

PROYECTO DAM

ACADEMY GESTOR

Fecha de presentación:

2024/25 IZAN FRANCO ANDUAGA - 72153424D

INDICE

* INTRODUCCION
* RESUMEN
* OBJETIVOS
* ANALISIS DE CONTEXTO
* ESTADO DEL ARTE
* INNOVACION
* ANALISIS DE REQUISITOS
* DISEÑO

INTRODUCCION

Para este proyecto realizaremos una aplicación para gestionar el alumnado y las actividades de una escuela local.

La idea nació de la situación de una persona cercana, que en su trabajo como profesora en la escuela de circo de Torrelavega advirtió, junto al resto del equipo, la necesidad de mejorar las capacidades de gestión de la escuela, para agilizar las tareas y poder realizar estas de forma más eficiente, hasta ahora no se había encontrado ninguna alternativa ya desarrollada que se ajuste a sus necesidades o, con demasiadas funciones, de las cuales utilizarían tan solo una parte mínima.

El planteamiento de este proyecto se centra en desarrollar una aplicación más sencilla pero centrada en las funciones esenciales que se van a utilizar, teniendo en cuenta a profesores y alumnos, desde las solicitudes de inscripción hasta el registro de pagos de cada uno.

De esta manera se pretende reducir las numerosas horas trabajo administrativo (hasta ahora todos estos datos se han manejado a mano) que, si se gestionan a través de las herramientas adecuadas se pueden ver notablemente reducidas, dedicando así más horas a otras labores igual de importantes pero quizá menos apremiantes.

En consecuencia, este proyecto tratará de proveer las herramientas para hacer de este trabajo más liviano.

RESUMEN

Academy Gestor

Se plantea como una aplicación para facilitar las tareas administrativas de en este caso una escuela de circo, se proporcionarán una aplicación de escritorio desde la cual se pueda gestionar tanto las inscripciones de alumnos como las resultantes matriculas, los cursos ofertados, el personal de la escuela y tener un control de los recibos generados para cada alumno; y una app móvil con un doble propósito, por un lado y mediante inicio de sesión permitirá a los profesores realizar los controles de asistencia y generación de informes sobre el objetivo de las clases. Por otro lado, será un espacio informativo para las familias donde se publicarán anuncios con respecto a las actividades de la escuela, contenido visual referente a los distintos cursos, información básica, contacto etc…

PALABRAS CLAVE

* Gestor
* Escuela
* Matriculas
* Extraescolares
* Academia
* Cursos
* Talleres
* Gestión académica

OBJETIVOS

* Objetivo principal: Agilizar las labores de gestión de una escuela de extraescolares y facilitar la comunicación con las familias.
* Objetivos específicos:

Obj 1- Automatizar registros de datos de alumnos

Obj 2- Gestión de alumnos (dar de alta/ baja alumnos, modificar sus datos…)

Obj 3- Gestión de matriculas (consulta, modificación, registro…)

Obj 4- Gestión de cursos (registro de cursos disponibles y capacidad para crear, modificar o eliminar los existentes)

Obj 5- Gestión de personal (registro de datos de profesores, pudiendo crear, eliminar y modificar)

Obj 6- Control de recibos (se registraran automáticamente al inicio del mes, y de cada uno se podrá generar un recibo y/o enviarlo por correo al alumno o tutor)

Obj 7- Publicación de noticias referentes a la escuela

Obj 8- Publicación de contenido visual de los distintos cursos o eventos

Obj 9- Canal de información y contacto para las familias

Obj 10- Control de asistencia e incidencias para los profesores

ANALISIS DE CONTEXTO

El presente proyecto busca modernizar la gestión y comunicación de una escuela de circo mediante la implementación de una base de datos centralizada y el desarrollo de aplicaciones de escritorio y móvil. Para comprender mejor el impacto de esta iniciativa se realiza el siguiente análisis de contexto.

En cuanto al contexto tecnológico actual, muestra una alta penetración de dispositivos inteligentes y un crecimiento sostenido en el uso de herramientas digitales en el ámbito educativo y de gestión administrativa. Según datos recientes el uso de *smartphone* es elevado en la población, lo que facilita la adopción de soluciones digitales. La automatización de procesos ha incrementado en los últimos años reduciendo errores y mejorando la eficiencia administrativa, también la implementación de sistemas de gestión automatizados permite optimizar el tiempo de los administradores.

El mantenimiento de procesos manuales puede verse traducido en costes ocultos en términos de ineficiencia y errores administrativos. La automatización de tareas mediante una base de datos y aplicaciones reducirá los tiempos administrativos, optimizará el uso de recursos y permitirá una mejor organización de la información, lo que a largo plazo puede generar ahorros significativos en la gestión. Además, la reducción del uso de papel para listas de asistencia y notificaciones contribuirá a una gestión más sostenible disminuyendo costes de impresión y materiales.

El público objetivo del proyecto incluye tanto a los administradores de la escuela de circo como a los estudiantes y sus familias. Se han identificado tendencias como la preferencia por la comunicación digital, donde la mayoría de las familias ya utiliza WhatsApp para la comunicación con la escuela lo que indica una disposición a adoptar soluciones digitales complementarias, también se observa un uso creciente de aplicaciones móviles lo que mejorará la interacción entre la escuela y los padres así como un fomento de la inclusión digital, que contribuirá a mejorar la organización de la escuela y la participación de las familias en las actividades.

También, se deben considerar aspectos legales relacionados con la protección de datos personales, asegurando el cumplimiento de normativas de privacidad en la implementación de la base de datos y aplicaciones, garantizando la seguridad de la información de estudiantes y familias. Asimismo, se deben tener en cuenta regulaciones sobre comunicaciones digitales, para el uso de aplicaciones en la difusión de información y la gestión de inscripciones ajustándose a las normativas vigentes en términos de consentimiento y acceso a la información.

Se espera que la digitalización de los procesos en la escuela de circo genere beneficios a medio y largo plazo como la reducción de errores y mayor eficiencia administrativa, una mayor participación de las familias a través de la aplicación móvil, la facilitación del acceso a la información y mejora en la organización interna. En conclusión este proyecto representa una solución que transformará la gestión optimizando procesos mejorando la comunicación y alineándose con las tendencias tecnológicas actuales.

ESTADO DEL ARTE

El presente análisis del estado del arte examina las tecnologías y soluciones actuales utilizadas en la gestión de inscripciones, matrículas y comunicación en escuelas similares, así como las tendencias tecnológicas y los productos existentes en el mercado que abordan problemas similares.

SITUACION ACTUAL

Actualmente, muchas instituciones pequeñas y medianas continúan utilizando herramientas tradicionales como hojas de cálculo (Excel) y comunicaciones manuales (WhatsApp) para la gestión de estudiantes y notificaciones. Sin embargo, la tendencia global se inclina hacia la digitalización y automatización mediante plataformas especializadas que ofrecen bases de datos centralizadas, aplicaciones móviles y soluciones en la nube. Herramientas como Google Classroom o Moodle han ganado popularidad en la gestión educativa, pero están diseñadas principalmente para la educación formal y no se adaptan fácilmente a un entorno de formación más informal y flexible.

En cuanto a la tecnología utilizada, la mayoría de las soluciones actuales emplean bases de datos SQL o documentales para almacenar información, aplicaciones web responsivas para gestión administrativa y aplicaciones móviles para la interacción con los usuarios. La integración de servicios de mensajería automática también es común en plataformas más avanzadas.

COMPETENCIA

El sector de la gestión educativa cuenta con diversas soluciones que pueden considerarse competencia para este proyecto, aunque no todas ofrecen un enfoque específico para la formación artística y circense. Algunas de las soluciones más destacadas incluyen:

* **ClassDojo**: Diseñada para la comunicación entre docentes y familias, permite gestionar asistencia y enviar notificaciones, pero no ofrece una solución completa para la gestión de inscripciones y matrículas.
* **EducaPro**: Una plataforma de pago que facilita la gestión académica y administrativa, pero su costo es elevado para escuelas pequeñas y su interfaz está más orientada a la educación escolar tradicional.

Las soluciones existentes destacan por facilitar la comunicación entre estudiantes, familias y administradores, y proporcionan automatización en ciertos aspectos como recordatorios y almacenamiento en la nube. Asimismo, cuentan con interfaces intuitivas y adaptadas para dispositivos móviles. Sin embargo, presentan carencias, como la ausencia de una gestión flexible de inscripciones y matrículas adaptada a clases no tradicionales o la falta de una integración fluida entre bases de datos de inscripciones, asistencia y comunicación.

|  |  |
| --- | --- |
| FORTALEZAS | OPORTUNIDADES |
| * Adaptación a las necesidades especificas * Facilidad de uso * Integración de funciones clave | * Auge de la digitalización educativa * Posible expansión a otras escuelas |
| DEBILIDADES | AMENAZAS |
| * No hay precedentes específicos * Curva de aprendizaje * Algunas opciones de mercado son costosas | * Resistencia al cambio en la institución * Pueden surgir soluciones especializadas |

En conclusión, si bien existen plataformas para la gestión educativa y la comunicación entre docentes y familias, ninguna está diseñada específicamente para un entorno extraescolar. Este proyecto representa una oportunidad para ofrecer una solución adaptada que cubra las carencias identificadas en la competencia y permita a la institución modernizar su gestión de manera eficiente y accesible.

INNOVACION

Este proyecto aporta una solución específica que cubre necesidades no resueltas por productos existentes. A diferencia de las plataformas educativas tradicionales, permite gestionar inscripciones y matrículas de manera flexible, adaptándose a clases no convencionales. Además, la automatización de asistencia y la integración de comunicación directa con las familias a través de una aplicación móvil mejoran la eficiencia del sistema actual.

La principal innovación radica en la centralización de funciones clave en una sola plataforma accesible desde dispositivos móviles, eliminando la necesidad de registros en papel y hojas de cálculo manuales.

El cambio de paradigma en este sector se da al reemplazar los métodos tradicionales de gestión manual por una solución digital específica para centros de formación artística. Además, se introduce un sistema de control de asistencia optimizado para dispositivos móviles, lo que permite a los instructores gestionar listas de alumnos en cualquier momento y lugar.

En cuanto a soluciones innovadoras aplicadas, este proyecto utiliza una combinación de metodologías ágiles para el desarrollo de software, bases de datos escalables y una arquitectura modular basada en microservicios, lo que permite una mejor adaptabilidad y evolución del sistema a futuro.

ANALISIS DE REQUISITOS

**Requisitos Funcionales**

| **ID** | **Nombre** | **Descripción** | **Prioridad** | **Objetivo Relacionado** |
| --- | --- | --- | --- | --- |
| RF-01 | Gestión de alumnos | Permite registrar, modificar y eliminar datos de los alumnos. | Alta | Automatizar registros de datos de alumnos. |
| RF-02 | Inscripción y matrículas | Permite la inscripción de nuevos alumnos y la gestión de matrículas. | Alta | Gestión de alumnos y matrículas. |
| RF-03 | Gestión de cursos | Posibilidad de crear, modificar y eliminar cursos con información detallada. | Alta | Gestión de cursos y personal. |
| RF-04 | Control de asistencia | Permitir a los profesores registrar la asistencia de los alumnos desde la app móvil. | Alta | Control de asistencia e incidencias para los profesores. |
| RF-05 | Generación de informes | Los profesores podrán generar informes sobre el desempeño de los alumnos. | Media | Control de asistencia e incidencias para los profesores. |
| RF-06 | Gestión de pagos y recibos | Registrar pagos y generar recibos. | Alta | Control de recibos. |
| RF-07 | Publicación de noticias | Publicar anuncios y noticias sobre la escuela. | Media | Publicación de noticias referentes a la escuela. |
| RF-08 | Espacio de comunicación con familias | Los padres podrán consultar información y comunicarse con la escuela a través de la app móvil. | Media | Canal de información y contacto para las familias. |
| RF-09 | Gestión del personal | Registrar y gestionar el personal de la escuela. | Media | Gestión de cursos y personal. |
| RF-10 | Integración con sistemas de mensajería | Posibilidad de enviar notificaciones a alumnos y familias sobre eventos o cambios en los cursos. | Media | Mejorar la comunicación con las familias. |

### ****Requisitos No Funcionales****

| **ID** | **Nombre** | **Descripción** | **Prioridad** |
| --- | --- | --- | --- |
| RNF-01 | Seguridad de datos | Protección de los datos de alumnos y profesores mediante cifrado. | Alta |
| RNF-02 | Cumplimiento de normativas | Cumplimiento con la normativa de protección de datos (GDPR). | Alta |
| RNF-03 | Plataforma multi-dispositivo | Accesible desde ordenadores y dispositivos móviles. | Alta |
| RNF-04 | Tiempo de respuesta óptimo | Las operaciones críticas deben ejecutarse en menos de 2 segundos. | Alta |
| RNF-05 | Usabilidad intuitiva | La interfaz debe ser intuitiva y fácil de usar para personas sin conocimientos técnicos. | Media |
| RNF-06 | Escalabilidad | Permitir la ampliación de funcionalidades sin afectar el rendimiento. | Media |
| RNF-07 | Copias de seguridad | Implementación de un sistema de respaldo automático de la base de datos. | Alta |
| RNF-08 | Alta disponibilidad | Garantizar una disponibilidad elevada para evitar interrupciones. | Alta |
| RNF-09 | Diseño adaptable | Ofrecer opciones de personalización para la interfaz | Media |

DISEÑO

**DIAGRAMA FISICO**

Servidor Ubuntu

Servicios:

MySQL Server: motor de base de datos central.

Apache2:

Aloja la API REST.

Almacena archivos accesibles vía HTTP.

Uso compartido:

Punto de conexión para las aplicaciones (escritorio y móvil).

Aplicación de Escritorio

Tecnología: C# con Windows Forms

Cliente:

Consume la API REST.

Puede descargar y subir archivos al servidor mediante la API.

Muestra o edita datos alojados en la base de datos.

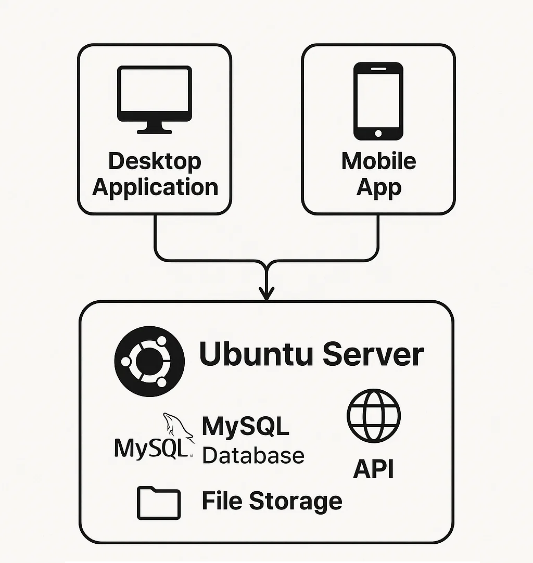
Aplicación Móvil

Tecnología: Kotlin (Android)

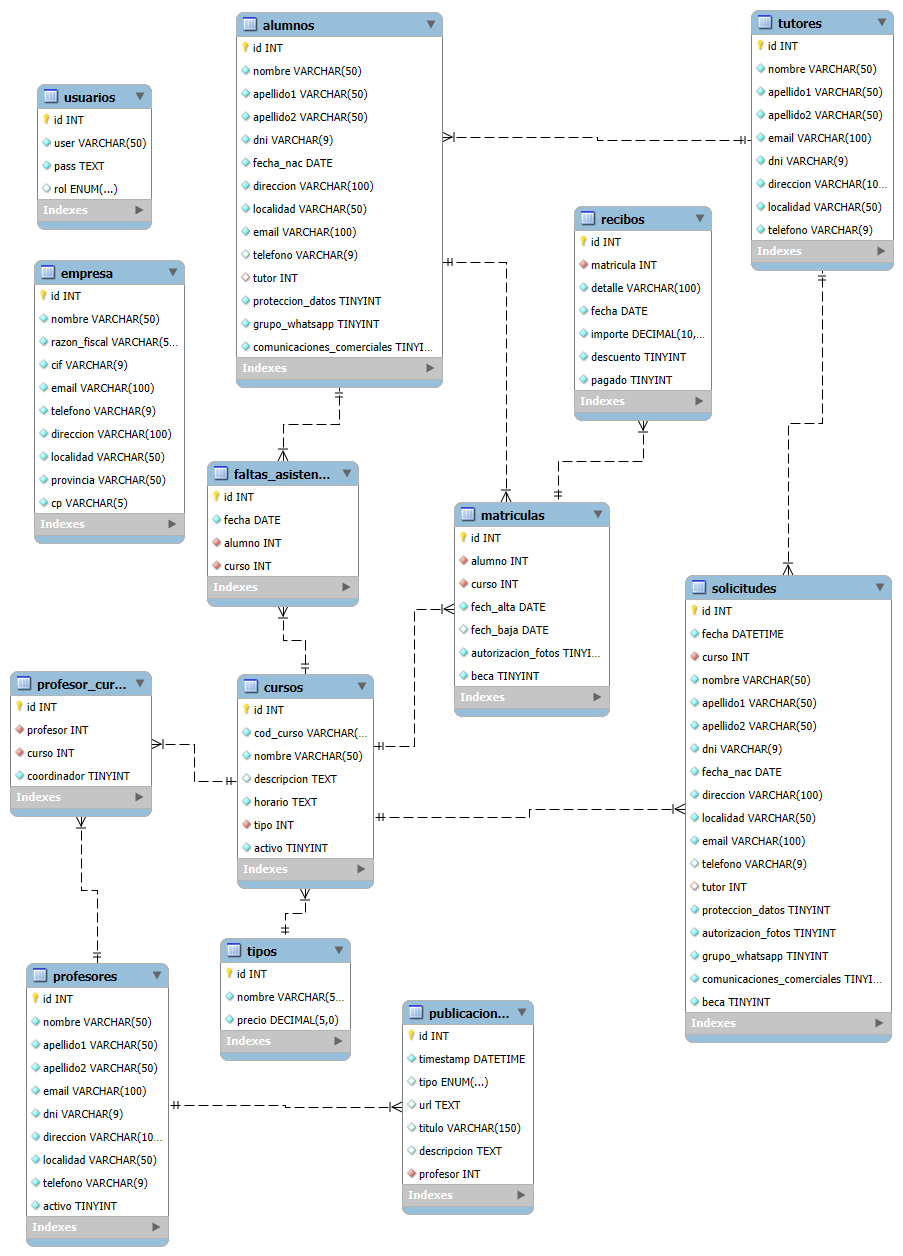
Cliente móvil:

Se conecta a la API REST usando Retrofit.

Accede también al almacenamiento para ver o enviar archivos.

****

**DIAGRAMA RELACIONAL**

****

**EXPLICACION:**

**La base de datos recoge los registros de la escuela de circo, reflejados en las siguientes tablas:**

* **Empresa: se recogen los datos de la empresa, id (PK), nombre, razón fiscal, CIF, email, teléfono, dirección, localidad, provincia y código postal. No guarda relación con ninguna otra tabla.**
* **Usuario: datos de los usuarios que pueden acceder a las aplicaciones, id (PK), usuario, contraseña y rol (admin para quienes gestionaran todos los registros de la escuela desde la aplicación de escritorio, user destinado a los profesores a la hora de usar la aplicación móvil para pasar lista, subir publicaciones etc…). No guarda relación con otras tablas.**
* **Solicitudes: recoge los datos de alumnos y tutores legales para solicitar nuevas inscripciones, que una vez aceptadas generaran registros en las tablas relacionadas. id (PK), fecha de la solicitud, curso que se solicita(FK de la tabla curso),(datos del alumno) nombre, apellido1, apellido2, DNI, fecha de nacimiento, dirección, localidad, email, teléfono, tutor (FK de la tabla tutores), protección de datos, autorización de fotos grupo de WhatsApp, comunicaciones comerciales y si tiene o no beca.**
* **Alumnos: todos los datos de los alumnos, id (PK), nombre, apellido1, apellido2, DNI, fecha de nacimiento, dirección, localidad, email, teléfono, tutor (FK de la tabla tutores), protección de datos, grupo de WhatsApp y comunicaciones comerciales, estas tres últimas son booleanos que indican si se aceptan o no dichas condiciones.**
* **Tutores: datos de los tutores legales de los alumnos menores de edad, id (PK), nombre, apellido1, apellido2, email, DNI, dirección, localidad, teléfono.**
* **Faltas de asistencia: registros de faltas de alumnos, id (PK), fecha, alumno (FK tabla alumnos), curso (FK tabla cursos).**
* **Matriculas: registros para cada curso que realiza un alumno, id (PK), alumno (FK tabla alumnos), curso (FK tablas cursos), fechas de alta y de baja la cual puede ser nula, si se tiene autorización para fotos y beca.**
* **Recibos: registros mensuales para cada matricula para control de pagos, id (PK), matricula (FK tabla matriculas), detalle, fecha, importe, si tiene descuento por beca, si esta pagado.**
* **Cursos: datos de cada curso ofrecido, id (PK), código, nombre, descripción, horario, tipo (FK tabla tipos), y si esta o no activo.**
* **Tipos: tipos de cursos impartidos, id (PK), nombre y precio.**
* **Profesores: datos de los profesores, id (PK), nombre, apellido1, apellido2, email, DNI, dirección, localidad, teléfono y si esta activo.**
* **Profesores\_Curso: tabla intermedia entre cursos y profesores que recoge los profesores participantes en cada curso y si son o no coordinadores, id (PK), profesor (FK tabla profesores), curso (FK tabla cursos), y si es coordinador (booleano).**
* **Publicaciones: datos de las publicaciones subidas por cada profesor.**

**DIAGRAMA DE CLASES**

DESCRIPCIÓN DE CLASES DEL SISTEMA

* Solicitud
  + Atributos:  
    id, nombre, apellido1, apellido2, email, telefono, dni, fechaNac, direccion, localidad, grupoWhatsapp, proteccionDatos, autorizacionFotos, comunicacionesComerciales, fecha, curso, tutor
  + Métodos: getters y setters para cada atributo.
  + Relaciones:

curso - referencia a Curso

tutor - referencia a Tutor

* Curso
  + Atributos:  
    id, nombre, descripcion, horario, activo, tipo
  + Métodos: getters y setters.
  + Relaciones:

tipo - referencia a Tipo

Relación con Alumno, ProfesorCurso, FaltasAsistencia, Matricula

* Tipo
  + Atributos:  
    id, nombre, precio
  + Métodos: getters y setters.
  + Relaciones:

Referenciado por Curso

* Tutor
  + Atributos:  
    id, nombre, apellido1, apellido2, email, telefono, dni, direccion, localidad
  + Métodos: getters y setters.
  + Relaciones:

Relacionado con Alumno (como padre/tutor)

Relacionado con Solicitud (si el alumno es menor de edad)

* Profesor
  + Atributos:  
    id, nombre, apellido1, apellido2, email, telefono, dni, direccion, localidad
  + Métodos: getters y setters.
  + Relaciones:

Relacionado con ProfesorCurso

* Publicacion
  + Atributos:  
    id, titulo, descripcion, profesor, uri, timestamp, email
  + Métodos: getters y setters.
  + Relaciones:

No se relaciona directamente con otras entidades

Profesor – referencia a Profesor

* ProfesorCurso
  + Atributos:  
    id, curso, profesor, coordinador
  + Métodos: getters y setters.
  + Relaciones:

curso - referencia a Curso

profesor - referencia a Profesor

Un solo profesor puede ser coordinador por curso

* Alumno
  + Atributos:  
    id, nombre, apellido1, apellido2, dni, fechaNac, direccion, localidad, grupoWhatsapp, proteccionDatos, comunicacionesComerciales, tutor
  + Métodos: getters y setters.
  + Relaciones:

tutor - referencia a Tutor

Asociado a Matricula, FaltasAsistencia

* FaltasAsistencia
  + Atributos:  
    id, fecha, curso, alumno
  + Métodos: getters y setters.
  + Relaciones:

curso - referencia a Curso

alumno - referencia a Alumno

* Matricula
  + Atributos:  
    id, fechaAlta, fechaBaja, beca, autorizacionFotos, curso, alumno
  + Métodos: getters y setters.
  + Relaciones:

curso - referencia a Curso

alumno - referencia a Alumno

* Recibo
  + Atributos:  
    id, detalle, descuento, pagado, importe, fecha, matricula
  + Métodos: getters y setters.
  + Relaciones:

matricula - referencia a Matricula

* Usuario
  + Atributos:  
    id, user, pass, rol
  + Métodos: getters y setters.
  + Relaciones:

No tiene relaciones directas en el modelo. Usado para autenticación del sistema.

* Empresa
  + Atributos:  
    id, nombre, razonFiscal, cif, direccion, localidad, provincia, telefono, email
  + Métodos: getters y setters.
  + Relaciones:

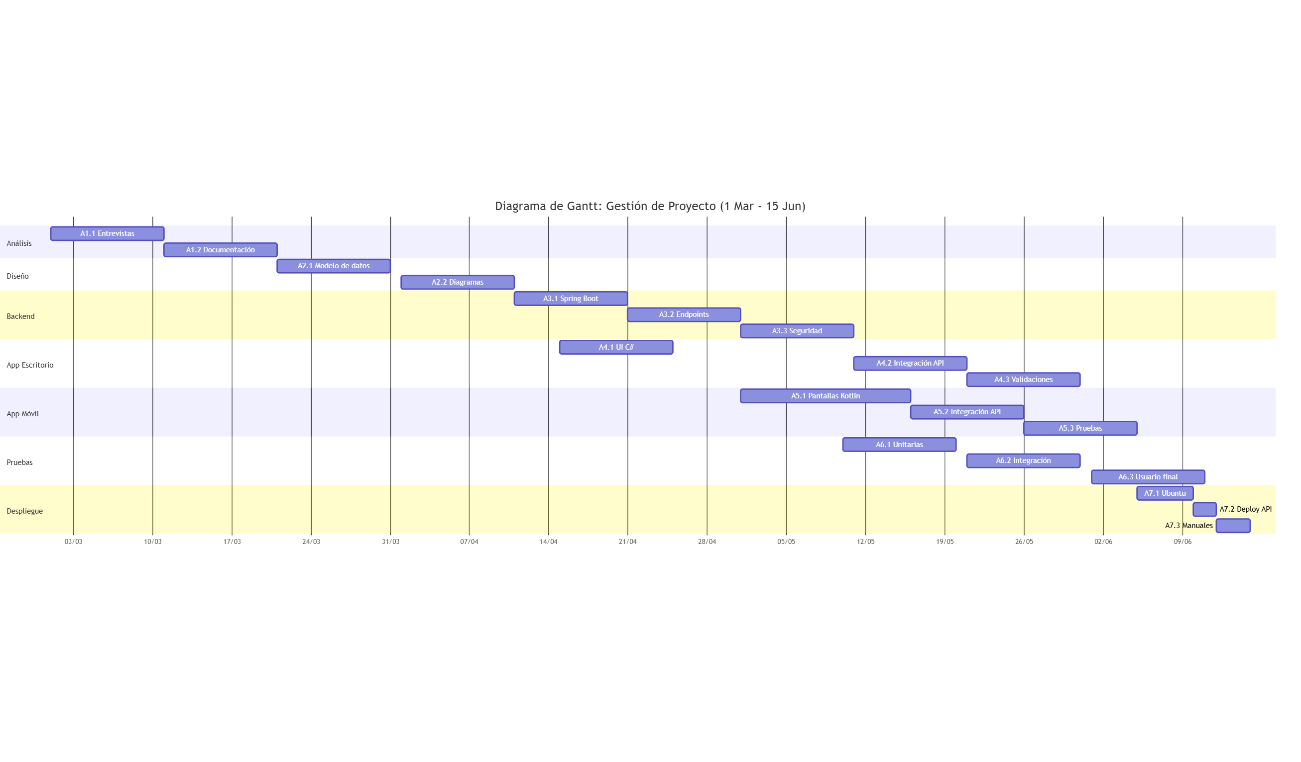
No se relaciona directamente con otras clases en el diagrama

**DIAGRAMA DE INTERFACES**

PLANIFICACION

DEFINICIÓN DE ACTIVIDADES Y SUBTAREAS

|  |  |  |
| --- | --- | --- |
| ID | Actividad Principal | Subtareas |
| A1 | Análisis y Recolección de Requisitos | A1.1 Entrevistas  A1.2 Documentación funcional |
| A2 | Diseño del Sistema | A2.1 Modelo de datos  A2.2 Diagramas de clases y arquitectura |
| A3 | Desarrollo Backend (API) | A3.1 Estructura Spring Boot  A3.2 Endpoints  A3.3 Seguridad |
| A4 | Desarrollo Aplicación de Escritorio | A4.1 UI C#  A4.2 Integración API  A4.3 Validaciones |
| A5 | Desarrollo App Móvil | A5.1 Pantallas Kotlin  A5.2 Integración API  A5.3 Pruebas |
| A6 | Pruebas del Sistema | A6.1 Unitarias  A6.2 Integración  A6.3 Usuario final |
| A7 | Despliegue y Documentación | A7.1 Instalación Ubuntu  A7.2 Deploy API  A7.3 Manuales |



IMPLEMENTACION / DESARROLLO

IDEs utilizados:

* IntelliJ, Visual Studio 2022, Android Studio.

Plataformas de implementación:

* Maquina virtual para simulación de servidor web y de base de datos.

Lenguajes de programación:

* Java, C#, Kotlin.

Frameworks:

* SpringBoot, .Net, Gradle.

Base de datos:

* Base de datos creada para el proyecto.

Control de versiones:

* Git.

PRUEBAS

Estrategias:

* Pruebas unitarias:  
  Se desarrollarán para todas las partes lógicas clave, especialmente en la API y en la lógica de negocio de las aplicaciones de escritorio y móvil. Se usarán frameworks como JUnit y NUnit.
* Pruebas de integración:  
  Se comprobará la correcta interacción entre componentes del. Se automatizarán en parte con herramientas como Postman o RestAssured para la API.
* Pruebas de sistema:  
  Se probará el sistema completo desde el punto de vista del usuario, asegurando que los flujos funcionales se ejecutan correctamente.
* Pruebas de rendimiento:  
  Principalmente aplicadas a la API, para verificar tiempos de respuesta, carga y concurrencia.
* Pruebas de seguridad:  
  Se verificarán aspectos como autenticación, autorización y protección contra inyecciones en la API.
* Pruebas de interfaz de usuario (UI):  
  Para la aplicación de escritorio y móvil, se realizarán pruebas funcionales en la interfaz para validar la correcta disposición, comportamiento e interacción de los elementos.

Diseño y ejecución de pruebas

* Diseño de las pruebas:  
  Las pruebas se diseñan en paralelo con el desarrollo, a medida que se finaliza cada módulo o funcionalidad.

Ejecución de las pruebas:

* Las pruebas unitarias se ejecutan automáticamente con cada nueva compilación.
* Las pruebas de integración y de sistema se ejecutan al finalizar cada sprint o bloque de desarrollo.
* Las pruebas por parte de los usuarios se ejecutan al final del desarrollo y durante la fase de entrega.

Responsables de las pruebas

* Desarrollador

Evaluación por parte del cliente

Objetivos de esta evaluación son:

* Confirmar que todas las funcionalidades cumplen los requisitos iniciales.
* Detectar posibles problemas de usabilidad o errores funcionales menores.
* Recoger observaciones finales para mejoras post-entrega.

Aspectos relevantes

* Se llevará un control de errores detectados, su prioridad, y su estado (pendiente / resuelto).
* Se aplicará control de versiones para garantizar la trazabilidad de las pruebas respecto a los cambios realizados.
* El sistema no se considera “apto para entrega” hasta que no supere el 100% de los casos de prueba críticos.