



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Análisis, diseño e implementación de una aplicación de
gestión de anticipos de nómina con énfasis en UX/IU

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Gómez Pérez, Víctor

Tutor/a: Carsí Cubel, José Ángel

CURSO ACADÉMICO: 2024/2025

Resumen

Este Trabajo de Fin de Grado tiene como objetivo el desarrollo de una nueva aplicación web para la gestión de Anticipos de Nómina del Cabildo de Gran Canaria. La idea surge de la necesidad de sustituir la aplicación actual, desarrollada con Oracle Forms y totalmente dependiente del obsoleto navegador Internet Explorer, por una solución moderna, accesible y alineada con los estándares tecnológicos actuales.

A lo largo del proyecto se definirá una arquitectura actualizada, se implementará una aplicación basada en tecnologías como Java Spring Boot, Maven, SQL Developer y el servidor de aplicaciones WildFly y se implementará un cambio de diseño que servirá como base de rediseño para el resto de aplicaciones del Cabildo de Gran Canaria. El desarrollo seguirá una metodología ágil, con herramientas colaborativas como JIRA, Microsoft Teams y Redmine, y el diseño de la interfaz se realizará con Figma y Bootstrap, priorizando la experiencia de usuario.

El resultado esperado es una aplicación funcional y escalable, capaz de gestionar las solicitudes de anticipos de forma eficiente, integrarse con el Sistema de Recursos Humanos del Cabildo y facilitar su despliegue en un entorno de preproducción para su validación por parte del cliente.

Palabras clave: Análisis, diseño, implementación, nóminas, UX/UI.

Abstract

This Final Degree Project is oriented towards the development of a new web application for the management of payroll advances within the Cabildo of Gran Canaria. The initiative arises from the necessity of replacing the current system, developed with Oracle Forms and entirely dependent on the obsolete Internet Explorer browser, with a modern, accessible solution aligned with contemporary technological standards.

Throughout the course of the project, an updated architecture will be defined, and an application will be implemented employing technologies such as Java Spring Boot, Maven, SQL Developer, and the WildFly application server. Furthermore, a design renewal will be undertaken, which will serve as a reference framework for the subsequent redesign of the remaining applications used by the Cabildo of Gran Canaria. The development process will adhere to an agile methodology, making use of collaborative tools such as JIRA, Microsoft Teams, and Redmine. The interface design will be carried out with Figma and Bootstrap, placing particular emphasis on enhancing the user experience.

The expected outcome is a functional and scalable application, capable of efficiently managing advance requests, ensuring seamless integration with the Human Resources System of the Cabildo, and facilitating deployment within a pre-production environment, thereby enabling thorough validation by the client.

Keywords: Analysis, design, implementation, payroll, UX/UI.

Índice

1 Introducción.....	9
1.1 Motivación.....	9
1.2 Objetivos.....	10
1.3 Metodología.....	11
1.3.1 Equipo multidisciplinar.....	11
1.3.2 Gestión de tareas mediante JIRA y asignación por especialidad.....	11
1.3.3 Reuniones de seguimiento semanales.....	12
1.3.4 Sprints de cuatro semanas.....	12
1.3.5 Entornos de preproducción y validación del cliente.....	12
1.4 Planificación.....	13
1.4.1 Análisis de la aplicación actual.....	13
1.4.2 Análisis de requisitos de la nueva aplicación.....	13
1.4.3 Diseño de la interfaz de usuario.....	13
1.4.4 Desarrollo del backend.....	13
1.4.5 Desarrollo del frontend.....	13
1.4.6 Integración.....	13
2 Estado del arte.....	14
2.1 Estado de la aplicación actual.....	14
2.2 UX/IU.....	14
2.3 Análisis con propuestas de mejora para la nueva aplicación.....	16
2.4 Análisis de la antigua aplicación.....	18
2.4.1 Flujo de una solicitud.....	18
2.4.2 Modelo lógico.....	19
2.4.3 Vistas de la antigua aplicación.....	20
3 Requisitos.....	26
3.1 Características exigidas en la nueva aplicación.....	26
3.1.1 Migración a nuevas tecnologías.....	26
3.1.2 Separación de ventanas según funcionalidad.....	26
3.1.3 Diseño más claro e intuitivo.....	26
3.1.4 Validación con mensajes de error.....	26
3.1.5 Ventanas de la aplicación.....	26
3.2 Requisitos funcionales.....	42
3.2.1 REQ-001 Roles de Usuario.....	42
3.2.2 REQ-002 Partes de la aplicación.....	42
3.2.3 REQ-003 Mantenimiento Límite de Importes y Plazos.....	42
3.2.4 REQ-004 Mantenimiento Partidas.....	42
3.2.5 REQ-005 Tramitación Solicitudes Consulta de empleados.....	43
3.2.6 REQ-006 Tramitación Solicitudes Validación.....	43
3.2.7 REQ-007 Tramitación Solicitudes con Urgencia.....	43
3.2.8 REQ-008 Gestión de Resolución por Concesión.....	43
3.2.9 REQ-009 Gestión de Anulaciones por Desestimación.....	43

3.2.10 REQ-010 Gestión de Anulaciones por Desistimiento.....	44
3.2.11 REQ-011 Tramitación de Pagos.....	44
3.2.12 REQ-012 Tramitación Gestión de Anticipos.....	44
3.2.13 REQ-013 Consulta de Solicitudes.....	44
3.2.14 REQ-014 Consulta de Anticipos.....	45
3.2.15 REQ-015 Generación de documentos.....	46
3.2.16 REQ-016 Plazos de amortización.....	48
3.3 Requisitos no funcionales.....	49
3.3.1 RNF-01-Rendimiento.....	49
3.3.2 RNF-02-Seguridad.....	49
3.3.3 RNF-03-Compatibilidad.....	49
3.3.4 RNF-04-Usabilidad.....	49
3.3.5 RNF-05-Usabilidad.....	49
3.3.6 RNF-06-Mantenibilidad.....	49
3.3.7 RNF-07-Despliegue y disponibilidad.....	49
3.4 Análisis de requisitos y diagramas UML (REPASAR).....	50
3.4.1 Diagrama de Casos de Uso.....	50
3.4.2 Diagrama de Clases de Análisis.....	51
3.4.3 Diagrama de Secuencia.....	52
4 Diseño.....	54
4.1 Diseño de la nueva IGU.....	54
4.1.2 Pantalla tabla.....	55
4.1.3 Pantalla detalle y edición.....	57
4.1.4 Menú navegable.....	59
4.1.5 Alertas y confirmaciones.....	60
4.1.6 Empty space.....	61
4.2 Arquitectura.....	62
4.2.1 Arquitectura actual: Oracle Forms.....	62
4.2.2 Nueva arquitectura: Aplicación multicapa.....	63
4.3 Tecnologías a utilizar.....	64
4.3.1 Java Spring Boot.....	64
4.3.2 JPA.....	64
4.3.3 Maven.....	64
4.3.4 Primefaces.....	64
4.3.5 WildFly.....	64
4.3.6 SQL Developer.....	64
4.3.7 JIRA.....	64
4.3.8 Figma.....	64
4.3.9 Redmine.....	65
4.3.10 Trello.....	65
4.3.11 Teams.....	65
4.3.12 IntelliJ IDEA.....	65
5 Desarrollo.....	66
5.1 Sprint 1.....	66

5.1.1 Preparación y bases del proyecto.....	66
5.1.2 Backend inicial y reglas básicas.....	67
5.2 Sprint 2.....	70
5.2.1 Prueba de conexión entre frontend y backend.....	70
5.2.2 Header y footer.....	70
5.2.3 Sistema de autenticación.....	71
5.3 Sprint 3.....	73
5.3.1 Primeras vistas.....	73
5.4 Sprint 4.....	75
5.4.1 Vistas de mantenimientos.....	75
5.4.2 Flujo de solicitudes.....	75
5.5 Sprint 5.....	76
5.5.1 Última parte del flujo de solicitudes.....	76
5.6 Sprint 6.....	78
5.6.1 Generación de documentos.....	78
5.6.2 Subida a preproducción y feedback.....	78
5.6.3 Ajustes finales.....	79
6 Pruebas.....	81
6.1 Pruebas unitarias.....	81
6.2 Pruebas de integración.....	81
6.3 Pruebas de validación.....	81
6.4 Pruebas automáticas.....	81
6.5 Entorno de preproducción.....	82
7 Conclusión y trabajos futuros.....	83
8 Bibliografía.....	84
9 Apéndices.....	85
9.1 Apéndice I. Objetivos de Desarrollo Sostenible.....	85
9.1.1 ODS 8. Trabajo decente y crecimiento económico.....	87
9.1.2 ODS 9. Industria, innovación e infraestructuras.....	87
9.1.3 ODS 16. Paz, justicia e instituciones sólidas.....	87
9.1.4 ODS 17. Alianzas para lograr objetivos.....	87
10 Glosario de términos y acrónimos.....	88

Índice de figuras

Imagen 1. Mantenimiento de ejemplo del aplicativo anterior	15
Imagen 2. Diseño propuesto para un mantenimiento de ejemplo	15
Imagen 3. Flujo de la solicitud	18
Imagen 3. Modelo lógico	19
Imagen 4. Vista principal de la antigua aplicación	20
Imagen 4. Vista del listado de importes y plazos de la antigua aplicación	20
Imagen 5. Vista del listado de partidas presupuestarias de la antigua aplicación	21
Imagen 6. Vista del listado de solicitudes de la antigua aplicación	21
Imagen 7. Vista del listado de solicitudes concedidas de la antigua aplicación	22
Imagen 8. Vista del listado de solicitudes a conceder de la antigua aplicación	22
Imagen 9. Vista del alta de desestimación de la antigua aplicación	23
Imagen 10. Vista del alta de desistimiento de la antigua aplicación	23
Imagen 11. Vista de la tramitación del pago de la antigua aplicación	24
Imagen 12. Vista de la tramitación del anticipo de la antigua aplicación	24
Imagen 13. Vista de la consulta de solicitudes de la antigua aplicación	25
Imagen 14. Vista de la consulta de anticipos de la antigua aplicación	25
Imagen 15. Vista del diseño propuesto del listado de importes	27
Imagen 16. Vista del diseño propuesto del alta de importes	27
Imagen 17. Vista del diseño propuesto del listado de plazos	28
Imagen 18. Vista del diseño propuesto del alta de plazos	28
Imagen 19. Vista del diseño propuesto del listado de partidas	29
Imagen 20. Vista del diseño propuesto del alta de partidas	30
Imagen 21. Vista del diseño propuesto del listado de solicitudes	31
Imagen 22. Vista del diseño propuesto del alta de solicitudes	32
Imagen 23. Vista del diseño propuesto del listado de concesiones	33
Imagen 24. Vista del diseño propuesto del alta de concesiones	34
Imagen 25. Vista del diseño propuesto del listado de desestimaciones	35
Imagen 26. Vista del diseño propuesto del alta de desestimaciones	36
Imagen 27. Vista del diseño propuesto del listado de desistimientos	37
Imagen 28. Vista del diseño propuesto del alta de desistimientos	38
Imagen 29. Vista del diseño propuesto del listado de pagos	39
Imagen 30. Vista del diseño propuesto del alta de pagos	40
Imagen 31. Vista del diseño propuesto del listado de anticipos	41
Imagen 32. Diagrama de Casos de Uso de la aplicación	51
Imagen 33. Diagrama de Clases de Análisis de la aplicación	52
Imagen 34. Diagrama de Secuencia simplificado del flujo de solicitudes	53
Imagen 35. Vista del diseño de la cabecera propuesta	54
Imagen 36. Vista del diseño de la pantalla tabla propuesta	55
Imagen 37. Vista del diseño de la pantalla detalles en modo vista propuesta	57
Imagen 38. Vista del diseño de la pantalla detalles en modo edición propuesta	58

Imagen 39. Vista del diseño del menú navegable propuesto	59
Imagen 40. Vista del diseño de las ventanas de alertas propuestas	60
Imagen 41. Vista del diseño del empty space de las tablas propuesto	61
Imagen 42. Archivo SQL de importación inicial	66
Imagen 43. Entidad de anticipo	67
Imagen 44. Servicio de anticipo	67
Imagen 45. Mapper de anticipo	68
Imagen 46. DTO de anticipo	68
Imagen 47. Repositorio de anticipo	69
Imagen 48. Anotación Autowired y como se utiliza	70
Imagen 49. Aplicación con el arquetipo inicial	71
Imagen 50. Pantalla de inicio de sesión del CAS	71
Imagen 51. Método que gestiona los permisos.	72
Imagen 52. Método que crea el LazyDataModel	73
Imagen 53. Vista de listado de partidas	74
Imagen 54. Vista de alta de partidas	74
Imagen 55. Métodos que gestionan los valores de importe/plaza	75
Imagen 56. Vista alta solicitud	75
Imagen 57. Vista listado de concesiones	76
Imagen 58. Vista de alta de desestimación	77
Imagen 59. Vista del alta de pago de anticipo	77
Imagen 60. Vista del listado anticipos	77
Imagen 61. Botón generar excel	78
Imagen 62. WinSCP	79
Imagen 63. Filtro extra	79
Imagen 64. Alta de concesiones modificado	80
Imagen 65. Vista del listado de usuarios	80

Índice de tablas

Tabla 1. Comparación de las tecnologías actuales y las propuestas	17
Tabla 2. Plazos de amortización de los anticipos a conceder	48
Tabla 3. Relación del proyecto con las 17 ODS actuales	86

1 Introducción

En las últimas décadas, las Tecnologías de la Información y las Comunicaciones (TIC) han transformado innegablemente el mundo laboral en prácticamente todos los ámbitos. Actualmente, resulta difícil encontrar un entorno de trabajo en el que no se utilicen herramientas tecnológicas para mejorar la eficiencia, optimizar procesos o reducir tiempos.

Esta digitalización creciente ha incrementado también el nivel de exigencia en el desarrollo de software. Ya no basta con que una aplicación funcione correctamente; se espera que además tenga un diseño intuitivo, ofrezca un rendimiento fluido y esté alineada con los estándares tecnológicos actuales (Nielsen, 2012). Este contexto ha dado lugar a una alta demanda de profesionales en el ámbito del desarrollo y mantenimiento de software.

En las administraciones públicas, el proceso de modernización tecnológica suele enfrentarse a ciertas resistencias debido a procedimientos administrativos complejos. No obstante, organismos como el Cabildo de Gran Canaria¹ han empezado a destinar recursos a la actualización de sus aplicaciones, con el objetivo de mejorar la experiencia de los usuarios y adaptarse a los nuevos requerimientos técnicos y normativos (Cabildo de Gran Canaria, 2024).

Para facilitar la lectura y comprensión de este documento, al final del trabajo se incluye un glosario con la definición de términos técnicos y acrónimos utilizados a lo largo del texto.

1.1 Motivación

En Läberit², contamos con un contrato con el Cabildo de Gran Canaria para la creación y el mantenimiento de sus aplicaciones informáticas. Entre ellas se encuentra una herramienta destinada a la gestión de Anticipos de Nómina, desarrollada con Oracle Forms y ejecutada sobre Oracle IAS (Oracle Corporation, 2019). Esta aplicación sólo es accesible desde el navegador Internet Explorer, un entorno tecnológicamente obsoleto y limitado en cuanto a compatibilidad con los estándares web actuales.

Ante esta situación, se plantea el desarrollo de una nueva versión de la aplicación, utilizando tecnologías modernas como Java 11 y Spring Boot³ (Spring Data Team, 2022), que permitan mejorar tanto la experiencia de usuario como la mantenibilidad del sistema.

Este proyecto supone también una oportunidad personal y profesional para aplicar conocimientos obtenidos en la carrera y profundizar en tecnologías ampliamente utilizadas en el entorno empresarial como Java Spring Boot, SQL, WildFly (Red Hat, 2021). A su vez ampliaré el conocimiento sobre metodologías ágiles ya que se trabajará con herramientas como JIRA (Atlassian, 2023) para el control y gestión de los tickets recibidos por parte del Cabildo de Gran Canaria. A través del desarrollo real de una aplicación para un cliente institucional, se amplían conocimientos prácticos que complementan la formación recibida durante la carrera y se adquiere una visión más completa del flujo de trabajo de una aplicación en un entorno laboral y profesional.

¹ <https://www.grancanaria.com/>

² Empresa en la que trabajo (<https://www.laberit.com/>)

³ Framework de Java (<https://spring.io/projects/spring-boot>)

1.2 Objetivos

El objetivo principal de este Trabajo de Fin de Grado es desarrollar una nueva aplicación web que permita gestionar de forma eficiente las solicitudes de Anticipos de Nómina por parte del personal del área de contabilidad del Cabildo de Gran Canaria.

Para lograrlo, se plantean los siguientes objetivos específicos:

- Analizar las limitaciones funcionales y técnicas de la aplicación actual.
- Diseñar una nueva interfaz de usuario con un enfoque moderno y accesible (Figma, 2023; PrimeFaces, 2025), siguiendo las directrices del equipo de diseño de Läberit.
- Implementar una arquitectura basada en tecnologías actuales como Java Spring Boot y Primefaces⁴ (Spring Data Team, 2022; PrimeFaces, 2025).
- Incorporar funcionalidades que permitan gestionar solicitudes en distintos estados (concesiones, desistimientos, pagos, etc.) y realizar los cálculos mensuales correspondientes de forma automatizada.
- Establecer la integración con el sistema de Recursos Humanos del Cabildo mediante una vista con los datos de empleados con derecho a solicitar anticipos.
- Desplegar la aplicación en un entorno de preproducción (Red Hat, 2021) para su validación por parte del cliente, recogiendo feedback y posibles incidencias o mejoras.

El desarrollo de esta aplicación no solo busca resolver los problemas del sistema actual, sino también sentar las bases para futuros desarrollos dentro del ecosistema tecnológico del Cabildo.

⁴ <https://www.primefaces.org/>

1.3 Metodología

El desarrollo que se seguirá al realizar este proyecto tendrá una metodología basada en la eficiencia del proceso, la adaptabilidad y la mejora continua a lo largo del proyecto. Para conseguir esto aprovecharemos las metodologías ágiles⁵ ya que nos permiten obtener todo lo mencionado anteriormente al implementarlas en un ámbito profesional del desarrollo del software. Más allá de alcanzar los objetivos técnicos planteados, trabajaremos con las herramientas que se utilizan en la actualidad en la gestión de proyectos para favorecer la organización, la colaboración y el seguimiento continuo del trabajo.

Para lograr trabajar de manera ágil se seguirán los principales aspectos organizativos y metodológicos:

1.3.1 Equipo multidisciplinar

El equipo de Läberit responsable del mantenimiento y evolución de las aplicaciones del Cabildo de Gran Canaria está compuesto por un conjunto especializado de profesionales en el desarrollo del software. Entre ellos se encuentran desarrolladores backend y frontend, especialistas en bases de datos, y expertos en integración de servicios web. Esta diversidad de perfiles permite abordar todas las distintas tareas y necesidades que plantea el cliente, cubriendo todos los aspectos del ciclo de vida de una aplicación software.

Dentro de este equipo, mi papel ha sido bastante amplio, ya que he participado en prácticamente todas las áreas del desarrollo de la nueva aplicación. Aunque en Läberit cada perfil suele estar más centrado en un ámbito concreto (backend, frontend, bases de datos o integración de servicios), en mi caso he trabajado de manera transversal.

He llevado a cabo el desarrollo de la lógica de negocio en el backend con Spring Boot (Spring Data Team, 2022), la implementación de las pantallas en el frontend siguiendo el diseño propuesto, y también la validación del funcionamiento general en el entorno de preproducción. Incluso, aunque esta aplicación en concreto no integra servicios web, me encargo de ese tipo de desarrollos en otros proyectos, por lo que también habría sido mi responsabilidad en caso de que se hubieran requerido.

En resumen, mi labor en este proyecto ha sido la de un desarrollador versátil, capaz de abarcar desde la implementación de la interfaz de usuario hasta la lógica del negocio. Esto me ha permitido tener una visión global del sistema y entender mejor cómo encajan todas las piezas, lo que considero una de las mayores aportaciones de este TFG a mi formación profesional.

1.3.2 Gestión de tareas mediante JIRA y asignación por especialidad

El seguimiento de tareas y la planificación del trabajo diario se realiza a través de la herramienta JIRA⁶ (Atlassian, 2023). En esta plataforma, el Product Owner registra las peticiones y necesidades que llegan desde el Cabildo de Gran Canaria, organizándose en forma de tickets. Posteriormente, se realiza una asignación de tareas considerando tanto la prioridad de cada solicitud como las habilidades técnicas y experiencia de cada miembro del equipo.

⁵ Métodos Ágiles. Scrum, Kanban, Lean (Rafael de las Heras del Dedo, 2017)

⁶ <https://www.atlassian.com/es/software/jira>

1.3.3 Reuniones de seguimiento semanales

El equipo celebra reuniones de seguimiento de forma semanal para conseguir una comunicación fluida y una visión compartida del avance del proyecto. En estas sesiones, cada miembro expone el trabajo realizado, las tareas en curso y la carga de trabajo actual. Este espacio también sirve para detectar posibles cuellos de botella y redistribuir tareas en caso de que algún integrante tenga una carga excesiva o se encuentre sin trabajo. Esta dinámica favorece la colaboración y la eficiencia del equipo.

1.3.4 Sprints de cuatro semanas

El trabajo se organiza en sprints⁷ de cuatro semanas, al final de cada sprint se realiza una revisión general del trabajo completado, asegurando que las funcionalidades desarrolladas estén finalizadas y listas para su validación. Esta planificación por ciclos permite medir el rendimiento, ajustar el ritmo de trabajo y realizar mejoras continuas en el proceso de desarrollo.

1.3.5 Entornos de preproducción y validación del cliente

Una vez finalizada la aplicación se despliega en un entorno de preproducción. Este servidor permite al Product Owner y al cliente realizar pruebas funcionales y validar las nuevas funcionalidades implementadas. A partir de este proceso de validación, se recopila feedback que se traduce en nuevos tickets en JIRA, ya sea para corregir errores detectados o para incorporar mejoras adicionales. Este ciclo continuo asegura que el producto evolucione de manera alineada con las necesidades reales del cliente.

⁷ Unidad de tiempo utilizada en metodologías ágiles

1.4 Planificación

El desarrollo del presente Trabajo de Fin de Grado se ha estructurado en una serie de fases secuenciales y siguiendo un enfoque ágil. Partiendo del análisis del sistema actual hasta la implementación completa de la nueva solución. A continuación, se detallan las fases del proceso:

1.4.1 Análisis de la aplicación actual

Como punto de partida, se realiza un análisis de la aplicación existente, desarrollada con Oracle Forms (Oracle Corporation, 2019). Esta fase tiene como objetivo identificar las principales limitaciones y deficiencias del aplicativo actual.

1.4.2 Análisis de requisitos de la nueva aplicación

Una vez detectadas las carencias de la solución anterior, se procede a crear la primera versión del análisis enfatizando los requisitos funcionales y no funcionales de la nueva aplicación.

1.4.3 Diseño de la interfaz de usuario

Con los requisitos definidos, se elabora un nuevo diseño para la interfaz de usuario, centrado en la usabilidad (Nielsen, 2012), la accesibilidad y la coherencia visual. El diseño es realizado con Figma (Figma Inc., 2023), siguiendo las propuestas del equipo de diseño de Läberit, y utilizando el framework Bootstrap⁸ para facilitar su implementación posterior en el frontend.

1.4.4 Desarrollo del backend

Una vez diseñada la base de datos, se procede al desarrollo del backend de la aplicación utilizando Java Spring Boot (Spring Data Team, 2022). Esta etapa consiste en implementar los servicios, los repositorios que se conectan con la base de datos y hacer la integración con la base de datos a través de JPA⁹ y consultas SQL (Spring JPA, 2025).

1.4.5 Desarrollo del frontend

Posteriormente, se implementa el frontend utilizando la biblioteca Primefaces, siguiendo el diseño definido previamente. Esta fase incluye la creación de formularios, componentes de interfaz y navegación entre ventanas.

1.4.6 Integración

Finalmente, se integran el backend y el frontend para tener un aplicativo válido que subir al servidor de preproducción con el objetivo de empezar a validar la aplicación e ir recibiendo feedback en Jira por parte de los usuarios.

⁸ <https://getbootstrap.com/>

⁹ <https://spring.io/projects/spring-data-jpa>

2 Estado del arte

2.1 Estado de la aplicación actual

Actualmente, la aplicación utilizada para la gestión de anticipos en el Cabildo de Gran Canaria está desarrollada con Oracle Forms (Oracle Corporation, 2019), una tecnología que en su momento fue ampliamente adoptada por las administraciones públicas debido a su facilidad de uso y a su capacidad para construir rápidamente interfaces basadas en operaciones CRUD¹⁰. Su estrecha integración con bases de datos Oracle¹¹ la convirtió en una solución efectiva durante muchos años para sistemas de gestión interna.

Sin embargo, con el paso del tiempo, Oracle Forms ha sido considerada una tecnología obsoleta en el contexto del desarrollo de aplicaciones modernas. Uno de los principales motivos es su arquitectura monolítica, que dificulta la escalabilidad, la reutilización de componentes y la integración con servicios externos. Además, presenta importantes limitaciones de mantenimiento, especialmente cuando se requieren actualizaciones, cambios funcionales o migraciones a nuevas plataformas tecnológicas. A esto se suma la falta de soporte para dispositivos móviles o entornos web responsivos, lo que impide adaptarse a los estándares actuales de accesibilidad y experiencia de usuario.

Otro aspecto crítico que afecta al sistema actual es su dependencia exclusiva del navegador Internet Explorer, ya que la aplicación fue diseñada específicamente para ejecutarse en dicho entorno. Internet Explorer fue oficialmente retirado por Microsoft¹² en 2022, y su motor de renderizado no es compatible con las tecnologías web modernas. Esto no solo limita gravemente la compatibilidad de la aplicación con otros navegadores, sino que también compromete su rendimiento, seguridad y usabilidad, dificultando la experiencia de los usuarios y la evolución del sistema.

2.2 UX/IU

Tradicionalmente, aplicaciones como la desarrollada con Oracle Forms daban prioridad al funcionamiento interno y la lógica de negocio, dejando en segundo plano el diseño visual o la experiencia del usuario (Nielsen, 2012). Como resultado, la interfaz era funcional pero poco intuitiva, con una estética obsoleta y escasa flexibilidad para adaptarse a distintos dispositivos, tamaños de pantalla o estándares de accesibilidad. Además, los formularios eran rígidos, con escaso feedback visual y una navegación poco fluida, lo que dificulta su uso por parte de usuarios no técnicos.

En la actualidad, las aplicaciones deben cumplir no solo con los requisitos funcionales, sino también con expectativas de usabilidad y diseño, ofreciendo una experiencia agradable, coherente y adaptada a distintos contextos. Para lograr esto, se emplean herramientas especializadas como Figma (Figma Inc., 2023), que permiten diseñar y prototipar interfaces centradas en el usuario, y frameworks frontend como Bootstrap (Bootstrap, 2023), que facilitan la creación de interfaces responsive, limpias y accesibles.

¹⁰ Create, read, Update, Delete.

¹¹ <https://www.oracle.com/>

¹² <https://www.microsoft.com>

La aplicación propuesta en este proyecto tiene como uno de sus principales objetivos mejorar significativamente la experiencia de usuario, con una interfaz más clara, moderna e intuitiva. Para ello, se seguirá un enfoque centrado en el usuario desde las fases iniciales del diseño, aplicando buenas prácticas de UX y principios de diseño visual actuales, como consistencia, retroalimentación inmediata, jerarquía visual, y accesibilidad.

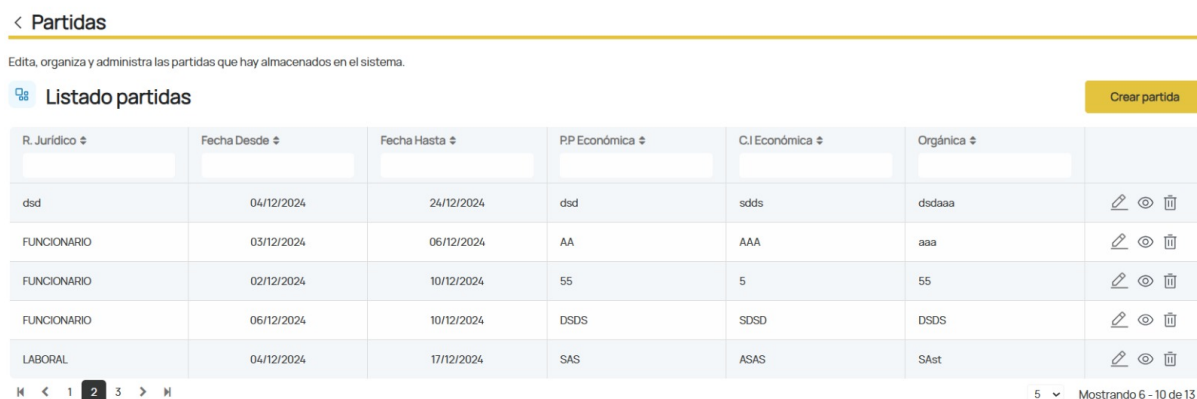
A continuación se muestra una comparación de una parte simple de la aplicación antigua con el diseño propuesto de la misma:



The screenshot shows a window titled 'Mantenimiento de Partidas'. Inside, there's a table with columns for 'Rjuridico', 'Fec. Desde', 'Fec. Hasta', 'Partida Presupuestaria' (subdivided into 'Organica', 'prog.', 'Economica'), and 'Concepto ingreso' (subdivided into 'Organica', 'prog.', 'Economica'). The table contains 8 rows of data for 'FUNCIONARIO' and 'LABORAL' across different years (2016-2019).

Rjuridico	Fec. Desde	Fec. Hasta	Partida Presupuestaria			Concepto ingreso		
			Organica	prog.	Economica	Organica	prog.	Economica
FUNCIONARIO	01/01/2019		02060	920	830000119	02060		831000119
LABORAL	01/01/2019		02060	920	830000219	02060		831000219
FUNCIONARIO	01/01/2018	31/12/2018	02060	920	830000118	02060		831000118
LABORAL	01/01/2018	31/12/2018	02060	920	830000218	02060		831000218
FUNCIONARIO	01/01/2017	31/12/2017	02060	920	830000117	02060		831000117
LABORAL	01/01/2017	31/12/2017	02060	920	830000217	02060		831000217
FUNCIONARIO	01/01/2016	31/12/2016	02060	920	830000116	02060		831000116
LABORAL	01/01/2016	31/12/2016	02060	920	830000216	02060		831000216

Imagen 1. Mantenimiento de ejemplo del aplicativo anterior



The screenshot shows a modern web interface for 'Partidas'. It includes a header with a back arrow and the title 'Partidas'. Below the header, there's a description: 'Edita, organiza y administra las partidas que hay almacenados en el sistema.' A 'Crear partida' button is on the right. The main section is titled 'Listado partidas' and contains a table with columns: 'R. Jurídico', 'Fecha Desde', 'Fecha Hasta', 'PP Económica', 'CI Económica', and 'Orgánica'. The table lists 6 rows of data. At the bottom, there's a pagination bar showing '5' and 'Mostrando 6 - 10 de 13'.

R. Jurídico	Fecha Desde	Fecha Hasta	PP Económica	CI Económica	Orgánica	
dsd	04/12/2024	24/12/2024	dsd	sdds	dsdaaa	  
FUNCIONARIO	03/12/2024	06/12/2024	AA	AAA	aaa	  
FUNCIONARIO	02/12/2024	10/12/2024	55	5	55	  
FUNCIONARIO	06/12/2024	10/12/2024	DSDS	SDSD	DSDS	  
LABORAL	04/12/2024	17/12/2024	SAS	ASAS	SAst	  

Imagen 2. Diseño propuesto para un mantenimiento de ejemplo

2.3 Análisis con propuestas de mejora para la nueva aplicación

Tras el análisis de la aplicación actual, desarrollada con Oracle Forms (Oracle Corporation, 2019) y basada en una arquitectura monolítica, se identificaron diversas limitaciones que afectan tanto al plano técnico como a la experiencia de usuario. Estos problemas no solo reducen la eficiencia en el uso diario de la herramienta, sino que dificultan su mantenimiento y evolución.

En primer lugar, se propone sustituir la arquitectura monolítica actual por una arquitectura basada en capas (MVC) que facilite el mantenimiento, la escalabilidad y la integración futura con otros sistemas del Cabildo.

Lógicamente también se ha propuesto cambiar las tecnologías obsoletas por unas actuales y más adaptadas al entorno empresarial como Java Spring Boot, Maven y Java 11 (Spring Data Team, 2022; Apache Maven, 2025).

Además del rediseño visual, se trabajará en mejorar la experiencia general de los usuarios que gestionan los anticipos. Para ello, se implementarán los siguientes cambios:

- Las funcionalidades de alta y edición de datos se trasladan a pantallas separadas para evitar el uso de ventanas modales con muchos datos.
- Validaciones en tiempo real a la hora de crear o editar objetos.
- Feedback visual claro mostrando mensajes de error, éxito o carga.

El nuevo aplicativo será diseñado de manera que pueda ser accesible desde navegadores modernos como Chrome¹³, Firefox¹⁴ o Microsoft Edge¹⁵, para garantizar que es compatible con los estándares web actuales y no depender de Internet Explorer.

¹³ <https://www.google.com/intl/es-es/chrome/>

¹⁴ <https://www.firefox.com>

¹⁵ microsoft.com/es-es/edge

En la siguiente tabla se puede observar una comparación de los cambios más importantes a realizar en la nueva aplicación de Gestión de Anticipos de Nómina:

Criterio	Tecnología Actual	Tecnología Propuesta
Tecnología base	Oracle Forms	Java con Spring Boot
Dependencia del navegador	Solo compatible con Internet Explorer	Compatible con todos los navegadores modernos
Arquitectura	Monolítica	Basada en capas (MVC)
Experiencia de usuario (UX)	Interfaz anticuada y limitada	Interfaz moderna, responsiva y centrada en el usuario
Diseño de interfaz	Basado en formularios estáticos	Diseñado con Figma y desarrollado con Bootstrap
Mantenimiento y evolución	Difícil y costoso	Mantenible y fácilmente escalable
Soporte y futuro	Tecnología en desuso	Ecosistema activo y en evolución constante

Tabla 1. Comparación de las tecnologías actuales y las propuestas

2.4 Análisis de la antigua aplicación

La aplicación actual se trata de una aplicación creada con Oracle Forms (Oracle Corporation, 2019), con un diseño y estructura muy antiguo (Antiguo_análisis_GDAN, 2025). La base de datos está creada en SQL¹⁶ y será la única parte de la aplicación que no recibirá grandes cambios debido a que el Cabildo de Gran Canaria quiere mantener la conexión a la base de datos.

2.4.1 Flujo de una solicitud

El objetivo principal de la aplicación es crear una solicitud que al final del proceso dará el alta a un anticipo si se concede y todos los parámetros son correctos. Hasta llegar a esto sigue un flujo de solicitud que se puede observar en la siguiente imagen:

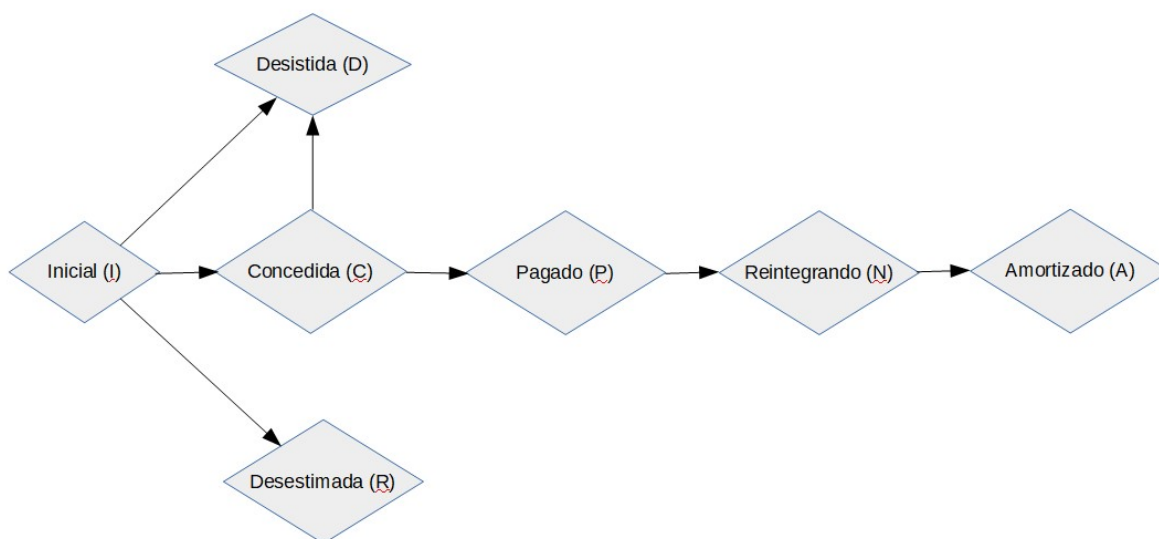


Imagen 3. Flujo de la solicitud

¹⁶ Lenguaje de programación para gestionar bases de datos

2.4.2 Modelo lógico

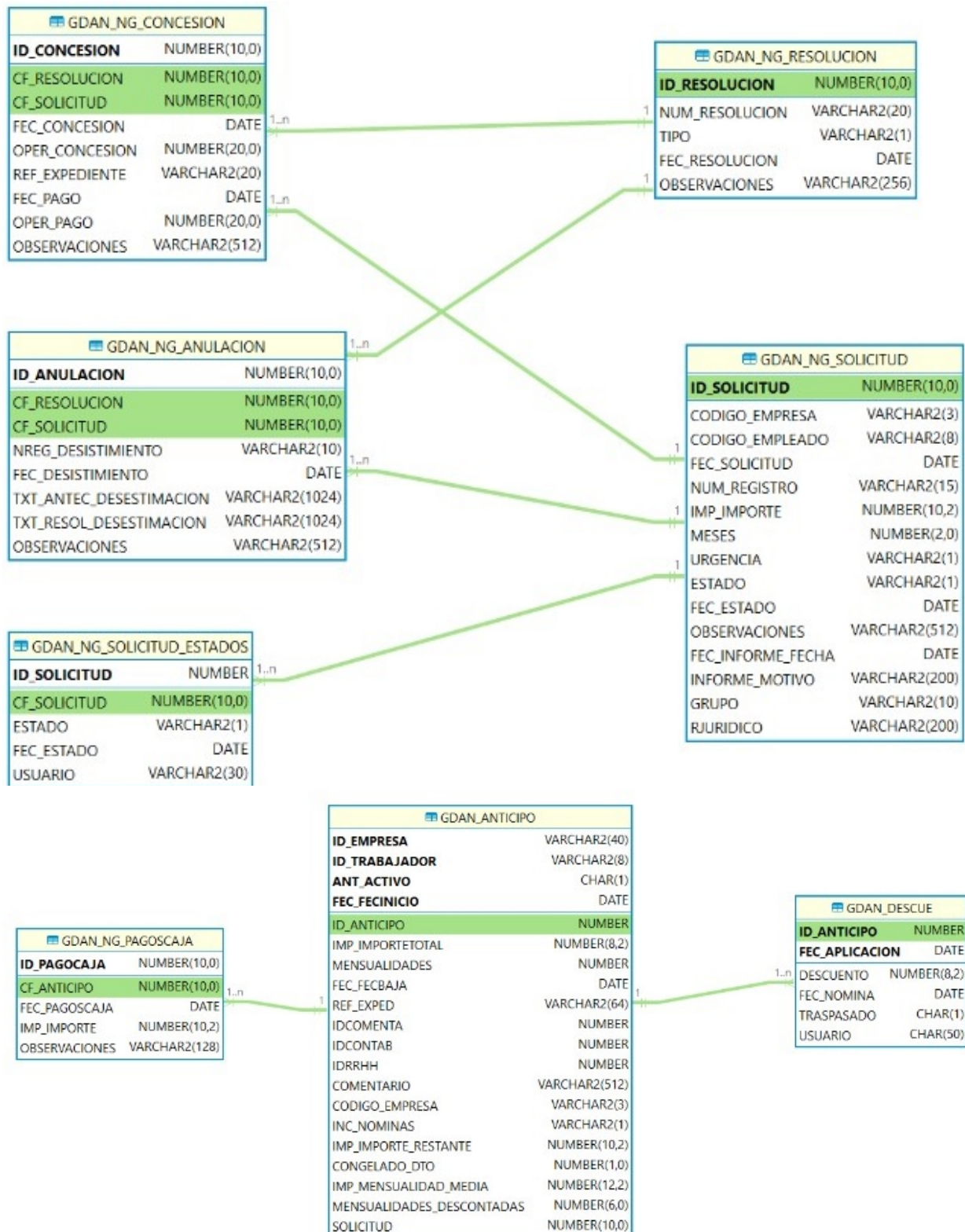


Imagen 3. Modelo lógico

2.4.3 Vistas de la antigua aplicación

Con el objetivo de ver el antiguo diseño de la aplicación y hacerse una idea de la estructura de la antigua aplicación se mostrarán las antiguas vistas:

Vista principal



Imagen 4. Vista principal de la antigua aplicación

Mantenimiento de límite importe/plazo

Fec. Desde		Fec. Hasta		Limite 1		Limite 2		Limite 3	
Desde	Hasta	Desde	Hasta	Desde	Hasta	Desde	Maximo		
01/01/2011		300.51\$	1502.52\$	1502.53\$	2103.53\$	2103.54\$	3005.06\$		

Fec. Desde		Fec. Hasta		Grupo	Limite 1 Meses	Limite 2 Meses	Limite 3 Meses
01/01/2011				A1 / I	8	10	12
01/01/2011				A2 / II	10	12	15
01/01/2011				C1 / III	12	15	18
01/01/2011				C2 / IV	16	18	21
01/01/2011				OAP / V	18	20	24

Imagen 4. Vista del listado de importes y plazos de la antigua aplicación

Mantenimiento de partidas presupuestarias

Mantenimiento de Partidas

Partidas

Rjuridico	Fec. Desde	Fec. Hasta	Partida Presupuestaria			Concepto ingreso		
			Organica	prog.	Economica	Organica	prog.	Economica
FUNCIONARIO	01/01/2019		02060	920	830000119	02060		831000119
LABORAL	01/01/2019		02060	920	830000219	02060		831000219
FUNCIONARIO	01/01/2018	31/12/2018	02060	920	830000118	02060		831000118
LABORAL	01/01/2018	31/12/2018	02060	920	830000218	02060		831000218
FUNCIONARIO	01/01/2017	31/12/2017	02060	920	830000117	02060		831000117
LABORAL	01/01/2017	31/12/2017	02060	920	830000217	02060		831000217
FUNCIONARIO	01/01/2016	31/12/2016	02060	920	830000116	02060		831000116
LABORAL	01/01/2016	31/12/2016	02060	920	830000216	02060		831000216

Imagen 5. Vista del listado de partidas presupuestarias de la antigua aplicación

Tramitación de solicitudes

Solicitudes

Empleados

Codigo: L0014 Nombre: [REDACTED] Apellido1: [REDACTED] Apellido2: [REDACTED]

NIF: [REDACTED] Grupo: IV reg.jur.: LABORAL Edad: [REDACTED]

Colectivo: LABORAL - CABILDO rel. emp.: LABORAL FIJO

Alta: 01/11/1986 Baja: [REDACTED] Situacion: ACTIVO

Solicitudes

solicitud	Fecha	nº Registro	Importe	Meses	Urg.	Estado	Fecha Estado
784	17/08/2023	20233200065	3000.00\$	21	☒	Reintegrand	19/09/2023 11:11:43
749	15/02/2023	20232000125	3000.00\$	21	☐	Pendiente	06/03/2023 00:00:00
613	30/06/2020	032010866	3000.00\$	21	☐	Desistido	29/03/2021 08:29:44
540	05/02/2019	2177	3000.00\$	21	☐	Desistido	04/09/2020 13:13:55
190	02/01/2014	03	3000.00\$	21	☐	Reintegrand	09/04/2014 12:00:18

Otros Datos

Informe favorable Junta/Comite Fecha Motivo

Solicitud Informe: [REDACTED]

Estado: Reintegrand Ver Resolucion

Observaciones: [REDACTED]

<< Volver

Imagen 6. Vista del listado de solicitudes de la antigua aplicación

Tramitación de concesiones

Concesiones de Anticipos

Resolucion

nº Resolucion: 216 Fecha:

Observaciones:

Solicitudes concedidas

Usuario GEMA ☐ Correcto ☒ Error

Empleado	Fec.solic.	Importe	Meses	F. Conc.	nº oper. Contab.
[REDACTED]	25/11/2016	3005.06\$	18	10/02/2017	12017000006000
[REDACTED]	03/01/2017	3000.00\$	24	10/02/2017	12017000006001
[REDACTED]	04/01/2017	3000.00\$	15	10/02/2017	12017000006003
[REDACTED]	10/01/2017	3000.00\$	21	10/02/2017	12017000006816
[REDACTED]	18/01/2017	3000.00\$	12	10/02/2017	12017000006004
[REDACTED]	23/01/2017	3000.00\$	12	10/02/2017	12017000006006

Imagen 7. Vista del listado de solicitudes concedidas de la antigua aplicación

Concesiones de Anticipos

Resolucion

nº Resolucion: 216 Fecha:

Observaciones:

Solicitudes pendientes

Inc.	Solicitud	Empleado	Fecha	nº Registro	Importe	Meses	Urg.	Grupo	Rjuridico
☐	803	[REDACTED]	23/01/2024	2024004712	3000.00\$	18	☒	C1	FUNCIONAR
☐	749	[REDACTED]	15/02/2023	20232000125	3000.00\$	21	☐	IV	LABORAL
☐	762	[REDACTED]	17/04/2023	2023031974	3000.00\$	24	☐	V	LABORAL
☐	776	[REDACTED]	19/07/2023	2023060397	3000.00\$	24	☐	V	LABORAL
☐	794	[REDACTED]	05/10/2023	2023078256	2500.00\$	24	☐	OAP	FUNCIONAR
☐							☐		
☐							☐		
☐							☐		
☐							☐		
☐							☐		
☐							☐		
☐							☐		
☐							☐		
☐							☐		
☐							☐		

Imagen 8. Vista del listado de solicitudes a conceder de la antigua aplicación

Tramitación de desestimaciones

Desestimación de Anticipos

Resolución

nº Resolución: 1197 Fecha:

Observaciones:

Solicitud desestimada

Empleado	Fec.solic.	Importe	Meses	R. juridico
	13/10/2020	5040.00\$	12	LABORAL

Observaciones:

MOTIVO

antecedentes ... en el que solicita un anticipo reintegrable,
visto el artículo 43 del IV Convenio Colectivo del Personal Laboral, que dice: ¿el trabajador con contrato indefinido tendrá derecho a percibir anticipos? y visto el contrato de trabajo de laboral interino en plaza vacante firmado por el Sr. Perez Jimenez con esta Corporación de fecha 13/08/2020
por la presente: RESUELVO

desestimacion ... presentada por [],
toda vez que no cumple el requisito de trabajador con contrato indefinido, pues mantiene una relación contractual con el Cabildo de Gran Canaria de laboral interino en plaza vacante de duración determinada

Imagen 9. Vista del alta de desestimación de la antigua aplicación

Tramitación de desistimientos

Desistimiento de Anticipos

Resolución

nº Resolución: 0 Fecha:

Observaciones:

Solicitud desistida

Empleado	Fec.solic.	Importe	Meses	R. juridico
	03/02/2017	3005.06\$	15	LABORAL

nº reg. ent. Fecha Desistimiento

Observaciones: POR ERROR SE LE MECANIZÓ LA SOLICITUD. LA SOLICITÓ BOCAJUDICIL COMPLETAR

Imagen 10. Vista del alta de desistimiento de la antigua aplicación

Empleado			
Cod. Empresa	AYU	Código Empleado	L0014
Vinculación Jur.	LABORAL	Dni/Mie	Letra
Primer apellido	id3274	Colectivo	LABORAL - CABILDO
Segundo apellido	id3274	Contrato	LABORAL FIJO
Nombre	id3274	Grupo	IV
Fecha de Alta	01/11/1986	Fecha de Baja	

Anticipo			
Importe total	3000.00\$	Mensualidades	21
Importe Restante	2428.56\$	Mens. Descontadas	4
Fecha Inicio	01/10/2023	Rev.	Ref. Exped. 50944/23
n° oper. contab.	12023000050944	Solicitud	17/08/2023 202332000654
Comentario	Anticipo Activo		

Consulta de solicitudes

[illegible]

Imagen 13. Vista de la consulta de solicitudes de la antigua aplicación

Consulta de anticipos

Consulta de Anticipos

Filtro

Situacion

Todos

Activo

ILT

Cese

Jubilado

Excedencia /ot...

Desde

Hasta

Régimen Jurídico

Pendientes

Congelados

Saldo

emp. no activo

cto. congelado

l.t.

Anticipos

código	Nombre	Apellido1	Apellido2	Fecha	Importe	meses	Cuota	Saldo	Situacion	Incidencia
L6126				01/02/2024	3000.00\$	24	125.00\$	3000.00\$	ACTIVO	
T4820				01/12/2023	3000.00\$	24	125.00\$	2750.00\$	ACTIVO	
T3758				01/12/2023	3000.00\$	12	250.00\$	2500.00\$	ACTIVO	
L6387				01/12/2023	3000.00\$	12	250.00\$	2500.00\$	ACTIVO	
L0504				01/12/2023	3000.00\$	24	125.00\$	2750.00\$	ACTIVO	
				01/11/2023	3005.06\$	12	250.43\$	2253.80\$	ACTIVO	
D2335				01/11/2023	3000.00\$	24	125.00\$	2625.00\$	ACTIVO	
				01/11/2023	3005.06\$	24	125.22\$	2629.43\$	ACTIVO	
X2159				01/10/2023	3005.06\$	21	143.10\$	2432.66\$	ACTIVO	
F1028				01/10/2023	3005.06\$	15	200.34\$	2203.70\$	ACTIVO	
D2017				01/10/2023	3000.00\$	24	125.00\$	2500.00\$	ACTIVO	
L6061				01/10/2023	3000.00\$	15	200.00\$	2200.00\$	ACTIVO	
L0014				01/10/2023	3000.00\$	21	142.86\$	2428.56\$	ACTIVO	
				01/09/2023	2500.00\$	15	166.67\$	1666.65\$	ACTIVO	
L6193				01/09/2023	3005.06\$	24	125.22\$	2379.01\$	ACTIVO	

205670.66\$

103576.00\$

Imagen 14. Vista de la consulta de anticipos de la antigua aplicación

3 Requisitos

Debido a que la aplicación ya fue creada en su momento y disponíamos de su primer análisis el análisis que se ha creado de la nueva aplicación se centra sobre todo en las mejoras a aplicar (CGC_GDAN-3_TOMREQ_v2 , 2025; CGC_GDAN-5_AF-Anticipos_v3 , 2025).

3.1 Características exigidas en la nueva aplicación

3.1.1 Migración a nuevas tecnologías

Una de las principales mejoras del nuevo sistema es la adopción de tecnologías modernas como Java Spring Boot, JPA, PrimeFaces y Java 11 (Spring JPA, 2025), con el objetivo de mejorar el rendimiento y eficiencia del sistema y hacer posible la ejecución en navegadores modernos, dejando atrás la dependencia de Internet Explorer. También abre la puerta a futuras ampliaciones, ya que estas tecnologías son compatibles con herramientas modernas como los servicios web, facilitando la integración con otros sistemas del Cabildo.

3.1.2 Separación de ventanas según funcionalidad

En la aplicación actual, las funciones de alta, consulta y edición se encontraban en la misma pantalla, abriendo ventanas modales, lo cual podría generar confusión. En la nueva versión, cada una de estas acciones cuenta con su propia ventana específica, excepto la opción de eliminar que ahora cuenta con una ventana de confirmación, lo que facilita al usuario entender qué está haciendo en cada momento y reduce la posibilidad de errores.

3.1.3 Diseño más claro e intuitivo

El rediseño de la interfaz de usuario ha sido uno de los puntos en los que más énfasis se ha querido poner. Se ha apostado por un aspecto visual más limpio y moderno, utilizando componentes gráficos que ayudan a guiar al usuario. La interfaz ha sido diseñada con Figma y construida con Bootstrap y Primefaces, lo que permite una experiencia más fluida y coherente en toda la aplicación.

3.1.4 Validación con mensajes de error

Para mejorar la experiencia de usuario y evitar errores en pleno uso de la aplicación, se han implementado mensajes de error o avisos que aparecen en tiempo real cuando se realiza alguna acción incorrecta o peligrosa (como dar de alta un objeto habiendo introducido datos inválidos o eliminar un registro que puede modificar otros registros en consecuencia).

3.1.5 Ventanas de la aplicación

A pesar de que el análisis está enfocado en las mejoras se han realizado mockups de las ventanas para tener claro los botones de cada vista, las columnas a mostrar en cada tabla...

Mantenimiento importes

**Cabildo de Gran Canaria**

Anticipos

PABLO FOS CLAVER
ROLE ADMINISTRADOR



 Cerrar

MENÚ 

Mantenimiento | Importes

Límites Importes

NuevoEditarEliminar

Fecha Desde ▾	Fecha Hasta ▾	Límite 1 Desde ▾	Límite 2 Desde ▾	Límite 3 Desde ▾	Límite Máximo ▾
01/01/2011		300.51	1502.52	2103.54	3005.06

◀ 1 ▶

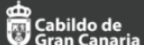
**Cabildo de Gran Canaria**

Cabildo de Gran Canaria © 2024

Versión 01.00.00
Fecha build: 12/07/2024 10:04:01

Imagen 15. Vista del diseño propuesto del listado de importes

Alta importe

**Cabildo de Gran Canaria**

Anticipos

PABLO FOS CLAVER
ROLE ADMINISTRADOR



 Cerrar

MENÚ 

Mantenimiento | Alta Límite Importe

Alta Límite Importe

Fecha Desde *

Fecha Hasta *

Límite 1 *

Límite 2 *

Límite 3 *

Máximo *

Guardar

Cancelar

**Cabildo de Gran Canaria**

Cabildo de Gran Canaria ©

Versión
Fecha build:

Imagen 16. Vista del diseño propuesto del alta de importes

Mantenimiento plazos


Anticipos
Anticipos
Cerrar
MENÚ

Mantenimiento | Plazo

Plazos

Nuevo
Editar
Eliminar

Fec.D.	Grupo	Mes1	Mes2	Mes3
08/05/2024	1	1	4	7

1



Cabildo de Gran Canaria ©

Versión
Fecha build:

Imagen 17. Vista del diseño propuesto del listado de plazos

Alta plazo


Anticipos
Anticipos
Cerrar
MENÚ

Mantenimiento | Alta Plazos

Alta Plazos

Fecha Desde *
Fecha Hasta
Grupo *
Mes1
Mes2
Mes3

Guardar
Cancelar



Cabildo de Gran Canaria ©

Versión
Fecha build:

Imagen 18. Vista del diseño propuesto del alta de plazos

Mantenimiento partidas



Anticipos

PABLO FOS CLAVER
ROLE_ADMINISTRADOR




Cerrar

MENÚ 

Mantenimiento | Partidas

Partidas

NuevoEditar

R. Jurídico	Fecha Desde	Fecha Hasta	PP Económica	C.I Económica	Orgánica
LABORAL	01/01/2019		830000219	831000119	02060
FUNCIONARIO	01/01/2019		830000119	831000119	02060
FUNCIONARIO	01/01/2019	31/12/2018	830000118	831000118	02060
LABORAL	01/01/2019	31/12/2018	830000218	831000218	02060

1



Cabildo de Gran Canaria

Cabildo de Gran Canaria © 2024

Versión 01.00.00
Fecha build: 12/07/2024 10:04:01

Imagen 19. Vista del diseño propuesto del listado de partidas

Alta partida

**Cabildo de Gran Canaria**

Anticipos

Cerrar

MENÚ

Mantenimiento | Alta Partidas

Alta Partida

R. Juridico *

0/200

Fecha desde *

Fecha hasta *

Orgánica *

0/20

Programa *

0/10

Económica *

0/20

Orgánica ingreso *

0/20

Programa ingreso *

0/10

Económica ingreso *

0/20

Guardar

Cancelar

**Cabildo de Gran Canaria**

Cabildo de Gran Canaria ©

Versión
Fecha build:

Imagen 20. Vista del diseño propuesto del alta de partidas

Listado solicitudes



Anticipos

PABLO FOS CLAVER
ROLE_ADMINISTRADOR





Cerrar

MENÚ



Tramitación | Solicitudes

Solicitudes

Reg. Juridico

DesdeHastaEstado

Concedidas

Buscar

Exportar a Excel

Nuevo

Editar

Eliminar

C. Empl	Nombre	Fecha Solicitud	Importe	Meses	Estado
L0014	Nombre1 Apellido1 Apellido2	17/08/2023	3000	21	Concedido

1



Cabildo de Gran Canaria

Cabildo de Gran Canaria © 2024

Versión 01.00.00

Fecha build: 12/07/2024 10:04:01

Imagen 21. Vista del diseño propuesto del listado de solicitudes

Alta solicitud

**Cabildo de Gran Canaria**

Anticipos

**Cerrar**

MENÚ 

Tramitación | Alta Solicitud

Alta Solicitud

Código Empresa *		0/20
Código Empleado *		0/20
Fecha Solicitud *		
Número Registro		0/20
Imp Importe *		
Meses *	0	
Estado *		0/20
Urgencia *		0/20
Fecha Estado *		
Observaciones		0/20
Fecha Informe Fecha		
Informe Motivo		0/20
Grupo		0/20
RJuridico		0/20

Guardar **Cancelar**

**Cabildo de Gran Canaria**

Cabildo de Gran Canaria © **Verificar**

Imagen 22. Vista del diseño propuesto del alta de solicitudes

Listado concesiones



Anticipos

PABLO FOS CLAVER
ROLE_ADMINISTRADOR




Cerrar

MENÚ 

Tramitación | Concesiones

Concesiones

Nuevo

Editar

Eliminar

Id. Resolución	Nº Resolución	Fecha Resolución
1	216	17/08/2023

1



Cabildo de Gran Canaria

Cabildo de Gran Canaria © 2024

Versión 01.00.00
Fecha build: 12/07/2024 10:04:01

Imagen 23. Vista del diseño propuesto del listado de concesiones

Alta concesión



Anticipos

PABLO FOS CLAVER
ROLE_ADMINISTRADOR





Cerrar

MENÚ

Tramitación | Alta Concesiones

Alta Concesiones

Número de Resolución *

Fecha Resolución *

dd/MM/yyyy

Selección de Solicitudes *

☐	Empleado ▾	Fecha Solic. ▾	Importe ▾	Meses ▾
☐	Nombre1 Apellido1 Apellido2	25/11/2023	3005	18
☐	Nombre2 Apellido1 Apellido2	03/01/2023	3000	18
☐	Nombre3 Apellido1 Apellido2	10/12/2023	3000	12
☐	Nombre4 Apellido1 Apellido2	10/11/2023	3000	15

1

Observaciones

0/255

Guardar

Cancelar



Cabildo de Gran Canaria

Cabildo de Gran Canaria © 2024

Versión 01.00.00

Fecha build: 04/07/2024 01:56:22

Imagen 24. Vista del diseño propuesto del alta de concesiones

Listado desestimaciones



Anticipos

PABLO FOS CLAVER
ROLE ADMINISTRADOR




Cerrar

MENÚ



Tramitación | Anulación | Desestimaciones

Desestimaciones

Nuevo

Editar

Eliminar

Nº Resolución	Fecha Resolución	Empleado	Fecha Solicitud	Meses
1001	01/01/2024	Empleado1 Apellido1 Apellido2	13/10/2023	12
1002	10/01/2024	Empleado2 Apellido1 Apellido2	13/11/2023	24
1197	20/07/2024	Empleado3 Apellido1 Apellido2	23/12/2023	18

1



Cabildo de Gran Canaria

Cabildo de Gran Canaria © 2024

Versión 01.00.00
Fecha build: 12/07/2024 10:04:01

Imagen 25. Vista del diseño propuesto del listado de desestimaciones

Alta desestimación



Anticipos

PABLO FOS CLAVER
ROLE ADMINISTRADOR





Cerrar

MENÚ

Tramitación

|

Anulación

|

Alta desestimaciones

Alta Desestimaciones

Número de Resolución *

Fecha Resolución *

dd/MM/yyyy

Selección de Solicitudes *

Empleado	Fecha Solic.	Importe	Meses
Nombre1 Apellido1 Apellido2	25/11/2023	3005	18
Nombre2 Apellido1 Apellido2	03/01/2023	3000	18
Nombre3 Apellido1 Apellido2	10/12/2023	3000	12
Nombre4 Apellido1 Apellido2	10/11/2023	3000	15

1

Observaciones

0/255

Motivo

No Fijo

Plazo de Amortización

Antecedentes

0/255

Motivo Desestimación

0/255

Guardar

Cancelar



Cabildo de Gran Canaria

Cabildo de Gran Canaria © 2024

Versión 01.00.00

Fecha build: 04/07/2024 01:56:22

Imagen 26. Vista del diseño propuesto del alta de desestimaciones

Listado desistimientos



Anticipos

PABLO FOS CLAVER
ROLE ADMINISTRADOR





Cerrar

MENÚ



Tramitación | Anulación | Desistimiento

Desistimiento

Nuevo

Editar

Eliminar

Nº Resolución	Fecha Resolución	Empleado	Fecha Solicitud	Meses
1001	01/01/2024	Empleado1 Apellido1 Apellido2	13/10/2023	12
1002	10/01/2024	Empleado2 Apellido1 Apellido2	13/11/2023	24
1197	20/07/2024	Empleado3 Apellido1 Apellido2	23/12/2023	18

«

«

1

»

»



Cabildo de Gran Canaria

Cabildo de Gran Canaria © 2024

Versión 01.00.00

Fecha build: 12/07/2024 10:04:01

Imagen 27. Vista del diseño propuesto del listado de desistimientos

Alta desistimiento



Anticipos

PABLO FOS CLAVER
ROLE ADMINISTRADOR





Cerrar

MENÚ



Tramitación | Anulación | Alta Desistimiento

Alta Desistimiento

Número de Resolución *

Fecha Resolución *

dd/MM/yyyy

Selección de Solicitudes *

Empleado	Fecha Solic.	Importe	Meses
Nombre1 Apellido1 Apellido2	25/11/2023	3005	18
Nombre2 Apellido1 Apellido2	03/01/2023	3000	18
Nombre3 Apellido1 Apellido2	10/12/2023	3000	12
Nombre4 Apellido1 Apellido2	10/11/2023	3000	15

1

Observaciones

0/255

Guardar

Cancelar



Cabildo de Gran Canaria

Cabildo de Gran Canaria © 2024

Versión 01.00.00

Fecha build: 04/07/2024 01:56:22

Imagen 28. Vista del diseño propuesto del alta de desistimientos

Listado pagos



Anticipos

PABLO FOS CLAVER
ROLE_ADMINISTRADOR





Cerrar

MENÚ 

Tramitación | Pagos

Pagos de Anticipo

NuevoEditarEliminar

Empleado	Importe	Meses	Fecha Pago	Ref. Exped
Empleado2 Apellido1 Apellido3	3000	24	20/08/2024	44451/24
Empleado1 Apellido1 Apellido2	1600	18	12/08/2024	33343/24

1



Cabildo de Gran Canaria

Cabildo de Gran Canaria © 2024

Versión 01.00.00
Fecha build: 12/07/2024 10:04:01

Imagen 29. Vista del diseño propuesto del listado de pagos

Alta pago



Anticipos

PABLO FOS CLAVER
ROLE ADMINISTRADOR





Cerrar

MENÚ



Tramitación | Alta Pagos

Alta de Pagos de Anticipo

Selección de Solicitudes Concedidas *

Empleado	Fecha Resolución	Nº Resol	Importe	Meses
Nombre1 Apellido1 Apellido2	25/11/2023	1009	3005	18
Nombre2 Apellido1 Apellido2	03/01/2023	1010	3000	18
Nombre3 Apellido1 Apellido2	10/12/2023	1011	3000	12
Nombre4 Apellido1 Apellido2	10/11/2023	1012	3000	15

1

Nº Operación Contab.

xxxxxxxxxxxx

Fecha Concesión

dd/MM/yyyy

Referencia Exp.

XXXXXXYY

Fecha de Pago *

dd/MM/yyyy

Observaciones

0/255

Guardar

Incidencia

Cancelar



Cabildo de Gran Canaria

Cabildo de Gran Canaria © 2024

Versión 01.00.00
Fecha build: 04/07/2024 01:56:22

Imagen 30. Vista del diseño propuesto del alta de pagos

Listado anticipos



Anticipos

PABLO FOS CLAVER
ROLE ADMINISTRADOR





Cerrar

MENÚ



Tramitación | Anticipos

Anticipos

Reg. Jurídico

DesdeHastaSituación

Todos

Buscar

Exporta a Excel

Listado Detallado

Listado Resumido

Nuevo

Editar

Eliminar

Empleado	Importe Total	Importe Restante	Cuota	Meses	Activo	Congelado
Empleado2 Apellido1 Apellido3	3000	2428,56	142.86	21	Si	No
Empleado3 Apellido1 Apellido3	2000	1428,56	142.86	18	Si	No
Empleado2 Apellido1 Apellido3	2000	0	142.86	18	No	No

1



Cabildo de Gran Canaria

Cabildo de Gran Canaria © 2024

Versión 01.00.00

Fecha build: 12/07/2024 10:04:01

Imagen 31. Vista del diseño propuesto del listado de anticipos

3.2 Requisitos funcionales

3.2.1 REQ-001 Roles de Usuario

Deben existir 4 perfiles o roles con distintos permisos:

- **Usuario de Nóminas.** El usuario de Nóminas tiene acceso a toda la aplicación excepto a los mantenimientos que serán gestionados por los administradores
- **Usuario de Intervención.** El usuario de Intervención sólo tiene acceso a las opciones de consulta de anticipos y a la comunicación mensual.
- **Usuario de Ingresos.** El usuario de Ingresos tiene acceso a las opciones de consulta de anticipos, comunicación mensual y tramitación de anticipos, pero de forma restringida.
- **Usuario Administrador.** El usuario administrador tiene acceso a toda la aplicación.

3.2.2 REQ-002 Partes de la aplicación

Se diferenciarán tres partes de la aplicación.

- **Mantenimientos.** Los mantenimientos de esta aplicación serán Importes, Plazos y Partidas.
- **Tramitación.** La tramitación constará de todo el flujo que debe seguir una solicitud hasta convertirse en un Anticipo concedido.
- **Consulta.** Las solicitudes creadas y los anticipos concedidos podrán ser consultados para ver la información asociada a estos.

3.2.3 REQ-003 Mantenimiento Límite de Importes y Plazos

En la sección de mantenimiento del menú habrá una subsección que será accesible por el usuario con el rol administrador donde se podrá establecer límites de los importes y los plazos máximos de devolución que se pueden conceder dependiendo del grupo al que pertenezca la plaza que ocupa el empleado y según lo estipulado en el convenio.

3.2.4 REQ-004 Mantenimiento Partidas

Se debe incluir una opción que permita gestionar la información sobre las partidas presupuestarias relativas a las distintas operaciones contables dependientes del régimen jurídico del empleado.

3.2.5 REQ-005 Tramitación Solicitudes Consulta de empleados

A la hora de dar de alta una solicitud el campo Empleado será rellenado seleccionando un valor de una lista de empleados obtenidos directamente de una vista conectada al aplicativo de recursos humanos del Cabildo de Gran Canaria.

3.2.6 REQ-006 Tramitación Solicitudes Validación

Al dar de alta una solicitud deberá de pasar las siguientes validaciones:

- Se debe de comprobar que el empleado no tenga solicitudes pendientes de liquidar.
- Debe de haber transcurrido un año desde la liquidación de anticipos previos
- El importe solicitado debe de estar dentro de los límites establecidos en el Convenio y parametrizado en el formulario de límites.
- Se debe validar el número de cuotas para la devolución. Se validará según lo establecido en el convenio al que pertenezca el grupo del empleado.

3.2.7 REQ-007 Tramitación Solicitudes con Urgencia

Se debe dar la posibilidad de tramitar la solicitud, aunque el empleado no cumpla con las validaciones descritas en el requisito 006. Esta opción debe ir acompañada de manera obligatoria con una descripción que indique el motivo por el cual se han ignorado las validaciones.

3.2.8 REQ-008 Gestión de Resolución por Concesión

Una resolución de concesión debe de poder incluir una o varias solicitudes de anticipos. Las solicitudes que se añadan a la concesión deben de estar en estado *pendiente* y cuando una solicitud es incorporada a una concesión esta debe de cambiar su estado a *concedida*.

Además, debe de incluir el dato de operación contable que se obtendrá vinculando la aplicación con el aplicativo de contabilidad.

Se debe establecer de manera automática el número de referencia del expediente pudiéndose calcular de manera automática a partir del código de referencia contable.

3.2.9 REQ-009 Gestión de Anulaciones por Desestimación

La gestión de una desestimación de una solicitud de anticipo se hará de manera individualizada por empleado, se deberá indicar un motivo en el cual se describen los antecedentes y motivo de desestimación de la solicitud.

La solicitud que incorporamos a la resolución de desestimación debe de estar en estado *pendiente*.

Cuando una solicitud es incorporada a la resolución de desestimación su estado pasa a *desestimada*.

3.2.10 REQ-010 Gestión de Anulaciones por Desistimiento

La gestión de un desistimiento de una solicitud de anticipo se hará de manera individualizada por empleado.

Sólo se podrá desistir mientras no se haya realizado el pago del anticipo y aquellas solicitudes que estén en estado *pendiente* o *concedida*.

Cuando una solicitud es incorporada a la resolución de desistimiento es pasada a estado *desistida*.

3.2.11 REQ-011 Tramitación de Pagos

La gestión de un pago de anticipo se hará de manera individualizada por concesión.

Sólo se podrá efectuar el pago de solicitudes en el estado *concedida*.

Cuando un pago es concedido el estado de la solicitud asociada es pasada a estado *pagado*. En el momento en el que se confirma a través del aplicativo de contabilidad del Cabildo de Gran Canarias que se ha realizado el pago se pasa al estado *reintegrando* y se calcula el valor de mensualidad de la solicitud.

3.2.12 REQ-012 Tramitación Gestión de Anticipos

La creación de un anticipo se hará de manera automática en la base de datos al avanzar una concesión a pago.

Deberá existir una ventana que permita visualizar y editar los pagos fuera de nómina, solo si el usuario tiene el rol necesario para ello. También deberá existir la opción de congelar de manera temporal el cobro del anticipo para parte de la administración, es decir, congelar la mensualidad que se le realiza al empleado en la nómina.

3.2.13 REQ-013 Consulta de Solicitudes

El listado de solicitudes de mostrará en formato de tabla con la siguiente información:

- Cod Empleado
- Nombre de empleado
- Fecha Solicitud
- Importe solicitado
- Número de meses en el que amortizará el anticipo
- Estado

Se deberá poder acceder al detalle de cada una de las solicitudes.

Se deberá poder generar un documento Excel con el listado de solicitudes.

3.2.14 REQ-014 Consulta de Anticipos

Deberá existir la posibilidad de acceder a la consulta o edición del anticipo pudiendo ver la siguiente información.

Relativa al empleado:

- Cod. Empresa
- Cod. Empleado
- Número documento de identidad
- Régimen jurídico
- Colectivo
- Datos personales (Nombre, apellidos)
- Fecha de alta y baja
- Contrato y grupo laboral

Relativa al anticipo concedido:

- Importe total
- N.º de mensualidades
- Importe de la cuota
- Importe restante
- N.º de mensualidades cobradas
- Código de referencia del expediente
- Fecha de inicio
- Fecha y código de solicitud
- N.º de operación contable
- Si el anticipo está activo
- Observaciones

Relativa a los pagos fuera de nómina y retenciones de nómina que se le han realizado al empleado para el cobro del anticipo:

- Fecha de la retención
- Importe del descuento realizado en la nómina
- Fecha de la nómina
- Número total de retenciones
- Importe total de las retenciones

3.2.15 REQ-015 Generación de documentos

Se deberá permitir la generación de documentos de comunicación con los usuarios de nóminas.

Documento con información por empleado relativa al propio anticipo

- Régimen Jurídico
- Código empleado
- Nombre y apellidos
- NIF
- Fecha Alta
- Importe Total
- Mensualidades
- Cuota
- Saldo
- Situación

3.2.16 REQ-016 Plazos de amortización

El sistema deberá permitir registrar solicitudes de anticipos teniendo en cuenta que el importe y el plazo debe de estar dentro de los rangos permitidos según el grupo profesional del empleado siguiendo esta tabla:

	DESDE	HASTA	DESDE	HASTA	DESDE	HASTA
GRUPO	300,51€	1.502,52€	1.502,53€	2.103,53€	2.103,54€	3.005,06€
1	8 meses		10 meses		12 meses	
2	10 meses		12 meses		15 meses	
3	12 meses		15 meses		18 meses	
4	16 meses		18 meses		21 meses	
5	18 meses		20 meses		24 meses	

Tabla 2. Plazos de amortización de los anticipos a conceder

3.3 Requisitos no funcionales

3.3.1 RNF-01-Rendimiento

La aplicación deberá ser capaz de cargar cada vista en un tiempo inferior a 2 segundos bajo condiciones normales de uso.

3.3.2 RNF-02-Seguridad

El acceso a la aplicación deberá requerir autenticación mediante usuario y contraseña, integrándose con el sistema de login corporativo del Cabildo.

3.3.3 RNF-03-Compatibilidad

La aplicación deberá ser compatible con los navegadores modernos: Google Chrome, Microsoft Edge y Mozilla Firefox.

3.3.4 RNF-04-Usabilidad

La interfaz debe seguir criterios de diseño responsive y accesible, de forma que se adapte correctamente a diferentes resoluciones de pantalla.

3.3.5 RNF-05-Usabilidad

La aplicación deberá utilizar un diseño coherente en todas sus vistas, siguiendo las guías de estilo proporcionadas por el equipo de diseño de Läberit.

3.3.6 RNF-06-Mantenibilidad

El código fuente deberá estar estructurado de forma modular siguiendo buenas prácticas (MVC, separación de capas...).

3.3.7 RNF-07-Despliegue y disponibilidad

La aplicación deberá estar desplegada en un entorno de preproducción accesible para los clientes del Cabildo para su validación.

3.4 Análisis de requisitos y diagramas UML

Tras la definición de los requisitos funcionales y no funcionales, resulta necesario llevar a cabo un análisis más gráfico que permita entender de forma clara cómo interactúan los usuarios con el sistema, cuáles son las entidades principales y qué procesos dinámicos deben cubrirse. Para ello, se utiliza el lenguaje de modelado unificado (UML).

3.4.1 Diagrama de Casos de Uso

El diagrama de casos de uso muestra cómo los distintos actores interactúan con la aplicación. En esta aplicación existen cuatro roles, cada uno con diferentes permisos: Administrador, Usuario de Nóminas, Usuario de Intervención y Usuario de Ingresos.

- El **Administrador** tiene acceso total, incluyendo la gestión de usuarios y los mantenimientos.
- El **Usuario de Nóminas** puede gestionar solicitudes, pagos, consultas y documentos.
- El **Usuario de Intervención** se centra en consultas y generación de documentación.
- El **Usuario de Ingresos** dispone de acceso restringido a la tramitación y consulta de anticipos.

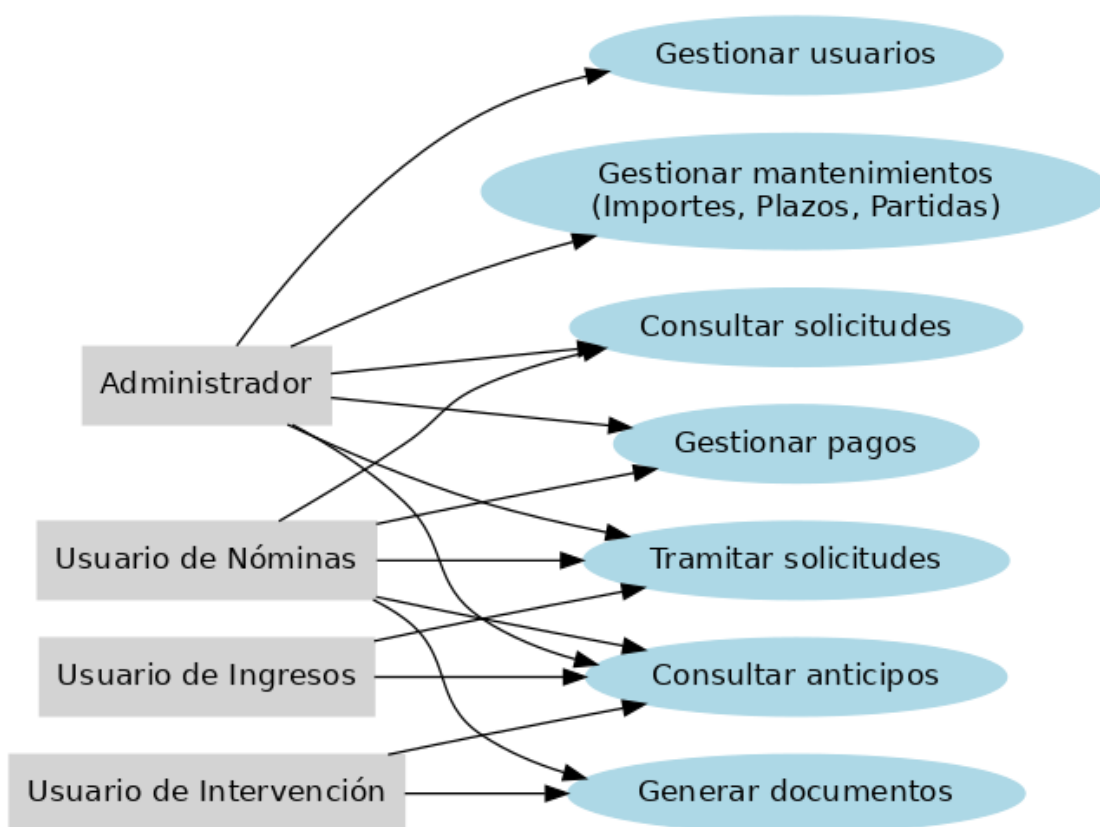


Imagen 32. Diagrama de Casos de Uso de la aplicación

3.4.2 Diagrama de Clases de Análisis

El diagrama de clases representa las entidades clave que componen el dominio del sistema:

- **Empleado:** contiene la información personal y laboral del solicitante.
- **Solicitud:** recoge los datos de cada petición de anticipo y su estado.
- **Concesión:** Contiene la información relevante a los detalles de la concesión del anticipo.
- **Desestimación:** Contiene información de cuándo y por qué se desestimó la solicitud.
- **Desestimación:** Contiene información de cuándo y por qué se desistió a la petición de la solicitud.
- **Pago:** Contiene información sobre cuando se le realizó el pago al empleado.
- **Anticipo:** se genera a partir de una solicitud concedida e incluye información de amortización.

Las relaciones entre estas entidades muestran cómo un empleado puede crear múltiples solicitudes y cómo una solicitud puede ser concedida, desestimada o desistida.

En el caso en que la solicitud se acepte se podrá proceder al pago del importe pedido y finalmente crear un anticipo para tener registrada la cantidad que le falta al cliente por devolver.

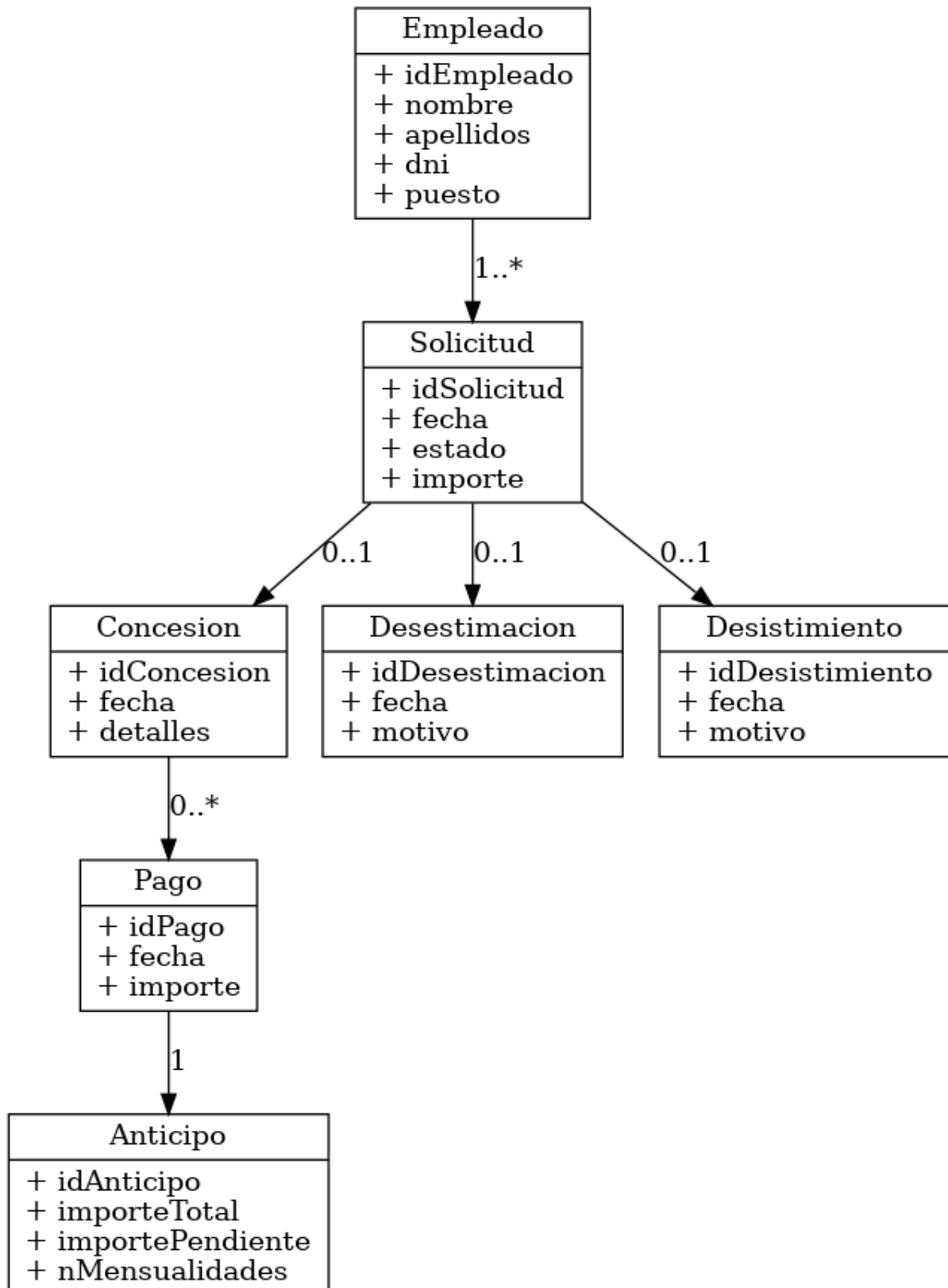


Imagen 33. Diagrama de Clases de Análisis de la aplicación

3.4.3 Diagrama de Secuencia

Para comprender el flujo dinámico del sistema, se presenta un diagrama de secuencia simplificado del proceso de solicitud:

1. El usuario da de alta una **Solicitud** desde la interfaz de usuario.
2. El sistema valida los datos y guarda la solicitud en la base de datos.
3. Posteriormente, el administrador puede crear la **Concesión**, lo que actualiza el estado de la solicitud.
4. Cuando el administrador lo decide se crea un Pago en la aplicación con lo que se actualiza el estado de la solicitud y mediante el aplicativo de contabilidad se abona la cantidad al empleado.
5. Finalmente al recibir la confirmación de que se ha procedido al pago automáticamente la aplicación crea el anticipo con los datos correspondientes.

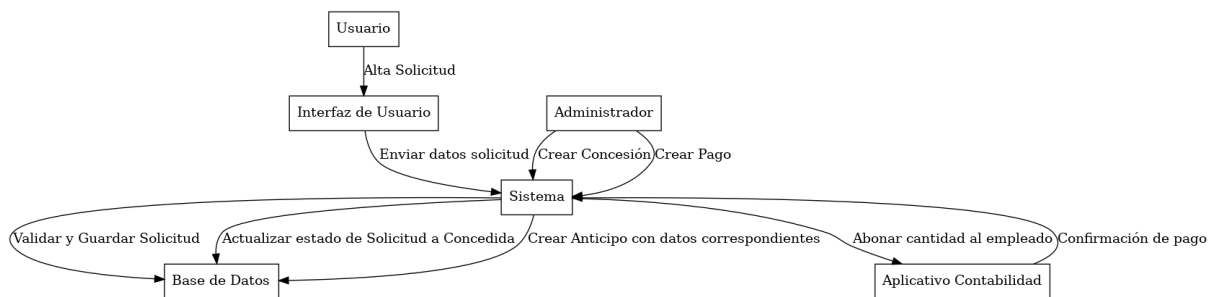


Imagen 34. Diagrama de Secuencia simplificado del flujo de solicitudes

4 Diseño

4.1 Diseño de la nueva IGU

En el Cabildo de Gran Canaria, se ha planteado la necesidad de diseñar una guía de estilos para estandarizar el diseño que van a seguir las aplicaciones, páginas web... relacionadas con el cabildo.

Con el objetivo de garantizar una experiencia de usuario coherente y eficiente, se propone un diseño de pantallas estándar que puedan ser aplicadas uniformemente a todas estas herramientas. Estas pantallas estándar servirán como base para desarrollar interfaces que se alineen con las necesidades específicas de cada módulo, proporcionando al mismo tiempo una estructura visual y funcional consistente.

A continuación, se enumeran las pantallas estándar, las funcionalidades a incluir y las propuestas de rediseño para cumplir con los objetivos de mejorar la coherencia visual y accesibilidad.

En todas las pantallas lo primero que se presenta es un breadcrumb¹⁷ que indica la ruta de navegación, mostrando que el usuario se encuentra en la pantalla de inicio. Seguido de esto, un título y subtítulo descriptivos proporcionan contexto al contenido y orientan al usuario sobre su ubicación actual y las acciones disponibles.



Imagen 35. Vista del diseño de la cabecera propuesta

¹⁷ Elemento de diseño común en las páginas web

4.1.2 Pantalla tabla

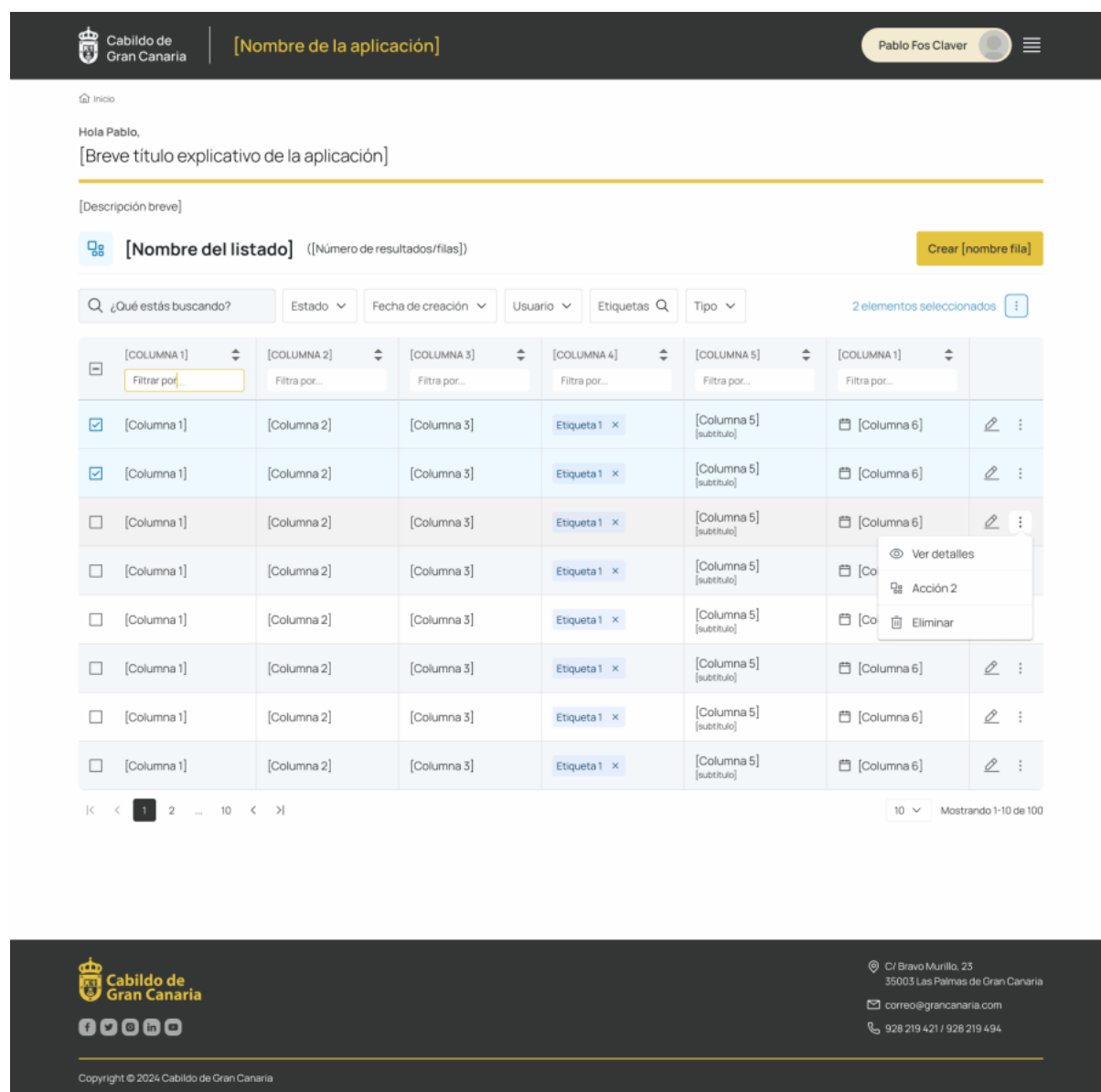


Imagen 36. Vista del diseño de la pantalla tabla propuesta

Al acceder a la pantalla tabla, lo primero que vemos es un filtro para que los usuarios puedan filtrar introduciendo datos en los campos de las cabeceras de las columnas.

Lo siguiente que podemos ver, dentro de la tabla es que cada fila es seleccionable, por lo que al seleccionar más de una fila/elemento, se pueden llevar a cabo acciones múltiples o masivas: como, por ejemplo, descargar todos los seleccionados en un CSV o eliminarlos todos de manera masiva.

Cada fila presenta distintos estados visuales para informar al usuario: el estado predeterminado (default) muestra la fila en su apariencia inicial y el estado seleccionado (selected) indica la interacción activa cuando se selecciona la fila. Ambos estados, cuentan también con un cambio de apariencia (resaltado) al hacer hover o pasar el ratón por encima.

Por otro lado, las filas pares e impares se representan con colores diferenciados para mejorar la legibilidad y facilitar la navegación por los datos.

Finalmente, la colocación estratégica de los botones de acción en la interfaz de la tabla está diseñada para optimizar la eficiencia y la claridad de uso.

El botón para crear una nueva entrada se sitúa por encima de la tabla a la derecha para facilitar su accesibilidad, ya que es una acción global que no está relacionada con ninguna fila específica. Esta ubicación asegura que los usuarios puedan iniciar el proceso de creación de datos de manera rápida y sencilla desde cualquier punto de vista de la tabla.

El botón de editar se deja visible dentro de cada fila, permitiendo un acceso inmediato y directo a la acción de edición, que suele ser la más común y relevante en la gestión de datos.

Por otro lado, el resto de las acciones secundarias como ver y borrar (o cualquier otra acción que requiera la interfaz según su casuística particular) se ubican dentro de cada fila, condensados en el icono de “tres puntos” y desplegados en su menú correspondiente, alineándose con el contexto específico de los datos a los que se aplican. Esto no solo reduce la posibilidad de errores al aplicar acciones a entradas incorrectas, sino que también proporciona un acceso directo y claro a las operaciones pertinentes sobre los datos existentes. Al asociar directamente estas acciones con las filas correspondientes, se mejora la precisión de la interacción del usuario con la tabla y se asegura que las acciones sean intencionales y específicas, mejorando así la usabilidad general de la interfaz.

4.1.3 Pantalla detalle y edición

Se pueden visualizar datos detallados que suelen provenir de campos de entrada previamente registrados. Esta vista integral muestra información exhaustiva y específica sobre un ítem o entidad seleccionada, organizada en un formato claro y accesible.

Los usuarios disfrutan de flexibilidad total al editar y guardar campos gracias a las dos visualizaciones diferenciadas: modo vista y modo edición.

Los botones principales son Cancelar, Editar (solo visible en modo vista) y Guardar (solo visible en modo edición)

Modo vista

En el modo vista, los datos se presentan de manera estática para facilitar una revisión rápida y clara.

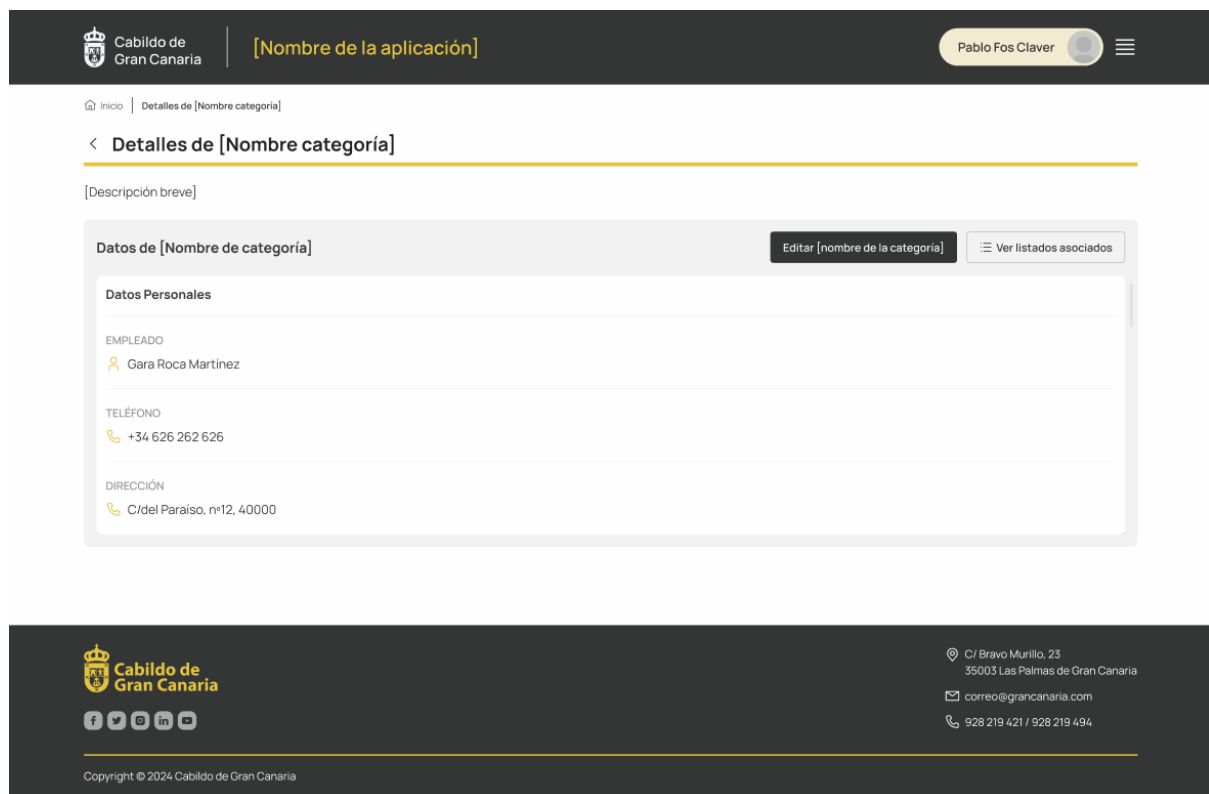


Imagen 37. Vista del diseño de la pantalla detalles en modo vista propuesta

Modo edición

Cuando se requiere realizar cambios, los usuarios pueden cambiar al modo edición, donde los campos se transforman en inputs interactivos, permitiendo la modificación directa de la información.

Cabildo de Gran Canaria | [Nombre de la aplicación] Pablo Fos Claver

[Inicio](#) | [Detalles de \[Nombre categoría\]](#)

< Detalles de [Nombre categoría]

[Descripción breve]

Datos de [Nombre de categoría] Cancelar Guardar cambios

Datos Personales

Empleado *

Teléfono *

Dirección *

[Categoría de datos]

[Label] *

Categoría de ejemplo x

Categoría de ejemplo x

Categoría de ejemplo x +2 más

Título *

Cuerpo de texto *

En el año 2024, un equipo de arqueólogos liderado por la doctora Elena Martínez hizo un descubrimiento asombroso en las remotas montañas de los Andes. Durante una expedición destinada a estudiar formaciones rocosas antiguas, tropezaron con un valle oculto que había permanecido invisible para el mundo exterior durante siglos.

El valle, al que llamaron "Valle de los Ecos", estaba rodeado por altas cumbres que lo mantenían aislado del resto del mundo. Dentro de este valle, el equipo encontró ruinas de una civilización desconocida hasta ahora. Las estructuras, construidas con una precisión sorprendente, estaban decoradas con intrincados grabados

Fecha *

Cabildo de Gran Canaria C/ Bravo Murillo, 23
35003 Las Palmas de Gran Canaria
correo@grancanaria.com
928 219 421 / 928 219 494

Copyright © 2024 Cabildo de Gran Canaria

Imagen 38. Vista del diseño de la pantalla detalles en modo edición propuesta

4.1.4 Menú navegable

Se presenta el menú navegable, siguiendo con la guía de estilo ofrecida por el Cabildo de Gran Canaria que incluye:

- Iconos representativos: Cada ítem del menú incluye un icono que facilita la identificación rápida de las opciones.
- Expandir y comprimir ítems: Los usuarios pueden expandir o comprimir ítems en el menú, permitiendo una navegación organizada y limpia.
- El menú está diseñado con estados default, hover y active que se pueden consultar en el diseño.

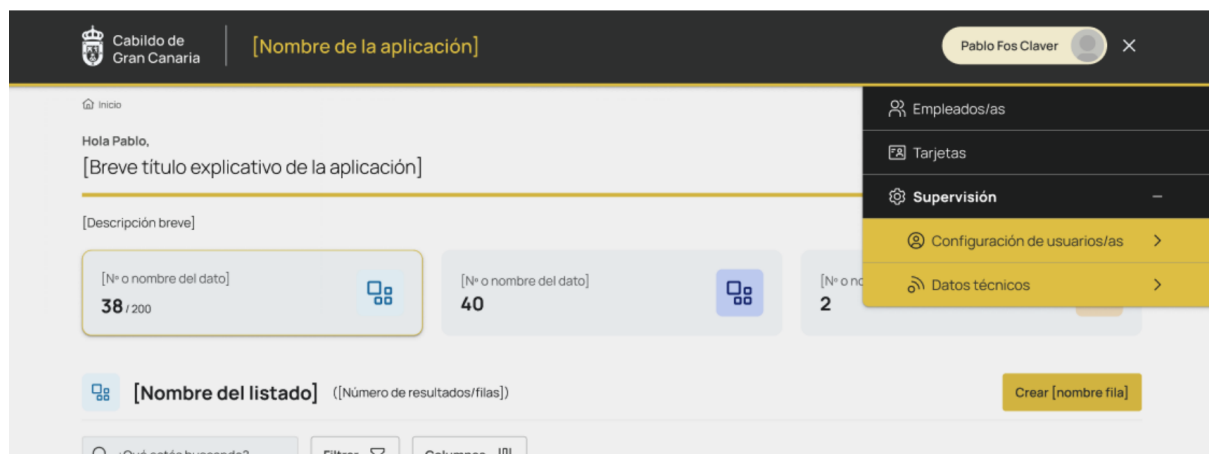


Imagen 39. Vista del diseño del menú navegable propuesto

4.1.5 Alertas y confirmaciones

Cuando un usuario intenta eliminar un elemento, se presenta un mensaje de confirmación solicitando confirmación de la acción. Esto ayuda a prevenir eliminaciones accidentales y asegura que los usuarios puedan revisar y confirmar su intención antes de proceder.

Tras la ejecución de la eliminación, se muestra un mensaje de éxito o fracaso dependiendo del resultado de la operación, proporcionando una retroalimentación inmediata sobre la acción realizada y permitiendo al usuario entender si la eliminación se llevó a cabo correctamente o si hubo algún problema.

Además, para reducir la probabilidad de errores y asegurar que las decisiones importantes se tomen con plena consciencia, el botón de confirmar la acción se ha diseñado como un botón secundario.

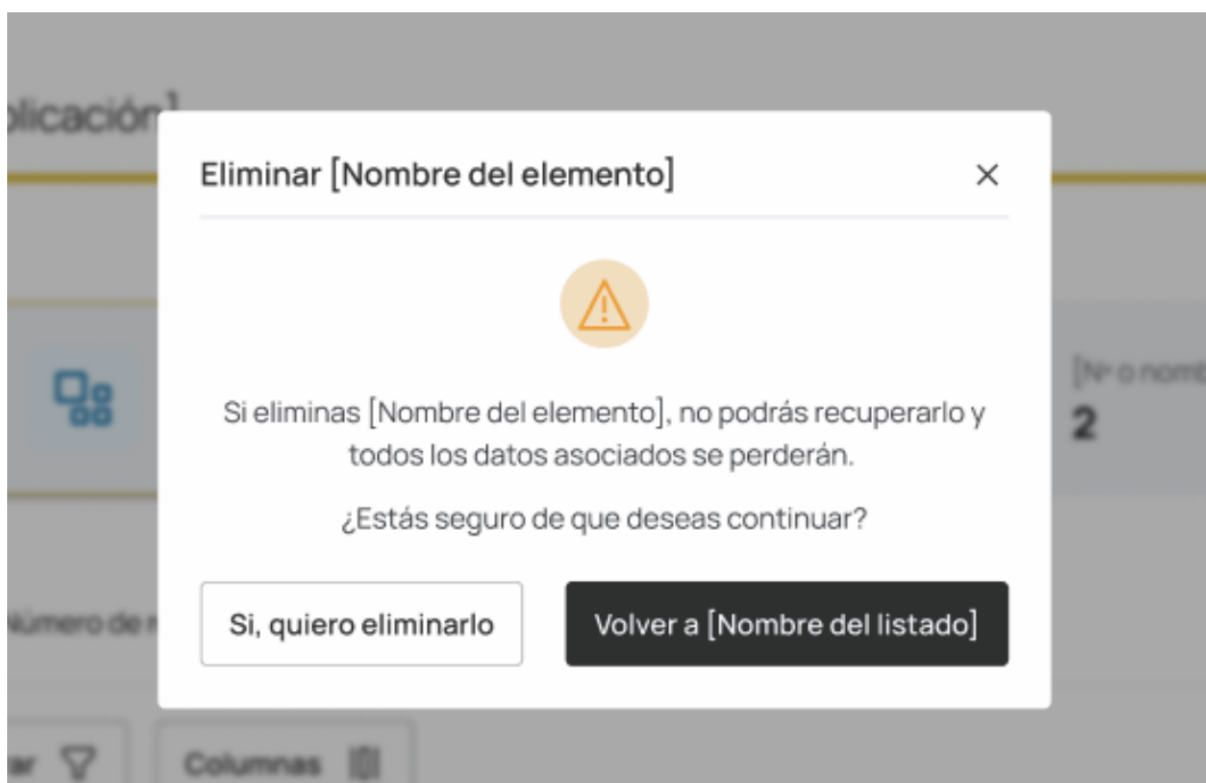


Imagen 40. Vista del diseño de las ventanas de alertas propuestas

4.1.6 Empty space

Para cuando la tabla está vacía o los resultados de la búsqueda no arrojan datos, se ha diseñado un “empty space”.

Este espacio vacío no solo indica que no hay datos disponibles, sino que también proporciona mensajes o sugerencias para guiar al usuario en cómo proceder. Por ejemplo, puede incluir un mensaje informativo como "No se encontraron resultados" o "La tabla está vacía", así como recomendaciones sobre posibles acciones a tomar, como ajustar los filtros de búsqueda, añadir nuevos datos o verificar la consulta realizada.

Este diseño ayuda a reducir la confusión al asegurar que el usuario comprenda la razón por la cual no se están mostrando datos y ofrece orientación sobre cómo solucionar o explorar la situación, mejorando así la usabilidad y la eficacia general de la interfaz.



Imagen 41. Vista del diseño del empty space de las tablas propuesto

4.2 Arquitectura

Una de las mejoras más significativas introducidas en el desarrollo de la nueva aplicación de Gestión de Anticipos de Nómina ha sido la definición de una estructura clara y organizada en el código. Esto permite que el desarrollo sea mucho más ordenado con el objetivo de que en cualquier momento sea más intuitivo saber donde empezar a programar a la hora de implementar una nueva funcionalidad o arreglar un posible error.

4.2.1 Arquitectura actual: Oracle Forms

La aplicación original está desarrollada en Oracle Forms (Oracle Forms, 2025), una tecnología que permite construir interfaces gráficas basadas directamente en formularios conectados a bases de datos de manera muy sencilla e intuitiva pero que con el tiempo se ha quedado obsoleta. Se trata de una arquitectura monolítica, en la que la lógica de negocio, la interfaz de usuario y el acceso a las bases de datos están altamente acoplados en un único bloque.

Esto implica varias limitaciones:

- **Alta dependencia de la base de datos.** Gran parte de la lógica se encuentra implementada directamente en vistas de bases de datos o dentro de los formularios y no en el código.
- **Difícil mantenimiento y escalabilidad.** Para realizar cualquier cambio o nueva implementación se debe modificar el bloque común, lo cual deja mucho espacio para errores y complica el mantenimiento.
- **Difícil despliegue de módulos separados.** Ya que no es posible separar la aplicación por módulos no se pueden desplegar de manera independiente, lo cual dificulta la escalabilidad.
- **Dependencia del navegador Internet Explorer.** Al estar diseñada para ejecutarse únicamente sobre este navegador, la estructura presenta incompatibilidades con otros entornos modernos.

4.2.2 Nueva arquitectura: Aplicación multicapa

La nueva aplicación sigue una arquitectura de capas MVC, para separar las responsabilidades y facilitar tanto el mantenimiento como la evolución futura del sistema. Se han definido las siguientes capas:

Capa modelo

La capa modelo es la encargada del acceso a los datos que hay en la base de datos. Se utiliza JPA (Spring JPA, 2025), una API para mapear las entidades en objetos Java y gestionar operaciones CRUD entre otras.

Capa vista

La capa vista es la que gestiona lo visible para el usuario final, implementada con los frameworks JSF y Primefaces. Está constituida por formularios y mensajes de validación que permiten realizar acciones como el alta, edición o listado de objetos de manera visual. Los formularios de la nueva aplicación se han diseñado con Figma y con un diseño enfocado en mejorar la experiencia de usuario.

Capa controlador

La capa controlador es la encargada de la lógica de la aplicación, desarrollada con Java Spring Boot (Spring Data Team, 2022). El objetivo es coordinar la capa de datos con la capa de presentación e implementar si son necesarias las reglas de negocio.

4.3 Tecnologías a utilizar

4.3.1 Java Spring Boot

Con el objetivo de facilitar la programación en Java se ha utilizado Java Spring Boot como framework. Ayuda a trabajar rápido y mantener el código más ordenado gracias a configuraciones como anotaciones que ahorran muchas líneas de código.

4.3.2 JPA

Java Persistence API permite manejar los datos como objetos y dejar que JPA se encargue de comunicarse con la base de datos para agilizar la programación en relación con los datos (Spring JPA, 2025).

4.3.3 Maven

Es una herramienta que tiene como objetivo gestionar las dependencias y compilar, empaquetar y ejecutar el código de forma ordenada.

4.3.4 Primefaces

Primefaces es una librería que ayuda a construir interfaces gráficas web funcionales de manera sencilla. Proporciona muchos componentes como tablas, formularios o botones ya creados que se integran fácil en Java.

4.3.5 WildFly

Es un servidor de aplicaciones que permite desplegar y ejecutar aplicaciones Java empresariales. Además de para el desarrollo en Läberit se utiliza para desplegar las aplicaciones en entornos de preproducción y producción.

4.3.6 SQL Developer

SQL Developer es una herramienta de Oracle que facilita mucho el trabajo relacionado con bases de datos gracias a su interfaz visual.

4.3.7 JIRA

El objetivo principal de JIRA es gestionar proyectos y tareas con metodologías ágiles, ya que permite crear tickets, organizarlos en sprints y asignarlos a personas de un equipo. En el caso de Läberit JIRA se utiliza como medio de comunicación con el cliente, en este caso el Cabildo de Gran Canaria.

4.3.8 Figma

Es una herramienta online que permite diseñar interfaces. Se utiliza para crear prototipos y maquetas de cómo se verá la aplicación antes de crearla.

4.3.9 Redmine

Redmine es una herramienta de gestión de proyectos que se utiliza en Läberit para imputar las horas que se trabaja en cada ticket para facilitar la gestión a los jefes de proyecto. También se puede utilizar para tener registradas las vacaciones de cada miembro del equipo.

4.3.10 Trello

El objetivo de Trello es organizar de manera muy visual y sencilla tareas mediante tableros y tarjetas. En el equipo se utiliza para exponer el trabajo actual que estás haciendo y crear una cola de tareas.

4.3.11 Teams

Teams es una de las herramientas de comunicación y colaboración más conocidas actualmente en el ámbito educativo y profesional. Permite chatear, hacer llamadas, reuniones, enviar archivos dentro del chat... Es muy útil para realizar reuniones de seguimiento semanales para conocer qué tarea está haciendo cada miembro del equipo.

4.3.12 IntelliJ IDEA

Es un entorno de desarrollo muy completo para programar en Java y otros lenguajes. Ayuda con el autocompletado, depuración, gestión de proyectos y un montón de herramientas integradas para hacer el trabajo más rápido.

5 Desarrollo

En este apartado cuento el proceso de construcción de la nueva aplicación de gestión de anticipos de nómina. Para organizar el trabajo, seguí una metodología ágil con sprints de 4 semanas, lo que me permitió avanzar de forma progresiva y tener siempre algo funcional al final de cada ciclo. A continuación, detallo lo que hice en cada sprint y cómo fue tomando forma la aplicación.

5.1 Sprint 1

5.1.1 Preparación y bases del proyecto

Al inicio de un proyecto se debe clonar la base de datos a utilizar en local para poder conectar el proyecto a esta. Esto lo hice a través de un archivo SQL que nos enviaron desde el Cabildo de Gran Canaria.

```
-----
-- DDL for Table GDAN_ANTICIPO
-----

CREATE TABLE "ANTICIPOS"."GDAN_ANTICIPO"
(
  "ID_ANTICIPO" NUMBER,
  "ID_EMPRESA" VARCHAR2(40 CHAR) DEFAULT NULL,
  "ID_TRABAJADOR" VARCHAR2(8 CHAR) DEFAULT NULL,
  "ANT_ACTIVO" CHAR(1 CHAR) DEFAULT 'S',
  "IMP_TOTAL" NUMBER DEFAULT 0,
  "MENSUALIDADES" NUMBER DEFAULT 0,
  "FEC_INICIO" DATE DEFAULT to_date('01/01/2008','dd/mm/yyyy'),
  "FEC_BAJA" DATE,
  "REF_EXPED" VARCHAR2(64 CHAR) DEFAULT '00/00',
  "ID_COMENTA" NUMBER DEFAULT NULL,
  "ID_CONTAB" VARCHAR2(20 BYTE),
  "ID_RRHH" NUMBER,
  "COMENTARIO" VARCHAR2(512 CHAR),
  "CODIGO_EMPRESA" VARCHAR2(3 CHAR),
  "INC_NOMINAS" VARCHAR2(1 CHAR),
  "IMP_RESTANTE" NUMBER,
  "CONGELADO_DTO" NUMBER DEFAULT 0,
  "MENSUALIDAD_MEDIA" NUMBER DEFAULT 0,
  "MENSUALIDADES_DESCONTADAS" NUMBER DEFAULT 0,
  "ID_SOLICITUD" NUMBER
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "GENERAL" ;

-----
-- DDL for Table GDAN_CALCULOS_DTOS_MES
-----

CREATE TABLE "ANTICIPOS"."GDAN_CALCULOS_DTOS_MES"
(
  "ID_CALCULOS" NUMBER,
  "ANNO" NUMBER,
  "MES" NUMBER,
  "EJECUTADO" NUMBER,
```

Imagen 42. Archivo SQL de importación inicial

Después de esto, creé el proyecto a partir de un arquetipo general que posee algunos logos, imágenes y estructuras de headers y footers ya hechas para seguir un estándar.

Finalmente para acabar la preparación configuré mi IDE, IntelliJ IDEA, seleccioné las versiones correspondientes de Java y Maven y creé un servidor local de WildFly para lanzar la aplicación. Una vez montado todo, hice algunas pruebas para comprobar que la base de datos y la aplicación estaban conectadas extrayendo datos de algún anticipo y mostrándolos en la aplicación.

5.1.2 Backend inicial y reglas básicas

Una vez configurado el IDE empecé a crear las clases relacionadas con el backend (Repositorios, DTO's, Mappers, Entidades, Servicios...) y añadiendo las anotaciones correspondientes.

```
package com.grancanaria.model;

import ...

@Entity @vgoomez+1
@Table(name = "GDAN_ANTICIPO")
@Getter
@Setter
@ToString
@AllArgsConstructor
@NoArgsConstructor
public class Anticipo implements Serializable {

    @Id
    @SequenceGenerator(name = "NGAnticipoGen", allocationSize = 1, sequenceName = "SEQ_GDAN_ANTICIPO")
    @GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "NGAnticipoGen")
    @NotNull
    @Column(name = "ID_ANTICIPO")
    private Long idAnticipo;

    @NotNull
    @Column(name = "ID_EMPRESA")
    private String idEmpresa;
}
```

Imagen 43. Entidad de anticipo

```
@Service @vgoomez*
@Transactional
public class AnticipoServiceImpl implements AnticipoService {

    private static final Logger LOG = LoggerFactory.getLogger(AnticipoServiceImpl.class); 5 usages

    @Autowired
    private AnticipoRepository anticipoRepository;
    @Autowired
    private EmpleadoService empleadoService;
    @Autowired
    private EmpleadoMapper empleadoMapper;
    @Autowired
    private AnticipoMapper mapper;
    @Autowired
    private NG_SolicitudMapper solicitudMapper;

    @Override 1 usage @vgoomez
    public Optional<AnticipoDTO> getAnticipoDtoByIdOptional(Long idAnticipo) {
        Anticipo anticipo = anticipoRepository.findById(idAnticipo)
            .orElseThrow(AnticipoNotFoundException::new);

        if (anticipo == null) {
            return Optional.of(new AnticipoDTO());
        }

        return Optional.ofNullable(mapper.toDto(anticipo));
    }
}
```

Imagen 44. Servicio de anticipo

```

package com.grancanaria.mapper;

import ...

@Component
public class AnticipoMapperImpl implements AnticipoMapper {

    @Override
    public AnticipoDTO toDto(Anticipo entity) {
        if ( entity == null ) {
            return null;
        }

        AnticipoDTObuilder anticipoDTO = AnticipoDTO.builder();

        anticipoDTO.idAnticipo( entity.getIdAnticipo() );
        anticipoDTO.idEmpresa( entity.getIdEmpresa() );
        anticipoDTO.idTrabajador( entity.getIdTrabajador() );
        anticipoDTO.activo( entity.getActivo() );
        anticipoDTO.importeTotal( entity.getImporteTotal() );
        anticipoDTO.mensualidades( entity.getMensualidades() );
        anticipoDTO.fecInicio( entity.getFecInicio() );
        anticipoDTO.fecBaja( entity.getFecBaja() );
        anticipoDTO.refExped( entity.getRefExped() );
        anticipoDTO.idComenta( entity.getIdComenta() );
        anticipoDTO.idContab( entity.getIdContab() );
        anticipoDTO.idRRHH( entity.getIdRRHH() );
        anticipoDTO.comentario( entity.getComentario() );
        anticipoDTO.codEmpresa( entity.getCodEmpresa() );
        anticipoDTO.incNominas( entity.getIncNominas() );
    }
}

```

Imagen 45. Mapper de anticipo

```

package com.grancanaria.dto;

import ...

@Data
@Builder
@NoArgsConstructor
@AllArgsConstructor
public class AnticipoDTO implements Serializable {
    private Long idAnticipo;
    private String idEmpresa;
    private String idTrabajador;
    private Boolean activo;
    private Double importeTotal;
    private Long mensualidades;
    private Date fecInicio;
    private Date fecBaja;
    private String refExped;
    private Long idComenta;
    private String idContab;
    private Long idRRHH;
    private String comentario;
    private String codEmpresa;
    private String incNominas;
    private Double impRestante;
    private Boolean congeladoDTO;
    private Double impMensualidadMedia;
}

```

Imagen 46. DTO de anticipo

```

package com.grancanaria.repository;

import ...

public class AnticipoRepositoryCustomImpl extends QuerydslRepositorySupport implements AnticipoRepositoryCustom {

    public AnticipoRepositoryCustomImpl() { super(Anticipo.class); }

    @Override
    public Page<Anticipo> findAnticipo(int page, int size, AnticipoFilter filtro, Map<String, Boolean> sortColumns) {
        QAnticipo qAnticipo = QAnticipo.anticipo;

        JPQLQuery<Anticipo> query = from(qAnticipo);
        BooleanBuilder where = new BooleanBuilder();

        if (filtro.isNombreCompleto()) {
            where.and(qAnticipo.solicitud.empleadoSol.nombre.containsIgnoreCase(filtro.getNombreCompleto())
                .or(qAnticipo.solicitud.empleadoSol.apellido1.containsIgnoreCase(filtro.getNombreCompleto())
                .or(qAnticipo.solicitud.empleadoSol.apellido2.containsIgnoreCase(filtro.getNombreCompleto())));
        }

        if (filtro.isImporteTotal()) {
            where.and(qAnticipo.importeTotal.stringValue().containsIgnoreCase(filtro.getImporteTotal()));
        }

        if (filtro.isImpRestante()) {
            where.and(qAnticipo.impRestante.stringValue().containsIgnoreCase(filtro.getImpRestante()));
        }

        if (filtro.isImpMensualidadMedia()) {
            where.and(qAnticipo.impMensualidadMedia.stringValue().containsIgnoreCase(filtro.getImpMensualidadMedia()));
        }
    }
}

```

Imagen 47. Repositorio de anticipo

5.2 Sprint 2

5.2.1 Prueba de conexión entre frontend y backend

Una vez preparado todo el backend creé una vista de prueba para comprobar que todo se está conectando correctamente, tanto la base de datos con el backend como el backend con el frontend.

Todo esto se implementa muy fácilmente gracias a la anotación “Autowired” de Java Spring Boot (Spring Data Team, 2022) que lo que hace es crear una instancia de una clase para utilizar sus métodos, normalmente lo utilizo para instanciar Servicios o Repositorios.

```
package com.grancanaria.controller;

import ...

@Controller @v Gomez *
@ViewScoped
public class AnticipoEditarController implements Serializable {

    //Inyección de componentes
    @Autowired
    private AnticipoService anticipoService;

    /**
     * Inicialización del controlador
     */

    public void init() { new *
        ExternalContext externalContext = FacesContext.getCurrentInstance().getExternalContext();
        String idAnticipo = externalContext.getRequestParameterMap().get("idAnticipo");
        anticipoDTO = anticipoService.getAnticipoDtoById(Long.valueOf(idAnticipo));
    }

    // Seguimiento de Loggs
    private static final Logger LOG = LoggerFactory.getLogger(AnticipoEditarController.class); no usages

    //Literales
    private static final String ANTICIPO_URL = "/ui/anticipo.xhtml?faces-redirect=true"; 2 usages

    private AnticipoDTO anticipoDTO = new AnticipoDTO(); 16 usages
```

Imagen 48. Anotación Autowired y como se utiliza

5.2.2 Header y footer

Después de resolver los errores que no me permitían conectar el backend con el frontend empecé creando las primeras vistas en HTML. Lo primero que tuve que programar fue el header y el footer que estarán presentes en toda la aplicación, el arquetipo ya tiene la estructura pero hay detalles propios de la aplicación que hay que cambiar como el nombre de la aplicación, la versión...



Imagen 49. Aplicación con el arquetipo inicial

5.2.3 Sistema de autenticación

Para implementar el inicio de sesión en la aplicación utilizamos el servicio CAS¹⁸ que es una solución de código abierto que permite a los usuarios iniciar sesión una sola vez para acceder a las múltiples aplicaciones web del CGC.

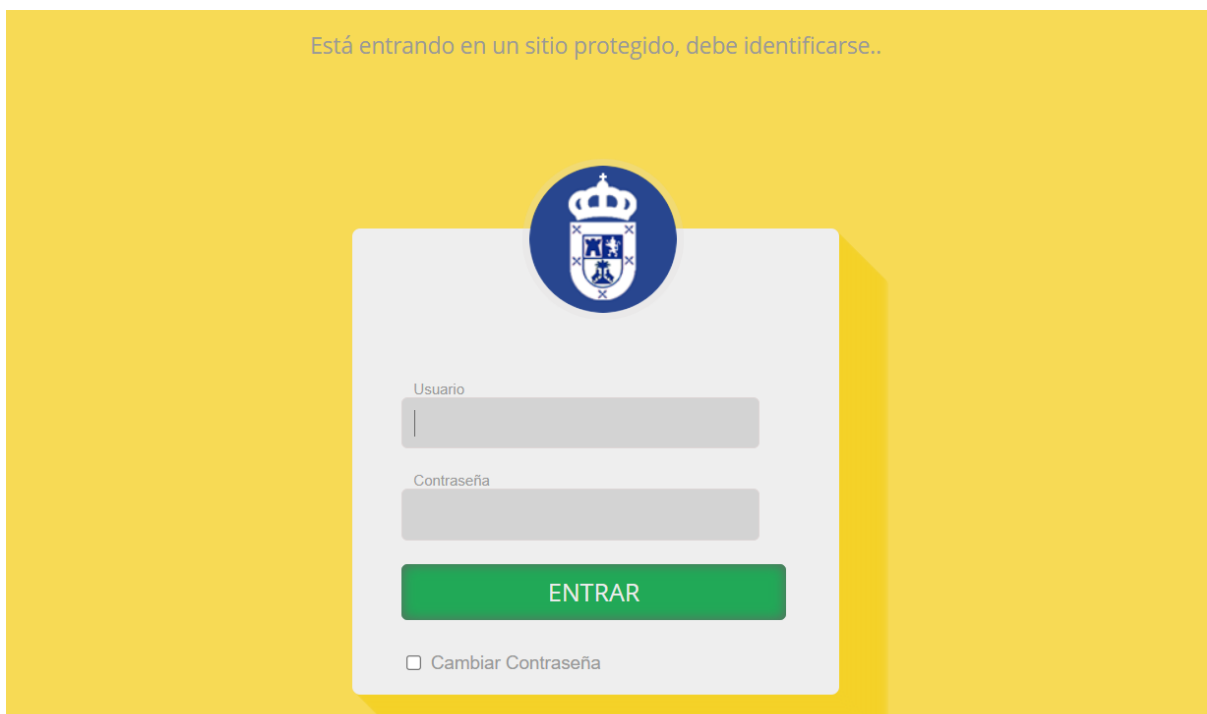


Imagen 50. Pantalla de inicio de sesión del CAS

¹⁸ <https://docs.spring.io/spring-security/reference/servlet/authentication/cas.html>

Además del inicio de sesión en la configuración del CAS tuve que especificar las vistas a las que podía acceder cada rol utilizando el siguiente método.

```
/**
 * Filtrado de acceso a páginas y autorización en función del rol del usuario
 */
@Override
protected void configure(HttpSecurity http) throws Exception {
    http.addFilterBefore(requestLogoutFilter(), LogoutFilter.class);
    http.addFilterBefore(singleSignOutFilter(), CasAuthenticationFilter.class);
    http.authorizeRequests() ExpressionInterceptUrlRegistry
        .antMatchers("@"/" AuthorizedUrl
        .permitAll() ExpressionInterceptUrlRegistry
        .antMatchers("@"/login", "@"/ui/inicio.xhtml") AuthorizedUrl
        .authenticated() ExpressionInterceptUrlRegistry
        .antMatchers("@"/ui/anticipo.xhtml", "@"/ui/anticipoAlta.xhtml", "@"/ui/anticipoEditar.xhtml",
            "@"/ui/anulacion.xhtml", "@"/ui/anulacionAlta.xhtml", "@"/ui/anulacionEditar.xhtml",
            "@"/ui/calculo.xhtml", "@"/ui/calculoAlta.xhtml", "@"/ui/calculoEditar.xhtml",
            "@"/ui/concesion.xhtml", "@"/ui/concesionAlta.xhtml", "@"/ui/concesionEditar.xhtml",
            "@"/ui/importe.xhtml", "@"/ui/importeAlta.xhtml", "@"/ui/importeEditar.xhtml",
            "@"/ui/pago.xhtml", "@"/ui/pagoAlta.xhtml", "@"/ui/pagoEditar.xhtml",
            "@"/ui/partida.xhtml", "@"/ui/partidaAlta.xhtml", "@"/ui/partidaEditar.xhtml",
            "@"/ui/plazo.xhtml", "@"/ui/plazoAlta.xhtml", "@"/ui/plazoEditar.xhtml",
            "@"/ui/solicitud.xhtml", "@"/ui/solicitudAlta.xhtml", "@"/ui/solicitudEditar.xhtml",
            "@"/ui/usuarios.xhtml", "@"/ui/EditarUsuarios.xhtml", "@"/ui/DetalleUsuarios.xhtml") AuthorizedUrl
        .hasAnyRole(roleAdministrador, roleIngresos, roleIntervencion, roleNominas, roleAdministrador) ExpressionInterceptUrlRegistry
        .antMatchers("@"/monitoring") AuthorizedUrl
        .hasAnyRole(roleAdministrador) ExpressionInterceptUrlRegistry
        .and() HttpSecurity
        .exceptionHandling() ExceptionHandlingConfigurer<HttpSecurity>
        .authenticationEntryPoint(casAuthenticationEntryPoint())
        .and() HttpSecurity
        .csrf().disable();
}
```

Imagen 51. Método que gestiona los permisos.

5.3 Sprint 3

5.3.1 Primeras vistas

A partir de este sprint es cuando empecé a implementar las vistas de los primeros mantenimientos, para empezar me estructuré el análisis de manera que implementaría primero las pantallas más sencillas como los listados de partidas presupuestarias, importes y plazos.

El framework Spring Boot (Spring Data Team, 2022) nos proporciona un objeto llamado *LazyDataModel* ideal para construir los listados de objetos que incluyen funciones como el propio listado, orden ascendente o descendente y filtrado.

Todo esto lo programo obteniendo el listado de base de datos y creando un filtro específico para cada objeto que le paso como parámetro.

```
public LazyDataModel<NG_PartidaDTO> getPartidaLazyDataModel() { 1 usage @vgomez+
    if (partidaLazyDataModel == null) {
        this.partidaLazyDataModel = new LazyDataModel<NG_PartidaDTO>() { @vgomez+

        @Override @vgomez+
        public List<NG_PartidaDTO> load(int first, int pageSize, Map<String, SortMeta> sortBy, Map<String, FilterMeta> filterBy) {

            NG_PartidaFilter filtros = createFilter(filterBy);
            Page<NG_Partida> page = service.getPartidaPage(
                page: first / pageSize, pageSize, filtros, createOrderColumns(sortBy)
            );

            setRowCount((int) page.getTotalElements());
            setPartidaFiltrado(service.getPartidaFiltrados(filtros, createOrderColumns(sortBy)));
            return mapper.toDto(page.getContent());
        }

        @Override @vgomez
        public String getRowKey(NG_PartidaDTO object) { return String.valueOf(object.getIdPartida()); }

        @Override @vgomez+
        public NG_PartidaDTO getRowData(String rowKey) {
            List<NG_PartidaDTO> data = getWrappedData();
            NG_PartidaDTO result = null;

            if (data != null) {
                Optional<NG_PartidaDTO> rowData = data.stream()
                    .filter( NG_PartidaDTO obj -> Integer.toString(Math.toIntExact(obj.getIdPartida())) .equals(rowKey)) .findAny();
                result = rowData.orElseGet(() -> service.getPartidaByIdOptional(Integer.parseInt(rowKey)).orElseThrow(NG_PartidaNotFoundException::new));
            }
            return result;
        }
    }
}
```

Imagen 52. Método que crea el *LazyDataModel*

En estas 4 semanas de sprint me dio tiempo a implementar las tres pantallas de listado junto con el alta y edición de partidas presupuestarias y el de plazos.

< Partidas

Edita, organiza y administra las partidas que hay almacenados en el sistema.

Listado partidas

Crear partida

R. Jurídico	Fecha Desde	Fecha Hasta	P.P Económica	C.I Económica	Orgánica	
Partida01	07/11/2024	08/11/2024	Eco01	Ecoln01	Org01	  
Partida02	01/12/2024		Eco02	Ecoln02	Org02	  
rr	18/12/2024	02/12/2024	aa	dd	rr	  
aa	02/12/2024	17/12/2024	dsdsasds	hfhfhfh	sasasa	  
dsd	08/12/2024	24/12/2024	dsd	gdfhgdh	dsds	  

1 2 3 >

5Mostrando 1 - 5 de 13

Imagen 53. Vista de listado de partidas

Cabildo de Gran Canaria

Gestion de Anticipos de Nóminas

PABLO FOS CLAVER





Inicio

Partida

Alta Partida

< Alta Partida

Da de alta una nueva partida y la guarda en el sistema.

Datos Partida

Guardar

Cancelar

Datos

R. Jurídico *

Seleccione...

Fecha Desde *

Fecha Hasta

Orgánica *

Programa

Económica *

Imagen 54. Vista de alta de partidas

5.4 Sprint 4

5.4.1 Vistas de mantenimientos

Durante este sprint me dediqué a acabar las funcionalidades de alta y edición de los importes los cuales tenían un poco más de complejidad debido a las condiciones esperadas en el requisito funcional [REQ 016](#).

```
/**
 * Métodos para controlar los valores de plazo e importe
 */
private Integer getMesesforRange(double x, double import1, double import2, double import3, double maximo, N6_Plazo plazo) {
    if (x >= import1 && x < import2) {
        return Integer.valueOf(plazo.getMeses1()); // Rango 1
    } else if (x >= import2 && x < import3) {
        return Integer.valueOf(plazo.getMeses2()); // Rango 2
    } else if (x >= import3 && x <= maximo) {
        return Integer.valueOf(plazo.getMeses3()); // Rango 3
    } else {
        return null;
    }
}

private Integer getRangeImporte(double x, double import1, double import2, double import3, double maximo) { 1usage 2vgomez
    if (x >= import1 && x < import2) {
        return 1; // Rango 1
    } else if (x >= import2 && x < import3) {
        return 2; // Rango 2
    } else if (x >= import3 && x <= maximo) {
        return 3; // Rango 3
    } else {
        return null;
    }
}
```

Imagen 55. Métodos que gestionan los valores de importe/plaza

5.4.2 Flujo de solicitudes

Al acabar esta parte empecé con el flujo de las solicitudes, lo primero fue crear el listado, el alta y la edición de las solicitudes, lo cual fue sencillo ya que lo único que lo distinguía de un mantenimiento era el número de parámetros del objeto.

Da de alta una nueva solicitud y la guarda en el sistema.

Datos Solicitud

Datos

Empleado*

Fecha Solicitud*

Número Registro

Importe*

Meses*

Guardar Cancelar

Completa los campos obligatorios antes de guardar

Imagen 56. Vista alta solicitud

Para acabar el sprint empecé a implementar el siguiente paso del flujo, los listados de concesiones, desestimaciones y desistimientos de una solicitud. Estos listados fueron más complejos porque tuve que relacionar las concesiones, desestimaciones y desistimientos con su solicitud relacionada para obtener información como los datos del empleado.

Cabildo de Gran Canaria | **Gestion de Anticipos de Nóminas** | PABLO FOS CLAVER

[Inicio](#) | [Concesiones](#)

< Concesiones

Edita, organiza y administra las concesiones que hay almacenados en el sistema.

Listado concesiones Crear concesión

Id Resolución	Número Resolución	Fecha Resolución	
261	9999	09/06/2025	✎ 👁 🗑
289	1000/2025	10/06/2025	✎ 👁 🗑
264	1000/2025	10/06/2025	✎ 👁 🗑

Mostrando 1 - 3 de 3

Cabildo de Gran Canaria | **EICAN** | FONDO DE DESARROLLO DE CANARIAS | Gobierno de Canarias

© C/ Bravo Murillo, 25 50035 Las Palmas de Gran Canaria
oscar@canariacanal.com
 928 219 229

Imagen 57. Vista listado de concesiones

5.5 Sprint 5

5.5.1 Última parte del flujo de solicitudes

Debido a que este sprint era el penúltimo para llegar a tiempo a la fecha de entrega acordada con cabildo me propuse acabar el flujo de la solicitud. Implementé las pantallas de alta y edición de las concesiones, desestimaciones y desistimientos.

Cabildo de Gran Canaria | **Gestion de Anticipos de Nóminas** | PABLO FOS CLAVER

[Inicio](#) | [Desestimación](#) | [Alta Desestimación](#)

< Alta Desestimación

Da de alta una nueva desestimación y la guarda en el sistema.

Guardar Cancelar

Datos Resolución

Número Resolución *

Fecha resolución *

Observaciones Resolución

Selección de solicitud *

	Empleado	Fecha Solicitud	Importe	Meses
☐	EMILIO HERNANDEZ PEREZ	13/12/2024	1.200,00	12
☐	LUIS RAMIREZ GONZALEZ	13/12/2024	1.200,00	12
☐	ROSA VEGA SANCHEZ	13/12/2024	1.200,00	2
☐	ANTONIO TOMAS CABRERA SANTANA	13/12/2024	1.200,00	2
☐	ANA ACOSTA JORGE	04/03/2025	110,00	15

Mostrando 1 - 5 de 9

Cabildo de Gran Canaria | **EICAN** | FONDO DE DESARROLLO DE CANARIAS | Gobierno de Canarias

© C/ Bravo Murillo, 25 50035 Las Palmas de Gran Canaria
oscar@canariacanal.com
 928 219 229

Imagen 58. Vista de alta de desestimación

Para acabar el flujo del anticipo me faltaba el último paso, dar de alta el anticipo. Debido a que el anticipo debe crearse cuando se realiza el pago al usuario, implementé un alta de pago con la información necesaria para dar de alta un anticipo en base de datos y cambiar el estado de la solicitud a *Reintegrando*.

Datos Pagos de Anticipo

Guardar Cancelar

Completa los campos obligatorios antes de guardar

Datos

Selección de Solicitudes Concedidas*

	Empleado	Fecha Resolución	Num. Resolución	Importe	Meses
☐	ACOSTA GONZALEZ JERONIMO	10/06/2025	1000/2025	122,00	12

1 < > 5 Mostrando 1 - 1 de 1

Operación Pago*

Fecha Pago*

Imagen 59. Vista del alta de pago de anticipo

Finalmente para implementar el listado de anticipos solo tenía que obtener los datos de la base de datos y crear con esto el listado.

Anticipos

Edita, organiza y administra los anticipos que hay almacenados en el sistema.

Listado anticipos

Empleado	Importe Total	Importe Restante	Mensualidad	Meses	Activo	Congelado	
ALICIA ESTHER BENITEZ SANTANA	1.200,00	1.000,00	600	2	Si	No	
GUACIMARA ALVAREZ SOCORRO	1.800,00	1.800,00	150	12	Si	No	
ROBERTO LUIS YANEZ YANEZ	150,00	150,00	38	4	Si	No	
YOLANDA AGUIAR VERA	2.200,00	2.200,00	183	12	Si	No	

1 < > 5 Mostrando 1 - 4 de 4

Cabildo de Gran Canaria **FICAN** FONDO DE DESARROLLO DE CANARIAS **Gobierno de Canarias**

C/ Bravo Murillo, 23 35003 Las Palmas de Gran Canaria
 info@icajegrancanaria.com
 928 219 229

Imagen 60. Vista del listado anticipos

5.6 Sprint 6

5.6.1 Generación de documentos

Empezando este sprint hice un repaso al análisis y vi que lo único que le quedaba a la aplicación para ser lanzada en preproducción era la generación de documentos, por lo tanto empecé a implementar un botón para generar un documento excel.

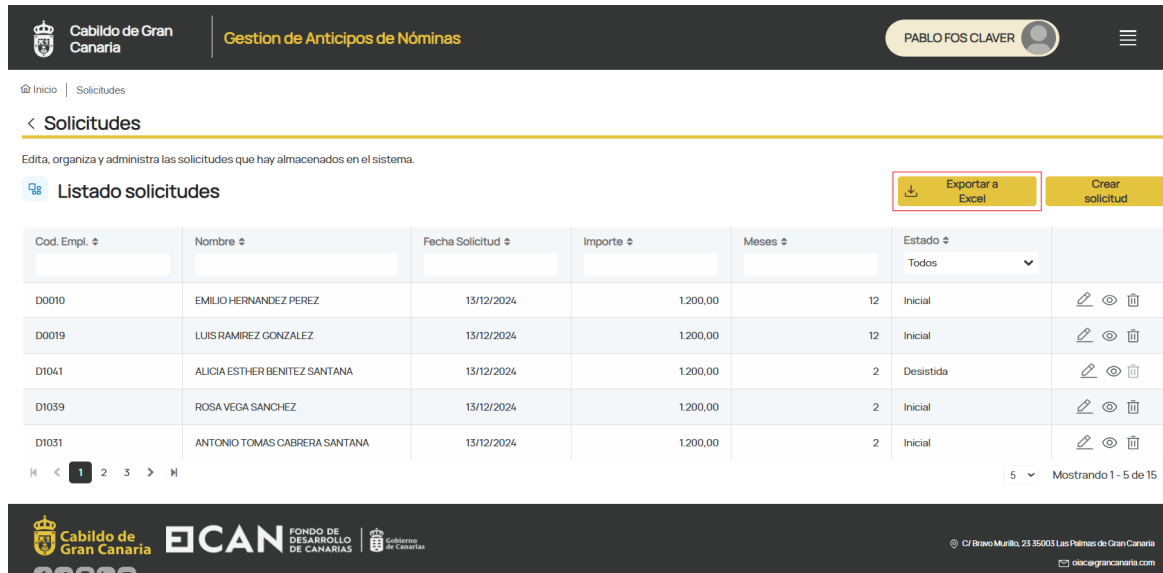


Imagen 61. Botón generar excel

5.6.2 Subida a preproducción y feedback

Después de todo el proceso de desarrollo de la aplicación hice el despliegue en el servidor de preproducción, para esto utilicé WinSCP¹⁹. Generé el archivo .war con la aplicación Maven y lo subí al servidor que nos proporcionó el CGC.

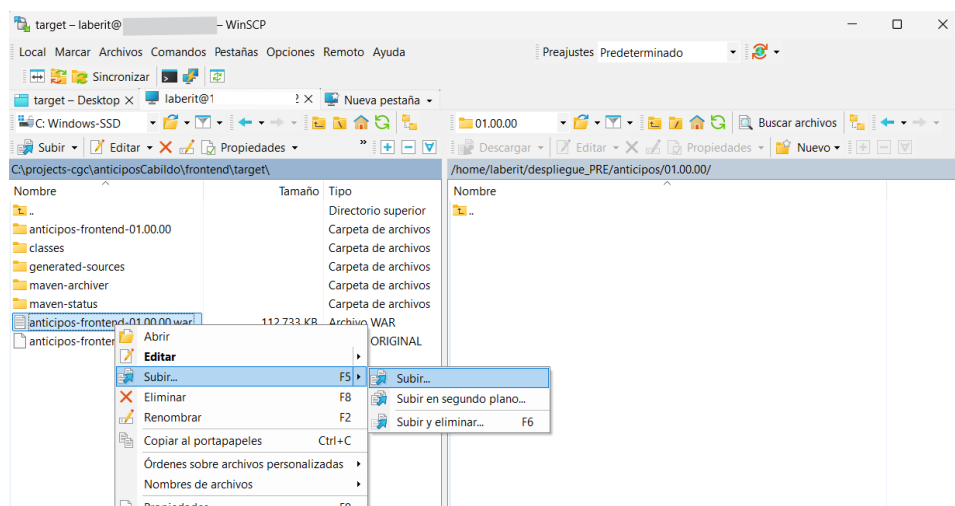


Imagen 62. WinSCP

¹⁹ <https://winscp.net/>

Finalmente, después de unos días los responsables de cabildo lo revisaron y nos solicitaron unas cuantas mejoras en la aplicación.

- Un filtro extra para el listado de anticipos y de solicitudes
- Modificar el alta de concesiones para que fuese posible conceder varias solicitudes en un mismo proceso de concesión
- Crear un mantenimiento de usuarios para evitar tener que gestionarlos únicamente desde la base de datos

5.6.3 Ajustes finales

Después de recibir todas las mejoras me las organicé de manera que las más sencillas las haría las primeras y de esta forma comencé a implementar los filtros extra.

Gestion de Anticipos de Nóminas

PABLO FOS CLAVER

Inicio | Anticipos

Anticipos

Edita, organiza y administra los anticipos que hay almacenados en el sistema.

Listado anticipos [Exportar a Excel](#)

Régimen jurídico Fecha desde Fecha hasta Situación [Buscar](#)

Empleado	Importe Total	Importe Restante	Mensualidad	Meses	Activo	Congelado	
ALICIA ESTHER BENITEZ SANTANA	1.200,00	1.000,00	600	2	Si	No	Editar Ver Eliminar
GUACIMARA ALVAREZ SOCORRO	1.800,00	1.800,00	150	12	Si	No	Editar Ver Eliminar
ROBERTO LUIS YANEZ YANEZ	150,00	150,00	38	4	Si	No	Editar Ver Eliminar
YOLANDA AGUIAR VERA	2.200,00	2.200,00	183	12	Si	No	Editar Ver Eliminar

Mostrando 1 - 4 de 4

Cabildo de Gran Canaria **ECAN** FONDO DE DESARROLLO DE CANARIAS

C/ Bravo Murillo, 25 35003 Las Palmas de Gran Canaria
daccagrancanaria.com

Imagen 63. Filtro extra

A continuación, modifiqué el alta de concesiones para ser capaces de seleccionar varias solicitudes en el mismo proceso de alta.

Añadir Solicitudes *

Núm. Operación: 12222 Ref. Expediente: 122

Observaciones: Prueba test Añadir

	Empleado	Rég. Jurídico	Fecha Solicitud	Importe	Meses
☒	LUIS RAMIREZ GONZALEZ	Laboral	13/12/2024	1.200,00	12
☐	ROSA VEGA SANCHEZ	Laboral	13/12/2024	1.200,00	2
☐	ANTONIO TOMAS CABRERA SANTANA	Laboral	13/12/2024	1.200,00	2
☐	ANA ACOSTA JORGE	Funcionario	04/03/2025	110,00	15

Mostrando 1 - 5 de 9

Solicitudes añadidas Eliminar

	Empleado	Fecha Solicitud	Importe	Meses
☒	EMILIO HERNANDEZ PEREZ	13/12/2024	1.200,00	12

Mostrando 1 - 1 de 1

Imagen 64. Alta de concesiones modificado

Por último, creé en el apartado del menú de mantenimientos el de usuarios con las operaciones CRUD correspondientes y desplegué la aplicación en el servidor de preproducción.

Mantenimiento de Usuarios

Edita, organiza y administra los Usuarios que hay almacenados en el sistema.

Crear Usuario

Usuario/a	Rol asignado	Fecha de baja
MBEJAR_ATS	Administrador	
DBELMONTE_ATS	Administrador	
ANREYESF	Administrador	
T_TEST	Nóminas	07/08/2025
FRODRIGUEZM	Administrador	

Mostrando 1 - 5 de 11

Imagen 65. Vista del listado de usuarios

6 Pruebas

6.1 Pruebas unitarias

En primer lugar, se realizaron pruebas unitarias con JUnit²⁰. Estas pruebas estaban centradas en comprobar que ciertas funciones del backend devolvieran los resultados esperados. Un ejemplo claro es el cálculo de los importes y plazos de devolución según el grupo del empleado. Gracias a estas pruebas se podía validar de forma rápida si la lógica de negocio funcionaba correctamente sin necesidad de levantar toda la aplicación.

6.2 Pruebas de integración

Después se llevaron a cabo pruebas de integración, que en este caso sirvieron para asegurarnos de que la comunicación entre la aplicación y la base de datos funcionaba bien.

Aquí se probaban cosas como: si los anticipos creados en la interfaz se guardaban correctamente en la base de datos, o si al consultar un empleado concreto se recuperaba la información adecuada.

Estas pruebas ayudaron a detectar pequeños problemas de mapeo entre las entidades JPA (Spring JPA, 2025) y las tablas.

6.3 Pruebas de validación

Por último, se pensaron unas pruebas de validación más orientadas al uso real de la aplicación. Aquí lo importante no era solo que el sistema funcionara, sino que funcionara como lo esperaba el usuario final.

Por ejemplo, se revisó que los mensajes de error aparecieran cuando un empleado intentaba pedir un anticipo fuera de los límites permitidos, o que el flujo de alta, edición y eliminación de solicitudes fuese claro e intuitivo.

Este tipo de pruebas resulta muy útil especialmente porque permiten detectar mejoras en la experiencia de usuario antes de que la aplicación pase a producción.

6.4 Pruebas automáticas

Para crear pruebas automáticas en Läberit utilizamos Playwright, es un framework desarrollado por Microsoft para automatizar pruebas web. El funcionamiento es muy parecido a un Scrapper pero en lugar de extraer datos de páginas web, se siguen unos pasos previamente programados y se comprueba si la respuesta obtenida es la deseada.

En este caso se han probado flujos enteros como el alta de una solicitud o la edición de una partida presupuestaria.

²⁰ <https://junit.org/>

6.5 Entorno de preproducción

Otro aspecto importante es, en sí, el uso del entorno de preproducción. Antes de llevar la aplicación a producción, se despliega primero en este servidor intermedio. De esta forma, el Product Owner y el cliente pueden probar las nuevas funcionalidades y asegurarse de que todo funciona como se espera. A partir de las pruebas y del feedback recibido, se crean tickets en JIRA para corregir fallos o añadir mejoras, lo que permite mantener un ciclo de validación y mejora continua.

7 Conclusión y trabajos futuros

Para mí este TFG no sólo representa lo que he aprendido con la aplicación de Gestión de Anticipos de Nómina, sino todo lo que he aprendido en Läberit, el proceso de una aplicación desde que se piensa hasta que está en producción, la forma de estructurar las peticiones de un cliente para darle forma y obtener una aplicación final, lo importante que es el trabajo en equipo y sobre todo lo beneficioso que es estar rodeado de un buen equipo en cualquier proyecto.

Cuando me planteé hacer el TFG sobre esta aplicación pensé que sería sencillo porque me pareció simple la idea pero después de haber recogido y redactado toda la información he aprendido que si en el ámbito del desarrollo del software se quiere crear algo de calidad se debe de dedicar tiempo, cada pequeño detalle lleva consigo una idea principal, un boceto, un análisis, una implementación y unas pruebas finales.

El futuro de la aplicación GDAN actualmente será pasar pruebas, tanto unitarias, funcionales, manuales en el servidor de preproducción... Después de esto, se subirá a producción para el uso en el cabildo y en ese momento se quedará en un stand by a la espera de nuevas implementaciones o arreglos de posibles bugs.

8 Bibliografía

IntelliJ IDEA. (2023). *IntelliJ IDEA Ultimate Documentation*. JetBrains. Recuperado de <https://www.jetbrains.com/es-es/idea/>

Oracle Corporation. (2019). *Oracle Database 19c Documentation*. Recuperado de <https://docs.oracle.com/en/database/>

Oracle Forms. (2025). *Oracle Forms Documentation*. Recuperado de https://docs.oracle.com/cd/E15523_01/index.htm

Red Hat, Inc. (2021). *WildFly 25 Administration Guide*. Recuperado de <https://docs.wildfly.org/25/>

Spring Data Team. (2022). *Spring Data JPA – Reference Documentation*. Recuperado de <https://spring.io/projects/spring-data-jpa>

Primefaces 8.0 (2025). *Primefaces – Reference Documentation*. Recuperado de https://primefaces.github.io/primefaces/8_0/#/

Maven 3.9.3 (2025). *Apache Maven – Reference Documentation*. Recuperado de <https://maven.apache.org/guides/index.html>

Spring Data JPA 3.9.3 (2025). *Spring JPA – Reference Documentation*. Recuperado de <https://docs.spring.io/spring-data/jpa/reference/jpa.html>

Atlassian. (2023). *JIRA Software Documentation*. Atlassian. Recuperado de <https://confluence.atlassian.com/alldoc/atlassian-documentation-32243719.html>

Figma Inc. (2023). *Figma Design Documentation*. Recuperado de <https://help.figma.com/hc/en-us>

Bootstrap. (2023). *Bootstrap Documentation*. Recuperado de <https://getbootstrap.com/docs/4.1/getting-started/introduction/>

Nielsen, J. (2012). *Usability Engineering*. Morgan Kaufmann.

Documentos internos del proyecto

Cabildo de Gran Canaria (2025). Antiguo_análisis_GDAN. Documento interno.

Cabildo de Gran Canaria (2025). CGC_GDAN-3_TOMREQ_v2: Toma de requisitos – Gestión de Anticipos de Nómina. Documento interno.

Cabildo de Gran Canaria (2025). CGC_GDAN-5_AF-Anticipos_v3: Análisis funcional – Gestión de Anticipos de Nómina. Documento interno.

9 Apéndices

9.1 Apéndice I. Objetivos de Desarrollo Sostenible

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No procede
ODS 1. El fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.				X
ODS 4. Educación de calidad.				X
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.	X			
ODS 9. Industria, innovación e infraestructuras.	X			
ODS 10. Reducción de las desigualdades.				X
ODS 11. Ciudades y comunidades sostenibles.				X

ODS 12. Producción y consumo responsables.				X
ODS 13. Acción por el clima.				X
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.		X		
ODS 17. Alianzas para lograr objetivos.	X			

Tabla 3. Relación del proyecto con las 17 ODS actuales

9.1.1 ODS 8. Trabajo decente y crecimiento económico

Con la nueva aplicación de GDAN, los trabajadores pueden gestionar los anticipos de nómina de una forma más sencilla y rápida. Esto mejora su día a día, ya que tienen más facilidades para gestionar este derecho, lo que a la larga también contribuye a un entorno laboral más justo y con mayor seguridad económica.

9.1.2 ODS 9. Industria, innovación e infraestructuras

El proyecto consiste en si en modernizar una aplicación antigua que se había quedado obsoleta. Al pasar a tecnologías más actuales como Spring Boot o JPA, no solo se gana en eficiencia y rendimiento, sino que también se asegura que la infraestructura tecnológica esté preparada para el futuro y sea más fácil de mantener y ampliar.

9.1.3 ODS 16. Paz, justicia e instituciones sólidas

Al mejorar el sistema, se consigue que el proceso de pedir y gestionar anticipos sea más transparente y ordenado. Esto genera más confianza en la institución, porque los trámites son claros, fiables y accesibles para todos los empleados.

9.1.4 ODS 17. Alianzas para lograr objetivos

Este proyecto no se hace solo, sino que implica la colaboración entre varios perfiles del equipo (programador, analista, jefe de proyecto...) y la coordinación constante con el Cabildo de Gran Canaria. Gracias a ese trabajo conjunto se pueden cumplir los objetivos y ofrecer una herramienta que realmente responda a lo que se necesita.

10 Glosario de términos y acrónimos

UX

Experiencia de Usuario.

UI

Interfaz de Usuario.

CGC

Cabildo de Gran Canaria.

Oracle

Empresa multinacional conocida sobre todo por su software de gestión de bases de datos.

Internet Explorer

Navegador web diseñado por Microsoft que está obsoleto actualmente.

SQL

Lenguaje de programación creado para gestionar bases de datos.

TIC

Tecnologías de la Información y la Comunicación.

Software

Sistema formal de un sistema informático que hace posible la realización de tareas específicas.

Läberit

Empresa dedicada al desarrollo de software y soluciones tecnológicas.

GDAN

Gestión De Anticipos de Nómina.

Metodología ágil

Enfoque para la gestión de proyectos que se centra en la flexibilidad, la colaboración y la entrega continua del producto.

Product Owner

Persona responsable de maximizar el valor de un producto desarrollado por un equipo.

Backend

Parte del sistema no visible que se encarga de la lógica de una aplicación.

Frontend

Parte del sistema visible que se encarga de la interfaz de una aplicación.

Base de datos

Almacén software donde se guardan los datos en un proyecto informático.

Servicio web

Tecnología utilizada para intercambiar datos entre distintas aplicaciones o sistemas a través de una red.

Servidor de preproducción

Servidor donde se almacenan las aplicaciones del CGC con el objetivo de poder revisar la aplicación sin necesidad de tenerla desplegada en local.

CRUD

Son las cuatro operaciones básicas a realizar sobre los datos (Create, Read, Update, Delete).

MVC

Modelo, Vista y Controlador.

ILT

Incapacidad laboral temporal.

Excel

Es un programa de hojas de cálculo desarrollado por Microsoft.

Modelo lógico

Es una representación gráfica de las clases y relaciones entre ellas que tiene una aplicación.

JPA

Java Persistence API.

CSV

Comma Separated Values.

CSS

Cascading Style Sheets (Hojas de estilo en cascada).

Input

Campo donde se introduce información en el contexto del software.

Default

Estado estándar.

Hover

Estado que se produce cuando se pasa el ratón sobre un objeto.

Active

Estado que se produce cuando se hace click con el ratón sobre un objeto.

API

Application Programming Interface.

JSF

JavaServer Faces.

PRE

Preproducción.

HTML

Lenguaje de etiquetas de hipertexto (HyperText Markup Language).

Playwright

Framework desarrollado por Microsoft para crear pruebas automáticas en un entorno de aplicaciones web.

Scraper

Programa o script que automatiza la extracción de datos de sitios web.

JUnit

Framework de pruebas de código abierto para Java, utilizado para escribir y ejecutar pruebas unitarias.

SoapUI

Herramienta de código abierto con interfaz intuitiva utilizada para probar aplicaciones con arquitecturas orientadas a servicios SOAP y REST.

JavaMelody

Herramienta de código abierto diseñada para monitorizar el rendimiento de aplicaciones Java.

Breadcrumb

Elemento de navegación que muestra la ruta jerárquica dentro de un sitio o aplicación web.

DTO

Data Transfer Object.

Servicio

Clase que comunica las clases controladoras más cercanas al frontend con las clases de repositorios más cercanas al backend.

Repositorio

Clase que comunica las clases de servicio más cercanas al frontend con la base de datos.

CAS

Central Authentication Service.

LazyDataModel

Lista de objetos que carga datos de forma perezosa o bajo demanda.

WinSCP

Aplicación que se utiliza para transferir archivos de forma segura entre tu ordenador local y un servidor remoto.

Archivo .war

Archivo comprimido que se utiliza para realizar el despliegue de una aplicación en un servidor.

UML

Estándar ampliamente extendido en ingeniería del software que facilita la transición entre el análisis de requisitos y el diseño de la aplicación