



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



TFG del Grado en Ingeniería
Informática

GCA

Agente para la Gestión
Centralizada de la
Configuración de Red en las
aulas



Presentado por Rafael Martín Guerrero
en Universidad de Burgos — 8 de junio
de 2025

Tutor: Pedro Renedo Fernández

Índice general

Índice general	i
Índice de figuras	iii
Índice de tablas	vi
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	1
A.3. Estudio de viabilidad	7
Apéndice B Especificación de Requisitos	17
B.1. Introducción	17
B.2. Objetivos generales	17
B.3. Catálogo de requisitos	17
B.4. Especificación de requisitos	19
Apéndice C Especificación de diseño	39
C.1. Introducción	39
C.2. Diseño de datos	39
C.3. Diseño arquitectónico	42
C.4. Diseño procedimental	47
C.5. Diseño de interfaces	51
Apéndice D Documentación técnica de programación	59
D.1. Introducción	59
D.2. Estructura de directorios	59

<i>Índice general</i>	II
D.3. Manual del programador	60
D.4. Compilación, instalación y ejecución del proyecto	60
D.5. Pruebas del sistema	69
Apéndice E Documentación de usuario	75
E.1. Introducción	75
E.2. Requisitos de usuarios	75
E.3. Instalación	76
E.4. Manual del usuario	80
Apéndice F Anexo de sostenibilización curricular	101
F.1. Introducción	101
Bibliografía	102

Índice de figuras

A.1. Clasificación de tareas por categoría.	6
A.2. Tareas y story points por sprint.	7
B.1. Diagrama de casos de uso para la plataforma web.	35
B.2. Diagrama de casos de uso para el aplicativo de configuraciones.	36
B.3. Diagrama de caso de uso de registro de un dispositivo.	36
B.4. Diagrama de caso de uso para la aplicación de configuración.	37
B.5. Diagrama de caso de uso para listar dispositivos.	38
C.1. Diagrama entidad relación	40
C.2. Diagrama exportado en IntelliJ	41
C.3. Diagrama de paquetes del backend	43
C.4. Diagrama de clases del backend	45
C.5. Diagrama de componentes Angular	46
C.6. Paquetes de Python	47
C.7. Diagrama de secuencia del inicio de sesión.	48
C.8. Diagrama de secuencia para la obtención de plantillas.	49
C.9. Diagrama de secuencia del algoritmo principal.	50
C.10. Prototipo inicio de sesión.	51
C.11. Prototipo listar grupos y dispositivos.	52
C.12. Prototipo acciones grupos.	53
C.13. Prototipo modificar dispositivo.	54
C.14. Prototipo crear comando.	55
C.15. Prototipo crear plantilla.	56
C.16. Prototipo asignación de comandos en plantillas.	57
C.17. Prototipo asignación de plantillas.	58
D.1. Contenedores de Docker.	62

D.2. Importar proyecto con Gradle.	63
D.3. Configuración del SDK de proyecto.	64
D.4. Ejecución del backend.	65
D.5. Ejecución del backend desde IntelliJ IDEA.	65
D.6. Versiones del proyecto frontend.	66
D.7. Script de Inno Setup.	68
D.8. Ejecución individual desde el IDE.	69
D.9. Ejecución de tests desde Gradle.	70
D.10.Resultado de pruebas.	70
D.11.Informe de pruebas.	71
D.12.Ejemplo de detalle de cobertura de una clase.	71
D.13.Ejemplo de detalle de cobertura de un método.	72
D.14.Análisis de SonarQube.	72
D.15.Plugin de SonarQube para IntelliJ IDEA.	73
D.16.Análisis de SonarQube desde el IDE.	73
D.17.GitHub Check.	73
D.18.Detalles del GitHub Check.	74
E.1. Instalador aplicación cliente	77
E.2. Ventana para elegir la ruta de instalación	78
E.3. Ventana resumen de la configuración seleccionada	78
E.4. Finalización de la instalación	79
E.5. Exclusión de la aplicación GCA	80
E.6. Página de Inicio de Sesión de la plataforma web	81
E.7. Página principal de la plataforma web	81
E.8. Menú de gestión de grupos	82
E.9. Asignación de plantilla a un grupo	83
E.10.Desasignación de plantilla a un grupo	83
E.11.Tabla de dispositivos	84
E.12.Icono para eliminar un dispositivo	84
E.13.Icono para editar un dispositivo	84
E.14.Icono para asignar una plantilla a un dispositivo	84
E.15.Formulario de edición de dispositivo	85
E.16.Asignación de plantilla a un dispositivo	86
E.17.Desasignación de plantilla a un dispositivo	87
E.18.Gestion de comandos	88
E.19.Formulario para un comando	89
E.20.Gestion de plantillas	90
E.21.Creación de una plantilla	91
E.22.Icono para asignar comandos a una plantilla	91
E.23.Asignar comandos a una plantilla	92

E.24. Comandos asignado a una plantilla	93
E.25. Inicio de la aplicación cliente	94
E.26. Mensaje de inicio de sesión	94
E.27. Panel de usuario	95
E.28. Mensaje de alta del equipo	96
E.29. Dispositivo dado de alta desde el cliente	96
E.30. Asignación de una plantilla al dispositivo registrado	97
E.31. Registro de un comando ejecutado satisfactoriamente	97
E.32. Firewall desactivado	98
E.33. Mensaje de baja del equipo	99

Índice de tablas

A.1. <i>Total de Issues por categorías.</i>	5
A.2. <i>Total de Issues y Story Points por sprint.</i>	6
A.3. <i>Nómina del empleado.</i>	8
A.4. <i>Coste para la empresa.</i>	8
A.5. <i>Costes del hardware.</i>	9
A.6. <i>Costes del software.</i>	9
A.7. <i>Costes varios.</i>	9
A.8. <i>Costes totales del proyecto.</i>	10
A.9. <i>Monetización del proyecto.</i>	10
A.10. <i>Suscripciones necesarias para cubrir gastos.</i>	10
A.11. <i>Dependencias del proyecto Python</i>	12
A.12. <i>Dependencias principales del proyecto Java</i>	13
A.13. <i>Dependencias principales del proyecto Angular</i>	14
A.14. <i>Resumen de los puntos clave de la licencia Apache License 2.0</i>	15
A.15. <i>Resumen de las licencias del proyecto.</i>	16
B.1. <i>CU-1 Registrar equipos.</i>	20
B.2. <i>CU-2 Editar equipos.</i>	21
B.3. <i>CU-3 Eliminar equipos.</i>	22
B.4. <i>CU-4 Ver y listar equipos.</i>	23
B.5. <i>CU-5 Añadir plantillas.</i>	24
B.6. <i>CU-6 Modificar plantillas.</i>	25
B.7. <i>CU-7 Eliminar plantillas.</i>	26
B.8. <i>CU-8 Ver y listar equipos.</i>	27
B.9. <i>CU-9 Añadir usuarios.</i>	28
B.10. <i>CU-10 Modificar usuarios.</i>	29
B.11. <i>CU-11 Eliminar usuarios.</i>	30
B.12. <i>CU-12 Ver y listar usuarios.</i>	31

B.13.CU-13 Registrar dispositivo.	32
B.14.CU-14 Aplicar configuración.	33
B.15.CU-15 Eliminar configuración.	34

Apéndice A

Plan de Proyecto Software

A.1. Introducción

La fase de planificación constituye un elemento clave e imprescindible en cualquier proyecto. Mediante una planificación adecuada, se incrementan notablemente las posibilidades de éxito, ya que permite anticipar problemas, distribuir adecuadamente los recursos y organizar las tareas de una forma eficiente.

Por este motivo, en este anexo se detallan aquellos aspectos necesarios para el desarrollo del proyecto.

Dividiremos el plan en dos apartados:

- **La planificación temporal:** donde se establece el modo de organizar las actividades a realizar, una estimación del tiempo que se va a dedicar a cada una de ellas y las fechas de inicio y fin del proyecto.
- **El estudio de viabilidad:** en el que se analiza si el proyecto es factible desde el punto de vista financiero y legal, atendiendo a los elementos necesarios para su realización.

A.2. Planificación temporal

El proyecto se ha realizado bajo un marco ágil de gestión de proyectos, basado en Scrum, aunque ha sido adaptado a las características específicas del contexto.

Debido a la inexistencia de un equipo, la falta de un entorno estructurado (roles definidos, reuniones diarias, retrospectivas y *demos*), esta planificación no se ha podido aplicar en su totalidad; aún así, se han mantenido varios de sus fundamentos claves, aunque con ciertos matices:

- **Iteraciones cortas:** organizadas en sprints de 2 semanas, durante los cuales se han desarrollado nuevas funcionalidades de manera incremental.
- **Equipos autoorganizados:** aunque no exista un equipo como tal, el desarrollo se ha producido de una manera autónoma.
- **Reuniones semanales:** se programó una reunión formal todos los lunes para revisar el estado del proyecto y resolver posibles dudas surgidas durante la semana anterior.
- **Backlog del producto:** en el que se van añadiendo de forma continua todas las tareas pendientes de realizar (ya sean de desarrollo, documentación...) y se mantiene un listado de tareas ordenadas por prioridad.
- **Planificación de los sprints:** en la que se estudian las tareas que se realizarán para entregar en el siguiente incremento, atendiendo a características como prioridad y complejidad.

A cada tarea se le asigna un valor estimado de esfuerzo, denominado *story point*, los cuales están basados en la secuencia de Fibonacci (1, 2, 3, 5, 8...).

Un *story point* representa una combinación de dificultad y tiempo estimado; en este sentido, decimos que una tarea de 3 puntos comparada con una de 5 se considera más sencilla, y, por tanto, se realizará en menor tiempo. Cabe mencionar que este sistema permite hacer estimaciones de tiempo para las tareas, pero no establece una equivalencia exacta con días de trabajo.

Esta forma de trabajar no es la más exacta, pero permite flexibilizar el trabajo, lo que resulta útil en este contexto específico en el que la capacidad disponible de trabajo puede variar en el tiempo: incluso en entornos ideales, los equipos pueden verse afectados por situaciones como bajas, vacaciones... Por eso, la planificación debe poder ser adaptable.

Sprint 0 (25/02/2025 - 05/03/2025) Este primer sprint sirvió como punto de partida: se estableció una reunión inicial con el tutor, **Pedro**

Renedo, en la que abordamos y concretamos el tema central del proyecto que se pretende realizar.

Se creó el repositorio inicial en GitHub y estudiamos las posibles herramientas y lenguajes de programación que se utilizarían en el desarrollo.

Además, se estimaron **7 *story points***, que correspondieron aproximadamente con la semana que duró el sprint. Esta duración fue una excepción, ya que se trató del primero; después se optó por alargar este tiempo a dos semanas, ya que observamos que un sprint de solo siete días era demasiado corto cuando se introducían tareas de desarrollo.

Sprint 1 (05/03/2025 - 20/03/2025)

Durante este sprint se profundizó en el análisis del proyecto, tratando de responder a preguntas clave como: **¿Qué se necesita realizar? ¿Cómo ha de hacerse? ¿Con base a qué surge esta necesidad? ...**

Además, se estudió el tipo de aplicación que se iba a necesitar en los dispositivos, se modeló una primera versión de base de datos y se creó la estructura inicial de la aplicación backend en Java. En este punto surgió un problema técnico: los ordenadores del centro son clonados, por lo que tienen el mismo GUID y no sirve como identificador único, por lo que hay que se hizo necesario seguir investigando para encontrar algún método alternativo.

Se estimaron **19 *story points*** de trabajo para los 15 días de trabajo.

Sprint 2 (20/03/2025 - 02/04/2025)

Continuamos con la línea de definición de requisitos y objetivos funcionales, se desarrollaron los CU principales, y además, se diseñó un método de autenticación para los equipos, el cual combina el GUID con el nombre de los equipos, aprovechando que estos sí son únicos para cada uno.

Por otro lado, se creó la estructura de la aplicación frontal.

Se estimaron **15.5 *story points*** para los 15 días de trabajo; en este caso, se ajustó a 1 *story point* por día prácticamente.

Sprint 3 (02/04/2025 - 16/04/2025)

En esta etapa comenzaron las tareas de desarrollo del backend. Para ello, se creó una base de datos inicial con MariaDB después de refinar el diagrama E-R actual, y se implementaron las bases de nuestra API con Spring Boot, configurando los primeros end points e implementando medidas de seguridad en las llamadas mediante login y generación de tokens.

Se estimaron **21 *story points*** en este sprint, ya que la disponibilidad prevista era superior a la de otros sprints.

Sprint 4 (16/04/2025 - 30/04/2025)

Avanzamos creando nuevos endpoints básicos, sentando las bases que utilizaremos en la aplicación web para gestionar los diferentes grupos y dispositivos. En paralelo, trabajamos en el desarrollo de nueva documentación, definiendo los requisitos no funcionales y las técnicas y herramientas que se pretenden utilizar, así como el estudio de viabilidad del proyecto.

Es estimaron **17 *story points*** para este sprint, ya que se consideró que la disponibilidad de tiempo iba a ser menor que en el anterior.

Sprint 5 (30/04/2025 - 14/05/2025)

Continuamos con el desarrollo de la API, pero en este caso, también diseñamos una posible aproximación a la interfaz de usuario, generamos diagramas de los casos de uso más importantes, y comenzamos a implementar la parte de la aplicación frontal utilizando Angular.

En la reunión semanal con el tutor, se aclararon puntos relevantes relativos a la aplicación: uno de ellos fue la posibilidad de generar comandos con valores dinámicos, de forma que, en el futuro, para cada plantilla, se puedan asignar valores diferentes en función de la plantilla utilizada.

Otro punto relevante que surgió fue la necesidad de investigar qué hacer con los equipos a los que se les deshabilita el acceso a la red, ya que no se podrían gestionar de manera remota. Una posibilidad que se barajó fue realizar una copia del estado del equipo antes de aplicar la plantilla, para que cuando se reinicie, se restaure el estado anterior, de manera que el equipo se mantenga tal y como estaba antes de aplicar la plantilla.

Se estimaron **20 *story points*** para este sprint, ya que se consideró que la disponibilidad iba a ser mayor y en este punto del proyecto, las bases ya estaban más asentadas, el trabajo se hacía más fluido y había menos dudas.

Sprint 6 (14/05/2025 - 28/05/2025)

Este Sprint es uno de los más críticos: se actualizó la aplicación cliente, permitiendo obtener y aplicar las plantillas de manera remota. De forma paralela, completamos la interfaz web, permitiendo gestionar comandos y plantillas de forma visual. En cuanto a test se refiere, mejoramos la cobertura del código, añadiendo pruebas unitarias y de integración.

Se estimaron **20 *story points*** para este sprint porque disponíamos de más tiempo para la realización de las tareas.

Sprint 7 (28/05/2025 - 04/06/2025)

El penúltimo sprint antes de la entrega final consistió en rellenar la documentación del proyecto, así como añadir logs en el backend para mejorar la trazabilidad de las acciones realizadas. Tras realizar una demostración del proyecto al tutor, se crearon nuevas tareas para mejorar la usabilidad de la aplicación cliente.

La duración de este sprint y el siguiente se redujeron a una semana, ya que al tratarse de la fase final, es más sencillo contar con períodos más cortos de tiempo y poder así ajustar si surgen problemas inesperados. Se estimaron **10 *story points*** para este sprint.

Sprint 8 (04/06/2025 - 11/06/2025)

En este último sprint se realizaron las últimas correcciones y ajustes necesarios para la entrega final del proyecto. Se añadió una mejora en la interfaz de la aplicación cliente, mostrando la información del dispositivo registrado: huella digital, grupo al que pertenece y plantilla aplicada. Se realizaron pruebas finales de la aplicación, tanto en el cliente como en el servidor, y se generó el instalador de la aplicación cliente para su distribución.

Se estimaron **8 *story points*** para este sprint.

Resumen

Por último, se muestra el resumen de los sprints realizados, con el total de issues y story points asignados a cada uno así como el total de issues por categorías.

Categoría	Total de Issues
documentation	19
back end	11
enhancement	9
design	6
front end	6
research	4
meeting	3
db	3
bug	1

Tabla A.1: *Total de Issues por categorías.*

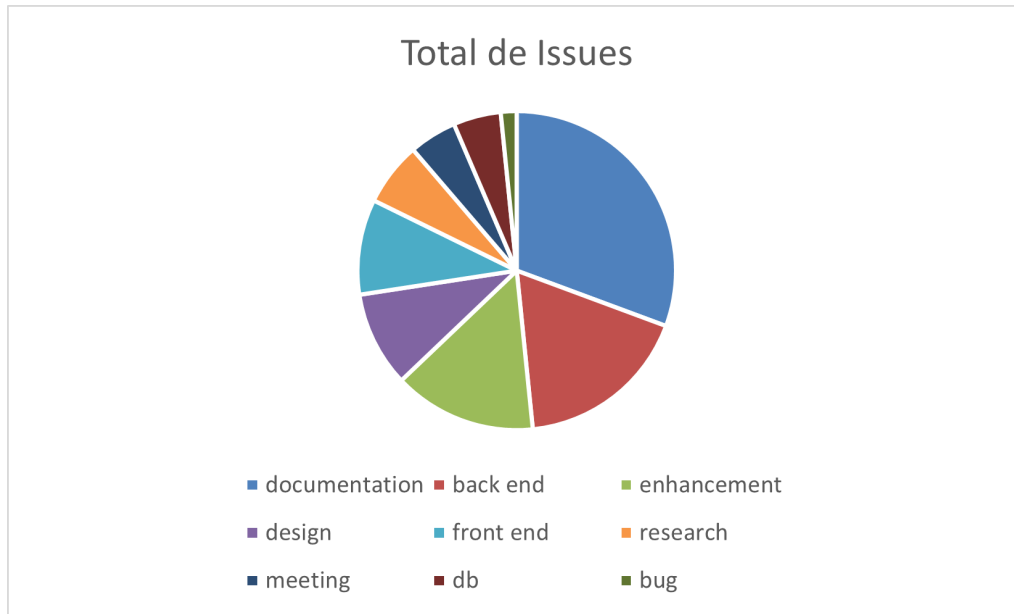


Figura A.1: Clasificación de tareas por categoría.

Sprint	Total de Issues	Story Points
Sprint 0	4	7
Sprint 1	5	19
Sprint 2	5	15
Sprint 3	4	21
Sprint 4	5	17
Sprint 5	7	20
Sprint 6	6	20
Sprint 7	6	10
Sprint 8	7	8

Tabla A.2: Total de Issues y Story Points por sprint.

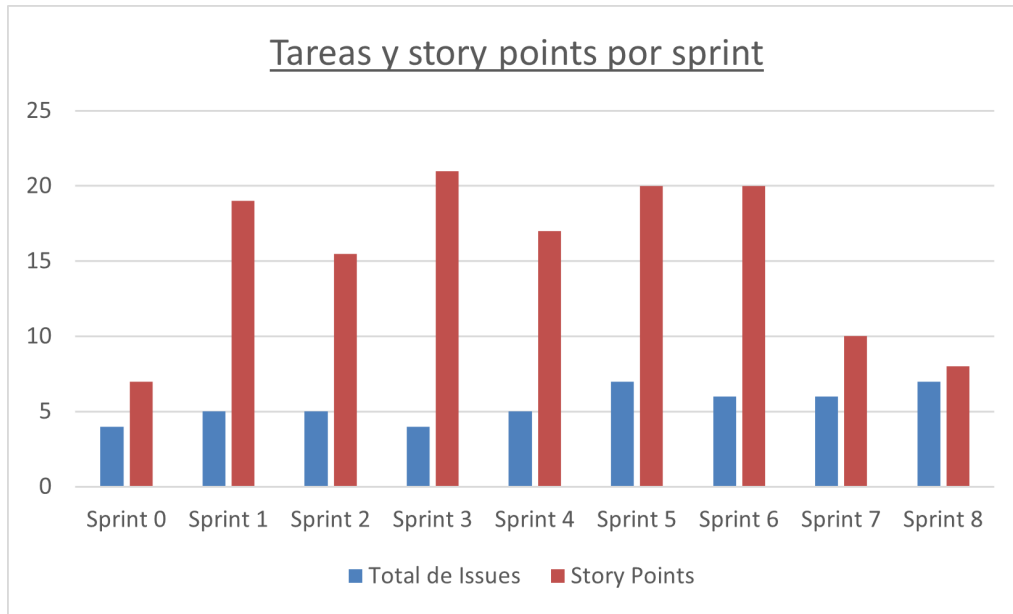


Figura A.2: Tareas y story points por sprint.

A.3. Estudio de viabilidad

Viabilidad económica

En este apartado analizaremos los costes y beneficios de haber implementado el proyecto con una intención real de llevarlo a cabo.

Costes

Costes de personal: todo el desarrollo ha sido realizado por un único desarrollador en un tiempo total de cinco meses comprendidos entre febrero y junio. El sueldo base promedio para un puesto de desarrollador “Full Stack Developer” en España es de 30 mil €/año brutos, de acuerdo con la información publicada en Glassdor[5]

Calculamos el salario en base a una persona de 30 años, soltera y sin hijos[8]

Concepto	Coste
Salario neto	1929,8€
Retención IRPF	411,5€
Cuota seguridad social	148,75€
Salario mensual bruto	2500€
Total 5 meses	12500€

Tabla A.3: Nómina del empleado.

Concepto	Coste
Salario bruto	2500€
Contingencias Comunes	590€
Accidentes de Trabajo y Enfermedades Profesionales (AT y EP)	22,5€
Desempleo	137,5€
Formación Profesional	15€
Fondo de garantía salarial (FOGASA)	5€
Total 5 meses	16350€

Tabla A.4: Coste para la empresa.

Costes de hardware

Para el hardware utilizado, no se emplearon demasiadas herramientas más allá de un ordenador para la realización del proyecto, que se hará a medida, a priori, con un dispositivo de 16GB de RAM, SSD NVMe, procesador actual y una tarjeta gráfica estándar será suficiente. En el precio viene ya incluido una licencia de Win 11.

$$\text{Coste amortizado} = \frac{\text{Coste total del equipo}}{\text{Vida útil (en meses)}} \times \text{Meses del proyecto}$$

$$\text{Coste amortizado} = \frac{1200 \text{ £}}{48 \text{ meses}} \times 5 \text{ meses} = 125 \text{ £}$$

Concepto	Coste
Ordenador a medida	1200€
Amortización durante 5 meses	125€
Total amortizado	125€

Tabla A.5: *Costes del hardware.*

Costes de software

En este apartado tendremos en cuenta las licencias necesarias para el desarrollo del proyecto: para los IDE's existen dos opciones, o bien utilizar las licencias gratuitas "Community Edition", o las de pago[6] Escogemos esta última porque cuenta con características que ayudan a desarrollar rápidamente nuestro proyecto.

Concepto	Coste
All Products Pack (mes)	94,26€
Visual Studio Code	0€
Total 5 meses	471,3€

Tabla A.6: *Costes del software.*

Costes variados

En este apartado incluimos otros costes a tener en cuenta:

Concepto	Coste
Dominio web	20€
Hosting / contendor	50€
Internet	100€
Total	170€

Tabla A.7: *Costes varios.*

Costes totales

Se muestra el sumatorio de todos los costes:

Concepto	Coste
Personal	16350€
Hardware	125€
Software	471,3€
Variados	170€
Total	17116,3€

Tabla A.8: *Costes totales del proyecto.*

Beneficios

Se trata de un proyecto con fines estrictamente académicos, pues la idea es que pueda llegar a ser desarrollado para la Universidad de Burgos.

No obstante, se podría considerar establecer algún tipo de suscripción (ya sea mensual o anual) y que pueda ser utilizado por otras entidades educativas o, incluso, por empresas privadas.

Tipo	Objetivo	Precio mensual	Precio anual
Básica	Centros educativos pequeños	49,90€	499€
Profesional	Instituciones grandes	89,90€	899€
Empresarial	Empresas privadas	149,90€	1149€
Premium	Incluye personalización total	250€	2500€

Tabla A.9: *Monetización del proyecto.*

Para calcular el número de suscripciones que necesitaríamos para cubrir gastos, utilizamos la siguiente fórmula:

$$\text{Suscripciones necesarias} = \frac{17,116,3 \text{ €}}{\text{Precio de la suscripción anual}}$$

Tipo	Precio anual	Suscripciones necesarias
Básica	499€	35
Profesional	899€	19
Empresarial	1149€	15
Premium	2500€	7

Tabla A.10: *Suscripciones necesarias para cubrir gastos.*

De esta manera, calculamos el número de suscripciones necesarias (si solo se tuviesen de un mismo tipo) para llegar a cubrir los gastos totales del proyecto. También se podría utilizar un modelo mixto e incluir suscripciones de cada tipo.

Viabilidad legal

Abordamos aquellos temas relacionados con el contexto legal del proyecto, así como lo relativo a las licencias.

La implementación del proyecto ha sido propia, a excepción de las librerías y dependencias de terceros.

En este punto, diferenciaremos las dependencias utilizadas en el cliente (Python) de las utilizadas en la API (Java) y la interfaz de gestión de usuario (Angular)

Python

Dependencia	Versión	Descripción	Licencia
altgraph	0.17.4	Análisis y representación de grafos de objetos de Python.	MIT
certifi	2025.1.31	Certificados raíz para validar la seguridad SSL.	MPL-2.0
charset-normalizer	3.4.1	Detección de codificación de texto similar a chardet.	MIT
idna	3.10	Soporte de nombres de dominio internacionalizados (IDNA).	BSD
packaging	24.2	Utilidades para manejo de versiones y requerimientos.	Apache-2.0
pefile	2023.2.7	Lectura y modificación de archivos PE (Portable Executable).	MIT
pillow	11.1.0	Biblioteca de procesamiento de imágenes (fork de PIL).	HPND
pip	25.0.1	Instalador oficial de paquetes Python.	MIT
pyinstaller	6.12.0	Empaquetado de apps Python en ejecutables.	GPL-2.0
pyinstaller-hooks-contrib	2025.2	Hooks adicionales para PyInstaller.	GPL-2.0

Tabla A.11 – *continuación*

Dependencia	Versión	Descripción	Licencia
pystray	0.19.5	Gestión de iconos en la bandeja del sistema.	MIT
pywin32-ctypes	0.2.3	Acceso a funcionalidades Win32 usando ctypes.	MIT
requests	2.32.3	Biblioteca HTTP simple y elegante para Python.	Apache-2.0
setuptools	78.1.0	Herramienta de empaquetado y distribución.	MIT
six	1.17.0	Compatibilidad entre Python 2 y 3.	MIT
urllib3	2.3.0	Cliente HTTP con soporte para conexión persistente.	MIT

Tabla A.11: Dependencias del proyecto Python

Java

Dependencia	Versión	Descripción	Licencia
spring-boot-starter	3.4.3	Starter básico para aplicaciones Spring Boot.	Apache-2.0
spring-boot-starter-web	3.4.3	Starter para crear aplicaciones web con Spring MVC.	Apache-2.0
spring-boot-starter-actuator	3.4.3	Starter para monitorización y gestión de aplicaciones.	Apache-2.0
spring-boot-starter-data-jpa	3.4.3	Starter para usar Spring Data JPA con Hibernate.	Apache-2.0
spring-boot-starter-security	3.4.3	Starter para habilitar seguridad con Spring Security.	Apache-2.0
spring-boot-starter-validation	3.4.3	Starter para validación de datos usando Bean Validation.	Apache-2.0

Tabla A.12 – *continuación*

Dependencia	Versión	Descripción	Licencia
mariadb-java-client	3.4.1	Driver JDBC para base de datos MariaDB.	LGPL-2.1
spring-boot-starter-test	3.4.3	Starter para pruebas unitarias y de integración en Spring.	Apache-2.0
spring-security-test	6.4.3	Utilidades para pruebas de seguridad con Spring Security.	Apache-2.0
jjwt-api	0.12.3	API para creación y verificación de JSON Web Tokens (JWT).	Apache-2.0
jacoco	0.8.12	Agente para la recolección de cobertura de código en tiempo de ejecución.	EPL-2.0
junit-jupiter-api	5.11.4	API principal de JUnit 5 para pruebas unitarias.	EPL-2.0
junit-jupiter-engine	5.11.4	Motor de ejecución de pruebas para JUnit 5.	EPL-2.0

Tabla A.12: Dependencias principales del proyecto Java

Angular

Dependencia	Versión	Descripción	Licencia
@angular/common	19.2.0	Funcionalidades comunes y directivas básicas de Angular.	MIT
@angular/compiler	19.2.0	Compilador para templates de Angular.	MIT
@angular/core	19.2.0	Núcleo del framework Angular.	MIT
@angular/forms	19.2.0	Módulo para crear formularios reactivos y por plantilla.	MIT

Tabla A.13 – *continuación*

Dependencia	Versión	Descripción	Licencia
@angular/platform-browser	19.2.0	Soporte para la ejecución de Angular en navegadores.	MIT
@angular/platform-browser-dynamic	19.2.0	Renderizado dinámico de Angular en el navegador.	MIT
@angular/router	19.2.0	Sistema de enrutamiento y navegación para Angular.	MIT
rxjs	7.8.0	Biblioteca para programación reactiva con observables.	Apache-2.0
tslib	2.3.0	Helpers de TypeScript para reducir el tamaño del bundle.	0BSD
zone.js	0.15.0	Implementación de zonas para la gestión del contexto de ejecución.	MIT
@fortawesome	6.7.2	Biblioteca de iconos escalables basada en Font Awesome.	MIT
tailwindcss	4.1.4	Framework de utilidad para crear interfaces modernas y responsivas.	MIT
daisyui	5.0.27	Plugin de componentes UI para Tailwind CSS.	MIT

Tabla A.13: Dependencias principales del proyecto Angular

Una vez estudiadas las dependencias y tipos de licencias de nuestro proyecto, observamos que las usadas son:

- MIT
- Apache-2.0
- BSD

- HPND
- 0BSD

Se ha optado por utilizar una licencia menos permisiva para no tener problemas con el uso de otros tipos de licencias. En este caso elegimos Apache License 2.0[1]:

Característica	Descripción
Libertad de uso	Permite usar el software para cualquier propósito, incluyendo comercial.
Distribución	Puedes redistribuir el código original o modificado, incluso bajo una licencia diferente compatible.
Modificación	Puedes modificar el código fuente, pero debes indicar los cambios realizados.
Licencia permisiva	No impone que el trabajo derivado use la misma licencia (no es copyleft).
Aviso de licencia	Se debe incluir un aviso de derechos de autor y la licencia en todas las distribuciones.
Compatibilidad	Compatible con muchas otras licencias, incluyendo GPL v3.
Patentes	Concede una licencia explícita de uso de patentes del contribuyente; si alguien inicia una demanda de patentes, pierde la licencia.
Marcas registradas	No concede ningún derecho sobre marcas registradas.

Tabla A.14: Resumen de los puntos clave de la licencia Apache License 2.0

De esta manera, aumentamos la posibilidad de que otros desarrolladores o empresas contribuyan a nuestro proyecto, pudiendo adoptar el software más fácilmente.

Documentación

Con tal de proteger la autoría del proyecto, que sea fácil de aplicar y compatible con usos académicos y comercial, se ha escogido Creative Commons Attribution 4.0 International[3]

Imágenes y vídeos

Para las imágenes y vídeos, utilizamos la licencia Creative Commons Attribution-NonCommercial 4.0 International[4], ya que las imágenes utilizadas son propias; así, protegemos el contenido pero permitimos su difusión.

Resumen

Tipo de contenido	Licencia	Resumen de derechos y restricciones
Código fuente	Apache License 2.0	Permite uso, modificación, distribución (incluso comercial), siempre que se incluya un aviso de licencia y derechos de autor. No obliga a licenciar el código derivado bajo la misma licencia.
Documentación	CC BY 4.0	Permite compartir y adaptar, incluso con fines comerciales, siempre que se dé atribución adecuada al autor original.
Imágenes y videos	CC BY-NC 4.0	Permite compartir y adaptar, pero no para uso comercial. Se requiere atribución al autor original.

Tabla A.15: Resumen de las licencias del proyecto.

Apéndice B

Especificación de Requisitos

B.1. Introducción

En esta sección se especifican los distintos requisitos del proyecto. Este punto es clave porque define qué debe hacer el sistema incluso antes de comenzar con el desarrollo.

B.2. Objetivos generales

- Crear una plataforma para el control de acceso a los recursos de los dispositivos del centro.
- Contribuir a la automatización de procesos de la Universidad.
- Solucionar las necesidades del personal docente.
- Guardar las distintas configuraciones y permitir una gestión eficiente.

B.3. Catálogo de requisitos

A continuación, se muestra el listado de los requisitos específicos del proyecto.

Requisitos funcionales

- **RF-1 Gestión de la plataforma web de configuraciones:** La aplicación debe ser capaz de gestionar las configuraciones de los distintos equipos.

- **RF-1.1 Registrar equipos:** El usuario debe poder dar de alta los ordenadores mediante un identificador que permita validar la veracidad del equipo.
- **RF-1.2 Editar equipos:** La aplicación podrá permitir modificar datos de los equipos, así como sus configuraciones.
- **RF-1.3 Eliminar equipos:** El usuario debe poder dar de baja equipos que no se deseen seguir gestionando y eliminar la información asociada.
- **RF-1.4 Listar equipos:** La aplicación debe ser capaz de realizar listados de equipos y agruparlos atendiendo a diferentes características para después aplicar las configuraciones pertinentes.
- **RF-1.5 Ver equipos:** La aplicación debe ser capaz de mostrar la configuración que aplica a un determinado equipo.
- **RF-2 Gestión de plantillas de configuraciones:** La aplicación debe ser capaz de ofrecer una gestión eficiente de las plantillas de configuraciones que se aplicarán a los equipos.
 - **RF-2-1 Añadir plantillas:** El usuario debe poder añadir plantillas de configuraciones para los equipos.
 - **RF-2.2 Modificar plantillas:** El usuario debe poder modificar plantillas existentes.
 - **RF-2.3 Eliminar plantillas:** El usuario podrá eliminar plantillas que no vayan a usarse más.
 - **RF-2.4 Listar plantillas:** El usuario podrá consultar las plantillas existentes.
 - **RF-2.5 Ver plantillas:** El usuario podrá consultar toda la información perteneciente a una plantilla.
- **RF-3 Gestión de usuarios:** La aplicación tiene que ser capaz de mantener una gestión de usuarios.
 - **RF-3.1 Añadir usuarios:** La aplicación permitirá dar de alta a nuevos usuarios que gestionen la aplicación.
 - **RF-3.2 Modificar usuarios:** Se podrán modificar atributos de usuarios ya existentes.
 - **RF-3.3 Eliminar usuarios:** La aplicación permitirá dar de baja a un usuario junto con sus datos existentes.
 - **RF-3.4 Listar usuarios:** La aplicación podrá listar los usuarios que existen en el sistema.
 - **RF-3.5 Ver usuarios:** La aplicación debe mostrar toda la información relativa al usuario.

- **RF-4 Aplicativo de configuraciones:** La aplicación tiene que ser capaz de ejecutarse en el sistema operativo y aplicar en los equipos las configuraciones necesarias que correspondan a cada uno.
 - **RF-4.1 Registrar dispositivo:** La aplicación debe realizar el registro del dispositivo antes de poder aplicar las configuraciones. Se debe identificar de manera única a cada equipo.
 - **RF-4.2 Aplicar configuración:** La aplicación tiene que ser capaz de aplicar la configuración registrada para cada dispositivo.
 - **RF-4.3 Eliminar configuración:** La aplicación tiene que ser capaz de revertir los cambios realizados a un dispositivo.

Requisitos no funcionales

Los requisitos no funcionales definen cómo debe comportarse un sistema.

- **RF-1 Rendimiento:** El sistema debe ofrecer unos tiempos de respuesta inferiores a 3 segundos bajo una carga media y ser capaz de escalar sin una degradación significativa del rendimiento.
- **RF-2 Seguridad:** La aplicación debe restringir el acceso a usuarios autorizados y cifrar datos sensibles, cumpliendo con buenas prácticas de seguridad.
- **RF-3 Usabilidad:** La aplicación debe ofrecer una interfaz intuitiva y accesible, garantizando una experiencia consistente y cumpliendo con unos principios básicos de accesibilidad.
- **RF-4 Disponibilidad:** La aplicación debe estar operativa de forma continua, a excepción de situaciones extraordinarias o mantenimientos programados.
- **RF-5 Mantenibilidad:** La aplicación debe seguir unos estándares y buenas prácticas, facilitando su mantenimiento, evolución y comprensión por parte de otros desarrolladores.
- **RF-6 Portabilidad:** La aplicación debe ser portable entre distintas plataformas, utilizando tecnologías que faciliten su migración, como pueden ser contenedores.
- **RF-7 Legalidad:** El sistema debe hacer un uso adecuado de herramientas de terceros y cumplir con las normas del territorio en el que se aloja.

B.4. Especificación de requisitos

Casos de uso

CU-1	Registrar equipos
Versión	1.0
Autor	Rafael Martín Guerrero
Requisitos asociados	RF-1.1
Descripción	Permite al usuario dar de alta un nuevo equipo.
Precondición	<ul style="list-style-type: none"> ▪ La aplicación se encuentra operativa. ▪ La base de datos es correcta. ▪ El usuario está autenticado en el sistema. ▪ El usuario tiene suficientes permisos para realizar la operación.
Acciones	<ol style="list-style-type: none"> 1. Acceder al área de gestión de equipos y presionar en añadir. 2. Rellenar el formulario (huella digital, nombre del equipo. . .). 3. Pulsar en guardar información.
Postcondición	El equipo se encuentra registrado en la BD.
Excepciones	<ul style="list-style-type: none"> ▪ La huella digital o el nombre ya existe. ▪ No se ha introducido algún campo obligatorio (se muestra mensaje).
Importancia	Alta

Tabla B.1: CU-1 Registrar equipos.

CU-2	Editar equipos
Versión	1.0
Autor	Rafael Martín Guerrero
Requisitos asociados	RF-1.2
Descripción	Permite al usuario modificar datos de un equipo.
Precondición	<ul style="list-style-type: none"> ▪ La aplicación se encuentra operativa. ▪ La base de datos es correcta. ▪ El usuario está autenticado en el sistema y el equipo debe existir. ▪ El usuario tiene suficientes permisos para realizar la operación.
Acciones	<ol style="list-style-type: none"> 1. Escoger el equipo que se quiere modificar. 2. Rellenar los campos necesarios (huella digital, nombre del equipo. . .). 3. Pulsar en modificar información.
Postcondición	El equipo se encuentra actualizado en la BD.
Excepciones	<ul style="list-style-type: none"> ▪ La huella digital o el nombre ya existe. ▪ No se ha introducido algún campo obligatorio (se muestra mensaje). ▪ El equipo a editar no existe.
Importancia	Alta

Tabla B.2: CU-2 Editar equipos.

CU-3	Eliminar equipos
Versión	1.0
Autor	Rafael Martín Guerrero
Requisitos asociados	RF-1.3
Descripción	Permite al usuario dar de baja equipo.
Precondición	<ul style="list-style-type: none"> ■ La aplicación se encuentra operativa. ■ La base de datos es correcta. ■ El usuario está autenticado en el sistema y el equipo debe existir. ■ El usuario tiene suficientes permisos para realizar la operación.
Acciones	<ol style="list-style-type: none"> 1. Escoger el equipo que se quiere eliminar. 2. Pulsar en eliminar equipo. 3. Confirmar eliminación.
Postcondición	El equipo se encuentra eliminado de la BD.
Excepciones	<ul style="list-style-type: none"> ■ No se ha podido borrar el registro (se muestra mensaje).
Importancia	Alta

Tabla B.3: CU-3 Eliminar equipos.

CU-4	Ver y listar equipos
Versión	1.0
Autor	Rafael Martín Guerrero
Requisitos asociados	RF-1.4, RF-1.5
Descripción	Permite al usuario obtener listados de equipos con determinados filtros.
Precondición	<ul style="list-style-type: none"> ▪ La aplicación se encuentra operativa. ▪ La base de datos es correcta. ▪ El usuario está autenticado en el sistema. ▪ El usuario tiene suficientes permisos para realizar la operación.
Acciones	<ol style="list-style-type: none"> 1. Acceder al área de gestión de equipos. 2. Escoger los distintos filtros para realizar la búsqueda. 3. Presionar en buscar. 4. Seleccionar un equipo entre los encontrados para ver sus detalles.
Postcondición	Se muestran correctamente los equipos, si se cumplen los criterios de búsqueda.
Excepciones	<ul style="list-style-type: none"> ▪ Se muestra un texto si no se han encontrado equipos para esos criterios.
Importancia	Alta

Tabla B.4: CU-4 Ver y listar equipos.

CU-5	Añadir plantillas
Versión	1.0
Autor	Rafael Martín Guerrero
Requisitos asociados	RF-2.1
Descripción	Permite al usuario crear plantillas de configuración que se aplicarán a los equipos.
Precondición	<ul style="list-style-type: none"> ■ La aplicación se encuentra operativa. ■ La base de datos es correcta. ■ El usuario está autenticado en el sistema. ■ El usuario tiene suficientes permisos para realizar la operación.
Acciones	<ol style="list-style-type: none"> 1. Acceder al área de gestión de plantillas y presionar en añadir. 2. Introducir los datos de la plantilla (nombre, descripción...) y los comandos a aplicar sobre ella. 3. Presionar en guardar.
Postcondición	La plantilla se almacena correctamente en base de datos.
Excepciones	<ul style="list-style-type: none"> ■ Se muestra un mensaje si no se tienen permisos para almacenar la plantilla.
Importancia	Alta

Tabla B.5: CU-5 Añadir plantillas.

CU-6	Modificar plantillas
Versión	1.0
Autor	Rafael Martín Guerrero
Requisitos asociados	RF-2.2
Descripción	Permite al usuario crear plantillas de configuración que se aplicarán a los equipos.
Precondición	<ul style="list-style-type: none"> ■ La aplicación se encuentra operativa. ■ La base de datos es correcta. ■ El usuario está autenticado en el sistema. ■ El usuario tiene suficientes permisos para realizar la operación. ■ La plantilla existe.
Acciones	<ol style="list-style-type: none"> 1. Acceder al área de gestión de plantillas y seleccionar aquella que se quiere modificar. 2. Introducir los datos modificados de la plantilla (nombre, descripción...) y los comandos sobre ella. 3. Presionar en modificar.
Postcondición	La plantilla se modifica correctamente en base de datos.
Excepciones	<ul style="list-style-type: none"> ■ Se muestra un mensaje si no se tienen permisos para modificar la plantilla.
Importancia	Media

Tabla B.6: CU-6 Modificar plantillas.

CU-7	Eliminar plantillas
Versión	1.0
Autor	Rafael Martín Guerrero
Requisitos asociados	RF-2.3
Descripción	Permite al usuario dar de baja plantillas.
Precondición	<ul style="list-style-type: none"> ▪ La aplicación se encuentra operativa. ▪ La base de datos es correcta. ▪ El usuario está autenticado en el sistema. ▪ El usuario tiene suficientes permisos para realizar la operación. ▪ La plantilla existe.
Acciones	<ol style="list-style-type: none"> 1. Acceder al área de gestión de plantillas y seleccionar aquella que se quiere eliminar. 2. Presionar en eliminar. 3. Confirmar eliminación.
Postcondición	<ul style="list-style-type: none"> ▪ La plantilla se elimina del sistema. ▪ No se vuelve a aplicar la plantilla eliminada sobre los equipos asociados.
Excepciones	<ul style="list-style-type: none"> ▪ Se muestra un mensaje si no se tienen permisos para eliminar la plantilla.
Importancia	Media

Tabla B.7: CU-7 Eliminar plantillas.

CU-8	Ver y listar plantillas
Versión	1.0
Autor	Rafael Martín Guerrero
Requisitos asociados	RF-2.4, RF-2.5
Descripción	Permite al usuario obtener listados de plantillas con determinados filtros.
Precondición	<ul style="list-style-type: none"> ▪ La aplicación se encuentra operativa. ▪ La base de datos es correcta. ▪ El usuario está autenticado en el sistema. ▪ El usuario tiene suficientes permisos para realizar la operación.
Acciones	<ol style="list-style-type: none"> 1. Acceder al área de gestión de plantillas. 2. Escoger los distintos filtros para realizar la búsqueda. 3. Presionar en buscar. 4. Seleccionar una plantilla entre las encontradas para ver sus detalles.
Postcondición	Se muestran correctamente las plantillas, si se cumplen los criterios de búsqueda.
Excepciones	<ul style="list-style-type: none"> ▪ Se muestra un texto si no se han encontrado plantillas para esos criterios.
Importancia	Alta

Tabla B.8: CU-8 Ver y listar equipos.

CU-9	Añadir usuarios
Versión	1.0
Autor	Rafael Martín Guerrero
Requisitos asociados	RF-3-1
Descripción	Permite dar de alta usuarios en el sistema con distintos roles.
Precondición	<ul style="list-style-type: none"> ▪ La aplicación se encuentra operativa. ▪ La base de datos es correcta.
Acciones	<ol style="list-style-type: none"> 1. Acceder al área de gestión de usuarios. 2. Seleccionar dar de alta un usuario. 3. Rellenar la información del usuario. 4. Pulsar en aceptar. 5. Si no hay ningún error, se envía un mail de confirmación para verificar la cuenta.
Postcondición	El usuario ha validado su cuenta y se encuentra registrado en la BD.
Excepciones	<ul style="list-style-type: none"> ▪ Si se ha introducido algún dato único ya almacenado (por ejemplo, email) se muestra un mensaje.
Importancia	Alta

Tabla B.9: CU-9 Añadir usuarios.

CU-10	Modificar usuarios
Versión	1.0
Autor	Rafael Martín Guerrero
Requisitos asociados	RF-3-2
Descripción	Permite modificar datos relativos al usuario.
Precondición	<ul style="list-style-type: none"> ▪ La aplicación se encuentra operativa. ▪ La base de datos es correcta. ▪ El usuario se encuentra previamente creado. ▪ El usuario tiene suficientes permisos para realizar la operación.
Acciones	<ol style="list-style-type: none"> 1. Acceder al área de gestión de usuarios. 2. Seleccionar modificar usuario. 3. Rellenar la información que se quiere modificar y puede actualizarse (por ejemplo, el correo electrónico no). 4. Pulsar en guardar. 5. Si no hay ningún error, se actualiza la cuenta.
Postcondición	El usuario ha sido actualizado en la BD.
Excepciones	<ul style="list-style-type: none"> ▪ Si se ha introducido algún dato único ya almacenado (por ejemplo, email) se muestra un mensaje.
Importancia	Media

Tabla B.10: CU-10 Modificar usuarios.

CU-11	Eliminar usuarios
Versión	1.0
Autor	Rafael Martín Guerrero
Requisitos asociados	RF-3-3
Descripción	Permite dar de baja un usuario.
Precondición	<ul style="list-style-type: none"> ▪ La aplicación se encuentra operativa. ▪ La base de datos es correcta. ▪ El usuario tiene suficientes permisos para realizar la operación.
Acciones	<ol style="list-style-type: none"> 1. Acceder al área de gestión de usuarios. 2. Seleccionar el usuario que se desea eliminar. 3. Pulsar en eliminar cuenta. 4. Confirmar la eliminación. 5. Si no hay ningún error, se elimina la cuenta.
Postcondición	El usuario ha sido eliminado de la BD.
Excepciones	<ul style="list-style-type: none"> ▪ Si no se ha podido eliminar se muestra un mensaje.
Importancia	Media

Tabla B.11: CU-11 Eliminar usuarios.

CU-12	Ver y listar usuarios
Versión	1.0
Autor	Rafael Martín Guerrero
Requisitos asociados	RF-3.4, RF-3.5
Descripción	Permite obtener listados de usuarios con determinados filtros.
Precondición	<ul style="list-style-type: none"> ■ La aplicación se encuentra operativa. ■ La base de datos es correcta. ■ El usuario tiene suficientes permisos para realizar la operación.
Acciones	<ol style="list-style-type: none"> 1. Acceder al área de gestión de usuarios. 2. Escoger los distintos filtros para realizar la búsqueda. 3. Presionar en buscar. 4. Seleccionar aquel usuario entre los encontrados para ver sus detalles.
Postcondición	Se muestran correctamente los usuarios, si se cumplen los criterios de búsqueda.
Excepciones	<ul style="list-style-type: none"> ■ Se muestra un texto si no se han encontrado usuarios para esos criterios.
Importancia	Alta

Tabla B.12: CU-12 Ver y listar usuarios.

CU-13	Registrar dispositivo
Versión	1.0
Autor	Rafael Martín Guerrero
Requisitos asociados	RF-4.1
Descripción	Permite realizar un registro del dispositivo para aplicar futuras configuraciones.
Precondición	<ul style="list-style-type: none"> ■ La aplicación se encuentra operativa. ■ La base de datos es correcta. ■ El usuario posee el ejecutable que permite iniciar el registro. ■ El usuario tiene suficientes permisos para realizar la operación.
Acciones	<ol style="list-style-type: none"> 1. El usuario ejecuta el programa. 2. Se solicitan credenciales para comprobar si se tienen permisos de registro. 3. Una vez introducidos y validados, el proceso continúa. 4. Se obtiene la huella digital para ese dispositivo, se almacena en el SO y se envía a la aplicación para su registro. 5. Se programa una tarea en la que la herramienta se ejecuta periódicamente.
Postcondición	Se registra correctamente el dispositivo y se establece una tarea periódica en el sistema.
Excepciones	<ul style="list-style-type: none"> ■ Error de usuario (no tiene suficientes permisos). ■ Error de registro (se muestra un mensaje).
Importancia	Alta

Tabla B.13: CU-13 Registrar dispositivo.

CU-14	Aplicar configuración
Versión	1.0
Autor	Rafael Martín Guerrero
Requisitos asociados	RF-4.2
Descripción	Permite aplicar configuraciones a determinados dispositivos.
Precondición	<ul style="list-style-type: none"> ■ La aplicación se encuentra operativa. ■ La base de datos es correcta. ■ El dispositivo se encuentra registrado en el sistema y se le ha asignado una configuración. ■ Existe una tarea programada en el dispositivo.
Acciones	<ol style="list-style-type: none"> 1. El dispositivo ejecuta la tarea programada. 2. Se comprueba que la huella digital esté registrada en el sistema. 3. Si no se encuentra registrada, se realiza el registro, en caso contrario, continúa el proceso. 4. Se obtiene la configuración a aplicar para el dispositivo. 5. Se aplica la configuración para el dispositivo.
Postcondición	Se aplica correctamente la configuración almacenada para el dispositivo.
Excepciones	<ul style="list-style-type: none"> ■ Error al aplicar la configuración (se muestra mensaje).
Importancia	Alta

Tabla B.14: CU-14 Aplicar configuración.

CU-15	Eliminar configuración
Versión	1.0
Autor	Rafael Martín Guerrero
Requisitos asociados	RF-4.3
Descripción	Permite eliminar configuraciones de los dispositivos.
Precondición	<ul style="list-style-type: none"> ▪ La aplicación se encuentra operativa. ▪ La base de datos es correcta. ▪ El dispositivo se encuentra registrado en el sistema y se ha dado de baja una configuración. ▪ Existe una tarea programada en el dispositivo.
Acciones	<ol style="list-style-type: none"> 1. El dispositivo ejecuta la tarea programada. 2. Se comprueba que la huella digital esté registrada en el sistema. 3. Si no se encuentra registrada, se realiza el registro, en caso contrario, continúa el proceso. 4. Se obtiene que hay que eliminar la configuración. 5. Se resetea la configuración para el dispositivo.
Postcondición	Se resetea correctamente la configuración del dispositivo.
Excepciones	<ul style="list-style-type: none"> ▪ Error al aplicar la configuración (se muestra mensaje).
Importancia	Alta

Tabla B.15: CU-15 Eliminar configuración.

Diagrama de casos de uso

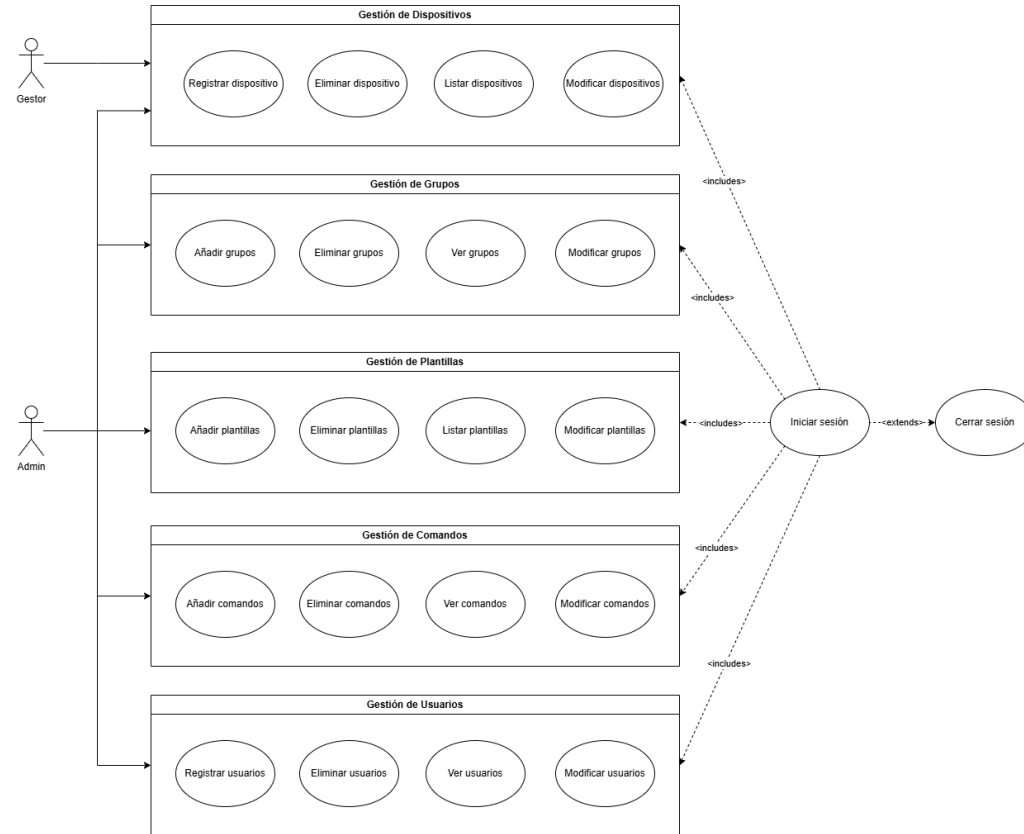


Figura B.1: Diagrama de casos de uso para la plataforma web.

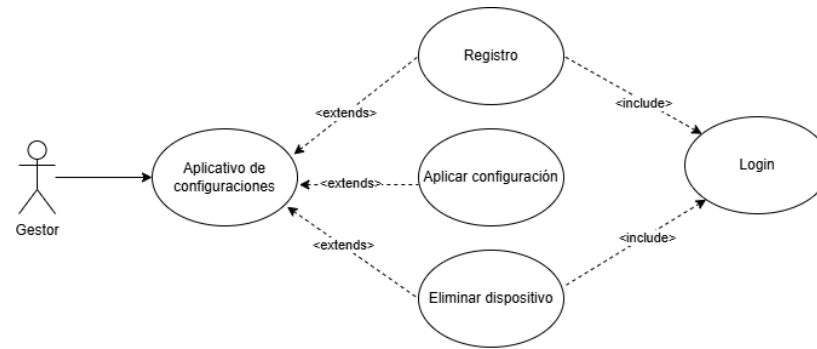


Figura B.2: Diagrama de casos de uso para el aplicativo de configuraciones.

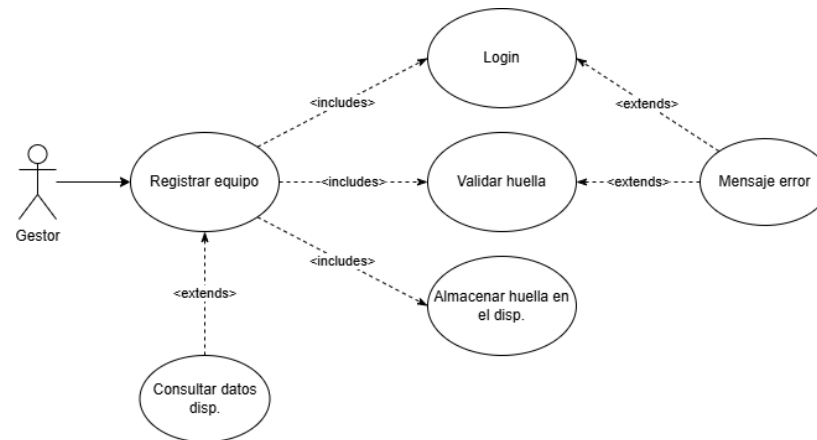


Figura B.3: Diagrama de caso de uso de registro de un dispositivo.

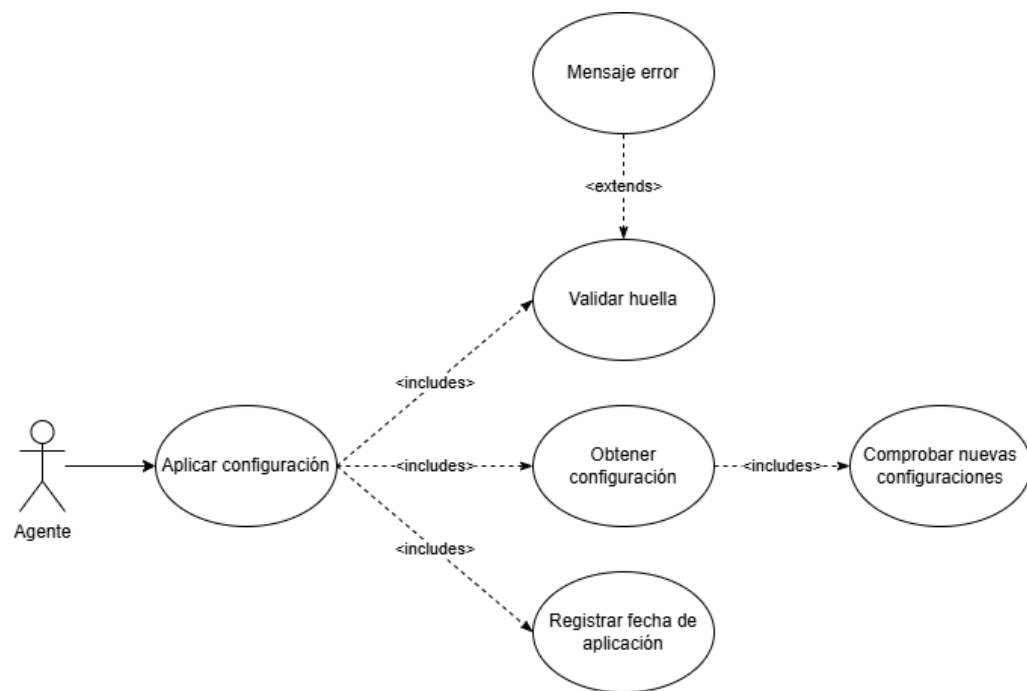


Figura B.4: Diagrama de caso de uso para la aplicación de configuración.

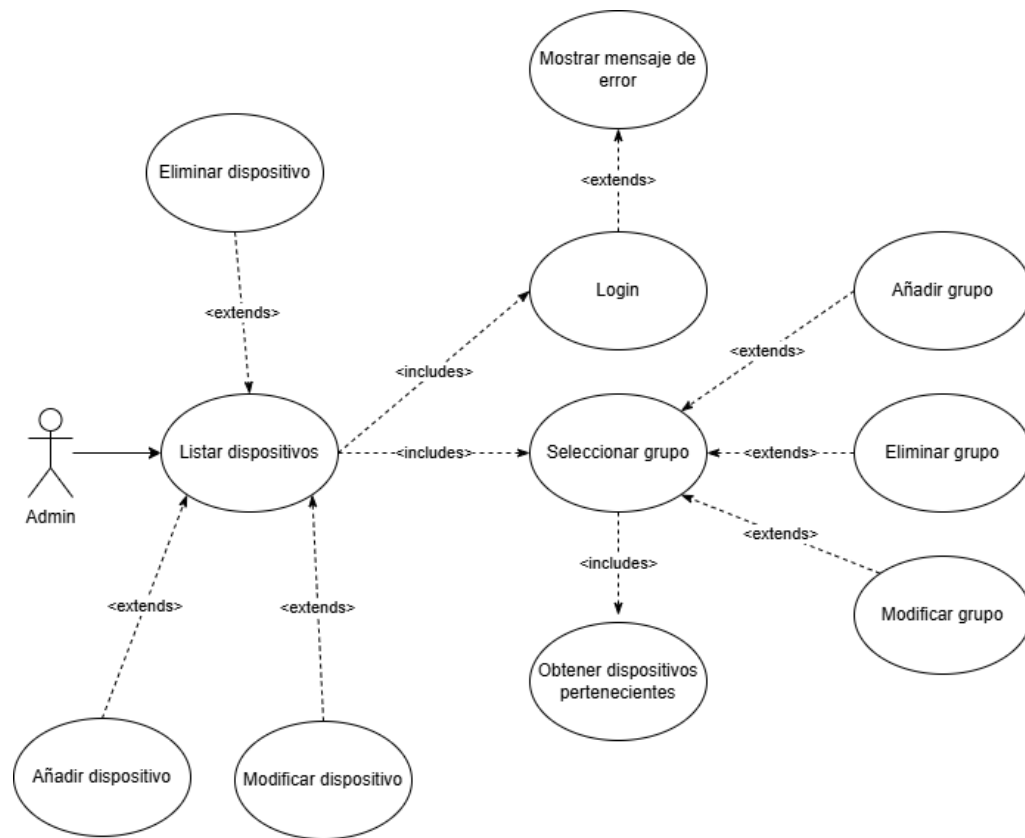


Figura B.5: Diagrama de caso de uso para listar dispositivos.

Apéndice C

Especificación de diseño

C.1. Introducción

Esta sección describe el diseño del sistema, incluyendo el diseño de datos, diseño arquitectónico, diseño procedimental y diseño de interfaces. El objetivo es proporcionar una visión general de cómo se estructuran los componentes del sistema y cómo interactúan entre sí para comprender mejor su funcionamiento.

C.2. Diseño de datos

GCA trabaja con los siguientes tipos de datos:

- **Usuarios:** figuras que interactúan con el sistema mediante un rol específico.
- **Grupos:** almacenan dispositivos y otros grupos, permitiendo una organización jerárquica, además de almacenar la plantilla asignada.
- **Dispositivos:** guardan información sobre cada dispositivo, el grupo al que pertenecen, así como su sistema operativo y la plantilla asignada.
- **Comandos:** nombre, descripción y valor del comando que se ejecutará en los dispositivos.
- **Plantillas:** nombre, descripción y lista de comandos que se ejecutarán en los dispositivos mediante un orden de ejecución.

- **Roles:** los roles que pueden tener los usuarios, como administrador o gestor.
- **Sistemas Operativos:** el conjunto de sistemas operativos que pueden tener los dispositivos.

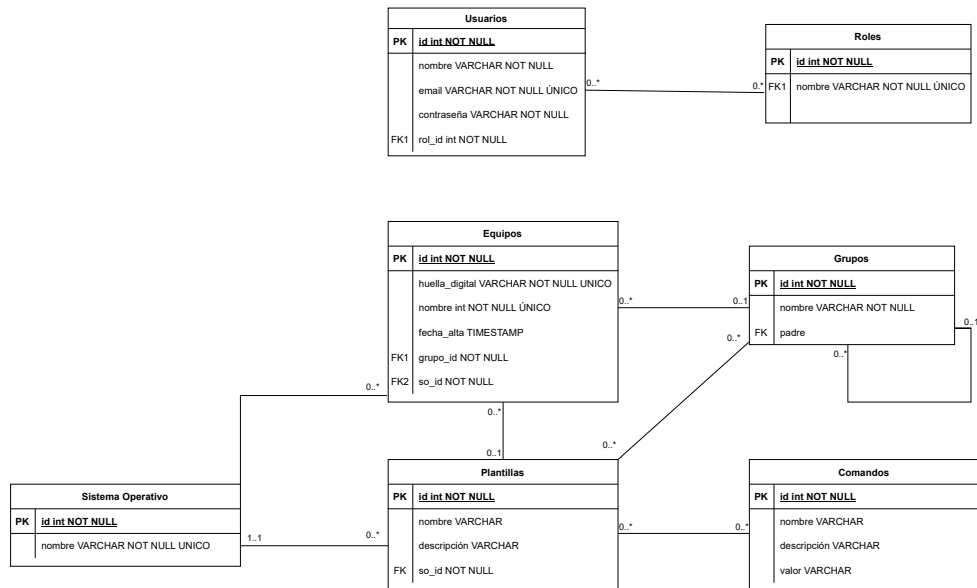


Figura C.1: Diagrama entidad relación

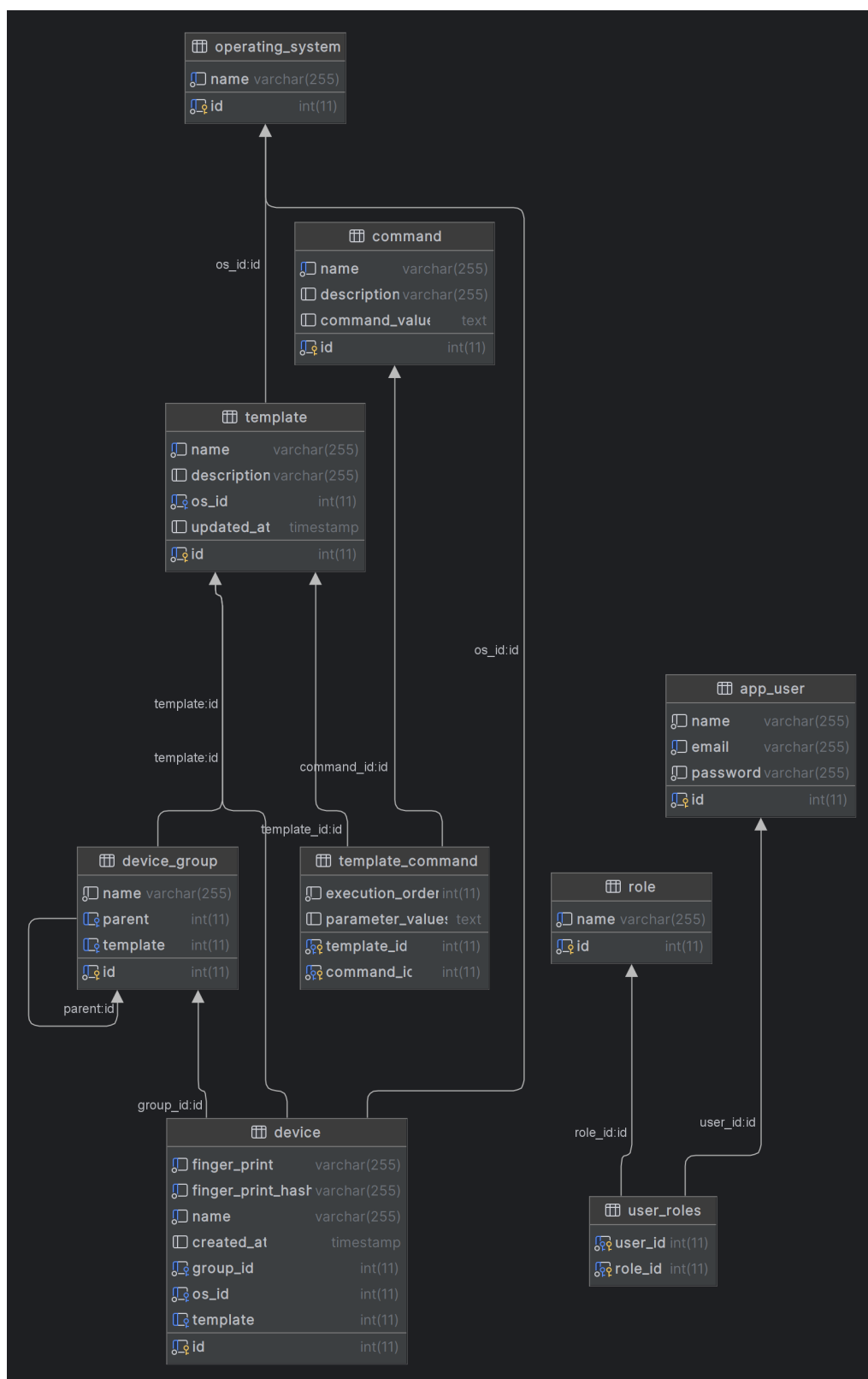


Figura C.2: Diagrama exportado en IntelliJ

C.3. Diseño arquitectónico

Antes de comenzar con la explicación de la arquitectura, hay que distinguir entre las tres partes en las que se divide este proyecto:

- **Backend:** aplicación desarrollada con Java y Spring Boot. Se trata de una API la cual consumen el resto de componentes.
- **Frontend:** aplicación Angular utilizada para llevar a cabo la gestión de dispositivos.
- **Cliente:** aplicación Python que genera un instalador para ejecutar en los distintos dispositivos sobre los que se quieren aplicar las configuraciones.

Backend

El patrón MVC (Modelo-Vista-Controlador) es un patrón de diseño que separa la lógica de negocio, la interfaz de usuario y el control de flujo de la aplicación[9].

En el caso del modelo, este se obtiene con ayuda de JPA, que mapea las entidades de la base de datos a objetos Java que se pueden manipular, sin necesidad de escribir consultas SQL directamente. La interacción con la base de datos se consigue mediante el uso del patrón de repositorio, que permite realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre las entidades de forma sencilla.

La vista se mantiene totalmente independiente del backend y se gestiona desde el frontend, que explicaremos en la siguiente sección.

El controlador, por su parte, se encarga de recibir las peticiones HTTP y dirigirlas a los servicios correspondientes. Para los servicios se ha aplicado una arquitectura de microservicios, donde cada servicio es responsable de una parte específica de la lógica de negocio: uno para gestionar usuarios, otro para grupos, dispositivos... Se consigue así una mayor modularidad y escalabilidad, permitiendo que cada servicio pueda evolucionar de forma independiente.

El backend se estructura de la siguiente manera:

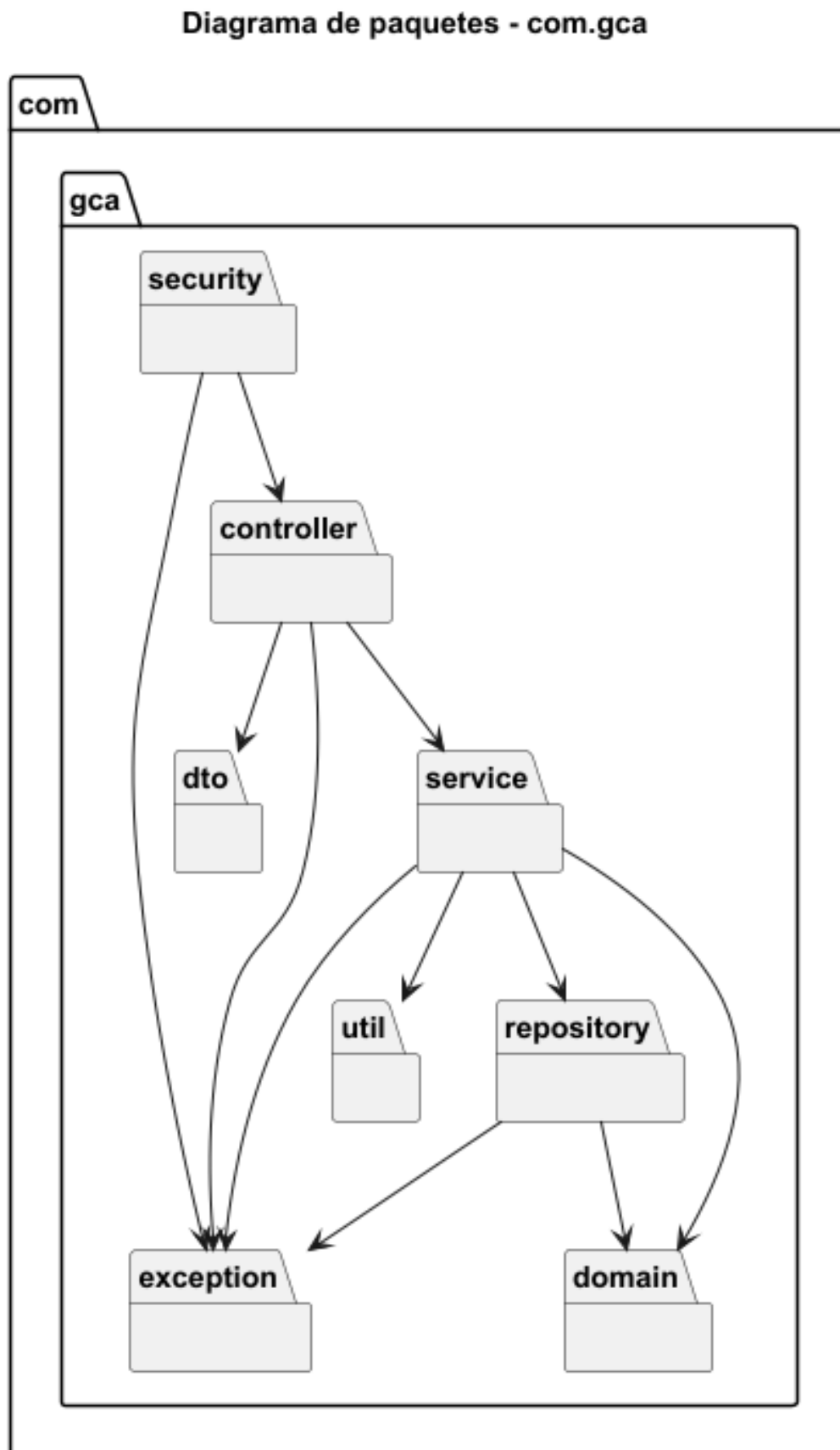


Figura C.3: Diagrama de paquetes del backend

Los paquetes que lo componen son:

- **Config**: se incluyen archivos para la configuración de la aplicación.
- **Controller**: contiene los controladores que gestionan las peticiones HTTP y utilizan los servicios.
- **Domain**: contienen las clases equivalentes a cada una de las tablas de la base de datos.
- **Dto**: se utilizan para enviar datos de los modelos a la vista, evitando así enviar información sensible o innecesaria.
- **Exception**: contiene las excepciones personalizadas que pueden lanzar el sistema.
- **Repository**: contiene las interfaces que extienden de JPA y permiten realizar operaciones CRUD.
- **Security**: contiene la configuración de seguridad de la aplicación; incluye la autenticación y autorización.
- **Service**: implementa la lógica de negocio y utiliza los repositorios.
- **Util**: clases de utilidad que pueden ser utilizadas en diferentes partes del sistema.

La estructura completa de clases del backend es la siguiente:

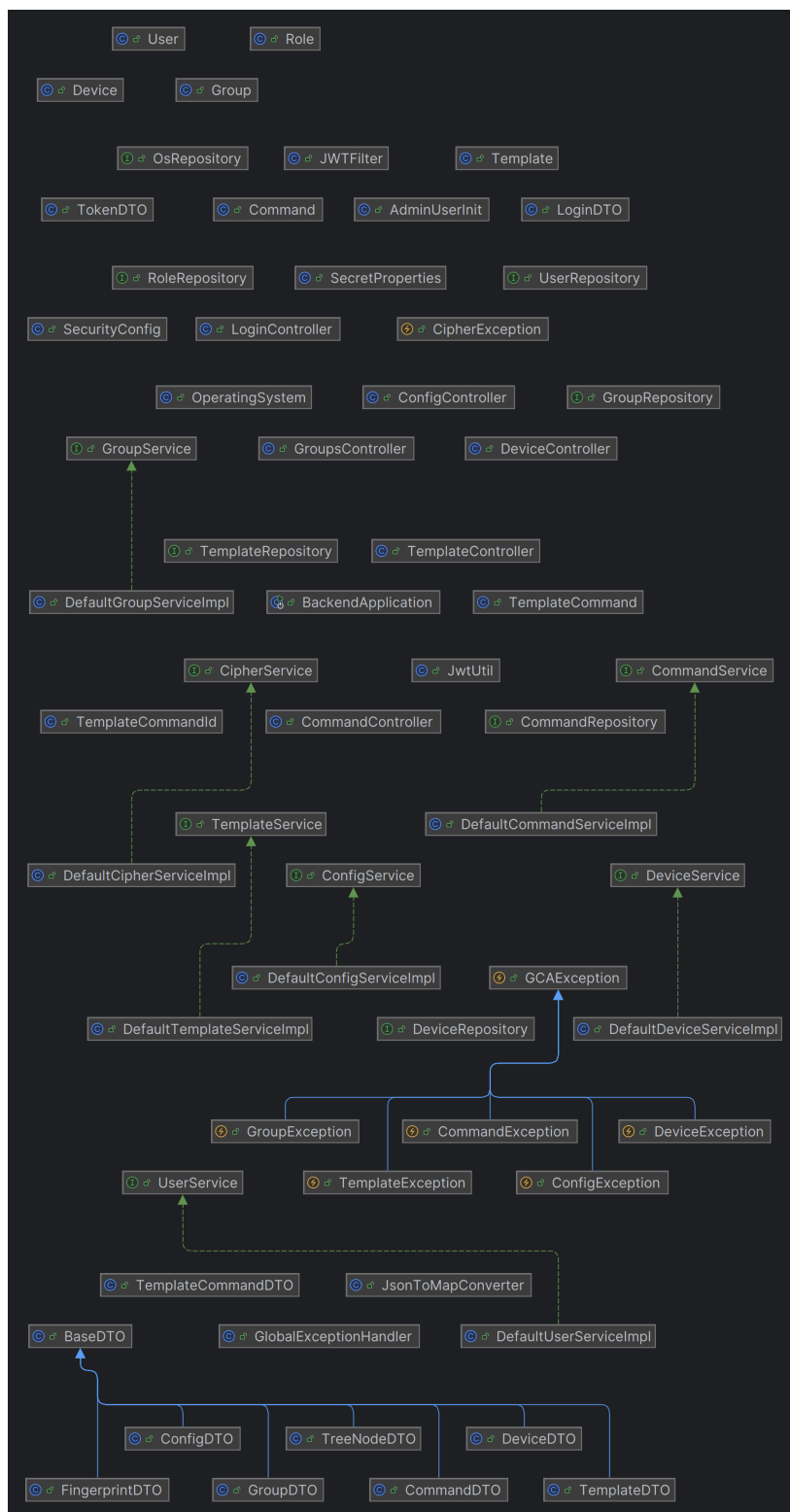


Figura C.4: Diagrama de clases del backend

Frontend

El front se estructura de forma similar al backend, siguiendo el patrón MVC, pero adaptado a Angular.

Dentro de la carpeta `src/app` se encuentra dividida la estructura de la siguiente manera:

- **Components:** contiene los componentes de la aplicación, que son las partes reutilizables de la interfaz de usuario.
- **Model:** contiene las clases que representan los modelos de datos utilizados en la aplicación.
- **Services:** contiene los servicios que se encargan de realizar peticiones HTTP al backend y gestionar la lógica de negocio del frontend.

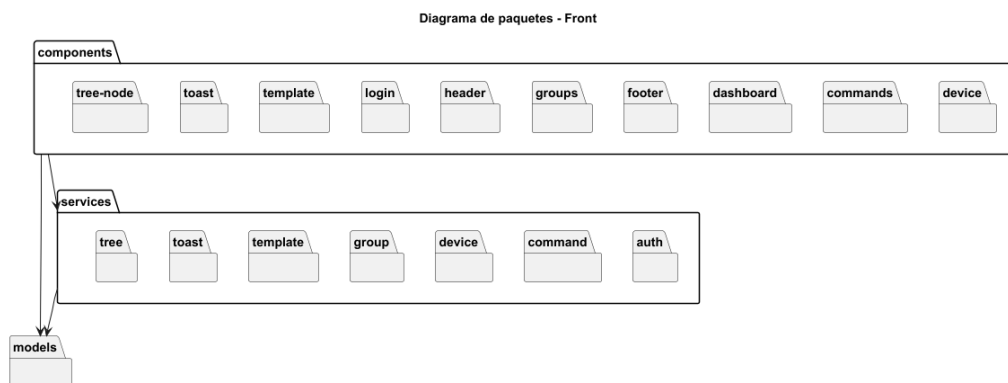


Figura C.5: Diagrama de componentes Angular

Cliente

El cliente Python se estructura de la siguiente manera:

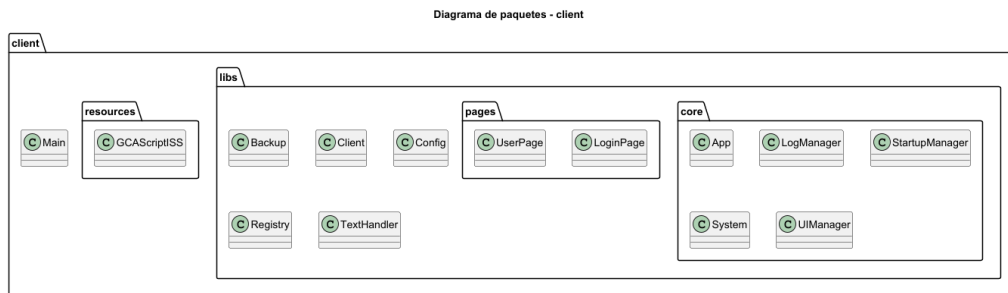


Figura C.6: Paquetes de Python

Se divide en los siguientes paquetes:

- **libs**: contiene las librerías utilizadas.
- **libs.core**: contiene las clases que implementan la lógica principal.
- **libs.pages**: contiene las páginas que se utilizan para interactuar con el usuario.

El punto de entrada del cliente es el archivo `main.py`, que se encarga de inicializar la aplicación y cargar la configuración necesaria.

C.4. Diseño procedimental

Esta sección muestra los diagramas de secuencia de los principales procesos del sistema, los cuales describen cómo interactúan los diferentes componentes y cómo se llevan a cabo las operaciones:

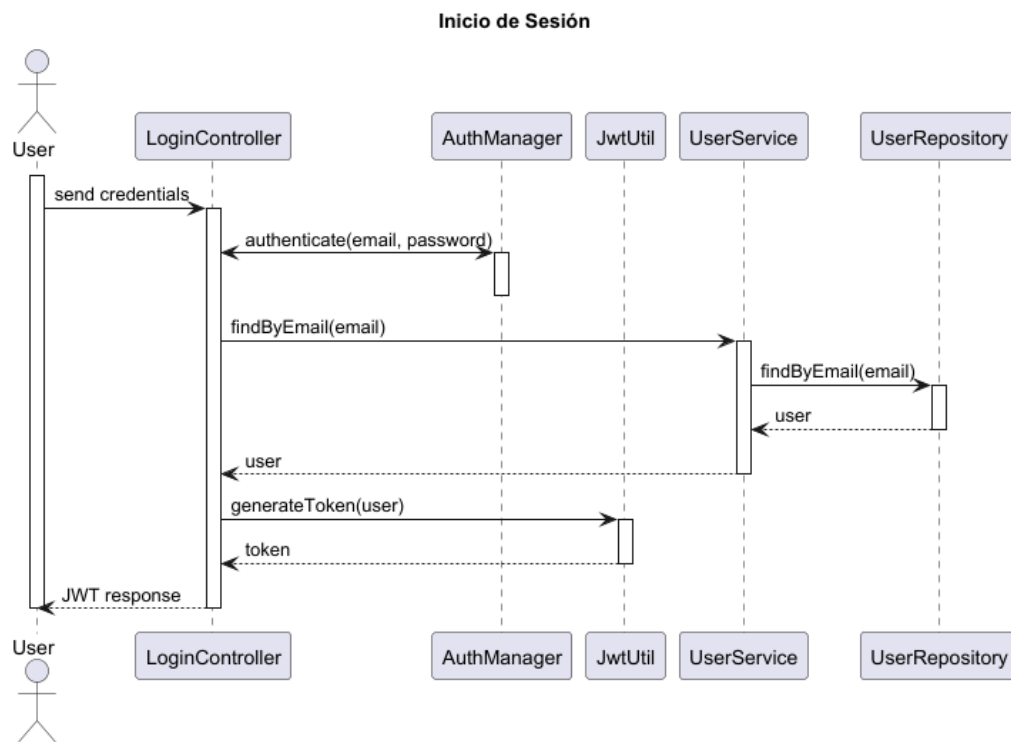


Figura C.7: Diagrama de secuencia del inicio de sesión.

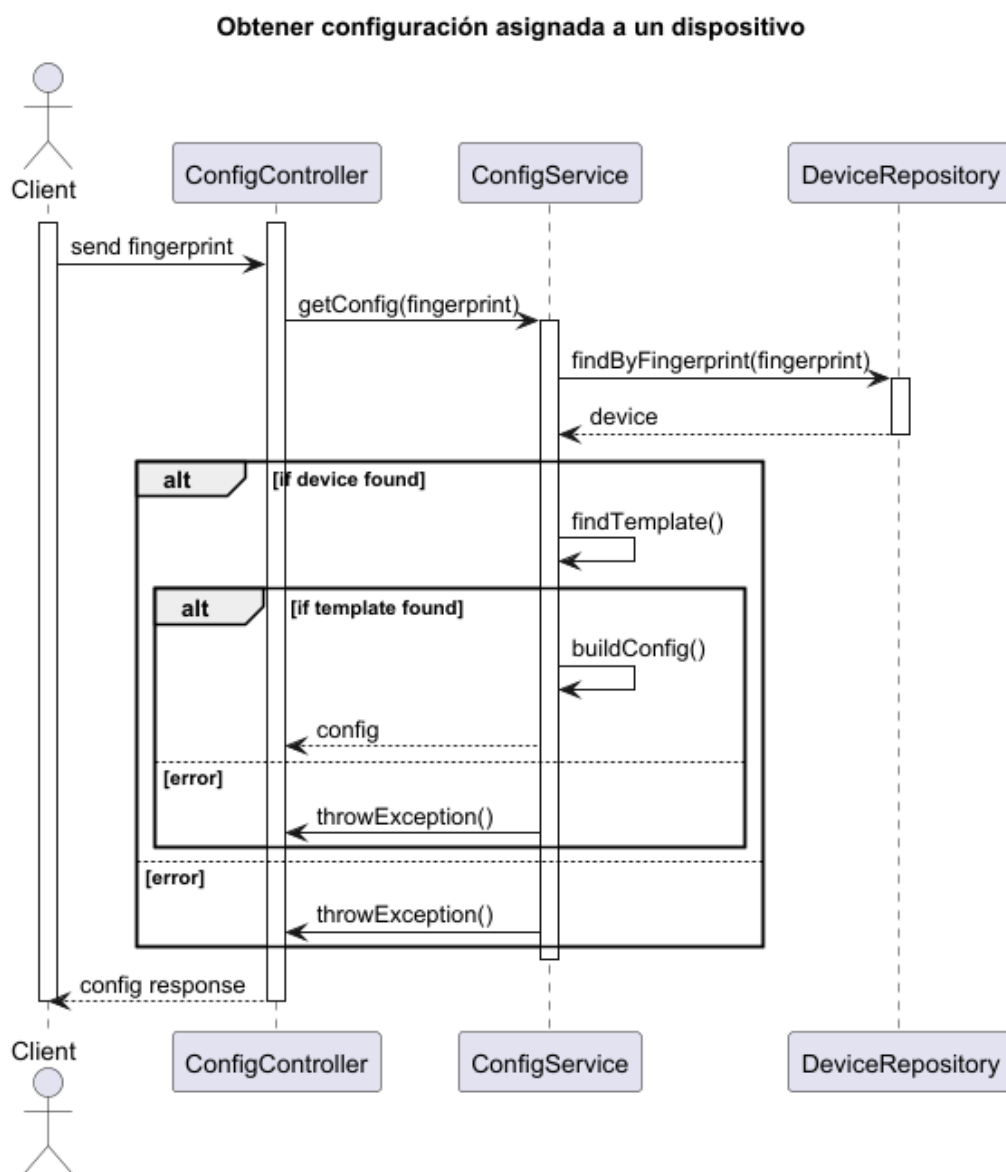


Figura C.8: Diagrama de secuencia para la obtención de plantillas.

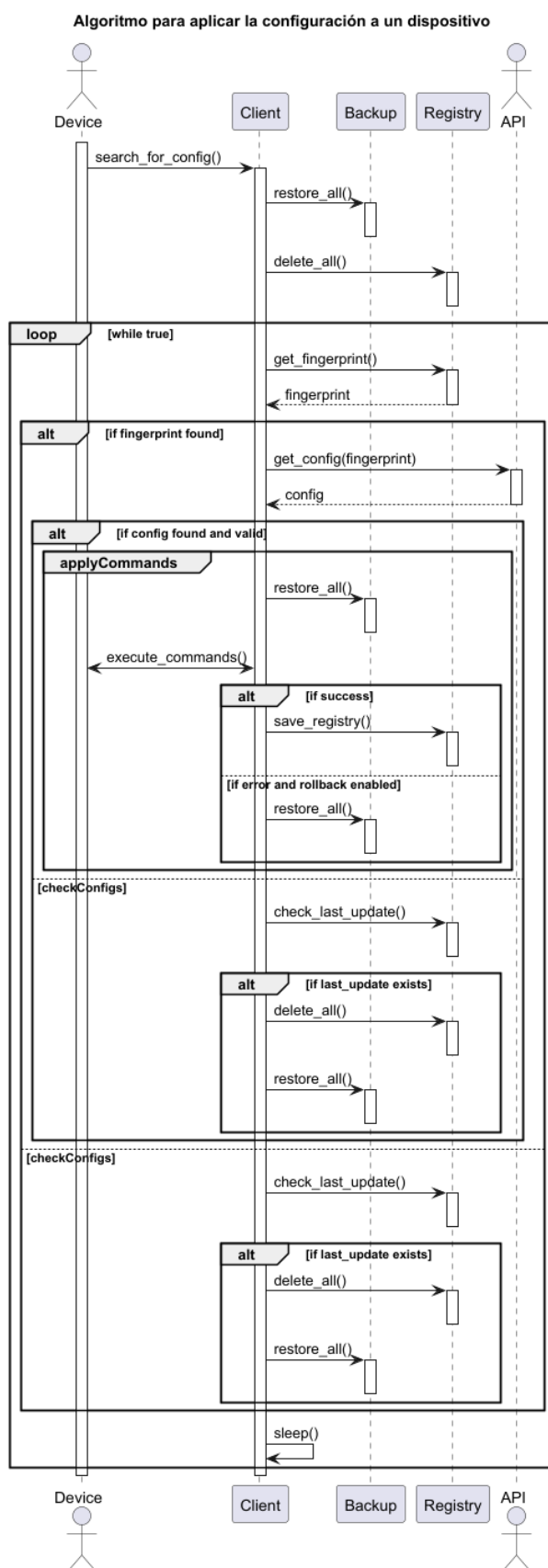
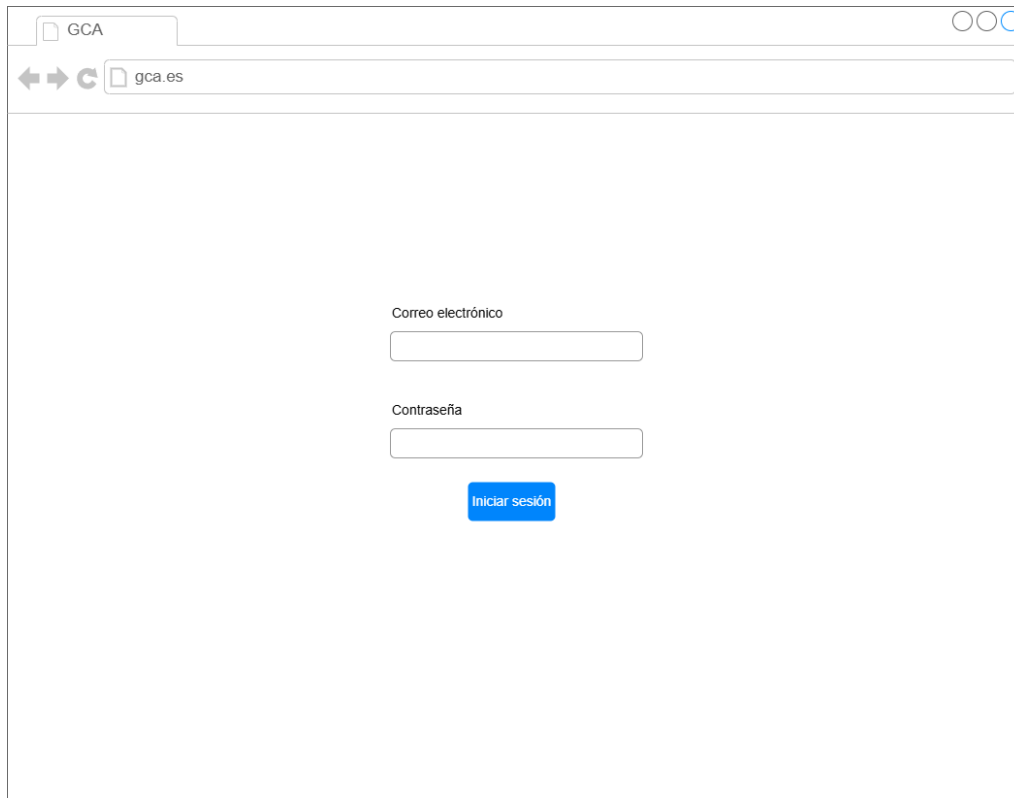


Figura C.9: Diagrama de secuencia del algoritmo principal.

C.5. Diseño de interfaces

Se han creado prototipos de las principales pantallas del sistema para facilitar la comprensión de la interfaz y partir de una base visual para el desarrollo.



El prototipo muestra una ventana de navegador con una pestaña etiquetada como 'GCA'. La barra de direcciones contiene 'gca.es'. El contenido principal de la página es un formulario de inicio de sesión centrado, que incluye:

- Una etiqueta 'Correo electrónico' sobre un campo de entrada rectangular.
- Una etiqueta 'Contraseña' sobre un campo de entrada rectangular.
- Un botón rectangular azul con el texto 'Iniciar sesión' en blanco, situado debajo de los campos de entrada.

Figura C.10: Prototipo inicio de sesión.

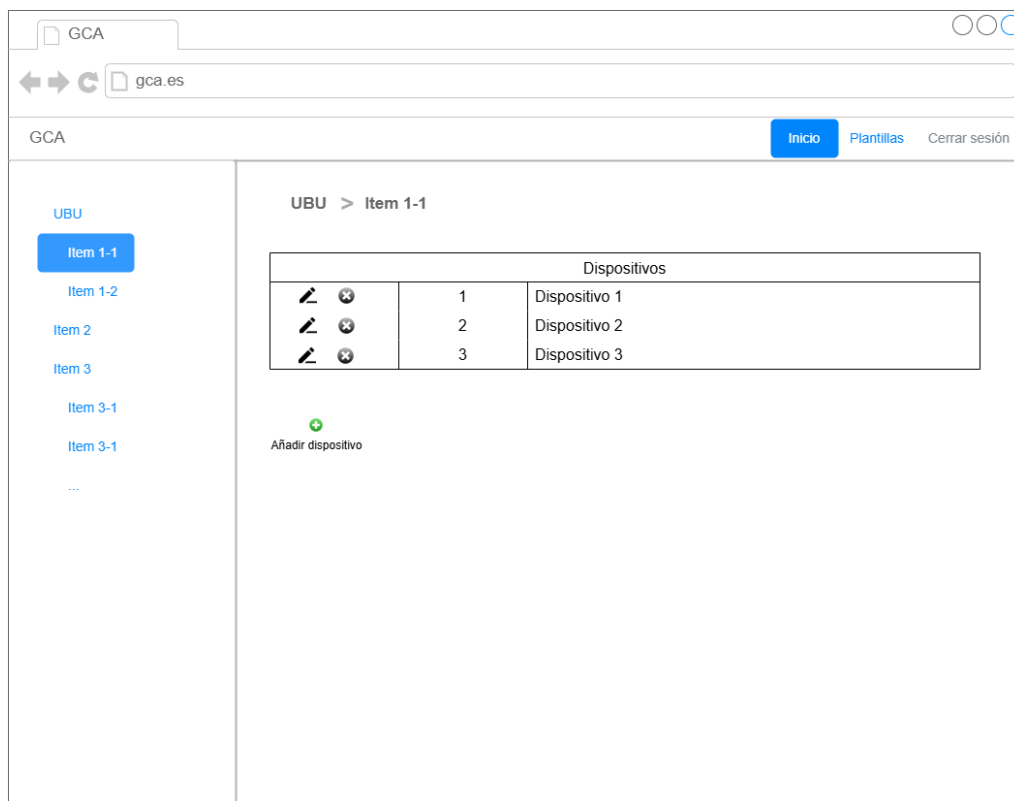


Figura C.11: Prototipo listar grupos y dispositivos.

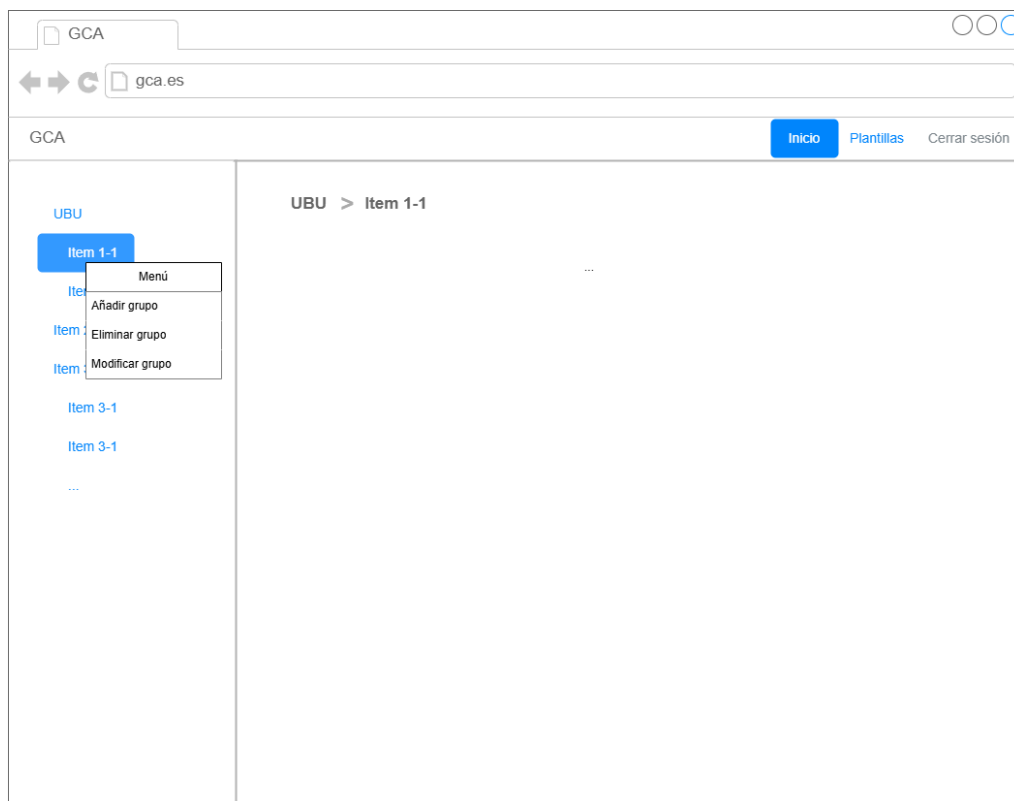


Figura C.12: Prototipo acciones grupos.

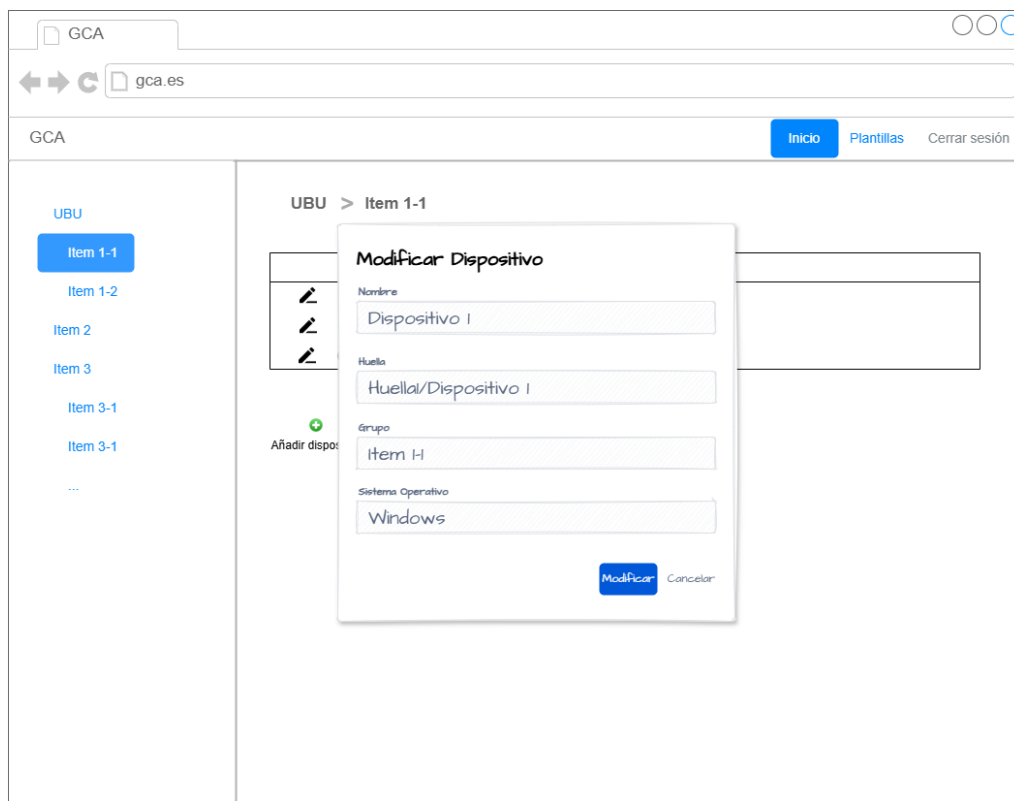
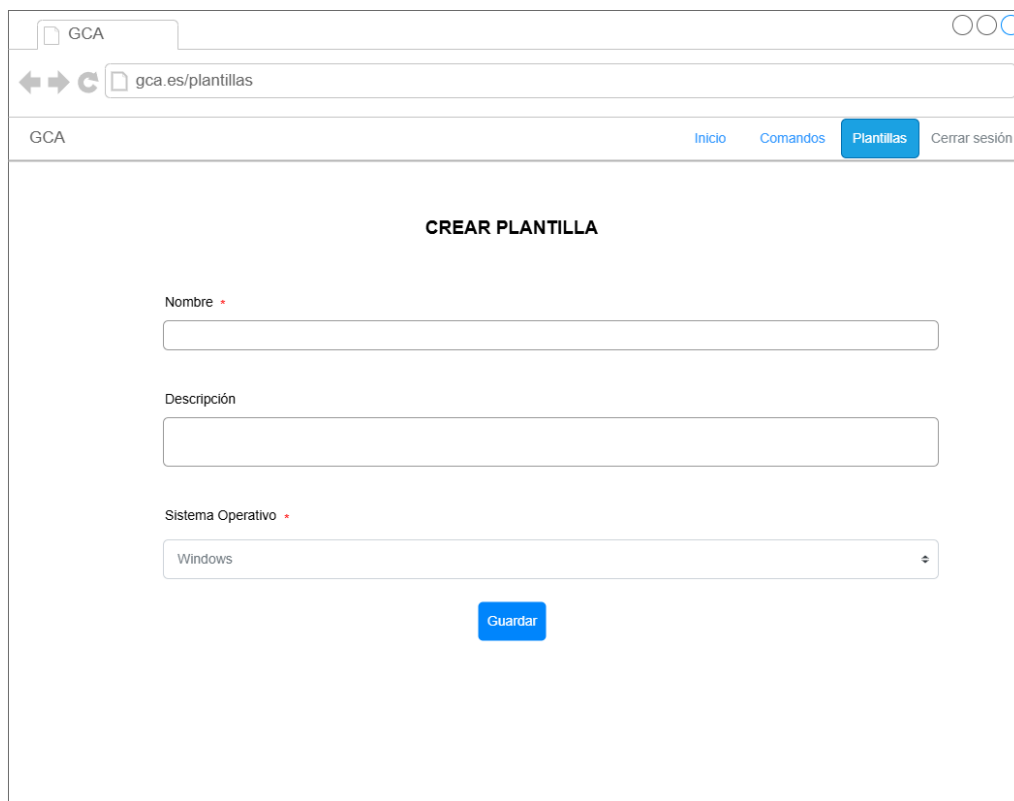


Figura C.13: Prototipo modificar dispositivo.

El prototipo muestra una ventana de navegador con la URL `gca.es/comandos`. La interfaz incluye una barra de navegación superior con el logo 'GCA' y los enlaces 'Inicio', 'Comandos' (destacado), 'Plantillas' y 'Cerrar sesión'. El contenido principal está encabezado por el título 'CREAR COMANDO'. A continuación, se encuentran tres campos de entrada: 'Nombre' (con un asterisco rojo indicando un campo obligatorio), 'Descripción' y 'Comando' (también con un asterisco rojo). Cada campo tiene un recuadro de texto correspondiente. En la parte inferior del formulario, hay un botón azul con el texto 'Guardar'.

Figura C.14: Prototipo crear comando.



The image shows a web browser window with the address bar displaying 'gca.es/plantillas'. The browser's title bar shows 'GCA' and standard window controls. The page has a navigation bar with 'Inicio', 'Comandos', 'Plantillas' (highlighted), and 'Cerrar sesión'. The main content area is titled 'CREAR PLANTILLA' and contains three form fields: 'Nombre' (with a red asterisk), 'Descripción', and 'Sistema Operativo' (with a red asterisk). The 'Sistema Operativo' field is a dropdown menu currently showing 'Windows'. A blue 'Guardar' button is positioned below the form fields.

GCA

Inicio Comandos **Plantillas** Cerrar sesión

CREAR PLANTILLA

Nombre *

Descripción

Sistema Operativo *

Windows

Guardar

Figura C.15: Prototipo crear plantilla.

Asignar comandos a plantilla

Seleccionar plantilla

Restricción completa [Windows]

Comandos disponibles

- + Deshabilitar red
- + Permitir acceso a Matlab
- + Desactivar firewall
- ...

Comandos asignados

Orden	
1	Abrir puerto 33
2	Reiniciar dispositivo
n	...

Asignar

Figura C.16: Prototipo asignación de comandos en plantillas.

The screenshot displays a web application interface for assigning templates. The browser's address bar shows 'gca.es/asignacion'. The page header includes the 'GCA' logo and navigation links: 'Inicio', 'Asignacion' (highlighted in blue), 'Plantillas', and 'Cerrar sesión'. A left sidebar shows a tree structure under 'UBU' with 'Item 1-1' selected. The main content area is titled 'UBU > Item 1-1' and contains form fields for 'Plantilla aplicada:' (Restricción básica), 'Descripción' (Plantilla básica para deshabilitar acceso a), 'Sistema operativo' (Windows), and 'Comandos' (a list of commands). At the bottom are two buttons: 'Cambiar plantilla' (green) and 'Eliminar asignación' (red).

Figura C.17: Prototipo asignación de plantillas.

El diseño final se ha realizado con la ayuda de Tailwind CSS y DaisyUI, ya que proporcionan una base sólida para el diseño de la interfaz, permitiendo una personalización rápida y sencilla. El resultado puede consultarse en el apartado de *Manual del usuario*, donde se muestran ejemplos de las pantallas del sistema.

Si desea obtener más información de cómo funciona Tailwind CSS, puede consultar su documentación oficial en <https://tailwindcss.com/docs/installation> y del mismo modo, para DaisyUI en <https://daisyui.com/docs/install/>.

Apéndice D

Documentación técnica de programación

D.1. Introducción

El objetivo de este apéndice es ofrecer una visión clara y estructurada de los procesos de implementación de la aplicación, de manera que pueda utilizarse por otros programadores como referencia para el mantenimiento y el desarrollo de nuevas funcionalidades. Al tratarse de documentación técnica, se entiende que el lector posee de nociones en la materia.

D.2. Estructura de directorios

El directorio del proyecto se puede consultar en el repositorio [GitHub](#).

La estructura de directorios del proyecto es la siguiente:

- **GCA/**: Directorio raíz del proyecto. Contiene todos los archivos y subdirectorios del proyecto, README.md, .env, docker-compose.yml y licencia.
- **GCA/.github/**: Acciones de GitHub. Contiene los archivos de configuración para las acciones de GitHub, utilizando workflows para la integración y despliegue continuo (CI/CD).
- **GCA/backend/**: Proyecto Spring Boot. Contiene el código fuente del backend, archivos de configuración y dependencias.
- **GCA/backend/src/main/java/**: Código fuente del backend. Contiene el código Java de la aplicación.

- `GCA/backend/src/main/resources/`: Recursos del backend. Contiene los archivos de configuración y ficheros para la base de datos.
- `GCA/backend/src/test/java`: Tests del backend. Contiene las pruebas unitarias y de integración.
- `GCA/backend/src/test/resources`: Recursos de pruebas. Contiene los archivos de configuración y ficheros necesarios para las pruebas.
- `GCA/client/`: Aplicación de escritorio Python. Contiene el código fuente de la aplicación de escritorio.
- `GCA/client/libs/`: Librerías del cliente. Contiene las librerías necesarias para la aplicación.
- `GCA/client/output/`: Instalador del cliente.
- `GCA/client/resources/`: Archivos de recursos del cliente. Contiene los archivos de configuración e imágenes.
- `GCA/docs/`: Documentación del proyecto. Contiene los documentos de diseño, manuales y otros archivos.
- `GCA/docs/img/`: Imágenes.
- `GCA/frontend/`: Proyecto Angular. Contiene el código fuente del frontend, archivos de configuración...
- `GCA/frontend/public/`: Archivos públicos del frontend (imágenes, iconos...).
- `GCA/frontend/src/`: Código fuente del frontend. Contiene el código TypeScript de la aplicación.
- `GCA/frontend/src/app/`: Código de la aplicación (componentes, servicios, modelos...).
- `GCA/frontend/src/environments/`: Configuración de los distintos entornos.

D.3. Manual del programador

Como hemos mencionado en la introducción, este apéndice está destinado a programadores que deseen comprender la implementación del proyecto. Se detallan todos aquellos aspectos necesarios para la instalación, compilación y ejecución del proyecto, así como las pruebas del sistema.

D.4. Compilación, instalación y ejecución del proyecto

En primer lugar, es importante mencionar que el proyecto se ha realizado bajo el sistema operativo Windows 11, por lo que las instrucciones de

instalación y ejecución están orientadas a este sistema operativo. No obstante, la gran mayoría de instrucciones son aplicables a otros entornos.

El proyecto se divide en tres partes principales: el backend, el frontend y el cliente. Aunque estén separadas, se encuentran en el mismo repositorio para favorecer una mayor facilidad de uso y despliegue.

Existen dos formas de ejecutar el proyecto: mediante Docker o de forma manual.

Docker

Para ejecutar el proyecto mediante Docker, es necesario tener instalado Docker en el sistema: se puede descargar [aquí](#).

Al instalar Docker Desktop, se nos permite tener una interfaz gráfica para gestionar los contenedores, sin perjuicio de poder utilizarse la línea de comandos si se prefiere.

Tras instalar Docker, podemos comprobar que se ha instalado correctamente desde nuestra terminal ejecutando:

```
docker --version
```

Si se ha instalado correctamente, nos mostrará la versión de Docker instalada. Iniciamos la aplicación Docker Desktop y nos aseguramos de que está en ejecución.

Nos dirigimos al directorio raíz del proyecto y ejecutamos el siguiente comando:

```
docker-compose up --build
```

Si al ejecutarlo se produce un error del tipo `yaml: did not find expected key`, es posible que haya un error de propiedades mal formadas en el archivo `docker-compose.yml`.

Si todo ha ido bien, se iniciarán los contenedores del backend, frontend y base de datos; podremos comprobarlo desde la interfaz de Docker Desktop:

<input type="checkbox"/>		Name
<input type="checkbox"/>	▼	gca
<input type="checkbox"/>		mariadb
<input type="checkbox"/>		backend
<input type="checkbox"/>		frontend

Figura D.1: Contenedores de Docker.

Una vez que los contenedores estén en ejecución, podemos acceder al frontend desde el navegador web en la dirección `http://localhost:4200` y al backend en `http://localhost:8080` o las urls que hayamos configurado.

Para que funcione correctamente, la base de datos debe levantarse cuando se inicie el backend, ya que este utiliza la base de datos para iniciar el contexto de la aplicación. Esto se consigue mediante la configuración del `docker-compose.yml`, donde se especifica que el backend depende del servicio de la base de datos.

Manual

Backend

Para ejecutar el backend de forma manual, es necesario tener instalar:

- JDK 17.
- Gradle.
- Git.
- MariaDB.

En este caso decidí utilizar IntelliJ IDEA como IDE, pero se puede utilizar cualquier otro IDE compatible con Java.

El proyecto se puede importar directamente utilizando la URL del repositorio de GitHub o clonando el repositorio y después importándolo en el IDE.

Para este proyecto en concreto, es conveniente clonar el repositorio y abrirlo desde el IDE, ya que la carpeta del proyecto que vamos a utilizar es la carpeta `backend/` y no la raíz del proyecto.

Para clonar el repositorio, abrimos una terminal y ejecutamos el siguiente comando:

```
git clone https://github.com/rmg1008/GCA.git
```

Una vez clonado el repositorio, abrimos IntelliJ IDEA y seleccionamos la opción de abrir un proyecto existente: seleccionamos la carpeta `GCA/backend/` e IntelliJ IDEA nos preguntará si queremos importar el proyecto como un proyecto de Gradle; le decimos que sí.

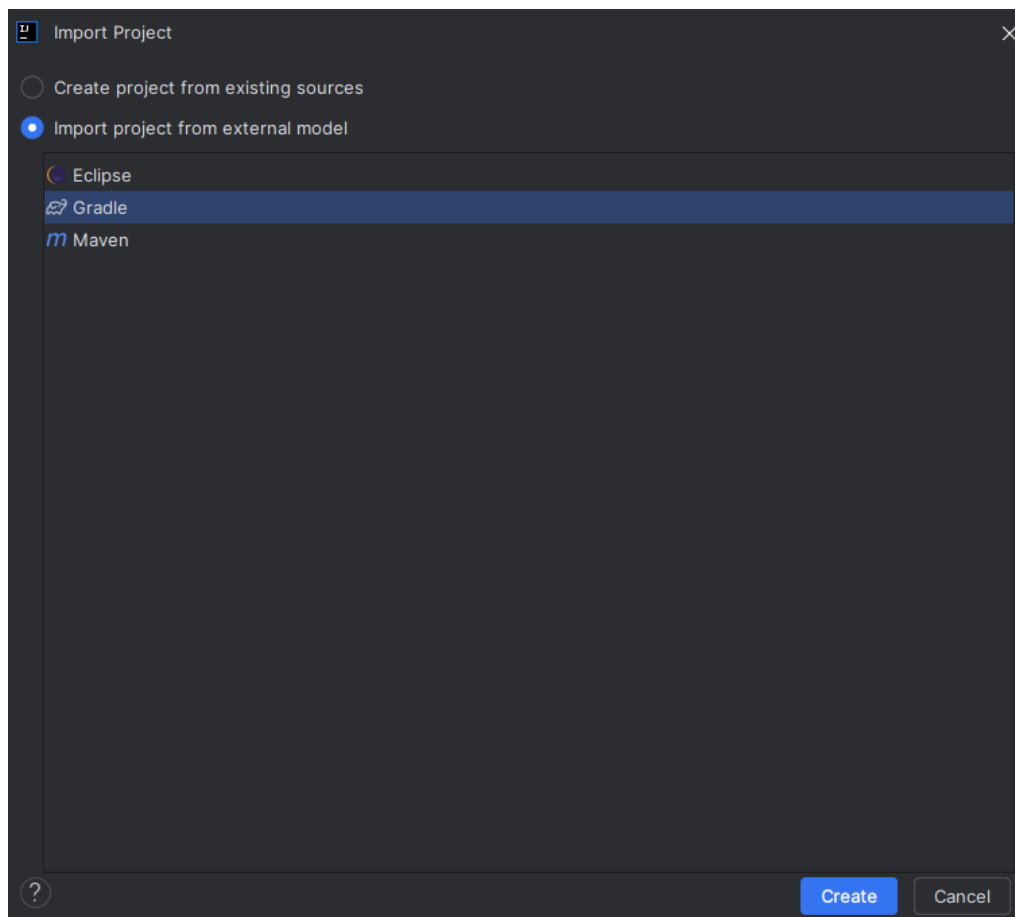


Figura D.2: Importar proyecto con Gradle.

Una vez importado el proyecto, IntelliJ IDEA descargará las dependencias necesarias y configurará el proyecto. Es importante configurar en el proyecto la versión sdk de Java 17, pues es la versión sobre la que se ha desarrollado el proyecto y es necesaria para que SonarQube funcione correctamente, porque no es compatible con versiones superiores. Para ello, vamos a **Archivo > Estructura del proyecto > SDK del proyecto** y seleccionamos la versión 17.

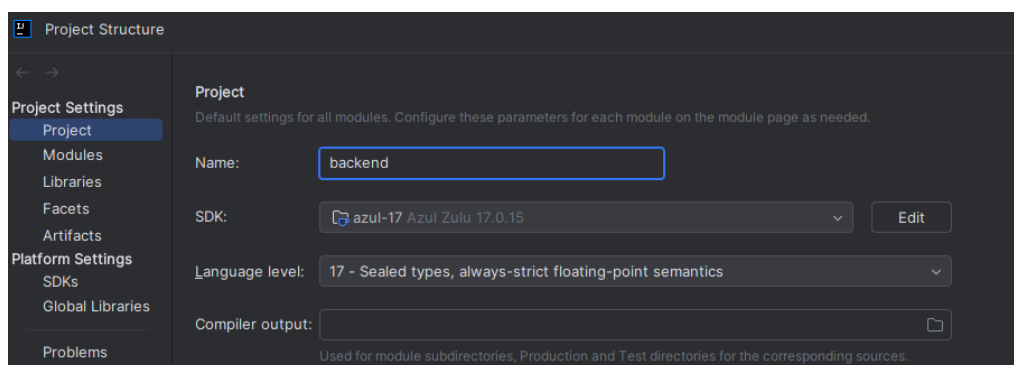


Figura D.3: Configuración del SDK de proyecto.

Se puede descargar a través del siguiente [enlace](#).

Una vez configurado el SDK, podemos proceder a configurar la base de datos. Para ello, debemos crear una base de datos en MariaDB para el entorno de desarrollo. Se puede descargar MariaDB desde su página oficial [aquí](#).

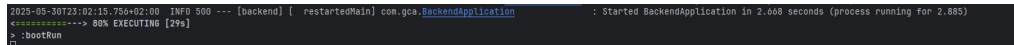
No importa cómo se haya nombrado la base de datos, pero es importante que se configure correctamente en el archivo `application.properties` que se encuentra en la carpeta `src/main/resources/` del proyecto. En este archivo encontramos todas aquellas configuraciones que podemos modificar antes de la ejecución del proyecto.

IntelliJ nos ofrece la posibilidad de configurar la conexión a la base de datos desde el IDE: para ello, vamos a **Ver > Herramientas de base de datos** y creamos una nueva conexión a la base de datos MariaDB. De esta forma, podremos ver las tablas y datos de la base de datos sin salir del IDE.

Por último, para iniciar el proyecto, podemos utilizar la opción de lanzarlo desde IntelliJ IDEA, o bien ejecutar el siguiente comando en la terminal:

```
./gradlew bootRun
```


Si todo ha ido bien, el backend se iniciará y podremos acceder a la API REST desde la dirección `http://localhost:8080`.



```
2025-05-30T23:02:19.756+02:00 INFO 500 --- [backend] [ restartedMain] com.gca.BackendApplication : Started BackendApplication in 2.668 seconds (process running for 2.885)
<=====--> 80% EXECUTING [29s]
> .bootRun
```

Figura D.4: Ejecución del backend.

No obstante, es recomendable ejecutar el proyecto desde IntelliJ IDEA, ya que nos ofrece una mejor experiencia de desarrollo y nos permite depurar el código ejecutándolo en modo debug.

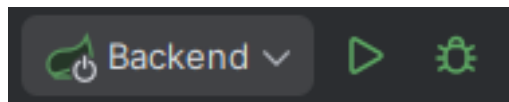


Figura D.5: Ejecución del backend desde IntelliJ IDEA.

Para las llamadas a la API REST, hemos utilizado la herramienta Postman, que permite realizar peticiones HTTP de forma sencilla y visual. Se puede descargar desde su página oficial [aquí](#).

Todas las librerías y dependencias del proyecto se gestionan mediante Gradle. Si deseamos añadir o modificar alguna extra, podemos hacerlo en el archivo `build.gradle` que se encuentra en la carpeta raíz del proyecto.

Frontend

Para ejecutar el frontend de forma manual, es necesario tener instalado:

- Node.js.
- Angular CLI.
- Visual Studio Code.

NPM se instala junto a Node.js, por lo que no es necesario instalarlo de forma independiente. Se puede descargar desde su página oficial [aquí](#).

Para instalar Angular CLI, podemos utilizar el siguiente comando en la terminal:

```
npm install -g @angular/cli
```

Las versiones actuales de Angular CLI y Node.js son las siguientes:



```
C:\Users\rafa_>ng version

Angular CLI
Angular CLI: 19.2.8
Node: 22.14.0
Package Manager: npm 10.9.2
OS: win32 x64
```

Figura D.6: Versiones del proyecto frontend.

Una vez instalado, ejecutamos Visual Studio Code y abrimos la carpeta del proyecto **frontend/**. Para una mejor experiencia de desarrollo, es recomendable instalar las extensiones de Angular y TypeScript.

Además, es necesario instalar las dependencias del proyecto la primera vez que lo importamos. Para ello, abrimos una terminal en Visual Studio Code y ejecutamos el siguiente comando:

```
npm install
```

De cara al futuro, si se desea añadir o modificar alguna dependencia, podemos hacerlo en el archivo **package.json** que se encuentra en la carpeta raíz del proyecto, y después, ejecutar de nuevo el comando anterior o bien utilizar NPM para instalar una dependencia concreta:

```
npm install nombre-dependencia
```

Para iniciar el proyecto, abrimos una terminal en Visual Studio Code y ejecutamos el siguiente comando:

```
ng serve
```

Una vez listo, el proyecto se iniciará y podremos acceder al frontend desde el navegador web en la dirección **http://localhost:4200/dashboard**.

Cliente

Para importar el cliente de forma manual, es necesario tener instalado:

- Python 3.10 o superior.
- Pip (incluido con Python).
- PyInstaller.
- PyCharm o cualquier otro IDE compatible con Python.
- Inno Setup.

Se puede descargar Python para Windows desde su página oficial [aquí](#).

Para instalar PyInstaller, podemos utilizar el siguiente comando en la terminal:

```
pip install pyinstaller
```

Para instalar PyCharm, podemos descargarlo desde su página oficial [aquí](#).

Para instalar Inno Setup, podemos descargarlo desde su página oficial [aquí](#).

Con todo instalado, abrimos PyCharm e importamos el proyecto desde la carpeta `client/`.

Desde PyCharm, podemos ejecutar el cliente directamente y utilizar el debug, siempre y cuando hayamos ejecutado el IDE como administrador para que pueda acceder a los recursos del sistema y el entorno sea Windows.

Antes de crear el ejecutable, podemos modificar el archivo `config.json` que se encuentra en la carpeta `resources/` del cliente y configurar la URL del backend, entre otras opciones.

Para crear un instalador del cliente, podemos ejecutar el siguiente comando en la terminal desde la carpeta del cliente:

```
pyinstaller --uac-admin --noconsole --onefile --add-data  
"resources/config.json;." --icon=resources/icono.ico main.py
```

o bien ejecutar este otro comando:

```
pyinstaller.exe .\main.spec
```

Esto generará un archivo (.exe) en la carpeta `dist/` del cliente, que se puede ejecutar directamente. Sin embargo, como queremos que el cliente pueda instalar en el sistema, utilizamos Inno Setup para crear un instalador.

InnoSetup funciona mediante scripts. Tenemos ya un script preparado en la carpeta `client/` que podemos utilizar. Abrimos Inno Setup y cargamos el script `gca_script.iss` que se encuentra en la carpeta `client/`.

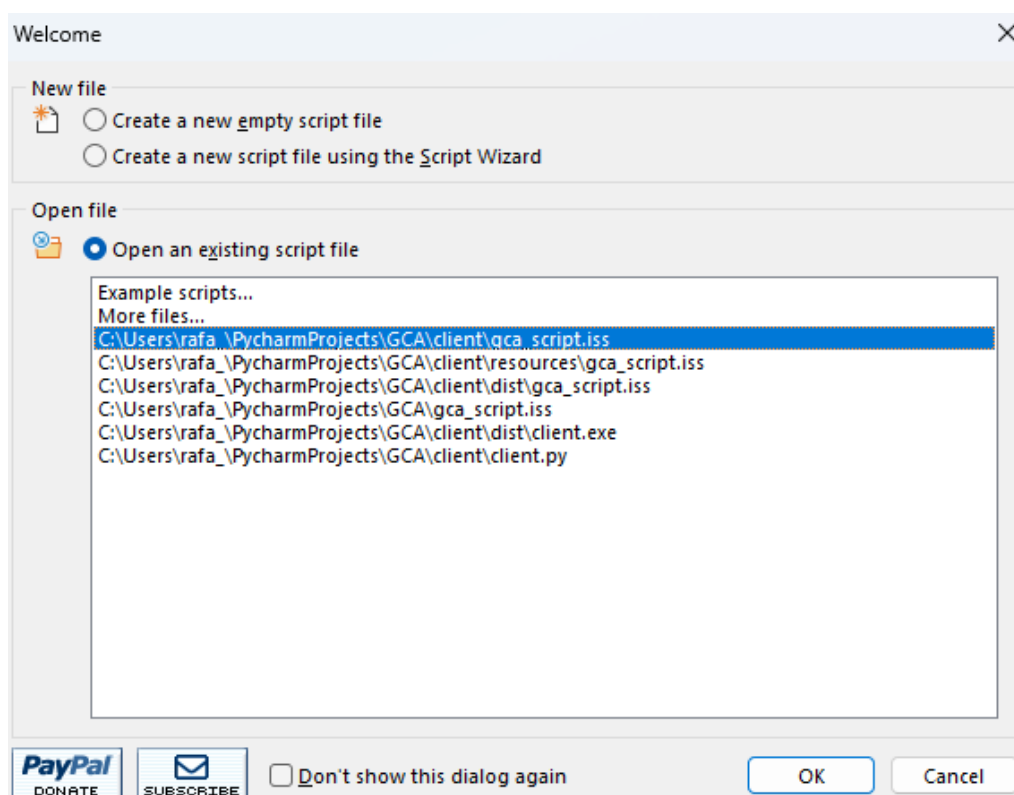


Figura D.7: Script de Inno Setup.

En el script se especifican las configuraciones del instalador, como el nombre del programa, la versión, el icono, los archivos a incluir, etc. El primer paso es compilar el script. Para ello, pulsamos en el botón de compilar en la barra de herramientas, o bien utilizamos el atajo de teclado `Ctrl + F9`.

Una vez compilado, se generará un instalador en la carpeta `output/` del cliente con el nombre `GCA_Installer.exe`.

Se pueden instalar nuevas dependencias en el cliente mediante PIP, al igual que en cualquier otro proyecto de Python. No obstante, cabe destacar que cuantas más dependencias se añadan, peor compatibilidad podrá tener el instalador con los dispositivos, pues algunas dependencias pueden requerir de librerías adicionales que no estén presentes en el sistema.

D.5. Pruebas del sistema

Las pruebas se han realizado en el backend utilizando Junit y Spring Boot Test. Todo el código de las pruebas se encuentra en la carpeta `src/test/java/` del proyecto y su configuración en la carpeta `resources/`.

Se utiliza una base de datos en memoria H2, por lo que no es necesario tener una base de datos configurada. De esta forma, las pruebas son más rápidas y no afectan a la base de datos real.

Dentro de los tests se pueden encontrar pruebas unitarias y de integración: las pruebas unitarias no necesitan levantar el contexto de la aplicación, mientras que las pruebas de integración sí lo necesitan, porque prueban la interacción entre los distintos componentes de la aplicación.

En relación a esto, se ha añadido Jacoco para la cobertura de código, que permite ver el porcentaje de código cubierto por las pruebas.

Se pueden ejecutar los tests de manera individual desde el IDE: basta con abrir las clases de tests y pulsar en el botón de ejecutar:

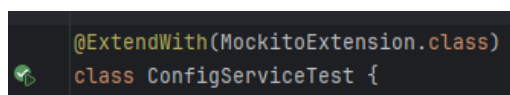


Figura D.8: Ejecución individual desde el IDE.

Si todo ha ido bien, aparece un tick verde en todos los métodos de prueba; en caso contrario, se marcan aquellos test que han fallado con una cruz roja.

Otra opción es utilizar Gradle para ejecutar todos los tests del proyecto. Para ello, abrimos una terminal en la carpeta `backend/` y ejecutamos el siguiente comando:

```
./gradlew test
```

O bien, utilizamos la herramienta de IntelliJ que tiene integrada para ejecutar las tareas Gradle. Para ello, vamos a la barra lateral derecha y buscamos la pestaña de Gradle, y allí buscamos la tarea `test` y la ejecutamos:

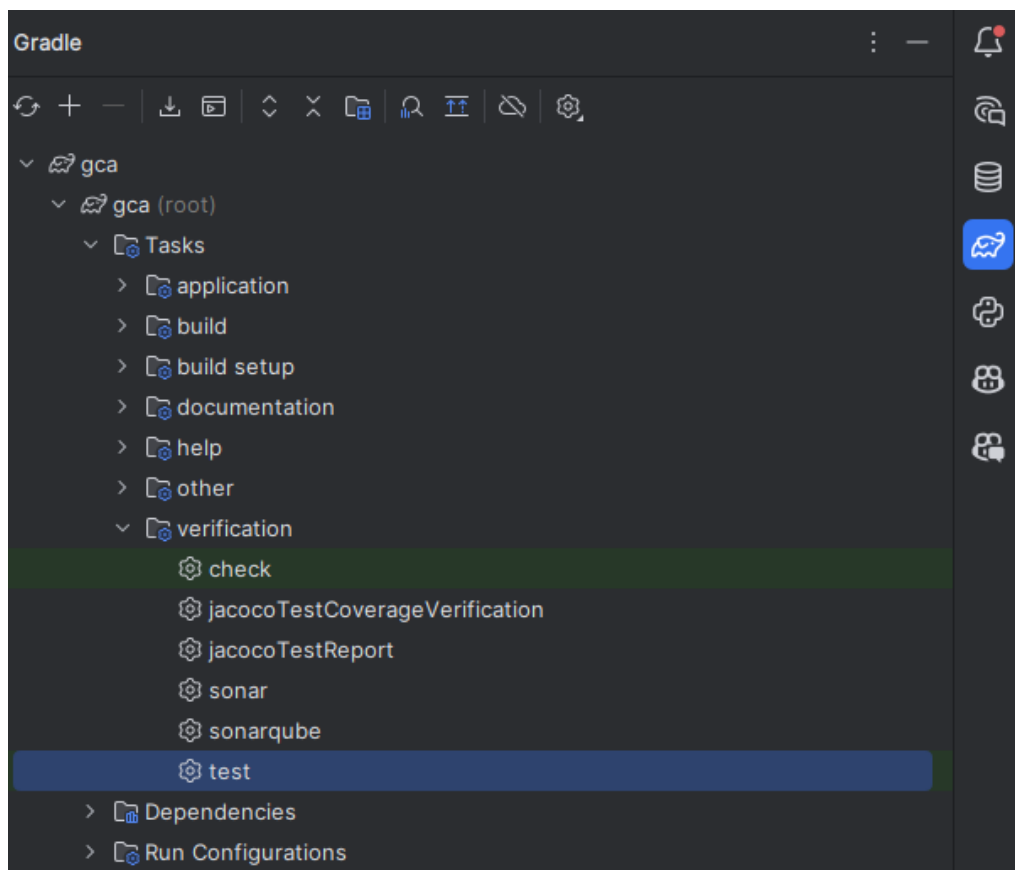


Figura D.9: Ejecución de tests desde Gradle.

Esta tarea ejecuta todos los tests del proyecto y genera un informe de los resultados en la carpeta `build/jacocoHtml`.

Si todo ha ido bien, en Spring se muestra el resultado:



Figura D.10: Resultado de pruebas.

Y podemos consultar el informe en nuestro navegador accediendo a la dirección `../backend/build/jacocoHtml/index.html`.

gca

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
com.gca.domain		80%		0%	15	93	16	137	11	89	0	9
com.gca.service.impl		96%		80%	11	110	8	337	0	82	0	7
com.gca.security		70%		n/a	2	10	10	30	2	10	0	2
com.gca.controller		94%		n/a	2	41	5	98	2	41	0	6
com.gca.dto		95%		n/a	3	59	5	91	3	59	1	11
com.gca.security.filter		95%		61%	7	20	2	40	0	11	0	1
com.gca		37%		n/a	1	2	2	3	1	2	0	1
com.gca.exception		100%		100%	0	18	0	43	0	17	0	10
com.gca.util		100%		100%	0	9	0	16	0	8	0	3
com.gca.config		100%		50%	1	5	0	12	0	4	0	1
Total	205 of 3,045	93%	27 of 88	69%	42	367	48	807	19	323	1	51

Figura D.11: Informe de pruebas.

En verde se muestran las líneas de código que han sido cubiertas por las pruebas y en rojo las que no, además del porcentaje de cobertura total del proyecto. Podemos navegar por el informe para ver en detalle la cobertura en cada clase y método (es decir, tanto las líneas de código que han sido cubiertas por las pruebas como las que no).

DefaultCipherServiceImpl

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
encrypt(String)		87%		n/a	0	1	2	12	0	1
calculateHash(String)		65%		n/a	0	1	2	5	0	1
decrypt(String)		100%		n/a	0	1	0	11	0	1
init()		100%		100%	0	2	0	5	0	1
DefaultCipherServiceImpl(SecretProperties)		100%		n/a	0	1	0	3	0	1
Total	15 of 171	91%	0 of 2	100%	0	6	4	36	0	5

Figura D.12: Ejemplo de detalle de cobertura de una clase.

```

39. public String encrypt(String value) {
40.     try {
41.         Cipher cipher = Cipher.getInstance(ALGORITHM);
42.         byte[] iv = new byte[IV_LENGTH];
43.         new SecureRandom().nextBytes(iv);
44.
45.         GCMParameterSpec spec = new GCMParameterSpec(TAG_LENGTH_BIT, iv);
46.         cipher.init(Cipher.ENCRYPT_MODE, keySpec, spec);
47.         byte[] encrypted = cipher.doFinal(value.getBytes(StandardCharsets.UTF_8));
48.
49.         byte[] combined = new byte[iv.length + encrypted.length];
50.         System.arraycopy(iv, 0, combined, 0, iv.length);
51.         System.arraycopy(encrypted, 0, combined, iv.length, encrypted.length);
52.
53.         return Base64.getEncoder().encodeToString(combined);
54.     } catch (Exception e) {
55.         throw new CipherException("Error al cifrar", CipherException.ErrorType.ENCRYPTION, e);
56.     }
57. }

```

Figura D.13: Ejemplo de detalle de cobertura de un método.

Además de todo esto, se ha utilizado SonarQube para el análisis del código. Esta herramienta se ejecuta en la nube y es válida para analizar el código que se encuentra en la rama principal del repositorio, dado que la versión utilizada es gratuita y no permite analizar ramas diferentes a la principal.

Los resultados del análisis se pueden consultar en la página de SonarQube del proyecto y muestra la siguiente información:

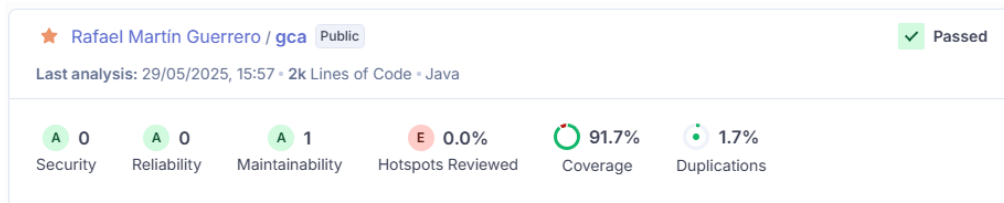


Figura D.14: Análisis de SonarQube.

Para no tener que subir constantemente el proyecto a SonarQube, se ha instalado el siguiente plugin en IntelliJ IDEA:

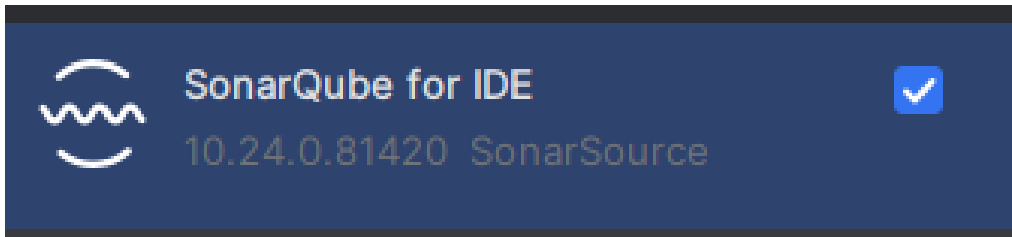


Figura D.15: Plugin de SonarQube para IntelliJ IDEA.

Esta herramienta nos permite analizar el código directamente desde el IDE y ver los resultados en tiempo real, como el que muestra la siguiente imagen:

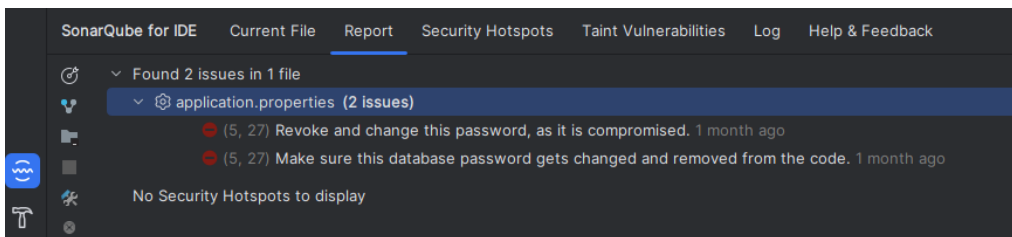


Figura D.16: Análisis de SonarQube desde el IDE.

Así, antes de subir el código al repositorio, podemos asegurarnos de que cumple con los estándares de calidad necesarios.

Por último, se ha incluido en GitHub Actions un workflow que se ejecuta cada vez que se realiza un push al repositorio; se encarga de compilar el proyecto y ejecutar los tests, y puede comprobarse directamente desde GitHub:

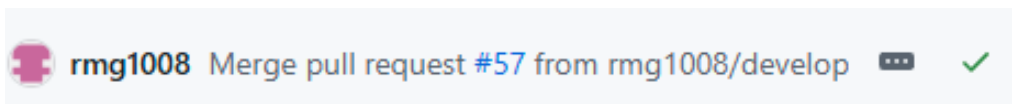


Figura D.17: GitHub Check.

Y pulsando sobre el check, podemos ver los detalles de la ejecución:

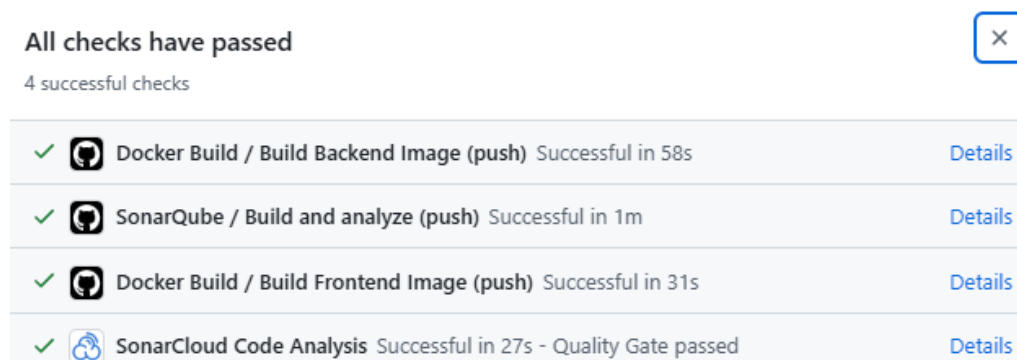


Figura D.18: Detalles del GitHub Check.

De esta manera, se consigue que cada vez que un usuario realice una aportación al repositorio, se garantice que el código cumple con los estándares de calidad y que no se introducen errores en el proyecto.

Apéndice E

Documentación de usuario

E.1. Introducción

El público al que va dirigido este manual son los usuarios de la aplicación GCA: Administradores y Gestores. Por lo que para entender su contenido, se asume que el lector tiene conocimientos en la administración de sistemas y redes. Este manual tiene como objetivo detallar todos aquellos aspectos que se consideran necesarios para los usuarios, desde la instalación y sus requisitos hasta el uso de la aplicación.

E.2. Requisitos de usuarios

Es importante mencionar que GCA utiliza dos herramientas:

- **Plataforma web**
- **Aplicación cliente**

Plataforma web

Para la utilización de la web, se requiere de:

- Navegador web moderno (Google Chrome, Microsoft Edge o Firefox actualizados a su última versión).
- Conexión a internet estable.
- Dispositivo con una resolución mínima de 1280x720 píxeles.

- Acceso a la URL de la plataforma y credenciales proporcionadas por el administrador del sistema.

Aplicación cliente

Para la aplicación de escritorio, es necesario instalar el agente en un dispositivo con sistema operativo Windows. Los requisitos son:

- Sistema operativo Windows (64 bits) o superior.
- Procesador con, al menos, dos núcleos para asegurar una ejecución fluida de los procesos en paralelo.
- Al menos 100 MB de espacio libre en el disco.
- Permisos de administrador.
- Conexión a internet para comunicarse con el servidor.
- Credenciales de acceso al sistema, las mismas que se utilizan en la web.

E.3. Instalación

Plataforma web

La plataforma web no requiere ninguna instalación, simplemente acceder a la URL proporcionada por el administrador del sistema.

Aplicación cliente

Para ello, es necesario descargar el instalador. Se puede acceder a él a través del siguiente [enlace](#) o utilizar el proporcionado por el administrador del sistema.



Figura E.1: Instalador aplicación cliente

Una vez descargado:

- Clic derecho sobre el instalador y pulsamos en “Ejecutar como administrador”: aparecerá un recuadro de confirmación para permitir su ejecución.
- Se abrirá un asistente de instalación. Escogemos la ruta para instalarlo: podemos utilizar la que viene por defecto y pulsamos en “Siguiente”.
- Elegir dónde colocar el acceso directo y pulsar “Siguiente”.
- Marcar la casilla si se quiere crear un acceso directo en el escritorio y pulsar “Siguiente”.
- Por último, cuando se muestra el resumen de la configuración selecciona, pulsar “Instalar”.
- Una vez instalado, se muestra una última ventana que nos permite lanzar directamente la aplicación.

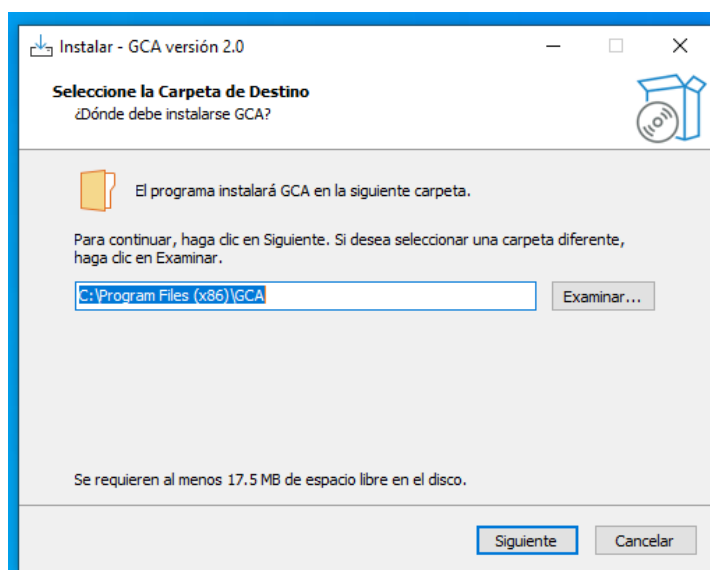


Figura E.2: Ventana para elegir la ruta de instalación

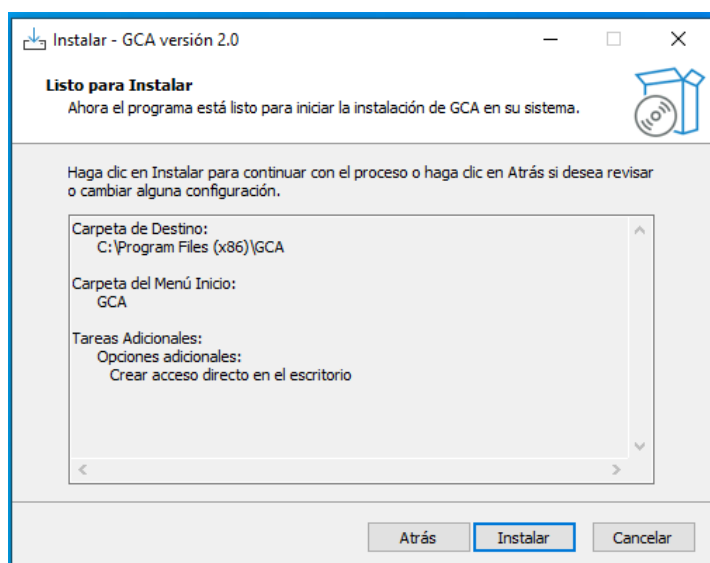


Figura E.3: Ventana resumen de la configuración seleccionada

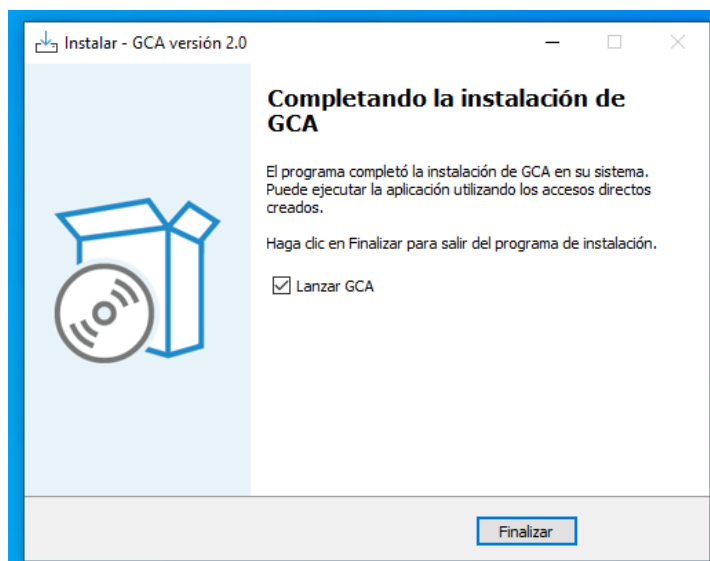


Figura E.4: Finalización de la instalación

Siguiendo estos pasos, la aplicación se instala correctamente. Sin embargo, es necesario realizar una última configuración para permitir su ejecución sin ser eliminada por Windows Defender.

- Acude a “Seguridad de Windows” - “Protección contra virus y amenazas”
- Haz clic en “Administrar configuración” y busca “Añadir o quitar exclusiones”.
- Añade la carpeta dónde se ha instalado la aplicación.
- Puedes obtener más información en el siguiente [tutorial](#).

Tras realizar la configuración, la aplicación se encuentra totalmente lista para ser ejecutada.

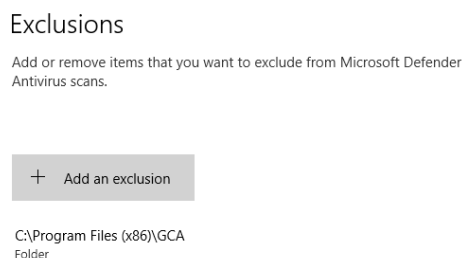


Figura E.5: Exclusión de la aplicación GCA

E.4. Manual del usuario

Se detallan los procesos necesarios para la utilización de ambas herramientas.

Plataforma web

Toda la plataforma web se divide en diferentes secciones, cada una de ellas con un propósito específico.

1. **Cabecera.** En la parte superior de la pantalla, se muestra el icono de la aplicación a la izquierda, y los enlaces a las diferentes secciones de la plataforma a la derecha.
2. **Contenido principal.** En el centro de la pantalla, se muestra el contenido principal de la página seleccionada. Dependiendo de la página, se mostrarán diferentes elementos, como tablas, formularios, buscadores, etc.
3. **Pie de página.** En la parte inferior de la pantalla, se muestra el nombre de la aplicación y la licencia bajo la que se distribuye.

El acceso a la plataforma web se realiza mediante el inicio de sesión, donde se debe introducir el correo del usuario y la contraseña.

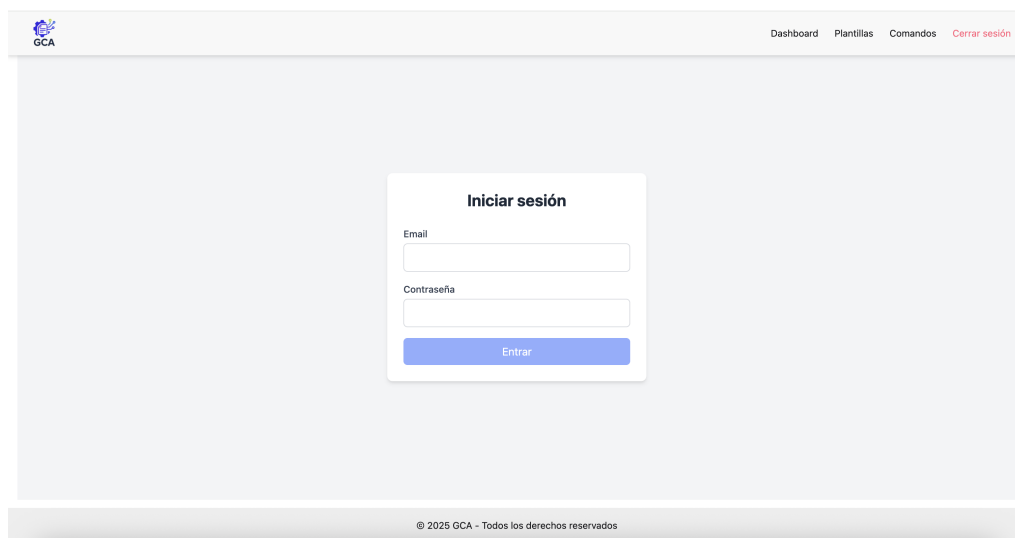


Figura E.6: Página de Inicio de Sesión de la plataforma web

Tras introducir las credenciales correctamente, se muestra la página principal de la plataforma:

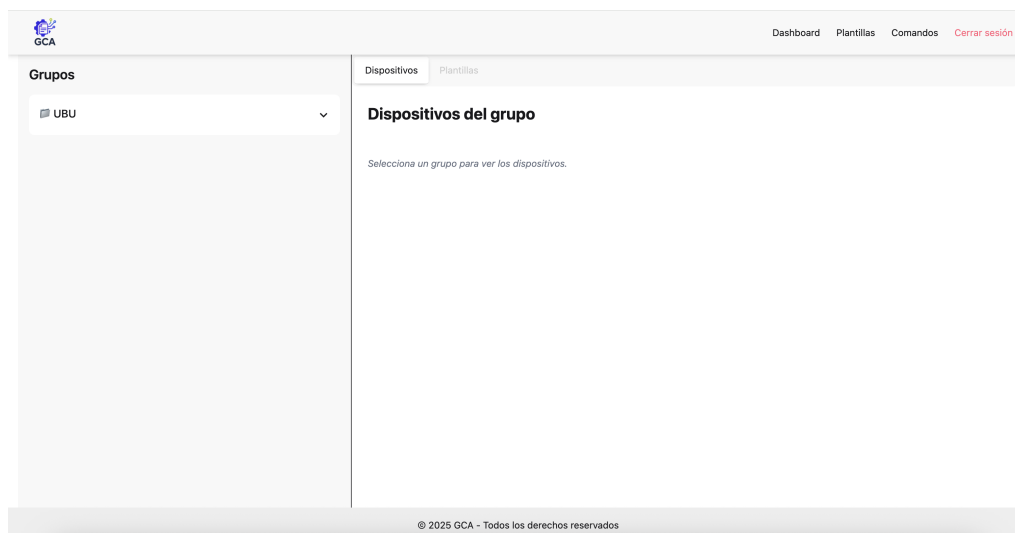


Figura E.7: Página principal de la plataforma web

Dentro del contenido principal, se observan diferentes secciones:

- **Menú lateral izquierdo.** A la izquierda de la pantalla, se muestra un menú con los diferentes grupos que conforman la plataforma. El grupo raíz “UBU” contiene a todos los demás grupos, y siempre se muestra al inicio. Para mostrar el resto de grupos (si los hay), se debe hacer clic en el desplegable, y se abrirá un listado de los diferentes grupos que tiene como hijos.
- **Sección central.** La sección central se distribuye en dos pestañas: dispositivos y plantillas. Por defecto, se muestra la pestaña de dispositivos. Si no se ha seleccionado ningún grupo o el grupo seleccionado no tiene dispositivos, se muestra una indicación de que no hay dispositivos en el grupo. Si se ha seleccionado un grupo que tiene dispositivos, se muestra una tabla con los dispositivos del grupo y diferentes opciones para interactuar con ellos. En cada pestaña se muestra la ruta completa del grupo seleccionado, desde el grupo raíz hasta el grupo actual, y se puede navegar por estos haciendo clic en el nombre del grupo.

Gestión de grupos

La gestión de grupos se realiza desde el menú lateral izquierdo, donde se pueden crear, editar y eliminar grupos. Para ello, se debe hacer clic derecho sobre un grupo y se desplegará un menú con las diferentes opciones:

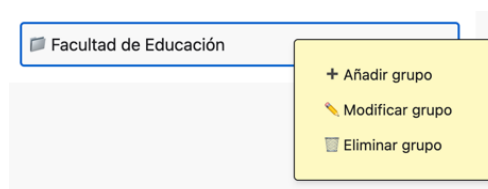


Figura E.8: Menú de gestión de grupos

- **Añadir grupo.** Inserta un nuevo grupo. Se debe introducir el nombre de este y se generará dentro del grupo seleccionado.
- **Modificar grupo.** Edita el nombre del grupo seleccionado.
- **Eliminar grupo.** Elimina el grupo seleccionado. Si este tiene dispositivos o grupos hijos, se mostrará un mensaje de error indicando que no se puede eliminar.

Para la asignación de plantillas a un grupo, hay que seleccionar la pestaña de plantillas. Si no se selecciona un grupo, la pestaña se mantiene

deshabilitada. Una vez seleccionado un grupo, si no tiene plantillas, se muestra un desplegable con las plantillas disponibles y los comandos que se ejecutarán al asignar la plantilla.

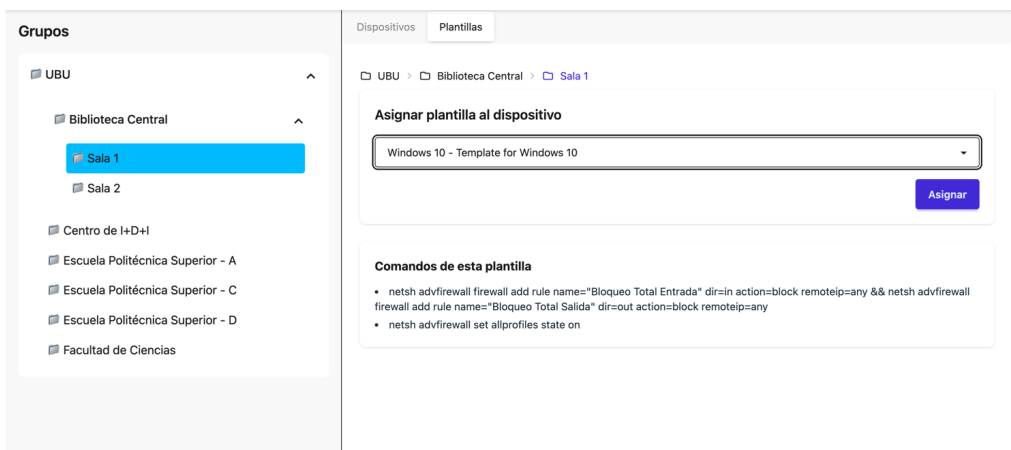


Figura E.9: Asignación de plantilla a un grupo

Si el grupo ya tiene una plantilla asignada, se muestra un botón de “Eliminar asignación”:

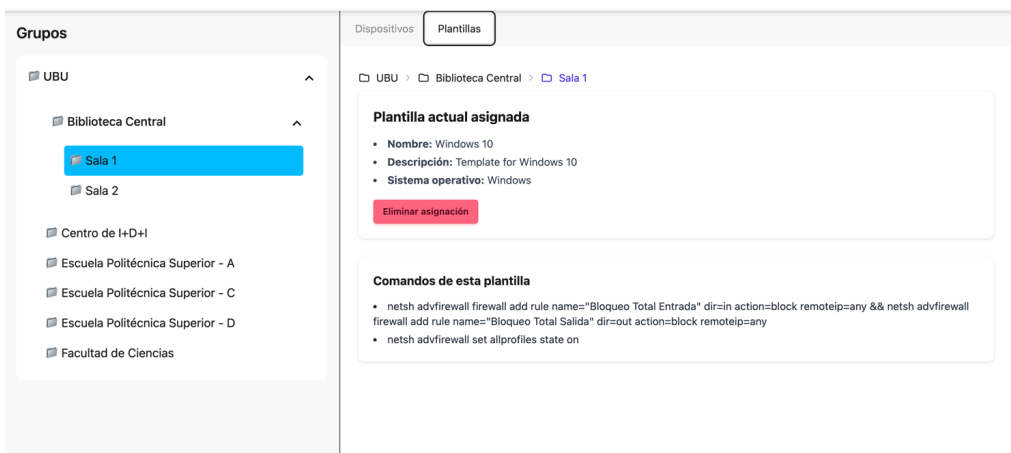


Figura E.10: Desasignación de plantilla a un grupo

Gestión de dispositivos

La gestión de dispositivos se realiza desde la pestaña de dispositivos: se muestra una tabla donde se pueden eliminar, editar y asignar plantillas a los dispositivos:



Acciones	ID	Nombre	Huella
  	1	Test	Huella1234

Figura E.11: Tabla de dispositivos



Figura E.12: Icono para eliminar un dispositivo



Figura E.13: Icono para editar un dispositivo



Figura E.14: Icono para asignar una plantilla a un dispositivo

- **Eliminar dispositivo.** Elimina el dispositivo seleccionado previa confirmación.
- **Añadir dispositivo.** Esta funcionalidad se utiliza a modo de ejemplo, ya que para añadir un dispositivo se debe utilizar la aplicación cliente.
- **Editar dispositivo** Edita la información del dispositivo seleccionado.
- **Asignar plantilla.** Asigna una plantilla al dispositivo seleccionado. Se debe seleccionar la plantilla de la lista de plantillas existentes.

- **Desasignar plantilla.** Desasigna la plantilla del dispositivo seleccionado.

El formulario para editar un dispositivo es el siguiente:

Editar dispositivo

Nombre	<input type="text" value="Test"/>
Huella	<input type="text" value="Huella1234"/>
OS	<div>Windows ▼</div>
Grupo	<div>UBU ▼</div>
<div><div>Guardar</div><div>Cancelar</div></div>	

Figura E.15: Formulario de edición de dispositivo

Se pueden editar los campos de nombre, sistema operativo y grupo al que pertenece. Una vez validados los datos, se pulsa en “Guardar” y se actualiza la tabla de dispositivos mostrando un mensaje de éxito.

Para la asignación de plantillas a un dispositivo, al hacer clic en el icono de asignar plantilla, se muestra un diálogo con una lista de las plantillas disponibles: se selecciona la plantilla deseada y se pulsa en “Asignar”:

Asignación de Plantilla

Asignar nueva plantilla al dispositivo

Windows 10 - Template for Windows 10

Comandos de esta plantilla

- netsh advfirewall firewall add rule name="Bloqueo Total Entrada" dir=in action=block remoteip=any && netsh advfirewall firewall add rule name="Bloqueo Total Salida" dir=out action=block remoteip=any
- netsh advfirewall set allprofiles state on

Cancelar
Asignar

Figura E.16: Asignación de plantilla a un dispositivo

Una vez seleccionada una plantilla, en la parte inferior del diálogo se muestra un resumen de los comandos que serán ejecutados en el dispositivo. Si no aparecen comandos es porque la plantilla no tiene ninguno asociado.

Para desasignar una plantilla, se utiliza el mismo icono de asignar plantilla: ahora se muestra un botón de “Desasignar plantilla”:

Asignación de Plantilla

Plantilla actual asignada

- **Nombre:** Windows 10
- **Descripción:** Template for Windows 10
- **Sistema operativo:** 1

Quitar asignación

Comandos de esta plantilla

- netsh advfirewall firewall add rule name="Bloqueo Total Entrada" dir=in action=block remoteip=any && netsh advfirewall firewall add rule name="Bloqueo Total Salida" dir=out action=block remoteip=any
- netsh advfirewall set allprofiles state on

Cancelar

Asignar

Figura E.17: Desasignación de plantilla a un dispositivo

Gestión de comandos











Para acceder a la gestión de comandos, se debe pulsar en el enlace de “Comandos” que se encuentra en la cabecera de la plataforma web.

Se muestra una tabla paginada con los comandos existentes y un buscador para filtrarlos por nombre o descripción. En cada fila de la tabla, se muestra un icono para editar el comando y otro para eliminarlo. Para añadir un nuevo comando, se debe pulsar en el botón “Añadir” que se encuentra en la parte superior derecha de la página.

Gestión de comandos

Añadir

Buscar comando por nombre o descripción...

Acciones	ID	Nombre	Descripción	Valor
 	1	Bloquear todo el tráfico	Bloquea todas las conexiones entrantes y salientes	netsh advfirewall firewall add rule name="Bloqueo Total Entrada" dir=in action=block remoteip=any && netsh advfirewall firewall add rule name="Bloqueo Total Salida" dir=out action=block remoteip=any
 	2	Permitir todo el tráfico	Elimina el bloqueo total de conexiones	netsh advfirewall firewall delete rule name="Bloqueo Total Entrada" && netsh advfirewall firewall delete rule name="Bloqueo Total Salida"
 	3	Bloquear acceso a IP	Bloquea todo el tráfico hacia y desde una IP específica	netsh advfirewall firewall add rule name="Bloquear IP Entrada" dir=in action=block remoteip={{direccionIP}} && netsh advfirewall firewall add rule name="Bloquear IP Salida" dir=out action=block remoteip={{direccionIP}}
 	4	Permitir acceso a IP	Permite el tráfico hacia y desde una IP específica	netsh advfirewall firewall add rule name="Permitir IP Entrada" dir=in action=allow remoteip={{direccionIP}} && netsh advfirewall firewall add rule name="Permitir IP Salida" dir=out action=allow remoteip={{direccionIP}}
 	5	Bloquear puerto	Bloquea conexiones en un puerto TCP específico	netsh advfirewall firewall add rule name="Bloquear Puerto Entrada" dir=in action=block protocol=TCP localport={{puerto}} && netsh advfirewall firewall add rule name="Bloquear Puerto Salida" dir=out action=block protocol=TCP remoteport={{puerto}}

<

1

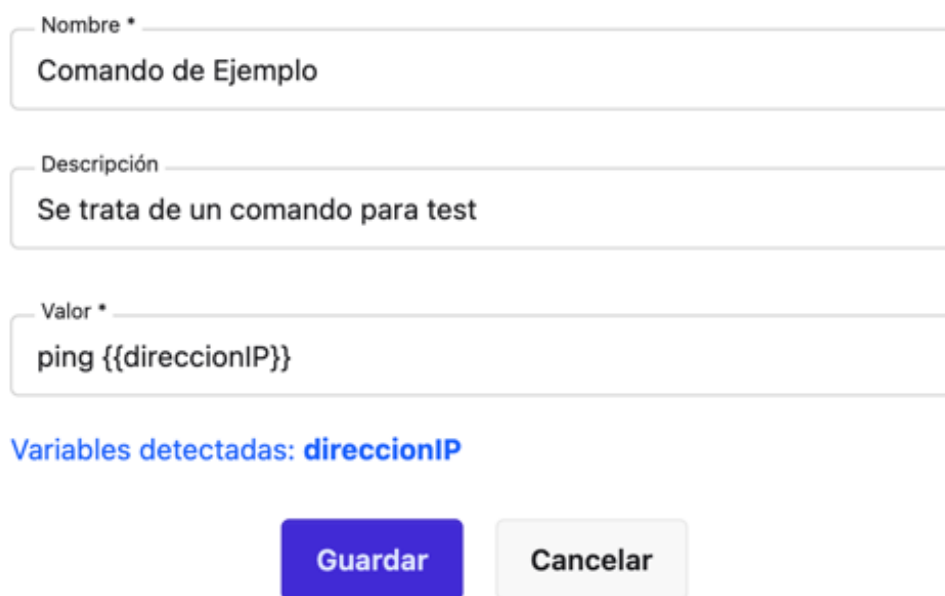
>

Figura E.18: Gestion de comandos

Tanto para añadir como para editar un comando, se muestra un formulario donde se deben introducir los siguientes campos:

- **Nombre:** nombre del comando.
- **Descripción:** descripción del comando.
- **Comando:** el comando que se pretende ejecutar en el dispositivo. Se pueden insertar variables en el comando, las cuales serán sustituidas por los valores que se definan en la plantilla a la que se asigne el comando. Deben tener el formato `{{nombre_variable}}`, donde `nombre_variable` se reemplazará por el valor correspondiente.

Nuevo comando



Nombre *

Comando de Ejemplo

Descripción

Se trata de un comando para test

Valor *

ping {{direccionIP}}

Variables detectadas: **direccionIP**

Guardar **Cancelar**

Figura E.19: Formulario para un comando

En este ejemplo, se inserta un comando de prueba que realiza un ping a una dirección IP de manera dinámica. A la hora de asignar este comando a una plantilla, se puede definir la variable `direccionIP` con el valor deseado, y el comando se ejecutará con ese valor en lugar de la variable.

El botón "Guardar" valida los datos introducidos y, si son correctos, añade o edita el comando en la tabla.

Para la eliminación de un comando, se debe pulsar en el icono de "Eliminar", y se mostrará un mensaje de confirmación para eliminar el comando. Si el comando no está asociado a ninguna plantilla, se eliminará; si lo está, se mostrará un mensaje de error.

Gestión de plantillas

La gestión de plantillas se realiza accediendo mediante el enlace de "Plantillas" en la cabecera de la plataforma web. Se comporta de manera similar a la gestión de comandos, con una tabla paginada y un buscador para filtrar las plantillas por nombre o descripción. Para añadir una nueva

plantilla, se debe pulsar en el botón “Añadir” que se encuentra en la parte superior derecha de la página. Adicionalmente, la asignación de comandos a la plantilla se realiza desde un nuevo botón que aparece en cada fila de la tabla.

Gestión de templates

Añadir

Buscar template por nombre o descripción...













Acciones	ID	Nombre	Descripción	Sistema Operativo
  	1	Windows 10	Template for Windows 10	Windows
  	2	Windows 10 - Restrict accesses	Template for Windows 10 with limited network access	Windows
  	3	Windows 11	Template for Windows 11	Windows
  	4	Windows 11 - Without network	Template for Windows 11 that disables network access	Windows

Figura E.20: Gestion de plantillas

El formulario para añadir o editar una plantilla se compone de los siguientes campos:

- **Nombre:** nombre de la plantilla.
- **Descripción:** descripción de la plantilla.
- **Sistema Operativo:** sistema operativo sobre el que se ejecutará la plantilla.

Nuevo template

Nombre *

Plantilla de prueba

Descripción

Simplemente se trata de una prueba

OS

Windows

▼

Guardar

Cancelar

Figura E.21: Creación de una plantilla

Mediante el botón “Guardar“, los datos son validados y almacenados, y se muestra un mensaje de éxito.

Para eliminar una plantilla, es necesario que no se encuentre asignada a un dispositivo o grupo; si estuviese asignada, se mostrará un mensaje de error.

La asignación de comandos a una plantilla existente se realiza en una página aislada, de manera que el usuario se pueda enfocar únicamente en la tarea. Este botón se encuentra disponible en cada registro de la tabla de plantillas.



Figura E.22: Icono para asignar comandos a una plantilla

La asignación de comandos consta de dos columnas:

- **Comandos disponibles.** Aquí se muestran todos los comandos que pueden aplicarse a la plantilla. Se pueden realizar búsquedas tal y como se hace para los comandos y plantillas. Para añadir un comando y asignarlo a la plantilla, basta con pulsar sobre el botón que tiene cada comando con el símbolo "+" y automáticamente se desplaza a la siguiente columna.
- **Comandos asignados.** Aquellos comandos que se encontraban previamente asignados aparecerán en esta columna. Cuando se añaden desde la anterior columna, se colocarán al final. No obstante, mediante las flechas ↑ y ↓ puede modificarse el orden en el que se ejecutarán los comandos, se ejecutan de arriba hacia abajo. Por último, se puede eliminar un comando pulsando sobre el botón que aparece al lado de las flechas.

Asignar comandos a la plantilla

Volver

Comandos disponibles

Comandos asignados

Buscar comandos...

Bloquear todo el tráfico

Bloquea todas las conexiones entrantes y salientes

Permitir todo el tráfico

Elimina el bloqueo total de conexiones

Bloquear acceso a IP

Bloquea todo el tráfico hacia y desde una IP específica

Permitir acceso a IP

Permite el tráfico hacia y desde una IP específica

Bloquear puerto

Bloquea conexiones en un puerto TCP específico

Anterior

Siguiente

Guardar

Figura E.23: Asignar comandos a una plantilla

Permitir acceso a IP

^

v

Permite el tráfico hacia y desde una IP específica

direccionIP*

192.168.1.145

Comando generado: netsh advfirewall firewall add rule name="Permitir IP Entrada" dir=in action=allow remoteip=192.168.1.145 && netsh advfirewall firewall add rule name="Permitir IP Salida" dir=out action=allow remoteip=192.168.1.145

Activar cortafuegos

^

v

Activa el cortafuegos de Windows en todos los perfiles (dominio, privado, público)

Comando generado: netsh advfirewall set allprofiles state on

Figura E.24: Comandos asignado a una plantilla

En aquellos comandos que posean variables dinámicas, deberá añadirse obligatoriamente un valor que sustituya a la variable; aparecerá además el comando resultante.

Cuando hayamos realizado los cambios necesarios, pulsamos sobre “Guardar” para aplicar los cambios.

Aplicación cliente

Al iniciar la aplicación (con permisos de administrador) por primera vez, cuando dispositivo aún no se encuentra registrado, se mostrará una ventana para iniciar sesión y un recuadro que muestra los logs de la aplicación en tiempo real.

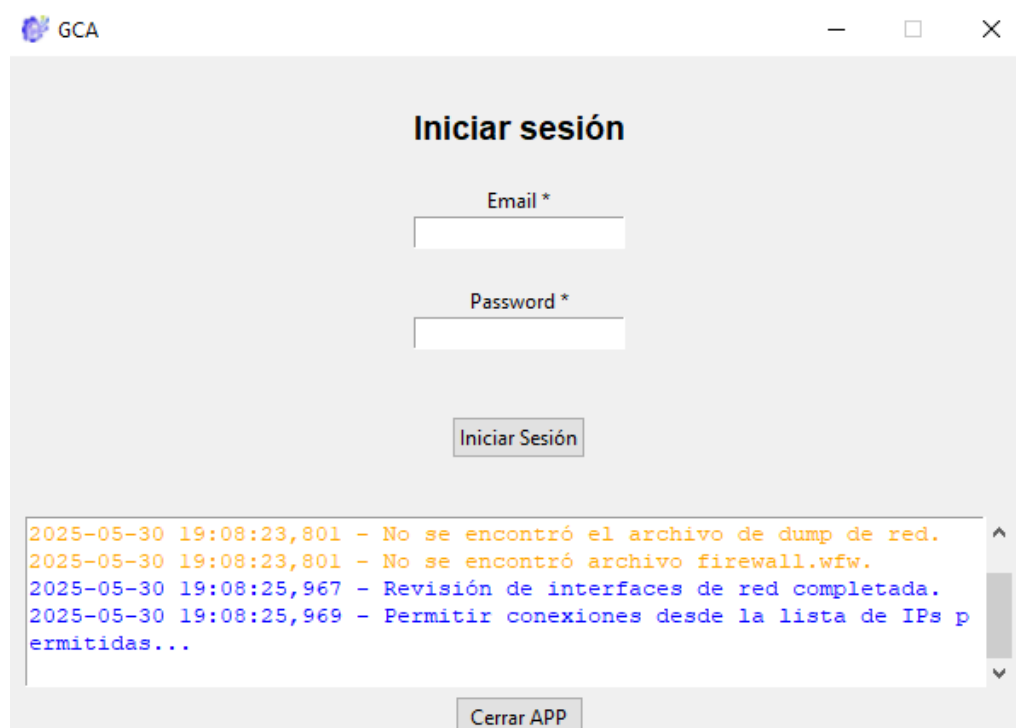


Figura E.25: Inicio de la aplicación cliente

Cada vez que se inicia la aplicación, se restaura la configuración base y se aplica la configuración más actual para el dispositivo, siempre y cuando esté registrado.

Dar de alta a un dispositivo

Para dar de alta un dispositivo es necesario iniciar sesión, de forma que se habilitan los botones para iniciar el proceso. Si las credenciales son correctas, se mostrará un mensaje y se carga el panel de usuario.

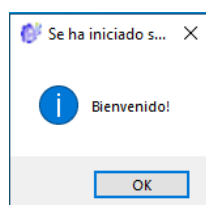


Figura E.26: Mensaje de inicio de sesión



Figura E.27: Panel de usuario

Cuando el equipo no se encuentre registrado, se muestra el botón “Registrar equipo”. Una vez realizado el registro (si se ha realizado de manera satisfactoria), se informa al usuario de la huella con la que ha sido dado de alta.

Tras realizar el alta, se realiza una copia de seguridad de las configuraciones de firewall y de red del dispositivo para ser restaurada al eliminar el equipo o tras una nueva aplicación de una plantilla.

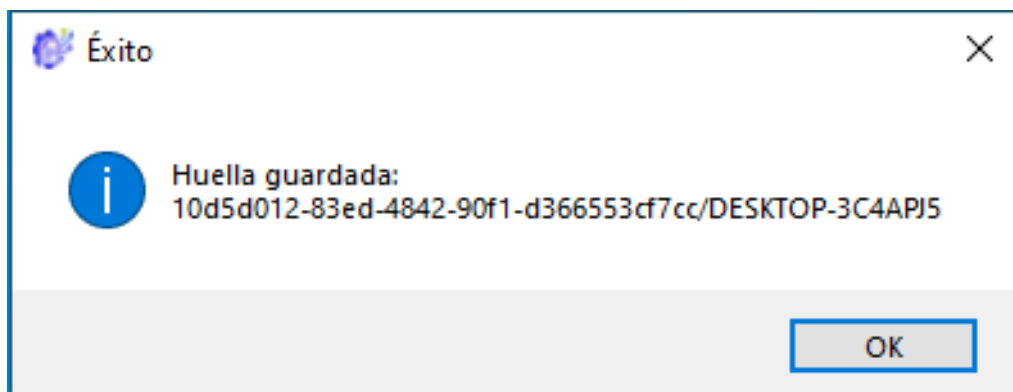


Figura E.28: Mensaje de alta del equipo

De forma paralela, la aplicación comprobará periódicamente si hay configuraciones que aplicar. Este registro queda reflejado en el recuadro de logs de la aplicación, que siempre se mantiene activo.

Para comprobar que se ha registrado correctamente, accedemos a la plataforma web, y en el grupo raíz “UBU” debería aparecer el dispositivo:

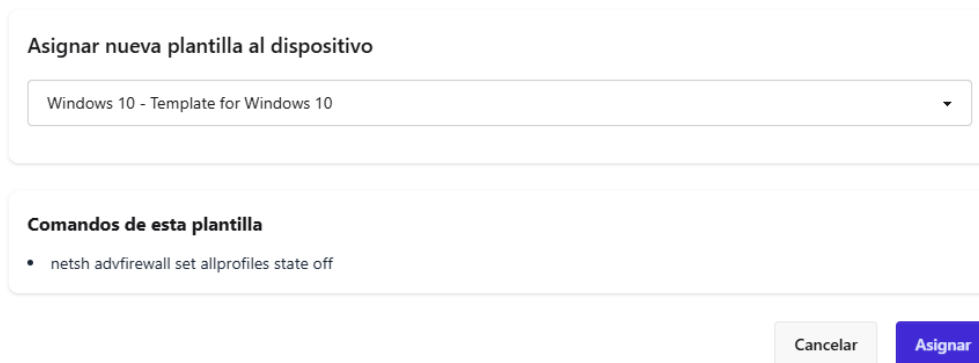


Figura E.29: Dispositivo dado de alta desde el cliente

Asignar plantilla a un dispositivo

Seleccionamos el dispositivo registrado, pulsamos en el botón de asignación de plantilla y escogemos la que queremos aplicar:

Asignación de Plantilla



Asignar nueva plantilla al dispositivo

Windows 10 - Template for Windows 10

Comandos de esta plantilla

- netsh advfirewall set allprofiles state off

Cancelar Asignar

Figura E.30: Asignación de una plantilla al dispositivo registrado

Mediante la plantilla que se ha escogido, deshabilitaremos el firewall del dispositivo. Hecho esto, el cliente se encargará de aplicar la configuración al equipo.

La aplicación registra la ejecución de comandos y lista aquellos que se han ejecutado:

```
2025-05-30 19:44:41,460 - Comando 1: netsh advfirewall set allprofiles state off
2025-05-30 19:44:41,634 - El comando se ha ejecutado correctamente.
```

Figura E.31: Registro de un comando ejecutado satisfactoriamente

Además, podemos comprobar en el equipo que el firewall se encuentra desactivado:

🔒 Firewall & network protection

Who and what can access your networks.

❌ Microsoft Defender Firewall is using settings that may make your device unsafe.

Restore settings

🏢 Domain network

Firewall is off.

Turn on

🏠 Private network

Firewall is off.

Turn on

🌐 Public network (active)

Firewall is off.

Turn on

Figura E.32: Firewall desactivado

Al aplicar la plantilla, el dispositivo registra el instante en el que fue aplicada, de forma que, mientras no se actualize la plantilla en la aplicación web, no se aplicará una nueva configuración. Del mismo modo, si asignamos una nueva plantilla al dispositivo, se ejecutarán los comandos de esta última. Ante cualquier cambio de comandos y plantillas, se restaura la copia de seguridad que se realizó previamente en el registro.

Dar de baja un dispositivo

El proceso de baja es similar al de alta: requiere iniciar sesión y que el dispositivo se encuentre registrado. Si se cumplen estas condiciones, podemos pulsar en el botón “Eliminar dispositivo” y confirmar la eliminación:

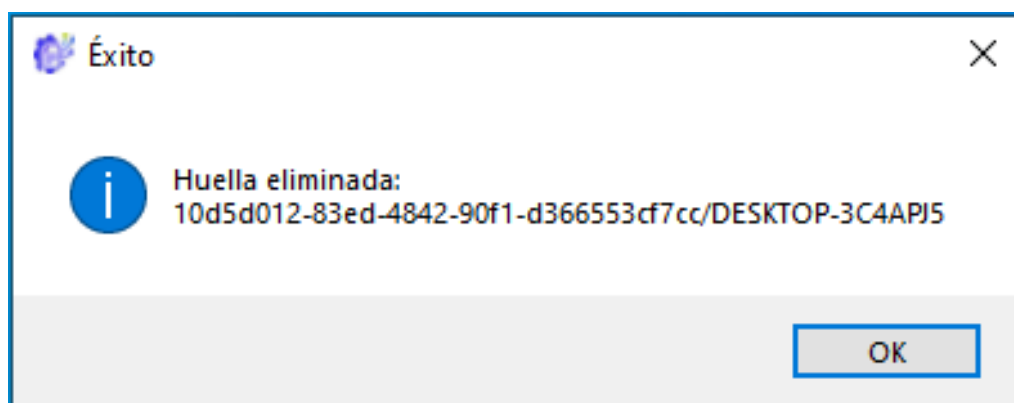


Figura E.33: Mensaje de baja del equipo

De esta manera, se borran todos los registros del equipo, tanto en el propio dispositivo como en la base de datos, y se restaura la configuración inicial. Como la plantilla que deshabilitaba el firewall se elimina, el firewall vuelve a estar activo.

Consideraciones a tener en cuenta

Existen comandos que deshabilitan el acceso a Internet en su totalidad, por lo que el acceso a la API se vuelve imposible. Para volver a tener acceso, al iniciar la aplicación nuevamente, se restaura la configuración de red y vuelve a comprobar qué plantilla tiene asignada: si la plantilla asignada sigue deshabilitando la conexión, volverá a aplicarse nuevamente.

Aunque se deshabilite el tráfico entrante y saliente, la conexión a la API estará siempre habilitada para que el dispositivo pueda seguir actualizando su configuración.

Anexo de sostenibilización curricular

F.1. Introducción

La implementación del proyecto podría tener un impacto social relevante al reducir la necesidad de personal en tareas manuales, lo que afectaría a ciertos puestos de trabajo. Por ello, se propone acompañar esta transición con formación continua y una comunicación interna transparente que facilite la adaptación del personal.

A nivel ambiental, se ha prestado especial atención al consumo energético del sistema, adoptando medidas como la reducción de consultas innecesarias a la API, la optimización de recursos y el uso de formatos de datos ligeros, con el fin de minimizar su impacto y garantizar una implementación más eficiente y sostenible.

Bibliografía

- [1] Apache Software Foundation. Apache license, version 2.0, 2025. Disponible en: <http://apache.org/licenses/LICENSE-2.0>. Accessed 19-04-2025.
- [2] Atlassian. ¿qué es scrum?, 2024. Disponible en: <https://www.atlassian.com/es/agile/scrum>. Accessed 18-04-2025.
- [3] Creative Commons. Attribution 4.0 international (cc by 4.0), 2013. Disponible en: <https://creativecommons.org/licenses/by/4.0/>. Accessed 19-04-2025.
- [4] Creative Commons. Attribution-noncommercial 4.0 international (cc by-nc 4.0), 2013. Disponible en: <https://creativecommons.org/licenses/by-nc/4.0/>. Accessed 19-04-2025.
- [5] Glassdoor. Sueldo de full stack developer, 2025. Disponible en: https://www.glassdoor.es/Sueldos/full-stack-developer-sueldo-SRCH_K00,20.htm. Accessed 18-04-2025.
- [6] JetBrains. Planes y precios de productos jetbrains, 2025. Disponible en: <https://www.jetbrains.com/es-es/store/?section=commercial&billing=monthly>. Accessed 18-04-2025.
- [7] Ken Schwaber and Jeff Sutherland. The scrum guide – the definitive guide to scrum: The rules of the game, 2020. Disponible en: <https://scrumguides.org/scrum-guide.html>. Accessed 18-04-2025.
- [8] Seguridad Social. Cotización a la seguridad social de los trabajadores, 2025. Disponible en: <https://www.seg-social.es/wps/portal/wss/>

[internet/Trabajadores/CotizacionRecaudacionTrabajadores/36537](#). Accessed 18-04-2025.

- [9] Spring Framework. Spring mvc, 2025. Disponible en: <https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/mvc.html>. Accessed 31-05-2025.
- [10] TLDRLegal. Tldrlegal - understand software licenses in plain english, 2025. Disponible en: <https://www.tldrlegal.com/>. Accessed 19-04-2025.