



UNIVERSIDAD TECNOLÓGICA METROPOLITANA  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE INFORMÁTICA  
ESCUELA DE INFORMÁTICA  
CARRERA INGENIERÍA EN INFORMÁTICA

## **REINGENIERÍA DE UN SISTEMA DE GESTIÓN DE UNIFORMES CORPORATIVOS**

**TRABAJO DE TITULACIÓN PARA OPTAR AL TÍTULO DE  
INGENIERO EN INFORMÁTICA**

### **AUTORES:**

ARANEDA MELLA, GERARDO GABRIEL  
GÓMEZ RODRÍGUEZ, ANDRÉS PATRICIO  
JIMÉNEZ ROMERO, NICOLÁS ISAÍAS

### **PROFESOR GUÍA:**

RUIZ ROCHA, YELKA MACARENA

SANTIAGO - CHILE

2025



## AUTORIZACIÓN PARA LA REPRODUCCIÓN DEL TRABAJO DE TITULACIÓN

### 1. Identificación del trabajo de titulación

Nombre del(os) alumno(s):

Gerardo Gabriel Araneda Mella.

Andrés Patricio Gómez Rodríguez.

Nicolás Isaías Jiménez Romero.

	Araneda	Gómez	Jiménez
Rut:	21061302-1	20749196-9	
Dirección:	Santiago, Lo Prado	Santiago, La Florida	
E-mail:	garaneda@utem.cl	agomezr@utem.cl	njimenezr@utem.cl
Teléfono:	+56945608143	+56979828311	

Título de la tesis:

Reingeniería de un Sistema de Gestión de Uniformes Corporativos

Escuela: Escuela de Informática

Carrera o programa: Carrera Ingeniería en Informática

Título al que opta: Título de Ingeniero en Informática

### 2. Autorización de Reproducción

Se autoriza la reproducción total o parcial de este trabajo de titulación, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica que acredita al trabajo y a su autor.

En consideración a lo anterior, se autoriza su reproducción de forma (marque con una X):

X	Inmediata
	A partir de la siguiente fecha: _____ (mes/año)

Fecha: \_\_\_\_\_

Firma: \_\_\_\_\_

Esta autorización se otorga en el marco de la ley N° 17.336 sobre Propiedad Intelectual, con carácter gratuito y no exclusivo para la Institución.



## NOTA OBTENIDA:

---

**Firma y timbre autoridad  
responsable**



## AGRADECIMIENTOS

Deseamos expresar nuestro agradecimiento a la institución académica por la formación entregada a lo largo del proceso formativo, la cual permitió adquirir los conocimientos y competencias necesarios para el desarrollo del presente proyecto.

Asimismo, agradecemos a la profesora guía Yelka Ruiz Rocha por su constante orientación, disposición y acompañamiento durante el desarrollo de este trabajo, así como a los docentes de la carrera que, mediante sus enseñanzas y observaciones, contribuyeron significativamente al fortalecimiento de la propuesta presentada.

Adicionalmente, expresamos nuestro agradecimiento a Julio Gómez, el propietario del sistema de tallajes (SISTAL) por su disposición y apertura durante el desarrollo del proyecto, así como por el tiempo dedicado a compartir su experiencia y conocimientos respecto a las reglas de negocio y el funcionamiento del sistema. Su colaboración fue fundamental para comprender el dominio del problema y asegurar que la solución propuesta se alineara adecuadamente con las necesidades reales del contexto operativo.

Finalmente, queremos agradecer a nuestras familias y personas cercanas por el apoyo brindado durante el proceso académico, cuyo respaldo consideramos fundamental para la elaboración de este proyecto.

## ÍNDICE GENERAL

<b>RESUMEN</b>	<b>XI</b>
<b>INTRODUCCIÓN</b>	<b>1</b>
<b>CAPÍTULO 1: DESCRIPCIÓN GENERAL DEL PROYECTO</b>	<b>4</b>
1.1 Descripción del Proyecto . . . . .	4
1.2 Objetivos . . . . .	6
1.2.1 Objetivo General . . . . .	6
1.2.2 Objetivos Específicos . . . . .	6
1.3 Alcances y Limitaciones . . . . .	7
1.4 Recursos . . . . .	8
1.4.1 Recursos Humanos . . . . .	8
1.4.2 Recursos de Hardware . . . . .	8
1.4.3 Recursos de Software . . . . .	8
<b>CAPÍTULO 2: DESCRIPCIÓN DE LA EMPRESA Y EL SISTEMA</b>	<b>10</b>
2.1 Descripción de la Empresa . . . . .	10
2.1.1 Historia . . . . .	10
2.1.2 Misión . . . . .	11
2.1.3 Visión . . . . .	12
2.1.4 Modelo de Negocios . . . . .	12
2.1.5 Análisis FODA . . . . .	15
2.2 Descripción del Sistema (SISTAL) . . . . .	17
2.2.1 Actores Principales y sus Módulos . . . . .	17
2.2.2 Procesos que Aborda SISTAL . . . . .	23
<b>CAPÍTULO 3: ANÁLISIS TÉCNICO DEL SISTEMA ACTUAL</b>	<b>25</b>
3.1 Infraestructura . . . . .	25
3.1.1 Tecnologías y Herramientas . . . . .	25
3.1.2 Arquitectura . . . . .	26
3.1.3 Proceso de Despliegue y Actualización . . . . .	28
3.1.4 Principales Problemas de esta Infraestructura . . . . .	29
3.2 Diagramas de Casos de Uso . . . . .	32
3.2.1 Módulo de Funcionario . . . . .	32
3.2.2 Módulo de Proveedor . . . . .	35
3.2.3 Módulo de Administrador . . . . .	38
3.3 Diagrama de Bases de Datos . . . . .	41
3.4 Diagramas de Flujo . . . . .	45



3.4.1	Proceso de Registro de Tallas . . . . .	45
3.4.2	Proceso de Envío a Confección . . . . .	47
3.4.3	Proceso de Postventa: Arreglo . . . . .	49
<b>CAPÍTULO 4: MARCO TEÓRICO</b>		<b>51</b>
4.1	Sistemas de información y su rol en las organizaciones . . . . .	51
4.2	Ingeniería de Software . . . . .	53
4.2.1	Metodologías de Desarrollo . . . . .	54
4.2.2	DevOps . . . . .	56
4.3	Reingeniería de Software . . . . .	56
4.4	Arquitectura de Software . . . . .	58
4.4.1	Arquitectura de Microservicios . . . . .	59
4.5	Contenedores y Orquestación . . . . .	60
4.5.1	Contenedores . . . . .	60
4.5.2	Kubernetes . . . . .	62
<b>CAPÍTULO 5: LEVANTAMIENTO DE REQUISITOS MEDIANTE HISTORIAS DE USUARIO</b>		<b>63</b>
5.1	Requerimientos . . . . .	63
5.1.1	Requerimientos funcionales . . . . .	64
5.1.2	Requerimientos no funcionales . . . . .	67
5.2	Historias de usuario . . . . .	68
<b>CAPÍTULO 6: DISEÑO Y ARQUITECTURA ACTUALIZADOS</b>		<b>72</b>
6.1	Diseño de Arquitectura . . . . .	72
6.1.1	Diagrama de Contexto . . . . .	73
6.1.2	Diagrama de Contenedores . . . . .	75
6.1.3	Diagrama de Arquitectura . . . . .	77
6.1.4	Diagrama de Infraestructura . . . . .	80
6.1.5	Diagrama de Despliegue . . . . .	82
6.2	Diseño Lógico . . . . .	84
6.2.1	Diagramas de Flujo . . . . .	84
6.3	Diseño Físico . . . . .	88
6.3.1	Modelo Entidad-Relación de Base de Datos . . . . .	88
6.4	Casos de Uso . . . . .	90
<b>CAPÍTULO 7: DESARROLLO DEL PROYECTO</b>		<b>100</b>
7.1	Stack tecnológico utilizado . . . . .	100
7.1.1	Tecnologías de frontend . . . . .	100
7.1.2	Tecnologías de backend . . . . .	102
7.1.3	Persistencia de datos . . . . .	103
7.1.4	Infraestructura y DevOps . . . . .	104



7.2	Gestión del proyecto y trabajo colaborativo . . . . .	106
7.2.1	Metodología de Trabajo . . . . .	106
7.2.2	Flujo de Trabajo con Git . . . . .	107
7.2.3	Organización en GitHub . . . . .	107
7.3	Diseño de Interfaz de Usuario . . . . .	108
<b>CONCLUSIONES</b>		<b>118</b>
<b>BIBLIOGRAFÍA</b>		<b>120</b>
<b>GLOSARIO</b>		<b>121</b>
	Glosario del Negocio . . . . .	121
	Glosario Técnico . . . . .	123
	Siglas . . . . .	124



## ÍNDICE DE FIGURAS

1.1	Interacción de los actores con los módulos del sistema, encerrado en un cuadro verde se encuentra el módulo a implementar. . . . .	5
2.1	Lienzo Canva del modelo de negocios actual de la empresa. . . . .	14
2.2	Diagrama explicativo de actores del sistema . . . . .	18
2.3	Página de inicio de sesión. Presenta los campos de Usuario, Contraseña y Recuperar contraseña con un <i>email</i> registrado. . . . .	20
2.4	Interfaz del módulo de funcionario. Muestra datos del funcionario, su sucursal de la empresa y sus tallas que puede editar. . . . .	21
2.5	Interfaz del módulo de proveedor. Muestra los funcionarios enviados por el administrador de la empresa solicitante que les corresponde uniforme. . . . .	22
2.6	Interfaz del módulo de administrador. El menú que se muestra permite al administrador aprobar o rechazar los funcionarios nuevos que se registraron en el sistema. . . . .	22
2.7	Flujo simplificado del proceso de dotación de uniformes apoyado por SISTAL . . . . .	23
2.8	Flujo simplificado de solicitud de arreglo . . . . .	24
3.1	Arquitectura actual de SISTAL . . . . .	27
3.2	Diagrama de despliegue del sistema actual . . . . .	28
3.3	Ejemplo de código PHP con código HTML incrustado más su salida generada. . . . .	29
3.4	Tablas de la Base de Datos MySQL . . . . .	30
3.5	Diagrama de casos de uso - Funcionario (Parte 1) . . . . .	33
3.6	Diagrama de casos de uso - Funcionario (Parte 2) . . . . .	34
3.7	Diagrama de casos de uso - Proveedor (Parte 1) . . . . .	36
3.8	Diagrama de casos de uso - Proveedor (Parte 2) . . . . .	37
3.9	Diagrama de casos de uso - Administrador (Parte 1) . . . . .	39
3.10	Diagrama de casos de uso - Administrador (Parte 2) . . . . .	40
3.11	Diagrama físico de bases de datos del sistema actual (Parte 1) . . . . .	43
3.12	Diagrama físico de bases de datos del sistema actual (Parte 2) . . . . .	44
3.13	Diagrama de flujo del sistema actual sobre el proceso de registro de tallas . . . . .	46
3.14	Diagrama de flujo del sistema actual sobre el proceso de envío a confección . . . . .	48
3.15	Diagrama de flujo del sistema actual sobre el proceso de postventa . . . . .	50
6.1	Diagrama C1 de contexto . . . . .	74



6.2	Diagrama C2 de Contenedores . . . . .	76
6.3	Diagrama de arquitectura . . . . .	77
6.4	Diagrama de infraestructura en Google Cloud Platform (GCP) . . . . .	81
6.5	Diagrama de despliegue con CI/CD . . . . .	83
6.6	Diagrama de flujo Login . . . . .	85
6.7	Diagrama de flujo Solicitud de uniforme . . . . .	86
6.8	Diagrama de flujo Postventa . . . . .	87
6.9	Diagrama entidad relación propuesto de base de datos, modelo lógico	89
6.10	Diagrama de casos de uso del actor Funcionario (Parte 1) . . . . .	91
6.11	Diagrama de casos de uso del actor Funcionario (Parte 2) . . . . .	92
6.12	Diagrama de casos de uso del actor Proveedor (Parte 1) . . . . .	94
6.13	Diagrama de casos de uso del actor Proveedor (Parte 2) . . . . .	95
6.14	Diagrama de casos de uso del actor Administrador (Parte 1) . . . . .	97
6.15	Diagrama de casos de uso del actor Administrador (Parte 2) . . . . .	98
7.1	GUI de inicio de sesión . . . . .	109
7.2	GUI de registro de usuario . . . . .	110
7.3	GUI Panel principal de Funcionario . . . . .	111
7.4	GUI Botón de notificaciones . . . . .	112
7.5	GUI solicitudes realizadas por el funcionario . . . . .	113
7.6	GUI de solicitud de cambio de sucursal . . . . .	114
7.7	GUI de solicitud de cambio de talla . . . . .	115
7.8	GUI de seguimiento de despachos . . . . .	116
7.9	GUI de información de cuenta funcionario . . . . .	117



## ÍNDICE DE TABLAS

1.1 Recursos Humanos del Proyecto . . . . .	8
1.2 Recursos de Software del Proyecto . . . . .	9
2.1 Análisis FODA con principales atributos identificados de la empresa	15
2.2 Matriz FODA con análisis estratégicos . . . . .	16
6.1 Descripción microservicios de la Figura 6.3 . . . . .	78
6.2 Resumen de casos de uso del actor Funcionario . . . . .	93
6.3 Resumen de casos de uso del actor Proveedor . . . . .	96
6.4 Resumen de casos de uso del actor Administrador . . . . .	99



## RESUMEN

El presente proyecto aborda la reingeniería del “Sistema de Tallajes” (SISTAL), un sistema web de gestión de uniformes corporativos desarrollado por la empresa GyV Inversiones para apoyar la administración de la dotación de vestimenta del personal en diferentes organizaciones. El sistema opera actualmente bajo una arquitectura monolítica basada en tecnologías obsoletas y sin prácticas modernas de desarrollo y despliegue, lo que limita su escalabilidad, mantenibilidad y capacidad de evolución.

El objetivo del proyecto es rediseñar la arquitectura de SISTAL para soportar de forma segura, escalable y mantenible el ciclo de gestión de uniformes, preservando los flujos de negocio existentes. Para ello, se adopta un enfoque basado en microservicios, prácticas DevOps y tecnologías en la nube, aplicando una metodología de reingeniería de software que incluye las fases de ingeniería inversa, rediseño arquitectónico e ingeniería directa.

Como resultado, se definió una arquitectura modernizada y se desarrolló un prototipo funcional del módulo de funcionario, el cual permitió validar la propuesta mediante pruebas funcionales y de usabilidad. La solución propuesta demuestra mejoras relevantes en modularidad, escalabilidad y mantenibilidad, confirmando la viabilidad técnica y el valor estratégico de la modernización planteada.

## PALABRAS CLAVE

Ingeniería de software, Reingeniería de software, Modernización tecnológica, Arquitectura de software, Microservicios, Computación en la nube, DevOps, Sistema de tallajes, Industria textil, Contenedores, Kubernetes.



## ABSTRACT

This project addresses the reengineering of “Sistema de Tallajes” (SISTAL), a web-based corporate uniform management system developed by GyV Inversiones to support organizations in administering employee uniform allocation. The system currently operates under a monolithic architecture based on obsolete technologies and lacks modern development and deployment practices, which limits its scalability, maintainability, and capacity for evolution.

The objective of the project is to redesign SISTAL’s architecture to securely, scalably, and maintainably support the uniform management lifecycle while preserving existing business workflows. To this end, an approach based on microservices, DevOps practices, and cloud technologies is adopted, applying a software reengineering methodology that includes the phases of reverse engineering, architectural redesign, and forward engineering. As a result, a modernized architecture was defined and a functional prototype of the employee module was developed, allowing the proposal to be validated through functional and usability testing. The proposed solution demonstrates notable improvements in modularity, scalability, and maintainability, confirming both the technical feasibility and the strategic value of the proposed modernization.

## KEYWORDS

Software engineering, Software reengineering, Technological modernization, Software architecture, Microservices, Cloud computing, DevOps, Sizing system, Textile industry, Containers, Kubernetes.



## INTRODUCCIÓN

La reingeniería de software corresponde al proceso de analizar, reestructurar y modernizar un sistema existente con el propósito de mejorar su diseño, su rendimiento y su mantenibilidad, preservando las funcionalidades esenciales que sustentan el negocio. En este contexto, el presente trabajo aborda la modernización del “Sistema de Tallajes” (SISTAL), un sistema *web* desarrollado por la empresa GyV Inversiones en el año 2009, para gestionar la dotación de uniformes corporativos en organizaciones que requieren administrar de forma centralizada las tallas, solicitudes, confección, despacho y trazabilidad de las prendas entregadas a su personal.

Al hablar de diseño arquitectónico y tecnológico, se hace referencia a la planificación y estructuración de los componentes de *software* y las tecnologías que los soportan, estableciendo la forma en que estos se organizan, se comunican e interactúan entre sí para dar cumplimiento a los objetivos del sistema. Por su parte, una aplicación, plataforma o sistema *web* se define como una solución informática accesible a través de un navegador *web*, que permite a los usuarios interactuar con servicios y datos mediante internet o una red interna. En consecuencia, en este documento se hará referencia a SISTAL como un “Sistema *Web*”.

A lo largo de más de una década de operación, SISTAL ha desempeñado un rol fundamental como herramienta de apoyo para grandes empresas. Sin embargo, su arquitectura monolítica (un diseño en el que todos los componentes del sistema se encuentran estrechamente acoplados), basada en las tecnologías PHP 5.6 y MySQL, y alojada en un servicio de *hosting* compartido, presenta actualmente limitaciones significativas en términos de escalabilidad, mantenibilidad, seguridad y capacidad de evolución. A ello se suma la ausencia de estándares de



diseño, la escasa adopción de buenas prácticas de desarrollo y la inexistencia de procesos de despliegue controlados, factores que dificultan la incorporación de nuevas funcionalidades y aumentan los riesgos operativos del sistema.

Por lo tanto, el proyecto se enfoca en el rediseño arquitectónico y tecnológico de este sistema *web*, definiendo una base tecnológica sólida para futuras innovaciones y expansiones de la plataforma, soportando el ciclo completo de dotación de uniformes y preservando los flujos de negocio existentes. A modo de validación de esta propuesta, se realiza el desarrollo del módulo de funcionarios del sistema, componente clave del flujo de dotación, por ser el responsable de registrar tallas, gestionar solicitudes y habilitar los procesos subsiguientes de aprobación, confección y despacho.

El desarrollo de este proyecto surge de la motivación por profundizar en el campo de la arquitectura de software, junto con nuestra cercanía al dominio y una comprensión detallada de las reglas de negocio vinculadas a la gestión de uniformes corporativos. Esta perspectiva, construida a partir del contacto directo con quienes han participado en la operación y el diseño del sistema en el pasado, nos permite identificar con precisión los puntos críticos, priorizar mejoras con un enfoque práctico y proponer soluciones viables que respondan a las necesidades reales de la operación.

Además, nos motiva la oportunidad de generar un impacto significativo en un sector donde escasean soluciones integrales comparables. Modernizar y fortalecer SISTAL constituye un desafío formativo de gran valor, ya que nos enfrenta a problemáticas de alta complejidad y, al mismo tiempo, nos brinda la posibilidad de aplicar los conocimientos teóricos adquiridos, que hasta ahora no habíamos podido poner en práctica durante nuestra etapa formativa. Asimismo, el proyecto nos permite aportar resultados concretos a un sistema de uso empresarial, mejo-



rando su trazabilidad, eficiencia operativa y capacidad de evolución tecnológica. En síntesis, elegimos este proyecto porque integra el conocimiento del negocio con una oportunidad real de aprendizaje y de impacto positivo en la organización.

La metodología aplicada en este proyecto se basa en un enfoque inspirado de la reingeniería de software, el cual comprende tres fases principales: Ingeniería Inversa (*Reverse Engineering*), Alteración (*Alteration*) e Ingeniería Directa (*Forward Engineering*). En la primera fase se analizó el sistema original para recuperar conocimiento y redocumentar su funcionamiento, arquitectura, procesos y dependencias internas. Posteriormente, en la etapa de alteración, se llevó a cabo la reorganización y modernización conceptual del sistema mediante actividades de reestructuración y rediseño arquitectónico basadas en principios de microservicios, computación en la nube y prácticas DevOps. Finalmente, la fase de ingeniería directa correspondió al desarrollo del Módulo del Funcionario dentro de la arquitectura propuesta, utilizando el marco de trabajo Scrum para gestionar el proceso de implementación mediante iteraciones cortas y validaciones continuas.

Los primeros capítulos del informe presentan el contexto general del proyecto, incluyendo la introducción y descripción del mismo, la historia de la empresa, así como la descripción y el análisis del sistema. En esta última sección se revisan aspectos como su arquitectura, la estructura de la base de datos, los flujos de información, entre otros componentes relevantes. Todo ello permite establecer los antecedentes y comprender la problemática que motiva la necesidad de una reingeniería. Posteriormente, se desarrolla el marco teórico que sustenta el proyecto, incorporando conceptos esenciales y otros enfoques relevantes que contribuyen a la propuesta técnica. Con esta base, se pasa al desarrollo del proyecto, donde se especifican los requerimientos, el diseño propuesto para el nuevo sistema, las herramientas y metodologías seleccionadas, finalizando con la documentación final, la implementación y conclusiones.



## CAPÍTULO 1

---

### DESCRIPCIÓN GENERAL DEL PROYECTO

Este capítulo presenta la descripción del proyecto y la identificación del módulo que será desarrollado como prototipo final, junto con sus objetivos, alcances y limitaciones y los recursos utilizados .

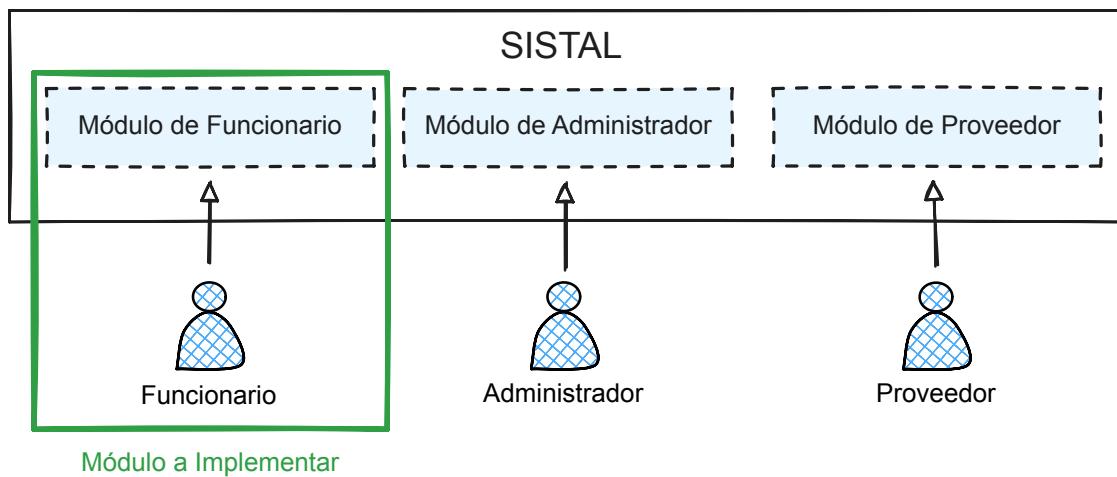
#### 1.1 DESCRIPCIÓN DEL PROYECTO

En línea con lo expuesto en la introducción, el proyecto se centra en la reingeniería de “Sistema de Tallajes” (SISTAL), comenzando con un diagnóstico exhaustivo del sistema monolítico actual, orientado a identificar sus principales limitaciones técnicas y estructurales. A partir de este análisis, se plantea el diseño de una arquitectura orientada a la nube basada en **microservicios**, enfoque que, a diferencia del modelo monolítico, divide el sistema en múltiples componentes independientes que se comunican entre sí, cada uno con una funcionalidad específica.

Este enfoque se complementa con el uso de **contenedores**, entendidos como unidades que encapsulan una aplicación, garantizando entornos de ejecución consistentes y reutilizables, y con la **orquestación** de estos a través de Kubernetes, tecnología que administra automáticamente cada contenedor, creando y destruyéndolos basado en la carga del sistema. Finalmente, se incorpora la adopción de prácticas **DevOps** incluyendo integración y despliegue continuo (CI/CD), prácticas orientadas tanto a la automatización de los procesos de despliegue y actualización del sistema como a la optimización del proceso de desarrollo, me-

diente la estandarización de flujos de trabajo, la ejecución automática de pruebas para la detección temprana de errores.

La implementación se realiza de forma incremental, comenzando exclusivamente por el Módulo del Funcionario como alcance para este proyecto. SISTAL cuenta con tres módulos principales asociados a los actores clave del sistema (Administrador, Funcionario y Proveedor) tal como se muestra en la [Figura 1.1](#).



**Figura 1.1:** Interacción de los actores con los módulos del sistema, encerrado en un cuadro verde se encuentra el módulo a implementar.

Fuente: *Diseño Propio*

La descripción detallada de estos actores y de sus respectivos módulos se presenta en la [Subsección 2.2.1](#) del [Capítulo 2](#).

Se seleccionó el módulo de Funcionario como punto de partida porque constituye la base operativa del sistema: es el primero en interactuar con el flujo de dotación, registra la información crítica de tallas y habilita los procesos posteriores de aprobación, confección y despacho.

Se construirán los componentes/microservicios destinados a permitir a un funcionario registrar sus tallas y gestionar sus solicitudes. Al cierre, se realizarán



pruebas funcionales y de usabilidad, validando criterios acordados y contrastando los resultados con la línea base del enfoque monolítico original.

## 1.2 OBJETIVOS

### 1.2.1 Objetivo General

Diseñar la nueva arquitectura del sistema de gestión de uniformes corporativos SISTAL, basada en microservicios y tecnologías en la nube, que permita mejorar la escalabilidad y mantenibilidad de la plataforma, y desarrollar el prototípo del módulo de funcionario.

### 1.2.2 Objetivos Específicos

- Analizar la arquitectura actual del sistema monolítico, identificando limitaciones y oportunidades de mejora en escalabilidad y mantenibilidad.
- Documentar los procesos de negocio y flujos de datos existentes con el propósito de modelarlos y replicarlos en la nueva arquitectura del sistema.
- Diseñar una nueva arquitectura basada en microservicios escalable en la nube.
- Proponer lineamientos de infraestructura, tecnologías y prácticas DevOps.
- Desarrollar el prototípo del módulo de funcionario del sistema, a modo de implementación inicial y validación de la propuesta.



## 1.3 ALCANCES Y LIMITACIONES

### Alcances

- Reestructurar el sistema monolítico de SISTAL hacia una arquitectura de microservicios.
- Diseñar los microservicios centrales (usuarios, pedidos, inventario y estados).
- Documentar la configuración de infraestructura en nube (Google Cloud Platform).
- Implementar integración y despliegue continuo (CI/CD) y buenas prácticas DevOps.
- Desarrollar e implementar el prototipo del módulo de funcionario del sistema.

### Limitaciones

- No se contempla el desarrollo de aplicaciones móviles nativas o complementarias.
- No se usarán datos reales.
- No habrá capacitación masiva a usuarios finales.
- No se implementarán integraciones con sistemas externos no contemplados en la versión original.
- Se desarrollará únicamente el módulo de funcionario.



## 1.4 RECURSOS

### 1.4.1 Recursos Humanos

El proyecto cuenta con el siguiente equipo de trabajo presentado en la [Tabla 1.1](#).

**Tabla 1.1:** Recursos Humanos del Proyecto

*Fuente: Elaboración Propia*

Rol	Responsabilidades principales	Nombre(s)
Product Owner	Definición de la visión del producto, priorización y gestión de requerimientos, y alineación con las necesidades del negocio.	Sr. Julio Gómez
Equipo de Desarrollo	Diseño, desarrollo, implementación y mantenimiento de la solución.	Gerardo Araneda Mella, Andrés Gómez Rodríguez, Nicolás Jiménez Romero.

### 1.4.2 Recursos de Hardware

- Computadora con capacidad para ejecutar entornos de desarrollo, contenedores y pruebas locales.
- Servidores en la nube proporcionados por el proveedor seleccionado.

### 1.4.3 Recursos de Software

Los recursos presentados en la [Tabla 1.2](#) representan las herramientas y tecnologías utilizadas para la realización del proyecto, una justificación más deta-



llada se detalla en capítulos posteriores.

**Tabla 1.2:** Recursos de Software del Proyecto

Fuente: *Elaboración Propia*

Componente	Herramienta / Tecnología Utilizada
Lenguaje Frontend	React
Lenguaje Backend	Go
Contenedорización y Orquestación	Docker y Kubernetes
Plataforma Cloud	GCP
Base de datos	PostgreSQL
Control de versiones	Git/GitHub
CI/CD	GitHub Actions



## CAPÍTULO 2

---

### DESCRIPCIÓN DE LA EMPRESA Y EL SISTEMA

Este capítulo presenta la descripción de la empresa responsable del desarrollo del sistema, revisando su funcionamiento y modelo de negocio, con el propósito de contextualizar el origen y los objetivos del sistema dentro del ámbito organizacional. Posteriormente, se describe de manera general el sistema actual, identificando sus principales actores, flujos de información y procesos operativos, lo que permite comprender el rol que cumple SISTAL en la gestión de uniformes corporativos y las dinámicas de interacción entre los distintos usuarios que lo utilizan.

#### 2.1 DESCRIPCIÓN DE LA EMPRESA

##### 2.1.1 Historia

En el año 2005, el ingeniero civil industrial Julio Gómez Vega, se integra a la empresa de la industria textil “La Scala”, dedicada a la producción y comercialización de prendas de vestir, asumiendo el rol de ingeniero de planificación de producción con el objetivo principal de desarrollar el departamento de venta y distribución de uniformes corporativos para sus diferentes clientes. Tras unos años, enfocado en automatizar y centralizar flujos de datos, dirige el desarrollo de una plataforma informática que permite obtener de forma rápida y eficientemente la información necesaria para producción y logística de cada prenda correspondiente a los uniformes comprometidos para la distribución. La herramienta que impul-



só permitía gestionar curvas de talla, coordinar la logística de entrega y recopilar retroalimentación en la postventa de los productos suministrados.

Tras el cierre de La Scala, debido a diferencias entre sus propietarios, Julio Gómez y Sebastián Gómez fundaron GyV Inversiones Ltda. en el año 2009, con el propósito de ofrecer soluciones tecnológicas a la industria textil. Aprovechando la experiencia adquirida, desarrollaron el Sistema de Tallajes (SISTAL), una plataforma que automatizó la toma de medidas, la generación de pedidos y la distribución de uniformes. Con el tiempo, SISTAL se consolidó como una herramienta clave en grandes empresas como BancoEstado y Banco de Chile.

Durante más de una década, SISTAL funcionó exitosamente como sistema de administración de uniformes. Sin embargo, los avances tecnológicos y las crecientes exigencias en materia de seguridad de la información hicieron necesaria una asesoría externa, con el fin de actualizar la plataforma e incorporar nuevos conceptos tecnológicos. A continuación, se presentan la Misión y Visión de la empresa.

## 2.1.2 Misión

Ser una empresa líder dentro del rubro textil, en lo que a sistemas de información se refiere, privilegiando la veracidad, rapidez y automatización de la información, con sistemas de bajo costo y fácil uso. Apostamos a la diferenciación en la comercialización mediante un servicio orientado al cliente final, con atención personalizada a cada usuario. Seremos considerados por nuestros clientes, más que un proveedor de servicios, un aliado estratégico que les ayudará a concentrarse específicamente en su negocio, dejando la administración de los unifor-



mes corporativos en nuestras manos.

### 2.1.3 Visión

Formar parte de las mejores empresas de gestión de uniformes corporativos a nivel nacional e internacional, con modelos de vanguardia, una confección que cumpla con todos los estándares de calidad del mercado, y un servicio integral que exceda las expectativas de nuestros clientes. Para lograrlo, nuestro foco está apuntando a dos principios básicos, en primer lugar, potenciar las fortalezas de nuestro recurso humano, considerándolo como nuestro principal activo, y por supuesto, comprometernos en ir adoptando continuamente mejoras tecnológicas a nuestros sistemas informáticos.

### 2.1.4 Modelo de Negocios

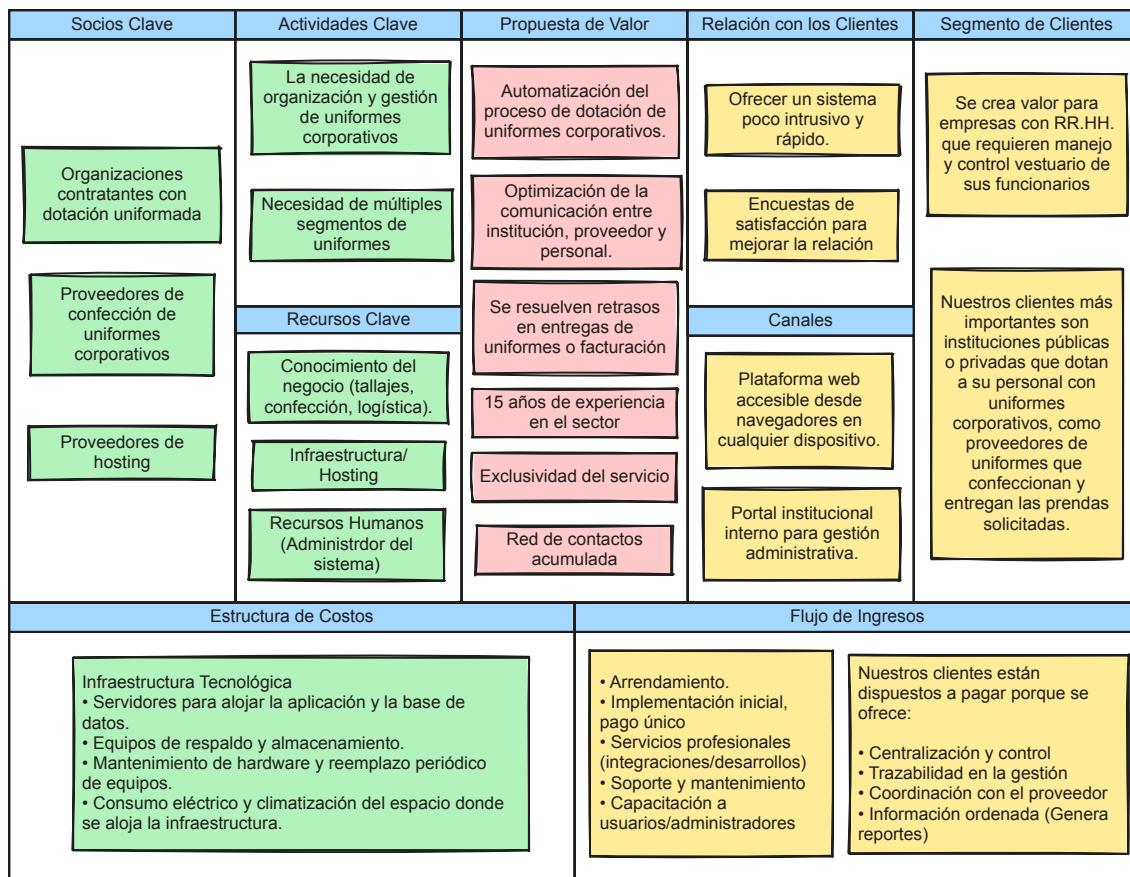
GyV Inversiones Ltda. se dedica al desarrollo, comercialización y arriendo de sistemas de gestión orientados principalmente a la industria textil. Desde su fundación en 2009, la empresa ha centrado sus esfuerzos en ofrecer soluciones informáticas que optimizan los procesos productivos y administrativos de las compañías del rubro, evolucionando hacia plataformas que integran a la gran empresa, sus colaboradores y proveedores. Su principal línea de negocio se enfoca en proveer sistemas especializados en la gestión de uniformes corporativos, destacando la plataforma SISTAL (Sistema de Tallas y Logística), diseñada para facilitar la administración y coordinación entre las empresas que comercializan uniformes, sus clientes y usuarios finales.



Los ingresos obtenidos por GyV Inversiones Ltda., corresponde principalmente al arriendo de plataformas de gestión, soporte informático de sistemas a medida, asesorías en diversos sistemas ERP como Softlan, Random, Manager y otras asesorías en Access y Excel.

El modelo de negocios actual presenta algunas características primordiales para el desarrollo de la operación. Una de las más importantes es el equipo de trabajo, demostrando alto profesionalismo en términos técnicos y experiencia en las actividades del negocio, la propuesta de valor tiene por objetivo ofrecer un servicio integral y de excelencia a los clientes, comprendiendo sus necesidades para ofrecerles la solución más eficiente para sus problemáticas. Se puede destacar además la mejora continua con respecto a los costos de la operación, privilegiando en lo posible el reacondicionamiento de equipos y reciclaje de insumos, permitiendo tener una estructura mínima de costos.

Basándonos en las características identificadas, se diseñó un esquema del modelo de negocios que se muestra en la [Figura 2.1](#), que permite tener una vista completa del modelo de negocios e identificar el valor que entrega el proyecto a la organización.



**Figura 2.1:** Lienzo Canva del modelo de negocios actual de la empresa.

Fuente: Diseño Propio

Como se observa, la propuesta de valor se conecta con todo el modelo de negocios: los clientes buscan precisamente una solución que centralice la gestión, reduzca retrasos y mejore la comunicación, por lo que el sistema se vuelve atractivo para áreas de Recursos Humanos y proveedores de confección. Asimismo, el funcionamiento del sistema depende de la participación coordinada de instituciones que requieren dotación uniformada, proveedores de confección y servicios de hosting. Estos socios aportan insumos, logística y soporte técnico que permiten que la automatización prometida sea efectiva y genere valor para todos los involucrados.



Para cumplir esa promesa, el conocimiento del rubro, infraestructura tecnológica y la experiencia acumulada, permite entregar un servicio confiable a través de una plataforma web accesible. Esto también determina los costos del proyecto, centrados en la tecnología, y la forma en que se generan ingresos, mediante arrendamiento, implementación y soporte.

### 2.1.5 Análisis FODA

En la [Tabla 2.1](#), se muestra el análisis FODA realizado, que lista los atributos identificados de la empresa que afectan el desarrollo y la sostenibilidad del sistema, como del modelo de negocio asociado.

**Tabla 2.1:** Análisis FODA con principales atributos identificados de la empresa

Fuente: *Elaboración Propia*

Fortalezas	Oportunidades
<ul style="list-style-type: none"><li>■ Plataforma integral que centraliza datos y procesos.</li><li>■ Propuesta de valor clara: eficiencia operativa y toma de decisiones basada en información.</li><li>■ Integración fluida con socios clave (proveedores tecnológicos, instituciones externas, sistemas colaboradores).</li><li>■ Conocimiento profundo del dominio del negocio.</li></ul>	<ul style="list-style-type: none"><li>■ Crecimiento de la demanda por sistemas integrados y analíticos.</li><li>■ Posibilidad de expandir el ecosistema con nuevos socios y alianzas o soporte de nuevos productos.</li><li>■ Tendencia del mercado a la digitalización de procesos.</li><li>■ Interés de clientes en reducir costos operativos mediante plataformas centralizadas.</li></ul>
Debilidades	Amenazas
<ul style="list-style-type: none"><li>■ Complejidad técnica que dificulta la incorporación de nuevos usuarios o mantenedores.</li><li>■ Procesos internos que aún requieren formalización o documentación.</li><li>■ Deuda técnica y limitación de escalabilidad.</li></ul>	<ul style="list-style-type: none"><li>■ Cambios constantes en estándares tecnológicos o regulatorios.</li><li>■ Riesgo de dependencia excesiva de algunos socios clave.</li><li>■ Expectativas crecientes de los clientes respecto a tiempos y calidad del servicio.</li><li>■ Vulnerabilidades inherentes a cualquier sistema digital frente a ciberataques.</li></ul>



A partir del análisis previamente expuesto, se construyó una matriz FODA presentada en la [Tabla 2.2](#), que cruza las variables internas y externas, para derivar un conjunto de estrategias orientadas a fortalecer la posición de la organización. La matriz resultante constituye una guía estratégica para la toma de decisiones y la evolución futura del sistema.

**Tabla 2.2:** Matriz FODA con análisis estratégicos

Fuente: *Elaboración Propia*

	<b>Fortalezas</b>	<b>Debilidades</b>
Oportunidades	<b>Estrategias FO:</b> <ul style="list-style-type: none"><li>■ Usar la integración fluida y el conocimiento del negocio para acelerar la creación de módulos o conectores que permitan sumar nuevos aliados, diversificar servicios y ampliar la cobertura del ecosistema.</li></ul>	<b>Estrategias DO:</b> <ul style="list-style-type: none"><li>■ La tendencia hacia sistemas integrados y fáciles de usar justifica inversiones en interfaces más intuitivas, automatización interna y reducción de complejidad técnica.</li><li>■ La incorporación de nuevos actores exige estándares más altos de orden y documentación, lo cual ayuda a superar brechas internas existentes.</li></ul>
Amenazas	<b>Estrategias FA:</b> <ul style="list-style-type: none"><li>■ La plataforma puede responder con mayor agilidad a cambios normativos al tener procesos y datos unificados, reduciendo impacto y tiempos de adaptación.</li></ul>	<b>Estrategias DA:</b> <ul style="list-style-type: none"><li>■ Fortalecer la arquitectura para evitar riesgos frente a regulaciones nuevas, aumentos de demanda o exigencias de rendimiento.</li><li>■ Disminuir complejidad interna y mejorar orden técnico para reducir la superficie de ataque y fortalece la resiliencia del sistema.</li></ul>

El análisis FODA y la matriz estratégica permiten obtener una visión clara del estado actual de la organización y del sistema, como de las rutas de acción más adecuadas para potenciar su desarrollo.

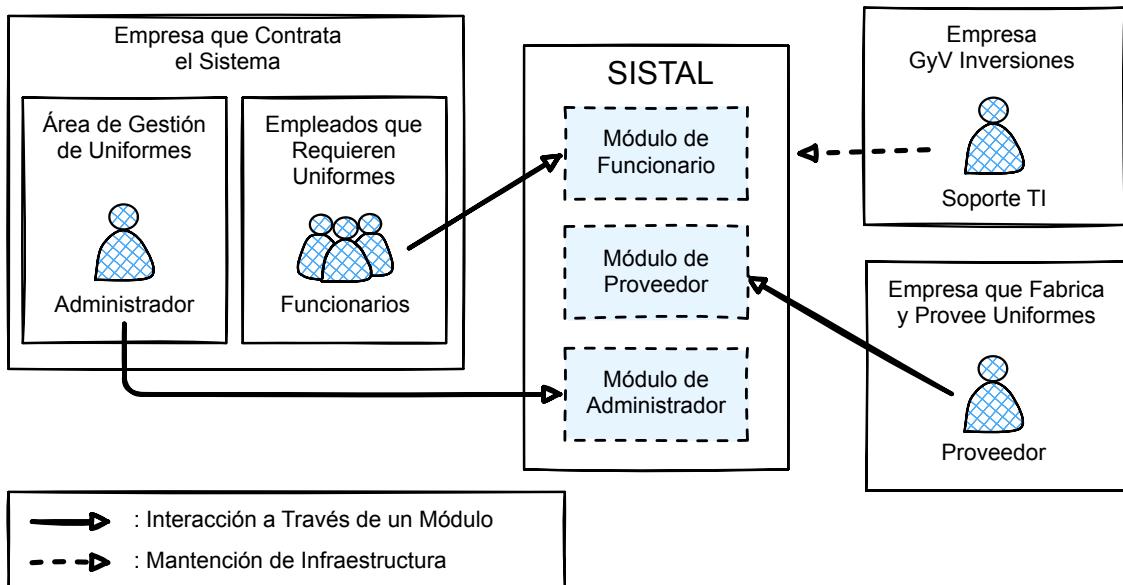


## 2.2 DESCRIPCIÓN DEL SISTEMA (SISTAL)

Como se ha mencionado anteriormente, SISTAL es un sistema web de gestión de uniformes corporativos, desarrollado para atender a organizaciones que requieren administrar de manera eficiente la dotación de uniformes a su personal. Este sistema cubre todo el flujo natural de dotación de uniformes, desde el registro de tallas de los funcionarios, hasta la confección y despacho del proveedor de uniformes. Para comprender bien todo este flujo, se comenzará explicando sobre los principales actores que interactúan en este sistema y sus respectivos módulos.

### 2.2.1 Actores Principales y sus Módulos

El sistema considera **tres actores principales**, denominados **Funcionario, Administrador y Proveedor**, los cuales interactúan tanto al interior de las organizaciones como dentro de la propia plataforma. Cada uno de ellos cumple un rol específico y dispone de una interfaz o módulos particulares que facilitan sus tareas dentro del proceso de gestión de uniformes corporativos. En la [Figura 2.2](#) se presenta un diagrama que representa cómo estos actores interactúan con el sistema.



**Figura 2.2:** Diagrama explicativo de actores del sistema

Fuente: *Diseño Propio*

Como se señaló previamente, los actores interactúan con una **Interfaz Gráfica de Usuario, o Módulo Funcional**, la cual se encuentra directamente asociada al rol que desempeña cada actor y que agrupa las tareas que cada uno puede realizar. A continuación se explican los tres actores y las funciones que puede realizar en su respectivo módulo.

- **Funcionario:** Representa al trabajador o personal de la organización que **utiliza uniformes corporativos como parte de su vestimenta laboral**. A través del módulo Funcionario de SISTAL, este usuario puede registrar y actualizar sus medidas de talla desde cualquier dispositivo con conexión a internet<sup>1</sup>, permitiéndole revisar la información de sus prendas asignadas y solicitar cambios o reposiciones cuando sea necesario.
- **Proveedor:** Representa a un trabajador de la empresa externa encargada de

<sup>1</sup>Al ser un sistema web, se puede acceder desde cualquier dispositivo que soporte un navegador, siendo una de las propuestas de valor iniciales del sistema.



**la confección y entrega de los uniformes corporativos.** A través del módulo Proveedor de SISTAL, el proveedor puede acceder a la información consolidada de tallas, cantidades y tipos de prendas requeridas, la cual puede descargar o consultar en línea para gestionar la producción. También puede actualizar el estado de los pedidos, registrar despachos y facturas.

- **Administrador:** Representa al trabajador de la institución que requiere dotar de uniformes a su personal y que cuenta con privilegios de administración a través del módulo Administrador de SISTAL. Es el responsable de **solicitar uniformes al proveedor, así como de gestionar y supervisar todo el proceso de dotación dentro de la organización solicitante.** Sus funciones incluyen gestionar funcionarios, controlar pedidos, aprobar solicitudes, generar reportes y mantener actualizada la información institucional. Asimismo, coordina la comunicación con el proveedor para asegurar la trazabilidad de los procesos y el cumplimiento de las políticas de dotación.

La [Figura 2.3](#) presenta la pantalla de inicio actual del sistema, cada actor puede acceder a su respectivo módulo desde aquí. Cuando uno de ellos ingresa sus credenciales (Usuario y Contraseña), el sistema automáticamente reconoce el su rol y le muestra el módulo correspondiente. Actualmente, un funcionario ingresaría con su RUT como su credencial de usuario, en caso de ser un Administrador o Proveedor, este tiene sus credenciales de usuario específicas.



INSET DATA

**SISTAL**

Bienvenido al Portal de Beneficios Corporativos

Usuario (Ej: 09458985-K)

Contraseña (Ej: 09458985-K)

Ingresar

Recuperar contraseña.

Email registrado:

Recuperar

Si tienes dudas y/o consultas sobre el proceso de la entrega del Uniforme Corporativo, puedes contactarnos telefónicamente al XXXXXX o al correo XXXXXXX@XXXXXXXX.cl

**Figura 2.3:** Página de inicio de sesión. Presenta los campos de Usuario, Contraseña y Recuperar contraseña con un *email* registrado.

Fuente: Interfaz de SISTAL alojada en un entorno de pruebas

Las figuras Figura 2.4, Figura 2.5 y Figura 2.6 presentan una vista de las interfaces gráficas de cada módulo con motivos ilustrativos, todos los módulos tienen un menú a la izquierda con botones para acceder a las distintas funciones, algunos de estos botones despliegan nuevos menús al pasar el cursor por encima. Una explicación más a fondo de estos módulos y sus funciones se encuentra en el Capítulo 3, Sección 3.2.



Cambiar Tallas	▼
Cambiar Sucursal	▼
Cambiar Contraseña	▼
Cerrar Sesión	▼

DIRECCIÓN LABORAL Y SEGMENTO		
DIRECCIÓN SUCURSAL	REGIÓN	SEGMENTO
[REDACTED]	11 REGION DE AYSEN	SEGMENTO TERRENO HOMBRE

DATOS PERSONALES	
NOMBRES	APELLIDO PAT APELLIDO MAT, NOMBRES
RUT	11111111-1
EMAIL	11111

INFORMACION DE TALLAS, COMPLETE TODOS LOS CAMPOS CLICK EN LA PRENDAS PARA VER CONVERSIÓN				
PRENDAS	TALLAS	LARGOS	MEDIDAS	
<a href="#">CHALECO</a>	M		ESTATURA	1.75
<a href="#">PANTALON</a>	48	PANTALON 108	BUSTO / PECHO(cm)	55
<a href="#">CAMISA</a>	40		CINTURA(cm)	55
<a href="#">SOFTSHELL</a>	M		CADERA(cm)	55

Comentario	[REDACTED]
------------	------------

REVISE BIEN SUS DATOS ANTES DE GUARDAR LA SOLICITUD.  
LOS DATOS INGRESADOS SERAN DE SU RESPONSABILIDAD.

**Guardar**

**Figura 2.4:** Interfaz del módulo de funcionario. Muestra datos del funcionario, su sucursal de la empresa y sus tallas que puede editar.

Fuente: *Interfaz de SISTAL alojada en un entorno de pruebas*



Cortes Disponibles ▾  
Funcionarios ▾  
Procesos ▾  
Consultas ▾  
Cambiar Contraseña  
Cerrar sesión ▾

### SEGMENTO MASCULINO

[Cortes temporadas anteriores](#)

Seleccione N° Corte 5 ▾

Total 6 Funcionarios enviados el 17-01-2018

[Enviar Excel](#)

Rut	Nombre Completo	Base	Maternal	Red	Direccion Laboral	Comuna
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]

**Figura 2.5:** Interfaz del módulo de proveedor. Muestra los funcionarios enviados por el administrador de la empresa solicitante que les corresponde uniforme.

Fuente: *Interfaz de SISTAL alojada en un entorno de pruebas*

Buscar Trabajador ▾  
Mantenedor ▾  
Nuevas Solicitudes ▾  
Envío a Corte ▾  
Consultas ▾  
Cambios Solicitudes ▾  
Facturación ▾  
Exportar Excel ▾  
Cambiar Contraseña  
Cerrar sesión ▾  
Nueva Temporada ▾  
Otras Temporadas ▾

### NUEVAS SOLICITUDES

Aprobar o rechazar: ▾ ENVIAR

[Enviar Excel](#)

[Marcar | Desmarcar](#)

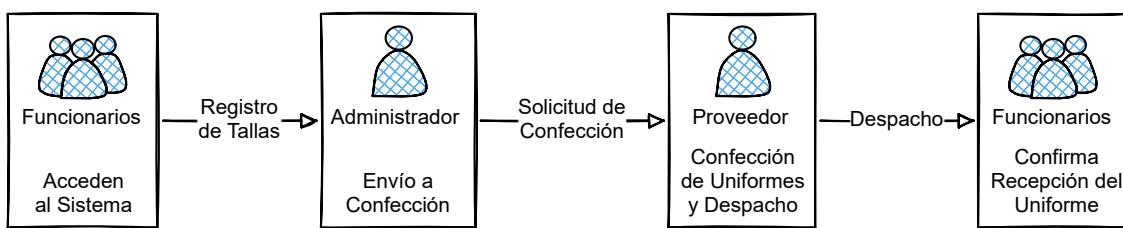
Enviar	Rut	Nombre Completo	Segmento	Tipo	Fecha Solicitud
<input type="checkbox"/>	11111111-1	APELLIDO PAT APELLIDO MAT NOMBRES	SEGMENTO TERRENO HOMBRE	NUEVO	2018-03-12
<input type="checkbox"/>	22222222-2	APELLIDO PATERNO APELLIDO MATERNO NOMBRES	SEGMENTO MASCULINO	NUEVO	2018-03-18
<input type="checkbox"/>	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
<input type="checkbox"/>	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
<input type="checkbox"/>	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
<input type="checkbox"/>	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]

**Figura 2.6:** Interfaz del módulo de administrador. El menú que se muestra permite al administrador aprobar o rechazar los funcionarios nuevos que se registraron en el sistema.

Fuente: *Interfaz de SISTAL alojada en un entorno de pruebas*

## 2.2.2 Procesos que Aborda SISTAL

Para comprender el flujo de información dentro del sistema, en la [Figura 2.7](#) se presenta, de forma general y simplificada, el proceso de dotación de uniformes al personal de una empresa cliente. Una explicación más detallada de estos flujos, junto con sus diagramas específicos, se encuentra en el [Capítulo 3, Sección 3.4](#).



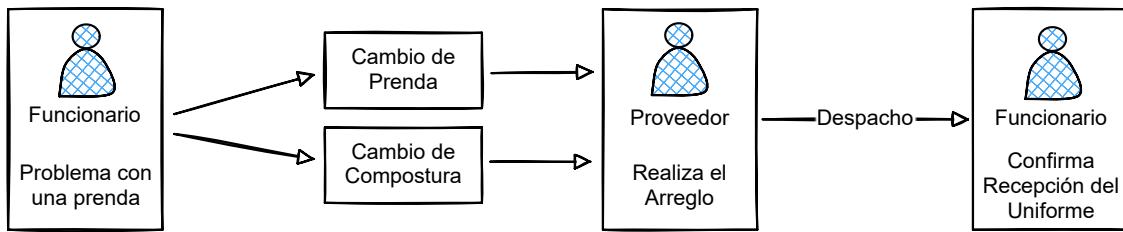
**Figura 2.7:** Flujo simplificado del proceso de dotación de uniformes apoyado por SISTAL

Fuente: *Diseño Propio*

El proceso se inicia cuando un funcionario requiere un uniforme y se registra en SISTAL para ingresar sus tallas correspondientes. Una vez enviadas, estas quedan almacenadas en el sistema junto con las del resto de los funcionarios, lo que permite al administrador continuar con la **solicitud de confección**. En esta etapa, se remite al proveedor toda la información consolidada sobre modelos, cantidades y medidas necesarias para la elaboración de las prendas.

Luego de la confección de los uniformes, el proveedor inicia el **despacho**, en este caso, los uniformes pueden ser enviados a la sucursal correspondiente del funcionario, pero puede darse el caso de que el funcionario pueda retirar su uniforme en la misma instalación del proveedor, en este caso, el proveedor aprueba la recepción directamente.

Posterior a la entrega del uniforme, el funcionario puede solicitar un **arreglo**, representado en la [Figura 2.8](#).



**Figura 2.8:** Flujo simplificado de solicitud de arreglo

Fuente: *Diseño Propio*

Este puede corresponder a dos tipos de procesos: el **cambio de prenda**, solicitado cuando alguna pieza del uniforme presenta un problema específico; y la **compostura**, mediante la cual se aplican ajustes o modificaciones a la misma prenda recibida. En este proceso es el funcionario quien directamente se comunica con el proveedor, dejando evidencia de las peticiones al administrador.

Con este repaso del sistema, del negocio al que entrega su solución y de su flujo de información, se dispone del contexto necesario para avanzar hacia el análisis técnico, instancia en la que se abordarán y examinarán en detalle las limitaciones del sistema.



## CAPÍTULO 3

### ANÁLISIS TÉCNICO DEL SISTEMA ACTUAL

En el siguiente capítulo se documenta el diseño y especificaciones técnicas del sistema actual, representando su estructura interna y sus principales componentes a través de diagramas de casos de uso, base de datos, flujo, arquitectura y despliegue. Se busca comprender la organización del sistema existente y las áreas que requieren modernización.

### 3.1 INFRAESTRUCTURA

Se comenzará el capítulo abordando la infraestructura del sistema, debido a la relevancia que esta tiene en el contexto del proyecto y en la comprensión de sus fundamentos técnicos. Se observarán las bases sobre las cuales se sostiene el servicio, así como los recursos que habilitan su operación diaria

#### 3.1.1 Tecnologías y Herramientas

A continuación describen las tecnologías y herramientas que permiten el funcionamiento del sistema:

- **PHP 5.6 como lenguaje de programación**

Elegido por su amplia adopción en el desarrollo web, su curva de aprendizaje accesible y su capacidad para construir páginas dinámicas de manera eficiente.

- **HTML y CSS como lenguajes de marcado para la interfaz de usuario**

Utilizados para estructurar y dar estilo a las vistas web del sistema.



- **MySQL como motor de base de datos**

Caracterizado por su estabilidad, facilidad de administración y compatibilidad con PHP, además de ser una solución ampliamente utilizada en aplicaciones web.

- **Servidor de alojamiento BlueHosting**

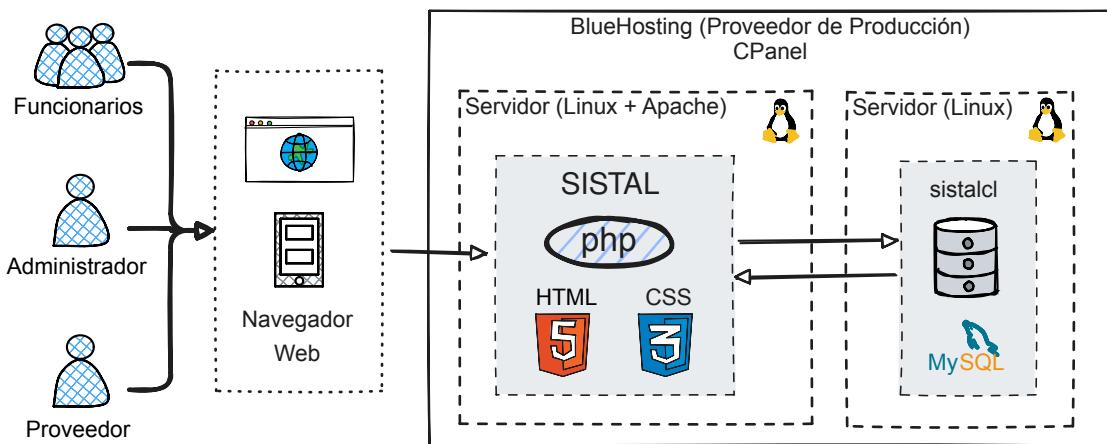
Proveedor que ofrece soporte para PHP y MySQL, garantizando la disponibilidad del sistema y simplificando su despliegue mediante un entorno compartido de fácil administración.

Dado que el sistema se aloja en BlueHosting, el proveedor pone a disposición un entorno administrado mediante cPanel, una plataforma ampliamente utilizada para gestionar servicios de alojamiento web. A través de cPanel es posible administrar aspectos clave del servidor (como dominios, bases de datos, cuentas de correo y archivos) utilizando herramientas comunes del alojamiento compartido, las cuales son **Apache**, **MySQL/MariaDB** y soporte para **PHP**. Esta combinación conforma una pila tecnológica típica en entornos de hosting compartido, similar a la arquitectura LAMP tradicional.

La pila LAMP se caracteriza por su estabilidad, facilidad de implementación y amplia compatibilidad entre sus componentes, lo que la ha convertido en una opción común para proyectos web pequeños y medianos, especialmente aquellos orientados a entornos de hosting compartido.

### 3.1.2 Arquitectura

En la [Figura 3.1](#), se muestra el diagrama de la arquitectura actual de SISTAL, con las herramientas tecnológicas utilizadas y actores que acceden desde un navegador web al sistema.



**Figura 3.1:** Arquitectura actual de SISTAL

Fuente: Diseño Propio

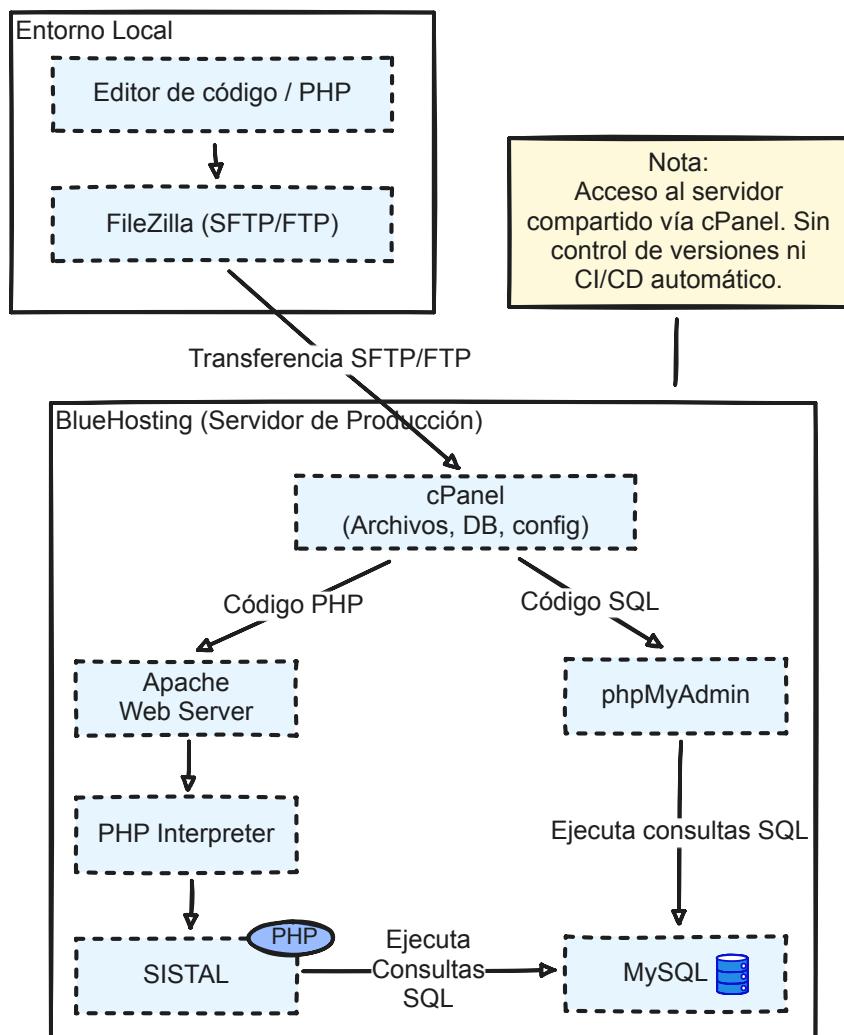
Dentro del proveedor existe un servidor Linux con Apache, el permite que el sistema donde se ejecuta SISTAL desarrollada en PHP, HTML y CSS. Este servidor es el encargado de recibir las peticiones del navegador, procesarlas y devolver las respuestas correspondientes.

El sistema se comunica con otro servidor dedicado a la base de datos, donde se aloja el motor MySQL y el esquema. Este servidor maneja exclusivamente la información almacenada y responde a las consultas que le realiza SISTAL.

cPanel actúa como intermediario que unifica todas estas herramientas, siendo la plataforma de administración que organiza los servidores dentro de BlueHosting, sin intervenir en el flujo directo que ocurre entre los usuarios, la aplicación y la base de datos.

### 3.1.3 Proceso de Despliegue y Actualización

En el diagrama de la [Figura 3.2](#) se muestra el proceso manual de despliegue y actualización del sistema, en el que desde un entorno local, a través del protocolo SFTP/FTP, se realiza una conexión al proveedor BlueHosting por medio de algún cliente como FileZilla, y se procede a añadir/reemplazar el código del sistema.



**Figura 3.2:** Diagrama de despliegue del sistema actual

Fuente: Diseño Propio



### 3.1.4 Principales Problemas de esta Infraestructura

Como se mostró anteriormente en la [Figura 3.1](#), SISTAL funciona como un único programa, lo que lo convierte en un sistema monolítico. Actualmente, está compuesto por aproximadamente 400 archivos de código PHP que contienen la lógica y el código del sistema. Estos archivos carecen de una estructura definida, se encuentran dispersos en el repositorio sin un orden claro y utilizan nombres ambiguos, lo que dificulta considerablemente el análisis del código.

También, como PHP permite “incrustar” directamente código HTML dentro de sus archivos, provoca que la lógica y la interfaz gráfica de usuario convivan en un mismo lugar sin una separación estricta entre vista y lógica (Un ejemplo de esto se muestra en la [Figura 3.3](#)).

---

```
1 // Trozo de solo PHP
2 <?php
3     $usuario = "María";
4     $mensaje = "Bienvenida al sistema";
5 ?>
6
7 // Trozo de HTML y PHP
8 <h1>SISTAL</h1>
9 <p><?php echo "$mensaje, $usuario."; ?></p>
```

---

## SISTAL

Bienvenida al sistema, María.

**Figura 3.3:** Ejemplo de código PHP con código HTML incrustado más su salida generada.

Fuente: *Elaboración Propia*

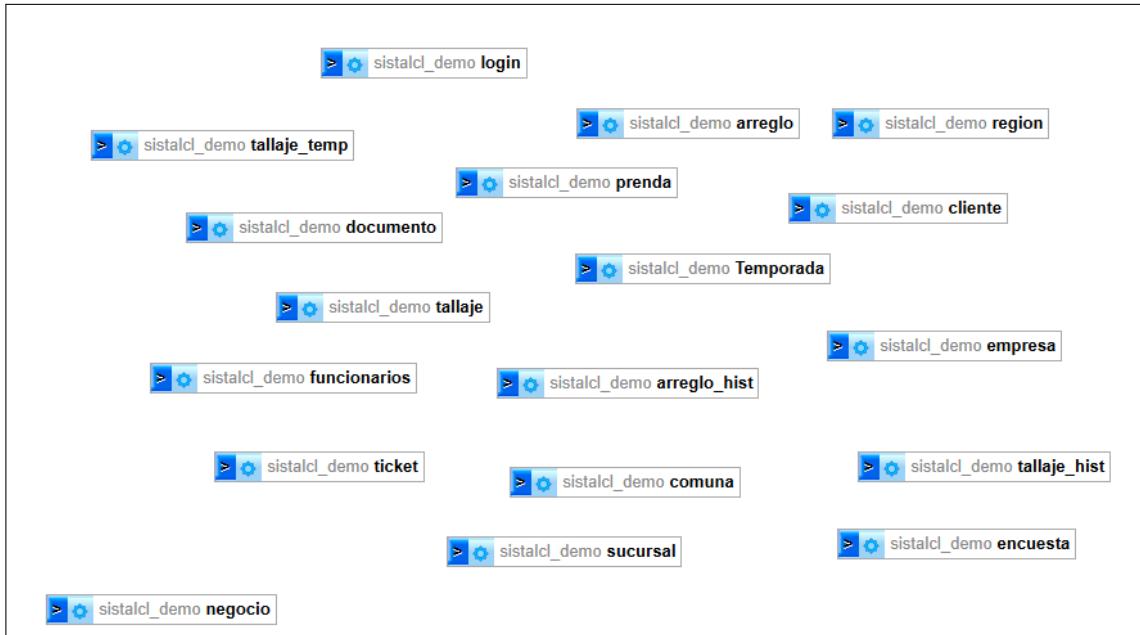
Como resultado, es común ver estructuras de control, variables y funciones PHP mezcladas con etiquetas HTML, facilitando la creación rápida de páginas dinámicas, aunque reduciendo la separación de responsabilidades típica de ar-



quitecturas más modernas. Una breve mención al patrón de arquitectura Modelo-Vista-Controlador (MVC), un patrón muy común que soluciona esto en sistemas monolíticos.

Esta falta de organización se conoce entre los arquitectos de software como *Big Ball of Mud* (gran bola de barro), un anti-patrón arquitectónico donde todos los componentes dependen entre sí de forma descontrolada, haciendo que cualquier modificación implique un alto riesgo de generar errores en otras partes del sistema.

Por otro lado, en la [Figura 3.4](#) se muestran las tablas de la base de datos obtenidas del software phpMyAdmin que permite exportar un diagrama con las tablas desde MySQL, más adelante en la [Sección 3.3](#) se encuentran las tablas completas y sus funciones.



**Figura 3.4:** Tablas de la Base de Datos MySQL

Fuente: Captura de pantalla de phpMyAdmin ofrecido en BlueHosting

La base de datos está conformada por tablas aisladas, sin relaciones ni re-



glas de integridad definidas. Las asociaciones entre los datos no están declaradas en la base de datos, sino que han sido incorporadas directamente en el código de la aplicación mediante lógica programada explícitamente, lo que dificulta mantener la coherencia, comprender la estructura real del modelo y realizar cualquier mejora sin afectar el funcionamiento general del sistema.

Además de estas observaciones estructurales, también se identificaron otros aspectos técnicos y operativos que afectan el funcionamiento y mantenimiento del sistema:

- **Procesos de desarrollo:** No existe control de versiones ni seguimiento de cambios a lo largo del tiempo. Tampoco se cuenta con mecanismos de despliegue automatizado, ni con ambientes separados para desarrollo, pruebas y producción, lo que dificulta la trazabilidad y aumenta el riesgo de errores.
- **Infraestructura:** El sistema y la base de datos se ejecutan en un servicio de alojamiento compartido, lo que genera latencias, inestabilidad y la ausencia de herramientas de observabilidad o configuraciones de alta disponibilidad.
- **Seguridad:** Los controles de acceso y mecanismos de cifrado son insuficientes, y no se aplican buenas prácticas ni estándares modernos de seguridad.
- **UX/UI:** La interfaz presenta un diseño obsoleto, poco intuitivo y sin capacidad de adaptarse adecuadamente a distintas resoluciones o dispositivos.
- **Escalabilidad:** Existen limitaciones para atender a múltiples empresas simultáneamente y para integrar servicios externos como proveedores, logística o sistemas ERP.

Una vez analizada la infraestructura del sistema y expuestas sus principales desventajas, podemos avanzar hacia la documentación del funcionamiento interno del sistema, examinando sus módulos, flujos y componentes operativos.



## 3.2 DIAGRAMAS DE CASOS DE USO

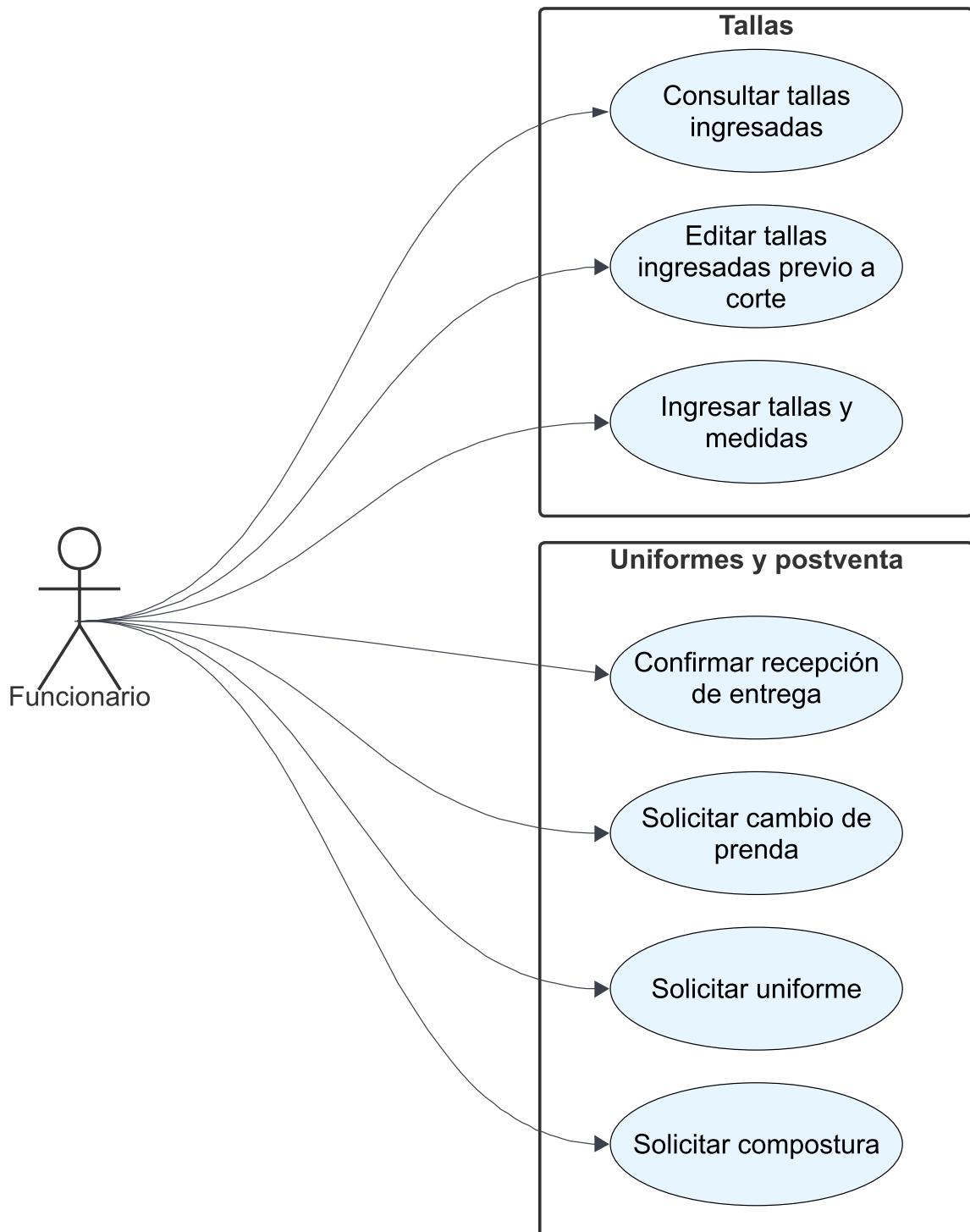
En los diagramas de casos de uso de a continuación, se describen todas las acciones que puede realizar un actor en su respectivo módulo. Estos diagramas representan correctamente las necesidades de los clientes y usuarios del sistema, por lo que no será necesario un replanteamiento de estos.

### 3.2.1 Módulo de Funcionario

En los diagramas de la [Figura 3.5](#) y [Figura 3.6](#) se describen las acciones que un funcionario puede realizar en el sistema para gestionar su dotación. El foco está en la captura y actualización de tallas, la solicitud/recepción de uniformes y las consultas de seguimiento asociadas. dentro de las funciones se incluye:

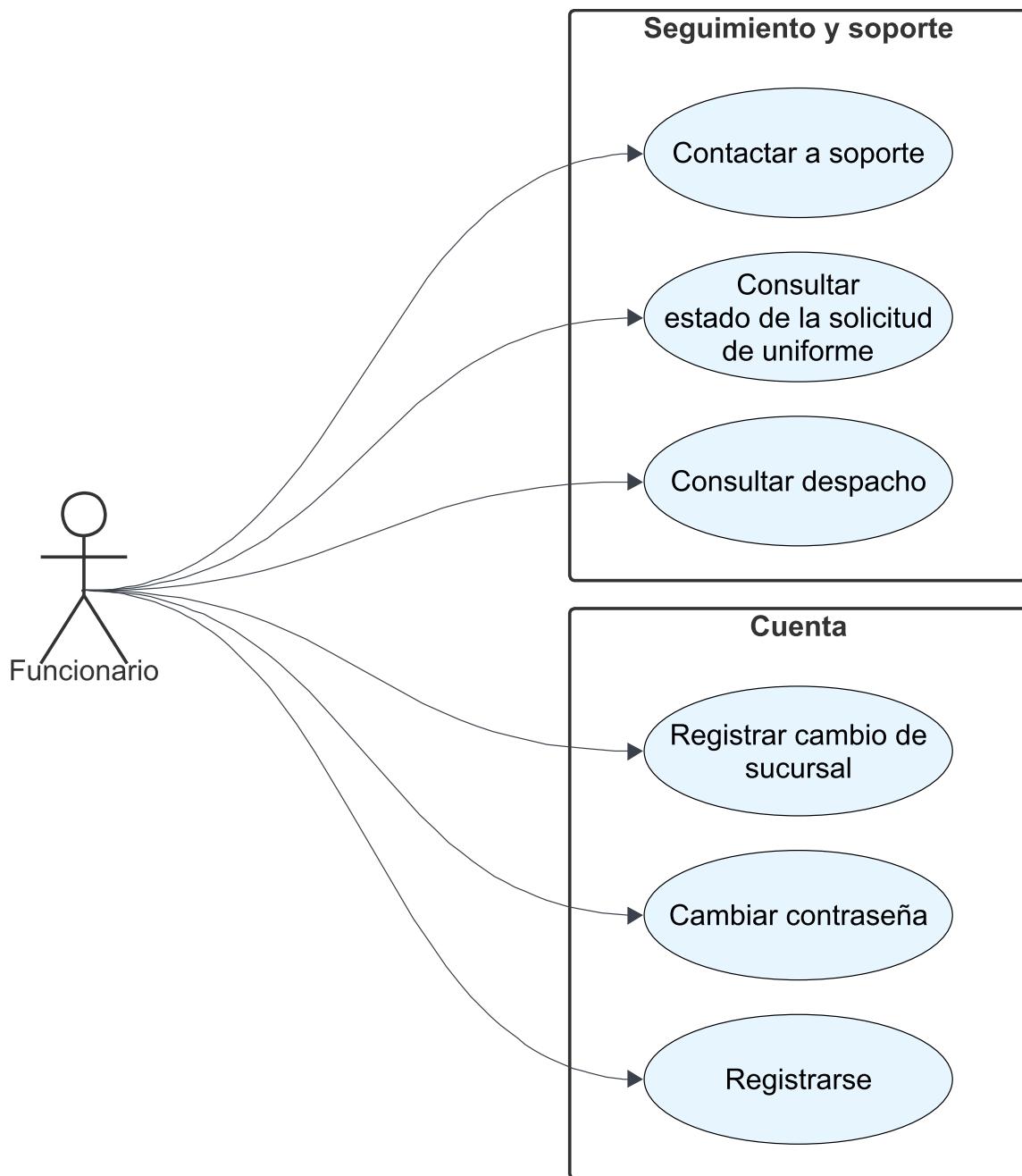
- **Gestión de tallas:** ingresar tallas y medidas, visualizar tallas registradas, solicitar cambios de talla.
- **Uniformes:** solicitar uniforme, informar la recepción del uniforme, solicitar cambio de prendas.
- **Seguimiento y comunicación:** revisar detalles del despacho del uniforme, cambiar sucursal donde trabaja, enviar comentarios.
- **Cuenta:** registro de nuevo funcionario y cambio de contraseña, donde cualquier tipo de usuario puede registrarse en el sistema con sus datos y depende del administrador aprobarlo.

Con ello, el funcionario mantiene su información actualizada y solicita dotaciones acordes a su perfil.



**Figura 3.5:** Diagrama de casos de uso - Funcionario (Parte 1)

Fuente: Diseño Propio



**Figura 3.6:** Diagrama de casos de uso - Funcionario (Parte 2)

Fuente: Diseño Propio

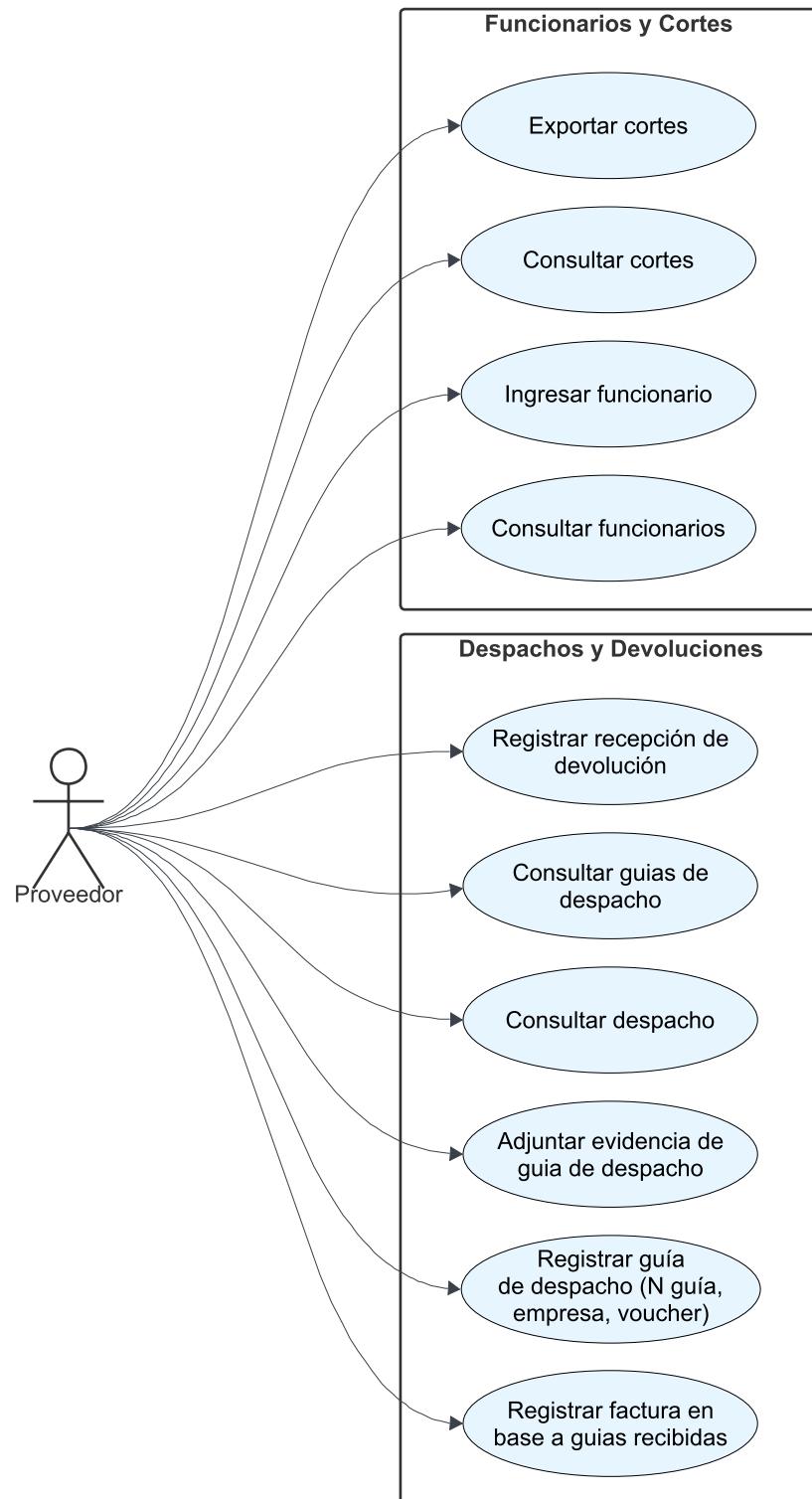


### 3.2.2 Módulo de Proveedor

El los diagramas de la [Figura 3.7](#) y [Figura 3.8](#) se representan las interacciones del proveedor responsable de confección, despacho y facturación. Se organiza en submódulos para facilitar la operación diaria y el control de avance:

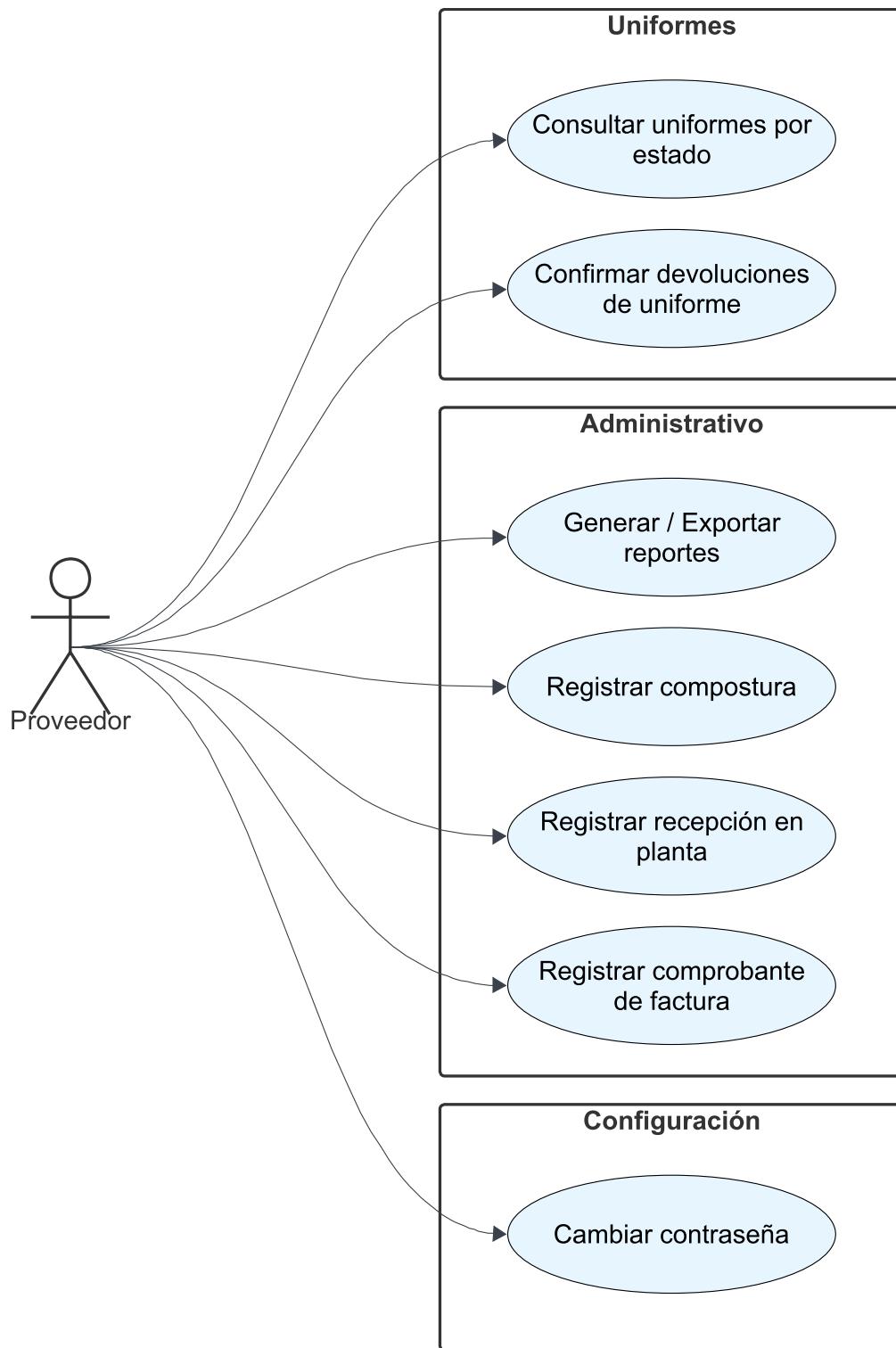
- **Gestión de funcionarios y tallas:** consultar funcionarios y curvas de talla por prenda/corte, ingresar tallas de un funcionario cuando aplique, exportar/consultar cortes (vigentes y de temporadas anteriores).
- **Despachos:** registrar guías de despacho (número, váucher y empresa), consultar detalles del despacho, confirmar la recepción de prendas devueltas.
- **Uniformes:** registrar entrega, confirmar devoluciones, consultar uniformes, des- pachados, recibidos y facturados.
- **Administrativo:** registrar datos de recepción, registrar factura, llevar registro de composturas.
- **Configuración:** cambiar sucursal y credenciales.

El diagrama evidencia el flujo completo del proveedor: desde la preparación (cortes) y confección, pasando por despacho y recepción, hasta la conciliación administrativa.



**Figura 3.7:** Diagrama de casos de uso - Proveedor (Parte 1)

Fuente: *Diseño Propio*



**Figura 3.8:** Diagrama de casos de uso - Proveedor (Parte 2)

Fuente: Diseño Propio

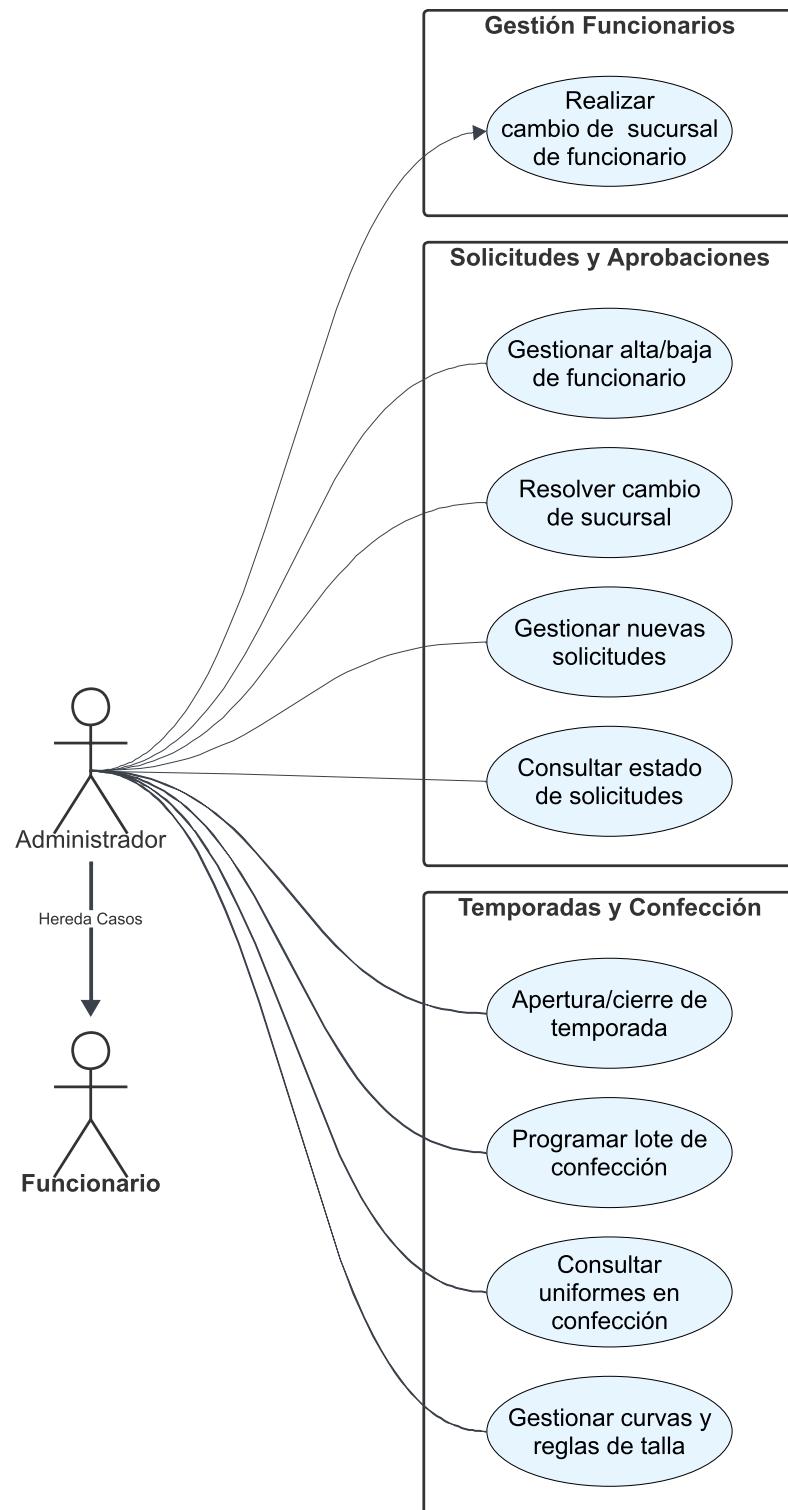


### 3.2.3 Módulo de Administrador

En los diagramas de la [Figura 3.9](#) y [Figura 3.10](#), el actor Administrador **hereda** todas las capacidades del *Funcionario* y añade funciones de control, operación y reportería a nivel global. Sus casos de uso propios se agrupan de la siguiente forma:

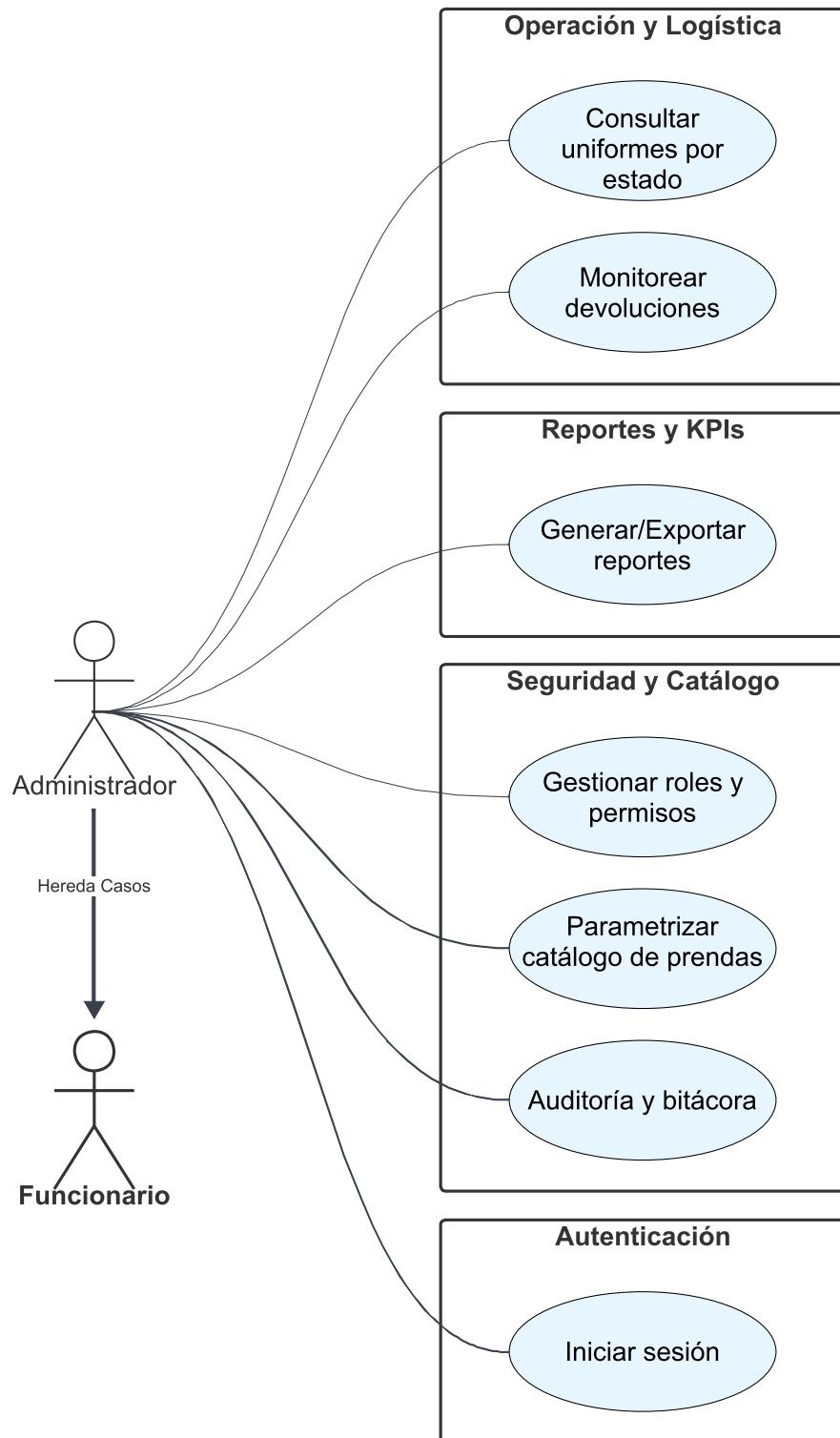
- **Gestión de solicitudes:** gestión de nuevas solicitudes, consulta de estado, revisión de cambios de sucursal, rechazo de funcionarios ya ingresados cuando corresponda.
- **Temporadas y confección:** apertura/cierre de temporada, iniciar procesos de confección para funcionarios con talla, consulta de uniformes en confección, consulta de curvas de tallas por proceso y de funcionarios sin talla.
- **Logística de uniformes:** consulta consolidada de uniformes recibidos, despa-chados y facturados.
- **Control de cambios de talla:** revisión por funcionario y por prenda para asegurar consistencia y trazabilidad.
- **Reportes/Exportaciones:** exportar encuestas y tallas de funcionarios, exportar totales y consultar detalle de facturas.
- **Autenticación:** inicio de sesión para acceso a módulos administrativos.

De esta forma, el administrador supervisa el ciclo completo de dotaciones y toma decisiones operativas con reportes del sistema.



**Figura 3.9:** Diagrama de casos de uso - Administrador (Parte 1)

Fuente: *Diseño Propio*



**Figura 3.10:** Diagrama de casos de uso - Administrador (Parte 2)

Fuente: *Diseño Propio*



### 3.3 DIAGRAMA DE BASES DE DATOS

El diagrama de base de datos presentados en ambas [Figura 3.11](#) y [Figura 3.12](#), son las tablas que se encuentran actualmente en el sistema. Como se aprecia, no existen relaciones entre ellas, muchas de los atributos no se utilizan y los nombres de las tablas son poco descriptivos. A continuación se realiza una explicación de cada una de las tablas:

- **Arreglo:** Se almacenan los cambios o composturas de prenda de cada funcionario.
- **Arreglo\_hist:** Registra los cambios o composturas de temporadas anteriores.
- **Cliente:** Se almacenan los proveedores de uniformes corporativos de cada empresa.
- **Comuna:** Listado de comunas de Chile.
- **Documento:** Registra datos de la tabla funcionarios, para almacenar temporadas anteriores.
- **Empresa:** Tabla que se vincula con la tabla Negocio, para el acceso de funcionarios nuevos.
- **Encuesta:** Se realizó una encuesta a los usuarios para calificar el sistema y se almacenó en esta tabla.
- **Funcionarios:** Se almacenan todos los funcionarios pertenecientes a la empresa y los pendientes de autorización.
- **Login:** Se almacenan credenciales de acceso para los actores del sistema.
- **Negocio:** Se almacena el segmento definido por cada empresa.
- **Prenda:** Se almacenan las prendas definidas por la empresa que se asignan de acuerdo al segmento.
- **Region:** Listado de regiones de Chile.
- **Sucursal:** Se almacenan las sucursales definidas por cada empresa que utiliza



el sistema.

- **Tallaje:** Se acumulan las tallas de los funcionarios de acuerdo a la estructura del segmento.
- **Tallaje\_hist:** Se almacena un historial de tallajes de cada funcionario, a medida que se va cambiando de temporadas.
- **Tallaje\_temp:** Registra momentáneamente el RUT del funcionario y las prendas que le corresponde.
- **Temporada:** Indica las temporadas vigentes y anteriores.
- **Ticket:** Pensada para la impresión de código de barra por cada uniforme.



Funcionarios	
ID	int
IDSucursal	int
IDNegocio	string
Rut	string
Nombre	string
ApellidoP	string
ApellidoM	string
Sexo	string
Cargo	string
Email	string
Telefono	string
Celular	date
Tipo	string
Comuna	string
Altura	string
Busto	string
Cadera	string
Cintura	string
Manga	string
Estado	string
EstadoE	string
EstadoT	string
Despacho	string
DespachoN	string
DespachoF	string
DespachoFF	string
IDDocumento	int
Maternal	string
CentroC	string
DescripcionCentro	string
Observacion	string
Prioridad	string
Guia	int
FechaTalla	date
PVenta	string
FechaCambio	date
FEnvioCambio	date
FechaGuia	date
Voucher	string
EmpTransp	string
Factura	int
FechaFact	date
Repcion	string
FechaRecep	date
GuiaC	int
VoucherC	string
EmpTranspC	string
FechaRecepC	date
RepcionC	string
FRRecipCambioP	date
Valija	string

Arreglo	
ID PK	int
IDDocumento	int
IDProducto	int
IDProductoA	int
RutFun	string
TallaDespacho	string
TallaArreglo	string
Tipo	string
FechaIngreso	date
FechaEntrega	date
FechaRepcion	date
FechaCorte	date
FechaModificacion	date
Observacion	string
Cantidad	string
RepcionP	string
EntregaP	string
Rollingreso	string
Largo	int
Maternal	string

Sucursal	
ID Empresa	string
Codigo	int
Nombre	string
Direccion	string
SucursalR	string
Comuna	string
Region	string
Telefono	string
Email	string
Encargado	string

Empresa	
ID PK	int
IDNegocio	int
Rut	string
RSocial	string
RFantasia	string
Nombre	string
Giro	string
Direccion	string
Contacto	string
Telefono	string
FechaOC	date
NumeroOC	string
ResponsableOC	string
EmailFactE	string

Arreglo_hist	
IDProductoA	int
RutFun	string
TallaDespacho	string
TallaArreglo	string
FechaIngreso	date
FechaEntrega	date
FechaRepcion	date
Observacion	string
Cantidad	string
RepcionP	string
EntregaP	string
Largo	int
Maternal	string
Tipo	string
MAT	string

Temporada	
ID	int
Temporada	string
Anno	string
Observacion	string
Estado	string

Ticket	
id	int
RutFun	string

Cliente	
ID PK	int
RSocial	string
RFantasia	string
Rut	string
Nombre	string
ApellidoP	string
ApellidoM	string
Direccion	string
Comuna	string
Ciudad	string
Telefono	string
Email	string
Celular	string
Cargo	string

**Figura 3.11:** Diagrama físico de bases de datos del sistema actual (Parte 1)

Fuente: Diseño Propio



Documento	
ID PK	int
Rut	string
Nombre	string
ApellidoP	string
ApellidoM	string
IDNegocio	string
TipoPrenda	string
Tipo	string
IDSucursal	int
DespachoN	string
Guia	int
FechaGuia	date
Repcion	string
DespachoF	date
Factura	int
Voucher	string
FechaRecep	date
Despacho	string
FechaTalla	date
EmpTransp	string
FechaFactura	date
Temporada	int
FechaCierre	date
GuiaC	int
FEnvioCambio	date
RepcionC	string
VoucherC	string
FechaRecepC	date
FechaCambio	date
EmpTranspC	string
PVenta	string
Ciclo	string
Celular	date
Observacion	string
DespachoFF	string
FRrecepcambioP	date
Prioridad	string
Valija	string

Comuna	
IDRegion	string
Pais	string
Codigo	string
Nombre	string

Region	
ID PK	string
Pais	string
Region	string

Tallaje	
ID PK	int
RutFun	string
IDProducto	int
Talla	string
TallaFinal	string
Cantidad	string
FechalIngreso	date
FechaModificacion	date
Observacion	string
Estado	string
Confeccion	string
IDDocumento	int
FechaEnvio	date
FechaArreglo	date
LargoManga	string
Largo	string
Maternal	string

Tallaje_hist	
ID	int
RutFun	string
Confeccion	string
TallaFinal	string
LargoManga	string
Largo	string
IDProducto	int
Cantidad	string
FechalIngreso	date
FechaModificacion	date
Maternal	string
MAT	string

Tallaje_temp	
ID	int
RutFun	string
Confeccion	string

Encuesta	
ID	int
RutFun	string
respuesta1	string
respuesta2	string
respuesta3	string
respuesta4	string
respuesta5	string
respuesta6	string
respuesta7	string
respuesta8	string
respuesta9	string

Negocio	
ID PK	int
Nombre	string
IDCliente	int
FechalInicio	date
Contacto	string
Telefono	string
Email	string
Maternal	string
CantBajo	int
CantBajoVaron	int
Proveedor	string
CantBajoNuevo	int
CantBajoNuevoV	int
RED	int

Prenda	
ID PK	int
IDNegocio	int
CodigoCliente	string
Modelo	string
Cantidad	int
Descripcion	string
Temporada	string
PrecioVtaCliente	string
Tipo	string
Genero	string
Ubicacion	string
Anno	string
Estado	string
IDTemp	int

Login	
ID	int
rol	int
usuario	string
password	string
nombres	string
ApellidoP	string
ApellidoM	string
Rut	string
Cargo	string
email	string
Genero	string
IDNegocio	int



**Figura 3.12:** Diagrama físico de bases de datos del sistema actual (Parte 2)

Fuente: Diseño Propio

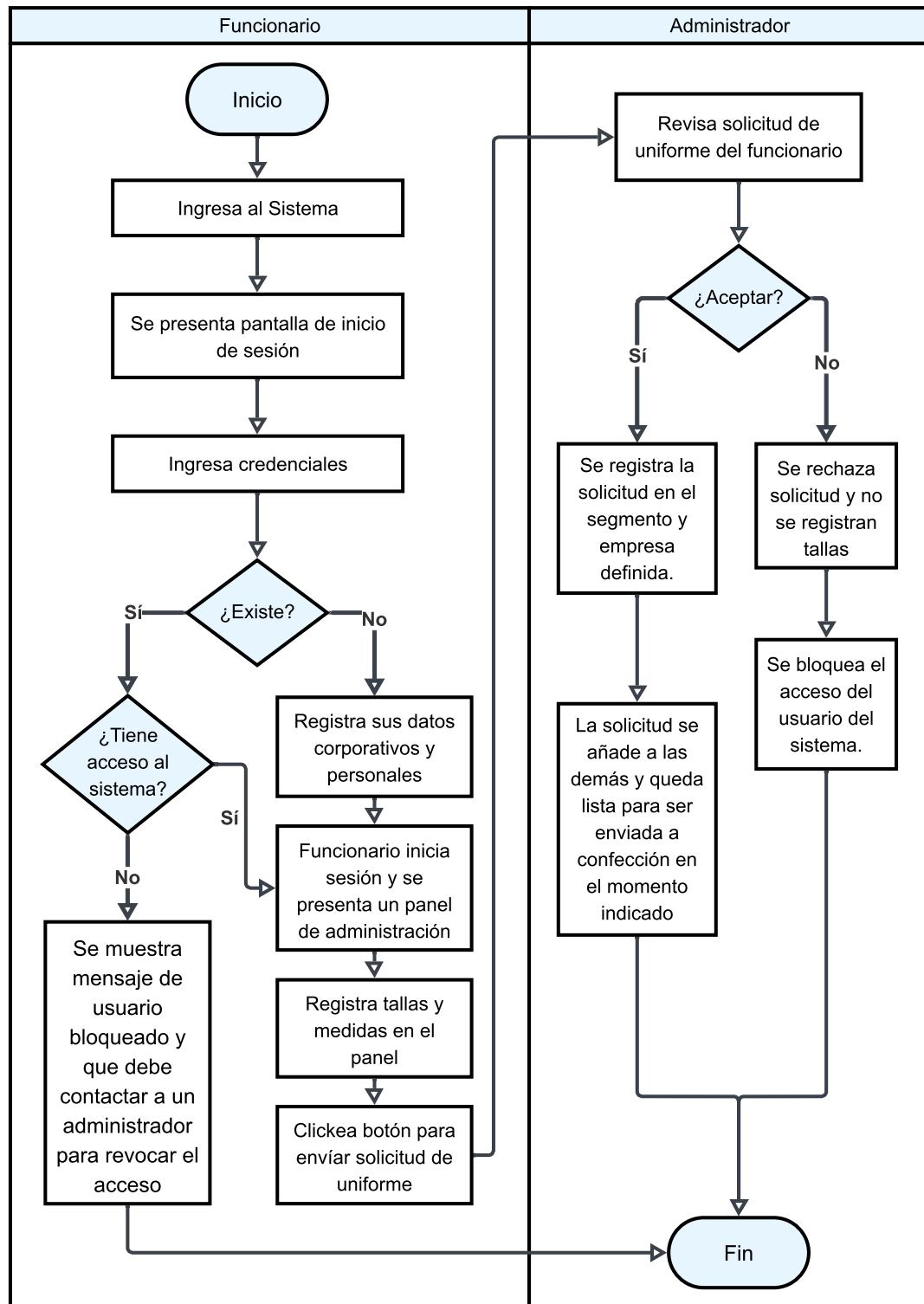


## 3.4 DIAGRAMAS DE FLUJO

A continuación se presenta una explicación detallada de los flujos principales que aborda SISTAL, junto con el diagrama de flujo respectivo para cada uno.

### 3.4.1 Proceso de Registro de Tallas

**Figura 3.13:** En primera instancia el funcionario inicia sesión o crea su cuenta, en donde completa sus datos corporativos y personales, e ingresa sus tallas y medidas. Luego de ingresar esa información, realiza la solicitud de uniforme. El administrador revisa la solicitud; si la acepta, la registra en el segmento y empresa definidos para dar continuidad al proceso; si la rechaza, el caso se cierra sin registrar tallas.



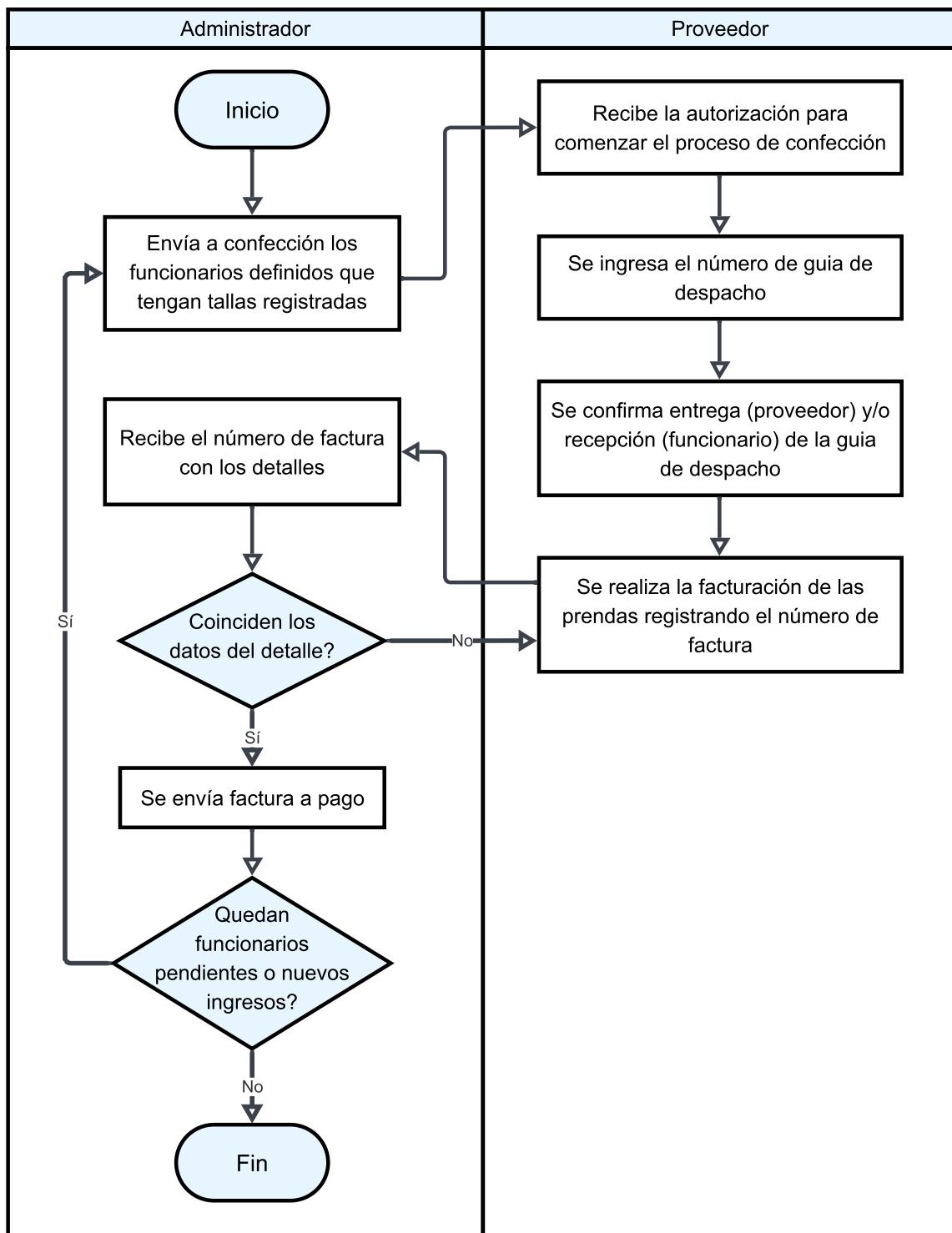
**Figura 3.13:** Diagrama de flujo del sistema actual sobre el proceso de registro de tallas

Fuente: Diseño Propio



### 3.4.2 Proceso de Envío a Confección

**Figura 3.14:** El administrador autoriza el envío a confección de los funcionarios con tallas registradas. El proveedor recibe la autorización, ingresa el número de guía de despacho y se confirma la entrega y/o recepción. Luego realiza la facturación registrando el número de factura. El administrador valida los detalles: si coinciden, la factura se envía a pago; si no, se solicita corrección. Finalmente, se verifica si quedan funcionarios pendientes o nuevos ingresos para repetir el ciclo.



**Figura 3.14:** Diagrama de flujo del sistema actual sobre el proceso de envío a confección

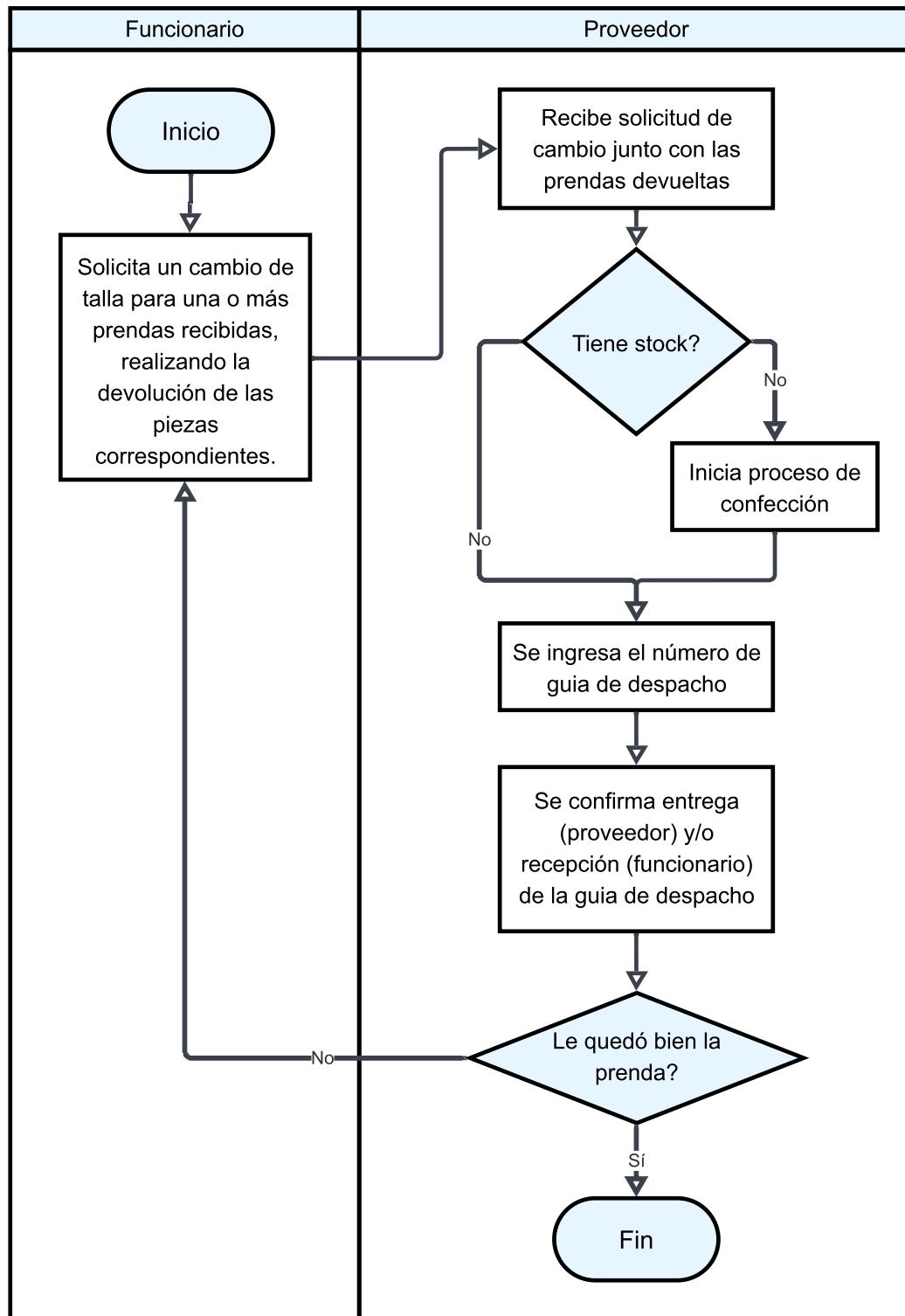
Fuente: Diseño Propio



### 3.4.3 Proceso de Postventa: Arreglo

**Figura 3.15:** El funcionario solicita el cambio de talla de una o más prendas recibidas. El proveedor evalúa disponibilidad: si hay stock, gestiona el despacho; de lo contrario, inicia confección. Se registra la guía de despacho y se confirma la entrega y/o recepción. El funcionario prueba la prenda; si el ajuste es satisfactorio, el caso se cierra; en caso contrario, se reinicia el ciclo de cambio hasta su resolución.

En este caso, el administrador no tiene un proceso propio, pero puede consultar los detalles de todo el flujo desde su módulo.



**Figura 3.15:** Diagrama de flujo del sistema actual sobre el proceso de postventa

Fuente: Diseño Propio



## CAPÍTULO 4

---

### MARCO TEÓRICO

#### 4.1 SISTEMAS DE INFORMACIÓN Y SU ROL EN LAS ORGANIZACIONES

La **Tecnología de Información (TI)** consiste en todo el *hardware* y *software* que necesita usar una empresa para poder cumplir con sus objetivos de negocios, siendo hardware todo equipo de cómputo físico tangible, como un dispositivo de almacenamiento o dispositivo móvil y software como un sistema operativo Windows o Linux, como los miles de programas o aplicaciones móviles. Mientras que un **Sistemas de Información (SI)**, puede plantearse como un conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar los procesos de toma de decisiones y de control en una organización (Laudon & Laudon, 2012).

De acuerdo a Gurbaxani y Whang (1991), los sistemas de información no solo automatizan tareas operativas, sino que transforman las estructuras y los mecanismos de coordinación dentro y fuera de las empresas. Es decir, dejan de ser simples herramientas de apoyo administrativo para convertirse en factores estratégicos que modifican el funcionamiento organizacional y las dinámicas del mercado. De esta manera, su uso estratégico puede modificar el tamaño, la forma y la distribución del poder dentro de las organizaciones, al mismo tiempo que transforma las relaciones entre las empresas y los mercados, teniendo un gran impacto en las mismas.

Los sistemas de información han transformado de manera profunda la es-



tructura y dinámica de las organizaciones modernas, convirtiéndose en un factor clave tanto en la eficiencia operativa como en la toma de decisiones estratégicas, sin nombrar las grandes inversiones que se realizan para adoptarlas. Su impacto económico se evidencia en la reducción de los costos de transacción (participación en el mercado) y agencia (administración de empleados), lo que permite a las empresas externalizar funciones, optimizar recursos y mantener estructuras más livianas sin disminuir su productividad. Desde una perspectiva organizacional, la tecnología de la información ha favorecido el aplanamiento jerárquico (disminución de gerencia y empleados) y la descentralización de la toma de decisiones, al proporcionar información oportuna y accesible a distintos niveles de la empresa. Esto ha impulsado la creación de organizaciones más flexibles y basadas en el conocimiento, donde el valor se centra en la gestión eficaz de la información. Sin embargo, la adopción de estos sistemas también plantea desafíos relacionados con la resistencia al cambio, la adaptación cultural y la reconfiguración de procesos, factores que deben ser cuidadosamente considerados para garantizar una implementación exitosa y sostenible. (Laudon & Laudon, 2012)

En este contexto, investigaciones más recientes en el campo de los Sistemas de Información amplían la comprensión del impacto organizacional de estas tecnologías, especialmente a partir de la incorporación de capacidades de inteligencia artificial. Benbya et al. (2021) sostienen que los sistemas de información basados en IA además de apoyar la toma de decisiones humanas, introducen nuevas formas de automatización, aprendizaje y autonomía que reconfiguran profundamente el trabajo, los mecanismos de control y las estructuras organizacionales. A diferencia de los SI tradicionales, estos sistemas pueden ejecutar acciones, generar recomendaciones complejas y adaptar su comportamiento en función de los datos, desplazando parcialmente la agencia desde los actores humanos hacia componentes algorítmicos. Aunque esto puede parecer un beneficio,



los autores recalcan en como este fenómeno genera tensiones organizacionales relevantes, como la automatización frente al aumento de capacidades humanas, la sustitución de tareas frente a la transformación de roles y la redistribución del poder en toma de decisiones dentro de la organización, donde el impacto de los SI deja de ser exclusivamente instrumental y se manifiesta como un fenómeno sociotécnico complejo, en el que factores tecnológicos, organizacionales y humanos interactúan de manera dinámica, planteando nuevos desafíos en términos de gobernanza, responsabilidad y sostenibilidad organizacional.

## 4.2 INGENIERÍA DE SOFTWARE

La ingeniería de software es una disciplina de ingeniería que se preocupa de todos los aspectos de la producción de software, desde la concepción inicial hasta la operación y mantenimiento (Sommerville, 2016, p. 21).

También, otra forma de definir software son: (1) instrucciones que se ejecutan para lograr funciones deseadas; (2) estructura de datos que permiten a los programas manipular información adecuadamente; y (3) información descriptiva tanto de forma impresa como virtual que describe la operación y uso de los programas (Pressman, 2020, p. 5). Esto implica que el software no solo son los programas computacionales, un buen software profesional incluye la documentación asociada, librerías, soporte y datos de configuración necesarios para que el programa sea de utilidad (Sommerville, 2016).

La importancia de esta disciplina radica en la necesidad de desarrollar sistemas confiables, eficientes y seguros que respondan a las demandas de una sociedad cada vez más dependiente de la tecnología. Por un lado, la disciplina de ingeniería se encarga de aplicar métodos, teorías y herramientas de forma con-



trolada y adaptativa, permitiendo obtener productos de alta calidad dentro de los límites de tiempo y presupuesto establecidos. Además, contribuye a reducir los costos a largo plazo asociados con la corrección, validación y mantenimiento de los sistemas, garantizando su capacidad de adaptación ante los constantes cambios tecnológicos y organizacionales. En este sentido, aunque desarrollar software pueda parecer una tarea sencilla, cuando no se aplican adecuadamente los principios, las consecuencias suelen ser significativas. La falta de una metodología estructurada puede derivar en proyectos con sobrecostos, retrasos en los plazos de entrega y productos de baja calidad que no satisfacen las necesidades del usuario (Somerville, 2016).

#### **4.2.1 Metodologías de Desarrollo**

Las metodologías de desarrollo de software surgen como la aplicación práctica de los procesos definidos en la ingeniería de software, estableciendo un marco estructurado que guía las actividades necesarias para crear un producto de calidad. Cada metodología organiza de manera distinta las etapas fundamentales del proceso: comunicación, planificación, modelado, construcción y despliegue, definiendo cómo se relacionan entre sí y cómo fluyen a lo largo del tiempo. Así, dependiendo del enfoque, estas etapas pueden ejecutarse de forma secuencial, iterativa, evolutiva o paralela, adaptándose a las características y necesidades del proyecto. (Pressman, 2020)

La metodología en cascada, también conocida como modelo secuencial lineal, representa uno de los enfoques más tradicionales dentro de la ingeniería de software. Se basa en un proceso ordenado y progresivo que avanza de manera lineal desde la comunicación y especificación de requisitos, pasando por la planificación, el modelado y la construcción, hasta llegar a la fase de despliegue y so-



porte. Este modelo resulta adecuado cuando los requerimientos del sistema están claramente definidos y son estables, como en casos de mejoras o adaptaciones a sistemas existentes. Sin embargo, su rigidez y la falta de retroalimentación temprana suelen dificultar su aplicación en proyectos complejos o cambiantes, donde la aparición tardía de errores o la redefinición de necesidades puede generar retrasos y mayores costos (Pressman, 2020).

Aquí es donde entra la agilidad (también referidas como metodologías ligeras o *lean*). En la ingeniería de software se refiere a la capacidad de los equipos de desarrollo para adaptarse de forma rápida y eficiente a los cambios constantes que surgen durante un proyecto. Más que una simple respuesta al cambio, la agilidad promueve una filosofía de trabajo flexible basada en la colaboración, la comunicación efectiva y la entrega continua de software funcional, poniendo un fuerte énfasis en el modelo incremental. Este enfoque integra al cliente dentro del proceso de desarrollo, fomenta la reducción de tareas innecesarias y favorece la planificación adaptable ante entornos inciertos. (Pressman, 2020).

*Scrum* es una metodología ágil basada en principios del manifiesto ágil y orientada a gestionar proyectos de forma flexible y colaborativa. Su estructura se organiza en ciclos cortos de trabajo llamados *sprints*, de entre dos y cuatro semanas, durante los cuales un equipo multidisciplinario desarrolla incrementos funcionales del producto. El proceso se apoya en artefactos clave como el *product backlog* (lista priorizada de requerimientos), el *sprint backlog* (tareas seleccionadas para cada *sprint*) y los incrementos de código entregables. A lo largo del desarrollo, se realizan reuniones periódicas (como las *daily scrums*, revisiones y retrospectivas) que facilitan la comunicación, la detección temprana de problemas y la mejora continua. Gracias a esta dinámica iterativa e incremental, Scrum permite adaptarse rápidamente a los cambios y mantener un enfoque constante en la entrega de valor al cliente. (Pressman, 2020).



#### 4.2.2 DevOps

DevOps (*Development & Operations*) es un enfoque que integra el desarrollo (*Development*) y las operaciones (*Operations*) con el objetivo de unificar ambos procesos bajo los principios ágiles y de desarrollo *lean*. Este modelo busca optimizar todo el ciclo de vida del software mediante una colaboración continua entre los equipos técnicos y operativos. Su flujo de trabajo se basa en etapas que se repiten de manera cíclica: desarrollo, prueba, integración, despliegue y monitoreo continuos. A través de la **automatización y la retroalimentación constante**, DevOps permite detectar errores tempranamente, reducir tiempos de entrega y mejorar la calidad del producto, acelerando la disponibilidad del software en entornos de producción y favoreciendo la innovación, la satisfacción del cliente y la capacidad de las organizaciones para adaptarse rápidamente a las necesidades del mercado (Pressman, 2020).

### 4.3 REINGENIERÍA DE SOFTWARE

De acuerdo a Ruhl y Gunn (1991), existen diversas formas de entender el término reingeniería de software. Entre las interpretaciones más citadas se encuentran

- *Forward engineering*, definido como el proceso tradicional de pasar de abstracciones de alto nivel y diseños independientes de la implementación a la construcción física de un sistema.
- *Reverse engineering*, que consiste en analizar un sistema existente para identificar sus componentes y relaciones, y generar representaciones del sistema en otro nivel de abstracción, sin modificarlo. Dentro de esta categoría se incluye



yen subáreas como redocumentación (creación o revisión de representaciones equivalentes) y recuperación de diseño (identificación de abstracciones de alto nivel mediante información externa y deducción).

- *Restructuring*, la transformación de un sistema de una forma de representación a otra al mismo nivel de abstracción, manteniendo su comportamiento externo.

Otros autores, como Rashid et al. (2013) definen reingeniería de software, como el examen, reorganización, análisis y modificación de un sistema de software existente. Donde su objetivo principal es mejorar la mantenibilidad del sistema y reconstituirlo en una nueva forma, implementando posteriormente el sistema modificado sin afectar su funcionalidad. Este proceso puede involucrar la reestructuración o recodificación de uno o todos los componentes del sistema legado y combina diversos procesos, tales como ingeniería inversa, redocumentación, traducción y *forward engineering*, aunque Sommerville (2016, p. 276) sugiere que idealmente, no deberían hacerse cambios muy grandes a la arquitectura de software con el objetivo de reducir riesgos y costos.

En una revisión más reciente realizada por Lim y Sian (2025), se amplía esta conceptualización al analizar de manera sistemática los enfoques contemporáneos de reingeniería de software y proponer una visión estructurada del proceso basada en etapas. Según esta revisión, la reingeniería de software suele organizarse en una secuencia de fases interrelacionadas: (1) análisis del sistema legado, orientado a comprender su arquitectura, dependencias y problemas técnicos; (2) ingeniería inversa, cuyo propósito es recuperar conocimiento de alto nivel sobre el diseño y la lógica del sistema; (3) restructuración y rediseño, donde se redefinen componentes, arquitecturas o tecnologías manteniendo el comportamiento funcional; (4) reimplementación o modernización, que implica la adaptación a nuevas plataformas, lenguajes o paradigmas; y finalmente (5) validación e implementación, asegurando que el sistema reingenierizado cumpla con los requisitos



funcionales y no funcionales originales.

Esto resulta en un proceso complejo que aborda las complejidades de un sistema existente, imponiendo una severidad de desafíos al equipo, como en el caso de que exista falta de documentación del sistema actual o conocimiento de los procesos intrínsecos. Asimismo, existen posibles riesgos que deben considerarse y evaluarse en el proceso, como costos, tiempos extendidos o problemas de rendimiento al utilizar una aproximación inapropiada Lim y Sian (2025).

## 4.4 ARQUITECTURA DE SOFTWARE

En el ámbito de la ingeniería de software, el concepto de arquitectura de software ha sido definido de diversas maneras, siendo comúnmente considerada como el “plano” o “hoja de ruta” de un sistema. Como señala Richards y Ford (2020), ha sido difícil llegar a una definición sólida, sobre todo considerando el constante cambio que sufre el rol de arquitecto, y la aparición de nuevas metodologías y diseños como lo es DevOps. Sin embargo, la definición entregada por los autores la entiende como la combinación de varios elementos fundamentales que, en conjunto, determinan la estructura y el comportamiento del sistema<sup>1</sup>.

Estos elementos son:

- **La estructura del sistema**, que representa el estilo o patrón arquitectónico adoptado (como microservicios, arquitectura en capas o microkernel). Esta describe cómo se organizan los componentes y cómo interactúan entre sí.
- **Las características arquitectónicas**, también conocidas como los “-ilities”, que definen los criterios de éxito del sistema más allá de su funcionalidad, tales

---

<sup>1</sup>Considerando la fecha del texto publicado por los autores Richards y Ford (2020), posiblemente se han puesto de acuerdo en una nueva definición.



como escalabilidad, rendimiento, o seguridad (*scalability, performance, security*).

- **Las decisiones arquitectónicas**, que establecen las reglas y restricciones que guían la construcción del sistema, determinando qué está permitido y qué no dentro del diseño técnico.
- **Los principios de diseño**, que funcionan como directrices o recomendaciones que orientan a los equipos de desarrollo sin imponer reglas estrictas, promoviendo buenas prácticas y consistencia en las implementaciones.

De acuerdo a Richards y Ford (2020), los estilos de arquitectura pueden ser clasificados en dos tipos principales, monolíticos (*monolithic*, una sola unidad de código desplegable) y distribuidos (*distributed*, múltiples unidades de código conectadas por protocolos de acceso remoto). También se menciona la falta total de una estructura como una “gran bola de barro (*big ball of mud*)”, un antipatrón de arquitectura que se debe evitar a toda costa y que ocurre frecuentemente.

#### 4.4.1 Arquitectura de Microservicios

La arquitectura de microservicios se basa en el desarrollo de pequeñas unidades de software autónomas, llamadas servicios, que trabajan en conjunto para conformar un sistema más amplio. Cada microservicio está diseñado para cumplir una única función o responsabilidad dentro del dominio del negocio, siguiendo el principio de responsabilidad única. Esta división permite que el código se mantenga más cohesionado, legible y fácil de modificar, evitando los problemas comunes de los sistemas monolíticos donde las dependencias internas dificultan el mantenimiento y la evolución del software. En lugar de un solo bloque de código, los microservicios establecen límites claros entre funcionalidades, haciendo más evidente la relación entre cada módulo y su propósito dentro del sistema. (Newman,



2015)

Una de las principales características de los microservicios es su autonomía. Cada servicio puede desarrollarse, desplegarse y mantenerse de manera independiente, comunicándose con los demás a través de interfaces bien definidas (APIs). Esto evita el acoplamiento entre componentes y permite realizar cambios o actualizaciones sin afectar el funcionamiento global del sistema. Esta independencia también ofrece la posibilidad de utilizar diferentes tecnologías o lenguajes en cada servicio, seleccionando las herramientas más adecuadas para cada caso, lo que se conoce como heterogeneidad tecnológica. (Newman, 2015)

Entre los beneficios más destacados de esta arquitectura se encuentran la escalabilidad, la resiliencia y la facilidad de despliegue. Los microservicios pueden escalarse individualmente, optimizando el uso de recursos y reduciendo costos operativos. Asimismo, en caso de fallos, estos quedan aislados dentro del servicio afectado, evitando que se propague al resto del sistema. Por último, al permitir despliegues pequeños y frecuentes, se reducen los riesgos asociados a las actualizaciones y se acelera la entrega de nuevas funcionalidades al usuario (Newman, 2015).

## 4.5 CONTENEDORES Y ORQUESTACIÓN

### 4.5.1 Contenedores

Los contenedores son una tecnología que permite empaquetar y ejecutar aplicaciones de forma aislada, junto con todas las dependencias necesarias para su funcionamiento, como bibliotecas, configuraciones y archivos del sistema. A diferencia de las máquinas virtuales, los contenedores comparten el mismo nú-



cleo del sistema operativo, lo que los hace más ligeros, portátiles y eficientes. Su propósito principal es garantizar que una aplicación se ejecute de la misma manera sin importar el entorno, eliminando los conflictos derivados de las diferencias entre sistemas. Un contenedor se crea a partir de una imagen de contenedor, la cual es un paquete binario que contiene todo lo necesario para ejecutar un programa dentro de un entorno aislado. Estas imágenes suelen construirse en capas, donde cada una agrega o modifica elementos sobre la anterior, permitiendo un almacenamiento y despliegue más eficiente. Existen dos tipos principales de contenedores: los contenedores de sistema, que imitan el comportamiento de una máquina virtual completa, y los contenedores de aplicación, que ejecutan un solo proceso o servicio, siguiendo la filosofía de modularidad y escalabilidad propia de los entornos modernos basados en microservicios.

En este contexto, Docker se ha consolidado como la plataforma más popular para la creación, distribución y ejecución de contenedores. Introducido como un proyecto de código abierto, Docker simplificó la adopción de esta tecnología mediante el uso de un formato estándar de imagen conocido como el formato de imagen Docker, compuesto por capas del sistema de archivos. Este formato permite construir imágenes reutilizables y distribuirlas a través de registros de contenedores (container registries), facilitando la automatización del despliegue en diferentes entornos. Gracias a Docker, los desarrolladores pueden construir aplicaciones portátiles, reproducibles y fácilmente escalables, convirtiendo esta herramienta en un componente esencial de la ingeniería de software moderna y un pilar para arquitecturas basadas en contenedores y microservicios (Burns et al., 2019).



#### 4.5.2 Kubernetes

Kubernetes es una plataforma de código abierto diseñada para la orquestación y gestión automatizada de aplicaciones en contenedores, permitiendo su implementación, escalado y mantenimiento de forma eficiente. Desarrollado inicialmente por Google, este sistema se basa en la experiencia adquirida en la administración de infraestructuras distribuidas a gran escala y se ha consolidado como el estándar para la construcción de aplicaciones cloud-native. Su función principal consiste en coordinar múltiples contenedores que conforman un sistema distribuido, garantizando la disponibilidad, escalabilidad y resiliencia de los servicios, incluso ante fallos en componentes individuales. Gracias a su capacidad para abstraer la infraestructura subyacente, Kubernetes puede ejecutarse en entornos locales, nubes públicas o híbridas, facilitando la portabilidad y la eficiencia en la administración de recursos.

Una de las principales fortalezas de Kubernetes radica en su arquitectura declarativa y su enfoque en la infraestructura inmutable. A través de estos principios, los desarrolladores describen el estado deseado del sistema, y Kubernetes se encarga de mantenerlo automáticamente mediante procesos de auto-recuperación (self-healing). Esto permite desplegar actualizaciones, escalar servicios y restaurar fallos sin intervención manual, lo que incrementa la velocidad de desarrollo (velocity) y reduce los errores operativos. En conjunto, estas características convierten a Kubernetes en una herramienta fundamental para la ingeniería de software moderna, al posibilitar la creación de sistemas distribuidos confiables y adaptables a las demandas cambiantes del entorno digital (Burns et al., 2019).



## CAPÍTULO 5

### LEVANTAMIENTO DE REQUISITOS MEDIANTE HISTORIAS DE USUARIO

Como se revisó con anterioridad, SISTAL contempla tres módulos principales: Funcionario, Proveedor y Administrador. Estos módulos cubren el ciclo completo de tallaje, solicitud, confección y despacho de uniformes, incluyendo también procesos de post-venta como cambios o composturas.

No obstante, en el marco de este proyecto el desarrollo se enfocará en la implementación del módulo de **Funcionario**. Los módulos de **Administrador** y **Proveedor** forman parte de la visión global de SISTAL y se incluyen en esta para describir el sistema objetivo; sin embargo, no se consideran dentro del alcance de la versión desarrollada en este trabajo de título.

La toma de requerimientos que se presenta a continuación se acota, por tanto, al funcionamiento del sistema desde la perspectiva del Funcionario y a los aspectos transversales necesarios para su operación, incorporando además los requerimientos de Proveedor y Administrador como base para futuras etapas de evolución del sistema.

#### 5.1 REQUERIMIENTOS

En esta sección se describen los requerimientos funcionales y no funcionales del sistema. A nivel de alcance de este proyecto, la relación entre módulos y desarrollo es la siguiente:



- El módulo **Funcionario** se considera **en alcance de implementación** para la versión desarrollada en este trabajo.
- Los módulos **Administrador** y **Proveedor** se consideran **fuerza del alcance de implementación** en esta versión, pero sus requerimientos se documentan para representar el comportamiento completo del sistema objetivo.

Se usará como notación para cada requerimiento XX-YNN, donde **XX** representa el Requerimiento Funcional o No Funcional (RF, RNF), **Y** como el Actor (F, P, A) y **NN** como número incremental que lista cada uno.

### 5.1.1 Requerimientos funcionales

A continuación se detallan los requerimientos funcionales derivados de los casos de uso identificados para cada módulo.

#### Módulo Funcionario

- RF-F01: Consultar tallas y medidas ingresadas.
- RF-F02: Editar tallas ingresadas previo a corte.
- RF-F03: Ingresar tallas y medidas.
- RF-F04: Confirmar recepción de entrega.
- RF-F05: Solicitar cambio de prenda.
- RF-F06: Solicitar uniforme.
- RF-F07: Solicitar compostura.
- RF-F08: Contactar a soporte.
- RF-F09: Consultar el estado de solicitudes de uniforme.
- RF-F10: Consultar estado de despacho.
- RF-F11: Registrar cambio de sucursal.



- RF-F12: Cambiar contraseña.
- RF-F13: Registrarse en el sistema.
- RF-F14: Completar encuesta de satisfacción.
- RF-F15: Consultar manuales de uso del sistema.

## Visión Global Módulo Proveedor

Estos requerimientos describen las capacidades esperadas del módulo Proveedor en el sistema objetivo. Su implementación concreta se considera para etapas futuras de evolución del sistema.

- RF-P01: Exportar cortes.
- RF-P02: Consultar cortes.
- RF-P03: Ingresar funcionario.
- RF-P04: Consultar funcionarios.
- RF-P05: Registrar recepción de devolución.
- RF-P06: Consultar guías de despacho.
- RF-P07: Consultar despacho.
- RF-P08: Adjuntar evidencia de guía de despacho.
- RF-P09: Registrar guía de despacho (número de guía, empresa, voucher).
- RF-P10: Registrar factura en base a guías recibidas.
- RF-P11: Consultar uniformes por estado.
- RF-P12: Confirmar devoluciones de uniforme.
- RF-P13: Generar y exportar reportes.
- RF-P14: Registrar compostura.
- RF-P15: Registrar recepción en planta.
- RF-P16: Registrar comprobante de factura.
- RF-P17: Cambiar contraseña.



- RF-P18: Exportar nóminas de despacho.

## Visión Global Módulo Administrador

El módulo Administrador actúa como conector de procesos entre el Funcionario y el Proveedor, permitiendo gestionar solicitudes, temporadas, reglas de talla, catálogo y estados. Los requerimientos que se presentan a continuación describen el comportamiento esperado del sistema objetivo y se consideran para etapas posteriores de desarrollo.

- RF-A01: Realizar cambio de sucursal de funcionario.
- RF-A02: Gestionar alta y baja de funcionario.
- RF-A03: Resolver cambios de sucursal de funcionarios.
- RF-A04: Gestionar nuevas solicitudes.
- RF-A05: Consultar estado de solicitudes.
- RF-A06: Realizar apertura y cierre de temporada.
- RF-A07: Programar lote de confección.
- RF-A08: Consultar uniformes en confección.
- RF-A09: Gestionar curvas y reglas de talla.
- RF-A10: Consultar uniformes por estado.
- RF-A11: Monitorear devoluciones.
- RF-A12: Generar y exportar reportes.
- RF-A13: Gestionar roles y permisos.
- RF-A14: Parametrizar catálogo de prendas.
- RF-A15: Iniciar sesión.
- RF-A16: Consultar datos de temporadas anteriores.
- RF-A17: Consultar datos y estado de los funcionarios.
- RF-A18: Gestionar factura del proveedor.



- RF-A19: Consultar encuestas de satisfacción de los funcionarios.

## 5.1.2 Requerimientos no funcionales

Los siguientes requerimientos no funcionales aplican al sistema en su conjunto, con foco en la experiencia del Funcionario en esta versión y considerando la evolución futura hacia los módulos de Proveedor y Administrador.

### Usabilidad

- RNF-U01: El sistema debe ser intuitivo y accesible desde dispositivos móviles y navegadores modernos.
- RNF-U02: Las interfaces deben incluir validaciones y mensajes claros para guiar al usuario durante la interacción.
- RNF-U03: Debe existir documentación accesible para cada tipo de actor (Funcionario, Proveedor, Administrador).

### Seguridad

- RNF-S01: El sistema debe implementar autenticación y autorización basada en roles.
- RNF-S02: Toda la información sensible debe transmitirse mediante HTTPS con cifrado TLS.
- RNF-S03: La actividad relevante debe registrarse en logs auditables.



## Rendimiento

- RNF-R01: Las consultas y listados deben responder en menos de 2 segundos bajo carga normal.
- RNF-R02: La exportación de reportes y archivos debe ejecutarse sin bloquear la interfaz de usuario.

## Disponibilidad

- RNF-D01: El sistema debe estar disponible al menos en un 99 % durante el período de temporada alta definido para su operación.
- RNF-D02: El sistema debe soportar múltiples proveedores y sucursales en simultáneo, siguiendo un modelo multi-tenant a nivel de diseño, aunque su explotación completa se considere para etapas posteriores.

## Integración

- RNF-I01: La arquitectura debe considerar la integración mediante API Gateway para la exposición controlada de servicios.

## 5.2 HISTORIAS DE USUARIO

Las historias de usuario permiten describir las necesidades del sistema desde la perspectiva de los actores, facilitando la priorización, planificación y validación de los desarrollos. En esta versión, la implementación se centra en el módulo Funcionario; sin embargo, se incluyen también historias asociadas a Pro-



veedor y Administrador para reflejar la visión global del sistema y servir de base a futuras iteraciones.

En este caso, se usará como notación para cada Historia de Usuario HU-XXNN, con **HU** como la Historia de Usuario, **XX** como el tipo, y **NN** como número incremental que lista cada uno.

### **Historias de usuario transversales**

- HU-AT01: Como usuario del sistema, quiero iniciar sesión con mis credenciales para acceder a las funcionalidades que me corresponden.
- HU-AT02: Como usuario del sistema, quiero poder recuperar o restablecer mi contraseña en caso de olvido para no perder el acceso.
- HU-AT03: Como usuario del sistema, quiero registrar un correo de recuperación asociado a mi cuenta para poder restablecer el acceso de forma segura en caso de problemas con mi contraseña.

En la versión desarrollada en este trabajo, estas historias transversales se implementan principalmente para el rol Funcionario.

### **Funcionario (en alcance de implementación)**

- HU-F01: Como funcionario, quiero registrarme en la plataforma para poder acceder al sistema de gestión de uniformes.
- HU-F02: Como funcionario, quiero iniciar sesión con mi RUT y contraseña para acceder a mis solicitudes, tallas y estados.
- HU-F03: Como funcionario, quiero cambiar mi contraseña desde el sistema para mantener la seguridad de mi cuenta.



- HU-F04: Como funcionario, quiero ingresar mis tallas y medidas para recibir prendas adecuadas a mi contextura.
- HU-F05: Como funcionario, quiero editar mis tallas antes del corte para corregir errores o cambios en mis medidas.
- HU-F06: Como funcionario, quiero consultar el estado de mis solicitudes de uniforme para saber en qué etapa se encuentran.
- HU-F07: Como funcionario, quiero solicitar un cambio de prenda cuando la talla recibida no es la correcta.
- HU-F08: Como funcionario, quiero confirmar la recepción del despacho para cerrar el proceso de entrega.
- HU-F09: Como funcionario, quiero contactar a soporte para resolver dudas o problemas con el sistema o con mis solicitudes.
- HU-F10: Como funcionario, quiero completar una encuesta de satisfacción para evaluar el proceso de solicitud y entrega de uniformes.
- HU-F11: Como funcionario, quiero consultar tutoriales o manuales de uso del sistema para entender cómo utilizar correctamente sus funcionalidades.

### **Proveedor (visión global, fuera de alcance de implementación)**

- HU-P01: Como proveedor, quiero iniciar sesión para acceder a la gestión de cortes, despachos y devoluciones.
- HU-P02: Como proveedor, quiero consultar los cortes generados para organizar la confección de uniformes.
- HU-P03: Como proveedor, quiero registrar guías de despacho para mantener la trazabilidad de las entregas al funcionario.
- HU-P04: Como proveedor, quiero consultar el estado de los uniformes por funcionario para coordinar producciones y despachos.



### Administrador (visión global, fuera de alcance de implementación)

- HU-A01: Como administrador, quiero iniciar sesión para acceder a la gestión de funcionarios, solicitudes y configuraciones del sistema.
- HU-A02: Como administrador, quiero gestionar altas y bajas de usuarios para mantener actualizado el universo de usuarios del sistema.
- HU-A03: Como administrador, quiero consultar el estado de las solicitudes de uniforme para monitorear la operación entre proveedor y funcionario.
- HU-A04: Como administrador, quiero parametrizar el catálogo de prendas y reglas de talla para asegurar que las solicitudes del funcionario se ajusten a la política de uniformes.

Las historias de usuario aquí presentadas se complementan con los requerimientos funcionales descritos previamente. En conjunto, estas secciones permiten diferenciar claramente la **visión completa del sistema del alcance efectivo de la versión desarrollada**, centrada en el módulo Funcionario e incorporando Proveedor y Administrador como referencia para futuras extensiones de SISTAL.



## CAPÍTULO 6

---

### DISEÑO Y ARQUITECTURA ACTUALIZADOS

El presente capítulo describe de manera detallada el diseño del sistema propuesto, estableciendo los fundamentos arquitectónicos, estructurales y operativos que permitirán su correcta implementación. El objetivo es definir, mediante una visión integral, cómo se organiza el software y qué decisiones técnicas sustentan su funcionamiento.

Para ello, se presentan los distintos modelos y diagramas que representan la estructura interna del sistema y su comportamiento esperado. Estos incluyen la arquitectura general de la solución, la modelación de la base de datos, los diagramas de flujo de procesos, los diagramas de clases y componentes, así como los modelos de despliegue e infraestructura necesarios para soportar el funcionamiento en entornos reales.

A través de estos elementos, se busca proporcionar una comprensión completa y coherente del sistema, asegurando que cada una de las partes que lo componen responda a los requerimientos funcionales y no funcionales definidos previamente. Este diseño constituye la base para la etapa de desarrollo, facilitando la toma de decisiones técnicas y garantizando que la solución final sea escalable, mantenable y alineada con los objetivos del proyecto.

#### 6.1 DISEÑO DE ARQUITECTURA

La arquitectura del sistema constituye la base estructural que define cómo se organizan, interactúan y operan los distintos componentes que conforman la

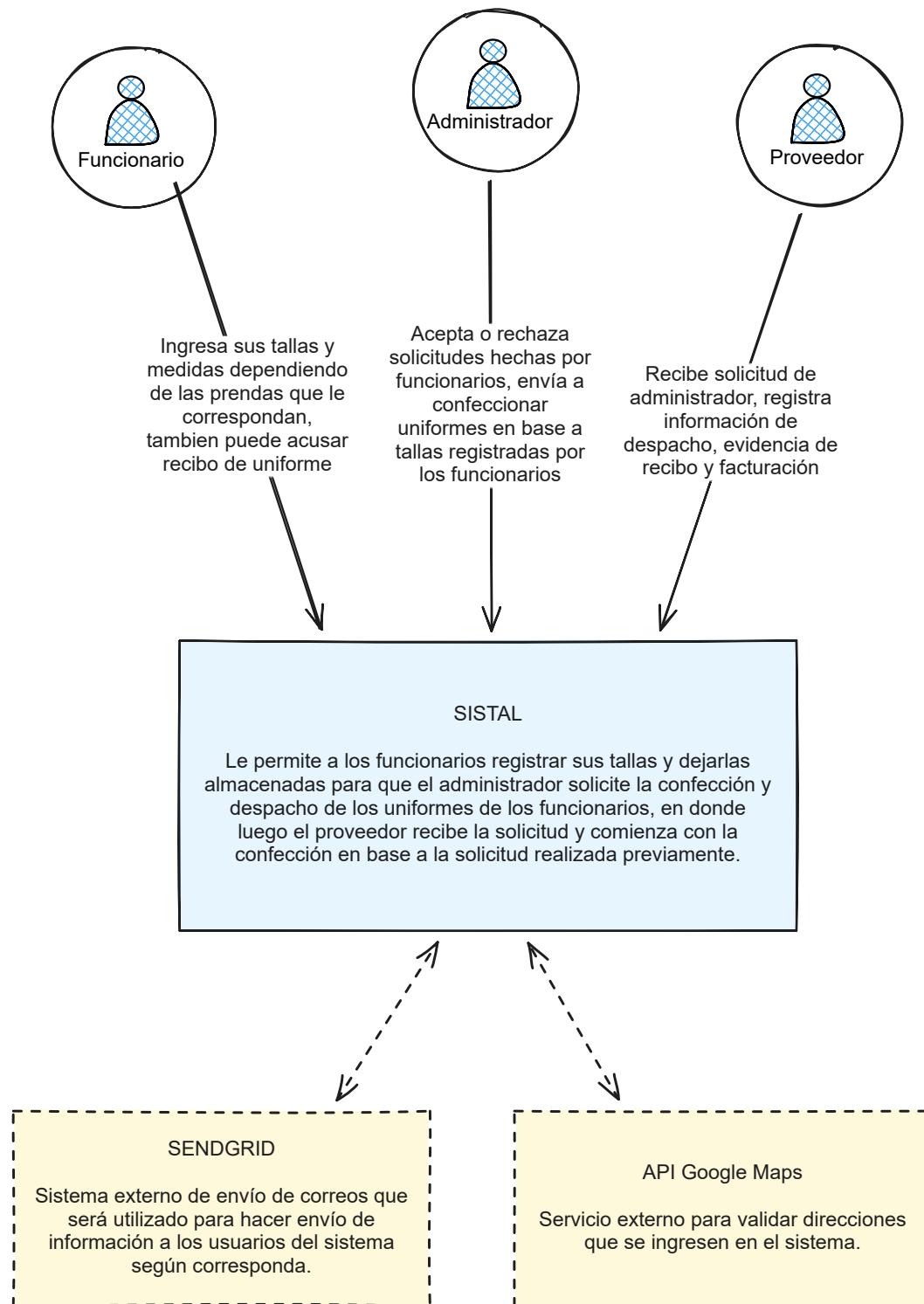


solución. En esta sección se presentan los modelos arquitectónicos que permiten comprender la visión global del sistema, su contexto operativo y la distribución de responsabilidades entre sus módulos internos.

Para ello, se utilizan distintos niveles de representación, comenzando con una vista de alto nivel que describe el entorno donde se inserta la solución, para luego profundizar en los contenedores, los microservicios y la forma en que el sistema es desplegado e implementado en la infraestructura cloud. Cada uno de estos diagramas facilita la comprensión del sistema desde múltiples perspectivas, asegurando que la arquitectura sea coherente, escalable y alineada con los requerimientos establecidos.

### 6.1.1 Diagrama de Contexto

El diagrama de contexto C1 presentado en la [Figura 6.1](#), ofrece una vista de alto nivel del sistema, mostrando cómo se relaciona con los actores externos y otros sistemas con los que interactúa. Su objetivo es situar la solución dentro de su entorno operativo, identificando los límites del sistema y los flujos de comunicación que lo conectan con usuarios, servicios externos o plataformas complementarias. Este nivel permite comprender la función global del sistema sin entrar aún en detalles técnicos internos.



**Figura 6.1:** Diagrama C1 de contexto

Fuente: Diseño Propio

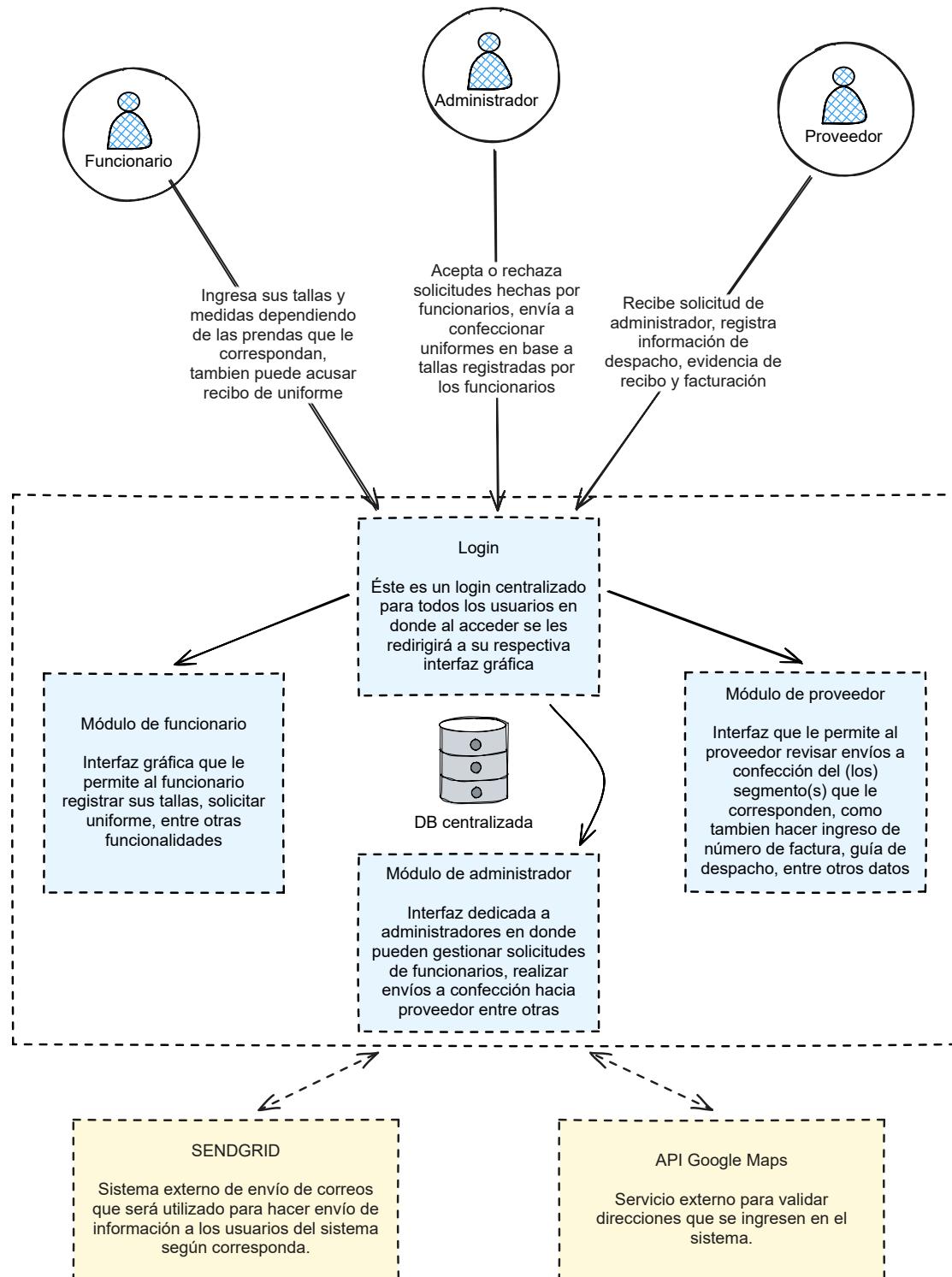


### 6.1.2 Diagrama de Contenedores

El diagrama de contenedores C2 presentado en la [Figura 6.2](#), profundiza en la estructura interna del sistema, representando los contenedores<sup>1</sup> o componentes principales que lo conforman. En este nivel se identifican los módulos, software, bases de datos, aplicaciones externas y cualquier elemento que cumple una responsabilidad específica dentro de la solución. Este diagrama permite visualizar cómo se divide el sistema, cómo se distribuyen sus funcionalidades y qué mecanismos de comunicación existen entre los distintos contenedores.

---

<sup>1</sup>En el contexto del modelo de diagramas C4, el término **contenedor** se emplea con un sentido arquitectónico y no debe confundirse con la contenedорización de aplicaciones mediante Docker u otras tecnologías similares.

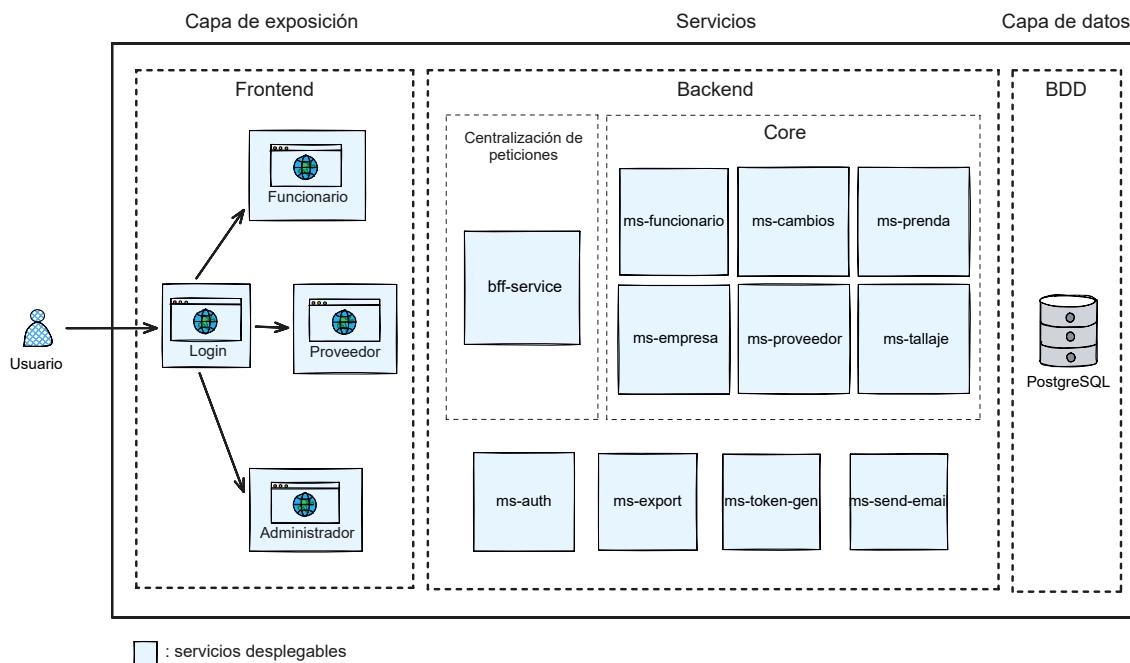


**Figura 6.2: Diagrama C2 de Contenedores**

Fuente: Diseño Propio

### 6.1.3 Diagrama de Arquitectura

El diagrama de arquitectura presentado en la [Figura 6.3](#), detalla la organización interna a nivel de microservicios, mostrando explícitamente cada uno de los servicios independientes que conforman el sistema. Este gráfico permite comprender la lógica modular de la solución, poniendo énfasis en los componentes compartidos y la forma en que los microservicios cooperan para cumplir los requerimientos funcionales. Además, evidencia los principios de escalabilidad, separación de responsabilidades y resiliencia adoptados en la arquitectura.



**Figura 6.3:** Diagrama de arquitectura

Fuente: *Diseño Propio*

Dentro de este diagrama tenemos:

- **Capa de exposición**

Consiste en los microservicios que exponen el sistema al usuario, separados



en *Login* y módulo para cada actor. A diferencia de la arquitectura anterior, aquí esta capa se encuentra separada de los otros servicios. Permitiendo implementar cada módulo por separado e incluso añadir nuevos en un futuro como la idea de una aplicación para dispositivos móviles, ya que solo tienen que conectar a la capa de servicios para mostrar la información necesaria.

- **Servicios:** Esta capa es responsable de procesar las reglas del negocio, gestionar la persistencia de datos y exponer capacidades bien definidas a través de interfaces claras. Los servicios se encuentran explicados en la [Tabla 6.1](#).
- **Capa de datos:** Para la capa de datos se consideró el mismo modelo de base de datos centralizada.

**Tabla 6.1:** Descripción microservicios de la [Figura 6.3](#)

Fuente: *Elaboración Propia*

Microservicio	Descripción
bff-service	El <i>Backend for Frontend</i> (BFF) actúa como una capa intermedia especializada, responsable de centralizar y orquestar las peticiones provenientes del frontend. Esto permite adaptar las respuestas del backend a las necesidades específicas de cada vista o perfil de usuario (funcionario, proveedor y administrador), evitando que los servicios frontend deban gestionar agregaciones de datos, transformaciones o llamadas encadenadas a múltiples microservicios. Como resultado, los servicios frontend se simplifican significativamente, enfocándose exclusivamente en la presentación y experiencia de usuario.
ms-auth	Este microservicio centraliza la autenticación y autorización del sistema. Su separación permite desacoplar la gestión de credenciales, roles y permisos del resto de los servicios de negocio, facilitando la aplicación de políticas de seguridad homogéneas y su evolución independiente.
ms-token-gen	Responsable de la generación y validación de tokens de acceso, este servicio complementa a ms-auth y permite implementar mecanismos de seguridad basados en tokens (por ejemplo, JWT). Su aislamiento mejora la seguridad y posibilita cambios en la estrategia de autenticación sin impactar al resto del sistema.
ms-funcionario	Gestiona la información y las operaciones asociadas a los funcionarios del sistema. Al encapsular este subdominio en un microservicio dedicado, se asegura una alta cohesión funcional y se facilita la evolución de las reglas específicas asociadas a este tipo de usuario.

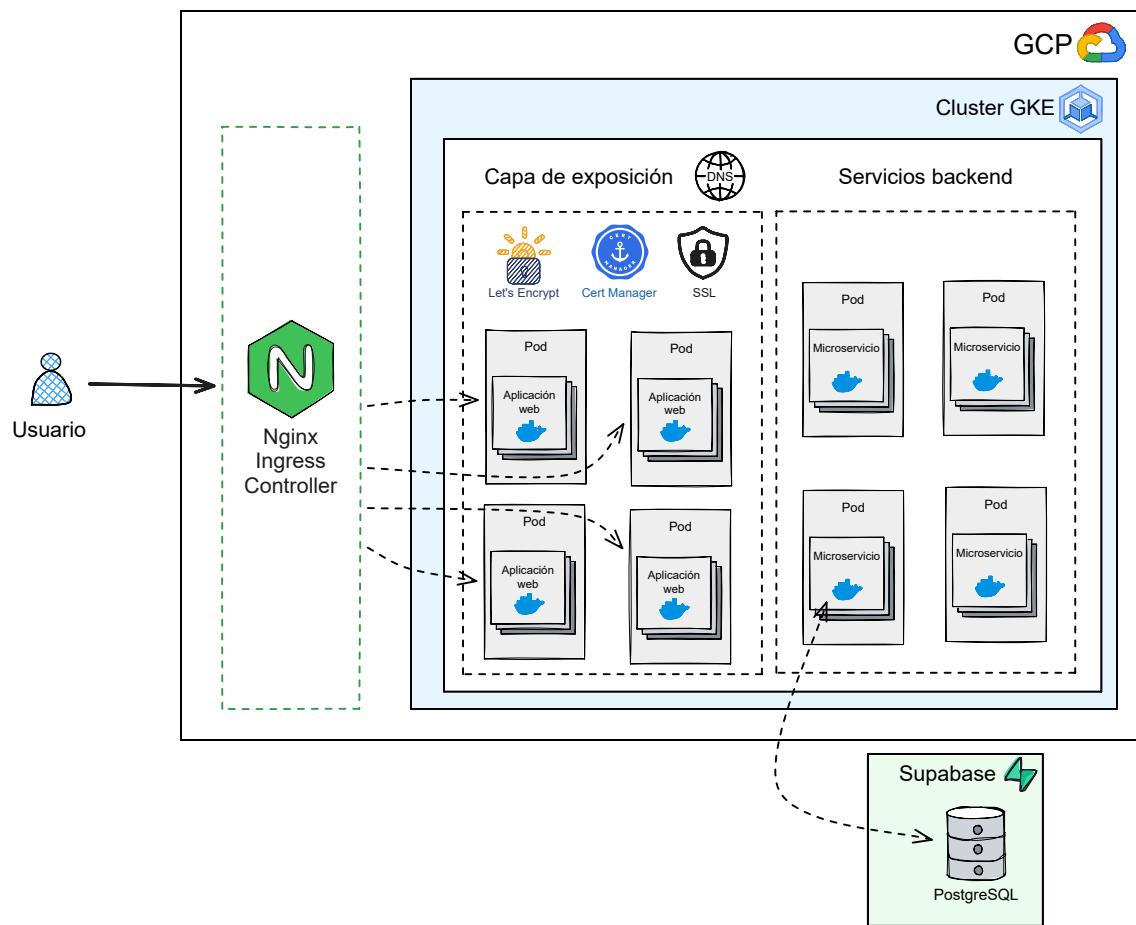


ms-empresa	Este microservicio administra los datos y procesos relacionados con las empresas. Su independencia permite manejar de forma clara las reglas del negocio asociadas a entidades organizacionales, evitando acoplamientos innecesarios con otros dominios.
ms-proveedor	Encargado de la gestión de proveedores, este servicio encapsula las operaciones vinculadas a su registro, actualización y consulta. Su separación responde a la necesidad de tratar al proveedor como un actor de negocio con comportamiento y reglas propias.
ms-prenda	Gestiona la información asociada a las prendas, constituyendo uno de los núcleos funcionales del sistema. Al aislar este dominio, se facilita la evolución del modelo de datos y de las reglas relacionadas con las prendas sin afectar a otros servicios.
ms-tallaje	Este microservicio se especializa en la administración de tallajes, permitiendo desacoplar esta lógica específica del servicio de prendas. Esto mejora la reutilización, la claridad del modelo y la mantenibilidad ante cambios en las reglas de tallas.
ms-cambios	Responsable de la gestión de cambios asociados a las prendas, este servicio permite encapsular procesos dinámicos del negocio, como modificaciones, ajustes o históricos. Su existencia favorece la trazabilidad y el control de la evolución de los datos.
ms-export	Este microservicio se encarga de la generación y exportación de información del sistema en distintos formatos. Su separación evita sobrecargar a los servicios de dominio con responsabilidades transversales y facilita la incorporación de nuevos formatos o integraciones externas.
ms-send-email	Gestiona el envío de notificaciones por correo electrónico. Al aislar esta funcionalidad, se mejora la resiliencia del sistema y se evita que fallas en servicios externos de mensajería impacten directamente en la lógica principal del negocio.



#### 6.1.4 Diagrama de Infraestructura

El diagrama de infraestructura presentado en la [Figura 6.4](#), describe la arquitectura física y cloud donde se aloja el sistema, en este caso utilizando Google Cloud Platform (GCP). Se representan los recursos que conforman el clúster de Kubernetes, los servicios gestionados utilizados, las redes, el ingress controller, los mecanismos de almacenamiento y los componentes de seguridad. Este diagrama permite visualizar la base sobre la cual se ejecuta el sistema, destacando aspectos como escalabilidad, tolerancia a fallos, orquestación de contenedores y capacidad de integración con el ecosistema cloud. Su propósito es demostrar cómo la infraestructura soporta la arquitectura lógica diseñada y garantiza la operación continua de la plataforma.



**Figura 6.4:** Diagrama de infraestructura en Google Cloud Platform (GCP)

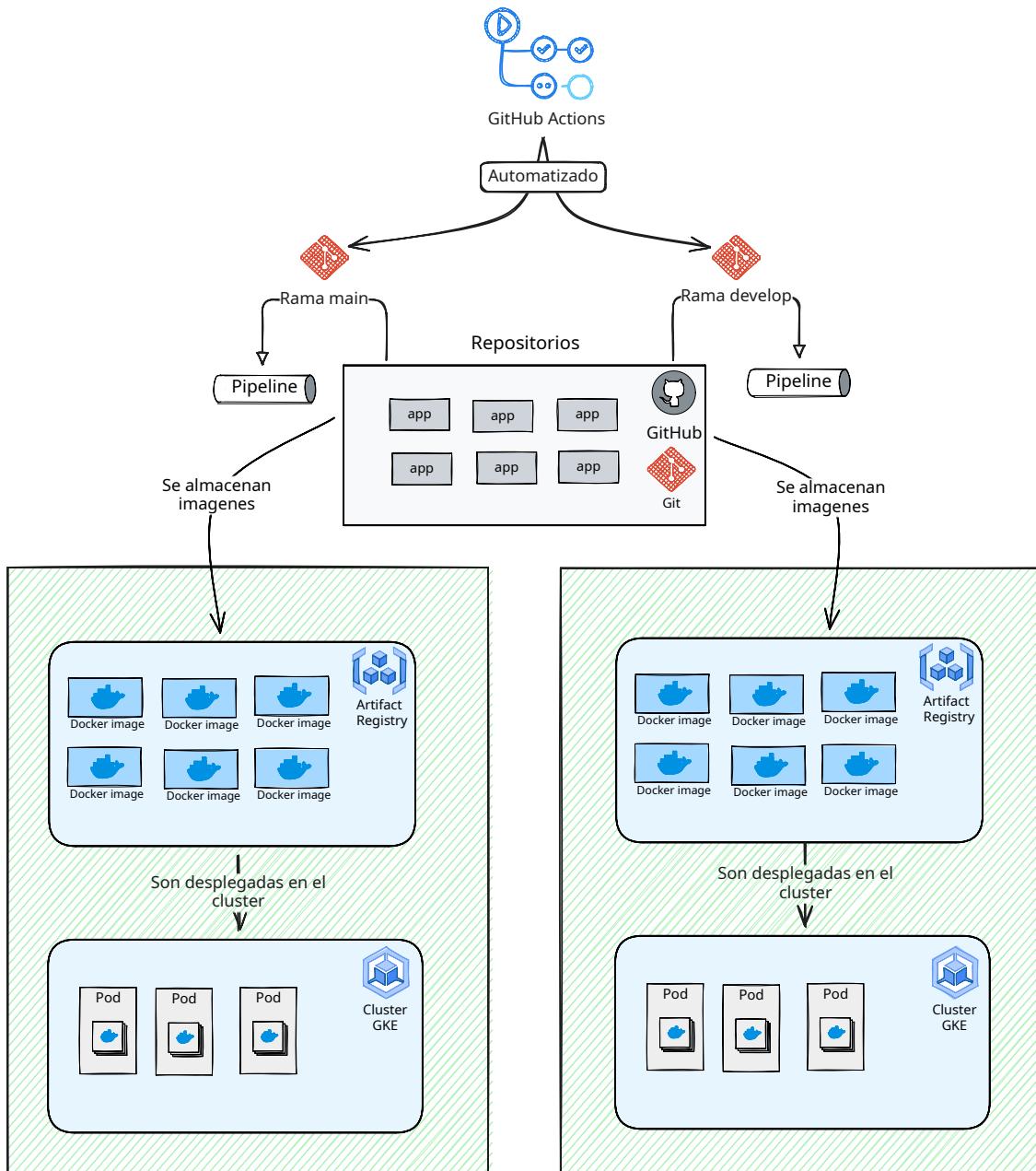
Fuente: Diseño Propio



### 6.1.5 Diagrama de Despliegue

El diagrama de despliegue presentado en la [Figura 6.5](#), representa el flujo automatizado que permite integrar, probar y desplegar el sistema de manera continua. A partir de los cambios realizados por el equipo de desarrollo, el código es versionado en un repositorio Git y procesado mediante un pipeline de integración continua, el cual construye y valida los contenedores Docker del sistema. Posteriormente, estos contenedores son desplegados mediante GitHub Actions, o algún otro sistema de flujos de trabajo, permitiendo contar con incrementos funcionales alineados con cada sprint definido en la metodología Scrum.

También permite comprender la forma en que los microservicios son desplegados a la nube, en el caso de los servicios backend, para ser consumidos por otros componentes, y para los servicios frontend, para ser expuestos a la web y que los usuarios puedan acceder a las respectivas interfaces gráficas del sistema.



**Figura 6.5:** Diagrama de despliegue con CI/CD

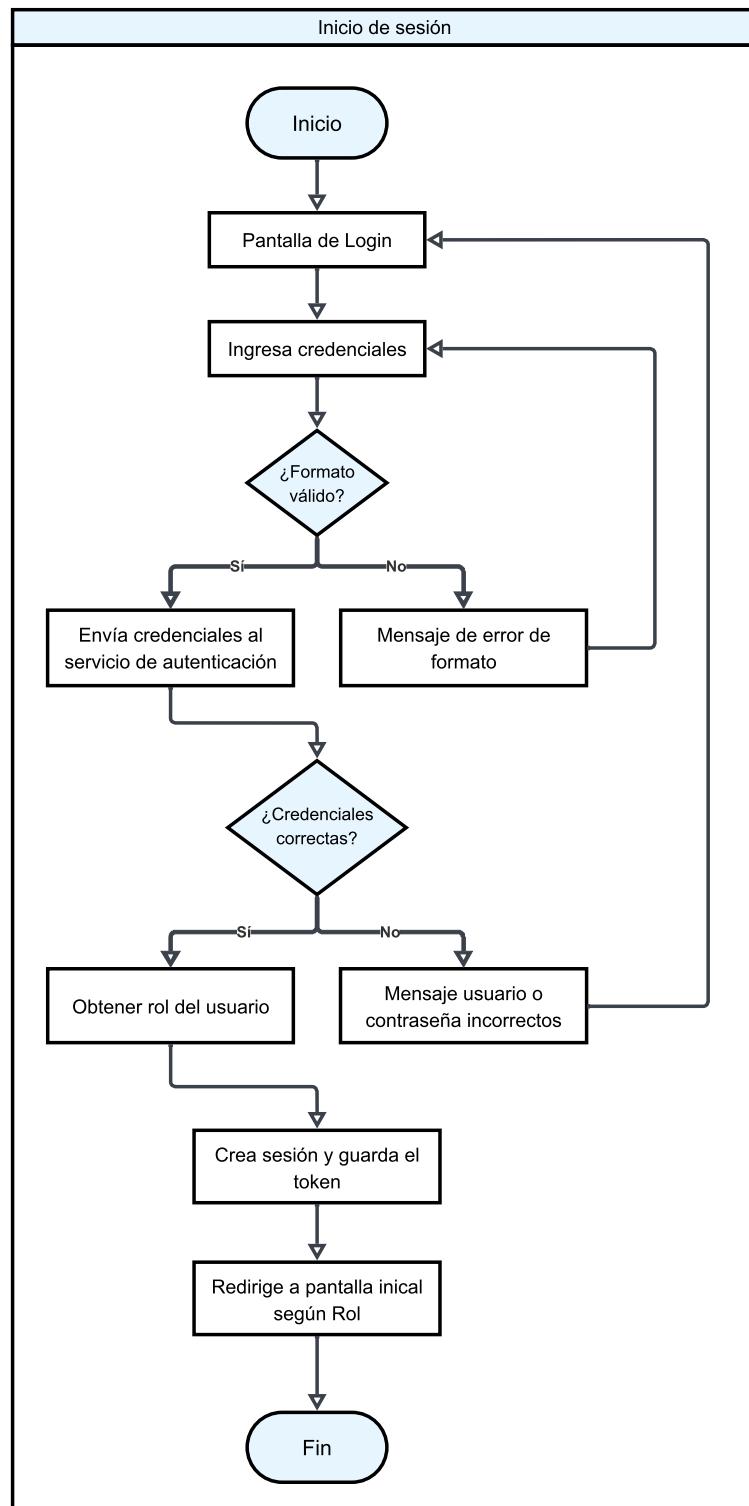
Fuente: Diseño Propio



## 6.2 DISEÑO LÓGICO

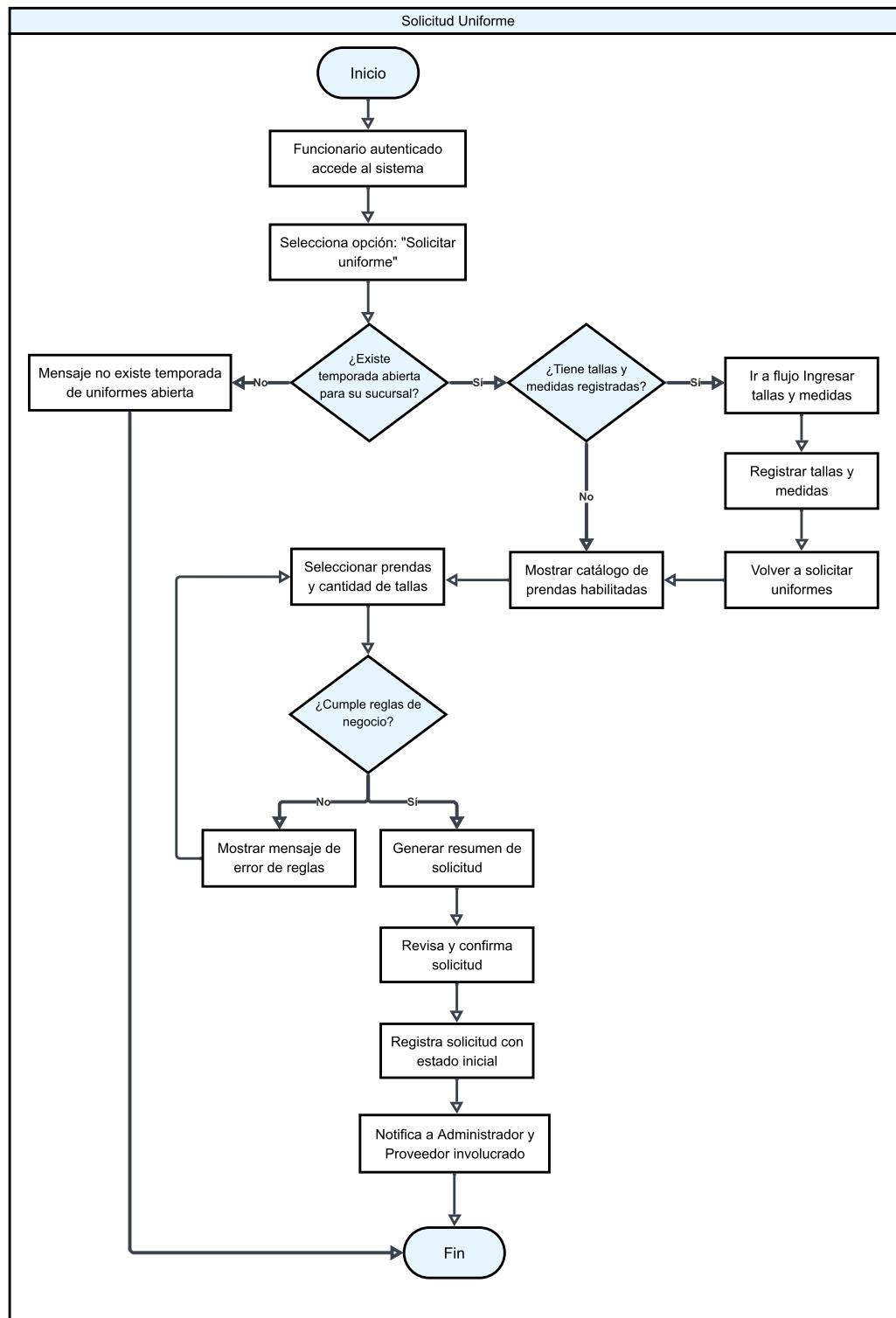
### 6.2.1 Diagramas de Flujo

En esta sección se presentan en la [Figura 6.6](#), [Figura 6.7](#) y [Figura 6.8](#); los diagramas de flujo asociados a los principales procesos del sistema, los cuales permiten identificar puntos de decisión y validaciones. Estos diagramas fueron prácticamente heredados del sistema actual de SISTAL ya que están bastante optimizados y se mantienen las reglas de negocio definidas por el cliente.



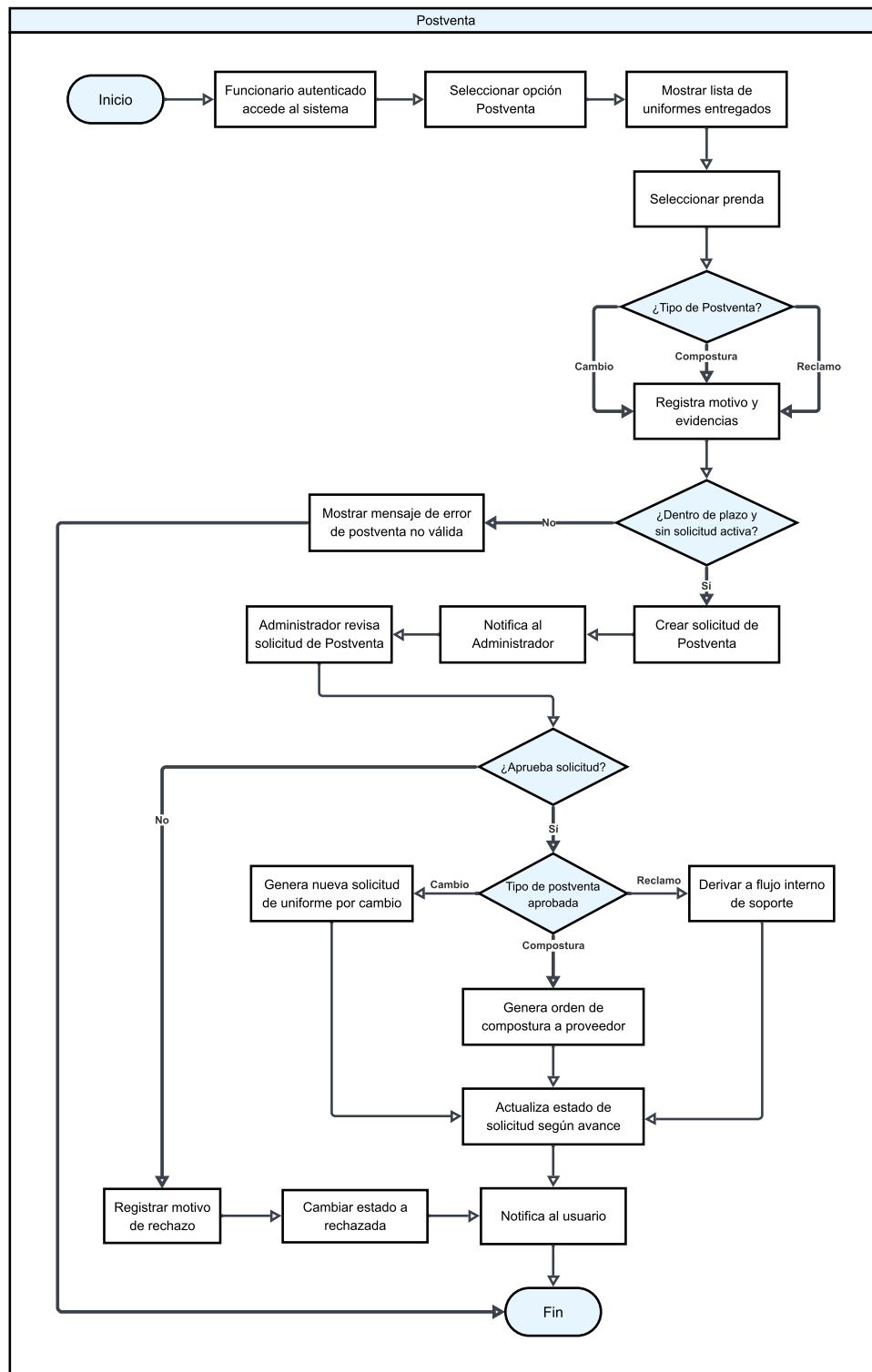
**Figura 6.6:** Diagrama de flujo Login

Fuente: *Diseño Propio*



**Figura 6.7:** Diagrama de flujo Solicitud de uniforme

Fuente: Diseño Propio



**Figura 6.8:** Diagrama de flujo Postventa

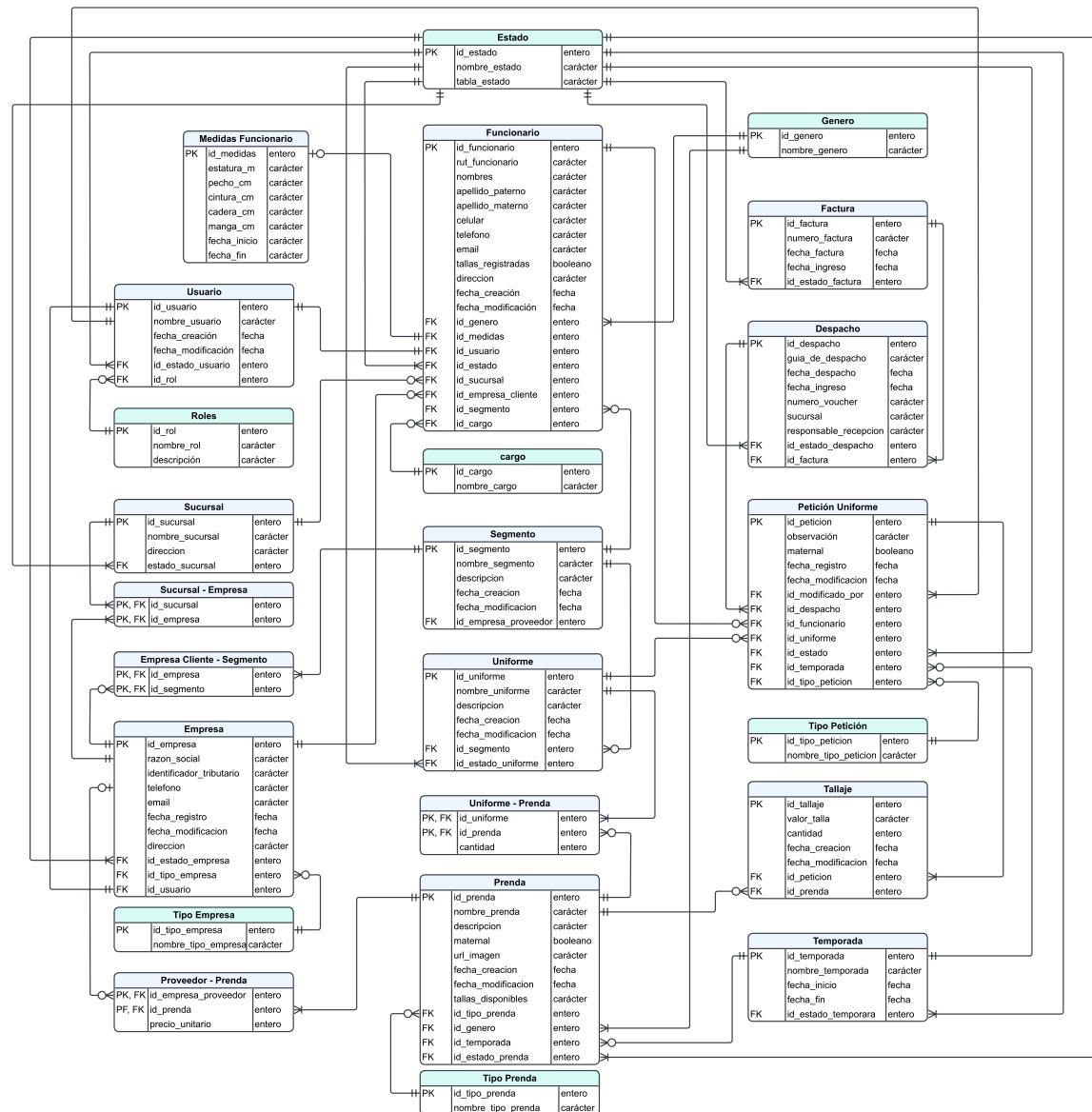
Fuente: *Diseño Propio*



## 6.3 DISEÑO FÍSICO

### 6.3.1 Modelo Entidad-Relación de Base de Datos

En la [Figura 6.9](#) se presenta la versión revisada del modelo de base de datos. Esta versión del modelo representa una mejora sustancial respecto al anterior, porque separa de forma clara los conceptos de catálogo, configuración y operación, eliminando ambigüedades y redundancias en las relaciones. Al introducir entidades intermedias bien definidas (como las relaciones asociativas entre uniforme y prenda, y entre prenda y talla), el modelo resuelve correctamente las relaciones muchos-a-muchos y permite expresar reglas de negocio reales, como la habilitación selectiva de tallas por prenda o la composición flexible de uniformes. Además, el nuevo diseño desacopla los elementos estructurales (cliente, segmento, uniforme, prenda) de los eventos operativos (asignaciones, cambios y arreglos), lo que mejora la normalización, facilita la trazabilidad y permite que el sistema evolucione sin reestructuraciones mayores.



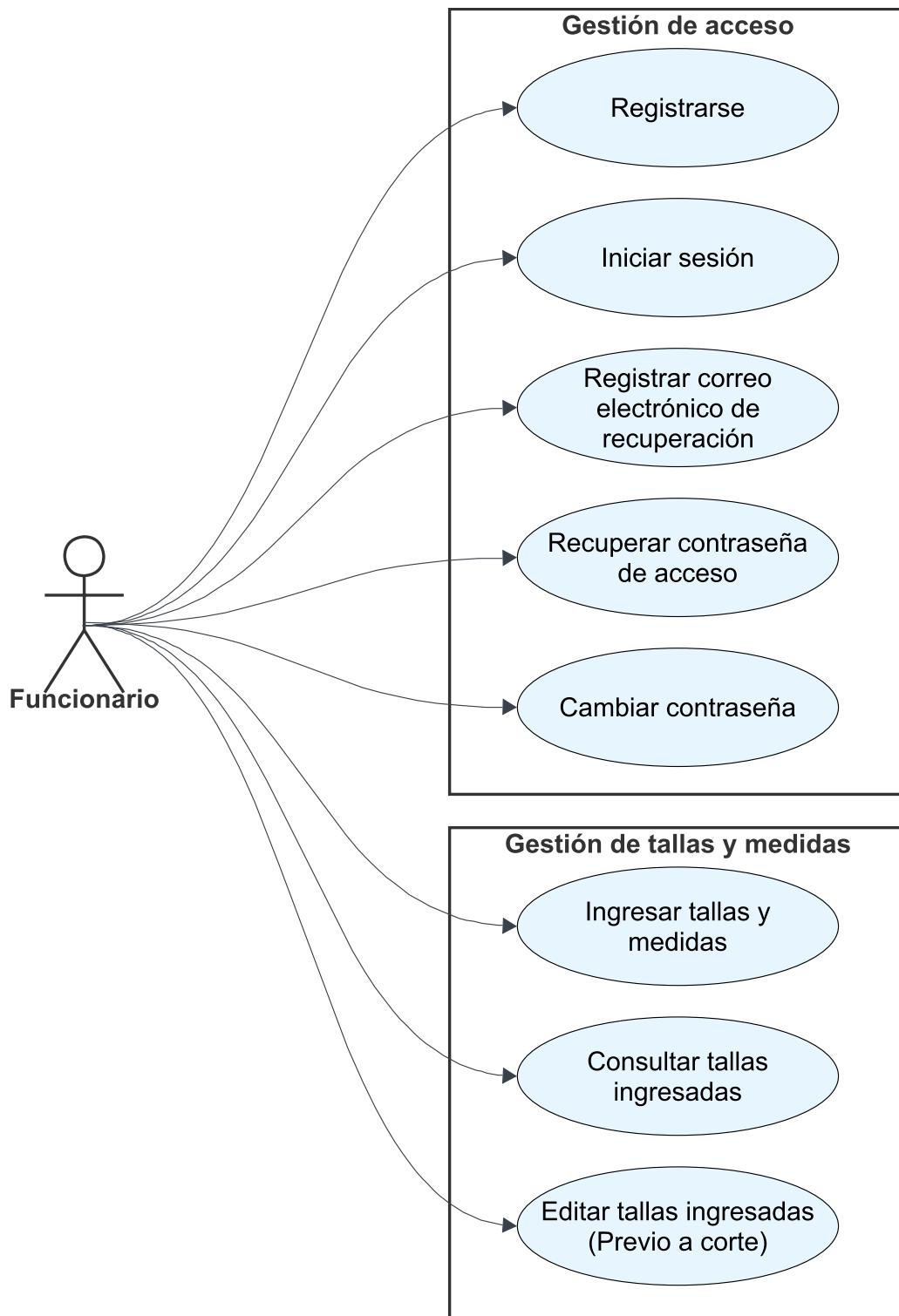
**Figura 6.9:** Diagrama entidad relación propuesto de base de datos, modelo lógico

Fuente: *Diseño Propio*



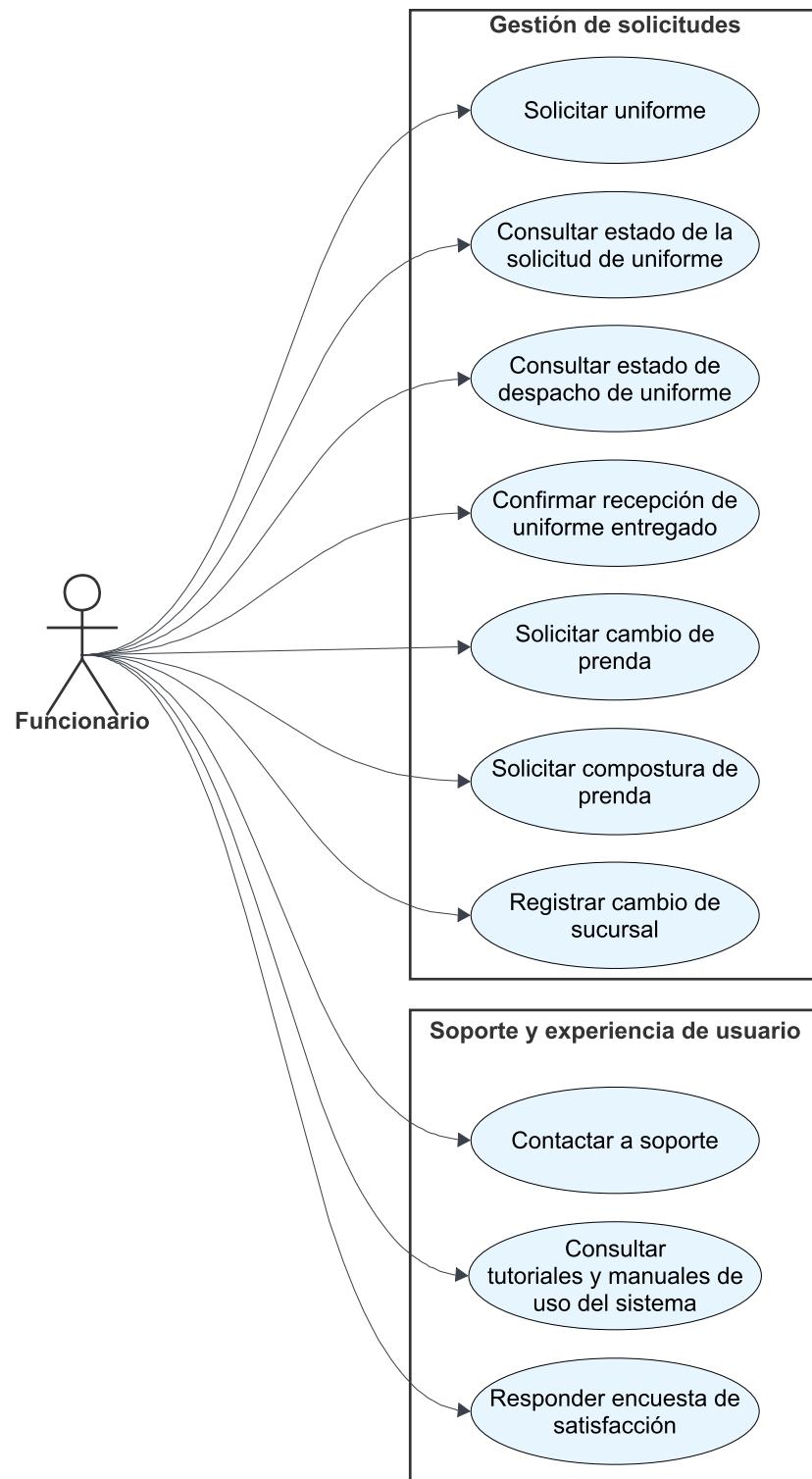
## 6.4 CASOS DE USO

En esta sección se presentan los diagramas de casos de uso que describen las principales funciones del sistema, estableciendo el alcance funcional de la solución y los límites de interacción entre el sistema y sus respectivos actores. Estos diagramas facilitan la comprensión de los requerimientos funcionales, sirven como base para el diseño de procesos y componentes, y aseguran que la implementación esté alineada con las necesidades de los usuarios y los objetivos del proyecto.



**Figura 6.10:** Diagrama de casos de uso del actor Funcionario (Parte 1)

Fuente: Diseño propio



**Figura 6.11:** Diagrama de casos de uso del actor Funcionario (Parte 2)

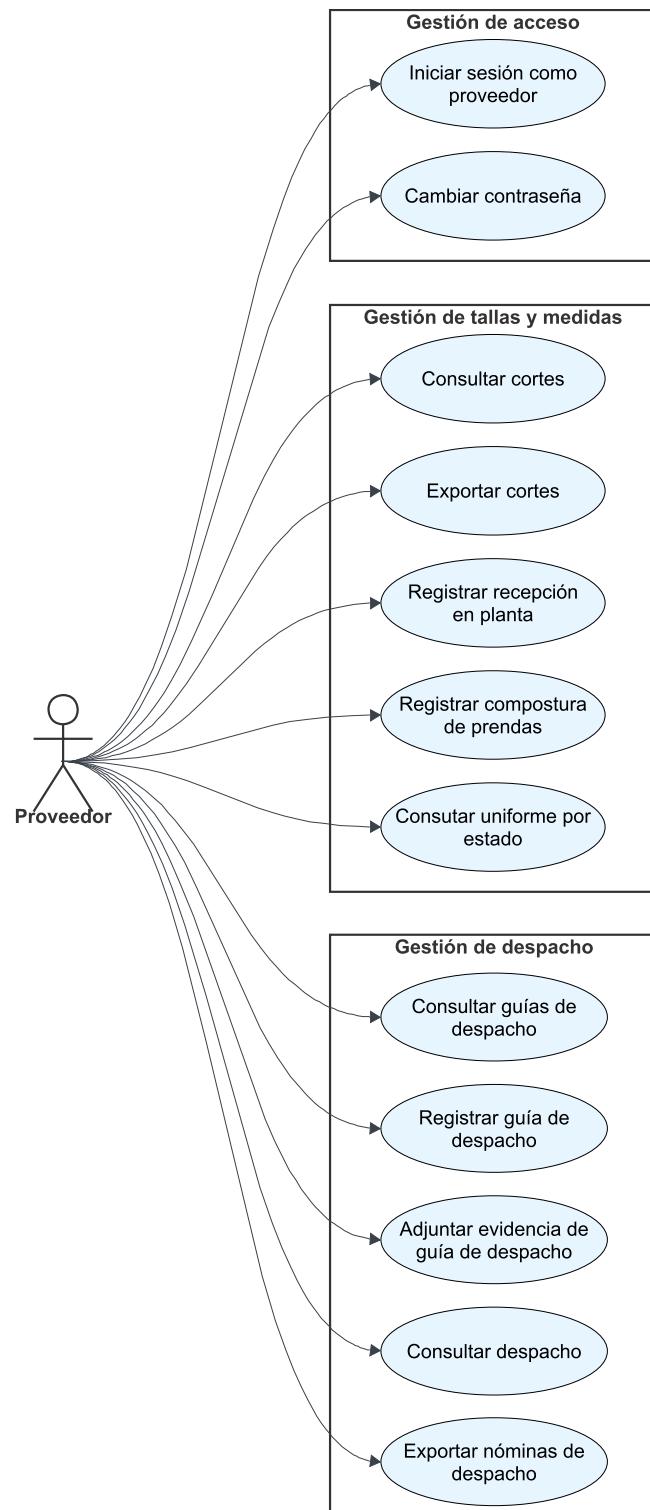
Fuente: Diseño propio



**Tabla 6.2:** Resumen de casos de uso del actor Funcionario

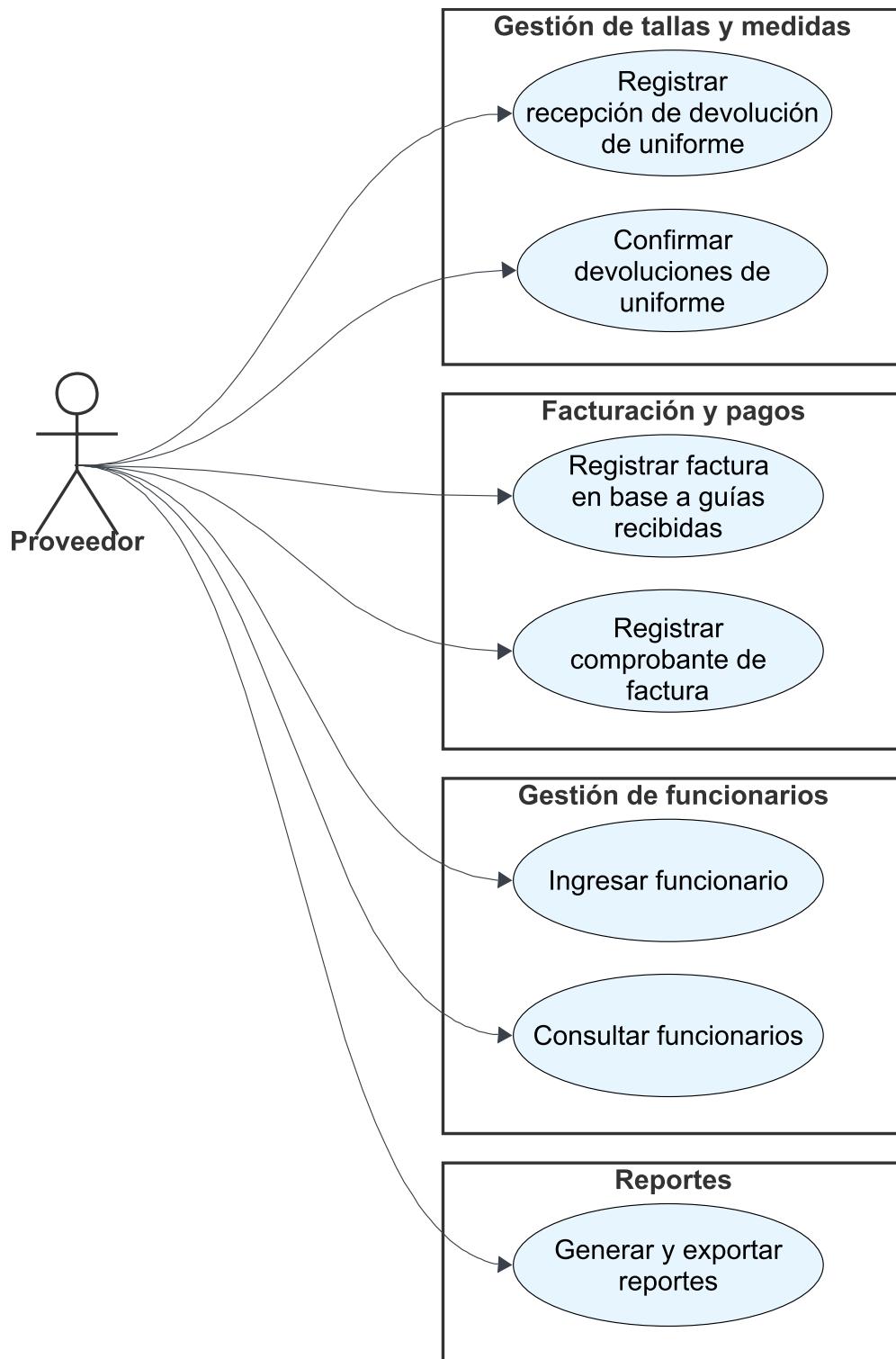
Fuente: *Elaboración Propia*

Caso de uso	Descripción breve
Registrarse como funcionario	El funcionario crea o activa su cuenta en el sistema por primera vez.
Iniciar sesión como funcionario	El funcionario se autentica para acceder a las funcionalidades del sistema.
Registrar correo electrónico de recuperación	El funcionario registra o actualiza el correo que se usará para recuperar su acceso.
Recuperar contraseña de acceso	El funcionario restablece su contraseña cuando no puede acceder al sistema.
Cambiar contraseña de funcionario	El funcionario modifica su contraseña mientras está autenticado.
Ingresar tallas y medidas	El funcionario registra sus tallas y medidas corporales en el sistema.
Consultar tallas ingresadas	El funcionario visualiza las tallas y medidas previamente registradas.
Editar tallas ingresadas previo a corte	El funcionario actualiza sus tallas y medidas mientras la orden aún no pasa a corte.
Solicitar uniforme	El funcionario genera una solicitud de entrega de uniforme según su perfil y tallas.
Consultar estado de la solicitud de uniforme	El funcionario revisa el estado actual de sus solicitudes de uniforme.
Consultar estado de despacho de uniforme	El funcionario revisa el avance del despacho asociado a su solicitud.
Confirmar recepción de uniforme entregado	El funcionario registra que recibió el uniforme solicitado.
Solicitar cambio de prenda	El funcionario solicita el cambio de una prenda ya entregada.
Solicitar compostura de prenda	El funcionario solicita arreglos o composturas sobre una prenda.
Solicitar cambio de sucursal	El funcionario registra una solicitud para cambiar su sucursal de adscripción.
Contactar a soporte	El funcionario envía consultas o incidencias al equipo de soporte del sistema.
Consultar tutoriales y manuales de uso del sistema	El funcionario accede a material de ayuda sobre el uso del sistema.
Responder encuesta de satisfacción	El funcionario completa una encuesta sobre su experiencia con el proceso de uniformes o con el sistema.



**Figura 6.12:** Diagrama de casos de uso del actor Proveedor (Parte 1)

Fuente: *Diseño Propio*



**Figura 6.13:** Diagrama de casos de uso del actor Proveedor (Parte 2)

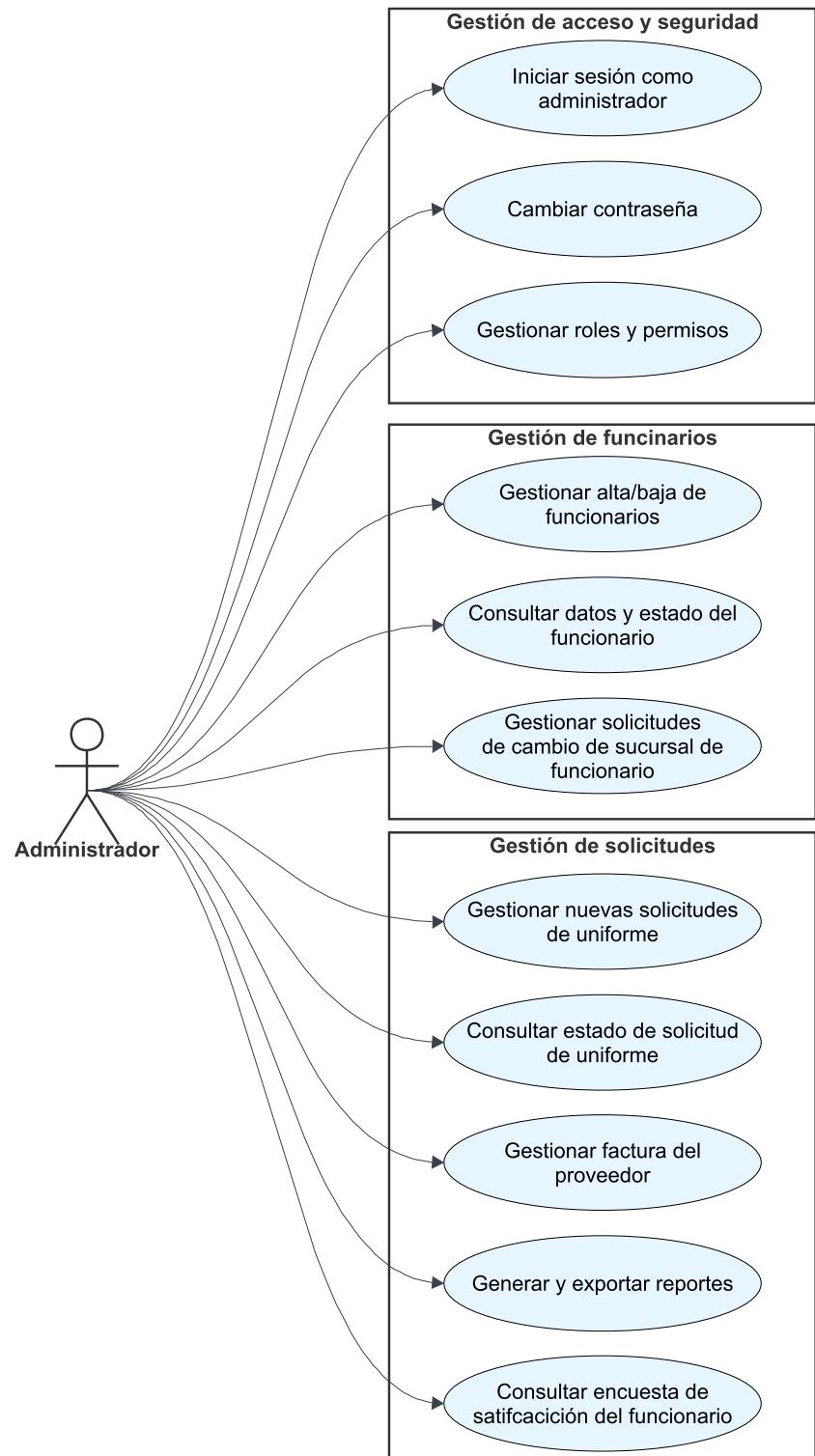
Fuente: Diseño Propio



**Tabla 6.3:** Resumen de casos de uso del actor Proveedor

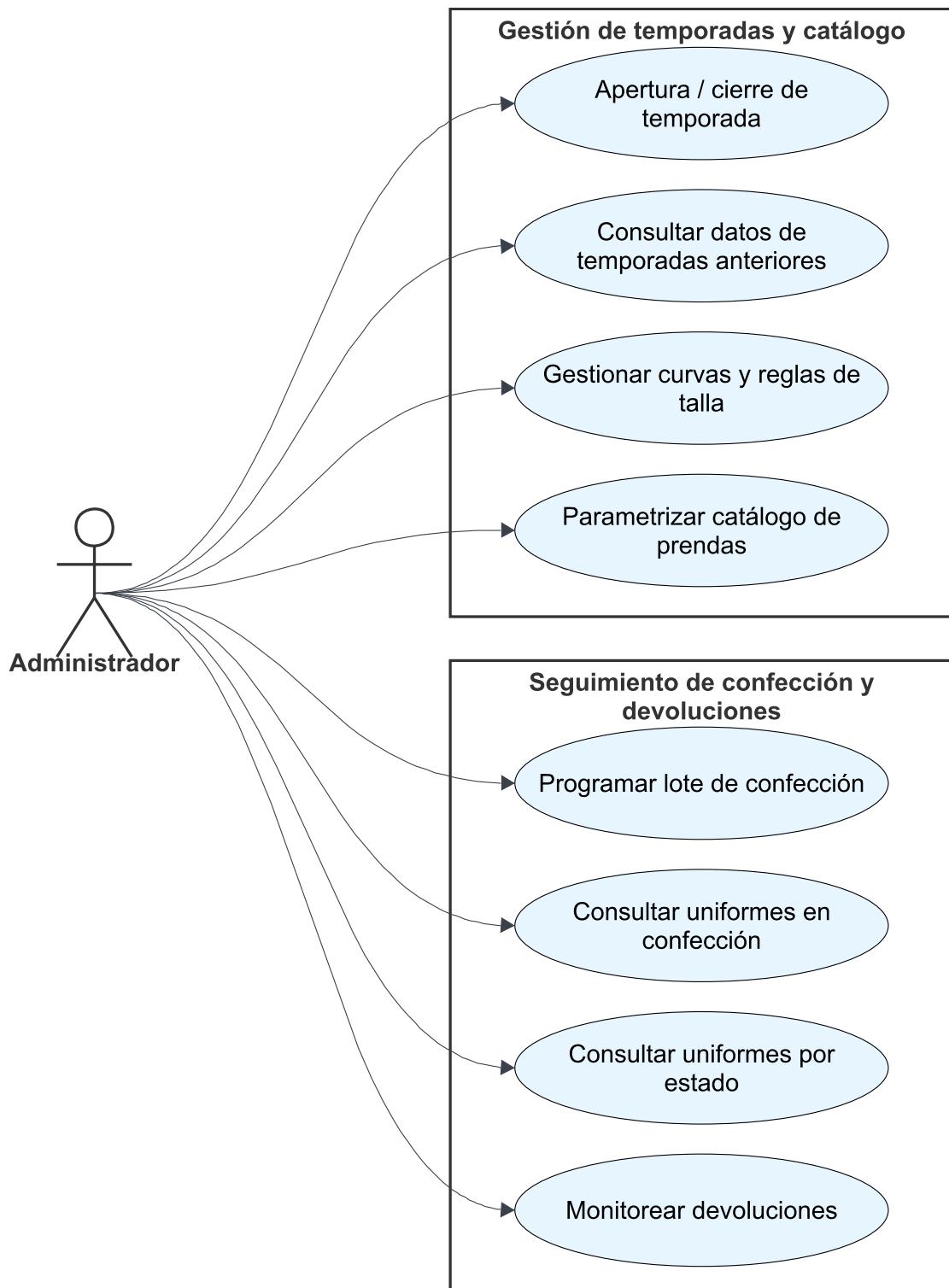
Fuente: *Elaboración Propia*

Caso de uso	Descripción breve
Iniciar sesión como proveedor	El proveedor se autentica para operar en el sistema.
Cambiar contraseña de proveedor	El proveedor modifica su contraseña de acceso.
Consultar cortes	El proveedor visualiza los cortes de confección generados para los uniformes.
Exportar cortes	El proveedor exporta la información de cortes a un archivo.
Registrar recepción en planta	El proveedor registra la recepción de uniformes o materiales en planta.
Registrar compostura de prendas	El proveedor registra las composturas realizadas sobre prendas devueltas.
Consultar uniformes por estado	El proveedor consulta los uniformes clasificados según su estado.
Consultar guías de despacho	El proveedor consulta las guías de despacho asociadas a envíos realizados.
Registrar guía de despacho	El proveedor registra una nueva guía de despacho indicando número, empresa de transporte y voucher.
Adjuntar evidencia de guía de despacho	El proveedor adjunta fotos o documentos de respaldo a una guía de despacho.
Consultar despacho	El proveedor revisa el estado de los despachos realizados hacia las sucursales.
Exportar nóminas de despacho	El proveedor exporta las nóminas de despacho de uniformes a un archivo externo.
Registrar recepción de devolución de uniforme	El proveedor registra que ha recibido una devolución de uniforme.
Confirmar devoluciones de uniforme	El proveedor valida el estado de las prendas devueltas y confirma la devolución en el sistema.
Registrar factura en base a guías recibidas	El proveedor registra una factura asociándola a las guías de despacho recibidas.
Registrar comprobante de factura	El proveedor registra comprobantes de pago o documentos contables asociados a las facturas.
Ingresar funcionario	El proveedor registra en el sistema a un nuevo funcionario asociado a sus procesos.
Consultar funcionarios	El proveedor consulta el listado y los datos de los funcionarios asociados.
Generar y exportar reportes	El proveedor genera reportes operacionales o de gestión y los exporta.



**Figura 6.14:** Diagrama de casos de uso del actor Administrador (Parte 1)

Fuente: Diseño propio



**Figura 6.15:** Diagrama de casos de uso del actor Administrador (Parte 2)

Fuente: Diseño propio



**Tabla 6.4:** Resumen de casos de uso del actor Administrador

Fuente: Elaboración Propia

Caso de uso	Descripción breve
Iniciar sesión como administrador	Actor se autentica en el sistema para gestionar la operación.
Cambiar contraseña de administrador	Actor modifica su contraseña de acceso.
Gestionar roles y permisos	Actor configura y mantiene los roles y permisos de los distintos usuarios del sistema.
Gestionar alta/baja de funcionario	Actor registra la incorporación o desvinculación de funcionarios en el sistema.
Consultar datos y estado del funcionario	Actor visualiza la información general y el estado de los funcionarios.
Gestionar solicitudes de cambio de sucursal de funcionario	Actor revisa, aprueba o rechaza solicitudes de cambio de sucursal y aplica el cambio correspondiente.
Gestionar nuevas solicitudes de uniforme	Actor recibe y gestiona nuevas solicitudes de uniforme provenientes de las sucursales o funcionarios.
Consultar estado de solicitudes de uniforme	Actor consulta el avance y el estado de las solicitudes de uniforme.
Apertura/cierre de temporada	Actor abre o cierra períodos de temporadas de uniformes.
Consultar datos de temporadas anteriores	Actor revisa información histórica de temporadas previas.
Gestionar curvas y reglas de talla	Actor configura las curvas de tallas y las reglas asociadas por tipo de prenda.
Parametrizar catálogo de prendas	Actor administra el catálogo de prendas disponibles en el sistema.
Programar lote de confección	Actor programa lotes de confección en base a solicitudes y parámetros definidos.
Consultar uniformes en confección	Actor consulta los uniformes que se encuentran en proceso de confección.
Consultar uniformes por estado	Actor consulta los uniformes agrupados por estados operacionales.
Monitorear devoluciones	Actor supervisa las devoluciones de uniformes registradas.
Gestionar factura del proveedor	Actor gestiona la revisión, registro y seguimiento de las facturas provenientes del proveedor.
Generar y exportar reportes	Actor genera reportes de gestión y los exporta.
Consultar encuestas de satisfacción del funcionario	Actor consulta y analiza los resultados de las encuestas de satisfacción respondidas por los funcionarios.



## CAPÍTULO 7

---

### DESARROLLO DEL PROYECTO

En este capítulo se describe el proceso de desarrollo del proyecto SISTAL, abordando la construcción de una solución funcional basada en la arquitectura y tecnologías definidas previamente. El desarrollo se enfocó en la implementación del módulo de funcionarios, validando la viabilidad técnica de la propuesta de reingeniería.

#### 7.1 STACK TECNOLÓGICO UTILIZADO

Haremos un repaso por las tecnologías definidas para realizar este proyecto, argumentando cada una de las decisiones e indicando los frameworks y todos los detalles respectivos.

##### 7.1.1 Tecnologías de frontend

###### React

React fue elegido como tecnología de frontend debido a su arquitectura<sup>1</sup> basada en componentes, la cual permite construir interfaces desacopladas, reutilizables y fácilmente mantenibles. Esta característica resulta fundamental en

---

<sup>1</sup>Arquitectura aquí hace referencia al patrón de diseño y organización de los archivos o componentes de código.



aplicaciones donde existen múltiples vistas y flujos de usuario diferenciados, como es el caso de SISTAL.

Desde un punto de vista técnico, React facilita:

- La separación clara entre lógica de presentación y lógica de negocio.
- El consumo eficiente de APIs REST.
- La escalabilidad del frontend sin afectar el backend.

Su adopción permite superar una de las principales limitaciones del sistema legacy, donde la interfaz y la lógica de negocio se encontraban fuertemente acopladas.

## Tailwind CSS

Tailwind CSS fue seleccionado como framework de estilos debido a su enfoque utility-first, el cual permite construir interfaces consistentes sin necesidad de escribir grandes cantidades de CSS personalizado y alineándose muy bien con la arquitectura de componentes de React.

Las razones técnicas de su elección incluyen:

- Reducción del acoplamiento entre estilos y componentes.
- Mayor velocidad de desarrollo y menor complejidad en el mantenimiento.

En el contexto de SISTAL, Tailwind facilita la rápida implementación de interfaces alineadas con los prototipos definidos, manteniendo un diseño coherente y controlado.



## 7.1.2 Tecnologías de backend

### Go

Go fue seleccionado como lenguaje de backend por su eficiencia, simplicidad y orientación al desarrollo de servicios.

Las razones técnicas principales son:

- Lenguaje compilado y tipado, que reduce errores en tiempo de ejecución.
- Alto rendimiento y bajo consumo de recursos.
- Excelente manejo de concurrencia mediante goroutines.

Estas características lo hacen especialmente adecuado para arquitecturas basadas en microservicios y despliegues contenerizados, como los propuestos en SISTAL.

### Gin

Gin se utiliza como framework HTTP debido a su ligereza y alto rendimiento.

Su elección se justifica porque:

- Proporciona un routing eficiente y claro.
- Introduce un mínimo de abstracciones, manteniendo control sobre la lógica.
- Facilita la construcción de APIs REST bien definidas.

Gin permite desarrollar servicios backend simples, rápidos y fáciles de mantener, alineados con la filosofía de Go.



## GORM

GORM fue seleccionado como ORM para Go debido a su madurez y facilidad de integración con PostgreSQL.

Desde el punto de vista técnico:

- Simplifica el acceso a datos mediante abstracción del SQL.
- Reduce errores comunes en consultas manuales.
- Facilita el mapeo entre modelos de dominio y tablas relacionales.

Su uso mejora la mantenibilidad del código y favorece la coherencia entre el modelo de datos y la lógica de negocio.

### 7.1.3 Persistencia de datos

#### PostgreSQL

PostgreSQL fue elegido como sistema gestor de base de datos por su robustez, confiabilidad y cumplimiento de estándares ACID.

Las razones técnicas incluyen:

- Soporte completo para integridad referencial.
- Manejo eficiente de transacciones.
- Alta estabilidad en entornos productivos.

Esta elección permite resolver las deficiencias estructurales del sistema legacy, garantizando consistencia y trazabilidad de la información.



## 7.1.4 Infraestructura y DevOps

### Docker

Docker se utiliza para contenerizar los distintos componentes del sistema.

Desde el punto de vista técnico:

- Garantiza consistencia entre entornos.
- Reduce errores de despliegue.
- Facilita la portabilidad de los servicios.

Docker es un pilar fundamental para una arquitectura cloud-native.

### Kubernetes

Kubernetes se emplea como plataforma de orquestación de contenedores.

Su elección se justifica por:

- Escalabilidad automática de servicios.
- Alta disponibilidad.
- Gestión centralizada del ciclo de vida de aplicaciones.

Kubernetes permite operar el sistema de forma robusta y preparada para crecimiento futuro.



## Google Cloud Platform

GCP fue seleccionada como plataforma cloud por su integración nativa con Kubernetes y su enfoque en servicios administrados.

Las razones técnicas incluyen:

- Google Kubernetes Engine (GKE) como servicio gestionado.
- Infraestructura altamente disponible.
- Reducción de carga operativa.

GCP permite desplegar SISTAL en un entorno real de producción, validando la arquitectura propuesta.

## GitHub Actions

GitHub Actions se utiliza como herramienta de integración y despliegue continuo.

Técnicamente aporta:

- Automatización del ciclo de desarrollo.
- Integración nativa con GitHub.
- Despliegues repetibles y controlados.

Su adopción asegura calidad, trazabilidad y eficiencia en el proceso de entrega de software.



## 7.2 GESTIÓN DEL PROYECTO Y TRABAJO COLABORATIVO

La siguiente sección explica como se utilizó la metodología de trabajo seleccionada y como se organizó el trabajo del proyecto.

### 7.2.1 Metodología de Trabajo

La metodología seleccionada para la etapa de desarrollo del proyecto fue *Scrum*, por tratarse de un marco de trabajo ágil orientado a la gestión incremental del software, que promueve la adaptación al cambio, la entrega continua de valor y una comunicación constante entre los integrantes del equipo, además de alinearse muy bien con las metodologías DevOps.

El desarrollo se estructuró en un ciclo de tres semanas, organizado en iteraciones cortas en las que se planificaron, implementaron y revisaron funcionalidades priorizadas. Este enfoque facilitó el seguimiento del avance, la detección temprana de problemas y la incorporación de mejoras de manera progresiva, alineando el resultado final con los objetivos planteados desde el inicio del proyecto.

Como herramienta de gestión de proyectos se utilizó **GitHub Projects**, ya que se integra directamente con los repositorios y el flujo de trabajo basado en ramas que se mencionan a continuación, centralizando las tareas asignadas y la revisión de código.



### 7.2.2 Flujo de Trabajo con Git

Como herramienta de control de versiones de los microservicios se utilizó el clásico **Git** y para complementar con un flujo de desarrollo organizado y alineado a prácticas DevOps se utilizó un modelo adaptado de **GitFlow** con las siguientes características:

- Rama principal **Main** donde está la versión estable de cada microservicio y la cual estará en uso por los clientes.
- Rama **QA** donde se aloja una nueva versión del microservicio que se encuentra en entorno de pruebas para pronto ser integrada en *Main*
- Rama **Develop** donde se encuentra la última versión con los cambios realizados por el equipo de desarrollo, si se decide lanzar esta versión se integra a la rama QA para que sea probada en un entorno de preparación.
- Rama **Feature** donde se implementa una nueva tarea acordada en la planificación. Esta rama primero se copia de *Develop* para trabajar sin alterar el trabajo ya realizado y al finalizarse se vuelve a integrar a *Develop* donde los desarrolladores deben aprobar su validación antes.

### 7.2.3 Organización en GitHub

Siguiendo los diagramas de arquitectura y despliegue previamente expuestos de la [Figura 6.3](#) y [Figura 6.5](#), se creó un repositorio por microservicio, cada uno siguiendo una plantilla de despliegue que permitira al comenzar el desarrollo que cada versión nueva se pueda testear inmediatamente en un servidor de pruebas.

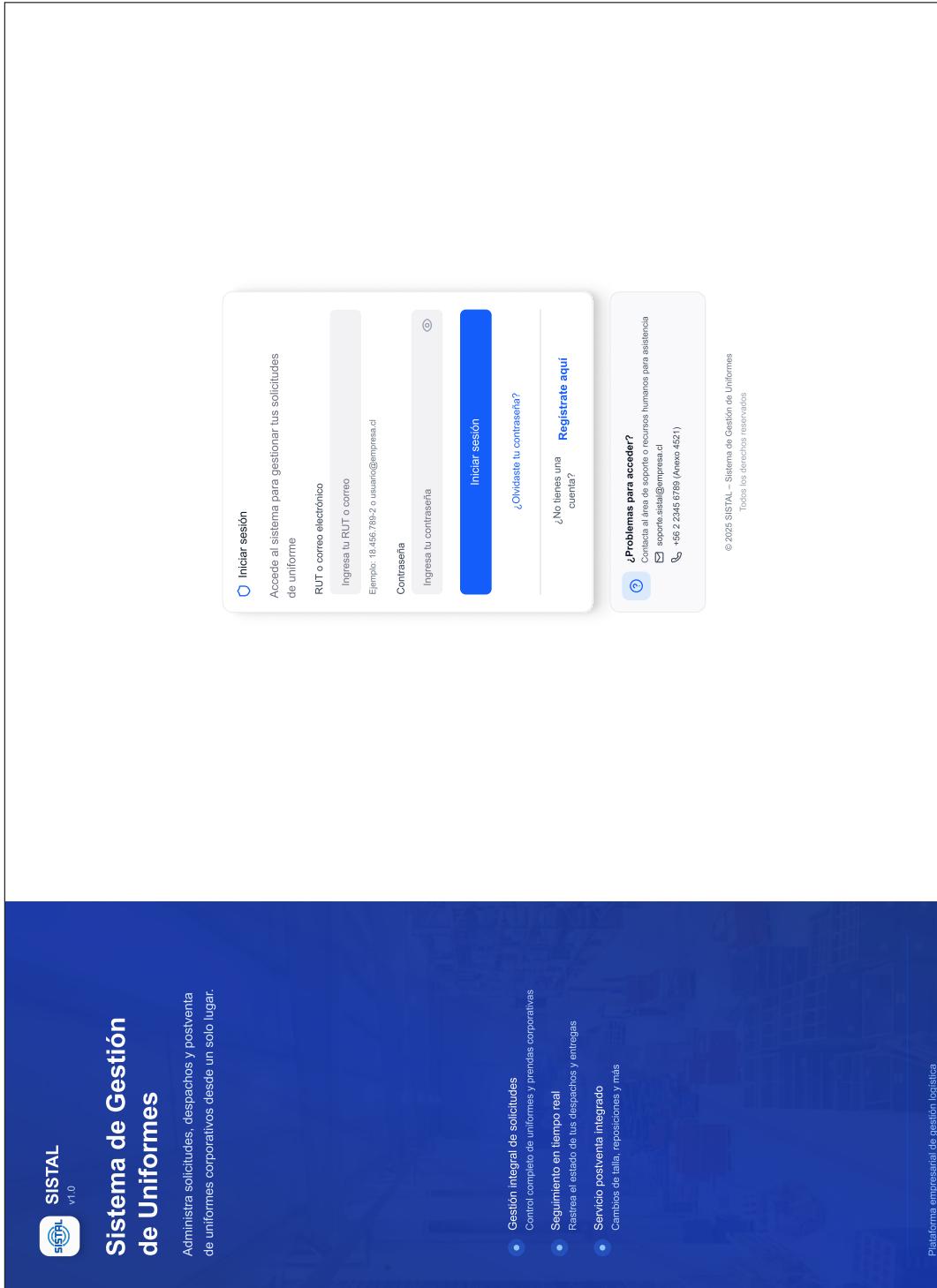


## 7.3 DISEÑO DE INTERFAZ DE USUARIO

En esta sección se presentan los diseños de la nueva interfaz de usuario destinada a los funcionarios del sistema. Esta introduce una serie de mejoras orientadas a optimizar la usabilidad, accesibilidad y coherencia visual del sistema, entre las cuales destacan las siguientes:

- **Estandarización de paneles de control:** se adopta un diseño más estandarizado en los paneles de control, alineado con patrones de diseño ampliamente utilizados en aplicaciones modernas. Esto facilita la comprensión de la información presentada, reduce la curva de aprendizaje y genera una experiencia más familiar e intuitiva para los usuarios.
- **Diseño adaptativo (responsive):** la interfaz fue concebida bajo principios de diseño adaptativo, permitiendo su correcta visualización y funcionamiento en distintas resoluciones y tamaños de pantalla. Esto garantiza una experiencia consistente tanto en equipos de escritorio como en dispositivos portátiles.
- **Mayor consistencia visual:** el uso homogéneo de colores, tipografías y componentes gráficos contribuye a una identidad visual coherente, fortaleciendo la percepción de profesionalismo y facilitando la navegación entre las distintas secciones del sistema.

Las interfaces fueron diseñadas utilizando la herramienta **Figma**, lo que permitió prototipar y validar visualmente la experiencia de usuario antes de su implementación. Con el objetivo de mejorar la legibilidad y visualización de los elementos, algunas de las figuras se presentan en formato horizontal.



**Figura 7.1:** GUI de inicio de sesión  
**Fuente:** *Elaboración Propia en Figma*



## SISTAL v1.0

### Sistema de Gestión de Uniformes

Administra solicitudes, despachos y postventa de uniformes corporativos desde un solo lugar.

Completa tus datos para acceder al sistema de gestión de uniformes

RUT \*

Ej: 18.456.789-2

Nombre completo \*

Juan Carlos Pérez González

Correo electrónico corporativo \*

usuario@empresa.cl

Contraseña \*

Crea una contraseña segura

Confirmar contraseña \*

Vuelve a ingresar tu contraseña

Crear cuenta

¿Ya tienes una cuenta?

Iniciar sesión

Problemas para acceder?

Contacta al área de soporte o recursos humanos para asistencia

soporte.sistal@empresa.cl

+56 2 2345 0789 (Anexo 4521)

© 2025 SISTAL – Sistema de Gestión de Uniformes

Todos los derechos reservados

Plataforma empresarial de gestión logística

**Figura 7.2: GUI de registro de usuario**  
*Fuente: Elaboración Propia en Figma*



The screenshot displays the main interface for employees of the Universidad Tecnológica Metropolitana (UTEM). At the top, the UTEM logo is visible. The header includes a search bar labeled "Buscar solicitudes", a notification bell icon, and a user profile for "Juan Pérez" (Funcionario JP).

The main content area features a large banner titled "Temporada de uniformes 2025" with the sub-instruction "Revisa tus solicitudes y asegúrate de tener tus tallas actualizadas". Below the banner are three buttons: "Solicitar uniforme" (Request uniform), "Ver calendario de entregas" (View delivery calendar), and tabs for "Resumen", "Historial", and "Tallas".

The "Mis solicitudes recientes" section lists four recent requests:

Fecha	Estado	Descripción
10 Dic 2024	En proceso	Uniforme completo
05 Dic 2024	Aprobado	Zapatos de seguridad
28 Nov 2024	Entregado	Chaleco reflectante
15 Nov 2024	En revisión	Camiseta corporativa

[Ver todas mis solicitudes](#)

The "Próximas entregas" section shows three scheduled deliveries:

Camiseta de manga corta	25 de noviembre
Pantalón cargo	28 de noviembre
Zapatos de seguridad	05 de diciembre

The "Atajos rápidos" section provides quick links:

- + Nueva solicitud
- Ver despachos
- Cambiar contraseña

The "Mis tallas" section displays current measurements for size "M":

Talla general	M
Pecho	96 cm
Cintura	82 cm
Cadera	98 cm
Actualizar tallas	<a href="#">Actualizar tallas</a>

The "Estado general" section shows a summary of 22 total requests across four categories:

Categoría	Cantidad
Entregados	12
En proceso de entrega	5
Pendientes de aprobación	3
Requieren acción	2

[Cerrar sesión](#)

**Figura 7.3: GUI Panel principal de Funcionario**

Fuente: Elaboración Propia en Figma



## Notificaciones

[Marcar todas como leídas](#)

Tienes 3 notificaciones nuevas

**Solicitud aprobada** ●  
Tu solicitud #2024-0847 ha sido aprobada y está en proceso de preparación.  
⌚ Hace 5 minutos

**Entrega programada** ●  
Tu pedido será entregado el 20 de diciembre en la Sucursal Centro.  
⌚ Hace 1 hora

**Documentación pendiente** ●  
Debes completar la documentación para tu solicitud de cambio de talla.  
⌚ Hace 3 horas

**Entrega confirmada**  
Has recibido correctamente tu pedido #2024-0832.  
Gracias por confirmar.

[Ver todas las notificaciones >](#)

**Figura 7.4:** GUI Botón de notificaciones

Fuente: Elaboración Propia en Figma

Inicio > Mis solicitudes

Juan Pérez JP Funcionario

Buscar solicitudes

Mis solicitudes

Consulta el estado y detalle de tus solicitudes de uniforme y postventa

Solicitudes activas

3 En proceso de gestión

Seguimiento de despachos

Cambio de sucursal

Próxima entrega

**20 Dic 2024**

Camisa, pantalón, zapatos - Talla M

Todos los estados > Todos los tipos > Todo el periodo >

+ Nueva solicitud

Nº Solicitud	Fecha	Tipo	Descripción	Estado	Última actualización	Acciones
SOL-2024-1547	12/12/2024	Uniforme completo	Camisa, pantalón, zapatos - Talla M	En proceso	13/12/2024 - 14:30	Ver detalle
SOL-2024-1432	28/11/2024	Reposición	Pantalón de trabajo - Talla M	Aprobado	05/12/2024 - 10:15	Ver detalle
SOL-2024-1389	15/11/2024	Cambio de talla	Camisa corporativa - De M a L	Pendiente	15/11/2024 - 16:45	Ver detalle
SOL-2024-1201	03/10/2024	Uniforme completo	Kit completo de invierno - Talla M	Entregado	25/10/2024 - 09:00	Ver detalle

Cerrar sesión

**Figura 7.5: GUI solicitudes realizadas por el funcionario**

Fuente: Elaboración Propia en Figma



SISTAL

Inicio > Cambio de sucursal

Buscar solicitudes Juan Pérez JP Funcionario

### Solicitud de cambio de sucursal

Solicita el cambio de tu sucursal de adscripción para futuras entregas y gestiones

1. Información actual

Sucursal actual: Sucursal Santiago Centro | Fecha de última actualización: 15/03/2024

Dirección: Av. Libertador Bernardo O'Higgins 1234 | Comuna: Santiago Centro

Región: Región Metropolitana

2. Nueva sucursal solicitada

Nueva sucursal: Selecione la nueva sucursal

Información importante: El cambio de sucursal se aplicará a futuras solicitudes y despachos. Las solicitudes activas no se verán afectadas.

3. Motivo del cambio

Motivo del cambio: Selecione el motivo del cambio

Descripción del motivo: Describe brevemente el motivo del cambio de sucursal

La información proporcionada será utilizada para validar la solicitud

4. Confirmación y reglas

Consideraciones importantes:

- El cambio está sujeto a aprobación del administrador.
- El proceso de validación puede tardar algunos días hábiles.
- Las solicitudes activas no se verán afectadas por este cambio.
- Solo podrás realizar un cambio de sucursal cada 90 días.

Confirmo que la información ingresada es correcta y que he leído las consideraciones del cambio de sucursal

Cerrar sesión

Historial de cambios de sucursal

Fecha de solicitud	Sucursal anterior	Nueva sucursal	Estado	Fecha de resolución
15/03/2024	Sucursal Las Condes	Sucursal Santiago Centro	Aprobado	18/03/2024
10/11/2023	Sucursal Providencia	Sucursal Las Condes	Aprobado	15/11/2023
22/08/2023	Sucursal Maipú	Sucursal Providencia	Rechazado	25/08/2023

**Figura 7.6: GUI de solicitud de cambio de sucursal**

Fuente: Elaboración Propia en Figma



Inicio > Cambios de talla / prenda

Buscar solicitudes

Juan Pérez JP Funcionario

### Solicitud de cambio de talla / prenda

Solicita el cambio de una prenda entregada por talla incorrecta o defecto

**1. Selección de solicitud base**

Solicitud entregada

Seleccione una solicitud entregada

**2. Motivo del cambio**

Seleccione el motivo

Talla incorrecta  
Prenda defectuosa  
Error en despacho  
Otro

Descripción del motivo

Describe brevemente el motivo del cambio

Proporcione detalles específicos que ayuden a validar la solicitud

**3. Nueva selección**

Nueva talla

Seleccione la nueva talla

Nueva prenda (opcional)

Seleccione si desea cambiar de prenda

**○ Información importante**  
El cambio está sujeto a validación según las reglas del sistema. El tiempo de procesamiento puede variar según disponibilidad de stock.

**4. Evidencia (opcional)**

Archivos adjuntos

Haga clic para cargar archivos o arrastre aquí

PNG, JPG, PDF hasta 10MB

Adjunte fotografías de la prenda o documentos que respalden la solicitud de cambio

Cerrar sesión

Cancelar

Enviar solicitud de cambio

**Figura 7.7: GUI de solicitud de cambio de talla**

Fuente: *Elaboración Propia en Figma*



Inicio > Seguimiento de despachos

Buscar despacho / solicitud

Juan Pérez JP Funcionario

Despachos entregados

5 Completados exitosamente

Próxima entrega

2 Dic 2024 Chaqueta de invierno – Talla M

Entregas en camino

Todos los estados

Últimos 30 días

Mis solicitudes

Cambios de talla / prenda

Seguimiento de despachos

Cambio de sucursal

Mi cuenta

Nº Solicitud Fecha de despacho Prenda(s) Dirección de entrega Estado Fecha estimad

SOL-2024-1547	13/12/2024	Chaqueta de invierno - Talla M	② Av. Libertador Bernardo O'Higgins 1234, Santiago Centro	En tránsito	22/12/2024
SOL-2024-1432	10/12/2024	Pantalón de trabajo - Talla M, Camisa corporativa - Talla M	② Av. Providencia 567, Providencia	En tránsito	18/12/2024
SOL-2024-1389	05/12/2024	Zapatos de seguridad - Talla 42	② Av. Apoquindo 3000, Las Condes	Entregado	10/12/2024
SOL-2024-1201	28/11/2024	Kit completo de invierno - Talla M (Chaqueta, Pantalón, Gorro)	② Av. Vicuña Mackenna 4567, Macul	Entregado	02/12/2024
SOL-2024-1156	20/11/2024	Chaleco reflectante - Talla L	② Av. Libertador Bernardo O'Higgins 1234, Santiago Centro	Entregado	25/11/2024
SOL-2024-1089	15/11/2024	Uniforme completo - Talla M (Camisa, Pantalón, Zapatos)	② Av. Providencia 567, Providencia	Entregado	20/11/2024
SOL-2024-0987	08/11/2024	Parka de invierno - Talla L	② Av. Apoquindo 3000, Las Condes	Entregado	13/11/2024

→ Cerrar sesión

**Figura 7.8: GUI de seguimiento de despachos**

Fuente: Elaboración Propia en Figma



The screenshot displays the 'Mi cuenta' (My Account) page of the SISTAL system. At the top right, there is a user profile for 'Juan Pérez' (Funcionario JP). A navigation sidebar on the left includes links for 'Inicio', 'Mis solicitudes', 'Cambios de talla / prenda', 'Seguimiento de despachos', 'Cambio de sucursal', and 'Mi cuenta'. The main content area is divided into several sections:

- Información personal:** Shows details like Name (Juan Carlos Pérez González), RUT (18.456.789-2), Email (juan.perez@empresa.cl), Cargo (Analista de Operaciones), and Current Branch (Sucursal Santiago Centro, status Active).
- Seguridad de la cuenta:** Includes fields for Recovery Email (juan.perez.personal@gmail.com) and Password (\*\*\*\*\*). A note at the bottom of this section advises changing the password periodically.
- Preferencias del sistema:** Allows setting the interface language (Español) and notification preferences (Receive notifications via email, Show notifications within the system). It also includes a text size accessibility option (Medium, Predefined).
- Actividad reciente de la cuenta:** A table showing recent activity logs:

Fecha y hora	Acción	Origen	Dispositivo
14/12/2024 - 14:32	Inicio de sesión exitoso	Santiago, Chile	Escritorio
13/12/2024 - 09:15	Solicitud de cambio de talla enviada	Santiago, Chile	Navegador
12/12/2024 - 16:45	Inicio de sesión exitoso	Santiago, Chile	Móvil
10/12/2024 - 11:20	Cambio de contraseña exitoso	Santiago, Chile	Escritorio
08/12/2024 - 10:05	Inicio de sesión exitoso	Santiago, Chile	Escritorio
05/12/2024 - 15:30	Solicitud de uniforme enviada	Santiago, Chile	Navegador

A note at the bottom of this section states: 'Este historial es solo informativo y no editable. Si detectas actividad sospechosa, contacta inmediatamente al administrador del sistema.'

- Acciones de la cuenta:** Contains a warning about sharing the device with others and a 'Cerrar sesión' (Logout) button.

**Figura 7.9: GUI de información de cuenta funcionario**

Fuente: Elaboración Propia en Figma



## CONCLUSIONES

El desarrollo del proyecto permitió comprobar que la reingeniería de un sistema monolítico, carente de estructura y buenas prácticas, puede abordarse de manera efectiva mediante una arquitectura moderna basada en servicios desacoplados, contenedores y despliegue en la nube. Los resultados obtenidos demuestran que es técnicamente viable migrar desde una solución obsoleta hacia una plataforma más mantenible, escalable y alineada con estándares actuales de la industria del software.

La posibilidad de mejorar la calidad estructural del sistema mediante la adopción de una arquitectura cloud-native y una pila tecnológica moderna, fue validada a través de la implementación funcional del módulo de funcionarios. Dicha implementación evidenció mejoras concretas en la separación de responsabilidades, integridad de los datos y automatización del ciclo de vida del software, además de lograr implementar un módulo completo en un corto periodo de tiempo.

En cuanto a los alcances del trabajo, se logró diseñar una arquitectura completa y desplegable, acompañada de un modelo de datos normalizado, flujos de integración y despliegue continuo, y una organización del desarrollo basada en buenas prácticas de control de versiones y gestión de tareas.

Desde el punto de vista disciplinar, el proyecto aporta un caso práctico de reingeniería de software aplicada, integrando arquitectura de microservicios, DevOps y computación en la nube, lo que resulta relevante tanto para contextos académicos como profesionales. La experiencia obtenida refuerza la importancia de abordar los problemas de sistemas heredados desde una perspectiva arquitectónica y no únicamente tecnológica.



## BIBLIOGRAFÍA

- Benbya, H., Pachidi, S., & Järvenpää, S. (2021). Special Issue Editorial: Artificial Intelligence in Organizations: Implications for Information Systems Research. *Journal of the Association for Information Systems*, 22(2). <https://doi.org/10.17705/1jais.00662>
- Burns, B., Beda, J., & Hightower, K. (2019). *Kubernetes: Up and Running* (2.<sup>a</sup> ed.). O'Reilly.
- Gurbaxani, V., & Whang, S. (1991). The impact of information systems on organizations and markets. *Communications of the ACM*, 59-73. <https://doi.org/10.1145/99977.99990>
- Laudon, K. C., & Laudon, J. P. (2012). *Sistemas de Información Gerencial* (12.<sup>a</sup> ed.). Pearson.
- Lim, F. J., & Sian, T. B. (2025). Overview of Software Re-Engineering Concepts, Models and Approaches. *International Journal On Informatics Visualization (JOIV)*. <https://www.joiv.org/index.php/joiv/article/view/3034/0>
- Newman, S. (2015). *Building Microservices*. O'Reilly.
- Pressman, R. S. (2020). *Software engineering. A practitioner's approach* (9.<sup>a</sup> ed.). McGraw Hill.
- Rashid, N., Salam, M., Sani, R. K. S., & Alam, F. (2013). Analysis of Risks in Re-Engineering Software Systems. *International Journal of Computer Ap-*



*plications*, 5-8. <https://ijcaonline.org/archives/volume73/number11/12783-9851/>

Richards, M., & Ford, N. (2020). *Fundamentals of Software Architecture: An Engineering Approach*. O'Reilly.

Ruhl, M., & Gunn, M. (1991, septiembre). *Software Reengineering: A Case Study and Lessons Learned* (inf. téc.). National Institute of Standards y Technology (NIST). Special Publication (NIST SP). <https://www.nist.gov/publications/software-reengineering-case-study-and-lessons-learned>

Sommerville, I. (2016). *Software Engineering* (10.<sup>a</sup> ed.). Pearson.



## GLOSARIO DEL NEGOCIO

**Corte** Proceso interno de confección de uniformes de la entidad que provee los uniformes corporativos (proveedor).

**Curva de talla** Distribución de tallas por cada prenda.

**Envío a Corte** Solicitud que realiza el administrador de la institución que necesita uniformes al proveedor para indicar cuántos uniformes deben confeccionarse, basándose en la cantidad de funcionarios registrados con sus tallas ingresadas durante la temporada. En una misma temporada pueden generarse varias solicitudes. Dependiendo de la cantidad de funcionarios con tallas ingresadas, el administrador puede decidir cuándo enviar a corte y a cuántos funcionarios enviar.

**Guía de despacho** Documento tributario electrónico destinado a acreditar el traslado de los uniformes desde la fábrica del proveedor a la respectiva sucursal definida.

**Segmento** Conjunto de prendas perteneciente a un área de desempeño en la empresa, por ejemplo vigilante, secretaria, etc. Masculinos y femeninos forman parte de segmentos distintos. Generalmente un proveedor se encarga de la confección de un solo segmento.

**Temporada** Período de tiempo definido por el administrador para organizar la solicitud y entrega de uniformes, por ejemplo una temporada de invierno y otra de verano, cada una con sus propios plazos y requerimientos.



## GLOSARIO TÉCNICO

**Autenticación** Proceso mediante el cual un sistema verifica la identidad de un usuario antes de permitir su acceso.

**Autorización** Proceso que determina los permisos y privilegios que tiene un usuario dentro del sistema.

**Back-end** Conjunto de servicios, lógica de negocio y operaciones del sistema ejecutadas en el servidor.

**Base de datos relacional** Modelo de almacenamiento que organiza información en tablas relacionadas mediante claves.

**Big Ball of Mud** Antipatrón que describe sistemas sin arquitectura clara, con alto acoplamiento y baja mantenibilidad.

**Cloud Computing** Modelo de provisión de servicios computacionales a través de Internet bajo demanda.

**Contenedor** Unidad portátil que empaqueta una aplicación junto con sus dependencias para ejecutarse de forma consistente en distintos entornos.

**Desacoplamiento** Principio de diseño que reduce dependencias entre componentes para facilitar su mantenimiento y evolución.

**Despliegue** Proceso de publicar una aplicación o servicio en un entorno de ejecución.

**DevOps** Conjunto de prácticas, principios y herramientas que integran el desarrollo de software y las operaciones de TI, con el objetivo de automatizar y optimizar los procesos de desarrollo, integración, despliegue y operación, promoviendo la colaboración continua y la entrega frecuente de software de calidad.

**Docker** Plataforma utilizada para crear y ejecutar contenedores de software.

**Escalabilidad** Capacidad del sistema para aumentar su rendimiento o recursos



según la demanda.

**Front-end** Interfaz gráfica y capa de presentación con la que interactúa directamente el usuario.

**Hosting** Servicio que proporciona infraestructura y recursos computacionales para alojar, ejecutar y mantener aplicaciones, sitios web o servicios digitales, permitiendo su acceso a través de Internet.

**Infraestructura como Código** Práctica que permite definir infraestructura mediante archivos de configuración versionables.

**Microservicio** Componente autónomo que implementa una funcionalidad específica y puede escalarse, desplegarse y actualizarse de forma independiente.

**Monolito** Arquitectura en la que toda la funcionalidad del sistema está contenida en un único bloque de software.

**Orquestación** Sistema que automatiza la administración, escalamiento y operación de contenedores.

**Pila Tecnológica** Una pila tecnológica o *tech stack* es el conjunto de herramientas de software utilizadas para desarrollar un proyecto..

**Pipeline** Secuencia automatizada de pasos que ejecutan compilación, pruebas, análisis y despliegue de software.

**Repositorio** Lugar donde se almacena y versiona el código fuente, documentación y archivos de un proyecto.

**Seguridad informática** Conjunto de prácticas destinadas a proteger datos, sistemas y comunicaciones frente a accesos no autorizados.

**Token** Credencial digital utilizada para autenticar o autorizar a un usuario.

**Workflow** Secuencia de actividades que conforman un proceso funcional.



## SIGLAS

**API** Application Programming Interface — Interfaz de Programación de Aplicaciones.

**CD** Continuous Deployment — Despliegue Continuo.

**CI** Continuous Integration — Integración Continua.

**DevOps** Development and Operations — Desarrollo y Operaciones.

**FTP** File Transfer Protocol — Protocolo de Transferencia de Archivos.

**GCP** Google Cloud Platform — Plataforma de Nube de Google.

**IaC** Infrastructure as Code — Infraestructura como Código.

**JSON** JavaScript Object Notation — Notación de Objetos de JavaScript.

**K8s** Kubernetes — Orquestador de Contenedores.

**UI** User Interface — Interfaz de Usuario.

**UX** User Experience — Experiencia de Usuario.