

JUNIO DE 2024



Centre universitari adscrit a la



Grado en Ingeniería Informática de Gestión y Sistemas de la Información

MEMORIA FINAL “NIMIO”

TRABAJO DE FINAL DE GRADO

AUTOR: POL MORENO CASTELAR
TUTOR: JOSEP ROURE ALCOBÈ
2023-2024

RESUMEN

Este proyecto de fin de carrera se enfoca en el desarrollo del backend de una aplicación destinada al control de las recogidas de alumnos en las escuelas. Asimismo, abarca temas de investigación relacionados con la comercialización a largo plazo de la solución y un análisis del público potencial que podría utilizarla en el futuro.

Para la ejecución de este proyecto, se ha dividido en dos secciones: una de naturaleza teórica y otra práctica. En la primera, se realiza un exhaustivo proceso de investigación que incluye la identificación y análisis de las necesidades del público potencial. En la segunda sección, se abordan estas necesidades desde la perspectiva del desarrollo del backend de la aplicación.

El objetivo principal de este proyecto no es desarrollar una aplicación completamente funcional, dado que el calendario de ejecución lo hace inviable. Sin embargo, el proyecto proporciona una base sólida para un desarrollo escalable que podría llevar a una aplicación capaz de satisfacer una necesidad real en la sociedad actual. Este enfoque inicial permite sentar las bases para futuras ampliaciones y mejoras que podrían culminar en una solución efectiva y ampliamente adoptada en el ámbito educativo.

RESUM

Aquest projecte de fi de carrera s' enfoca en el desenvolupament del backend d' una aplicació destinada al control de les recollides d' alumnes a les escoles. Així mateix, abasta temes de recerca relacionats amb la comercialització a llarg termini de la solució i una anàlisi del públic potencial que podria utilitzar-la en el futur.

Per a l' execució d' aquest projecte, s' ha dividit en dues seccions: una de naturalesa teòrica i una altra pràctica. En la primera, es realitza un exhaustiu procés de recerca que inclou la identificació i anàlisi de les necessitats del públic potencial. En la segona secció, s' aborden aquestes necessitats des de la perspectiva del desenvolupament del backend de l' aplicació.

L' objectiu principal d' aquest projecte no és desenvolupar una aplicació completament funcional, atès que el calendari d' execució el fa inviable. Tanmateix, el projecte proporciona una base sòlida per a un desenvolupament escalable que podria portar a una aplicació capaç de satisfer una necessitat real en la societat actual. Aquest enfocament inicial permet establir les bases per a futures ampliacions i millores que podrien culminar en una solució efectiva i àmpliament adoptada en l' àmbit educatiu.

SUMMARY

This final project focuses on the development of the backend of an application aimed at controlling student collections in schools. It also covers research topics related to the long-term commercialization of the solution and an analysis of the potential audience that could use it in the future.

For the execution of this project, it has been divided into two sections: one of theoretical nature and the other practical. In the first, an exhaustive research process is carried out that includes the identification and analysis of the needs of the potential audience. In the second section, these needs are addressed from the perspective of backend development of the application.

The main objective of this project is not to develop a fully functional application, since the execution schedule makes it unfeasible. However, the project provides a solid foundation for scalable development that could lead to an application capable of satisfying a real need in today's society. This initial approach allows laying the foundations for future extensions and improvements that could culminate in an effective solution widely adopted in the educational field.

ÍNDICE

<i>Resumen</i>	1
<i>Resum</i>	2
<i>Summary</i>	3
<i>Introduccion</i>	7
<i>Justificacion</i>	9
<i>Planificación y calendario</i>	11
Entrega intermedia	11
Entrega final	11
Presentacion final	11
Definición de las tareas a realizar y estimación temporal	12
Diagrama de Gantt.....	21
Camino critico	21
<i>Definición de requerimientos</i>	22
<i>Definicion de la necesidad</i>	24
<i>Metodología</i>	27
<i>Limites del Proyecto</i>	28
<i>Argumentacion de la solucion</i>	30
Objetivo	30
Actores.....	30
Funcionamiento	31
Flujo de Recogida 1	34
Flujo de Recogida 2	35
<i>Entrevista a potenciales clientes</i>	36
<i>Ley GDPR y datos</i>	44
Que datos se van a recoger concretamente	45
<i>Definición de la solución planteada en la práctica</i>	47
Metodo de planificacion.....	47
Diagrama de clases	48
Base de datos y estructura de la información	48

Base de datos con JPA	50
Descripción de funcionalidades y requerimientos.....	51
Endpoints	51
School	53
Course	54
Teacher.....	55
Schedule.....	56
Class.....	56
Collector.....	57
Person.....	58
Student	59
Extra Endpoints.....	60
Definición de classes	64
Center.....	64
School	64
Course	64
Teacher.....	65
Schedule.....	65
Class.....	65
Student	66
Collector.....	66
Person.....	66
Acceso a la gestión del proyecto y código Fuente	67
Funcionamiento de JIRA y GIT.....	67
Guía de ejecución del proyecto	70
Requerimientos	70
Logins y enlaces.....	70
Fin guía de ejecución del proyecto	77
<i>Creación del modelo de negocio y venta de la solución.....</i>	<i>79</i>
Propuesta de valor:	79
Segmento de mercado:	79
Estrategia de precios:	80
Canales de distribución:	80
Estructura de costes:	81
Potencial de beneficios:	83
Posicionamiento competitivo:	84
Estrategia de marketing:	84
Plan de acción:	84

Conclusiones	86
Parte teórica	86
Parte practica	87
Valoración Final	89
Objetivo del Proyecto	89
Alcance del Proyecto	89
Objetivos Cumplidos Durante el TFG	89
Límites del Proyecto	90
Alcance del Proyecto	90
Objetivos del Producto y del Cliente	90
Conclusión General.....	91
Valoración Personal	92
Posibles ampliaciones	93
Bibliografía	94

INTRODUCCION

El proyecto que se ha llevado a cabo durante todo este tiempo se puede resumir en el desarrollo del backend de una aplicación que gestiona el control y autorización de la recogida de alumnos en centros escolares y el plan de comercialización de esta.

Hoy en día se está viviendo una época donde existe una gran preocupación por el control y la seguridad en todos los ámbitos, dada la gran cantidad de exposición de los datos de las personas en lugares tan inseguros como internet o simplemente la seguridad como es la de llevar una aplicación en el móvil que permita a tus familiares y amigos saber dónde estás en caso de emergencia.

Es por ello por lo que en este proyecto se desarrolla una nueva medida de seguridad y control, pero aplicada al sector escolar. Se ha desarrollado el backend de una aplicación que permite gestionar quien está autorizado y quien no, para la recogida de un alumno en las escuelas, de manera que en todo momento se dispone de un histórico de quien ha venido a buscar a quien cada día y quien está autorizado y quien no. A simple vista puede parecer que no se comprende al 100% que necesidad cubre la solución, pero a medida que se lea la memoria se comprenderá en su totalidad esta necesidad a la que se le ha dado una solución.

La idea de la creación de esta solución viene dada de una captación de la necesidad derivada directamente de una entrevista con una profesional del equipo psicopedagogo, a la que se le propusieron diversas soluciones enfocadas al sector escolar y la profesional concluyo en que el desarrollo de NIMIO sería una buena solución aplicable al mundo escolar y hoy en día sin competencia existente.

El desarrollo de esta solución y la aplicación de esta en centros escolares contribuiría a aparte de a dar una solución al problema que se detallara más adelante en la memoria, también dará un plus de seguridad a los centros escolares que implanten la solución y por lo cual generara una necesidad de compra para los centros que quieran vender una imagen de seguridad de cara a las familias.

La motivación principal por la que se ha llevado a cabo esta solución viene dada por la gran cantidad de robos/secuestros de alumnos de pequeña edad en centros escolares. Se han investigado múltiples casos con situaciones totalmente diferentes entre sí donde el alumno

en cuestión ha sido entregado a una persona que no era la correcta. Estos casos que a simple vista parece que no existan, sí que existen, puede venir des de una simple equivocación en la que una persona mayor por ejemplo se lleva a un niño que no es su nieto o puede derivarse a casos como parejas con problemas de custodia donde la persona sin custodia del hijo se lleva al alumno sin autorización, o podría extrapolarse hasta el robo/secuestro de alumnos donde el que recoge al alumno se hace pasar por su responsable o familiar.

En el periodo del desarrollo de la aplicación y de la comercialización de esta han aparecido dudas preguntas problemas hipótesis a las que se les ha ido dando una respuesta, preguntas como:

- ¿Como se identificará si el usuario que recoge al alumno es el correcto?
- ¿Como se registrarán los usuarios?
- ¿Como cumpliremos la ley de protección de datos?
- ¿Cuál es el flujo de uso de la aplicación puesta en funcionamiento?
- ¿Como los usuarios pueden autorizar a otro usuario para recoger a su hijo/a?
- ¿Qué modelo de venta se presentará al cliente potencial?
- ¿Como se quiere facturar el producto final?

Todas estas quedan claramente contestadas en las instrucciones del funcionamiento de la aplicación y a lo largo de la memoria se verán respuestas y justificadas.

La metodología que se ha usado para el desarrollo del proyecto ha sido una metodología AGILE se ha dividido en etapas, comenzando con la definición de requisitos y prototipos seguido de iteraciones de desarrollo de funcionalidades y pruebas de estas. I para la gestión de todo ello se ha hecho uso de la herramienta de planificación JIRA integrada con Github para un control más exhaustivo de los cambios.

JUSTIFICACION

El proyecto propuesto aborda una problemática crucial en el sistema educativo actual, centrándose en la seguridad en el aula, específicamente en el control de acceso y salida de los alumnos. Hasta la fecha, este control se ha limitado principalmente a registros rudimentarios, como apuntar en un papel o en un portal web si un alumno ha faltado a clase. Sin embargo, carece de detalles importantes, como quién ha llevado al alumno, a qué hora y quién está autorizado para recogerlo en caso de que la persona designada no pueda hacerlo.

La importancia del problema al que se le ha encontrado una solución es un tema muy preocupante de cara a las familias, ya que son ellas las que viven de primera mano la importancia de saber que su hijo/a, este a salvo. Cabe tener en cuenta que se pueden diferenciar distintos públicos dentro de las familias ya que puede que haya familias que no vean la utilidad del proyecto que se propone ya que en su situación no tienen ningún problema con el simple hecho de “¿Quién va a recoger al niño hoy?” aunque hay muchas otras como familias divorciadas, familias con problemas de tutela, familias desestructuradas, situaciones anormales o simplemente familias con problemas de organización para recoger a su hijo/a donde la solución les será de gran utilidad.

Según los datos recopilados por la INE Instituto Nacional de Estadística [1], se puede observar que la tendencia de Divorcios sin llegar a un acuerdo de custodia llego a un pico en el año 2020 pero des del 2020 en adelante se prevé un despunte de divorcios con problemas de custodia que se podría traducir directamente a potenciales clientes para la solución propuesta.

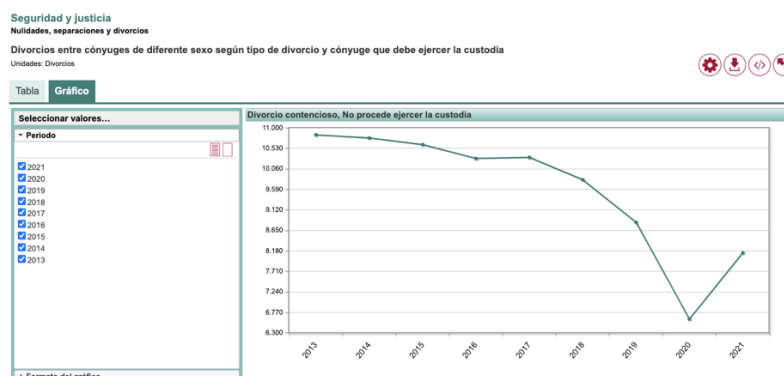


Ilustración 1 : Datos INE Divorcios

Aun analizando estos datos y teniendo en cuenta cual sería el cliente potencial del proyecto aún quedan brechas en la investigación que se abordaran a medida que se avanza en su desarrollo, es por ello que se realizaran entrevistas tanto al sector profesional como a familias de diferentes distribuciones para tener en cuenta posibles conocimientos datos o aspectos importantes que se hayan pasado por alto.

En cuanto al aspecto más personal dentro del desarrollo del proyecto cabe decir que la idea de esta solución viene dada por la escucha de un problema escolar narrado por una profesional del sector escolar, por la parte de un ingeniero con más de 25 años de experiencia en el sector de las soluciones aplicadas a problemas reales. Concretamente el padre y madre del desarrollador que es finalmente quien le motivo la idea de llevar esta solución a un proyecto palpable y con un objetivo junto con el problema o necesidad determinado por la profesional del sector educativo junto con la solución propuesta por el ingeniero.

Teniendo en cuenta todo lo mencionado cabe decir que la viabilidad del desarrollo y comercialización del proyecto es sumamente alta ya que han sido contemplados los escenarios en los que se encontrara el proyecto y se han hecho presupuestos y análisis temporales para determinar hasta donde llegara el proyecto dentro del periodo del TFG. En primera instancia cabe remarcar otra vez que en este periodo solo se llevara a cabo el desarrollo del backend junto con el estudio de mercado y creación del modelo de negocio.

En conclusión, este proyecto no solo aborda una necesidad apremiante en el sistema educativo, sino que también demuestra una respuesta innovadora y eficaz para mejorar la seguridad en las escuelas. Al considerar las diversas situaciones familiares y los datos estadísticos actuales, se confirma la relevancia y la amplitud del impacto potencial de esta solución. La combinación de la experiencia técnica y la perspicacia profesional ha dado lugar a un enfoque integral que promete beneficios tangibles para las familias y el entorno educativo en general.

PLANIFICACIÓN Y CALENDARIO

La planificación inicial del proyecto se gestionará de la siguiente manera:

En primer lugar, se debe tener en cuenta que se debe desarrollar tanto la parte teórica como la práctica por lo que será necesario identificar la cantidad de tiempo que se dedicará a cada una y la simultaneidad de en algún momento hacer teórica y práctica.

Para ello se ha desarrollado un calendario en formato mensual que permitirá tener un cierto control y orden de las tareas y de cuando deberían estar listas.

Dado que el trabajo tiene 2 entregas, la memoria intermedia y la final, y la presentación final como tal. De manera que se ha organizado de la siguiente manera.

ENTREGA INTERMEDIA

Los meses de enero febrero y marzo que son los 3 primeros meses, se destinarán única y exclusivamente al desarrollo de la parte teórica. Para organizar la parte teórica se ha creado un índice con los puntos teóricos a tratar y se ha asignado un tiempo de realización para cada uno de ellos aproximado para poder hacer una idea del tiempo de demora que supondrán y la dependencia que habrá entre ellos.

ENTREGA FINAL

Una vez concluida la parte teórica y ya hecha la entrega intermedia del proyecto, se empezará con el desarrollo es decir con la parte práctica. Este desarrollo se complementará también con una documentación teórica que formará parte de la parte teórica del trabajo por lo que será algo simultáneo la teoría más la práctica. Esta parte de desarrollo más documentación se llevará a cabo durante los meses de abril y mayo.

PRESENTACION FINAL

Finalmente, el mes de junio se dedicará única y exclusivamente a ultimar detalles si es necesario y a la preparación de la presentación oral, i también se realizará la entrega final del trabajo.

DEFINICIÓN DE LAS TAREAS A REALIZAR Y ESTIMACIÓN TEMPORAL

Para poder determinar una planificación sobre que se va a hacer cada mes se deben identificar las tareas tanto de la parte teórica como de la parte práctica.

Para determinar las tareas de la parte teórica se ha desarrollado un índice con los puntos a tratar y el periodo de tiempo que llevara el desarrollo de cada uno de ellos. Los puntos son los siguientes

Tarea 1

- **Descripción:** Redactar la introducción y justificación del proyecto.
- **Contexto:** se redactarán la introducción y la justificación del trabajo donde se especificará el objetivo de este y de que tratará el conjunto de todo el proyecto.
- **Duración:** se prevén 4 días para completar esta tarea.

Tarea 2

- **Descripción:** Definir la necesidad detectada
- **Contexto:** se definirá en profundidad la necesidad que se ha detectado al igual que se detallara la fuente de la necesidad basándonos en la justificación del porqué del desarrollo de la aplicación.
- **Duración:** se prevén 5 días para completar esta tarea.

Tarea 3

- **Descripción:** Argumentar la solución encontrada
- **Contexto:** se concretará la solución planteada para el desarrollo y se presentarán los requerimientos y objetivos de la aplicación que se habrán desarrollado previamente.
- **Duración:** se prevén 5 días para completar esta tarea.

Tarea 4

- **Descripción:** Investigar cumplimiento y requisitos de la ley de GDPR

- **Contexto:** se redactará información relacionada con el GDPR que incumba al desarrollo de la aplicación para verificar que el producto final cumpla con las cláusulas legales.
- **Duración:** se prevén 4 días para completar esta tarea.

Tarea 5

- **Descripción:** Argumentar Límites del proyecto
- **Contexto:** se justificará y aclarará los límites del proyecto dentro de la entrega del TFG y se concretará cuáles son los objetivos a largo plazo para el proyecto.
- **Duración:** se prevén 5 días para completar esta tarea.

Tarea 6

- **Descripción:** Definir la solución planteada junto con la parte práctica
- **Contexto:** se justificará y explicará todo el trabajo realizado en la parte práctica del proyecto de manera que quede claro cuáles han sido todos sus pasos y porque han sido efectuados (metodología usada, tecnologías, clases usadas, arquitectura, bases de datos). Esta tarea es la que más demora supone ya que se debe esperar a finalizar la parte práctica para desarrollarla.
- **Duración:** se prevén 6 días para completar esta tarea.

Tarea 7

- **Descripción:** Creación del modelo de negocio y la venta de la solución
- **Contexto:** se creará un modelo de negocio para la comercialización de la aplicación y se argumentará su viabilidad a largo plazo junto con su rentabilidad.
- **Duración:** se prevén 6 días para completar esta tarea.

Tarea 8

- **Descripción:** Redacción de las Conclusiones
- **Contexto:** se sintetizarán todos los conocimientos adquiridos a lo largo del proyecto y se plasmarán sobre la conclusión.
- **Duración:** se prevén 4 días para completar esta tarea.

Tarea 9

- **Descripción:** Bibliografía
- **Contexto:** se recopilarán todas las fuentes usadas durante toda la parte teórica y se añadirán a la bibliografía en su formato especificado.
- **Duración:** se prevén 3 días para completar esta tarea.

Tarea 10

- **Descripción:** Recopilar y organizar documentación utilizada
- **Contexto:** se anexarán al proyecto toda la documentación utilizada para la realización de toda la parte teórica junto con las entrevistas realizadas.
- **Duración:** se prevén 4 días para completar esta tarea.

N °	Tarea	Tiempo (Días)
1	Redactar la introducción y justificación del proyecto	4
2	Definir la necesidad detectada	5
3	Argumentar la solución encontrada	5
4	Investigar cumplimiento y requisitos de GDPR	4
5	Argumentar los límites del proyecto	5
6	Definir la solución planteada junto con la parte practica	6
7	Creación del modelo de negocio i la venta de la solución	6
8	Redacción de las conclusiones	4
9	Bibliografía	3
10	Recopilar y organizar documentación utilizada	3

Dados estas tareas que se convertirán finalmente en puntos de teoría de la parte teórica del proyecto, se distribuirán de la siguiente manera:

- Durante el mes de enero se llevarán a cabo las tareas 1,2,3,4 lo que harán un total de 20 días de trabajo teniendo en cuenta la realización documentación y revisión de cada una de ellas.
- Durante el mes de febrero se llevarán a cabo las tareas 5,6,7 lo que harán un total de 17 días de trabajo teniendo en cuenta la realización documentación y revisión de cada una de ellas.

- Durante el mes de marzo se llevarán a cabo las tareas 8, 9, 10 lo que harán un total de 10 días de trabajo teniendo en cuenta la realización documentación y revisión de cada una de ellas y teniendo en cuenta que aún no se darán por finalizadas ya que se completaran a lo largo de la parte práctica.

Se han eliminado algunas tareas respecto a la primera entrega del anteproyecto ya que se han excedido de los límites establecidos i no se ha podido dedicar ese tiempo de más. Es por ello que en el siguiente apartado la primera tarea es la 13.

Por otra parte, una vez realizada la entrega intermedia se empezará con la parte práctica que podrán identificarse las siguientes tareas:

Tarea 13

- **Descripción:** Redacción de requerimientos y funcionalidades
- **Contexto:** se redactarán todos los requerimientos captados en la parte teórica de manera que queden claros y directos para que el desarrollo sea lo más fiel posible a lo que se pide por parte del usuario final.
- **Duración:** se prevén 2 días para completar esta tarea.

Tarea 14

- **Descripción:** Definición de la arquitectura y tecnologías a usar
- **Contexto:** se especificarán con exactitud las tecnologías que se usarán para el desarrollo del proyecto y se justificara el porqué de su elección.
- **Duración:** se prevén 2 días para completar esta tarea.

Tarea 15

- **Descripción:** Documentación y justificación de las decisiones (Tarea 7)
- **Contexto:** se redactarán todas las decisiones tomadas y se dejarán por escrito en la parte teórica para la correcta documentación del proyecto, apoyándonos en las conclusiones descritas en el apartado anterior.
- **Duración:** se prevé 1 día para completar esta tarea.

Tarea 16

- **Descripción:** Configuración del entorno
- **Contexto:** se tomarán en cuenta las decisiones tomadas y se configurara el entorno de trabajo para poder empezar con el desarrollo como tal.
- **Duración:** se prevé 1 día para completar esta tarea.

Tarea 17

- **Descripción:** Definición de Clases
- **Contexto:** se decidirán y crearán las clases necesarias para la aplicación siguiendo las pautas determinadas por la arquitectura que se haya decidido.

- **Duración:** se prevén 5 días para completar esta tarea.

Tarea 18

- **Descripción:** Desarrollo de las funcionalidades
- **Contexto:** se tendrán en cuenta todos los requerimientos captados y se desarrollarán todas las funcionalidades necesarias para cubrirlos al cien por cien.
- **Duración:** se prevén 9 días para completar esta tarea.

Tarea 19

- **Descripción:** Documentación y justificación de las decisiones (Tarea 7)
- **Contexto:** se explicarán todas las clases y funcionalidades creadas y el porqué de la creación de cada una de ellas y el requerimiento que cumplen y se dejarán por escrito en la parte teórica para la correcta documentación del proyecto.
- **Duración:** se prevén 3 días para completar esta tarea.

Tarea 20

- **Descripción:** Pruebas
- **Contexto:** se realizarán pruebas sobre todas las funcionalidades para verificar su correcto funcionamiento. Esta tarea en concreto puede alargarse ya que puede haber fallos que retrasen el proyecto.
- **Duración:** se prevén 4 días para completar esta tarea.

Tarea 21

- **Descripción:** Compilado e instalación del “Swagger”
- **Contexto:** se compilará el backend de la aplicación una vez dado por testado y se instalará el “Swagger” para facilitar las pruebas a la persona que evalué el proyecto y poder realizar demostraciones en directo el día de la presentación.
- **Duración:** se prevén 2 días para completar esta tarea.

Tarea 22

- **Descripción:** Documentación (Tarea 7)

- **Contexto:** se explicarán todas las pruebas realizadas y el funcionamiento del “Swagger” para que cualquier lector del proyecto pueda poner a prueba las funcionalidades. Todo se dejará por escrito en el punto 7 de la parte teórica.
- **Duración:** se prevén 4 días para completar esta tarea.

N°	Tarea	Tiempo (Días)
13	Redacción de requerimientos y funcionalidades	2
14	Definición de arquitectura y tecnologías a usar	2
15	Documentación y justificación de las decisiones (Tarea 7)	1
16	Configuración del entorno	1
17	Definición de clases	5
18	Desarrollo de funcionalidades	9
19	Documentación y justificación de las decisiones (Tarea 7)	3
20	Pruebas	4
21	Compilado e instalación de “Swagger”	2
22	Documentación (Tarea 7)	4

Dadas las tareas de la tabla al igual que con las anteriores, serán distribuidas de la siguiente manera:

- Durante el mes de abril se llevarán a cabo las tareas 13,14,15,16,17,18 lo que harán un total de 20 días de trabajo teniendo en cuenta la realización documentación y revisión de cada una de ellas
- Durante el mes de mayo se llevarán a cabo las tareas 19,20,21,22 lo que harán un total de 13 días de trabajo teniendo en cuenta la realización documentación y revisión de cada una de ellas. En este mes se ha decidido dar más margen de tiempo ya que las tareas de pruebas siempre pueden alargarse inesperadamente.

Finalmente dada por concluida y revisada la parte teórica y práctica y realizada la entrega final, ya solo quedara la preparación de la presentación oral junto con la entrega final.

Tarea 23

- **Descripción:** Preparación presentación Oral
- **Contexto:** se creará una presentación con los conceptos más importantes a destacar el día de la presentación y se realizaran ensayos para poder asegurar que se transmite correctamente la información al oyente.
- **Duración:** se prevén 20 días para completar esta tarea.

N °	Tarea	Tiempo (Días)
23	Preparación presentación oral	20

Con esta tarea concluirá el trabajo ya que el mes de junio hasta el día de la presentación se dedicará todo el tiempo a la preparación de la presentación oral. En esta tarea se incluye el proceso de el desarrollo de la presentación y de la mejora de las habilidades orales de cara a la presentación final.

Las tareas deberán ser llevadas a cabo en el orden que se ha planteado en las tablas para poder mantener una coherencia y sentido en su realización, aunque cabe destacar que la tarea 7 se retomara en la parte práctica ya que esta es complementaria a la práctica

DIAGRAMA DE GANTT

Para dar una visión más simple al conjunto de desarrollo del proyecto en su totalidad se ha creado un diagrama de Gantt con la ayuda de la aplicación **GanttProject**, que permite identificar todas las tareas tal y como han sido definidas i poder ver las dependencias entre sí.

En el diagrama que se puede ver a continuación se pueden diferenciar el espacio temporal de cada tarea tal y como se veía en las tablas anteriores, y también se puede ver la dependencia entre las tareas.

Se puede observar que la única tarea con dependencia de la parte teórica es la tarea 6, 9 y 10 ya que como se ha remarcado anteriormente, estas tareas teóricas se complementan de la parte práctica por lo que será necesario disponer de la parte teórica para poder completarlas.

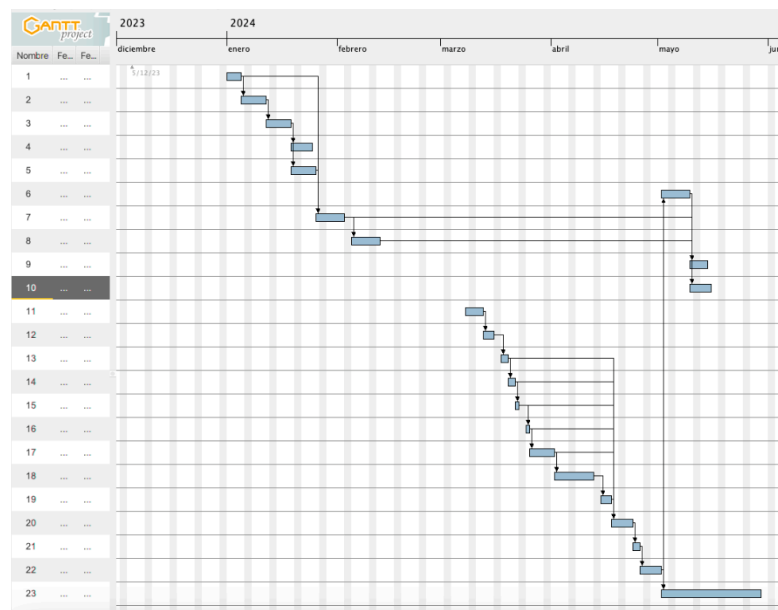


Ilustración 2: Diagrama de Gantt del Proyecto

CAMINO CRITICO

13-14-15-16-17-18-19-20-21-22-23

DEFINICIÓN DE REQUERIMIENTOS

Requerimientos Funcionales

1. Registrar Usuarios:

- Permitir a los usuarios crear cuentas con información personal.

2. Gestión de Perfiles:

- Permitir a los usuarios configurar perfiles con detalles específicos, como la información de contacto y la relación con el estudiante.

3. Sistema de Autenticación:

- Implementar un sistema seguro de inicio de sesión para garantizar la autenticidad de los usuarios. (Aunque para el desarrollo no será posible dentro del periodo de entrega del trabajo, pero si el desarrollo del sistema de “login” normal)

4. Registro de Recogida:

- Permitir a los padres registrar las recogidas autorizadas, especificando fechas y horas.

5. Alertas y Notificaciones:

- Enviar notificaciones a los padres sobre la llegada de sus hijos o cualquier cambio en los arreglos de recogida.

6. Gestión de Autorizaciones:

- Proporcionar una función para gestionar y verificar las autorizaciones de recogida, incluyendo la capacidad de actualizarlas.

7. Historial de Recogidas:

- Mantener un historial detallado de las recogidas, accesible para los padres y las autoridades escolares.

Requerimientos Tecnológicos

1. Escalable:

- Utilizar tecnologías actualizadas que permitan escalar el proyecto sin grandes dificultades.

2. Integrable:

- Aunque en el periodo de desarrollo dentro del TFG no se podrá desarrollar una integración como tal, cabe tener en cuenta que el sistema deba de ser integrable con centros escolares para su posterior comercialización.

3. Seguridad de Datos:

- Implementar medidas de seguridad robustas para proteger la información personal de los usuarios.

DEFINICION DE LA NECESIDAD

Para llevar a cabo una solución es importante una definición clara y concisa de la necesidad encontrada por lo que será de vital importancia definir la necesidad.

1. **Identificación del Problema:**

Se ha detectado una necesidad en la seguridad relacionada con la recogida de alumnos en los centros escolares. El problema principal es que no existe ninguna herramienta que permita al equipo escolar identificar si la persona quien viene a recoger un alumno es o no es la correcta.

2. **Contextualización:**

Por el momento hoy en día no existe ninguna solución al problema por lo que se trata de un desarrollo des de cero. También cabe añadir que aun no habiendo competencia se ha detectado una gran necesidad frente al problema

3. **Efectos y Consecuencias:**

La solución planteada da como efecto un control más exhaustivo sobre los alumnos y sobre sus responsables. A consecuencia de esto se evitarían problemas de equívocaciones, pérdidas, secuestro de niños. También cabe decir que se realizar un control más exhaustivo de los alumnos de manera que invadirá la privacidad de muchas familias sin darse cuenta, para ello se desarrollara una solución que se centre únicamente en los datos relevantes que realmente aporten valor a la solución final.

4. **Público Afectado:**

El público afectado por el problema o necesidad son directamente las familias y centros escolares, por un lado, las familias quieren tener el control y la seguridad de sus hijos/as asegurada, y por otro lado las escuelas quieren poder asegurarse de que cumplen con su trabajo i no realizan ningún error a la hora de entregar un niño a su correspondiente responsable.

5. Descripción Detallada:

La ausencia de un sistema de identificación adecuado se traduce en la imposibilidad de verificar la autenticidad de la persona que recoge a un alumno, lo que genera vulnerabilidades significativas.

Los riesgos incluyen situaciones potenciales de recogida equivocada, pérdida de niños durante el proceso y, en casos extremos, posibles situaciones de robo.

Se requiere el desarrollo de una solución integral desde cero que aborde no solo la identificación, sino también la seguridad y la privacidad de las familias.

6. Situaciones Específicas:

Recogida equivocada, pérdida de niños en el proceso, casos potenciales de robo, problemas de tutela de un menor, familias desestructuradas, casos de órdenes de alejamiento entre responsables.

7. Perspectivas Múltiples:

La perspectiva de las familias se centra en la seguridad y privacidad de sus hijos. La implementación de un sistema debe abordar estas preocupaciones de manera efectiva para ganar la confianza de los padres.

Desde la perspectiva de los centros escolares, la eficiencia y la prevención de errores son fundamentales. La solución debe ser práctica y fácil de implementar, garantizando la efectividad en la gestión de la seguridad.

8. Sensibilidad Cultural y Social:

Se deben tener en cuenta las variaciones culturales en las expectativas de privacidad para garantizar que la solución sea culturalmente sensible y aceptable de manera que el público acceda al uso de la solución sin tener la sensación de que esta comprometiéndolo sus datos.

La solución propuesta es adaptable a diferentes contextos sociales para abordar las necesidades específicas de diversas comunidades, es decir será tan válida como para una familia normal como para una familia con problemas o desestructurada.

9. Colaboración y Retroalimentación:

La colaboración con expertos en seguridad, tecnología y psicología infantil es esencial para desarrollar una solución integral y efectiva. Es por ello que se dispone de contacto directo con el equipo psicopedagogo de una escuela la cual no permite mencionar su nombre en este proyecto.

La obtención de retroalimentación continua de padres, personal escolar y otros interesados directos garantizará la validez y la aceptación de la solución propuesta. También cabe mencionar que será de vital importancia tener en cuenta el “feedback” que aportaran para estar en constante mejora.

METODOLOGÍA

Se ha realizado una revisión exhaustiva de investigaciones académicas relacionadas con la seguridad de los niños en entornos escolares. También se han llevado a cabo entrevistas con profesionales del ámbito educativo para obtener perspectivas prácticas.

La información recopilada se ha analizado en función de su relevancia y aportación. Se han establecido criterios para evaluar la calidad de las investigaciones académicas, y se ha aplicado un enfoque cualitativo al analizar las respuestas de las entrevistas para identificar patrones y tendencias significativas, es decir, opiniones que se repiten constantemente o quejas frecuentes relacionadas con la seguridad del entorno escolar.

Desarrollo de la aplicación

Se ha seguido una metodología ágil para el desarrollo de la aplicación. Se ha dividido en etapas, comenzando con la definición de requisitos y prototipos, seguido de iteraciones de desarrollo de funcionalidades y pruebas de estas. Se han empleado las tecnologías mencionadas en los puntos anteriores (Java, SpringBoot, MySQL).

Para evaluar diferentes enfoques en el desarrollo de la aplicación, se han establecido criterios de evaluación, incluyendo la seguridad, la usabilidad y la escalabilidad. Se han considerado alternativas y se ha seleccionado la solución más adecuada en base a un análisis comparativo.

Organización de la información

Se ha organizado la información utilizando una base de datos centralizada para los datos prácticos (Github + Memoria Local) y un sistema de gestión de proyectos (Onedrive + Word) para el material teórico. Esto garantiza la accesibilidad y la trazabilidad de la información a lo largo de todo el proceso de desarrollo del proyecto.

LIMITES DEL PROYECTO

En el caso de este proyecto se parte de un objetivo muy claro desde un principio, que es desarrollar una aplicación que dé respuesta a un problema presente en la actualidad y aparte desarrollar el plan de negocio para comercializar este producto, es decir de la manera que se podrá vender la solución propuesta.

Actualmente el alcance del proyecto como tal se centra en poder desarrollar el backend de la aplicación de manera que sea funcional, y desarrollar el plan de negocio de la venta de la aplicación una vez sea confeccionada al 100%.

Se consideraría fuera del alcance del proyecto dentro de este TFG el tema del lanzamiento de la aplicación al mercado real y el desarrollo de la parte visual (frontend) y funcional de la misma ya que esto ya formaría parte de los objetivos del proyecto a largo plazo y no los definidos de un principio para el TFG de este año 2023-2024.

Es decir, quedan claros los objetivos a alcanzar durante este curso al igual que quedan claros los límites y alcance del proyecto dentro del desarrollo que se llevara a cabo durante este curso.

Objetivos dentro del periodo de desarrollo del TFG:

- Dar un remedio a un problema existente dentro del sector escolar
- Plantear la comercialización de esta solución
- Demostrar la viabilidad del producto final y crear necesidad
- Desarrollo en JavaSpringBoot
- Control de BBDD con MySQL

Límites dentro del periodo de desarrollo del TFG:

- Mantener dentro del sector escolar
- Desarrollo de únicamente Backend

Alcance dentro del periodo de desarrollo del TFG:

- Desarrollo completo del Backend en Java SpringBoot de las principales funcionalidades

- Desarrollo de un plan de negocio
- Contacto con profesionales del sector para la validación de la aplicación y consultas sobre el sector

Objetivos del Producto

- Vender el software a centros escolares
- Ser una validación de seguridad para el sector escolar
- Crear un software usable e intuitivo
- Respetar las leyes de GDPR

Objetivos del Cliente

- Tener un control de recogida de sus niños/as
- Poder hacer cargo a otros usuarios con seguridad
- Tener confirmación de quién ha recogido al niño/a es el autorizado

Público potencial y target

Es cierto que el proyecto se podría usar en muchos ámbitos ya no solo en el sector escolar, pero en este caso está orientado al público de las familias/tutores legales, ya que son el cliente potencial de la aplicación.

Todos los objetivos mencionados cumplen la metodología SMART, por lo tanto, son específicos, medibles, posibles, realistas y de duración limitada, con respecto que estén totalmente alineados con el desarrollo del proyecto.

ARGUMENTACION DE LA SOLUCION

Para argumentar correctamente la solución que se propone en este proyecto será de vital importancia comprender el objetivo de la solución, los actores que forman parte de esta y el funcionamiento básico de la aplicación sin centrarnos de pleno en sus funcionalidades ya que de estas se hablara en adelante.

OBJETIVO

El objetivo principal como ya se ha remarcado en puntos anteriores es dar una solución a un problema con el control de recogida de los alumnos en centros escolares. A parte de solventar el problema el objetivo también es aportar un producto de valor al mercado que pueda satisfacer una necesidad real.

También aparte de objetivos como aplicación también hay objetivos des del punto de vista económico y personal, que es el hecho de desarrollar una solución des de cero y que esta pueda ser comercializada para poder obtener una rentabilidad sobre ella y tener un proyecto prometedor.

ACTORES

Para argumentar correctamente el funcionamiento de la solución se debe de tener en cuenta los actores que formaran parte de su funcionamiento. Para hacer una pequeña muestra de los roles o perfiles que pueden hacer uso de esta aplicación se ha creado una pequeña simulación:

Rol de Tutor/Madre/Padre

- **Rol de Profesor/a**
- **Rol de Autorizado/a**
- **Rol de Alumno**

Teniendo en cuenta estos 4 roles debemos de saber qué papel juega cada uno de ellos, para poder comprender a continuación el funcionamiento de la aplicación:

- **Tutor/Madre/Padre**

Tiene el rol de la persona responsable del alumno, es decir la que legalmente consta como la responsable de los actos del alumno. Tiene como poderes sobre el alumno el hecho de poder autorizar a terceros para la recogida de su niño/a de manera que también tiene el control de las recogidas de este.

- **Profesor/a**

Es la persona a cargo del alumno una vez ingresa en el centro escolar de alguna manera se podría considerar el encargado o el responsable dentro del centro. Tiene como deber, comprobar que el alumno es entregado a la persona correcta.

- **Autorizado/a**

Tiene el rol de la persona que es autorizada por un **Tutor/Madre/Padre** para realizar la recogida de un alumno, este sujeto no tiene ningún poder si no es autorizado previamente por uno de ellos. Sus límites de autorización tampoco pueden ser modificados por el sujeto con lo que de alguna manera está totalmente gestionado por otro rol.

- **Alumno/a**

Es el centro de toda la solución i será la persona que estará a cargo de un rol de **Tutor/Padre/Madre** y bajo la responsabilidad escolar de un **Profesor/a**. Este actor tampoco tiene ningún poder sobre sí mismo o los demás ya que no puede autorizar a terceros y tampoco puede decidir desautorizar a alguno de sus autorizados. Todos esos trámites recaen directamente sobre el **Tutor/Padre/Madre**.

FUNCIONAMIENTO

En este apartado solo se hará una pequeña descripción del flujo de funcionamiento de la aplicación sin indagar en las funcionalidades que tiene cada rol dentro de la aplicación por lo que únicamente haremos hincapié en el flujo de recogida de un alumno y más adelante en otros puntos se harán los flujos con las funcionalidades pertinentes de cada rol, como son la de añadir un autorizado, configurar un autorizado, consultar el histórico de un alumno, etc.

Antes de aclarar el flujo de funcionamiento de una recogida, se debe de tener en cuenta que es imprescindible tener identificados cada uno de los alumnos con una ID es decir un identificador único para cada uno que permita diferenciarlos. Cada uno de los alumnos

deberá disponer de un tag RFID para que estos puedan ser escaneados y consultados des de la aplicación por un profesor para poder verificar en el momento de la recogida si el que recoge al alumno es el indicado. Se ha decidido identificarlos con un tag RFID ya que este podría llevarse colgando de la mochila de manera que el alumno nunca se dejara la mochila en el aula ya que sino no podría abandonarla y a parte se ha hecho de esta manera ya que pensando en alumnos de pequeña edad es la única manera que no pierdan el identificador.



Ilustración 3 : Llavero RFID

Por otro lado, también cabe aclarar que los que recojan al alumno sin importar su rol, dispondrán de un QR en su aplicación que les permitirá identificar su ID de Tutor/Madre/Padre o Autorizado/a, con lo que este ID permitirá a la aplicación comprobar si pueden recoger a X alumno.

El flujo principal de una recogida de un alumno podría ser de dos maneras diferentes:

Flujo 1

1. Profesor escanea a alumno
2. Profesor escanea a recogedor
3. Aplicación valida si el recogedor puede llevarse al alumno
4. Recogedor se lleva al alumno o Recogedor no se lleva al alumno

Flujo 2

1. Recogedor solicita un alumno
2. Profesor escanea el tag RFID del alumno
3. Comprueba que el recogedor es el autorizado
4. Recogedor se lleva o no el alumno



**Ilustración 4:
Identificador Recogedor**

Para aclarar este flujo de funcionamiento se ha hecho un diagrama de flujo para poder aclarar correctamente como sería el funcionamiento de la funcionalidad principal.

FLUJO DE RECOGIDA 1

En este primer flujo se puede ver el “modus operandi” de que el recogedor es quien enseña su código QR y posteriormente se va a buscar al alumno/a con el match del QR en la aplicación.

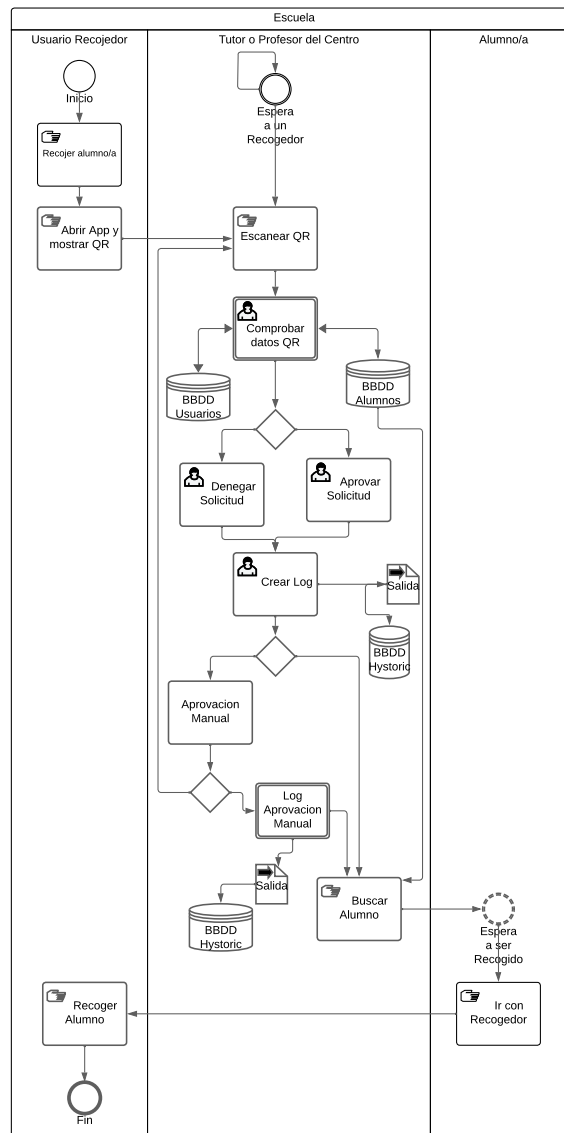


Ilustración 5: Flujo de Recogida 1

FLUJO DE RECOGIDA 2

En este segundo flujo a diferencia del primero se plantea un caso donde el recogedor nombra al alumno y el profesor/a va a buscarlo físicamente y comprueba si el usuario que lo va a recoger concuerda con el autorizado del alumno, escaneando el tag RFID del alumno.

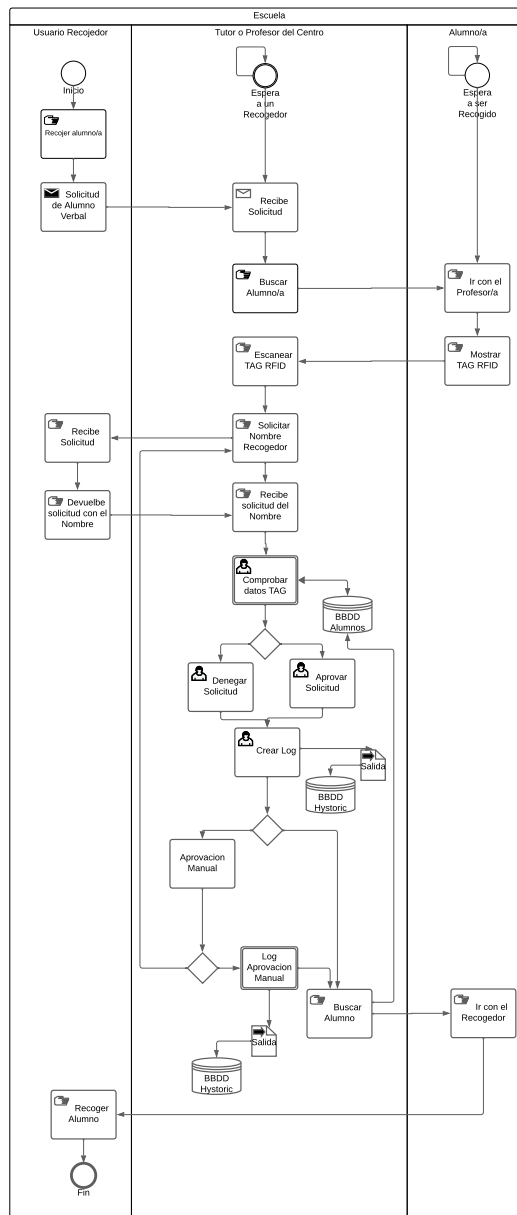


Ilustración 6: Flujo de Recogida 2

ENTREVISTA A POTENCIALES CLIENTES

Para acercarse más al cliente final, se ha creado una encuesta destinada a familias con infantes a su cargo, para conocer en profundidad la opinión que tienen las familias respecto al tema de la seguridad en los centros escolares y aparte conocer necesidades que se puedan haber pasado por alto en primera instancia.

Las preguntas que se han incluido en la encuesta serán adjuntadas a continuación i para cada una de ellas se argumentara el porqué de la pregunta, y que datos relevantes pueden aportar:

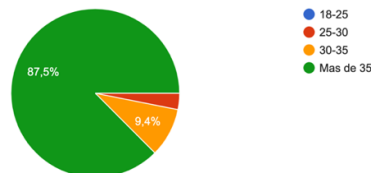
Pregunta 1

¿Qué edad tienes?

Esta pregunta proporciona información demográfica útil para segmentar la audiencia. Conocer la edad de los encuestados ayudará a entender mejor las preferencias y necesidades de diferentes grupos de edad.

Resultados

¿Qué edad tienes?
32 respuestas

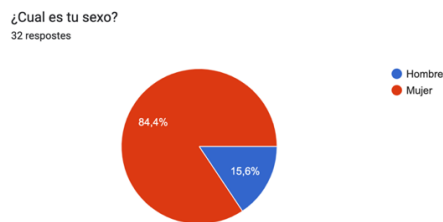


Pregunta 2

¿Cuál es tu sexo?

Esta pregunta ayudará a comprender la composición de género de la audiencia objetivo. Esto puede ser útil para personalizar la experiencia del usuario, adaptar el lenguaje y el diseño de la aplicación para que sea inclusiva y relevante para todos los usuarios.

Resultados



Pregunta 3

¿Tienes un infante o más a tu cargo?

Esta pregunta básicamente dará la información necesaria para contemplar la respuesta con más o menos interés ya que si no tiene infantes a cargo no es un potencial cliente por lo que su opinión no la tomaremos tan en cuenta como la de una persona que pueda acabar siendo un cliente final.

Resultados



Pregunta 4

¿Valoras la seguridad cuando inscribes un infante centro escolar?

Esta pregunta ayudará comprender la importancia que los encuestados dan a la seguridad al inscribir a sus hijos en un centro escolar. Sus respuestas brindarán información valiosa sobre las preocupaciones de los padres en relación con la seguridad de sus hijos en entornos escolares. Esto puede influir en el enfoque de la aplicación y en las características que deberían priorizarse

Resultados



Pregunta 5

¿Qué elementos de seguridad tiene el centro escolar donde esta escolarizado?

La información que proporcionará esta pregunta permitirá comprender mejor el contexto en el que se utilizará tu aplicación y cómo puede complementar o mejorar los sistemas de seguridad existentes en las escuelas.

Resultados

Tras hacer un análisis sobre los resultados obtenidos los más repetidos han sido los siguientes:

- Alarma de incendios
- Alarma de robo
- Sistemas de simulacro
- Señalización de Incendios
- Sistemas de comunicación de emergencia

Todas las respuestas se han centrado mucho en un nicho de seguridad que está realmente explotado, por lo que en ningún centro existe un elemento de seguridad relacionado con nada de lo que se quiere desarrollar en esta solución.

Pregunta 6

¿Qué elementos de seguridad te gustaría que tuviera la escuela?

Esta pregunta proporcionará información que ayudará a identificar áreas de oportunidad para el desarrollo de tu aplicación, centrándote en las características de seguridad que los padres consideran más importantes y necesarias.

Resultados

Tras hacer un análisis sobre los resultados obtenidos, los más repetidos han sido los siguientes:

- Cámaras de vigilancia
- Revisión de recogida de alumnos / Control de Acceso
- Personal de seguridad

Resultados que dan aún más pie a considerar la solución como una buena idea de cara a lo que está buscando el público potencial.

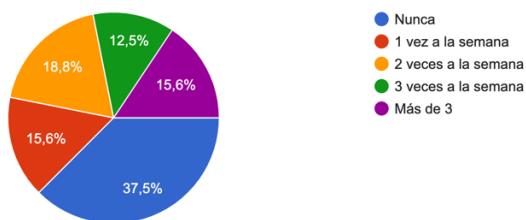
Pregunta 7

Con que frecuencia dejas a cargo tus infantes de personas terceras ya sean familia o conocidos para la recogida del infante en el centro escolar.

Esta pregunta ayudara a hacer una idea de la cantidad de operaciones que se llevaran a cabo durante el uso de la aplicación y ver si realmente se hará uso de la aplicación como tal en un entorno real.

Resultados

Con que frecuencia dejas a cargo tus infantes de personas terceras ya sean familia o conocidos para la recogida del infante en el centro escolar.
32 respuestas



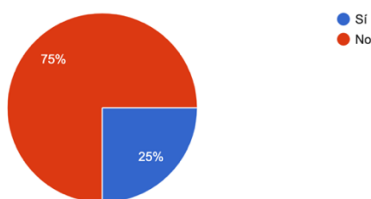
Pregunta 8

¿Alguna vez te has planteado la situación de que tu hijo/hija que has acordado que recogiera una tercera persona en la escuela, fuera recogido por otra no autorizada?

Las respuestas esta pregunta proporcionarían información sobre las experiencias pasadas o las preocupaciones potenciales de los padres en cuanto a la posibilidad de que sus hijos sean recogidos por personas no autorizadas.

Resultados

¿Alguna vez te has planteado la situación de que tu hijo/hija que has acordado que recogiera una tercera persona en la escuela, fuera recogido por otra no autorizada?
32 respuestas



Aunque parezca contradictorio el generar el pensamiento de esta desagradable situación hace que se cree una necesidad para evitarla tal y como se ve en los resultados de la siguiente pregunta.

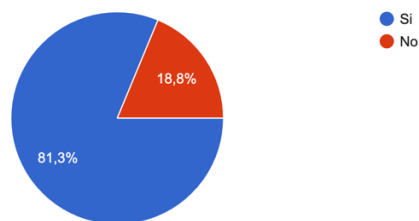
Pregunta 9

¿Te gustaría disponer de un control sobre las entradas y salidas de tu hijo en la escuela junto con la persona que lo ha recogido como medida de seguridad activa?

Las respuestas a esta pregunta proporcionarían información sobre la disposición de los padres a utilizar herramientas de seguimiento y control para garantizar la seguridad de sus hijos. Esta información es crucial para determinar la viabilidad y la aceptación de características específicas en la aplicación, como la capacidad de monitorear y registrar las entradas y salidas de los niños y las personas autorizadas que los recogen.

Resultados

De haber respondido que si crees que estarías interesado/a en una solución que permitiera evitar esa inesperada situación.
32 respuestas



Dados los resultados se puede interpretar que claramente aun no habiéndose planteado las situaciones, les interesaría abordarla a toda costa.

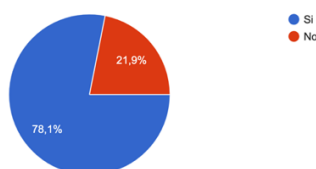
Pregunta 10

¿Si has respondido que no en alguna de las anteriores, porque no te interesaría teniendo en cuenta que se está luchando por la seguridad de tu hijo/a?

Las respuestas a esta pregunta proporcionarían información sobre posibles barreras o preocupaciones relacionadas con la privacidad, la confianza en el sistema escolar, la conveniencia o la percepción de la efectividad de tales medidas de seguridad. Comprender estas razones es crucial para adaptar la aplicación y abordar las preocupaciones de los usuarios de manera efectiva y confiable para ellos.

Resultados

¿Te gustaría disponer de un control sobre las entradas y salidas de tu hijo en la escuela junto con la persona que lo ha recogido como medida de seguridad activa?
32 respuestas



Estos resultados son útiles para comprender si realmente el público está de acuerdo con el tratamiento que se haría sobre la información de su infante.

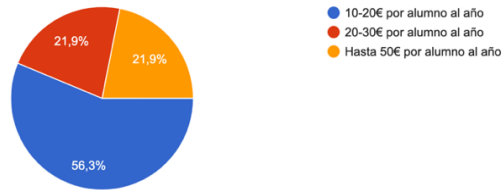
Pregunta 11

¿Si nos encontrásemos en una situación donde el centro escolar al que asiste el infante dispone de un servicio de seguridad contratable con un suplemento de pago anual de X€ por infante, que cifra estarías dispuesto/a a pagar por este servicio?

Las respuestas a esta pregunta proporcionarían información sobre el valor percibido de este servicio por parte de las familias y su disposición a invertir en la seguridad de sus hijos. Esta información es crucial para determinar la viabilidad financiera de implementar este servicio.

Resultados

¿Si nos encontrásemos en una situación donde el centro escolar al que asiste el infante dispone de un servicio de seguridad contratable con un suplemento...¿estarías dispuesto/a a pagar por este servicio?
32 respuestas



Los resultados a esta pregunta nos servirán de cara a la realización del modelo de negocio, para poder determinar de manera provisional un umbral de precios posibles para la solución.

Pregunta 12

¿Qué forma de pago convendría más a su núcleo familiar?

Las respuestas podrían proporcionar información sobre la capacidad financiera de los padres, así como sus preferencias en cuanto a comodidad y flexibilidad en los métodos de pago. Esto es importante para garantizar que el servicio sea accesible y atractivo para la mayoría de las familias.

Resultados

Los resultados a esta pregunta han sido generalmente que preferirían que se añadiera a la cuota escolar del infante, anualmente.

LEY GDPR Y DATOS

Para llevar a cabo un correcto desarrollo de la aplicación, es necesario conocer a la perfección el marco legal donde se encuentra el proyecto. Cabe destacar que, al tratarse de datos relacionados con la identidad de las personas, son datos que requieren de una cierta atención a su cumplimiento para evitar problemas legales en el momento de lanzarla al mercado.

La ley principal que debe ser tomada en cuenta es el Reglamento General de Protección de Datos (GDPR). Aquí hay algunas pautas clave dentro del GDPR que se deberán de tener en cuenta para el desarrollo:

1. **Consentimiento:** Se debe obtener el consentimiento explícito de los individuos cuyos datos se están recopilando y procesando. Este consentimiento debe ser libre, específico, informado e inequívoco. Es decir, en el momento que un usuario se da de alta en la plataforma se tendrá que disponer de su aprobación para el uso de los datos dentro de la aplicación.
2. **Transparencia y divulgación:** Se debe proporcionar información clara y transparente sobre cómo se recopilan, procesan y utilizan los datos personales. Esto incluye detalles sobre quién es el responsable del tratamiento de datos, la finalidad del procesamiento y cualquier otro aspecto relevante. Deberá de quedar reflejado que los datos recopilados solo se usaran para fines totalmente relacionados con el hecho de controlar la recogida del alumno y estos no se venderán a terceros o se usaran para ningún otro aspecto.
3. **Derechos de los interesados:** Se debe asegurar que los individuos cuyos datos se están procesando puedan ejercer sus derechos según lo establecido en el GDPR. Esto incluye el derecho de acceso, rectificación, supresión, limitación del procesamiento, portabilidad de datos y oposición al procesamiento. Dentro de la aplicación el usuario es libre en todo momento de modificar sus datos personales siempre y cuando no se haga pasar por una persona que no es para falsificar una autorización. Por lo cual dentro de los derechos de modificación de los datos se podría deducir la destrucción de ellos, más que la edición.

4. **Seguridad de los datos:** Se deben implementar medidas técnicas y organizativas adecuadas para garantizar la seguridad de los datos personales y protegerlos contra el acceso no autorizado, la divulgación, la alteración o la destrucción accidental o ilícita. Ya como se mencionaba en el punto anterior se deberán establecer medidas de seguridad para garantizar la integridad de los datos de los usuarios, que, aunque en esta primera versión de desarrollo no se vaya a hacer hincapié en este tema, cabe tenerlo en cuenta para cuando se haga la producción del producto final.
5. **Evaluaciones de impacto en la protección de datos (EIPD):** En ciertos casos, puede ser necesario realizar una EIPD para evaluar y mitigar los riesgos para la privacidad asociados con el procesamiento de datos personales.
6. **Responsabilidad y cumplimiento:** Se debe ser capaz de demostrar el cumplimiento con los principios y requisitos del GDPR, lo que puede requerir llevar registros detallados de las actividades de procesamiento de datos y cooperar con las autoridades de protección de datos. Para ellos se crearán LOG's dentro del sistema para poder demostrar que se está haciendo con cada uno de los datos en todo momento.

QUE DATOS SE VAN A RECOGER CONCRETAMENTE

Aunque en el modelo relacional de clases quedara mucho más claro los datos que se recogeran de cada uno de los usuarios/roles/perfiles que harán uso de la aplicación, deben de quedar claros los grupos de datos que necesitara la aplicación para dar un servicio mínimo:

- **Infante**
 - Nombre y Apellidos
 - DNI NIF o Numero identificativo
 - DNI NIF o Numero Identificativo de sus tutores legales
 - Lista de DNI NIF o Numero Identificativo de sus autorizados
- **Persona**
 - Nombre y Apellidos
 - DNI NIF o Numero identificativo

- Lista de DNI NIF o Numero Identificativo de sus autorizados
- Autorizado
- Padre/Madre

- **Escuela**

- Nombre
- Ubicación
- Centro
- Lista de Cursos
- Clases dentro de la escuela

- **Centro Escolar**

- Nombre
- Lista de Escuelas adscritas

DEFINICIÓN DE LA SOLUCIÓN PLANTEADA EN LA PRÁCTICA

A continuación, en este apartado se desarrollara toda la parte práctica del trabajo final, partiendo des de la planificación hasta el desarrollo como tal en Java. Es necesario recalcar que en esta parte práctica solo se realizara el desarrollo del backend de la aplicación con algunas de sus funcionalidades principales, ya que el tiempo limita mucho la posibilidad de extenderse en esta parte.

METODO DE PLANIFICACION

Antes de empezar con el desarrollo del código como tal hay que comprender en totalidad el problema al que nos vamos a enfrentar, cosa que se ha hecho a lo largo de toda la parte teórica. Una vez comprendido a la perfección el problema, el próximo paso sería diseñar un modelo de clases para poder determinar que clases serán las principales para desarrollar nuestro proyecto y a partir de ahí decidir las funcionalidades que desarrollaremos para cada una de ellas, en los siguientes puntos se entrara en detalles y se justificara la creación de cada una de las clases y de su principal utilidad.

Para llevar a cabo toda la parte práctica, es necesario desarrollar un método de planificación que permita realizar una cierta trazabilidad de cómo se va a avanzando en el desarrollo del proyecto. Para ello se ha hecho uso de las tecnologías de JIRA + Git. Para cada funcionalidad se ha asignado un “ticket” con un identificador que nos permitirá identificar el código que se ha realizado para desarrollar dicha funcionalidad, en apartados posteriores se hará hincapié en esta funcionalidad de JIRA + Git.

Para poder corregir y realizar pruebas sobre el proyecto, al igual que poder ver toda la trazabilidad del JIRA+GIT, se ha creado una sección con la **“Guía para la corrección y ejecución del proyecto”** donde queda detallado como acceder a todo lo necesario para disponer del proyecto en local.

DIAGRAMA DE CLASES

Para realizar el primer diagrama de clases se ha hecho uso de la aplicación de LucidChart que ha permitido plasmar de la manera más simple y entendedora como quedara nuestro diseño distribuido en las siguientes clases.

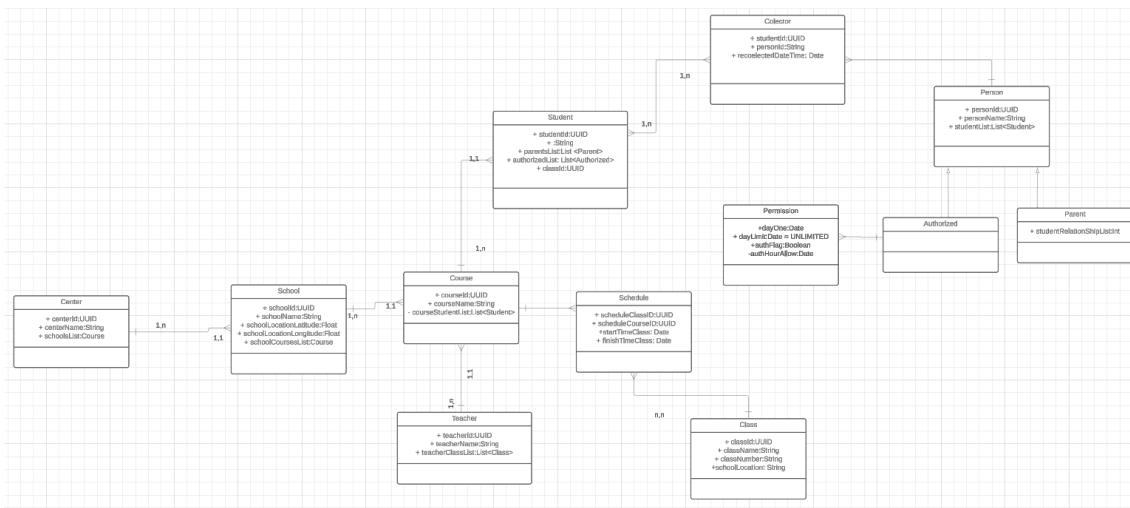


Ilustración 7 : Diagrama de clases

A medida que se avance en el desarrollo el modelo sufrirá cambios y posiblemente la agregación de nuevas clases. Con lo que esto es solo una imagen general de como quedara distribuido el diseño de la solución final.

BASE DE DATOS Y ESTRUCTURA DE LA INFORMACIÓN

Inicialmente, se consideró la opción de crear una base de datos desde cero y establecer la conexión utilizando JDBC a través de Spring. Sin embargo, tras un análisis más profundo de las necesidades del proyecto y las opciones disponibles, se tomó la decisión de adoptar la tecnología JPA (Java Persistence API).

JPA ofrece numerosas ventajas que hacen que sea una elección sólida para la capa de persistencia de datos en aplicaciones Java, entre las cuales se destacan:

Productividad Mejorada: JPA simplifica el desarrollo al eliminar la necesidad de escribir consultas SQL manuales y gestionar la asignación de objetos a tablas de base de datos. Las

anotaciones de JPA permiten mapear directamente clases Java a entidades de base de datos, lo que agiliza el proceso de desarrollo.

Portabilidad: Al utilizar JPA, se reduce la dependencia del proveedor de base de datos específico, lo que facilita la migración entre diferentes sistemas de gestión de bases de datos (DBMS) en el futuro. Esto proporciona una mayor flexibilidad y evita el bloqueo del proveedor.

Abstracción de la Base de Datos: JPA proporciona una capa de abstracción sobre la base de datos subyacente, lo que permite a los desarrolladores interactuar con los objetos de la base de datos de manera más intuitiva y orientada a objetos. Esto simplifica el código y mejora la legibilidad y mantenibilidad de este.

Rendimiento Optimizado: Los proveedores de JPA suelen implementar optimizaciones que mejoran el rendimiento de las consultas y transacciones, lo que puede traducirse en una mejor respuesta y tiempos de carga para la aplicación.

Soporte para Transacciones: JPA facilita la gestión de transacciones, lo que garantiza la integridad de los datos y la consistencia de la base de datos en entornos concurrentes.

En resumen, la adopción de JPA ofrece una solución robusta y eficiente para la capa de persistencia de datos en el proyecto, mejorando la productividad del desarrollo, la portabilidad, la abstracción de la base de datos, el rendimiento y el soporte para transacciones. Esto contribuye a la creación de una base de datos sólida y fácil de mantener, al tiempo que reduce la complejidad del código y facilita futuras actualizaciones y expansiones del sistema.

BASE DE DATOS CON JPA

Finalmente, después de realizar todas las relaciones entre clases y la arquitectura del software, el modelo de base de datos ha quedado distribuido de la siguiente manera.

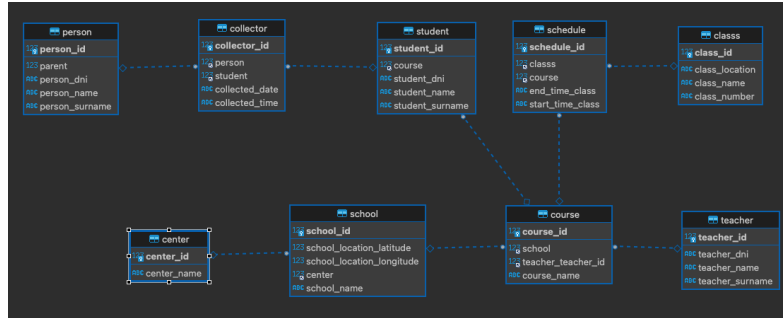


Ilustración 8: Esquema base de datos JPA

DESCRIPCIÓN DE FUNCIONALIDADES Y REQUERIMIENTOS

En este caso no partimos de un cliente que proporcione cuales son los requerimientos o needs que tiene sobre el producto final, pero, en un caso real el paso previo a determinar las funcionalidades de nuestra aplicación partiría de eso. Para poder hacerlo lo más fiel a la realidad posible se han creado requerimientos funcionales y se han determinado los endpoints que solventaran estos requerimientos.

Para ello se ha realizado un Excel de URS (User Requirements) donde se especifica un **identificador para cada requerimiento** junto con un **identificador para cada funcionalidad** i de cada funcionalidad su correspondiente **endpoint** con un enlace al **ticket de JIRA** donde se desarrolla esa funcionalidad en concreto.

A continuación, adjunto un enlace al Excel donde se puede visualizar de forma mucho más clara las URS:

https://tecnocampus-my.sharepoint.com/:x:/g/personal/pmorenoc_edu_tecnocampus_cat/EUtfgEgn1GZEorDAWGd2bhUBtMSg9tHzuu22ckQZYBmZSw?rttime=iBIDybN03Eg

ENDPOINTS

A continuación se mostraran cada uno de los endpoints que se han diseñado junto con la clase a la que pertenece cada uno de ellos y de que tipo son es decir POST, PUT, GET, DELETE. Toda la información sobre este apartado debe de ser complementada con el documento de URS donde queda referenciado a que funcionalidad hace referencia cada endpoint y a que requerimiento da respuesta y el ticket de Jira para cada uno.

A parte para hacer mucho más simple el test de todo el proyecto, se ha diseñado una colección de “PostMan”, con todos los endpoints prediseñados y identificados con el código de funcionalidad que se puede ver en el Excel de URS o en los tickets de Jira, para facilitar al usuario el test.

Es necesario tener presente esta librería de “PostMan”, previamente preparada para testear la aplicación, ya que en la descripción de los endpoints no se hace mención de todos los

atributos que se deben de rellenar en el body de cada petición ya que haría este documento algo largo y difícil de comprender.

Toda la información respecto a la ejecución y testeo del proyecto se encuentra en el apartado de “Guía de ejecución del proyecto”.

CENTER

ID Requerimiento	Descripción	Clase	ID Funcionalidad	Tipo	Ticket JIRA	Endpoints	Parámetros de entrada	Return
CE1	El sistema debe poder crear un centro escolar al que se le asociaran escuelas, al crearlo también se deben de poder editar las características posteriormente.	Center	CE1F	POST	https://nimio2024.atlassian.net/browse/NFP-9	/centers	Center center	Center
CE2	El sistema debe poder obtener todas las escuelas que pertenecen a un centro	Center	CE2F	GET	https://nimio2024.atlassian.net/browse/NFP-10	/centers/{centerId}/schools	Float centerId	List <School>
CE3	El sistema debe poder obtener todos los centros que hacen uso de la solución	Center	CE3F	GET	https://nimio2024.atlassian.net/browse/NFP-11	/centers	IDNAME	List <Center>
CE4	El sistema debe poder modificar los datos de un centro en concreto	Center	CE4F	PUT	https://nimio2024.atlassian.net/browse/NFP-12	/centers/{centerId}	Float centerId	Center
CE5	El sistema debe de poder eliminar un centro y con ello todas sus escuelas	Center	CE5F	DELETE	https://nimio2024.atlassian.net/browse/NFP-13	/centers/{centerId}	Float centerId	IDNAME

Ilustración 9: Endpoints de la clase Center

CE1F POST CENTER

Este endpoint permite al usuario crear un nuevo centro insertando en el body de la request únicamente un JSON con el atributo “centerName”. Este lo guarda en la base de datos y le asigna una ID, i de momento una lista de escuelas vacía.

CE2F GET SCHOOLS BY CENTER ID

Este endpoint permite al usuario consultar las escuelas de un centro, para ello se deberá agregar en el path del endpoint el Id del centro sobre el que queremos realizar la consulta.

CE3F GET ALL CENTERS

Este endpoint permite al usuario consultar todos los centros que existen en la base de datos, en este caso no es necesario agregar nada en el body ni en el path de la consulta.

CE4F UPDATE CENTER NAME BY ID

Este endpoint permite al usuario modificar el nombre de un centro ya existente, para ello es necesario agregar sobre el path el id del centro que deseamos modificar su nombre e insertar en el body un JSON con el atributo “centerName”, con el nuevo nombre que queremos asignar al centro.

CE5F DELETE CENTER BY ID

Este endpoint permite al usuario eliminar un centro, para ello es necesario agregar sobre el path el id del centro que se desea eliminar.

SCHOOL

ID Requerimiento	Descripción	Clase	ID Funcionalidad	Tipo	Ticket JIRA	Endpoints	Parametros de entrada	Return
SC6	El sistema debe de poder crear una escuela que forme parte de un centro	School	SC6F	POST	https://nimio2024.atlassian.net/browse/NIP-14	/schools/{centerId}	School school, Float centerId	School
SC7	El sistema debe de poder obtener todos los datos de una escuela en concreto	School	SC7F	GET	https://nimio2024.atlassian.net/browse/NIP-15	/schools/{schoolId}	Float schoolId	School
SC8	El sistema debe de poder obtener todos los cursos que se cursan en la escuela	School	SC8F	GET	https://nimio2024.atlassian.net/browse/NIP-16	/schools/{schoolId}/courses	Float schoolId	List <Course>
SC9	El sistema debe permitir modificar los datos de la escuela en concreto	School	SC9F	PUT	https://nimio2024.atlassian.net/browse/NIP-17	/schools/{schoolId}	Float schoolId	School
SC10	El sistema debe de poder eliminar una escuela	School	SC10F	DELETE	https://nimio2024.atlassian.net/browse/NIP-18	/schools/{schoolId}	Float schoolId	Float

Ilustración 10: Endpoints de la clase School

SC6F POST SCHOOL

Este endpoint permite al usuario crear una nueva escuela que pertenezca a un centro, es decir no pueden existir escuelas que no pertenezcan a ningún centro. Para ello, se debe de agregar en el path del endpoint el id del centro al que pertenece y en el body un JSON con los atributos necesarios para crear una escuela.

SC7F GET SCHOOL BY ID

Este endpoint permite al usuario obtener los datos de una escuela en concreto, para ello es necesario agregar en el path el id de la escuela sobre la que queremos consultar los datos, nos mostrara la información de la escuela junto con sus cursos y estudiantes.

SC8F GET SCHOOL COURSES BY ID

Este endpoint permite al usuario consultar los cursos que existen en una escuela en concreto, para ello solo es necesario agregar en el path el id de la escuela sobre la que queremos consultar los cursos, nso mostrara los cursos junto son su lista de alumnos.

SC9F UPDATE SCHOOL BY ID

Este endpoint permite al usuario modificar los datos de una escuela en concreto, para ello es necesario agregar la id de la escuela que desea modificar junto con el body en JSON con los nuevos datos para la escuela.

SC10F

Este endpoint permite al usuario eliminar una escuela, agregando sobre el path el id de la escuela que desea eliminar

COURSE

ID Requerimiento	Descripción	Clase	ID Funcionalidad	Tipo	Ticket JIRA	Endpoints	Parametros de entrada	Return
CO11	El sistema debe poder crear un curso nuevo	Course	CO11F	POST	https://nimio2024.atlassian.net/browse/NFP-18	/courses/{courseId}	Course, course	Course
CO12	El sistema debe poder modificar datos de un curso en concreto	Course	CO12F	PUT	https://nimio2024.atlassian.net/browse/NFP-20	/courses/{courseId}	Float, courseId	Course
CO13	El sistema debe permitir añadir alumnos a los cursos	Course	CO13F	PUT	https://nimio2024.atlassian.net/browse/NFP-21	/courses/{courseId}/addStudent{studentId}	Float, courseId, Float, studentId	Course
CO14	El sistema debe de permitir añadir una lista de alumnos a un curso	Course	CO14F	PUT	https://nimio2024.atlassian.net/browse/NFP-22	/courses/{courseId}/addStudentList	List <Student>, studentIdAdd	Course
CO15	El sistema debe de poder obtener todos los alumnos de un curso en concreto y los datos del curso	Course	CO15F	GET	https://nimio2024.atlassian.net/browse/NFP-23	/courses/{courseId}	Float, courseId	Course
CO16	El sistema debe de poder obtener los alumnos de un curso en concreto	Course	CO16F	GET	https://nimio2024.atlassian.net/browse/NFP-24	/students/{courseId}	Float, courseId	List <Student>
CO17	El sistema debe de permitir eliminar un curso sin borrar los alumnos	Course	CO17F	DELETE	https://nimio2024.atlassian.net/browse/NFP-25	/courses/{courseId}	Float, courseId	HTTP 204

Ilustración 11: Endpoints de la clase Course

CO11F POST COURSE

Este endpoint permite al usuario crear un nuevo curso, agregando en el path el id de la escuela a la que pertenece el curso y en el body en forma de JSON el “courseName” es decir el nombre que tiene el curso.

CO12F UPDATE COURSE NAME BY ID

Este endpoint permite al usuario modificar los datos de un curso en concreto, para ello es necesario agregar en el path la id del curso que se quiere modificar y en el body en forma de JSON el nuevo “courseName”.

CO13F ADD STUDENT BY ID

Este endpoint permite al usuario añadir a un curso un estudiante ya creado, será necesario el Id del estudiante que se quiere agregar y el id del curso al que queremos añadir al estudiante, ambos parámetros se agregaran en el path del endpoint.

CO14F ADD STUDENT LIST TO A COURSE

Este endpoint permite al usuario agregar uno o más estudiantes a un curso, para ello es necesario agregar al path el id del curso sobre el que se va a realizar la agregación y también se debe de agregar en el body en forma de JSON una lista de estudiantes con sus respectivos atributos.

CO15F GET COURSE BY ID

Este endpoint permite al usuario consultar los datos sobre un curso en concreto, para ello es necesario agregar en el path el id del curso que se desea consultar.

CO16F GET STUDENTS FROM A COURSE BY ID

Este endpoint permite al usuario obtener los estudiantes de un curso en concreto, para ello es necesario agregar en el path el id del curso sobre el que se desea consultar los estudiantes que forman parte.

CO17F DELETE A COURSE

Este endpoint permite al usuario eliminar un curso sin eliminar los alumnos, para ellos solo es necesario agregar en el path el id del curso que se desea eliminar.

TEACHER

ID Requerimiento	Descripción	Clase	ID Funcionalidad	Tipo	Ticket JIRA	Endpoints	Parámetros de entrada	Retorno
TE18	El sistema debe de permitir crear un profesor junto con sus datos correspondientes	Teacher	TE18F	POST	https://nimio2024.atlassian.net/browse/NFP-28	/teachers	Teacher teacher	Teacher
TE19	El sistema debe de permitir modificar los datos de un profesor	Teacher	TE19F	PUT	https://nimio2024.atlassian.net/browse/NFP-27	/teachers/{teacherId}	Float teacherId, Teacher teacher	Teacher
TE20	El sistema debe de permitir suscribir a un profesor y asignarlo	Teacher	TE20F	PUT	https://nimio2024.atlassian.net/browse/NFP-28	/teachers/{teacherId}/replace/{teacherIdNew}	Float teacherIdOld, Float TeacherIdNew	Teacher
TE21	El sistema debe de permitir obtener la lista de cursos a cargo de un profesor	Teacher	TE21F	GET	https://nimio2024.atlassian.net/browse/NFP-26	/teachers/courses/{teacherId}	Float teacherId	List <Course>
TE22	El sistema debe de permitir eliminar un profesor	Teacher	TE22F	DELETE	https://nimio2024.atlassian.net/browse/NFP-30	/teachers/{teacherId}	Float teacherId	NOTA

Ilustración 12: Endpoints de la clase Teacher

TE18F

Este endpoint permite al usuario crear un profesor para ello es necesario introducir los datos correspondientes.

TE19F

Este endpoint permite modificar los datos de un profesor identificándolo por su ID, para ello es necesario agregar en el path el ID del profesor que deseamos editar junto con el body con el contenido que sería el profesor editado.

TE20F

Este endpoint permite identificar dos profesores por id y sustituir los cursos del profesor nuevo por los del antiguo, sin necesidad de eliminar el profesor al que se le desasignan los

cursos que tenía; para ello es necesario incluir en el path el ID del profesor origen y el ID del profesor destino.

TE21F

Este endpoint permite obtener la lista de cursos que están a cargo de un profesor a través del ID del profesor. Para ello es necesario agregar en el path únicamente el ID del profesor a consultar.

TE22F

Este endpoint permite eliminar un profesor identificándolo con su ID, para ello únicamente es necesario agregar en el path la ID del profesor que se desea eliminar.

SCHEDULE

ID Requerimiento	Descripción	Clase	ID Funcionalidad	Tipo	Ticket JIRA	Endpoints	Parámetros de entrada	Retorno
SC23	El sistema debe permitir consultar donde se encuentra un curso en que clase esta	Schedule	SC23F	GET	https://nimio2024.atlassian.net/browse/NFP-31	/schedule/course/{courseId}	Float courseId	Schedule

Ilustración 13: Endpoints de la clase Schedule

SC23

Este endpoint permite consultar donde se encuentra un curso es decir en que clase esta para ello únicamente es necesario agregar al path la ID del curso a consultar.

CLASS

ID Requerimiento	Descripción	Clase	ID Funcionalidad	Tipo	Ticket JIRA	Endpoints	Parámetros de entrada	Retorno
CL24	El sistema debe poder crear clases o espacios	Class	CL24F	POST	https://nimio2024.atlassian.net/browse/NFP-32	/class	Class class	Class
CL25	El sistema debe permitir consultar toda la información de una clase	Class	CL25F	GET	https://nimio2024.atlassian.net/browse/NFP-33	/class/{classId}	Float classId	Class
CL26	El sistema debe permitir eliminar una clase	Class	CL26F	DELETE	https://nimio2024.atlassian.net/browse/NFP-34	/class/{classId}	Float classId	NOTAF

Ilustración 14: Endpoints de la clase Class

CL24F

Este endpoint permite crear classes o espacios con sus respectivos atributos, para ello únicamente es necesario introducir los respectivos datos en el body.

CL25F

Este endpoint permite al usuario consultar toda la información de un espacio o clase identificándolo por ID, para ello únicamente es necesario agregar al path el ID de la clase que deseamos consultar.

CL26F

Este endpoint permite al usuario eliminar una clase o espacio identificándolo por el ID de la clase a eliminar, para ello únicamente es necesario agregar el ID de la clase que deseamos eliminar al path.

COLLECTOR

ID Requerimiento	Descripción	Clase	ID Funcionalidad	Tipo	Ticket JIRA	Endpoints	Parámetros de entrada	Return
CO27	El sistema debe permitir consultar los recolectores de cada infante	Colector	CO27F	GET	https://nimio2024.atlassian.net/browse/NIP-35	/colectors/auth/{studentid}	Float studentid	List <Person>
CO28	El sistema debe permitir consultar el infante de un recolector X	Colector	CO28F	GET	https://nimio2024.atlassian.net/browse/NIP-36	/persons/auth/{personid}	Float personid	List <Student>
CO29	El sistema debe permitir consultar si un infante X pertenece a un recolector Y	Colector	CO29F	GET	https://nimio2024.atlassian.net/browse/NIP-37	/colectors/auth/{personid}/{studentid}	Float personid, Float Studentid	Boolean

Ilustración 15: Endpoints de la clase Colector

CO27F

Este endpoint permite al usuario consultar la lista de recolectores para cada infante, para ello únicamente es necesario agregar al path la ID del estudiante sobre el que se quiere realizar la consulta.

CO28F

Este endpoint permite al usuario consultar los infantes que están a cargo de un recolector, para ello únicamente es necesario agregar al path el id del recolector sobre el que se quiere realizar la consulta.

CO29F

Este endpoint permite verificar si un estudiante pertenece a un recolector, para ello es necesario agregar al path el id del estudiante y el id del recolector, i devolverá un booleano en función de si el recolector es el autorizado para ese estudiante o no.

PERSON

ID Requerimiento	Descripción	Clase	ID Funcionalidad	Tipo	Ticket IRA	Endpoints	Parámetros de entrada	Retorno
PE30	El sistema debe permitir a la escuela crear una persona parent	Person AUTH	PE30F	POST	https://nimio2024.atlassian.net/browse/NIP-38	/persons	Person person	Person
PE31	El sistema debe permitir a el parent crear una persona autorizada	Person AUTH	PE31F	POST	https://nimio2024.atlassian.net/browse/NIP-38	/persons/parent	Person person	Person
PE32	El sistema debe permitir a una persona parent modificar los permisos de una persona autorizada (ELIMINAR)	Person AUTH		PUT				
PE33	El sistema debe de permitir a la escuela modificar la información del parent	Person AUTH	PE33F	PUT	https://nimio2024.atlassian.net/browse/NIP-40	/persons/parent	Person person/parent	Person
PE34	El sistema debe permitir consultar la lista de alumnos autorizados para un parent	Person AUTH	PE34F	GET	https://nimio2024.atlassian.net/browse/NIP-41	/persons/parent	Float personid	List <Student>
PE35	El sistema debe permitir consultar quien son los autorizados de un alumno X	Person AUTH	PE35F	GET	https://nimio2024.atlassian.net/browse/NIP-38	/persons/parent	Float studentid	List <Person>
PE36	El sistema debe permitir a un parent consultar la lista de permisos que tiene un autorizado sobre su infante	Person AUTH	PE36F	GET	https://nimio2024.atlassian.net/browse/NIP-42	/collectors/auth/studentid	Float studentid	Person
PE37	El sistema debe permitir dar de baja un autorizado	Person AUTH	PE37F	DELETE	https://nimio2024.atlassian.net/browse/NIP-43	/persons/personid	Float authid	None

Ilustración 16: Endpoints de la clase Person

PE30F

Este endpoint permite crear una persona es decir un individuo al que se le podrán poner a cargo estudiantes, para ello únicamente hay que introducir en el body los atributos necesarios.

PE31F

Este endpoint permite crear una persona parent es decir la persona con más permisos sobre un alumno, para ello únicamente hay que introducir en el body los atributos necesarios para crear una persona.

PE33F

Este endpoint permite modificar la información de una persona en concreto, para ello únicamente es necesario agregar al path el id de la persona que queremos modificar, y agregar en el body la información de la nueva persona actualizada.

PE34F

Este endpoint permite consultar que alumnos están a cargo de una persona, para ello se debe de agregar en el path el id de la persona sobre la que se quiere consultar que alumnos tiene a su cargo.

PE35F & PE36F

Este endpoint permite consultar que personas están autorizadas para un alumno y que permisos tienen, para ello se debe de agregar en el path el id del estudiante sobre el que se quieren consultar sus autorizados.

PE37F

Este endpoint permite dar de baja un autorizado, para ello únicamente es necesario agregar en el path e id de la persona que queremos dar de baja.

STUDENT

ID Requerimiento	Descripción	Clase	ID Funcionalidad	Tipo	Ticket JIRA	Endpoints	Parametros de entrada	Return
ST38	El sistema debe de permitir crear un alumno con sus respectivos atributos	Student	ST38F	POST	https://nimio2024.atlassian.net/browse/NP-44	/students/{courseId}	Student student, Float courseId	Student
ST39	El sistema debe de permitir modificar los atributos de un alumno en concreto	Student	ST39F	PUT	https://nimio2024.atlassian.net/browse/NP-45	/students/{studentId}	Float studentId	Student
ST40	El sistema debe de permitir añadir responsables a un alumno	Student	ST40F	PUT	https://nimio2024.atlassian.net/browse/NP-46	/students/{studentId}/auth/{personId}	Float studentId, Float personId	Student
ST41	El sistema debe de permitir eliminar un responsable	Student	ST41F	PUT	https://nimio2024.atlassian.net/browse/NP-47	/students/deleteAuth/{studentId}/{personId}	Float studentId, Float personId	Student
ST42	El sistema debe de permitir consultar todos los alumnos de un curso	Student	ST42F	GET	https://nimio2024.atlassian.net/browse/NP-48	/students/course/{courseId}	Float courseId	List <Student>
ST43	El sistema debe de permitir consultar todos los datos de un alumno	Student	ST43F	GET	https://nimio2024.atlassian.net/browse/NP-49	/students/{studentId}	Float studentId	Student
ST44	El sistema debe de permitir dada una ID Responsable comprobar si es realmente el responsable	Student	ST44F	GET	https://nimio2024.atlassian.net/browse/NP-37	/students/auth/{studentId}/{personId}	Float personId, Float StudentId	Boolean
ST45	El sistema debe permitir eliminar un infante	Student	ST45F	DELETE	https://nimio2024.atlassian.net/browse/NP-54	/students/{studentId}	Float studentId	Boolean
ST46	El sistema debe permitir crear un student en course asignado	Student	ST46F	POST	https://nimio2024.atlassian.net/browse/NP-55	/students	Student student	Student

Ilustración 17: Endpoints de la clase Student

ST38F POST STUDENT

Este endpoint permite al usuario crear un nuevo estudiante con sus respectivos datos, para ello es necesario agregar al path el id del curso que formara parte y en el body en forma de JSON los datos necesarios para crear un estudiante.

ST39F

Este endpoint permite al usuario modificar los atributos de un alumno en concreto identificándolo por ID, para ello únicamente es necesario volver a introducir todos los atributos de este actualizados.

ST40F

Este endpoint permite añadir personas responsables a un alumno, para ello es necesario agregar en el path la ID del alumno al que queremos añadir la persona, y la ID de la persona que va a ser añadida.

ST41F

Este endpoint permite eliminar una persona responsable de un alumno sin necesidad de eliminar a la persona por completo, es decir desasignarla. Para ello es necesario agregar en el path la ID del alumno y la ID de la persona que se quiere desasignar.

ST42F

Este endpoint permite al usuario consultar todos los alumnos que están dentro de un curso, para ello únicamente es necesario agregar al path el ID del curso que queremos consultar sus alumnos.

ST43F

Este endpoint permite al usuario consultar todos los datos de un alumno en concreto identificándolo por ID, para ello únicamente es necesario agregar al path el ID del alumno que se desea consultar.

ST44F

Este endpoint permite verificar que una persona X puede recoger a un alumno X, para ello es necesario introducir en el path el ID del alumno que se va a recoger, i el ID de la persona que lo va a recoger.

ST45F

Este endpoint permite eliminar un alumno identificándolo por ID, para ello agregaremos en el path el ID del usuario que queremos eliminar.

ST46F

Este endpoint permite al usuario crear un nuevo estudiante con sus respectivos datos, pero sin estar asignado a ningún curso en concreto, para ello es necesario agregar en el body únicamente el estudiante con sus datos correspondientes.

EXTRA ENDPOINTS

Durante la ejecución del proyecto bajo la metodología ágil, se han identificado necesidades adicionales que no estaban contempladas inicialmente. Estos nuevos requerimientos han surgido a medida que avanzaba el proyecto, y aquellos que han sido factibles dentro del calendario se han desarrollado. A continuación, se enumeran los endpoints añadidos fuera de la planificación inicial:

En el Excel de URS, se han identificado como EXTRA de extra functionalities al igual que en sus correspondientes tickets de JIRA.

A	B	C	D	E	F	G	H	I
EX47	Des de la escuela se debe de poder saber que nombre tiene el centro al que pertenece	ExtraSchool	EX47F	GET	https://nimio2024.atlassian.net/browse/NIP-56	/school/centerName/{schoolId}	Float schoolId	Center
EX48	El sistema debe de permitir modificar los datos de la escuela por separado	ExtraSchool	EX48F	PUT	https://nimio2024.atlassian.net/browse/NIP-57	/school/{schoolId}/schoolName	Float schoolId, String newName	School
EX49	El sistema debe permitir añadir alumnos con el DNI a un curso	ExtraCourse	EX49F	PUT	https://nimio2024.atlassian.net/browse/NIP-58	/course/{courseId}/addStudentByDNI/{studentDNI}	Float courseId, String studentDNI	Course
EX50	El sistema debe de permitir mover todos los alumnos de un curso a otro	ExtraCourse	EX50F	PUT	https://nimio2024.atlassian.net/browse/NIP-59	/course/{courseId}/migrateToCourse/{courseId}	Float courseId, Float dCourseId	Course
EX51	El sistema debe de permitir obtener todos los profesores de una escuela	ExtraTeacher	EX51F	GET	https://nimio2024.atlassian.net/browse/NIP-60	/school/teachersFromSchool/{schoolId}	Float schoolId	School
EX52	El sistema debe permitir establecer una clase a unas horas para un curso	ExtraSchedule	EX52F	PUT	https://nimio2024.atlassian.net/browse/NIP-68	/schedule/{courseId}/classId	Float courseId, Float classId	Schedule
EX53	El sistema debe de permitir obtener todos los horarios para una clase	ExtraSchedule	EX53F	GET	https://nimio2024.atlassian.net/browse/NIP-67	/schedule/{courseId}/classId	Float classId	List <Schedule>
EX54	El sistema debe de permitir obtener todos los horarios para un curso	ExtraSchedule	EX54F	GET	https://nimio2024.atlassian.net/browse/NIP-68	/schedule/course/{courseId}	Float courseId	List <Schedule>
EX55	El sistema debe de permitir comprobar si es el auth con el DNI del auth	ExtraStudent	EX55F	GET	https://nimio2024.atlassian.net/browse/NIP-70	/student/authByDNI/{personDNI}/{studentId}	String personDNI, Float studentId	List <Schedule>
EX56	El sistema debe permitir asignar un curso a un teacher	ExtraTeacher	EX56F	PUT	https://nimio2024.atlassian.net/browse/NIP-79	/teacher/{teacherId}/addCourse/{courseId}	Float teacherId, Float courseId	Course
EX57	El sistema debe permitir eliminar un horario creado para una clase y curso	ExtraSchedule	EX57F	DELETE	https://nimio2024.atlassian.net/browse/NIP-78	/schedule/{scheduleId}	Float scheduleId	REMOVE

Ilustración 18: Extra endpoints

EXTRASCHOOL

EX47F

Este endpoint permite al usuario saber a qué centro pertenece una escuela a través del ID de la escuela, para ello únicamente es necesario agregar en el path el ID de la escuela sobre la que queremos consultar el nombre de su correspondiente centro.

EX48F

Este endpoint permite modificar únicamente el nombre de la escuela sin necesidad de modificar la escuela entera, para ello únicamente es necesario agregar en el path el id de la escuela que deseamos modificar, i en el body sin ningún tipo de formato el nombre que se le quiere dar a la escuela.

EX51F

Este endpoint permite obtener todos los profesores de una escuela en concreto, para ellos únicamente hay que añadir al path el ID de la escuela sobre la que queremos consultar sus profesores.

EXTRACOURSE

EX49F

Este endpoint permite añadir a un curso identificado por ID alumnos que ya estén creados, pero en vez de identificarlos por ID se pueden añadir introduciendo su DNI. Para ello únicamente es necesario añadir en el path el ID del curso al que se quiere añadir, y el DNI de alumno que se quiere añadir.

EX50F

Este endpoint permite migrar alumnos de un curso a otro es decir no elimina el curso, sino que cambia los alumnos de curso, para ello únicamente es necesario agregar en el path el ID del curso antiguo y el ID del curso nuevo.

EXTRA SCHEDULE

EX52F

Este endpoint permite al usuario asignar un horario para una clase, controlando que no se solape con otras clases, para ello es necesario agregar en el path el id de la clase y el id del curso que la va a ocupar.

EX53F

Este endpoint permite al usuario obtener todos los horarios que se van a llevar a cabo en una clase, únicamente agregando en el path el ID de la clase que se quiere consultar

EX54F

Este endpoint permite al usuario obtener todos los horarios para un curso es decir todos los horarios que hace un curso, para ello únicamente es necesario agregar en el path el ID del curso que se desea consultar.

EX57F

Este endpoint permite al usuario eliminar un horario ya creado para una clase y course, para ello se debe agregar al path el id del horario que se desea eliminar.

EXTRA STUDENT

EX55F

Este endpoint permite comprobar si una persona es la autorizada para recoger a un alumno, pero identificando a la persona por su DNI. Para ello es necesario agregar en el path el dni del recogedor y el id del alumno a recoger.

EXTRA TEACHER

EX56F

Este endpoint permite asignar un curso a un profesor, para ello es necesario agregar al path el id del profesor previamente creado y el id del curso previamente creado.

DEFINICION DE CLASSES

A continuación, se determinaran cada una de las clases a que hacen referencia y sus características más importante respecto a las relaciones y atributos que las definen.

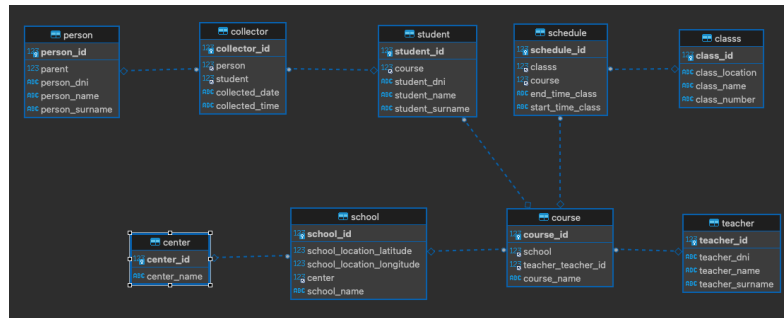


Ilustración 19: Esquema de base de datos JPA

CENTER

La tabla **center** hace referencia al centro al que pertenece una escuela, es decir como el grupo educativo al que pertenece una escuela, por ejemplo considerariamos “**CENTER**” a la UPF y “**SCHOOL**” a el Tecnocampus Mataró como tal.

Para la tabla **center** se ha definido únicamente el *center_name* que da nombre al grupo de escuelas. Un centro puede tener una o muchas escuelas.

SCHOOL

La tabla **SCHOOL** hace referencia directamente a lo que conocemos como escuela es decir el sitio físico, una escuela siempre pertenece a un **CENTER**, y una **SCHOOL** tiene uno o muchos **COURSE**.

Para la tabla **SCHOOL** se han definido campos como las coordenadas de latitud i longitud en el mapa para poder determinar su ubicación (*school_location_latitude*, *school_location_longitude*) también contiene el centro al que pertenece (*center*) y el nombre de la escuela (*school_name*).

COURSE

La tabla **COURSE** hace referencia a un curso escolar, no se quiere referir a una asignatura sino que hace referencia a un curso en completo, es decir, un **COURSE** por ejemplo podría

ser P3 en una **SCHOOL** de infantil. Cada **COURSE** esta asignado a un **TEACHER**, un **COURSE** tiene uno o muchos **STUDENT**, y un **COURSE** tiene uno o muchos **SCHEDULE**.

Para la tabla **COURSE** se han definido únicamente el campo del *course_name*, junto con las referencias a *school* y a *teacher*.

TEACHER

La tabla **TEACHER** hace referencia a un profesor como tal. **TEACHER** puede tener a cargo uno o muchos **COURSE**.

Para la tabla **TEACHER** se han definido los campos necesarios para identificar a un **TEACHER** con su dni (*teacher_dni*), su primer nombre (*teacher_name*) y su segundo nombre (*teacher_surname*).

SCHEDULE

La tabla **SCHEDULE** hace la función de calendario y es la encargada de almacenar a qué hora empieza una clase y cuando se acaba; relacionando a la vez donde se hace la clase es decir la tabla **CLASS** y quien hay en esa clase es decir que **COURSE** está dentro. Un **SCHEDULE** pertenece únicamente a una **CLASS** y a un **COURSE**.

Para la tabla **SCHEDULE** como bien se ha mencionado solo se dispondrá de los campos que hacen referencia a la class y el course, junto con los campos que hacen referencia al inicio de la clase en formato 24h (*start_time_class*) y el final de la clase en formato 24h (*end_time_class*).

CLASS

La tabla **CLASS** hace referencia a una clase como espacio de trabajo para los estudiantes, es decir un espacio dentro de la escuela. Una **CLASS** puede ser ocupada en diferentes **SCHEDULE** por diferentes **COURSE**.

Para la tabla **CLASS** se han definido los campos para determinar la ubicación de la clase en la escuela es decir si está en la primera planta en la segunda o una ubicación específica

(*class_location*), junto con los campos de el nombre de la clase (*class_name*) y el numero de la clase (*class_number*).

STUDENT

La tabla **STUDENT** hace referencia a un estudiante como tal es decir a un infante. Un estudiante pertenece a un único **COURSE** y un estudiante puede tener uno o muchos **COLLECTOR** es decir personas que estén autorizadas a recogerlo.

Para la tabla **STUDENT** se han definido los campos necesarios para identificar al estudiante es decir el dni (*student_dni*), el nombre (*student_name*) y el segundo nombre (*student_surname*); y tambien esta la instancia de *course* que hace referencia al curso al que pertenece el estudiante.

COLLECTOR

La tabla **COLLECTOR** hace una función parecida a la de la tabla **SCHEDULE**, esta relaciona los **STUDENTS** con las **PERSON** es decir con las personas que recogen a los infantes en la escuela. Una instancia de **COLLECTOR** pertenece únicamente a un **STUDENT** y a una **PERSON**.

Para la tabla **COLLECTOR** se han definido los campos que hacen referencia a la persona recogedora (*person*), el que hace referencia al estudiante que es recogido (*student*), e información relevante que sobre la fecha y la hora (*collected_date*, *collected_time*) que está permitido que una **PERSON** recoja a un **STUDENT**.

PERSON

La tabla **PERSON** hace referencia a la persona que recoge al infante, una persona puede recoger a uno o muchos infantes es por ello que **PERSON** puede tener una o muchas instancias de **COLLECTOR**.

Para la tabla **PERSON** se han definido los campos que hacen referencia a la identificación de la persona, el dni (*person_dni*), el nombre y apellido (*person_name*, *person_surname*), y un booleano que hace referencia a si es el padre o madre o es un autorizado externo (*parent_flag*).

ACCESO A LA GESTIÓN DEL PROYECTO Y CÓDIGO FUENTE

FUNCIONAMIENTO DE JIRA Y GIT

Como se puede ver en el Excel, cada una de las funcionalidades dispone de un ticket de Jira asociado que permite realizar una trazabilidad al detalle de cada una de las funcionalidades. Jira y Git son herramientas utilizadas en el desarrollo de software para gestionar proyectos y controlar versiones de código.

En Jira, cada tarea o funcionalidad tiene un "ticket" asociado, identificado por un código como NPP-XXX. Estos tickets permiten un seguimiento detallado de cada parte del proyecto.

Cuando se realizan cambios en el código utilizando Git, se puede vincular cada cambio con el ticket correspondiente en Jira. Por ejemplo, si se arregla un error o se implementa una nueva función, el código relacionado se "commitea" en Git con el número de ticket en el mensaje del commit.

Esta conexión entre Git y Jira permite a los desarrolladores y usuarios seguir la evolución del proyecto. Pueden ver qué cambios se han realizado en el código y relacionarlos con las tareas específicas del proyecto en Jira. Esto facilita la colaboración y el seguimiento del progreso del proyecto.

ACLARACIONES SOBRE COMO RESOLVER O REVISAR UN TICKET

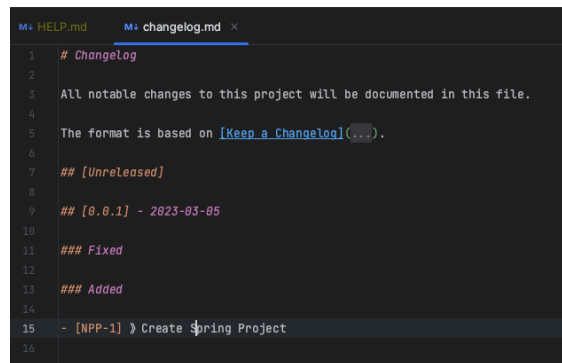
Para resolver un ticket y asociarlo a un commit sobre el código, se realizará el commit de la siguiente manera:

A parte de sincronizar el ticket de Jira con el repositorio de Github, también será necesario añadir los cambios al changelog del proyecto que permitirá llevar un histórico de todas las funcionalidades que se han ido añadiendo al proyecto, a más a más de en un futuro poder llevar un control de versiones del software diseñado.

Para crear este changelog se ha seguido un patrón universal de la mano de keepachangelog.com que nos permitirá tener un changelog con una documentación propia.

Una vez aclarado el funcionamiento del changelog ya se puede resolver un ticket correctamente.

- Añadimos al changelog.md el id del ticket en el siguiente formato “[NPP-XXX] » Breve descripción de la tarea”. Lo añadiremos en su sitio correspondiente dependiendo de si estamos añadiendo una nueva funcionalidad o estamos realizando arreglos sobre una ya existente, todo siguiendo la documentación del changelog que determina donde deben de quedar anotados cada uno de los cambios.



```
1 # Changelog
2
3 All notable changes to this project will be documented in this file.
4
5 The format is based on [Keep a Changelog](...),
6
7 ## [Unreleased]
8
9 ## [0.0.1] - 2023-03-05
10
11 ### Fixed
12
13 ### Added
14
15 - [NPP-1] » Create Spring Project
16
```

Ilustración 20 Changelog del proyecto

- Añadimos a la descripción del commit el id del ticket en el siguiente formato “[NPP-XXX] » Breve descripción de la tarea”, es decir lo mismo que en el changelog.md

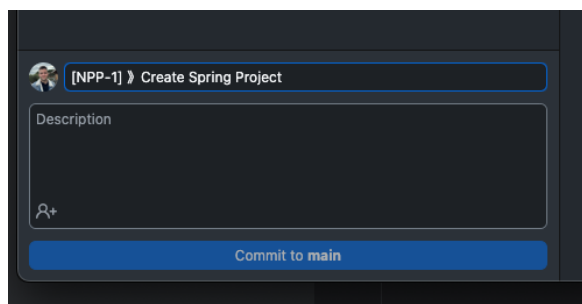


Ilustración 21 Descripción del commit

- Realizamos commit y push

Para poder revisar los cambios realizados sobre un ticket X, se hará de la siguiente manera:

- Seleccionamos el ticket sobre el que queremos indagar

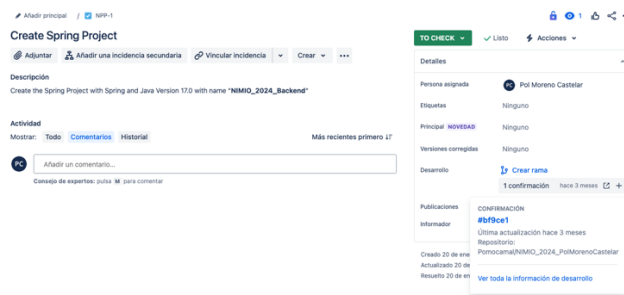


Ilustración 22: Ticket de Jira

- Seleccionamos el apartado de confirmaciones

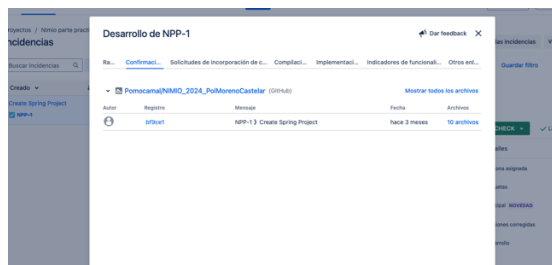


Ilustración 23: Confirmaciones Ticket de Jira

- Observamos las modificaciones del código y archivos modificados en ese ticket

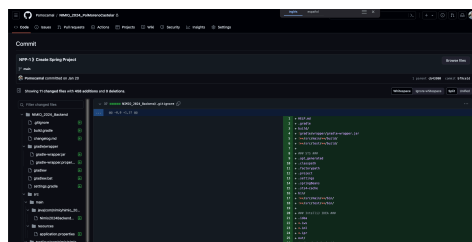


Ilustración 24: Ticket de Jira en Git

GUIA DE EJECUCIÓN DEL PROYECTO

En este apartado se detallan todos los procedimientos a seguir para poder ejecutar el proyecto y comprobar su funcionamiento en local en una nueva máquina.

REQUERIMIENTOS

Antes de nada, se debe descargar en local los siguientes programas:

- GitHub desktop (o cualquiera que permita clonar repositorios en local) [16]
- Navegador con conexión a internet (para acceder al tablero de JIRA)
- Postman desktop [17]
- Intel IJ (o cualquier IDE que permita ejecutar un proyecto de Spring) [19]
- MySqlWorkbench [18]

LOGINS Y ENLACES

En los siguientes apartados se detallarán paso por paso los procedimientos a seguir para *clonar el repositorio de github en local*, *acceder al tablero de JIRA* donde poder ver todas las tareas, *configurar el MySqlWorkbench*, *configurar Postman* para las pruebas, y *ejecutar el proyecto en local* correctamente.

CLONAR PROYECTO DE GITHUB

Para poder tener el proyecto en local será imprescindible iniciar sesión en GitHub Desktop con la siguiente cuenta:

Correo: nimiocorrector2024@gmail.com

Contraseña: nimiocorrector2024_TCM

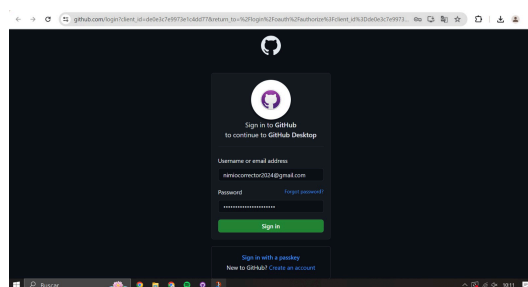


Ilustración 25: Página de registro en Git

Una vez iniciada la sesión correctamente se podrá ver la siguiente pantalla:

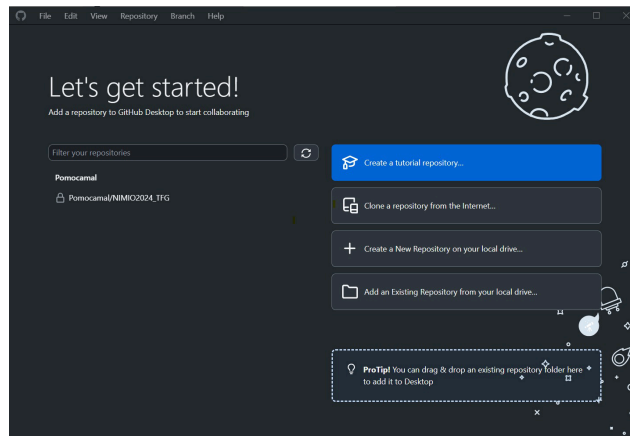


Ilustración 26: GitHub Desktop

Se debe seleccionar el único repositorio compartido que aparece que esta compartido por el usuario “Pomocamal” y lo que se debe de clonar en local en una ubicación donde después se accederá des de la IDE. Una vez clonado correctamente la ventana debería quedar de la siguiente forma:

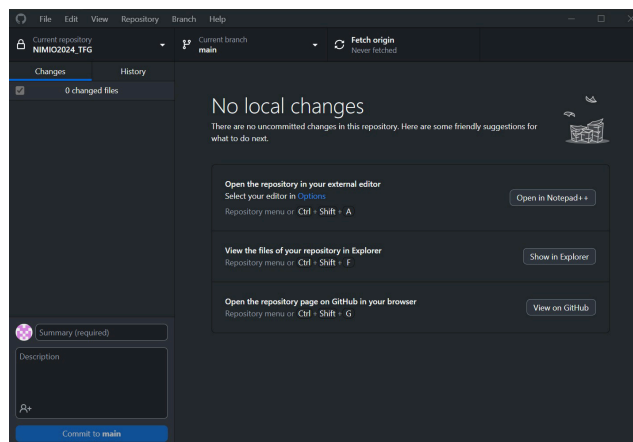


Ilustración 27: GitHub Desktop

ACCESO A JIRA

A continuación, se detallarán los pasos para acceder al tablero de JIRA, donde se podrán ver cada una de las tareas realizadas en el desarrollo, junto con las confirmaciones vinculadas con git para poder ver para cada tarea únicamente los cambios realizados.

Para ello se debe acceder al siguiente enlace y entrar seleccionando la opción de iniciar sesión con Google con la siguiente cuenta de Google:

Correo: nimiocorrector2024@gmail.com

Contraseña: nimiocorrector2024_TCM

Enlace JIRA: <https://nimio2024.atlassian.net/jira/software/projects/NPP/boards/3>

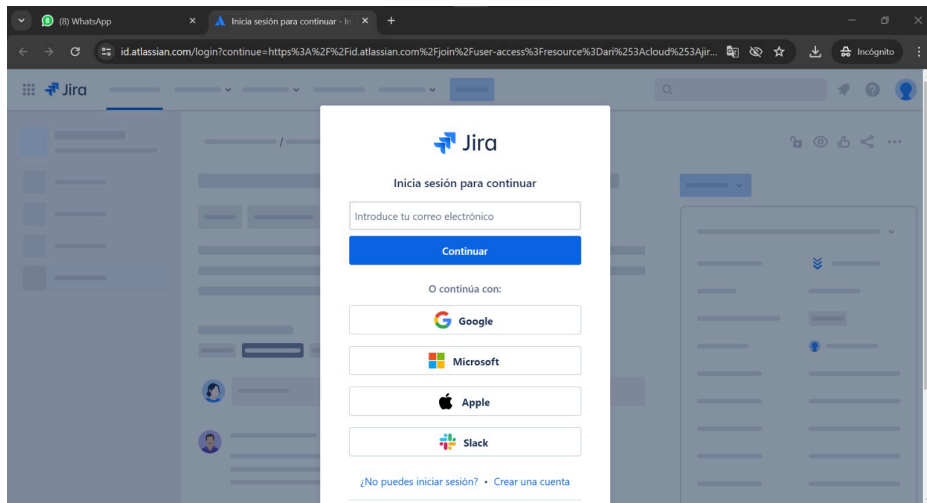


Ilustración 28: Página de LogIn de Jira

Ya iniciada la sesión con Google ya se tendrá acceso al tablero de JIRA, donde se podrán consultar todos los tickets y su contenido.

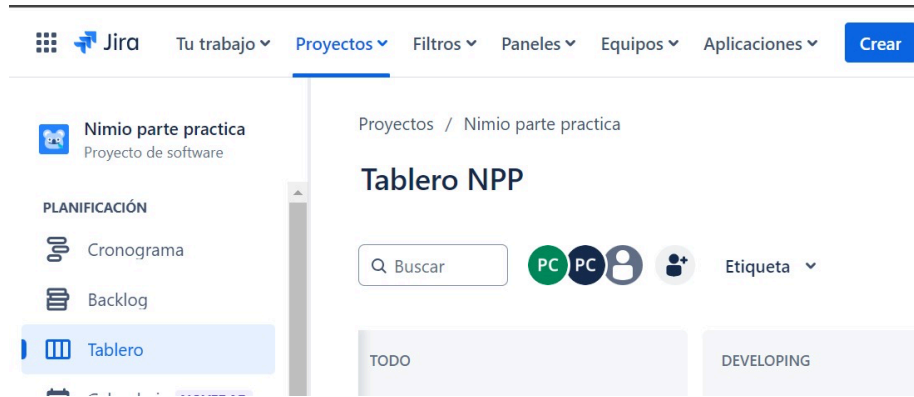


Ilustración 29: Tablero de Jira

En los apartados anteriores (Aclaraciones sobre cómo resolver o revisar un ticket), se incluye como consultar un ticket y ver las modificaciones que se han hecho en este.

ACCESO POSTMAN

Una vez descargada la aplicación de escritorio de PostMan se debe iniciar sesión con Google para poder disponer de todos los endpoints necesarios para realizar todas las pruebas de los endpoints descritos anteriormente. Para ello se iniciará sesión en la aplicación de escritorio con la siguiente cuenta con la opción de iniciar con Google:

Correo: nimiocorrector2024@gmail.com

Contraseña: nimiocorrector2024_TCM

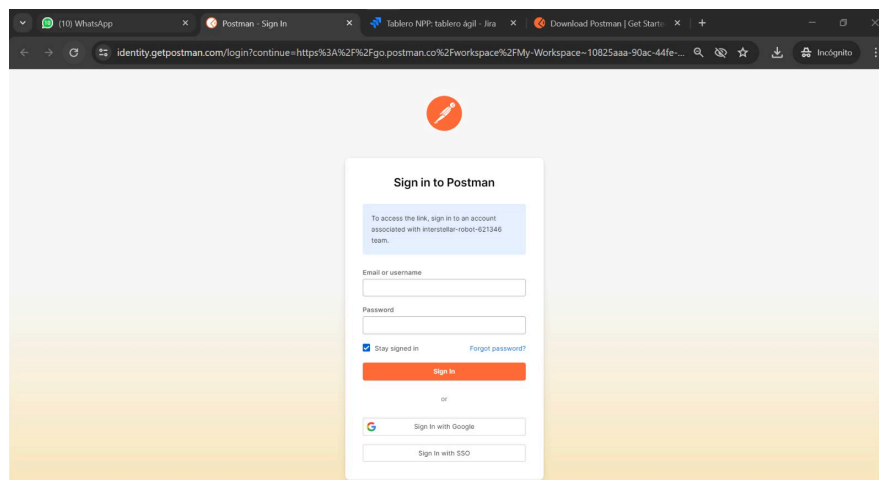


Ilustración 30: Inicio de Sesión en Postman

Una vez iniciada la sesión se podrá comprobar que existe una colección creada donde se encuentran todas las peticiones identificadas para poder realizar las pruebas de cada uno de los endpoints.

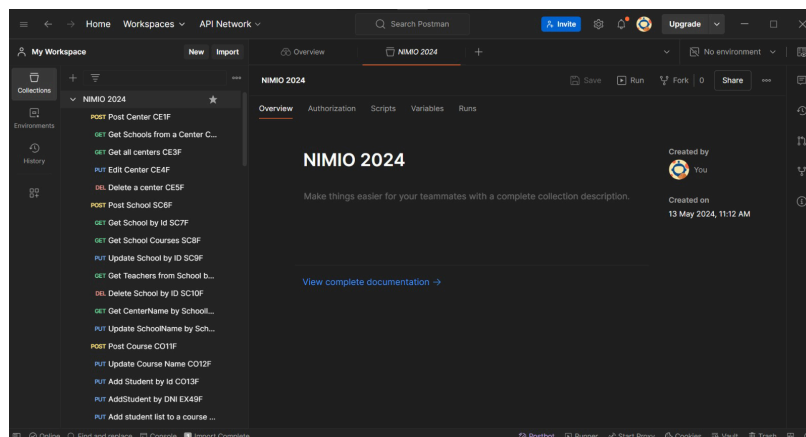


Ilustración 31: Colección de Postman

CONFIGURACIÓN DE MYSQLWORKBENCH

Una vez descargado e instalado MysqlWorkbench en su instalación más completa, se deberá crear una conexión con una contraseña y un “Schema” que tenga el siguiente nombre para que pueda conectarse correctamente con el proyecto de Spring.

Nombre de la Conexión: NimioTest

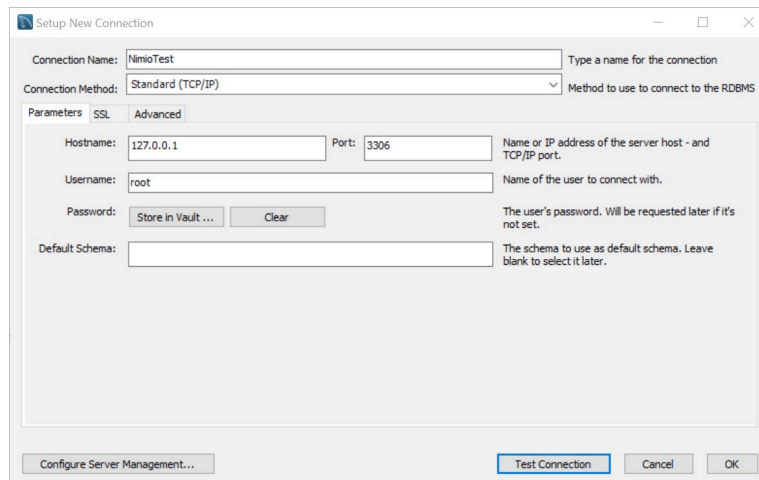


Ilustración 32: Configuración de MySQL

Nombre del Schema: nimioTest

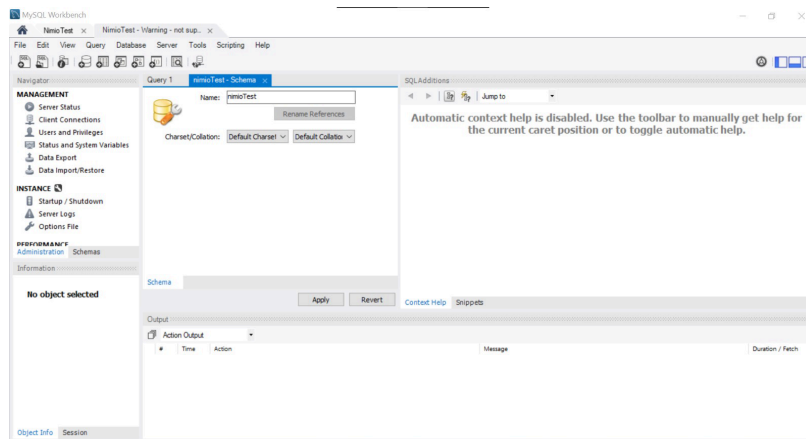
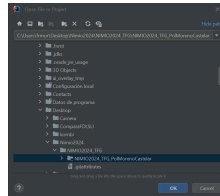


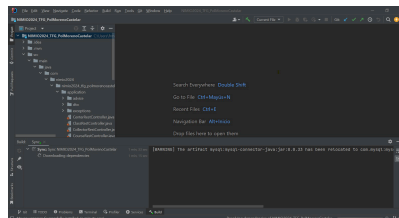
Ilustración 33: Configuración de MySQL

EJECUCIÓN EN INTEL IJ

Una vez completados los pasos anteriores ya está todo listo para la ejecución del proyecto en local. Para ello deberemos de abrir IntelliJ ya previamente descargado e instalado y abrir el repositorio previamente clonado con Git.

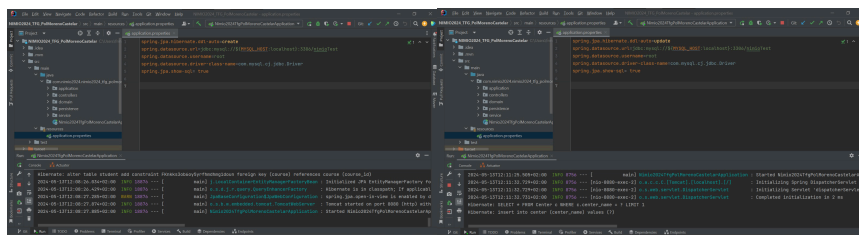


Una vez cargado esperaremos a que instale todas las dependencias automáticamente para poder iniciar correctamente el entorno.



Una vez se hayan instalado todas las dependencias se deberá de actualizar el fichero de configuración para que la primera ejecución cree todas las instancias de base de datos para ello, se deberá poner el atributo **“create”** en la primera línea del archivo de configuración, y en la segunda ejecución ya lo se podrá cambiar a **“update”** para que no tenga que crear todas las instancias para cada ejecución i poder guardar datos entre ejecuciones. Aunque si se prefiere se puede dejar en **“create-drop”**, pero no guardará los datos entre ejecuciones y la inicialización será más lenta ya que creará siempre de cero toda la base de datos a través de JPA.

También se debe tener en cuenta que dependiendo de la base de datos que hayamos creado en MySql puede que no necesitemos contraseña como en este caso para esta máquina en concreto.



FIN GUÍA DE EJECUCIÓN DEL PROYECTO

Una vez finalizados todos los pasos anteriores para la configuración del entorno, ya solo quedaría probar cada una de las funcionalidades a través de PostMan, un buen flujo de pruebas sería:

1. Seleccionar el ticket en JIRA y comprender su funcionalidad.

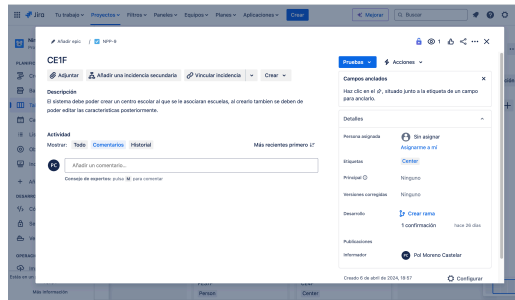


Ilustración 34: Ticket de Jira

2. Iniciar el proyecto tal y como se ha mostrado en los apartados anteriores

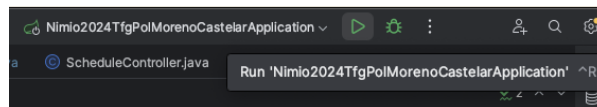


Ilustración 35: Ejecución del Proyecto

3. Iniciar PostMan y buscar en la colección la petición que comparta la misma ID que el ticket de JIRA.

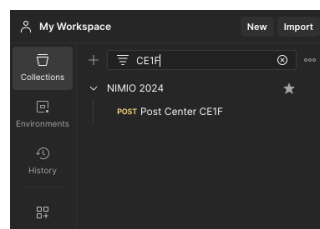


Ilustración 36: Postman

4. Ejecutar la petición

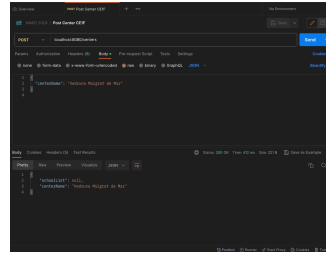


Ilustración 37: Ejecución de petición des de Postman

Durante la ejecución se pueden mostrar mensajes de error controlados que ayudaran al usuario a ejecutar correctamente en orden las peticiones para poder hacer uso de todas las funcionalidades.

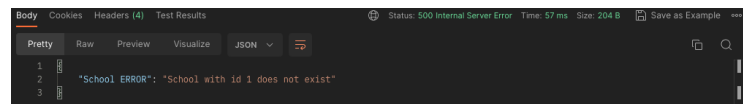


Ilustración 38: Ejemplo de error controlado des de Postman

CREACIÓN DEL MODELO DE NEGOCIO Y VENTA DE LA SOLUCIÓN

A continuación, se ha desarrollado un modelo de negocio para el planteamiento de como sería la salida al mercado de la aplicación. En este modelo de negocio la focalización ya no será directamente del ámbito de recursos informáticos, sino que irá dirigida a como se va a vender el producto y como se va a obtener una rentabilidad de este.

PROPUESTA DE VALOR:

Se ha analizado que es lo que ofrece esta solución respecto a las que ya existen y que es aquello que la va a hacer diferente a la competencia. Después de realizar un estudio de la competencia se ha llegado a destacar los siguientes puntos:

- **Seguridad:** Mayor control y seguridad en la recogida de alumnos, evitando confusiones y posibles riesgos.
- **Tranquilidad:** Los padres pueden estar seguros de que sus hijos solo serán recogidos por personas autorizadas.
- **Comunicación:** Facilita la comunicación entre el centro escolar y los padres.

Estas serían las 3 grandes ideas que transmitiremos en el momento de vender la solución, es decir debemos vender que esta aplicación tiene como fin optimizar la seguridad, la tranquilidad de las familias y facilitar la comunicación entre los centros escolares y las familias. Cabe destacar que no solo hay que vender a las familias, sino que también hay que convencer a los centros escolares y crearles una necesidad.

SEGMENTO DE MERCADO:

También se ha investigado cuáles serán los segmentos de mercado donde se deberá de atacar para poder llevar a cabo esta solución, y en este caso el mercado es claro ya que va dirigido a un público muy concreto, por lo que el segmento de mercado se basará en:

- Centros escolares de educación infantil, primaria y secundaria.
- Asociaciones de padres y madres de alumnos (AMPA).

ESTRATEGIA DE PRECIOS:

En el momento de desarrollar un plan de negocio cabe tener en cuenta cómo vamos a vender o como vamos a facturar aquello que vendemos. Para ello es imprescindible valorar todas las opciones posibles y que mejor que preguntar directamente al cliente potencial como le gustaría realizar el pago de este servicio. Por ello se ha realizado una pregunta dentro de la encuesta que respondieron los principales clientes donde se les preguntaba cuál sería su método de pago preferido, con qué frecuencia querrían realizar el pago y que monto estarían dispuestos a pagar.

Analizadas las respuestas de las encuestas se ha llegado a la conclusión que:

- El precio que las familias estarían dispuestas a pagar por el servicio sería de entre 10-20€ por alumno al año.
- El pago se realizaría junto con la matricula del alumno y en un solo pago, es decir justo a principio del curso escolar
- El método de pago sería por transferencia des de la escuela hasta el proveedor de la solución (nosotros).

Antes de hacer una estimación de la facturación que se podría generar, cabe destacar que se llegaría a un acuerdo con el centro escolar ya que este se deberá de llevar una comisión por la implantación del servicio. Se ha comentado la idea de la comisión a más de 4 escuelas de las cuales no se puede dar más información por leyes de protección de datos, y todas ellas han coincidido en que aceptarían en que se llevarían un 10% aproximadamente de la facturación, es decir aproximadamente 1,5€ por alumno que contrate el servicio.

CANALES DE DISTRIBUCIÓN:

- **Venta directa:**

Para vender la solución es claro que se contactaría directamente con centros escolares u organizaciones con representación dentro de los centros escolares, y se intentaría acordar una presentación de la solución junto con demostraciones para crear esa necesidad en los centros. Es sumamente desarrollar una buena presentación de la idea para dejar claro a los clientes finales que es lo que deben de esperar de esta solución.

- **Marketing online:**

Para la promoción de la aplicación se plantean opciones como la creación de redes sociales como Instagram y Tiktok para hacer-se ver al público, y a parte hacer colaboraciones con canales específicos focalizados en la educación que sean de referencia para el sector de manera que se dé una imagen confiable del servicio.

También aparte de publicitarse en redes sociales, también se atacará directamente a los centros escolares con contacto directo con ellos como se ha mencionado en la venta directa (email marketing).

Por último, también se creará una página web informativa para que las personas interesadas en la solución puedan informarse y puedan encontrarla de manera fácil e intuitiva.

ESTRUCTURA DE COSTES:

El presupuesto estimado para este proyecto parte de los siguientes puntos, sin tener en cuenta todo el tema de publicitarse ya que no entraría dentro de los límites de proyecto para este TFG:

- Implicación estimada de 500h por parte de un Ingeniero Informático Junior
- Equipo informático (Ordenador, Monitores, Software)
- Luz
- Internet

Siguiendo la media de precio la hora que cobra un Software Developer de Java Spring con la carrera de Ingeniería Informática en España, esta ronda los 25€ (brutos) partiendo de aquí estaríamos hablando de los siguientes costes por la implicación en el proyecto:

- Desarrollador de Software Junior (Spring)
 - o 25€/h (Cuota Seguridad Social Incluida)
 - o 500h
 - o Coste para el proyecto **12500€**

El sueldo neto es el resultado de descontar al sueldo bruto, un 4,70% de cotización a la seguridad social del trabajador, un 0,10% de formación y un 1,55% de cotización por

desempleo: un total de 6,35 %. Además, es necesario aplicar el porcentaje de IRPF, que, por estos niveles de sueldo son alrededor del 20%-25%. La seguridad social sólo se descuenta doce meses, y el IRPF en las catorce pagas o prorrateado en las 12.

Por parte del equipo informático se trabajará con ordenador Apple ya que es el entorno de desarrollo en el que el desarrollador se encuentra más cómodo, por lo que serán 2300€ por parte del ordenador y dos monitores de aproximadamente 200€ por monitor es decir 400€ en total. Pero como el tiempo de desarrollo del proyecto será de aproximadamente 9 meses, deberíamos amortizarlo de la siguiente manera:

- Ordenador:
 - o Tiempo de vida 5-6 años
 - o Precio por mes 32€
 - o Coste para el proyecto **288€**
- Monitores:
 - o Tiempo de vida 2-4 años
 - o Precio por mes 8.50€
 - o Coste para el proyecto **77€**

También debemos tener en cuenta el software que se utilizará, el desarrollo se hará con la IDE de IntelliJ que tiene un coste de:

- Programas:
 - o Intel IJ 16.90€/mes
 - o 9 meses
 - o Coste para el proyecto **152.10€**

Como el proyecto se desarrollará en local no será necesario contratar ningún servidor en el cloud por lo que no será un gasto hasta la aplicación desarrollada saliese al mercado real.

Por parte de luz e internet se debe tener en cuenta un coste de luz de unos 15€ al mes ya que se trata de un portátil y dos monitores lo que no consume mucha luz contando que están una media de unas 20h a la semana en funcionamiento. Junto con una tarifa de internet de aproximadamente 25€ al mes. Todo esto hará un total de:

- Luz:
 - o 15€ al mes

- 9 meses
- Coste para el proyecto **135€**
- Internet:
 - 25€ al mes
 - 9 meses
 - Coste para el proyecto **225€**

El total del presupuesto estaría en **13.377€** para el desarrollo del proyecto dentro de los límites especificados en el alcance del proyecto. Aunque claramente en el momento de la salida al mercado se agregarían todos los costes pertinentes al desarrollo de una solución a gran escala.

POTENCIAL DE BENEFICIOS:

- El mercado de las aplicaciones para centros escolares es amplio y creciente.
- La propuesta de valor de la aplicación es atractiva para los centros escolares y los padres.
- La estrategia de precios es flexible y se adapta a las necesidades de diferentes tipos de centros.
- La seguridad en el ámbito escolar esta poco desarrollada
- Mercado muy poco explotado

POSICIONAMIENTO COMPETITIVO:

- La aplicación se diferenciará de la competencia por su enfoque en la seguridad y la tranquilidad de los padres.
- Se ofrecerá un servicio de atención al cliente de alta calidad.
- Se realizarán actualizaciones constantes de la aplicación para mejorar sus funcionalidades.

ESTRATEGIA DE MARKETING:

Para la estrategia de marketing se crearán campañas dirigidas a un público muy específico que serán literalmente las familias y para atacar a este público se atacará directamente a AMPA, ya que estos son los que validarán si es o no viable pasar esta propuesta de solución a las familias.

En esta campaña se destacarán dos ideas principales que capten al completo la atención de los clientes, que serán Seguridad y Tranquilidad, ya que realmente es lo que quiere proporcionarte la solución y es lo que buscan las familias cuando dejan a sus hijos en la escuela.

Finalmente, paralelo a atacar a las familias directamente, se ofrecerán pruebas gratuitas del servicio o demostraciones en escuelas para vender la solución más fácilmente y que el entorno comprenda el funcionamiento al cien por cien.

PLAN DE ACCIÓN:

- Desarrollar un prototipo funcional de la aplicación.
- Realizar pruebas con usuarios reales para obtener feedback.
- Definir la estrategia de precios y los canales de distribución.
- Crear una página web y materiales de marketing.
- Contactar con centros escolares y AMPA para presentar la aplicación.

Cabe destacar, que dentro de los límites de este TFG solo se llegaría a desarrollar el prototipo de la aplicación en su backend, los demás pasos quedarían fuera del alcance de este trabajo.

CONCLUSIONES

Para realizar las conclusiones sobre este trabajo, se considera interesante dividir las conclusiones en tres partes: la parte teórica, la parte práctica y, finalmente, la relación entre ambas junto con una valoración personal sobre el trabajo finalizado en su totalidad. Es importante dividir la conclusión en partes ya que es necesario determinar las ideas clave a las que se ha llegado con el transcurso del proyecto y resulta interesante identificar aquellos puntos donde se han encontrado dificultades y cómo se han afrontado.

PARTE TEÓRICA

Después de realizar todas las tareas planificadas para la parte teórica en el calendario previsto, se pueden extraer las siguientes conclusiones:

En primer lugar, se determinaron las metodologías de trabajo a utilizar para desarrollar el proyecto tanto desde el punto de vista teórico como práctico, permitiendo hacer un seguimiento de todas las tareas realizadas y las pendientes a través de las tecnologías de JIRA, GitHub y OneDrive, que facilitan una gestión de cambios detallada y completa. También se definieron las tecnologías a emplear para el desarrollo, siendo estas Java SpringBoot y MySQL.

En el apartado de límites del proyecto, se aclararon los límites en el marco temporal de la entrega del trabajo, basándose en que en estas entregas solo se desarrollaría el modelo de negocio junto con el backend de la aplicación y la parte teórica.

Una vez establecida la metodología y los límites, se pasó al apartado de definición de la solución, donde se identificaron los principales comportamientos que podría tener la aplicación para el usuario final, permitiendo comprender la problemática de los distintos flujos de la recogida de un alumno. Se determinó que ambos flujos tenían casos excepcionales y requerían un control exhaustivo del diseño para evitar imprevistos. Finalmente, se definieron ambos flujos después de una revisión detallada y se utilizaron para determinar los casos de uso y los requerimientos en la parte teórica.

En el apartado de entrevistas, se consideró vital conocer la opinión de los potenciales clientes para validar si se estaba cumpliendo el objetivo de solucionar un problema real. Se confirmó que las hipótesis eran correctas y se obtuvieron datos interesantes, como el rango de precios

que el público estaría dispuesto a pagar, lo cual fue útil para desarrollar el modelo de negocio. Las entrevistas ayudaron a validar la idea principal del proyecto y a identificar puntos débiles pasados por alto.

En cuanto al tema legal, se realizó una investigación sobre las leyes que la aplicación debería cumplir para su lanzamiento al mercado y se identificaron los datos a recopilar. Aunque no se profundizó en gran detalle sobre el marco legal, se destacaron los aspectos legales más importantes a tener en cuenta.

Una vez aterrizada toda la idea de la aplicación y conociendo sus puntos fuertes, se procedió al desarrollo del modelo de negocio. Con la ayuda de jóvenes emprendedores y personas con empresas activas, se confeccionó un modelo de negocio directo, identificando todos los puntos clave a tener en cuenta. Los resultados de las entrevistas fueron de gran ayuda para agilizar el proceso de cómo atacar al cliente y ofrecer la solución de manera más cercana y con mayor posibilidad de éxito. No se profundizó en el lanzamiento de la aplicación al mercado real, ya que no se encontraba dentro de los límites del proyecto, y se optó por centrarse en cómo atraer al cliente y potenciar sus necesidades respecto a la aplicación.

En resumen, en la parte teórica se pudieron desarrollar sin problemas todos los puntos determinados en el calendario inicial del proyecto, extrayendo información valiosa que da valor a la investigación y al trabajo realizado.

PARTE PRACTICA

Por otro lado, después de haber realizado la mayoría de las tareas previstas para la parte práctica, se han llegado a las siguientes conclusiones:

En primer lugar, antes de empezar con el desarrollo del código en Java, se desarrolló la arquitectura del software. Se inició creando un borrador de la estructura de la base de datos, es decir, cuántas entidades y qué relaciones establecer entre ellas. Posteriormente, se creó una base de datos en MySQL, ya que inicialmente se planeaba desarrollar Spring con conexión a base de datos por JDBC. Sin embargo, se valoró la posibilidad de usar JPA, por lo que se optó por eliminar la base de datos inicial y recrearla desde Spring utilizando JPA,

simplificando así el desarrollo y permitiendo dedicar más tiempo a la creación de funcionalidades.

Una vez creada la base de datos, se definió un flujo de trabajo estructurado y con un control de cambios constante. Para ello, se utilizaron tecnologías de "Project Management" como JIRA y GitHub, que permitieron gestionar los cambios y controlar cada una de las funcionalidades a desarrollar y su ubicación en el código.

Definido el flujo de trabajo, se empezaron a definir las funcionalidades basadas en los requerimientos que la aplicación debía cumplir. Para facilitar la estructuración, se diseñó un Excel de URS (User Requirements), donde se especificaba cada requerimiento, la funcionalidad correspondiente y su enlace al ticket de JIRA correspondiente.

A partir de este punto, se desarrollaron cada una de las funcionalidades siguiendo el flujo especificado y se documentaron explicando brevemente su uso, acompañadas de una librería de Postman y una guía de ejecución para facilitar al corrector la verificación del correcto funcionamiento de cada una.

Durante el desarrollo, surgieron contratiempos y funcionalidades adicionales que obligaron a crear una sección de "Extra Functionalities", donde se incluyeron todas aquellas funcionalidades imprescindibles que se habían pasado por alto inicialmente.

Finalizadas todas las funcionalidades tanto en el aspecto práctico (código) como en el teórico (documentación), se creó una guía para ayudar al lector a comprender el uso de Git con JIRA y la trazabilidad de los cambios aplicados a lo largo del trabajo práctico. También se desarrolló una guía de ejecución desde cero, explicando cómo ejecutar toda la parte práctica y comprobar su correcto funcionamiento.

Finalmente, se destaca que se desarrollaron la mayoría de las funciones previstas para la entrega final, siguiendo los tiempos determinados en la planificación del proyecto. Sin embargo, algunas funcionalidades se descartaron debido a que alargaban mucho el calendario e impedían llegar a la fecha de entrega. Ejemplos de estas funcionalidades son la instalación de "Swagger" en el proyecto, el desarrollo de un log-in con OAuth para registrar estudiantes, alumnos, profesores o administradores, así como la descripción de roles dentro del backend para controlar el acceso a determinados endpoints.

VALORACIÓN FINAL

El proyecto se ha desarrollado de manera ordenada y efectiva, cumpliendo con los objetivos iniciales y adaptándose de forma adecuada a los desafíos surgidos durante el proceso. La planificación detallada y el uso de herramientas avanzadas de gestión de proyectos, como JIRA, GitHub y OneDrive, permitieron una ejecución eficiente y controlada

OBJETIVO DEL PROYECTO

Se debe tener en cuenta que, desde el principio, el proyecto tuvo un objetivo muy claro: desarrollar una aplicación que dé respuesta a un problema presente en el sector escolar, además de confeccionar un plan de negocio para la comercialización de este producto. El objetivo final es que la solución propuesta no solo sea técnicamente viable, sino también comercialmente atractiva y capaz de satisfacer una necesidad real en el mercado escolar.

ALCANCE DEL PROYECTO

También es importante recordar que el alcance del proyecto se limitó a desarrollar el backend de la aplicación, asegurando su funcionalidad, y en desarrollar el plan de negocio para su eventual comercialización. Se estableció claramente que el lanzamiento de la aplicación al mercado real y el desarrollo del frontend quedaban fuera del alcance del Trabajo de Fin de Grado (TFG) para el curso 2023-2024.

OBJETIVOS CUMPLIDOS DURANTE EL TFG

DAR UN REMEDIO A UN PROBLEMA EXISTENTE EN EL SECTOR ESCOLAR

Se logró identificar un problema concreto en el sector escolar relacionado con la recogida de alumnos y se desarrolló una solución viable y práctica para abordarlo. La aplicación está diseñada para ofrecer una manera segura y eficiente de gestionar este proceso, aportando un valor significativo tanto para los padres como para las instituciones escolares.

PLANTEAR LA COMERCIALIZACIÓN DE ESTA SOLUCIÓN

Se desarrolló un plan de negocio detallado que incluye marketing, análisis de mercado, y una propuesta de valor clara para los potenciales clientes. Este plan de negocio está diseñado para facilitar la comercialización de la aplicación una vez esté completamente desarrollada.

DEMOSTRAR LA VIABILIDAD DEL PRODUCTO FINAL Y CREAR NECESIDAD

Las entrevistas con los potenciales usuarios proporcionaron datos valiosos que confirmaron la viabilidad del producto. Se demostró que existe una necesidad real y una disposición del mercado a adoptar la solución propuesta. Esta validación es crucial para asegurar que el proyecto no solo es viable desde una perspectiva técnica, sino también comercial.

DESARROLLO EN JAVA SPRINGBOOT Y CONTROL DE BBDD CON MYSQL

El backend de la aplicación se desarrolló utilizando Java SpringBoot, con una base de datos gestionada mediante MySQL. Si bien es cierto que se decidió optar por otra modalidad de base de datos, pero siempre usando las dos tecnologías mencionadas, ya que se pasó de crear una base de datos con MySQL y conectarla con JDBC a Spring a hacerlo con la tecnología de JPA des de Spring.

LÍMITES DEL PROYECTO

MANTENERSE DENTRO DEL SECTOR ESCOLAR

El proyecto se mantuvo enfocado en el sector escolar, asegurando que todas las funcionalidades y el plan de negocio estuvieran alineados con las necesidades específicas de este sector.

DESARROLLO ÚNICAMENTE DEL BACKEND

El desarrollo se centró exclusivamente en el backend de la aplicación, cumpliendo con los límites establecidos. Esto permitió un enfoque más profundo y detallado en esta parte.

ALCANCE DEL PROYECTO

DESARROLLO COMPLETO DEL BACKEND EN JAVA SPRINGBOOT DE LAS PRINCIPALES FUNCIONALIDADES

Se completó el desarrollo del backend, incluyendo todas las funcionalidades principales necesarias para el correcto funcionamiento de la aplicación. Esto asegura que la base técnica del proyecto está sólidamente establecida.

DESARROLLO DE UN PLAN DE NEGOCIO

Se creó un plan de negocio detallado que incluye estrategias de comercialización y un análisis profundo del mercado objetivo. Este plan es esencial para la futura implementación y éxito comercial de la aplicación.

CONTACTO CON PROFESIONALES DEL SECTOR

Se mantuvieron contactos con profesionales del sector escolar para validar la aplicación y obtener insights valiosos sobre las necesidades y desafíos específicos del sector. Este feedback fue crucial para ajustar y mejorar la propuesta de valor del proyecto.

OBJETIVOS DEL PRODUCTO Y DEL CLIENTE

VENDER EL SOFTWARE A CENTROS ESCOLARES

El proyecto está diseñado para ser comercializado directamente a centros escolares, ofreciendo una solución específica y adaptada a sus necesidades. Aunque durante el calendario del desarrollo de todo el trabajo no ha sido posible la presentación de este a centros escolares.

PÚBLICO POTENCIAL Y TARGET

El proyecto está orientado principalmente a familias y tutores legales, que son los clientes potenciales de la aplicación. El sistema está diseñado para proporcionarles un control seguro y eficiente sobre la recogida de sus hijos, creando un valor significativo para este grupo demográfico.

CONCLUSIÓN GENERAL

En conjunto, el proyecto logró cumplir con gran parte de los objetivos planteados, demostrando una capacidad efectiva para gestionar y ejecutar proyectos complejos en el ámbito del desarrollo de software. Se estableció una base teórica robusta, se validaron las hipótesis de mercado, se desarrolló un plan de negocio detallado y se desarrolló una parte práctica ajustándose a los límites del proyecto.

VALORACIÓN PERSONAL

Concluidas las partes teórica y práctica, puedo dar por finalizado este Trabajo de Fin de Grado. A lo largo de su desarrollo, he aplicado y potenciado los conocimientos adquiridos durante mi carrera universitaria. He abarcado desde aspectos relacionados con la gestión, organización e investigación, hasta el desarrollo, diseño y realización de un proyecto entregable.

La finalización de este proyecto me ha permitido reflexionar sobre la cantidad de conocimientos adquiridos durante mi formación académica y sobre los muchos que aún me quedan por aprender en mi futura inserción en el mundo laboral. He comprendido que cada tecnología es un universo en sí mismo, con una vasta cantidad de información y funcionalidades que no se pueden aprender completamente en un entorno académico, sino que requieren de una dedicación autodidacta. Me he podido dar cuenta que la mejor manera de aprender y desarrollarse en este sector es a través de la autoformación, aprovechando la comunidad y todas las fuentes de información que disponemos hoy en día como es el internet, para maximizar el conocimiento adquirido.

Si bien es cierto que a veces uno se da cuenta de lo mucho que le falta por saber, también he llegado a valorar los conocimientos prácticos que he aplicado en este trabajo y que he adquirido durante la carrera. Las asignaturas con una base práctica sólida han sido fundamentales para la realización de la parte más técnica del proyecto. Asimismo, las asignaturas relacionadas con la gestión han sido de gran ayuda para la organización y planificación del calendario del proyecto y asignaturas más relacionadas con la investigación y explotación de datos han sido de gran utilidad para la parte de investigación relacionada con la parte más teórica del proyecto.

En conclusión, este trabajo me ha ayudado a comprender a nivel personal que siempre hay algo nuevo por aprender y que la formación continua es esencial en este sector. He entendido que los grandes proyectos requieren de grandes equipos, ya que es difícil llevar a cabo desarrollos significativos de manera individual. Este Trabajo de Fin de Grado ha sido una experiencia enriquecedora que subraya la importancia del aprendizaje constante y del trabajo colaborativo en el ámbito profesional.

POSIBLES AMPLIACIONES

Las posibles ampliaciones que se proponen son las siguientes:

- **Arquitectura:** En primer lugar, la ampliación más atractiva sería migrar el proyecto a una arquitectura aún más escalable que la de capas, que es la que se ha utilizado. Si bien es cierto que se ha optado por esta debido al calendario disponible para el desarrollo, lo más correcto hubiera sido desarrollarlo todo en una arquitectura basada en microservicios o hexagonal. Esta migración permitiría una mayor flexibilidad y escalabilidad de la aplicación.
- **Roles y permisos:** En segundo lugar, como se ha mencionado en las conclusiones de la parte práctica, sería interesante disponer de roles en la aplicación para poder delimitar el acceso a los endpoints según el tipo de usuario registrado. Esta funcionalidad mejoraría la seguridad y personalización de la experiencia del usuario.
- **Autenticación:** En tercer lugar, se propone una ampliación relacionada con los roles y permisos, que consiste en mejorar la autenticación dentro del registro de la aplicación. Esto se refiere a disponer de un registro verificado por estatutos legales, es decir, que para registrarse o crear un usuario se utilicen sistemas de respaldo existentes, como los del gobierno, para acceder a las páginas legales de los ciudadanos. Un ejemplo de esto sería la autenticación mediante DNI electrónico o cl@ve pin.
- **Testeo TDD:** Finalmente, otra ampliación que sería prácticamente obligatoria si se decidiera escalar el proyecto en el futuro, sería la implementación de Test Driven Development (TDD). Este enfoque sería imprescindible para la sostenibilidad del sistema y del código, asegurando que el desarrollo futuro se realice de manera ordenada y con menos errores.

Si bien es cierto que pueden existir muchas otras posibles ampliaciones, se ha puesto especial énfasis en estas, ya que se consideran esenciales para la escalabilidad del proyecto y pensando en la finalidad de poner la solución en un entorno real, como sería el mercado de software para centros escolares o empresas.

BIBLIOGRAFIA

- [1] *Soluciones y software informático para centros educativos.* (s.f.). Sumosa, proveedor global de soluciones para empresas. <https://www.sumosa.com/educacion/soluciones-y-software/>
- [2] Navarro, D. (2021, 8 de octubre). Evita problemas legales entre escuela familias. *enlaceoccidente*. <https://www.enlaceeducacionsuperior.com/post/patria-potestad-y-escuela>
- [3] Enlace Occidente A.C. (2021, 21 de julio). *Evita problemas legales con las familias de tus alumnos - T2 E20 De profes a Masters* [Video]. YouTube. <https://www.youtube.com/watch?v=WWZMkIIWX7U>
- [4] *W3Schools Online Web Tutorials.* (s.f.). W3Schools Online Web Tutorials. <https://www.w3schools.com/>
- [5] *Stack Overflow - Where Developers Learn, Share, & Build Careers.* (s.f.). Stack Overflow. <https://stackoverflow.com/>
- [6] *Forums Mysql – Forum Developers* (s.f.). Mysql. <https://forums.mysql.com/>
- [7] *Mysql – DEV Community.* <https://dev.mysql.com/doc/>
- [8] Springboot. (s.f.). *DEV Community.* <https://dev.to/t/springboot>
- [9] *Spring | Community.* (s.f.). Community. <https://spring.io/community>
- [10] *Java Documentation - Get Started.* (2018, 30 de julio). Oracle Help Center. <https://docs.oracle.com/en/java/>
- [11] *Inicio | FEMP - Federación Española de Municipios y Provincias.* (s.f.). Inicio | FEMP - Federación Española de Municipios y Provincias. <https://www.femp.es/>
- [12] GanttProject (s.f.). *GanttProject* (Versión 3.2.3240) [Software de computadora].
- [13] *INE - Instituto Nacional de Estadística.* (s.f.). INE. <https://www.ine.es/jaxiT3/Datos.htm?tpx=58546#!tabs-grafico>

- [14] Vladmihalcea. (2023, 15 octubre). *The best way to map a projection query to a DTO (Data Transfer Object) with JPA and Hibernate*. Vlad Mihalcea.
<https://vladmihalcea.com/the-best-way-to-map-a-projection-query-to-a-dto-with-jpa-and-hibernate/>
- [15] Bonagiri, M. (2022, 5 enero). *Spring Boot Data JPA — Part IV (Projections) - Javarevisited - Medium*. Medium. <https://medium.com/javarevisited/spring-boot-data-jpa-part-iv-projections-9d25a5637dd9>
- [16] GitHub Desktop. (s. f.). *GitHub Desktop*. <https://desktop.github.com/>
- [17] *Download Postman | Get Started for Free*. (s. f.). Postman API Platform. <https://www.postman.com/downloads/>
- [18] MySQL :: MySQL Downloads. (s. f.). <https://www.mysql.com/downloads/>
- [19] *Download IntelliJ IDEA – The Leading Java and Kotlin IDE*. (2021, 1 junio). JetBrains. <https://www.jetbrains.com/idea/download/?section=mac>
- [20] *API Documentation & Design Tools for Teams | Swagger*. (s. f.). <https://swagger.io/>