



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**GCA**



Presentado por Rafael Martín Guerrero  
en Universidad de Burgos — 26 de mayo  
de 2025

Tutor: Pedro Renedo Fernández





UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



D. nombre tutor, profesor del departamento de nombre departamento, área de nombre área.

Expone:

Que el alumno D. Rafael Martín Guerrero, con DNI 80087034E, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado GCA.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 26 de mayo de 2025

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. nombre tutor

D. nombre co-tutor



## **Resumen**

Este proyecto tiene como objetivo desarrollar e implementar una plataforma web y un agente que permitan la gestión centralizada de la configuración de red en los equipos de las aulas. La solución permite la segmentación de los dispositivos en grupos y la aplicación de restricciones personalizadas para el acceso a recursos internos y externos, incluyendo Internet. Se busca optimizar el control y administración de la red educativa, mejorando la seguridad y la eficiencia en la gestión de los permisos de acceso.

## **Descriptores**

Plataforma web, configuración de red, administración educativa, agente, control, segmentación, aplicación java

## **Abstract**

The objective of this project is to develop and implement a web platform and an agent that allows the centralized management of the network configuration in the classrooms. The solution allows the segmentation of devices into groups and the application of customized restrictions for access to internal and external resources, including the Internet. The aim is to optimize the control and administration of the educational network, improving security and efficiency in the management of access permissions.

## **Keywords**

Web platform, network configuration, educational management, agent, control, segmentation, java application

---

# Índice general

---

|  |           |
|--|-----------|
| Índice general   | iii       |
| Índice de figuras  | v         |
| Índice de tablas   | vi        |
| <b>1. Introducción</b>   | <b>1</b>  |
| <b>2. Objetivos del proyecto</b>                                   | <b>3</b>  |
| 2.1. Objetivos generales . . . . .                                 | 3         |
| 2.2. Objetivos técnicos . . . . .                                  | 3         |
| 2.3. Objetivos personales . . . . .                                | 4         |
| <b>3. Conceptos teóricos</b>                                       | <b>5</b>  |
| 3.1. Comunicación mediante APIs REST . . . . .                     | 5         |
| 3.2. Gestión de configuraciones . . . . .                          | 6         |
| 3.3. Comandos y protocolos para la gestión del acceso a la red . . | 7         |
| <b>4. Técnicas y herramientas</b>                                  | <b>11</b> |
| 4.1. Metodologías . . . . .  | 11        |
| 4.2. Gestión del proyecto . . . . .                                | 12        |
| 4.3. Lenguajes de programación . . . . .                           | 13        |
| 4.4. Arquitectura . . . . .  | 15        |
| 4.5. Herramientas de desarrollo . . . . .                          | 15        |
| 4.6. Integración continua . . . . .                                | 16        |
| 4.7. Documentación . . . . .                                       | 17        |
| 4.8. Librerías . . . . .   | 17        |

|   |           |
|---|-----------|
| <i>ÍNDICE GENERAL</i>                                     | IV        |
| <b>5. Aspectos relevantes del desarrollo del proyecto</b> | <b>19</b> |
| <b>6. Trabajos relacionados</b>                           | <b>20</b> |
| 6.1. Artículos científicos . . . . .                      | 20        |
| 6.2. Herramientas . . . . .                               | 21        |
| 6.3. Fortalezas y debilidades del proyecto . . . . .      | 22        |
| <b>7. Conclusiones y Líneas de trabajo futuras</b>        | <b>23</b> |
| <b>Bibliografía</b>                                       | <b>24</b> |



---

## Índice de figuras

---

---

# Índice de tablas

---

|  |    |
|--|----|
| 6.1. Comparativa de las herramientas de los proyectos con codificación de colores. . . . . | 21 |
| 6.2. Fortalezas y Debilidades del Proyecto . . . . .                                       | 22 |

---

# 1. Introducción

---

La Universidad de Burgos (UBU) cuenta con más de 1000 equipos informáticos distribuidos en aulas entre 8 y 10 centros. Cada una de estas aulas dispone de dispositivos destinados a satisfacer una serie de necesidades distintas, como por ejemplo, la realización de operaciones matemáticas complejas, redacción de documentos, edición de contenido multimedia, entre otras.

Sin embargo, un mismo equipo no está destinado a un único uso y con las aplicaciones instaladas pueden permitir el acceso a otros recursos y dispositivos no deseados, esto supone un riesgo para la seguridad de la información y puede derivar en un uso indebido, como acceder a información no autorizada durante una prueba de evaluación.

En los últimos años, se ha observado un incremento de las peticiones entre los docentes, para limitar los recursos a los que se pueden acceder desde los ordenadores. Por este motivo, surge la necesidad de implementar una gestión centralizada en las aulas, que permita, de una manera sencilla, controlar los accesos de los distintos equipos, atendiendo a características diferentes, como nombres de equipo, aulas, centros. . .

De acuerdo con Microsoft[22] existen cuatro formas de control de acceso:

- Control de acceso discrecional (DAC): un propietario concede acceso a otros usuarios.
- Control de acceso basado en roles (RBAC): se otorga acceso según funciones empresariales definidas.
- Control de acceso obligatorio (MAC): se concede acceso en forma de autorización regulado por una autoridad central.

- Control de acceso basado en atributos (ABAC): se da de manera flexible en función de variables y condiciones.

Actualmente, los usuarios de los equipos, mediante el protocolo 802.1X[40] han de ingresar sus credenciales y tras una autenticación satisfactoria, se les asigna una dirección IP, que les permite hacer uso del dispositivo, navegar por Internet, acceder a los distintos recursos de red. . . .

Existen herramientas que pueden ayudar a limitar el acceso a los recursos, como software específico, no obstante, estas opciones pueden resultar costosas. También se pueden configurar uno a uno los permisos de cada dispositivo, pero ello requiere que manualmente, un administrador tenga que cambiar los distintos permisos, y llevar un control de cada uno de ellos.

La plataforma propuesta, busca optimizar costes y ofrecer una solución práctica, diseñada a medida para la UBU. A diferencia de otras propuestas, esta iniciativa se enfoca en resolver problemas reales sin añadir funciones innecesarias que acaban dificultando el uso. Para su implementación, solo requiere un servidor web encargado de gestionar las configuraciones y una puesta a punto inicial en cada equipo para su registro en el sistema. Posteriormente, un software se encargará de aplicar las configuraciones necesarias en cada dispositivo de forma automática.

---

## 2. Objetivos del proyecto

---

Existen diferentes objetivos que han derivado en el nacimiento del proyecto:

### 2.1. 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.

### 2.2. Objetivos técnicos

- Crear una aplicación web para permitir la gestión de los dispositivos de una manera sencilla e intuitiva.
- Desarrollar una aplicación Java para gestionar las configuraciones a realizar.
- Utilizar herramientas de control de versiones como Git y GitHub.
- Utilizar una arquitectura MVC (Modelo-Vista-Controlador).
- Aplicar una metodología Ágil (Scrum) en el desarrollo del proyecto.
  - Utilizar ZenHub para gestión de proyectos.

- Aplicar TDD (Test Driven Development) durante todo el desarrollo, para garantizar calidad en el producto.

## **2.3. Objetivos personales**

- Aportar una herramienta beneficiosa a la Universidad.
- Aplicar los conocimientos aprendidos durante la ingeniería.
- Ampliar conocimientos técnicos:
  - Gestión de dispositivos.
  - Java.
  - Angular.
  - Herramientas de testeo.

---

## 3. Conceptos teóricos

---

El funcionamiento del proyecto se divide en varias fases.

1. **Registro de dispositivos:** cada dispositivo se registra usando la aplicación cliente y se mantiene a la espera de recibir configuraciones.
2. **Configuración inicial:** una vez registrado, el gestor asigna una localización específica al dispositivo
3. **Asignación de configuraciones:** se asignan configuraciones a cada dispositivo o al grupo que pertenece.
4. **Aplicación de configuraciones:** mediante la aplicación cliente, se obtiene la configuración asignada para el dispositivo y se aplican los comandos correspondientes.

Para entender mejor cómo se desarrollan cada fase, es recomendable explicar algunos conceptos teóricos relacionados con la gestión de configuraciones, protocolos de red y herramientas de administración.

### 3.1. Comunicación mediante APIs REST

Las comunicaciones entre el cliente y el servidor, así como entre el servidor y la interfaz web, se realizan mediante APIs REST.

Una API REST (Representational State Transfer) es un conjunto de definiciones y protocolos utilizados para integrar sistemas entre aplicaciones[15]. Permite realizar comunicaciones entre consumidores (aquellos que realizan las llamadas) y proveedores (aquellos que reciben la llamada y devuelven

la respuesta). La información se transmite en formato JSON (JavaScript Object Notation), que es un formato de intercambio de datos ligero y fácil de leer. Esta comunicación utiliza el protocolo HTTP (Hypertext Transfer Protocol), que es el protocolo de transferencia de hipertexto utilizado en la World Wide Web. HTTP utiliza una serie de métodos para definir las diferentes acciones, los más comunes son:

- **GET:** Recupera información del servidor.
- **POST:** Envía datos al servidor para crear un nuevo recurso.
- **PUT:** Actualiza un recurso existente en el servidor.
- **DELETE:** Elimina un recurso del servidor.

## Autenticación

Para la autenticación de los usuarios, se utilizan tokens JWT (JSON Web Tokens)[11]. Estos tokens son objetos que contienen información codificada de datos del usuario así como la fecha de expiración y permite transmitir información entre cliente y servidor de forma segura, ya que pueden ser firmados y verificados.

La estructura de un token JWT consta de tres partes:

- **Header:** Contiene información sobre el tipo de token y el algoritmo utilizado para firmarlo.
- **Payload:** Contiene la información del usuario y los datos adicionales.
- **Signature:** Se utiliza para verificar que el emisor del token es quien dice ser y para asegurar que no ha sido modificado.

El token se envía en la cabecera de la solicitud HTTP, lo que permite al servidor verificar la autenticidad del usuario y otorgar acceso (o no) al recurso solicitado.

## 3.2. Gestión de configuraciones

Para la gestión de configuraciones utilizamos una herramienta propia, la cual se puede instalar en cualquier sistema operativo, ya sea Windows, Linux o MacOS . Esta herramienta se mantendrá ejecutándose en segundo



plano de manera ininterrumpida, y se encargará de solicitar a la API las configuraciones que le correspondan.

Se gestiona a través de una interfaz web, mediante la cual se crean comandos, se agrupan en plantillas y se aplican a cada dispositivo de forma individual o grupos de dispositivos. Las plantillas incluyen un orden de ejecución de los comandos, lo que permite establecer prioridades. Si una plantilla es aplicada a un grupo de dispositivos, y otra plantilla es aplicada a un dispositivo de ese mismo grupo, la plantilla del dispositivo tiene prioridad sobre la plantilla del grupo.

### 3.3. Comandos y protocolos para la gestión del acceso a la red

Para la gestión de acceso a la red en sistemas operativos Windows, se utilizan comandos y protocolos específicos.

#### Netsh

Netsh (Network Shell) es una herramienta de línea de comandos de Microsoft que permite la configuración y gestión de dispositivos de red en sistemas operativos Windows.

Su sintaxis es la siguiente:

```
netsh [-a <Aliasfile>]
      [-c <Context>]
      [-r <Remotecomputer>]
      [-u [<domainname>\<username>]
      [-p <Password> |
      [{<NetshCommand> | -f <scriptfile>}]
```

Una explicación mas detallada puede encontrarse en la documentación oficial de Microsoft[23].

#### Firewall

El firewall[6] es un sistema de seguridad de red que actúa como una barrera entre una red interna y externa. Su principal función es controlar el tráfico de red, permitiendo o bloqueando paquetes de datos según un conjunto de reglas de seguridad. Los firewalls pueden implementarse en

hardware, software o de manera híbrida. En el caso de Windows, el firewall se implementa como un software integrado dentro del sistema operativo. En nuestro caso particular, administramos el el firewall de Windows mediante la consola de comandos:

```
netsh advfirewall firewall
[add | delete | set | show]
[<RuleName>]
[<RuleType>]
[<LocalIP>]
[<RemoteIP>]
[<LocalPort>]
[<RemotePort>]
[<Protocol>]
[<Action>]
[<Direction>]
[<Program>]
[<Service>]
[<InterfaceType>]
```

Mediante la ejecución de este comando, se pueden añadir, eliminar o modificar reglas del firewall, por ejemplo, habilitar o deshabilitar un puerto específico, bloquear un programa o servicio, e incluso deshabilitar el firewall [26]

## PowerShell

PS (PowerShell)[30] es una solución para la automatización de tareas multiplataforma, desarrollada por Microsoft. Combina un shell de línea de comandos, un lenguaje de scripting y un marco de administración de configuración.

La sintaxis de un comando de PowerShell es la siguiente:

```
Get-Command [[-ArgumentList] <Object[]>] [-Verb <string[]>] [-Noun <string[]>]
[-Module <string[]>] [-FullyQualifiedModule <ModuleSpecification[]>]
[-TotalCount <int>] [-Syntax] [-ShowCommandInfo] [-All] [-ListImported]
[-ParameterName <string[]>] [-ParameterType <PSTypeName[]>]
[<CommonParameters>]
```

```
Get-Command [[-Name] <string[]>] [[-ArgumentList] <Object[]>]
[-Module <string[]>] [-FullyQualifiedModule <ModuleSpecification[]>]
[-CommandType <CommandTypes>] [-TotalCount <int>] [-Syntax] [-ShowCommandInfo]
[-All] [-ListImported] [-ParameterName <string[]>]
[-ParameterType <PSTypeName[]>] [-UseFuzzyMatching]
[-FuzzyMinimumDistance <uint>] [-UseAbbreviationExpansion]
[<CommonParameters>]
```

Puede ser ejecutado mediante la consola de windows, o mediante un script. En concreto, para este proyecto, se utilizará para automatizar la configuración de sistemas y aplicaciones e implementar políticas de seguridad y configuración de red.

## Protocolos

Los protocolos de red son un conjunto de reglas y convenciones que permiten la comunicación entre dispositivos dentro de una red. A continuación se describen aquellos más relevantes en el contexto del presente proyecto:

- **TCP/IP:** conjunto de protocolos fundamentales que permite la comunicación entre dispositivos en red. Incluye mecanismos para diferenciar aplicaciones mediante puertos. Cualquier configuración de red o del firewall afecta directamente a este protocolo. [46]
- **HTTPS:** extensión segura del protocolo HTTP que utiliza SSL/TLS para cifrar la comunicación entre cliente y servidor. Es utilizado por la aplicación para el registro de dispositivos y la consulta de configuraciones a través de la API[44]
- **DNS:** sistema encargado de traducir nombres de dominio (más fáciles de recordar) a direcciones IP . Desde comandos como **netsh** es posible añadir o eliminar entradas relacionadas con el DNS[43]
- **DHCP:** protocolo que asigna direcciones IP dinámicas a dispositivos en una red. Aunque es utilizado por los dispositivos de la Universidad, no se contempla su modificación dentro del proyecto[27].
- **ICMP:** protocolo empleado para el envío de mensajes de diagnóstico y control, como ocurre con el comando **ping**. Su tráfico puede ser habilitado o bloqueado mediante reglas del firewall[45].

- **SMB:** protocolo de nivel de aplicación que permite compartir archivos, impresoras y otros recursos dentro de una red. Mediante SMB, se puede restringir el acceso a determinados recursos por dispositivo[38]
- **ARP:** protocolo responsable de resolver direcciones IP en direcciones MAC dentro de una red local. Aunque es más avanzado, se puede utilizar para establecer restricciones a nivel de dispositivos concretos[42]

---

## 4. Técnicas y herramientas

---

### 4.1. Metodologías

#### Scrum

La metodología ágil utilizada ha sido Scrum[34], mediante la cual se ha organizado el trabajo en interacciones cortas llamadas sprints. En cada uno de estos se establecen una serie de tareas, ordenadas por prioridad.

Este enfoque permite una mayor flexibilidad, ya que no se necesita tener todo definido desde el inicio del proyecto, si surgen imprevistos, como cambios de requisitos, nuevas ideas, o cualquier tipo de problema, se introducen nuevas tareas en los siguientes sprints.

Este marco de trabajo encaja muy bien en equipos pequeños que trabajan de manera autónoma, favoreciendo una autogestión eficiente y una mejora continua del proyecto.

#### Test Driven Development (TDD)

En la misma línea de metodologías ágiles, se ha optado por seguir la práctica de TDD[5], en la cual se escriben los tests antes de implementar el código que se necesita probar.

TDD consta de tres fases repetitivas:

1. **Fase roja** : se escriben los test destinados a comprobar una determinada funcionalidad. Inicialmente, estos tests fallan.
2. **Fase verde** : se implementa el mínimo código necesario para que los tests anteriores pasen.

3. **Refactorización** : se optimiza el código a la vez que se asegura que todos los test siguen siendo correctos.

## 4.2. Gestión del proyecto

### Control de versión

Para la gestión de versiones del proyecto se ha escogido Git[12], una herramienta de control de versiones distribuido.

Permite llevar un seguimiento de todos los cambios realizados en el código, además de facilitar la creación de ramas independientes que pueden usarse para implementar nuevas funcionalidades o corregir errores.

Existen otras herramientas de gestión como Subversion (SVN)[37], la cual se había considerado también:

- **Modelo:** Git es un sistema distribuido, mientras que Subversion (SVN) es centralizado.
- **Repositorio local:** Git almacena todo el historial del proyecto localmente; SVN solo guarda una copia de trabajo.
- **Velocidad:** Git realiza operaciones locales mucho más rápido que SVN, que depende del servidor.
- **Gestión de ramas:** Git facilita la creación y fusión de ramas de forma ligera; en SVN, este proceso es más pesado y complejo.
- **Integración con plataformas:** Git se integra de forma nativa con plataformas como GitHub o GitLab, mientras que SVN tiene una integración más limitada.
- **Soporte y comunidad:** Git cuenta con una comunidad más amplia, con mayor soporte y documentación actualizada que SVN.

### Repositorio

Las opciones contempladas fueron GitHub o Gitlab, debido a su integración nativa con Git. Finalmente se ha optado por utilizar GitHub[39], ya que se trata de una plataforma muy popular en proyectos open source, es muy intuitiva y está enfocada a desarrolladores individuales y equipos colaborativos.

Por su parte, Gitlab está más enfocada a entornos de empresas, es más completo pero más complejo de utilizar.

Además, GitHub incluye una serie de herramientas de integración CI / CD llamada GitHub Actions que se utilizará en conjunto con otras herramientas que describiremos en otras secciones.

## Plataforma de gestión

Existen infinidad de plataformas enfocadas a la gestión de proyectos; Trello, Jira, Asana, ZenHub ...

Hemos optado por utilizar ZenHub<sup>[47]</sup>, ya que es una herramienta que se integra fácilmente con GitHub, permite crear sprints, épicas y tableros Kanban.

Si bien es cierto que la herramienta más utilizada por excelencia es Jira, no obstante considero que tiene demasiadas funcionalidades que no van a ser aprovechadas en este proyecto, por lo que es mas conveniente utilizar algo menos potente.

## Comunicación

Para cualquier tipo de comunicación, se utilizará email y Teams.

## 4.3. Lenguajes de programación

### Cliente

Para el desarrollo de la aplicación cliente, se debía escoger un lenguaje de scripts o cercano a estos, ya que el objetivo era construir una herramienta capaz de obtener configuraciones y aplicarlas a los dispositivos de forma sencilla.

Entre los candidatos estaban PowerShell, Java, y Python. Todos son lenguajes que permiten satisfacer los requisitos del proyecto, con ligeras diferencias:

- **PowerShell:** aunque es potente, limita su uso a dispositivos con sistema operativo Windows, restringiendo así la portabilidad.
- **Java:** gracias a su arquitectura multiplataforma, permite ejecutar programas en cualquier entorno. No obstante, generar un archivo

ejecutable (.exe) que funcione universalmente resulta complejo, y en caso de distribuir un archivo .jar, se requeriría que los dispositivos clientes tuvieran Java instalado previamente.

- **Python:** en cambio, ofrece una solución más práctica: mediante sus librerías, es posible generar ejecutables de forma sencilla, compatibles con distintos sistemas operativos, y sin necesidad de instalar software adicional en los dispositivos cliente.

Por esto, se decide utilizar Python[9] para desarrollar la aplicación de escritorio.

## API

En cuanto al desarrollo de APIs, los dos candidatos por excelencia son Spring Boot y Node.js.

Node.js ha adquirido gran relevancia en los últimos años, ya que permite crear APIs de una manera muy sencilla, siendo especialmente útil en aplicaciones que precisan de una gran velocidad y eficiencia en tiempo real. Por otro lado, Spring Boot, ofrece un amplio conjunto de módulos, que permiten configurar la aplicación atendiendo a las necesidades de cada proyecto, a cambio de una mayor complejidad de desarrollo.

Se ha escogido Spring Boot[10] debido a que, aunque el proyecto inicial es relativamente pequeño, ofrece la posibilidad de escalar la aplicación y gestionar un mayor número de usuarios en un futuro. Además, integra módulos (beans, core, security...) que facilitan la implementación de mecanismos avanzados de desarrollo, como el control de accesos de usuarios.

## Frontal

Finalmente, para el frontend, se evaluaron dos alternativas: Angular o React. React proporciona gran flexibilidad y es ideal para proyectos en los que se quiere tener control total de la arquitectura y librerías usadas, aunque depende en gran medida de librerías de terceros para su desarrollo. Angular, en cambio ofrece un framework completo, con una gran cantidad de funcionalidades predefinidas, lo que facilita el mantenimiento y la optimización del código, especialmente en proyectos grandes.

Se ha optado por Angular[2], priorizando su robustez, escalabilidad y la gran cantidad de herramientas que proporciona.



## 4.4. Arquitectura

### Patrón de diseños

En este proyecto se han aplicado distintos patrones de diseño, adaptados a las diferentes necesidades.

La aplicación de escritorio del cliente, desarrollada en Python, emplea una estructura basada en la gestión dinámica de una única ventana, ocultando y mostrando distintos elementos. Esto es similar al concepto de SPA (Single Page Application) en el ámbito web.

La API REST desarrollada en Spring Boot, sigue una arquitectura MVC (Modelo-Vista-Controlador)[41].

El frontend en Angular adopta un patrón similar a MVC, donde los componentes representan la vista, y el controlador son los servicios que encapsulan el modelo de datos y la comunicación con la API.

## 4.5. Herramientas de desarrollo

### Cliente

Para el desarrollo de la aplicación cliente se ha utilizado el Entorno de Desarrollo Integrado (IDE) PyCharm[17], ya que se encuentra plenamente enfocado a Python, ofrece multitud de herramientas y permite realizar configuraciones rápidas.

Otra opción era Visual Studio Code, pero para este lenguaje en concreto, se necesitan descargar multitud de extensiones para poder desarrollar cómodamente.

Para la instalación de librerías externas, se utiliza PIP[3], un gestor de paquetes que permite instalar, actualizar y gestionar librerías.

### API

Para nuestra API con Spring Boot, se ha optado por el IDE IntelliJ IDEA[16] frente a Eclipse, por la sencilla razón de que IntelliJ es mucho más potente, intuitivo y dispone de todas las herramientas necesarias, sin necesidad de instalar nada.

En conjunto con este IDE, se ha añadido una herramienta de automatización de compilación moderna y potente, como es Gradle[14], permite compilar, construir, testear y desplegar aplicaciones de software.

En combinación con Spring Boot, destaca su uso frente al de Maven, ya que es más ligero, flexible y permite tener un control de los despliegues.

## Frontend

El frontend se ha desarrollado utilizando Visual Studio Code[24], también es posible utilizar IntelliJ y tener ambos proyectos (API y Frontend) en el mismo IDE, sin embargo, no está tan optimizado para la parte front y la sensación de programar el frontal dista mucho de ser óptima.

Además, utilizamos NPM (Node Package Manager)[31] como gestor de paquetes para instalar librerías externas de una manera simple y segura.

## 4.6. Integración continua

### Integración y despliegue continuos (IC/DC)

Para automatizar compilaciones, pruebas y despliegue, se ha utilizado GitHub actions[13], en este caso, al escoger GitHub como repo, viene incluido sin necesidad de instalar nada más.

### Cobertura y calidad de código

SonarQube Cloud[36] se ha seleccionado como plataforma para analizar la calidad y cobertura del código fuente. La razón de su uso se debe a que no es necesario instalar ningún servidor, basta con conectarlo al repositorio de GitHub y se puede consultar todos los resultados online.

### Despliegue

Docker[8] es una plataforma que permite empaquetar una aplicación y todas sus dependencias en un contenedor portable y que puede ejecutarse en cualquier entorno

## 4.7. Documentación

La documentación se ha desarrollado utilizando LaTeX[32], en lugar de otras herramientas de procesamiento de texto como Word. Word funciona muy bien para documentos simples y además es fácil e intuitivo desde el principio. No obstante, LaTeX permite un control de todo el formato, genera documentos de una calidad tipográfica excelente y es ideal para documentos técnicos. Sin embargo, es necesario un periodo de adaptación para aprender a usarlo.

Como editor LaTeX se utiliza IntelliJ Idea junto con el plugin TeXiFy-IDEA y el compilador pdfLaTeX.

Visualizamos los documentos PDF generados con SumatraPDF por integrarse de manera excelente con las herramientas mencionadas.

## 4.8. Librerías

En este apartado mencionamos las librerías más importantes, sin contar aquellas que se encuentran preinstaladas en los frameworks utilizados.

### Cliente

#### PyInstaller

PyInstaller[33] es una herramienta que permite convertir programas escritos en Python en ejecutables independientes para Windows, Linux y macOS.

#### Inno Setup Compiler

Inno Setup[35] es una herramienta gratuita para crear instaladores para aplicaciones de Windows. Permite empaquetar archivos, configuraciones y recursos necesarios, generando un asistente de instalación (.exe) que facilita la distribución y el despliegue de programas.

### API

#### Jjwt-api

Jjwt-api forma parte de la biblioteca JJWT (Java JWT)[18], una de las librerías más populares para trabajar con JSON Web Tokens (JWT) en aplicaciones Java.

## **JUnit**

JUnit[19] es uno de los frameworks de testing más populares para aplicaciones Java. Permite crear y ejecutar de manera estructurada tests unitarios, ayudando a garantizar que cada componente de un programa funcione correctamente de forma aislada.

## **Frontend**

### **Angular CLI**

Angular CLI (Command Line Interface)[1] es una herramienta oficial para crear, desarrollar, construir y mantener aplicaciones Angular desde la línea de comandos.

### **Tailwind CSS**

Tailwind CSS[20] es un framework de CSS utilitario que permite diseñar interfaces de usuario rápidamente mediante la combinación de pequeñas clases directamente en el HTML .

### **DaisyUI**

DaisyUI[7] es un complemento de componentes UI construido sobre Tailwind CSS .

Ofrece componentes predefinidos como botones, modales, menús, tabs, alertas, etc., todos diseñados usando clases de Tailwind.

---

## 5. Aspectos relevantes del desarrollo del proyecto

---

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros<sup>3</sup>, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.

---

## 6. Trabajos relacionados

---

La gestión centralizada de dispositivos, es una práctica habitual en la administración de sistemas, y casi desde sus inicios, se han desarrollado herramientas para la gestión de dispositivos de red. Esta gestión ha evolucionado, y sigue evolucionando hacia una automatización de tareas, la orquestación de servicios y la gestión de configuraciones.

En este apartado se exponen aquellos artículos científicos y herramientas existentes que guardan similitud con el presente trabajo.

### 6.1. Artículos científicos

#### **An Application to Automate the Configuration Management and Assessment of Practically Based Learning Outcomes in Computer Networking**

Este estudio describe una aplicación desarrollada para automatizar la configuración y evaluación de escenarios prácticos en laboratorios de redes[21]. La herramienta permite aplicar configuraciones a dispositivos reales y simulados, facilitando la evaluación de resultados de aprendizaje prácticos. Aunque se centra en el ámbito educativo, su enfoque en la automatización y gestión de configuraciones es aplicable a entornos universitarios más amplios.

#### **Configuration Management and Security**

Este artículo[4] analiza los desafíos de la gestión de configuración en sistemas de gran escala, destacando la importancia de políticas centralizadas para garantizar la seguridad y el cumplimiento normativo. Aunque no se enfoca específicamente en la educación, se describe la necesidad de herra-

mientas que permitan la gestión de configuraciones de manera eficiente y segura.

## 6.2. Herramientas

### System Center Configuration Manager (SCCM)

SCCM[28] es una solución de Microsoft para la gestión centralizada de configuraciones y actualizaciones. Permite aplicar configuraciones a grupos de dispositivos, gestionar actualizaciones de software ... Aunque es una herramienta comercial, su enfoque en la gestión de configuraciones es similar al propuesto en este trabajo. Para su uso, es necesario tener instalado además Microsoft AD para permitir gestionar dispositivos y usuarios dentro de la red.

### Active Directory (AD) or Entra ID

AD[25] y Entra ID[29] (anteriormente conocido como Azure AD) son soluciones de Microsoft para la gestión centralizada mediante servicios que controlan el acceso a recursos y la configuración de dispositivos en entornos empresariales. La diferencia principal entre ambos es que Entra ID está diseñado para entornos en la nube, mientras que AD se utiliza principalmente en entornos locales.

### Netsh

Netsh[23] es una herramienta de comandos de Microsoft que permite la configuración y gestión de dispositivos de red en sistemas operativos Windows, ya sea de forma local o remota.

| Características                    | GCA  | SCCM     | AD       | Netsh         |
|------------------------------------|------|----------|----------|---------------|
| Automatización de configuraciones  | Sí   | Sí       | Sí       | No            |
| Gestión centralizada               | Sí   | Sí       | Sí       | No            |
| Priorización por dispositivos      | Si   | Limitada | Sí       | No            |
| Interfaz de usuario                | Sí   | Sí       | Sí       | No (terminal) |
| Curva de aprendizaje               | Baja | Alta     | Alta     | Media         |
| Dependencia con otras herramientas | No   | Sí       | Sí       | No            |
| Multiplataforma                    | Sí*  | Limitada | Limitada | No            |

Tabla 6.1: Comparativa de las herramientas de los proyectos con codificación de colores.

Esta comparativa revela que, aunque soluciones como SCCM y Active Directory ofrecen capacidades avanzadas de gestión de configuraciones y escalabilidad, requieren de hardware específico, son económicamente costosas y están diseñadas para entornos empresariales. En contraste, la propuesta de GCA se centra en la simplicidad y accesibilidad, permitiendo su uso en entornos educativos y de investigación sin la necesidad de infraestructura costosa y compleja, pudiendo ser implementada en cualquier tipo entorno, si así se desea.

No obstante, la elección de la herramienta adecuada dependerá de las necesidades específicas de cada entorno y de los recursos disponibles.

### 6.3. Fortalezas y debilidades del proyecto

| Fortalezas                                 | Debilidades                          |
|--|--------------------------------------|
| Automatización de tareas repetitivas       | Dependencia de la conectividad       |
| Gestión centralizada y escalable           | Complejidad en la implementación     |
| Priorización de configuraciones            | Riesgo de obsolescencia tecnológica  |
| Integración con herramientas existentes    | Requiere primera instalación manual  |
| Mejora en la seguridad                     | Posibles problemas de compatibilidad |
| Compatible con cualquier sistema operativo | Necesidad de mantenimiento continuo  |

Tabla 6.2: Fortalezas y Debilidades del Proyecto



---

## **7. Conclusiones y Líneas de trabajo futuras**

---

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

---

## Bibliografía

---

- [1] Angular. Angular cli - command line interface. "<https://angular.dev/tools/cli>", 2024. [Accessed 27-04-2025].
- [2] Angular. Angular overview. "<https://angular.dev/overview>", 2024. [Accessed 27-04-2025].
- [3] Python Packaging Authority. pip: The python package installer. "<https://pypi.org/project/pip/>", 2024. [Accessed 27-04-2025].
- [4] Steven M. Bellovin and William R. Cheswick. Configuration management and security. 2001. [Accessed 01-05-2025].
- [5] BrowserStack. What is test driven development (tdd)? — browsers-tack guide. "<https://www.browserstack.com/guide/what-is-test-driven-development>", 2024. [Accessed 27-04-2025].
- [6] Cloudflare. What is a firewall? "<https://www.cloudflare.com/es-es/learning/security/what-is-a-firewall/>", 2025. [Accessed 01-05-2025].
- [7] DaisyUI. Daisyui - tailwind css components. "<https://daisyui.com/>", 2024. [Accessed 27-04-2025].
- [8] Docker. Docker official site. "<https://www.docker.com/>", 2024. [Accessed 27-04-2025].
- [9] Python Software Foundation. Python 3 documentation. "<https://docs.python.org/3/>", 2024. [Accessed 27-04-2025].

- [10] Spring Framework. Spring framework documentation - 4.0.x overview. "<https://docs.spring.io/spring-framework/docs/4.0.x/spring-framework-reference/html/overview.html>", 2024. [Accessed 27-04-2025].
- [11] FusionAuth. Jwt components explained. "<https://fusionauth.io/articles/tokens/jwt-components-explained>", 2025. [Accessed 01-05-2025].
- [12] Git. Git - distributed version control system. "<https://git-scm.com/>", 2024. [Accessed 27-04-2025].
- [13] GitHub. Github actions - documentation. "<https://github.com/features/actions>", 2024. [Accessed 27-04-2025].
- [14] Gradle. Gradle build tool. "<https://gradle.org/>", 2024. [Accessed 27-04-2025].
- [15] Red Hat. What is a rest api? "<https://www.redhat.com/es/topics/api/what-is-a-rest-api>", 2025. [Accessed 01-05-2025].
- [16] JetBrains. IntelliJ idea: The leading java and kotlin ide. "<https://www.jetbrains.com/idea/>", 2024. [Accessed 27-04-2025].
- [17] JetBrains. Pycharm: the python ide for professional developers. "<https://www.jetbrains.com/pycharm/>", 2024. [Accessed 27-04-2025].
- [18] JWT. Jjwt: Java jwt library. "<https://github.com/jwtk/jjwt>", 2024. [Accessed 27-04-2025].
- [19] JUnit. Junit 5 user guide. "<https://junit.org/junit5/docs/current/user-guide/>", 2024. [Accessed 27-04-2025].
- [20] Tailwind Labs. Tailwind css - rapidly build modern websites. "<https://tailwindcss.com/>", 2024. [Accessed 27-04-2025].
- [21] Christos Malliarakis, Maria Satratzemi, and Stelios Xinogalos. An application to automate the configuration management and assessment of practically based learning outcomes in computer networking. 2019. [Accessed 01-05-2025].
- [22] Microsoft. ¿Qué es Access Control? | Seguridad de Microsoft — microsoft.com. <https://www.microsoft.com/es-es/security/business/security-101/what-is-access-control>. [Accessed 05-03-2025].

- [23] Microsoft. Microsoft netsh. "<https://learn.microsoft.com/es-es/windows-server/administration/windows-commands/netsh>", 2024. [Accessed 01-05-2025].
- [24] Microsoft. Visual studio code. "<https://code.visualstudio.com/>", 2024. [Accessed 27-04-2025].
- [25] Microsoft. Active directory domain services overview. "<https://learn.microsoft.com/es-es/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview>", 2025. [Accessed 01-05-2025].
- [26] Microsoft. Configure windows firewall with command line. "<https://learn.microsoft.com/en-us/windows/security/operating-system-security/network-security/windows-firewall/configure-with-command-line?tabs=powershell>", 2025. [Accessed 01-05-2025].
- [27] Microsoft. Dhcp technologies. "<https://learn.microsoft.com/es-es/windows-server/networking/technologies/dhcp/dhcp-top>", 2025. [Accessed 01-05-2025].
- [28] Microsoft. Microsoft configuration manager documentation. "<https://learn.microsoft.com/es-es/intune/configmgr/>", 2025. [Accessed 01-05-2025].
- [29] Microsoft. Microsoft entra documentation. "<https://learn.microsoft.com/es-es/entra/>", 2025. [Accessed 01-05-2025].
- [30] Microsoft. Powershell scripting overview. "<https://learn.microsoft.com/en-us/powershell/scripting/overview?view=powershell-7.5>", 2025. [Accessed 01-05-2025].
- [31] Inc. npm. npm: Node package manager. "<https://www.npmjs.com/>", 2024. [Accessed 27-04-2025].
- [32] LaTeX Project. The latex project. "<https://www.latex-project.org/>", 2024. [Accessed 27-04-2025].
- [33] PyInstaller. Pyinstaller documentation. "<https://pyinstaller.org/>", 2024. [Accessed 27-04-2025].
- [34] Scrum.org. What is scrum? — scrum.org learning series. "<https://www.scrum.org/learning-series/what-is-scrum/>", 2024. [Accessed 27-04-2025].

- [35] Inno Setup. Inno setup compiler. "<https://jrsoftware.org/isinfo.php>", 2024. [Accessed 27-04-2025].
- [36] SonarSource. Sonarqube cloud documentation. "<https://docs.sonarsource.com/sonarqube-cloud/>", 2024. [Accessed 27-04-2025].
- [37] Apache Subversion. Apache subversion. "<https://subversion.apache.org/>", 2024. [Accessed 27-04-2025].
- [38] TechTarget. Server message block protocol. "[https://www.techtarget.com/searchnetworking/definition/Server-Message-Block-Protocol#:~:text=The%20Server%20Message%20Block%20\(SMB\)%20protocol%20is%20a%20client%2D,other%20resources%20on%20a%20network.](https://www.techtarget.com/searchnetworking/definition/Server-Message-Block-Protocol#:~:text=The%20Server%20Message%20Block%20(SMB)%20protocol%20is%20a%20client%2D,other%20resources%20on%20a%20network.)", 2025. [Accessed 01-05-2025].
- [39] Wikipedia. Github — wikipedia, the free encyclopedia. "<https://en.wikipedia.org/wiki/GitHub>", 2024. [Accessed 27-04-2025].
- [40] Wikipedia. Ieee 802.1x — wikipedia, la enciclopedia libre. "[https://es.wikipedia.org/w/index.php?title=IEEE\\_802.1X](https://es.wikipedia.org/w/index.php?title=IEEE_802.1X)", 2024. [Accessed 05-03-2025].
- [41] Wikipedia. Model–view–controller — wikipedia, the free encyclopedia. "<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>", 2024. [Accessed 27-04-2025].
- [42] Wikipedia. Address resolution protocol. "[https://en.wikipedia.org/wiki/Address\\_Resolution\\_Protocol](https://en.wikipedia.org/wiki/Address_Resolution_Protocol)", 2025. [Accessed 01-05-2025].
- [43] Wikipedia. Domain name system. "[https://en.wikipedia.org/wiki/Domain\\_Name\\_System](https://en.wikipedia.org/wiki/Domain_Name_System)", 2025. [Accessed 01-05-2025].
- [44] Wikipedia. Https. "<https://en.wikipedia.org/wiki/HTTPS>", 2025. [Accessed 01-05-2025].
- [45] Wikipedia. Protocolo de control de mensajes de internet. "[https://es.wikipedia.org/wiki/Protocolo\\_de\\_control\\_de\\_mensajes\\_de\\_Internet](https://es.wikipedia.org/wiki/Protocolo_de_control_de_mensajes_de_Internet)", 2025. [Accessed 01-05-2025].
- [46] Wikipedia. Protocolo de control de transmisión. "[https://es.wikipedia.org/wiki/Protocolo\\_de\\_control\\_de\\_transmisi%C3%B3n](https://es.wikipedia.org/wiki/Protocolo_de_control_de_transmisi%C3%B3n)", 2025. [Accessed 01-05-2025].
- [47] ZenHub. Getting started with zenhub. "<https://www.zenhub.com/getting-started>", 2024. [Accessed 27-04-2025].