

SEMINARIO DE PRÁCTICA

Licenciatura en Informática
Desarrollo de un Sistema de Gestión de
Ausentismo Hospitalario
Solución Especializada para la Administración novedades
Hospital de Urgencias de Córdoba

Profesor: Pablo Alejandro Virgolini
Alumno: Pablo Alejandro Hamann
Legajo: VINF010782
Año: 2024

Tabla de contenido

1.	Introducción	1
1.1	Antecedentes.....	2
1.2	Descripción del área problemática	3
1.3	Formulación de la problemática	3
2.	Justificación.....	3
3.	Definiciones del proyecto	5
3.1	Objetivo general del proyecto	5
3.2	Objetivos específicos del proyecto	5
4.	Definiciones del sistema.....	6
4.1	Objetivo general del sistema	6
4.2	Límites.....	6
4.3	Alcances	7
4.4	No Contempla.....	7
5.	Elicitación	7
5.1	Actividad del cliente o dominio	8
5.2	T.I.C (Tecnologías de la Información y Comunicación).....	8
5.3	Relevamiento estructural y tecnológico	8
5.4	Relevamiento funcional.....	9
	Organigrama.....	9
	Funciones de las áreas involucradas	9
	Diagrama de Dominio.....	9
	Procesos de negocio.....	10
6.	Diagnóstico.....	19
	Diagnóstico 1: ABM (Alta, Baja, Modificación) de empleados.....	19
	Diagnóstico 2: Planificación de los Diagramas de Servicio y Asignación de RRHH	19
	Diagnóstico 3: Registro de novedades en el parte diario de personal	20
	Diagnóstico 4: Generación de reportes personalizados	20
	Diagnóstico 5: Monitoreo de la disponibilidad del personal	20
7.	Propuesta de solución.....	21
7.1	Propuesta funcional.....	21
7.2	Propuesta técnica	22
7.3	Propuesta de despliegue	23
7.4	Diagrama de Arquitectura	23
	Diagrama de componentes:	24

Diagrama de Despliegue:	24
7.5 Listado de requerimientos funcionales en matriz de trazabilidad	25
7.6 Listado de requerimientos no funcionales	25
7.7 Listado de requerimientos candidatos	26
Listado de requerimientos candidatos Funcionales.....	26
Listado de requerimientos candidatos No Funcionales	27
8. Desarrollo del prototipo.....	27
8.1 Diagrama de casos de uso	27
Diagrama de CU General: Sistema de Gestión de RRHH.....	27
Diagrama de CU01: Gestionar Altas, Bajas y Modificaciones de Empleados.....	30
Diagrama de CU02: Generar Diagrama de Servicio Personalizado	31
Diagrama de CU03: Consultar Disponibilidad del Personal en Tiempo Real	32
Diagrama de CU04: Registrar Horas Extras y Calcular Francos Compensatorios.....	32
Diagrama de CU05: Registrar Pases de Salida y Calcular Horas Pendientes de Devolución.....	33
Diagrama de CU06: Registrar y Ajustar Omisiones de Marcación en el Parte Diario	33
Diagrama de CU07: Enviar Memorándums.....	34
Diagrama de CU08: Generar Reportes Personalizados.....	34
Diagrama de CU09: Enviar Alertas Automáticas	35
Diagrama de CU10: Controlar Acceso según el Rol del Usuario	35
8.2 Descripción de casos de uso	35
Descripción de CU01: Gestionar Altas, Bajas y Modificaciones de Empleados	35
Descripción de CU02: Generar Diagrama de Servicio Personalizado.....	37
Descripción de CU03: Consultar Disponibilidad del Personal en Tiempo Real	38
Descripción de CU04: Registrar Horas Extras.....	39
Descripción de CU05: Registrar Pases de Salida y Calcular Horas Pendientes de Devolución .	40
Descripción de CU06: Registrar y Ajustar Omisiones de Marcación en el Parte Diario	41
Descripción de CU07: Enviar Memorándums	43
Descripción de CU08: Generar Reportes Personalizados	44
Descripción de CU09: Enviar Alertas Automáticas.....	45
Descripción de CU10: Controlar Acceso según el Rol del Usuario	46
Descripción de CU11: Gestión de Francos Compensatorios	48
9. Etapa de análisis.....	49
Diagramas de secuencia.....	49
Diagrama de secuencia para CU01.....	50
Diagrama de secuencia para CU02.....	52

Diagrama de secuencia para CU04.....	54
Diagrama de secuencia para CU08.....	56
Diagrama de secuencia para CU10.....	58
10. Etapa de diseño.....	59
Diagrama de clases de diseño	59
11. Etapa de implementación	62
Diagrama de Despliegue	62
12. Etapa de pruebas	62
Plan de pruebas.....	62
Objetivos	62
Alcance	63
Equipo	63
Estrategia.....	63
Criterios	63
Ambientes	63
Entregables.....	64
Gestión de incidentes.....	64
Riesgos.....	64
Cuadro de riesgos	64
Procedimiento de prueba	68
Preparar los Datos de Prueba	68
Asignación de testers y acceso.....	68
Generar los Casos de Prueba	68
Responsables de las pruebas	69
Planificación y Calendario de Pruebas	69
Automatización de pruebas	70
Casos de Prueba	71
Matriz de Trazabilidad de Casos de Pruebas vs. Requerimientos (Func. y No Func.).....	71
Tabla de CP01: Autenticación en el inicio de sesión	72
Tabla de CP02: Gestión de las asignaciones de empleados a servicios	73
Tabla de CP03: Generación de diagramas de servicio	75
Tabla de CP04: Carga de marcaciones	77
Tabla de CP05: Generación de Solicitudes de empleados: Horas extra / Francos compensatorios / Faltas justificadas.....	78
Tabla de CP06: Validación y aprobación de solicitudes de empleados: Horas extra / Francos compensatorios / Faltas justificadas.....	80

Tabla de CP07: Simulación de sobrecarga: Generación de múltiples solicitudes en paralelo..	82
Informe de pruebas realizadas.....	83
Tabla de Resultados de pruebas	83
Tabla de Porcentaje de pruebas.....	84
Procedimiento de Corrección	84
13. Interfaz gráfica	85
Pantalla de inicio de sesión	85
Pantalla de Menú Principal	85
Pantallas del módulo de Diagramas de Servicios.....	86
Pantalla 1 de 3.....	87
Pantalla 2 de 3.....	88
Pantalla 3 de 3.....	89
14. Definición de base de datos para el prototipo.....	90
15. Diagrama entidad-relación.....	91
16. Creación de tablas.....	93
Definición de funciones personalizadas.....	93
Función para convertir (y almacenar) un UUID en un binario de 16 bytes.....	93
Función para recuperar y convertir de nuevo a UUID	94
Tabla HorarioBase	94
Tabla Horario.....	94
Tabla HorarioConFranquicia.....	95
Tabla HorarioEstandar	95
Tabla HorarioSemanal.....	95
Tabla HorarioNocturno	96
Tabla HorarioFeriante	96
Tabla HorarioDXI	96
Tabla HorarioGuardiaMedica.....	96
Tabla HorarioGuardiaEnfermeria.....	97
Tabla HorarioJefeServicioGuardiaPasiva.....	97
Tabla HorarioAbierto.....	97
Tabla Cargo.....	97
Tabla Domicilio.....	98
Tabla Usuario	98
Tabla Direccion.....	99
Tabla Servicio	99

Tabla JefaturaDeServicio.....	99
Tabla OficinaDePersonal	99
Tabla Empleado.....	100
Trigger BEFORE INSERT	100
Trigger BEFORE UPDATE.....	101
Tabla Roles	102
Tabla Usuario_Rol	102
Tabla Autorizacion.....	103
Tabla EstadoTramite	103
Inicialización de Tabla EstadoTramite	103
Tabla Novedad	103
Tabla Intermedia Empleado_Novedad	104
Tabla JornadaLaboral	104
Tabla DiagramaDeServicio	104
Tabla Planificacion.....	105
Tabla Memorandum.....	105
Tabla Memorandum_Destinatario.....	105
Tabla Memorandum_Firmante	105
Tabla Memorandum_Autorizacion	106
Tabla FrancoCompensatorio	106
Tabla HoraExtra.....	106
Tabla ParteDiario.....	107
Tabla Intermedia ParteDiario_Empleado.....	107
Tabla MarcacionEmpleado.....	107
Tabla RegistroJornadaLaboral.....	108
Implementación de índices en Claves Foráneas (FK).....	108
17. Inserción, consulta y borrado de registros.....	109
Insertar Domicilios	110
Uso de variables	110
Insertar usuarios	113
Insertar los usuarios en las respectivas tablas de especialización de usuarios	116
Insertar Servicios.....	116
Asignar Jefaturas de Servicio.....	118
Asignar Oficina De Personal	119
Asignar Empleados	119

Desasignar Empleados	119
Insertar usuarios en Empleado	119
Listar usuarios Empleados.....	120
Salida de la consulta.....	121
Listar Directores cargados en la BD.....	121
Salida de la consulta:.....	122
Listar Servicios con sus Jefes de Servicio y Directores del cual dependen	122
Salida de la consulta.....	123
Contabilizar cantidad de empleados por servicio incluyendo jefes.....	123
Agregar/modificar un dato a un usuario.....	124
Borrado de un usuario	125
18. Presentación de las consultas SQL.....	126
19. Definiciones de comunicación	127
Diagrama	127
20. Explicación del desarrollo en Java.....	127
Alcance del Prototipo.....	128
Estructura de Desarrollo	128
Estructura de Paquetes	129
Estructura de ramas en Github	130
Software de desarrollo utilizado	130
Versión de Java.....	130
Desarrollo de la Interfaz Gráfica	130
Entorno de desarrollo integrado (IDE)	131
Gestión y Construcción del Proyecto	131
Otras herramientas	131
Clase de inicio principal.....	132
Otras consideraciones.....	132
Cifrado de contraseñas	132
Consideraciones de diseño.....	132
Archivo de configuración del programa.....	133
Logo del programa	133
Icono del programa	133
21. Referencias al Código y Documentación del Proyecto	134
Código Fuente del Proyecto	134
Documentación JavaDoc	134

22.	Video de Presentación del Proyecto	135
23.	Referencias.....	136
	Anexo I: Diagrama de Gantt	137
	Anexo II: Organigrama del Hospital Municipal de Urgencias (HMU).....	139

1. Introducción

El **Hospital de Urgencias de Córdoba** es una institución pública que depende del municipio de la ciudad de Córdoba. Desde el punto de vista administrativo, el hospital es una Dirección que depende de la **Secretaría de Salud del municipio**. En la municipalidad de Córdoba, la **Subsecretaría de Capital Humano** se encarga de gestionar diversos aspectos relacionados con todos los empleados municipales, independientemente de su tipo de contrato (empleados contratados, de planta permanente o de planta política) o área de trabajo. Entre las responsabilidades de esta subsecretaría se incluyen:

- **Liquidación de sueldos**
- **Gestión de licencias** de cualquier tipo (ordinarias, sanitarias, por fallecimiento, etc.)
- **Manejo de carpetas médicas**

Los empleados gestionan sus trámites a través de una plataforma web integrada al portal **CIDI (Ciudadano Digital)** de la provincia de Córdoba. Sin embargo, ciertos aspectos específicos a cada institución, como la modalidad de trabajo y los horarios, se gestionan internamente en cada entidad.

En particular, el **Hospital de Urgencias de Córdoba** presenta características singulares debido a su funcionamiento continuo las 24 horas del día, los 365 días del año. Las modalidades de trabajo en el hospital incluyen:

1. **Horario estándar:** 35 Hs. semanales, distribuidas en 7 horas por día, en días hábiles.
2. **Horario semanal:** 35 Hs. semanales, distribuidas de forma irregular durante cada día, pero con regularidad a lo largo de las semanas (ejemplo: lunes a viernes: 6 Hs por día, sábados 10 Hs).
3. **Horario nocturno:** 140 Hs al mes, distribuidas en 14 jornadas de 10 Hs. al mes, en turno nocturno (usualmente de 21:00 a 07:00 de la mañana del día siguiente).
4. **Guardias de feriante:** 120 Hs (mínimas) al mes, distribuidas en 10 guardias de 12 Hs en días no laborables; si no hay suficientes días no laborables en el mes, se completan las horas faltantes según diagramación de servicio. No hay máximo definido (ejemplo: si en el mes se dan 12 feriados, se deberán cumplimentar 144 Hs). De forma excepcional, puede darse que un feriante complete 2 guardias continuas (sumando 24 Hs), por necesidad de servicio, por ejemplo.
5. **Guardias de enfermería:** 140 Hs al mes, distribuidas en 10 guardias de 12 Hs, más 2 guardias de 10 Hs.
6. **Guardias médicas:** 36 Hs semanales, usualmente distribuidas en 3 guardias de 12 Hs, pudiendo ser hasta 2 guardias continuas, y pasar no menos de 12 Hs hasta la próxima guardia.
7. **Diagnóstico por imágenes:** 24 Hs semanales (legislación especial por exposición a radiación), distribuidas según diagramación de servicio.
8. **Jefes de servicio con guardia pasiva:** cuando aplican al sistema, trabajan una hora menos por día, porque queda a disponibilidad de guardia pasiva (que puede ser completada o no, según lo requiera el hospital).
9. **Horario abierto:** 35 Hs. semanales, distribuidas en cualquier horario. El HMU no cuenta actualmente con empleados activos bajo este régimen, pero se prevé en sistema.
10. **En comisión:** el empleado no registra marcaciones. Son casos especiales cuando un empleado trabaja en varias reparticiones y debe trasladarse constantemente entre ellas. El HMU no cuenta actualmente con empleados activos bajo este régimen y tampoco se prevé que vaya a haber en el corto plazo. Dado que incrementaría la complejidad del

modelo de diseño, si en algún momento se requiriera, se asignará un horario estándar y se utilizará un código de licencia especial en su lugar.

11. **Franquicias horarias:** se trata de una reducción horaria (casi siempre de 2 Hs.) a la jornada laboral para ciertos empleados que cuentan con dicha franquicia. Las franquicias horarias se asignan por períodos de tiempo determinados (aunque la mayoría de las veces son renovadas).
12. **Condicionalidad general:** En cada caso, no pueden transcurrir menos de 10 Hs corridas entre un egreso y un ingreso.

Además, la estructura organizativa del hospital incluye directores, jefes y subjefes que deben autorizar ciertas solicitudes. La oficina de personal del hospital es responsable de registrar todas las novedades de ausentismo y reportarlas semanal y mensualmente a la Subsecretaría de Capital Humano, que a su vez procesa esta información en su área de ausentismo.

Con el objetivo de optimizar el proceso de gestión de ausentismo en el Hospital de Urgencias de la ciudad de Córdoba, se desarrollará un proyecto que implementará un sistema especializado para la administración de horarios, cambios de guardia y control de inasistencias. Este sistema busca mejorar la eficiencia y la precisión en el manejo de la información relacionada con el personal de salud, permitiendo una integración más efectiva con la Subsecretaría de Capital Humano del municipio de la ciudad y reduciendo la dependencia de tediosos, inseguros y complejos procesos manuales actualmente en uso.

1.1 Antecedentes

En antecedentes se hace una reseña histórica, descripción en el tiempo de la problemática y/o los avances o trabajos previos que se hayan realizado sobre el tema.

Originalmente, en cada institución municipal de la ciudad de Córdoba, la respectiva área de RRHH (a la que se denomina “Oficina de Personal”), se encargaba de gestionar todos los procesos de ausentismo y horarios mediante **métodos manuales y planillas de cálculo**, con un impacto significativo en la eficiencia y precisión de los datos.

Durante los últimos años, se ha implementado un proceso de informatización y autogestión en el municipio, lo que ha afectado a todas las oficinas de personal de las direcciones y áreas municipales, incluido el Hospital de Urgencias.

No obstante, las tareas específicas de la oficina de Personal en el Hospital de Urgencias aún se realizan de manera “artesanal” debido a la falta de un sistema especializado que contemple las particularidades del entorno hospitalario. A pesar de los esfuerzos realizados a lo largo de los años para integrar estos procesos en el marco de las mejoras generales implementadas en el municipio, los avances han sido insuficientes para cubrir las necesidades específicas del hospital. De hecho, en ocasiones, estos avances han generado nuevos problemas precisamente por no haber considerado adecuadamente dichas necesidades particulares.

Estos problemas surgen principalmente por la forma en que se abordan los proyectos de gestión de RRHH en el municipio. De los 8,793 empleados municipales, la gran mayoría trabaja en jornadas laborales de 6 horas diarias y en horarios fijos bien establecidos. Solo un poco más del 5% de estos empleados pertenecen al Hospital de Urgencias, que cuenta con 478 empleados formales y diversas modalidades de trabajo, como se ha descrito anteriormente. Al intentar adaptar los procesos de gestión de RRHH diseñados para la mayoría de los empleados del municipio a las

complejas necesidades del hospital, los sistemas implementados han demostrado ser inadecuados y han fallado en abordar efectivamente los desafíos específicos del entorno hospitalario.

1.2 Descripción del área problemática

La registración manual de **novedades de ausentismo** en el Hospital de Urgencias de Córdoba **afecta significativamente** la eficiencia operativa. Las tareas incluyen:

- **Gestión de los diagramas de servicios**
- **Gestión de cambios de horarios y cambios de guardia**
- **Control de inasistencias, y tipo de inasistencia (faltas justificadas, injustificadas, por fuerza mayor, etc.)**
- **Administración de francos compensatorios**
- **Gestión de pases de salida oficiales, y particulares**

Estas tareas se realizan con **extensas planillas de cálculo**, que frecuentemente se dañan, resultando en pérdida de datos y retrasos en la presentación de informes a la Subsecretaría de Capital Humano.

La problemática afecta a **todas las áreas involucradas** en la gestión del personal hospitalario y tiene **consecuencias trascendentes**, impactando la precisión de la información y la eficiencia en la gestión de recursos humanos, y en el peor de los casos, hasta una incorrecta liquidación de sueldo. Además, el servicio de Informática del hospital no dispone de un área de desarrollo propio. Las mejoras implementadas hasta la fecha se han centrado principalmente en la **infraestructura**, como la actualización de computadoras, redes rápidas y seguras, un sistema de archivos en la nube y un servidor de documentos para la edición simultánea y respaldo de información. Sin embargo, estos avances no abordan el problema principal: la necesidad de un **sistema de gestión de RRHH** que se ajuste a las especificidades del entorno hospitalario.

1.3 Formulación de la problemática

¿Cómo puede el Hospital de Urgencias de Córdoba optimizar la gestión de RRHH? Se propone implementar un sistema que permita automatizar las gestiones de ausentismo, cambios de horarios, guardias, francos compensatorios y demás novedades de ausentismo, propias del entorno hospitalario, con el objetivo de reducir la dependencia de procesos manuales y mejorar la eficiencia y precisión administrativa.

2. Justificación

La gestión de los recursos humanos (o del capital humano) se lleva a cabo mediante la oficina de personal del propio nosocomio, y de la Subsecretaría de Capital Humano del municipio de la ciudad de la cual administrativamente depende el propio hospital. Mientras que la parte a cargo del municipio se ha modernizado, la parte que depende del nosocomio no lo ha hecho, y esto provoca múltiples problemas. En este sentido, la adopción de un sistema de sistema informático adaptado a las necesidades específicas de la institución, brindará los siguientes beneficios:

1. Mejorará la eficiencia reduciendo tiempos y carga administrativa, permitiendo que el personal dedique su tiempo a tareas más estratégicas.
2. Reducirá errores, al eliminar la dependencia de planillas de cálculo y registros manuales, mejorando la precisión de los registros.
3. Brindará mayor control y trazabilidad, facilitando datos cruciales (accesibles en tiempo real), tales como faltas justificadas o injustificadas, cambios de guardias, entre otros.
4. Optimizará la toma de decisiones, ya que, si los datos se vuelven precisos, la dirección hospitalaria, y los jefes de servicios, podrán tomar decisiones basadas en información

confiable, concreta y precisa, mejorando la asignación del personal y asegurando que se cumplan los requisitos operativos del hospital las 24 horas del día.

5. Aportará al cumplimiento normativo y transparencia, ayudando a cumplir con las regulaciones laborales y de gestión pública, aumentando la transparencia en la administración de recursos humanos y permitiendo auditorías más efectivas.

El desarrollo de un Sistema de Gestión de Ausentismo, específico a las necesidades del Hospital de Urgencias de Córdoba, proporcionaría beneficios en diferentes aspectos clave en la operación diaria y administrativa del nosocomio, entre las que podemos enumerar:

- 1. Reducción de errores administrativos**

Al digitalizar la gestión de novedades de empleados (entendemos por esto lo que refiere a faltas, cambios de horarios y/o de guardia, francos compensatorios, pases de salida, entre otros), se eliminan errores humanos propios de un trabajo manual, como lo suelen ser la pérdida de datos o errores de carga de datos. Esto tendrá un **impacto directo en la precisión de los datos**, lo que también influirá en una posterior liquidación de sueldos correcta (desde el interés del propio empleado) y en contar con información confiable para la toma de decisiones (desde el interés del empleador).

- 2. Ahorro de tiempo**

Los empleados administrativos de la oficina de Personal del hospital, pasarán de gestionar planillas manuales y procesos repetitivos, a utilizar un sistema automatizado que reducirá drásticamente el tiempo invertido en estas tareas. Esto permitirá dedicar tiempo a funciones de mayor valor, como la planificación estratégica de los recursos humanos.

- 3. Mejora en la planificación y asignación de personal**

Con un sistema informatizado, la dirección hospitalaria, podrá contar con datos precisos, confiables y actualizados sobre la disponibilidad del personal, lo que por supuesto, facilitará la coordinación con los jefes de servicios en lo referido a la organización adecuada de guardias, como así también de los períodos de licencias, evitando problemas como falta de cobertura.

- 4. Mayor control y trazabilidad de la información**

La automatización permitirá un control más riguroso sobre las entradas y salidas del personal, ausencias justificadas e injustificadas, y la asignación de guardias. Esto generará informes precisos y facilitará la toma de decisiones basada en datos concretos, así como una mayor trazabilidad y transparencia en la gestión del personal.

- 5. Mejora en la interacción entre las áreas de RRHH del hospital con las Capital Humano**

Un sistema adecuado, permitirá una mejor interacción entre las áreas de RRHH del hospital y de la Subsecretaría de Capital Humano, al reducir la brecha de tiempo existente en la producción de información útil y necesaria de cada empleado; los datos que requiere Capital Humano por parte del nosocomio estarán disponibles tan o más rápido que los que genera Capital Humano para ese mismo grupo de empleados.

- 6. Aumento de la satisfacción del personal**

Con una mejor administración de horarios, guardias y francos, el personal de salud experimentará mayor claridad y organización en sus rutinas laborales, lo que se traducirá en una mayor satisfacción y moral en el trabajo, al reducir problemas derivados de malentendidos o errores en la asignación de horarios, guardias y modificaciones imprevistas en la diagramación de los servicios.

- 7. Facilidad de acceso a datos históricos**

La informatización permitirá un acceso rápido y sencillo al historial de cada empleado, como faltas y tipo de faltas, cambios de horario y/o guardias, pases de salida y tipos de

pases de salida, etc. Esto facilitará la planificación futura y permitirá un análisis más profundo de patrones de asistencia y rendimiento, por empleado, y por servicio.

8. Mayor transparencia y cumplimiento normativo

El sistema garantizará una mayor transparencia en la gestión de los RRHH y facilitará el cumplimiento de las normativas laborales y administrativas¹, al mantener registros precisos y accesibles para auditorías internas o externas.

Estos beneficios tangibles mejorarán tanto la eficiencia operativa como la calidad del servicio médico que el hospital presta la comunidad, optimizando la administración del personal, lo que finalmente contribuye a lograr una atención más eficaz y oportuna a los pacientes.

3. Definiciones del proyecto

3.1 Objetivo general del proyecto

Desarrollar un sistema de gestión de recursos humanos para el Hospital de Urgencias de Córdoba, que automatice y optimice la gestión de novedades del personal, garantizando una administración más precisa y segura de la información. El sistema resolverá problemas de ineficiencia y errores administrativos, integrando estos procesos en una plataforma digital que facilite la toma de decisiones y mejore la coordinación con los sistemas municipales de recursos humanos.

3.2 Objetivos específicos del proyecto

Propósitos particulares y alcanzables para lograr el objetivo general del proyecto:

1. Analizar los requisitos del sistema:

- **Propósito:** Identificar y documentar las necesidades específicas del Hospital de Urgencias en relación con la gestión de RRHH.
- **Meta:** Completar un informe detallado de requisitos en un plazo de 4 semanas.

2. Diseñar el prototipo del sistema:

- **Propósito:** Crear un diseño funcional y técnico del sistema que abarque la gestión de ausencias, cambios de horarios, guardias y francos compensatorios.
- **Meta:** Finalizar el diseño del prototipo en 6 semanas, incluyendo diagramas y especificaciones técnicas.

3. Desarrollar el prototipo del sistema:

- **Propósito:** Implementar las funcionalidades del sistema de gestión de recursos humanos según el diseño aprobado.
- **Meta:** Completar el desarrollo del prototipo en 8 semanas, con un sistema funcional que permita la gestión de todas las novedades del personal.

4. Realizar pruebas del prototipo:

- **Propósito:** Verificar el funcionamiento del sistema mediante pruebas exhaustivas para asegurar que cumple con los requisitos y no presenta errores críticos.
- **Meta:** Llevar a cabo pruebas de funcionalidad, seguridad y usabilidad durante 4 semanas, resolviendo problemas identificados y ajustando el sistema según sea necesario.

5. Capacitar al personal del hospital:

¹ Los empleados municipales, están normados por el Estatuto del empleado municipal. Adicionalmente, los empleados del hospital de Urgencias, se rigen también por la Ordenanza Municipal Nro. XXXX

- **Propósito:** Capacitar a los usuarios finales en el uso del nuevo sistema para garantizar una transición fluida y efectiva.
 - **Meta:** Completar la capacitación del personal en 2 semanas, asegurando que todos los usuarios puedan operar el sistema de manera eficiente.
6. **Implementar el sistema en el entorno operativo del hospital:**
- **Propósito:** Desplegar el sistema en el entorno real del Hospital de Urgencias de Córdoba y asegurar su integración con los procesos existentes.
 - **Meta:** Implementar el sistema en 2 semanas, monitoreando el rendimiento y realizando ajustes finales según los comentarios de los usuarios.
7. **Evaluar el impacto del sistema:**
- **Propósito:** Evaluar el impacto del nuevo sistema en la eficiencia administrativa y en la calidad de la gestión de recursos humanos.
 - **Meta:** Realizar una evaluación de impacto y generar un informe de resultados en 4 semanas después de la implementación completa.

Se acompaña un diagrama de Gantt de las actividades que se desarrollarán para lograr estos objetivos en el [Anexo I: Diagrama de Gantt](#).

4. Definiciones del sistema

4.1 Objetivo general del sistema

Desarrollar un sistema de información que permita **gestionar, controlar y administrar** de manera automatizada las novedades del personal del Hospital de Urgencias, optimizando los procesos relacionados con inasistencias, cambios de horario y/o de guardias, francos compensatorios y pases de salida, para garantizar una administración precisa y segura de la información.

Funcionalidades generales del prototipo de sistema:

1. **Gestionar** las ausencias del personal, distinguiendo entre faltas justificadas, injustificadas, por fuerza mayor, y de otros tipos menos frecuentes.
2. **Controlar** los cambios de guardia y de horarios, asegurando que no se produzcan inconsistencias en la diagramación de servicios.
3. **Administrar** las horas extras para francos compensatorios, como así también una planificación eficiente de estos, conforme a las normativas vigentes.
4. **Registrar** los pases de salida oficiales y particulares del personal, proporcionando un seguimiento completo y centralizado.
5. **Generar** reportes periódicos de novedades semanales, y mensuales, para facilitar la comunicación con la Subsecretaría de Capital Humano y otras áreas relevantes.
6. **Proporcionar una herramienta** que facilite a cada jefe de servicio, la confección de diagramas de servicio mensuales, con las asignaciones del personal, integrando esta información en tiempo real con la jefatura de la oficina de Personal y la dirección hospitalaria, asegurando una gestión centralizada y eficiente de los RRHH.

4.2 Límites

Desde que se registra el alta de nuevo empleado asignado al nosocomio, gestionando su asignación a uno o más servicios y la diagramación de su modalidad de trabajo y horarios o

guardias acorde a dicha modalidad, **hasta que** se gestionan las novedades del personal (faltas, cambios de horarios, francos compensatorios y pases de salida), integrándose esta información en los reportes oficiales y coordinándose con las áreas pertinentes para la correcta administración de los RRHH.

4.3 Alcances

1. Gestión de altas, bajas y modificaciones (ABM) del personal y de los servicios.
2. Planificación de los diagramas de servicio y asignación de RRHH.
3. Registro de novedades en el parte diario de personal.
4. Generación de reportes personalizados.
5. Monitoreo de la disponibilidad del personal.

4.4 No Contempla

- El sistema no gestiona licencias de ningún tipo: ordinarias, sanitarias, por fallecimiento, entre otras, que ya son manejadas por el sistema Ve Di de Capital Humano.
- Se excluyen otros procesos externos: cualquier otra tarea que ya sea gestionada por sistemas externos o que no se relacione directamente con la administración específica de ausentismo, horarios y diagramación de servicios del hospital.

5. Elicitación

Para este proyecto, se definió realizar:

- **Encuesta mixta vía web:** preguntas de selección múltiple con opción a brindar una respuesta personalizada (casilla "Otro") y algunas preguntas abiertas, con respuesta de texto tipo largo.
- **Entrevistas personales:** de formato libre, tipo charla, con preguntas puntuales.

La encuesta se publicó en la nube hospitalaria y se distribuyó a la mayoría de los empleados a través de grupos de WhatsApp. En el transcurso de una semana, llamó la atención que menos del 3% del personal activo la respondiera (solo 13 respuestas). Inicialmente, esto se atribuyó a una posible falta de motivación por parte de los empleados para colaborar en el desarrollo de un sistema que, *a priori*, redundaría en su propio beneficio. Sin embargo, indagando posteriormente se determinó que, aunque en muchos casos solo una persona envió la respuesta, lo hizo en representación de sus compañeros de servicio (lo que se confirmó al observar que cada respuesta provenía de un servicio diferente). Al considerar la cantidad de respuestas por servicio recibidas (exceptuando el servicio de enfermería, donde solo se consideró un tercio de su personal debido a su gran tamaño y dispersión en distintos turnos), se puede estimar que entre el 25% y el 30% de los empleados participaron en la encuesta y aportaron sugerencias que no habían sido tenidas en cuenta previamente.

A partir de estas entrevistas, se obtuvo la información utilizada para redactar los puntos anteriores de este documento, como los detalles de la introducción, y las definiciones de proyecto y de sistema. En base a esto también, se diseñó la breve encuesta que luego fue distribuida al resto de los empleados. Además, permitió profundizar en las problemáticas identificadas:

La jefa de personal indicó que los tres hospitales municipales tienen diferentes dinámicas entre sí respecto del manejo del personal, y que, aun así, a ninguno de los tres hospitales les sirven los

sistemas de RRHH de Capital Humano. No obstante, por disposiciones de carácter legal y obligaciones administrativas, deben utilizar dichos sistemas. La falta de contar con información centralizada, en contraposición a tener que buscarlos en diferentes subsistemas, es algo tedioso, y un sistema, que pudiera integrar dicha información sería relevante.

5.1 Actividad del cliente o dominio

Aunque el Hospital de Urgencias tiene como actividad principal la atención de urgencias médicas, en el contexto de este proyecto, lo relevante es el **dominio de la gestión de recursos humanos (RRHH)** dentro del nosocomio. Entonces, aclarado esto, y enfocados ya en nuestro problema, diremos que para que la atención funcione de manera eficiente, es esencial que los aspectos internos, como la **gestión del personal**, estén bien organizados:

La actividad del cliente se enmarca en el área de Recursos Humanos del hospital de Urgencias, encargada de la administración del personal de salud (médico, administrativo, técnico, de servicios, etc.), garantizando la correcta diagramación de servicios, control de ausencias, y gestión de francos compensatorios, entre otros aspectos.

5.2 T.I.C (Tecnologías de la Información y Comunicación)

De acuerdo con los objetivos establecidos, es fundamental seleccionar los recursos tecnológicos adecuados para implementar una solución integrada por una aplicación de escritorio en **Java**, ya que este lenguaje permite aplicar los principios de la **Programación Orientada a Objetos (POO)** de manera robusta y escalable. Además, cumple con los requisitos establecidos por este Seminario de Práctica Profesional.

Para el *backend*, se utilizará **MariaDB**, fork ampliamente popular de MySQL con las mismas funcionalidades, pero sin los problemas de licencia asociados a MySQL (principal razón por la cual se produjo el fork). En la actualidad, MariaDB es el reemplazo natural en la mayoría de distribuciones *GNU/Linux*, lo que facilita su implementación y mantenimiento, no solo en el Hospital de Urgencias, sino en la mayoría de los *datacenters* modernos.

5.3 Relevamiento estructural y tecnológico

En cuanto a la infraestructura informática y de redes, el Hospital de Urgencias cuenta con un *datacenter* basado en *ProxmoxVE*, donde se ejecutan diversas VMs (mayormente *Windows Server 2022*) y contenedores LXC (mayormente *GNU/Linux Debian 12*). Los servicios que prestan estas VMs y LXC son múltiples y diversos: servicio de directorio, bases de datos de diferentes tipo y finalidades, sistemas de imágenes (PACS), etc. Independientemente de esto, para el desarrollo del proyecto, se destinarán recursos específicos a tal fin.

Para el desarrollo de la aplicación, el Servicio de Informática del nosocomio, se comprometió a proporcionar un contenedor LXC *GNU/Debian 12* con un servidor de bases de datos *MariaDB* con posibilidad de ser accesible desde Internet si fuera necesario para pruebas remotas o consultas externas. Además, se dispone de una estación de trabajo *GNU/Linux Deepin 23* con el IDE *IntelliJ IDEA Ultimate* para el desarrollo de la aplicación en *Java*, y *MySQL Workbench* para el modelado de la base de datos y su gestión, entre otras aplicaciones, siempre que se trata de software libre, o que se dispongan de las licencias correspondientes.

La red interna opera a través de ethernet gigabit con doble pila (IPv4 e IPv6) y se gestiona mediante un enrutador *FreeBSD pfSense*. El sistema de autenticación y autorización de usuarios está centralizado mediante *Active Directory*.

Si bien en el mercado existen sistemas diseñados para la gestión general de RRHH, la mayoría enfocados en el control de asistencia, nómina y seguimiento evaluación de desempeño, e incluso algunos como **Kronos Workforce Ready**, tienen herramientas que permitiría diagramar los servicios, no se adaptan fácilmente a las necesidades específicas del Hospital de Urgencias, con horarios irregulares, turnos rotativos y francos compensatorios. Pero el **principal problema**, es que todos estos sistemas son **costosos**, se cobran en divisas, y su personalización es compleja (y peor aún, probablemente nunca cubra las funcionalidades necesarias para este proyecto).

5.4 Relevamiento funcional

Organigrama

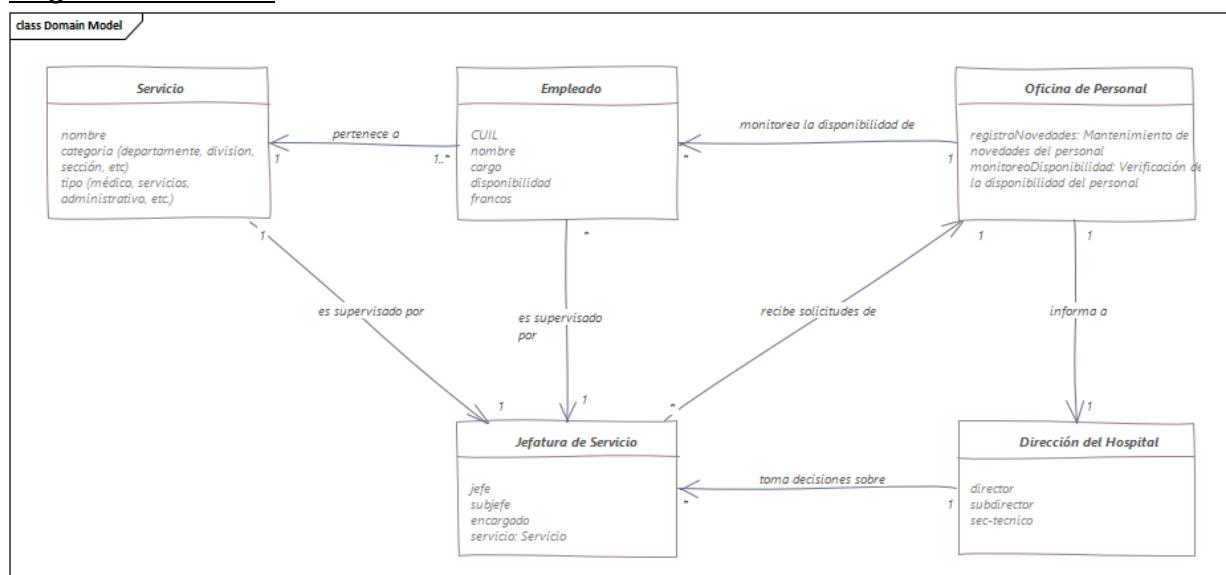
El organigrama de la institución, se adjunta en el [Anexo II: Organigrama del Hospital Municipal de Urgencias \(HMu\)](#).

Funciones de las áreas involucradas

Los servicios clave involucrados en los procesos relevados son, en primer lugar, la **Oficina de Personal**, que se encarga del alta, baja y modificación (ABM) de empleados, la recepción de los diagramas de servicio, y el control interno del cumplimiento de las normas y reglamentos establecidos. Por otro lado, el **Servicio de Informática** es responsable de la infraestructura tecnológica y de red del hospital, asegurando su correcto funcionamiento.

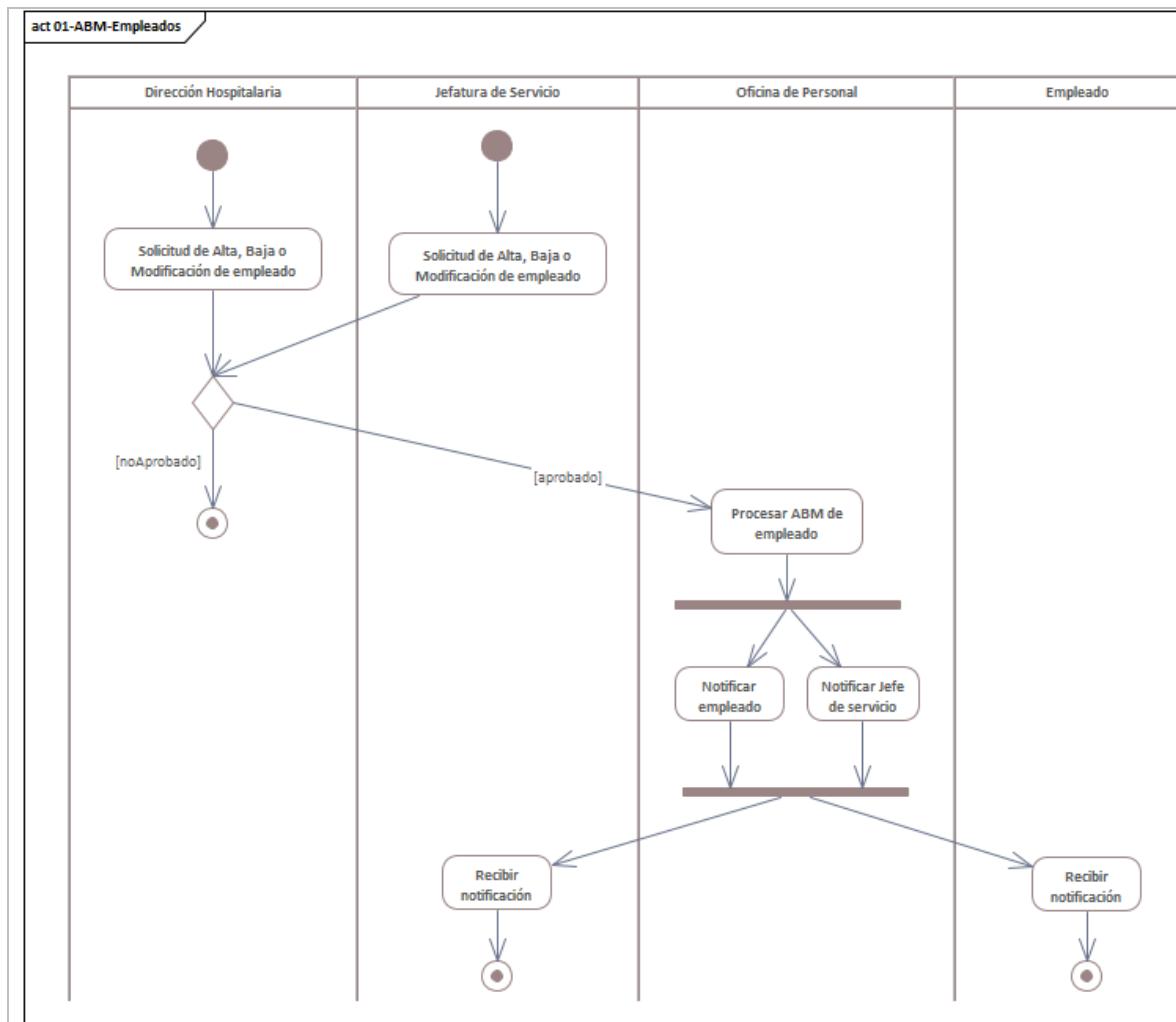
Finalmente, los **jefes o encargados de todos los servicios** tienen la responsabilidad de elaborar los diagramas de servicio correspondientes, los cuales serán revisados y controlados posteriormente por la Oficina de Personal, y eventualmente, por la Dirección hospitalaria.

Diagrama de Dominio



Procesos de negocio**Proceso 1: ABM (Alta, Baja, Modificación) de Empleados**

Proceso	ABM (Alta, Baja, Modificación) de Empleados
Roles	<p>Dirección del Hospital: Responsable primaria de solicitar altas, bajas o modificaciones, ya sea por decisión interna o por órdenes de una dependencia superior.</p> <p>Jefatura de Servicio: Puede solicitar altas o bajas de empleados en su área, sujeto a aprobación de la Dirección del Hospital.</p> <p>Oficina de Personal: Ejecuta las acciones solicitadas (altas, bajas o modificaciones) en el sistema y actualiza los registros.</p> <p>Empleado: Recibe notificaciones de cambios en su estatus (alta, baja, modificación).</p>
Pasos	<ol style="list-style-type: none"> 1. La Dirección del Hospital solicita el alta, baja o modificación de un empleado, ya sea por decisión interna o en cumplimiento de una orden superior. 2. Alternativamente, el Jefe de Servicio puede solicitar una alta, baja o modificación de personal de su área, pero esta solicitud debe ser revisada y aprobada por la Dirección del Hospital. 3. La Oficina de Personal recibe la solicitud aprobada e ingresa los cambios en el sistema: <ol style="list-style-type: none"> 1. Si es un alta, se registran los datos del nuevo empleado en el sistema, se generan las credenciales de acceso y se asignan turnos iniciales. 2. Si es una baja, se desactiva al empleado en el sistema, eliminando su acceso y actualizando los registros en la Subsecretaría de Capital Humano. 3. Si es una modificación, se actualizan los datos del empleado en el sistema (cambio de servicio, rol, etc.). 4. Oficina de Personal notifica automática y simultáneamente al empleado involucrado sobre su nueva condición (alta, baja, modificación) y al jefe de servicio a cargo sobre los cambios realizados en su equipo. 5. El empleado y Jefe de Servicio reciben la confirmación de los cambios realizados.

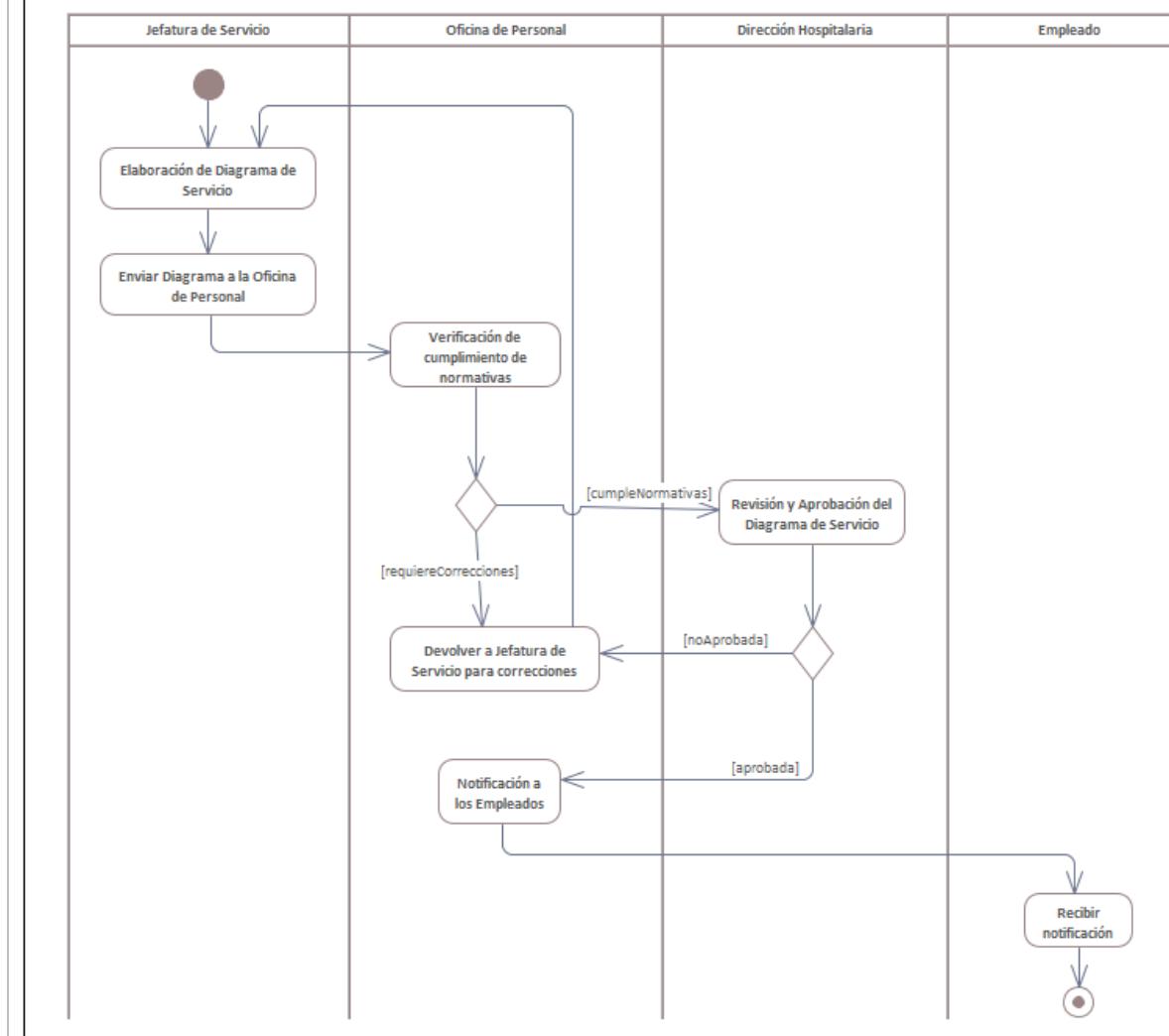


Proceso 2: Planificación de los Diagramas de Servicio y Asignación de RRHH

Proceso	Planificación de los Diagramas de Servicio y Asignación de RRHH
Roles	<p>Jefatura de Servicio: Encargada de elaborar los diagramas de servicio para su área y asignar los RRHH disponibles.</p> <p>Dirección del Hospital: Revisa y aprueba los diagramas de servicio propuestos por las Jefaturas de Servicio.</p> <p>Oficina de Personal: Coordina la distribución general del personal, asegurando que se cumplan las normativas y reglas laborales, y da el visto bueno final a los diagramas.</p> <p>Empleado: Recibe notificaciones acerca de su asignación de modalidad laboral y horarios para un mes determinado.</p>
Pasos	<ol style="list-style-type: none"> Jefatura de Servicio elabora su Diagrama de Servicio: cada jefe de servicio elabora un diagrama de servicio, en donde asigna los RRHH disponibles en su área a los distintos turnos y guardias. Envío del diagrama a la Oficina de Personal: La Jefatura de Servicio envía el diagrama a la Oficina de Personal para su revisión y aprobación. Revisión de los diagramas por la Oficina de Personal: La Oficina de Personal recibe los diagramas de servicio, verifica que no contengan errores, que cumplan con las normativas internas y la legislación vigente. <ul style="list-style-type: none"> Si los diagramas son correctos, se envían a la Dirección para su aprobación.

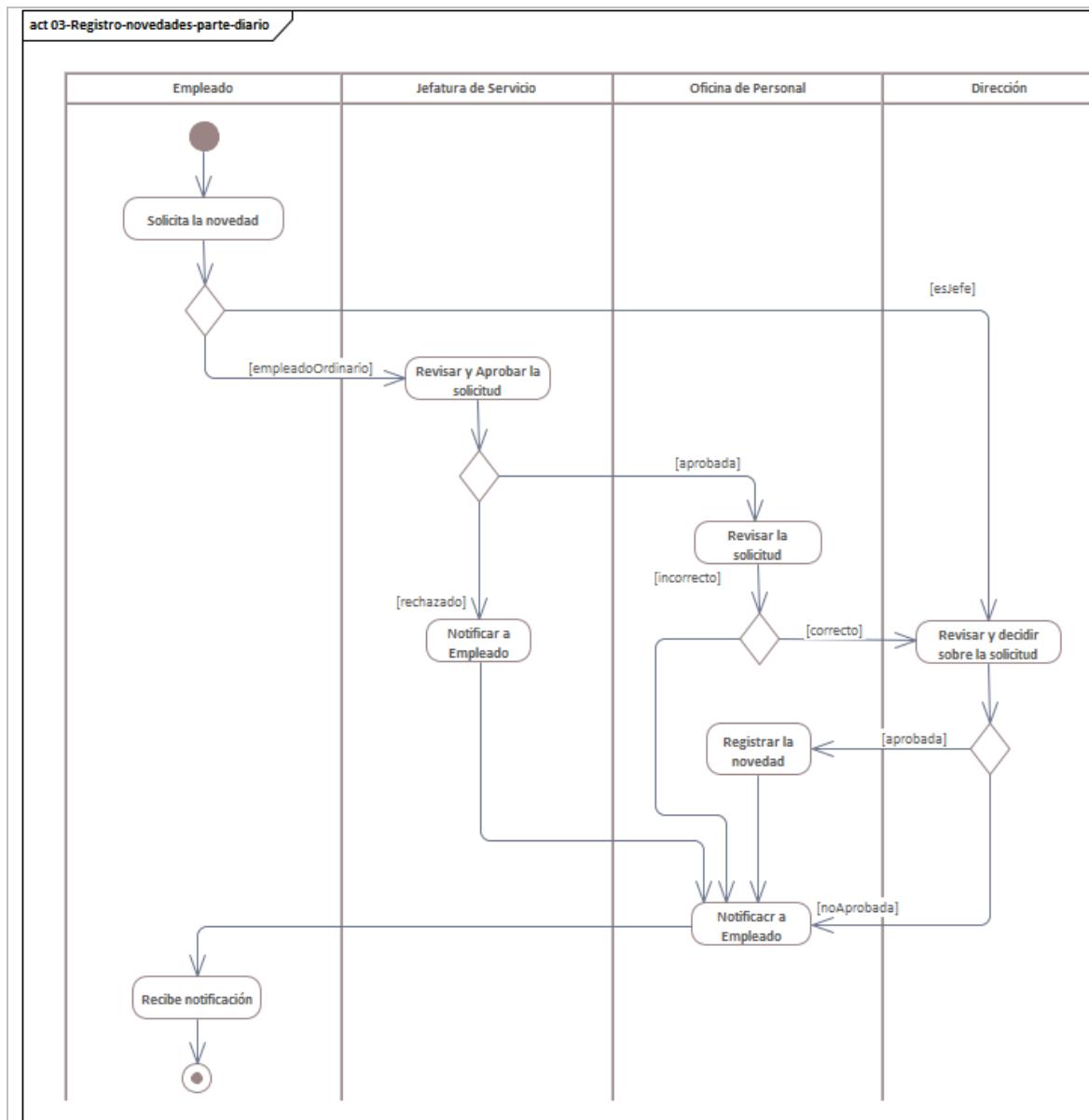
	<ul style="list-style-type: none"> • Si contienen errores, la Oficina de Personal notifica a la Jefatura de Servicio involucrada, solicitando las correcciones necesarias. <p>4. Revisión de los diagramas por la Dirección del Hospital:</p> <ul style="list-style-type: none"> • Si la Dirección aprueba los diagramas, los devuelve a la Oficina de Personal para que realice las notificaciones correspondientes. • Si la Dirección no aprueba los diagramas, notifica directamente a la Jefatura de Servicio para que confeccione un nuevo diagrama y también informa a la Oficina de Personal sobre la situación. <p>5. Notificación a los empleados: Una vez que la Oficina de Personal recibe la aprobación de la Dirección, notifica a los empleados afectados sobre sus nuevos horarios.</p> <p>6. Confirmación de recepción por parte del empleado: Cada empleado recibe la notificación de sus horarios correspondientes para el mes determinado.</p>
--	---

act 02-Planif-Diag-Serv-Asign-RRHH

**Proceso 3: Registro de novedades en el parte diario de personal**

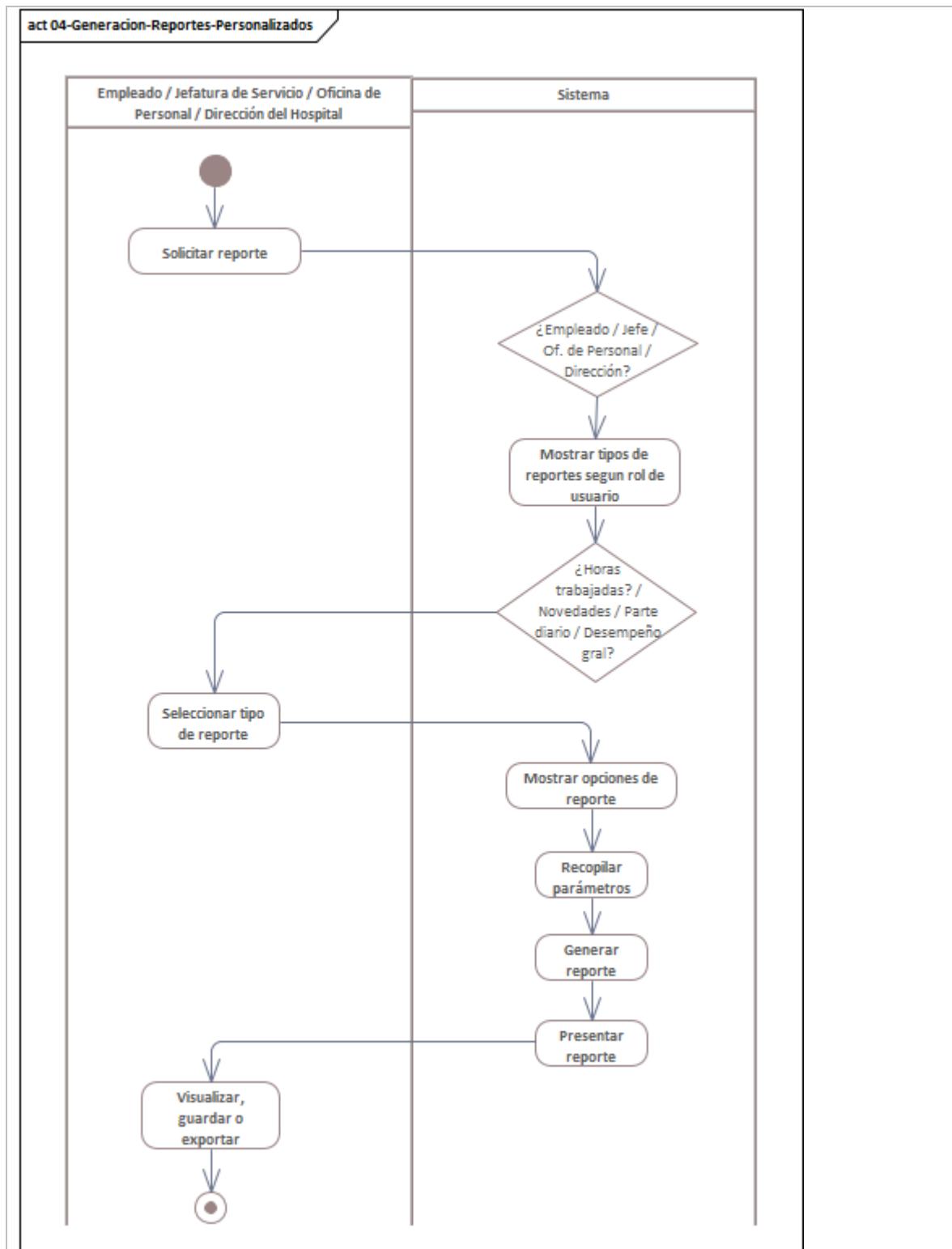
Proceso	Registro de novedades en el parte diario de personal
Roles	Empleado: Responsable de solicitar novedades (faltas, licencias, cambios de horario, guardias, etc.) a la Jefatura de Servicio o a la Dirección, según corresponda. También

	<p>puede ser notificado en caso de que alguna novedad afecte sus registros (por ejemplo, faltas justificadas agotadas, devolución horaria próxima a vencer, etc.).</p> <p>Oficina de Personal: Recibe, registra y valida las novedades en el sistema. Se asegura de que estén correctamente registradas para la generación de reportes y el cumplimiento de las normativas laborales.</p> <p>Dirección del Hospital: Aprueba las solicitudes de novedades previamente revisadas por la Oficina de Personal. También se encarga de aprobar directamente las solicitudes de los jefes de servicio.</p>
	<p>Jefatura de Servicio: Aprueba las solicitudes de novedades presentadas por los empleados bajo su responsabilidad, antes de enviarlas a la Oficina de Personal.</p>
Pasos	<ol style="list-style-type: none"> 1. Solicitud de novedades por el empleado: El empleado solicita la aprobación de una novedad a la Jefatura de Servicio (si el empleado es un jefe de servicio, realiza la solicitud directamente a la Dirección del Hospital). <ul style="list-style-type: none"> • Si la solicitud es aprobada por la Jefatura, se envía a la Oficina de Personal. • Si no es aprobada, la Oficina de Personal registrará la novedad según corresponda (generalmente aplicando un código disciplinario, en caso de rechazo). 2. Verificación por la Oficina de Personal: La Oficina de Personal recibe las solicitudes de novedades y las revisa para asegurarse de que no contengan errores (por ejemplo, códigos o fechas incorrectas). <ul style="list-style-type: none"> • Si la solicitud es correcta, se envía a la Dirección para su aprobación. • Si contiene errores, se rechaza y se notifica al empleado, quien deberá corregir la solicitud según las indicaciones. 3. Revisión y decisión de la Dirección del Hospital: La Dirección revisa la solicitud de novedades enviada por la Oficina de Personal y decide si aprobarla o no. <ul style="list-style-type: none"> • Si la aprueba, devuelve la solicitud a la Oficina de Personal para su registro en el parte diario correspondiente. • Si no la aprueba, la devuelve a la Oficina de Personal para que notifique al empleado sobre el rechazo. 4. Registro de la novedad y notificación al empleado: Una vez que la Oficina de Personal recibe la decisión de la Dirección, procede a registrar la novedad aprobada o, en caso de rechazo, a aplicar el código de sanción correspondiente. A continuación, notifica al empleado sobre el estado de su solicitud. 5. Confirmación de recepción por el empleado: El empleado recibe la notificación sobre la aprobación o rechazo de su solicitud de novedad y puede revisar los detalles correspondientes.

**Proceso 4: Generación de reportes personalizados**

Proceso	Generación de reportes personalizados
Roles	Empleado: Genera reportes de su propia actividad laboral, como horas trabajadas, horas disponibles para francos compensatorios, etc.
	Jefatura de Servicio: Genera reportes de su servicio, como el porcentaje de cada tipo de novedades de todos los empleados de su servicio, entre otros.
	Oficina de Personal: Genera reportes globales, como el parte diario de asistencia, ausencias, faltas, y cualquier otro informe relevante para la gestión del personal.
	Dirección del Hospital: Genera reportes de alto nivel, como el cumplimiento de guardias por servicio, comparativos de asistencia entre distintos servicios, entre otros.
Pasos	<ol style="list-style-type: none"> Acción del actor: Cada actor (Empleado, Jefatura de Servicio, Oficina de Personal, Dirección) accede al sistema y solicita la generación de un reporte específico. Selección del tipo de reporte: <ul style="list-style-type: none"> El sistema muestra las opciones de reportes disponibles según el rol del actor:

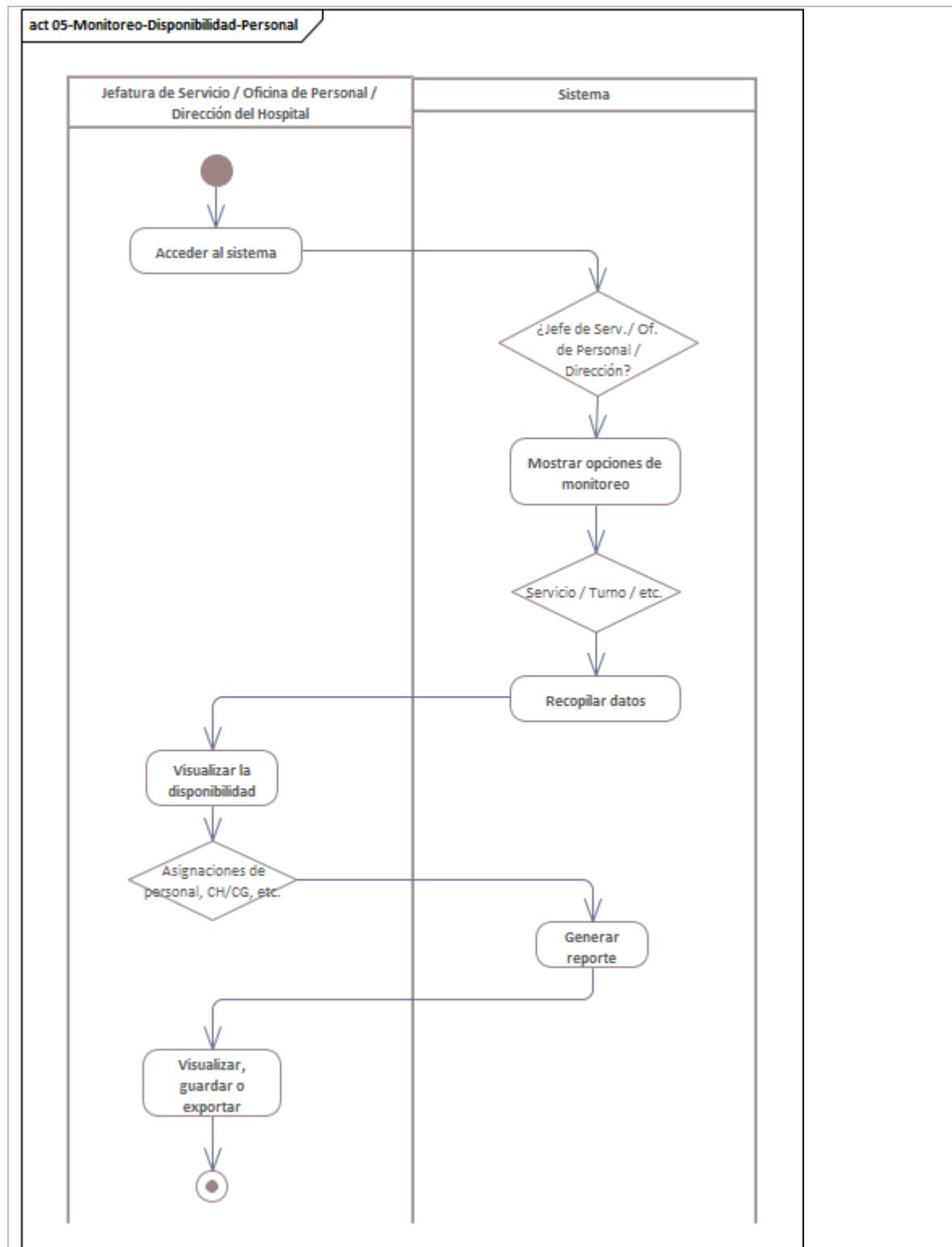
	<ul style="list-style-type: none">• Empleado: Reportes de horas trabajadas, horas para francos, turnos asignados, etc.• Jefatura de Servicio: Reportes como el porcentaje de cada tipo de novedad de todos los empleados de su servicio, entre otros, etc.• Oficina de Personal: Reporte completo del parte diario (faltas, licencias, horas trabajadas, etc.).• Dirección: Reportes globales por servicio, como cumplimiento de horas, guardias, comparativos de rendimiento.
	<p>3. Parámetros de generación:</p> <ul style="list-style-type: none">• El actor puede ajustar parámetros para personalizar el reporte (por ejemplo, rango de fechas, servicio específico, tipo de guardia, etc.).
	<p>4. Generación del reporte:</p> <ul style="list-style-type: none">• El sistema recopila la información solicitada, genera el reporte y lo presenta al actor en el formato adecuado (PDF, Excel, etc.).
	<p>5. Acciones posteriores:</p> <ul style="list-style-type: none">• Los actores pueden guardar el reporte, enviarlo por correo, o exportarlo en distintos formatos.• Si el reporte tiene fines oficiales, el sistema puede tener un paso adicional para validar o firmar electrónicamente.

**Proceso 5: Monitoreo de la disponibilidad del personal**

Proceso	Monitoreo de la disponibilidad del personal
Roles	Jefatura de Servicio: Revisa la disponibilidad del personal en su área para la planificación de turnos, guardias y coberturas de emergencias.

	<p>Oficina de Personal: Monitorea la disponibilidad de todo el personal en el hospital para gestionar la asignación global de recursos humanos.</p> <p>Dirección del Hospital: Supervisa la disponibilidad a nivel macro para evaluar si los servicios cuentan con el personal necesario y tomar decisiones estratégicas en casos de alta demanda o contingencias.</p> <p>Sistema: Muestra el estado de disponibilidad en tiempo real, proporcionando información sobre la ubicación y estatus de cada empleado.</p>
Pasos	<ol style="list-style-type: none"> 1. Acceso al sistema: <ul style="list-style-type: none"> • La Jefatura de Servicio, la Oficina de Personal o la Dirección acceden al sistema para monitorear la disponibilidad del personal. 2. Selección de criterios de búsqueda: <ul style="list-style-type: none"> • El actor selecciona los criterios según los cuales quiere monitorear la disponibilidad, como: <ul style="list-style-type: none"> • Servicio específico. • Fecha y hora. • Turno o guardia. • Tipo de disponibilidad (por ejemplo, si necesitan saber quién tiene francos disponibles, quién está de licencia, quién está en guardia, etc.). 3. Recopilación y visualización de datos: <ul style="list-style-type: none"> • El sistema recopila la información en tiempo real desde la base de datos del personal, mostrando: <ul style="list-style-type: none"> • Empleados disponibles (sin guardias asignadas, ni licencias). • Empleados en guardia. • Empleados con francos compensatorios disponibles. • Empleados ausentes (por licencia, falta u otro motivo). • El sistema puede mostrar la información visualmente (por ejemplo, usando colores para indicar la disponibilidad o ausencia). 4. Acciones según la disponibilidad: <ul style="list-style-type: none"> • Basado en la información presentada: <ul style="list-style-type: none"> • La Jefatura de Servicio puede reasignar personal disponible a cubrir turnos o guardias. • La Oficina de Personal puede coordinar coberturas entre servicios. • La Dirección puede tomar decisiones operativas estratégicas (como habilitar nuevos turnos en caso de necesidad). 5. Reporte opcional: <ul style="list-style-type: none"> • El sistema también puede permitir la generación de un reporte de disponibilidad que se puede exportar o imprimir para mantener un registro de la disponibilidad del personal en un determinado período de tiempo.

Sistema de Gestión de Ausentismo Hospitalario



6. Diagnóstico

Diagnóstico 1: ABM (Alta, Baja, Modificación) de empleados

Proceso	ABM (Alta, Baja, Modificación) de empleados
Problema/s	<ol style="list-style-type: none"> Ausencia de una base de datos unificada y centralizada: Actualmente, no existe una base de datos única que consolide la información de todos los empleados del hospital. En su lugar, se utilizan múltiples hojas de cálculo de Excel, cada una con datos fragmentados y, en muchos casos, inconsistentes entre sí. Por ejemplo, algunas hojas contienen el número de DNI del empleado, mientras que otras incluyen el servicio al que pertenece, sin que ninguna de ellas contenga toda la información de manera completa y precisa. Retrasos en la actualización de los datos del personal: Los cambios en el estado del personal (altas, bajas o modificaciones) no se reflejan de manera inmediata en todos los registros. La información se actualiza de forma progresiva a medida que se accede y modifica cada hoja de cálculo individualmente, lo que genera retrasos y problemas en la gestión administrativa y la planificación de los recursos humanos.
Causa/s	<ol style="list-style-type: none"> Falta de una base de datos SQL centralizada: La inexistencia de una base de datos estructurada y gestionada por un motor de bases de datos (como SQL) obliga al uso de hojas de cálculo de Excel para almacenar y gestionar la información del personal, lo que dificulta la unificación y centralización de los datos. Procesos manuales y descentralizados: La ausencia de un sistema automatizado y centralizado provoca que las actualizaciones de los datos de los empleados se realicen de manera manual y dispersa. Esto ocasiona que los cambios en el estado de los empleados no se apliquen de manera simultánea y uniforme en todos los registros, lo que deriva en inconsistencias y retrasos en la actualización de la información.

Diagnóstico 2: Planificación de los Diagramas de Servicio y Asignación de RRHH

Proceso	Planificación de los Diagramas de Servicio y Asignación de RRHH
Problema/s	<ol style="list-style-type: none"> Falta de una interfaz flexible que se adapte a las necesidades de cada servicio: Debido a la diversidad de modalidades horarias entre los distintos servicios del hospital, no existe un formulario estándar para la planificación de horarios y guardias. Cada servicio utiliza una plantilla personalizada, adaptada a sus propias necesidades, la cual es entregada en formato papel a la Oficina de Personal. Esto genera un retraso en el control de los horarios, ya que los empleados de la Oficina de Personal deben verificar manualmente cada plantilla contra el reporte de marcaciones de empleados (generados por los relojes de marcación), y luego registrar las novedades que se detectasen en el parte diario; evidentemente, la variedad de formatos y estructuras y el procesamiento artesanal complica en demasía la tarea. Retrasos en la planificación de los servicios: Las jefaturas de servicio suelen tardar en enviar los diagramas de servicio a la Oficina de Personal, lo que ocasiona problemas de organización. Este retraso impide un control eficiente de los horarios en el parte diario, ya que no se cuenta con la planificación de turnos a tiempo.
Causa/s	<ol style="list-style-type: none"> Falta de un sistema automatizado que permita gestionar los horarios y guardias: No existe una herramienta digital que permita a cada servicio crear y gestionar su propia planificación horaria en un calendario mensual. Además, no se genera automáticamente una vista estandarizada ni reportes personalizados que puedan ser fácilmente utilizados por la Oficina de Personal para verificar y registrar los horarios de los empleados. Falta de un sistema de notificación y control de fechas límite: La ausencia de un sistema que envíe alertas a las jefaturas de servicio y establezca fechas límite para la presentación de los diagramas contribuye a la tardanza en la entrega de los mismos.

	Además, algunos servicios que no realizan cambios significativos en la planificación de un mes a otro tienden a relajarse y omiten enviar una nueva planificación bajo la premisa de que "los horarios y guardias siguen siendo los mismos", lo que agrava el problema.
--	---

Diagnóstico 3: Registro de novedades en el parte diario de personal

Proceso	Registro de novedades en el parte diario de personal
Problema/s	<ol style="list-style-type: none"> Retrasos en la actualización del parte diario: Las novedades del personal, como ausencias, llegadas tarde o cambios de turno, no se actualizan en tiempo real. Esto dificulta la toma de decisiones operativas diarias, ya que la información no está completamente actualizada al momento de necesitarla. Falta de herramientas adecuadas que complican la carga de datos: Los agentes de la Oficina de Personal trabajan simultáneamente en la carga de novedades utilizando hojas de cálculo compartidas en un servidor de documentos. Sin embargo, este método de trabajo es susceptible a errores de edición. Cuando un agente comete un error sin percatarse, puede dañar el archivo. Al detectarse estos errores, es necesario restaurar versiones previas del archivo, lo que provoca la pérdida de las entradas que se habían realizado correctamente. Esto duplica el esfuerzo de trabajo y genera inefficiencia en el proceso.
Causa/s	<ol style="list-style-type: none"> Falta de integración y automatización entre sistemas: La ausencia de un sistema centralizado que permita gestionar las novedades del personal de manera automatizada y en tiempo real retrasa la actualización de la información, lo que afecta la operatividad del hospital. Procedimientos manuales y riesgo de errores: La carga de novedades se realiza de forma manual en hojas de cálculo, lo que la hace vulnerable a errores humanos. Además, la edición simultánea en un archivo compartido sin un sistema de control adecuado aumenta el riesgo de dañar los datos y perder información previamente ingresada.

Diagnóstico 4: Generación de reportes personalizados

Proceso	Generación de reportes personalizados
Problema/s	<ol style="list-style-type: none"> Dificultad para generar reportes actualizados: Los usuarios (agentes de la Oficina de Personal y Dirección) no pueden generar reportes en tiempo real, lo que impide un análisis preciso de la situación laboral. Datos incompletos o inconsistentes en los reportes: Los reportes generados no reflejan adecuadamente la situación actual, lo que afecta las decisiones estratégicas.
Causa/s	<ol style="list-style-type: none"> Sistema de reportes desactualizado: El sistema de reportes no está diseñado para gestionar los cambios en tiempo real, lo que genera demoras en la actualización de los datos. Fragmentación de la información: La información sobre los empleados, guardias y ausencias está dispersa en diferentes sistemas y no se encuentra integrada, lo que genera inconsistencias.

Diagnóstico 5: Monitoreo de la disponibilidad del personal

Proceso	Monitoreo de la disponibilidad del personal
Problema/s	<ol style="list-style-type: none"> Incapacidad para prever ausencias y planificar coberturas: No se cuenta con un sistema que permita anticipar ausencias programadas o faltas, lo que dificulta la planificación preventiva.

	2. Falta de visibilidad en tiempo real de la disponibilidad del personal: Los responsables de la planificación no tienen acceso inmediato a la disponibilidad actualizada del personal, lo que genera problemas para cubrir turnos o guardias.
Causa/s	1. Descoordinación entre servicios: Los servicios no tienen mecanismos de comunicación eficaces entre sí, lo que retrasa la redistribución del personal en situaciones de emergencia.
	2. Falta de un sistema de monitoreo centralizado: No existe un sistema que integre en tiempo real la disponibilidad del personal, lo que limita la capacidad de reacción ante cambios o emergencias.

7. Propuesta de solución

El proyecto tiene como objetivo principal desarrollar un **Sistema de Gestión de RRHH** centralizado y automatizado, diseñado para optimizar y mejorar la administración de las novedades del personal del Hospital de Urgencias de Córdoba. Este sistema permitirá gestionar de manera eficiente y precisa las Altas, Bajas, Modificaciones, asignaciones de horarios, guardias, horas extras para francos compensatorios, devoluciones de pases de salida, omisiones de marcación, y cualquier otra novedad del personal, proporcionando a la Oficina de Personal, Jefaturas de Servicio y Dirección del Hospital, una herramienta unificada que garantice una toma de decisiones más rápida y basada en información actualizada.

La solución propuesta consta de tres dimensiones: Funcional, Técnica y de Despliegue.

7.1 Propuesta funcional

El sistema proporcionará una plataforma digital integrada, accesible para los diferentes actores del hospital (Oficina de Personal, Jefaturas de Servicio, Dirección del Hospital, y Empleados), cada uno con sus respectivos niveles de acceso y capacidades. Los componentes funcionales clave del sistema serán los siguientes:

- **ABM (Alta, Baja y Modificación) de personal en la base de datos centralizada:** Los agentes de la Oficina de Personal podrán gestionar el alta, baja y modificación de empleados directamente en una base de datos centralizada, asegurando que la información esté siempre actualizada y accesible para los distintos procesos del hospital. Esta funcionalidad permitirá una administración más precisa de los recursos humanos, mejorando la coordinación y evitando la dispersión de información en múltiples hojas de cálculo.
- **Planificación de diagramas de servicio:** Se integrará una herramienta de planificación de horarios que permita a cada servicio generar diagramas personalizados y adaptados a las diferentes modalidades horarias del hospital. Los jefes de servicio podrán ingresar los turnos horarios y/o guardias directamente en el sistema, que los validará y los enviará a la Oficina de Personal para su revisión y aprobación, reduciendo los errores y eliminando el uso de múltiples formatos en papel o planillas de Excel.
- **Presentación de horas extras y confección automática de francos compensatorios:** El sistema permitirá a los empleados presentar las horas extras trabajadas de manera digital. Estas horas serán automáticamente calculadas para la creación de francos compensatorios, que podrán ser gestionados por la Oficina de Personal. Esta funcionalidad permitirá un control más preciso y automatizado de las horas adicionales trabajadas y los días de descanso a los que tiene derecho el personal.
- **Informe de horas para la devolución de pases de salida:** El sistema permitirá que las horas utilizadas para pases de salida oficiales o particulares se informen automáticamente. Esto facilitará

el cálculo de las horas que deben ser devueltas, asegurando que la gestión de las horas pendientes sea clara y esté al día.

- **Registro de omisiones de marcación de ingreso o egreso:** Los empleados y la Oficina de Personal podrán registrar omisiones de marcación de ingreso o egreso en el sistema. Esta funcionalidad garantizará que las ausencias o errores en las marcaciones sean identificados rápidamente y corregidos en el parte diario, asegurando que los empleados sean debidamente notificados y los ajustes se realicen con prontitud.
- **Envío de memorandos digitales:** El sistema permitirá a los empleados enviar notas tipo memorándum a sus respectivos jefes de servicio, a la Oficina de Personal o incluso a la Dirección del Hospital. Estos memorandos podrán ser aprobados por la Jefatura de Servicio antes de su envío, garantizando un flujo de comunicación formal y controlado. Esta funcionalidad ayudará a mejorar la comunicación interna y la trazabilidad de las solicitudes y notas entre los distintos niveles del hospital.
- **Generación de reportes personalizados:** El sistema permitirá la generación de reportes personalizados para cada actor (empleados, Jefatura de Servicio, Oficina de Personal y Dirección del Hospital). Estos reportes incluirán, entre otros, el parte diario del personal, reportes de cumplimiento de horarios, horas trabajadas, licencias y francos compensatorios disponibles.
- **Automatización de alertas y notificaciones:** El sistema enviará alertas automáticas para recordar a las Jefaturas de Servicio las fechas límite para la entrega de los diagramas de servicio. También notificará a los empleados sobre las novedades que afecten su situación laboral (por ejemplo, cambios de turno o aprobaciones de licencias).
- **Monitoreo de la disponibilidad del personal:** El sistema ofrecerá una interfaz que muestre en tiempo real la disponibilidad del personal, permitiendo a las Jefaturas de Servicio y a la Dirección tomar decisiones informadas respecto a la asignación de recursos humanos. También se podrá consultar la disponibilidad futura basada en las licencias y francos programados.

7.2 Propuesta técnica

El sistema estará basado en una arquitectura modular y escalable que facilite su uso interno y permita optimizar la administración de los recursos humanos del Hospital de Urgencias de Córdoba. Las principales características técnicas del sistema serán:

- **Base de datos centralizada con *MariaDB*:** Se implementará una base de datos SQL utilizando el motor *MariaDB*, que es una opción robusta y de código abierto, basada en *MySQL*. Esta base de datos centralizada almacenará toda la información relacionada con los empleados, sus novedades, horarios, licencias, guardias y reportes. El uso de *MariaDB* garantizará la integridad y consistencia de los datos, eliminando la dependencia de planillas de *Excel* dispersas y desactualizadas.
- **Desarrollo en lenguaje de programación robusto:** El sistema será desarrollado en *Java*, lenguaje orientado a objetos ampliamente utilizado, que garantiza flexibilidad, seguridad y mantenibilidad. El uso de *Java* permitirá la construcción de una aplicación que pueda cumplir con los requisitos técnicos exigidos por la cátedra de Seminario de Práctica Informática, asegurando la portabilidad y la eficiencia de la solución.
- **Interfaz de usuario:** el sistema tendrá como interfaz principal una **aplicación de escritorio**, únicamente; es decir que no se desarrollarán (en lo inmediato al menos), ni interfaces para dispositivos móviles, ni *webapps* para navegadores. En cambio, se pondrá el foco en ofrecer una interfaz intuitiva, diseñada para facilitar el uso por parte de los empleados de la Oficina de Personal, Jefaturas de Servicio y Dirección del Hospital, que permita un acceso fácil y amigable a las funciones clave, como el registro de novedades, la planificación de diagramas de servicio, cambios de horario, y generación de reportes, entre otras funcionalidades.

- **Integración con sistemas existentes:** Dado que los sistemas municipales existentes (como VeDi u otros) no proveen APIs que permitan la interacción para la consulta de datos o el intercambio de información automatizado, se buscará una **convivencia fluida con estos sistemas**. Aunque no será posible una integración automatizada, se implementarán mecanismos que minimicen la intervención manual del usuario, tendientes a facilitar el intercambio de información entre ambos sistemas, que permita a los usuarios trabajar de manera más eficiente con datos provenientes de diferentes plataformas.
- **Seguridad y control de acceso:** Se implementarán políticas de control de acceso que limiten las funciones y datos disponibles para cada tipo de usuario, garantizando la seguridad y confidencialidad de la información. Solo el personal autorizado podrá realizar modificaciones o consultar ciertos datos, dependiendo de su rol dentro del hospital.

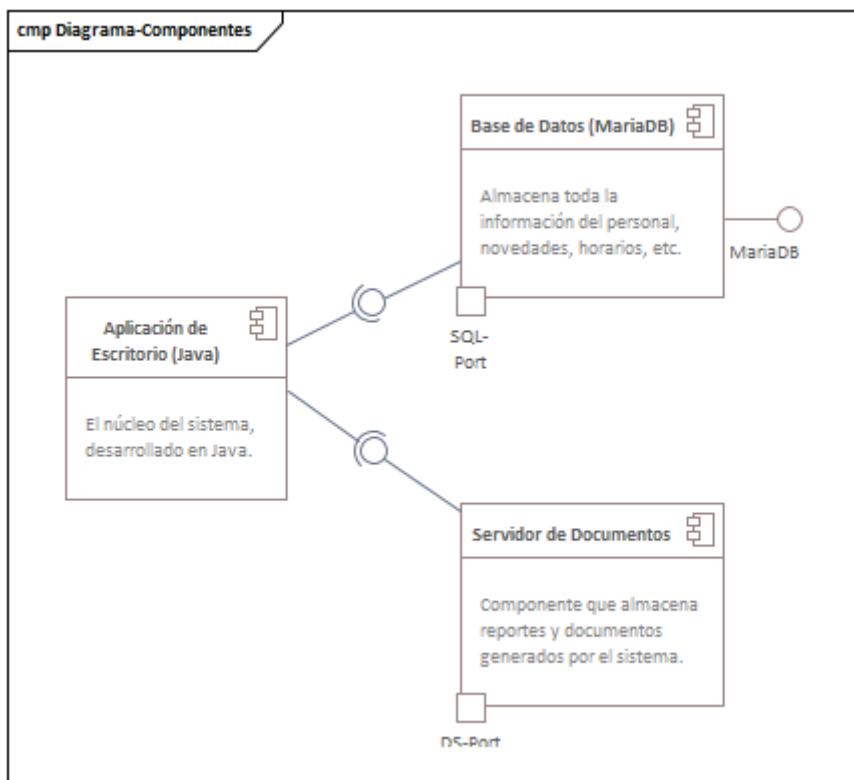
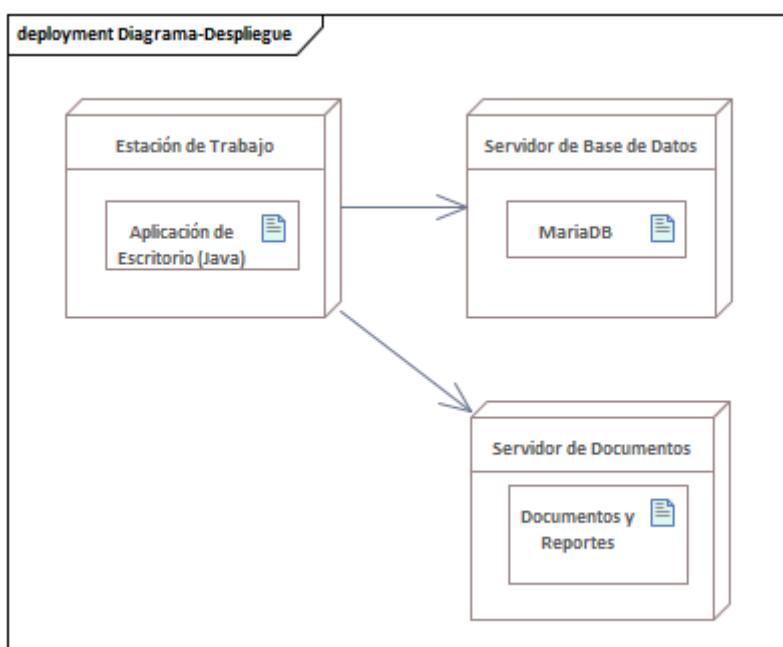
7.3 Propuesta de despliegue

El despliegue del sistema se llevará a cabo en fases que permitan una implementación ordenada y controlada, minimizando las interrupciones en los procesos actuales del hospital. El enfoque propuesto es el siguiente:

- **Fase 1: Diseño e implementación de la base de datos centralizada:** Esta primera fase se centrará en consolidar la información dispersa en una única base de datos SQL que sirva como la columna vertebral del sistema.
- **Fase 2: Desarrollo de módulos básicos:** Se desarrollarán los módulos principales, como el ABM de personal, la planificación de los diagramas de servicio, la presentación de horas extras para francos compensatorios, la presentación de faltas justificadas, injustificadas y por fuerza mayor, y los módulos de aprobación de todo estos, asegurando que cada funcionalidad esté bien integrada y probada.
- **Fase 3: Capacitación del personal y pruebas piloto:** Antes del despliegue completo, se realizará una capacitación intensiva para los usuarios clave (Oficina de Personal y Jefaturas de Servicio). También se llevará a cabo una prueba piloto para asegurar el correcto funcionamiento del sistema en un entorno real.
- **Fase 4: Despliegue completo y Desarrollo de módulos extras:** Una vez completadas las pruebas piloto, el sistema será desplegado en todo el hospital y se comenzarán con la implementación los módulos restantes, como la funcionalidad de envíos de memorándums, y el monitoreo de la disponibilidad del personal.
- **Fase 5: Mantenimiento y mejoras continuas:** Despues del despliegue, el sistema será monitoreado y mejorado constantemente para adaptarse a las necesidades cambiantes del hospital y para implementar nuevas funcionalidades según sea necesario

7.4 Diagrama de Arquitectura

En UML no existe un Diagrama de Arquitectura como tal. Pero acorde a lo tratado en la bibliografía (Unidad 1, Bloque 2, Desafío Propuesta Técnica), se entiende que se solicita una representación de alto nivel que muestre los componentes clave del sistema, su distribución y cómo se relacionan entre sí. En UML, tenemos dos diagramas que nos ayudan con esto: **Diagrama de componentes:** que proporciona una vista más bien lógica, de los componentes de software del sistema, y **Diagrama de despliegue:** que refleja cómo se despliegan los componentes de software en los diferentes nodos físicos.

Diagrama de componentes:**Diagrama de Despliegue:**

7.5 Listado de requerimientos funcionales en matriz de trazabilidad

Id Req	Descripción	Id Caso de Uso	Id Caso de Prueba	Validado
RFS01	El sistema deberá permitir a la Oficina de Personal gestionar las altas, bajas y modificaciones de los empleados en la base de datos centralizada.	CURFS01	CPRFS01	Sí
RFS02	El sistema deberá permitir a la Jefatura de Servicio generar diagramas de servicio personalizados, asignando empleados a turnos y guardias específicos.	CURFS02	CPRFS02	Sí
RFS03	El sistema deberá permitir consultar en tiempo real la disponibilidad del personal, incluyendo horarios programados, licencias y faltas.	CURFS03	CPRFS03	Sí
RFS04	El sistema deberá permitir a los empleados registrar horas extras y actualizar automáticamente el total acumulado de horas extras disponibles cada vez que se realice un nuevo registro.	CURFS04	CPRFS04	Sí
RFS05	El sistema deberá registrar las horas utilizadas en pases de salida y calcular las horas pendientes de devolución.	CURFS05	CPRFS05	Sí
RFS06	El sistema deberá permitir registrar omisiones de marcación de ingreso o egreso, y añadirlas al parte diario.	CURFS06	CPRFS06	Sí
RFS07	El sistema deberá permitir a los empleados enviar memorándums a sus respectivos jefes, la Oficina de Personal o la Dirección del Hospital.	CURFS07	CPRFS07	Sí
RFS08	El sistema deberá generar reportes personalizados de horas trabajadas, francos compensatorios, licencias y cumplimiento de horarios, según el rol del usuario.	CURFS08	CPRFS08	Sí
RFS09	El sistema deberá enviar alertas automáticas para recordar las fechas límite de presentación de diagramas de servicio y eventos importantes.	CURFS09	CPRFS09	Sí
RFS10	El sistema deberá controlar el acceso a las funcionalidades según el rol del usuario (Empleado, Jefatura de Servicio, Oficina de Personal, Dirección).	CURFS10	CPRFS10	Sí
RFS11	El sistema deberá permitir a los empleados generar francos compensatorios a partir de las horas extras disponibles, y solicitar los mismos a sus jefatura de servicio correspondiente.	CURFS11	CPRFS11	Sí

7.6 Listado de requerimientos no funcionales

Id Req	Descripción	Categoría
RFN01	El sistema será una aplicación de escritorio y deberá funcionar en los sistemas operativos Windows y Linux. El entorno de desarrollo estará basado en Java , y la base de datos deberá utilizar MariaDB como motor de base de datos.	Compatibilidad y plataforma
RNF02	La interfaz de la aplicación de escritorio deberá ser intuitiva y fácil de usar , con un diseño que permita a los usuarios encontrar fácilmente las funcionalidades que necesitan.	Usabilidad
RFN03	El sistema deberá ser capaz de manejar cargas simultáneas de usuarios , con al menos 50 usuarios concurrentes sin que el rendimiento del sistema se vea afectado significativamente.	Rendimiento
RNF04	El sistema deberá implementar mecanismos de autenticación para garantizar que solo los usuarios autorizados puedan acceder al sistema. Se deberá controlar el acceso mediante usuario y contraseña, y aplicar políticas	Seguridad y autenticación

	de seguridad como la caducidad de contraseñas y la complejidad de las mismas.	
RNF05	El sistema deberá estar disponible al menos el 95% del tiempo , permitiendo que los usuarios puedan acceder y utilizar las funcionalidades sin interrupciones importantes.	Disponibilidad
RFN06	Los datos del personal deberán ser almacenados de manera segura, cumpliendo con los estándares de protección de datos. Solo los usuarios autorizados tendrán acceso a información confidencial, como datos de licencias o reportes de novedades	Confidencialidad de los datos
RNF07	Las consultas a la base de datos deberán tener un tiempo de respuesta inferior a 2 segundos para garantizar una experiencia fluida.	Rendimiento
RNF08	La primera versión del sistema deberá estar disponible dentro del plazo de tiempo acordado con los responsables del hospital, respetando las fases de desarrollo previstas.	Entrega
RNF09	El sistema deberá facilitar la convivencia con los sistemas municipales existentes, generando reportes exportables en formatos estándar como Excel o CSV.	Interoperabilidad
RNF10	El sistema deberá cumplir con las regulaciones legales sobre protección de datos, como las leyes locales de protección de datos personales aplicables al hospital.	Legales

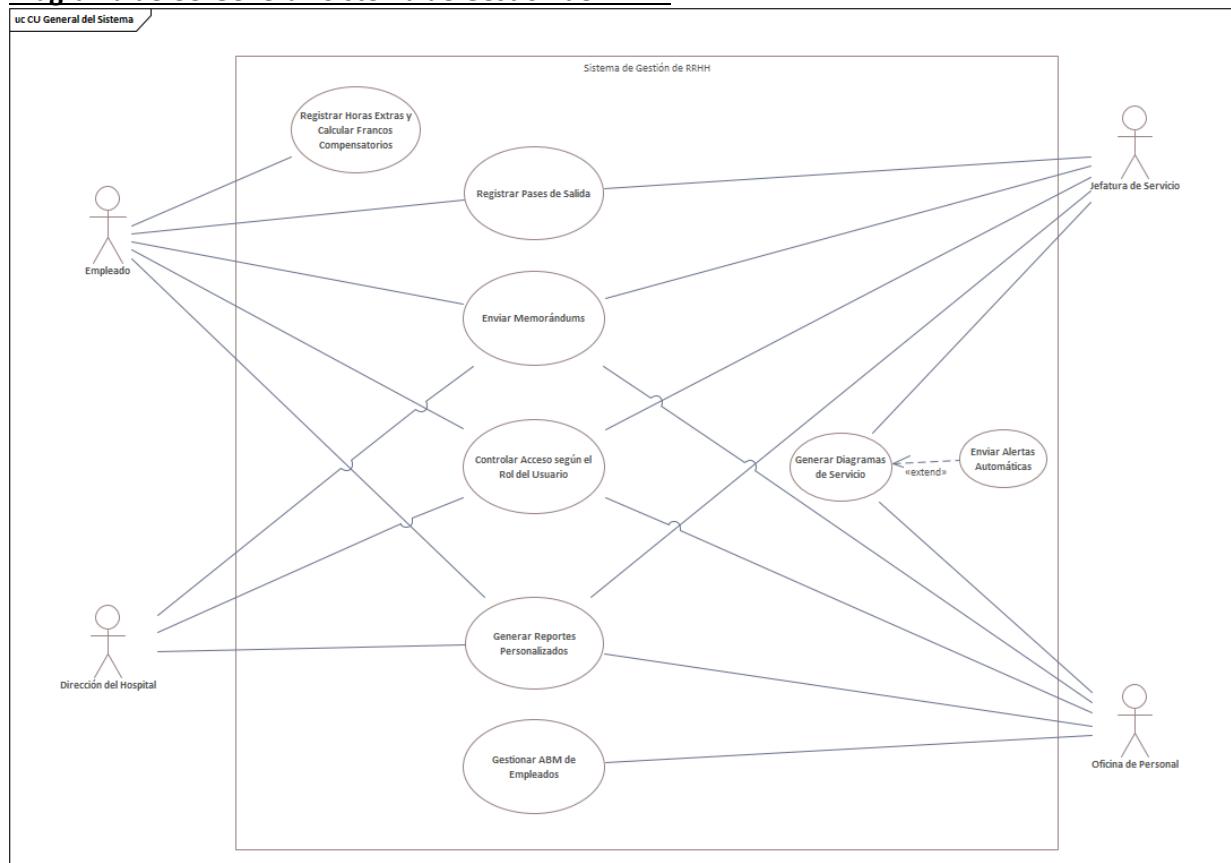
7.7 Listado de requerimientos candidatos

Listado de requerimientos candidatos Funcionales

Id Req	Descripción	Categoría
RCSF01	El sistema podrá automatizar la solicitud, aprobación y gestión de licencias (por enfermedad, vacaciones, etc.), permitiendo que los empleados envíen solicitudes directamente desde el sistema. Estas solicitudes pasarán por un proceso de aprobación que involucra a la Jefatura de Servicio y la Oficina de Personal .	Gestión de licencias con automatización
RCSF02	Se integrará el sistema con sistemas municipales externos (como VeDi) mediante APIs si en el futuro estos sistemas proporcionan interfaces para la consulta o el intercambio de información automática.	Integración con sistemas externos (VeDi)
RCSF03	El sistema generará reportes avanzados que incluyan análisis predictivo sobre la disponibilidad futura del personal y posibles ausencias basadas en datos históricos, permitiendo una mejor planificación.	Reportes avanzados con análisis predictivo
RCSF04	El sistema podrá generar automáticamente los horarios y guardias del personal según las reglas de disponibilidad y horarios, optimizando la asignación de recursos humanos sin intervención manual.	Automatización de asignación horaria / Asistencia de IA
RCSF05	El sistema proporcionará una funcionalidad de auditoría , que permitirá hacer un seguimiento detallado de los cambios en los datos de los empleados y las decisiones tomadas por los supervisores	Auditoría y seguimiento de cambios

Listado de requerimientos candidatos No Funcionales

Id Req	Descripción	Deseabilidad
RCNF01	Se implementará un portal de autogestión web donde los empleados puedan gestionar aspectos de su perfil, como actualizar su información personal, consultar sus licencias, francos, y cargar horas extras directamente.	Aplicación web / Portabilidad
RCNF02	Se desarrollará una app móvil que permita a los empleados consultar sus horarios, enviar solicitudes de licencias o memorandos, y verificar sus francos compensatorios, desde cualquier lugar y dispositivo móvil.	Aplicación móvil / Portabilidad
RCNF03	Para mejorar la seguridad, se implementará una autenticación multifactor que requiera no solo usuario y contraseña, sino también un código temporal enviado a dispositivos móviles para verificar la identidad del usuario.	Mejora en la seguridad con autenticación multifactor (MFA)

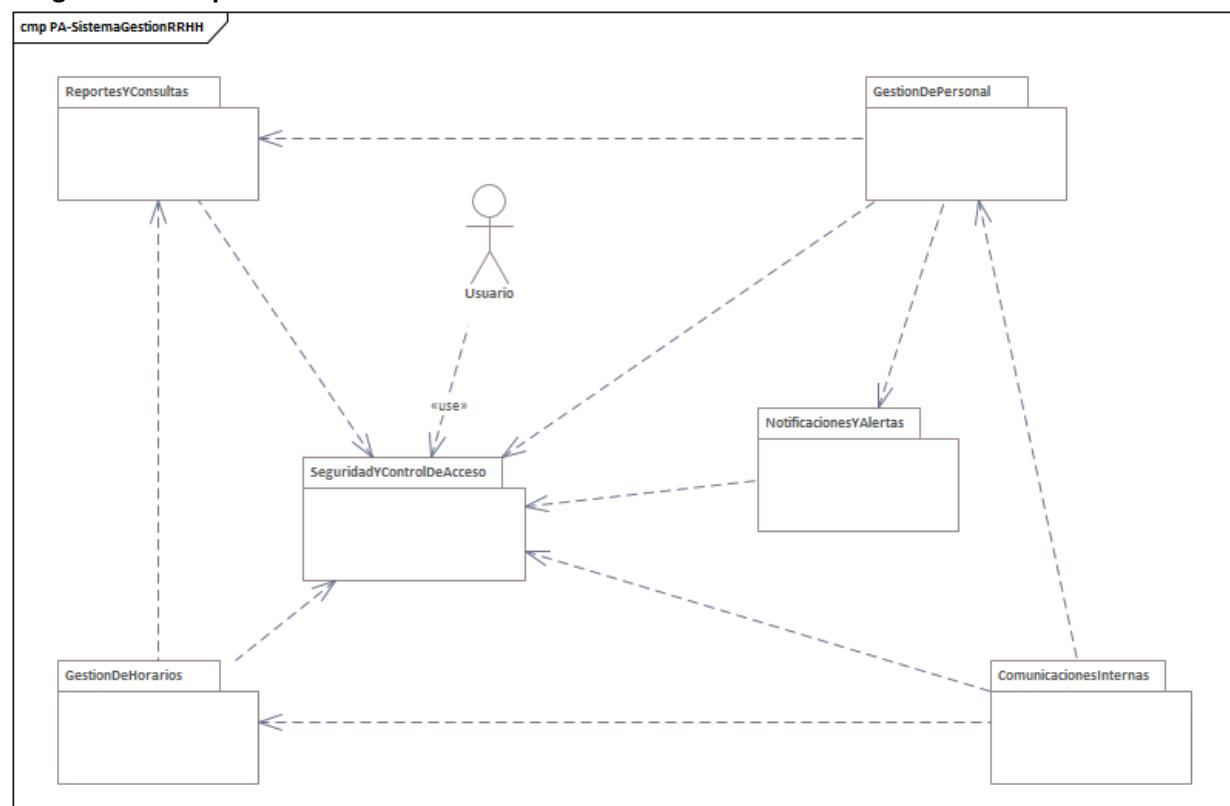
8. Desarrollo del prototipo**8.1 Diagrama de casos de uso****Diagrama de CU General: Sistema de Gestión de RRHH**

Identificación de actores

- **Empleado:** Usuario que interactúa con el sistema para registrar sus horas, consultar reportes, enviar memorándums, etc.
- **Jefatura de Servicio:** Usuario con más responsabilidades que puede gestionar diagramas de servicio, recibir alertas, validar horas extras, entre otras tareas.
- **Oficina de Personal:** Encargados de gestionar personal y recibir reportes detallados sobre la asistencia, control de horarios, ABM de empleados, entre otras tareas.
- **Dirección del Hospital:** El actor de mayor nivel que puede generar reportes globales, acceder a información general, recibir alertas importantes, entre otras funciones.

Trazabilidad

Es importante plantear una trazabilidad para mantener la relación entre los requerimientos, casos de uso y el resto de los artefactos correspondientes al PUD. Pero dado que una de las columnas de esta tabla hace referencia al paquete de análisis, parece oportuno presentar antes, un Diagrama de Paquete de Análisis, que muestre como se entendió la organización lógica del sistema.

Diagrama de Paquete de Análisis

1. Paq. de Gestión de Personal: agrupa todas las funcionalidades relacionadas con la gestión de empleados, cambios en la BD de empleados y control del registro del personal.

Abarca:

- CU01: Gestionar Altas, Bajas y Modificaciones de Empleados
- CU03: Consultar Disponibilidad del Personal en Tiempo Real
- CU05: Registrar Pases de Salida y Calcular Horas Pendientes de Devolución
- CU06: Registrar y Ajustar Omisiones de Marcación en el Parte Diario

2. Paq. de Gestión de Horarios: agrupa los casos de uso relacionados con la planificación de horarios y la administración de guardias y horas extras del personal.

- Abarca:** CU02: Generar Diagrama de Servicio Personalizado
CU04: Registrar Horas Extras y Calcular Francos Compensatorios
- 3. Paq. de Comunicaciones Internas:** se centra en la comunicación entre los empleados, jefaturas de servicio y otras áreas del hospital.
- Abarca:** CU07: Enviar Memorándums
- 4. Paq. de Reportes y Consultas:** agrupa los casos de uso relacionados con la generación de reportes personalizados para diferentes roles.
- Abarca:** CU08: Generar Reportes Personalizados
- 5. Paq. de Notificaciones y Alertas:** está relacionado con las alertas automáticas y recordatorios que el sistema envía a los usuarios.
- Abarca:** CU09: Enviar Alertas Automáticas
- 6. Paq. de Seguridad y Control de Acceso:** engloba las funcionalidades relacionadas con el control de acceso al sistema y la seguridad basada en el rol de cada usuario.
- Abarca:** CU10: Controlar Acceso según el Rol del Usuario

Tabla de Trazabilidad

Caso de uso	Requerimiento	Actor principal	Paquete del análisis	Comentario
CU01	RFS01	Oficina de Personal	Gestión de Personal	ABM de empleados en la base de datos del sistema
CU02	RFS02	Jefatura de Servicio	Gestión de Horarios	Planificación y asignación de recursos
CU03	RFS03	Oficina de Personal	Gestión de Personal	Consulta en tiempo real de la disponibilidad
CU04	RFS04	Empleado	Gestión de Horarios	Registro de horas extras y cálculo de francos
CU05	RFS05	Empleado, Jefatura de Servicio	Gestión de Personal	Registro y cálculo de devolución de horas
CU06	RFS06	Empleado, Oficina de Personal	Gestión de Personal	Ajustes de omisiones en el parte diario
CU07	RFS07	Empleado	Comunicaciones Internas	Envío de memorándums entre empleados y áreas
CU08	RFS08	Jefatura de Servicio, Oficina de Personal	Reportes y Consultas	Generación de reportes de horas trabajadas
CU09	RFS09	Sistema (es un proceso interno)	Notificaciones y Alertas	Alertas automáticas de fechas y eventos importantes
CU10	RFS10	Empleado, Jefatura de Servicio, Oficina de Personal, Dirección Hospital	Seguridad y Control de Acceso	Control de acceso basado en rol del usuario

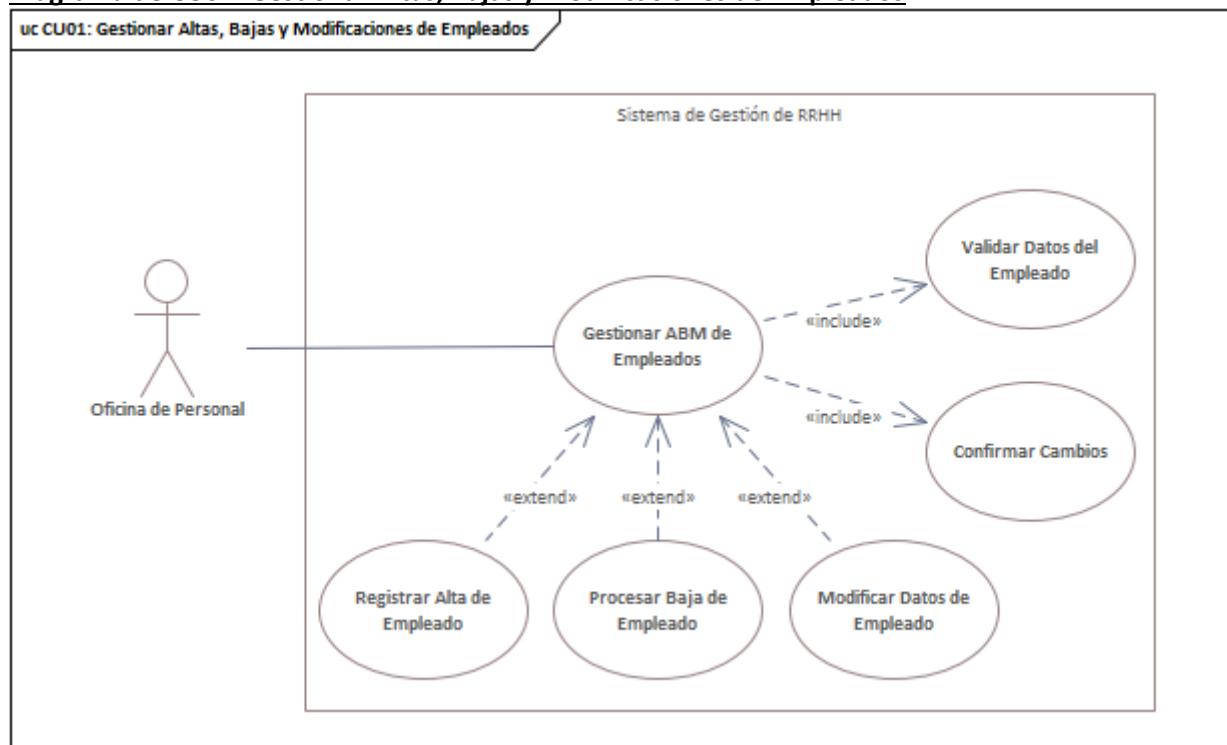
Diagrama de CU01: Gestionar Altas, Bajas y Modificaciones de Empleados

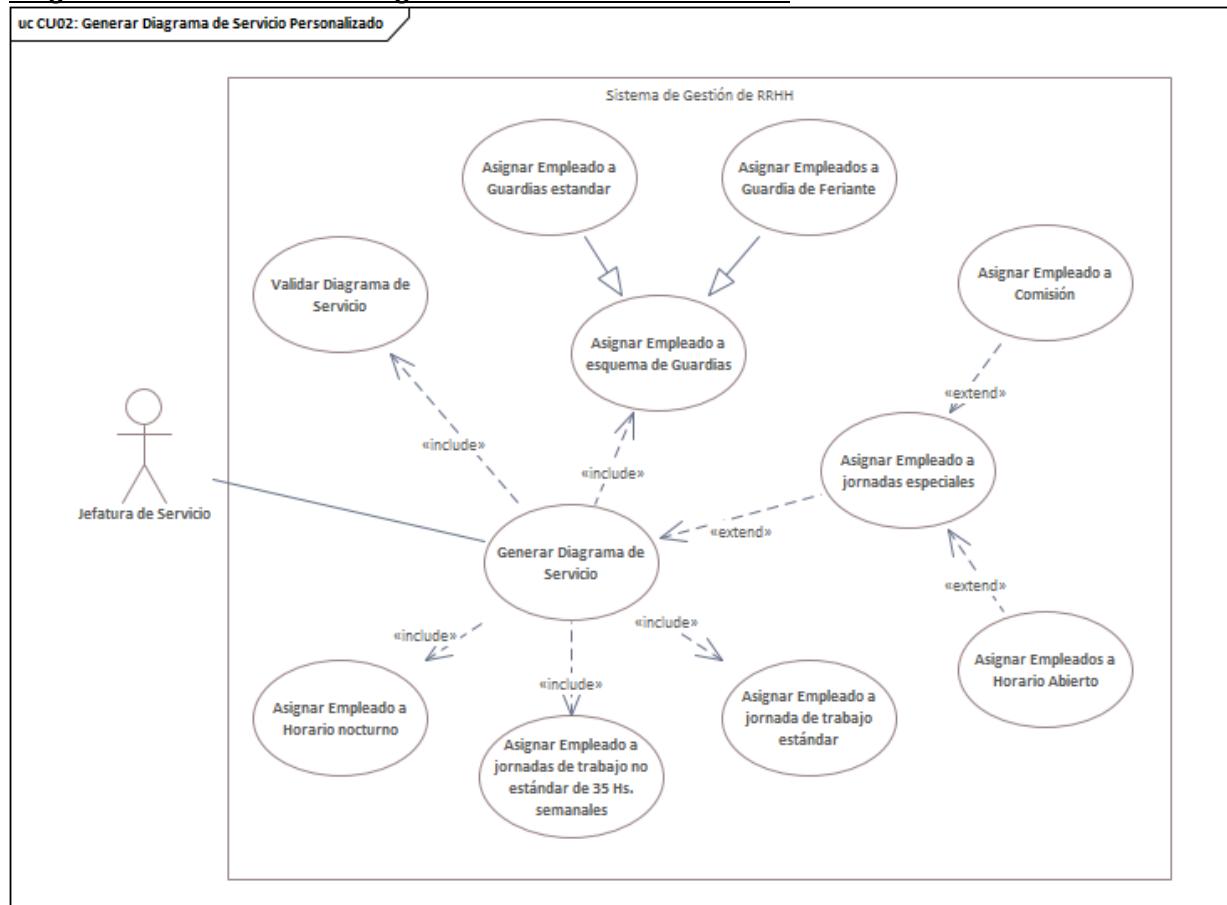
Diagrama de CU02: Generar Diagrama de Servicio Personalizado

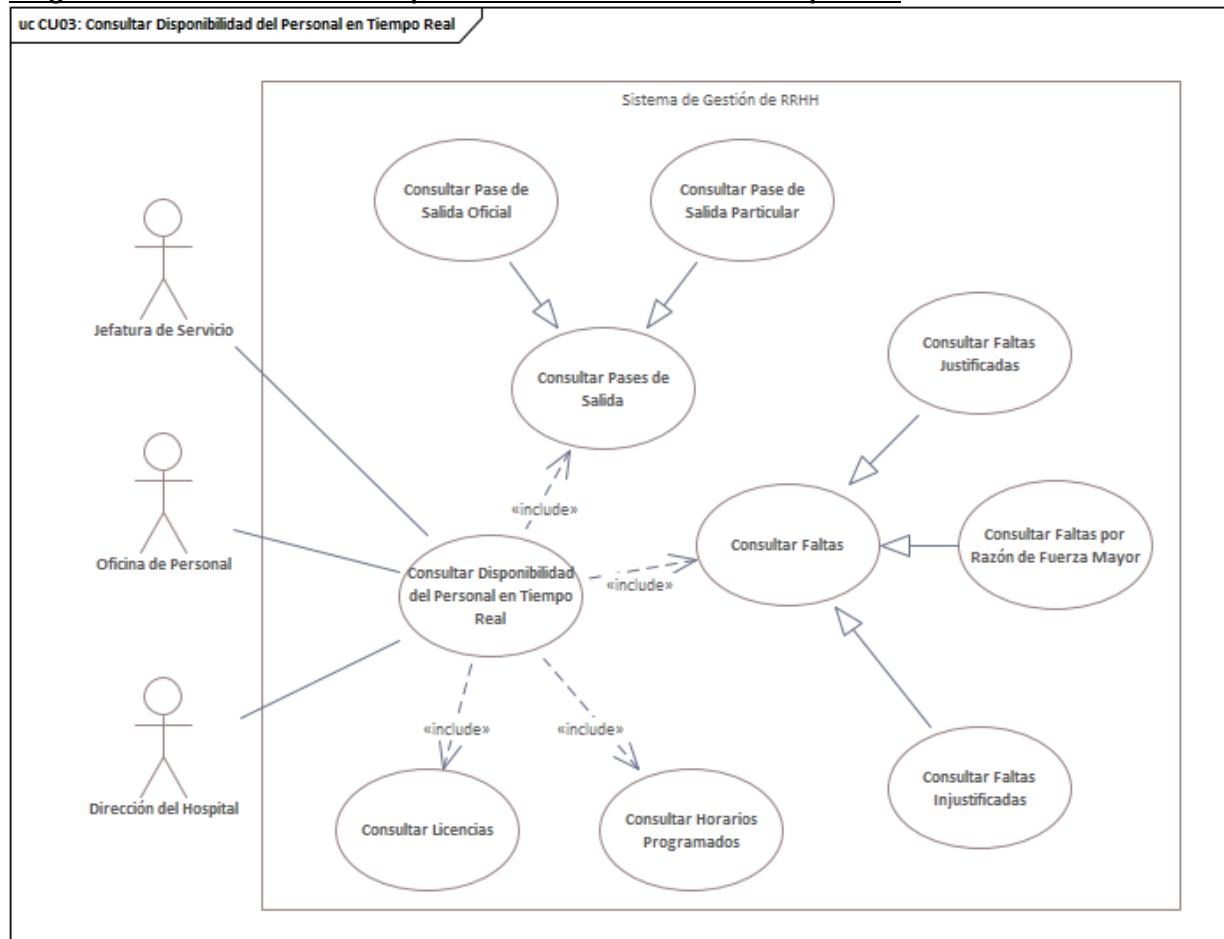
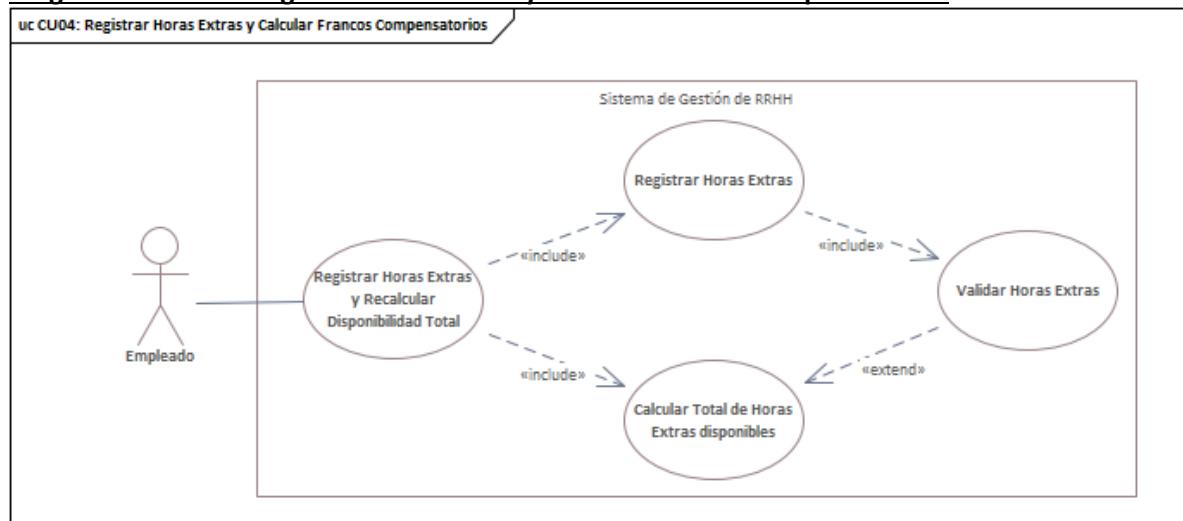
Diagrama de CU03: Consultar Disponibilidad del Personal en Tiempo Real**Diagrama de CU04: Registrar Horas Extras y Calcular Francos Compensatorios**

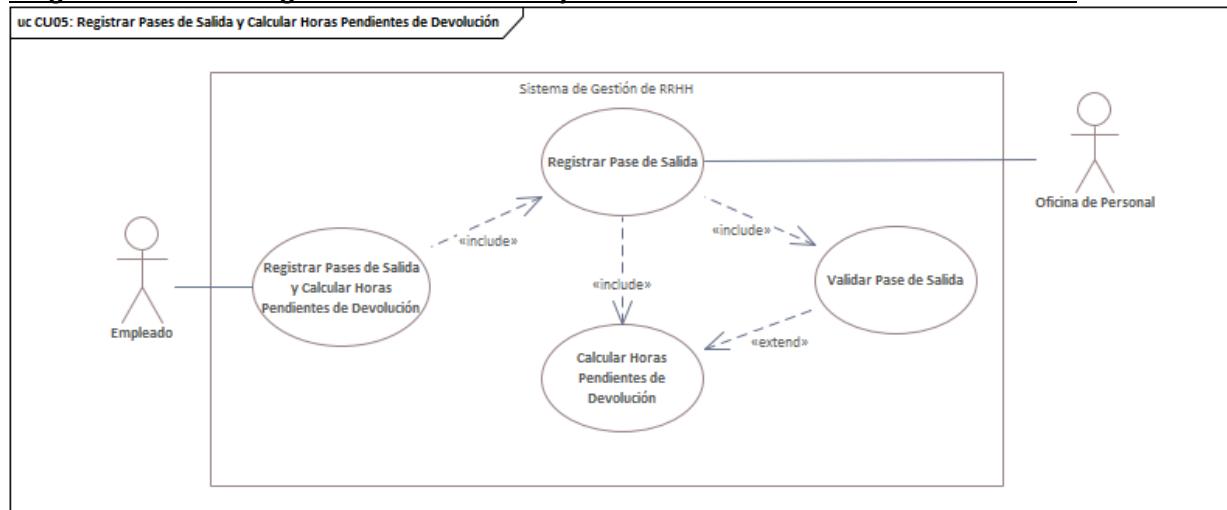
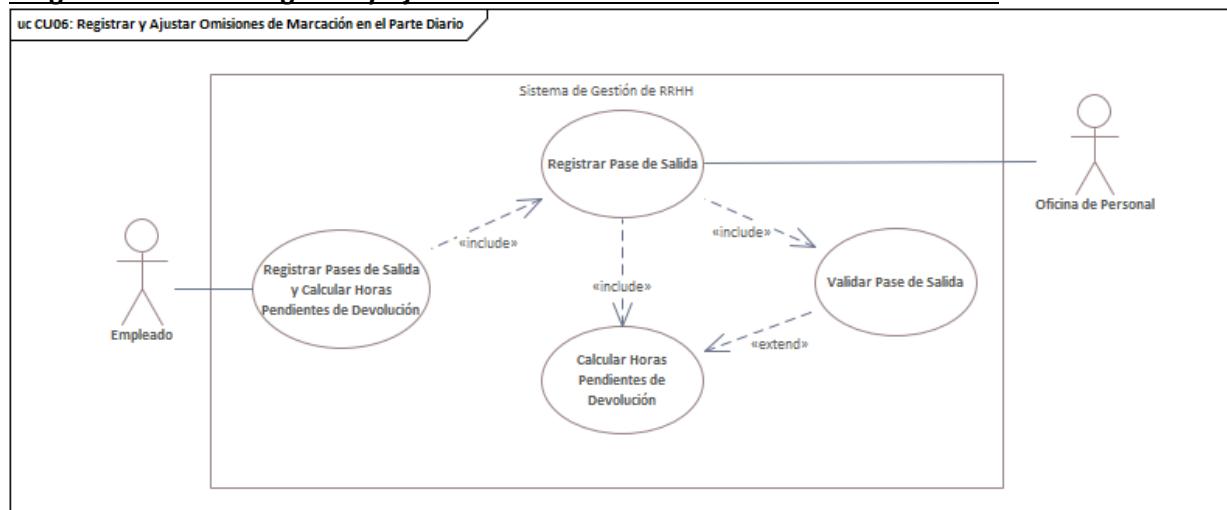
Diagrama de CU05: Registrar Pases de Salida y Calcular Horas Pendientes de Devolución**Diagrama de CU06: Registrar y Ajustar Omisiones de Marcación en el Parte Diario**

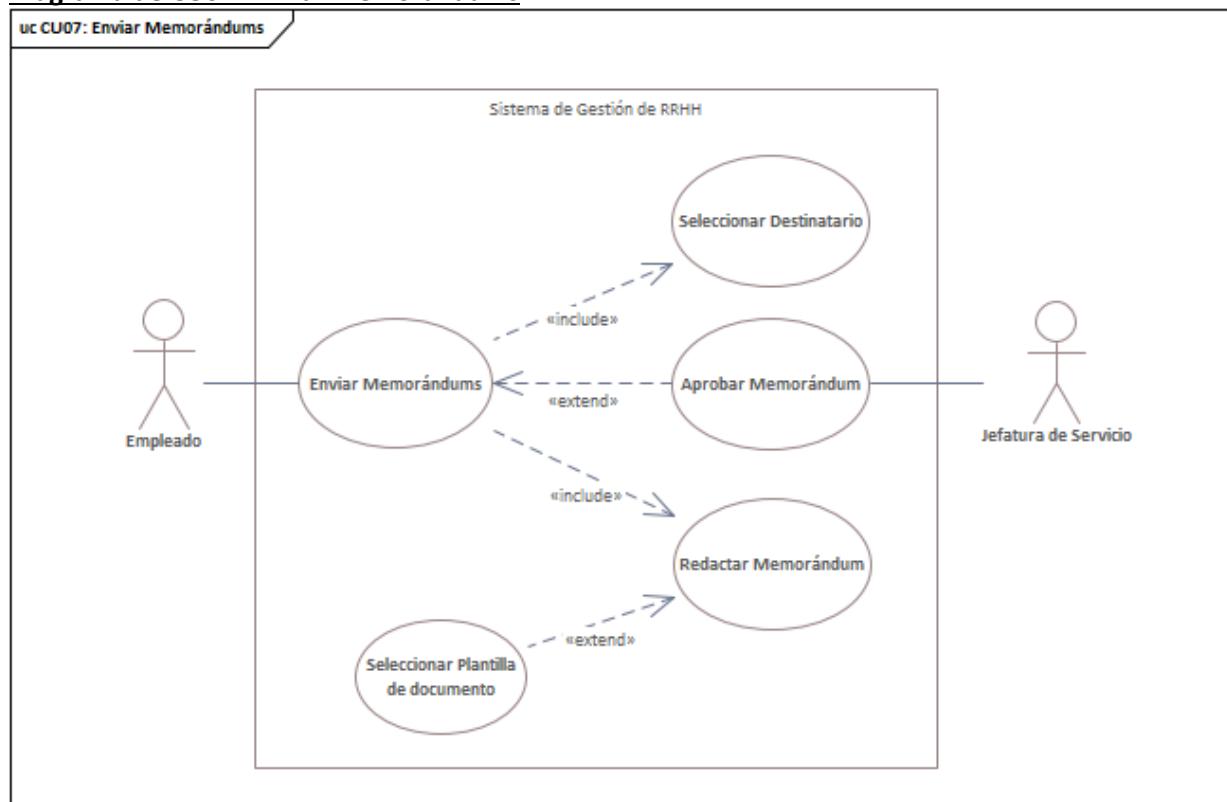
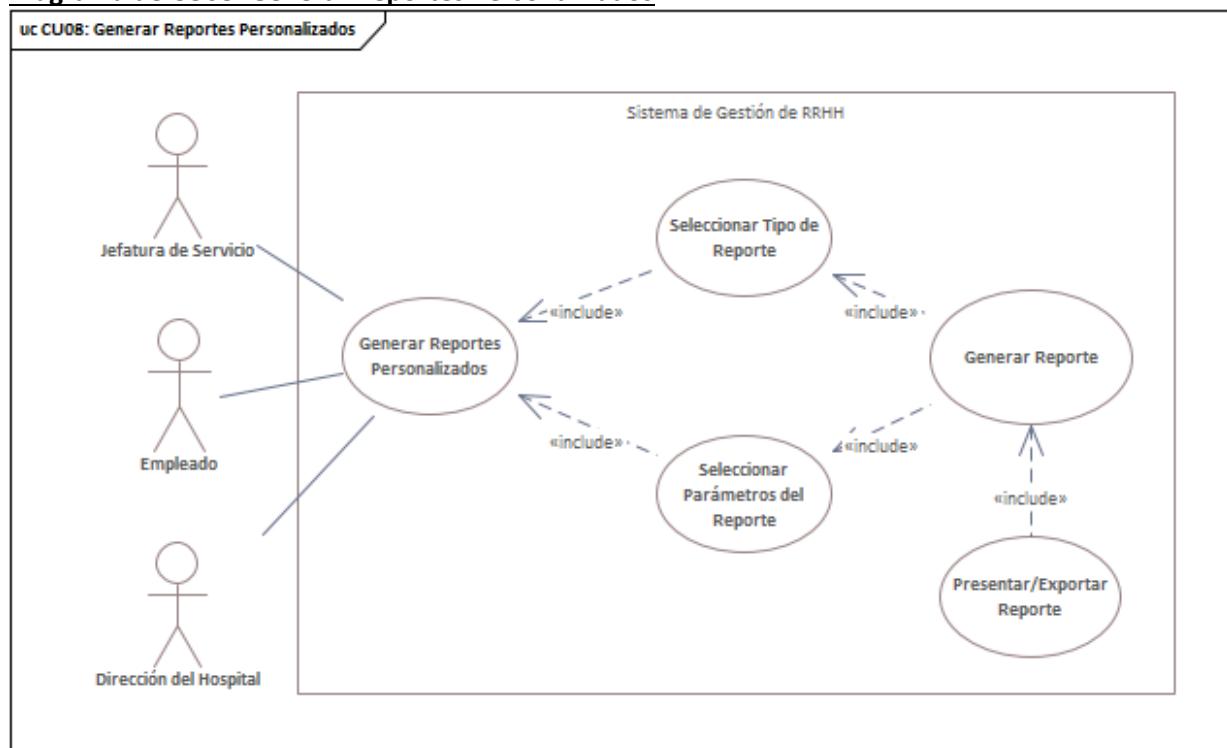
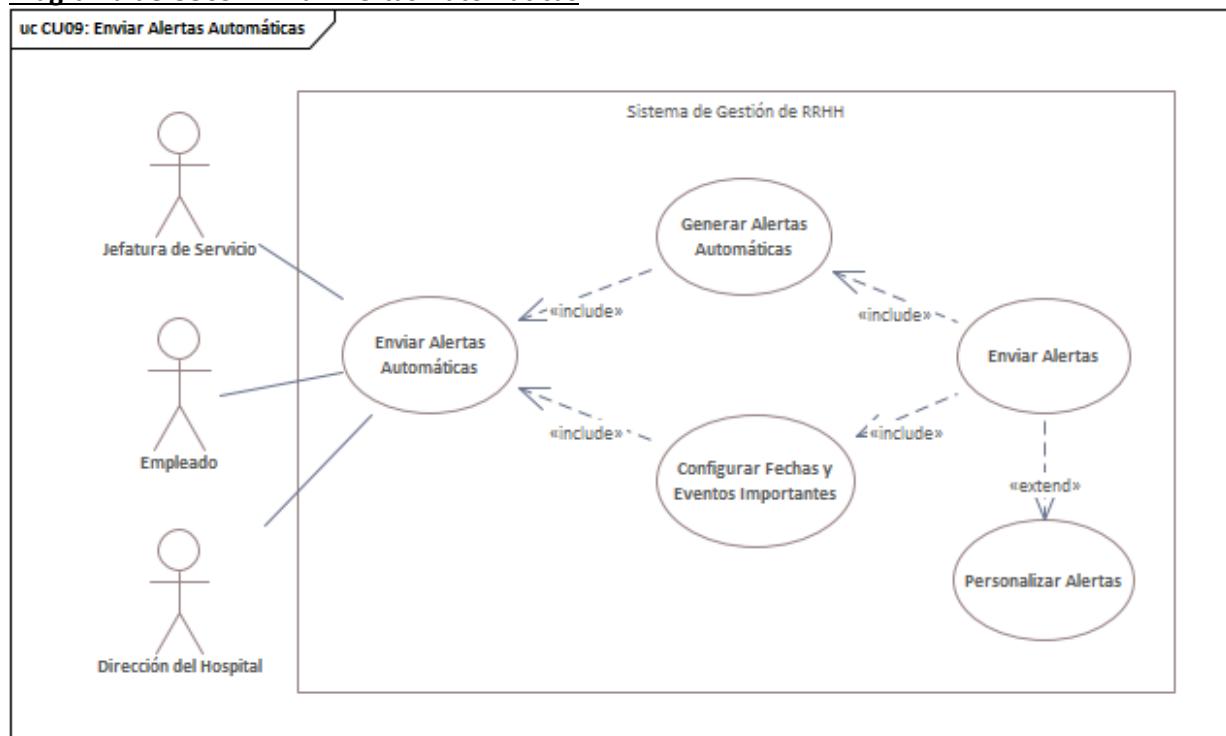
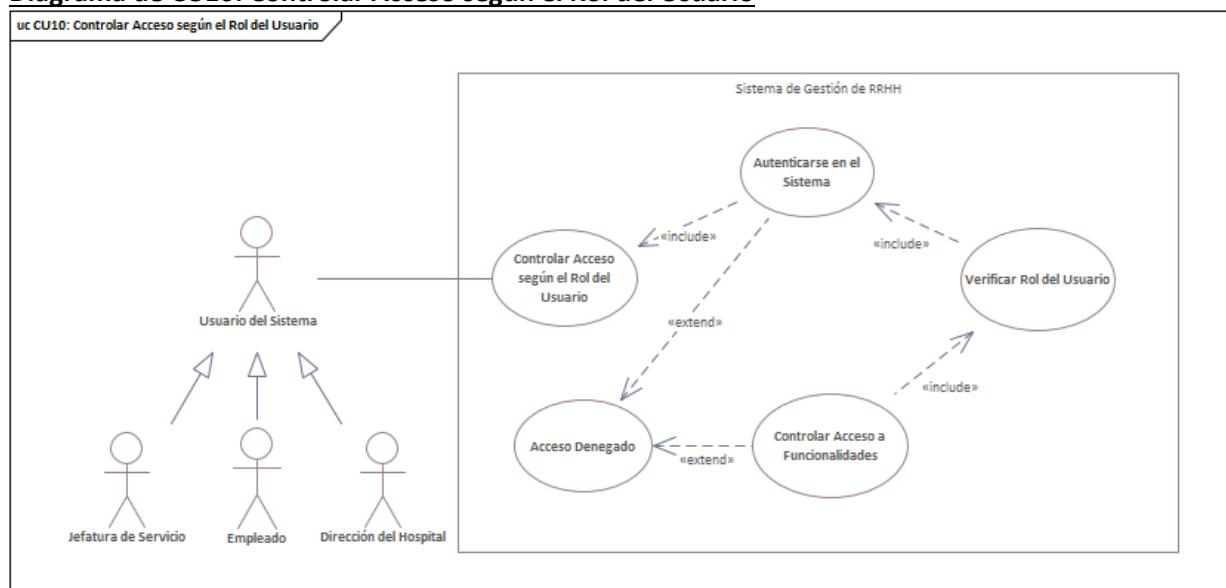
Diagrama de CU07: Enviar Memorándums**Diagrama de CU08: Generar Reportes Personalizados**

Diagrama de CU09: Enviar Alertas Automáticas**Diagrama de CU10: Controlar Acceso según el Rol del Usuario**

8.2 Descripción de casos de uso

Se describen x casos de uso, elegidos por la interacción que muestran dentro de un contexto para lograr un propósito.

Descripción de CU01: Gestionar Altas, Bajas y Modificaciones de Empleados

Caso de uso	C001: Gestionar Altas, Bajas y Modificaciones de Empleados					
Nivel del CU:	<input type="checkbox"/> Negocio	<input checked="" type="checkbox"/> Sistema de Inf.	Prioridad:	<input type="checkbox"/> Baja	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media

Tipo de CU:	<input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto	Complejidad:	<input type="checkbox"/> Baja <input checked="" type="checkbox"/> Media <input type="checkbox"/> Alta
Referencia(s):	RF01		
Actor(es):	Oficina de Personal		
Descripción:	Permite a la Oficina de Personal gestionar las altas, bajas y modificaciones de los empleados en la BD centralizada del sistema. Las altas implican el registro de un nuevo empleado en el sistema, las bajas representan la eliminación o desactivación del empleado, y las modificaciones son cambios en la información del empleado (servicio, rol, etc.)		
Precondicion(es):	La Oficina de Personal debe haber iniciado sesión en el sistema con los permisos adecuados.		
Flujo principal:			
01	El usuario (Oficina de Personal) accede al módulo: «Gestión de Empleados».		
02	Selecciona si desea realizar una Alta, Baja o Modificación de un empleado.		
03	El sistema presenta el formulario correspondiente: <ul style="list-style-type: none"> Para Altas, se ingresan los datos del nuevo empleado (nombre, DNI, servicio, etc.). Para Bajas, se selecciona el empleado a dar de baja. Para Modificaciones, se elige el empleado y se modifican los datos necesarios. 		
04	El usuario confirma la operación.		
05	El sistema valida los datos ingresados.		
06	El sistema actualiza la base de datos con la acción correspondiente.		
07	El sistema genera una notificación al empleado afectado (en caso de alta o modificación) y al jefe del servicio correspondiente.		
08	Se genera un reporte interno para la Oficina de Personal con el registro de la acción realizada.		
09	Fin del Flujo principal.		
Postcondicion(es):	<ul style="list-style-type: none"> La base de datos de empleados se actualiza correctamente según la acción ejecutada (alta, baja o modificación). El empleado afectado y su jefatura de servicio reciben una notificación de los cambios realizados (si corresponde). 		
Flujo alternativo			
S03	El sistema presenta el formulario correspondiente: <ul style="list-style-type: none"> Para Altas, si el usuario no proporciona todos los datos requeridos, el sistema muestra un mensaje de error solicitando completar los campos faltantes antes de proceder. Para Bajas, si el empleado a dar de baja tiene información pendiente (como horas por devolver o procesos sin concluir), el sistema alerta al usuario y sugiere revisar esa información antes de continuar. Para Modificaciones, si los cambios introducen inconsistencias (por ejemplo, asignar un rol o servicio no permitido), el sistema alerta al usuario y sugiere las modificaciones adecuadas. 		
S04	Fin del Flujo alternativo		
Excepcion(es):	Si el sistema no tiene conexión con la base de datos, el proceso se interrumpe y se informa al usuario que no se pudo completar la acción, solicitando que reintente más tarde.		
Autor: Pablo Alejandro Hamann	Ultima Modificación: 13/09/2024		

Descripción de CU02: Generar Diagrama de Servicio Personalizado

Caso de uso		CU02: Generar Diagrama de Servicio Personalizado				
Nivel del CU:	<input type="checkbox"/> Negocio	<input checked="" type="checkbox"/> Sistema de Inf.	Prioridad:	<input type="checkbox"/> Baja	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Alta
Tipo de CU:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto	Complejidad:	<input type="checkbox"/> Baja	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Alta
Referencia(s):	RF02					
Actor(es):	Jefatura de Servicio					
Descripción:	Permite a la Jefatura de Servicio generar un diagrama de servicio personalizado para su área. El diagrama incluye la asignación de modalidades y horarios laborales a empleados, guardias específicos (estándar, feriante, etc.), teniendo en cuenta las necesidades del servicio, y de los propios empleados. El objetivo es proporcionar una planificación de horarios ajustada a las necesidades operativas de cada servicio, la cual será validada posteriormente por la Oficina de Personal y la Dirección del Hospital.					
Precondicion(es):	<ul style="list-style-type: none"> El usuario (Jefatura de Servicio) debe haber iniciado sesión con los permisos adecuados. Los empleados a asignar deben estar registrados en el sistema. El calendario de turnos y guardias para el mes debe estar disponible. 					
Flujo principal:						
01	El usuario (Jefatura de Servicio) accede al módulo: «Generar Diagrama de Servicio».					
02	Selecciona el mes y los días para los cuales generará el diagrama.					
03	El sistema muestra una lista de los empleados del servicio, junto con sus modalidades horarias.					
04	El usuario asigna a cada empleado los turnos y guardias disponibles: <ul style="list-style-type: none"> Jornada estándar (35 horas semanales). Guardias de feriante (sábados, domingos y feriados). Guardias nocturnas. Jornadas abiertas u horarios flexibles. 					
05	El usuario revisa y confirma el diagrama de servicio.					
06	El sistema valida la asignación de turnos (por ejemplo, evitando que se asignen guardias consecutivas o turnos que excedan las horas permitidas).					
07	El sistema guarda el diagrama de servicio en el sistema.					
08	El diagrama se envía a la Oficina de Personal para su revisión y aprobación.					
09	Fin del Flujo principal.					
Postcondicion(es):	<ul style="list-style-type: none"> El diagrama de servicio se guarda en el sistema y queda disponible para revisión y aprobación por la Oficina de Personal. Los empleados reciben notificación sobre su asignación de turnos y guardias una vez que el diagrama es aprobado. 					
Flujo alternativo						
S04	Si un empleado es asignado a un turno que viola las normas laborales (por ejemplo, guardias consecutivas o exceso de horas), el sistema alerta al usuario acerca del error, y sugiere las correcciones necesarias.					
S05	Si el usuario decide no confirmar el diagrama, el sistema ofrece la opción de guardar un borrador o cancelar la operación sin guardar cambios.					
S06	Fin del Flujo alternativo					

Excepcion(es): <ul style="list-style-type: none"> • Si el sistema no tiene acceso al calendario de turnos o no encuentra los datos de los empleados, la operación se cancela y se notifica al usuario que no puede proceder con la generación del diagrama. • Si el sistema detecta que no hay empleados disponibles en las fechas seleccionadas (por vacaciones, licencias, etc.), se notifica al usuario para que asigne los turnos en otras fechas o con otros empleados. 	
<i>Autor: Pablo Alejandro Hamann</i>	<i>Ultima Modificación: 13/09/2024</i>

Descripción de CU03: Consultar Disponibilidad del Personal en Tiempo Real

Caso de uso	CU03: Consultar Disponibilidad del Personal en Tiempo Real					
Nivel del CU:	<input type="checkbox"/> Negocio	<input checked="" type="checkbox"/> Sistema de Inf.	Prioridad:	<input type="checkbox"/> Baja	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Alta
Tipo de CU:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto	Complejidad:	<input type="checkbox"/> Baja	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Alta
Referencia(s):	RF03					
Actor(es):	<ul style="list-style-type: none"> • Oficina de Personal • Jefatura de Servicio • Dirección del Hospital 					
Descripción:	<p>Permite a los actores consultar la disponibilidad del personal en tiempo real, mostrando la asignación de horarios programados, licencias, faltas justificadas e injustificadas, así como los pases de salida oficiales y particulares. El objetivo es proporcionar una visión clara y actualizada sobre el estado del personal para la correcta toma de decisiones y la gestión de recursos humanos.</p>					
Precondicion(es):	<ul style="list-style-type: none"> • El usuario debe haber iniciado sesión en el sistema con los permisos adecuados. • Los datos de horarios, licencias, faltas y pases de salida deben estar actualizados en el sistema. 					
Flujo principal:						
01	El usuario (Oficina de Personal, Jefatura de Servicio o Dirección del Hospital) accede al módulo: «Consultar Disponibilidad del Personal».					
02	Selecciona el período de consulta (día, semana o mes).					
03	<p>El sistema muestra una lista del personal con la siguiente información:</p> <ul style="list-style-type: none"> • Diagramación horaria. • Licencias (vacaciones, enfermedades, etc.). • Faltas justificadas e injustificadas (y otras). • Pases de salida (oficiales y/o particulares). 					
04	El usuario puede filtrar los datos según el tipo de información que desea visualizar (licencias, faltas, pases de salida, etc.).					
05	El sistema actualiza en tiempo real los cambios que se produzcan, como la aprobación de una licencia o la justificación de una falta.					
06	El usuario puede generar un reporte de disponibilidad en el período consultado y exportarlo en formato PDF o Excel si es necesario.					
07	Fin del Flujo principal.					
Postcondicion(es):	<ul style="list-style-type: none"> • Se muestra la disponibilidad del personal en tiempo real, permitiendo a los usuarios tomar decisiones basadas en la información presentada. 					

	<ul style="list-style-type: none"> Los reportes de disponibilidad pueden generarse y exportarse si es necesario.
Flujo alternativo	
S03	Si el usuario intenta consultar la disponibilidad de un empleado que no tiene información registrada para el período seleccionado, el sistema muestra un mensaje indicando que no se encontraron registros para dicho empleado en las fechas consultadas.
S05	Si existen solicitudes de licencias, faltas o pases que aún no han sido aprobadas o revisadas, el sistema muestra un mensaje indicando que la información está pendiente de actualización y proporciona la opción de continuar la consulta con los datos disponibles.
S06	Fin del Flujo alternativo
Excepcion(es):	<ul style="list-style-type: none"> Si el sistema no tiene acceso a la BD o no puede cargar la información en tiempo real, se notifica al usuario con un mensaje de error y se sugiere reintentar más tarde. Si el sistema detecta un conflicto de datos (por ejemplo, un empleado registrado en dos turnos horarios simultáneamente), se alerta al usuario para que corrija el conflicto antes de continuar.
<i>Autor: Pablo Alejandro Hamann</i>	
<i>Ultima Modificación: 13/09/2024</i>	

Descripción de CU04: Registrar Horas Extras

Caso de uso	CU04: Registrar Horas Extras y Calcular Francos Compensatorios					
Nivel del CU:	<input type="checkbox"/> Negocio	<input checked="" type="checkbox"/> Sistema de Inf.	Prioridad:	<input type="checkbox"/> Baja	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Alta
Tipo de CU:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto	Complejidad:	<input type="checkbox"/> Baja	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Alta
Referencia(s):	RF04					
Actor(es):	<ul style="list-style-type: none"> Empleado Jefatura de Servicio Oficina de Personal 					
Descripción:	Permite a los empleados registrar sus horas extras en el sistema (necesarias para la posterior confección de un franco compensatorio). Las horas extras ingresadas deben ser autorizadas por la Jefatura de servicio, y luego validadas por la Oficina de Personal. Esto garantiza que las mismas se encuentren en regla, de acuerdo a la normativa laboral del hospital, y al régimen laboral del propio empleado.					
Precondicion(es):	<ul style="list-style-type: none"> El empleado debe haber iniciado sesión en el sistema con sus credenciales. Las horas de trabajo regulares del empleado deben estar registradas en el sistema. 					
Flujo principal:						
01	El empleado accede al módulo: «Registrar Horas Extras».					
02	Selecciona la fecha y el horario en el que realizó las horas extras.					
03	Introduce la cantidad de horas extras trabajadas y confirma el registro.					
04	El sistema valida automáticamente si las horas extras cargadas, corresponden a un período de tiempo con marcación horaria registrada por el empleado.					
05	El sistema marca las horas extras como "Pendientes de validación".					
06	La Oficina de Personal recibe una notificación para validar las horas extras.					

07	La Oficina de Personal accede al módulo de validación, revisa las horas extras y las aprueba o rechaza.
08	Si las horas extras son aprobadas, el sistema marca las horas como aprobadas y las mismas quedan a disponibilidad para ser usadas en un franco compensatorio a futuro. NOTA: el FC no se calcula en el momento, debido a que depende de la jornada laboral del empleado de la solicitud.
09	El sistema registra las horas extra en el perfil del empleado.
10	El empleado recibe una notificación acerca de que ha sumado horas extras a su legajo, y le informa el nuevo total de horas extras disponibles para usar en la confección de FC.
11	Fin del Flujo principal.
Postcondicion(es):	<ul style="list-style-type: none"> Las horas extras quedan registradas y pendientes de validación por la Oficina de Personal. Una vez validadas, las horas extra totales se recalculan nuevamente y se reflejan en el perfil del empleado.
Flujo alternativo	
S04	Si el empleado ingresa horas extra que no coinciden con los turnos registrados, el sistema alerta al usuario indicando la inconsistencia y solicita que revise los datos ingresados.
S08	Si la Oficina de Personal rechaza las horas extras ingresadas, el sistema notifica al empleado indicando el motivo del rechazo y solicita que se pongan en contacto con la Oficina de Personal si es necesario.
S09	Fin del Flujo alternativo
Excepcion(es):	<ul style="list-style-type: none"> Si el sistema no tiene acceso a la BD para registrar o validar las horas extras, se notifica al empleado y a la Oficina de Personal, y el registro queda marcado como pendiente para ser procesado cuando el sistema esté operativo. Si se detectan inconsistencias en la suma de horas trabajadas (por ejemplo, horas extras que exceden los límites permitidos por la normativa), se alerta al usuario y no se permite continuar hasta que la situación sea corregida.
<i>Autor: Pablo Alejandro Hamann</i>	<i>Ultima Modificación: 30/09/2024</i>

Descripción de CU05: Registrar Pases de Salida y Calcular Horas Pendientes de Devolución

Caso de uso	CU05: Registrar Pases de Salida y Calcular Horas Pendientes de Devolución					
Nivel del CU:	<input type="checkbox"/> Negocio	<input checked="" type="checkbox"/> Sistema de Inf.	Prioridad:	<input type="checkbox"/> Baja	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Alta
Tipo de CU:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto	Complejidad:	<input type="checkbox"/> Baja	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Alta
Referencia(s):	RF05					
Actor(es):	<ul style="list-style-type: none"> Empleado Oficina de Personal Jefatura de Servicio 					
Descripción:	Permite a los empleados registrar sus pases de salida (oficiales y/o particulares), para luego calcular automáticamente las horas pendientes de devolución. La Oficina de Personal y la Jefatura de Servicio validan estos pases antes de que el sistema registre las horas correspondientes para devolver. El sistema también realiza un seguimiento de las horas					

		pendientes para garantizar que el empleado devuelva las horas dentro de los plazos establecidos.
Precondicion(es):		<ul style="list-style-type: none"> • El empleado debe haber iniciado sesión en el sistema con sus credenciales. • La solicitud de pase de salida debe cumplir con las políticas del hospital.
Flujo principal:		
01	El empleado accede al módulo: « Registrar Pase de Salida ».	
02	Selecciona si el pase de salida es Oficial o Particular.	
03	Introduce la fecha y la hora de salida y regreso (si aplica).	
04	El empleado confirma el registro del pase de salida.	
05	El sistema marca el pase como "Pendiente de Validación".	
06	La Jefatura de Servicio revisa el pase y lo aprueba o rechaza: <ul style="list-style-type: none"> • Si es aprobado, el pase se envía a la Oficina de Personal para su registro. • Si es rechazado, el sistema notifica al empleado con los motivos del rechazo. 	
07	La Oficina de Personal verifica el pase aprobado y calcula automáticamente las horas pendientes de devolución, en función del tiempo que el empleado estuvo fuera.	
08	El sistema actualiza las horas pendientes en el perfil del empleado y le envía una notificación indicando cuántas horas debe devolver y el plazo correspondiente.	
09	Fin del Flujo principal.	
Postcondicion(es):		<ul style="list-style-type: none"> • El pase de salida queda registrado en el sistema y queda pendiente de validación por parte de la Jefatura de Servicio o la Oficina de Personal. • Las horas pendientes de devolución se calculan y se registran para seguimiento una vez aprobado el pase.
Flujo alternativo		
S03	Si el empleado ingresa una solicitud de pase con fechas o horas incorrectas (por ejemplo, una salida futura que excede el límite permitido, o una salida sin retorno superior a las 3 Hs.), el sistema alerta al usuario y solicita corregir la información antes de continuar.	
S07	Si la Jefatura de Servicio o la Oficina de Personal rechazan el pase de salida, el sistema notifica al empleado y le ofrece la opción de corregir o anular la solicitud.	
S08	Fin del Flujo alternativo	
Excepcion(es):		<ul style="list-style-type: none"> • Si el sistema no puede acceder a la BD datos para registrar el pase de salida, alerta al empleado y le recomienda reintentar más tarde. El sistema guarda la solicitud como pendiente para ser procesada cuando esté disponible el acceso a la BD. • Si se detecta que el empleado ya tiene horas pendientes de devolución de un pase anterior, el sistema puede emitir una advertencia para que el empleado devuelva las horas antes de registrar un nuevo pase.
Autor: Pablo Alejandro Hamann		Última Modificación: 13/09/2024

Descripción de CU06: Registrar y Ajustar Omisiones de Marcación en el Parte Diario

Caso de uso	CU06: Registrar y Ajustar Omisiones de Marcación en el Parte Diario					
Nivel del CU:	<input type="checkbox"/> Negocio	<input checked="" type="checkbox"/> Sistema de Inf.	Prioridad:	<input type="checkbox"/> Baja	<input type="checkbox"/> Media	<input type="checkbox"/> Alta

Tipo de CU:	<input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto	Complejidad:	<input type="checkbox"/> Baja <input checked="" type="checkbox"/> Media <input type="checkbox"/> Alta Media
Referencia(s):	RF06		
Actor(es):	<ul style="list-style-type: none"> • Empleado • Oficina de Personal • Jefatura de Servicio 		
Descripción:	Permite a los empleados y a la Oficina de Personal registrar y ajustar las omisiones de marcación de entrada o salida en el sistema de control horario. Las omisiones pueden deberse a fallos del sistema o errores humanos al no marcar adecuadamente los registros de entrada o salida. La Oficina de Personal se encarga de validar los ajustes, garantizando que las horas registradas sean correctas antes de actualizar el parte diario.		
Precondicion(es):	<ul style="list-style-type: none"> • El empleado debe haber iniciado sesión en el sistema con sus credenciales. • El parte diario debe estar disponible y accesible para su edición. 		
Flujo principal:			
01	El empleado accede al módulo: « Registrar Omisión de Marcación ».		
02	Selecciona el tipo de omisión que desea registrar: “Omisión de entrada” u “Omisión de salida”.		
03	Introduce la fecha y la hora en que debería haberse realizado la marcación.		
04	El empleado confirma la solicitud de ajuste de la omisión.		
05	El sistema marca la solicitud como "Pendiente de Validación".		
06	La Oficina de Personal recibe la solicitud y la revisa: <ul style="list-style-type: none"> • Si la omisión es válida, la Oficina de Personal aprueba la solicitud y añade la novedad al parte diario. • Si la omisión no es válida, la solicitud es rechazada y se notifica al empleado. 		
08	El sistema actualiza el parte diario y genera un registro de auditoría con los ajustes realizados.		
09	Fin del Flujo principal.		
Postcondicion(es):	<ul style="list-style-type: none"> • El sistema registra la corrección de la omisión de marcación en el parte diario. • La Oficina de Personal actualiza el parte diario con los datos ajustados y notifica a la Jefatura de Servicio. 		
Flujo alternativo			
S03	Si el empleado ingresa una fecha o una hora que no se corresponde al de horario diagramado, el sistema alerta al usuario indicando la inconsistencia y le solicita corregir la información.		
S08	Si la Jefatura de Servicio o la Oficina de Personal rechazan la solicitud de ajuste, el sistema notifica al empleado con los motivos del rechazo y le permite realizar las correcciones necesarias o retirar la solicitud.		
S09	Fin del Flujo alternativo		
Excepcion(es):	<ul style="list-style-type: none"> • Si el sistema no puede acceder al parte diario o si este ya ha sido cerrado para modificaciones, se notifica al empleado y a la Oficina de Personal, y la solicitud de ajuste queda pendiente hasta que el parte se vuelva a abrir para edición. 		

	<ul style="list-style-type: none"> Si existen inconsistencias graves en los datos ingresados (como omisiones repetidas o recurrentes), se alerta al usuario y se solicita la revisión por parte de la Oficina de Personal antes de continuar.
Autor: Pablo Alejandro Hamann	Última Modificación: 13/09/2024

Descripción de CU07: Enviar Memorándums

Caso de uso CU07: Enviar Memorándums						
Nivel del CU:	<input type="checkbox"/> Negocio	<input checked="" type="checkbox"/> Sistema de Inf.	Prioridad:	<input type="checkbox"/> Baja	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Alta
Tipo de CU:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto	Complejidad:	<input checked="" type="checkbox"/> Baja	<input type="checkbox"/> Media	<input type="checkbox"/> Alta
Referencia(s):	RF07					
Actor(es):	<ul style="list-style-type: none"> Empleado Jefatura de Servicio Oficina de Personal Dirección del Hospital 					
Descripción:	Permite a los empleados enviar memorándums a sus jefaturas de servicio, a la Oficina de Personal o a la Dirección del Hospital. Los memorándums pueden incluir solicitudes, notificaciones o comunicaciones internas relacionadas con la gestión del personal o con temas laborales. Las jefaturas y la Oficina de Personal pueden revisar y responder los memorándums recibidos.					
Precondicion(es):	<ul style="list-style-type: none"> El empleado debe haber iniciado sesión en el sistema con los permisos adecuados. La jefatura de servicio, Oficina de Personal o Dirección deben estar registrados en el sistema para recibir memorándums. 					
Flujo principal:						
01	El empleado accede al módulo: «Enviar Memorándum».					
02	Selecciona el destinatario del memorándum (Jefatura de Servicio, Oficina de Personal o Dirección del Hospital).					
03	El empleado redacta el memorándum en el sistema, indicando el asunto y el cuerpo del mensaje.					
04	El empleado confirma el envío del memorándum.					
05	El sistema envía el memorándum al destinatario seleccionado.					
06	El destinatario recibe una notificación indicando que ha recibido un nuevo memorándum.					
07	El destinatario puede acceder al módulo de Consultar Memorándums para leer el memorándum y enviar una respuesta si es necesario.					
08	El sistema almacena el memorándum enviado y la respuesta (si aplica) en el perfil del empleado.					
09	Fin del Flujo principal.					
Postcondicion(es):	<ul style="list-style-type: none"> El memorándum queda registrado en el sistema y enviado al destinatario correspondiente. El destinatario recibe una notificación del memorándum recibido. 					
Flujo alternativo						

S02	Si el empleado selecciona un destinatario incorrecto o no autorizado para recibir memorándums, el sistema alerta al usuario y solicita que seleccione el destinatario adecuado antes de continuar.
S03	Fin del Flujo alternativo

- Excepcion(es):**
- Si el sistema no puede enviar el memorándum debido a problemas de conexión o fallo del sistema, el memorándum queda marcado como "Pendiente de envío" y se reintentará su envío cuando el sistema esté disponible.
 - Si el destinatario no está disponible (por ejemplo, si no se encuentra registrado en el sistema o tiene el buzón lleno), el sistema notifica al usuario y ofrece la opción de guardar el memorándum como borrador o seleccionar otro destinatario.

Autor: Pablo Alejandro Hamann

Ultima Modificación: 13/09/2024

Descripción de CU08: Generar Reportes Personalizados

Caso de uso	CU08: Generar Reportes Personalizados					
Nivel del CU:	<input type="checkbox"/> Negocio	<input checked="" type="checkbox"/> Sistema de Inf.	Prioridad:	<input type="checkbox"/> Baja	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Alta
Tipo de CU:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto	Complejidad:	<input type="checkbox"/> Baja	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Alta
Referencia(s):	RF08					
Actor(es):	<ul style="list-style-type: none"> • Empleado • Jefatura de Servicio • Oficina de Personal • Dirección del Hospital 					
Descripción:	Permite a los actores (empleados, jefaturas de servicio, Oficina de Personal y Dirección del Hospital) generar reportes personalizados sobre diferentes aspectos de la gestión de personal. Los reportes pueden incluir información sobre horas trabajadas, francos compensatorios, licencias, cumplimiento de horarios, entre otros. El sistema ofrecerá opciones de personalización de los reportes según el rol del usuario.					
Precondicion(es):	<ul style="list-style-type: none"> • El usuario debe haber iniciado sesión en el sistema con los permisos adecuados. • Los datos a ser reportados deben estar actualizados y disponibles en la base de datos. 					
Flujo principal:						
01	El usuario accede al módulo: «Generar Reportes Personalizados».					
02	Selecciona el tipo de reporte que desea generar, según las opciones disponibles para su rol: <ul style="list-style-type: none"> • Empleado: Reportes de horas trabajadas, licencias, inasistencias, omisiones y horas disponibles p/Francos Compensatorios. • Jefatura de Servicio: Reportes de cumplimiento de horarios y asignaciones de guardias. • Oficina de Personal: Reportes del parte diario, ausentismos y novedades. • Dirección del Hospital: Reportes de cumplimiento por servicio. Gráficos comparativos entre servicios (ranking de los servicios más y menos eficientes) 					

03	El sistema solicita al usuario seleccionar los parámetros del reporte (período de tiempo, tipo de información, filtros adicionales).
04	El usuario confirma los parámetros seleccionados.
05	El sistema genera el reporte basado en los datos disponibles.
06	El sistema presenta el reporte generado en formato digital.
07	El usuario tiene la opción de exportar el reporte en diferentes formatos (PDF, Excel) o guardarlo en su perfil para futuras consultas.
08	Fin del Flujo principal.
Postcondicion(es):	<ul style="list-style-type: none"> • El reporte es generado y presentado al usuario en formato digital. • El usuario tiene la opción de exportar o guardar el reporte en diversos formatos (PDF, Excel, etc.).
Flujo alternativo	
S03	Si el usuario introduce parámetros inválidos o inconsistentes (por ejemplo, un rango de fechas incorrecto), el sistema alerta al usuario y le solicita corregir los parámetros antes de continuar con la generación del reporte.
S04	Fin del Flujo alternativo
Excepcion(es):	<ul style="list-style-type: none"> • Si el sistema no tiene acceso a los datos necesarios para generar el reporte, se muestra un mensaje de error al usuario y se ofrece la opción de reintentar más tarde. • Si el sistema detecta una falta de datos (por ejemplo, no hay información suficiente para generar un reporte en el período solicitado), el sistema se lo notifica al usuario y ofrece la opción de ajustar los parámetros o cambiar el tipo de reporte.
Autor: Pablo Alejandro Hamann	
Ultima Modificación: 13/09/2024	

Descripción de CU09: Enviar Alertas Automáticas

Caso de uso	CU09: Enviar Alertas Automáticas					
Nivel del CU:	<input type="checkbox"/> Negocio	<input checked="" type="checkbox"/> Sistema de Inf.	Prioridad:	<input type="checkbox"/> Baja	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Alta
Tipo de CU:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto	Complejidad:	<input checked="" type="checkbox"/> Baja	<input type="checkbox"/> Media	<input type="checkbox"/> Alta
Referencia(s):	RF09					
Actor(es):	<ul style="list-style-type: none"> • Oficina de Personal • Jefatura de Servicio • Dirección del Hospital • Empleado (recibe alertas) 					
Descripción:	Permite al sistema enviar alertas automáticas a los empleados, jefaturas de servicio, Oficina de Personal y Dirección del Hospital para recordar eventos importantes y fechas límite. Las alertas pueden incluir recordatorios sobre la presentación de diagramas de servicio, vencimiento de licencias, horarios pendientes de devolución de pases de salida, entre otros. El envío de las alertas es gestionado automáticamente por el sistema, con base en reglas preconfiguradas y eventos que disparan las alertas.					
Precondicion(es):	<ul style="list-style-type: none"> • El sistema debe tener acceso a los datos y fechas relevantes de los empleados y jefaturas. 					

	<ul style="list-style-type: none"> Los usuarios deben estar registrados en el sistema y tener un canal de notificación habilitado (correo electrónico, SMS, etc.).
Flujo principal:	
01	El sistema detecta un evento o una fecha límite configurada (por ejemplo, plazo para presentar el diagrama de servicio).
02	El sistema genera automáticamente la alerta correspondiente.
03	El sistema selecciona a los destinatarios adecuados para la alerta (<i>Empleado, Jefatura de Servicio, Oficina de Personal, Dirección del Hospital</i>).
04	El sistema envía la alerta por el canal de notificación configurado (correo electrónico, SMS, notificación en la plataforma, etc.).
05	Los usuarios destinatarios reciben la alerta y, si corresponde, toman acción en relación con el contenido de la alerta (presentar un diagrama de servicio, ajustar un horario, etc.).
06	El sistema registra el envío de la alerta y almacena un registro del mismo para su seguimiento y auditoría.
07	Fin del Flujo principal.
Postcondicion(es):	<ul style="list-style-type: none"> Se envían las alertas correspondientes a los usuarios según los eventos o fechas límite configuradas. El sistema actualiza el estado de las alertas enviadas y almacena el registro del envío para su trazabilidad.
Flujo alternativo	
S04	Si el canal de notificación (correo electrónico, SMS) no está disponible o no es accesible, el sistema reintentará enviar la alerta más tarde o enviará la alerta por un canal alternativo si está disponible.
S05	Fin del Flujo alternativo
Excepcion(es):	<ul style="list-style-type: none"> Si el sistema no puede acceder a la BD para obtener la información necesaria para enviar la alerta, el envío se marca como pendiente y se reintentará una vez que los datos estén disponibles. Si el usuario no tiene configurado un canal de notificación, el sistema genera un mensaje de error interno y lo notifica a la Oficina de Personal para que actualice los datos de contacto del usuario.
Autor: Pablo Alejandro Hamann	
Última Modificación: 13/09/2024	

Descripción de CU10: Controlar Acceso según el Rol del Usuario

Caso de uso	CU10: Controlar Acceso según el Rol del Usuario					
Nivel del CU:	<input type="checkbox"/> Negocio	<input checked="" type="checkbox"/> Sistema de Inf.	Prioridad:	<input type="checkbox"/> Baja	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Alta
Tipo de CU:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto	Complejidad:	<input type="checkbox"/> Baja	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Alta
Referencia(s):	RF10					
Actor(es):	<ul style="list-style-type: none"> Empleado Jefatura de Servicio Oficina de Personal Dirección del Hospital Administrador del Sistema 					
Descripción:	Describe cómo el sistema controla el acceso a las diferentes funcionalidades según el rol asignado a cada usuario. Los roles incluyen					

	<p>“Empleado”, “Jefatura de Servicio”, “Oficina de Personal”, “Dirección del Hospital” y “Administrador del Sistema”. Cada rol tiene permisos específicos para acceder a determinadas áreas del sistema y realizar ciertas acciones. El sistema garantiza que los usuarios solo puedan ver y modificar la información para la que tienen autorización, acorde a su rol.</p>
Precondicion(es):	<ul style="list-style-type: none"> • El usuario debe estar registrado en el sistema y haber iniciado sesión con su rol asignado. • El sistema debe tener definidos los permisos correspondientes a cada rol.
Flujo principal:	
01	El usuario accede al sistema e ingresa sus credenciales de autenticación.
02	El sistema verifica las credenciales y autentica al usuario.
03	El sistema asigna el rol correspondiente al usuario autenticado (Empleado, Jefatura de Servicio, Oficina de Personal, Dirección del Hospital, Administrador del Sistema).
04	<p>Según el rol asignado, el sistema permite o restringe el acceso a determinadas funcionalidades:</p> <ul style="list-style-type: none"> • Empleado: Puede consultar sus horarios, sus novedades (licencias, faltas, horas disponibles p/ francos compensatorios, además de enviar memorándums). • Jefatura de Servicio: Puede generar diagramas de servicio, validar pases de salida, aprobar solicitudes de sus empleados a cargo, y consultar disponibilidad del personal bajo su supervisión. • Oficina de Personal: Puede gestionar el ABM de empleados, aprobar/rechazar diagramas de servicio, validar y registrar omisiones de E/S, generar reportes detallados, entre otros. • Dirección del Hospital: Puede generar reportes globales, aprobar/rechazar diagramas de servicio y gestionar altos niveles de administración. • Administrador del Sistema: Puede gestionar roles de usuario y configurar los permisos del sistema.
05	El sistema permite al usuario acceder únicamente a las funcionalidades y datos permitidos según su rol.
06	Todas las acciones realizadas se registran en el sistema para fines de auditoría.
07	Fin del Flujo principal.
Postcondicion(es):	<ul style="list-style-type: none"> • El sistema permite o deniega el acceso a las funcionalidades o datos según el rol del usuario. • Se almacena un registro de la actividad del usuario (acciones permitidas o denegadas) para fines de auditoría.
Flujo alternativo	
S04	Si el usuario intenta acceder a una funcionalidad o información que no está permitida para su rol, el sistema muestra un mensaje de acceso denegado y redirige al usuario a la pantalla de inicio o a una sección permitida.
S05	Fin del Flujo alternativo
Expcion(es):	<ul style="list-style-type: none"> • Si el sistema no puede autenticar al usuario debido a problemas de conexión o fallos en la BD, se muestra un mensaje de error y se solicita reintentar más tarde. • Si un usuario no tiene un rol asignado o su rol ha sido desactivado, el sistema muestra un mensaje de error y no permite el acceso a

	ninguna funcionalidad hasta que el Administrador del Sistema revise su configuración.
Autor: Pablo Alejandro Hamann	Ultima Modificación: 13/09/2024

Descripción de CU11: Gestión de Francos Compensatorios

CU04: Registrar Horas Extras y Calcular Francos Compensatorios						
Nivel del CU:	<input type="checkbox"/> Negocio	<input checked="" type="checkbox"/> Sistema de Inf.	Prioridad:	<input type="checkbox"/> Baja	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Alta
Tipo de CU:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto	Complejidad:	<input type="checkbox"/> Baja	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Alta
Referencia(s):	RF04, RF11					
Actor(es):	<ul style="list-style-type: none"> • Empleado • Jefatura de Servicio • Oficina de Personal 					
Descripción:	Permite a los empleados generar francos compensatorios (FC) en el sistema. Los francos compensatorios generados, deben ser autorizados por la jefatura de servicio, y luego validadas por la Oficina de Personal. Esto garantiza que el servicio no se verá resentido (responsabilidad del jefe que lo autoriza), y que se usufructúan acorde a las reglas laborales del empleado en cuestión					
Precondicion(es):	<ul style="list-style-type: none"> • El empleado debe haber iniciado sesión en el sistema con sus credenciales. • El usuario debe disponer de la suficiente cantidad de horas extras necesarias para generar el franco compensatorio. 					
Flujo principal:						
01	El empleado accede al módulo: «Generar Franco Compensatorio».					
02	Selecciona la fecha laboral para la cual se usufructuará el FC.					
03	El sistema verifica automáticamente si las horas extras disponibles son suficientes para generar el FC para la fecha.					
04	De forma automática, el sistema selecciona la cantidad necesaria de horas extra de las disponibles comenzando a consumir siempre primero las horas extra más antiguas, y cambia la disponibilidad de las mismas marcándolas como “asignadas a FC pendiente de usufructo”.					
05	La Jefatura de Servicio recibe una notificación para autorizar el franco compensatorio.					
06	La Jefatura de Servicio accede al módulo de validación, revisa los FC solicitados y los aprueba o rechaza.					
07	Si el FC es aprobado, el sistema envía una solicitud de validación a la Oficina de Personal, y una notificación al Empleado avisando que su FC ha sido aprobado y que está pendiente de validación por Oficina de Personal.					
08	La Oficina de Personal accede al módulo de validación, revisa los FC autorizados por validar, y los valida u observa.					
09	Si los FC son validados, el sistema marca las horas extra como usufructuadas, y notifica al Empleado que el FC ha sido autorizado y validado, y que no deberá asistir a trabajar el día solicitado.					
10	Fin del Flujo principal.					

Postcondicion(es):	<ul style="list-style-type: none"> El pedido de los FC queda registrado y pendiente de autorización y validación por la Jefatura de Servicio y la Oficina de Personal respectivamente. Una vez autorizados y validadas, los FC se almacenan en un historial de FCs usufructuados en el perfil del empleado.
Flujo alternativo	
S04	El sistema le informa al usuario que no dispone de horas extra suficientes necesarias para generar el FC para la fecha seleccionada. Se le sugiere trabajar X cant. de horas extra para poder completar el FC para la fecha solicitada.
S05	Fin del Flujo alternativo
Excepcion(es):	<ul style="list-style-type: none"> Si se detectan inconsistencias en la fecha de solicitud del FC (por ejemplo, para una fecha en la cual el empleado no trabaja), se alerta al usuario y no se permite continuar hasta que la situación sea corregida.
<i>Autor: Pablo Alejandro Hamann</i>	<i>Ultima Modificación: 30/09/2024</i>

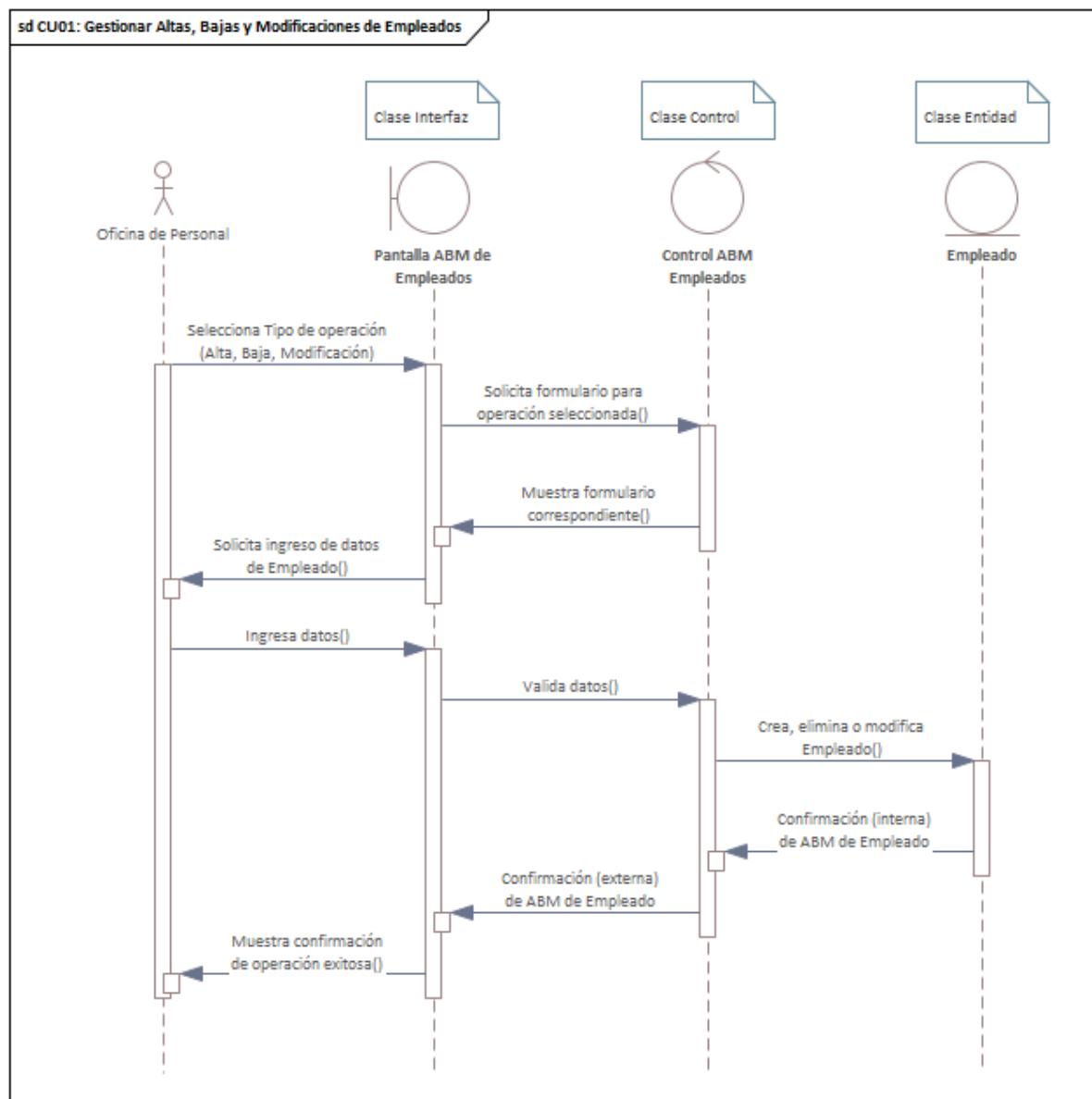
9. Etapa de análisis

Diagramas de secuencia

Para la confección de los diagramas de secuencia, distinguiremos entre clases de Interfaz, Control y Entidad:

- **Clases de Interfaz:** responsables de la interacción directa con el usuario (por ejemplo, Empleados, Oficina de personal, Jefaturas de Servicio, etc.), que nos permiten la interacción con el sistema, enviando y recibiendo datos.
- **Clases de Control:** gestionan la lógica del sistema. Coordinan la recepción de datos desde la interfaz, los validan y hacen que las entidades los procesen o que interactúen con otras clases. Conectan las acciones del usuario (a través de la interfaz) con el modelo del sistema (las entidades).
- **Clases de Entidad:** representan los datos del dominio. Corresponden a las tablas de la BD o los objetos principales del sistema (Empleados, Horarios, Licencias, etc.).

A continuación, se presentan los diagramas de secuencia para los casos de uso CU01, CU02, CU03, CU04, CU07 y CU010.

Diagrama de secuencia para CU01**Curso Normal**

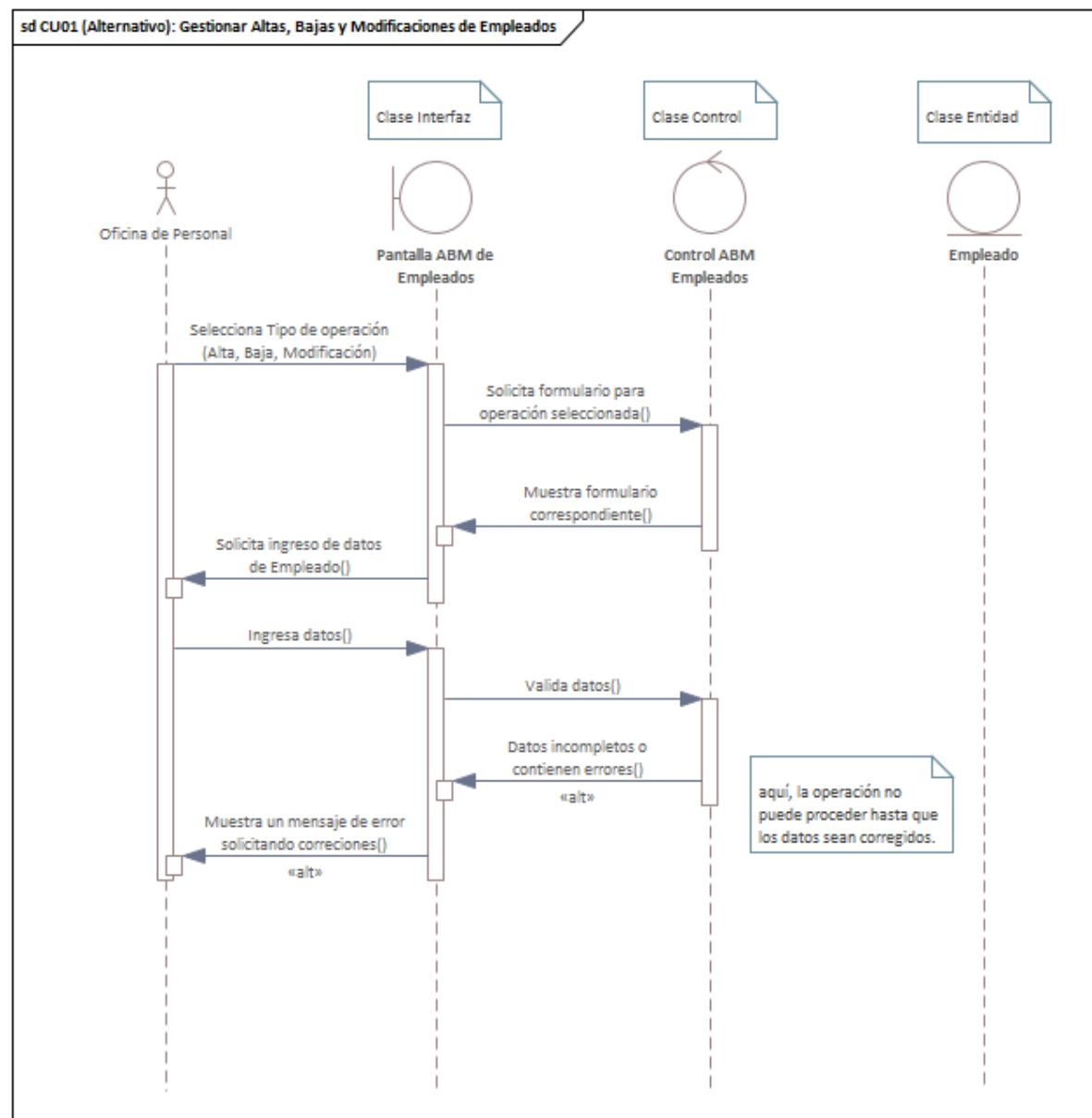
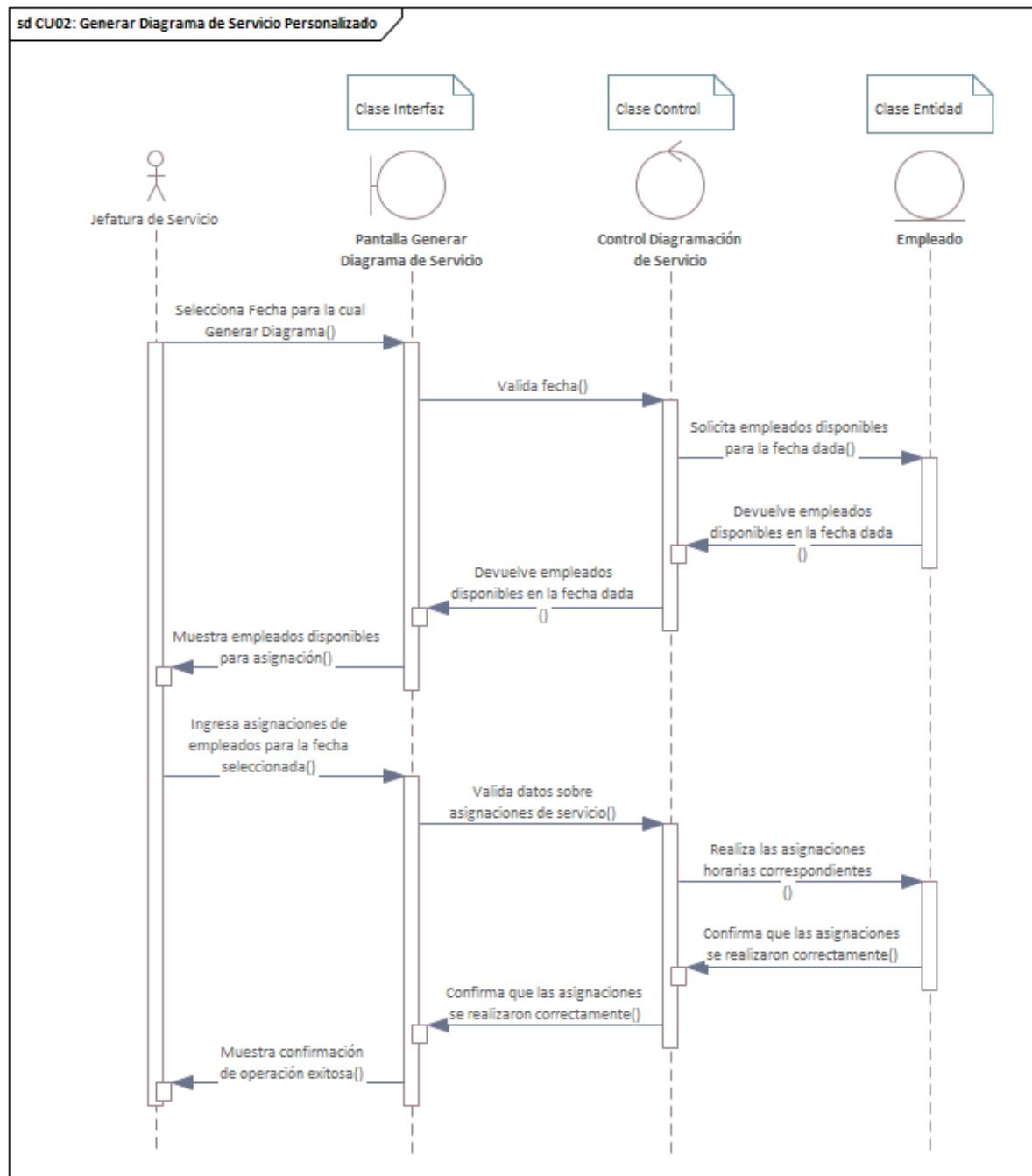
Curso Alternativo

Diagrama de secuencia para CU02**Curso Normal**

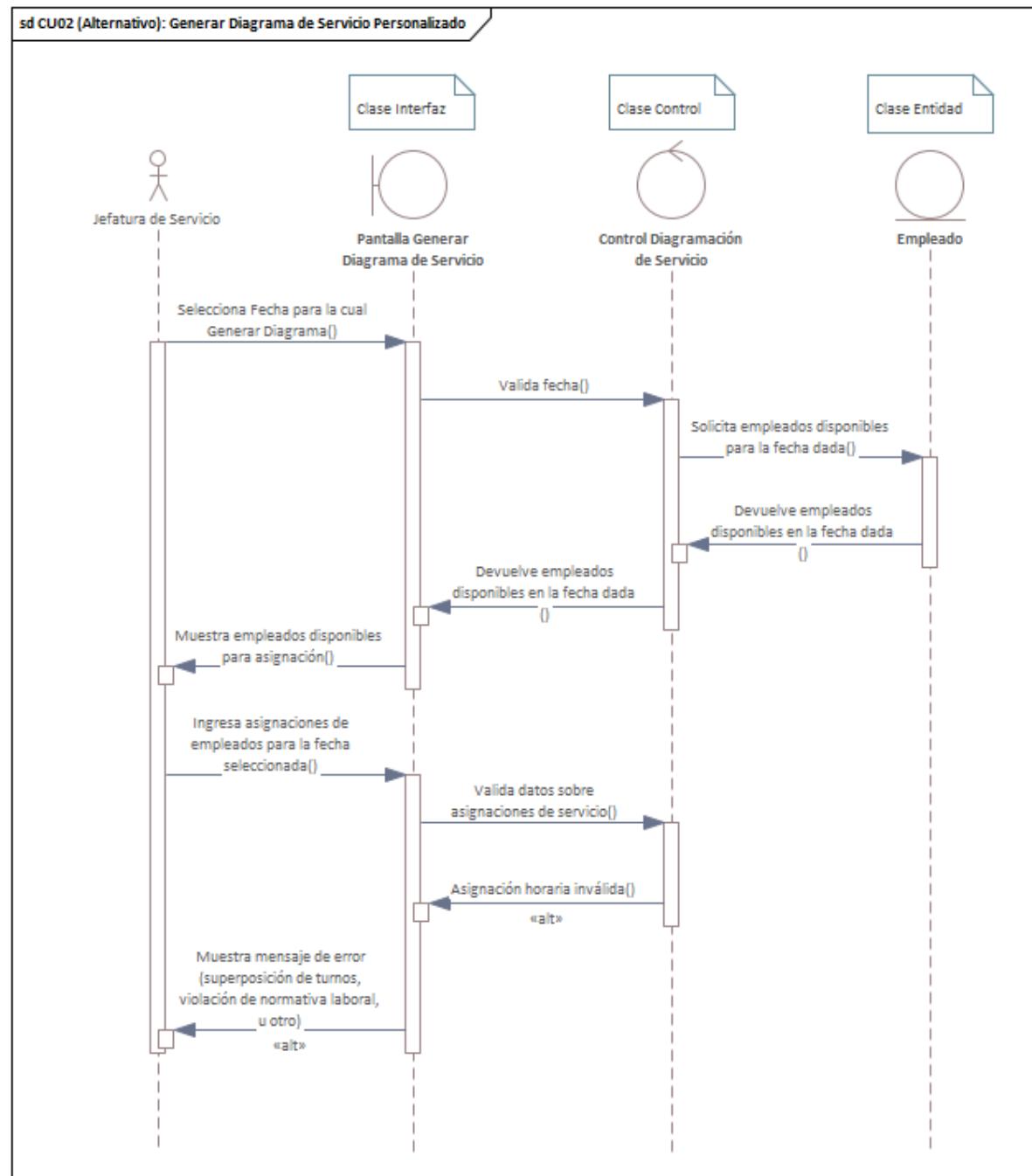
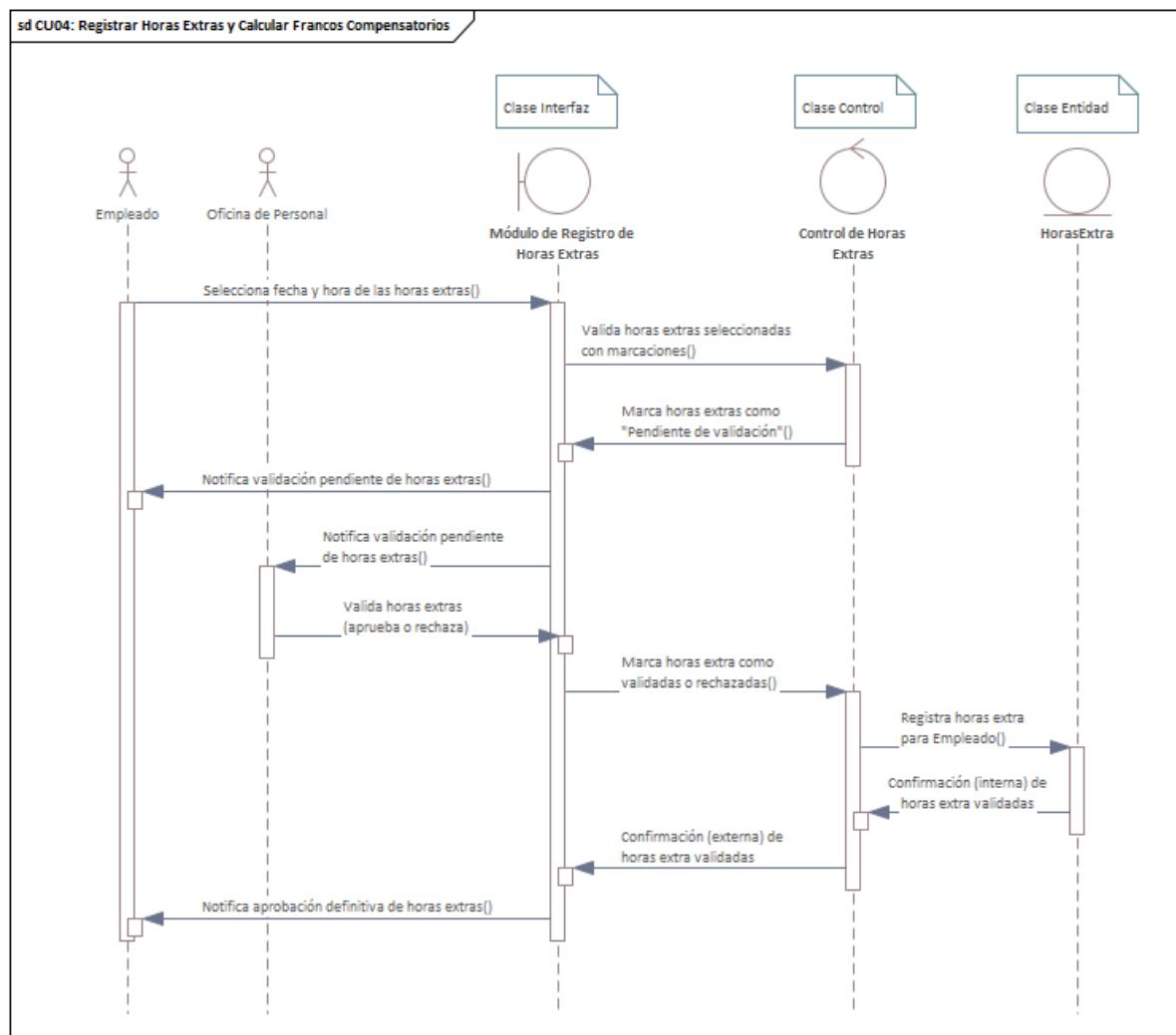
Curso Alternativo

Diagrama de secuencia para CU04**Curso Normal**

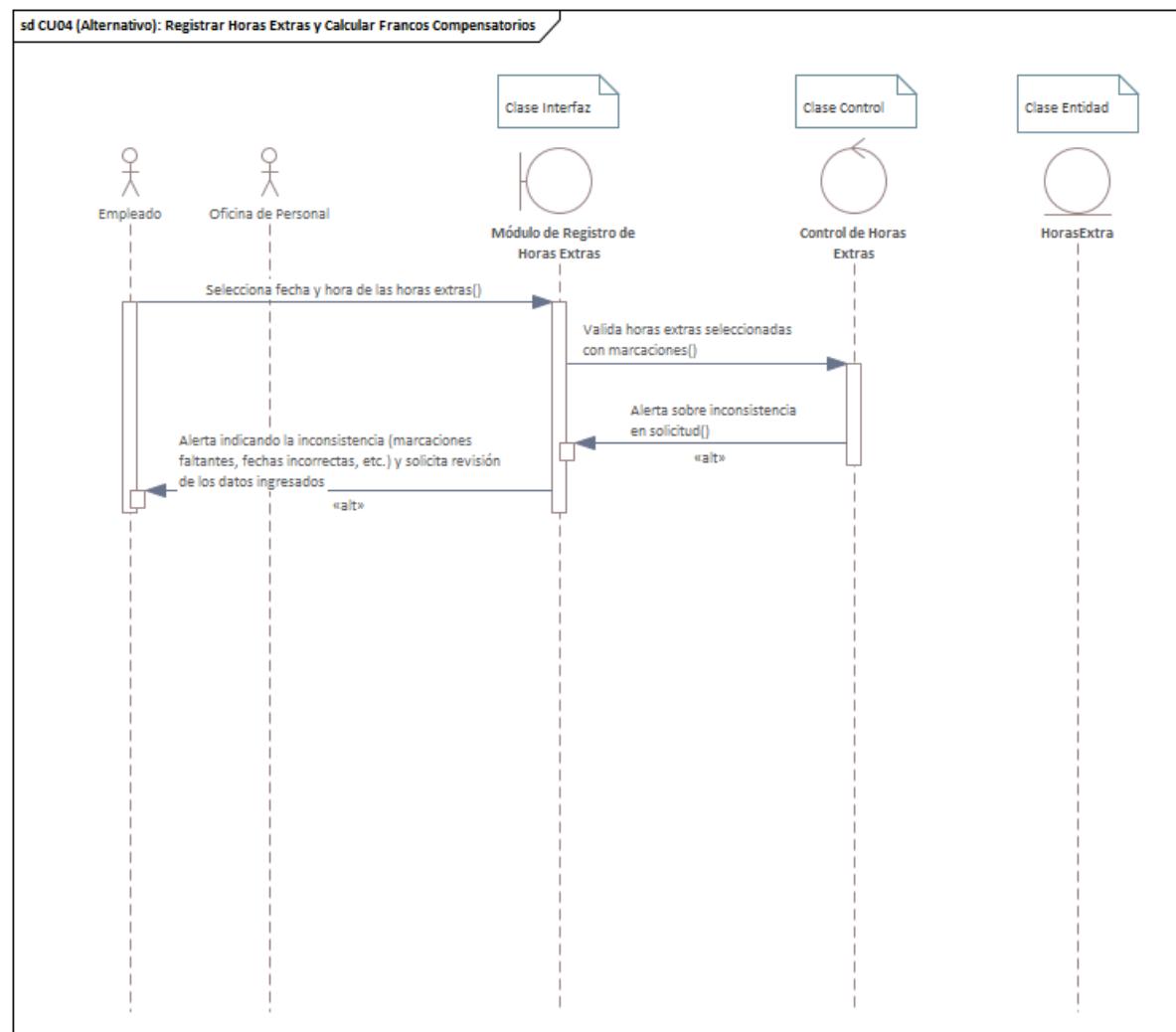
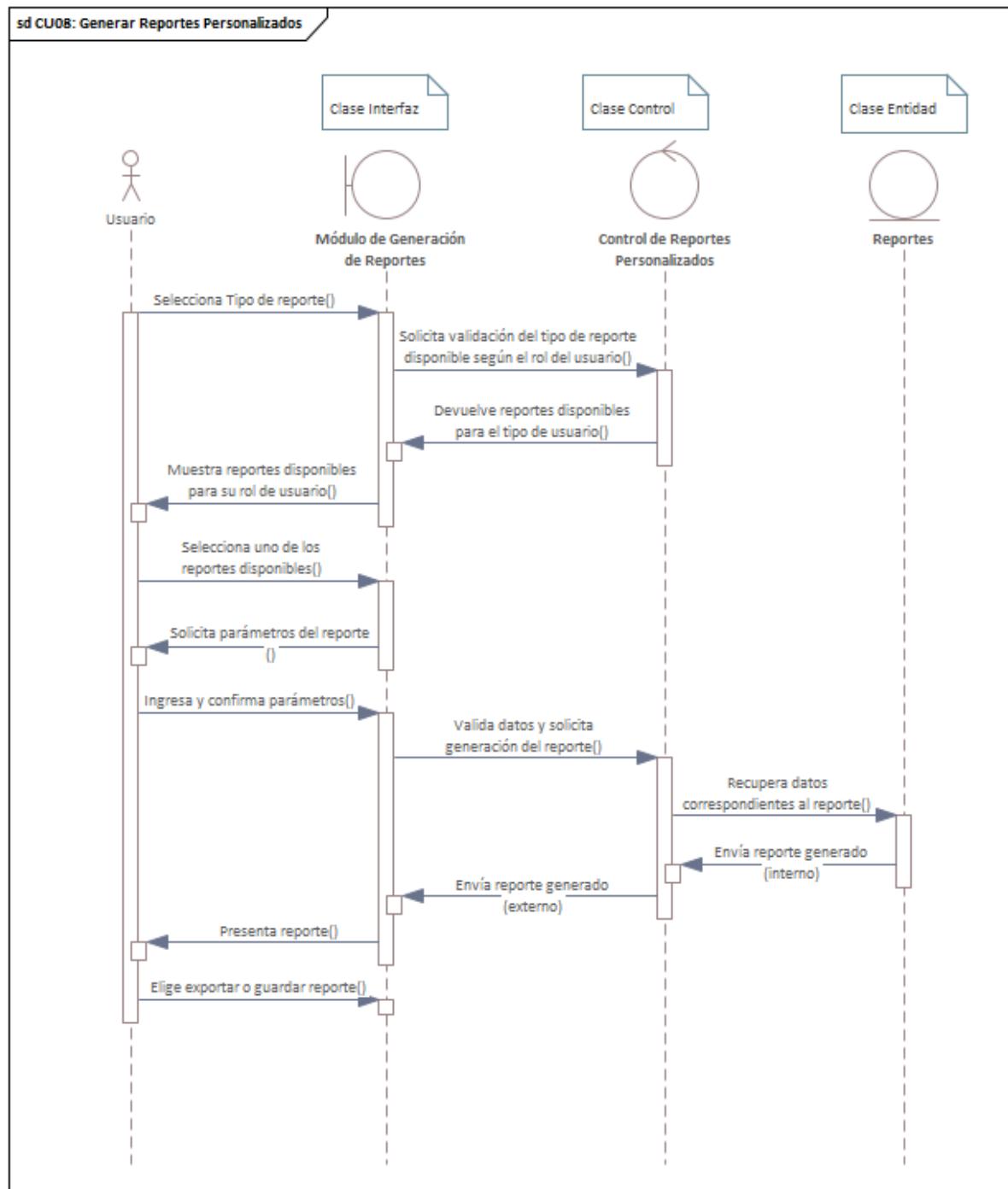
Curso Alternativo

Diagrama de secuencia para CU08**Curso Normal**

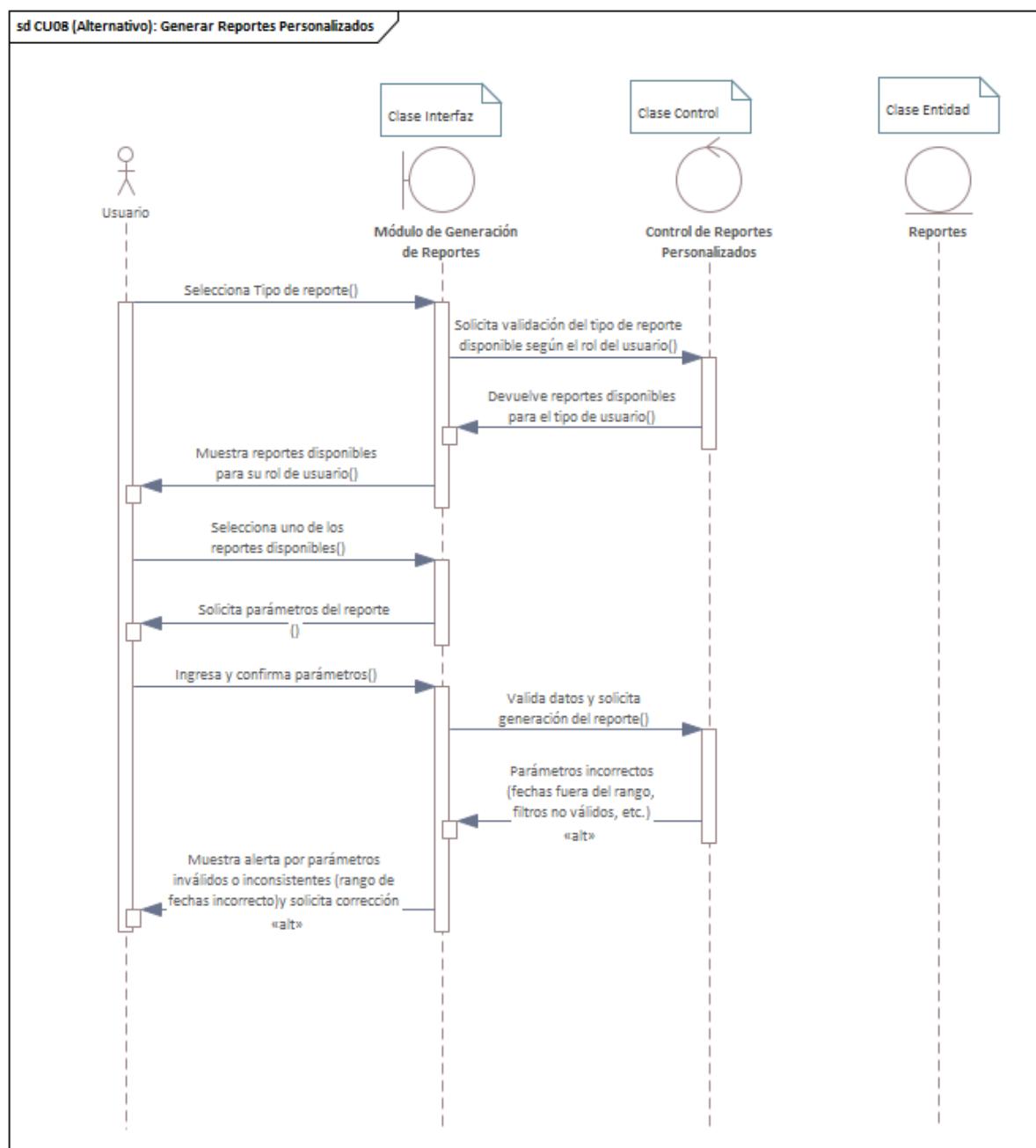
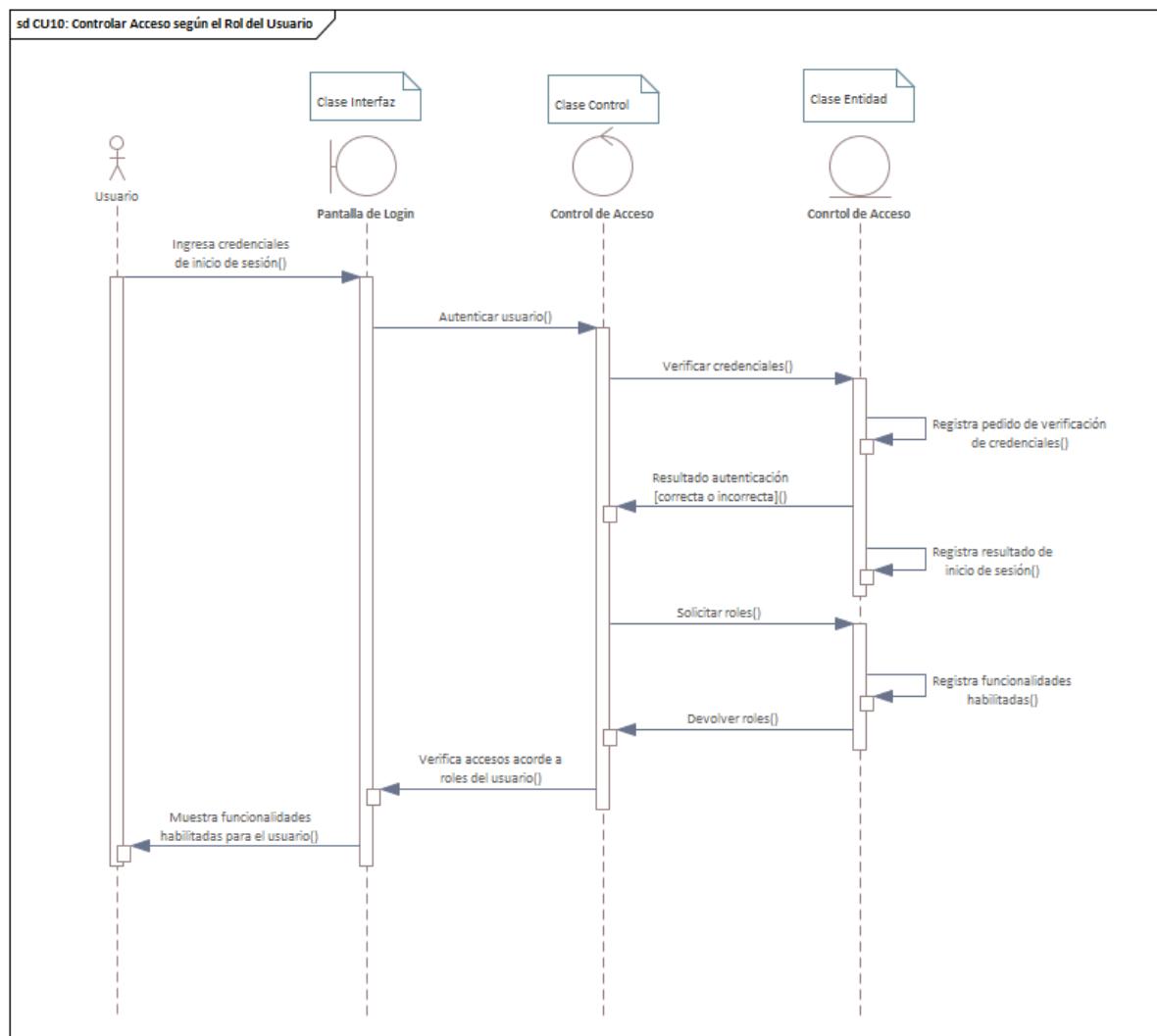
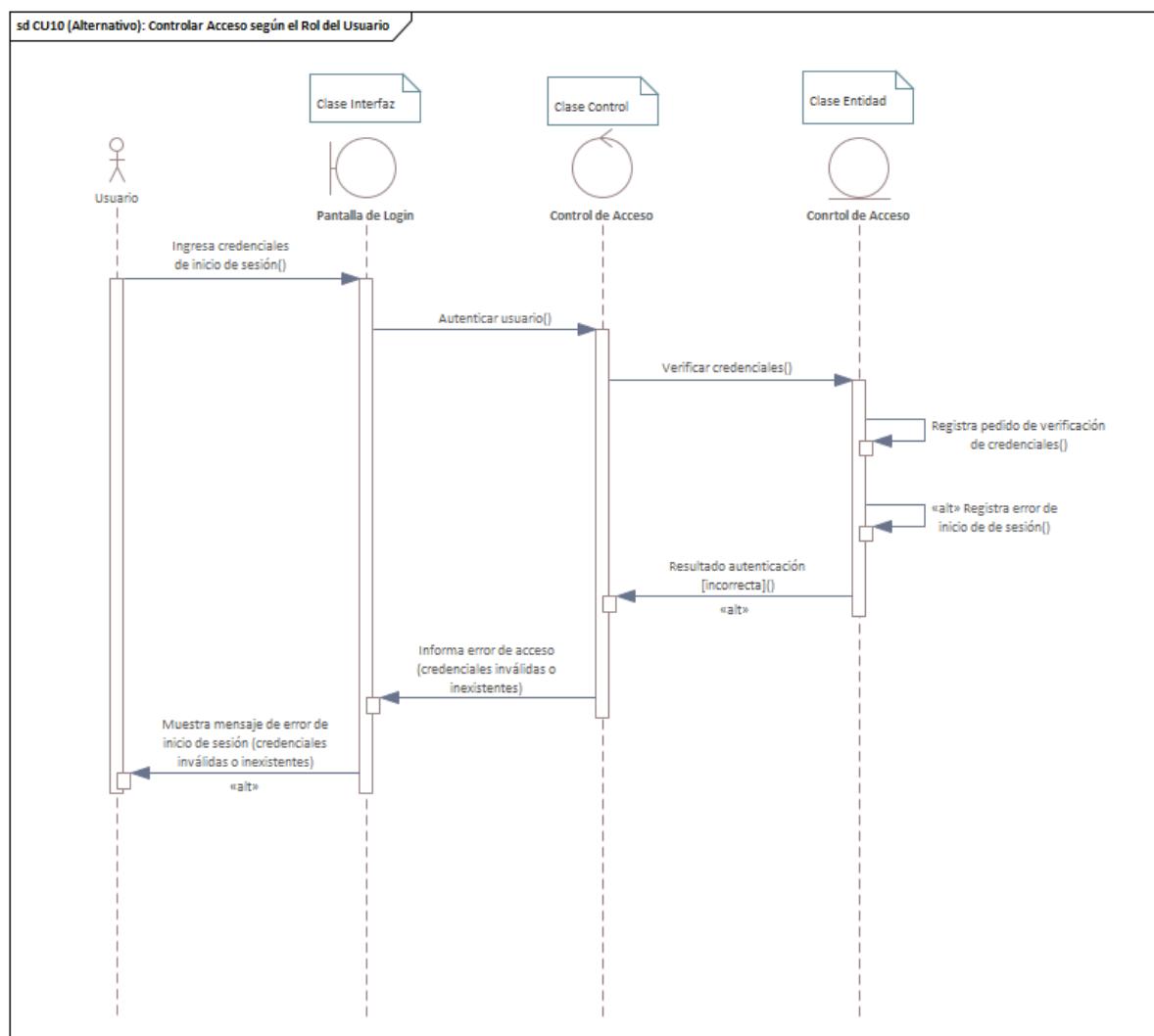
Curso Alternativo

Diagrama de secuencia para CU10**Curso Normal**

Curso Alternativo**10. Etapa de diseño****Diagrama de clases de diseño**

La complejidad del diseño reside en el modelado de la clase **Horario**. En el Hospital de Urgencias coexisten múltiples regímenes horarios y un empleado puede cambiar de régimen de un mes a otro, dependiendo de la necesidad del servicio. Por esto, se decidió trabajar **Horario** como una clase abstracta e implementar en diferentes clases cada tipo de horario, con sus particularidades. (*Patrón de Diseño Strategy*). Resumiendo lo expuesto al principio del documento, tenemos:

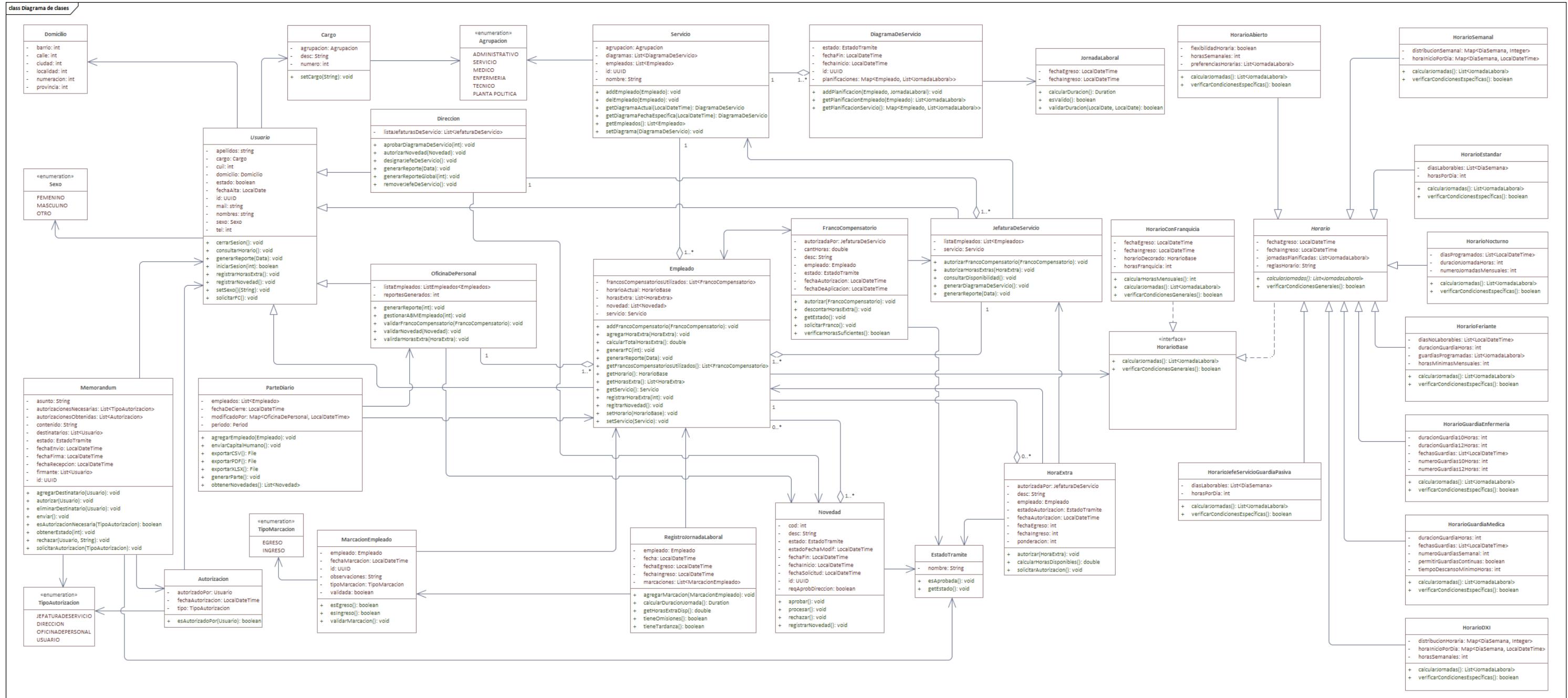
1. **Horario estándar**
2. **Horario semanal**
3. **Horario nocturno**
4. **Guardias de feriante**
5. **Guardias de enfermería**
6. **Guardias médicas**

7. **Diagnóstico por imágenes**
8. **Jefes de servicio con guardia pasiva**
9. **Horario abierto**
10. **En comisión:** como se expuso al principio, este caso particular lo omitiremos
11. **Franquicias horarias:** utilizaremos el patrón de Diseño *Decorator*.
12. **Condicionalidad general:** no menos de 10 Hs entre ingresos y egresos

Otras consideraciones de diseño: (contrariamente a las recomendaciones la cátedra de Análisis y Diseño de Software), se optó en tres ocasiones muy precisas, implementar una relación bidireccional entre las clases:

- **Servicio y Empleado.** Esto produce redundancia en el modelo, pero facilita el acceso desde ambas clases (Empleado y Servicio necesitan interactuar frecuentemente entre sí), y para garantizar la coherencia de los datos se desarrollan métodos adicionales encapsulando las operaciones que gestionan la asignación y desvinculación de un empleado a un servicio. Las alternativas (relación unidireccional).
- **Servicio y DiagramaDeServicio.** De forma similar al caso anterior: se encapsulan las operaciones para asegurar la coherencia de los datos.
- **Empleado y FrancoCompensatorio.** Ídem a anterior. Se mantiene utiliza también encapsulamiento de las operaciones que modifican los datos.
Para el registro laboral del empleado, tenemos dos clases muy parecidas:
- **JornadaLaboral** es la planificación previa que define lo que debería suceder según el plan de trabajo establecido.
- **RegistroJornadaLaboral** registro posterior que registra lo que realmente sucedió, incluyendo diferencias respecto a lo planificado.

Sistema de Gestión de Ausentismo Hospitalario



Disponible en otros formatos en *GitHub*:

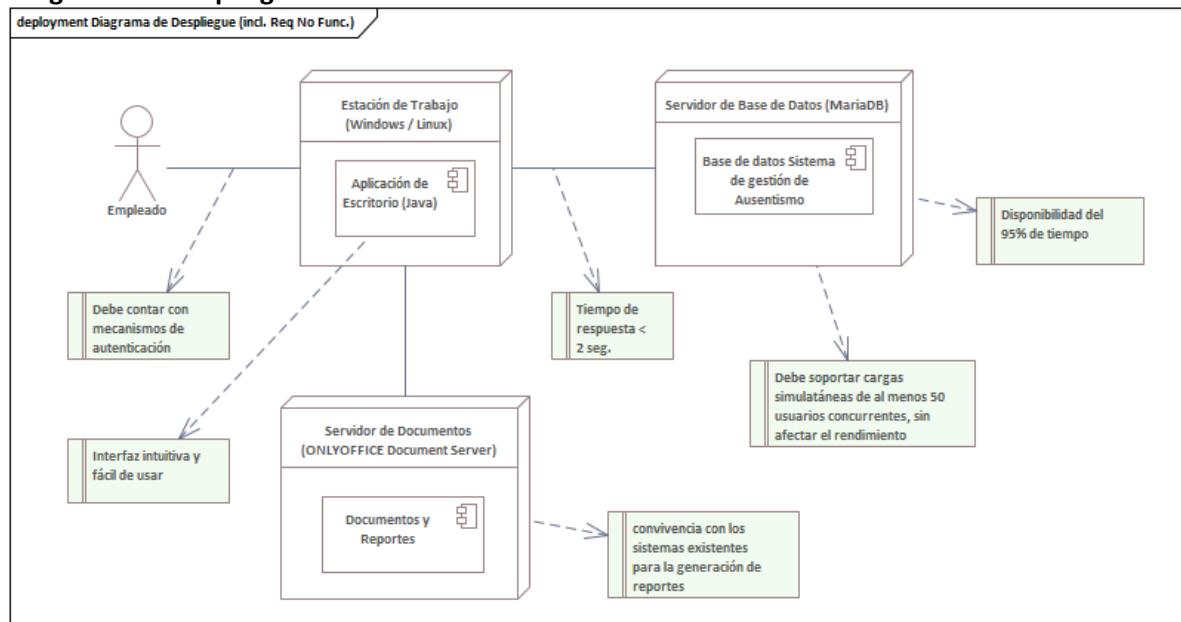
Imagen PNG: <https://raw.githubusercontent.com/linkstat/seminare/refs/heads/main/diagrams/DiagramaDeClases.png>

Documento PDF: <https://github.com/linkstat/seminare/blob/main/diagrams/DiagramaDeClases.pdf>

Fuente (Enterprise Architect): <https://github.com/linkstat/seminare/blob/main/diagrams/DiagramaDeClases.eapx>

11. Etapa de implementación

Diagrama de Despliegue



12. Etapa de pruebas

Para el presente proyecto, se generará un plan de pruebas para la aplicación Java del Sistema de control de Ausentismo del HMU, en donde se planificará y ejecutarán las pruebas necesarias para verificar y validar la correcta implementación del sistema (tanto versiones intermedias como final)

Plan de pruebas

Atento a las formas de clasificar, las pruebas se corresponden a caja blanca, caja negra y caja gris:

- **Pruebas de caja negra:** realizaremos pruebas a través del conocimiento que tenemos sobre los requerimientos funcionales y no funcionales; de esta forma sabremos qué salida esperar ante una entrada ingresada.
- **Pruebas de caja blanca:** cuando conocemos las estructuras: código fuente, la jerarquía de los componentes, el flujo de control, el flujo de datos. Un usuario normal (externo al equipo de desarrollo) no tiene este conocimiento. Nuestras pruebas serán desde el punto de vista de un usuario del sistema, por lo cual la mayoría de las pruebas serán de caja negra.
- **Pruebas de caja gris:** básicamente son pruebas de caja negra, pero en un marco en el cual poseemos conocimiento, información adicional que podemos usar como entrada. Para mantenerlo simple, tomamos caja negra y caja gris de forma indistinta.

Objetivos

Para responder esto, nos planteamos las siguientes preguntas: ¿para qué vamos a probar, y qué queremos lograr con estas pruebas?

- Se persigue la identificación de errores potenciales reduciendo la probabilidad de que aparezcan defectos en la aplicación.
- Lograr la satisfacción de los usuarios mediante el cumplimiento de sus expectativas (queremos que los mismos estén conformes).
- Asegurar el cumplimiento de los RF (requisitos funcionales) y RNF (requisitos no funcionales); que la aplicación funcione acorde a éstos.

Alcance

Para responder esto, queremos saber qué vamos a probar, pero de manera genérica. ¿Qué va a estar dentro de las pruebas, y qué es lo que no se va a contemplar en las mismas? Realizaremos pruebas funcionales acerca de:

- Fallas de datos
- Fallas de control
- Fallas de entrada/salida
- Fallas de interfaz
- Fallas de almacenamiento
- Pruebas de seguridad
- Pruebas de performance

Equipo

Nuestros individuos usuarios de pruebas serán usuarios con conocimiento de informática, pertenecientes al servicio de Informática del hospital, y la jefa de personal, con conocimiento estándar de las tecnologías actuales. Se trata de usuarios *normales*, que no son parte del equipo de desarrollo, ni tampoco *testers profesionales*. Todos utilizarán estaciones de trabajo PC con Windows conectadas a la red hospitalaria. Es decir, las pruebas serán en el entorno real de ejecución primaria.

Estrategia

- Tipo de pruebas a realizar:
 - Pruebas funcionales y no automatizadas.
- Organización de las pruebas:
 - Al final del proceso de diseño (estamos probando un producto ya desarrollado).
- Cantidad de ciclos de prueba:
 - uno por cada Caso de Prueba

Criterios

- Inicio de pruebas:
 - cuando tengamos un *ejecutable Java*.
- Cantidad de defectos aceptados:
 - Ninguno grave.
 - No deben producirse errores graves para que las pruebas sean consideradas exitosas.
- Final de pruebas:
 - cuando no existan errores graves y se haya completado al menos el 90% de las pruebas, dado que se trata de una aplicación que gestiona el ausentismo y el historial completo de novedades de todos los empleados.

Ambientes

Dado que lo que se va a probar es una aplicación Java 8, cualquier plataforma con soporte JRE 8 o superior funcionará bien. Utilizaremos entonces:

- Para plataforma Windows: PCs marca y modelo *Dell OptiPlex 3080 (JFC48B3)*, con hardware basado en arquitectura Intel x64, 8GB de RAM, 240GB de almacenamiento local, y sistema operativo Windows 10 LTSC, y Windows 11 Enterprise 22H2, con conectividad ethernet gigabit y salida a Internet mediante proxy.

Entregables

Se generarán los **Casos de Prueba** que se desarrollan más abajo en el presente documento.

Gestión de incidentes

No se utilizarán **Diagramas de Transición de Estados** para el seguimiento de los defectos. En su lugar, se implementará una **Escala de Severidad (CRÍTICA – ALTA – MEDIA – BAJA)** que permitirá clasificar los defectos en función de su impacto en el sistema, sin tener en cuenta la prioridad de su resolución. Esta escala se utilizará para asignar la severidad a los defectos según el siguiente criterio:

- CRÍTICA**: El defecto impide completamente el uso de una funcionalidad clave del sistema.
- ALTA**: El defecto afecta significativamente la funcionalidad, pero el sistema aún es usable.
- MEDIA**: El defecto tiene un impacto moderado, pero existe una solución alternativa o no afecta funciones esenciales.
- BAJA**: El defecto es menor, generalmente relacionado con aspectos cosméticos o de poca repercusión funcional.

Riesgos

Para gestionar los riesgos asociados a la etapa de pruebas, nos anticiparemos a posibles sucesos que podrían afectar el desarrollo o el funcionamiento del sistema. Implementaremos un plan tanto de **mitigación** (para reducir la probabilidad de que el riesgo ocurra) como de **contingencia** (para reaccionar adecuadamente si el riesgo finalmente se materializa). El razonamiento aplicado para la confección de la tabla de riesgos desarrollada a continuación es el siguiente:

DADO QUE...	ENTONCES (POSSIBLEMENTE)
[condición o circunstancia que genera el riesgo]	[consecuencia del riesgo si no es mitigado]

Cuadro de riesgos

Si bien siempre existen riesgos externos al sistema que no podemos controlar completamente (como por ejemplo, la pérdida de conectividad a la red o que el usuario olvide sus credenciales de acceso), siempre intentaremos tener un plan de contingencia para minimizar el impacto de estos eventos.

El cuadro que se desarrolla en las páginas siguientes, adopta el siguiente formato y lógica de razonamiento:

Riesgo	Prob.	Impacto	Mitigación	Contingencia
<i>Descripción del riesgo</i>	<i>ALTA / MEDIA / BAJA</i>	<i>CRÍTICO / ALTO / MEDIO / BAJO</i>	<i>Acciones preventivas que tomaremos para evitar que ese riesgo ocurra</i>	<i>Acciones correctivas que aplicaremos, una vez que ese riesgo ocurrió</i>

(cuadro en sección siguiente)

Riesgo	Prob	Impacto	Mitigación	Contingencia
Que el usuario no pueda iniciar sesión, por falla de conectividad	MEDIA	CRÍTICO	Utilizar un mecanismo de validación que almacene localmente las últimas credenciales de inicio de sesión utilizadas de forma local, con un tiempo de expiración de 10 Hs, y que revalide las mismas durante el próximo inicio de sesión éxitos contra el servidor.	Permitir un inicio de sesión «fuera de línea», que le indique al usuario que, por problemas de conectividad, solo podrá acceder a consultas previamente guardadas en el dispositivo que está utilizando, durante las próximas 10 horas.
Que el usuario se olvide sus credenciales para iniciar sesión en la aplicación	MEDIA	CRÍTICO	Una vez validado el primer inicio de sesión con credenciales introducidas por el usuario, ofrecer activar 2FA de forma tal que por más que se olvide su contraseña, aun pueda iniciar sesión mediante 2FA.	Cuando el usuario no recuerde sus credenciales de inicio de sesión y no tenga activado 2FA, hacerle 3 preguntas secretas que, tras ser respondidas correctamente, genere y envíe una nueva contraseña a la dirección de correo electrónico registrada.
Que no se puedan realizar consultas ni otras operaciones, por saturación de servidor.	BAJA	ALTO	Si debido a un problema de comunicación, lentitud o saturación del servidor, no se obtiene una respuesta en tiempo real durante una solicitud de consulta o de realización de una operación, se deberán encolar las solicitudes, marcándolas como pendientes de respuesta, hasta que el servidor responda.	Si el servidor no responde a una solicitud en un plazo de 5 minutos, se deberá notificar al usuario mediante un mensaje emergente, indicando que debe realizar nuevamente su consulta u operación, ya que no se completó exitosamente. Además, se precargarán los últimos datos ingresados para evitar que el usuario tenga que cargarlos de nuevo.
Actualización del modelo de datos y/o del servidor de bases de datos que genere una incompatibilidad con versiones anteriores de la aplicación utilizada por el cliente.	MEDIA	CRÍTICO	Implementar un sistema de verificación de compatibilidad de versiones, de forma tal que al iniciar la aplicación cliente, se verifiquen la versión del modelo de datos y de la aplicación, y su compatibilidad con el cliente, ofreciendo una actualización en línea en caso de ser necesario.	Mostrar mensaje de error, indicando que la versión en ejecución de la aplicación es muy vieja, y que por motivos de seguridad el usuario debe actualizar manualmente la misma (ofrecer enlace de descarga).

Sistema de Gestión de Ausentismo Hospitalario

Riesgo	Prob	Impacto	Mitigación	Contingencia
Que la Base de Datos se corrompa de forma tal que se pierda toda la información en ella almacenada	BAJA	CRÍTICO	Se utilizarán servidores virtuales (específicamente contenedores LXC), almacenados sobre un espejo en sistema de archivos ZFS, que aseguren la redundancia de datos (al contar con dos copias simultaneas de la información almacenada), tanto a nivel de base de datos, como a nivel de sistema operativo.	Se realizarán respaldos completos cada 12 Hs, y respaldos diferenciales cada hora de la BD. Adicionalmente, se contarán con respaldos diarios de la VM a nivel de sistema, a forma de contar con dos niveles de respaldo: a nivel de base de datos y a nivel de sistema operativo, para restaurar en caso de catástrofe.
Que un empleado pueda solicitar una Hora Extra desde un rango de fecha y horario parcial o totalmente ya utilizada.	BAJA	MEDIO	Al momento de asignar la Hora Extra, verificar que, entre las horas extras de dicho usuario, no exista una Hora Extra que se superponga total o parcialmente con la fecha y hora solicitada para la Hora Extra.	En el caso de que un empleado tenga más de una Hora Extra para una fecha y hora dada, notificar mediante mensaje emergente a la Oficina de Personal, un error de superposición de Hora Extra, permitiéndole eliminársela del registro de Horas Extras del usuario.
Que un empleado solicite y obtenga aprobación de un FC para una fecha futura y que luego no sea aplicable, por ejemplo, por declaración de asueto para esa fecha, o causa de fuerza mayor.	BAJA	MEDIO	Al momento de generar el FC, se le da la opción al usuario de “descartar automáticamente el FC en caso de no ser aplicable por fuerza mayor”, de esta forma, el FC se aprueba con esa condicionalidad, que, de cumplirse, llegado el momento se desestima automáticamente y se notifica a los actores involucrados (Empleado, Jefatura de Servicio y Oficina de Personal).	Al momento de decretarse el asueto o la causa de fuerza mayor, verificar qué usuarios tenían FC programados para esa fecha (si hubiere alguno), desestimar el FC, devolver las Horas Extra y notificar a los usuarios involucrados (Empleado, Jefatura de Servicio y Oficina de Personal).

Sistema de Gestión de Ausentismo Hospitalario

Riesgo	Prob	Impacto	Mitigación	Contingencia
Que la modificación de un Diagrama de Servicio produzca cambios en la fecha de regreso a uno o más Empleados en período de licencia	BAJA	CRÍTICO	Cualquier modificación de un Diagrama de Servicio que afecte la fecha de regreso a las actividades laborales al menos a un empleado con licencia en curso, deberá mostrar mensajes de advertencia a la Jefatura de Servicio, de forma tal que pueda haber una comunicación y un acuerdo previo con el o los empleados afectados.	En el caso de que la modificación del Diagrama de servicio que afecte la fecha de regreso a las actividades laborales al menos a un empleado, sea llevada a cabo haciendo caso omiso de las advertencias, el sistema deberá notificar, de forma instantánea y automáticamente por todos los medios a los actores involucrados. Adicionalmente, al Empleado afectado, se le notificará diariamente la fecha de regreso al trabajo desde 5 días corridos antes, solicitando confirmación de recepción para los medios que lo permitan (por ejemplo, mail).
Que un servicio quede sin Jefe de servicio que pueda autorizar los pedidos de los empleados (por ejemplo, por licencia o enfermedad).	MEDIA	ALTO	El sistema deberá verificar que un servicio tenga siempre un jefe activo. En caso de licencia, el jefe de servicio deberá designar a un empleado como encargado (jefe temporal) de servicio, y notificar a la Oficina de Personal y a la Dirección para que presten su conformidad o soliciten otro empleado a cargo.	En el caso de que un jefe de servicio no haya designado a nadie como empleado a cargo, la Dirección y/o la Oficina de Personal deberán autorizar un encargado de servicio (jefe temporal) que autorice los pedidos de los empleados del Servicio.

Procedimiento de prueba

Dado el alcance del sistema, se definen las siguientes tareas para llevar a cabo el proceso de prueba. Estas tareas abarcan desde la preparación de datos hasta la ejecución de los casos de prueba por parte de diferentes perfiles de usuario, asegurando que todas las funcionalidades del sistema se verifiquen adecuadamente.

Preparar los Datos de Prueba

1. Base de datos de prueba:

- Se creará una base de datos de prueba con la misma estructura que la base de datos de producción, pero utilizando datos ficticios generados específicamente para este entorno.

2. Población inicial de la base de datos:

- Se generarán **50 usuarios** ficticios cuyos datos personales, como nombre, apellido, DNI y cargo, se tomarán de un archivo CSV exportado desde un documento de Excel con la información generada.
- A cada usuario se le asignará un correo electrónico ficticio, ya que no existen registros de correos en la base de datos original de producción.
- Los servicios (oficinas / áreas) disponibles en el sistema se cargarán desde otro archivo CSV que contenga el listado de servicios hospitalarios predefinidos (por ejemplo: Traumatología, Admisión, Cirugía, etc.).

3. Asignación de usuarios a roles:

- Desde la base de datos de pruebas ya poblada:
 - 2 usuarios serán promovidos al rol de **Dirección**.
 - 3 usuarios serán promovidos al rol de **Oficina de Personal**.
 - 5 usuarios serán promovidos al rol de **Jefatura de Servicio**.
 - Los 40 usuarios restantes serán asignados al rol de **Empleado**.

Asignación de testers y acceso

1. Acceso por perfiles:

- El **primer tester** tendrá acceso a dos usuarios:
 - Un usuario con perfil de **Dirección**.
 - Un usuario con perfil de **Oficina de Personal**.
- El **segundo y tercer tester** tendrán acceso a dos usuarios cada uno:
 - Un usuario con perfil de **Jefatura de Servicio**.
 - Un usuario con perfil de **Empleado**.
- El **cuarto tester** tendrá acceso a dos usuarios con el perfil de **Empleado**.

Generar los Casos de Prueba

1. CP01: Verificar la autenticación en el inicio de sesión:

- Todos los testers, independientemente de su perfil de usuario, deberán intentar iniciar sesión con datos faltantes, incorrectos o inexistentes. El objetivo de esta prueba es asegurar que la funcionalidad de autenticación funcione correctamente, evitando accesos no autorizados o incorrectos.

2. CP02: Gestión de asignaciones:

- El tester con perfil de **Dirección** distribuirá la asignación de empleados a los servicios. Esta tarea verificará que la funcionalidad de asignación de empleados a los servicios esté funcionando correctamente.

3. CP03: Generación de diagramas de servicio:

- Los testers con perfil de Jefatura de Servicio deberán generar los respectivos diagramas de servicio para los empleados asignados a su área. El objetivo es probar la funcionalidad de creación y modificación de diagramas de servicio, verificando la asignación de horarios y guardias.

4. CP04: Carga de marcaciones:

- El tester con perfil de Oficina de Personal será responsable de cargar un listado de marcaciones ficticias para los empleados. Esto validará la funcionalidad de registro y almacenamiento de marcaciones.

5. CP05: Solicitudes de empleados:

- Los testers con perfiles de Empleado realizarán las siguientes acciones:
 - Generar solicitudes de horas extra.
 - Solicitar francos compensatorios.
 - Registrar solicitudes de faltas justificadas.
 - Esto permitirá verificar que el sistema gestione correctamente las solicitudes de los empleados.

6. CP06: Validación y aprobación:

- Los usuarios con perfiles de responsabilidad (Dirección, Oficina de Personal y Jefatura de Servicio) aprobarán o rechazarán de manera aleatoria las solicitudes de los empleados (horas extra, francos compensatorios, faltas justificadas). Este proceso validará las funcionalidades relacionadas con la aprobación, rechazo y gestión de solicitudes en el sistema.

Casos adicionales

7. CP07: Simulación de sobrecarga:

Se probarán escenarios en los que se generen múltiples solicitudes en paralelo para evaluar cómo responde el sistema bajo carga.

Responsables de las pruebas

1. Testers:

- El equipo de testers estará compuesto por usuarios que asumirán diferentes perfiles (Dirección, Oficina de Personal, Jefatura de Servicio, y Empleado). Cada tester verificará las funcionalidades correspondientes a su perfil para asegurar que los diferentes roles en el sistema estén correctamente implementados.

2. Desarrollador (alumno):

- El desarrollador apoyará el proceso de prueba respondiendo a los errores reportados y realizando correcciones en el código. También participará en la ejecución de pruebas unitarias y pruebas de integración antes de liberar el sistema a los usuarios beta.

3. Usuarios Finales (Beta Testers):

- En una fase final del ciclo de pruebas, usuarios finales del hospital (especialmente seleccionados) podrían participar en una fase de pruebas beta o de aceptación del usuario (UAT), donde verifican que el sistema cumpla con las expectativas y requisitos desde su perspectiva de uso.

Planificación y Calendario de Pruebas

En esta sección se define el cronograma de ejecución de las pruebas, incluyendo las diferentes etapas del desarrollo en las que se realizarán pruebas y la planificación de cada fase.

1. Pruebas Unitarias:

- Las **pruebas unitarias** serán realizadas por los desarrolladores inmediatamente después de implementar cada funcionalidad. Estas pruebas se ejecutan en el entorno de desarrollo para verificar que cada módulo de código funcione correctamente de forma individual.
 - **Duración estimada:** Continúa durante todo el ciclo de desarrollo.
2. **Pruebas de Integración:**
- Después de que varios módulos hayan sido implementados y pasen las pruebas unitarias, se ejecutarán las **pruebas de integración** para verificar que los módulos se comuniquen correctamente entre sí.
 - **Duración estimada:** Fase intermedia del desarrollo.
3. **Pruebas Funcionales:**
- Las pruebas funcionales se llevarán a cabo cuando las funcionalidades clave del sistema estén completas. Estas pruebas se enfocarán en validar que el sistema cumpla con los requerimientos funcionales especificados.
 - Duración estimada: Durante el ciclo de desarrollo y previo al lanzamiento de la versión de prueba.
4. **Pruebas de Rendimiento:**
- Estas pruebas se realizarán para evaluar el rendimiento del sistema bajo diferentes condiciones de carga. Se simulará un gran número de solicitudes y usuarios concurrentes para comprobar la estabilidad y capacidad de respuesta del sistema.
 - **Duración estimada:** Previo a la entrega final.
5. **Pruebas de Usuario (UAT):**
- Antes del despliegue final, se realizarán pruebas de aceptación con usuarios finales para validar que el sistema cumpla con las expectativas de los usuarios y se ajuste a los casos de uso reales del hospital.
 - **Duración estimada:** Fase final antes del despliegue.
6. **Pruebas de Regresión:**
- Se ejecutarán pruebas de regresión después de cada corrección de errores o cambios en el sistema, para asegurar que las funcionalidades previamente aprobadas no se vean afectadas.
 - **Duración estimada:** Tras cada actualización significativa.

Automatización de pruebas

En esta etapa del proyecto, no se implementarán pruebas automatizadas. Aunque el desarrollador tiene un interés personal en explorar herramientas como **JUnit** para pruebas unitarias e integración en Java, así como en el uso de **Jenkins** para la ejecución automática de pruebas periódicas o tras cada modificación del código, actualmente no existen planes concretos ni recursos suficientes, tanto en términos de personal como de tiempo, para integrar dichas pruebas en el proceso de desarrollo. Sin embargo, se contempla la posibilidad de implementar estas herramientas en fases futuras del proyecto, dependiendo de la evolución de los recursos y la necesidad de automatización.

Casos de Prueba

Según la norma ISO 29119, se recomienda que, por cada caso de pruebas se cuente con un:

- Identificador único
- Objetivo
- Prioridad
- Trazabilidad (aunque se aclara que puede llevarse en un documento aparte, como una matriz de trazabilidad)
- Precondiciones
- Entradas
- Resultados esperados
- Resultados Actuales (describimos los resultados reales de la ejecución del caso de prueba).

Para construir las tablas de Casos de Prueba, se toman en cuenta algunas de estas recomendaciones y, además, se incorporan otras que resultan necesarias durante el desarrollo de los casos de prueba.

Se expone a continuación entonces, la relación entre Casos de Prueba y Requerimientos Funcionales y Requerimientos No Funcionales, a través de la siguiente matriz:

Matriz de Trazabilidad de Casos de Pruebas vs. Requerimientos (Func. y No Func.)

CP vs Req	RF01	RF02	RF03	RF04	RF05	RF06	RF08	RF09	RF10	RF11	RNF01	RNF02	RNF03	RNF04	RNF05	RNF06	RNF07	RNF09
CP01									✓		✓	✓		✓	✓		✓	
CP02	✓									✓		✓						
CP03	✓	✓	✓					✓	✓			✓						
CP04	✓		✓	✓	✓	✓	✓			✓				✓	✓		✓	
CP05				✓	✓	✓				✓		✓					✓	
CP06			✓			✓		✓	✓					✓				
CP07		✓									✓		✓		✓		✓	

NOTA: se omiten los requisitos que no resultan relevantes para los casos de prueba propuestos.

Tabla de CP01: Autenticación en el inicio de sesión

ID	Descripción	Precondiciones	Entradas	Pasos	Resultados esperados	Resultados observados	Estado	Prioridad
1	Iniciar sesión dejando todos los campos vacíos	La aplicación debe estar sin datos previos (recién instalada)	Ninguna	1. Abrir la aplicación desde su acceso directo. 2. Tocar el botón INGRESAR en la pantalla de inicio (sin haber llenado ningún campo)	Al estar vacíos los campos de inicio de sesión, el botón INGRESAR, no debería habilitarse, y por ende no debería responder en absoluto.	El botón de inicio de sesión no se habilita hasta que los campos de CUIL y contraseña contengan al menos un carácter.	EXITOSA	ALTA
2	Iniciar sesión ingresando solo el nombre de usuario, dejando vacío el campo de contraseña	La aplicación debe estar sin datos previos (recién instalada)	Solo nombre de usuario válido y ningún otro dato	1. Abrir la aplicación desde su acceso directo. 2. Introducir el nombre de usuario 3. Tocar el botón INGRESAR en la pantalla de inicio (sin haber ingresado ninguna contraseña)	Al no estar completos los campos de inicio de sesión, el botón INGRESAR, no debería habilitarse, y por ende no debería responder en absoluto.	Teniendo un CUIL cargado en el campo de CUIL, el botón de inicio de sesión no se habilita hasta que se haya ingresado al menos un carácter en el capo de contraseña.	EXITOSA	ALTA
3	Iniciar sesión ingresando solo el nombre de usuario, y una contraseña inválida	La aplicación debe estar sin datos previos (recién instalada)	Nombre de usuario y contraseña inválida	1. Abrir la aplicación desde su acceso directo. 2. Introducir el nombre de usuario 3. Introducir una contraseña incorrecta 4. Tocar el botón INGRESAR en la pantalla de inicio de sesión	Mostrar por pantalla mensaje de error indicando que 'la contraseña ingresada es inválida'.	Se indica un mensaje: "¡Contraseña incorrecta o usuario no encontrado!" y se explicita el CUIL ingresado en el mismo mensaje.	EXITOSA	CRÍTICA

Sistema de Gestión de Ausentismo Hospitalario

ID	Descripción	Precondiciones	Entradas	Pasos	Resultados esperados	Resultados observados	Estado	Prioridad
4	Iniciar sesión ingresando un nombre de usuario existente, y una contraseña cualquiera	La aplicación debe estar sin datos previos (recién instalada)	Nombre de usuario existente y una contraseña cualquiera	1. Abrir la aplicación desde su acceso directo. 2. Introducir un nombre de usuario existente 3. Introducir una contraseña cualquiera 4. Tocar el botón INGRESAR en la pantalla de inicio de sesión	Mostrar por pantalla mensaje de error indicando que 'el CUIL ingresado no se corresponde a ningún usuario activo en el sistema'.	Se indica un mensaje: "¡Contraseña incorrecta o usuario no encontrado!" y se explicita el CUIL ingresado en el mismo mensaje.	EXITOSA	ALTA

Tabla de CP02: Gestión de las asignaciones de empleados a servicios

ID	Descripción	Precondiciones	Entradas	Pasos	Resultados esperados	Resultados observados	Estado	Prioridad
1	Usuario de tipo empleado intenta asignar empleados a un servicio	El usuario debe haber iniciado sesión con credenciales válidas, aunque sin permisos de asignación de empleados	Nombre de usuario y contraseña válidos	1. En el menú de la pantalla principal (o dashboard), seleccionar la opción «Asignación de empleados a Servicios» e intentar realizar una asignación.	El sistema no debería mostrar la opción de asignación de empleados a un usuario de tipo Empleado. Si de alguna manera, el usuario lograra alcanzar dicha opción, el sistema debería mostrar un mensaje indicando que no tiene permisos para la acción solicitada.	Pendiente	PENDIENTE	ALTA

Sistema de Gestión de Ausentismo Hospitalario

ID	Descripción	Precondiciones	Entradas	Pasos	Resultados esperados	Resultados observados	Estado	Prioridad
2	Usuario de tipo OficinaDePersonal intenta asignar empleados a un servicio	El usuario debe haber iniciado sesión con credenciales válidas, y debe ser de tipo OficinaDePersonal	Nombre de usuario y contraseña válidos	1. En el menú de la pantalla principal (o dashboard), seleccionar la opción «Asignación de empleados a Servicios» e intentar realizar una asignación.	El sistema permite asignar empleados a servicios y muestra un mensaje con los detalles de la operación solicitando confirmar la acción. Al confirmar la acción, realiza la asignación correctamente.	Pendiente.	PENDIENTE	CRÍTICA
3	Asignar un empleado ya asignado a otro servicio	El usuario debe haber iniciado sesión con credenciales válidas, y debe contar con permisos de asignación de empleados	Empleado a ser asignado a un servicio. Servicio de destino.	1. En el menú de la pantalla principal (o dashboard), seleccionar la opción «Asignación de empleados a Servicios». 2. Seleccionar un empleado ya asignado a otro servicio 3. Intentar reasignarlo a otro servicio.	El sistema bloquea la acción y muestra un mensaje de error indicando que el empleado ya está asignado. Adicionalmente, ofrece la opción de enviar un pedido de autorización a la Dirección y los jefes de los servicios involucrados, para desbloquear la acción.	Pendiente.	PENDIENTE	ALTA
4	Asignar múltiples empleados en un solo paso	El usuario debe haber iniciado sesión con credenciales válidas, y debe contar con permisos de asignación de empleados	Listado de Empleados a asignar a un servicio. Servicio de destino.	1. En el menú de la pantalla principal (o dashboard), seleccionar la opción «Asignación de empleados a Servicios». 2. Seleccionar múltiples empleados. 3. Asignar en lote a un servicio de destino.	El sistema muestra un mensaje con los detalles de la operación: listado enumerado de empleados y servicio de destino, solicitando confirmar la acción. Al confirmar la acción, realiza la asignación en lote correctamente.	Pendiente.	PENDIENTE	ALTA

ID	Descripción	Precondiciones	Entradas	Pasos	Resultados esperados	Resultados observados	Estado	Prioridad
5	Desasignar un empleado de un servicio	El usuario debe haber iniciado sesión con credenciales válidas, y debe contar con permisos de asignación de empleados	Empleado ya asignado a un servicio.	1. En el menú de la pantalla principal (o dashboard), seleccionar la opción «Asignación de empleados a Servicios». 2. Seleccionar un usuario perteneciente a un servicio. 3. Intentar desasignar el empleado de alguna forma.	El sistema no debe permitir empleados sin servicio una vez que fueron asignados a un servicio, y muestra el mensaje de error correspondiente.	Pendiente.	PENDIENTE	BAJA

Tabla de CP03: Generación de diagramas de servicio

ID	Descripción	Precondiciones	Entradas	Pasos	Resultados esperados	Resultados observados	Estado	Prioridad
1	Generar diagrama con todos los empleados asignados	1. Sesión iniciada con perfil de Jefatura de Servicio. 2. Servicios y empleados cargados en el sistema.	Empleados asignados al servicio. Fechas y Horarios de jornadas laborales.	1. En el menú de la pantalla principal (o dashboard), seleccionar la opción «Diagrama de Servicio». 2. Generar el diagrama asignando todos los empleados a turnos y guardias.	El diagrama se genera correctamente y es enviado a Oficina de Personal para validaciones posteriores. El detalle de la operación se muestra en un mensaje por pantalla.	Pendiente	PENDIENTE	CRÍTICA

Sistema de Gestión de Ausentismo Hospitalario

ID	Descripción	Precondiciones	Entradas	Pasos	Resultados esperados	Resultados observados	Estado	Prioridad
2	Intentar generar un diagrama sin empleados asignados	1. Sesión iniciada con perfil de Jefatura de Servicio. 2. Servicios cargados, pero sin empleados asignados	Jefe de servicio. Fechas y Horarios de jornadas laborales.	1. En el menú de la pantalla principal (o dashboard), seleccionar la opción «Diagrama de Servicio». 2. Intentar generar un diagrama sin más empleados que el propio jefe de servicio.	El sistema muestra una advertencia porque solo hay un empleado (el propio jefe de servicio), pero permite generar un diagrama de servicio unipersonal (ya que los jefes se autoasignan su propio horario).	Pendiente.	PENDIENTE	MEDIA
3	Generar diagrama sin la totalidad de los empleados del servicio	1. Sesión iniciada con perfil de Jefatura de Servicio. 2. Servicios y empleados cargados en el sistema.	Solo algunos empleados asignados al servicio (pero no la totalidad de ellos). Fechas y Horarios de jornadas laborales.	1. En el menú de la pantalla principal (o dashboard), seleccionar la opción «Diagrama de Servicio». 2. Generar el diagrama con solo algunos empleados asignados a turnos.	El sistema muestra un mensaje de error, advirtiendo que hay empleados sin asignación horaria. Pero permite guardar el borrador de la asignación para completarla a futuro.	Pendiente.	PENDIENTE	MEDIA
4	Generar diagrama y modificar horarios previamente asignados	1. Sesión iniciada con perfil de Jefatura de Servicio. 2. Servicios y empleados cargados en el sistema.	Empleados. Fechas y Horarios de jornadas laborales. Diagrama previamente generado.	1. En el menú de la pantalla principal (o dashboard), seleccionar la opción «Diagrama de Servicio». 2. Intentar modificar el diagrama en curso.	El sistema debe mostrar el diagrama en curso (y cualquier anterior), pero informa que o no se permite modificar diagramas en curso o pasados.	Pendiente.	PENDIENTE	MEDIA

Tabla de CP04: Carga de marcaciones

ID	Descripción	Precondiciones	Entradas	Pasos	Resultados esperados	Resultados observados	Estado	Prioridad
1	Cargar marcaciones con datos completos y correctos	1. Sesión iniciada con perfil de Oficina de Personal. 2. Servicios y empleados cargados en el sistema.	Listado de marcaciones completo y válido	1. En el menú principal, seleccionar «Cargar marcaciones». 2. Cargar archivo de marcaciones válido. 3. Confirmar la carga.	Las marcaciones se cargan correctamente sin errores y se actualiza el registro del personal.	Pendiente	PENDIENTE	CRÍTICA
2	Intentar cargar marcaciones con empleados no registrados	1. Sesión iniciada con perfil de Oficina de Personal.	Listado de marcaciones con empleados no registrados	1. En el menú principal, seleccionar «Cargar marcaciones». 2. Cargar archivo con empleados no registrados. 3. Confirmar la carga.	El sistema bloquea la carga y muestra un mensaje de error indicando que el archivo de marcaciones contiene empleados que no están registrados.	Pendiente.	PENDIENTE	CRÍTICA
3	Cargar marcaciones con datos faltantes (ej: hora de salida faltante)	1. Sesión iniciada con perfil de Oficina de Personal.	Listado de marcaciones con omisiones de INGRESO/EGRESO	1. En el menú principal, seleccionar «Cargar marcaciones». 2. Cargar archivo con datos faltantes. 3. Confirmar la carga.	El sistema realiza la carga sin problemas, ya que es una situación prevista y soportada. Adicionalmente, informa por pantalla que el archivo de marcaciones contenía potenciales omisiones.	Pendiente.	PENDIENTE	CRÍTICA
4	Cargar un archivo de marcaciones en formato incorrecto	1. Sesión iniciada con perfil de Oficina de Personal.	Archivo en formato no compatible	1. En el menú principal, seleccionar «Cargar marcaciones». 2. Cargar archivo en formato incorrecto. 3. Confirmar la carga.	El sistema rechaza la carga y muestra un mensaje de error indicando “Formato no válido o no reconocido. Intentar nuevamente o contactar a Informática”.	Pendiente.	PENDIENTE	ALTA

ID	Descripción	Precondiciones	Entradas	Pasos	Resultados esperados	Resultados observados	Estado	Prioridad
5	Cargar marcaciones duplicadas	1. Sesión iniciada con perfil de Oficina de Personal.	Listado de marcaciones con duplicados	1. En el menú principal, seleccionar «Cargar marcaciones». 2. Cargar archivo con duplicados. 3. Confirmar la carga.	El sistema realiza la carga sin problemas, ya que es una situación prevista y soportada. Adicionalmente, informa por pantalla que se compactaron las marcaciones duplicadas.	Pendiente.	PENDIENTE	BAJA
6	Cargar marcaciones de empleados que ya tienen registradas horas para el mismo día	1. Sesión iniciada con perfil de Oficina de Personal. 2. Empleados con horas registradas.	Listado de marcaciones con empleados que ya tienen marcaciones para el mismo día.	1. Seleccionar «Cargar marcaciones» 2. Cargar archivo con conflictos de horas. 3. Confirmar la carga.	El sistema advierte que ya se cargaron marcaciones para los usuarios, y muestra el listado de marcaciones existentes. No obstante, ofrece volver a cargar, descartando las existentes.	Pendiente.	PENDIENTE	MEDIA

Tabla de CP05: Generación de Solicitudes de empleados: Horas extra / Francos compensatorios / Faltas justificadas

ID	Descripción	Precondiciones	Entradas	Pasos	Resultados esperados	Resultados observados	Estado	Prioridad
1	Generar solicitud de horas extra con datos correctos	1. Sesión iniciada con perfil de Empleado. 2. Empleado asignado a un servicio y con horas extra disponibles.	Horas extra trabajadas por el empleado	1. Seleccionar «Solicitar Horas Extra». 2. Introducir las horas extra y confirmar la solicitud.	La solicitud de horas extra se registra correctamente y queda pendiente de aprobación.	Pendiente	PENDIENTE	ALTA

Sistema de Gestión de Ausentismo Hospitalario

ID	Descripción	Precondiciones	Entradas	Pasos	Resultados esperados	Resultados observados	Estado	Prioridad
2	Generar solicitud de Franco Compensatorio (FC) con horas extra disponibles	1. Sesión iniciada con perfil de Empleado. 2. Empleado con horas extra registradas y disponibles.	1. Fecha de solicitud del FC. 2. Horas extra acumuladas.	1. Seleccionar «Solicitar Franco Compensatorio» 2. Introducir la fecha en la cual usufructuar el FC. 3. Confirmar cantidad de horas extras necesarias para conformar el FC. 3. Tocar «Generar y enviar solicitud».	El FC se genera correctamente, descontando las horas extra disponibles (marcándolas como pendientes de usufructo), y la solicitud se envía a la jefatura de servicio para su aprobación.	Pendiente.	PENDIENTE	ALTA
3	Intentar generar solicitud de FC sin suficientes horas extra disponibles	1. Sesión iniciada con perfil de Empleado. 2. Empleado sin horas extra disponibles.	1. Fecha de solicitud de FC. 2. Horas extra acumuladas.	1. Seleccionar «Solicitar Franco Compensatorio». 2. Intentar registrar la solicitud sin horas disponibles.	El sistema no habilita el botón de «Generar y enviar solicitud», y muestra un mensaje de error indicando la cantidad de horas necesarias, las disponibles, y las faltantes.	Pendiente.	PENDIENTE	CRÍTICA
4	Generar solicitud de falta justificada (FJ), para una fecha laborable para el empleado.	1. Sesión iniciada con perfil de Empleado. 2. Empleado asignado a un servicio.	1. Fecha de solicitud de FJ para un día laborable para el empleado.	1. Seleccionar «Solicitar Falta Justificada» 2. Indicar la fecha confirmar la solicitud. 3. Tocar «Generar y enviar solicitud».	La solicitud de FJ se genera correctamente y se informa por pantalla que fue enviada a la jefatura de servicio para su aprobación.	Pendiente.	PENDIENTE	ALTA
5	Generar solicitud de falta justificada (FJ), para una fecha no laborable por el empleado.	1. Sesión iniciada con perfil de Empleado. 2. Empleado asignado a un servicio.	1. Fecha de solicitud de FJ para un día no laborable para el empleado.	1. Seleccionar «Solicitar Falta Justificada». 2. Indicar la fecha confirmar la solicitud. 3. Tocar «Generar y enviar solicitud».	El sistema bloquea la solicitud y muestra un mensaje de error indicando que la FJ no es aplicable para la fecha seleccionada. Solicitud verificar la fecha.	Pendiente.	PENDIENTE	CRÍTICA

Tabla de CP06: Validación y aprobación de solicitudes de empleados: Horas extra / Francos compensatorios / Faltas justificadas

ID	Descripción	Precondiciones	Entradas	Pasos	Resultados esperados	Resultados observados	Estado	Prioridad
1	Validar y aprobar una solicitud de horas extra	1. Sesión iniciada con perfil de Oficina de Personal o Jefatura de Servicio. 2. Solicitud de horas extra pendiente de validación.	Solicitud de aprobación de horas extra	1. Acceder al módulo de «Aprobación / validación de solicitudes». 2. Revisar la solicitud de horas extra. 3. Aprobar la solicitud.	La solicitud se aprueba correctamente y se actualizan las horas extra del empleado. Se envía una notificación de la aprobación al empleado que originó la solicitud.	Pendiente	PENDIENTE	ALTA
2	Validar y rechazar una solicitud de horas extra	1. Sesión iniciada con perfil de Oficina de Personal o Jefatura de Servicio. 2. Solicitud de horas extra pendiente de validación.	Solicitud de aprobación horas extra	1. Acceder al módulo de «Aprobación / validación de solicitudes». 2. Revisar la solicitud de horas extra. 3. Rechazar la solicitud.	La solicitud se rechaza correctamente y el empleado recibe una notificación del rechazo.	Pendiente.	PENDIENTE	ALTA
3	Validar y aprobar una solicitud de francos compensatorios (FC)	1. Sesión iniciada con perfil de Oficina de Personal o Jefatura de Servicio. 2. Solicitud de FC pendiente de validación.	Solicitud de aprobación de FC	1. Acceder al módulo de «Aprobación / validación de solicitudes». 2. Revisar la solicitud de francos compensatorios. 3. Aprobar la solicitud.	La solicitud de FC se aprueba correctamente y se envía notificación al empleado que originó la solicitud.	Pendiente.	PENDIENTE	ALTA

Sistema de Gestión de Ausentismo Hospitalario

ID	Descripción	Precondiciones	Entradas	Pasos	Resultados esperados	Resultados observados	Estado	Prioridad
4	Validar y rechazar una solicitud de francos compensatorios (FC)	1. Sesión iniciada con perfil de Oficina de Personal o Jefatura de Servicio. 2. Solicitud de francos compensatorios pendiente de validación.	Solicitud aprobación de FC	1. Acceder al módulo de «Aprobación / validación de solicitudes». 2. Revisar la solicitud. 3. Rechazar la solicitud.	La solicitud de FC se rechaza correctamente y el empleado recibe la notificación.	Pendiente.	PENDIENTE	CRÍTICA
5	Validar y aprobar una solicitud de falta justificada (FJ)	1. Sesión iniciada con perfil de Oficina de Personal o Jefatura de Servicio. 2. Solicitud de FJ pendiente de validación.	Solicitud de aprobación de FJ.	1. Acceder al módulo de «Aprobación / validación de solicitudes». 2. Revisar la solicitud. 3. Aprobar la solicitud.	La FJ se aprueba correctamente y se refleja en el parte diario del empleado, el cual es notificado al instante.	Pendiente.	PENDIENTE	ALTA
6	Validar y rechazar una solicitud de falta justificada (FJ) con motivo	1. Sesión iniciada con perfil de Oficina de Personal o Jefatura de Servicio. 2. Solicitud de FJ pendiente de validación.	Solicitud de aprobación de FJ	1. Acceder al módulo de «Aprobación / validación de solicitudes». 2. Revisar la solicitud. 3. Rechazar la solicitud indicando motivo (obligatorio).	La solicitud se rechaza correctamente y el empleado recibe una notificación indicando la causa.	Pendiente.	PENDIENTE	CRÍTICA

Tabla de CP07: Simulación de sobrecarga: Generación de múltiples solicitudes en paralelo

ID	Descripción	Precondiciones	Entradas	Pasos	Resultados esperados	Resultados observados	Estado	Prioridad
1	Ejecutar 50 solicitudes de inserción de horas extra simultáneas	1. <i>JMeter</i> configurado para conectarse a <i>MariaDB</i> mediante <i>JDBC</i> . 2. Base de datos con tabla de horas extra configurada.	Script de inserción de horas extra para 50 usuarios.	1. Configurar <i>JMeter</i> con <i>JDBC Connection Config</i> . 2. Crear un <i>Thread Group</i> con 50 hilos (usuarios). 3. Ejecutar el script de inserción para generar 50 horas extra.	El sistema debe procesar todas las inserciones sin degradación del rendimiento. Los tiempos de respuesta deben ser < 2s. por solicitud.	Pendiente	PENDIENTE	CRÍTICA
2	Ejecutar 100 solicitudes de actualización de francos compensatorios en paralelo	1. <i>JMeter</i> configurado con <i>JDBC Connection</i> a la BD de pruebas. 2. Tabla de FC en la base de datos.	Script de actualización de FC.	1. Crear un <i>Thread Group</i> con 100 hilos en <i>JMeter</i> . 2. Cargar el script de actualización. 3. Ejecutar las solicitudes en paralelo.	El sistema debe manejar todas las actualizaciones sin errores ni degradación. Los tiempos de respuesta deben ser < 2s.	Pendiente.	PENDIENTE	CRÍTICA
3	Medir tiempo de respuesta para 200 consultas simultáneas de horas trabajadas	1. <i>JMeter</i> configurado con <i>JDBC Connection</i> a <i>MariaDB</i> . 2. Tabla de horas trabajadas con datos ficticios cargados.	Script de consulta de horas trabajadas	1. Configurar <i>JMeter</i> para ejecutar 200 consultas simultáneas. 2. Crear un <i>Thread Group</i> con 200 hilos. 3. Ejecutar el script de consulta.	El sistema debe procesar todas las consultas sin interrupciones. El tiempo de respuesta no debe exceder > 2s por consulta.	Pendiente.	PENDIENTE	ALTA

ID	Descripción	Precondiciones	Entradas	Pasos	Resultados esperados	Resultados observados	Estado	Prioridad
4	Sobrecarga máxima: Ejecutar 500 solicitudes combinadas (inserciones, actualizaciones y consultas) en paralelo	1. <i>JMeter</i> configurado para manejar inserciones, actualizaciones y consultas. 2. <i>MariaDB</i> debe tener tablas con datos ficticios.	Scripts de inserción, actualización y consulta	1. Crear un <i>Thread Group</i> con 500 hilos. 2. Dividir las solicitudes entre inserciones, actualizaciones y consultas. 3. Ejecutar el script de prueba.	El sistema debe procesar todas las solicitudes sin errores. Los tiempos de respuesta deben mantenerse dentro de límites aceptables < 2s.	Pendiente.	PENDIENTE	CRÍTICA
5	Monitoreo de uso de CPU y memoria durante la simulación de sobrecarga	1. <i>JMeter</i> configurado para conectarse a <i>MariaDB</i> . 2. Herramienta de monitoreo de sistema activa.	500 solicitudes combinadas (inserciones, actualizaciones, consultas)	1. Ejecutar las pruebas anteriores (500 solicitudes en paralelo). 2. Monitorear el uso de CPU, memoria y otros recursos del sistema durante la prueba.	El sistema no debe alcanzar un uso crítico de CPU o memoria. Los recursos del servidor deben mantenerse en niveles aceptables <80%.	Pendiente.	PENDIENTE	ALTA

Informe de pruebas realizadas

Tabla de Resultados de pruebas

CP	IDs Exitosos	CANT	IDs Fallidos	CANT	OBSERVACIONES
CP01	1,2,3,4	4	Ninguno	0	El funcionamiento es el esperado en todos los casos.
CP02	1,2,3,4,5		1,2,3,4,5		Pendiente de realización de pruebas.
CP03	1,2,3,4		1,2,3,4		Pendiente de realización de pruebas.
CP04	1,2,3,4,5,6		1,2,3,4,5,6		Pendiente de realización de pruebas.
CP05	1,2,3,4,5		1,2,3,4,5		Pendiente de realización de pruebas.
CP06	1,2,3,4		1,2,3,4		Pendiente de realización de pruebas.
CP07	1,2,3,4,5		1,2,3,4,5		Pendiente de realización de pruebas.
TOTAL	--		--		<i>Total de pruebas planificadas</i>

Explicación:

ID (Exitosa) e ID (Fallida): Los números de prueba exitosos o fallidos (por ejemplo, CP01 tiene 4 pruebas planificadas).

CANT: La cantidad de pruebas exitosas o fallidas.

Tabla de Porcentaje de pruebas

Cuando podamos realizar las pruebas, en la tabla se podrá observar que se encontraron N fallas. La siguiente tabla nos ayudará a calcular el porcentaje de fallas, y ver si está dentro de los valores planteados:

Pruebas exitosas	Pruebas fallidas	Pruebas totales	Porcentaje de éxitos	Porcentaje de fallos
M	N	M+N	$(M/(M+N))*100$	$(N/(M+N))*100$

Hasta el momento del desarrollo, las pruebas indican un **100% de éxito**.

Procedimiento de Corrección

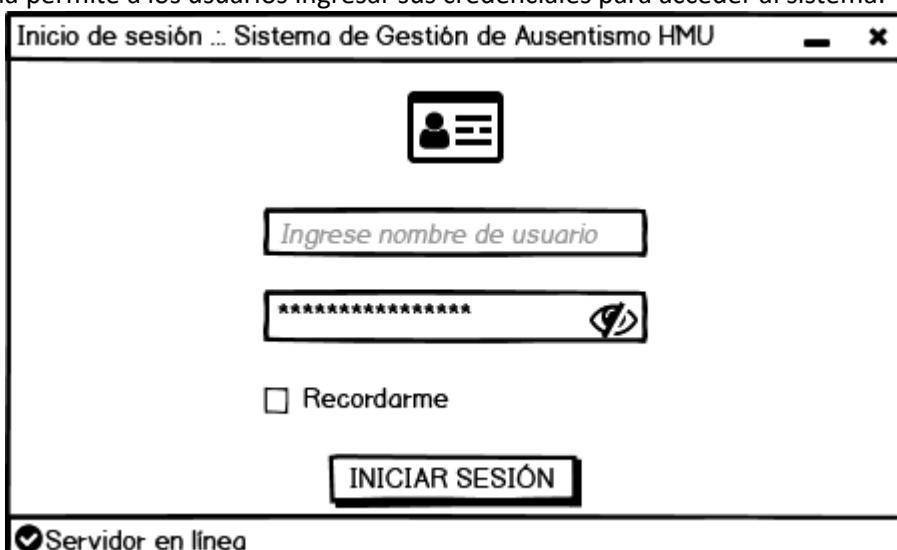
Si el porcentaje de fallos está por encima del umbral definido en los criterios iniciales, se procederá a corregir las fallas detectadas y se repetirá el plan de pruebas para verificar las correcciones en la nueva versión del producto. En caso contrario, el sistema se considerará mínimamente estable y apto para ser lanzado a producción, siempre y cuando no haya defectos críticos.

13. Interfaz gráfica

A continuación, se presentan algunos ejemplos preliminares de las posibles pantallas del sistema, los cuales servirán como base para el desarrollo de la interfaz de usuario:

Pantalla de inicio de sesión

Esta pantalla permite a los usuarios ingresar sus credenciales para acceder al sistema.



La captura de pantalla muestra una ventana de inicio de sesión titulada "Inicio de sesión ... Sistema de Gestión de Ausentismo HMU". En el centro, hay un icono de usuario con tres líneas horizontales. Abajo de él, un cuadro de texto dice "Ingrese nombre de usuario". A continuación, hay un cuadro de texto que muestra asteriscos (*) y un icono de ojo para alternar visibilidad. Debajo de estos campos, hay un checkbox vacío seguido de la etiqueta "Recordarme". A continuación, un botón grande que dice "INICIAR SESIÓN". En la parte inferior, hay un cuadro de texto que dice "Servidor en línea" con un checkmark.

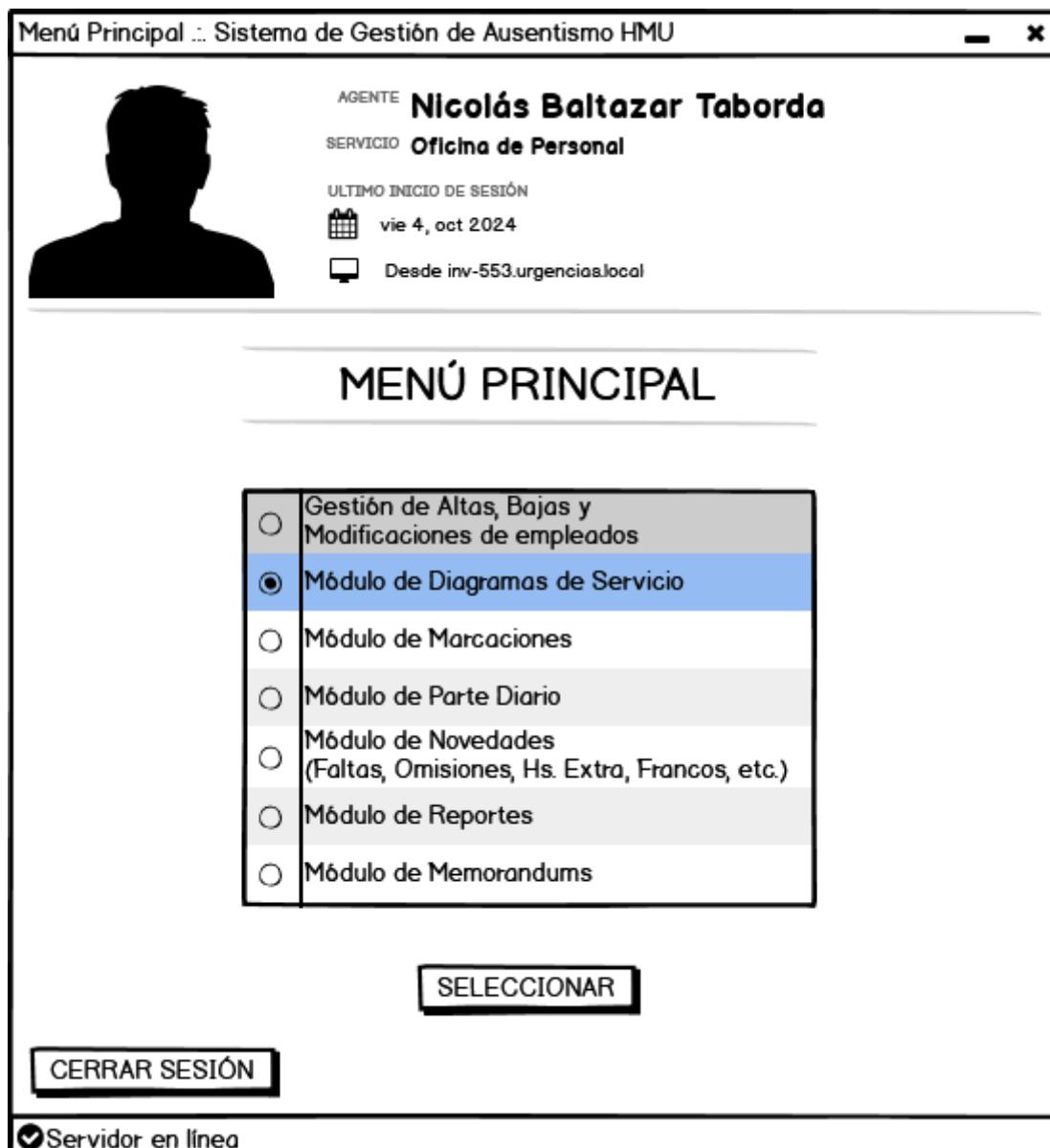
Incluye los siguientes elementos:

- Campo de nombre de usuario.
- Campo de contraseña (con opción de mostrar/ocultar).
- Checkbox para recordar el nombre de usuario solamente (no la contraseña, por cuestiones de seguridad).
- Botón de inicio de sesión.
- Barra de estado inferior que muestra el estado del servidor.

Pantalla de Menú Principal

Esta es la pantalla principal a la que el usuario accede tras iniciar sesión. Está diseñada para ser intuitiva y contiene los siguientes elementos:

- **Información del Usuario:** En la parte superior, se muestra el nombre, el servicio al que pertenece, la fecha del último inicio de sesión y la dirección IP desde la que se está conectando.
- **Cuadro de lista para selección de módulo:** Un menú de donde se puede seleccionar la opción del módulo al cual se desea ingresar.
- **Botón para ingresar al módulo escogido.**
- **Botón de cierre de sesión:** para desconectarse del sistema.
- **Barra de Estado:** En todas las pantallas se indica el estado del servidor (en este caso "Servidor en línea").

**Pantallas del módulo de Diagramas de Servicios**

Las siguientes pantallas, forman parte del módulo de «Generación de Diagramas de Servicio».

Pantalla 1 de 3

La primera etapa de este módulo, permite seleccionar un período correspondiente a un mes particular (lo usual), o a un rango de fechas específico (inusual, por ejemplo, durante la pandemia de COVID-19, los diagramas de servicio se confeccionaban por quincenas).

Diagrama de Servicio ... Sistema de Gestión de Ausentismo HMU

AGENTE **Nicolás Baltazar Taborda**
SERVICIO **Oficina de Personal**

ULTIMO INICIO DE SESIÓN
vie 4, oct 2024
Desde inv-553.urgencias.local

DIAGRAMA DE SERVICIO

1. Fecha | 2. Empleados | 3. Ejecución

Seleccionar un mes completo

MES DEL AÑO
10 :: OCTUBRE

Ó

Seleccionar período personalizado

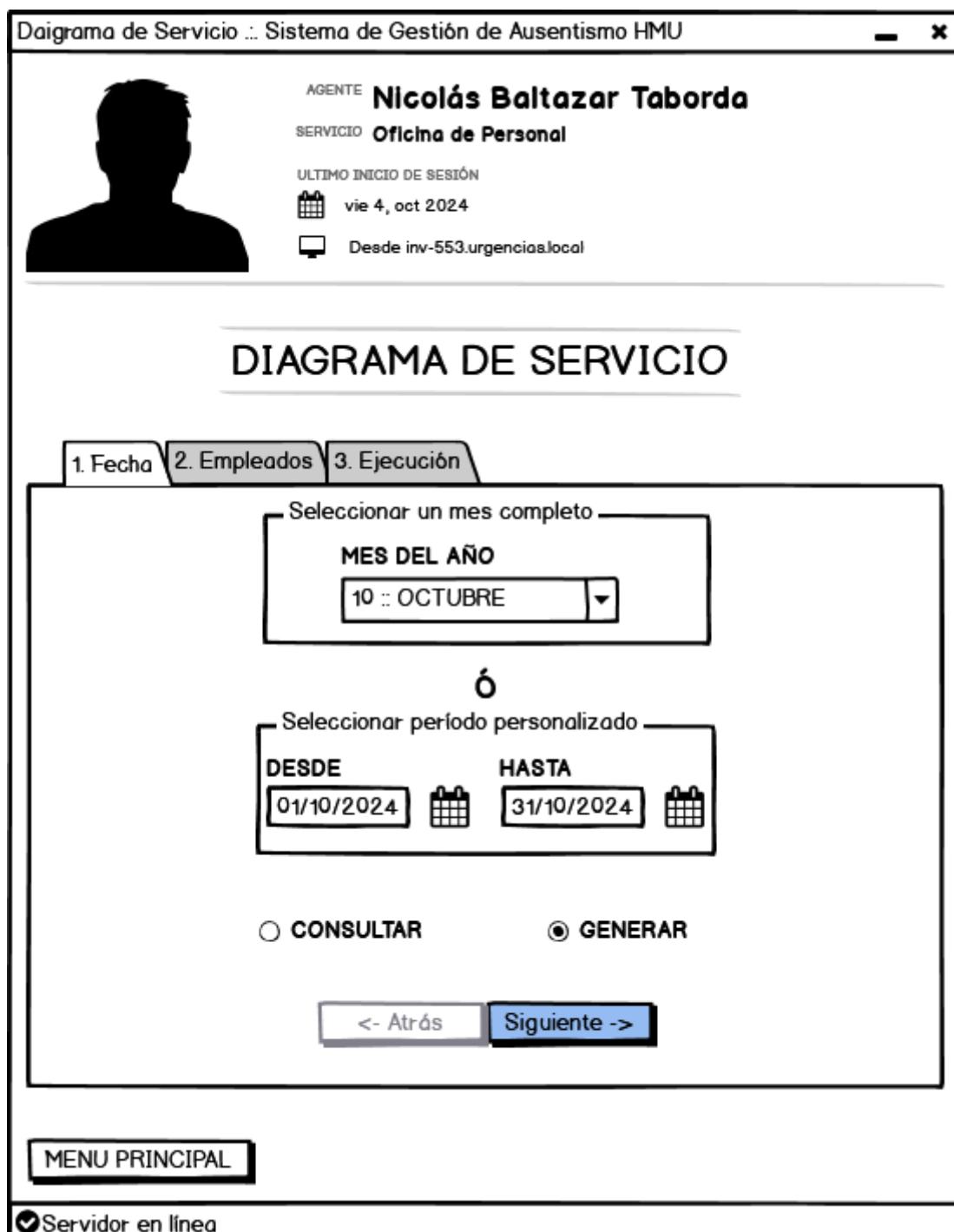
DESDE 01/10/2024 HASTA 31/10/2024

CONSULTAR GENERAR

<- Atrás Siguiente ->

MENU PRINCIPAL

Servidor en línea



Pantalla 2 de 3

La segunda etapa de este módulo, permite seleccionar los empleados que serán asignados a un servicio específico, y el tipo de modalidad horaria que se asignará luego.

Diagrama de Servicio :: Sistema de Gestión de Ausentismo HMU

AGENTE **Nicolás Baltazar Taborda**
SERVICIO **Oficina de Personal**
ULTIMO INICIO DE SESIÓN
vie 4, oct 2024
Desde inv-553.urgencias.local

DIAGRAMA DE SERVICIO

1. Fecha 2. Empleados 3. Ejecución

Empleados disponibles

- Juan Pérez
- Claudia Villafañe
- Leonardo Pinto**
- Andrea Sanders

Agregar ->
<- Remover
Agregar todos
Remover todos

Empleados seleccionados

- Nicolás Baltazar Taborda**
- Patricia Abalos
- Jorgelina Arancibia

MODALIDAD HORARIA PARA ESTA ASIGNACIÓN 35 Hs semanales

<- Atrás Siguiente ->

MENU PRINCIPAL

Servidor en línea



Pantalla 3 de 3

La tercera etapa de este módulo permite seleccionar rangos horarios en un calendario que puede alternar entre las vistas por **Día**, **Semana** o **Mes**.

En la imagen, se muestra la vista semanal (indicando el número de la semana), con un rango de horarios ya seleccionado. La selección se realiza haciendo clic y arrastrando hacia arriba o hacia abajo, delimitando el rango horario deseado. La selección siempre comienza y termina en los minutos **00** o **30** de la hora seleccionada.

El sistema muestra los horarios asignados y calcula automáticamente el total de horas seleccionadas para cada bloque. En la parte inferior, se despliega un mensaje con el **total de horas seleccionadas**, en **verde** si cumplen con la modalidad horaria previamente definida, o en **rojo** si no se ajustan a los requisitos establecidos.

Finalmente, cuando la selección satisface los requisitos de la modalidad horaria, se habilita el botón **Finalizar**. En caso de que falten empleados por asignar (por ejemplo, para otra modalidad horaria), un mensaje notificará que las selecciones realizadas se guardarán como borrador y se ofrecerá la opción de continuar con la asignación ahora o más tarde.

14. Definición de base de datos para el prototipo

Para el desarrollo del **Sistema de Gestión de RRHH del Hospital de Urgencias**, es fundamental contar con una base de datos relacional que garantice la persistencia y gestión eficiente de los datos. En cumplimiento con los requerimientos no funcionales, el prototipo emplea una base de datos **MariaDB**. Si bien la cátedra exige el uso de *MySQL*, *MariaDB* se ha seleccionado por estar basada en *MySQL*, manteniendo su rendimiento, flexibilidad y escalabilidad, pero sin los problemas de licencia asociados con *MySQL*. Esta elección asegura compatibilidad con el motor originalmente propuesto, manteniendo la calidad y desempeño requeridos para el sistema.

La persistencia de datos clave en el sistema es esencial, ya que abarca la gestión del personal, los diagramas de servicio y los registros del parte diario, entre otras entidades fundamentales. Estos componentes constituyen el núcleo del funcionamiento del sistema, permitiendo la correcta diagramación de servicios, así como la gestión de solicitudes, autorizaciones y validaciones de horas extras, francos compensatorios, y faltas justificadas, entre otras novedades. Además, facilitan la generación de partes diarios semanales y mensuales, junto con otros reportes personalizados, que son una necesidad primordial para la Oficina de Personal, el área responsable de los recursos humanos en el hospital.

En sistemas complejos, como el abordado en esta actividad, se puede comenzar a plantear un prototipo que “es la creación de un modelo operacional que incluya solo algunas características del sistema final” (Kendall y Kendall, 2011). De acuerdo con los lineamientos del PUD, las clases identificadas en la fase de análisis estarán directamente relacionadas con las tablas que se deben crear en la base de datos. Esto incluye entidades como **Usuario**, **Servicio**, **ParteDiario**, **Horario** y **Novedad**, entre otras, que permiten asegurar la integridad y consistencia de los datos a lo largo del tiempo.

Para las claves primarias (**PK**) de las tablas, se decidió utilizar **UUID**² (eliminando los guiones) en vez de **INT AUTO_INCREMENT** (enteros auto incrementables). Por esta razón, las **PK** se definen y almacenan como **BINARY(16)**.

¿Por qué eliminar los guiones? Bueno, en el formato estándar un UUID tiene 36 caracteres: 32 hexadecimales y 4 guiones (por ejemplo: 550e8400-e29b-41d4-a716-446655440000). Estos guiones son solo para mejorar la legibilidad del UUID cuando se almacena como texto, pero no son parte esencial del identificador.

Almacenarlo sin guiones en una base de datos tiene varias ventajas:

² **UUID** (Universally Unique Identifier) y **GUID** (Globally Unique Identifier) son esencialmente lo mismo. Ambos son identificadores únicos que se utilizan para distinguir registros de manera global, y la diferencia principal radica en el nombre: **UUID** es más común en entornos no *Microsoft*, mientras que **GUID** es el término usado principalmente por *Microsoft*. Ambos tienen el mismo formato y cumplen la misma función.

1. **Ahorro de espacio:** Al eliminar los guiones, se puede reducir el tamaño de almacenamiento de 36 caracteres a 32 caracteres. Al mismo tiempo, si se almacena el UUID en formato binario (BINARY(16)), se logra ahorrar aún más espacio en comparación con el formato de texto.
2. **Rendimiento:** Operaciones como comparaciones o búsquedas son más rápidas si el UUID se almacena en formato binario en lugar de texto, porque se está procesando un tipo de dato más compacto.

Por esta razón (y como se verá más adelante en la etapa de inserción de registros), en lugar de generar los valores automáticamente (como lo haríamos con AUTO_INCREMENT si los **PK** fuera del tipo INT), asignamos el UUID mediante la función **UUID()** de MySQL al insertar registros. Se definieron un par de funciones para lidiar con la conversión de texto a binario y viceversa.

En resumen, el diseño de la base de datos para el prototipo busca asegurar la correcta gestión de la información relacionada con la administración del personal hospitalario, lo que sienta las bases para su uso en producción una vez completada la fase de pruebas.

15. Diagrama entidad-relación

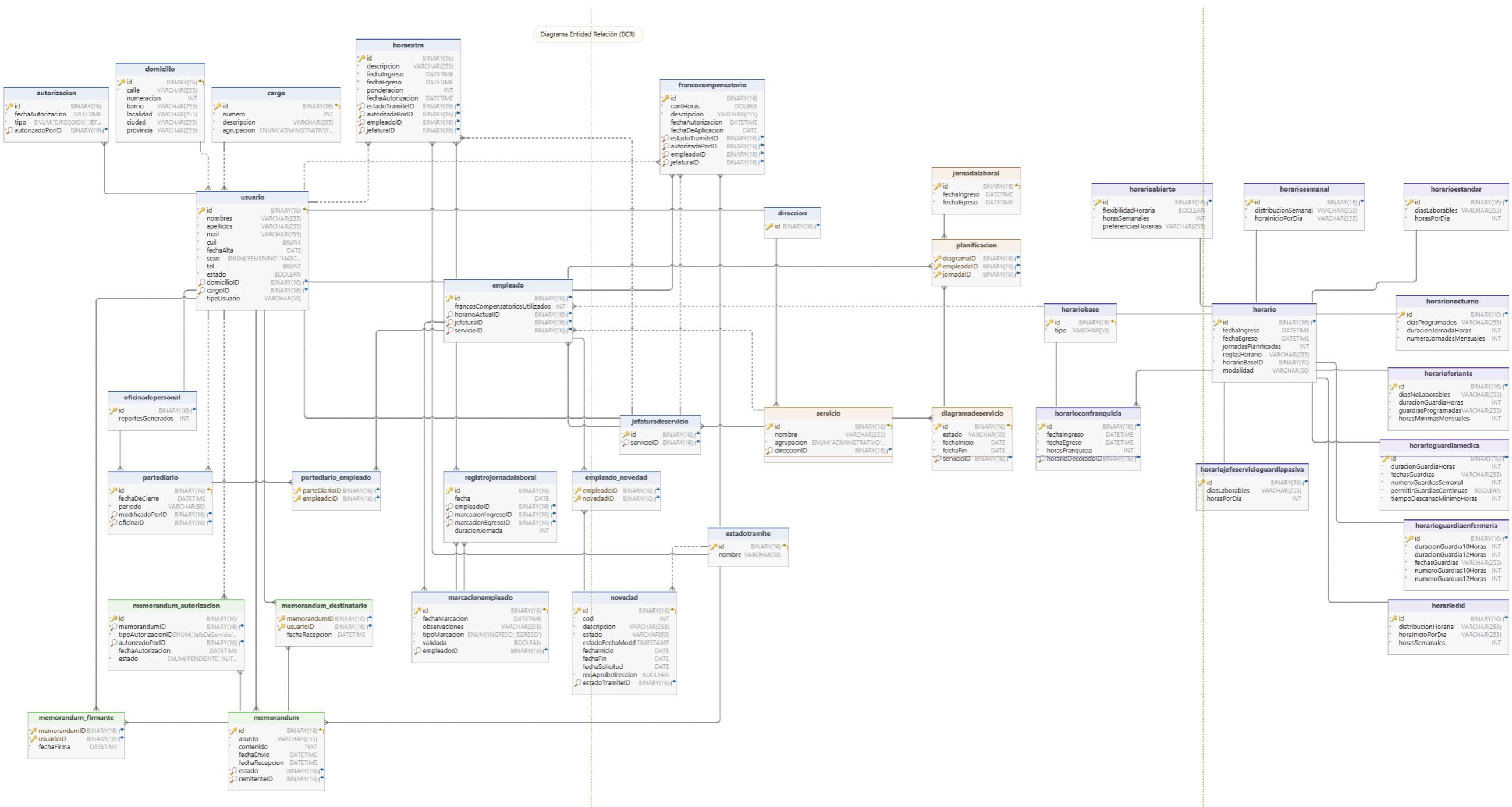
Como se mencionó anteriormente, un diagrama de entidad-relación (DER) es una representación gráfica de la estructura de una base de datos, que facilita la visualización de cómo se relacionan entre sí las diferentes entidades o tablas, así como las interacciones entre ellas. La correcta definición de la estructura es clave, ya que impacta directamente en el rendimiento y la integridad del sistema.

Conceptos que se tuvieron en cuenta al momento de armar el DER:

- **Entidades.**
- **Atributos.**
- **Relaciones.**
- **Claves primarias (PK) y foráneas (FK)**
- **Cardinalidad:** Se presta especial atención al tratamiento de las relaciones mucho a muchos.

La base de datos en la que se crearán las tablas se denomina «**aromito**», en homenaje a la *Acacia caven*, el árbol nativo más común en las Sierras de Córdoba. Aunque se contempla el diseño completo del sistema, para este prototipo se han definido veinticuatro tablas que permiten gestionar los atributos principales de los usuarios, los servicios y las novedades.

Sistema de Gestión de Ausentismo Hospitalario



Disponible en otros formatos en [GitHub](#):

Imagen PNG: <https://raw.githubusercontent.com/linkstat/seminare/refs/heads/main/diagrams/DiagramaEntidadRelacion.png>

Documento PDF: <https://github.com/linkstat/seminare/blob/main/diagrams/DiagramaEntidadRelacion.pdf>

Fuente (DbSchema): <https://github.com/linkstat/seminare/blob/main/diagrams/DiagramaEntidadRelacion.dbs>

16.Creación de tablas

Las tablas se crearon siguiendo el Diagrama de Clases antes diseñado. Durante el proceso de creación de las tablas, se dieron algunos replanteos, que obligaron a modificar el diagrama de clases, y en algunos casos, se tuvieron que implementar tablas intermedias para salvaguardar las relaciones mucho a muchos. Como ya se mencionó, una decisión de diseño e implementación del esquema de datos, fue la de utilizar UUID como claves primarias y almacenarlos como datos binarios.

El script SQL con las todas las sentencias para la creación de todas las tablas de la base de datos que se detallan a continuación en este documento, se puede descargar desde GitHub en:

<https://github.com/linkstat/seminare/tree/main/sql>

Definición de funciones personalizadas

Dado que se decidió utilizar UUID y almacenarlos en formato binario, contar con una función que facilite la inserción y consulta de estos datos, nos facilitará enormemente la vida.

Las funciones propuestas³, usan la representación estándar de UUID y no la propuesta de MySQL 8.0+ (UUID_TO_BIN(uuid, swap_flag=1)). Es decir, los bytes no se reordenan.

Gracias a esto, se puede utilizar la biblioteca java.util.UUID para leer y manipular los UUID de la base de datos sin problemas, ya que son totalmente compatibles.

Función para convertir (y almacenar) un UUID en un binario de 16 bytes

```
-- Función para convertir (y almacenar) un UUID en un binario de 16 bytes
DELIMITER $$

CREATE FUNCTION `UUID_TO_BIN`(uuid CHAR(36))
RETURNS BINARY(16)
DETERMINISTIC
BEGIN
    RETURN UNHEX(CONCAT(
        SUBSTRING(uuid, 1, 8),      -- aaaaaaaaaa
        SUBSTRING(uuid, 10, 4),     -- bbbb
        SUBSTRING(uuid, 15, 4),     -- cccc
        SUBSTRING(uuid, 20, 4),     -- dddd
        SUBSTRING(uuid, 25, 12)     -- eeeeeeeeeeee
    ));
END$$
DELIMITER ;
```

³ Rediseño en base a las propuestas en: <https://mariadb.com/kb/en/uuid-data-type/>

Función para recuperar y convertir de nuevo a UUID

```
-- Función para recuperar y convertir de nuevo a UUID
DELIMITER $$

CREATE FUNCTION `BIN_TO_UUID`(b BINARY(16))
RETURNS CHAR(36) CHARSET ascii
DETERMINISTIC
BEGIN
    DECLARE hexStr CHAR(32);
    SET hexStr = HEX(b);
    RETURN LOWER(CONCAT(
        SUBSTR(hexStr, 1, 8), '-',
        SUBSTR(hexStr, 9, 4), '-',
        SUBSTR(hexStr, 13, 4), '-',
        SUBSTR(hexStr, 17, 4), '-',
        SUBSTR(hexStr, 21, 12)
    ));
END$$
DELIMITER ;
```

Tabla HorarioBase

```
/* Tabla HorarioBase
 * Implementación de la Herencia entre Horarios
 * Utilizamos la estrategia de Tabla por Subclase, donde:
 * HorarioBase tabla base que contiene el identificador y el tipo de horario.
 * Horario y HorarioConFranquicia son subclases que extienden HorarioBase.
 */
CREATE TABLE HorarioBase (
    id BINARY(16) PRIMARY KEY,
    tipo VARCHAR(50) NOT NULL -- 'Horario', 'HorarioConFranquicia'
);
```

Tabla Horario

```
/* Tabla Horario
 * Agregamos una columna 'modalidad' para identificar el tipo de horario y
facilitar consultas:
 * Posibles valores: 'HorarioEstandar', 'HorarioSemanal', 'HorarioNocturno',
 * 'HorarioFeriante', 'HorarioDXI', 'HorarioGuardiaEnfermeria',
'HorarioGuardiaMedica',
 * 'HorarioJefeServicioGuardiaPasiva', 'HorarioAbierto'
*/
CREATE TABLE Horario (
    id BINARY(16) PRIMARY KEY,
    fechaIngreso DATETIME NOT NULL,
```

```
fechaEgreso DATETIME NOT NULL,
jornadasPlanificadas INT,
reglasHorario VARCHAR(255),
horarioBaseID BINARY(16) NOT NULL,
FOREIGN KEY (id) REFERENCES HorarioBase(id),
modalidad VARCHAR(50) NOT NULL
);
```

Tabla HorarioConFranquicia

```
-- Tabla HorarioConFranquicia
CREATE TABLE HorarioConFranquicia (
    id BINARY(16) PRIMARY KEY,
    fechaIngreso DATETIME NOT NULL,
    fechaEgreso DATETIME NOT NULL,
    horasFranquicia INT NOT NULL,
    horarioDecoradoID BINARY(16) NOT NULL,
    FOREIGN KEY (id) REFERENCES HorarioBase(id),
    FOREIGN KEY (horarioDecoradoID) REFERENCES Horario(id)
);
```

Tabla HorarioEstandar

```
-- Tabla HorarioEstandar
CREATE TABLE HorarioEstandar (
    id BINARY(16) PRIMARY KEY,
    diasLaborables VARCHAR(255) NOT NULL,
    horasPorDia INT NOT NULL,
    FOREIGN KEY (id) REFERENCES Horario(id)
);
```

Tabla HorarioSemanal

```
-- Tabla HorarioSemanal
CREATE TABLE HorarioSemanal (
    id BINARY(16) PRIMARY KEY,
    distribucionSemanal VARCHAR(255) NOT NULL,
    horaInicioPorDia VARCHAR(255) NOT NULL,
    FOREIGN KEY (id) REFERENCES Horario(id)
);
```

Tabla HorarioNocturno

```
-- Tabla HorarioNocturno
CREATE TABLE HorarioNocturno (
    id BINARY(16) PRIMARY KEY,
    diasProgramados VARCHAR(255) NOT NULL,
    duracionJornadaHoras INT NOT NULL,
    numeroJornadasMensuales INT NOT NULL,
    FOREIGN KEY (id) REFERENCES Horario(id)
);
```

Tabla HorarioFeriante

```
-- Tabla HorarioFeriante
CREATE TABLE HorarioFeriante (
    id BINARY(16) PRIMARY KEY,
    diasNoLaborables VARCHAR(255) NOT NULL,
    duracionGuardiaHoras INT NOT NULL,
    guardiasProgramadas VARCHAR(255) NOT NULL,
    horasMinimasMensuales INT NOT NULL,
    FOREIGN KEY (id) REFERENCES Horario(id)
);
```

Tabla HorarioDXI

```
-- Tabla HorarioDXI
CREATE TABLE HorarioDXI (
    id BINARY(16) PRIMARY KEY,
    distribucionHoraria VARCHAR(255) NOT NULL,
    horaInicioPorDia VARCHAR(255) NOT NULL,
    horasSemanales INT NOT NULL,
    FOREIGN KEY (id) REFERENCES Horario(id)
);
```

Tabla HorarioGuardiaMedica

```
-- Tabla HorarioGuardiaMedica
CREATE TABLE HorarioGuardiaMedica (
    id BINARY(16) PRIMARY KEY,
    duracionGuardiaHoras INT NOT NULL,
    fechasGuardias VARCHAR(255) NOT NULL,
    numeroGuardiasSemanal INT NOT NULL,
    permitirGuardiasContinuas BOOLEAN NOT NULL,
    tiempoDescansoMinimoHoras INT NOT NULL,
    FOREIGN KEY (id) REFERENCES Horario(id)
);
```

Tabla HorarioGuardiaEnfermeria

```
-- Tabla HorarioGuardiaEnfermeria
CREATE TABLE HorarioGuardiaEnfermeria (
    id BINARY(16) PRIMARY KEY,
    duracionGuardia10Horas INT NOT NULL,
    duracionGuardia12Horas INT NOT NULL,
    fechasGuardias VARCHAR(255) NOT NULL,
    numeroGuardias10Horas INT NOT NULL,
    numeroGuardias12Horas INT NOT NULL,
    FOREIGN KEY (id) REFERENCES Horario(id)
);
```

Tabla HorarioJefeServicioGuardiaPasiva

```
-- Tabla HorarioJefeServicioGuardiaPasiva
CREATE TABLE HorarioJefeServicioGuardiaPasiva (
    id BINARY(16) PRIMARY KEY,
    diasLaborables VARCHAR(255) NOT NULL,
    horasPorDia INT NOT NULL,
    FOREIGN KEY (id) REFERENCES Horario(id)
);
```

Tabla HorarioAbierto

```
-- Tabla HorarioAbierto
CREATE TABLE HorarioAbierto (
    id BINARY(16) PRIMARY KEY,
    flexibilidadHoraria BOOLEAN NOT NULL,
    horasSemanales INT NOT NULL,
    preferenciasHorarias VARCHAR(255),
    FOREIGN KEY (id) REFERENCES Horario(id)
);
```

Tabla Cargo

```
-- Tabla Cargo
CREATE TABLE Cargo (
    id BINARY(16) PRIMARY KEY,
    numero INT NOT NULL,
    descripcion VARCHAR(255) NOT NULL,
    agrupacion ENUM('ADMINISTRATIVO', 'SERVICIO', 'MEDICO', 'ENFERMERIA',
    'TECNICO', 'PLANTA POLITICA') NOT NULL
);
```

Tabla Domicilio

```
-- Tabla Domicilio
CREATE TABLE Domicilio (
    id BINARY(16) PRIMARY KEY,
    calle VARCHAR(255) NOT NULL,
    numeracion INT,
    barrio VARCHAR(255),
    localidad VARCHAR(255),
    ciudad VARCHAR(255),
    provincia VARCHAR(255)
);
```

Tabla Usuario

```
-- Tabla Usuario
/* Representa a la clase Usuario (y es base para sus herederas):
JefaturaDeServicio, OficinaDePersonal y Direccion
* La herencia puede manejarse de varias formas en SQL.
* Usaremos la estrategia de Tabla por Subclase (Class Table Inheritance)
* donde c/subclase tiene su propia tabla que extiende la tabla de la
* superclase mediante una PK que es la FK de la tabla de la superclase.
*/
CREATE TABLE Usuario (
    id BINARY(16) PRIMARY KEY,
    fechaAlta DATE NOT NULL,
    estado BOOLEAN NOT NULL,
    cuil BIGINT UNIQUE NOT NULL,
    apellidos VARCHAR(255) NOT NULL,
    nombres VARCHAR(255) NOT NULL,
    sexo ENUM('FEMENINO', 'MASCULINO', 'OTRO') NOT NULL,
    mail VARCHAR(255) NOT NULL,
    tel BIGINT,
    domicilioID BINARY(16),
    cargoID BINARY(16),
    servicioID BINARY(16),
    tipoUsuario VARCHAR(50) NOT NULL, -- Indica si: 'Direccion', 'Empleado',
'JefaturaDeServicio', 'OficinaDePersonal'
    passwd VARCHAR(255) NOT NULL,
    profile_image BLOB,
    FOREIGN KEY (domicilioID) REFERENCES Domicilio(id),
    FOREIGN KEY (cargoID) REFERENCES Cargo(id),
    FOREIGN KEY (servicioID) REFERENCES Servicio(id)
);
```

Tabla Direccion

```
-- Tabla Direccion
CREATE TABLE Direccion (
    id BINARY(16) PRIMARY KEY,
    FOREIGN KEY (id) REFERENCES Usuario(id)
);
```

Tabla Servicio

```
-- Tabla Servicio
CREATE TABLE Servicio (
    id BINARY(16) PRIMARY KEY,
    nombre VARCHAR(255) NOT NULL,
    agrupacion ENUM('ADMINISTRATIVO', 'SERVICIO', 'MEDICO', 'ENFERMERIA',
'TECNICO', 'PLANTA POLITICA') NOT NULL,
    direccionID BINARY(16) NOT NULL,
    FOREIGN KEY (direccionID) REFERENCES Direccion(id)
);
```

Tabla JefaturaDeServicio

```
-- Tabla JefaturaDeServicio
CREATE TABLE JefaturaDeServicio (
    id BINARY(16) PRIMARY KEY,
    FOREIGN KEY (id) REFERENCES Usuario(id)
);
```

Tabla OficinaDePersonal

```
-- Tabla OficinaDePersonal
CREATE TABLE OficinaDePersonal (
    id BINARY(16) PRIMARY KEY,
    FOREIGN KEY (id) REFERENCES Usuario(id),
    reportesGenerados INT
);
```

Tabla Empleado

```
-- Tabla Empleado
CREATE TABLE Empleado (
    id BINARY(16) PRIMARY KEY,
    FOREIGN KEY (id) REFERENCES Usuario(id),
    francesCompensatoriosUtilizados INT,
    horarioActualID BINARY(16),
    jefaturaID BINARY(16) NOT NULL,
    FOREIGN KEY (horarioActualID) REFERENCES HorarioBase(id),
    FOREIGN KEY (jefaturaID) REFERENCES JefaturaDeServicio(id)
);
```

Debemos verificar:

- Integridad Referencial:** Cada Empleado está asociado a una JefaturaDeServicio y a un Servicio.
- Coherencia:** Dado que la JefaturaDeServicio está asociada al Servicio, podemos validar que el Empleado y la JefaturaDeServicio pertenecen al mismo Servicio.
- Restricciones Adicionales:** Agregamos una restricción para asegurar que el servicioID del Empleado coincida con el servicioID de la JefaturaDeServicio a la que está asignado.

Trigger BEFORE INSERT

```
-- Trigger BEFORE INSERT
-- cambiamos el delimitador por defecto para no tener problemas
DELIMITER $$

CREATE TRIGGER trg_check_servicio_consistente_before_insert
BEFORE INSERT ON Empleado
FOR EACH ROW
BEGIN
    DECLARE empleadoServicioID BINARY(16);
    DECLARE jefaturaServicioID BINARY(16);

    -- Obtener el servicioID del Empleado desde la tabla Usuario
    SELECT servicioID INTO empleadoServicioID
    FROM Usuario
    WHERE id = NEW.id;

    -- Verificar si NEW.jefaturaID no es NULL
    IF NEW.jefaturaID IS NOT NULL THEN
        -- Obtener el servicioID de la JefaturaDeServicio desde la tabla Usuario
        SELECT servicioID INTO jefaturaServicioID
        FROM Usuario
        WHERE id = NEW.jefaturaID;

        -- Si no se encuentra el servicioID de la JefaturaDeServicio, generarError
        IF jefaturaServicioID IS NULL THEN
            SIGNAL SQLSTATE '45000'
```

```
        SET MESSAGE_TEXT = 'La JefaturaDeServicio especificada no tiene un
servicio asignado.';
        END IF;

        -- Verificar si los servicioID coinciden
        IF empleadoServicioID != jefaturaServicioID THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'El servicioID del Empleado debe coincidir con el
servicioID de la JefaturaDeServicio.';
            END IF;
        END IF;
        -- Si NEW.jefaturaID es NULL, no se realiza la validación
END$$
-- restablecemos el delimitador por defecto
DELIMITER ;
```

Trigger BEFORE UPDATE

```
-- Trigger BEFORE UPDATE
-- cambiamos el delimitador por defecto para no tener problemas
DELIMITER $$

CREATE TRIGGER trg_check_servicio_consistente_before_update
BEFORE UPDATE ON Empleado
FOR EACH ROW
BEGIN
    DECLARE empleadoServicioID BINARY(16);
    DECLARE jefaturaServicioID BINARY(16);

    -- Obtener el servicioID del Empleado desde la tabla Usuario
    SELECT servicioID INTO empleadoServicioID
    FROM Usuario
    WHERE id = NEW.id;

    -- Verificar si NEW.jefaturaID no es NULL
    IF NEW.jefaturaID IS NOT NULL THEN
        -- Obtener el servicioID de la JefaturaDeServicio desde la tabla Usuario
        SELECT servicioID INTO jefaturaServicioID
        FROM Usuario
        WHERE id = NEW.jefaturaID;

        -- Si no se encuentra el servicioID de la JefaturaDeServicio, genera Error
        IF jefaturaServicioID IS NULL THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'La JefaturaDeServicio especificada no tiene un
servicio asignado.';
```

```

        END IF;

        -- Verificar si los servicioID coinciden
        IF empleadoServicioID != jefaturaServicioID THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'El servicioID del Empleado debe coincidir con el
servicioID de la JefaturaDeServicio.';
        END IF;
    END IF;
    -- Si NEW.jefaturaID es NULL, no se realiza la validación
END$$
-- restablecemos el delimitador por defecto
DELIMITER ;

```

Tabla Roles

Del modelado de clases de diseño, surge una jerarquía de clases donde:

- **Clase Base:** Usuario (abstracta)
- **Clases Hijas:** Empleado, JefaturaDeServicio, OficinaDePersonal, Direccion

Este diseño funciona bien cuando cada usuario pertenece a un único tipo. Sin embargo, cuando un usuario necesita tener múltiples roles (por ejemplo, la jefa de Personal es de tipo OficinaDePersonal y JefaturaDeServicio al mismo tiempo) la herencia múltiple (que es una característica de la POO) no es una opción en Java, lo que complica el diseño y la mantenibilidad. Una alternativa a esto, es modelar los roles directamente en la **base de datos**, lo que facilita la asignación dinámica de roles a los usuarios. Este enfoque es especialmente útil para el caso del HMU, en donde los roles cambian frecuentemente ya que no requiere modificar el código en Java.

```

-- Tabla Rol
CREATE TABLE Rol (
    id BINARY(16) PRIMARY KEY,
    nombre VARCHAR(50) UNIQUE NOT NULL,
    descripcion VARCHAR(255)
);

```

Tabla Usuario_Rol

Dado que un usuario puede tener muchos roles y que un rol puede pertenecer a muchos usuarios, para evitar esa relación mucho a muchos en el modelo relacional, necesitamos una tabla intermedia de **Usuarios x Roles**:

```

-- Tabla Intermedia Usuario_Rol
CREATE TABLE Usuario_Rol (
    usuario_id BINARY(16) NOT NULL,
    rol_id BINARY(16) NOT NULL,
    PRIMARY KEY (usuario_id, rol_id),
    FOREIGN KEY (usuario_id) REFERENCES Usuario(id) ON DELETE CASCADE,
    FOREIGN KEY (rol_id) REFERENCES Rol(id) ON DELETE CASCADE
) ENGINE=InnoDB;

```

Tabla Autorizacion

```
-- Tabla Autorizacion
CREATE TABLE Autorizacion (
    id BINARY(16) PRIMARY KEY,
    fechaAutorizacion DATETIME NOT NULL,
    tipo ENUM('DIRECCION', 'JEFATURADESERVICIO', 'OFICINADEPERSONAL', 'USUARIO')
NOT NULL,
    autorizadoPorID BINARY(16) NOT NULL,
    FOREIGN KEY (autorizadoPorID) REFERENCES Usuario(id)
);
```

Tabla EstadoTramite

```
-- Tabla EstadoTramite
CREATE TABLE EstadoTramite (
    id BINARY(16) PRIMARY KEY,
    nombre VARCHAR(50) NOT NULL
);
```

Inicializamos en tiempo de creación, la tabla EstadoTramite, con (prácticamente todos) los posibles valores que tendrá EstadoTramite.

Inicialización de Tabla EstadoTramite

```
-- Inicializar tabla EstadoTramite
INSERT INTO EstadoTramite (id, nombre) VALUES
    (UUID_TO_BIN(UUID()), 'BORRADOR'),
    (UUID_TO_BIN(UUID()), 'ENVIADO'),
    (UUID_TO_BIN(UUID()), 'PENDIENTE DE FIRMA'),
    (UUID_TO_BIN(UUID()), 'PENDIENTE DE AUTORIZACION'),
    (UUID_TO_BIN(UUID()), 'AUTORIZADO'),
    (UUID_TO_BIN(UUID()), 'RECHAZADO'),
    (UUID_TO_BIN(UUID()), 'OBSERVADO'),
    (UUID_TO_BIN(UUID()), 'LEIDO'),
    (UUID_TO_BIN(UUID()), 'COMPLETADO');
```

Tabla Novedad

```
-- Tabla Novedad
CREATE TABLE Novedad (
    id BINARY(16) PRIMARY KEY,
    cod INT NOT NULL,
    descripcion VARCHAR(255) NOT NULL,
    estado VARCHAR(50) NOT NULL,
    estadoFechaModif TIMESTAMP,
```

```

fechaInicio DATE,
fechaFin DATE,
fechaSolicitud DATE,
reqAprobDireccion BOOLEAN NOT NULL,
estadoTramiteID BINARY(16),
FOREIGN KEY (estadoTramiteID) REFERENCES EstadoTramite(id)
);

```

Tabla Intermedia Empleado_Novedad

Cuando un empleado no tiene novedades, se le asigna la novedad con código 99: Presente. Es común entonces, tener una relación mucho-a-muchos que salvaguardar, ya que por es una situación normal que, durante un período muchos empleados no tengan novedades (relación mucho a muchos).

```

-- Tabla Intermedia Empleado_Novedad (Por la novedad 99, tenemos una relación
-- mucho-a-muchos que salvaguardar)
CREATE TABLE Empleado_Novedad (
    empleadoID BINARY(16) NOT NULL,
    novedadID BINARY(16) NOT NULL,
    PRIMARY KEY (empleadoID, novedadID),
    FOREIGN KEY (empleadoID) REFERENCES Empleado(id),
    FOREIGN KEY (novedadID) REFERENCES Novedad(id)
);

```

Tabla JornadaLaboral

```

-- Tabla JornadaLaboral
CREATE TABLE JornadaLaboral (
    id BINARY(16) PRIMARY KEY,
    fechaIngreso DATETIME NOT NULL,
    fechaEgreso DATETIME NOT NULL
);

```

Tabla DiagramaDeServicio

```

-- Tabla DiagramaDeServicio
CREATE TABLE DiagramaDeServicio (
    id BINARY(16) PRIMARY KEY,
    estado VARCHAR(50) NOT NULL,
    fechaInicio DATE NOT NULL,
    fechaFin DATE NOT NULL,
    servicioID BINARY(16) NOT NULL,
    FOREIGN KEY (servicioID) REFERENCES Servicio(id)
);

```

Tabla Planificacion

Esta tabla nos relaciona DiagramaDeServicio, Empleado y JornadaLaboral.
Representa las planificaciones de cada empleado en un diagrama de servicio.

```
-- Tabla Planificacion: para el atributo 'planificaciones' de la clase
DiagramaDeServicio
CREATE TABLE Planificacion (
    diagramaID BINARY(16) NOT NULL,
    empleadoID BINARY(16) NOT NULL,
    jornadaID BINARY(16) NOT NULL,
    PRIMARY KEY (diagramaID, empleadoID, jornadaID),
    FOREIGN KEY (diagramaID) REFERENCES DiagramaDeServicio(id),
    FOREIGN KEY (empleadoID) REFERENCES Empleado(id),
    FOREIGN KEY (jornadaID) REFERENCES JornadaLaboral(id)
);
```

Tabla Memorandum

```
CREATE TABLE Memorandum (
    id BINARY(16) PRIMARY KEY,
    asunto VARCHAR(255) NOT NULL,
    contenido TEXT NOT NULL,
    fechaEnvio DATETIME,
    fechaRecepcion DATETIME,
    estado BINARY(16) NOT NULL,
    remitenteID BINARY(16) NOT NULL,
    FOREIGN KEY (estado) REFERENCES EstadoTramite(id),
    FOREIGN KEY (remitenteID) REFERENCES Usuario(id)
);
```

Tabla Memorandum_Destinatario

Salvaguarda la relación muchos-a-muchos entre 'Memorandum' y 'Usuario' para los destinatarios.

```
CREATE TABLE Memorandum_Destinatario (
    memorandumID BINARY(16) NOT NULL,
    usuarioID BINARY(16) NOT NULL,
    fechaRecepcion DATETIME,
    PRIMARY KEY (memorandumID, usuarioID),
    FOREIGN KEY (memorandumID) REFERENCES Memorandum(id),
    FOREIGN KEY (usuarioID) REFERENCES Usuario(id)
);
```

Tabla Memorandum_Firmante

Salvaguarda la relación muchos-a-muchos entre 'Memorandum' y 'Usuario' para los firmantes,
(incluyendo la fecha y hora de la firma)

```
CREATE TABLE Memorandum_Firmante (
```

```

memorandumID BINARY(16) NOT NULL,
usuarioID BINARY(16) NOT NULL,
fechaFirma DATETIME NOT NULL,
PRIMARY KEY (memorandumID, usuarioID),
FOREIGN KEY (memorandumID) REFERENCES Memorandum(id),
FOREIGN KEY (usuarioID) REFERENCES Usuario(id)
);

```

Tabla Memorandum_Autorizacion

Registra las autorizaciones necesarias y las obtenidas para cada memorándum.

```

-- Tabla Memorandum_Autorizacion
CREATE TABLE Memorandum_Autorizacion (
    id BINARY(16) PRIMARY KEY,
    memorandumID BINARY(16) NOT NULL,
    tipoAutorizacionID ENUM('JefeDeServicio', 'OficinaDePersonal', 'Direccion',
'Usuario') NOT NULL,
    autorizadoPorID BINARY(16),
    fechaAutorizacion DATETIME,
    estado ENUM('PENDIENTE', 'AUTORIZADO', 'RECHAZADO') NOT NULL,
    FOREIGN KEY (memorandumID) REFERENCES Memorandum(id),
    FOREIGN KEY (autorizadoPorID) REFERENCES Usuario(id)
);

```

Tabla FrancoCompensatorio

```

-- Tabla FrancoCompensatorio
CREATE TABLE FrancoCompensatorio (
    id BINARY(16) PRIMARY KEY,
    cantHoras DOUBLE NOT NULL,
    descripcion VARCHAR(255) NOT NULL,
    fechaAutorizacion DATETIME,
    fechaDeAplicacion DATE,
    estadoTramiteID BINARY(16) NOT NULL,
    autorizadaPorID BINARY(16),
    empleadoID BINARY(16) NOT NULL,
    jefaturaID BINARY(16),
    FOREIGN KEY (estadoTramiteID) REFERENCES EstadoTramite(id),
    FOREIGN KEY (autorizadaPorID) REFERENCES Usuario(id),
    FOREIGN KEY (empleadoID) REFERENCES Empleado(id),
    FOREIGN KEY (jefaturaID) REFERENCES JefaturaDeServicio(id)
);

```

Tabla HoraExtra

```
-- Tabla HoraExtra
```

```
CREATE TABLE HoraExtra (
    id BINARY(16) PRIMARY KEY,
    descripcion VARCHAR(255) NOT NULL,
    fechaIngreso DATETIME NOT NULL,
    fechaEgreso DATETIME NOT NULL,
    ponderacion INT NOT NULL,
    fechaAutorizacion DATETIME,
    estadoTramiteID BINARY(16) NOT NULL,
    autorizadaPorID BINARY(16),
    empleadoID BINARY(16) NOT NULL,
    jefaturaID BINARY(16),
    FOREIGN KEY (estadoTramiteID) REFERENCES EstadoTramite(id),
    FOREIGN KEY (autorizadaPorID) REFERENCES Usuario(id),
    FOREIGN KEY (empleadoID) REFERENCES Empleado(id),
    FOREIGN KEY (jefaturaID) REFERENCES JefaturaDeServicio(id)
);
```

Tabla ParteDiario

```
-- Tabla ParteDiario
CREATE TABLE ParteDiario (
    id BINARY(16) PRIMARY KEY,
    fechaDeCierre DATETIME,
    periodo VARCHAR(50) NOT NULL,
    modificadoPorID BINARY(16),
    oficinaID BINARY(16) NOT NULL,
    FOREIGN KEY (modificadoPorID) REFERENCES Usuario(id),
    FOREIGN KEY (oficinaID) REFERENCES OficinaDePersonal(id)
);
```

Tabla Intermedia ParteDiario_Empleado

Salvaguarda la relación muchos-a-muchos entre ‘ParteDiario’ y ‘Empleado’.

```
-- Tabla Intermedia ParteDiario_Empleado (relaciona ParteDiario con Empleado)
CREATE TABLE ParteDiario_Empleado (
    parteDiarioID BINARY(16) NOT NULL,
    empleadoID BINARY(16) NOT NULL,
    PRIMARY KEY (parteDiarioID, empleadoID),
    FOREIGN KEY (parteDiarioID) REFERENCES ParteDiario(id),
    FOREIGN KEY (empleadoID) REFERENCES Empleado(id)
);
```

Tabla MarcacionEmpleado

```
-- Tabla MarcacionEmpleado
CREATE TABLE MarcacionEmpleado (
```

```

    id BINARY(16) PRIMARY KEY,
    fechaMarcacion DATETIME NOT NULL,
    observaciones VARCHAR(255),
    tipoMarcacion ENUM('INGRESO', 'EGRESO') NOT NULL,
    validada BOOLEAN NOT NULL,
    empleadoID BINARY(16) NOT NULL,
    FOREIGN KEY (empleadoID) REFERENCES Empleado(id)
);

```

Tabla RegistroJornadaLaboral**Notas:**

- **fecha:** Representa la fecha de la jornada laboral.
- **empleadoID:** Referencia al empleado al que pertenece la jornada.
- **marcacionIngresoID** y **marcacionEgresoID:** *FK* a las marcaciones específicas de ingreso y egreso.

Restricciones adicionales:

- Las marcaciones referenciadas DEBEN pertenecer al mismo empleado.
- La marcación de ingreso debe ser de tipo INGRESO.
- La marcación de egreso debe ser de tipo EGRESO.

```

-- Tabla RegistroJornadaLaboral
CREATE TABLE RegistroJornadaLaboral (
    id BINARY(16) PRIMARY KEY,
    fecha DATE NOT NULL,
    empleadoID BINARY(16) NOT NULL,
    marcacionIngresoID BINARY(16) NOT NULL,
    marcacionEgresoID BINARY(16) NOT NULL,
    duracionJornada INT, -- No es bueno guardar datos calculados, pero me puede
facilitar la vida al momento de programar la lógica de negocios
    FOREIGN KEY (empleadoID) REFERENCES Empleado(id),
    FOREIGN KEY (marcacionIngresoID) REFERENCES MarcacionEmpleado(id),
    FOREIGN KEY (marcacionEgresoID) REFERENCES MarcacionEmpleado(id)
);

```

Implementación de índices en Claves Foráneas (FK)

Es una buena práctica agregar índices en las columnas que son claves foráneas y se utilizan en consultas frecuentes.

```

-- Índices en Claves Foráneas:
CREATE INDEX idx_empleado_jefatura ON Empleado (jefaturaID);
CREATE INDEX idx_servicio_direccion ON Servicio (direccionID);

```

17. Inserción, consulta y borrado de registros

El script SQL con las sentencias expuesta en este documento (y otras más) se encuentra en GitHub en:

<https://github.com/linkstat/seminare/tree/main/sql>

Como ya se mencionó anteriormente, manejaremos los UUIDs haciendo uso de las funciones UUID_TO_BIN y BIN_TO_UUID para las inserciones y las consultas SQL.

Primero, seleccionamos la base de datos sobre la cual trabajaremos las sentencias SQL.

```
USE aromito;
```

Insertar Domicilios

```
-- Insertar domicilios para poder asignárselos a usuarios
INSERT INTO Domicilio (id, calle, numeracion, barrio, ciudad, localidad, provincia)
VALUES
    (UUID_TO_BIN(UUID()), 'Defensa', 1200, NULL, 'Córdoba', NULL, 'Córdoba'),
    (UUID_TO_BIN(UUID()), 'Lavalleja', 3050, NULL, 'Córdoba', NULL, 'Córdoba'),
    (UUID_TO_BIN(UUID()), 'Av Belgrano Sur', 134, NULL, 'La Rioja', NULL, 'La Rioja'),
    (UUID_TO_BIN(UUID()), 'Av. Marcelo T. de Alvear', 120, NULL, 'Córdoba', NULL, 'Córdoba'),
    (UUID_TO_BIN(UUID()), 'Av. Libertador', 1450, NULL, 'Alta Gracia', NULL, 'Córdoba'),
    (UUID_TO_BIN(UUID()), 'Catamarca', 441, NULL, 'Córdoba', NULL, 'Córdoba'),
    (UUID_TO_BIN(UUID()), 'Junín', FLOOR(RAND() * 1000 + 1), NULL, 'Rosario', NULL, 'Santa Fe'),
    (UUID_TO_BIN(UUID()), 'Mate de Luna', FLOOR(RAND() * 1000 + 1), NULL, 'Famaillá', NULL, 'Tucumán'),
    (UUID_TO_BIN(UUID()), 'Gobernación', FLOOR(RAND() * 1000 + 1), NULL, 'Santa Fe', NULL, 'Santa Fe'),
    (UUID_TO_BIN(UUID()), 'Güemes', FLOOR(RAND() * 1000 + 1), NULL, 'Iruya', NULL, 'Jujuy'),
    (UUID_TO_BIN(UUID()), '9 de julio', FLOOR(RAND() * 1000 + 1), NULL, 'Corrientes', NULL, 'Corrientes'),
    (UUID_TO_BIN(UUID()), 'Av. del Trabajo', FLOOR(RAND() * 1000 + 1), NULL, 'Rosario', NULL, 'Santa Fe'),
    (UUID_TO_BIN(UUID()), 'Calle del Río', FLOOR(RAND() * 1000 + 1), NULL, 'Paraná', NULL, 'Entre Ríos'),
    (UUID_TO_BIN(UUID()), 'Av. Central', FLOOR(RAND() * 1000 + 1), NULL, 'Salta', NULL, 'Salta'),
    (UUID_TO_BIN(UUID()), 'Av. Siempre Viva', FLOOR(RAND() * 1000 + 1), NULL, 'Rawson', NULL, 'Chubut'),
    (UUID_TO_BIN(UUID()), 'San Martín', FLOOR(RAND() * 1000 + 1), NULL, 'Bariloche', NULL, 'Río Negro');
```

Uso de variables

```
-- Obtener el ID del domicilio del HMU (Catamarca 441)
SELECT id INTO @HMUdomID
FROM Domicilio
WHERE calle = 'Catamarca' AND numeracion = 441 AND ciudad = 'Córdoba' AND provincia = 'Córdoba'
LIMIT 1;
-- Opcional: verificar que se obtuvo el ID correctamente
SELECT BIN_TO_UUID(@HMUdomID) AS domicilioHMU;
```

Sistema de Gestión de Ausentismo Hospitalario

```
-- Obtener el ID del domicilio del HIM (Lavalleja 3050)
SELECT id INTO @HIMdomID
FROM Domicilio
WHERE calle = 'Lavalleja' AND numeracion = 3050 AND ciudad = 'Córdoba' AND provincia = 'Córdoba'
LIMIT 1;
-- Opcional: verificar que se obtuvo el ID correctamente
SELECT BIN_TO_UUID(@HIMdomID) AS domicilioHIM;
-- Generar UUIDs y almacenarlos en variables, para luego usar con JefaturaDeServicios
SET @JefeAdmisionID = UUID_TO_BIN(UUID());
SET @JefeBiomedicaID = UUID_TO_BIN(UUID());
SET @JefeCardiologiaID = UUID_TO_BIN(UUID());
SET @JefeCirugiaID = UUID_TO_BIN(UUID());
SET @JefeDiagPorImagenesID = UUID_TO_BIN(UUID());
SET @JefeEnfermeriaID = UUID_TO_BIN(UUID());
SET @JefeEsterilizacionID = UUID_TO_BIN(UUID());
SET @JefeFarmaciaID = UUID_TO_BIN(UUID());
SET @JefeHabilitacionID = UUID_TO_BIN(UUID());
SET @JefeInformaticaID = UUID_TO_BIN(UUID());
SET @JefeNutricionID = UUID_TO_BIN(UUID());
SET @JefeTraumatoID = UUID_TO_BIN(UUID());
SET @JefeToxicologiaID = UUID_TO_BIN(UUID());
-- Generar UUIDs y almacenarlos en variables, para luego usar con Servicios
SET @AdministracionID = UUID_TO_BIN(UUID());
SET @AdmisionID = UUID_TO_BIN(UUID());
SET @AnatomiaPatologicaID = UUID_TO_BIN(UUID());
SET @AnestesiaID = UUID_TO_BIN(UUID());
SET @AuditoriaMedicaID = UUID_TO_BIN(UUID());
SET @BiomedicaID = UUID_TO_BIN(UUID());
SET @BioquimicaID = UUID_TO_BIN(UUID());
SET @CamillerosID = UUID_TO_BIN(UUID());
```

Sistema de Gestión de Ausentismo Hospitalario

```
SET @CapacitacionDocenciaID = UUID_TO_BIN(UUID());
SET @CirugiaID = UUID_TO_BIN(UUID());
SET @CirugiaPlasticaID = UUID_TO_BIN(UUID());
SET @ClinicaMedicaID = UUID_TO_BIN(UUID());
SET @DiagPorImagenesID = UUID_TO_BIN(UUID());
SET @DireccionID = UUID_TO_BIN(UUID());
SET @EnfermeriaID = UUID_TO_BIN(UUID());
SET @EsterilizacionID = UUID_TO_BIN(UUID());
SET @FacturacionID = UUID_TO_BIN(UUID());
SET @FarmaciaID = UUID_TO_BIN(UUID());
SET @HabilitacionID = UUID_TO_BIN(UUID());
SET @HemoterapiaID = UUID_TO_BIN(UUID());
SET @InformaticaID = UUID_TO_BIN(UUID());
SET @InstrumentacionID = UUID_TO_BIN(UUID());
SET @KinesiologiaID = UUID_TO_BIN(UUID());
SET @LaboratorioID = UUID_TO_BIN(UUID());
SET @LavaderoID = UUID_TO_BIN(UUID());
SET @MantenimientoID = UUID_TO_BIN(UUID());
SET @MedicinaLegalID = UUID_TO_BIN(UUID());
SET @NeurocirugiaID = UUID_TO_BIN(UUID());
SET @NutricionID = UUID_TO_BIN(UUID());
SET @PersonalID = UUID_TO_BIN(UUID());
SET @QuirofanoID = UUID_TO_BIN(UUID());
SET @RegMedicoID = UUID_TO_BIN(UUID());
SET @SaludMentalID = UUID_TO_BIN(UUID());
SET @SecTecnicaID = UUID_TO_BIN(UUID());
SET @ServicioSocialID = UUID_TO_BIN(UUID());
SET @TecnoelectricaID = UUID_TO_BIN(UUID());
SET @TraumatologiaID = UUID_TO_BIN(UUID());
```

Insertar usuarios

Dado que aún no generamos los servicios (solo tenemos los UUID que utilizaremos), deshabilitaremos temporalmente las restricciones de clave primaria para poder generar los usuarios.

```
-- Deshabilitar temporalmente restricciones de clave primaria
SET FOREIGN_KEY_CHECKS = 0;

-- Insertar en Usuario
-- Insertar en Usuario
INSERT INTO Usuario (id, nombres, apellidos, mail, cuil, fechaAlta, sexo, estado, domicilioID, servicioID,
tipoUsuario, passwd)
VALUES
    -- Dirección
    (UUID_TO_BIN(UUID()), 'Mariano Gustavo', 'Marino', 'direccion@hmu.com.ar', 20224448885, CURDATE(),
'MASCULINO', TRUE, @HMuDomID, @DireccionID, 'Direccion', @initPass),
    (UUID_TO_BIN(UUID()), 'Federico', 'Huergo Sánchez', 'fedesubdir@hmu.com.ar', 20259993331, CURDATE(),
'MASCULINO', TRUE, @HIMdomID, @DireccionID, 'Direccion', @initPass),
    (UUID_TO_BIN(UUID()), 'Gloria', 'Longoni', 'subdirectora@hmu.com.ar', 27129997773, CURDATE(), 'FEMENINO',
TRUE, (SELECT id FROM Domicilio WHERE ciudad = 'Alta Gracia' LIMIT 1), @DireccionID, 'Direccion', @initPass),
    (UUID_TO_BIN(UUID()), 'Fabricio', 'Vitali', 'subdireccion@hmu.com.ar', 20239997772, CURDATE(),
'MASCULINO', TRUE, @HMUdomID, @DireccionID, 'Direccion', @initPass),

    -- Jefaturas de Servicio
    (@JefeAdmisionID, 'Andrea', 'Balconte', 'andrea@hmu.com.ar', 27295554447, CURDATE(), 'FEMENINO', TRUE,
(SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @AdmisionID, 'JefaturaDeServicio', @initPass),
    (@JefeBiomedicaID, 'Silvina', 'Maestro', 'smaestro@hmu.com.ar', 27274422442, CURDATE(), 'FEMENINO', TRUE,
(SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @BiomedicaID, 'JefaturaDeServicio', @initPass),
    (@JefeCirugiaID, 'Maximiliano', 'Titarelli', 'drtita@hmu.com.ar', 20268944448, CURDATE(), 'MASCULINO',
TRUE, (SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @CirugiaID, 'JefaturaDeServicio', @initPass),
    (@JefeDiagPorImagenesID, 'Juan Ignacio', 'Morales', 'jimorales@hmu.com.ar', 20281324547, CURDATE(),
'MASCULINO', TRUE, (SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @DiagPorImagenesID,
'JefaturaDeServicio', @initPass),
```

```
(@JefeEnfermeriaID, 'Tania', 'Plaza', 'tplaza@hmu.com.ar', 27224444445, CURDATE(), 'FEMENINO', TRUE,
(SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @EnfermeriaID, 'JefaturaDeServicio', @initPass),
(@JefeEsterilizacionID, 'Matías', 'Pérez Cabral', 'mcp@hmu.com.ar', 20289445441, CURDATE(), 'MASCULINO',
TRUE, (SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @EsterilizacionID, 'JefaturaDeServicio', @initPass),
(@JefeFarmaciaID, 'María Pía', 'Arancibia', 'piaarancibia@hmu.com.ar', 27254344447, CURDATE(), 'FEMENINO',
TRUE, (SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @FarmaciaID, 'JefaturaDeServicio', @initPass),
(@JefeHabilitacionID, 'Carlos Fernando', 'Roberts', 'ferroberts@hmu.com.ar', 20276444446, CURDATE(),
'MASCULINO', TRUE, (SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @HabilitacionID, 'JefaturaDeServicio',
@initPass),
(@JefeInformaticaID, 'Juan Manuel', 'Roqué', 'jmroque@hmu.com.ar', 20220361118, CURDATE(), 'OTRO', TRUE,
(SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @InformaticaID, 'JefaturaDeServicio', @initPass),
(@JefeNutricionID, 'Alejandra', 'Boqué', 'aleboque@hmu.com.ar', 27174444446, CURDATE(), 'FEMENINO', TRUE,
(SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @NutricionID, 'JefaturaDeServicio', @initPass),
(@JefeTraumatologiaID, 'Omar Wenceslao', 'Sánchez', 'owsanchez@hmu.com.ar', 20124644445, CURDATE(),
'MASCULINO', TRUE, (SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @TraumatologiaID, 'JefaturaDeServicio',
@initPass),
(@JefeToxicologiaID, 'Andrea', 'Vilkelis', 'avilkelis@hmu.com.ar', 27214434447, CURDATE(), 'FEMENINO',
TRUE, (SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @ToxicologiaID, 'JefaturaDeServicio', @initPass),

-- Oficina de Personal
(@JefePersonalID, 'Florencia', 'Maurino', 'flormaurino@hmu.com.ar', 27284644443, CURDATE(), 'FEMENINO',
TRUE, (SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @PersonalID, 'OficinaDePersonal', @userAPass),

-- Empleados
(UUID_TO_BIN(UUID()), 'Juan', 'Aniceto', 'janiceto@hmu.com.ar', 20124543421, CURDATE(), 'MASCULINO', TRUE,
(SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @DiagPorImagenesID, 'Empleado', @initPass),
(UUID_TO_BIN(UUID()), 'Baltazar', 'Garzón', 'baltig@hmu.com.ar', 20554644448, CURDATE(), 'MASCULINO',
TRUE, (SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @PersonalID, 'Empleado', @initPass),
(UUID_TO_BIN(UUID()), 'Pedro', 'Taborda', 'pltaborda@hmu.com.ar', 20154786445, CURDATE(), 'MASCULINO',
TRUE, (SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @PersonalID, 'Empleado', @initPass),
```

```
(UUID_TO_BIN(UUID()), 'María Laura', 'Vargas Ruiz', 'mlvargas@hmu.com.ar', 27554664563, CURDATE(),
'FEMENINO', TRUE, (SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @EsterilizacionID, 'Empleado',
@initPass),
(UUID_TO_BIN(UUID()), 'María Celeste', 'Vignetta', 'mcv@hmu.com.ar', 27554685443, CURDATE(), 'FEMENINO',
TRUE, (SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @CirugiaID, 'Empleado', @initPass),
(UUID_TO_BIN(UUID()), 'Alicia', 'Vivas', 'avivas@hmu.com.ar', 27554567443, CURDATE(), 'FEMENINO', TRUE,
(SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @HabilitacionID, 'Empleado', @initPass),
(UUID_TO_BIN(UUID()), 'Matías Quique', 'Canga Castellanos', 'lestat@hmu.com.ar', 20554649743, CURDATE(),
'MASCULINO', TRUE, (SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @InformaticaID, 'Empleado', @initPass),
(UUID_TO_BIN(UUID()), 'Mauricio Elio', 'Garay', 'mgaray@hmu.com.ar', 24574741745, CURDATE(), 'OTRO', TRUE,
(SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @InformaticaID, 'Empleado', @initPass),
(UUID_TO_BIN(UUID()), 'Eva Patricia', 'Usandivares', 'epu@hmu.com.ar', 27556786453, CURDATE(), 'FEMENINO',
TRUE, (SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @PersonalID, 'Empleado', @initPass),
(UUID_TO_BIN(UUID()), 'Edith', 'Tolay', 'etolay@hmu.com.ar', 20454676443, CURDATE(), 'FEMENINO', TRUE,
(SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @EnfermeriaID, 'Empleado', @initPass),
(UUID_TO_BIN(UUID()), 'José Luis', 'Terrieris', 'jterrieris@hmu.com.ar', 20554543453, CURDATE(),
'MASCULINO', TRUE, (SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @EnfermeriaID, 'Empleado', @initPass),
(UUID_TO_BIN(UUID()), 'Claudia', 'Tarifa', 'clautarifa@hmu.com.ar', 27540678443, CURDATE(), 'FEMENINO',
TRUE, (SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @EnfermeriaID, 'Empleado', @initPass),
(UUID_TO_BIN(UUID()), 'Erica', 'Tapia', 'erikatapia@hmu.com.ar', 27554560863, CURDATE(), 'FEMENINO', TRUE,
(SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @ToxicologiaID, 'Empleado', @initPass),
(UUID_TO_BIN(UUID()), 'Demetrio', 'Tampares', 'demetam@hmu.com.ar', 24784044443, CURDATE(), 'OTRO', TRUE,
(SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @ToxicologiaID, 'Empleado', @initPass),
(UUID_TO_BIN(UUID()), 'Alejandro', 'Suizer', 'amsuizer@hmu.com.ar', 20526456443, CURDATE(), 'MASCULINO',
TRUE, (SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @CirugiaID, 'Empleado', @initPass),
(UUID_TO_BIN(UUID()), 'David', 'Suárez', 'dsuarez@hmu.com.ar', 20554897343, CURDATE(), 'MASCULINO', TRUE,
(SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @BiomedicaID, 'Empleado', @initPass),
(UUID_TO_BIN(UUID()), 'Soledad', 'Romero', 'soleromero@hmu.com.ar', 27864567893, CURDATE(), 'FEMENINO',
TRUE, (SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @AdmisionID, 'Empleado', @initPass),
```

```
(UUID_TO_BIN(UUID()), 'Pedro', 'Rius', 'pedrorius@hmu.com.ar', 20554674843, CURDATE(), 'MASCULINO', TRUE,
(SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @DiagPorImagenesID, 'Empleado', @initPass),
(UUID_TO_BIN(UUID()), 'Damian', 'Rabbat', 'drabbat@hmu.com.ar', 20558244443, CURDATE(), 'MASCULINO', TRUE,
(SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @TraumatologiaID, 'Empleado', @initPass),
(UUID_TO_BIN(UUID()), 'Ismael', 'Puig', 'ipuig@hmu.com.ar', 20567974443, CURDATE(), 'MASCULINO', TRUE,
(SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @TraumatologiaID, 'Empleado', @initPass),
(UUID_TO_BIN(UUID()), 'Diego', 'Pascolo', 'dpascolo@hmu.com.ar', 20456645843, CURDATE(), 'MASCULINO',
TRUE, (SELECT id FROM Domicilio ORDER BY RAND() LIMIT 1), @CirugiaID, 'Empleado', @initPass),
(UUID_TO_BIN(UUID()), 'Sebastián', 'Bustos', 'sebustos@hmu.com.ar', 24554978443, CURDATE(), 'OTRO', TRUE,
(SELECT id FROM Domicilio WHERE calle = 'Av. Siempre Viva' LIMIT 1), @BiomedicaID, 'Empleado', @userBPass);

-- Habilitar nuevamente restricciones de claves primarias
SET FOREIGN_KEY_CHECKS = 1;
```

Insertar los usuarios en las respectivas tablas de especialización de usuarios

```
-- Insertar en la tabla 'Direccion' utilizando los 'id' guardados previamente para esos usuarios
INSERT INTO Direccion (id)
SELECT id FROM Usuario WHERE tipoUsuario = 'Direccion';

-- Insertar en Oficina de Personal
INSERT INTO OficinaDePersonal (id, reportesGenerados)
VALUES ( (SELECT id FROM Usuario WHERE tipoUsuario = 'OficinaDePersonal' LIMIT 1), 0 );
```

Insertar Servicios

```
-- Obtener el ID del actual director (cuyo apellido es 'Marino')
SELECT D.id INTO @dirID
FROM Direccion D
JOIN Usuario U ON D.id = U.id
WHERE U.apellidos = 'Marino' AND U.tipoUsuario = 'Direccion'
```

Sistema de Gestión de Ausentismo Hospitalario

```
LIMIT 1;
-- Opcionalmente, verificar que se obtuvo el ID correctamente
SELECT BIN_TO_UUID(@dirID) AS dirID;

-- Obtener el ID del actual subdirector (cuyo apellido es 'Vitali')
SELECT D.id INTO @subdirID
FROM Direccion D
JOIN Usuario U ON D.id = U.id
WHERE U.apellidos = 'Vitali' AND U.tipoUsuario = 'Direccion'
LIMIT 1;
-- Opcionalmente, verificar que se obtuvo el ID correctamente
SELECT BIN_TO_UUID(@subdirID) AS subdirID;
-- Insertar Servicios
INSERT INTO Servicio (id, nombre, agrupacion, direccionID)
VALUES
    (@AdministracionID, 'Administración', 'ADMINISTRATIVO', @dirID),
    (@AdmisionID, 'Admisión', 'ADMINISTRATIVO', @dirID),
    (@AnatomiaPatologicaID, 'Anatomía Patológica', 'MEDICO', @dirID),
    (@AnestesiaID, 'Anestesia', 'MEDICO', @dirID),
    (@AuditoriaMedicaID, 'Auditoría Médica', 'MEDICO', @dirID),
    (@BiomedicaID, 'Biomédica', 'TECNICO', @subdirID),
    (@BioquimicaID, 'Bioquímica', 'SERVICIO', @subdirID),
    (@CamillerosID, 'Camilleros', 'SERVICIO', @subdirID),
    (@CapacitacionDocenciaID, 'Capacitación y Docencia', 'MEDICO', @dirID),
    (@CentralEsterilizacionID, 'Central de Esterilización', 'SERVICIO', @subdirID),
    (@CirugiaID, 'Cirugía', 'MEDICO', @dirID),
    (@CirugiaPlasticaID, 'Cirugía Plástica', 'MEDICO', @dirID),
    (@ClinicaMedicaID, 'Clínica Médica', 'MEDICO', @dirID),
    (@DiagPorImagenesID, 'Diagnóstico por Imágenes', 'MEDICO', @dirID),
    (@DireccionID, 'Dirección', 'ADMINISTRATIVO', @dirID),
```

```
(@EnfermeriaID, 'Enfermería', 'ENFERMERIA', @dirID),
(@FacturacionID, 'Facturación', 'ADMINISTRATIVO', @dirID),
(@FarmaciaID, 'Farmacia', 'SERVICIO', @subdirID),
(@HabilitacionID, 'Habilitación', 'ADMINISTRATIVO', @dirID),
(@HemoterapiaID, 'Hemoterapia', 'SERVICIO', @subdirID),
(@InformaticaID, 'Informática', 'TECNICO', @subdirID),
(@InstrumentacionID, 'Instrumentación Quirúrgica', 'MEDICO', @dirID),
(@KinesiologiaID, 'Kinesiología', 'MEDICO', @dirID),
(@LaboratorioID, 'Laboratorio', 'SERVICIO', @subdirID),
(@LavaderoID, 'Lavadero', 'SERVICIO', @subdirID),
(@MantenimientoID, 'Mantenimiento', 'TECNICO', @subdirID),
(@MedicinaLegalID, 'Medicina Legal', 'ADMINISTRATIVO', @dirID),
(@NeurocirugiaID, 'Neurocirugía', 'MEDICO', @dirID),
(@NutricionID, 'Nutrición', 'MEDICO', @dirID),
(@PersonalID, 'Personal', 'ADMINISTRATIVO', @dirID),
(@QuirofanoID, 'Quirófano', 'MEDICO', @dirID),
(@RegMedicoID, 'Registro Médico (Archivo)', 'ADMINISTRATIVO', @dirID),
(@SaludMentalID, 'Salud Mental', 'MEDICO', @dirID),
(@SecTecnicaID, 'Secretaría Técnica', 'ADMINISTRATIVO', @dirID),
(@ServicioSocialID, 'Servicio Social', 'ADMINISTRATIVO', @dirID),
(@TecnoelectricaID, 'Tecnoeléctrica', 'TECNICO', @subdirID),
(@TraumatologiaID, 'Traumatología y Ortopedia', 'MEDICO', @dirID);
```

Asignar Jefaturas de Servicio

```
-- Insertar en JefaturaDeServicio, todos los usuarios de tipo JefeDeServicio
INSERT INTO JefaturaDeServicio (id)
SELECT id FROM Usuario WHERE tipoUsuario = 'JefaturaDeServicio';
```

Asignar Oficina De Personal

```
-- Insertar en OficinaDePersonal
INSERT INTO OficinaDePersonal (id, reportesGenerados)
SELECT id, 0 FROM Usuario WHERE tipoUsuario = 'OficinaDePersonal';
```

Asignar Empleados**Desasignar Empleados**

```
-- Desasignar todos los Empleados de sus servicios
UPDATE Empleado
SET servicioID = NULL;
```

Insertar usuarios en Empleado

Al insertar en Empleado, solo necesitamos especificar `id`, `francosCompensatoriosUtilizados`, `horarioActualID`, y `jefaturaID`, asegurándonos de que `jefaturaID` sea coherente con el `servicioID` previamente asignado al Empleado.

```
-- Insertar Empleados en la tabla Empleado
/* Al insertar en Empleado, solo necesitamos especificar
 * id, francosCompensatoriosUtilizados, horarioActualID, y jefaturaID,
 * asegurándonos de que jefaturaID sea coherente con el servicioID asignado al Empleado.
 */
INSERT INTO Empleado (id, francosCompensatoriosUtilizados, horarioActualID, jefaturaID)
SELECT u.id, 0, NULL, jds.id
FROM Usuario u
JOIN Usuario ujds ON ujds.servicioID = u.servicioID AND ujds.tipoUsuario = 'JefaturaDeServicio'
JOIN JefaturaDeServicio jds ON jds.id = ujds.id
WHERE u.tipoUsuario = 'Empleado';
```

Listar usuarios Empleados

```
SELECT
    BIN_TO_UUID(U.id) AS empleado_id,
    U.apellidos,
    U.nombres,
    U.cuil,
    U.mail,
    U.tel,
    S.nombre AS area_de_servicio
FROM Usuario U
JOIN Empleado E ON U.id = E.id
JOIN Servicio S ON U.servicioID = S.id
WHERE U.tipoUsuario = 'Empleado';
```

Salida de la consulta

empleado_id	apellidos	nombres	cuil	mail	tel	area_de_servicio
4d127332-a3d0-11ef-a1e9-d05099803c6d	Romero	Soledad	27864567893	soleromero@hmu.com.ar	(Null)	Admisión
4d127098-a3d0-11ef-a1e9-d05099803c6d	Suárez	David	20554897343	dsuarez@hmu.com.ar	(Null)	Biomédica
4d127e16-a3d0-11ef-a1e9-d05099803c6d	Bustos	Sebastián	24554978443	sebustos@hmu.com.ar	(Null)	Biomédica
4d125b36-a3d0-11ef-a1e9-d05099803c6d	Vignetta	Maria Celeste	27554685443	mcv@hmu.com.ar	(Null)	Cirugía
4d126eca-a3d0-11ef-a1e9-d05099803c6d	Suizer	Alejandro	20526456443	amsuizer@hmu.com.ar	(Null)	Cirugía
4d127c56-a3d0-11ef-a1e9-d05099803c6d	Pasco	Diego	20456645843	dpasco@hmu.com.ar	(Null)	Cirugía
4d124a14-a3d0-11ef-a1e9-d05099803c6d	Aniceto	Juan	20124543421	janiceto@hmu.com.ar	(Null)	Diagnóstico por Imágenes
4d12769b-a3d0-11ef-a1e9-d05099803c6d	Rius	Pedro	20554674843	pedrorius@hmu.com.ar	(Null)	Diagnóstico por Imágenes
4d1264f8-a3d0-11ef-a1e9-d05099803c6d	Tolay	Edith	20454676443	etolay@hmu.com.ar	(Null)	Enfermería
4d1266e7-a3d0-11ef-a1e9-d05099803c6d	Terrieris	José Luis	20554543453	jterrieris@hmu.com.ar	(Null)	Enfermería
4d1268aa-a3d0-11ef-a1e9-d05099803c6d	Tarifa	Claudia	27540678443	clautarifa@hmu.com.ar	(Null)	Enfermería
4d12588e-a3d0-11ef-a1e9-d05099803c6d	Vargas Ruiz	Maria Laura	27554664563	mlvargas@hmu.com.ar	(Null)	Esterilización
4d125d67-a3d0-11ef-a1e9-d05099803c6d	Vivas	Alicia	27554567443	avivas@hmu.com.ar	(Null)	Habilitación
4d125f4a-a3d0-11ef-a1e9-d05099803c6d	Canga Castellanos	Matías Quique	20554649743	lestat@hmu.com.ar	(Null)	Informática
4d12615d-a3d0-11ef-a1e9-d05099803c6d	Garay	Mauricio Elio	24574741745	mgaray@hmu.com.ar	(Null)	Informática
4d127896-a3d0-11ef-a1e9-d05099803c6d	Rabbat	Damian	20558244443	drabbat@hmu.com.ar	(Null)	Traumatología y Ortopedia
4d127a7e-a3d0-11ef-a1e9-d05099803c6d	Puig	Ismael	20567974443	ipuig@hmu.com.ar	(Null)	Traumatología y Ortopedia

Listar Directores cargados en la BD

```
-- Listar usuarios de tipo Dirección cargados
SELECT
    BIN_TO_UUID(id) AS usuario_id,
    nombres,
    apellidos,
    mail,
    cuil,
```

```

fechaAlta,
sexo,
estado,
tipoUsuario
FROM Usuario
WHERE tipoUsuario = 'Direccion';

```

Salida de la consulta:

usuario_id	nombrs	apellidos	mail	cuil	fechaAlta	sexo	estado	tipoUsuario
4d12288d-a3d0-11ef-a1e9-d05099803c6d	Mariano Gustavo	Marino	direccion@hmu.com.ar	20224448885	2024-11-16	MASCULINO	1	Direccion
4d122be7-a3d0-11ef-a1e9-d05099803c6d	Federico	Huergo Sánchez	fedesubdir@hmu.com.ar	20259993331	2024-11-16	MASCULINO	1	Direccion
4d122d49-a3d0-11ef-a1e9-d05099803c6d	Gloria	Longoni	subdirectora@hmu.com.ar	27129997773	2024-11-16	FEMENINO	1	Direccion
4d122efe-a3d0-11ef-a1e9-d05099803c6d	Fabricio	Vitali	subdireccion@hmu.com.ar	20239997772	2024-11-16	MASCULINO	1	Direccion

Listar Servicios con sus Jefes de Servicio y Directores del cual dependen

```

-- Listar asignaciones de Jefes a Servicios y Director superior
SELECT
    BIN_TO_UUID(s.id) AS ServicioID,
    s.nombre AS ServNombre,
    s.agrupacion AS ServAgrupacion,
    BIN_TO_UUID(jds.id) AS JefaturaID,
    CONCAT(jefe.apellidos, ' ', jefe.nombrs) AS JefeFullName,
    BIN_TO_UUID(dir.id) AS DirectorID,
    CONCAT(director.apellidos, ' ', director.nombrs) AS DirFullName
FROM JefaturaDeServicio jds
JOIN Usuario jefe ON jds.id = jefe.id
JOIN Servicio s ON jefe.servicioID = s.id
JOIN Direccion dir ON s.direccionID = dir.id
JOIN Usuario director ON dir.id = director.id;

```

Salida de la consulta

ServicioID	ServNombre	ServAgrupacion	JefaturalD	JefeFullName	DirectorID	DirFullName
4cc2e0a1-a3d0-11ef-a1e9-d05099803c6d	Admisión	ADMINISTRATIVO	4cf98fc1-a3d0-11ef-a1e9-d05099803c6d	Balconte, Andrea	4d12288d-a3d0-11ef-a1e9-d05099803c6d	Marino, Mariano Gustavo
4cc9c085-a3d0-11ef-a1e9-d05099803c6d	Biomédica	TECNICO	4cfac8b1-a3d0-11ef-a1e9-d05099803c6d	Maestro, Silvina	4d122efe-a3d0-11ef-a1e9-d05099803c6d	Vitali, Fabricio
4cd07b8e-a3d0-11ef-a1e9-d05099803c6d	Cirugía	MEDICO	4cf727fc-a3d0-11ef-a1e9-d05099803c6d	Titarelli, Maximiliano	4d12288d-a3d0-11ef-a1e9-d05099803c6d	Marino, Mariano Gustavo
4cd57378-a3d0-11ef-a1e9-d05099803c6d	Diagnóstico por Imágenes	MEDICO	4cfe6dc0-a3d0-11ef-a1e9-d05099803c6d	Morales, Juan Ignacio	4d12288d-a3d0-11ef-a1e9-d05099803c6d	Marino, Mariano Gustavo
4cd86e49-a3d0-11ef-a1e9-d05099803c6d	Enfermería	ENFERMERIA	4cffbd5c-a3d0-11ef-a1e9-d05099803c6d	Plaza, Tania	4d12288d-a3d0-11ef-a1e9-d05099803c6d	Marino, Mariano Gustavo
4cda375e-a3d0-11ef-a1e9-d05099803c6d	Esterilización	SERVICIO	4d012be8-a3d0-11ef-a1e9-d05099803c6d	Pérez Cabral, Matías	4d122efe-a3d0-11ef-a1e9-d05099803c6d	Vitali, Fabricio
4cdd3b96-a3d0-11ef-a1e9-d05099803c6d	Farmacia	SERVICIO	4d0275d2-a3d0-11ef-a1e9-d05099803c6d	Arancibia, María Pía	4d122efe-a3d0-11ef-a1e9-d05099803c6d	Vitali, Fabricio
4cde6df3-a3d0-11ef-a1e9-d05099803c6d	Habilitación	ADMINISTRATIVO	4d03cb91-a3d0-11ef-a1e9-d05099803c6d	Roberts, Carlos Fernando	4d12288d-a3d0-11ef-a1e9-d05099803c6d	Marino, Mariano Gustavo
4ce0c674-a3d0-11ef-a1e9-d05099803c6d	Informática	TECNICO	4d0518f7-a3d0-11ef-a1e9-d05099803c6d	Roqué, Juan Manuel	4d122efe-a3d0-11ef-a1e9-d05099803c6d	Vitali, Fabricio
4cea533c-a3d0-11ef-a1e9-d05099803c6d	Nutrición	MEDICO	4d065f00-a3d0-11ef-a1e9-d05099803c6d	Boqué, Alejandra	4d12288d-a3d0-11ef-a1e9-d05099803c6d	Marino, Mariano Gustavo
4cf864c7-a3d0-11ef-a1e9-d05099803c6d	Traumatología y Ortopedia	MEDICO	4d097b04-a3d0-11ef-a1e9-d05099803c6d	Sánchez, Omar Wenceslao	4d12288d-a3d0-11ef-a1e9-d05099803c6d	Marino, Mariano Gustavo

Contabilizar cantidad de empleados por servicio incluyendo jefes

```
-- Contar la cantidad de empleados de cada servicio (incluidos sus jefes)
SELECT
    S.nombre AS Servicio,
    COUNT(U.id) AS Cant_empleados
FROM Servicio S
LEFT JOIN Usuario U ON U.servicioID = S.id
WHERE U.tipoUsuario IN ('Empleado', 'JefaturaDeServicio')
GROUP BY S.id, S.nombre
ORDER BY Cant_empleados DESC;
ORDER BY total_empleados DESC;
```

Salida de la consulta

Servicio	Cant_empleados
Cirugía	4
Enfermería	4
Traumatología y Or	3
Personal	3
Biomédica	3
Diagnóstico por Im	3
Informática	3
Admisión	2
Esterilización	2
Habilitación	2
Nutrición	1
Farmacia	1

Agregar/modificar un dato a un usuario

```
-- Agregar/modificar el teléfono a un usuario cualquiera
-- Por ejemplo, elegimos modificar el teléfono al jefe de Informática (que ya teníamos almacenado en
@JefeInformaticaID)
-- Actualizamos el teléfono del usuario
UPDATE Usuario
SET tel = '3517553799'
WHERE id = @JefeInformaticaID;
-- Verificamos nuevo valor:
SELECT nombres, apellidos, tel FROM Usuario WHERE id = @JefeInformaticaID;
```

Borrado de un usuario

```
-- Borrar un usuario
-- Supongamos que el usuario a eliminar es 'Soledad Romero'
SET @usuarioIdEliminar = (SELECT id FROM Usuario WHERE apellidos = 'Romero' AND nombres = 'Soledad' LIMIT 1);

-- Verificar de qué tipo de usuario se trata
SELECT tipoUsuario
FROM Usuario
WHERE id = @usuarioIdEliminar;

-- Eliminación de tablas específicas
-- Si es Empleado
DELETE FROM Empleado WHERE id = @usuarioIdEliminar;

-- Si es JefaturaDeServicio
DELETE FROM JefaturaDeServicio WHERE id = @usuarioIdEliminar;

-- Si es Dirección
DELETE FROM Direccion WHERE id = @usuarioIdEliminar;

-- Si está en OficinaDePersonal
DELETE FROM OficinaDePersonal WHERE id = @usuarioIdEliminar;

-- Finalmente, eliminar de la tabla Usuario
DELETE FROM Usuario WHERE id = @usuarioIdEliminar;
-- Opcionalmente, si el usuario tuviera registros en otras tablas (como Planificacion, Novedad, etc.), también
los eliminaríamos (a fines de evitar inconsistencias)
-- No es el caso
```

18. Presentación de las consultas SQL

Como se mencionó anteriormente, las consultas están en un repositorio de **GitHub**. Se indican los enlaces directos a los archivos SQL:

- Archivo de creación de las tablas: `aromito_creacion_de_tablas.sql`
https://raw.githubusercontent.com/linkstat/seminare/refs/heads/main/sql/aromito_creacion_de_tablas.sql
- Archivo de consultas CRUD: `aromito_operaciones_CRUD.sql`
https://raw.githubusercontent.com/linkstat/seminare/refs/heads/main/sql/aromito_operaciones_CRUD.sql
- Archivo de borrado total: `aromito_borrado_preinit.sql`
https://raw.githubusercontent.com/linkstat/seminare/refs/heads/main/sql/aromito_borrado_preinit.sql

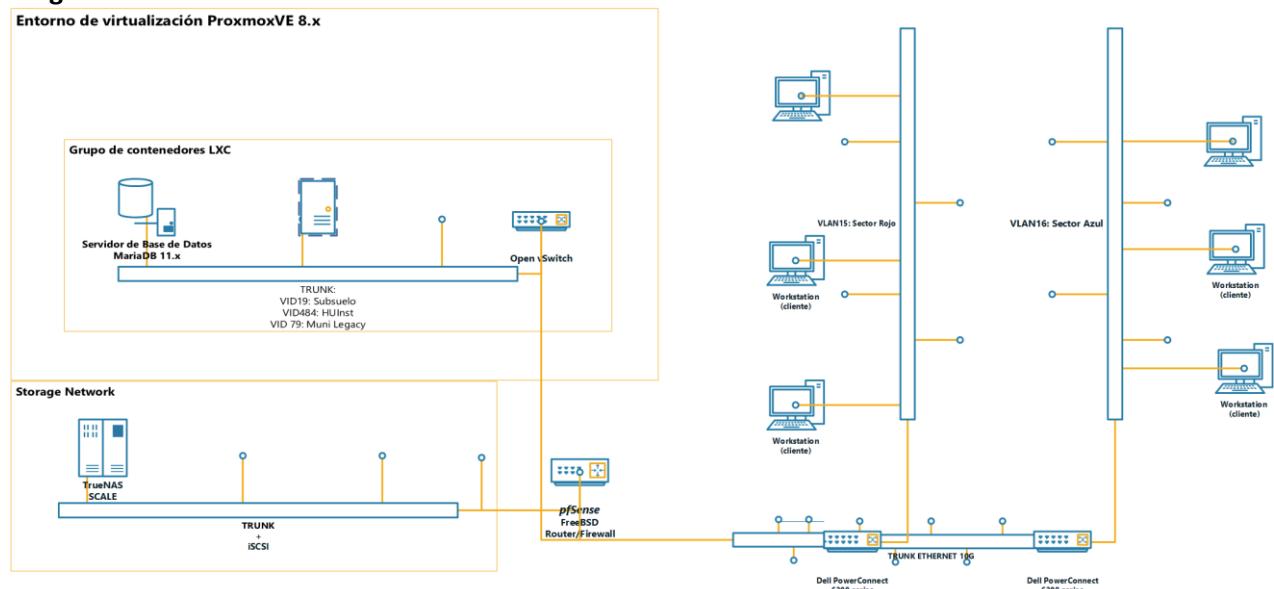
19. Definiciones de comunicación

El lugar en donde se planea desplegar el sistema, cuenta con una buena infraestructura de red. Se trata de una red con cableado estructurado Cat-6, soportada casi enteramente por unos cerca de 15 switchs *Dell PowerConnect 6200 series*, conectados en topología tipo estrella, instalados en 13 racks distribuidos por el lugar.

El edificio está sectorizado en diversas VLANs:

- 10 VLANs IPv4 /25 para estaciones de trabajo, (**desde las cuales se conectarán nuestros clientes**). Cableadas: giga Ethernet.
- 3 VLANs IPv4 para: dispositivos WiFi del hospital, equipos inalámbricos de empleados y una red de pública para acceso a Internet.
- 1 VLAN “*Baremetal*” para hipervisores, storage y equipos comunicaciones.
- 1 VLAN de Servidores, accesible además desde el resto de la red del municipio. **Sobre esta red correrá nuestro servidor de base de datos MariaDB**, ejecutándose como un **contenedor LXC, inicialmente, con IPv4 172.16.79.39/24**.
- Finalmente, todo el tráfico entre VLANs internas y con Internet, está interconectado y regulado por un router/firewall *pfSense* (pronto a ser relevado por *OPNsense*, cuando se implemente IPv6 definitivamente).

Diagrama



20. Explicación del desarrollo en Java

Sin perder de vista que el proyecto debe contemplar el desarrollo completo del Sistema de Gestión Ausentismo Hospitalario, en esta fase se trabajará con un **prototipo** que permitirá obtener un **Producto Mínimo Viable (MVP)**. Este prototipo proporcionará las funcionalidades esenciales que

serán utilizadas como base para el desarrollo completo, y ofrecerá una visión preliminar de lo que el sistema final podría alcanzar.

De acuerdo con las definiciones tomadas en el **Proceso Unificado de Desarrollo (PUD)**, el sistema requiere el uso de más de **30 clases**, algunas de las cuales son **abstractas**, así como **interfaces y enumeraciones (enum)**. Además, se incluirán nuevas clases e interfaces específicas para la construcción de la **Interfaz Gráfica de Usuario (GUI)**, asegurando una experiencia de usuario adecuada.

Alcance del Prototipo

El objetivo mínimo de este prototipo es ofrecer las siguientes funcionalidades clave:

- Autenticación y validación de usuarios.
- Gestión de Altas, Bajas y Modificaciones (ABM) de empleados.
- Asignación y administración de roles.
- Registro de novedades (inicialmente, realizadas de forma manual por la Oficina de Personal).
- Generación y exportación del reporte "Parte Diario" (un tipo específico de reporte de que la Oficina de Personal debe enviar a Capital Humano semanal y mensualmente).

El objetivo mínimo, es que este prototipo cubra una de las principales necesidades actuales de la Oficina de Personal, facilitando la gestión de las novedades de ausentismo de los empleados del nosocomio.

En una segunda fase, se tiene prevista la implementación del envío de memorándums, digitalizando así un proceso que actualmente se realiza de forma manual en formato papel. La incorporación del módulo de memorándums permitiría reducir la carga de papeleo manual y agilizar la comunicación interna, completando aún más el alcance del prototipo.

Estructura de Desarrollo

El proyecto está organizado siguiendo principios sólidos de separación de responsabilidades, tomando como referencia una arquitectura **modular** que combina el patrón **MVC (Modelo-Vista-Controlador)** con otras capas específicas para la lógica de negocio, autenticación, persistencia de datos, y utilidades generales. Esta estructura permite una división clara y lógica de las funcionalidades del sistema:

- **Model (Modelo):** Contiene las clases que representan los datos de la aplicación. Estas clases se utilizan para definir las entidades que corresponden al dominio del sistema, como **Usuario** o **Empleado**. Estas entidades están organizadas en el paquete **model** y no contienen lógica adicional; únicamente los atributos y métodos para gestionar el estado de los datos.
- **View (Vista):** Se encarga de la presentación y la interacción con el usuario. En el paquete **ui** se encuentran las clases que implementan la interfaz gráfica de usuario, utilizando **JavaFX**. Estas clases manejan la lógica relacionada con la visualización y la interacción directa con los usuarios, como la pantalla de inicio de sesión.
- **Auth (Autenticación):** El paquete **auth** contiene clases y lógica relacionadas con la autenticación y autorización de usuarios. Aquí se gestiona la seguridad del sistema y la validación de credenciales de inicio de sesión.

- **Controller** (Controlador): El paquete `controller` contiene las clases responsables de coordinar la comunicación entre la vista (`ui`) y las capas de servicio y repositorio. Actúan como intermediarios que manejan las interacciones del usuario, procesan las entradas, y actualizan las vistas.
- **Service** (Servicio): En el paquete `service` se encuentran las clases que encapsulan la lógica de negocio. Estas clases se encargan de procesar la información y aplicar reglas de negocio específicas, como `LoginService`, que valida las credenciales de usuario antes de proceder con el inicio de sesión. Esta capa también coordina entre la vista y los repositorios.
- **Repository / DAO**: Para gestionar la persistencia de los datos, se utiliza el patrón **DAO** (*Data Access Object*), ubicado en el paquete `repository`, el cual contiene un subpaquete `dao` para gestionar operaciones específicas de acceso a la BD. Esta capa se encarga de la interacción directa con la BD, gestionando operaciones **CRUD** (crear, leer, actualizar, eliminar). El `DatabaseConnector` y las clases como `UsuarioDAO` encapsulan el acceso a la BD, proporcionando métodos específicos para realizar consultas, de tal manera que la lógica de acceso a datos se desacopla de la lógica de negocio.
Para clarificar aún más: el paquete `repository` se enfoca en repositorios de alto nivel o servicios que requieren integración con la capa de acceso a datos, mientras que el paquete `repository/dao` agrupa específicamente el acceso directo a la BD.
- **Factory** (Fabrica): contiene clases y métodos responsables de la creación de instancias complejas de objetos de acuerdo con su tipo específico. La idea es centralizar la lógica de instantiación (respetando el principio de responsabilidad única). Aquí se incluye la clase `UsuarioFactory`, que permite la creación de diferentes tipos de usuarios (como `Empleado`, `JefaturaDeServicio`, `OficinaDePersonal`, y `Direccion`) a partir de datos obtenidos de la base de datos.
- **Utils** (Utilidades): El paquete `utils` contiene clases que proporcionan funcionalidades generales y reutilizables que no pertenecen a una capa específica del modelo, tales como métodos auxiliares para validación, conversión de formatos, o manejo de configuraciones comunes.
- **Config** (Configuración): En el paquete `config` se encuentran las clases relacionadas con la configuración del sistema. `ConfigReader` y `DatabaseConfig` están aquí para centralizar la carga de configuraciones desde archivos externos, como `config.yaml`, y proporcionar una única fuente de verdad para los parámetros de conexión y otros ajustes de la aplicación.

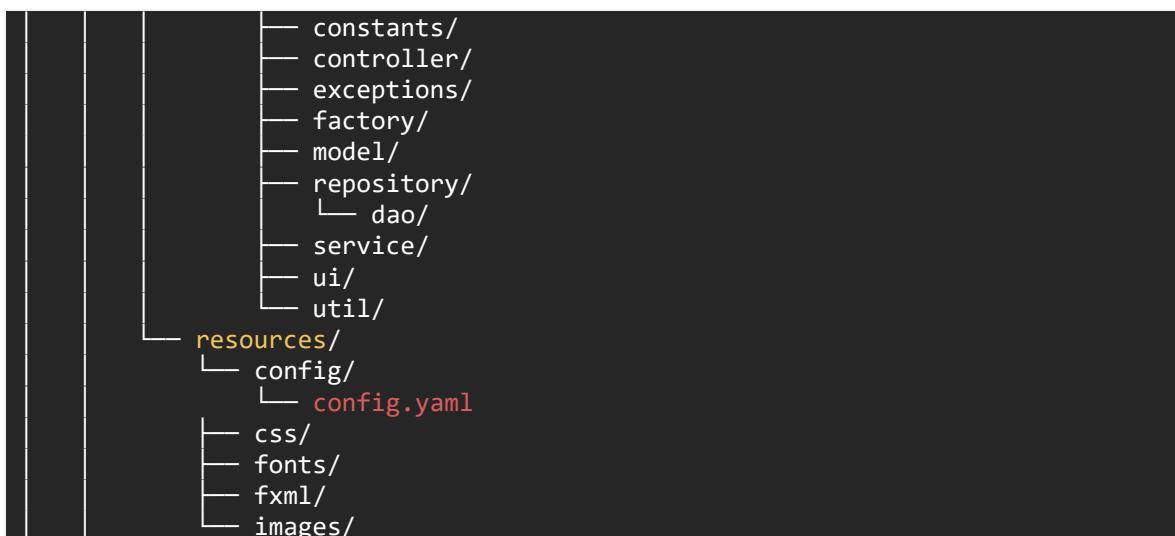
Este enfoque modular no solo permite una división clara de responsabilidades, sino que también mejora la **mantenibilidad** y **escalabilidad** del sistema. Al mantener la lógica de presentación, negocio, autenticación, y persistencia en capas claramente separadas, es posible extender cada componente del sistema de forma independiente sin afectar al resto, lo que contribuye a la robustez y flexibilidad del desarrollo.

Estructura de Paquetes

La estructura de paquetes del proyecto se ha diseñado para seguir una **organización clara y lógica**:

[raíz del proyecto]/





Dentro de **src.main.java**, tenemos todos los archivos referidos al **código fuente** del proyecto. Dentro de **src.main.resources**, tenemos todos los archivos referidos a **recursos no compilables** del proyecto.

Estructura de ramas en Github

Todo el material del proyecto, incluidos archivos de documentación, imágenes, código, etc., se almacena en un repositorio de GitHub dedicado. Este repositorio cuenta con dos ramas principales: **main** y **testing**.

- **Rama testing:** Es la rama de desarrollo activa, donde se realiza el trabajo diario, incluyendo nuevas funcionalidades, ajustes y pruebas.
- **Rama main:** Es la rama de producción, que refleja una versión estable del proyecto. Los cambios en esta rama se realizan cuando se alcanza un hito significativo y se validan los avances realizados en la rama **testing**.

Esta estructura de ramas facilita un flujo de trabajo ordenado, permitiendo realizar pruebas de manera segura antes de incorporar los cambios definitivos al proyecto principal.

Software de desarrollo utilizado

Versión de Java

La versión del **Java Development Kit (JDK)** seleccionada para el desarrollo es **OpenJDK 23.0.1**. Esta decisión se tomó para aprovechar la última versión disponible al momento del inicio del proyecto, dado que es un **sistema nuevo** y se desea contar con todas las mejoras de rendimiento, compatibilidad y seguridad que incluye la versión más reciente.

Desarrollo de la Interfaz Gráfica

Para el desarrollo de la interfaz gráfica, se optó por **JavaFX 23.0.1**. El enfoque es ofrecer la mejor experiencia de usuario posible, y mantener el proyecto con un aspecto visual tecnológicamente actualizado. En este sentido, a comparación de **Swing** (la opción robusta de antaño), **JavaFX** es la más actual y ofrece ventajas significativas, como:

1. Aspecto Moderno y Personalizable.

2. Mejores Herramientas para Diseño: (*Scene Builder*).
3. Mayor Flexibilidad.
4. Compatibilidad con FXML.

Para la asistencia en el diseño de las pantallas, se hará uso de ***Scene Builder 23***.

Entorno de desarrollo integrado (IDE)

La IDE (*Integrated Development Environment*) utilizada será ***IntelliJ IDEA 2024.2.3 (Ultimate Edition)***. Es un software popular, con mucha documentación disponible en línea y su numerosa comunidad de usuarios desarrolladores la hacen ideal para recurrir a consultas e intercambio de ideas tendientes a la resolución de potenciales problemas.

El **Entorno de Desarrollo Integrado (IDE: Integrated Development Environment)** elegido para el proyecto es ***IntelliJ IDEA 2024.2.3 (Ultimate Edition)***. Propiciaron esta decisión:

1. Gusto personal.
2. Facilidad de uso.
3. Funcionalidades disponibles.
4. Amplia comunidad de usuarios, lo cual garantiza acceso a abundante documentación y soporte en línea.

IntelliJ proporciona herramientas avanzadas de desarrollo, como la gestión integrada de control de versiones, herramientas de depuración y soporte mejorado para tecnologías modernas.

Gestión y Construcción del Proyecto

Para la **gestión y construcción del proyecto**, se utilizará ***Apache Maven***, una herramienta de automatización que facilita la gestión del ciclo de vida del proyecto Java. Se decidió hacer uso de esta herramienta por las siguientes razones:

- **Gestión de dependencias:** Permite la integración de bibliotecas externas, como *JavaFX* y el *driver JDBC de MariaDB*, etc. de forma sencilla.
- **Compilación y Empaquetado:** Automatiza la compilación del código fuente y el empaquetado de la aplicación en archivos ejecutables (.jar o .war).
- **Ejecución de pruebas:** Facilita la ejecución de pruebas unitarias y la validación continua del código.
- **Consistencia del entorno:** Utilizando Maven, el proyecto mantiene un entorno de desarrollo homogéneo en diferentes máquinas, asegurando que todos los equipos trabajen con versiones compatibles de librerías, eliminando así problemas de compilación.

Maven utiliza un archivo llamado pom.xml (*Project Object Model*), el cual define las dependencias del proyecto, el proceso de construcción, y las tareas específicas, garantizando un entorno reproducible y consistente en todas las fases del desarrollo.

Otras herramientas

- **GIMP:** Software libre utilizado para la edición y creación de imágenes en formatos como PNG e ICO, destinado principalmente a íconos y otros elementos gráficos del programa.
- **Inkscape:** Software libre utilizado para la edición y creación de gráficos vectoriales en formato SVG, ideal para generar íconos y otros elementos gráficos del programa.

- **Javadoc:** Herramienta estándar de Java utilizada para generar la documentación del proyecto en formato HTML. A partir de los comentarios incluidos en cada clase y método, *Javadoc* crea una documentación detallada que facilita la comprensión y uso del código.

Clase de inicio principal

El método `main`, se encuentra en la clase `LoginScreen`.

Otras consideraciones

Cifrado de contraseñas

Para la validación de usuario, utilizamos el atributo de CUIL como nombre de usuario y el atributo `pass` como contraseña. Almacenamos las contraseñas en forma de hash (nunca en texto plano) utilizando **BCrypt**, (una librería de cifrado para Java), esto mejora la seguridad contra ataques de fuerza bruta.

Consideraciones de diseño

Durante el diseño inicial, se incluyeron métodos como `iniciarSesion(int credenciales)` y `cerrarSesion()` en la clase `Usuario` para gestionar la autenticación. Sin embargo, en la implementación final, estos métodos fueron eliminados ya que la autenticación se delega a `LoginService`, una clase dedicada a manejar el proceso de validación. La clase `Usuario` se enfoca únicamente en representar y mantener el estado del usuario. Este cambio respeta los principios de **responsabilidad única y desacoplamiento**, evitando que `Usuario` gestione lógica de negocio o conexiones a la BD.

En la base de datos, la tabla `Usuario` no tiene una clave foránea hacia la tabla `Servicio`, al igual que las tablas `Direccion` y `OficinaDePersonal`. Sin embargo, las tablas `Empleado` y `JefeDeServicio` sí cuentan con dicha relación. Esto se debe a que, en el modelo de datos, los usuarios de tipo `Direccion` siempre pertenecen al servicio de Dirección, mientras que los usuarios de tipo `OficinaDePersonal` siempre pertenecen al servicio de Oficina de Personal. Debido a que estas asociaciones son fijas, no es necesario almacenarlas como claves foráneas en la base de datos.

En la implementación del diseño de clases, se decidió incluir un atributo `servicio` (de tipo `Servicio`) en la clase abstracta `Usuario` y heredar dicho atributo en todas las subclases. Esta decisión aporta uniformidad en el modelo de clases y facilita la gestión de los objetos, ya que todos los tipos de usuarios comparten una estructura común.

Este enfoque representa una forma de **normalización a nivel de base de datos y desnormalización a nivel del modelo de clases de aplicación**, lo cual implica agregar un poco de redundancia en el código para simplificar el trabajo con objetos y mejorar la consistencia. En el nivel de base de datos, se gestionan estas relaciones con lógica específica (o bien se establece el valor fijo a nivel del código de aplicación cuando se instancian los objetos). Esto permite que el diseño de clases sea más limpio y el código más mantenible, evitando excepciones o lógica condicional para diferenciar entre los distintos tipos de usuarios.

Archivo de configuración del programa

Para el almacenamiento de los parámetros de configuración, como los relacionados con la conexión a la base de datos, se optó por utilizar un archivo en formato **YAML**. Originalmente, se había considerado el uso de un archivo **config.properties**, el cual es tradicional en Java. Sin embargo, se prefirió **YAML** debido a varias ventajas significativas:

- **Modernidad y Legibilidad:** YAML es un formato más moderno y legible, lo cual facilita la comprensión y edición manual de las configuraciones.
- **Soporte Multilingüe:** YAML tiene un excelente soporte para **UTF-8**, permitiendo configuraciones que incluyan caracteres especiales y múltiples idiomas, lo cual es ideal para entornos multilingües. En comparación, los archivos **config.properties** más antiguos usan codificación **ANSI**, que puede presentar problemas con caracteres no latinos (como acentos o la letra ñ, usados principalmente en comentarios).

YAML permite estructurar la configuración en subniveles y hace que el archivo sea más flexible y fácil de mantener en el futuro, a medida que se vayan agregando más parámetros de configuración.

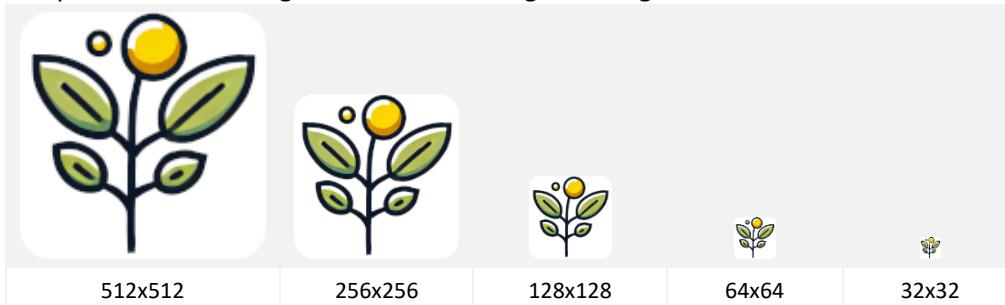
Logo del programa

A partir de una fotografía de un aromito serrano, se generó una imagen simplificada que resultara agradable y fácilmente reconocible de un vistazo, acompañada por la palabra Aromito, con tipografía Pristina tamaño 72px, color **DARKGOLDENROD** y trazo de acho 0.25px color **DARKSLATEGRAY**, y rotación -5°, resultando esto en lo siguiente:



Icono del programa

A partir de la imagen simplificada del logo, se generó una imagen que resultara visualizable y fácilmente distinguible en tamaños pequeños, de 32x32 o 64x64, que son dimensiones típicas para íconos de aplicaciones. La imagen resultante del logo es la siguiente:



21. Referencias al Código y Documentación del Proyecto

Dado el volumen significativo del código fuente y de la documentación generada mediante *JavaDoc*, su inclusión directa en este documento no resulta práctica. En su lugar, se proporciona una referencia al repositorio de GitHub donde están disponibles:

Código Fuente del Proyecto

El código Java del proyecto está disponible para descarga y consulta en el siguiente repositorio de GitHub:

GitHub - Código del Proyecto

- Raíz del proyecto (puede cargarse en IntelliJ):
<https://github.com/linkstat/seminare/tree/main>
- Clases estructuradas en paquetes:
<https://github.com/linkstat/seminare/tree/main/src/main/java/ar/com/hmu>

Documentación JavaDoc

La documentación detallada del proyecto, generada con *JavaDoc*, puede ser consultada en:

GitHub - Documentación JavaDoc

- <https://github.com/linkstat/seminare/tree/main/docs/javadoc>

22. Video de Presentación del Proyecto

Para ilustrar el alcance del proyecto, se desarrolló un video que muestra el funcionamiento del programa en sus primeras fases de desarrollo. Este puede visualizarse en el siguiente enlace:

[GitHub – Video de Presentación del Proyecto](#)

- <https://github.com/linkstat/seminare/tree/main/videos>

En el video se presenta:

- **Pantalla de inicio de sesión:** Incluyendo opciones adicionales.
- **Menú principal:** Estructurado en mosaicos.
- **Barra de menú:** Acceso organizado según roles de usuario.
- **Pantallas ABM (Altas, Bajas, Modificaciones):**
 - Usuarios.
 - Servicios.
 - Cargos.
- **Pantalla de reportes:**
 - Listado de usuarios.
 - Listado de servicios.
 - Listado de usuarios por servicio.
- **Pantalla de licencias de uso.**
- **Pantalla "Acerca de".**

El video también incluye breves referencias a la implementación del código fuente que sustenta estas funcionalidades.

23. Referencias

Ordenanza N° 7244 Estatuto del empleado municipal (texto actualizado al 2019), Recuperado de:

<https://documentos.cordoba.gob.ar/TDC/ordenanza-n-7244-estatuto-de-personal-2019.pdf>

Digesto Web Municipalidad de Córdoba, textos modificatorios a la ordenanza 7244. Recuperado de: <https://servicios2.cordoba.gov.ar/DigestoWeb/Page/Documento.aspx?Nro=1624>

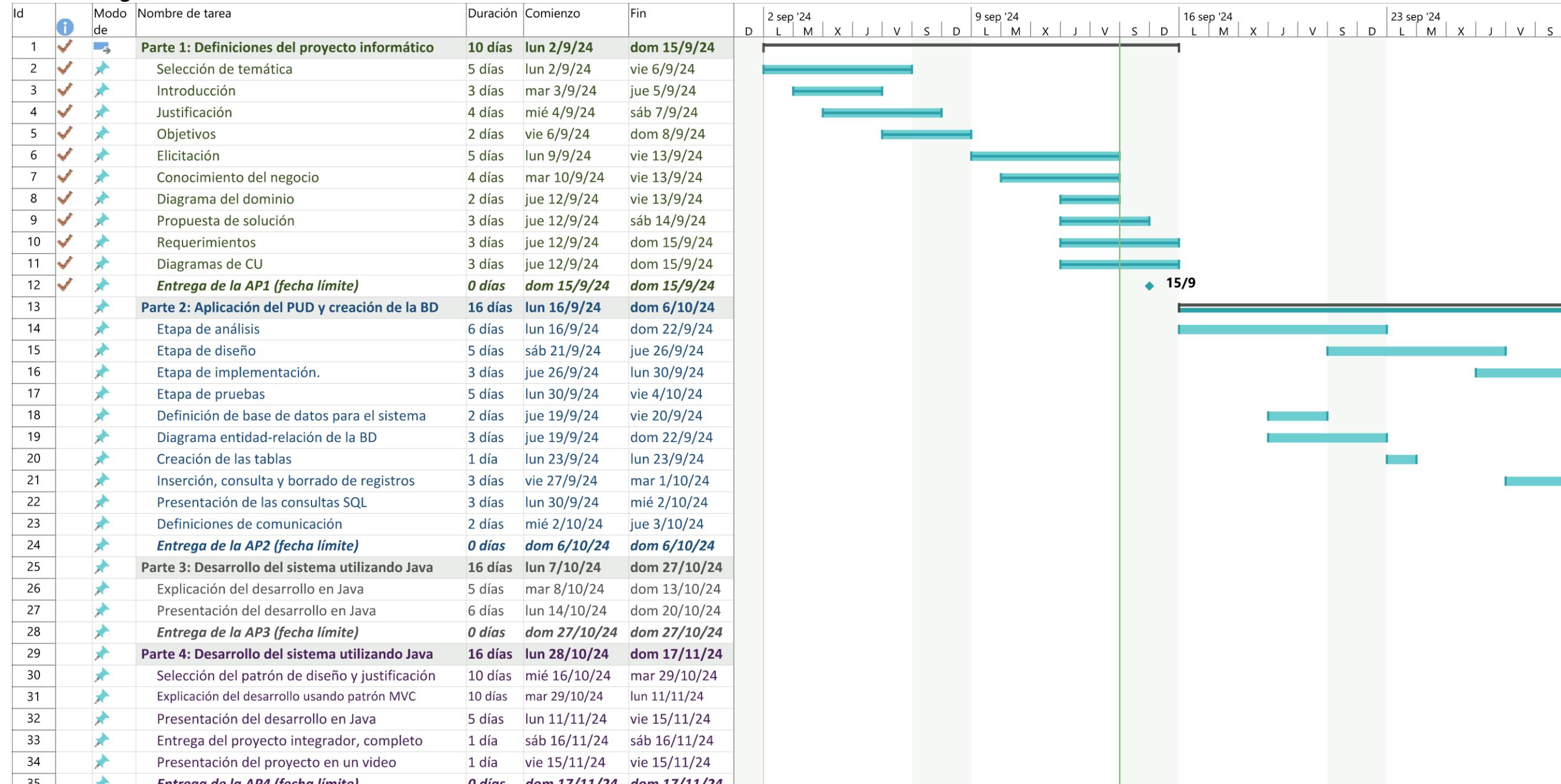
Grady Booch, James Rumbaugh, Ivar Jacobson, (2006) El Lenguaje Unificado de Modelado. Guía del usuario. Segunda Edición. Pearson Educación S.A., Madrid, 2006.

Ian Sommerville, (2011) Ingeniería de Software. Novena Edición. Pearson Educación de México S.A., 2011.

Laurent Debrauwer, Fien Van Der Heyde, (2020), UML 2.0 Iniciación, ejemplos y ejercicios corregidos. 5ª edición. Ediciones ENI. Barcelona, 2020.

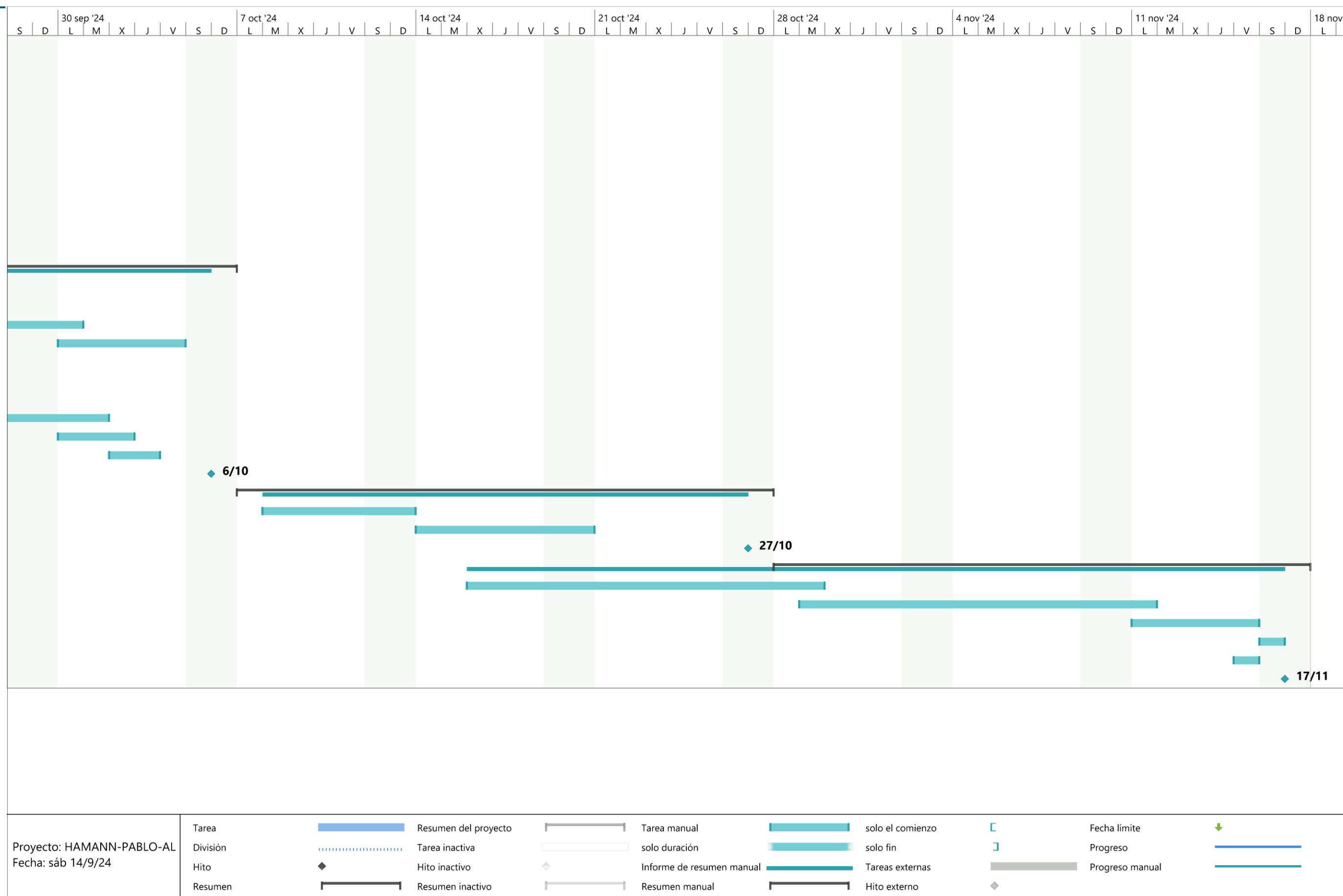
Repositorio GitHub de Pablo Alejandro Hamann para este proyecto, (2024), Disponible en:
<https://github.com/linkstat/seminare/>

Anexo I: Diagrama de Gantt



Proyecto: HAMANN-PABLO-AL	Tarea	Resumen del proyecto	Tarea manual	Resumen manual	Fecha límite
Fecha: sáb 14/9/24	División	Tarea inactiva	solo duración	solo el comienzo	↓
	Hito	Hito inactivo	informe de resumen manual	solo fin	Progreso
	Resumen	Resumen inactivo	Tareas externas	Resumen externo	Progreso manual
			Hito externo		

Sistema de Gestión de Ausentismo Hospitalario



Anexo II: Organigrama del Hospital Municipal de Urgencias (HMU)

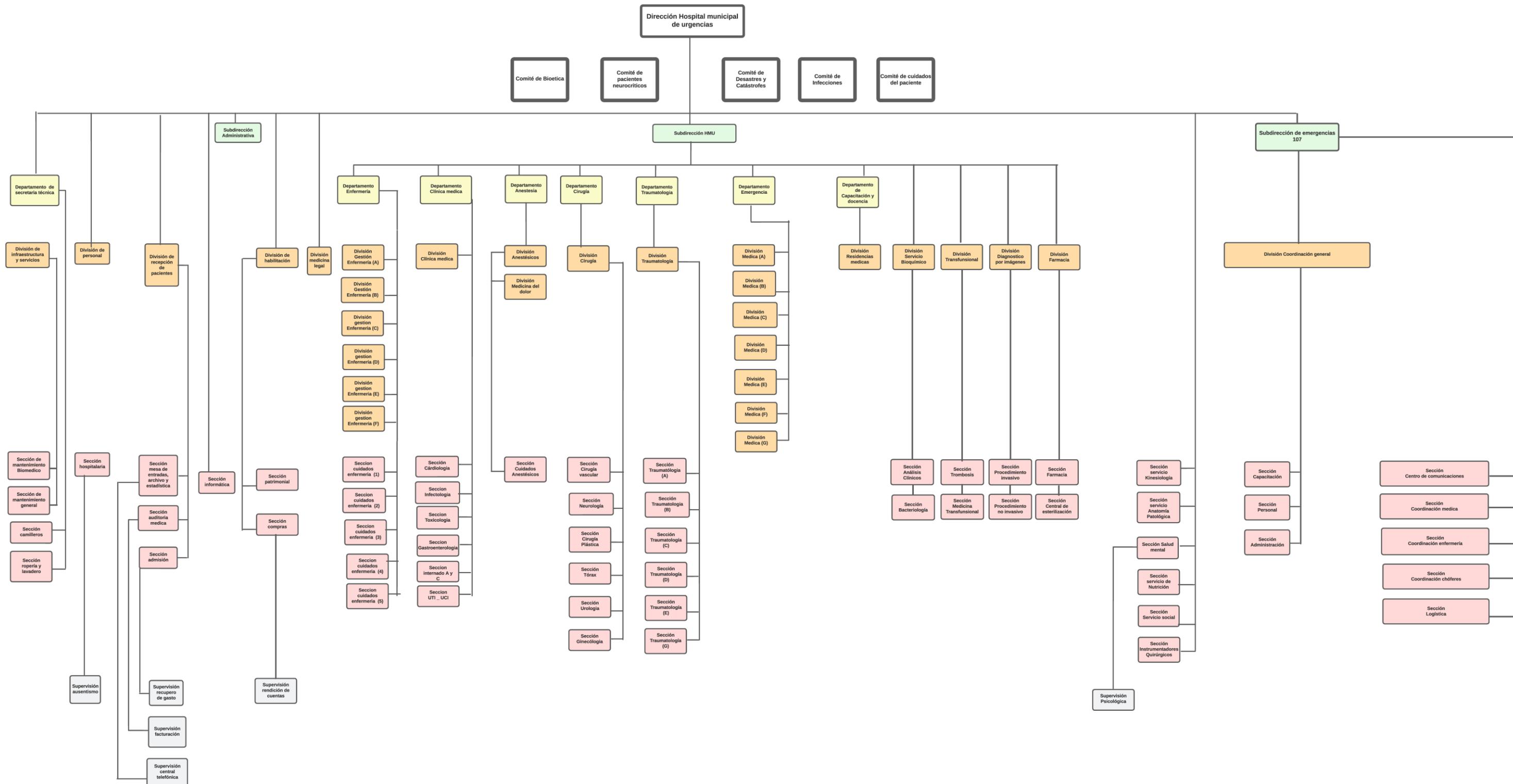


Imagen original disponible en: <https://raw.githubusercontent.com/linkstat/seminare/refs/heads/main/docs/organigrama/HMU.Organigrama.png>

Archivo fuente (formato PDF) disponible en: <https://github.com/linkstat/seminare/blob/6d1c39b5d34db6f45fb990763c4fcba5abafb436/docs/organigrama/HMU.Organigrama.pdf>