

******TODO LO QUE ESTÁ SUBRAYADO EN COLOR AMARILLO O AZUL ES PARA ELIMINAR
SOLO SIRVE PARA INFORMAR DE QUE HACER EN DICHO APARTADO******

PROYECTO FIN DE GRADO

DESARROLLO DE APLICACIONES

MULTIPLATAFORMA

NOMBRE Y APELLIDOS: **Kleivan Alexander Torrealba Blanco**

TÍTULO DEL PROYECTO: **(Rellenar por el alumno)**

CURSO ESCOLAR 2024-2025



Tutora de proyecto integrado

M.^a Carmen Buenestado Fernández

RESUMEN

Es necesario escribirlo de una manera objetiva e interesante, dejando muy claro, y de inmediato, de qué trata el proyecto.

- No exceder las 200 o 250 palabras.
- Debe incluir sólo los puntos más importantes del proyecto, ser claro y conciso, evitando comentarios y generalidades.
- Debe estar escrito con oraciones sencillas y coherentes.
- No debe contener datos o consideraciones que no figuren en el desarrollo del documento.
- Se deben evitar las referencias a secciones, figuras, tablas, citas bibliográficas, gráficos, etc.
- Escrito en voz activa y forma impersonal (por ejemplo, “se analiza..., etc.”)

Mas información: <https://hazmitrabajo.es/blog/abstract/>

(Dejar para últimas entregas)

ABSTRACT

Escribir el mismo contenido que en apartado anterior ‘Resumen’ pero en idioma inglés. Se puede usar traductor de Google.

(Dejar para últimas entregas)

ÍNDICE

1. INTRODUCCIÓN.....	5
1.1. PLANTEAMIENTO DEL PROBLEMA.....	5
1.2. ANÁLISIS DEL MERCADO.....	5
1.3. JUSTIFICACIÓN DEL PROYECTO.....	5
2. DEFINICIÓN DEL PROYECTO.....	5
2.1. OBJETIVOS GENERALES Y ESPECÍFICOS.....	5
2.2. CONTEXTO Y PÚBLICO OBJETIVO.....	5
3. ANÁLISIS DE LA APLICACIÓN.....	5
3.1. MEDIOS HARDWARE Y SOFTWARE A UTILIZAR.....	5
3.2. METODOLOGÍA DE DESARROLLO.....	6
3.3. PRESUPUESTO Y ANÁLISIS DE COSTOS.....	6
4. DISEÑO DE LA APLICACIÓN.....	6
4.1. ESPECIFICACIÓN DE REQUISITOS.....	6
4.1.1. REQUISITOS FUNCIONALES.....	6
4.1.2. CASOS DE USO.....	7
4.1.3. REQUISITOS NO FUNCIONALES.....	7
4.1.4. INTERFACES DE USUARIO.....	7
4.2. RESTRICCIONES DE DISEÑO.....	8
5. DISEÑO DE LA BASE DE DATOS.....	8
5.1. DESCRIPCIÓN GENERAL.....	8
5.2. MODELO DE DATOS.....	9
5.2.1. MODELO ENTIDAD-RELACIÓN (Si tu base de datos es del tipo entidad-relación)...	9
5.2.2. MODELO XXXX (Si tu base de datos sigue otro modelo, por ejemplo NoSQL).....	9
5.3. PASO A ESTRUCTURAS DE ALMACENAMIENTO.....	9
5.4. DIAGRAMA DE TABLAS DE LA BASE DE DATOS.....	11
5.5. DICCCIONARIO DE DATOS.....	11
5.6. SCRIPT DE CREACIÓN E INSERCIÓN DE LA BASE DE DATOS.....	11
6. VISIÓN GENERAL DEL CÓDIGO.....	12
7. PRUEBAS Y VALIDACIONES.....	12
8. DIFICULTADES DETECTADAS Y POSIBLES MEJORAS.....	12
9. TEMPORALIZACIÓN DEL PROYECTO.....	12
10. DESPLIEGUE DE LA APLICACIÓN.....	12
11. MANUAL DE USUARIO.....	13
12. MANTENIMIENTO.....	13
13. CONCLUSIONES.....	13
14. GLOSARIO DE SIGLAS Y ABREVIATURAS.....	13

15. BIBLIOGRAFÍA.....	14
16. ANEXOS.....	14

1. INTRODUCCIÓN

Es una aplicación de Gestión y Planificación de Espectáculos Musicales la cual diseñada para facilitar la organización y coordinación de eventos musicales de manera eficiente y accesible. Con esta app, los organizadores de espectáculos podrán gestionar todos los aspectos relacionados con los eventos, como la planificación de fechas, la asignación de artistas y la gestión de recursos.

1.1. PLANTEAMIENTO DEL PROBLEMA.

Este proyecto será un programa para Android enfocada en la gestión y planificación de espectáculos musicales. Cuyo objetivo es el de facilitar la organización de eventos, permitiendo a los usuarios coordinar horarios, ubicaciones y recursos de manera eficiente. La app busca resolver los desafíos logísticos que enfrentan los organizadores, optimizando la comunicación y la planificación en un solo lugar.

1.2. ANÁLISIS DEL MERCADO

En el mercado de hoy en día, hay diversas herramientas orientadas a la gestión de eventos. Y aunque muchas de ellas no están específicamente diseñadas para el ámbito de los espectáculos musicales. Existen algunas aplicaciones populares como Eventbrite, Asana, Trello o Google Calendar ofrecen soluciones para organizar tareas, agendar eventos y coordinar equipos. Sin embargo, abordan de forma integral las necesidades específicas del sector musical como lo son: programación de shows, asignación de ubicaciones, gestión de riders técnicos o la logística del personal artístico, entre otros.

Por otro lado, plataformas como Bandsintown Manager y Gigwell están más orientadas a músicos y agentes para la gestión de giras y presentaciones, sin enfocarse tanto en la coordinación general de espectáculos desde el punto de vista de un productor o planificador de eventos.

1.3. JUSTIFICACIÓN DEL PROYECTO

La industria del espectáculo requiere una planificación precisa y una gestión eficiente de recursos humanos y materiales para garantizar montajes exitosos, pero la falta de herramientas digitales especializadas dificulta la coordinación entre los equipos de producción.

Este programa busca resolver estos desafíos proporcionando una plataforma centralizada para la gestión de espectáculos musicales.

2. DEFINICIÓN DEL PROYECTO

2.1. OBJETIVOS GENERALES Y ESPECÍFICOS.

Los objetivos generales los cuales busco cumplir con esta aplicación, son:

- facilitar la organización de espectáculos
- permitir gestionar los roles de usuario.
- mejorar la asignación de los recursos y de equipos
 - Integrar un sistema de comunicación para facilitar la coordinación entre el equipo encargado del montaje y las bandas.
 - Ofrecer un historial y seguimiento de eventos
 - garantizar la accesibilidad desde dispositivos móviles.

Los objetivos específicos de la aplicación son:

- implementar un sistema de autenticación
- desarrollar una base de datos con información sobre los eventos, usuarios, equipos, recursos necesarios para el montaje e incluso lugares.
- Permitir la creación, edición y eliminación de eventos
- integrar un sistema de notificaciones para mantener informados a los usuarios sobre cambios en el montaje
- diseñar una interfaz intuitiva y responsive
- incorporar un sistema de registro de equipos y materiales
- optimizar la aplicación para un rendimiento eficiente
- garantizar la seguridad de los datos

2.2. CONTEXTO Y PÚBLICO OBJETIVO

El propósito de esta aplicación es facilitar la gestión y coordinación del montaje de espectáculos, proporcionando una plataforma centralizada donde los usuarios puedan planificar eventos y comunicarse de manera eficiente.

El alcance de esta aplicación estará dirigida a organizadores de eventos, técnicos de montaje y demás personal involucrado en la producción de espectáculos musicales en las que permitirá el acceso mediante un sistema de autenticación.

3. ANÁLISIS DE LA APLICACIÓN

3.1. MEDIOS HARDWARE Y SOFTWARE A UTILIZAR

Los medios hardware son:

- Un dispositivo Android o un emulador de Android para comprobar el funcionamiento del programa.

- Un ordenador con S.O Windows para el desarrollo del programa
- Un servidor encargado de almacenar los datos-

Los medios software planteados en un inicio son:

- visual estudio code: Será el editor de código del programa.
- .NET SDK: Será la base para desarrollar en C#
- Android SDK: para compilar y probar la aplicación en Android
- .NET MAUI Workload: será necesario para el desarrollo de aplicaciones móviles en C#.
- PostgreSQL: es la base de datos.

3.2. METODOLOGÍA DE DESARROLLO

Escogeré la metodología Scrum porque ayuda a organizar tareas, priorizar y avanzar de forma estructurada en el desarrollo.

3.3. PRESUPUESTO Y ANÁLISIS DE COSTOS

En un principio usaré solamente herramientas gratuitas, al igual que me encargaré yo de todo el diseño. Lo único que tengo que costear sería la tarifa única de Google Play Developer (25\$)

4. DISEÑO DE LA APLICACIÓN

4.1. ESPECIFICACIÓN DE REQUISITOS

La especificación de requisitos es una parte fundamental que ayuda a definir qué funcionalidades y características debe tener el software que se va a desarrollar. En este punto deja una introducción sobre lo que contiene este punto. Debes incluir los siguientes puntos:

4.1.1. REQUISITOS FUNCIONALES

Gestión de Eventos

- Crear, editar y eliminar espectáculos musicales.
- Asignar fechas, horarios y ubicaciones a cada evento.
- Gestionar listas de artistas y bandas participantes.

Gestión de Recursos

- Registrar y administrar equipos, instrumentos y personal.
- Asignar recursos a cada evento según disponibilidad.

Usuarios y Roles

- Registro e inicio de sesión para organizadores y artistas.
- Diferentes roles de usuario con permisos específicos (administrador, staff, artista).

Comunicación y Coordinación

- Enviar notificaciones sobre cambios o recordatorios de eventos.
- Chat interno o foro de discusión entre organizadores y artistas.

Reportes y Seguimiento

- Visualización de eventos en un calendario interactivo.
- Generación de reportes sobre uso de recursos y asistencia esperada.

4.1.2. CASOS DE USO

Es una herramienta gráfica que se utilizará para representar las interacciones entre los actores (usuarios externos) y el sistema en sí. Cada caso de uso representa una funcionalidad específica que el sistema puede realizar. Los diagramas de casos de uso ayudan a identificar y documentar los requisitos funcionales del sistema, lo que facilita la planificación del desarrollo.

En este apartado debes incluir el diagrama de casos de uso general de la aplicación en el que se refleje cada una de las funcionalidades que puede realizar cada uno de los actores participantes.

En este link tienes información y ejemplos. <https://diagramasuml.com/casos-de-uso/>

Puedes utilizar PowerPoint, Word o la herramienta online drawio (link <https://app.diagrams.net/>) para realizarlos.

También debes incluir una tabla como la mostrada en este link y al final de esa página para representar los pasos que el actor principal debe realizar para realizar una función de alta y baja del elemento que escogas.

<https://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/416>.

4.1.3. REQUISITOS NO FUNCIONALES

Rendimiento

- La app debe cargar y mostrar eventos en menos de 2 segundos.
- El procesamiento de datos no debe bloquear la interfaz de usuario.

Usabilidad

- Implementar una Interfaz intuitiva y fácil de usar para organizadores y artistas.
- Diseño responsive para adaptarse a diferentes tamaños de pantalla.

Seguridad

- Autenticación de usuarios.
- Cifrado de datos sensibles en la base de datos.

Escalabilidad

- La app debe permitir agregar más usuarios y eventos sin perder rendimiento.
- Posibilidad de migrar la base de datos a un servidor más potente si es necesario.

Mantenimiento

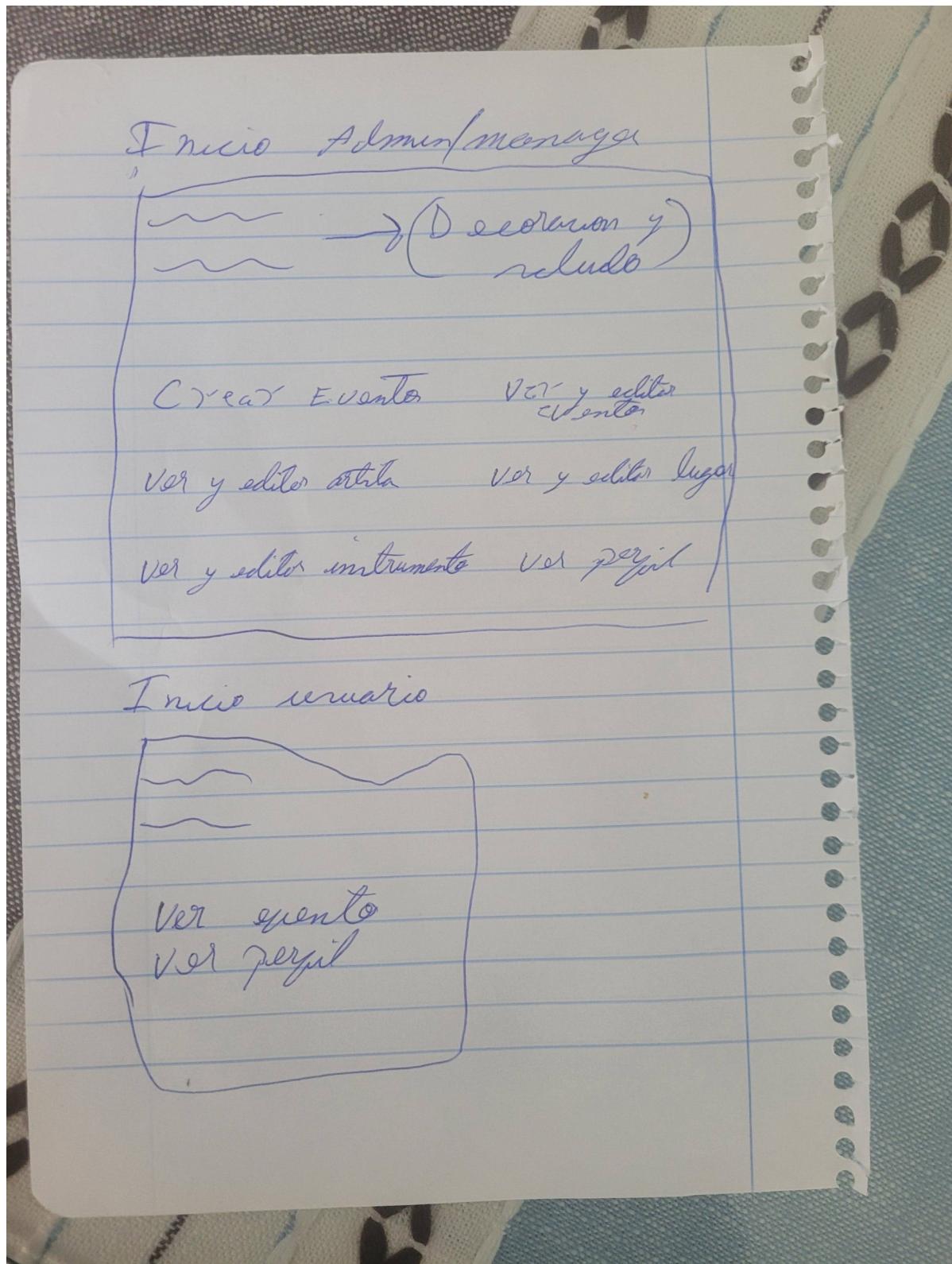
- Código modular para facilitar futuras actualizaciones.
- Registro de errores para depuración y mejoras continuas.

4.1.4. INTERFACES DE USUARIO

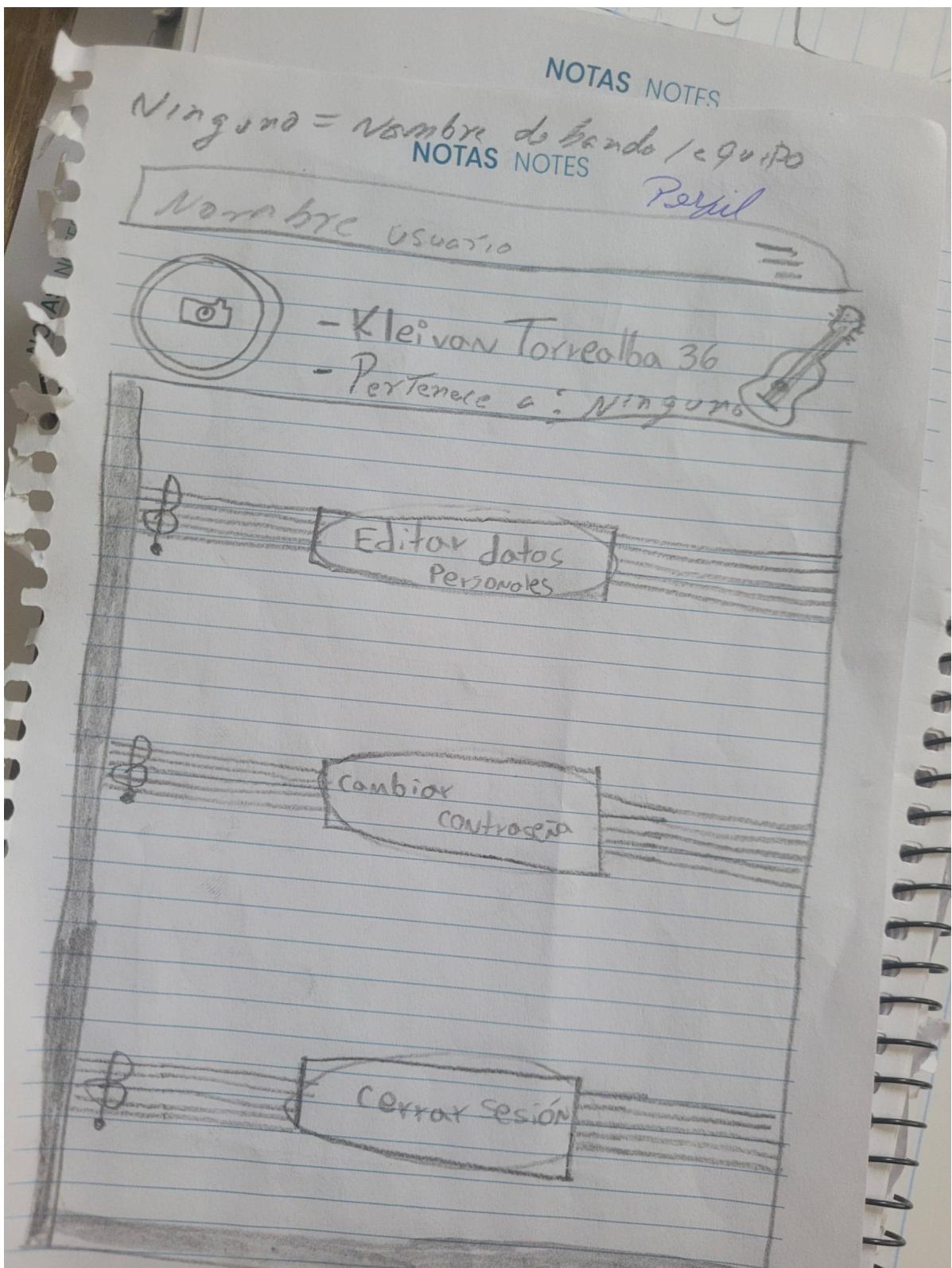
Pantalla de login



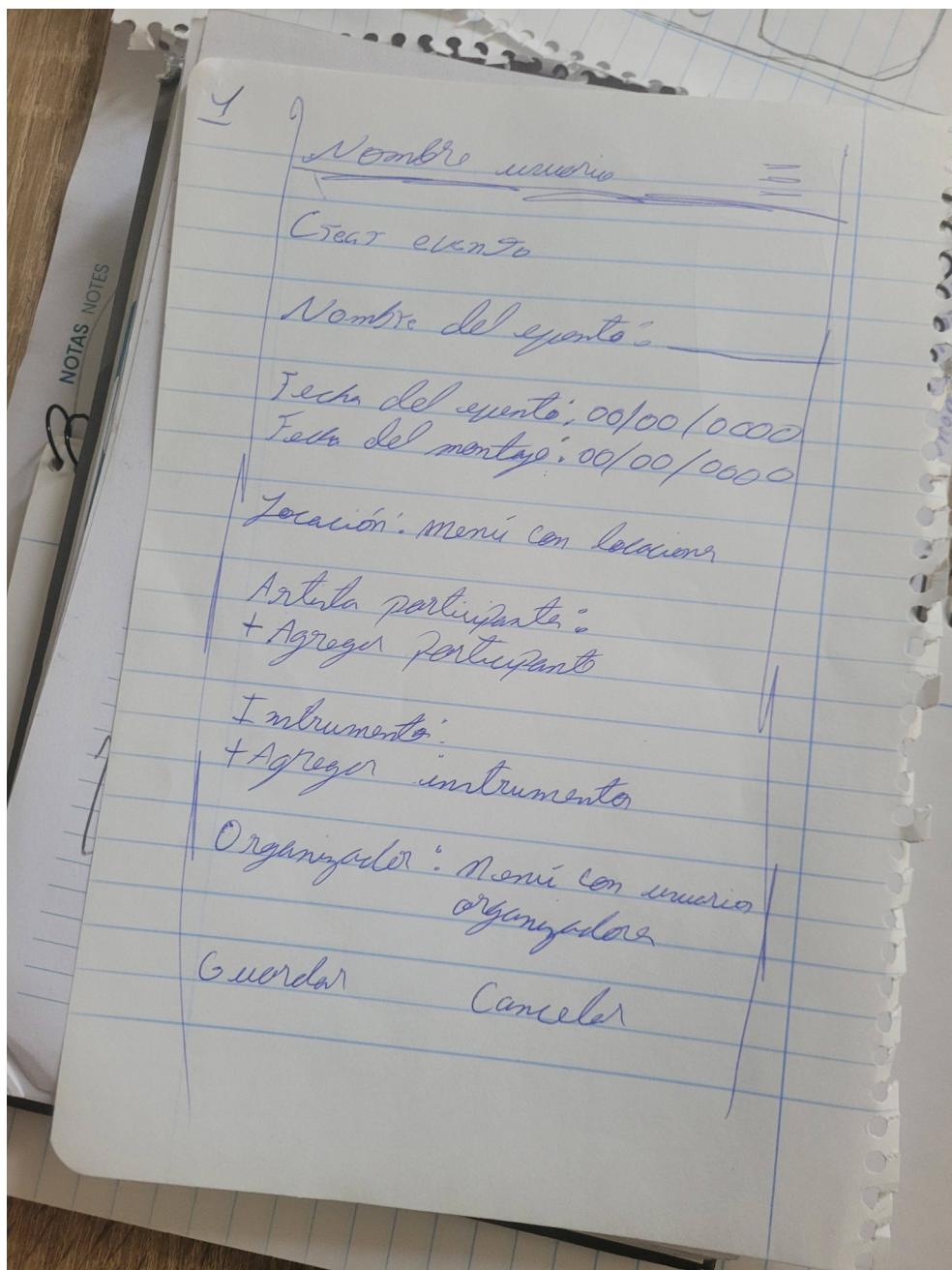
Pantalla de home (para el administrador/manager y para el usuario artista)



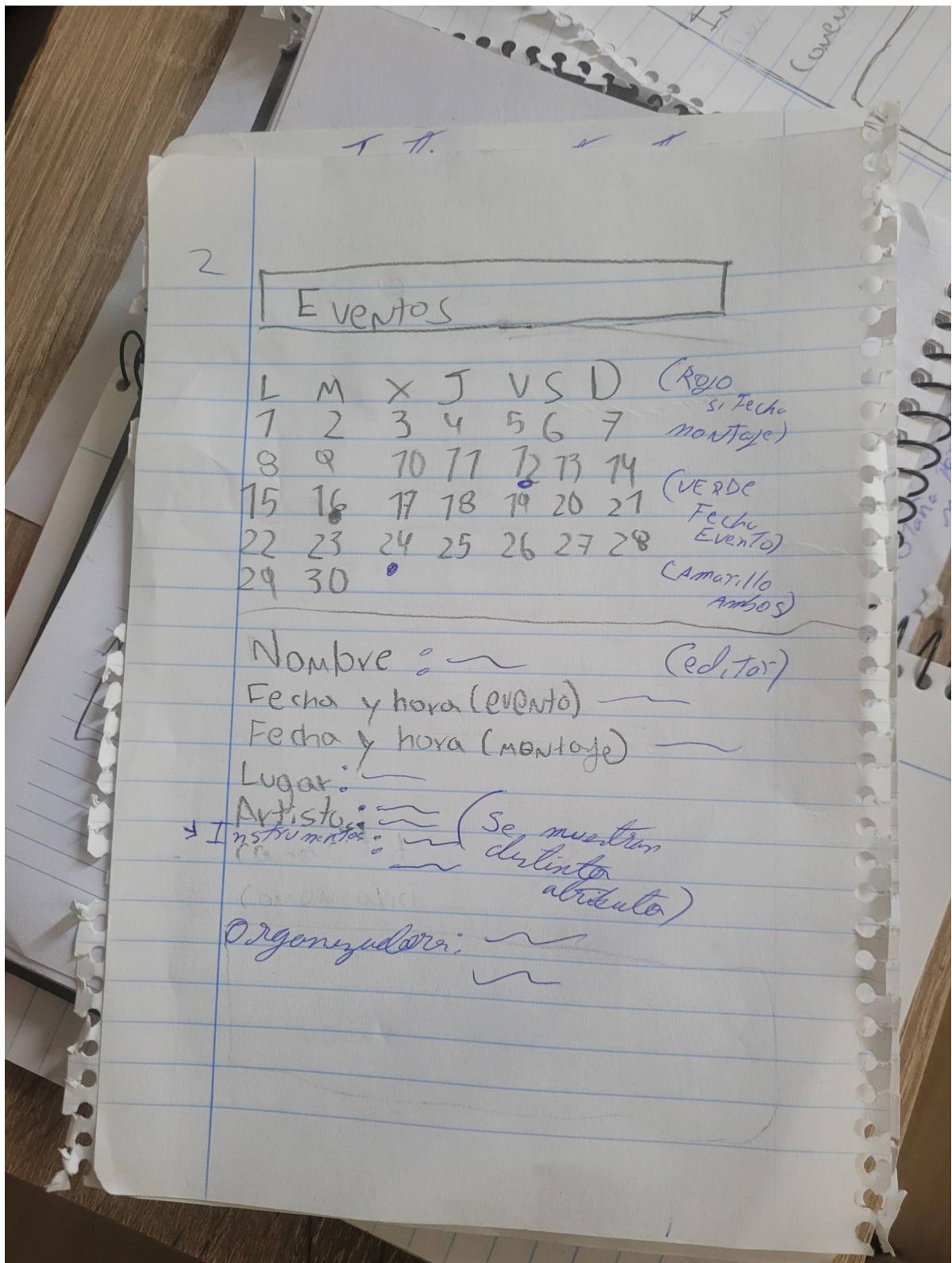
Pantalla de ver el perfil



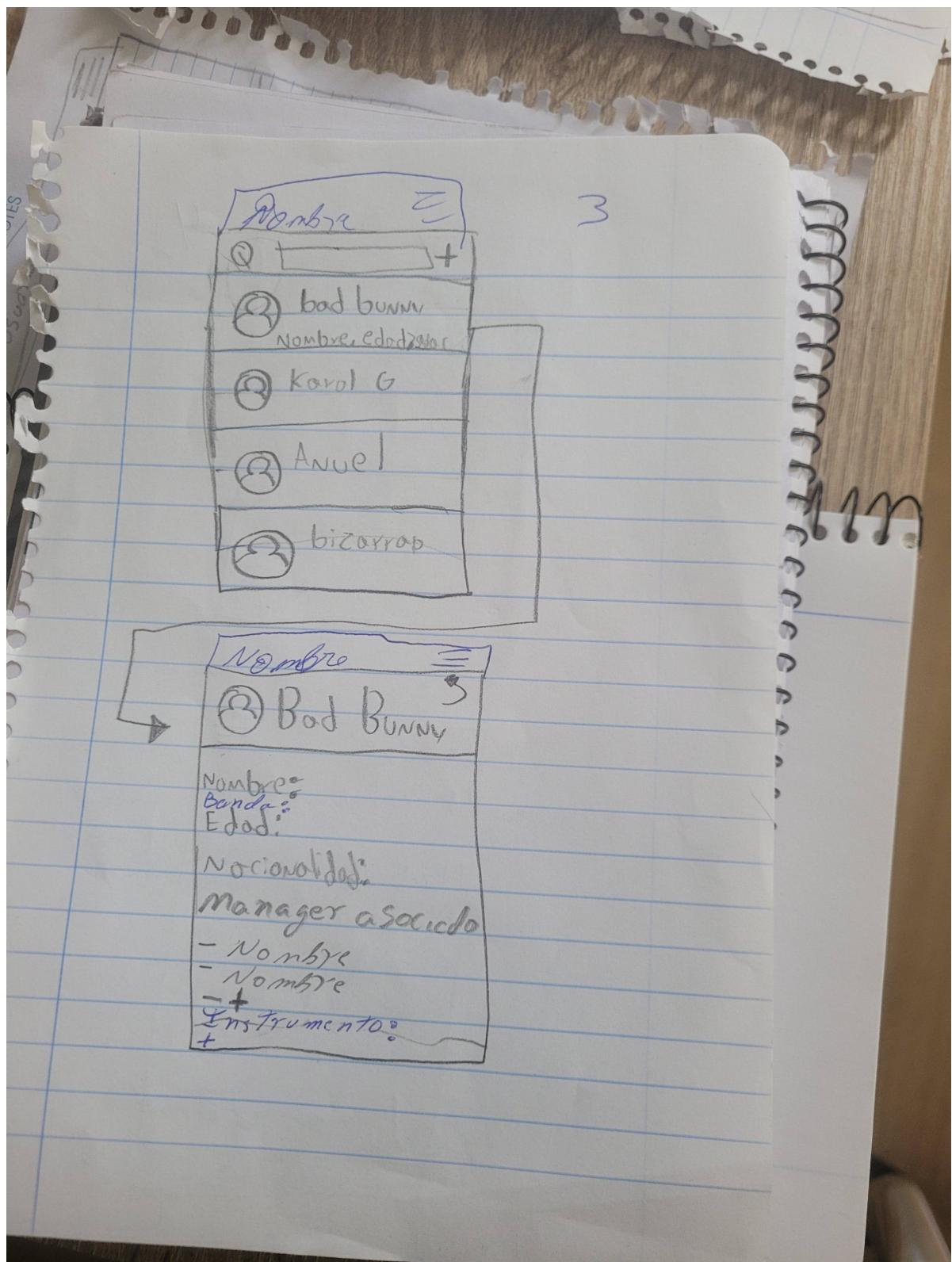
Pantalla de creación de evento



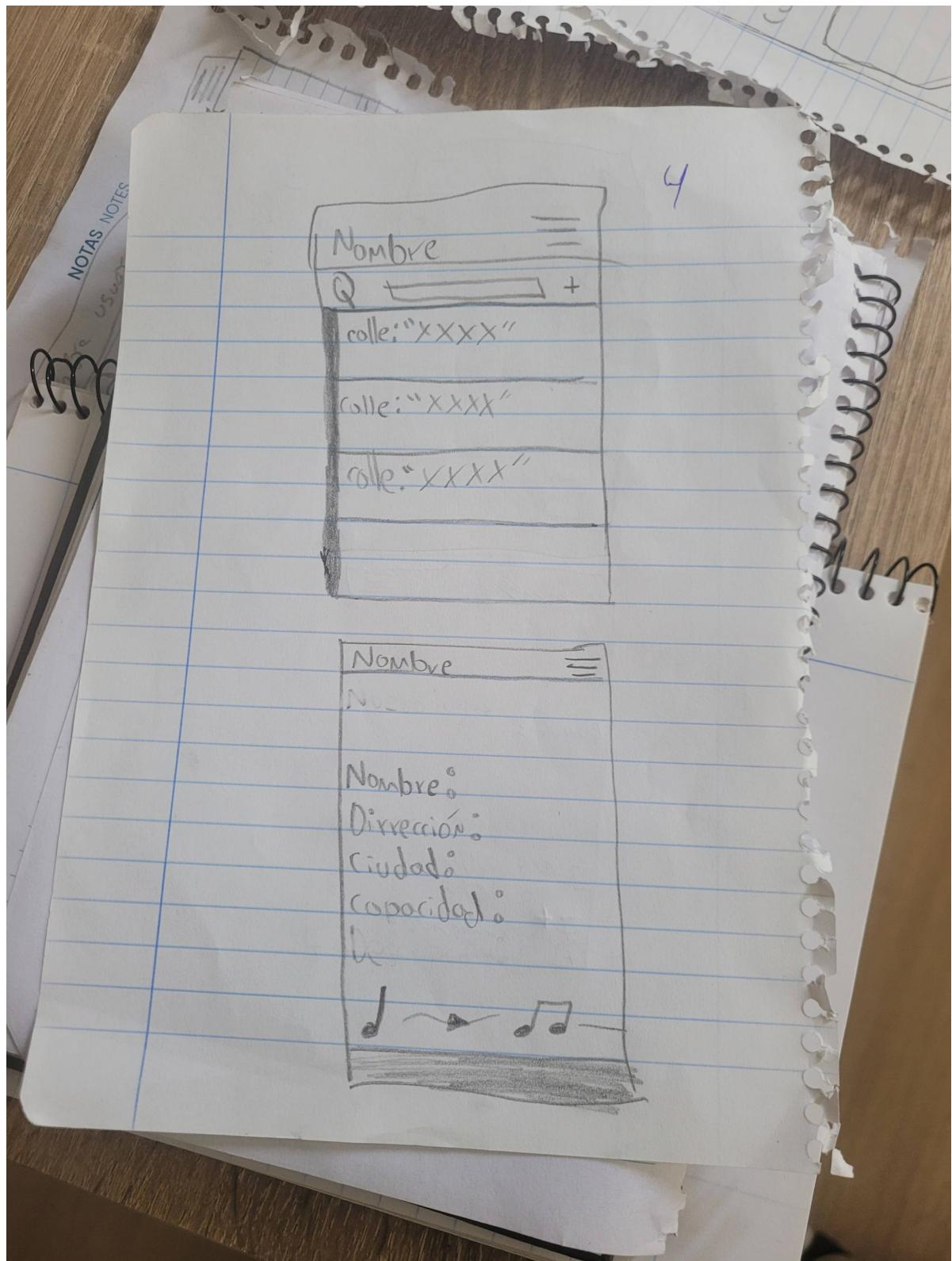
Pantalla de mostrar y edición de eventos



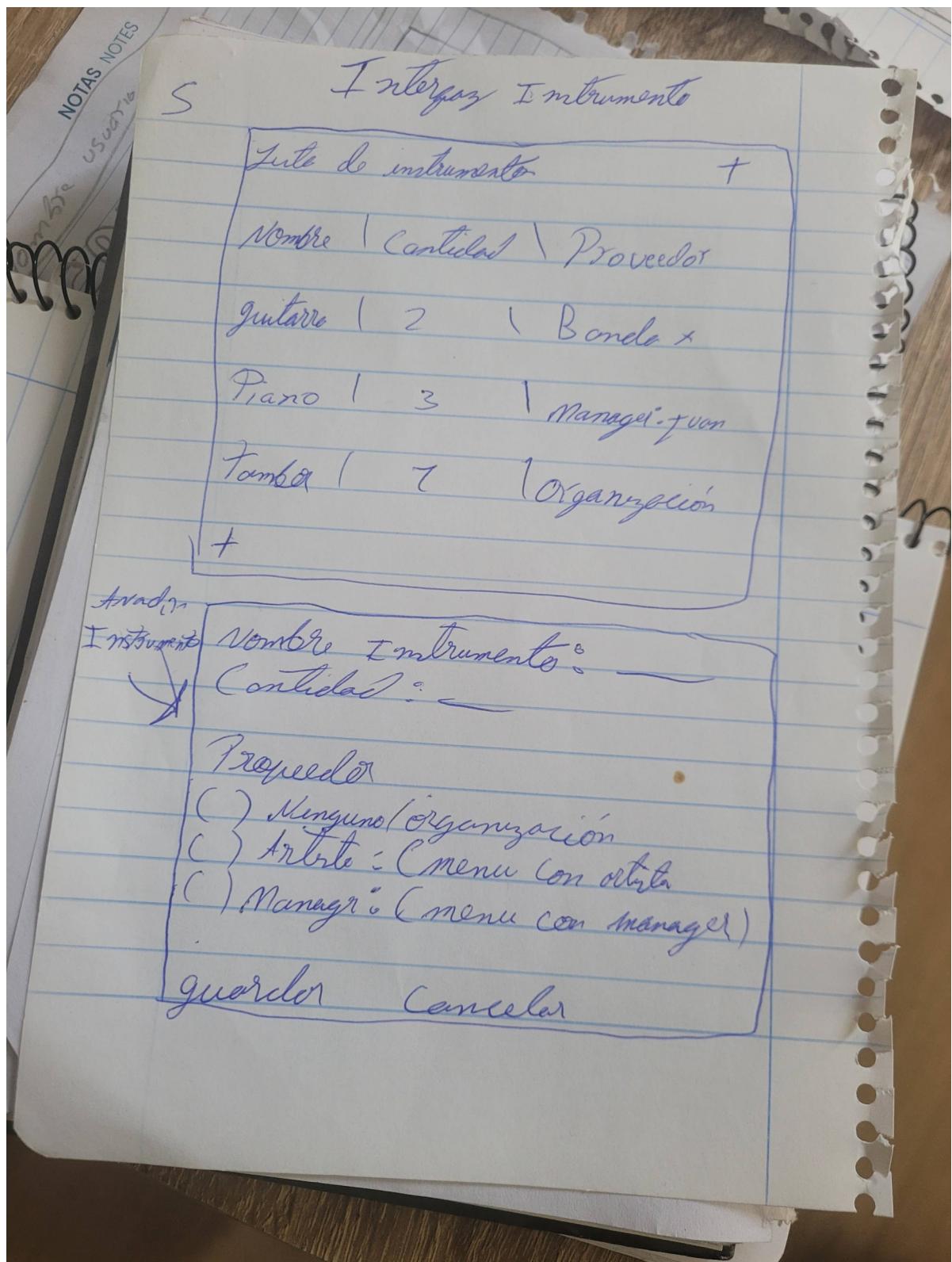
Pantalla de ver los artistas, editarlos y añadir nuevos



Pantalla de ver las localizaciones y añadir localizaciones



Pantalla de instrumentos y añadir nuevos instrumentos



4.1.5. INTERFACES DE COMUNICACIÓN

Incluir en caso de que se utilice cualquier interface de comunicación con otro sistema o dispositivo como redes, dispositivos remotos, API, etc. Si no se usan se puede suprimir este epígrafe.

4.2. RESTRICCIONES DE DISEÑO

Estas restricciones representan limitaciones y condiciones que los desarrolladores deben tener en cuenta al diseñar y construir este software. Por ejemplo:

- Restricciones sobre el lenguaje o lenguajes a utilizar, metodologías o procesos específicos que se deben seguir durante el desarrollo del software.
- El uso de metodologías ágiles, procesos de desarrollo en cascada, o cumplimiento de estándares específicos.
- Pueden existir restricciones sobre las herramientas de desarrollo que se deben utilizar, como entornos de desarrollo integrados (IDE), sistemas de control de versiones, o herramientas de gestión de proyectos.
- Restricciones hardware, que se refieren a las características o capacidades del hardware en el que se ejecutará el software. Por ejemplo, si el software debe ser compatible con dispositivos de hardware específicos, esas restricciones deben ser consideradas.
- Restricciones de librerías y componentes, que pueden incluir el uso de librerías específicas o componentes de terceros, o restricciones sobre la creación de librerías internas.

5. DISEÑO DE LA BASE DE DATOS

5.1. DESCRIPCIÓN GENERAL

La base de datos elegida ha sido **PostgreSQL**, una bases de datos relacional (RDBMS)

Características y Ventajas

- **Fiabilidad y rendimiento:** Maneja grandes volúmenes de datos sin comprometer la velocidad.
- **Soporte ACID:** Garantiza integridad y seguridad en las transacciones.
- **Escalabilidad y concurrencia:** Permite múltiples accesos simultáneos sin bloqueos.
- **Compatibilidad con .NET MAUI**
- **Código abierto y sin costos de licencia.**

Forma de Uso en el Proyecto

La base de datos estará alojada en un servidor, conectándose con la aplicación a través de una API en C# para manejar eventos, artistas y locaciones.

Motivo de Elección

Se eligió PostgreSQL por su estructura relacional, ideal para gestionar datos interconectados de espectáculos. A diferencia de Firebase, que no maneja bien relaciones complejas, y MySQL, PostgreSQL ofrece mejor concurrencia y optimización para consultas avanzadas.

5.2. MODELO DE DATOS

Se utiliza un modelo relacional usando PostgreSQL. Este sistema de gestión de bases de datos relacional organiza la información en tablas las cuales se componen de columnas y filas. Las entidades que forman parte de esta base de datos son: **usuarios**, **eventos**, **artistas**, **locaciones** y **recursos**.

Las tablas clave incluyen:

1. **Usuarios**: Almacena la información de los usuarios como organizadores, artistas y staff.
2. **Eventos**: Representa los espectáculos musicales, con atributos como nombre, fecha y ubicación.
3. **Artistas**: Contiene detalles de los artistas que participan en los eventos.
4. **Locaciones**: Información sobre las ubicaciones de los eventos.
5. **Instrumentos**: Almacena los instrumentos de los eventos.

Las tablas están **relacionadas** entre sí a través de **claves primarias**, lo que garantiza la integridad y permite la realización de consultas complejas sin perder rendimiento.

Estructura de Tablas y Relaciones

1. **Usuarios** (*id_usuario, nombre, correo, contraseña, rol*)
 - Relacionada con eventos y recursos según su rol (organizador, artista, staff).
2. **Eventos** (*id_evento, nombre, fechaEvento, fechaMontaje, id_locacion, id_artista, id_manager, id_instrumentos*)
 - Relacionada con locaciones y artistas.
3. **Artistas** (*id_artista, nombre, banda, género, generoMusical, nacionalidad, edad, id_manager*)
 - Relacionada con eventos (muchos a muchos)
4. **Manager** (*id_manager, nombre, email, telefono, id_usuario*)
 - Relacionada con Artistas (muchos a muchos)

5. **Locaciones** (*id_locacion, nombre, dirección, capacidad, ciudad*)
 - Relacionada con eventos.
6. **instrumentos** (*id_instrumento, nombre, cantidad, id_artista, id_manager*)
 - Asignados a eventos según disponibilidad.

5.2.1. MODELO ENTIDAD-RELACIÓN (Si tu base de datos es del tipo entidad-relación)

Entidades y Relaciones

1. Usuario

- Atributos: *id_usuario, nombre, correo, contraseña, rol*
- Relación: Un usuario puede ser **organizador, artista o staff**.

2. Evento

- Atributos: *id_evento, nombre, fecha, id_locacion*
- Relación: Un evento se asocia con una **locación** (1 evento a 1 locación).

3. Artista

- Atributos: *id_artista, nombre, género, id_usuario*
- Relación: Un artista puede participar en varios **eventos** y un evento puede tener varios artistas (**muchos a muchos**).

4. Locación

- Atributos: *id_locacion, nombre, dirección, capacidad*
- Relación: Una locación puede tener varios **eventos**.

5. Instrumento

- Atributos: *id_recurso, nombre, cantidad, id_evento*
- Relación: Un instrumento se asigna a un **evento** (1 a muchos).

5.3. PASO A ESTRUCTURAS DE ALMACENAMIENTO

En esta sección, debes describir cómo se traducen las entidades o documentos y relaciones definidas en el modelo conceptual representado en los epígrafes 5.2.1 y 5.2.2, a las estructuras específicas del sistema de base de datos que se está utilizando.

Dependiendo del tipo de base de datos, nos centraremos solo en estas estructuras:

- **Bases de datos relacionales (epígrafe 5.2.1):** Las entidades se convierten en tablas, donde los atributos de las entidades se representan como columnas, y las relaciones entre entidades se traducen a claves foráneas.

Ejemplo: La entidad "Cliente" se convierte en una tabla con columnas como IdCliente, nombre, y dirección. Las relaciones entre entidades, como "Cliente" y "Pedido", pueden representarse mediante una clave foránea en la tabla "Pedidos" que apunte a la tabla "Clientes".

Tabla Clientes (**idCliente**, nombre, dirección),

Tabla Pedidos (**idPedido**,idCliente)

* La convención elegida para diferenciar cual es la clave principal y foránea debe especificarse e indicarse de forma clara. En este caso se ha escogido negrita para clave principal y subrayada para foránea(puedes escoger otra).

Ver el siguiente link, para saber cómo aplicar las reglas para el paso de entidades a tablas en **modelo entidad-relación**.

https://josejuansanchez.org/bd/unidad-03-teoria/resources/resumen_paso_a_tablas.pdf

- **Bases de datos documentales (NoSQL) (epígrafe 5.2.2):** Las entidades se traducen en documentos. Los documentos pueden tener una estructura flexible, donde los atributos pueden variar de un documento a otro. Las relaciones se suelen representar de forma anidada, es decir, un documento puede contener subdocumentos o listas de elementos relacionados.

Ejemplo: La entidad "Cliente" se convierte en un documento JSON que puede tener una estructura como:

Si esta parte (NoSQL) ya la incluiste en el epígrafe 5.2.2. entonces este **epígrafe 5.3. Paso a estructuras de almacenamiento lo puedes eliminar

Ej:json

```
{
  "ID": "123",
  "nombre": "Juan",
  "dirección": "Calle 123",
  "pedidos": [
    {"ID": "001", "Fecha": "2024-01-01", "Total": 200},
    {"ID": "002", "Fecha": "2024-02-01", "Total": 150}
  ]
}
```

}

5.4. DIAGRAMA DE TABLAS DE LA BASE DE DATOS

Aquí tienes que incluir el diagrama las tablas que has incluido en tu gestor de base de datos (modelo relacional). Tienes unos links para saber cómo hacerlo.

Una vez esté generado guarda o captura el diagrama e insértalo en este epígrafe. Puedes utilizar phpMyAdmin , Workbench o cualquier otra aplicación.

Diagrama con phpMyAdmin

<https://www.youtube.com/watch?v=emIF3wfODkE>

Diagrama con Workbench

<https://www.youtube.com/watch?v=Q18Zb31Vc5M>

**Si tu modelo es NoSQL entonces busca una forma de representar la base de datos de tipo documental.

5.5. DICCIONARIO DE DATOS

Proporciona información más específica sobre cada elemento de datos y asociada al lenguaje de programación utilizado. incluyendo definiciones detalladas, características y metadatos, se debe usar una tabla que contenga los siguientes datos:

Ejemplo a seguir: de diccionario de datos para una **tabla de ejemplo llamada Alumnos**.

Incluye tantos diccionarios de datos como tablas tengas en tu base de datos

COLUMN A	KEY	TIPO	NULO	TIPO DATO	LONG	DESCRIPCIÓN
idAlu	SI	AUTOINCREMENT	NO	INT	4	Número identificador de alumno.
dniAlu	NO			VARCHAR	9	Cadena que identifica el documento nacional de identidad del alumno

5.6. SCRIPT DE CREACIÓN E INSERCIÓN DE LA BASE DE DATOS

Crea un script de **creación** de las tablas y otro de **inserción** de datos en dichas tablas en lenguaje SQL. En el script de inserción como mínimo debes incluir 5 registros por tabla. Por ejemplo si la tabla se llama Clientes entonces por lo menos se deben introducir 5 clientes en ella mediante la forma que se indica aquí.

Para generar un script completo de tu base de datos tienes estos links...

Con Workbench <https://www.youtube.com/watch?app=desktop&v=tWwwC7qdP-Y>

Y también realiza otro que inserte datos en la BD.

<https://es.khanacademy.org/computing/computer-programming/sql/sql-basics-pt/creating-a-table-and-inserting-data>

Agrega el resultado en este epígrafe e identifícalos con un título. (Borra aquella información generada automáticamente y que no sea relevante, solo sentencias de creación e inserción)

6. VISIÓN GENERAL DEL CÓDIGO

Explica la estructura de carpetas y directorios de tu aplicación. Puedes insertar pequeños fragmentos de parte del código que creas necesarios e importantes. Pero **no** se debe incluir páginas completas llenas de código fuente, recuerda solo pequeños fragmentos que tengan relevancia.

7. PRUEBAS Y VALIDACIONES

Explica que son las pruebas y expón los tipos de pruebas y tipos de validaciones realizadas para comprobar el correcto funcionamiento del producto desarrollado. Puedes incluir imágenes (siempre que no ocupen mas de la mitad de la página sino deben ir a apartado Anexos y ser referenciadas en este epígrafe).

8. DIFICULTADES DETECTADAS Y POSIBLES MEJORAS

Describe los problemas encontrados. Y en cuanto a las mejoras futuras plantea aquellas funcionalidades que se podrían añadir al proyecto y que en la actualidad no se han podido implementar por falta de tiempo u otros factores.

9. TEMPORALIZACIÓN DEL PROYECTO

Aquí, proporcionaras la estimación del tiempo que ha tomado completar las tareas que han sido necesarias para realizar el proyecto. Puedes utilizar un cronograma, una línea de tiempo, una tabla que muestre fases o tareas clave y las fechas de entrega.

En este link puedes ver imágenes de tipos de gráficas de tiempo.

<https://talentohumanocali.jimdofree.com/cronograma-de-actividades-y-flujograma/>

10. DESPLIEGUE DE LA APLICACIÓN

Se debe indicar como se ha desplegado la aplicación, tecnologías utilizadas, Hosting, etc. O en su defecto como se accede a la aplicación de forma local.

11. MANUAL DE USUARIO

Un manual de usuario es una guía que proporciona información detallada sobre cómo usar un producto, sistema, software o servicio, pero será un manual orientado al usuario. Se explicará con detalle y de forma estructurada cómo será el proceso de instalación o puesta en marcha por parte del usuario final, la dirección web(en caso de estar alojado en un servidor o hosting) y/o lugar donde se encuentran los archivos (los archivos se entregan en la 4º entrega en fichero comprimido).

En un manual de usuario, es importante explicar los elementos clave que permitan a los usuarios comprender y aprovechar al máximo el producto o servicio.

- Breve descripción del producto y servicio, propósito y utilidad y público al que va dirigido.
- Instrucciones de instalación o ejecución de la página web, bien claras.
- Debe indicar los requerimientos mínimos de software y hardware, etc. que debe tener el usuario antes de la instalación o ejecución, como acceder a la aplicación, y principales funcionalidades que el usuario puede realizar con la misma.
- Interfaz de usuario. (Si se ha explicado antes en apartado 'Diseño de la interfaz' en este punto solo hacer un breve resumen o explicar un caso de ejemplo para que el usuario entienda el funcionamiento, ya no haría falta volver a explicar de nuevo cada interfaz), explica cómo navegar por la interfaz y cómo acceder a las funcionalidades.

12. MANTENIMIENTO

Explica cómo mantener el producto o servicio en buen estado, cómo aplicar actualizaciones y el rol de la persona encargada de hacerlo en un futuro.

13. CONCLUSIONES

Evaluación de los resultados investigados en el proyecto, cumplimiento de objetivos marcados, diferentes tipos de dificultad encontrada y como se ha superado, etc.

14. GLOSARIO DE SIGLAS Y ABREVIATURAS

Indicar en forma de tabla o similar las siglas o abreviaturas utilizadas a lo largo del proyecto.

Ejemplo a seguir:

Acrónimo	Significa...
FIFO	Primero en entrar, primero en salir
PHP	Preprocesador de hipertexto
HTML	Lenguaje de marcado de hipertexto
JS	JavaScript
CSS	Hojas de estilo en cascada

15. BIBLIOGRAFÍA

Plantea aquí los recursos externos, libros, documentos técnicos u otros materiales de referencia utilizados en el desarrollo del proyecto, menciona cuáles son. Utiliza las normas APA.

En plataforma hay un archivo sobre como utilizar las normas APA

Fíjate en este link para crear estas referencias desde un menú en Word de forma rápida

<https://www.youtube.com/watch?v=VgR1bPo9uTU>

**Recibirás también un documento pdf con explicaciones sobre normas APA

16. ANEXOS

En este apartado, se incluirán aquellos recursos que ocupan una hoja, casi una hoja o son de gran tamaño, como por ejemplo gráficos, tablas, infografías, imágenes, entre otros datos que se hayan necesitado incorporar a esta planificación de proyecto.

Cada imagen debe tener un título y número, que debe figurar en el índice automático que se incluye al principio del documento debajo del epígrafe Anexos.

En este documento ya está creado un índice automático que solo tendrás que actualizar con un clic de ratón cada vez que añadas nuevo contenido, pero si no sabes cómo trabajar con este tipo de índice, mira este link

<https://www.youtube.com/watch?v=kvomMdjAdKk>

****TODO LO QUE ESTÁ SUBRAYADO EN COLOR AMARILLO O AZUL ES PARA ELIMINAR Y RELLENAR CON INFORMACIÓN SOBRE EL PROYECTO, SOLO SIRVE PARA INDICAR QUE HACER EN DICHO APARTADO****

Número de páginas mínimo 30 y máximo 50 se excluyen los Anexos.

Tutora: María del Carmen Buenestado Fernández