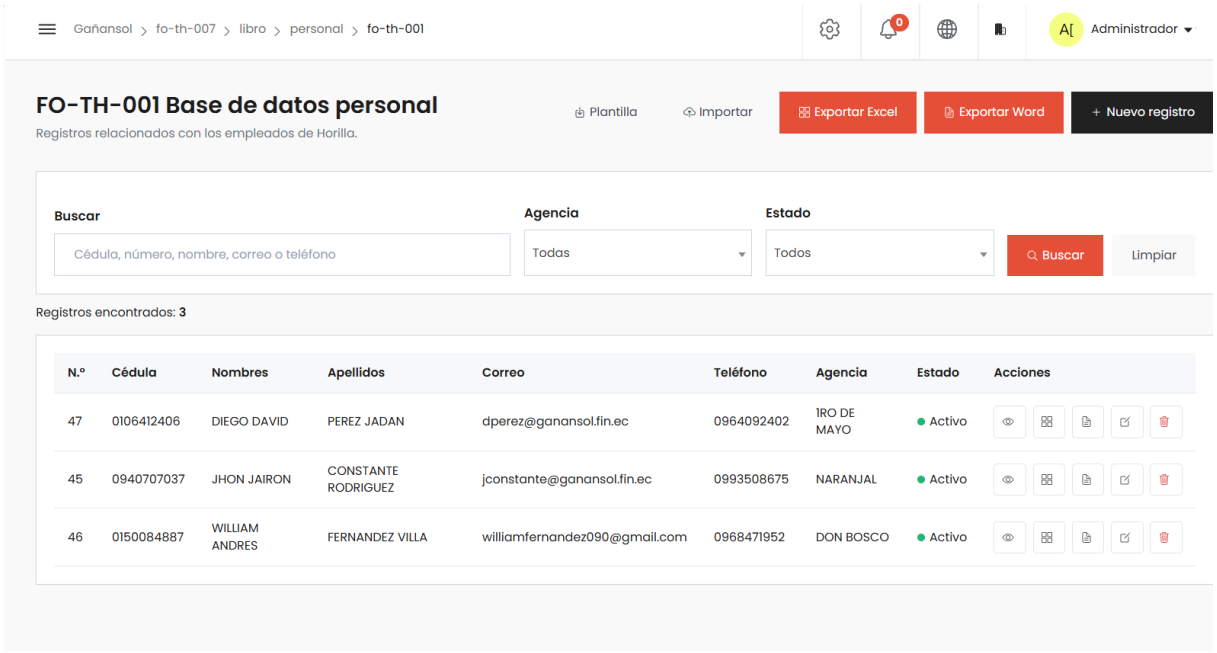
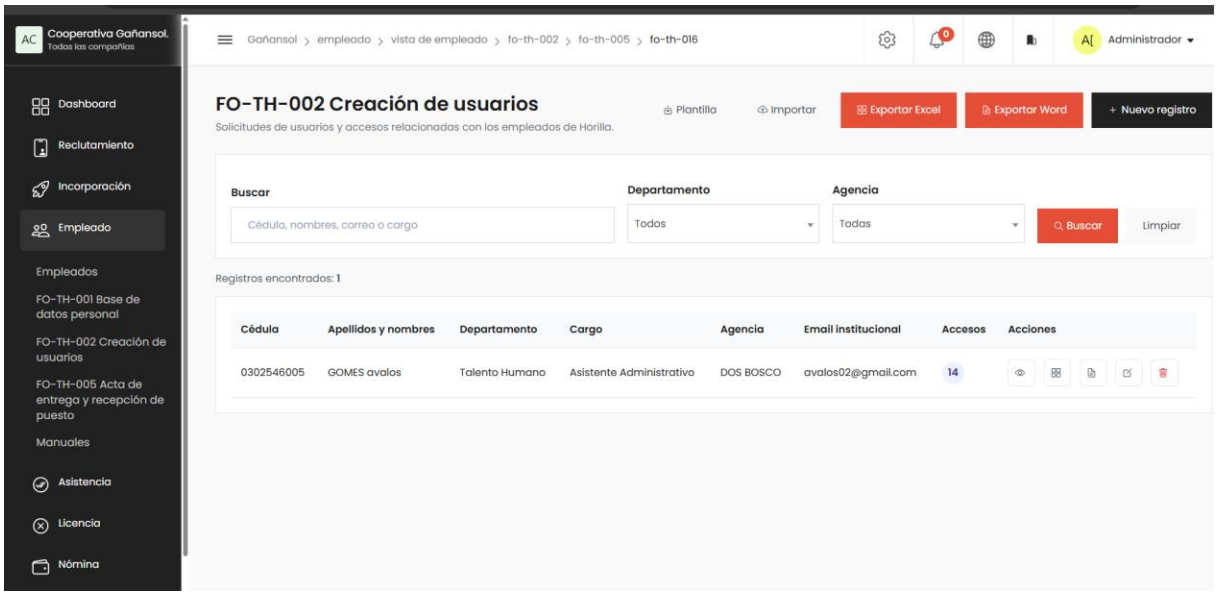
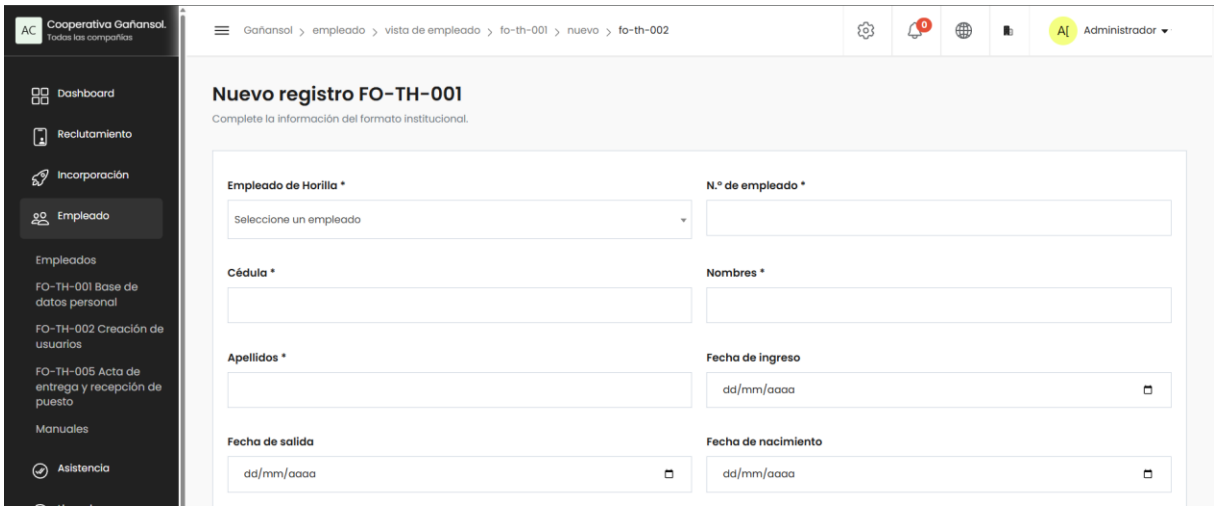
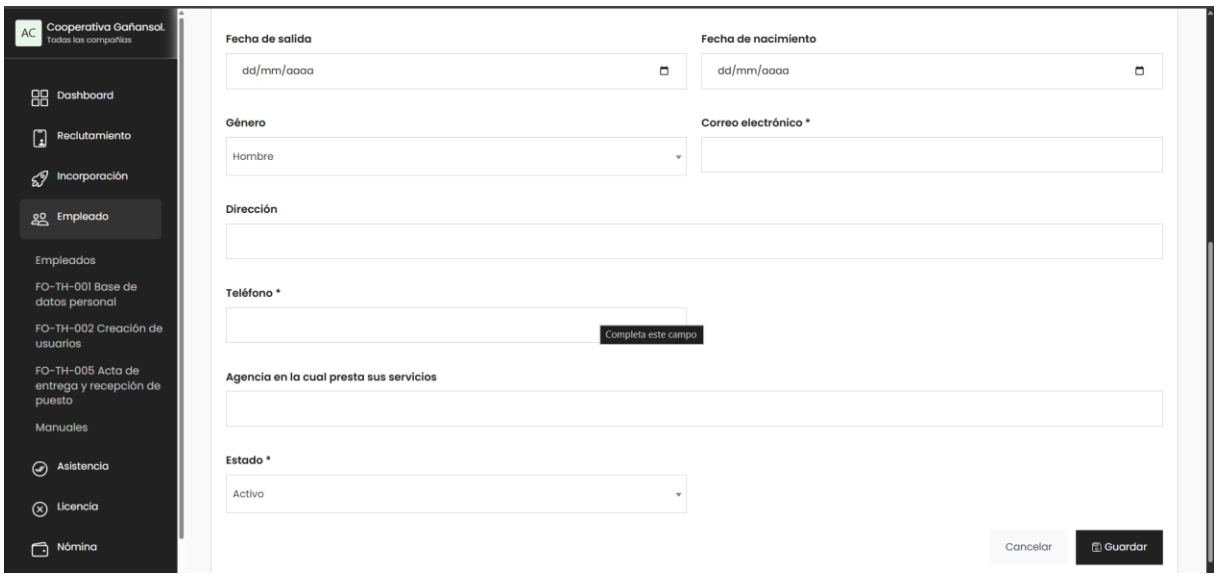


Figura 9. Listado FO-TH-001 Base de datos personal, con buscador y filtros por agencia y estado.







Figuras 10–12. Vista de detalle, edición y exportación de un registro de la base de datos personal.

5. FO-TH-002 Creación de usuarios

Permite gestionar las solicitudes de usuario y accesos asociadas a cada empleado dentro del sistema.

1. Utilice el campo "Buscar" para localizar registros por cédula, nombres, correo o cargo.
2. Aplique los filtros disponibles (Departamento, Agencia) y presione "Buscar" para refinar el listado.
3. Presione "+ Nuevo registro" para abrir el formulario de registro.
4. Complete los campos solicitados y guarde la información.
5. Use las opciones "Exportar Excel" / "Exportar Word" o "Descargar todos" para obtener respaldos del listado.
6. En la columna "Acciones" puede ver, editar, descargar o eliminar cada registro individualmente.

El listado muestra: Cédula, Apellidos y nombres, Departamento, Cargo, Agencia, Email institucional y número de "Accesos" solicitados. El administrador define, en el formulario de registro, los accesos que tendrá cada usuario dentro del sistema.

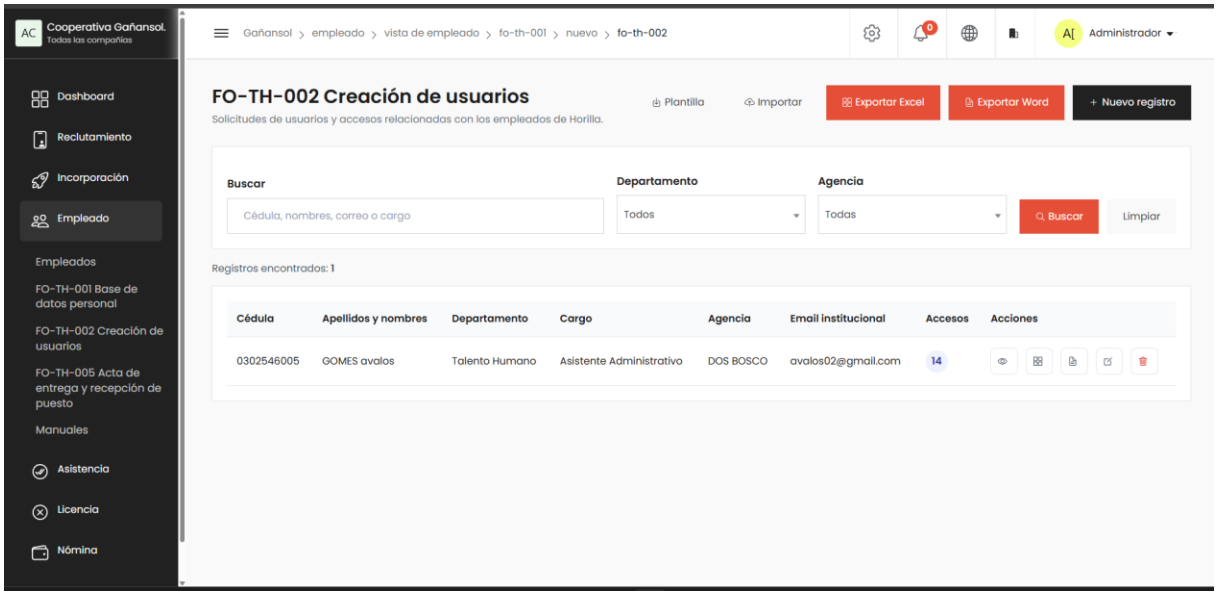
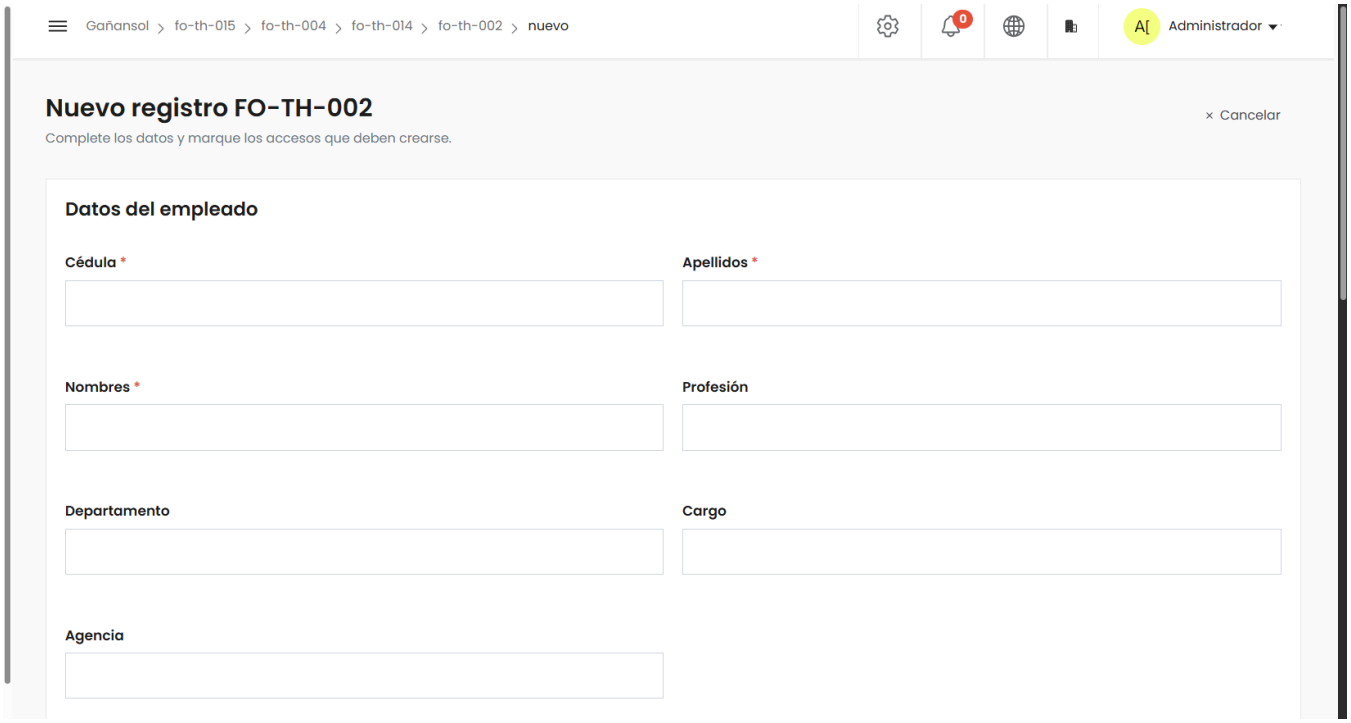
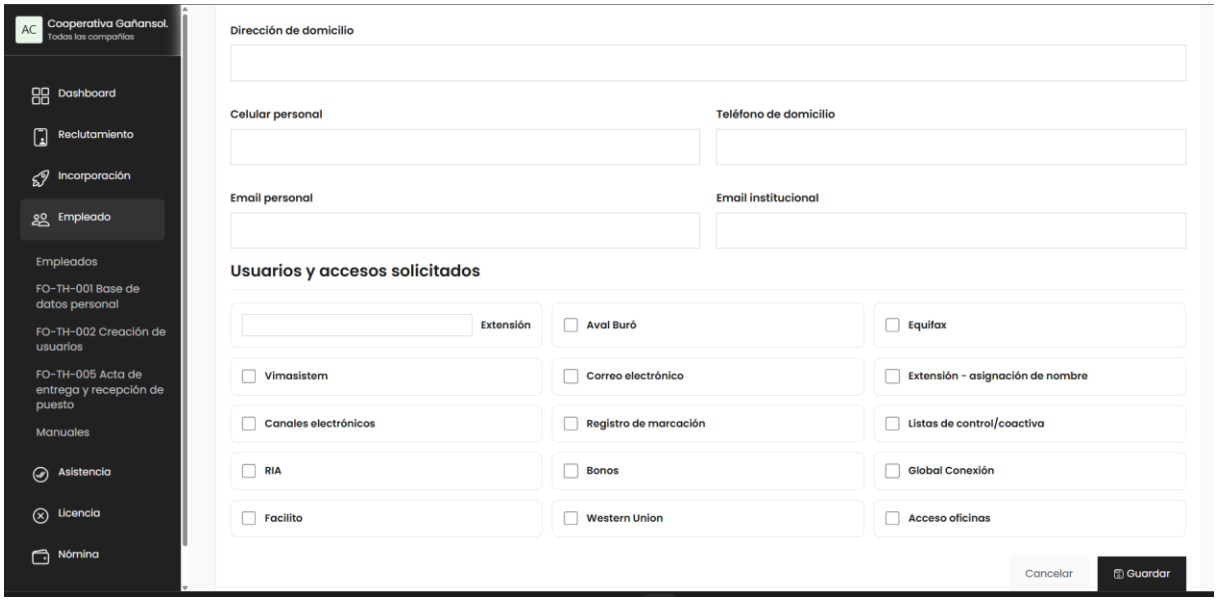


Figura 13. Listado FO-TH-002 Creación de usuarios.





Figuras 14–15. Formulario de solicitud de usuario, con datos del empleado y accesos a otorgar.

6. FO-TH-003 Acumulación / mensualización de décimos

Módulo del menú "Nómina" que gestiona la acumulación o mensualización de los décimos tercero y cuarto de cada colaborador.

1. Utilice el campo "Buscar" para localizar registros por empleado, cédula o cargo.
2. Aplique los filtros disponibles (modalidad, año) y presione "Buscar" para refinar el listado.
3. Presione "+ Nuevo registro" para abrir el formulario de registro.
4. Complete los campos solicitados y guarde la información.
5. Use las opciones "Exportar Excel" / "Exportar Word" o "Descargar todos" para obtener respaldos del listado.
6. En la columna "Acciones" puede ver, editar, descargar o eliminar cada registro individualmente.

Permite registrar y consultar solicitudes relacionadas con el décimo tercero, décimo cuarto y la modalidad seleccionada (acumulado o mensualizado), además de generar reportes.



Figura 16. Listado de solicitudes de décimos (acumulación/mensualización).



Figura 17. Registro y detalle de una solicitud de décimos.

7. FO-TH-004 Formato de inducción / reinducción

Módulo del menú "Incorporación" que permite registrar los procesos de inducción y reinducción del personal.

1. Utilice el campo "Buscar" para localizar registros por empleado, cédula o cargo.
2. Aplique los filtros disponibles (estado, tipo de proceso) y presione "Buscar" para refinar el listado.
3. Presione "+ Nuevo registro" para abrir el formulario de registro.
4. Complete los campos solicitados y guarde la información.
5. Use las opciones "Exportar Excel" / "Exportar Word" o "Descargar todos" para obtener respaldos del listado.
6. En la columna "Acciones" puede ver, editar, descargar o eliminar cada registro individualmente.

En el formulario se puede ingresar: empleado, tipo de proceso, fecha, estado, cargo, temas tratados y documentos entregados durante la inducción.

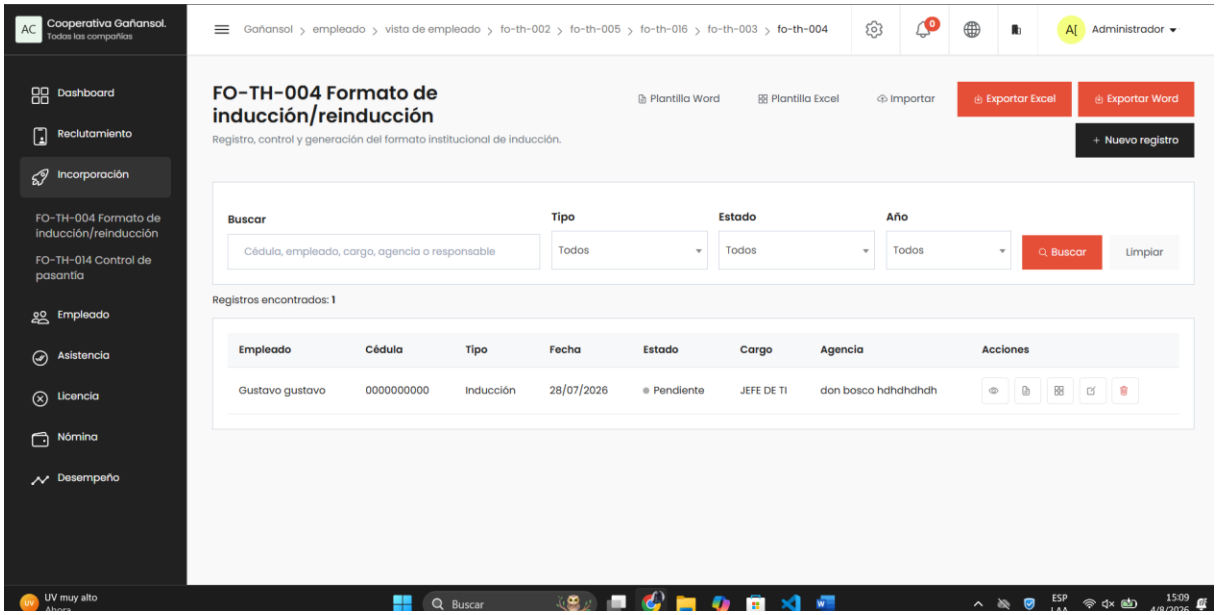
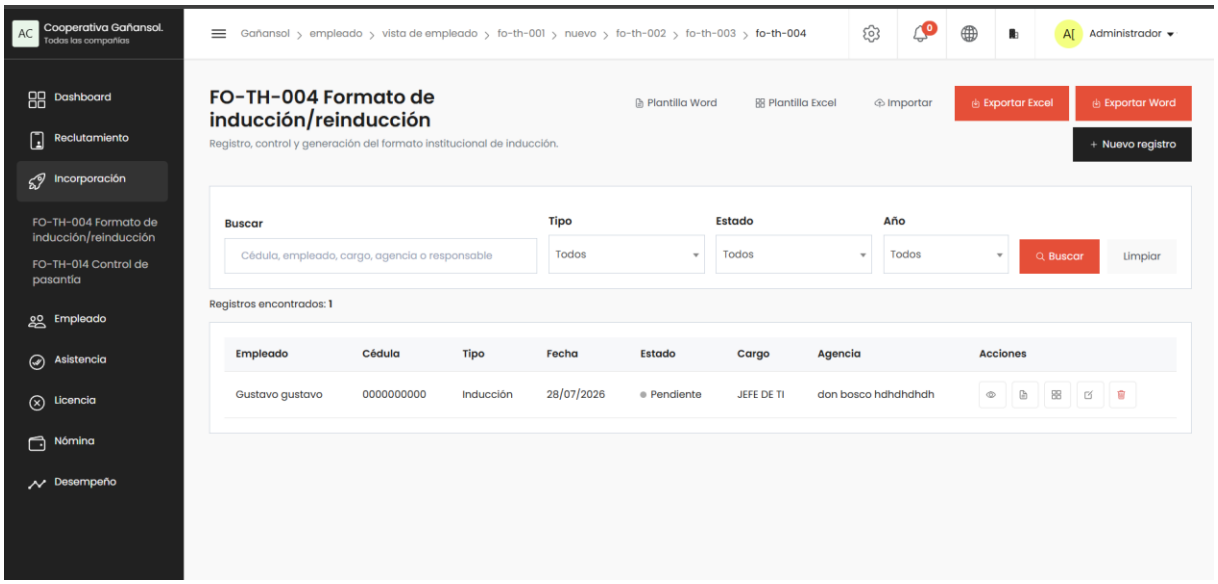
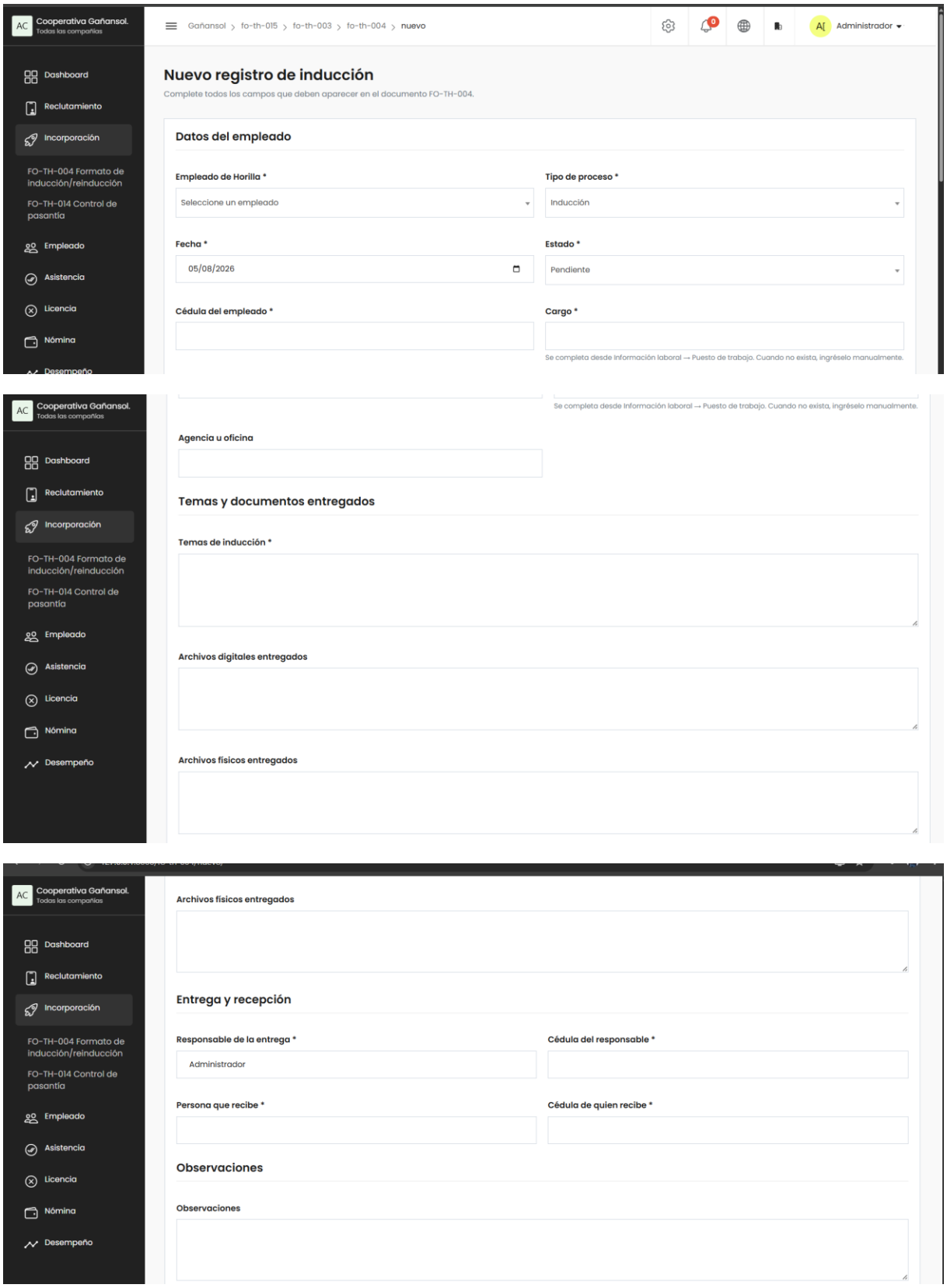


Figura 18. Listado de procesos de inducción/reinducción registrados.





Figuras 19–22. Formulario de registro de inducción/reinducción, con los temas tratados y documentos entregados.

8. FO-TH-005 Acta de entrega y recepción de puesto

Submódulo del apartado "Empleado" que permite documentar la entrega y recepción de un puesto de trabajo.

1. Utilice el campo "Buscar" para localizar registros por empleado, cédula o cargo.
2. Aplique los filtros disponibles (estado) y presione "Buscar" para refinar el listado.
3. Presione "+ Nuevo registro" para abrir el formulario de registro.
4. Complete los campos solicitados y guarde la información.
5. Use las opciones "Exportar Excel" / "Exportar Word" o "Descargar todos" para obtener respaldos del listado.
6. En la columna "Acciones" puede ver, editar, descargar o eliminar cada registro individualmente.

El acta registra: datos del empleado, motivo de entrega, archivos digitales y físicos, equipos o muebles entregados, pendientes, y los responsables de entrega y recepción.

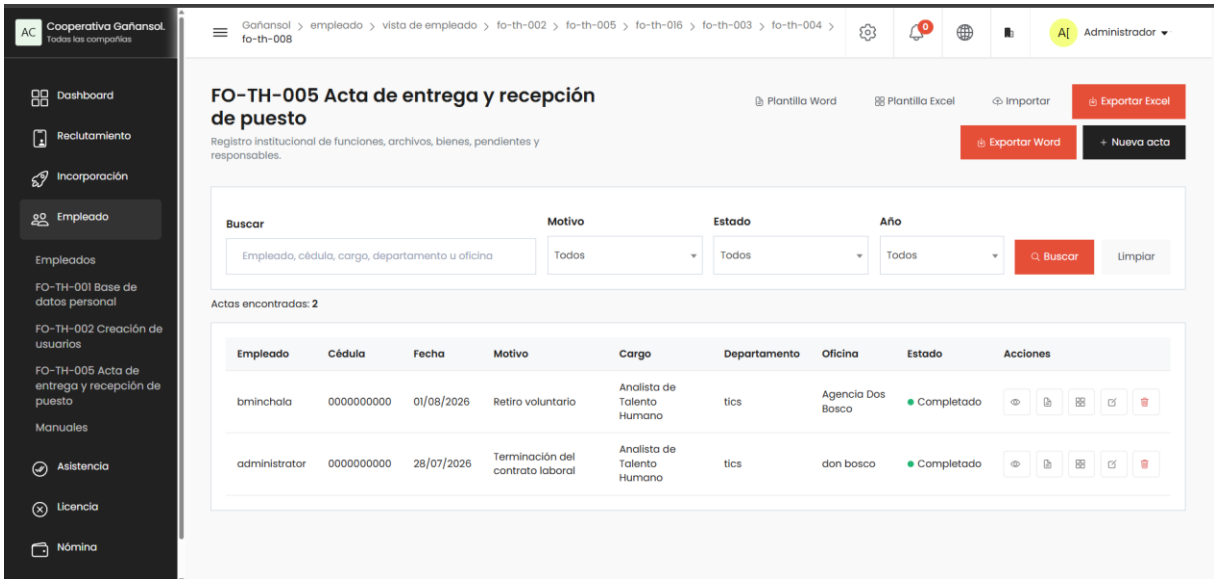
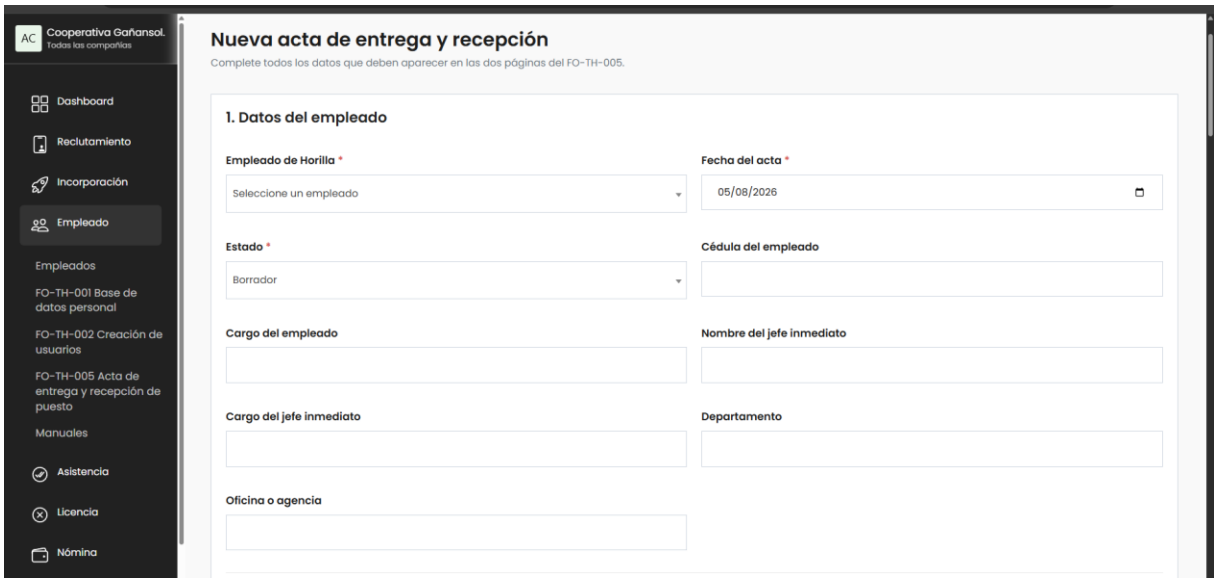
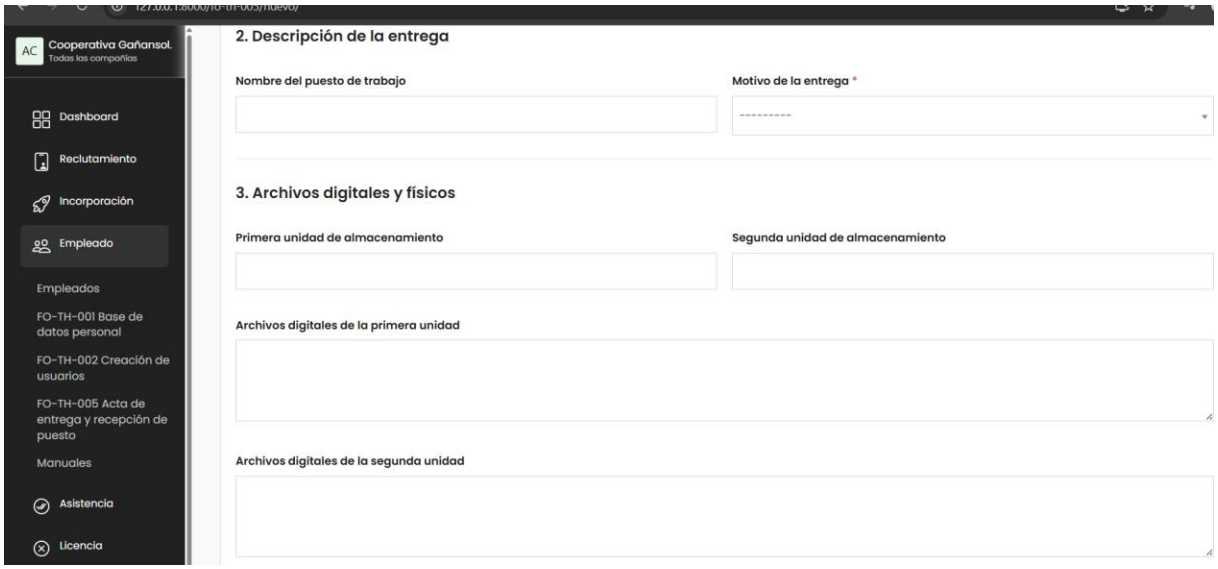
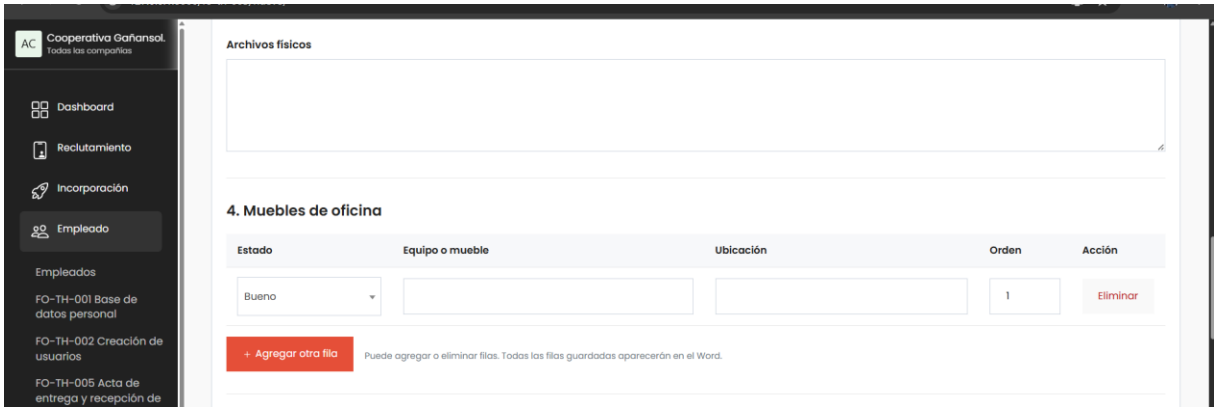


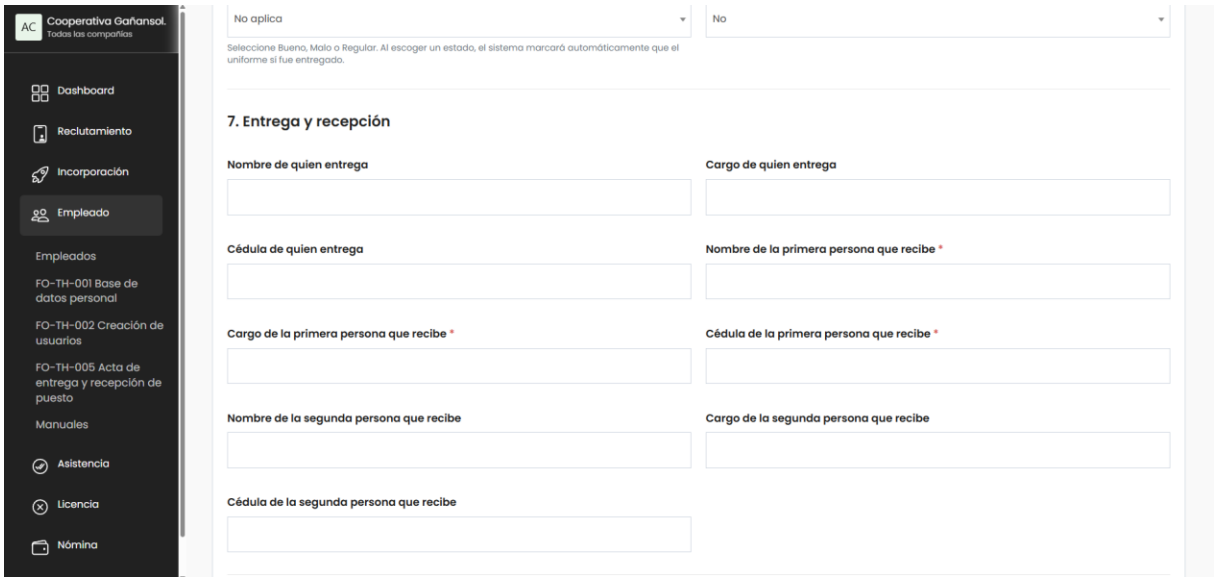
Figura 23. Listado de actas de entrega y recepción de puesto.

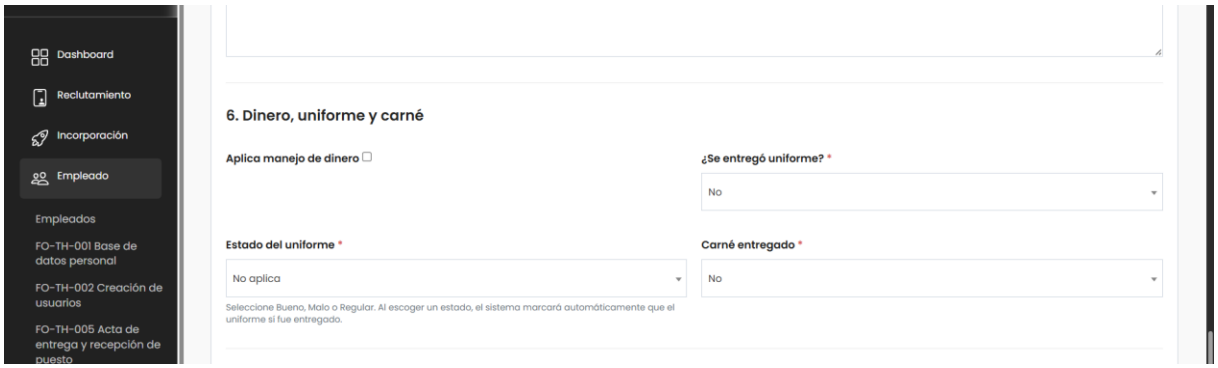



Cooperativa Gañansol.
Todas las compañías
Dashboard
Reclutamiento
Incorporación
Empleado
Empleados
FO-TH-001 Base de datos personal
FO-TH-002 Creación de usuarios
FO-TH-005 Acta de entrega y recepción de puesto
Manuales
Asistencia
Licencia
2. Descripción de la entrega
Nombre del puesto de trabajo
Motivo de la entrega *
----
3. Archivos digitales y físicos
Primera unidad de almacenamiento
Segunda unidad de almacenamiento
Archivos digitales de la primera unidad
Archivos digitales de la segunda unidad

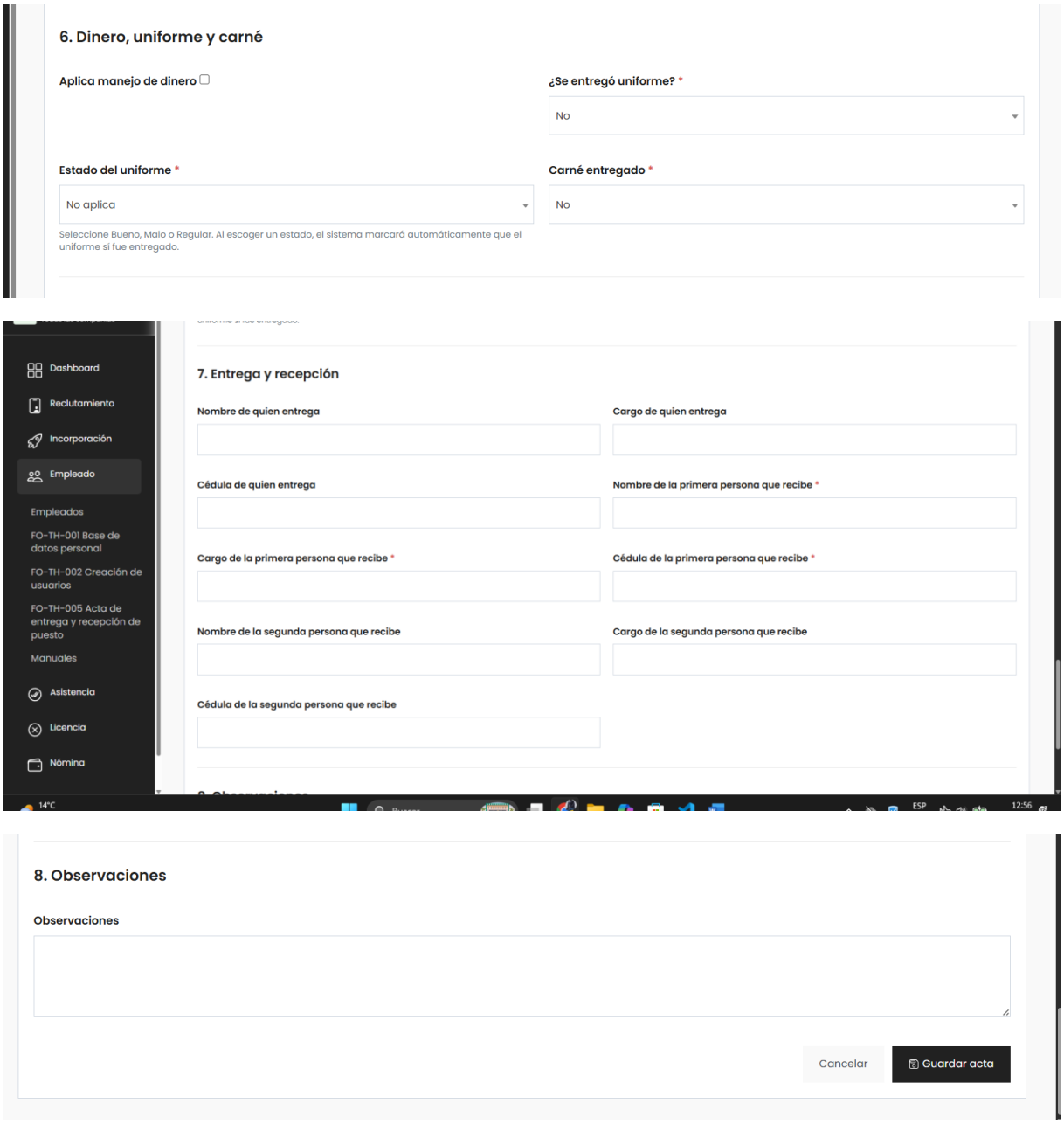

Cooperativa Gañansol.
Todas las compañías
Dashboard
Reclutamiento
Incorporación
Empleado
Empleados
FO-TH-001 Base de datos personal
FO-TH-002 Creación de usuarios
FO-TH-005 Acta de entrega y recepción de
Archivos físicos
4. Muebles de oficina
Estado
Equipo o mueble
Ubicación
Orden
Acción
Bueno
1
Eliminar
+ Agregar otra fila
Puede agregar o eliminar filas. Todas las filas guardadas aparecerán en el Word.


Cooperativa Gañansol.
Todas las compañías
Dashboard
Reclutamiento
Incorporación
Empleado
Empleados
FO-TH-001 Base de datos personal
FO-TH-002 Creación de usuarios
FO-TH-005 Acta de entrega y recepción de puesto
Manuales
Asistencia
Licencia
Nómina
5. Suministros, pendientes y otros asuntos
Suministros de oficina
Varios
Asuntos procesados
Pendientes


AC
Cooperativa Gañansol.
Todas las compañías
Dashboard
Reclutamiento
Incorporación
Empleado
Empleados
FO-TH-001 Base de datos personal
FO-TH-002 Creación de usuarios
FO-TH-005 Acta de entrega y recepción de puesto
Manuales
Asistencia
Licencia
Nómina
No aplica
No
Seleccione Bueno, Malo o Regular. Al escoger un estado, el sistema marcará automáticamente que el uniforme si fue entregado.
7. Entrega y recepción
Nombre de quien entrega
Cargo de quien entrega
Cédula de quien entrega
Nombre de la primera persona que recibe *
Cargo de la primera persona que recibe *
Cédula de la primera persona que recibe *
Nombre de la segunda persona que recibe
Cargo de la segunda persona que recibe
Cédula de la segunda persona que recibe


Dashboard
Reclutamiento
Incorporación
Empleado
Empleados
FO-TH-001 Base de datos personal
FO-TH-002 Creación de usuarios
FO-TH-005 Acta de entrega y recepción de puesto
6. Dinero, uniforme y carné
Aplica manejo de dinero
¿Se entregó uniforme? *
No
Estado del uniforme *
No aplica
Carné entregado *
No
Seleccione Bueno, Malo o Regular. Al escoger un estado, el sistema marcará automáticamente que el uniforme si fue entregado.


AC
Cooperativa Gañansol.
Todas las compañías
Dashboard
Reclutamiento
Incorporación
Empleado
Empleados
FO-TH-001 Base de datos personal
FO-TH-002 Creación de usuarios
FO-TH-005 Acta de entrega y recepción de puesto
Manuales
Asistencia
Licencia
Nómina
5. Suministros, pendientes y otros asuntos
Suministros de oficina
Varios
Asuntos procesados
Pendientes



Figuras 24–33. Formulario completo del acta: datos del empleado, motivo, bienes entregados, pendientes y responsables.

9. FO-TH-006 Evaluación de desempeño en período de prueba

Módulo del menú "Desempeño" para el registro, calificación y descarga del formato institucional de evaluación sobre 100 puntos.

1. Utilice el campo "Buscar" para localizar registros por empleado, cédula, cargo o evaluador.
2. Aplique los filtros disponibles (estado, resultado, año) y presione "Buscar" para refinar el listado.
3. Presione "+ Nueva evaluación" para abrir el formulario de registro.
4. Complete los campos solicitados y guarde la información.
5. Use las opciones "Exportar Excel" / "Exportar Word" o "Descargar todos" para obtener respaldos del listado.
6. En la columna "Acciones" puede ver, editar, descargar o eliminar cada registro individualmente.

El listado muestra Empleado, Cédula, Fecha, Cargo, Evaluador, Puntaje, Resultado (por ejemplo "Aprobado") y Estado (por ejemplo "Completada"). También se dispone de "Ver/descargar plantilla", "Plantilla para importar", "Importar Excel", "Exportar resumen" y "Descargar todos llenos".

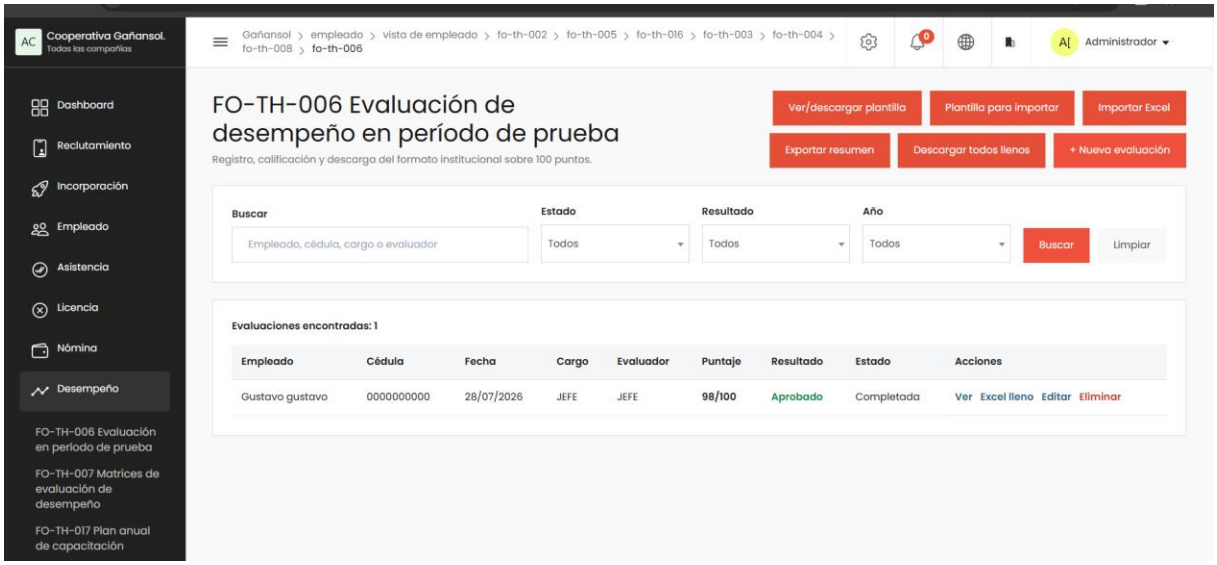
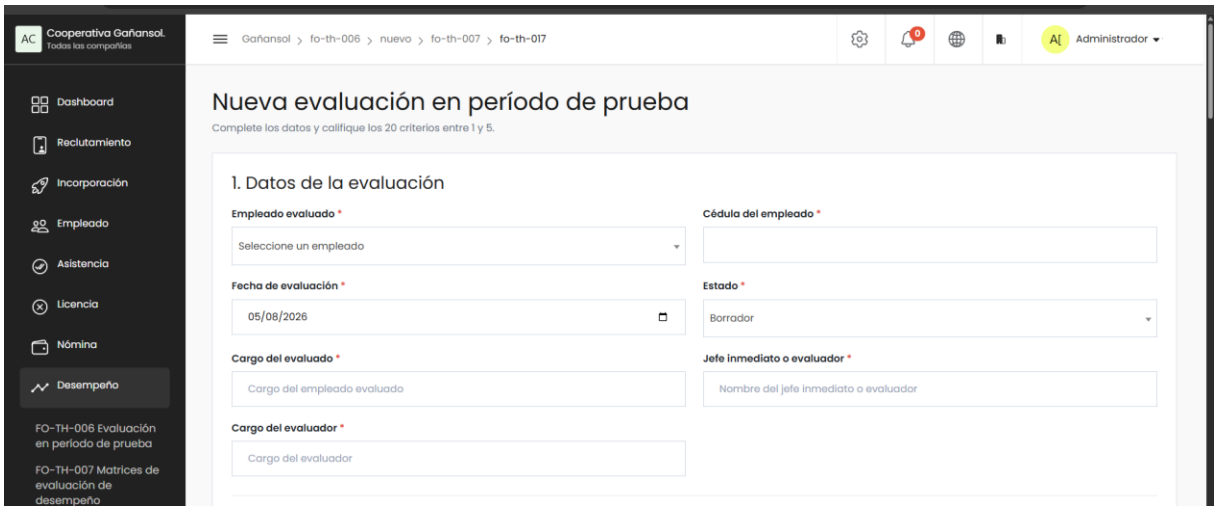


Figura 34. Listado FO-TH-006 Evaluación de desempeño en período de prueba.



2. Criterios de evaluación

Escala: 1 Muy bajo, 2 Bajo, 3 Moderado, 4 Alto y 5 Muy alto.

ORIENTACIÓN A RESULTADOS	
1. Termina su trabajo oportunamente	◉ --------- ○ Muy bajo ○ Bajo ○ Moderado ○ Alto ○ Muy alto
2. Cumple con las tareas que se le encomienda	◉ --------- ○ Muy bajo ○ Bajo ○ Moderado ○ Alto ○ Muy alto
3. Realiza un volumen adecuado de trabajo	◉ --------- ○ Muy bajo ○ Bajo ○ Moderado ○ Alto ○ Muy alto

CALIDAD	
4. No comete errores en el trabajo	◉ --------- ○ Muy bajo ○ Bajo ○ Moderado ○ Alto ○ Muy alto
5. Hace uso racional de los recursos	◉ --------- ○ Muy bajo ○ Bajo ○ Moderado ○ Alto ○ Muy alto
6. No requiere supervisión frecuente	◉ --------- ○ Muy bajo ○ Bajo ○ Moderado ○ Alto ○ Muy alto
7. Se muestra profesional en el trabajo	◉ --------- ○ Muy bajo ○ Bajo ○ Moderado ○ Alto ○ Muy alto
8. Se muestra profesional en el trabajo	◉ --------- ○ Muy bajo ○ Bajo ○ Moderado ○ Alto ○ Muy alto

RELACIONES INTERPERSONALES	
9. Se muestra cortés con el personal y con sus compañeros	◉ --------- ○ Muy bajo ○ Bajo ○ Moderado ○ Alto ○ Muy alto
10. Brinda una adecuada orientación a sus compañeros	◉ --------- ○ Muy bajo ○ Bajo ○ Moderado ○ Alto ○ Muy alto
11. Evita los conflictos dentro del trabajo	◉ --------- ○ Muy bajo ○ Bajo ○ Moderado ○ Alto ○ Muy alto

TRABAJO EN EQUIPO

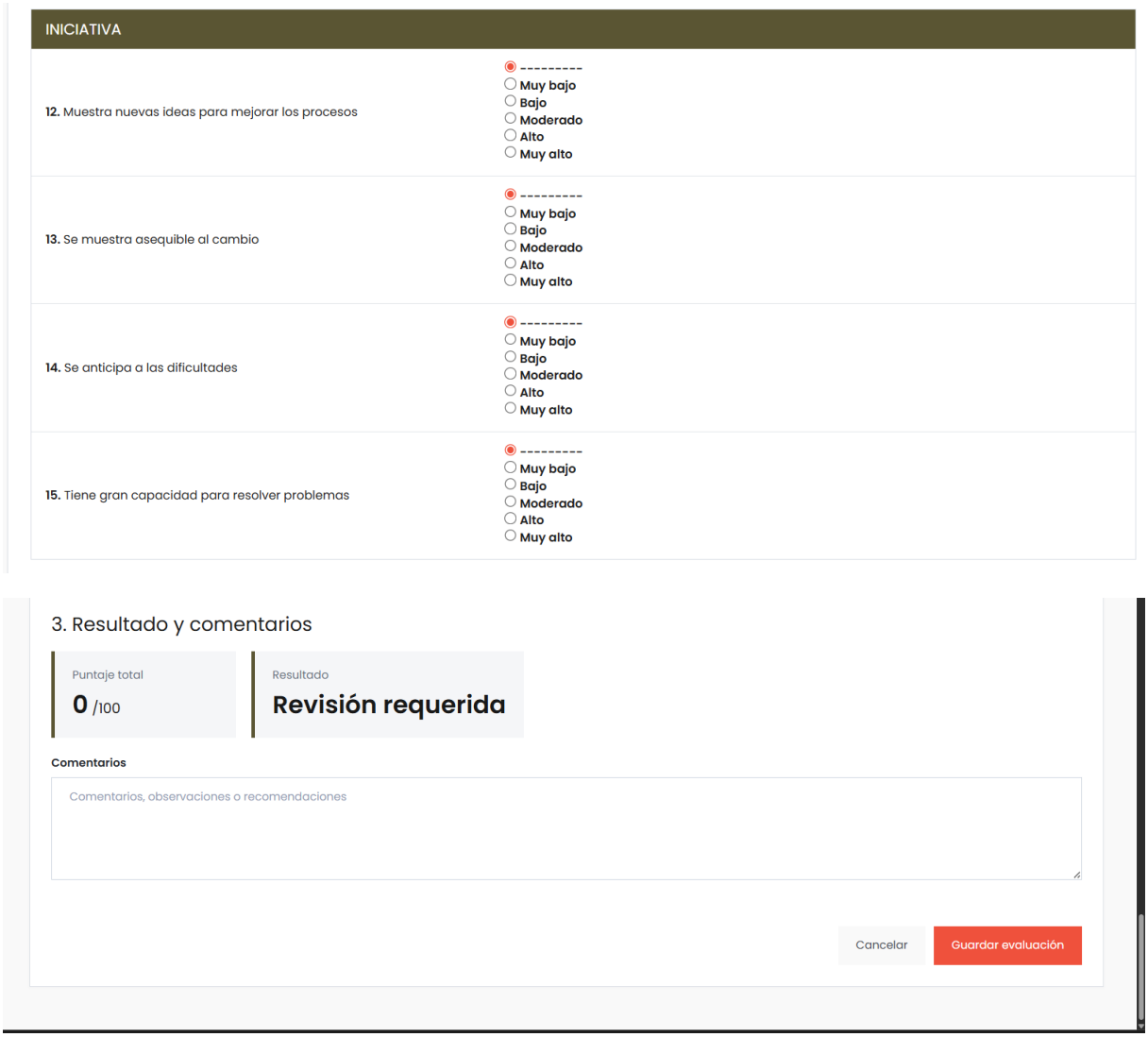
16. Muestra aptitud para integrarse al equipo	○ --------- ○ Muy bajo ○ Bajo ○ Moderado ○ Alto ○ Muy alto
17. Se identifica fácilmente con los objetivos del equipo	○ --------- ○ Muy bajo ○ Bajo ○ Moderado ○ Alto ○ Muy alto

ORGANIZACIÓN

18. Planifica sus actividades	○ --------- ○ Muy bajo ○ Bajo ○ Moderado ○ Alto ○ Muy alto
19. Hace uso de indicadores	○ --------- ○ Muy bajo ○ Bajo ○ Moderado ○ Alto ○ Muy alto
20. Se preocupa por alcanzar las metas	○ --------- ○ Muy bajo ○ Bajo

INICIATIVA

12. Muestra nuevas ideas para mejorar los procesos	○ --------- ○ Muy bajo ○ Bajo ○ Moderado ○ Alto ○ Muy alto
13. Se muestra asequible al cambio	○ --------- ○ Muy bajo ○ Bajo ○ Moderado ○ Alto ○ Muy alto
14. Se anticipa a las dificultades	○ --------- ○ Muy bajo ○ Bajo ○ Moderado ○ Alto ○ Muy alto
15. Tiene gran capacidad para resolver problemas	○ --------- ○ Muy bajo ○ Bajo ○ Moderado ○ Alto ○ Muy alto



Figuras 35–42. Formulario de evaluación, criterios de calificación y resultado final.

10. FO-TH-007 Matrices de evaluación de desempeño

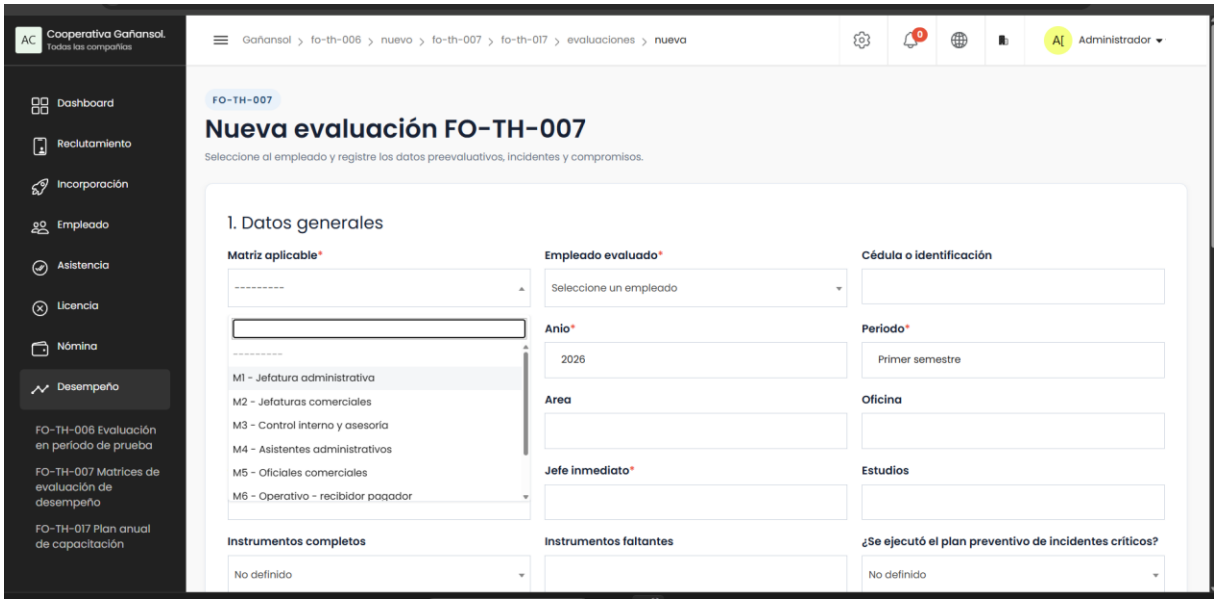
Submódulo de "Desempeño" que contiene las matrices utilizadas para evaluar el desempeño del personal.

1. Utilice el campo "Buscar" para localizar registros por empleado, cédula o cargo.
2. Aplique los filtros disponibles (estado, año) y presione "Buscar" para refinar el listado.
3. Presione "+ Nueva evaluación" para abrir el formulario de registro.
4. Complete los campos solicitados y guarde la información.
5. Use las opciones "Exportar Excel" / "Exportar Word" o "Descargar todos" para obtener respaldos del listado.
6. En la columna "Acciones" puede ver, editar, descargar o eliminar cada registro individualmente.

Permite seleccionar al empleado evaluado, registrar los criterios de la matriz, calificar el desempeño, agregar comentarios y obtener los resultados finales.



Figura 43. Listado de matrices de evaluación de desempeño.



2. Incidentes críticos

Registre hechos positivos o negativos con su fecha y descripción.

+ Agregar incidente

Tipo*	Fecha	Descripcion*	Orden*
---------	dd/mm/aaaa		1

☐ Eliminar fila

3. Planes y compromisos

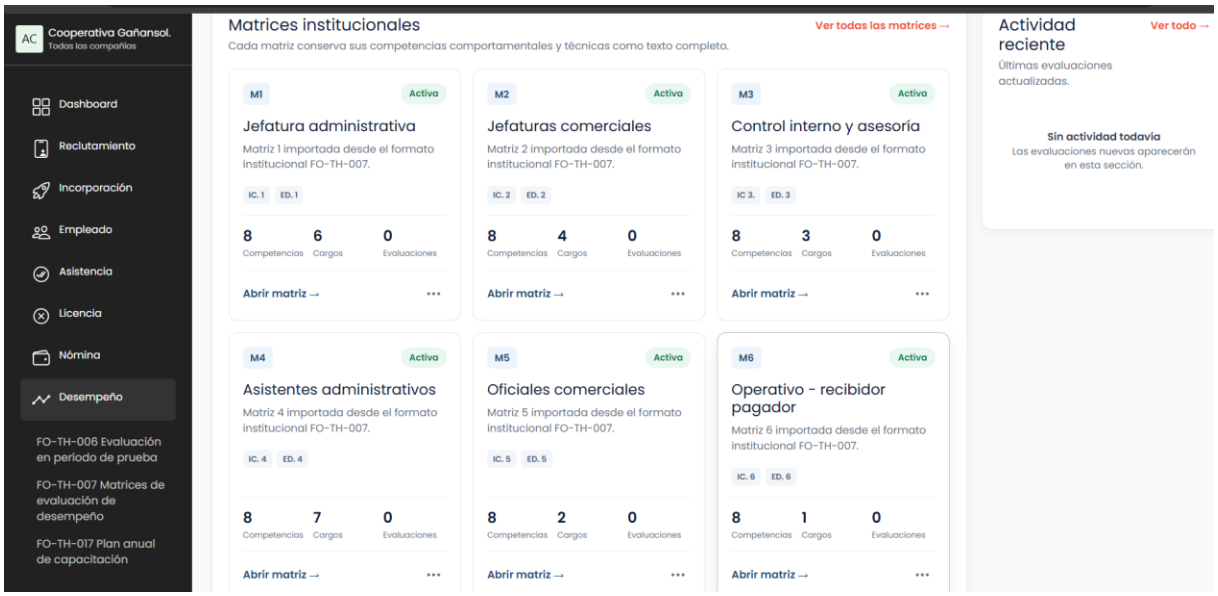
Puede registrar compromisos del empleado, jefe inmediato o Cooperativa.

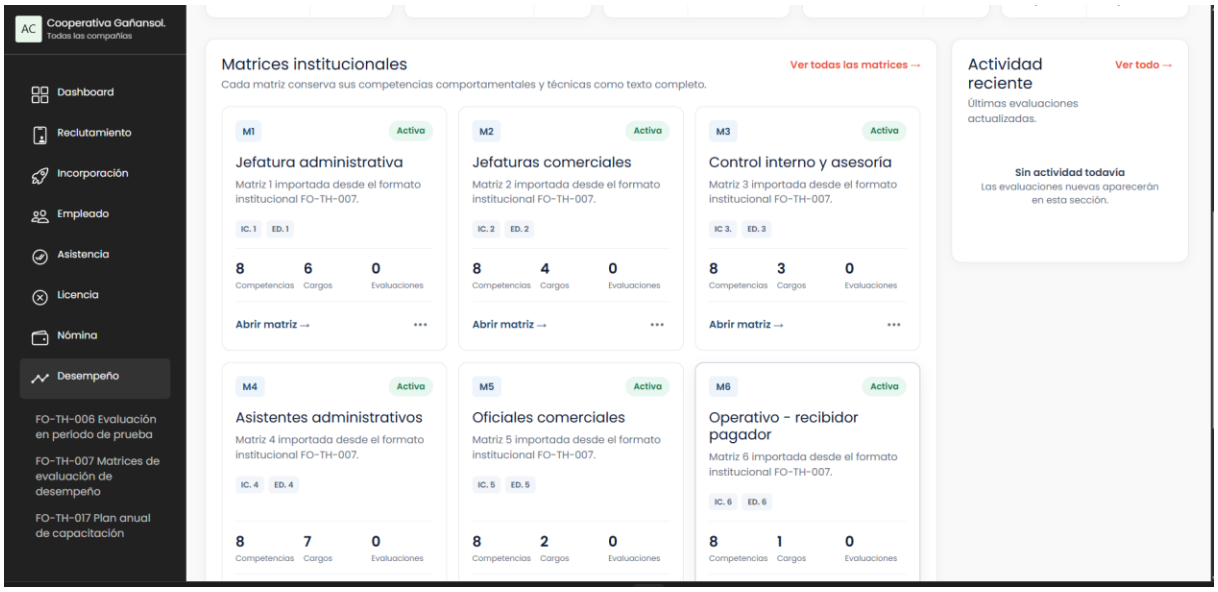
+ Agregar compromiso

Tipo*	Compromiso*	Responsable	Fecha limite
---------			dd/mr

Estado*	Orden*
Pendiente	1

☐ Eliminar fila





Figuras 44–47. Formulario de la matriz de evaluación con criterios, calificaciones y comentarios.

11. FO-TH-008 Control de vacaciones

Módulo del menú "Licencia" que administra, por empleado, los períodos, días asignados, vacaciones tomadas y saldos disponibles.

1. Utilice el campo "Buscar" para localizar registros por empleado, cédula, cargo o correo.
2. Aplique los filtros disponibles (estado, saldo) y presione "Buscar" para refinar el listado.
3. Presione "+ Nuevo control" para abrir el formulario de registro.
4. Complete los campos solicitados y guarde la información.
5. Use las opciones "Exportar Excel" / "Exportar Word" o "Descargar todos" para obtener respaldos del listado.
6. En la columna "Acciones" puede ver, editar, descargar o eliminar cada registro individualmente.

La pantalla resume "Controles" (empleados registrados), "Períodos" administrados y "Saldo disponible" en días consolidados. El listado detalla Empleado, Cédula, Cargo, Ingreso, Períodos, Asignados, Tomados, Disponibles y Estado.

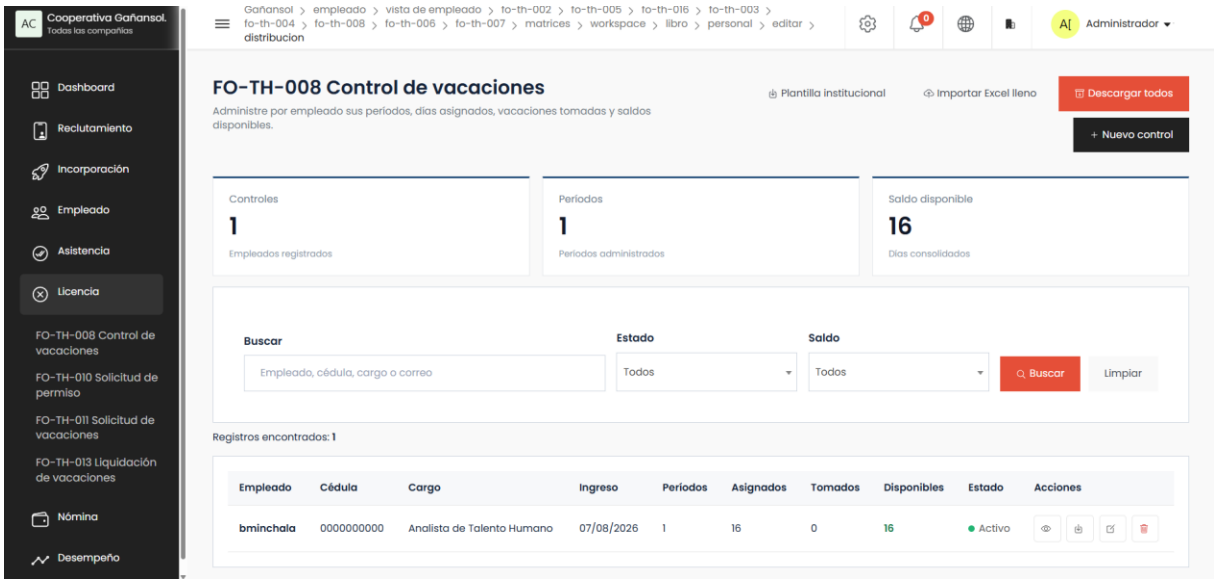


Figura 48. Panel FO-TH-008 Control de vacaciones, con indicadores de controles, períodos y saldo disponible.

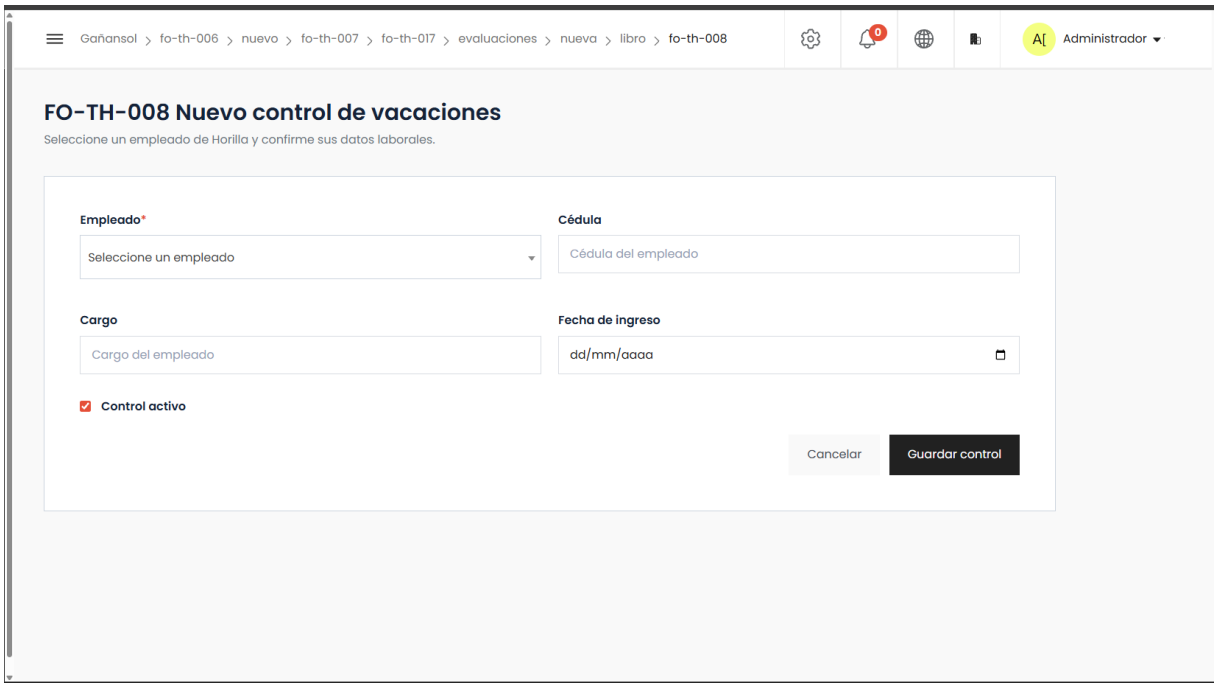


Figura 49. Detalle y edición de un control de vacaciones.

12. FO-TH-009 Reporte de novedades de Talento Humano

Módulo del menú "Nómina" que permite registrar y consultar novedades del personal reportadas por Talento Humano.

1. Utilice el campo "Buscar" para localizar registros por empleado, cédula o cargo.
2. Aplique los filtros disponibles (estado, tipo de novedad) y presione "Buscar" para refinar el listado.
3. Presione "+ Nuevo reporte" para abrir el formulario de registro.
4. Complete los campos solicitados y guarde la información.
5. Use las opciones "Exportar Excel" / "Exportar Word" o "Descargar todos" para obtener respaldos del listado.
6. En la columna "Acciones" puede ver, editar, descargar o eliminar cada registro individualmente.

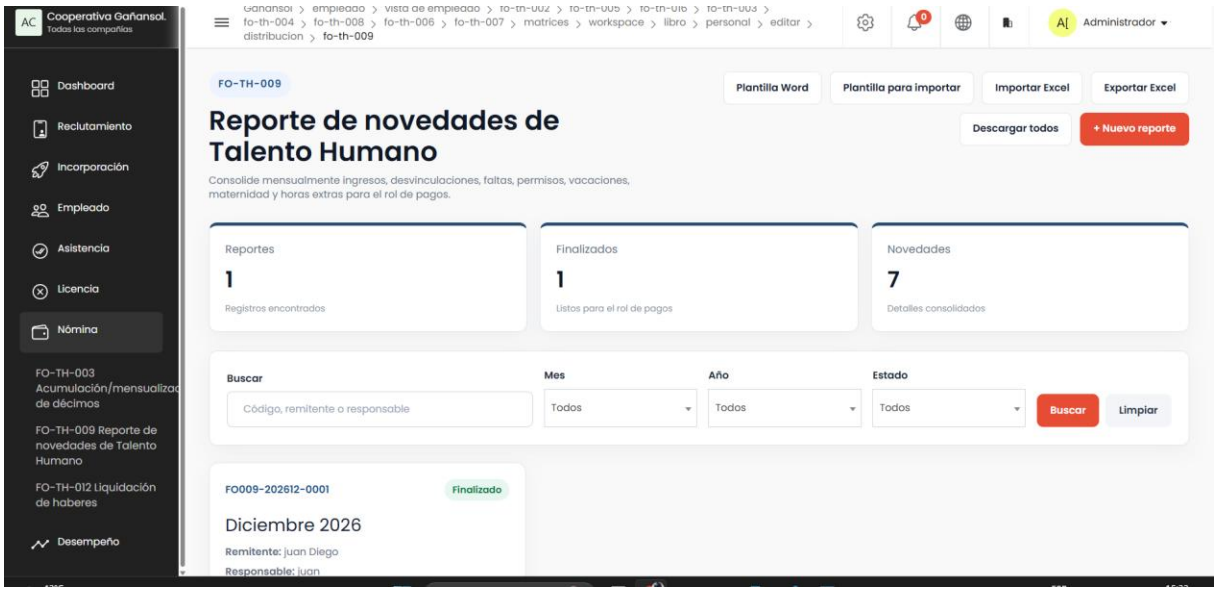
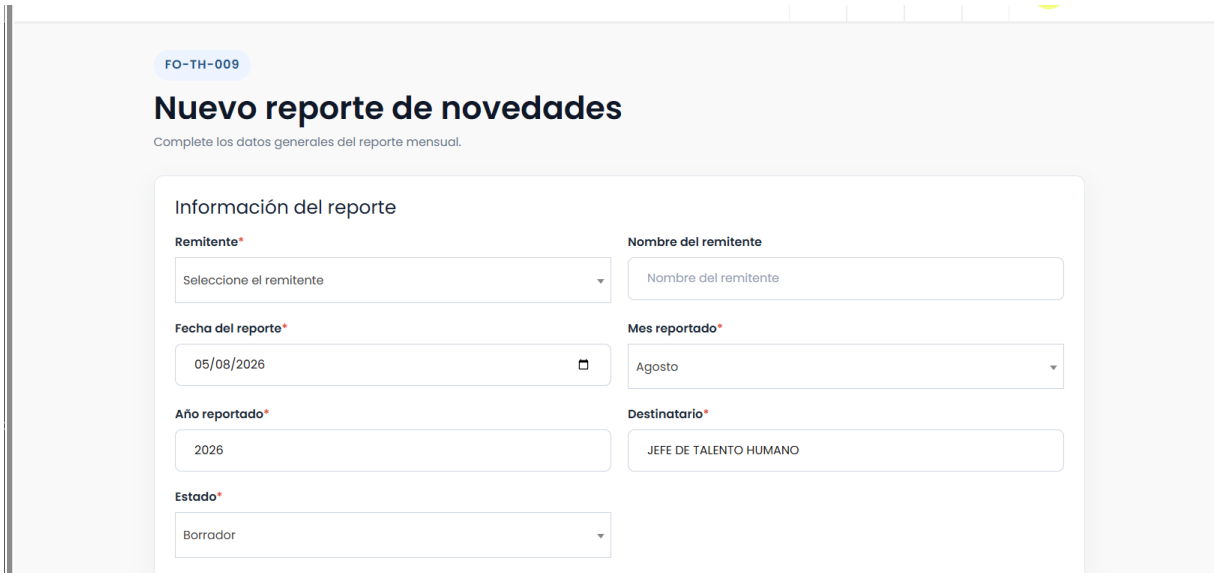
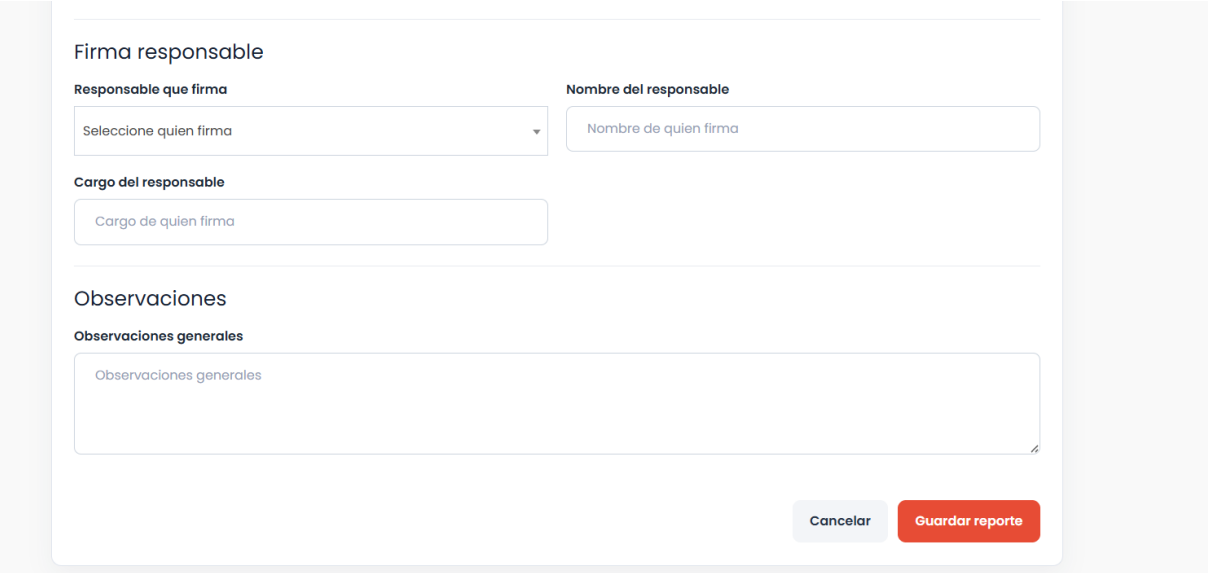


Figura 50. Listado FO-TH-009 Reporte de novedades de Talento Humano.





Figuras 51–52. Formulario de registro y detalle de una novedad reportada.

13. FO-TH-010 Solicitud de permiso

Submódulo de "Licencia" donde los usuarios registran solicitudes de permiso y los responsables revisan su estado.

1. Utilice el campo "Buscar" para localizar registros por empleado, cédula o cargo.
2. Aplique los filtros disponibles (estado) y presione "Buscar" para refinar el listado.
3. Presione "+ Nueva solicitud" para abrir el formulario de registro.
4. Complete los campos solicitados y guarde la información.
5. Use las opciones "Exportar Excel" / "Exportar Word" o "Descargar todos" para obtener respaldos del listado.
6. En la columna "Acciones" puede ver, editar, descargar o eliminar cada registro individualmente.

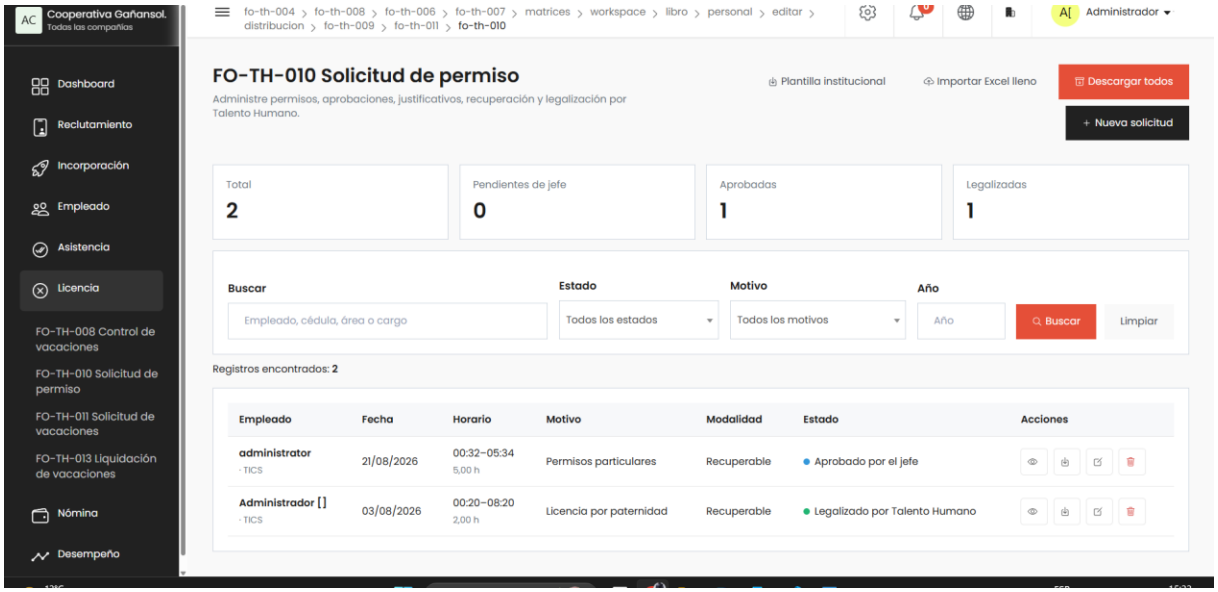


Figura 53. Listado FO-TH-010 Solicitud de permiso.

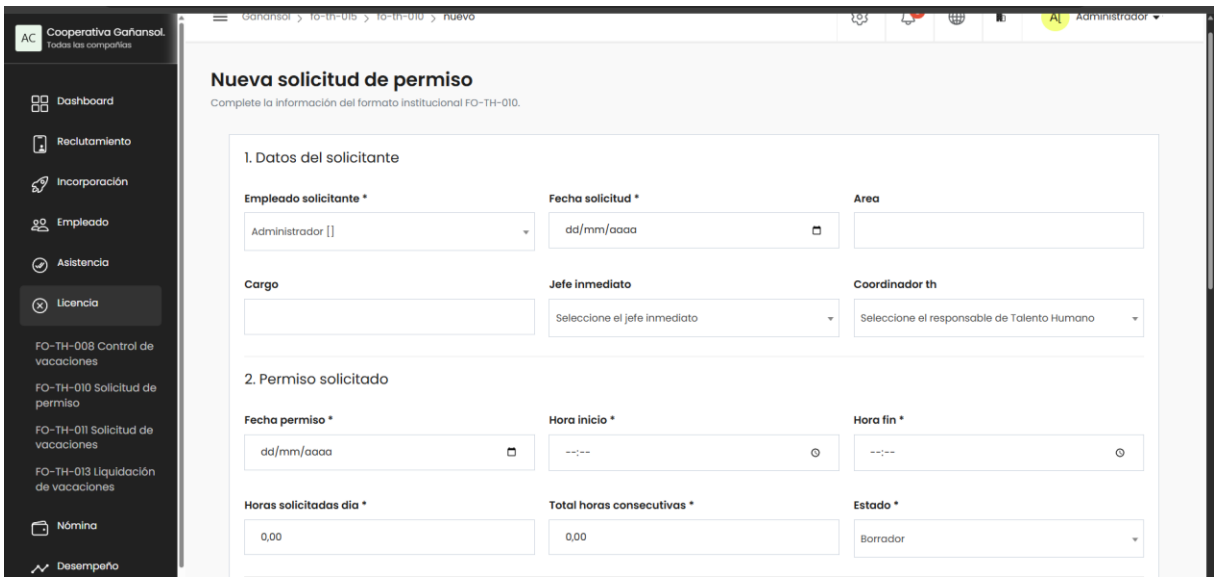


Figura 54. Formulario para registrar una nueva solicitud de permiso.

14. FO-TH-011 Solicitud de vacaciones

Submódulo de "Licencia" que permite registrar solicitudes de vacaciones y dar seguimiento a su legalización y recuperación.

1. Utilice el campo "Buscar" para localizar registros por empleado, cédula o cargo.
2. Aplique los filtros disponibles (estado) y presione "Buscar" para refinar el listado.
3. Presione "+ Nueva solicitud" para abrir el formulario de registro.
4. Complete los campos solicitados y guarde la información.
5. Use las opciones "Exportar Excel" / "Exportar Word" o "Descargar todos" para obtener respaldos del listado.
6. En la columna "Acciones" puede ver, editar, descargar o eliminar cada registro individualmente.

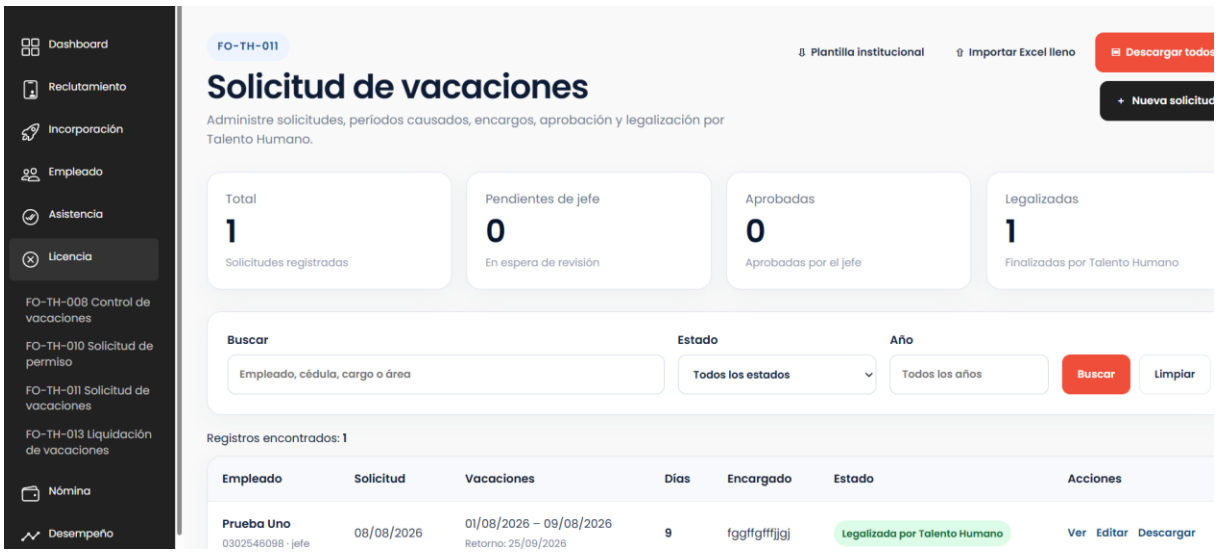
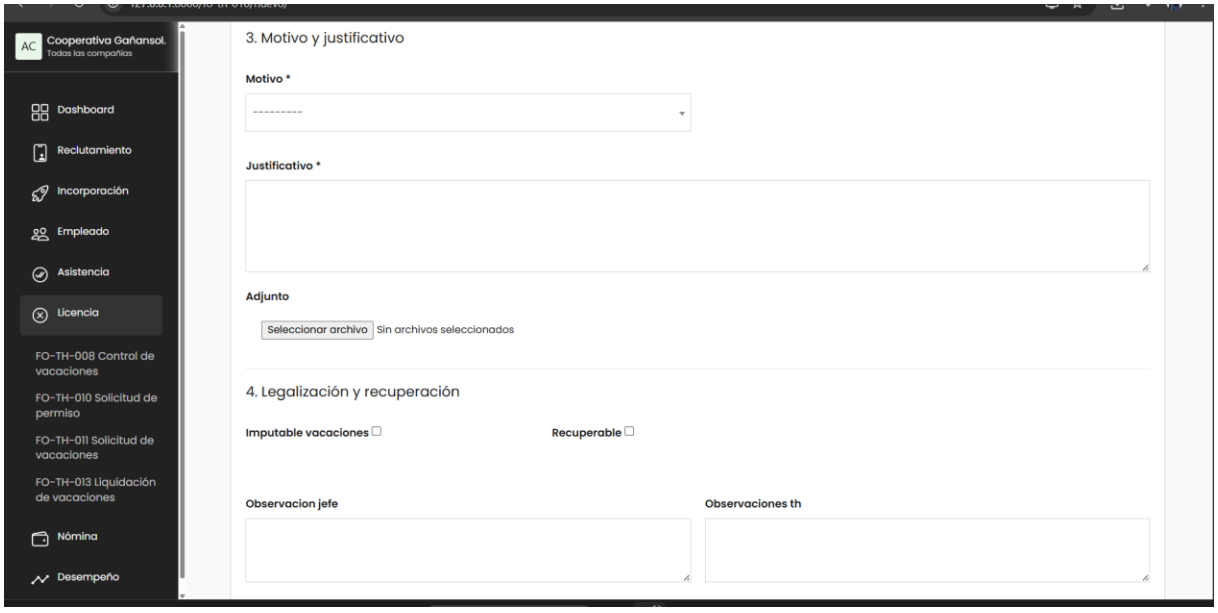


Figura 55. Listado FO-TH-011 Solicitud de vacaciones.





Figuras 56–57. Formulario de solicitud, con fechas y estado de aprobación.

15. FO-TH-012 Liquidación de haberes

Módulo del menú "Nómina" que administra liquidaciones con décimos acumulados o mensualizados, vacaciones, desahucio e indemnización.

1. Utilice el campo "Buscar" para localizar registros por empleado, cédula, cargo o clave.
2. Aplique los filtros disponibles (modalidad, estado, año) y presione "Buscar" para refinar el listado.
3. Presione "+ Nueva liquidación" para abrir el formulario de registro.
4. Complete los campos solicitados y guarde la información.
5. Use las opciones "Exportar Excel" / "Exportar Word" o "Descargar todos" para obtener respaldos del listado.
6. En la columna "Acciones" puede ver, editar, descargar o eliminar cada registro individualmente.

El panel resume "Registros", "Borradores", "Aprobadas" y "Total calculado". El listado muestra Empleado, Modalidad, Salida, Estado (ej. "Pagada") y Total, con acciones para abrir, editar, descargar, anular o eliminar cada liquidación.

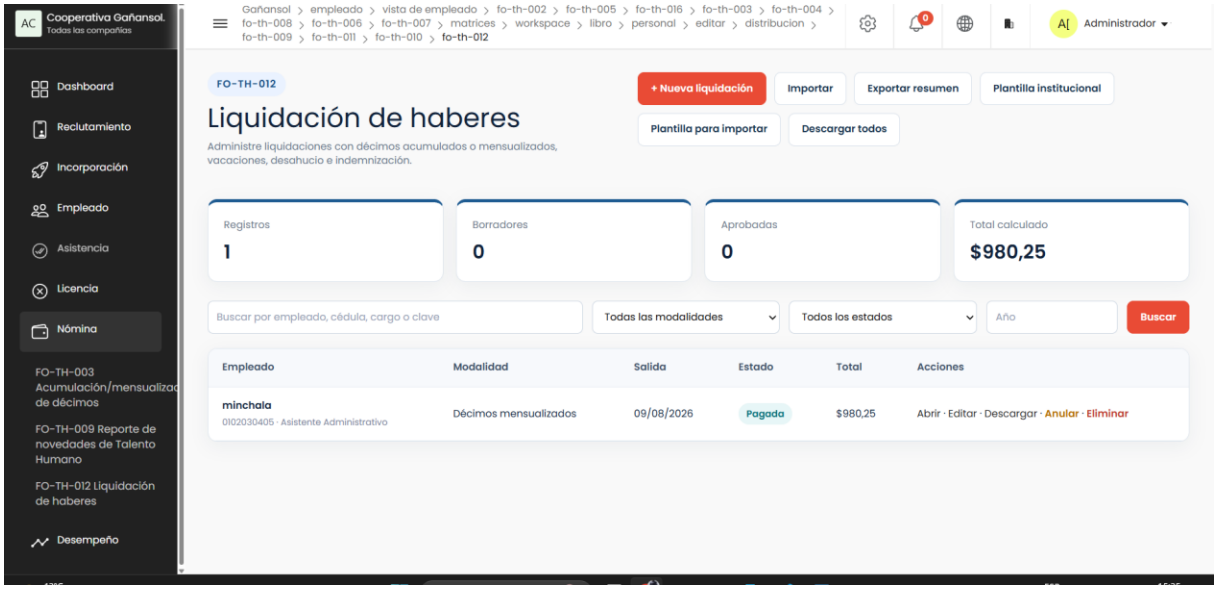


Figura 58. Panel FO-TH-012 Liquidación de haberes.



Figura 59. Formulario de una nueva liquidación de haberes.

16. FO-TH-013 Liquidación de vacaciones

Submódulo de "Licencia" que gestiona la liquidación de vacaciones, incluyendo datos laborales, valores calculados, responsables y observaciones.

1. Utilice el campo "Buscar" para localizar registros por empleado, cédula o cargo.
2. Aplique los filtros disponibles (estado, año) y presione "Buscar" para refinar el listado.
3. Presione "+ Nueva liquidación" para abrir el formulario de registro.
4. Complete los campos solicitados y guarde la información.
5. Use las opciones "Exportar Excel" / "Exportar Word" o "Descargar todos" para obtener respaldos del listado.
6. En la columna "Acciones" puede ver, editar, descargar o eliminar cada registro individualmente.

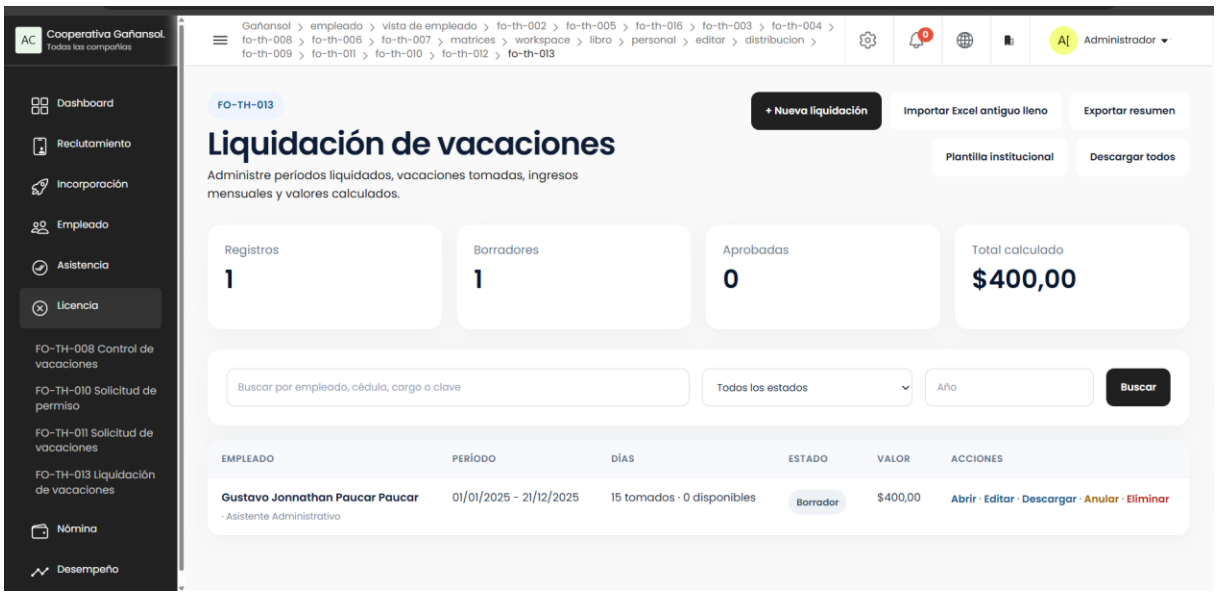
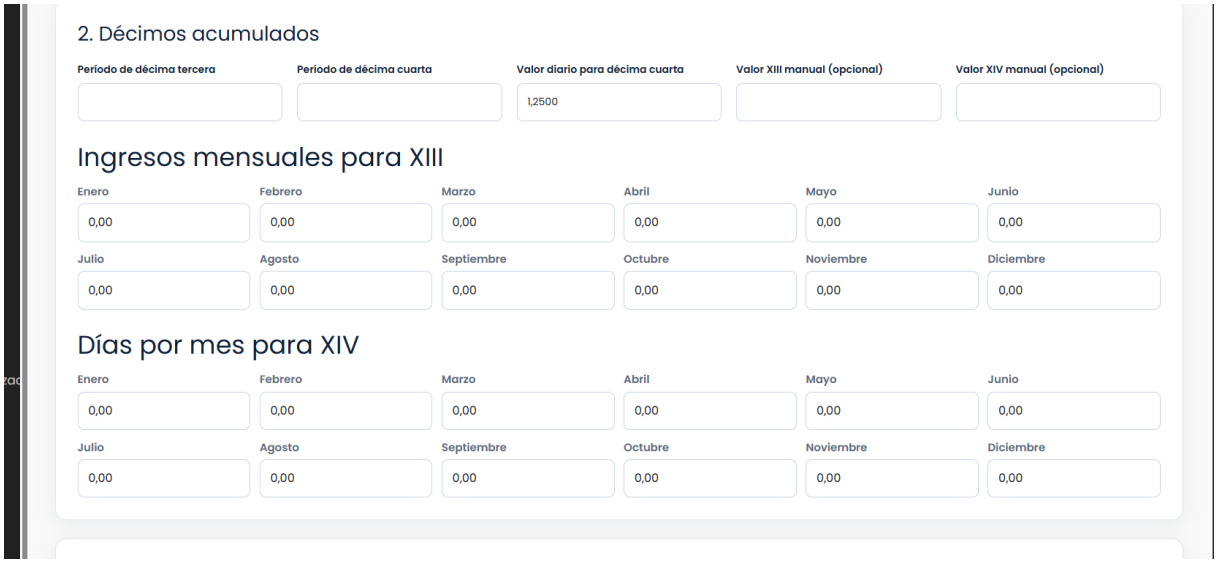
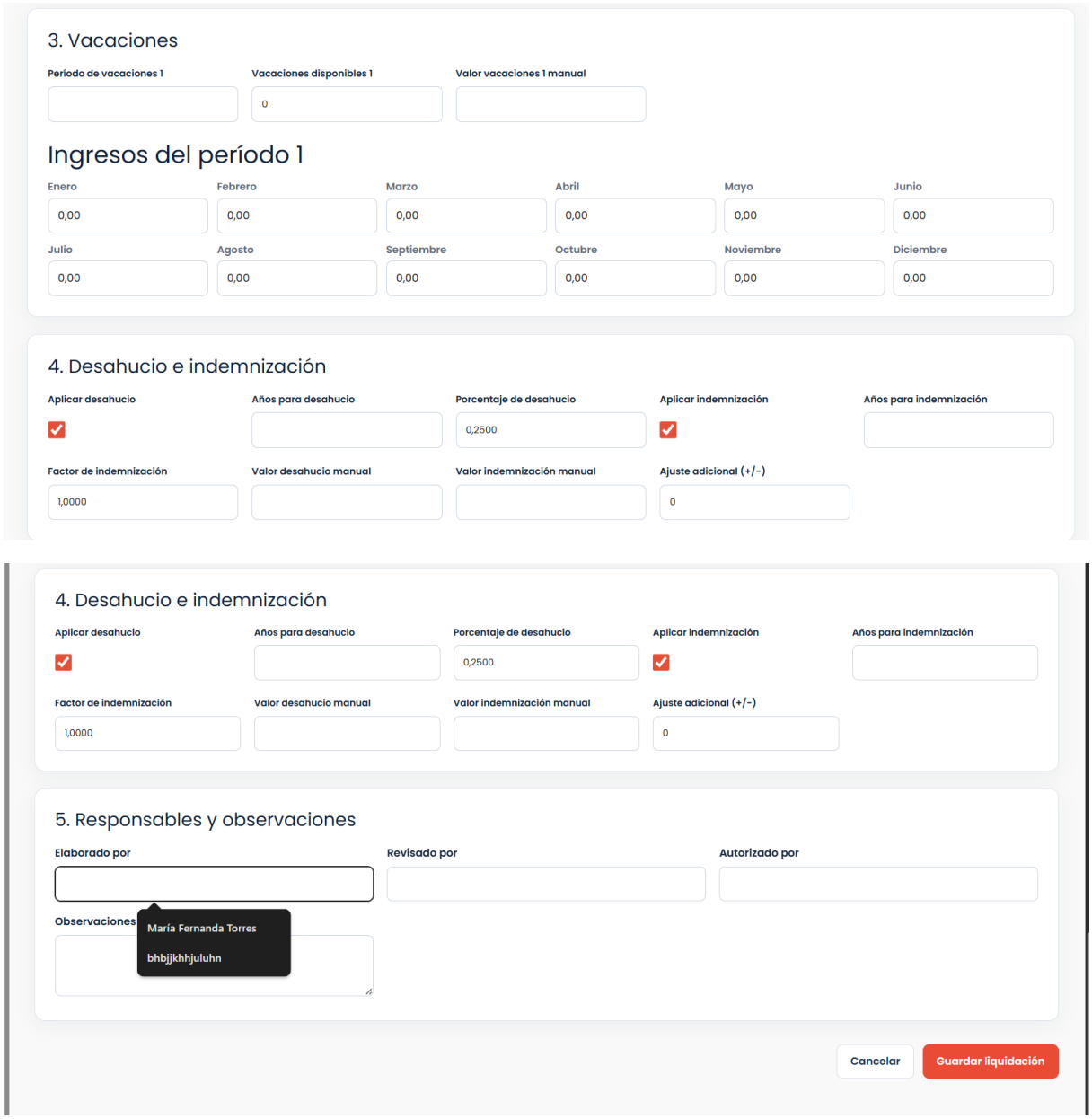


Figura 60. Listado FO-TH-013 Liquidación de vacaciones.





Figuras 61–63. Formulario de liquidación de vacaciones con valores calculados y observaciones.

17. FO-TH-014 Control de pasantías

Submódulo de "Incorporación" que administra colaboradores en formación, instituciones formadoras, jornadas, tutores empresariales y actividades de pasantía.

1. Utilice el campo "Buscar" para localizar registros por código, colaborador, cédula, institución, área o tutor.
2. Aplique los filtros disponibles (estado, año) y presione "Buscar" para refinar el listado.
3. Presione "+ Nuevo control" para abrir el formulario de registro.
4. Complete los campos solicitados y guarde la información.
5. Use las opciones "Exportar Excel" / "Exportar Word" o "Descargar todos" para obtener respaldos del listado.
6. En la columna "Acciones" puede ver, editar, descargar o eliminar cada registro individualmente.

El panel resume "Controles", "Finalizados", "Actividades" y "Horas registradas". El listado detalla Colaborador, Institución/Área, Período, Tutor, Actividades y Estado (ej. "Aprobado").

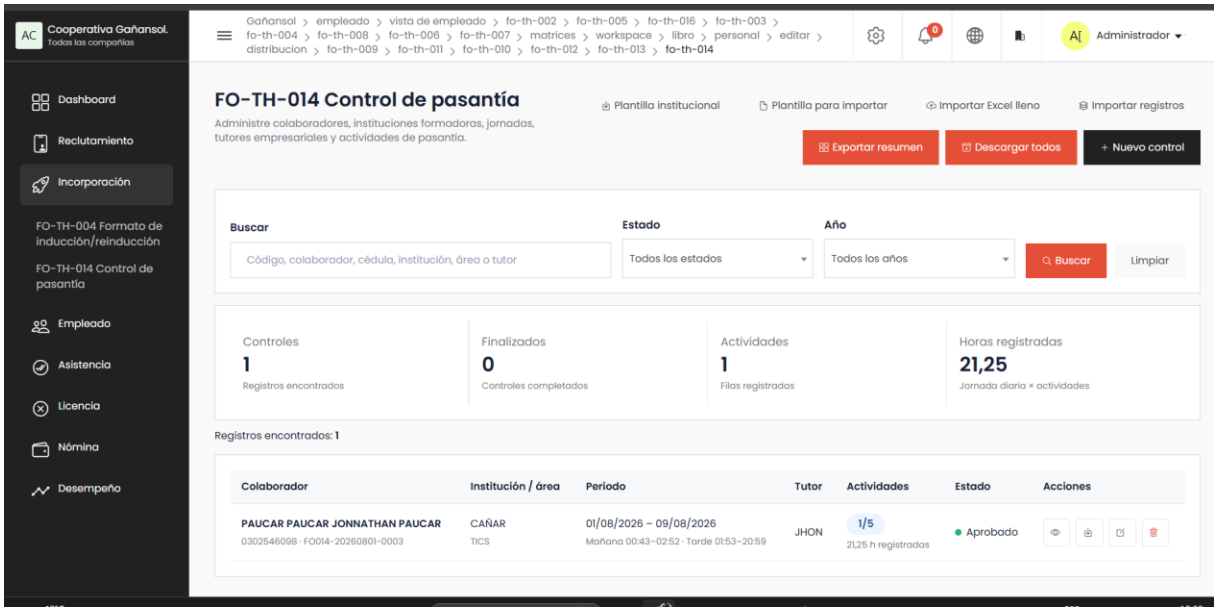
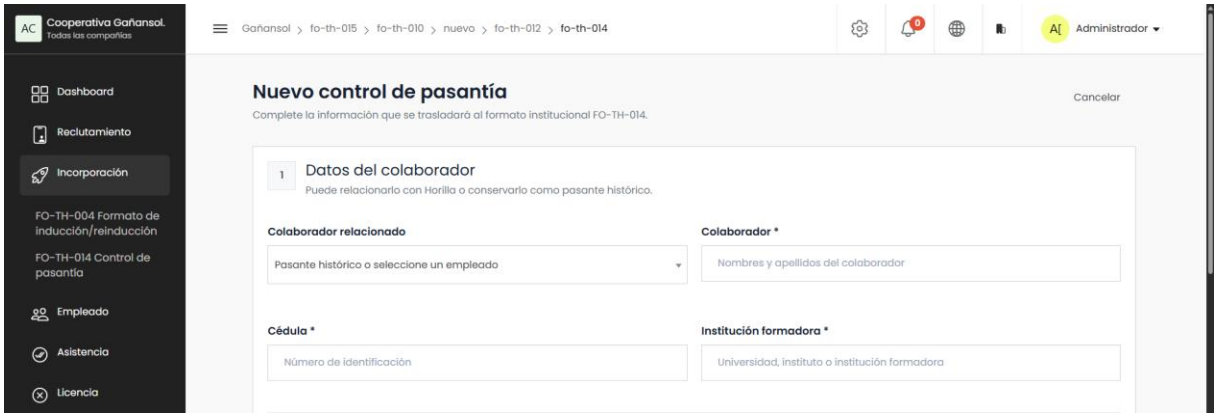
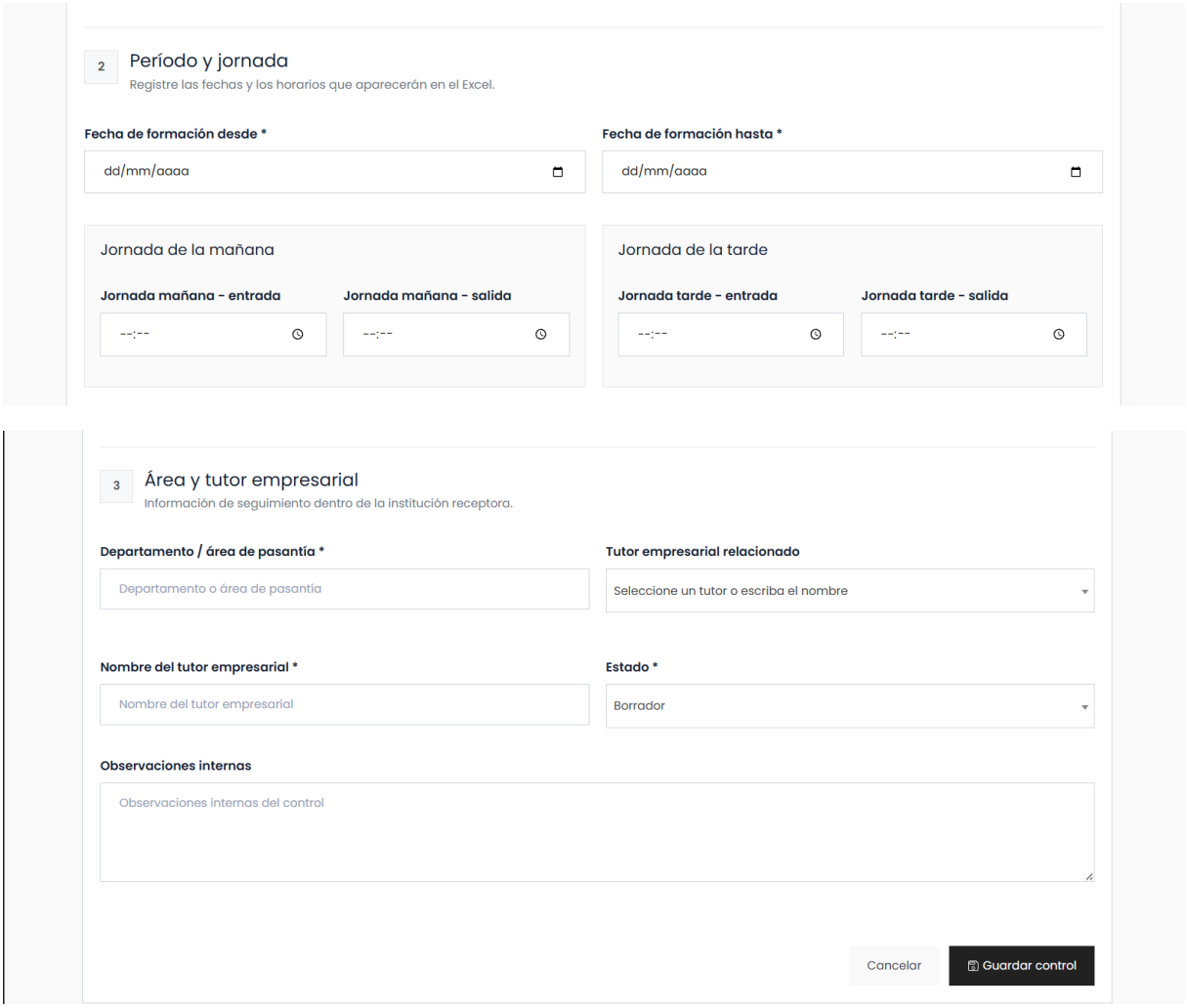


Figura 64. Panel FO-TH-014 Control de pasantías.





Figuras 65–67. Registro de jornadas y actividades de la pasantía.

18. FO-TH-015 Ficha de entrevista

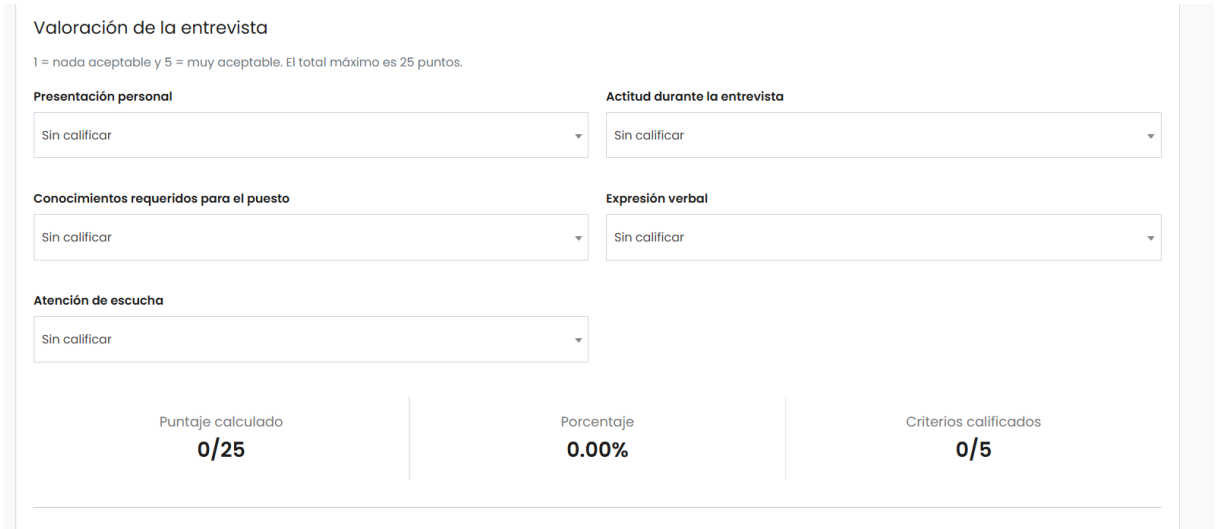
Módulo del menú "Reclutamiento" para registrar, consultar e importar entrevistas realizadas en el formato institucional.

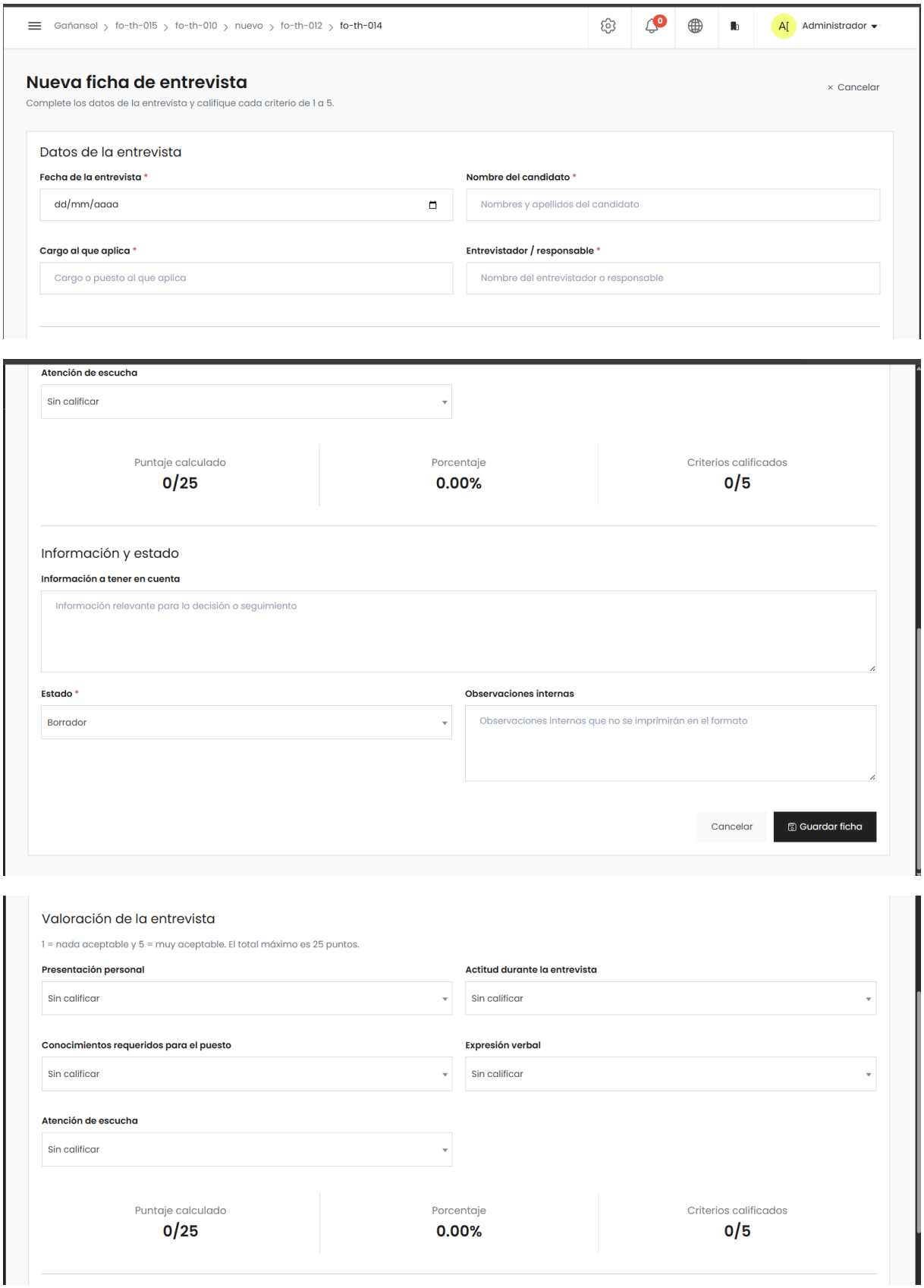
1. Utilice el campo "Buscar" para localizar registros por código, candidato, cargo o entrevistador.
2. Aplique los filtros disponibles (estado, año) y presione "Buscar" para refinar el listado.
3. Presione "+ Nueva ficha" para abrir el formulario de registro.
4. Complete los campos solicitados y guarde la información.
5. Use las opciones "Exportar Excel" / "Exportar Word" o "Descargar todos" para obtener respaldos del listado.
6. En la columna "Acciones" puede ver, editar, descargar o eliminar cada registro individualmente.

El listado muestra Código, Candidato, Fecha, Cargo, Responsable, Resultado (puntaje y porcentaje) y Estado (ej. "Completada").



Figura 68. Listado FO-TH-015 Ficha de entrevista.





Figuras 69–72. Formulario de la ficha de entrevista, con la valoración del candidato.

19. FO-TH-016 Informe de actividades de modalidad virtual

Submódulo de "Asistencia" que registra los períodos, actividades, evidencias y datos de entrega del trabajo en modalidad virtual.

1. Utilice el campo "Buscar" para localizar registros por código, empleado, cédula, cargo, dirección o motivo.
2. Aplique los filtros disponibles (estado, año) y presione "Buscar" para refinar el listado.
3. Presione "+ Nuevo informe" para abrir el formulario de registro.
4. Complete los campos solicitados y guarde la información.
5. Use las opciones "Exportar Excel" / "Exportar Word" o "Descargar todos" para obtener respaldos del listado.
6. En la columna "Acciones" puede ver, editar, descargar o eliminar cada registro individualmente.

El listado muestra Empleado, Cargo/dirección, Período, Motivo, Actividades (evidencias completas) y Estado (ej. "Borrador").

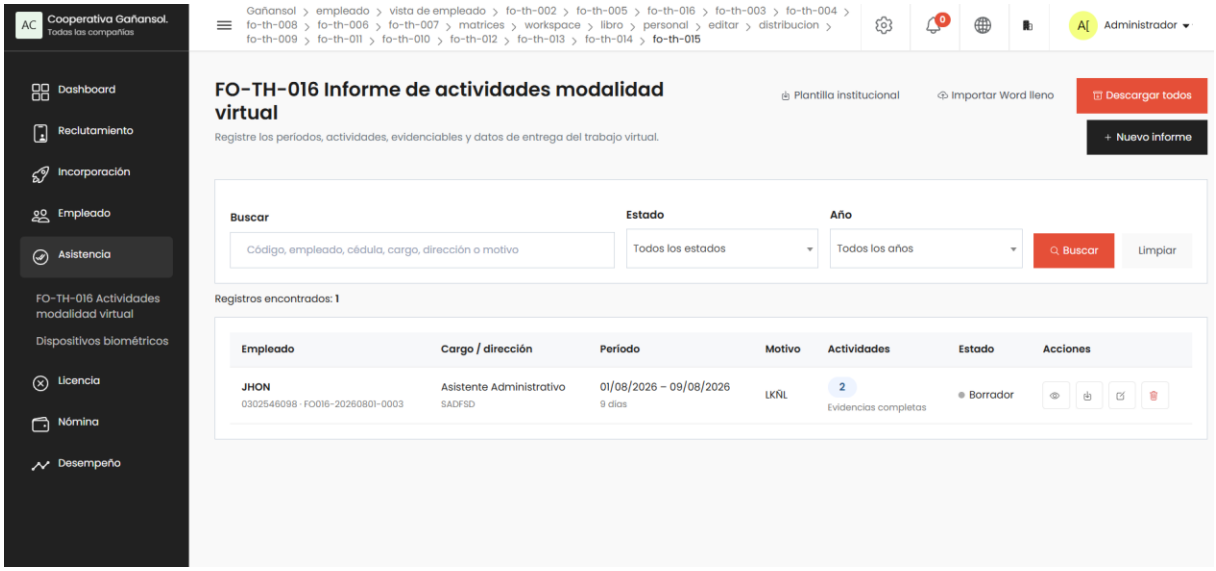
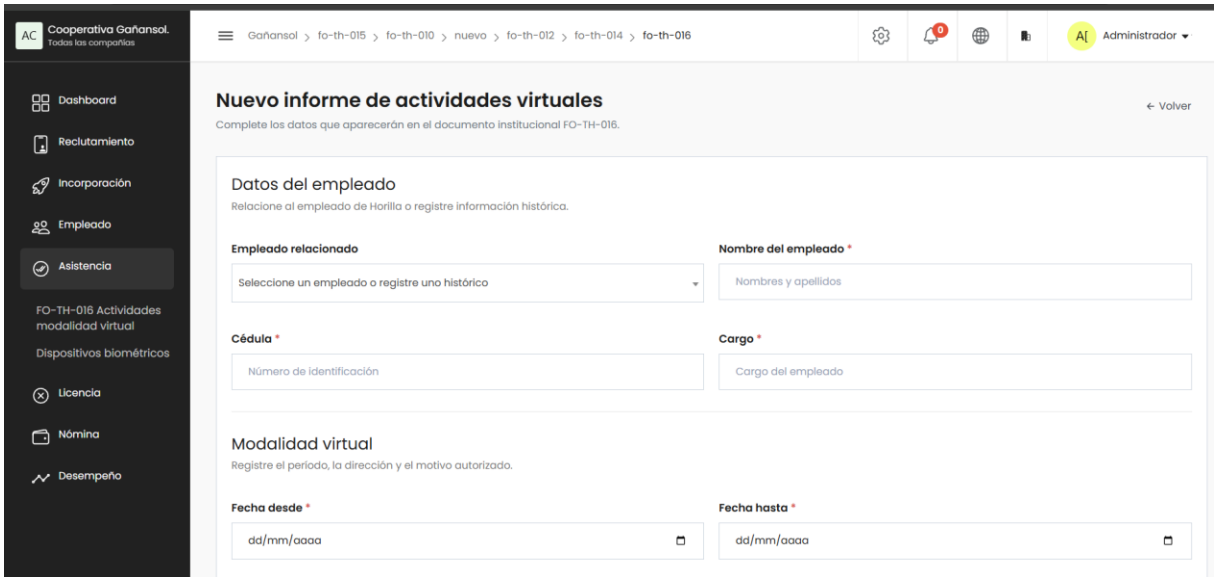
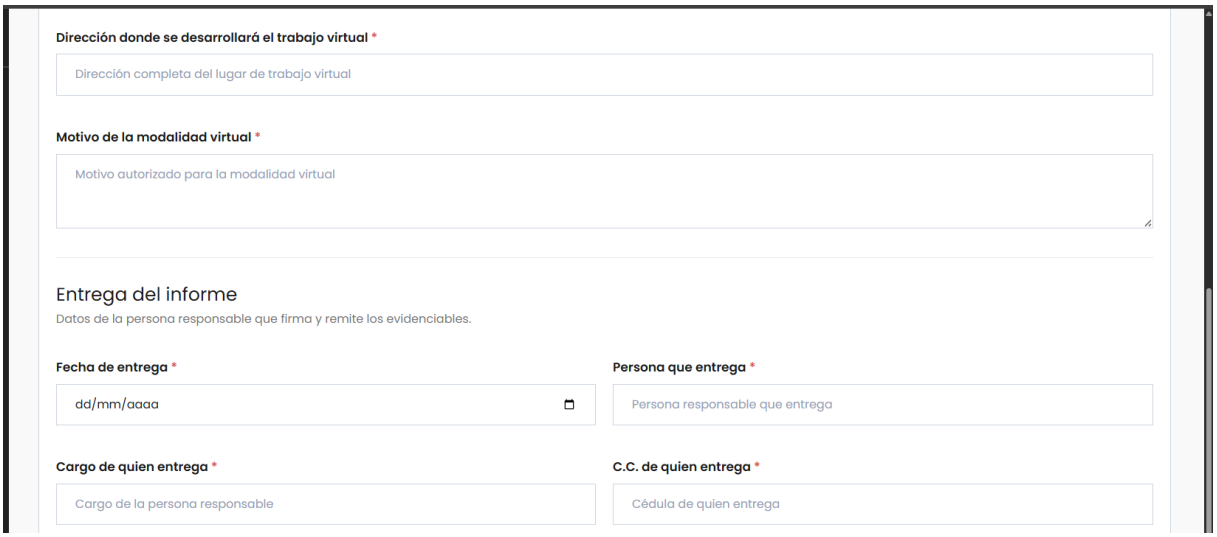
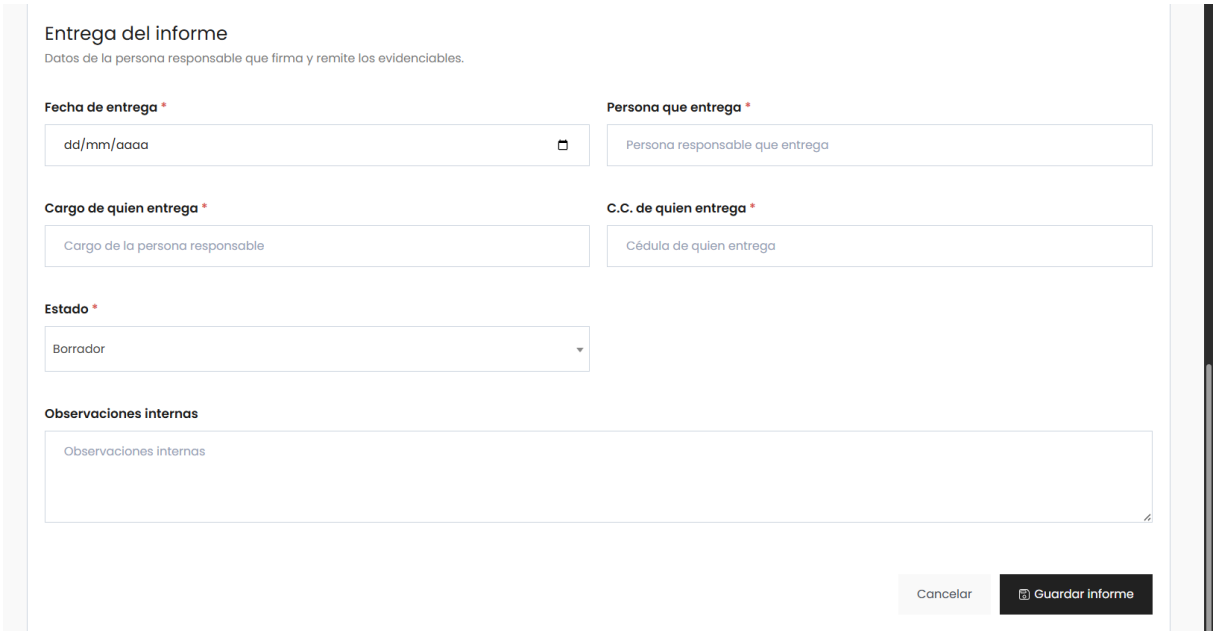
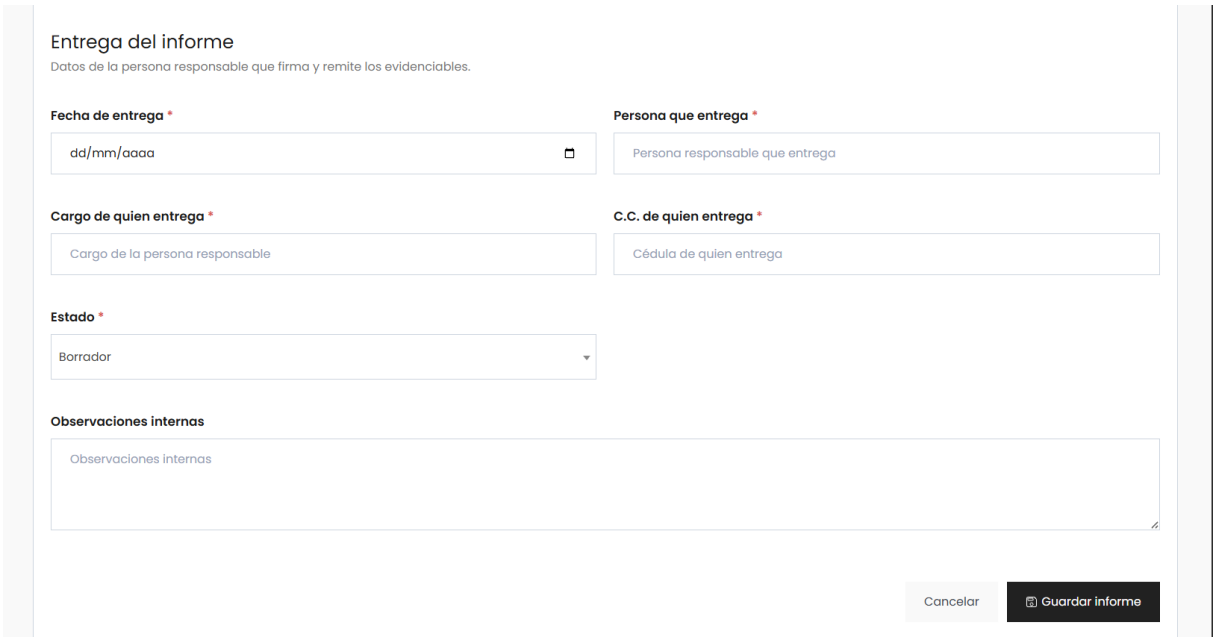


Figura 73. Listado FO-TH-016 Informe de actividades modalidad virtual.




Dirección donde se desarrollará el trabajo virtual *
Dirección completa del lugar de trabajo virtual
Motivo de la modalidad virtual *
Motivo autorizado para la modalidad virtual
Entrega del informe
Datos de la persona responsable que firma y remite los evidenciables.
Fecha de entrega *
dd/mm/aaaa
Persona que entrega *
Persona responsable que entrega
Cargo de quien entrega *
Cargo de la persona responsable
C.C. de quien entrega *
Cédula de quien entrega


Entrega del informe
Datos de la persona responsable que firma y remite los evidenciables.
Fecha de entrega *
dd/mm/aaaa
Persona que entrega *
Persona responsable que entrega
Cargo de quien entrega *
Cargo de la persona responsable
C.C. de quien entrega *
Cédula de quien entrega
Estado *
Borrador
Observaciones internas
Observaciones internas
Cancelar
Guardar informe



Figuras 74–77. Formulario del informe, con las evidencias del período virtual.

20. FO-TH-017 Plan anual de capacitación

Submódulo de "Desempeño" que permite administrar los planes institucionales de capacitación.

1. Utilice el campo "Buscar" para localizar registros por responsable o año del plan.
2. Aplique los filtros disponibles (estado, año) y presione "Buscar" para refinar el listado.
3. Presione "+ Nuevo plan" para abrir el formulario de registro.
4. Complete los campos solicitados y guarde la información.
5. Use las opciones "Exportar Excel" / "Exportar Word" o "Descargar todos" para obtener respaldos del listado.
6. En la columna "Acciones" puede ver, editar, descargar o eliminar cada registro individualmente.

El plan incluye: año del plan, responsable, documentación de respaldo, capacitaciones programadas, control de cambios y estado del plan.

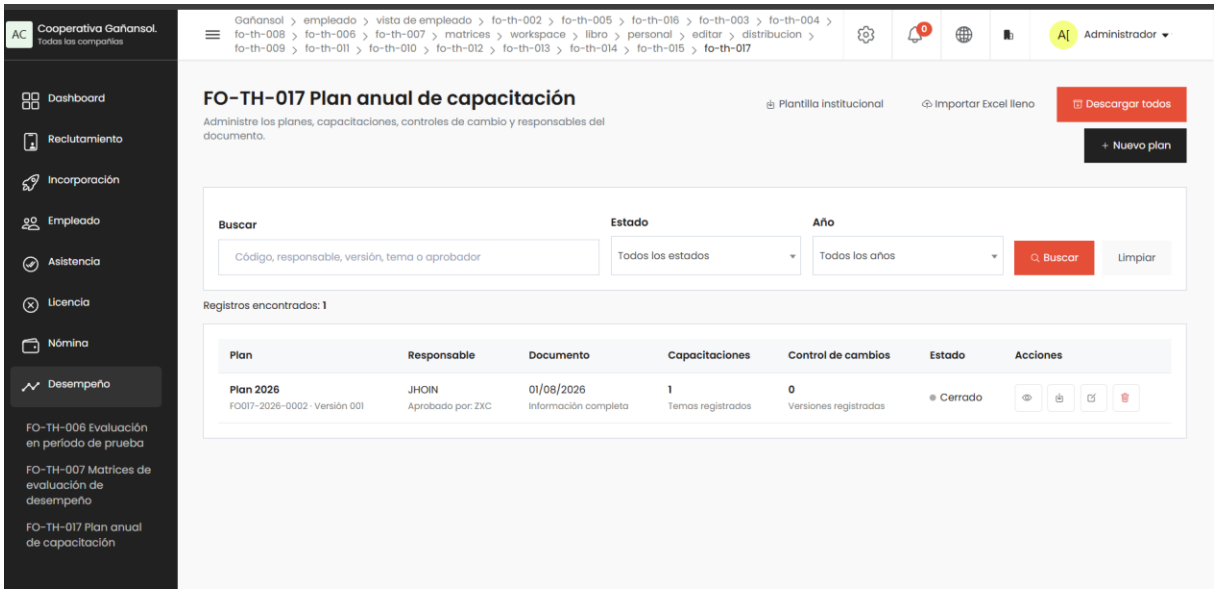
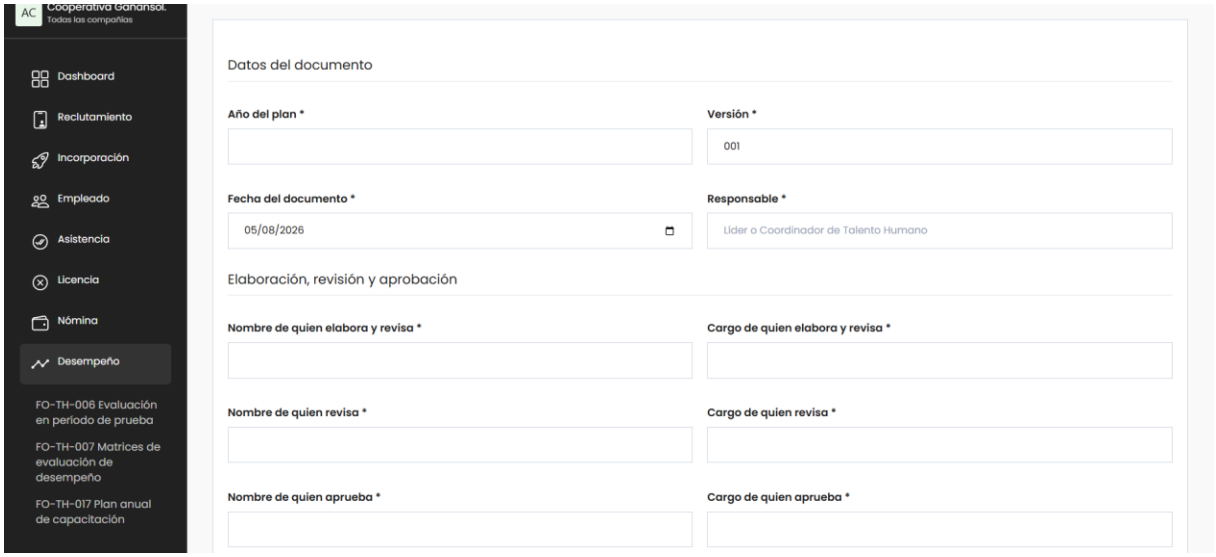
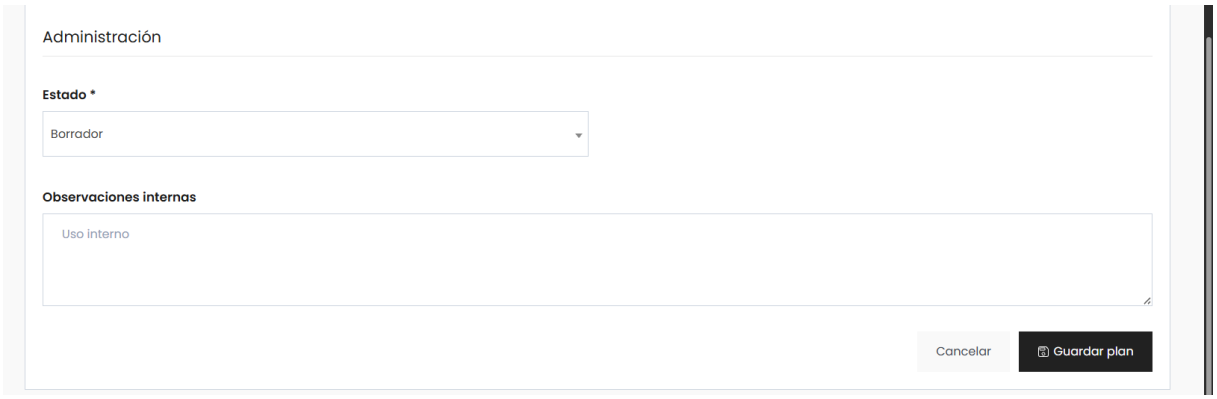


Figura 78. Listado FO-TH-017 Plan anual de capacitación.





Figuras 79–80. Detalle del plan de capacitación y sus capacitaciones programadas.

21. Manuales y documentos institucionales

El módulo "Manuales", ubicado dentro del apartado Empleado, permite almacenar y compartir documentos institucionales de Talento Humano (manuales, políticas, planes, etc.), sin necesidad de asignar categorías, códigos o versiones.

6. Presionar "+ Subir manual" (o "Subir varios") para cargar uno o más documentos.
7. Usar el buscador "Buscar por nombre" para localizar un documento específico.
8. Desde la columna "Acciones" puede "Descargar", "Renombrar / reemplazar" o "Eliminar" cada documento.
9. Usar "Descargar todos" para obtener un respaldo completo de los documentos disponibles.

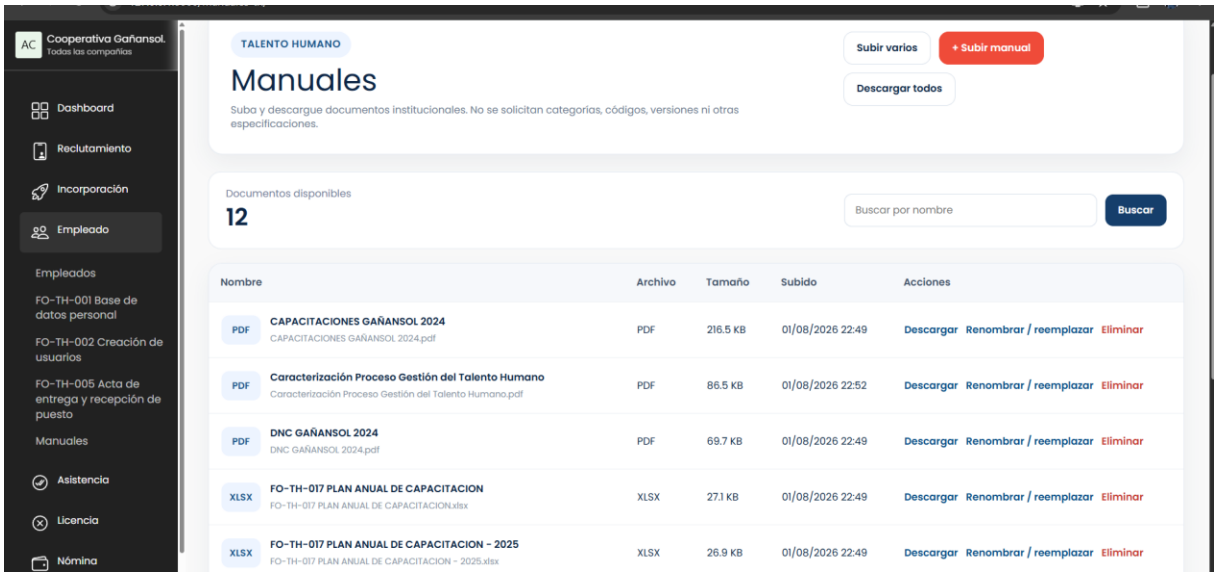


Figura 81. Módulo de Manuales, con el listado de documentos institucionales disponibles para descarga.

ANEXO F. MANUAL DEL PROGRAMADOR

Arquitectura, permisos, instalación y guía de desarrollo de módulos

1. Introducción

Este manual describe, con nivel técnico, la arquitectura, la configuración y el mantenimiento de la personalización realizada sobre el sistema Horilla para la Cooperativa Gañansol. La integración incorpora un conjunto de módulos propios de Talento Humano (identificados con el prefijo fo_th_*), manteniendo intacta la arquitectura original del framework y añadiendo compatibilidad total con el sistema de permisos, configuración y accesos ya existente en Horilla.

El objetivo de este documento es servir como guía de referencia para cualquier desarrollador —actual o futuro— encargado del mantenimiento, la ampliación o la actualización del sistema. Se explica no solo "qué" se hizo, sino "cómo" y "por qué", de modo que una persona sin contexto previo pueda incorporarse al proyecto con autonomía.

¿Qué es Horilla?

Horilla es un sistema de gestión de recursos humanos (HRMS) de código abierto, construido sobre el framework Django (Python). Provee de fábrica módulos como Reclutamiento, Incorporación (Onboarding), Empleados, Asistencia, Licencias, Nómina (Payroll) y Desempeño. Sobre esa base, este proyecto añade 17 formatos institucionales propios de la cooperativa (FO-TH-001 a FO-TH-017) sin alterar el núcleo del framework.

Alcance de este manual

- Arquitectura general del proyecto y organización de carpetas.
- Detalle de cada módulo fo_th_* y su responsabilidad.
- Cambios realizados sobre el Horilla original.
- Sistema de permisos de Django aplicado a los nuevos módulos.
- Instalación y puesta en marcha del proyecto, paso a paso.
- Estructura interna que debe seguir cada módulo nuevo.
- Buenas prácticas, mantenimiento, ampliaciones futuras y resolución de problemas.

2. Arquitectura general del sistema

El sistema está construido sobre la siguiente pila tecnológica:

Componente	Tecnología
Backend / framework	Django (Python)
Lenguaje principal	Python 3.x
Base de datos	Compatible con los motores soportados por Django (PostgreSQL recomendado; también MySQL/MariaDB o SQLite en desarrollo)
Frontend	Plantillas de Django (Django Templates), HTML5, CSS3 y JavaScript
Autenticación y autorización	Sistema nativo de usuarios, grupos y permisos de Django (django.contrib.auth)
Servidor de aplicaciones (producción)	WSGI/ASGI, típicamente Gunicorn o uWSGI detrás de Nginx

2.1 Organización por aplicaciones

Django organiza el código en "aplicaciones" (apps): paquetes de Python independientes, cada uno con su propio modelo de datos, vistas, rutas y plantillas, que luego se registran en el proyecto principal. Esta arquitectura se respeta y se extiende de la siguiente forma:

- Aplicaciones base de Horilla: módulos originales del framework (empleados, reclutamiento, asistencia, licencias, nómina, etc.). No se modifican.
- Aplicaciones propias de Talento Humano: los 17 módulos fo_th_* desarrollados para la cooperativa, cada uno como una app Django independiente.
- Configuración general: settings.py, urls.py raíz y variables de entorno del proyecto.
- Gestión de permisos: aprovecha el sistema de permisos de Django, extendido con los permisos propios de cada módulo fo_th_*.

2.2 Estructura de carpetas de referencia

A modo de referencia, la estructura de alto nivel del proyecto sigue este esquema:

```
horilla/
├── horilla/                  # Paquete de configuración del proyecto
│   ├── settings.py           # INSTALLED_APPS, DB, middlewares, estáticos
│   ├── urls.py               # Enrutador raíz (incluye los de cada app)
│   └── wsgi.py / asgi.py
├── employee/, recruitment/, ...   # Apps originales de Horilla (sin tocar)
├── fo_th_001_personal/        # Módulo propio: Base de datos personal
├── fo_th_002_usuarios/        # Módulo propio: Creación de usuarios
├── fo_th_003_decimos/         # ... y así sucesivamente hasta
├── ...
├── fo_th_017_capacitacion/
├── manage.py
└── requirements.txt
```

Nota: Cada carpeta fo_th_XXX_nombre es una app Django autocontenida: tiene sus propios models.py, views.py, urls.py, forms.py, templates/ y migrations/. Esto facilita mantener, probar o incluso desinstalar un módulo sin afectar a los demás.

3. Módulos personalizados de Talento Humano

A continuación, se detallan los 17 módulos incorporados, con el nombre técnico de la app Django, el formato institucional al que corresponde y su responsabilidad funcional. Estos módulos son los que se documentaron, con sus pantallas, en el Manual de Usuario.

App Django	Formato	Responsabilidad
fo_th_001_personal	FO-TH-001	Gestión de información personal y datos laborales del colaborador.
fo_th_002_usuarios	FO-TH-002	Administración de usuarios y accesos relacionados al sistema.
fo_th_003_decimos	FO-TH-003	Control y gestión de la acumulación/mensualización de décimos.
fo_th_004_induccion	FO-TH-004	Procesos de inducción y reinducción del personal.
fo_th_005_entrega_puesto	FO-TH-005	Control documental del acta de entrega y recepción de puesto.
fo_th_006_evaluacion_prueba	FO-TH-006	Evaluaciones de desempeño durante el período de prueba.
fo_th_007_evaluacion_desempeno	FO-TH-007	Matrices de evaluación del desempeño laboral.
fo_th_008_control_vacaciones	FO-TH-008	Administración y control de saldos y períodos de vacaciones.
fo_th_009_novedades_th	FO-TH-009	Registro de novedades reportadas por Talento Humano.
fo_th_010_permiso	FO-TH-010	Gestión de solicitudes de permisos laborales.
fo_th_011_vacaciones	FO-TH-011	Gestión de solicitudes y registros de vacaciones.
fo_th_012_liquidacion_haberes	FO-TH-012	Procesos de liquidación de haberes (décimos, desahucio, indemnización).
fo_th_013_liquidacion_vacaciones	FO-TH-013	Liquidación correspondiente a vacaciones no gozadas.
fo_th_014_control_pasantia	FO-TH-014	Seguimiento de pasantías, tutores y actividades.
fo_th_015_ficha_entrevista	FO-TH-015	Registro y valoración de entrevistas de reclutamiento.
fo_th_016_actividades_virtuales	FO-TH-016	Gestión de informes de actividades en modalidad virtual.
fo_th_017_capacitacion	FO-TH-017	Administración del plan anual de capacitación.

Nota: Cada módulo se registra en INSTALLED_APPS de settings.py y expone sus rutas propias mediante un include() en el urls.py raíz del proyecto, tal como se detalla en la sección 8 de este manual.

4. Cambios realizados sobre Horilla

Esta sección detalla, con precisión, qué se modificó y qué se mantuvo intacto respecto al proyecto Horilla original. Esta distinción es crítica para futuras actualizaciones del framework base.

4.1 Resumen de cambios

- Integración visual del acceso de Talento Humano: se añadió una entrada en el menú lateral de Horilla que agrupa los módulos fo_th_* bajo las secciones existentes (Empleado, Incorporación, Licencia, Nómina, Desempeño, Asistencia, Reclutamiento).
- Incorporación de los módulos en la configuración del proyecto (INSTALLED_APPS y urls.py raíz).
- Integración del sistema de permisos: cada módulo declara sus propios permisos (ver, crear, editar, eliminar) siguiendo la convención nativa de Django.
- Asociación de permisos por usuario: los administradores pueden activar o desactivar el acceso a cada módulo desde la pantalla de Permisos de empleados.
- Compatibilidad total con usuarios administradores y usuarios normales, respetando la jerarquía de roles ya definida en Horilla.

4.2 Qué NO se modificó

Por decisión de diseño, no se modificaron los módulos originales de Horilla (employee, recruitment, onboarding, attendance, leave, payroll, etc.). Todo el desarrollo propio vive en apps nuevas (fo_th_*) que consumen los modelos y utilidades de Horilla cuando es necesario (por ejemplo, referenciando al modelo Employee mediante ForeignKey), pero sin alterar su código fuente.

> **Importante:** Mantener esta separación es lo que permite actualizar Horilla a versiones futuras del framework base con el menor riesgo posible de conflictos.

5. Sistema de permisos

El sistema utiliza el modelo de permisos nativo de Django (django.contrib.auth.models.Permission), asociado a los ContentType de cada modelo. Por cada modelo de Django se generan automáticamente 4 permisos base al ejecutar las migraciones:

Acción	Codename generado	Descripción
Ver información	view_<modelo>	Permite consultar/listar registros del módulo.
Crear registros	add_<modelo>	Permite crear nuevos registros.
Editar registros	change_<modelo>	Permite modificar registros existentes.
Eliminar registros	delete_<modelo>	Permite eliminar registros.

Por ejemplo, para el módulo FO-TH-001 (fo_th_001_personal), con un modelo llamado DatosPersonales, Django genera automáticamente los permisos: view_datospersonales, add_datospersonales, change_datospersonales y delete_datospersonales.

5.1 Verificación de permisos en las vistas

En las vistas basadas en función se recomienda usar el decorador @permission_required; en vistas basadas en clase, el mixin PermissionRequiredMixin:

```
# views.py (vista basada en función)
from django.contrib.auth.decorators import permission_required

@permission_required('fo_th_001_personal.view_datospersonales', raise_exception=True)
def lista_personal(request):
    registros = DatosPersonales.objects.all()
    return render(request, 'fo_th_001_personal/lista.html', {'registros': registros})

# views.py (vista basada en clase)
from django.contrib.auth.mixins import PermissionRequiredMixin
from django.views.generic import CreateView

class DatosPersonalesCreateView(PermissionRequiredMixin, CreateView):
    permission_required = 'fo_th_001_personal.add_datospersonales'
    model = DatosPersonales
    form_class = DatosPersonalesForm
    template_name = 'fo_th_001_personal/form.html'
```

5.2 Verificación de permisos en las plantillas

En las plantillas HTML se puede condicionar la visibilidad de botones o secciones usando la variable perms, disponible por defecto gracias al context processor de autenticación de Django:

```
{% if perms.fo_th_001_personal.add_datospersonales %}
  <a href="{% url 'fo_th_001_personal:crear' %}" class="btn btn-primary">+ Nuevo registro</a>
{% endif %}
```

5.3 Permisos personalizados adicionales

Cuando una acción no corresponde a un CRUD estándar (por ejemplo, "aprobar" una solicitud de vacaciones o "anular" una liquidación), se declara un permiso personalizado en la clase Meta del modelo:

```
# models.py
class SolicitudVacaciones(models.Model):
    # ... campos del modelo ...

    class Meta:
        permissions = [
            ('aprobar_solicitudvacaciones', 'Puede aprobar solicitudes de vacaciones'),
            ('anular_solicitudvacaciones', 'Puede anular solicitudes de vacaciones'),
        ]
```

Nota: Después de declarar permisos personalizados es obligatorio generar y aplicar una migración (makemigrations y migrate) para que Django los registre en la base de datos.

6. Administración de usuarios y asignación de permisos

La asignación de permisos a nivel funcional (para usuarios que no son superusuarios de Django) se realiza desde la interfaz web del sistema, en la pantalla "Permisos de empleados". Esta pantalla es una capa de administración construida sobre el modelo de permisos de Django, pensada para que un administrador de Talento Humano —sin conocimientos técnicos— pueda gestionar accesos sin entrar al panel de administración de Django.

Pasos para asignar permisos desde la interfaz

1. Ingresar al sistema como administrador.
2. Ir al módulo "Empleado".
3. Abrir "Permisos de empleados".
4. Seleccionar el usuario al que se le asignarán o revocarán accesos.
5. Activar o desactivar los permisos del módulo requerido (ver, crear, editar, eliminar, o permisos personalizados como aprobar/anular).
6. Guardar los cambios.

Alternativa desde el panel de administración de Django

Para tareas masivas o de soporte técnico, también es posible gestionar permisos directamente desde /admin/, agrupando permisos en Grupos (django.contrib.auth.models.Group) y asignando esos grupos a los usuarios, en lugar de asignar permisos uno por uno:

```
# Ejemplo: crear un grupo "Analista TH" con permisos de solo lectura
# sobre varios módulos, vía shell de Django (python manage.py shell)

from django.contrib.auth.models import Group, Permission

grupo, _ = Group.objects.get_or_create(name='Analista TH')
codenames = [
    'view_datospersonales',
    'view_solicitudvacaciones',
    'view_ficha_entrevista',
]
permisos = Permission.objects.filter(codename__in=codenames)
grupo.permissions.set(permisos)
```

Nota: Usar Grupos es preferible cuando varios usuarios comparten el mismo perfil de acceso, ya que evita repetir la asignación individual y facilita auditar quién tiene qué nivel de acceso.

7. Instalación del proyecto

7.1 Requisitos previos

- Python 3.10 o superior instalado en el servidor o equipo de desarrollo.
- Gestor de paquetes pip actualizado.
- Motor de base de datos disponible (MySQL, PostgreSQL recomendado en producción; SQLite es suficiente para desarrollo local).
- Git para clonar/actualizar el repositorio del proyecto.
- Node.js (opcional) si se requiere compilar assets de frontend adicionales.

7.2 Pasos de instalación

1. Copiar o clonar el proyecto en el servidor o equipo local.
2. Crear un entorno virtual de Python, para aislar las dependencias del proyecto.

```
python -m venv venv
```

3. Activar el entorno virtual.

```
# Windows
venv\Scripts\activate

# Linux / macOS
source venv/bin/activate
```

4. Instalar las dependencias del proyecto.

```
pip install -r requirements.txt
```

5. Configurar las variables de entorno (base de datos, clave secreta, modo debug) normalmente en un archivo .env leído por settings.py.

```
# .env (ejemplo)
DEBUG=False
SECRET_KEY=coloque-aqui-una-clave-larga-y-aleatoria
DATABASE_URL=postgres://usuario:clave@localhost:5432/horilla_th
ALLOWED_HOSTS=talentohumano.ganansol.fin.ec
```

6. Ejecutar las migraciones de base de datos, incluyendo las de los módulos fo_th_*.

```
python manage.py migrate
```

7. Crear un usuario administrador (superusuario).

```
python manage.py createsuperuser
```

8. Recolectar archivos estáticos (CSS, JS, íconos) para producción.

```
python manage.py collectstatic --noinput
```

9. Ejecutar el servidor.

```
# Desarrollo
python manage.py runserver

# Producción (ejemplo con Gunicorn)
gunicorn horilla.wsgi:application --bind 0.0.0.0:8000
```

Nota: En producción, colocar Gunicorn/uWSGI detrás de un proxy inverso como Nginx, habilitar HTTPS y desactivar DEBUG en settings.py.

8. Estructura de desarrollo de un módulo

Cada módulo fo_th_XXX_nombre es una app Django independiente y debe mantener siempre la misma estructura interna, para que cualquier desarrollador pueda orientarse rápidamente entre módulos:

```
fo_th_XXX_nombre/
├── __init__.py
├── apps.py           # Configuración de la app (AppConfig)
├── admin.py           # Registro de modelos en /admin
├── models.py          # Modelos de base de datos
├── forms.py           # Formularios (ModelForm)
├── views.py           # Lógica de negocio (vistas)
├── urls.py            # Rutas propias del módulo
├── templates/
│   └── fo_th_XXX_nombre/
│       ├── lista.html
│       ├── form.html
│       └── detalle.html
├── static/            # CSS/JS específicos del módulo (si aplica)
└── migrations/        # Historial de cambios de base de datos
```

8.1 models.py

Define la estructura de datos del módulo. Debe declarar explícitamente los permisos personalizados que necesite en su clase Meta (ver sección 5.3).

```
from django.db import models
from employee.models import Employee  # modelo original de Horilla

class DatosPersonales(models.Model):
    empleado = models.ForeignKey(Employee, on_delete=models.CASCADE)
    cedula = models.CharField(max_length=13, unique=True)
    direccion = models.CharField(max_length=255, blank=True)
    fecha_nacimiento = models.DateField(null=True, blank=True)
    estado_civil = models.CharField(max_length=30, blank=True)
    creado_en = models.DateTimeField(auto_now_add=True)

    class Meta:
        verbose_name = 'Dato personal'
        verbose_name_plural = 'FO-TH-001 Base de datos personal'
```

8.2 forms.py

```
from django import forms
from .models import DatosPersonales

class DatosPersonalesForm(forms.ModelForm):
    class Meta:
        model = DatosPersonales
        fields = ['empleado', 'cedula', 'direccion', 'fecha_nacimiento', 'estado_civil']
```

8.3 views.py

```
from django.contrib.auth.mixins import PermissionRequiredMixin
```

```
from django.views.generic import ListView, CreateView, UpdateView, DeleteView
from .models import DatosPersonales
from .forms import DatosPersonalesForm

class DatosPersonalesListView(PermissionRequiredMixin, ListView):
    permission_required = 'fo_th_001_personal.view_datospersonales'
    model = DatosPersonales
    template_name = 'fo_th_001_personal/lista.html'
    context_object_name = 'registros'

class DatosPersonalesCreateView(PermissionRequiredMixin, CreateView):
    permission_required = 'fo_th_001_personal.add_datospersonales'
    model = DatosPersonales
    form_class = DatosPersonalesForm
    template_name = 'fo_th_001_personal/form.html'
    success_url = '/fo-th-001/'
```

8.4 urls.py

```
from django.urls import path
from . import views

app_name = 'fo_th_001_personal'

urlpatterns = [
    path('', views.DatosPersonalesListView.as_view(), name='lista'),
    path('nuevo/', views.DatosPersonalesCreateView.as_view(), name='crear'),
    path('<int:pk>/editar/', views.DatosPersonalesUpdateView.as_view(), name='editar'),
    path('<int:pk>/eliminar/', views.DatosPersonalesDeleteView.as_view(), name='eliminar'),
]
```

8.5 Registro del módulo en el proyecto

Cada módulo nuevo debe registrarse en dos lugares del proyecto principal:

```
# horilla/settings.py
INSTALLED_APPS = [
    # ... apps de Django y de Horilla ...
    'fo_th_001_personal',
    'fo_th_002_usuarios',
    'fo_th_003_decimos',
    # ... resto de módulos fo_th_* ...
]

# horilla/urls.py
from django.urls import path, include

urlpatterns = [
    # ... rutas de Horilla ...
    path('fo-th-001/', include('fo_th_001_personal.urls')),
    path('fo-th-002/', include('fo_th_002_usuarios.urls')),
    # ... resto de módulos fo_th_* ...
]
```

8.6 admin.py

Registrar el modelo en el panel de administración facilita las tareas de soporte y depuración:

```
from django.contrib import admin
from .models import DatosPersonales

@admin.register(DatosPersonales)
class DatosPersonalesAdmin(admin.ModelAdmin):
    list_display = ('empleado', 'cedula', 'estado_civil', 'creado_en')
    search_fields = ('cedula', 'empleado__nombre')
```

8.7 migrations

Toda modificación a models.py debe ir acompañada de su migración correspondiente, generada y aplicada así:

```
python manage.py makemigrations fo_th_001_personal
python manage.py migrate
```

9. Convenciones de código y buenas prácticas

- Nombrar las apps siempre con el prefijo fo_th_XXX_nombre, usando el número de formato con tres dígitos (001, 002, …).
- Los nombres de modelos, campos y variables en español o inglés, pero de forma consistente dentro de un mismo módulo (se recomienda mantener el idioma ya usado en el módulo).
- Usar ModelForm en lugar de construir formularios HTML manuales, para aprovechar validación automática.
- Usar vistas basadas en clase (Class-Based Views) con PermissionRequiredMixin para mantener consistencia con el resto del sistema.
- Cada módulo debe tener su propio archivo de rutas (urls.py) con un app_name único, evitando colisiones de nombres de URL.
- No editar directamente archivos de las apps originales de Horilla; si se requiere extender su comportamiento, hacerlo mediante señales (signals), sobreescritura de plantillas o nuevas apps que las referencien.
- Mantener las migraciones pequeñas y descriptivas; evitar migraciones acumuladas sin aplicar en el repositorio.
- Escribir mensajes de commit claros indicando el módulo afectado, por ejemplo: fo_th_011_vacaciones: valida fechas de solicitud.

10. Recomendaciones de mantenimiento

Antes de modificar código en producción o en la rama principal del repositorio:

- Realizar copia de seguridad de la base de datos y, de ser posible, del código desplegado.
- Probar todo cambio primero en un ambiente de desarrollo o staging idéntico a producción.
- Crear migraciones controladas: revisar el archivo generado por makemigrations antes de aplicarlo, especialmente si involucra pérdida de datos (eliminar o renombrar campos).
- Evitar modificar archivos centrales de Horilla; si es indispensable, documentar el cambio y evaluar el impacto en futuras actualizaciones del framework.
- Mantener actualizado el archivo requirements.txt cuando se añadan nuevas dependencias.
- Ejecutar pruebas (si existen) antes de fusionar cambios a la rama principal.

Nota: Se recomienda establecer un ambiente de staging que refleje producción para validar migraciones y cambios de permisos antes de aplicarlos en vivo.

11. Solución de problemas

Si un módulo no aparece en el sistema o presenta errores, revisar en este orden:

Síntoma	Posible causa	Acción recomendada
El módulo no aparece en el menú	No está registrado en el menú o el usuario no tiene el permiso view_*	Revisar la integración visual y los permisos asignados al usuario
Error 404 al abrir el módulo	Rutas no incluidas en el urls.py raíz	Revisar que exista el path(...) con include() correspondiente
Error "No module named ..."	App no instalada o entorno virtual no activado	Revisar INSTALLED_APPS y activar el entorno virtual antes de ejecutar
"403 Forbidden" al usar una acción	Falta el permiso específico para esa acción	Revisar permisos del usuario en Permisos de empleados o en /admin/
Error de columna/tabla inexistente	Migraciones pendientes de aplicar	Ejecutar python manage.py makemigrations y migrate
Cambios de código no se reflejan	Servidor de desarrollo no reiniciado, o caché de estáticos	Reiniciar el servidor Django y ejecutar collectstatic si aplica

Comandos de diagnóstico útiles

```
python manage.py check               # valida la configuración del proyecto
python manage.py showmigrations       # lista migraciones aplicadas/pendientes
python manage.py migrate --plan       # muestra qué migraciones se aplicarían
python manage.py shell                # consola interactiva para inspeccionar datos
```

AUTORIZACIÓN DE PUBLICACIÓN EN EL REPOSITORIO INSTITUCIONAL

Jonnathan Gustavo Paucar Paucar portador(a) de la cédula de ciudadanía Nº **0302546098** En calidad de autor/a y titular de los derechos patrimoniales del trabajo de investigación **"Desarrollo de un sistema web para la gestión de talento humano en la COAC Gañansol utilizando técnicas de Vibe coding",** de conformidad a lo establecido en el artículo 114 Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, reconozco a favor de la Universidad Católica de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos y no comerciales. Autorizo además a la Universidad Católica de Cuenca**,** para que realice la publicación de éste trabajo de investigación en el Repositorio Institucional de conformidad a lo dispuesto en el artículo 144 de la Ley Orgánica de Educación Superior.

Cañar, 07 de octubre de 2026



……

Jonnathan Gustavo Paucar Paucar
C.I. **0302546098**