



## **Implementación de back end para sistema de generación de certificados**

Luis Mateo Hincapié Martínez 1

Informe de práctica como requisito para optar al título de:  
Ingeniero de sistemas

Asesores

Interno: Catalina Maria Cespedes Toro, Especialista en gerencia de proyectos  
Externo: Johana Saldarriaga Naranjo, Magíster en gestión y desarrollo de proyectos de software

Universidad de Antioquia  
Facultad de Ingeniería  
Ingeniería de Sistemas  
Pregrado  
Medellín, Colombia  
2024

---

<b>Cita</b>	(Hincapie, Luis, 2024)
<b>Referencia</b>	Hincapie Martinez, L. M (2024). <i>Informe final de práctica para optar al título de Ingeniero de Sistemas: Implementación de back end para sistema de generación de certificados, 2024</i> [Pregrado]. Universidad de Antioquia, Medellín.
<b>Estilo APA 7 (2020)</b>	

---



Centro de Documentación Ingeniería (CENDOI)

**Repositorio Institucional:** <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - [www.udea.edu.co](http://www.udea.edu.co)

**Rector:** John Jairo Arboleda Céspedes.

**Decano/Director:** Julio César Saldarriaga Molina.

**Jefe departamento:** José Luis Botía Valderrama.

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

## **Agradecimientos**

Quisiera expresar mi más profundo agradecimiento a todas las personas e instituciones que hicieron posible la realización de estas prácticas profesionales y culminación de mi proceso académico como Ingeniero de Sistemas.

En primer lugar, a mi familia por brindarme su apoyo incondicional y ánimo durante todo este proceso de formación. Su comprensión y palabras de aliento fueron fundamentales para culminar con éxito este capítulo.

A la Universidad de Antioquia por darme la oportunidad de poner en práctica los conocimientos adquiridos durante la carrera y confrontarlos con los retos del mundo laboral real. Igualmente por facilitar espacios de aprendizaje activo impulsando el desarrollo de competencias profesionales.

A la empresa Blankfactor por abrirme las puertas, acogerme como parte de su equipo y permitirme contribuir a sus objetivos. El poder trabajar en un ambiente profesional real ha enriquecido enormemente mi experiencia.

Finalmente a mis asesoras, Catalina Céspedes y Johana Sladarriga, por compartir sus valiosos conocimientos y dedicar tiempo a guiarme durante todo el proceso de prácticas. Su visión y retroalimentación ayudaron a potenciar mis habilidades y crecimiento profesional.

## Tabla de contenido

Resumen	7
Abstract	8
Introducción	9
1 Objetivos	10
1.1 Objetivo general	10
1.2 Objetivos específicos	10
4 Resultados	16
4.1 Producto	16
4.2 Funcionalidades principales:	18
5 Análisis	23
6 Conclusiones	25
Referencias	27

## Lista de figuras

<b>Figura 1</b>	Infografía sobre los eventos de scrum	13
<b>Figura 2</b>	Esquema gráfico de las etapas de desarrollo	14
<b>Figura 3</b>	Diagrama de arquitectura	16
<b>Figura 4</b>	Diagrama de componentes del sistema	17
<b>Figura 5</b>	Diagrama de flujo del proceso de autenticación	18
<b>Figura 6</b>	Diagrama de flujo del proceso de creación de plantillas	19
<b>Figura 7</b>	Diagrama de flujo del proceso de generación de certificados	20
<b>Figura 8</b>	Diagrama de flujo del proceso de generación de reporte	21
<b>Figura 9</b>	Modelo entidad-relación	22

## **Siglas, acrónimos y abreviaturas**

<b>CI</b>	Continuous Integration
<b>CD</b>	Continuous Delivery
<b>CRUD</b>	Create Read Update Delete
<b>UdeA</b>	Universidad de Antioquia
<b>IDE</b>	Integrated Development Environment
<b>REST</b>	Representational State Transfer
<b>HTTP</b>	Hypertext Transfer Protocol
<b>API</b>	Application programming interface
<b>QA</b>	Quality Assurance
<b>AWS</b>	Amazon Web Services

## Resumen

El enfoque del trabajo es el desarrollo del back end mediante la implementación de una API REST para automatizar la generación de certificados en una compañía. La gestión manual actual ha causado demoras y deficiencias del área de recursos humanos. Se plantea una aplicación web que elimine la intervención manual. El back end se integrará con un front end el cual permitirá a los empleados generar certificados de manera independiente y a líderes del área de recursos humanos supervisar el proceso. La aplicación contendrá autenticación para diferentes roles, almacenamiento de plantillas de certificados, generación de certificados para empleados activos e inactivos, informes históricos. El sistema deberá respetar políticas de privacidad y gobernanza de datos personales, además deberá generar certificados en al menos dos idiomas: inglés y español . El objetivo es agilizar el proceso, reducir retrasos y errores, mejorar la satisfacción de empleados y recursos humanos mediante una solución que cumpla con los estándares de una aplicación empresarial, respetando los lineamientos de gobernanza de datos, seguridad y privacidad de datos personales.

*Palabras clave:* Generación de certificados, Automatización, Gestión de plantillas, Aplicación empresarial, desarrollo back end.

### **Abstract**

The focus of the work is the development of the back end by implementing a REST API to automate the generation of certificates in a company. The current manual management has caused delays and deficiencies in the human resources area. We propose a web application that eliminates manual intervention. The back end will be integrated with a front end that will allow employees to generate certificates independently and HR leaders to supervise the process. The application will contain authentication for different roles, certificate template storage, certificate generation for active and inactive employees, historical reports. The system will have to respect privacy and personal data governance policies, as well as generate certificates in at least two languages: English and Spanish. The objective is to streamline the process, reduce delays and errors, improve employee satisfaction and human resources through a solution that meets the standards of an enterprise application, respecting the guidelines of data governance, security and privacy of personal data.

*Keywords:* Certificate generation, Automation, Template management, Enterprise application, back end development.



## Introducción

En el marco de la búsqueda constante de mejorar los procesos de la empresa Blankfactor, se presenta la propuesta para desarrollar el back end de una aplicación web que automatice la generación de certificados. El proceso manual actual ha resultado en demoras y complejidades que obstaculizan la eficiencia operativa. La necesidad de combinar datos de diferentes fuentes y personalizar cada certificado ha creado ineficiencias en dicho proceso.

Para abordar este problema, se propone el desarrollar un back end bajo la arquitectura de una API REST robusta para una aplicación web que automatice la generación de certificados. Además, se diseñará un sistema de templates de plantillas y reportes que garanticen la escalabilidad y personalización de otro tipo de de certificados, por último se complementará a un sistema de reportes para garantizar la trazabilidad y auditoría dentro del sistema, para que sea accesible se integrará con el Front end el cual será desarrollado durante el mismo espacio de tiempo y que servirá como interfaz gráfica para consumir la API de la que se habla en el presente documento.

Se espera lograr mejoras significativas en la eficiencia operativa. Esto se traducirá en una reducción notable en los tiempos de respuesta para la emisión de certificados y, en última instancia, en una experiencia más satisfactoria tanto para el equipo de recursos humanos como para los empleados de la empresa. La generación automatizada de certificados brindará un flujo de trabajo más rápido y preciso, eliminando las ineficiencias asociadas al proceso manual y asegurando que la información sea precisa y confiable.

## 1 Objetivos

### 1.1 Objetivo general

Desarrollar una API REST robusta y funcional para una aplicación web que automatice la generación de diferentes tipos de certificados.

### 1.2 Objetivos específicos

- Diseñar e implementar una arquitectura escalable mediante una API REST que permita a los empleados generar certificados de manera autónoma.
- Desarrollar módulos que permitan la gestión de múltiples plantillas y gestión de firmas de certificados.
- Desarrollar un sistema de informes que ofrezca una visión clara y detallada de los certificados generados en un período de tiempo específico.
- Integrar principios de gobernanza de datos, seguridad y privacidad de datos personales en la API REST para garantizar que toda la información manejada por la aplicación esté protegida y sea utilizada de manera responsable.
- Asegurar la integración con una interfaz gráfica, front end, de forma que la API sea consumible de forma simple por cualquier empleado de la compañía.

## 2 Marco teórico

Un certificado es un documento por medio del cual una persona puede demostrar o dar por verdad la propiedad o dominio sobre una cosa (Lengnick-Hall, 2012), específicamente dentro de la empresa Blankfactor se generan principalmente certificados laborales, los cuales certifican que la persona que está consignada en el mismo tiene o tuvo algún tipo de relación laboral con la compañía.

“Las presiones competitivas están motivando a las organizaciones a hacer un uso cada vez más eficiente, eficaz e innovador de las tecnologías de la información para transformar los procesos intra e interorganizativos. Esta necesidad particular de la vida organizativa es una de las que abordan la investigación y la práctica de los sistemas de información, ya que ocupa un hueco entre, por un lado, la ingeniería de software y, por otro, la empresa/organización. Así pues, el punto de vista tecnológico destaca la importancia de las TI para el desarrollo, la implantación y el uso de los SI en la práctica” (Boell, 2015). Dado la evolución constante de las empresas a usar sistemas de información dentro de sus procesos, este enfoque enfatiza cómo los sistemas de información desempeñan un papel fundamental en la recopilación, almacenamiento y distribución de datos en una organización. En este proyecto, el desarrollo del producto consiste en una aplicación web que actúa como un sistema de información que centraliza y gestiona la generación de certificados.

La importancia de adoptar este tipo de soluciones en las empresas se explica según Tesoro y Sakona (1993), porque “uno de los grandes retos de empresarios y gerentes es la productividad, y desde esta óptica, gestionar adecuadamente los procesos es cada vez más importante”. Con esto se pretende “aumentar la eficiencia de los procesos administrativos de rutina” y “aumentar la eficacia y la eficiencia de la actividad gerencial, mejorando el desempeño humano ante problemas complejos y dinámicos”.

Para automatizar este proceso se requerirá construir el back end implementado una API que servirá como el servicio encargado de gestionar la lógica y almacenar la información, el cual deberá integrarse con la interfaz de usuario que le provea a los empleados la capacidad de interactuar con el sistema.

“El desarrollo back end (a menudo llamado desarrollo "del lado del servidor"), es la creación de todo lo que ocurre entre bastidores de un sitio web o aplicación que el usuario no

puede ver. Este consta normalmente de una aplicación, un servidor y una base de datos. Al interactuar con un sitio introduciendo información, ésta se almacena en una base de datos que reside en un servidor. A continuación, se le devuelven los resultados para que los muestre en el sitio como “código front end” (Mejia, 2009). La automatización de tareas manuales y repetitivas puede liberar recursos humanos para dedicar sus esfuerzos a operaciones más críticas. En el contexto del proyecto, la automatización permitirá la generación de certificados de manera eficiente y precisa.

Sin embargo, el término back end no define específicamente cómo se construirá la propuesta de solución, la cual será un API Rest; “REST (Representational State Transfer) es una arquitectura API (Application Programming Interface) que proporciona comunicaciones cliente-servidor para Aplicaciones Web sobre el protocolo HTTP, haciéndolo fácilmente implementable, ya que no está vinculada a ningún protocolo de transferencia. Los tres principales principios de diseño de REST son la direccionalidad, la interfaz uniforme y la ausencia de estado. REST aborda la aceptabilidad definiendo puntos finales en una estructura de directorio a través de diferentes URI para extraer los datos. La API funciona según el principio CRUD (Create, Read, Update, Delete)” (Prayogi et al.,2020). Esta arquitectura será la base de la implementación y posterior desarrollo del sistema, la cual en integración con el front end, dará al usuario una experiencia adecuada.

Finalmente se debe mencionar el concepto de gobernanza de datos, que es de suma importancia dentro de los procesos de trabajo de la empresa y específicamente en el proceso de generación de certificados con todo lo subyacente a la gestión de datos personales e información sensible de carácter privado, que puede abarcar cargos, fechas, salarios, entre otros, pues según “el régimen normativo nacional e internacional de protección de datos personales, así como sus fundamentos científicos y filosóficos han evolucionado en función de las condiciones tecnológicas y económicas en las que se ejerce globalmente el poder informático” (Silva et al., 2019). Teniendo en cuenta esta evolución social y normativa frente al tratamiento de datos es necesario explorar enfoques para garantizar la privacidad y seguridad de los datos personales en sistemas de información. La consideración de estos aspectos es crucial para asegurar el cumplimiento de las regulaciones de privacidad en el manejo de información de los empleados.

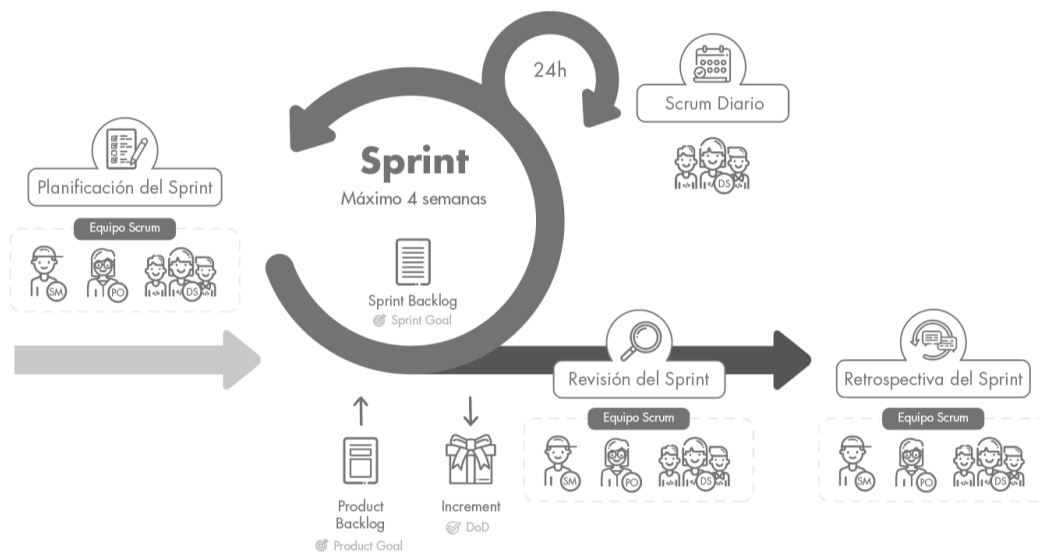
### 3 Metodología

El desarrollo de la propuesta se llevó a cabo utilizando la metodología ágil Scrum, la cual proporciona un marco de trabajo flexible y colaborativo que se adapta a la naturaleza incremental y evolutiva del proyecto, sin embargo al ser un proyecto nuevo se agregaron ciertos pasos de elicitación de requerimientos, análisis y diseño del sistema.

Scrum es un marco de trabajo que los equipos utilizan para auto organizarse y trabajar en pro de un objetivo común. Describe un conjunto de reuniones, herramientas y roles para la entrega eficiente de proyectos. Al igual que un equipo deportivo que practica para un gran partido, las prácticas de Scrum permiten a los equipos auto gestionarse, aprender de la experiencia y adaptarse al cambio. Los equipos de software utilizan Scrum para resolver problemas complejos de forma rentable y sostenible.

#### Figura 1

*Infografía sobre los eventos de scrum.*



Durante el proceso de desarrollo del proyecto se utilizaron herramientas como tableros Kanban y software de gestión de proyectos ágiles para visualizar el progreso y asignar tareas. Las reuniones diarias, las revisiones de sprint y las retrospectivas permitieron una comunicación constante y una mejora continua del proceso, principalmente Jira y confluence:

Jira: Plataforma de gestión de proyectos ágiles que permite la creación de historias de usuario, asignación de tareas, seguimiento del progreso y administración de sprints. Se utilizará para planificar y rastrear el trabajo en curso, así como para garantizar que las historias de usuario se aborden de manera efectiva.

Confluence: Herramienta de colaboración que facilita crear y compartir documentación, informes y recursos importantes. Será utilizada para documentar decisiones, almacenar información relevante del proyecto y crear una base de conocimientos accesible para todos los miembros del equipo.

Para tener una comunicación constante y efectiva con el equipo de trabajo y partes interesadas se utilizaron herramientas como Slack y la suite de herramientas de Google:

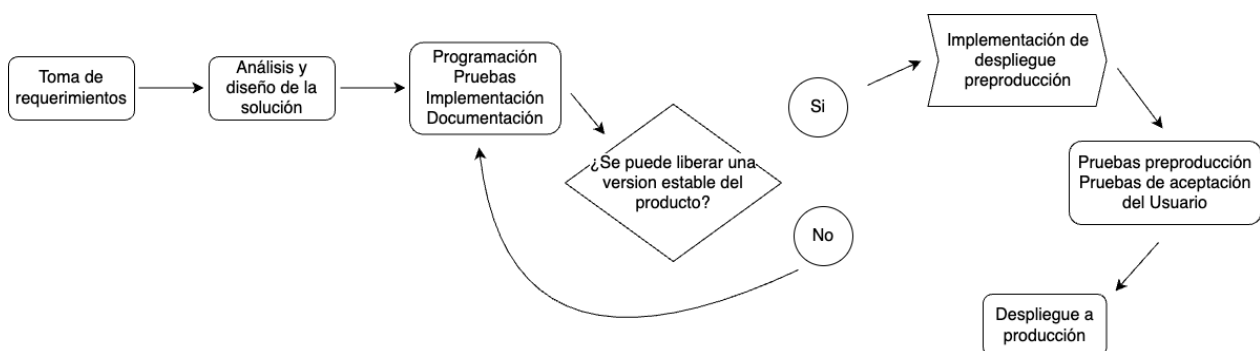
Slack: Plataforma de comunicación en tiempo real que permitirá la comunicación constante y fluida entre los miembros del equipo. Se utilizó para mantener conversaciones, discutir problemas, resolver obstáculos y compartir actualizaciones de manera instantánea.

Google Workspace: Proporciona herramientas como *Calendar* y *Meet* para programar y realizar reuniones, Docs, Sheets, Slides y Drive para la creación, colaboración y almacenamiento de documentos importantes, hojas de cálculo, y otros recursos del proyecto.

Las etapas del proceso de desarrollo del proyecto se estructuraron de la siguiente manera:

## Figura 2

*Esquema gráfico de las etapas de desarrollo.*



Aunque el marco de trabajo Scrum no incorpora explícitamente etapas pre o post desarrollo, se establecieron las siguientes fases:

**Toma de Requerimientos:** Durante esta fase, se recopilaron y documentaron los requisitos esenciales del sistema. La comunicación efectiva entre el equipo y los stakeholders fue fundamental para comprender las necesidades del proyecto. Se llevó a cabo una reunión entre todos los involucrados para definir el producto mínimo viable.

**Análisis y Diseño de la Solución:** En esta etapa, se analizaron los requisitos identificados para definir la arquitectura y el diseño del sistema. Se crearon diagramas y especificaciones técnicas para guiar la implementación. Se realizaron esquemas UML de flujo del sistema, de modelo entidad-relación (MER) y de arquitectura.

**Desarrollo:** Durante esta fase, se llevó a cabo la implementación del sistema según las especificaciones y el diseño establecidos. Se iteró según lo definido por el marco de trabajo Scrum, con reuniones y comunicación constante entre los interesados. También se realizaron pruebas y se documentó lo implementado en cada Sprint. Una vez se tuvo una versión mínima estable y funcional, se pasó a la siguiente etapa.

**Implementación y Despliegue en Preproducción:** Después del desarrollo, se realizó una implementación inicial en un entorno preproducción. Esto facilitó pruebas adicionales y la corrección de posibles problemas antes del despliegue final. En este punto, se aseguró de que el ambiente fuera replicable y semejante al futuro entorno productivo.

**Pruebas de Preproducción y Aceptación del Usuario:** Se llevaron a cabo pruebas exhaustivas en el entorno preproducción para garantizar el correcto funcionamiento del sistema. Los usuarios finales participaron en pruebas de aceptación para validar que la solución satisfacía sus necesidades.

**Despliegue a Producción:** Tras la aprobación de las pruebas de preproducción y la aceptación del usuario, la aplicación se desplegó en el entorno de producción, estando así disponible para su uso general.

## 4 Resultados

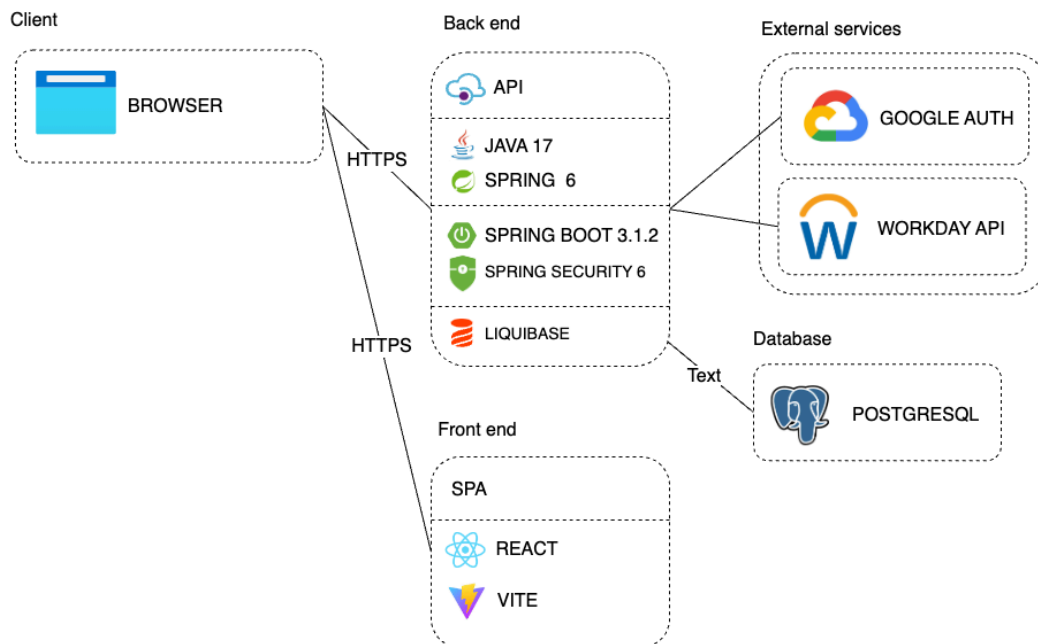
### 4.1 Producto

El back end forma parte de los cimientos del proyecto, ya que se encarga de la autenticación y autorización de usuarios, la gestión de certificaciones, la gestión de plantillas, la gestión de firmas, y la comunicación con servicios externos.

La arquitectura adoptada es un API REST, la cual proporciona una solución flexible, escalable y eficaz para gestionar el flujo de trabajo de generación de certificados, la integración con el front end y las integraciones externas. Al seguir los estándares REST, el proyecto garantiza una clara separación de preocupaciones, divide las funcionalidades en diferentes recursos; en puntos de acceso específicos con una URL única que identifica cada recurso y puede gestionarse de forma independiente. Es flexible y fácil de integrar con otros servicios porque permite estandarizar el intercambio de datos entre diferentes servicios web, basados en métodos HTTP que facilitan la comprensión y el uso para la posterior gestión por parte de la interfaz de usuario.

**Figura 3**

*Diagrama de arquitectura.*



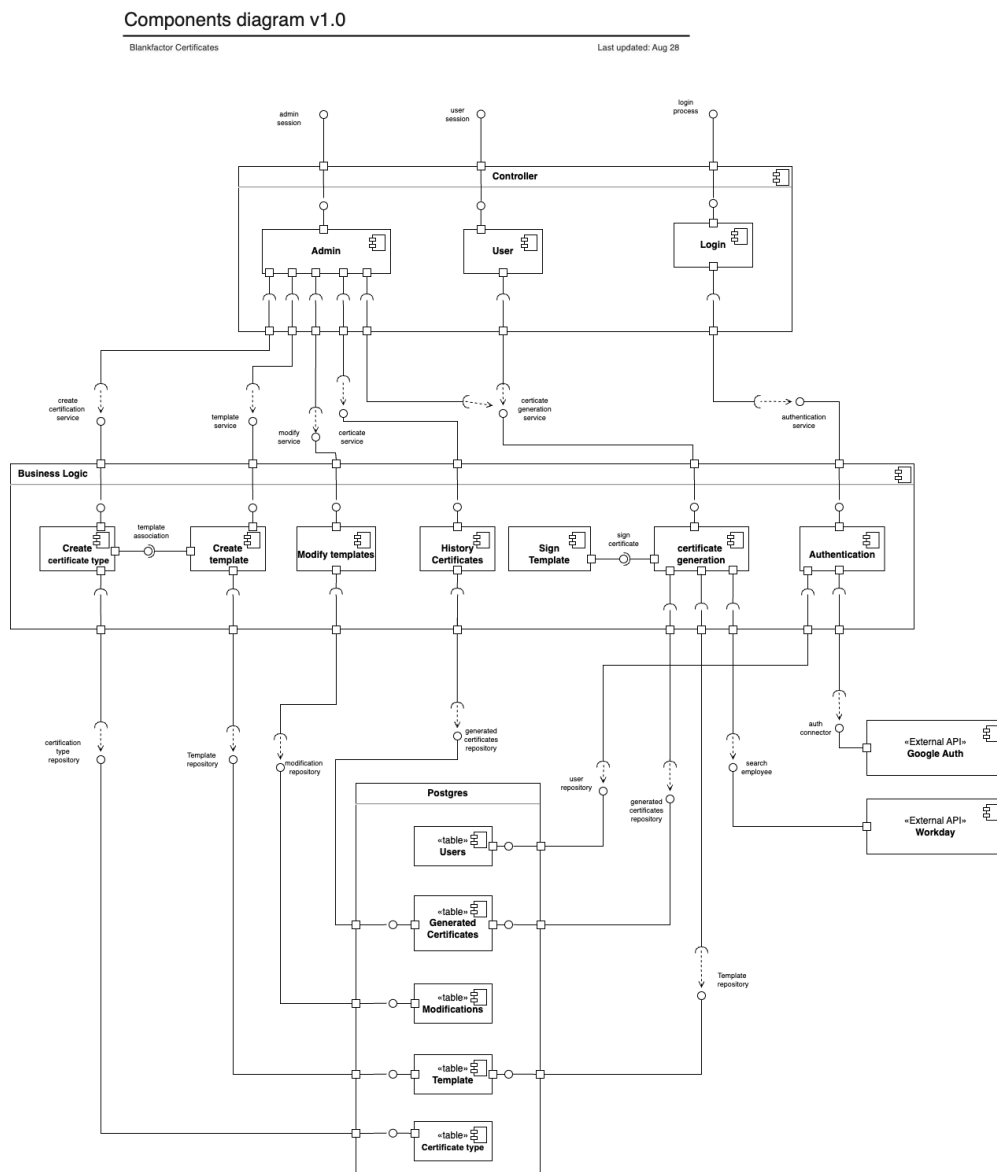


La API se desarrolló en Java en la versión 17 y fue implementada bajo el marco de trabajo Spring. Se seleccionaron núcleos específicos que permitieron simplificar la implementación de funcionalidades y la integración de distintos servicios y recursos como: Spring Boot, Spring Data Rest, y Spring Security. La base de datos es SQL para permitir manejar los datos de forma estructurada y llevar un registro fiable de los certificados generados, esta funciona bajo el motor de Postgres, que es un proyecto de código abierto, robusto con rendimiento sobresaliente, suficiente para este proyecto.

El proyecto se divide en tres capas principales, como se ve en el diagrama: controlador, capa de lógica de negocio y la capa de datos.

#### Figura 4

*Diagrama de componentes del sistema.*



## 4.2 Funcionalidades principales:

Mediante la creación de un mapeo visual de historias (visual story mapping) se definieron las historias de usuario y las funcionalidades que se llevaron a cabo para el producto mínimo viable del proyecto.

Módulo de autenticación y autorización: Acceso mediante inicio de sesión con autenticación/autorización a través de Google.

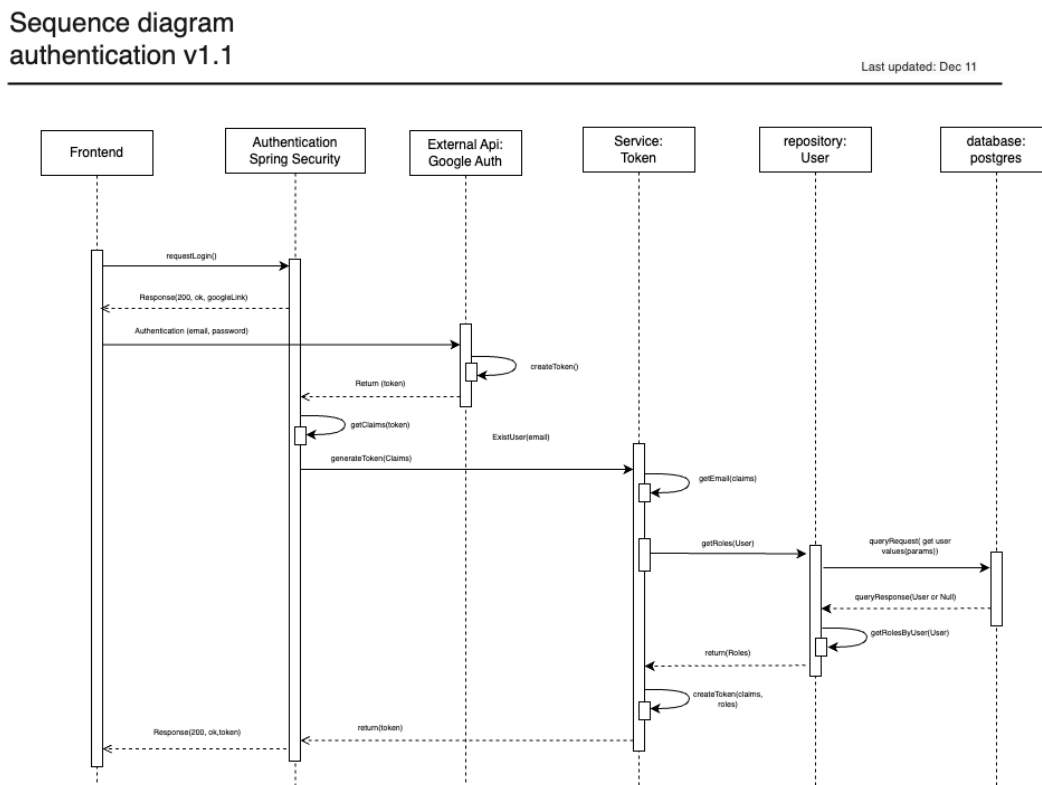
Roles: Líder de Recursos Humanos (HR Leader): Puede seleccionar campos de plantillas y generar certificados para empleados inactivos en la plataforma Workday<sup>1</sup>.

Empleado: Permite generar certificados sin intervención de Recursos Humanos, con la opción de seleccionar información específica para incluir en los certificados.

Administrador: Gestiona la asignación de roles mediante una tabla que mapea los grupos con los grupos de Google.

### Figura 5

Diagrama de flujo del proceso de autenticación.



<sup>1</sup> Workday es una plataforma en la nube que proporciona soluciones integrales para la gestión empresarial, centrándose especialmente en recursos humanos, finanzas y planificación. Es donde se centraliza la información de los empleados de la compañía

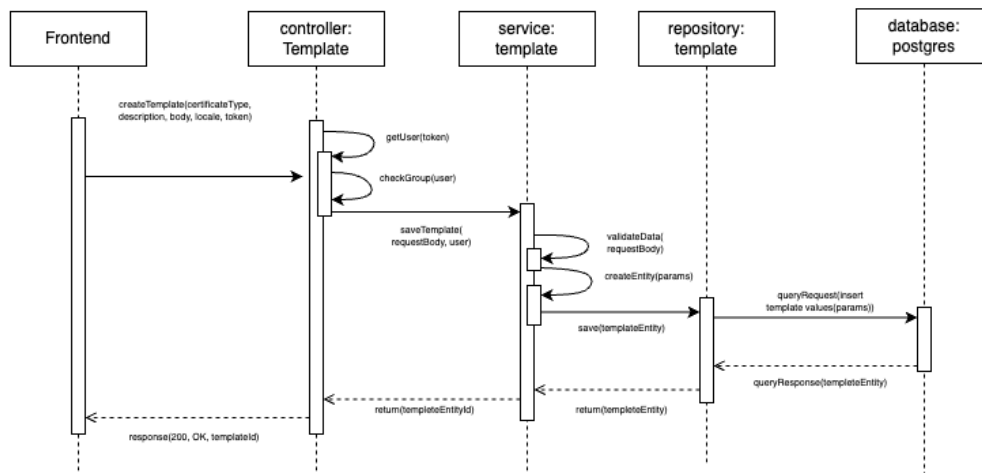
Módulo de plantillas: Permite crear y listar todas las plantillas de certificados almacenadas en el sistema.

### Figura 6

Diagrama de flujo del proceso de creación de plantillas.

#### Sequence diagram template creation v1.0

Last updated: Aug 30



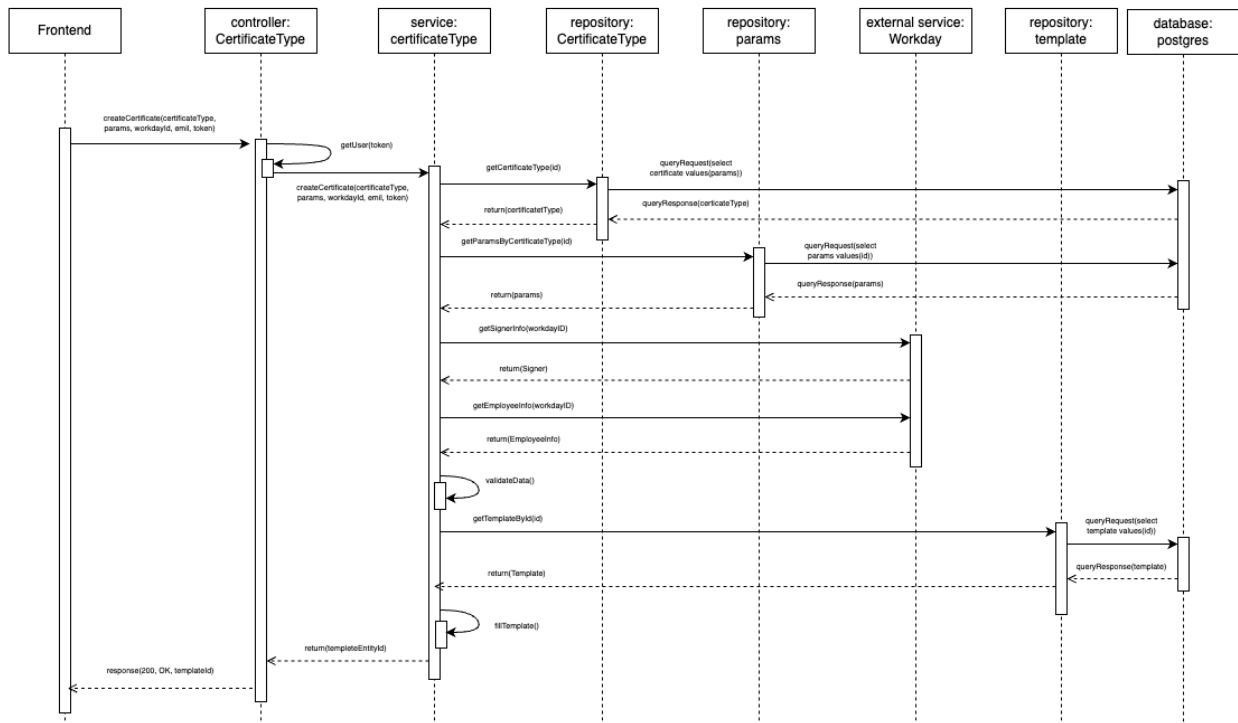
Módulo de generación de certificados: Permite a los empleados generar y descargar certificados firmados sin intervención de Recursos Humanos. Disponible solo para empleados registrados en Workday (activos/inactivos). Permite la descarga de certificados con una selección de tipos de certificados. El Líder de Recursos Humanos puede filtrar empleados por nombre o ID de Workday.

**Figura 7**

Diagrama de flujo del proceso de generación de certificados.

Sequence diagram  
generate certificate v1.1

Last updated: Dec 11



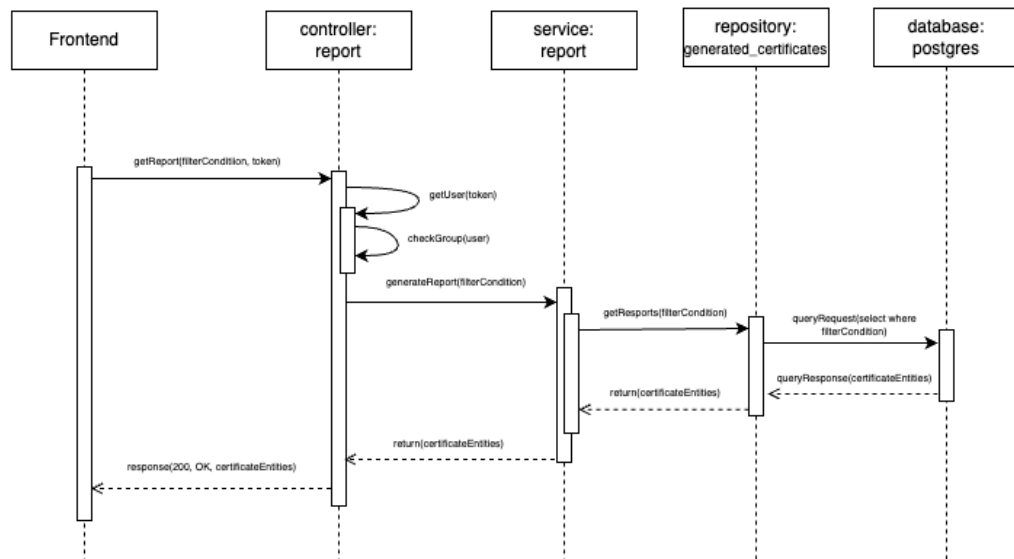
Módulo de reportes: Muestra un historial de certificados solicitados con toda la información de las certificaciones creadas en un período específico. El usuario puede seleccionar un rango de tiempo y campos ( nombres del empleado, ID de Workday del empleado, fechas) sin límites en los rangos de las fechas.

**Figura 8**

Diagrama de flujo del proceso de generación de reporte.

**Sequence diagram  
generate report v1.0**

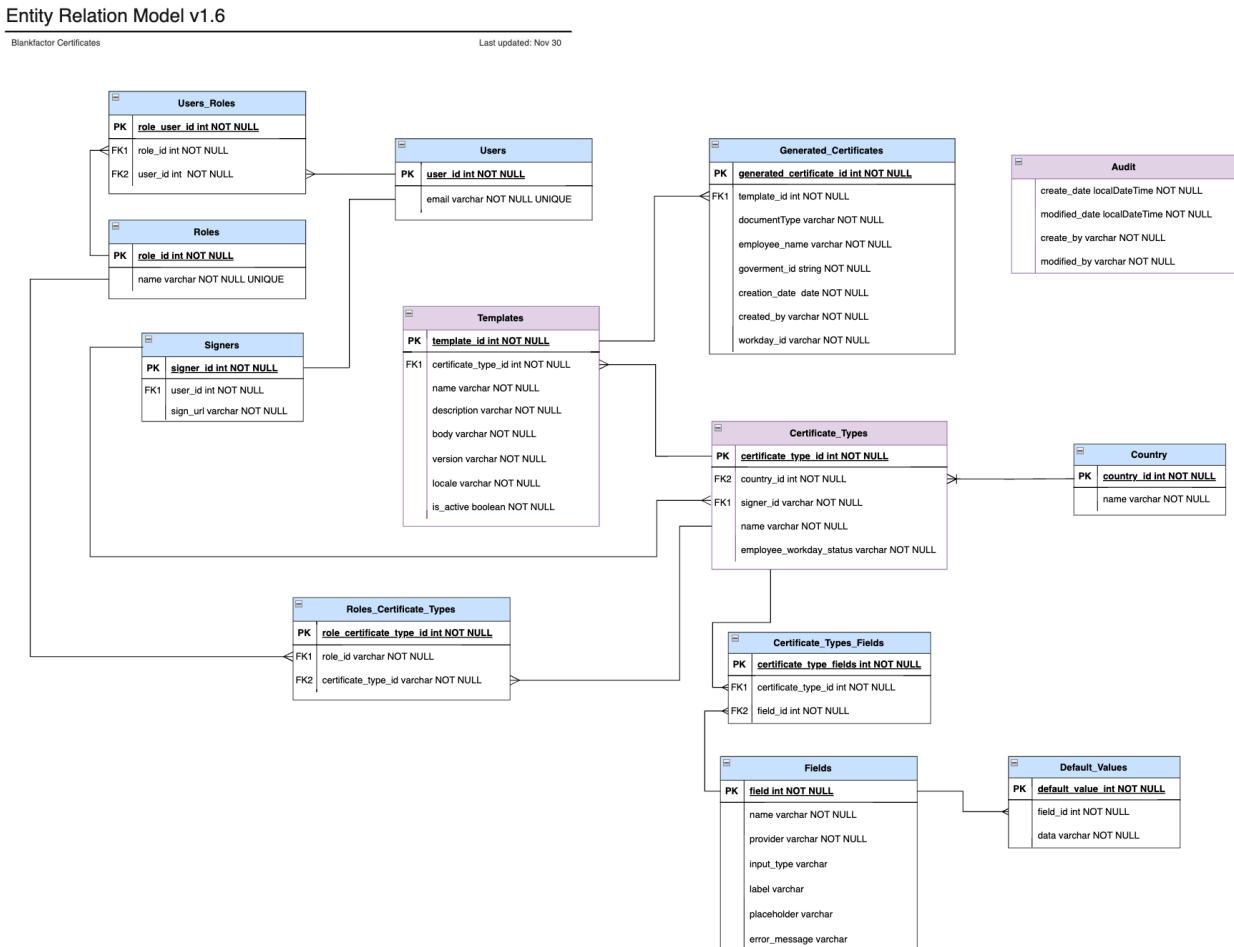
Last updated: Aug 30



El siguiente modelo entidad relación abstrae las partes o entidades presentes en el sistema, este modelo se traduce a tablas en la base de datos.

Figura 9

Modelo entidad-relación.



De izquierda a derecha se tienen los usuarios, los cuales pueden tener asociados varios roles dentro del sistema, a su vez cada usuario tendrá una firma, esto con el fin de que los líderes de recursos humanos puedan definir una única firma para todos los certificados que dependan de ese usuario. Se pueden tomar como entidades principales la entidad de tipos de certificados, que se componen de plantillas y campos, estos últimos sirven de referencia para que el usuario sepa qué información debe ser proveída por el usuario y de qué manera. La entidad de plantillas puede tener uno o varios certificados generados, esto quiere decir que de esa plantilla ha sido usada 1 o más veces para generar un certificado y se almacena información de la plantilla y el usuario al que se le genera el certificado para la generación de reportes de certificados generados. La entidad de auditoría sirve para almacenar información de marcas de tiempo y usuarios, cuando se interactúa con el sistema.

## 5 Análisis

El resultado fue una API REST que se construyó basándose en buenas prácticas y mediante la selección de herramientas adecuadas. El análisis inicial guió el comienzo del desarrollo con la abstracción y diagramación de la solución, mientras que la metodología ágil permitió adaptarse a cambios e iterar rápidamente durante los sprints. El resultado fue una solución flexible y extensible, cumpliendo con los objetivos planteados.

En cuanto a la gobernanza de datos se logró la implementación de servicios de auditoría dentro de los módulos que manejan datos personales lo que permitió tener un mayor control sobre la seguridad y privacidad de los mismos. Se contaba con registros detallados sobre los cambios y actualizaciones realizadas al sistema, como en los templates o la generación de certificados. Esto brindó trazabilidad y posibilidad de investigar incidentes, se consideraron 3 herramientas diferentes para implementar diferentes estándares, y políticas para controlar el uso y flujo dentro del sistema. En primer lugar la librería Envers para tener un historial de los cambios hechos a ciertos valores, otra parte se hizo usando el propio framework, *JPA Auditing*, que permitió almacenar timestamps e información que el usuario usa mientras interactúa con el sistema. Por último, se integró la herramienta *Liquibase* que versiona los cambios que realizamos en la base de datos durante el desarrollo. Al reunir estas herramientas, se garantizó el flujo, el control, la trazabilidad y la observabilidad de los datos.

La autenticación y autorización se realizaron a través de Spring Security con autenticación mediante Google. De esta forma el acceso al sistema está limitado solo a personas con correos activos de la compañía, una vez dentro la autorización de acceso a los módulos o puntos de acceso es controlada desde el servidor de autorización de Spring Security, el cual se configuró según los roles del sistema dividiendo permisos principalmente para empleados y administradores dentro del sistema.

Uno de los puntos cruciales del sistema fue la integración con Workday, el principal repositorio de información de los empleados. Mediante esta, se pudo acceder a datos actualizados y consistentes para generar los certificados. El aprovechar servicios externos existentes en la compañía facilitó el desarrollo y funcionamiento de la aplicación.

Durante el desarrollo de funcionalidades clave para la generación de certificados se presentaron algunos desafíos para llevar a cabo ciertas funcionalidades. Uno de estos retos fue

contribuir a un proyecto open source (Tradukisto<sup>2</sup>) que convierte números a palabras, agregando soporte para español, dado que inicialmente no estaba disponible. Otro desafío fue seleccionar el motor de plantillas más adecuado, decantándose finalmente por Freemarker por su facilidad de uso e integración con librerías como iText para generar archivos PDF.

Con el objetivo de garantizar el óptimo rendimiento de los diversos servicios desarrollados, así como de validar cada restricción e integración, se llevaron a cabo pruebas exhaustivas que abarcaron tanto aspectos unitarios como de integración. Estas pruebas se enfocaron en analizar cada uno de los casos de uso y flujos de trabajo que podrían surgir en la aplicación. Como criterio de aceptación, se estableció una cobertura del 80% en instrucciones y branches (if y switch) para verificar el funcionamiento adecuado de los componentes de las aplicaciones. Estas métricas de cobertura fueron evaluadas mediante el uso de Jacoco, una dependencia que posibilita el análisis de código línea por línea en todas las clases del proyecto. Jacoco permitió recopilar diversas métricas de cobertura y generar informes con los resultados, identificando así los componentes con niveles insuficientes de pruebas. Los resultados de estas métricas de cobertura se verificaron con la integración de Jacoco en el pipeline en GitHub Actions, y los informes resultantes se publicaron en cada *Pull Request* al repositorio. Este enfoque se implementó para asegurar que todas las nuevas funcionalidades cumplieran con las métricas predefinidas, permitiendo así una validación continua del correcto funcionamiento y la calidad del código a lo largo del ciclo de desarrollo.

Finalmente, uno de los principales retos fue garantizar el acceso a la aplicación para cualquier empleado sin inconvenientes. Para lograrlo, se implementó un pipeline de integración y despliegue continuo. Este automatiza la compilación, ejecución de pruebas y validaciones para cada nueva funcionalidad desarrollada. Luego, genera una imagen del proyecto que se despliega en el repositorio versionando los cambios, para finalmente publicarla en la infraestructura de AWS, siguiendo los lineamientos de la empresa. De esta manera se consiguió tener una constante integración y entrega de nuevas funcionalidades a producción.

Una posible mejora para incrementar la mantenibilidad y reducir la complejidad del código en el futuro sería incorporar una librería para mapear automáticamente entre objetos DTO y entidades de modelo. Por ejemplo, herramientas como MapStruct, Orika o Dozer podrían

---

<sup>2</sup> Tradukisto es una potente biblioteca Java diseñada para convertir números en sus correspondientes representaciones de palabras.



utilizarse para generar clases que se encargan de convertir de forma transparente entre estos tipos de objetos. Actualmente en el proyecto se tienen varias clases que realizan manualmente estas conversiones requeridas en los servicios para separar las capas de modelo y API. Si bien cumplen su función, al agregar nuevos campos o atributos se debe recordar actualizar estos mapeos en todos los lugares requeridos. Una librería dedicada para este propósito encapsularía esta lógica en una única localización, reduciendo posibles errores y el esfuerzo en mantenimiento futuro.

Otra mejora relevante sería la estandarización de los logs de la aplicación para hacer más eficiente su uso. Se podría investigar e implementar alguno de los patrones comunes como LOGBack o el estándar de logging de Java (JUL). Esto permitiría definir un formato consistente para los mensajes, estandarizar los niveles de logs disponibles (info, debug, warn, error), mejorar el performance al permitir deshabilitar ciertos niveles en producción, y simplificar la configuración de manejo y rotación de archivos de logs. También se podrían implementar funcionalidades avanzadas como enmascaramiento de datos sensibles, mensajes parametrizados, y registros estructurados en JSON para facilitar su consumo e indexación en herramientas como Elasticsearch. Contar con logs estandarizados y más ricos en contenido Incrementaría la observabilidad del sistema y facilitaría la depuración y monitoreo. Se podría también integrar analítica de logs con Dashboards para obtener métricas e insights operativos del aplicativo.

## 6 Conclusiones

La adopción de Scrum en el desarrollo de software se traduce en una serie de ventajas sustanciales. Facilita la adaptación ágil a los cambios, asegura la entrega temprana de resultados incrementales, fomenta una comunicación continua y colaborativa entre los miembros del equipo mediante ceremonias clave como la Planificación, Daily, Review, Retrospectiva y Refinamiento de Sprint. En entornos de desarrollo donde los requisitos son dinámicos y no completamente definidos, Scrum se revela como una metodología efectiva al aumentar la probabilidad de éxito, tal como se experimentó en este proyecto. Aquí, el producto evolucionó de manera constante, alcanzando finalmente los objetivos acordados en el Producto Mínimo Viable (MVP) dentro de los plazos establecidos, validando así la eficacia de Scrum en la gestión de proyectos de software.

Al construir diagramas anticipados del proyecto se pudieron identificar patrones reutilizables y escalables para la API y los diferentes módulos. De otra manera, sin esta visión inicial, los desarrollos de los distintos miembros del equipo podrían no estar alineados, esto permite establecer una ruta a seguir para facilitar la etapa de desarrollo, se tuvieron diagramas de arquitectura, del modelo de la base de datos, de componentes del sistema, de paquetes y de flujo del sistema, los cuales sirvieron de apoyo y referencia para tener un mapa visual del proyecto.

La integración con sistemas externos como Workday mediante API fue un acierto dado que evitó tener que construir servicios desde cero. Aprovechar recursos existentes aportó valor y permitió enfocarse en la lógica de negocio propia de la aplicación. Para el futuro, se podrían evaluar integraciones adicionales con otros sistemas internos que agreguen datos de valor al proceso de generación de certificados.

La manera en que se abordó todo lo concerniente a gobernanza de datos, privacidad y seguridad durante el desarrollo fue acertado dada la naturaleza sensible de la información manejada. La implementación de funcionalidades de auditoría, control de acceso y manejo de permisos aseguró que se protegieran adecuadamente los datos personales de los empleados. El cumplimiento de estas buenas prácticas genera confianza en el uso responsable de la información por parte del sistema.

La experiencia de práctica no solo enriqueció las habilidades técnicas, sino que también brindó un crecimiento a nivel personal y profesional. El ambiente laboral, la retroalimentación

constante y el desarrollo de habilidades blandas promovieron mejoras significativas en adaptabilidad, inteligencia emocional y comunicación asertiva.

El desarrollo de este proyecto demuestra la importancia y el valor de la automatización de procesos manuales y repetitivos en las organizaciones. Al implementar una solución tecnológica como la API REST se logró optimizar sustancialmente la generación de certificados, reduciendo los tiempos de respuesta de un proceso que antes tomaba días con intervención humana a uno automatizado que puede ejecutarse en cuestión de segundos o minutos.

## Referencias

- Amazon Web Services, Inc. (2023). What is Scrum? Amazon. <https://aws.amazon.com/what-is/scrum/>
- Atlassian. (2023). Confluence. Your Remote-Friendly Team workspace | Atlassian. <https://www.atlassian.com/software/confluence>
- Atlassian. (2023). Jira. Issue & Project Tracking Software. <https://www.atlassian.com/software/jira>
- Boell, Sebastian & Cecez-Kecmanovic, Dubravka. (2015). What is an Information System?. 2015. 10.1109/HICSS.2015.587.
- Google. (2023). Business Apps & Collaboration Tools. Google Workspace. <https://workspace.google.com/>
- Infografía Novedades Scrum. Adaptado de Scrum: el pasado y el futuro [Figura 1], por Miguel Rodríguez. (2020). Netmind (<https://netmind.net/es/scrum-el-pasado-y-el-futuro>)
- Lengnick-Hall, M. L., & Aguinis, H. (2012). What is the value of human resource certification? A multi-level framework for research. *Human Resource Management Review*, 22(4), 246–257. <https://doi.org/10.1016/j.hrmr.2011.03.001>
- Mejía Henao, V. (2009). La informática y su contribución a la automatización de procesos. *Lupa Empresarial*, 9(9). <https://revistas.ceipa.edu.co/index.php/lupa/article/view/385>
- Silva, J., Solano, D., Fernandez, C., Romero, L., & Villa, J. V. (2019). Privacy Preserving, Protection of Personal Data, and Big Data: a Review of the Colombia Case. *Procedia Computer Science*, 151, 1213–1218. doi:10.1016/j.procs.2019.04.174
- Slack. (2023). ¿Qué es Slack? Slack Help Center. <https://slack.com/intl/es-co/help/articles/115004071768-%c2%bfQu%c3%a9-es-Slack>
- TESORO, J, y SAROKA, H. (1993): Automatización administrativa y desempeño gerencial. 249-257. En ADMINISTRACIÓN DE EMPRESAS, Vol XV
- Toal, R. (2021, Noviembre 19). What is Back-End Development? - Code Institute IE. Code Institute IE. <https://codeinstitute.net/global/blog/what-is-back-end-development/>