



Proyecto final:

Sistema de información para la gestión de procesos administrativos de RRHH en YPFB- Andina

ESTUDIANTES:

- Dax Kenji Tellez Duran
- Alejandro Alpire Céspedes.
- Pablo Ignacio Kantuta Eguez
- Emilio Arturo Saucedo Melgar

DOCENTE: Eivy Pereyra Carvalho

MATERIA: Desarrollo de Software y Base de datos

FECHA: XX/08/2025

SANTA CRUZ – BOLIVIA

índice

Tabla de contenido

1.....	Introducción
2.....	Objetivo general
3.....	Objetivo específico
4.....	Planificación temporal
5.....	Marco teórico conceptual
1.1	Historias de Usuarios
1.2	Casos de Uso:
1.3	Requisitos funciones
1.4	Requisitos No Funcionales
6.....	Teoría de la metodología
7.....	Estimaciones
8.....	Actividades específicas para el ACS
9.....	Metas, atributos y métricas aplicadas (Cap. 16)
10.....	Factores evaluados del software (Cap. 14)
11.....	Líneas para las revisiones técnicas (Cap. 15)
12.....	Registro de las revisiones técnicas (Cap. 17)
13.....	Justificación de las estrategias de prueba (Cap. 22)
14.....	Cómo se administró la configuración
15.....	Diagramas de clases
16.....	Mapeo

17.....	Diseño de reportes parametrizados
18.....	Despliegue de datos
19.....	Registro de las pruebas
20.....	Presupuesto
21.....	Conclusiones, recomendaciones y bibliografía

1. Introducción

La gestión de Recursos Humanos (RRHH) se ha consolidado como un pilar estratégico indispensable en el panorama corporativo y energético actual, particularmente en sectores de gran escala como el petrolero. La modernización de procesos y sistemas en esta área es crucial para que las empresas mantengan su competitividad y eficiencia operativa en un entorno de mercado dinámico y en constante evolución.

No obstante, muchas organizaciones enfrentan desafíos significativos en la administración de RRHH, que a menudo se derivan de la dependencia de procesos manuales y la fragmentación de sistemas existentes. Estos problemas comunes resultan en ineficiencias operativas, dificultades en el control de tiempos y una gestión subóptima del capital humano. En el caso específico de YPFB-Andina, una empresa boliviana con 335 empleados dedicada a la exploración y perforación petrolera, la gestión manual y poco automatizada de su información de RRHH ha generado problemas directos en procesos clave. Esta situación se traduce en retrasos, errores humanos, y un impacto negativo en la satisfacción de los empleados, además de posibles sanciones externas.

Para abordar esta problemática crítica, se propone el desarrollo de un sistema integral de gestión de Recursos Humanos. Este proyecto busca optimizar y automatizar los procesos esenciales, permitiendo una organización mejorada de los datos de los empleados, minimizando errores y agilizando los tiempos de procesamiento para asegurar el cumplimiento normativo. La implementación de esta solución tecnológica no solo mejorará la eficiencia operativa y reducirá los errores, sino que también contribuirá a la satisfacción laboral y la competitividad de la empresa.

El presente documento detalla el proceso sistemático para el desarrollo de este software. Se seguirá una estructura de ingeniería de software formal, que abarca las actividades fundamentales de comunicación, planificación, modelado, construcción y despliegue. El objetivo es proporcionar una guía clara y estructurada para la ejecución del proyecto, desde la definición de requisitos iniciales hasta la fase de pruebas rigurosas, garantizando la entrega de un producto de alta calidad que satisfaga las necesidades de YPFB-Andina

2. Objetivo general

Desarrollar un sistema de información para gestionar procesos administrativos de recursos humanos para la empresa YPFB-Andina.

3. Objetivo específico

Registrar los requisitos funcionales y no funcionales del sistema de gestión de Recursos Humanos para YPFB-Andina. Esto incluye procesos clave como reclutamiento, creación de kardex, gestión de pagos, afiliaciones al seguro médico, control de vacaciones y elaboración de finiquitos, mediante la elaboración de casos de uso y modelos de requisitos que representen la visión del usuario y del dominio de información.

Diseñar la arquitectura del software, la estructura de la base de datos y las interfaces de usuario del sistema de RRHH para (definir) YPFB-Andina. Este diseño se realizará utilizando diagramas UML (casos de uso, clases, comportamiento y despliegue) que reflejen los requisitos identificados y aseguren un diseño comprensible y de alta calidad.

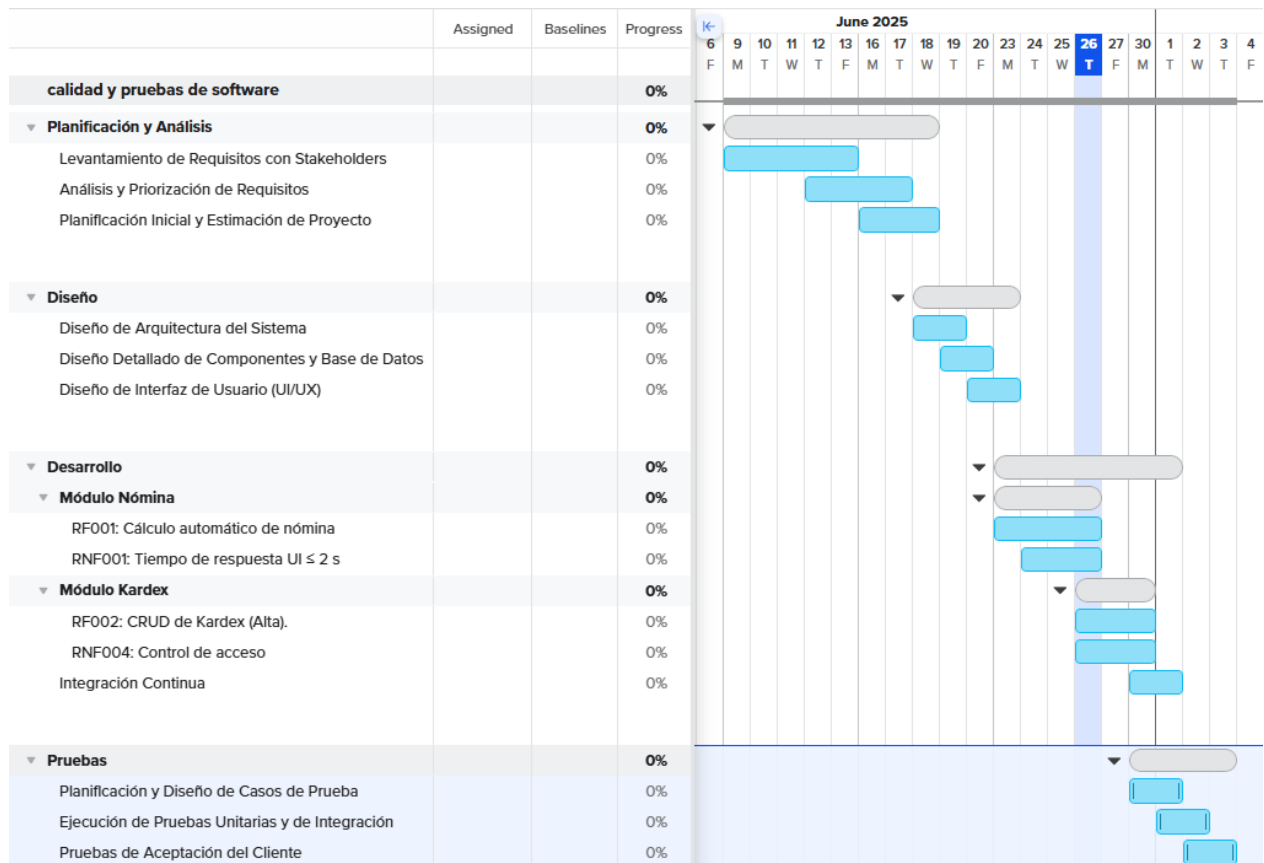
Codificar los módulos y componentes del sistema de Recursos Humanos, siguiendo el diseño establecido y aplicando buenas prácticas de programación. Se asegurará la coherencia con la arquitectura definida y la implementación de los requisitos funcionales y no funcionales.

Planificar un conjunto exhaustivo de pruebas (unitarias, de integración, de validación y de sistema) para el software de RRHH. El objetivo es validar que el sistema cumple con todos los requisitos funcionales y no funcionales de YPFB-Andina, y que satisface los atributos de calidad del software definidos por el modelo McCall, como funcionalidad, usabilidad, confiabilidad, eficiencia e integridad.

Implementar el sistema de Recursos Humanos en un entorno controlado dentro de YPFB-Andina. Se realizarán las pruebas de despliegue y configuración necesarias y se

facilitará la validación final por parte de los usuarios, asegurando la adopción y el soporte continuo del sistema.

4. Planificación temporal



5. Marco teórico conceptual

Este marco define los conceptos fundamentales que sustentan el diseño, desarrollo e implementación del sistema de Recursos Humanos, siguiendo los principios de la Ingeniería de Software de Roger Pressman y los atributos de calidad.

Área de Dominio (Negocio):

1. Gestión de Recursos Humanos (RRHH)

Definición: La Gestión de Recursos Humanos se refiere al conjunto de procesos y actividades que buscan administrar el capital humano de una organización de manera eficiente y efectiva. En el contexto de la ingeniería de software, implica el análisis de las necesidades del usuario para desarrollar, operar y mantener un software que gestione el personal de una empresa de forma rentable y confiable. Abarca desde la adquisición y administración de información del personal hasta la gestión de beneficios y desvinculaciones.

Relevancia para el proyecto de RRHH en YPFB-Andina: Este sistema está diseñado para centralizar toda la información y los procesos de RRHH de YPFB-Andina, permitiendo una gestión eficiente del personal. Su relevancia radica en la automatización del cálculo preciso de sueldos, descuentos, aportes y bonificaciones, la actualización del kardex de empleados, el control de vacaciones, la afiliación y desafiliación al seguro médico, la generación de reportes y la gestión de finiquitos. Esto busca asegurar el cumplimiento normativo y evitar inconsistencias en la gestión del capital humano de una empresa con 335 empleados.

2. Optimización de Procesos de Negocio (BPO)

Definición: Aunque las fuentes no usan el término exacto "Optimización de Procesos de Negocio (BPO)", se refieren a la Reingeniería de Procesos de Empresa (RPE) como la "búsqueda, e implementación, de cambios radicales en los procesos de las empresas para lograr resultados innovadores". Estos cambios tienen como motores clave la reducción de costo, reducción de tiempo, mejora de la calidad y desarrollo y fortalecimiento del personal. Un proceso rediseñado debe ser prototipado antes de su integración plena para asegurar refinamientos.

Relevancia para el proyecto de RRHH en YPFB-Andina: La implementación de este sistema de RRHH tiene un objetivo fundamental de optimizar los procesos internos. Por ejemplo, busca reducir en un 50% el tiempo de registro y actualización de información en el Kardex y

disminuir el tiempo de procesamiento de nóminas en un 70%, al mismo tiempo que aumenta la precisión de los cálculos al 99.9%. Esto se alinea directamente con los principios de RPE al buscar una mejora radical en la eficiencia, productividad y precisión, minimizando los tiempos y costos operativos asociados a la gestión de personal.

3. Ciclo de Vida del Empleado

Definición: Las fuentes no proporcionan una definición formal del "Ciclo de Vida del Empleado". Sin embargo, las funcionalidades descritas para el sistema de RRHH abarcan las principales etapas de la relación de un individuo con la empresa. Esto incluye la gestión de información desde el reclutamiento (mencionado por el rol de "Reclutador"), el registro y actualización de información (kardex), la gestión de beneficios (nómina, seguro médico, vacaciones) durante su permanencia, hasta el proceso de desvinculación (finiquitos).

Relevancia para el proyecto de RRHH en YPFB-Andina: El sistema busca dar soporte a todo el ciclo de vida del empleado dentro de YPFB-Andina. Esto es crucial para mantener un registro completo y preciso del personal a lo largo de su trayectoria en la empresa. La digitalización y automatización de estos procesos aseguran que la información esté disponible y sea precisa para la toma de decisiones, desde la planificación de recursos humanos hasta el cumplimiento de obligaciones laborales al finalizar una relación de trabajo.

4. Evaluación de Desempeño

Definición: La fuente menciona la "evaluación de desempeño" como una de las "consideraciones de diseño" para las que se sugiere usar "sentido común". No se proporciona una definición detallada ni se describe su funcionalidad específica dentro del alcance del sistema de RRHH.

Relevancia para el proyecto de RRHH en YPFB-Andina: Aunque la "evaluación de desempeño" es una parte integral de la gestión de RRHH y es una "consideración de diseño", las especificaciones actuales del proyecto de YPFB-Andina no detallan funcionalidades específicas para este aspecto. Esto implica que, en la fase actual del proyecto, la prioridad está en otros módulos críticos (nómina, kardex, vacaciones, seguros, finiquitos). Su inclusión futura podría complementar el sistema, proporcionando herramientas para el seguimiento y desarrollo del personal, lo que se alinea con la meta de un registro completo del personal.

Conceptos básicos del caso en Bolivia:

Kardex de empleados: Se refiere a un registro completo y preciso del personal. El rol de Administrador RRHH tiene la responsabilidad de registrar y actualizar la información del kardex de los empleados para mantener este registro. Los Empleados pueden solicitar actualizaciones de datos en su Kardex.

Nómina: Es la gestión automática de los pagos a los empleados, asegurando el cálculo preciso de sueldos, descuentos, aportes a AFP y bonificaciones. El Administrador RRHH se encarga de gestionar automáticamente la nómina.

Gestión Salarial: Implica la existencia de doce sueldos al año, un sueldo por mes.

Vacaciones: Se gestionan las solicitudes y el control de los días de descanso de los empleados para llevar un seguimiento eficiente y evitar inconsistencias. Un Empleado puede hacer una o muchas solicitudes de vacaciones. El Administrador RRHH gestiona estas solicitudes y su control. El Gerente de Área recibe notificaciones automáticas cuando un miembro de su equipo solicita vacaciones para poder aprobarlas o rechazarlas oportunamente.

Seguro Médico / Afiliación y Desafiliación a la CPS: Se refiere al proceso de afiliar y desafiliar empleados al seguro médico para garantizar la correcta cobertura y el cumplimiento

con entidades externas como la Caja Petrolera de Salud (CPS). Este caso de uso es importante porque garantiza que la empresa cumpla con las normas laborales.

Finiquitos: Concepto relacionado con la gestión de los finiquitos de los empleados salientes. El objetivo es asegurar un proceso de desvinculación correcto y evitar sanciones legales o financieras.

Reportes Automatizados: Se refiere a la capacidad de generar reportes automatizados de pagos, vacaciones y afiliaciones para obtener información que apoye la toma de decisiones y el cumplimiento normativo.

Centralización de Información de RRHH: La necesidad de integrar todos los procesos de RRHH en una única plataforma accesible a todas las áreas correspondientes, lo que permite al Administrador RRHH acceder a una única plataforma que integre la gestión del personal de manera eficiente.

Anticipos: Un empleado puede tener uno o muchos anticipos a lo largo de su contrato, y un anticipo corresponde únicamente a un empleado.

Documentos de empleados y dependientes: Tanto los empleados como sus dependientes pueden tener uno o más documentos, y un tipo de documento puede ser para muchas personas.

Roles de Usuario en RRHH:

- **Administrador RRHH:** Encargado de las funciones centrales de gestión del personal.
- **Empleado:** Trabajador que interactúa con el sistema para gestionar y consultar su información.

- **Gerente de Área:** Supervisor que aprueba solicitudes y consulta información de su equipo.
- **Reclutador:** Persona que gestiona el proceso de selección, incluyendo la publicación de vacantes, el seguimiento de postulaciones y la programación de entrevistas y evaluaciones. Esto implica conceptos como vacantes de empleo, candidatos, postulaciones, entrevistas y evaluaciones dentro del proceso de reclutamiento.

Consideraciones adicionales: Aunque no se detallan, se mencionan como "demás consideraciones de diseño" relacionadas con RRHH, elementos como la lactancia, la evaluación de desempeño y el IVA.

Área de Solución (Tecnología):

1. Sistemas de Información

Definición: Un sistema de información es un grupo de procesos, habilidades, procedimientos y documentación que ayuda a los desarrolladores a lograr la implementación de nuevos sistemas de información. La Ingeniería de Software, como estudio práctico sistemático y disciplinado, se encarga de analizar las necesidades del usuario para desarrollar, operar y mantener software que sea rentable, confiable y funcional en máquinas reales.

Relevancia para el proyecto de RRHH en YPFB-Andina: El proyecto "Sistema de RRHH para YPFB-Andina" es, en esencia, la implementación de un nuevo sistema de información. Su desarrollo requiere la aplicación de principios de ingeniería de software para transformar las necesidades de gestión de RRHH de YPFB-Andina en un software funcional. El sistema busca centralizar y digitalizar los procesos de RRHH, lo cual es el objetivo principal de un sistema de información para mejorar la eficiencia y la toma de decisiones en la empresa.

2. Bases de Datos Relacionales (y el concepto de Mapeo Objeto-Relacional - ORM)

Definición: Una base de datos es un centro de acumulación y almacenamiento de información. En el modelado de requerimientos, si el software necesita crear, extender o hacer interfaz con una base de datos, o manipular estructuras de datos complejas, se puede crear un modelo de datos. El Diagrama Entidad-Relación (DER) es una herramienta utilizada para representar todos los datos que se introducen, almacenan, transforman y generan dentro de una aplicación, así como la relación entre ellos. El análisis de datos es fundamental y debe ser el inicio de cualquier diseño de software. El concepto de Mapeo Objeto-Relacional (ORM) no es explícitamente definido en las fuentes proporcionadas.

Relevancia para el proyecto de RRHH en YPFB-Andina: La gestión de los recursos humanos en YPFB-Andina implica un gran volumen de datos estructurados (kardex de empleados, información de nómina, registros de vacaciones, afiliaciones, finiquitos). Una base de datos relacional es esencial para almacenar, organizar y gestionar esta información de manera eficiente y con integridad. La capacidad de definir relaciones entre diferentes conjuntos de datos asegura la consistencia y la facilidad de consulta y generación de reportes.

3. Arquitectura de Software (Ej: Modelo-Vista-Controlador o Arquitectura en 3 Capas)

Definición: La arquitectura de un sistema es el marco general que describe su forma y estructura: sus componentes y la manera en la que ajustan entre sí. Debe modelar la estructura de un sistema y la forma en que los datos y los componentes de procesamiento colaboran. La descripción arquitectónica se representa mediante múltiples perspectivas, cada una ofreciendo una vista del sistema desde un conjunto de preocupaciones relacionadas de un participante. Los componentes de software pueden ser de control, del dominio del problema o de infraestructura. Aunque las fuentes mencionan "capas de abstracción" en el contexto de reingeniería, los patrones específicos como Modelo-Vista-

Controlador (MVC) o Arquitectura en 3 Capas no se definen explícitamente en los materiales proporcionados.

Relevancia para el proyecto de RRHH en YPFB-Andina: Una arquitectura de software bien definida es fundamental para la construcción de un sistema robusto, escalable y mantenible. Para un sistema de RRHH, que maneja información sensible y procesos complejos, una arquitectura clara permite separar las preocupaciones (datos, lógica de negocio, interfaz de usuario), facilitando el desarrollo modular, las pruebas y el mantenimiento a largo plazo. Esto asegura que el sistema pueda adaptarse a futuros cambios y extensiones sin una reingeniería completa, lo cual es vital para la longevidad del sistema en YPFB-Andina.

Aplicación Web

Definición: Las aplicaciones web (webapps) son un tipo de software cuyo desarrollo implica consideraciones especiales para su análisis de requerimientos, incluyendo el modelo funcional (operaciones sobre contenido), el modelo de navegación (estrategia general de navegación) y el modelo de configuración (ambiente e infraestructura).

Relevancia para el proyecto de RRHH en YPFB-Andina: El sistema de RRHH está diseñado para ser utilizado por empleados, gerentes de área y administradores de RRHH, lo que implica la necesidad de una amplia accesibilidad y facilidad de uso. Desarrollarlo como una aplicación web permitirá el acceso desde cualquier ubicación con conexión a internet, lo cual es ideal para una empresa con una gran fuerza laboral distribuida como YPFB-Andina. Las consideraciones de diseño de webapps (funcionalidad, navegación y configuración) serán clave para asegurar una experiencia de usuario eficiente y un despliegue y mantenimiento adecuados.

Área de Calidad:

1. Modelo de Calidad de McCall (y sus factores principales: Operación, Revisión, Transición)

Definición: Las fuentes discuten la importancia de la calidad del software, la necesidad de establecer criterios técnicos para un buen diseño, y la aplicación de métricas de calidad para evaluar la eficacia de las acciones de ingeniería de software. Se mencionan atributos de calidad como la eficiencia, confiabilidad, seguridad, usabilidad y mantenibilidad [283, 3; RNF previos]. Sin embargo, el "Modelo de Calidad de McCall" como tal, y su clasificación específica en los factores de Operación, Revisión y Transición, no se detallan explícitamente en las fuentes proporcionadas de Pressman. No obstante, los factores de calidad que definen este modelo, como Rendimiento (Eficiencia), Seguridad (Integridad), Usabilidad, Fiabilidad (Confiabilidad) y Mantenibilidad, son atributos ampliamente reconocidos en la ingeniería de software y se alinean con los principios discutidos en Pressman sobre la entrega de software confiable y eficiente [283, 330, RNF previos].

Relevancia para el proyecto de RRHH en YPFB-Andina: La calidad es un pilar fundamental para el éxito del sistema de RRHH. Un sistema de RRHH debe ser altamente confiable (Fiabilidad) para garantizar la precisión del cálculo de salarios y vacaciones, seguro (Seguridad) para proteger la información sensible de los empleados, eficiente (Rendimiento) en sus tiempos de respuesta para una experiencia de usuario fluida [RNF previos], usable (Usabilidad) para facilitar su adopción por parte de todos los usuarios (empleados, gerentes, RRHH) [RNF previos], y mantenible (Mantenibilidad) para permitir futuras adaptaciones a políticas cambiantes y asegurar su longevidad. La aplicación de estos factores de calidad, aunque no se haga referencia explícita al modelo de McCall en las fuentes, es crucial para la satisfacción del cliente y el éxito a largo plazo del proyecto.

1.1 Historias de Usuarios

Roles Identificados

Basado en el documento de levantamiento de información y la problemática de YPFB-Andina, los roles principales que interactuarán con el sistema son:

1. Administrador RRHH 1
2. Empleado 2
3. Gerente de Área (deducido por la necesidad de aprobación de vacaciones y estructura organizacional en una empresa con 335 empleados)
4. Reclutador (deducido del "Proceso de reclutamiento" y la necesidad de gestionar el capital humano)

Historias de Usuario por Rol

A continuación, se presenta el listado de Historias de Usuario clave para cada rol:

Rol: Administrador RRHH

Como Administrador RRHH, quiero gestionar automáticamente la nómina de los empleados, para que pueda asegurar el cálculo preciso de sueldos, descuentos, aportes a AFP y bonificaciones¹.

Como Administrador RRHH, quiero registrar y actualizar la información del kardex de los empleados, para que pueda mantener un registro completo y preciso del personal de YPFB-Andina³....

Como Administrador RRHH, quiero gestionar las solicitudes y el control de vacaciones de los empleados, para que pueda llevar un seguimiento eficiente de los días de descanso y evitar inconsistencias³....

Como Administrador RRHH, quiero afiliar y desafiliar empleados al seguro médico, para que pueda garantizar la correcta cobertura y el cumplimiento con entidades externas como la CPS³....

Como Administrador RRHH, quiero generar reportes automatizados de pagos, vacaciones y afiliaciones, para que pueda obtener información para la toma de decisiones y el cumplimiento normativo⁸.

Como Administrador RRHH, quiero gestionar los finiquitos de los empleados salientes, para que pueda asegurar un proceso de desvinculación correcto y evitar sanciones legales o financieras⁴⁶.

Como Administrador RRHH, quiero establecer roles de usuario y permisos de acceso al sistema, para que pueda garantizar que solo el personal autorizado acceda a información sensible⁸.

Como Administrador RRHH, quiero centralizar toda la información de los procesos de RRHH, para que pueda acceder a una única plataforma que integre la gestión del personal de manera eficiente².

Rol: Empleado

Como Empleado, quiero solicitar y consultar mis días de vacaciones disponibles, para que pueda planificar mi tiempo libre de manera efectiva²⁶.

Como Empleado, quiero consultar y solicitar actualizaciones de mi información personal en mi kardex, para que pueda mantener mis datos al día y accesibles en todo momento.

Como Empleado, quiero revisar mi información centralizada en el sistema, para que pueda tener visibilidad sobre mis datos, beneficios y situación laboral².

Rol: Gerente de Área

Como Gerente de Área, quiero recibir una notificación automática cuando un miembro de mi equipo solicita vacaciones, para que pueda aprobarlas o rechazarlas de manera oportuna [Ejemplo proporcionado en la tarea].

Como Gerente de Área, quiero consultar la información relevante de RRHH de los miembros de mi equipo, para que pueda gestionar mejor a mi personal y apoyar sus necesidades.

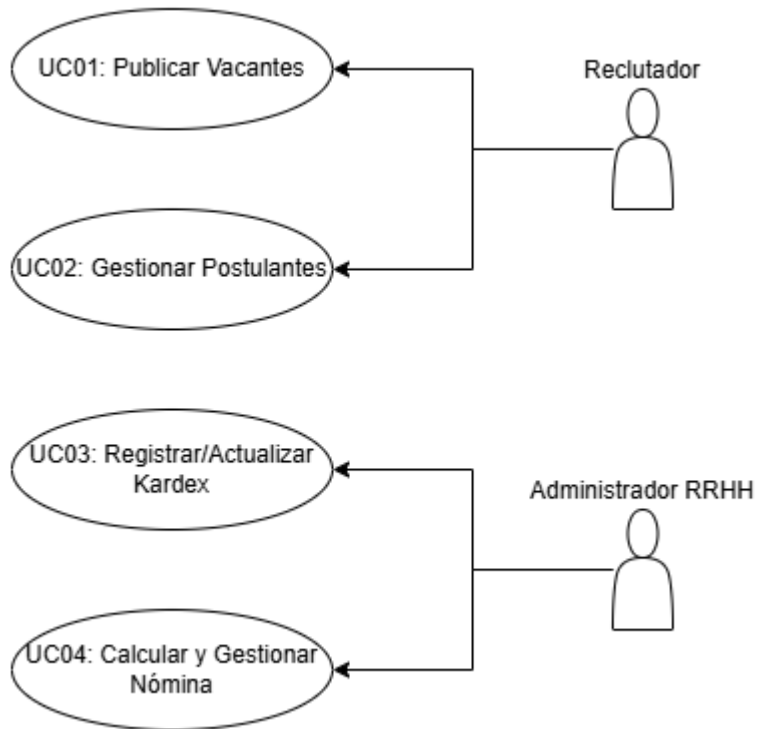
Rol: Reclutador

Como Reclutador, quiero publicar y gestionar nuevas vacantes de empleo, para que pueda atraer a candidatos cualificados para YPFB-Andina de manera eficiente³⁴.

Como Reclutador, quiero realizar el seguimiento de las postulaciones de los candidatos, para que pueda seguir el progreso de cada aplicante a través de las diferentes etapas del proceso de selección³⁴.

Como Reclutador, quiero programar y gestionar entrevistas y evaluaciones para los candidatos, para que pueda organizar el proceso de selección de manera estructurada y ágil.

1.2 Casos de Uso:



UC01: Publicar Vacantes

Actores: Reclutador (primario)

Precondición: Reclutador autenticado con rol "Reclutador".

Flujo Principal:

Reclutador selecciona "Crear nueva vacante".

Sistema despliega formulario con campos (título, descripción, requisitos, fecha límite).

Reclutador completa y presiona "Publicar".

Sistema valida datos y publica la vacante.

Sistema muestra confirmación de publicación.

Flujos Alternos:

A1: Faltan campos obligatorios → sistema resalta errores y vuelve al paso 2.

A2: Fecha límite inválida → mensaje de error y retorno al paso 2.

UC02: Gestionar Postulantes

Actores: Reclutador (primario), Gerente de Área (secundario, en entrevistas)

Precondición: Existe al menos una vacante publicada.

Flujo Principal:

Reclutador abre “Ver postulantes” de una vacante.

Sistema lista candidatos por estado (recibido, entrevista, rechazado).

Reclutador selecciona candidato y opta por “Programar entrevista”.

Sistema muestra calendario; reclutador fija fecha/hora y notifica a candidato.

Entrevista se realiza; reclutador registra resultado (aprobar/rechazar) y, si procede, involucra al Gerente para decisión final.

Flujos Alternos:

A1: Candidato no disponible → reclutador reprograma (volver al paso 4).

A2: Reclutador rechaza sin entrevista → sistema marca estado “Rechazado” y termina.

UC03: Registrar/Actualizar Kardex

Actores: Administrador RRHH (primario), Empleado (secundario, consulta)

Precondición: Empleado existe en la base de datos.

Flujo Principal:

Administrador busca empleado por ID o nombre.

Sistema muestra historial (kardex).

Administrador elige "Agregar" o "Editar entrada".

Introduce datos (cargo, departamento, fechas) y guarda.

Sistema actualiza kardex y confirma.

Flujos Alternos:

A1: Fechas inconsistentes → sistema muestra validación y devuelve al paso 4.

UC04: Calcular y Gestionar Nómina

Actores: Administrador RRHH (primario)

Precondición: Configuración de calendario y feriados cargada.

Flujo Principal:

Administrador lanza proceso de cálculo de nómina mensual.

Sistema recopila datos (horas, descuentos, AFP, bonificaciones).

Sistema aplica reglas y calcula montos individuales.

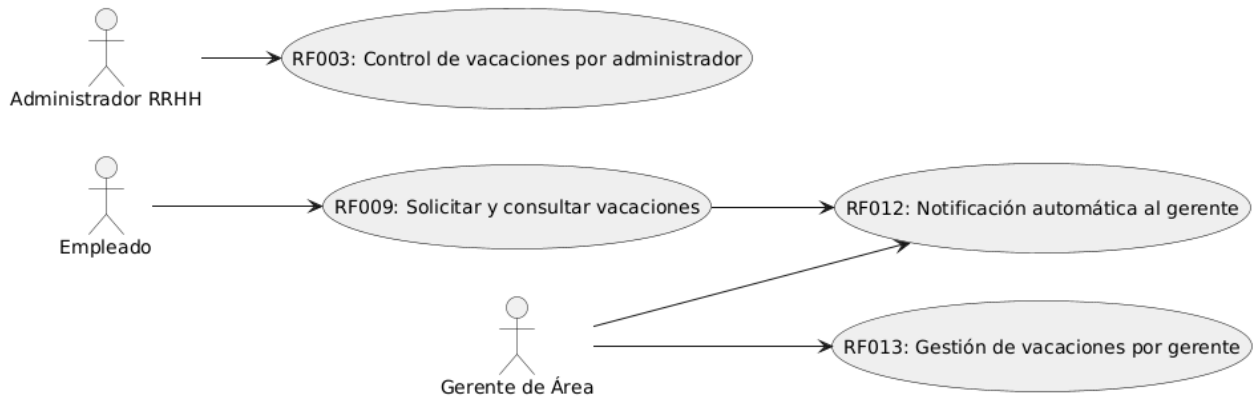
Para cesantes, genera finiquito según antigüedad.

Sistema presenta resultados en pantalla y en archivo exportable.

Administrador revisa, aprueba y el sistema registra el pago.

Flujos Alternos:

A1: Datos faltantes → sistema detiene cálculo y notifica errores.



UC05: Solicitar Vacaciones

Actores: Empleado (primario)

Gerente de Área (Secundario)

Administrador RRHH (secundario)

Precondición: Empleado autenticado y con saldo de días disponible.

Flujo Principal:

Empleado accede al módulo y elige “Solicitar vacaciones”.

Indica rango de fechas.

Sistema valida saldo y bloqueos (feriados/proyectos críticos).

Si procede, crea solicitud y notifica al Gerente de Área.

Gerente accede a solicitudes pendientes y revisa la solicitud.

Gerente aprueba o rechaza:

- Si aprueba, el Administrador RRHH registra las fechas en el calendario corporativo.
- Si rechaza, se solicita motivo.

Empleado recibe notificación del resultado.

Flujos Alternos:

A1: Saldo insuficiente → sistema rechaza y avisa al Empleado.

A2: Fechas en blackout → sistema sugiere alternativas o rechaza.

A3: Proyecto crítico bloquea fechas → Gerente propone nuevo rango y se reinicia el proceso.



UC06: Generar Reportes de RRHH

Actores: Administrador RRHH (primario)

Precondición: Datos de nómina, vacaciones y afiliaciones actualizados.

Flujo Principal:

Selecciona tipo de reporte (Nómina, Vacaciones, Afiliaciones, Finiquitos).

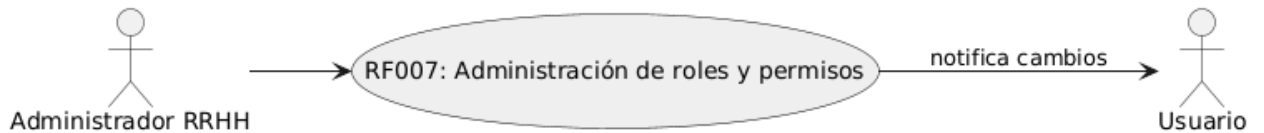
Define rango de fechas y formato (PDF/Excel).

Sistema compila datos y genera archivo.

Administrador descarga o envía por e-mail.

Flujos Alternos:

A1: No hay registros en el rango → sistema muestra “No hay datos”.



UC07: Gestionar Roles y Permisos

Actores: Administrador RRHH (primario), Usuario (secundario, recibe cambios)

Precondición: Usuario autenticado como Administrador.

Flujo Principal:

Accede a “Gestión de permisos”.

Selecciona usuario o rol a modificar.

Asigna o revoca accesos a módulos.

Guarda cambios; sistema actualiza políticas.

Notifica al Usuario afectado (si está en línea).

Flujos Alternos:

A1: Conflicto de permisos → sistema alerta y solicita corrección.

1.3 Requisitos Funcionales

ID Requerimiento	Descripción	Rol	Prioridad
RF001	Gestión automática de nómina: El sistema calculará sueldos, descuentos, aportes AFP y bonificaciones automáticamente.	Administrador RRHH	Alta
RF002	Registro y actualización del kardex: Permite crear, editar y consultar historial laboral: cargos, departamentos, fechas.	Administrador RRHH	Alta
RF003	Control de vacaciones por administrador: El administrador revisa, edita y gestiona todas las solicitudes de vacaciones registradas por empleados.	Administrador RRHH	Alta
RF004	Gestión de afiliaciones médicas: Afiliar, desafiliar y actualizar datos del seguro médico del empleado (CPS).	Administrador RRHH	Media
RF005	Generación de reportes de RRHH: Generar reportes exportables (PDF/Excel) de nómina, vacaciones, afiliaciones y finiquitos.	Administrador RRHH	Media

	Gestión de finiquitos: Calcula y		
RF006	registra liquidaciones finales según antigüedad, beneficios y descuentos.	Administrador RRHH	Alta
	Administración de roles y permisos: Define perfiles de usuario y controla accesos a cada módulo del sistema.		
RF007		Administrador RRHH	Alta
	Panel central de administración:		
RF008	Acceso unificado a todos los módulos, con indicadores clave y acceso rápido.	Administrador RRHH	Alta
	Solicitud y consulta de vacaciones del empleado: Empleado puede solicitar vacaciones y consultar saldo y estado de sus solicitudes.		
RF009		Empleado	Alta
	Solicitud de actualización de datos personales: El empleado puede solicitar correcciones o cambios en su información del kardex.		
RF010		Empleado	Media
	Visualización de datos laborales centralizados: El empleado puede revisar su información: contrato, beneficios, historial laboral.		
RF011		Empleado	Media
	Notificación automática de solicitudes al gerente: Notificación		
RF012		Gerente de Área	Alta

inmediata al gerente cuando un subordinado solicita vacaciones.

Gestión de vacaciones por			
RF013	gerente: Visualiza solicitudes pendientes, aprueba/rechaza y programa reemplazos temporales si es necesario.	Gerente de Área	Alta

Consulta de información del			
equipo a cargo: Permite al gerente ver			
RF014	estado contractual, ausencias y desempeño del personal bajo su responsabilidad.	Gerente de Área	Media

Creación y publicación de			
RF015	vacantes: Publicar y modificar vacantes laborales con requisitos y fechas límite.	Reclutador	Media

Seguimiento de postulantes: Ver			
RF016	postulaciones por vacante, filtrar candidatos por etapa y observaciones.	Reclutador	Media

Gestión de entrevistas y			
RF017	evaluaciones: Programar entrevistas y registrar evaluaciones de desempeño y resultados de candidatos.	Reclutador	Media

RF018	Configuración de calendario laboral: Permite al administrador definir	Administrador RRHH	Baja
-------	---	--------------------	------

feriados, días no laborables y
configuraciones para cálculo de nómina.

1.4 Requisitos No Funcionales

ID Requerimiento	Descripción	Rol	Prioridad
RNF001	Tiempo de respuesta UI: Todas las pantallas deben cargar en ≤2 segundos.	Todos	Alta
RNF002	Soporte usuarios concurrentes: Soporta hasta 2,500 usuarios simultáneos.	Todos	Alta
RNF003	Seguridad en autenticación: Autenticación segura; contraseñas hasheadas; cifrado TLS en tránsito.	Todos	Alta
RNF004	Control de acceso: Roles y permisos controlados por módulo.	Todos	Alta
RNF005	Cifrado en reposo: Datos sensibles cifrados con AES-256.	Todos	Alta
RNF006	Auditoría de acciones: Registro log de operaciones sensibles (usuario, acción, fecha/hora).	Administrador	Media
RNF007	Disponibilidad: 99% uptime en horario laboral.	Todos	Media
RNF008	Compatibilidad de navegadores: Compatible con Chrome, Firefox y Edge últimas versiones.	Usuarios Front-end	Media

RNF009	Accesibilidad UI: Cumplimiento WCAG; formularios con validaciones claras.	Todos	Media
RNF010	Mantenibilidad técnica: Código modular, pruebas unitarias/integración, CI/CD.	DevOps	Alta
RNF011	Escalabilidad de arquitectura: Arquitectura preparada para escalamiento horizontal.	Arquitecto	Alta
RNF012	Backups y restauración: Backup diario automático; restauración en <1hora.	Admin TI	Media
RNF013	Portabilidad del backend: Backend deployable en Linux/Docker.	DevOps	Media
RNF014	Cumplimiento legal/normativo: Cumplimiento normativa boliviana laboral y protección de datos.	Admin TI	Alta

6. Teoría de la metodología

Desde la perspectiva de ingeniería de software, XP (eXtreme Programming) aporta un marco disciplinado, alineado con los principios del ciclo de vida iterativo y los estándares modernos de calidad de software. Para este proyecto, como el tiempo estimado es de 4 meses, aproximadamente, XP es una buena alternativa para realizar proyectos en poco tiempo. Entre las características más destacadas se encuentran:

Disciplina incremental y centrada en el usuario

XP opera mediante iteraciones cortas (1–2 semanas) que permiten entregar software funcional

de forma continua, facilitando la incorporación de retroalimentación del cliente antes de que los requisitos evolucionen o se malinterpreten

Elevación de la calidad técnica mediante TDD e integración continua

La adopción de Test-Driven Development (TDD) garantiza que cada módulo desarrollado cumpla requisitos definidos desde el inicio, promoviendo cobertura de pruebas y detección de errores temprana.

Diseño simple y refactorización constante

XP promueve la “sencillez intencionada” (no implementar más de lo necesario), y requiere refactorización continua para mantener la mantenibilidad del código, evitando la acumulación de deuda técnica.

Eficiencia mediante trabajo colaborativo y propiedad compartida

La programación en parejas y la propiedad colectiva del código fomentan distribución efectiva del conocimiento, reducen dependencias individuales y elevan la calidad gracias a revisión constante y estándares compartidos.

Retroalimentación continua y respuesta al cambio

Al incorporar al cliente durante toda la ejecución —con pruebas de aceptación frecuentes— XP permite validar alineación funcional y detectar desvíos de forma temprana, adaptándose a cambios necesarios sin impactar la calidad o el tiempo de entrega.

Fases de XP:

Planificación: Definición de historias de usuario, estimación de tareas y priorización del backlog.

Diseño: Creación de diseños simples y tarjetas CRC para cada historia.

Codificación: Desarrollo iterativo con TDD y programación en pareja.

Pruebas: Ejecución de pruebas unitarias y de aceptación para cada incremento.

Refactorización: Mejora continua del código después de cada iteración.

Lanzamiento: Entrega del software funcional a producción y feedback del cliente.

Esta combinación prescriptiva-ágil mantiene la disciplina documental de Pressman y aprovecha la flexibilidad y velocidad de XP.

7. Estimaciones

Basado en Roger Pressman (Capítulos 15 «Planificación y estimación de proyectos», 16 «Programación de proyectos» y 17 «Métricas y medición») de *Ingeniería de Software*, se aplican las siguientes prácticas: descomposición funcional, uso de métricas de productividad y refinamiento iterativo.

Técnica de Estimación Aplicada

Se emplea la **descomposición por módulos**, combinado con puntos de función aproximados para cada bloque funcional, tal como sugiere Pressman. Cada módulo —definido por casos de uso y requisitos clave— se dimensiona considerando complejidad, prioridad y factor de ajuste.

Módulo	Método de cálculo	Esfuerzo (PM)
Gestión de Personal	Descomposición + PF ($18 \text{ PF} \times 0.36 \text{ PM/PF}$)	6.5

Nómina	Descomposición + PF (16 PF × 0.375 PM/PF)	6.0
Reclutamiento	Descomposición simple (basado en UC)	4.5
Autoservicio del Empleado	PF ligero (10 PF × 0.4 PM/PF)	4.0
Total		21.0

3. Duración y Recursos (4 meses)

Para completar los 21 PM en **4 meses**, se distribuirá el esfuerzo de la siguiente manera, asumiendo un equipo de **2 desarrolladores y 1 ingeniero de pruebas**:

Mes	Personas disponibles	PM planificados	Hitos clave
1	2 Dev + 1 QA	5.25	Análisis y diseño de Kardex
2	2 Dev + 1 QA	5.25	Codificación y pruebas unitarias de Kardex
3	2 Dev + 1 QA	5.25	Análisis, diseño y codificación de Nómina

Mes	Personas disponibles	PM planificados	Hitos clave
4	2 Dev + 1 QA	5.25	Pruebas de integración y sistema; refinamiento de Autoservicio y Reclutamiento

4. Distribución del Esfuerzo (40-20-40)

Según la regla de Pressman (Cap.16), el esfuerzo se reparte en las actividades: análisis/diseño (40%), codificación (20%) y pruebas (40%).

- **Análisis y Diseño:** 40% → 8.4 PM
- **Codificación:** 20% → 4.2 PM
- **Pruebas:** 40% → 8.4 PM

4. Recursos Requeridos

Con base en Pressman (Cap.15), se definen roles críticos y herramientas para asegurar planificación y control:

- **Equipo Humano:**
 - Líder de Proyecto (seguimiento y coordinación)
 - 2 Desarrolladores Full-Stack (programación y revisión)
 - 1 Ingeniero de Pruebas (automatización y métricas de defectos)
 - 1 Administrador de Base de Datos (optimización y rendimiento)
- **Entorno de Desarrollo:**

- Repositorio Git con CI/CD (GitLab/GitHub)
- Servidores Linux para dev/pruebas/producción
- Herramientas UML y Jira (casos de uso, trazabilidad)
- Framework de pruebas automatizadas (JUnit/Selenium)

5. Principio de Actualización

Tal como recomienda Pressman (Cap.17), las estimaciones se revisan tras cada hito y se ajustan usando métricas reales de avance (productividad, defectos, horas consumidas), garantizando precisión y control continuo. **Técnica de Estimación**

Aplicada

Se emplea la **descomposición por módulos** según requisitos y casos de uso. Los módulos y sus esfuerzos son:

Módulo	Esfuerzo (PM)
Gestión de Personal	6.5
Nómina	6.0
Reclutamiento	4.5
Autoservicio del Empleado	4.0
Total	21.0

3. Distribución del Esfuerzo (40-20-40)

- **Análisis y Diseño (40%):** 8.4 PM
- **Codificación (20%):** 4.2 PM

- **Pruebas (40%):** 8.4 PM

4. Recursos Requeridos

- **Equipo Humano:**

- Líder de Proyecto
- 2 Analistas de Requisitos y Diseño
- 3 Desarrolladores Full-Stack
- 2 Ingenieros de Pruebas
- Administrador de Base de Datos

- **Entorno de Desarrollo:**

- Repositorio Git con CI/CD
- Servidores Linux para desarrollo, pruebas y producción
- Herramientas UML y gestión ágil (Jira)
- Framework de pruebas automatizadas

5. Principio de Actualización

Las estimaciones se refinan iterativamente tras cada hito y al completar iteraciones, incorporando métricas reales y feedback de usuarios para mantener precisión.

6. Plan de Aseguramiento de la Calidad del Software (ACS)

Propósito del Plan de ACS

El objetivo del Plan de ACS para el proyecto "Sistema de información para la gestión de procesos administrativos de RRHH en YPFB-Andina" es garantizar que todas las actividades de desarrollo, pruebas y despliegue cumplan con los estándares de calidad

definidos, minimizando defectos y asegurando la conformidad con los requisitos funcionales y no funcionales.

8. Actividades específicas para el ACS

Revisión de especificaciones y requisitos: El equipo de ACS validará que los RF y RNF registrados (p. ej., RF001-RF018, RNF001-RNF014) estén completos, claros y trazables. Se realizarán revisiones quincenales usando checklists de trazabilidad basados en UML y casos de uso.

Verificación de diseño y arquitectura: Se programarán sesiones de revisión de diagramas UML (casos de uso, clases y despliegue) para asegurar que la arquitectura propuesta satisface la separación de responsabilidades y escalabilidad. El equipo ACS marcará incongruencias en Jira con prioridad alta.

Auditoría de estándares de codificación: Semanalmente, mediante herramientas de análisis estático (SonarQube), se comprobará que el código fuente cumple con convenciones y métricas de calidad (complejidad ciclomática, cobertura de pruebas). Se generará un reporte de métricas en cada sprint.

Inspección de casos de prueba: El SQA Manager revisará los planes de pruebas unitarias, integración y sistema, asegurando que cubren todos los casos de uso (UC01–UC07) y los requisitos de rendimiento y seguridad. Las inspecciones se documentarán en plantillas de revisión de pruebas.

Monitorización de métricas de calidad: Se implementarán dashboards en Jenkins/Grafana para hacer seguimiento de métricas clave: defectos por módulo, tiempo medio de corrección, cobertura de pruebas y tiempo de respuesta de la UI. Estas métricas alimentarán reuniones de métricas semanales.

Registro y seguimiento de no conformidades: Cualquier desviación del proceso, defecto o incumplimiento se documentará en Jira, asignando responsable, severidad y fecha de resolución. El SQA Manager hará reportes quincenales de estado de no conformidades.

Auditoría de la gestión de configuración: Se verificará el uso correcto de ramas Git, etiquetado de versiones e implementación de CI/CD en Jenkins. El equipo ACS validará que cada build pase pruebas y análisis estático antes de su despliegue.

Revisión de entregas y despliegues: Antes de cada release a producción, el ACS llevará a cabo una auditoría de checklist de despliegue, incluyendo validación de backups, scripts de migración y evidencia de pruebas de aceptación.

Capacitación y concienciación en calidad: Se llevarán a cabo talleres al inicio de cada sprint para reforzar prácticas TDD, revisión por pares y estándares de codificación. El SQA Manager documentará asistencia y resultados en actas de reunión.

Este plan de ACS se revisará y ajustará al final de cada iteración, incorporando las lecciones aprendidas y métricas reales de calidad para garantizar la mejora continua del proceso.

9. Metas, atributos y métricas aplicadas (Cap. 16)

9.1 Metas de Calidad del Proyecto

- **Exactitud e integridad de datos:** Garantizar que los cálculos de nómina, kardex y finiquitos sean 100% precisos y estén libres de inconsistencias.
- **Disponibilidad y rendimiento óptimo:** Asegurar que el sistema soporte hasta 2 500 usuarios concurrentes con tiempos de respuesta ≤ 2 s en todas las pantallas.
- **Seguridad y cumplimiento normativo:** Proteger datos sensibles (cifrado TLS y AES-256) y cumplir con regulaciones bolivianas y CPS.

9.2 Atributos de Calidad Prioritarios

- **Fiabilidad:** Crítica para un sistema que gestiona cálculos salariales y finiquitos; fallos podrían generar sanciones o errores financieros.
- **Usabilidad:** Clave para la adopción por parte de empleados y gerentes; una interfaz clara y rápida permite minimizar errores de entrada y agilizar procesos.
- **Seguridad:** Fundamental para proteger información sensible y garantizar el cumplimiento de normativas (RNF003, RNF005, RNF014).

9.3 Métricas Aplicadas

Atributo de Calidad	Métrica Específica	Descripción de su Uso en el Proyecto
Fiabilidad	Tasa de defectos críticos	Número de defectos de alta severidad por módulo dividido entre el total de casos de prueba; objetivo <1%.
Usabilidad	Tiempo medio de tarea	Medición del tiempo promedio que tarda un usuario en completar procesos críticos (p. ej., generar nómina), meta ≤ 2 min.
Seguridad	Violaciones de acceso y cifrado	Conteo de incidentes de seguridad (accesos no autorizados o cifrado inactivo) por mes; objetivo 0 incidentes.

Estos indicadores se recolectarán mediante herramientas de monitoreo (Grafana, logs de auditoría) y se revisarán en reuniones de métricas semanales para asegurar mejora continua.

10. Factores evaluados del software (Cap. 14)

La calidad del software se mide a través de factores específicos y cuantificables durante todas las fases del ciclo de vida, desde el diseño hasta el mantenimiento. Estos factores proporcionan indicadores objetivos que guían la mejora continua y el aseguramiento de que el "Sistema de información para la gestión de procesos administrativos de RRHH en YPFB-Andina" cumple con los estándares de calidad esperados.

Factores de Calidad a Evaluar

- **Usabilidad**

- **Definición:** Grado en que el sistema es fácil de aprender, comprender y operar.
- **Importancia crítica:** Garantiza que empleados y gerentes puedan usar funciones clave (solicitar vacaciones, consultar kardex) eficientemente con mínima capacitación, aumentando la adopción y reduciendo solicitudes de soporte.
- **Método de evaluación:** Se realizarán sesiones de prueba con usuarios piloto, aplicando listas de verificación de atributos de interfaz y midiendo el tiempo promedio por tarea y tasa de éxito por escenario.

- **Fiabilidad**

- **Definición:** Capacidad del sistema para funcionar sin defectos bajo condiciones especificadas y recuperarse de fallas.
- **Importancia crítica:** Asegura cálculos exactos de nómina y finiquitos, previniendo errores que puedan provocar sanciones legales o financieras.

- **Método de evaluación:** Pruebas de resistencia y análisis de métricas como la tasa de defectos críticos por módulo y el Tiempo Medio Hasta Falla (TMHF) en condiciones de uso representativas.
- **Rendimiento**
 - **Definición:** Eficiencia con la que el sistema responde a solicitudes y utiliza recursos (CPU, memoria, red).
 - **Importancia crítica:** Esencial para procesar Nómina y reportes a 2 500 usuarios concurrentes sin demoras, garantizando continuidad operativa.
 - **Método de evaluación:** Pruebas de carga y estrés con herramientas como JMeter, midiendo tiempos de respuesta y utilización de recursos frente a los umbrales establecidos.
- **Mantenibilidad**
 - **Definición:** Facilidad con que el software puede modificarse, corregirse o ampliarse.
 - **Importancia crítica:** Fundamental para adaptarse a cambios normativos o nuevos requisitos, reduciendo costos y tiempo de desarrollo de futuras versiones.
 - **Método de evaluación:** Análisis de métricas de código (acoplamiento, cohesión y complejidad ciclomática) con SonarQube, y revisión de la integridad de la documentación técnica y gestión de configuración en Git.

11. Líneas para las revisiones técnicas (Cap. 15)

Las Revisiones Técnicas (RT) tienen como propósito identificar y corregir defectos en artefactos de ingeniería (requisitos, diseños, código) en etapas tempranas, para minimizar

retrabajos y reducir significativamente los costos asociados a fallas detectadas en fases posteriores.

Procedimiento de Revisión Técnica Formal (RTF)

1. Planificación

- El Líder del Proyecto nombra un Coordinador de Revisión y define el alcance y los artefactos a inspeccionar (p. ej., diagramas de casos de uso).
- Se programa la sesión de revisión, asignando fecha, hora y participantes, y se distribuyen los materiales con antelación.
- Se elabora y comparte una agenda que incluye objetivos, duración estimada y roles (coordinador, revisores, autor, secretario).

2. Preparación Individual

- Cada revisor recibe la documentación correspondiente y dedica 1–2 horas a examinarla detenidamente.
- Durante la inspección individual, se anotan observaciones, inconsistencias y preguntas siguiendo una lista de verificación estándar.

3. Ejecución de la Reunión

- Abre el Coordinador de Revisión, resume el objetivo y revisa la agenda.
- El Autor del artefacto presenta brevemente el contenido y contexto del material.
- Los Revisores exponen hallazgos uno a uno, mientras el Secretario documenta cada punto, sin entrar en debates extensos.

- Al concluir, se valida la lista de pendientes, se asignan responsables y se acuerdan plazos de corrección.

4. Reglas Clave durante la Reunión

- **No resolver problemas en el momento:** Registrar las incidencias para su tratamiento posterior.
- **Seguir la agenda:** Mantener el enfoque y la eficiencia del proceso.
- **Limitar el número de participantes a 4–6 personas:** Garantizar dinamismo y evitar redundancias.
- **Registrar todo por escrito:** El Secretario utiliza herramientas colaborativas (Jira/Confluence) para documentar hallazgos y acuerdos.
- **Fomentar el respeto mutuo:** Formular observaciones mediante preguntas que guíen al autor a reconocer posibles errores.

Lista de Verificación para Diagrama de Casos de Uso

Para garantizar la exhaustividad y coherencia del Diagrama de Casos de Uso del "Sistema de información para la gestión de procesos administrativos de RRHH en YPFB-Andina", se aplicará la siguiente checklist:

1. ¿Se describen todos los flujos principales y alternativos para cada caso de uso (p. ej., validación de saldo en "Solicitar Vacaciones")?
2. ¿Están correctamente asignados y nombrados los actores (Administrador RRHH, Empleado, Gerente de Área, Reclutador)?
3. ¿Cada caso de uso refleja una funcionalidad completa, con precondiciones y resultados esperados claramente definidos?

4. ¿El nivel de abstracción se centra en las necesidades de negocio, sin incluir detalles de implementación o infraestructura?
5. ¿Se han identificado y agregado todos los casos críticos (gestión de kardex, cálculo de nómina, generación de finiquitos, roles y permisos)?

12. Registro de las revisiones técnicas (Cap. 17)

La **documentación sistemática** de cada Revisión Técnica Formal (RTF) es esencial para garantizar la trazabilidad, el control de calidad y la gestión adecuada de los artefactos del proyecto. Este registro constituye la **evidencia oficial** de los hallazgos, decisiones y acciones correctivas, y sirve como base para auditorías, métricas de proceso y lecciones aprendidas.

12.1 Propósito del Registro

- **Trazabilidad:** Permite rastrear qué artefactos (requisitos, diseños, código) han sido revisados, por quién y cuándo.
- **Control de líneas base:** Certifica la aprobación de cada producto de trabajo antes de consolidarlo como “línea base” del proyecto.
- **Seguimiento de acciones correctivas:** Facilita el monitoreo de las tareas derivadas de los hallazgos, asegurando que se resuelvan en tiempo y forma.
- **Evidencia histórica:** Proporciona datos para evaluar la efectividad del ACS y para preparar auditorías internas o externas.

12.2 Plantilla de Informe de Revisión Técnica

Campo	Descripción
--------------	--------------------

ID de Revisión	RT-01
-----------------------	-------

Artefacto Revisado	Módulo de sueldos
---------------------------	-------------------

Fecha	26/06/2025
--------------	------------

Participantes	<ul style="list-style-type: none">• Líder de Revisión: Nombre• Autor: Kenji Tellez• Secretario: Ivan Longaric• Revisores: Gerentes de YPFB
----------------------	---

Resumen de Objetivos	Validar el diseño con los solicitantes Verificar funcionamiento del modulo
-----------------------------	---

Tabla de Hallazgos

ID	Descripción del	Ubicación	Severidad
Hallazgo	Problema	(pág./sección/línea)	(Crítica/Major/Minor/Sugerencia)
H-001	Si el empleado que se busca tiene más de un cargo, solo se muestra el primer cargo y los otros se omiten.	sueldos.ejs	Minor

ID	Descripción del	Ubicación	Severidad
Hallazgo	Problema	(pág./sección/línea)	(Crítica/Major/Minor/Sugerencia)
	El endpoint no está recibiendo el parametro mes, por lo que no se puede buscar sueldos de otros meses que no sea el actual.	App.ts	Major

Decisión Final del Equipo

☐ Aprobado

Acciones a Seguir

ID	Acción Requerida	Responsable	Fecha	Estado
Hallazgo			Límite	
H-001	Modificar la función que muestra los datos para que acepte más de un cargo	Desarrollador Frontend	05/07/2025	Pendiente
H-002	Obtener parámetro mes para hacer el select en la base de datos y devolver el sueldo de esa fecha	Desarrollador Backend	04/07/2025	En curso

- 12.3 Proceso de Archivo y Difusión

- Almacenamiento

- Cada Informe de Revisión se guarda en el repositorio de documentación (/docs/revisiones/RT-YYYYMMDD-XX.md).

2. Notificación

- Al concluir la RTF, el Secretario envía un enlace al informe en Confluence y notifica por correo a todos los participantes y al SQA Manager.

3. Seguimiento

- Las **Acciones a Seguir** se crean como tareas en Jira con vinculación al ID del hallazgo; el Test Lead monitorea el cierre de cada una.

4. Auditoría

- En la **auditoría de configuración** al cierre de cada hito, se verifica que todos los hallazgos marcados como “Pendiente” o “En curso” tengan un dueño y una fecha límite.

Con este proceso y plantilla, el equipo de “**Sistema de información para la gestión de procesos administrativos de RRHH en YPFB-Andina**” contará con un registro riguroso y accionable de todas sus Revisiones Técnicas, asegurando la calidad y la continuidad del proyecto.

13. Justificación de las estrategias de prueba (Cap. 22)

La adopción de una estrategia de pruebas sistemática que avance “**de lo pequeño a lo grande**” es esencial para el **Sistema de información para la gestión de procesos administrativos de RRHH en YPFB-Andina**. Al comenzar validando los componentes más básicos y progresar gradualmente hacia entornos cada vez más integrados, garantizamos la detección temprana de defectos, evitamos que los errores se propaguen, reducimos los costos

de corrección en fases posteriores y reforzamos la confianza de los usuarios en la calidad del software.

13.1 Niveles de Prueba a Aplicar

1. Prueba de Unidad

- **Objetivo principal:** Verificar que cada componente o función individual opere correctamente de forma aislada.
- **Ejemplo en el proyecto:** Se diseñarán y ejecutarán pruebas unitarias para la función de **cálculo automático de nómina**, asegurando que los algoritmos de sueldos, descuentos, aportes AFP y bonificaciones devuelvan resultados exactos antes de pasar a la integración.

2. Prueba de Integración

- **Objetivo principal:** Detectar fallos en las interfaces y en la interacción entre módulos probados unitariamente.
- **Ejemplo en el proyecto:** Se probará la conexión entre el módulo de **Reclutamiento** y el de **Gestión de Personal**, validando que los datos de un nuevo contratado —como su kardex y afiliación al seguro médico— se transfieran y almacenen correctamente.

3. Prueba de Validación

- **Objetivo principal:** Confirmar que el software cumple con los requisitos funcionales y no funcionales definidos por el cliente y el usuario.
- **Ejemplo en el proyecto:** A partir de las historias de usuario, se elaborarán casos de prueba para verificar que un **Gerente de Área** pueda aprobar un lote

de solicitudes de vacaciones en **menos de 5 clics**, y que los **reportes automatizados** de pagos y afiliaciones coincidan con los formatos y datos estipulados.

4. Prueba del Sistema

- **Objetivo principal:** Evaluar el sistema completo en un entorno de producción simulado, incluyendo funcionalidades, rendimiento, seguridad y despliegue.
- **Ejemplo en el proyecto:**
 - **Seguridad:** Verificar que solo los **Administradores de RRHH** autorizados modifiquen información sensible (nómina, datos personales).
 - **Rendimiento:** Medir el tiempo de procesamiento de nóminas para **2 500 empleados**, garantizando que los cálculos y la generación de reportes se completen dentro de los umbrales definidos.
 - **Despliegue/Configuración:** Probar la instalación y operación del sistema en desarrollo, prueba y producción, asegurando que la configuración de bases de datos y servidores cumpla con los requisitos de disponibilidad y escalabilidad.

14. Cómo se administró la configuración

La Gestión de la Configuración del Software (GCS) para el proyecto “**Desarrollo de un Sistema de Recursos Humanos para la Optimización de Procesos en YPFB-Andina**” tiene como propósito garantizar que todos los artefactos—desde requisitos y modelos hasta código fuente y pruebas—sean controlados, versionados y trazables a lo largo de todo su ciclo de vida.

Una GCS bien implementada evita el caos derivado de cambios no documentados, asegura la integridad de cada entrega y facilita la recuperación ante errores o regresiones.

Objetivos clave de la GCS:

1. **Trazabilidad completa:** Asegurar que cada requisito aprobado pueda rastrearse hasta su implementación en el código y las pruebas asociadas.
2. **Control de versiones uniforme:** Mantener un histórico claro de todas las versiones de documentos, modelos UML, código fuente y artefactos de prueba.
3. **Gestión formal de cambios:** Establecer un flujo de trabajo para solicitudes, evaluaciones y aprobaciones de cambios que minimice riesgos y evite impactos inesperados.

Identificación de Ítems de Configuración (ICs)

Un Ítem de Configuración (IC) es cualquier artefacto del proyecto cuyo control de versión, revisión y trazabilidad es esencial. A continuación, se listan los principales ICs agrupados por categoría:

Categoría	Ítem de Configuración (IC) Específico
Documentación	Documento de Especificación de Requisitos, Plan de Proyecto, Manual de Usuario
Modelado (UML)	Diagrama de Casos de Uso, Diagrama de Clases, Diagrama de Secuencia
Código Fuente	Módulo de Nómina (código), Módulo de Autoservicio del Empleado (código), Repositorio CI/CD

Categoría	Ítem de Configuración (IC) Específico
Pruebas	Plan General de Pruebas, Casos de Prueba de Aceptación, Reportes de Pruebas

2 Control de Versiones

- **Herramienta:** Se empleará el sistema de control de versiones distribuido **Git**, alojado en un servidor privado de la organización (GitLab/GitHub Enterprise).
- **Esquema de versionado:** Se seguirá **Versionado Semántico** (SemVer) con el formato **MAYOR.MENOR.PARCHE** para todos los ICs principales:
 - **MAYOR:** Cambios incompatibles en la API o despliegue.
 - **MENOR:** Nuevas funcionalidades sin romper compatibilidad.
 - **PARCHE:** Corrección de defectos o documentación.
- **Ramas principales:**
 - main (o master): Contiene sólo versiones liberadas y aprobadas.
 - develop: Integra las nuevas funciones en desarrollo.
 - Ramas de característica (feature/...), corrección (hotfix/...) y despliegue (release/...) según flujo GitFlow.

Proceso de Control de Cambios

Para cualquier actualización de un IC se seguirá este flujo formal:

1. Solicitud de Cambio

- **Quién la inicia:** Cualquier miembro del equipo (desarrollador, analista, QA) o un representante del cliente.
- **Cómo:** Mediante un **Formulario de Solicitud de Cambio (FSC)** en la herramienta de gestión (Jira/Trello), detallando descripción, motivación, prioridad y posible impacto.

2. Evaluación del Cambio

- **Responsable:** El **Comité de Control de Cambios (CCC)**, compuesto por el líder de proyecto, un desarrollador senior y un representante del cliente/usuario final.
- **Tarea:** Analizar impacto técnico, esfuerzo estimado, riesgos y beneficios. Documentar conclusiones en una minuta de evaluación.

3. Aprobación / Rechazo

- El **CCC** emite una decisión formal:
 - **Aprobado:** Se planifica la implementación.
 - **Aprobado con Condiciones:** Requiere modificaciones al plan o redefinición de alcance.
 - **Rechazado:** Motivos documentados y notificación al solicitante.

4. Implementación y Verificación

- **Asignación:** El líder de desarrollo asigna la tarea a un responsable.
- **Implementación:** Se crea una rama específica (feature/ o hotfix/), se realiza el cambio y se generan pruebas unitarias/integración.

- **Pruebas de Regresión:** El equipo de QA ejecuta casos de regresión para asegurar que el cambio no introdujo errores nuevos.
- **Merge y Liberación:** Tras la aprobación en **Merge Request**, se fusiona a develop y luego, tras las pruebas finales, a main.

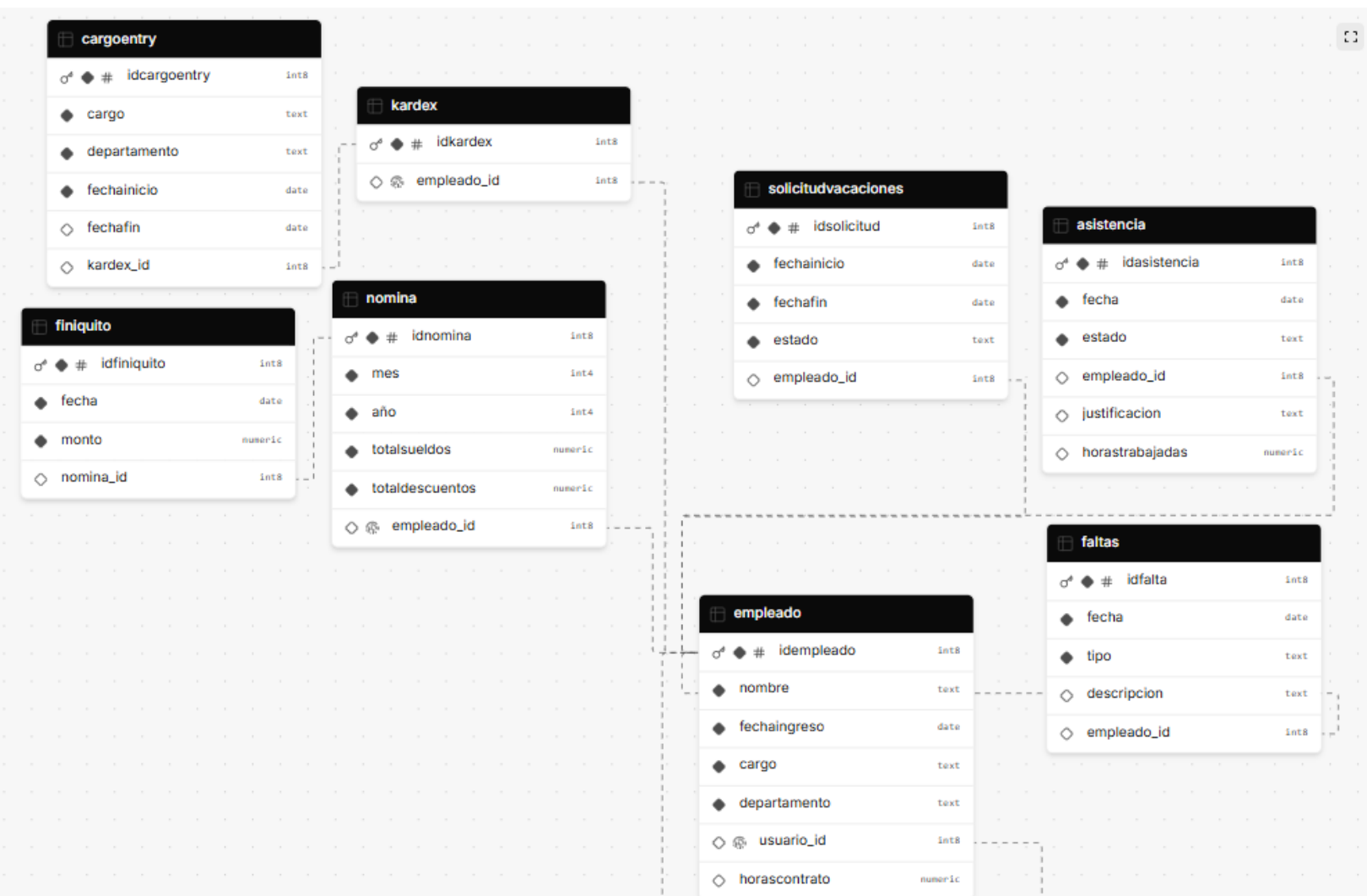
Auditoría y Reporte de Estado de la Configuración

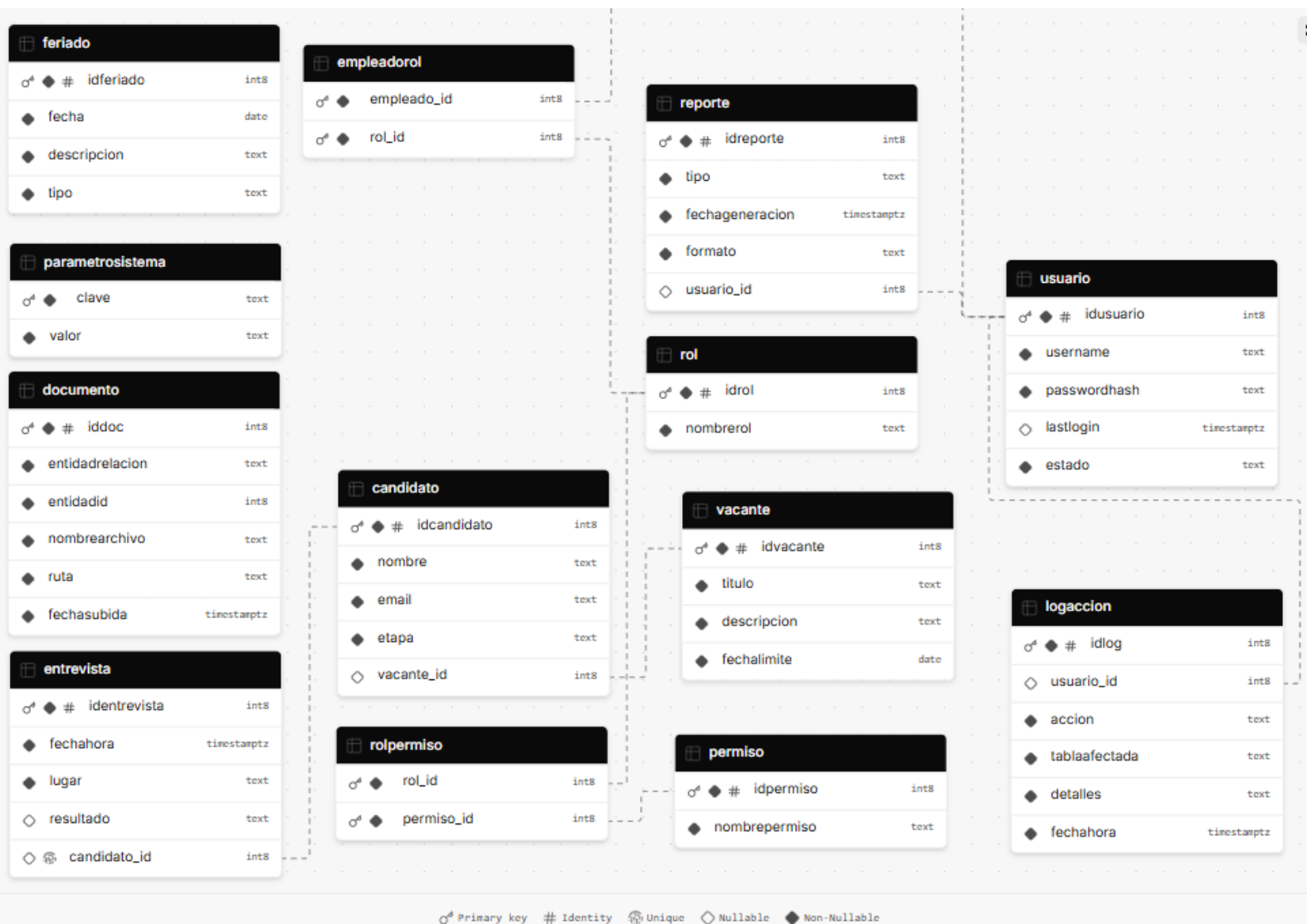
Al finalizar cada hito (por ejemplo, diseño completo, módulo implementado, pruebas de aceptación), se realizarán **auditorías formales de configuración**. Estas consisten en:

- **Revisión de Ramas y Tags:** Verificar que las ramas y etiquetas (tags) correspondan con las versiones aprobadas.
- **Concordancia de Artefactos:** Confirmar que la documentación, modelos UML, código y planes de prueba estén sincronizados y versionados según SemVer.
- **Informe de Auditoría:** Documento breve que recoja hallazgos, desviaciones y acciones correctivas, distribuido al equipo de proyecto y al SQA Manager.

Este proceso de auditoría garantiza que **la configuración implementada coincide con la documentación aprobada** y que todas las solicitudes de cambio han seguido el procedimiento, reforzando la integridad y calidad del producto final.

15. Diagramas de clases





- 16. Mapeo**
- 17. Diseño de reportes parametrizados**
- 18. Despliegue de datos**
- 19. Registro de las pruebas**
- 20. Presupuesto**
- 21. Conclusiones, recomendaciones y bibliografía**