Anexo V
Informe Final del TFI

Universidad Nacional del Noroeste de la Provincia de Buenos Aires

Título: Autogestión de salud: Desarrollo e implementación de un módulo para la digitalización de la venta de bonos de La Pequeña Familia
Carrera: **Ingeniería Informática**
Práctica Profesional Supervisada (PPS)

Estudiante: Mijael Andrés Koroluk
Tutor Docente: Sabrina Pompei
Tutor de Empresa/Institución/Organización: Roberto García
Fecha de presentación: 13 de abril de 2026

Anexo V
Informe Final del TFI

1. Introducción.....pág 3
2. Objetivos.....pág 3
3. Plan de Trabajo y Carga Horaria.....pág 4
4. Descripción del Trabajo Final Integrador efectuado.....pág 6
5. Conclusiones.....pág 21
6. Bibliografía.....pág 22
7. Anexos.....pág 22
8. Agradecimientos.....pág 24

Anexo V
Informe Final del TFI

Presentación del Informe

Lineamientos Generales

1. *Introducción*

La Pequeña Familia es un grupo de salud originario de Junín, con múltiples sedes a lo largo de la provincia de Buenos Aires y provincias adyacentes.

El presente trabajo se desarrolló en su Casa Central (Gandini 128, Junín), núcleo administrativo y comercial de sus planes de medicina prepaga y medicina integral.

En estas oficinas se encuentra una parte del departamento de Sistemas (ya que la clínica aloja la otra parte), el cual cuenta con aproximadamente 10 personas, y repartidas en múltiples subáreas, como infraestructura, procesos y desarrollo de software.

En esta última se llevaron a cabo las prácticas profesionales supervisadas. Como parte de las actividades, se asignó la resolución de una problemática operativa concreta: la venta presencial de bonos para consultas médicas. Este proceso manual, gestionado por el personal de atención al cliente, presentaba demoras significativas y exigía el traslado físico del afiliado a las oficinas administrativas previo a la atención médica. Esta dificultad se acentuaba críticamente en el caso de los bonos de "guardia", ante situaciones de urgencia, donde el tiempo y la movilidad son factores limitantes.

Para solucionar a esta problemática desde la perspectiva de la ingeniería de software, se diseñó e implementó un nuevo módulo de gestión de bonos dentro del Portal de Afiliados de La Pequeña Familia, un sistema web de autogestión preexistente para los afiliados. Esta solución tecnológica digitaliza y automatiza el proceso, permitiendo a los usuarios visualizar, solicitar, cancelar, pagar y descargar sus bonos de forma remota y centralizada.

Finalmente, el objetivo del presente informe es documentar el proceso técnico llevado a cabo para la construcción de dicho módulo, detallando las etapas de análisis de requerimientos, diseño arquitectónico, desarrollo, integración con sistemas externos (ERP y pasarelas de pago) y las pruebas de validación efectuadas durante la práctica profesional.

2. *Objetivos*

Objetivo General

Desarrollar e implementar un módulo de gestión de bonos dentro del Portal de Afiliados de La Pequeña Familia, con el fin de resolver la problemática operativa de la venta presencial mediante la digitalización y centralización del proceso. El alcance funcional de esta gestión comprenderá la visualización del historial de bonos, la adquisición y compra de nuevos bonos y su posterior descarga e impresión.

Objetivos Específicos

Para alcanzar el objetivo general propuesto, se definieron los siguientes niveles de logro:

- **Análisis y Diseño Arquitectónico**: Definir los requerimientos funcionales y no funcionales del sistema, y diseñar la arquitectura del módulo web, modelando su integración estructural con las reglas de negocio y la base de datos del ERP subyacente (Odoo).
- **Desarrollo e integración de Sistemas**: Construir tanto la lógica de negocio, así como la interfaz del usuario, garantizando la interoperabilidad bidireccional tanto del sistema central (Odoo) para la sincronización de información, como con la pasarela externa (Mercado Pago) para la gestión financiera.
- **Validación y Documentación**: Certificar la estabilidad y el correcto funcionamiento del software mediante ciclos de pruebas funcionales e integrales, consolidando simultáneamente el acervo documental técnico y los manuales de usuario finales.

3. Plan de Trabajo y Carga Horaria

CRONOGRAMA DE TAREAS											
Diagrama de Gantt de la distribución en el tiempo de las actividades a realizar en el TFI y entrevistas con el/la Tutor/a Docente para informar el grado de avance del Plan de Trabajo											
N°	***ACTIVIDADES***	***TIEMPO DE DURACIÓN***									
		SEMANAS									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	Relevamiento y definición de Requerimientos	X									

2	Análisis de tablas y restricciones de Odoo (Documentación)	X	X								
3	Diseño de la implementación			X							
4	Desarrollo e Integración 1: Visualización y lógica (Frontend / Backend)				X						
5	Desarrollo e Integración 2: Conexión API Odoo y sincronización					X	X				
6	Desarrollo de la Integración con la API de Mercado Pago y conexión con el frontend y Odoo.							X			
7	Testing iterativo, pruebas integrales y				X	X	X	X	X	X	

	corrección de errores										
8	Documentación técnica y manuales de uso	X	X	X	X	X	X	X	X	X	X
9	Reuniones con el docente tutor		X	X	X		X	X			X

Durante la ejecución del plan de trabajo, se registró un único desfasaje menor respecto al cronograma original. La actividad N° 4 ("Desarrollo e Integración 1: Visualización y lógica (Frontend / Backend)") requirió un tiempo de implementación superior al estimado en la planificación inicial, extendiendo su ejecución durante los primeros días de la quinta semana.

No obstante, la tarea logró estabilizarse y completarse de manera exitosa dentro de ese mismo periodo transicional. Esta leve extensión temporal fue absorbida de manera eficiente, por lo que no comprometió el inicio de las actividades de integración subsiguientes ni generó demoras en el cumplimiento de los plazos generales del proyecto.

4. Descripción de la Práctica Profesional Efectuada

El presente trabajo consistió en la ejecución de un proyecto integral de ingeniería de software estructurado en etapas fundamentales. Estas fases abarcaron el relevamiento y diseño del sistema, el desarrollo e integración de la solución tecnológica, y los respectivos ciclos de pruebas y verificación.

Paralelamente, se elaboró la documentación técnica correspondiente para garantizar la mantenibilidad del código y del proyecto a lo largo de su ciclo de vida.

El proyecto se gestionó mediante la adopción de metodologías ágiles, específicamente bajo el marco de trabajo Scrum. Para ello, se establecieron reuniones de seguimiento semanales que permitieron monitorear el avance, resolver bloqueos y adaptar los requerimientos dinámicamente.

4.1. Contexto tecnológico y Sistema Base

El desarrollo del nuevo módulo exigió su integración sobre un entorno preexistente: el Portal de Afiliados de La Pequeña Familia. Este portal consiste en un sistema web de autogestión y visualización de datos, donde los usuarios pueden registrarse e interactuar con múltiples módulos para consultar información u operar sobre su plan de salud.

Con el fin de asegurar la compatibilidad y mantener la coherencia arquitectónica, se adoptó el lenguaje Java y el framework Spring Boot para la construcción de la lógica del backend. Esta decisión tecnológica se fundamentó en criterios estrictos de diseño de software. Por un lado, Spring Boot proporciona altos estándares de seguridad y robustez, características críticas para el procesamiento de transacciones financieras y el manejo de datos médicos sensibles. Por otro lado, su ecosistema facilita la escalabilidad horizontal y permite una integración nativa y eficiente con APIs externas.

En cuanto a la capa de presentación (frontend), se emplearon estándares web nativos como HTML, CSS y JavaScript puro. Esta combinación garantizó la construcción de una interfaz dinámica, adaptable a dispositivos móviles (responsive design) y exenta de dependencias pesadas que pudieran comprometer el rendimiento del cliente.

Finalmente, la persistencia de datos se implementó sobre PostgreSQL. La elección de este motor de base de datos relacional respondió a su probada capacidad para procesar consultas complejas (mediante el uso de múltiples uniones lógicas), lo cual resultó indispensable para cruzar de manera eficiente la información de los afiliados, sus coberturas y la conformación de los grupos familiares.

4.2. Relevamiento y análisis

La fase inicial del proyecto consistió en el relevamiento y análisis de los requerimientos funcionales del módulo de bonos. Esta etapa resultó fundamental para establecer el alcance del sistema y sentar las bases de la planificación bajo los estándares y buenas prácticas de la Ingeniería de Software.

Para llevar a cabo esta definición, se ejecutaron diversas reuniones de trabajo que involucraron al departamento de Sistemas y la gerencia general. El propósito central fue identificar las necesidades operativas del negocio, priorizar la digitalización de los distintos tipos de bonos y delimitar el alcance funcional del proyecto en función de las restricciones temporales.

4.2.1. Definición de Requerimientos

A partir de las reuniones de relevamiento, se estructuró el alcance del módulo de bonos, categorizando las necesidades operativas en requerimientos funcionales, flujos específicos y reglas de negocio críticas para la integración con el sistema central.

El requerimiento principal del proyecto consiste en incorporar al Portal de Afiliados un módulo integral que permita a los usuarios visualizar, solicitar, abonar y descargar bonos de consulta médica programada y de atención por guardia, centralizando toda la operatividad de forma digital.

Requerimientos Funcionales Generales

- **Control de acceso y Edad**: El sistema debe validar la mayoría de edad del usuario logueado (titular o adherente) para permitir el ingreso y la solicitud de bonos. En caso de que el usuario sea menor de edad, el sistema restringe su acceso al módulo. Como excepción a esta regla, el sistema debe permitir el acceso a menores de edad si figuran como titulares de su propio grupo familiar.

- **Selección de Beneficiario**: El sistema debe permitir al usuario gestionar bonos tanto para sí mismo como para cualquier integrante asociado a su grupo familiar.

- **Gestión del Ciclo de Vida y Visualización**: El sistema debe gestionar las transacciones a través de los estados lógicos de "borrador", "aprobado" y "emitido". Asimismo, el sistema debe proveer al afiliado un historial detallado para visualizar los bonos aprobados, emitidos, consumidos y de reintegro.

- **Clasificación de Bonos y Gestión de Guardia**: El sistema debe permitir la gestión de bonos para consulta programada (Consultorio, Nutricionista, Plan Materno Infantil y "Cuarto Bono") y atenciones por Guardia (Consulta, Laboratorio y Diagnóstico por Imágenes). Para el flujo de guardia, el sistema debe autocompletar a la clínica central como prestador obligatorio, y debe exigir la emisión previa de un bono de consulta (dentro de las últimas 24 horas) para habilitar la solicitud de prácticas derivadas (laboratorio o imágenes).

- **Filtro Dinámico de prestadores**: El sistema debe listar exclusivamente a los profesionales habilitados para emitir bonos de consulta, adaptando dinámicamente las prestaciones disponibles según la cobertura del afiliado. El sistema debe excluir automáticamente a

entidades institucionales (farmacias, laboratorios) y prestadores categorizados como "Alta Consulta".

- **Valorización y Ejecución de Pagos**: El sistema debe delegar el cálculo del coseguro al ERP (Odoo), tomando como parámetros de entrada el afiliado (y su cobertura), el prestador, el tipo de bono y la ubicación geográfica del afiliado. Si la valorización retorna un importe superior a cero, el sistema debe integrar la pasarela de pagos; de lo contrario, el sistema debe concretar la emisión de forma automática.

- **Vigencia y Depuración**: El sistema debe aplicar un tiempo de vigencia máximo de 48 horas a los bonos impagos o en estado "aprobado". Una vez cumplido este lapso, el sistema debe eliminar automáticamente dichos registros para prevenir futuros desfasajes en las listas de precios.

Reglas de Negocio y Validaciones (Anomalías)

- **Jerarquía de Coberturas**: El sistema debe identificar y priorizar automáticamente las coberturas especiales (como el Plan Materno Infantil) por sobre la cobertura general del plan, evaluando dinámicamente el perfil clínico del beneficiario.

- **Control de Topes**: El sistema debe implementar contadores lógicos para restringir la cantidad de prestaciones permitidas en un lapso de tiempo. Para ello, el sistema debe agrupar y contabilizar los consumos según su naturaleza: consultas regulares junto con reintegros, consultas del Plan Materno Infantil de forma aislada y prestaciones de guardia de manera independiente. Asimismo, para efectuar el cálculo, el sistema debe auditar exclusivamente los bonos que hayan alcanzado el estado "emitido" o "consumido".

- **Manejo de Anomalías (Exceso de Topes)**: El sistema debe interceptar aquellas solicitudes que excedan el límite de prestaciones permitidas por su cobertura. Ante la detección de una anomalía, el sistema debe retener el bono en estado "borrador", bloquear las acciones de pago o emisión en la interfaz y redirigir automáticamente al usuario al canal de asistencia digital (WhatsApp), inyectado los metadatos de la solicitud para su resolución administrativa.

4.3. Análisis del sistema Odoo

El desarrollo del módulo de bonos se fundamentó en un análisis técnico del sistema central de la organización, basado en la plataforma ERP (Enterprise Resource Planning) Odoo. Esta solución

de código abierto actúa como núcleo integrador de la información operativa, médica y comercial, constituyendo la fuente primaria de datos para el Portal de Afiliados.

El sistema provee la infraestructura de datos necesaria para la ejecución de los requerimientos funcionales, permitiendo la identificación de grupos familiares, la consulta de coberturas, la validación de topes y la gestión de prestadores habilitados.

Asimismo, Odoo centraliza el motor de reglas para la valorización y el registro de la emisión de comprobantes. El estudio de la arquitectura se centró en el relevamiento de las tablas relacionales, las entidades involucradas y la lógica de los ciclos de estados, permitiendo establecer las siguientes definiciones estructurales para la implementación.

4.3.1. Mapeo de Entidades y Base de Datos

El modelado y la relación de los actores del sistema de salud en la base de datos de Odoo se estructuró de la siguiente manera:

- **Afiliados y Prestadores**: Se identificó que ambas entidades coexisten bajo el modelo central de contactos (res.partner), diferenciándose mediante banderas lógicas y categorizaciones. Para la visualización de médicos habilitados en el portal, se estableció un filtro sobre el atributo de emisión de bonos, excluyendo mediante sus etiquetas de especialidad y categoría a las entidades financieras y los prestadores de "Alta Consulta".
- **Planes y Coberturas**: La información comercial se almacena en el modelo de plantillas de productos (product.template). Se categorizaron las coberturas en dependientes del plan general y coberturas independientes asignadas a perfiles específicos. Arquitectónicamente, se definió que el sistema local debe priorizar la lectura de coberturas independientes sobre las del plan general del afiliado.
- **Bonos y Prácticas Médicas**: Las transacciones se modelan mediante dos entidades clave: medical.bond (cabecera del bono) y medical.bond.practice (líneas de detalle). Estas entidades actúan como receptoras de las peticiones web y ejecutan la validación final de importes.

4.3.2. Análisis del Motor de Reglas y Valorización

La lógica de cálculo del costo de un bono (coseguro) se delega al ERP, dependiendo de la intersección de las siguientes variables:

- **Listas de Precios**: El sistema central administra múltiples listas de precios activas segmentadas por ubicación geográfica. El módulo web debe identificar la región del usuario para consultar la tarifa correspondiente.
- **Categoría del Prestador**: El valor base de la consulta médica está vinculado a la categoría del profesional (A, B, C, D). El motor de reglas cruza el valor del convenio de dicha categoría con el porcentaje de cobertura del plan para calcular el monto final.
- **Topes**: Se evaluaron los campos de restricciones paramétricas de las prácticas médicas, los cuales definen los topes máximos de consumo permitidos por lapso temporal.

4.3.3. Sincronización de Estados y Escritura en el ERP

Se definió la estructura de datos requerida para persistir transacciones en Odoo manteniendo la integridad referencial del sistema:

- **Estructura del Registro**: La petición de creación exige un bloque de cabecera con los datos demográficos y comerciales del afiliado, el prestador seleccionado, el punto de venta digital y el tipo de bono. Adicionalmente, requiere líneas de detalle para especificar la práctica médica y su respectiva valorización.
- **Integridad de Estados**: Para garantizar la consistencia transaccional y evitar registros residuales, el portal inserta inicialmente el bono en estado "Borrador" y lo transiciona a "Aprobado". El avance definitivo al estado "Emitido", junto con la asignación de su número de secuencia oficial, se condiciona a la confirmación asíncrona de la pasarela de pagos, o bien se ejecuta automáticamente si el importe de la operación es nulo.

4.4. Diseño de la Solución

Tras el relevamiento de requerimientos y el análisis de la infraestructura de datos en Odoo, se definió una arquitectura basada en el patrón Modelo-Vista-Controlador (MVC). Esta elección garantiza una separación clara de responsabilidades, factor determinante para asegurar la escalabilidad del módulo y facilitar futuras tareas de mantenimiento. La implementación se

estructuró en tres capas independientes: el Modelo para la gestión de datos y lógica de negocio, la Vista para la interfaz de usuario, y el Controlador como orquestador central de peticiones.

4.4.1. El Controlador (Orquestador de Peticiones)

Se definió un controlador centralizado como único punto de entrada para las interacciones del usuario con el módulo de bonos. Esta decisión de diseño permite centralizar la lógica de control y garantiza que cada petición sea procesada bajo criterios de seguridad uniformes. Sus responsabilidades principales incluyen:

- **Gestión del Contexto y Seguridad**: El controlador identifica el rol del afiliado (titular o adherente), para restringir el acceso y recuperar exclusivamente la información correspondiente a su grupo familiar. Esta gestión asegura que los datos entregados a la Vista sean coherentes con los permisos del usuario logueado.
- **Orquestación del Flujo Transaccional**: El componente intercepta las solicitudes de creación de nuevos bonos y delega la validación de las reglas de negocio al Modelo. Basándose en el resultado de dicha validación, el controlador dirige el flujo hacia la emisión del comprobante o hacia el manejo de excepciones, preservando la integridad del proceso en todo momento.
- **Control de Concurrencia y Asincronismo**: Con el objetivo de evitar condiciones de carrera y consumos indebidos, se estableció una validación de límites mandatoria previa a la habilitación de la pasarela de pagos. Asimismo, el controlador administra de forma asíncrona la descarga de documentos PDF y las órdenes de cancelación, optimizando la respuesta del sistema ante operaciones concurrentes.

4.4.2. El Modelo (Lógica de Negocio y Conexión a Odoo)

La capa del Modelo concentra el núcleo duro de la aplicación. Su función abarca la persistencia local de datos (PostgreSQL), la validación algorítmica de reglas de negocio y la gestión transaccional previa a la integración con Odoo. Para garantizar modularidad y eficiencia, su arquitectura se dividió en tres componentes clave:

- **Entidades y Repositorios**: Se modelaron representaciones de datos locales para gestionar los estados transicionales de los bonos. La persistencia se diseñó mediante repositorios específicos para centralizar consultas complejas, tales como la extracción de

historiales y el cálculo de disponibilidad futura, optimizando así las peticiones a la base de datos.

- **Servicios Internos (Lógica Transaccional)**: Se encapsuló la auditoría de topes y lapsos de consumo en servicios dedicados. Ante la detección de anomalías operativas (exceso de límites de cobertura), la arquitectura dicta el bloqueo estricto de la transacción y la construcción de un enrutamiento dinámico hacia canales de atención administrativa externa, garantizando un manejo de contingencias seguro.
- **Estrategia Dual de Integración externa**: Para mitigar cuellos de botella ante el alto volumen transaccional, se descartó el uso de un canal único de comunicación. En su lugar, se implementó una estrategia de conectores especializados:
    - **Integración Transaccional (XML-RPC)**: Conector orientado a operaciones de escritura crítica. Ejecuta la autenticación, mapea dinámicamente los puntos de venta y fuerza la transición de estados en el ERP (validación y emisión), asegurando la consistencia comercial.
    - **Integración Ligera (API REST)**: Canal asíncrono y ligero exclusivo para la transferencia de archivos. Al limitarse a la recepción y decodificación de documentos PDF, evita la sobrecarga del flujo transaccional principal.

4.4.3. La Vista (Interfaz de Usuario)

El diseño de la capa de presentación partió de la integración de componentes visuales preexistentes con el motor de plantillas del servidor. Las decisiones arquitectónicas en este nivel se focalizaron en optimizar la carga de datos y estructurar una interacción fluida para el usuario final:

- **Renderizado Dinámico y Adaptabilidad**: Se adoptó Thymeleaf en conjunto con las estructuras responsivas (Bootstrap) para garantizar la compatibilidad de múltiples dispositivos. La arquitectura permite al servidor inyectar variables sintéticas para construir cuadros de resumen familiar dinámicos, exponiendo de forma intuitiva reglas de negocio complejas, tales como los límites de consumo mensual o las coberturas de planes especiales.

- **Comportamiento Condicional de la Interfaz**: Se estableció que los elementos visuales debían reaccionar estrictamente al ciclo de vida de la transacción. A modo de control, los botones de interacción financiera se habilitan de forma exclusiva bajo el estado "Aprobado", mientras que la opción de descarga de documentos queda restringida al estado "Emitido".
- **Navegación Interactiva y Eficiencia (SPA)**: Para reducir las peticiones redundantes al servidor, la creación y visualización de bonos se estructuró mediante ventanas modales interactivas. Este diseño permite gestionar el flujo de datos en el cliente, simulando el comportamiento de una Single Page Application (SPA).
- **Lógica de Interacción en el Cliente**: Se proyectó un árbol de decisiones embebido en el navegador, diseñado para reaccionar a los atributos de datos del afiliado (data-attributes). En consecuencia, la interfaz muta en tiempo real para habilitar lógicas de negocio específicas; por ejemplo, revelando opciones ocultas ante un perfil con embarazo, o autoseleccionando a la clínica central y desbloqueando prácticas derivadas (laboratorio/imágenes) al detectarse un flujo de guardia.
- **Asincronismo y Flujos de Contingencia**: Se estipuló que la validación de formularios y la comunicación cliente-servidor se ejecutaran bajo un esquema estrictamente asíncrono. Esta directriz habilitó la implementación de flujos de contingencia avanzados:
    - **Manejo de Anomalías**: Se diseñó un mecanismo automático para el rechazo por exceso de topes. Ante la excepción devuelta por el servidor, la interfaz captura el error y acciona un temporizador de redirección hacia el canal de asistencia digital (WhatsApp), transfiriendo el contexto de la operación.
    - **Prevención de Condiciones de Carrera**: Para garantizar la integridad financiera, se programó una validación concurrente silenciosa contra el servidor de forma previa al redireccionamiento a la pasarela externa. Este control certifica la validez del bono en el instante exacto del cobro.

Los resultados visuales de esta implementación pueden consultarse en el Anexo A del presente informe.

4.5. Desarrollo de la solución local

Concluida la fase de diseño arquitectónico, se procedió a la construcción del software centrándose en las capas internas de la aplicación. El objetivo de esta etapa fue materializar las reglas del negocio, la persistencia de los datos y la interfaz del usuario, operando de manera aislada para estabilizar los componentes antes de su acoplamiento con el sistema ERP.

La implementación se ejecutó utilizando el framework Spring Boot para la lógica del backend y un ecosistema compuesto por HTML, CSS, JavaScript, y Thymeleaf para la capa de presentación (frontend).

4.5.1. Implementación del backend y Lógica de Negocio

La construcción de la capa lógica se estructuró en componentes especializados para garantizar el aislamiento de responsabilidades y la mantenibilidad del sistema:

- **Mapeo de Entidades y Transferencia**: Se codificaron las entidades principales para representar el modelo de dominio local y gestionar las transiciones de estado del ciclo de vida de los bonos. Asimismo, se implementaron Objetos de Transferencia de Datos (DTO) para optimizar el transporte de parámetros de cobertura y límites entre las distintas capas de aplicación.
- **Persistencia de Datos**: El acceso a la base de datos se resolvió mediante la implementación de repositorios personalizados. Se construyeron consultas relacionales complejas (mediante lenguaje de consulta sobre objetos) para extraer historiales precisos y calcular la disponibilidad prestacional dentro de ventanas temporales específicas, optimizando así el rendimiento de lectura.
- **Servicios y Reglas Dinámicas**: El núcleo algorítmico se centralizó en una capa de servicios encargada de procesar las validaciones de topes mensuales. En este nivel se programó la lógica condicional que altera los márgenes de control basándose en el estado clínico del afiliado (por ejemplo, excepciones ante perfiles con embarazo activo) y se automatizó la generación de enlaces para la derivación de contingencias.
- **Controladores Web**: Se configuraron los puntos de enlace (endpoints) para orquestar la recepción de peticiones web, garantizando que toda solicitud atraviese el motor de validación interno de manera estricta y previa a cualquier intento de integración externa.

4.5.2. Implementación del Frontend y Experiencia de Usuario

La capa de presentación se desarrolló priorizando la interactividad y la adaptación dinámica a las reglas de negocio, delegando carga de procesamiento al cliente para agilizar la respuesta visual:

- **Renderizado del Servidor**: Se utilizó el motor de plantillas del servidor para generar cuadros de resumen familiar interactivos. La lista evalúa el estado clínico de los miembros del grupo e imprime condicionalmente las métricas de consumo disponibles, traduciendo la lógica interna a una interfaz de usuario intuitiva.
- **Manipulación Condicional de la Interfaz**: Mediante la captura de eventos en el navegador, se programó un árbol de decisiones que reacciona a los atributos de datos del afiliado. Esta lógica permite habilitar flujos específicos en tiempo real, como la selección obligatoria de la clínica central y el desbloqueo de prácticas derivadas ante un escenario de atención por guardia.
- **Optimización de Búsquedas**: Para el filtrado de prestadores médicos, se integraron componentes de búsqueda avanzada acoplados a la API local. Se desarrolló un interceptor de resultados (matcher) personalizado que flexibiliza los criterios de coincidencia, mitigando errores humanos y mejorando la tolerancia a búsquedas desordenadas.
- **Gestión Asíncrona y Manejo de Excepciones**: El envío de formularios se encapsuló utilizando peticiones asíncronas, asegurando la inyección de los protocolos de seguridad pertinentes (tokens CSRF). Paralelamente, se implementó un mecanismo de intercepción de respuestas HTTP diseñado para capturar rechazos por violaciones a las reglas de negocio, accionando de forma automática una alerta visual y la posterior redirección del usuario al canal de asistencia.

4.6. Integración de Sistemas Externos

Estabilizada la lógica de negocio local y la interfaz de usuario, se procedió a la vinculación del módulo con los sistemas externos críticos. Esta integración resultó fundamental para transformar las peticiones aisladas en transacciones efectivas, requiriendo el establecimiento de canales de comunicación seguros y de alta disponibilidad con el sistema ERP central (Odoo) y la pasarela de pagos externa (Mercado Pago).

4.6.1. Integración Bidireccional con el Sistema Central (Odoo)

Debido a la criticidad y el volumen de reglas transaccionales del dominio médico, la comunicación con el ERP se orquestó mediante una estrategia dual, separando las responsabilidades de escritura y lectura documental:

- **Conector Transaccional (Escritura y Estados)**: Se implementó un servicio dedicado a operaciones críticas mediante el protocolo XML-RPC. Este componente gestiona la autenticación y determina dinámicamente el entorno del destino (compañías y puntos de venta) en función del contexto geográfico del afiliado. Su función principal es insertar los registros y forzar el avance del comprobante a través de la máquina de estados nativa del ERP, asegurando el impacto comercial y contable.
- **Conector Documental (Lectura asíncrona)**: Con el objetivo de evitar la saturación del canal transaccional principal debido a la transferencia de archivos pesados, se estableció un servicio paralelo basado en tecnología API REST. La responsabilidad de este canal se limita a la solicitud, decodificación y empaquetado de los comprobantes PDF para su posterior descarga.
- **Mantenimiento automatizado**: Se programó una rutina de ejecución periódica en el servidor (background task), encargada de auditar y anular sistemáticamente aquellos registros en el sistema central que hubiesen excedido su tiempo máximo de vigencia sin confirmación de pago, garantizando la consistencia de la base de datos externa.

4.6.2. Integración Financiera (Mercado Pago)

La gestión de cobro de coseguros exigió la implementación del SDK oficial de la pasarela financiera, estableciendo un flujo de pago validado bajo los siguientes lineamientos:

- **Configuración y Entornos**: Se habilitó la integración tecnológica estableciendo los parámetros de seguridad y configurando credenciales aisladas para los entornos de prueba y producción.
- **Orquestación de Pagos**: Ante operaciones con valorización positiva por parte del ERP, el sistema local construye dinámicamente una preferencia de cobro, la cual encapsula el detalle financiero de la práctica médica y los datos del titular.
- **Validación Concurrente (Pre-Checkout)**: Como medida de seguridad crítica, previo a la redirección del usuario hacia la plataforma de pagos, se ejecuta una validación

transaccional concurrente. Este control mitiga vulnerabilidades al certificar en milisegundos que los topes de consumo del afiliado no hayan sido alterados desde otras instancias.

- **Sincronización Asíncrona (Webhooks)**: Se configuraron puertos de escucha para interceptar las notificaciones automáticas de la pasarela. Al recibir la confirmación de la acreditación de los fondos, el portal orquesta la transición definitiva del comprobante hacia el estado de emisión, habilitando su disponibilidad para el usuario.

4.7. Pruebas y Validación Funcional

En concordancia con las metodologías ágiles adoptadas, el proceso de aseguramiento de calidad (QA) se ejecutó de forma iterativa y continua durante todo el ciclo de desarrollo. Las validaciones se llevaron a cabo en entornos aislados no productivos (máquinas virtuales, bases de datos de prueba y sandbox financieros), garantizando la integridad de los sistemas operativos de la organización.

4.7.1. Pruebas de Lógica Local e Interfaz

El primer ciclo de validación se enfocó en certificar la estabilidad de las reglas de negocio internas y el comportamiento de la interfaz de usuario. A partir de estas iteraciones, se detectaron y resolvieron diversas desviaciones funcionales:

- **Gestión de Permisos y Topes**: Se ajustó la lógica de autorización para garantizar que los usuarios adherentes posean los privilegios necesarios para operar sobre todo su grupo familiar. Asimismo, se refactorizaron los algoritmos de auditoría de consumo para que contabilicen estrictamente las transacciones consolidados, evitando falsos positivos por bonos en estado borrador.
- **Optimización de Persistencia**: Se depuraron las consultas relacionales a la base de datos para excluir perfiles profesionales no habilitados (como entidades bioquímicas) y para reflejar con precisión el historial de movimientos, omitiendo registros expirados.
- **Estabilidad de la interfaz**: Se corrigieron estados transicionales de la vista, garantizando el reseteo de la memoria de los formularios al cancelar operaciones y refinando el motor de búsqueda de profesionales para tolerar consultas asíncronas y desordenadas, así como la optimización de la lógica condicional de los botones de pago y cancelación.

4.7.2. Pruebas de Integración y Validación End-to-End

Durante la fase final, se ejecutó un plan de pruebas integrales (End-to-End) para certificar la interoperabilidad entre el Portal de Afiliados, el entorno de pruebas del ERP (Test Odoo) y la pasarela de pagos. Operando con credenciales de prueba y simulando fallas de red, se consolidó la estabilidad del sistema mediante los siguientes ajustes críticos:

- **Sincronización de Lógicas Complejas**: Se detectaron desfasajes en la valorización de perfiles con lógicas paramétricas dinámicas (como coberturas por embarazo o recién nacidos). Esto demandó una refactorización matemática en el modelo local para alinear el cálculo de los lapsos temporales con las exigencias del ERP, logrando una sincronización de datos perfecta.
- **Manejo de Excepciones Externas**: Se identificó que las inconsistencias de afiliación en Odoo provocaban caídas no controladas en el flujo local. Para mitigarlo, se implementaron manejadores de excepciones (Error Handlers) que interceptan los códigos de error del ERP y los traducen en alertas informativas para el usuario, preservando la continuidad del sistema.
- **Integridad Transaccional y Financiera**: Los simulacros de extremo a extremo certificaron la resiliencia de la arquitectura. Se validó que el sistema bloquea eficazmente intentos de pago ante condiciones de carrera, asigna con precisión los puntos de venta geográficos y garantiza que el impacto contable en la pasarela se refleje de manera idéntica en el sistema central de la organización.
- **Refinamiento de la Interfaz y Controles de Validación**: Durante los simulacros de uso se identificaron vulnerabilidades en la validación dinámica de los formularios, específicamente al alternar entre flujos de atención por guardia y consultas regulares. Para solucionarlo, se ajustaron los algoritmos del cliente para forzar re-validaciones estrictas ante cambios de estado, bloqueando el envío de peticiones incompletas. Asimismo, se aplicaron mejoras integrales en la experiencia de usuario (UX), corrigiendo la adaptabilidad en dispositivos móviles, optimizando la paginación y actualizando las consultas a la base de datos para excluir registros que no debían auditarse en el portal.

4.8. Documentación y manuales de uso

Durante el ciclo de vida del desarrollo de software, se llevó a cabo la estructuración de la documentación técnica, funcional y operativa del proyecto. El objetivo de esta etapa consistió en asegurar la mantenibilidad y escalabilidad del módulo por parte del departamento de Sistemas, así como facilitar la adopción tecnológica por parte de los usuarios finales.

El acervo documental del proyecto se estructuró en los siguientes ejes principales:

- **Documento Funcional y de Diseño Arquitectónico**: Archivo maestro que asienta las bases de ingeniería del proyecto. Contiene el relevamiento inicial, la definición de los requerimientos funcionales, las reglas de negocio, el mapeo de las entidades del ERP y el diseño de la implementación basado en el patrón MVC, sirviendo como guía de referencia para el equipo de desarrollo.
- **Registro de Pruebas y Validación (Testing)**: Documento de trazabilidad de calidad (QA) donde se asientan las iteraciones de prueba. Registra los casos de uso evaluados, las anomalías detectadas y las resoluciones algorítmicas aplicadas, constituyendo el historial de estabilización del sistema.
- **Manuales de Usuario Final**: Instructivos y guías visuales orientadas a los afiliados de La Pequeña Familia. Detallan la navegación del módulo, la gestión de autorizaciones del grupo familiar, el ciclo de vida de las transacciones y los procedimientos estandarizados para efectuar pagos y descargas.

Nota sobre la confidencialidad de la información: Si bien se incluyen capturas de la interfaz gráfica en el **Anexo A** (utilizando datos de prueba), la documentación técnica profunda, los diagramas de arquitectura detallados y el código fuente completo se encuentran resguardados en los repositorios privados de la organización por razones de seguridad y confidencialidad corporativa.

4.9. Reflexión personal, aprendizajes y errores.

El desarrollo e implementación de este módulo representó una oportunidad invaluable para consolidar y llevar a la práctica los conocimientos académicos asimilados durante la carrera universitaria. La formación provista por la universidad me otorgó las bases teóricas y metodológicas necesarias para abordar un proyecto de esta magnitud, brindándome las herramientas fundamentales en áreas críticas como el ciclo de vida del software, el diseño de

bases de datos relacionales, la programación estructurada y orientada a objetos, el análisis de sistemas de información y la aplicación de patrones de diseño arquitectónico.

Sin embargo, la inmersión en un entorno corporativo real exigió una pronunciada curva de aprendizaje y adaptación. Más allá de la base teórica, el proyecto demandó el desarrollo de nuevas competencias prácticas, tales como la integración de código sobre una arquitectura de software preexistente, el manejo de entornos separados (desarrollo, testing y producción) y la interacción con bases de datos pobladas y en constante operatividad; escenarios que trascienden el ámbito académico. Asimismo, el trabajo requirió una profunda investigación técnica para asimilar el funcionamiento de un sistema ERP de la magnitud de Odoo, comprender cómo orquestar la conexión de múltiples APIs externas en Spring Boot, y dominar librerías específicas para cumplir con los requisitos funcionales exigidos por la gerencia.

Finalmente, el proceso de desarrollo no estuvo exento de obstáculos y errores técnicos, propios de una primera experiencia profesional en este sector. No obstante, estos desafíos funcionaron como el motor principal de la mejora continua del módulo. Gracias a la investigación proactiva, el esfuerzo iterativo y, fundamentalmente, al apoyo y la guía constante del equipo de Sistemas de La Pequeña Familia, cada anomalía pudo ser refactorizada y solucionada. Esta sinergia no solo permitió entregar un producto de software robusto, documentado y funcional, sino que forjó una experiencia de aprendizaje invaluable para mi consolidación como futuro profesional de la ingeniería.

5. Conclusiones

A partir de las actividades ejecutadas, se concluye que el objetivo general del proyecto fue alcanzado. Se implementó el módulo de gestión de bonos en el Portal de Afiliados, lo que permitió la digitalización y centralización del proceso de venta. Mediante esta solución tecnológica, se redujo el volumen de tareas administrativas, se mitigaron las demoras y se eliminó el requerimiento de traslado presencial de los usuarios. Como resultado, se agilizó el acceso a las visitas médicas y se resolvieron las problemáticas operativas identificadas al inicio del informe.

Asimismo, se dio cumplimiento íntegro a los objetivos específicos propuestos para esta práctica profesional. Se relevaron los requerimientos funcionales y se analizaron con éxito las estructuras

y reglas de negocio del sistema Odoo. Sobre esta base, se diseñó la arquitectura del módulo y se desarrolló la lógica de negocio junto con la interfaz de usuario.

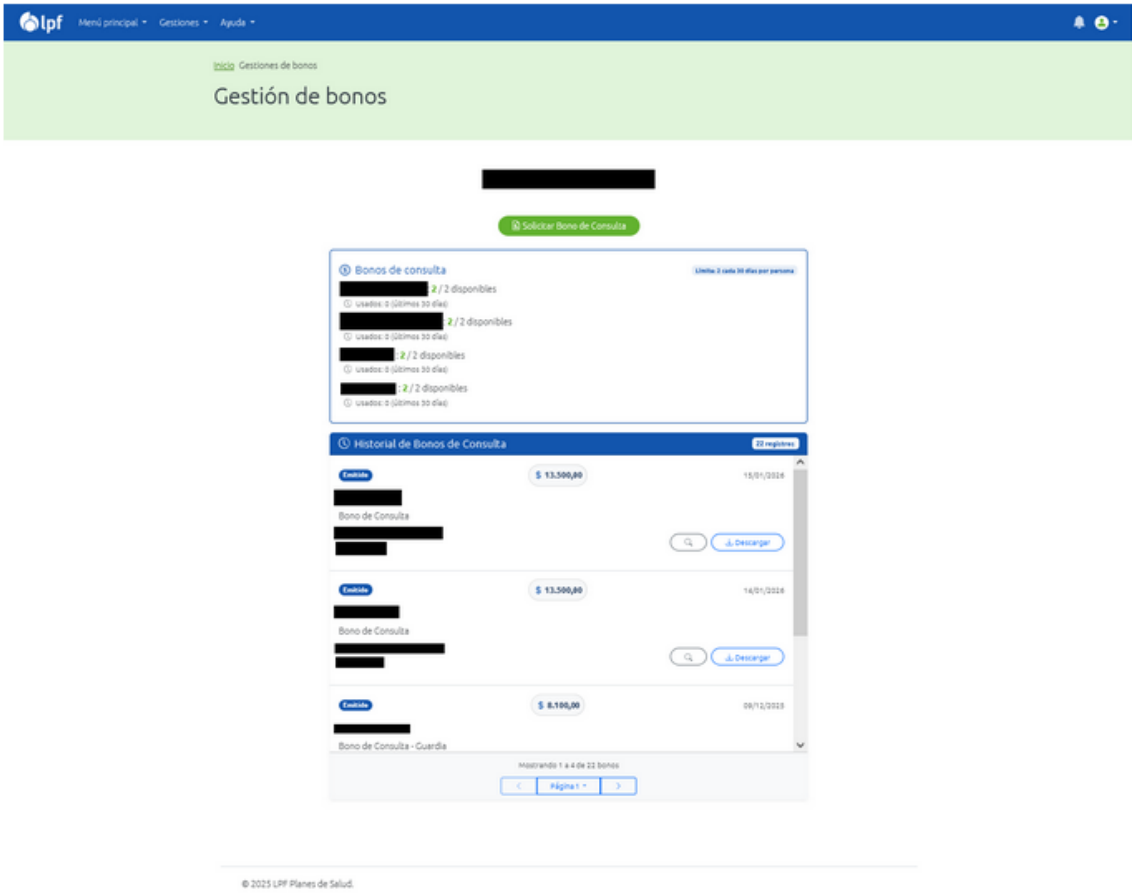
La integración técnica se materializó a través de la conexión con las APIs de Odoo y la pasarela de pagos de Mercado Pago, asegurando la sincronización de datos. Por último, la aplicación de pruebas iterativas validó la estabilidad del software, el cual dispone de su documentación técnica y manuales de uso.

6. Bibliografía

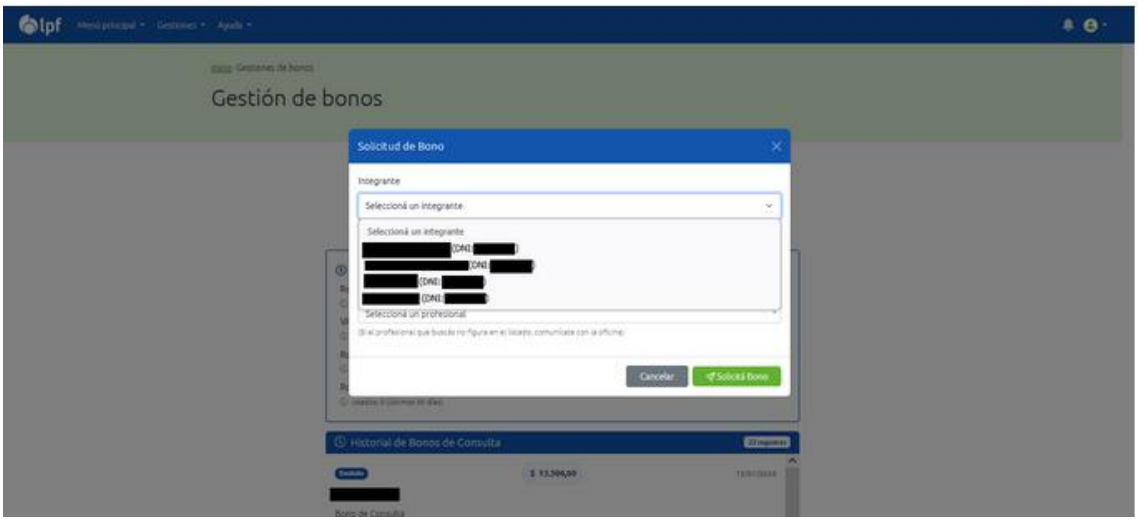
[1] Spring, "Spring Boot Reference Documentation", Spring Framework, [En línea]. Disponible en: https://docs.spring.io/spring-boot/
[2] Mercado Pago, "Documentación para Desarrolladores - Checkout Pro y SDK para Java", Mercado Pago Developers, [En línea]. Disponible en: https://www.mercadopago.com.ar/developers/es/docs
[3] Odoo S.A., "Odoo External API - XML-RPC", Odoo Documentation, [En línea]. Disponible en: https://www.odoo.com/documentation/14.0/developer/reference/external_api.html
[4] The Thymeleaf Team, "Tutorial: Using Thymeleaf", Thymeleaf, [En línea]. Disponible en: https://www.thymeleaf.org/doc/tutorials/3.1/usingthymeleaf.html
[5] PostgreSQL Global Development Group, "PostgreSQL Official Documentation", PostgreSQL, [En línea]. Disponible en: https://www.postgresql.org/docs/
[6] Select2, "Select2 Documentation - The jQuery replacement for select boxes", Select2, [En línea]. Disponible en: https://select2.org/
[7] I. Sommerville, Ingeniería de Software, 9na ed. México: Pearson Educación, 2011.

7. Anexos

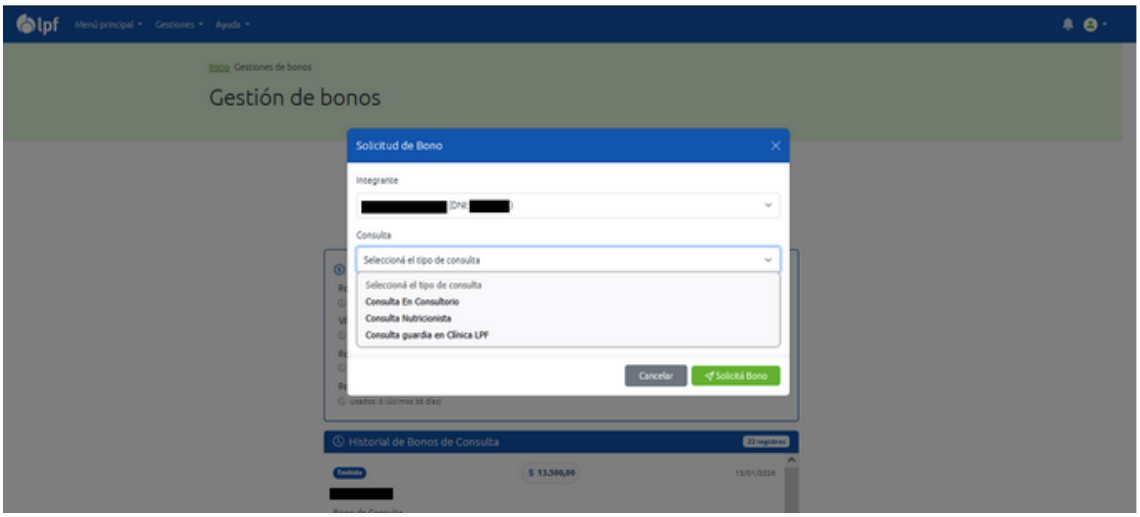
Anexo A: Interfaz Gráfica del Módulo de Bonos
A continuación, se presentan capturas de pantalla obtenidas del entorno de pruebas (Sandbox), utilizando datos tapados para demostrar la funcionalidad y el diseño del módulo desarrollado:



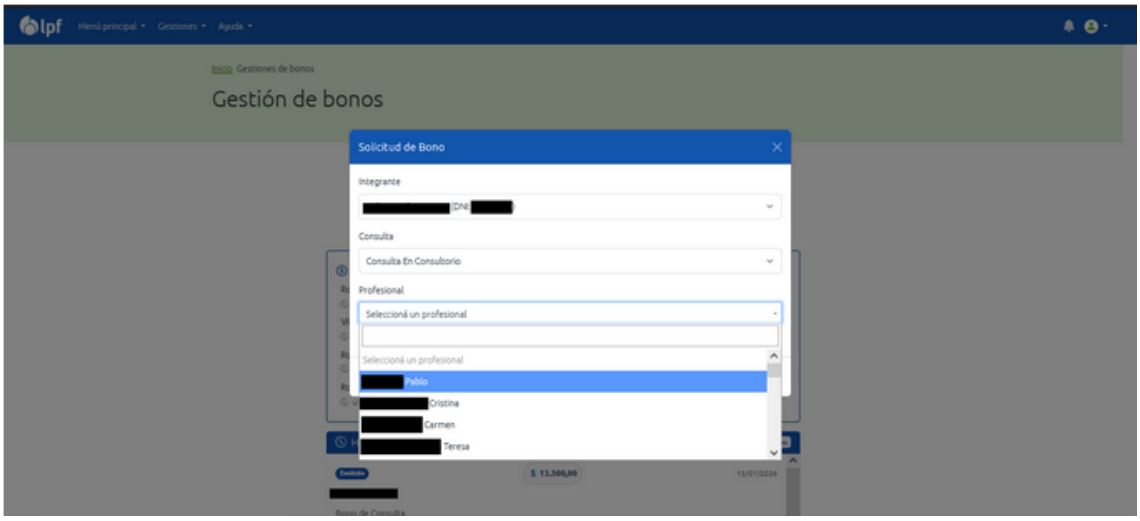
- *Figura 1.* **Gestión de bonos:** Visualización de la página principal con el historial de bonos y los botones.



- *Figura 2.* **Modal de Solicitud de Bonos:** Interfaz dinámica que permite la selección del beneficiario.



- *Figura 3.* **Modal de Solicitud de Bonos:** Interfaz dinámica que permite la selección del tipo de prestación.



- *Figura 4.* **Buscador de Prestadores:** Demostración del filtrado avanzado y la integración con la librería Select2.

8. Agradecimientos

En primer lugar, quiero agradecer a mi madre, María Belén Lavatti, padre, Mario Oscar Koroluk, novia, Valentina Lanzillotta y hermanos, Mirko Adrián Koroluk y Milena Alexandra Koroluk. Gracias por acompañarme a lo largo de mi carrera y ayudarme a poder llevar a cabo la realización de la misma de una manera más sencilla. Agradecerles por tanto apoyo, amor y comprensión durante estos cinco años.

Extiendo mi gratitud a los profesores de la UNNOBA, por impartirme el conocimiento necesario para poder llegar a ser el profesional en el que me he convertido y por darme las herramientas necesarias para seguir creciendo en el futuro. Cabe mencionar también al personal no docente y personal administrativo, quienes han permitido que pueda cursar mis estudios en perfectas condiciones.

Asimismo, agradezco a mis compañeros de la Universidad, desde el primer hasta el quinto año, con quienes he compartido trabajos, parciales y momentos de estudio, que han hecho más llevadera, enriquecedora y divertida mi experiencia académica.

A su vez, quiero agradecer a La Pequeña Familia por brindarme la oportunidad de realizar una pasantía y hacer mis prácticas profesionales en la empresa, una etapa fundamental para recibirme. Dentro de la organización, extiendo mi agradecimiento a todas las personas con quienes me he relacionado, en especial a mi tutor de la pasantía y jefe, Roberto García, quien no solo me ayudó en el desarrollo de mis tareas y en mi primera experiencia laboral, sino que también se convirtió en un gran amigo. Además, mencionar a un compañero de la facultad, Germán Torres, con quien ingresamos a la vez a la empresa y el habernos ayudado mutuamente hizo que este primer trabajo fuera mucho más fácil y ameno.

De igual manera, quiero agradecer a mi tutora del informe, Sabrina Pompei, por su valiosa ayuda y orientación con mis numerosas preguntas y dudas respecto al informe y los anexos necesarios.

Por último, quiero agradecer a aquel que esté leyendo esto, por tomarse el tiempo de leer este informe que tanto significa tanto para mí.

MORAN Sebastian
Administrativo
14/04/2026 11:02
hora estándar de Argentina