



Área Académica de Ingeniería en Computadores

Bases de Datos

Estudiantes:

Meibel Ceciliano Picado

Sebastián Hidalgo Vargas

Valeria Morales Alvarado

Juan Daniel Rodríguez Montero

Estefany Villalta Segura

Profesor: Marco Rivera Meneses

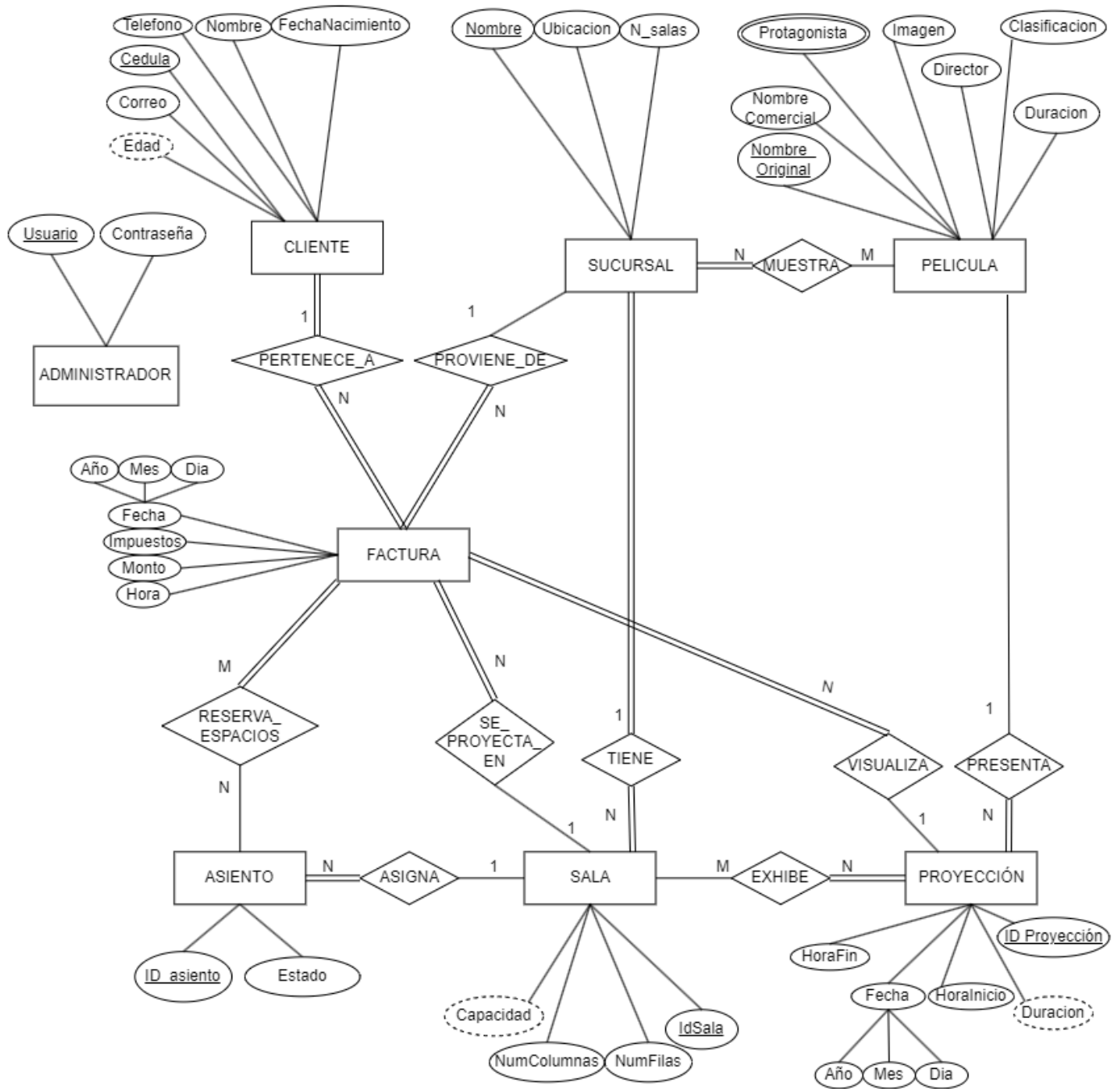
**Prototipo**

**CineTEC**

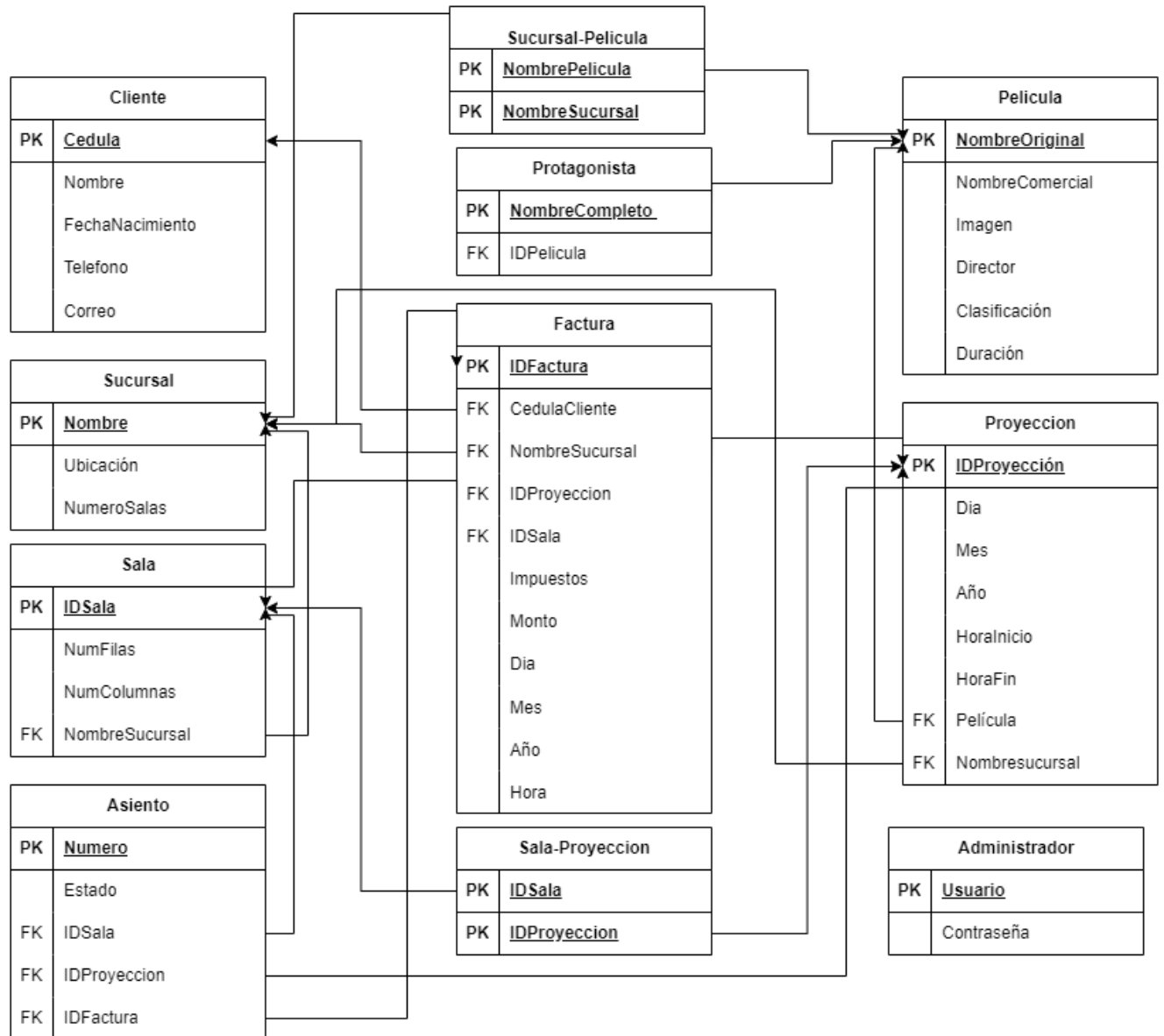
## **Tabla de Contenidos**

Modelo Conceptual .....	3
Modelo Relacional .....	4
Descripción de las estructuras de datos desarrolladas .....	5
Descripción detallada de la arquitectura desarrollada .....	8
Diagrama de Clases.....	9
Problemas conocidos: .....	10
Conclusiones .....	11
Recomendaciones .....	11
Plan de Proyecto .....	12
Bibliografía .....	31

## Modelo Conceptual



## Modelo Relacional



## Descripción de las estructuras de datos desarrolladas

Las estructuras de datos utilizadas en este prototipo fueron principalmente tablas. Mediante el formato de texto json se logra simular una base de datos que permite almacenar la información de cada entidad. Cada json contenía una lista, se elige esta forma ya que es posible tener una colección ordenada de los elementos, siendo estos del mismo o diferente tipo, de forma simular a POO en la lista se guardaban los objetos generados al instanciar las clases “entidades”, junto a sus atributos. Para el desarrollo de este prototipo fueron utilizadas las siguientes tablas de entidades

```
[
  {
    "IDSala": 1,
    "NumFilas": 10,
    "NumColumnas": 15,
    "NombreSucursal": "Paseo Metropoli"
  },
  {
    "IDSala": 2,
    "NumFilas": 15,
    "NumColumnas": 20,
    "NombreSucursal": "Paseo Metropoli"
  },
  {
    "IDSala": 3,
    "NumFilas": 14,
    "NumColumnas": 18,
    "NombreSucursal": "Terramall"
  },
  {
    "IDSala": 4,
    "NumFilas": 8,
    "NumColumnas": 10,
    "NombreSucursal": "Paseo Las Flores"
  }
]
```

Figura 1: Datos de tabla “Salas” guardados mediante una lista en un archivo json

A continuación, se procede a describir cada una de las tablas:

**Cliente:** esta tabla contiene la información de cada usuario que opta por el servicio de CineTEC, se almacena para cada uno su Cedula la cual será su PK, Nombre, Teléfono, Fecha de Nacimiento que es un atributo almacenado y es de tipo DateTime, finalmente su correo donde será enviada posteriormente la factura.

**Sala:** se crea un IDSala que será su identificador, por lo tanto, este campo no puede ser nulo, tiene un atributo de numero de filas y otro de numero de columnas, para saber el tamaño de cada una, se le agrego una FK de NombreSursal para que sea posible determinar en que lugar se encuentra esta específicamente.

**Sucursales:** La sucursal va a tener un Nombre que será su llave primaria, una Ubicación y un número de salas, este último atributo si podría ser nulo y es de tipo int.

**Películas:** cada película cuenta con un NombreOriginal que será su llave primaria, un NombreComercial, una imagen asociada, un director, clasificación de todos los mencionados anteriormente serán de tipo string, solamente duración que será un entero y nunca nulo, pues una película no puede durar 0 segundos.

**Protagonistas:** protagonistas es un atributo multivaluado de Películas, al ser un atributo de este tipo debía ser colocado en una tabla aparte, este tiene un atributo de tipo string que es NombreCompleto y es la PK, también contiene una FK de NombreOriginal que pertenece a la tabla películas.

**Proyecciones:** cada proyección tiene un ID que es su llave primaria, un día, mes y año, de tipo int, que no pueden ser nulos. Tiene una fecha de inicio y finalización de tipo DateTime que tampoco puede ser vacio, además lleva 2 FK pertenecientes a NombreSucursal y Película.

**Asientos:** tiene un numero que es su llave primaria, además de un atributo estado que define su disponibilidad, al inicio todos se encuentran disponibles se almacenada su estado ocupado al generarse la factura, cada asiento debe ir asociado entonces a una factura, una sala, y una proyección, por lo anterior se agregan FK de IDSala, IDProyección y IDFactura.

**Factura:** una factura generada debe ser única para cada cliente, es por esto que contiene un ID como PK, tiene una CedulaCliente, un monto, impuesto y una fecha, de ellos ninguno null. Cada factura debe vincularse con la información de sucursal, sala, proyección, razón por la que se agregan esos FK a la tabla.

Finalmente, fue necesario realizar dos tablas intermedias que hacen referencia a las relaciones N:M

**SucursalPelicula:** esta contiene dos llaves primarias, la PK NombrePelicula vinculada la entidad Película, y la PK NombreSucursal vinculada a la entidad Sucursal, de esta manera al usuario seleccionar una sucursal se le podrán desplegar todas las películas que están disponibles en ella.

**SalaProyeccion:** de manera similar a la tabla anterior, SucursalProyeccion tiene dos llaves primarias, la PK de Sala que es IDSala, y la PK de proyección que es IDProyeccion, fue necesario hacer esto ya que una proyección puede verse en diferentes salas a la vez, incluso en el mismo horario.

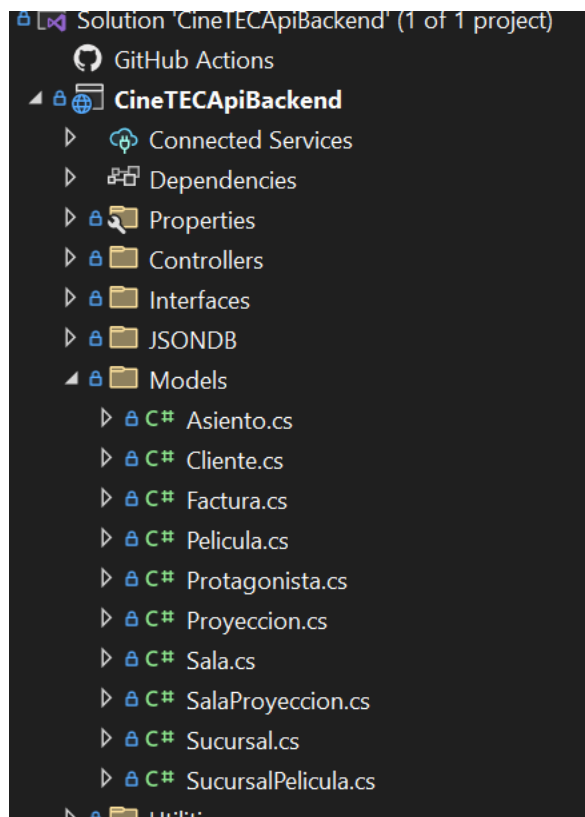
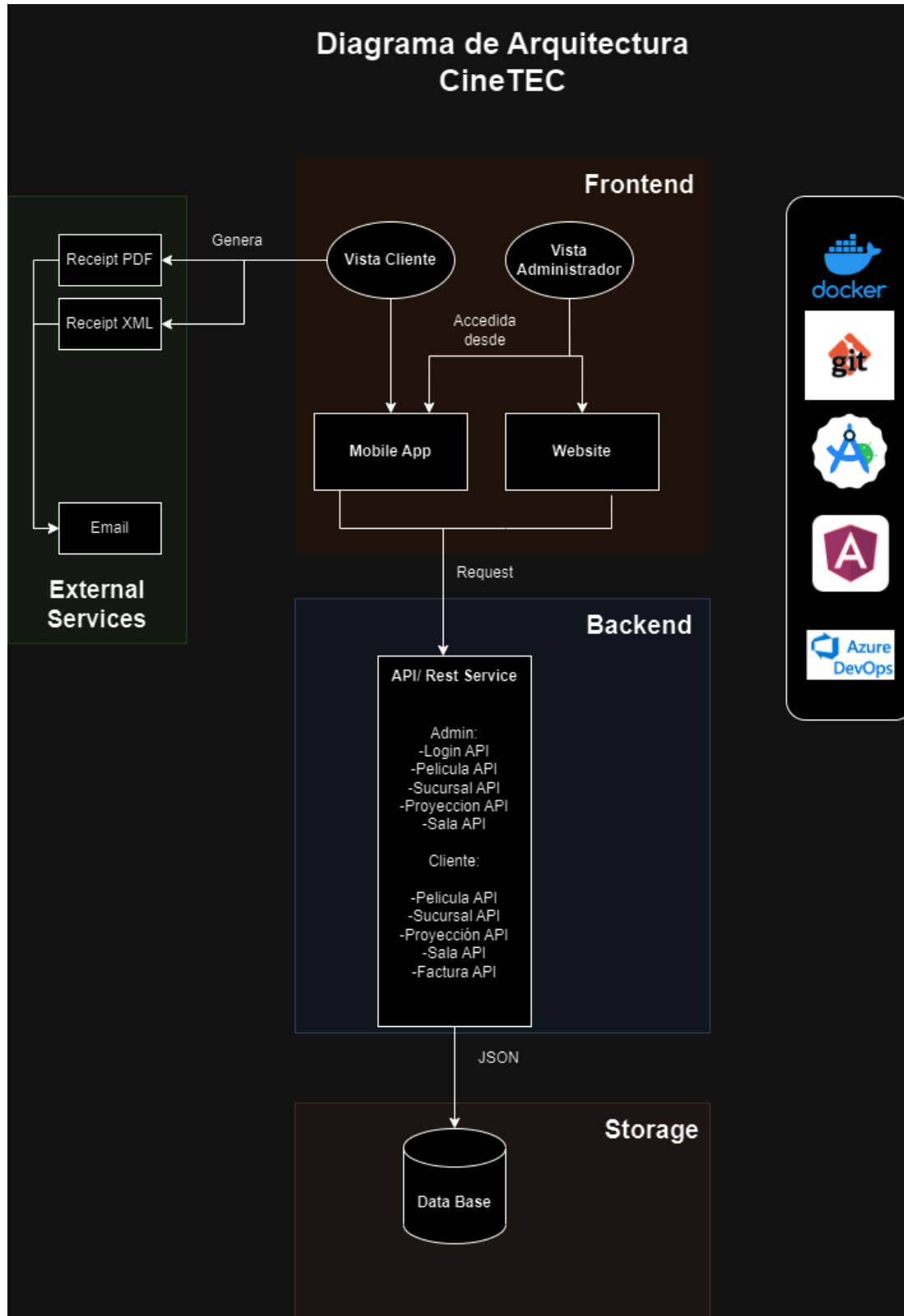


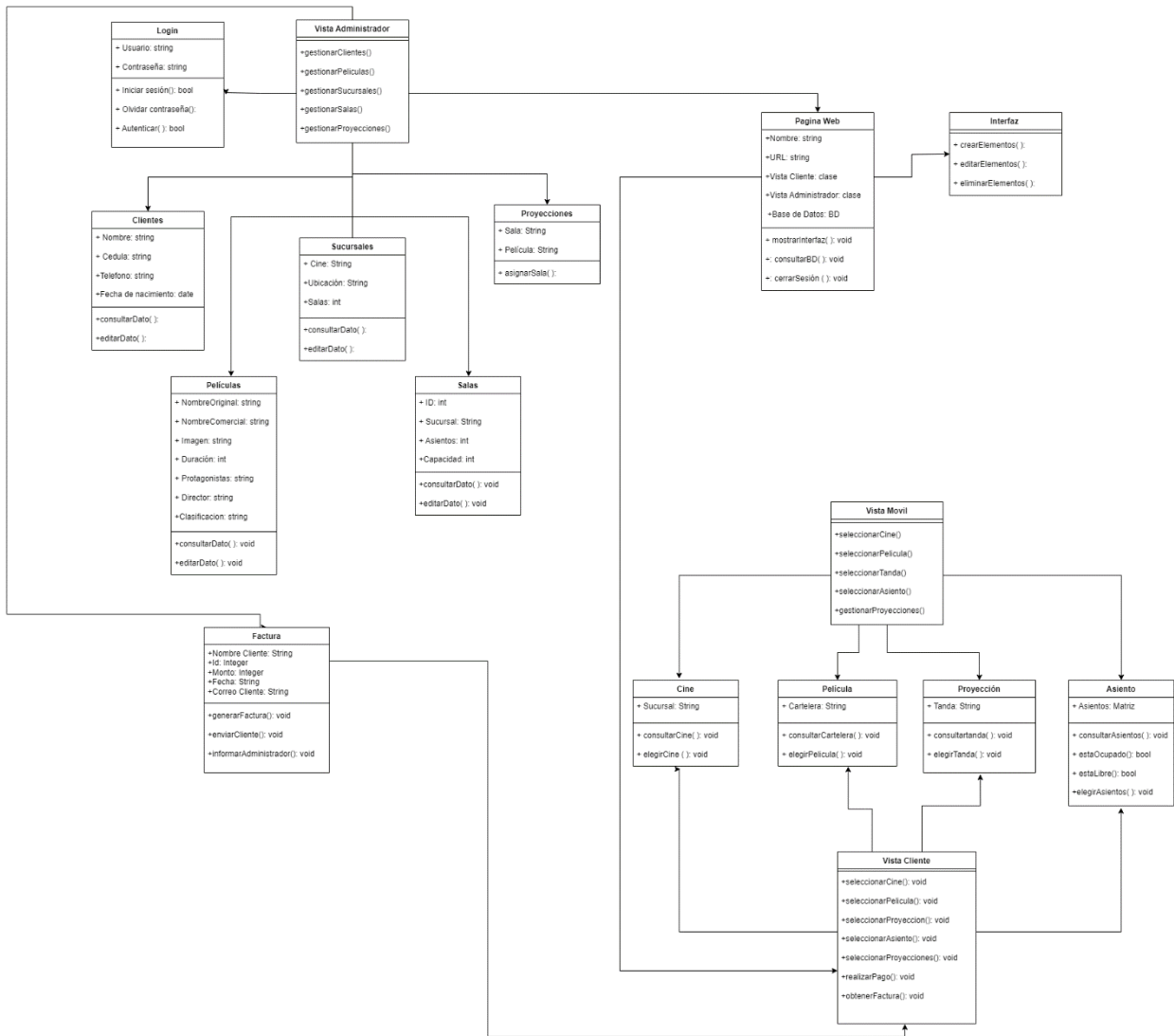
Figura 2: Tablas creadas para el prototipo CineTEC

## Descripción detallada de la arquitectura desarrollada





## Diagrama de Clases



Se puede apreciar mejor en el siguiente [enlace](#)

**Descripción:** Se decide representar el proyecto y sus módulos de esta manera y aprovechar la capacidad de representar las entidades necesarias mediante abstracción. De igual manera, se buscó organizar el código en unidades coherentes y lo más independientes posibles, para facilitar su mantenimiento y modificación a futuro. También así se puede tener una clara visibilidad de los atributos y métodos de cada entidad. Otro aspecto importante es que proporciona la capacidad de facilitar el entendimiento del proyecto y la colaboración de los integrantes del equipo.

## **Problemas conocidos:**

### **Problemas solucionados**

1. Se había presentado un problema al inicio al no poder abrir el Swagger esto se solucionó rápidamente al modificar la configuración de CORS para que de esta manera el servidor devolviera los encabezados http y fuera posible probar los diferentes endpoints.
2. Los asientos fueron un inconveniente, pues al principio tanto en el modelo conceptual, como en el relacional habían sido colocados como un atributo multivaluado, sin embargo, al trabajar en el código se decide que es mejor asignarle a este un estado, el cual se ocupará cuando el usuario reserve su entrada, de manera contraria sigue disponible. Esto hizo más efectivo su manejo.
3. Se tenía una tabla de relación intermedia llamada Factura-Asiento, la cual fue eliminada al dar complicaciones al obtener los asientos por factura, se solucionó agregando una foreign key (FK) de ID\_factura a la tabla asiento.
4. En el FrontEnd se tuvieron algunos problemas con el mal modelado de la respuesta del api, enrutamiento, y sintaxis del framework angular, estos fueron solucionados haciendo reuniones grupales pequeñas donde se revisaba la estructura del código y se detectaban los errores, los cuales pudieron corregirse de manera exitosa en poco tiempo.
5. Se le agregó a proyecciones una FK hacia nombre sucursal para poder filtrar todas las proyecciones de cada película por sucursal, pues al inicio fue complicado poder obtener este dato.

### **Problemas sin solucionar:**

1. La aplicación móvil no fue completada en el prototipo, esto debido a un mal manejo del tiempo, se considera hubo mala comunicación, pues en caso de haberse informado estos problemas con anticipación se hubiesen podido distribuir entre los miembros y atender oportunamente.

## **Conclusiones**

1. Una etapa crucial y a la cual no se le da la atención necesaria es la confección de los modelos conceptuales y relaciones de un caso, pues al momento de programar pueden presentarse problemas si estos no fueron realizados correctamente, en nuestro caso se tuvo que reestructurar varias relaciones y llaves foráneas que generaban inconvenientes al obtener y enlazar los datos.
2. Trabajar un diseño modular en angular es lo ideal para mantener una estructura clara, separar los componentes y dependencias vuelve más sencillo el proceso de añadir, modificar o eliminar funcionalidades.
3. Es necesario familiarizarse y comprender bien el flujo de datos que va desde la BD hasta la interfaz de usuario, fue útil realizar una especie de boceto donde se colocaron las diferentes paginas web y los servicios que debían solicitarse en cada una de ellas, de esta manera es más sencillo saber los endpoints que deben desarrollarse en el API REST.

## **Recomendaciones**

1. Angular al ser un framework muy popular y utilizado por desarrolladores, cuenta con gran cantidad de recursos, documentación y soluciones a problemas comunes, se recomienda consultar su página en caso de presentarse posibles errores durante su utilización, esto le ayudará a entender y corregir los problemas de una mejor manera.
2. Se sugiere a los desarrolladores de páginas web la utilización de Bootstrap, este entorno cuenta con un conjunto de componentes y plantillas modificables que le permitirán crear su interfaz de usuario de una forma más eficaz y sencilla, acelerando sus tiempos de desarrollo y brindándole al usuario una plataforma interactiva.
3. Aproveche la capacidad de Swagger para generar documentación interactiva automática a partir de la especificación OpenAPI. Usted será capaz de poner a prueba la lógica implementada en los endpoints directamente desde la documentación. Realizar la validación de solicitudes y respuestas desde esta herramienta le permite asegurarse que los métodos HTTP y parámetros utilizados son correctos.

## **Plan de Proyecto**

### **METAS:**

En esta sección se describirán cada uno de los entregables que deben completar los estudiantes para cumplir con lo solicitado en el proyecto

**1. Tarea Corta 1:** Se busca elaborar el modelo conceptual del caso de estudio “CineTEC” poniendo en práctica la notación de Chen para describir las entidades, atributos y relaciones de la base de datos. Por su parte, se lleva a cabo el diagrama de clases para describir el contexto del problema teniendo en cuenta la página web, aplicación móvil y base de datos.

Adicionalmente, debe realizarse el plan de proyecto, donde se planificarán las diferentes tareas y forma de trabajo. Todo esto debe estar listo para ser entregado el 30 de agosto

**2. Tarea Corta 2:** Una vez finalizada la tarea corta 1 y contando con el modelo conceptual, puede iniciarse con el mapeo para generar el modelo relacional. Además de generar el diagrama de arquitectura para implementar el prototipo de la tarea corta. Este entregable debe ser elaborado como fecha límite entre el 6 y 13 de septiembre.

**3. Resumen Ejecutivo Avance 1:** Como una estrategia de transparencia y comunicación se realizará cada cierto tiempo entrega de avances al profesor, como un control e identificación temprana de problemas, de esta manera será posible tener un tiempo de reacción oportuno ante cualquier desafío que se presente. En este primer avance debe evidenciarse las debilidades y fortalezas que se han ido detectando durante la tarea corta 1 y 2. Tiempo de entrega 6 de septiembre.

**4. Resumen Ejecutivo Avance 2:** Para este punto ya se debería tener un 75% del prototipo realizado, el reporte de avance 1 ha ayudado a mejorar la planificación futura, pues se detecta que ciertas tareas tardan más de lo esperado teniendo una visión más clara de cómo administrar el tiempo. Se debería entregar el 13 septiembre

**5. Prototipo Completo:** Entregar el código fuente terminado que incluye una página web con una vista cliente y administrador, funcionando junto a la base de datos y aplicación móvil. Deben estar

listos ciertos documentos como el manual de usuario, documentación técnica y documento de instalación para el 20 de septiembre.

**6. Atributos:** De manera integral como equipo de trabajo se desea generar un ambiente idóneo, cumplir a cabalidad con los roles establecidos, adquirir nuevos conocimientos y reforzar aquellos vistos en clase mediante la práctica, haciendo un análisis crítico en la resolución de problemas, todo lo anterior con el objetivo de completar el prototipo en el tiempo establecido.

## **ROLES:**

### **1. Developer (Desarrollador):**

- a. Los desarrolladores son responsables de escribir y mantener el código fuente de un proyecto.
- b. Implementan las funcionalidades según los requisitos del proyecto.
- c. Trabajan en estrecha colaboración con otros miembros del equipo, como diseñadores y analistas de negocios, para llevar a cabo el desarrollo de software.

### **2. Business Analyst (Analista de Negocios):**

- a. Los analistas de negocios se enfocan en comprender las necesidades del cliente y traducirlas en requisitos técnicos.
- b. Desarrollan documentación detallada, como casos de uso, historias de usuario o especificaciones funcionales.
- c. Actúan como un puente entre los interesados del negocio y el equipo de desarrollo.

### **3. Architect/Tech Lead (Arquitecto / Líder Técnico):**

- a. Los arquitectos o líderes técnicos son responsables de diseñar la arquitectura técnica del proyecto.
- b. Definen las tecnologías y enfoques a utilizar.
- c. Supervisan la implementación técnica y garantizan que se cumplan los estándares de codificación.

### **4. QA Lead (Líder de Control de Calidad):**

- a. El líder de control de calidad es responsable de la gestión de la calidad del proyecto.
- b. Diseña estrategias de pruebas, crea casos de prueba y supervisa la ejecución de pruebas.
- c. Asegura que el software cumpla con los estándares de calidad y los requisitos del cliente.

**5. Database Administrator (Administrador de Bases de Datos):**

- a. Los administradores de bases de datos son responsables de gestionar y mantener las bases de datos del proyecto.
- b. Realizan copias de seguridad, optimizan el rendimiento y garantizan la seguridad de los datos.
- c. Trabajan en colaboración con otros equipos para asegurar que los datos estén disponibles y sean coherentes.

**6. Database Designers (Diseñadores de Bases de Datos):**

- a. Los diseñadores de bases de datos crean la estructura y el diseño de las bases de datos utilizadas en el proyecto.
- b. Definen las tablas, relaciones y esquemas de datos según los requisitos del sistema.

**7. Project Manager (Gerente de Proyecto):**

- a. El gerente de proyecto es responsable de planificar, coordinar y supervisar todo el proyecto.
- b. Gestiona el presupuesto, el cronograma y los recursos, y comunica el progreso a las partes interesadas.
- c. Se asegura de que el proyecto se entregue a tiempo y dentro del alcance y presupuesto definidos.

**8. Infrastructure Engineer (Ingeniero de Infraestructura):**

- a. Los ingenieros de infraestructura se encargan de diseñar, implementar y mantener la infraestructura técnica necesaria para el proyecto, como servidores, redes y sistemas de almacenamiento.

**9. Graphic Designer (Diseñador Gráfico):**

- a. Los diseñadores gráficos crean elementos visuales, como gráficos, imágenes y diseños, para el proyecto.

- b. Su enfoque está en la estética y la presentación visual del producto.

#### **10. UX Designer (Diseñador de Experiencia de Usuario):**

- a. Los diseñadores de experiencia de usuario se concentran en crear una experiencia de usuario satisfactoria y eficiente.
- b. Diseñan la interfaz de usuario y se aseguran de que sea intuitiva y fácil de usar.

#### **11. Documenter (Documentador)**

- a. El documentador se encarga de crear y mantener la documentación crucial del proyecto.
- b. Su función es organizar información como manuales, especificaciones técnicas y guías de usuario.
- c. Asegurar que toda la documentación sea clara y esté disponible para el equipo y las partes interesadas.
- d. Trabaja en colaboración con otros miembros del equipo para recopilar información necesaria.

### **REGLAS**

#### **1. Roles y Responsabilidades:**

- a. Establecer claramente los roles y responsabilidades de cada miembro del equipo desde el principio del proyecto.
- b. Definir quién será el líder del proyecto y cómo se tomarán las decisiones.

#### **2. Comunicación:**

- a. Establecer canales de comunicación efectivos para mantener a todos los miembros del equipo informados.
- b. Fomentar la comunicación abierta y la retroalimentación constante entre los miembros del equipo.

#### **3. Planificación y Seguimiento:**

- a. Desarrollar un plan de proyecto detallado que incluya cronogramas, backlog y objetivos claros.

- b. Realizar reuniones regulares de seguimiento para evaluar el progreso y ajustar el plan según sea necesario.

#### **4. Control de Versiones:**

- a. Utilizar un sistema de control de versiones, como Git, para rastrear y gestionar el código fuente del proyecto.
- b. Asegurarse de que todos los miembros del equipo comprendan cómo utilizar el sistema de control de versiones de manera efectiva.

#### **5. Código Limpio y Comentarios:**

- a. Mantener un estándar de codificación uniforme y escribir código limpio y legible.
- b. Agregar comentarios claros y explicativos en el código para facilitar la comprensión y el mantenimiento.

#### **6. Pruebas y Calidad:**

- a. Implementar pruebas unitarias y pruebas de integración de manera continua.
- b. Realizar pruebas de calidad para garantizar que el software cumpla con los estándares definidos y los requisitos del cliente.

#### **7. Documentación:**

- a. Mantener documentación actualizada que incluya especificaciones técnicas, manuales de usuario y cualquier documentación relevante del proyecto.
- b. Asegurarse de que la documentación sea accesible para todos los miembros del equipo.

#### **8. Gestión de Cambios:**

- a. Establecer un proceso para gestionar cambios y nuevas características en el proyecto.
- b. Evaluar cuidadosamente el impacto de los cambios antes de su implementación.

#### **9. Resolución de Conflictos:**

- a. Establecer un procedimiento claro para la resolución de conflictos dentro del equipo.



- b. Fomentar un ambiente de trabajo colaborativo y resolver las diferencias de manera constructiva.

**10. Entrega a Tiempo:**

- a. Comprometerse a cumplir con los plazos establecidos y a entregar el software a tiempo.

**11. Evaluación y Retroalimentación:**

- a. Realizar una evaluación periódica del progreso del proyecto y del desempeño del equipo.
- b. Utilizar la retroalimentación para mejorar continuamente el proceso de desarrollo.

**Cronograma:**

			Agosto						
ACTIVIDAD	Encargado	Duración	24	25	26	27	28	29	30
Reunión inicial	Todos	1h 30 min							
Organización en AzureDevops y creación del repositorio	Juan Daniel	1 hora							
Elaboración diagrama de clases	Meibel y Sebastian	2 horas							
Creación de modelo conceptual	Juan, Valeria y Stephanie	2h y 30 min							
Realización del plan de proyecto	Todos	2 horas							

Entrega de la tarea corta 1	Por definir	10 min							
-----------------------------	-------------	--------	--	--	--	--	--	--	--

			Setiembre						
ACTIVIDAD	Encargado	Duración	1	2	3	4	5	6	7
Reunión grupal 2	Todos	1h 30 min							
Elaborar el modelo entidad relación	Por definir	2 horas							
Creación del diagrama de arquitectura para la implementación del prototipo	Por definir	2 horas							
Investigación sobre herramientas Angular, HTML...	Todos	2 días							
Realizar resumen Ejecutivo Avance 1	Por definir	1 hora							
Entrega tarea corta 2	Por definir	10 min							
Entrega resumen Ejecutivo Avance 1	Meibel	10 min							
			8	9	10	11	12	13	14

Reunión grupal 3 para distribuir tareas	Todos	1 hora							
Iniciar con funciones de código fuente	Todos	1 semana							
Realizar resumen Ejecutivo Avance 2	Por definir	1 hora							
Entregar resumen Ejecutivo Avance 2	Por definir	10 min							

			Setiembre						
ACTIVIDAD	Encargado	Duración	15	16	17	18	19	20	
Informe 1 avances sobre código fuente	Todos	1 hora							
Desarrollo de prototipo	Todos	3 días							
Ejecucion de manual de usuario	Por definir	1 h, 30 min							
Creación de documentación técnica y de instalación	Por definir	2h y 30 min							
Revisión de detalles finales	Todos	2 horas							
Entrega del prototipo	Por definir	10 min							

## Minutas de las sesiones de trabajo:

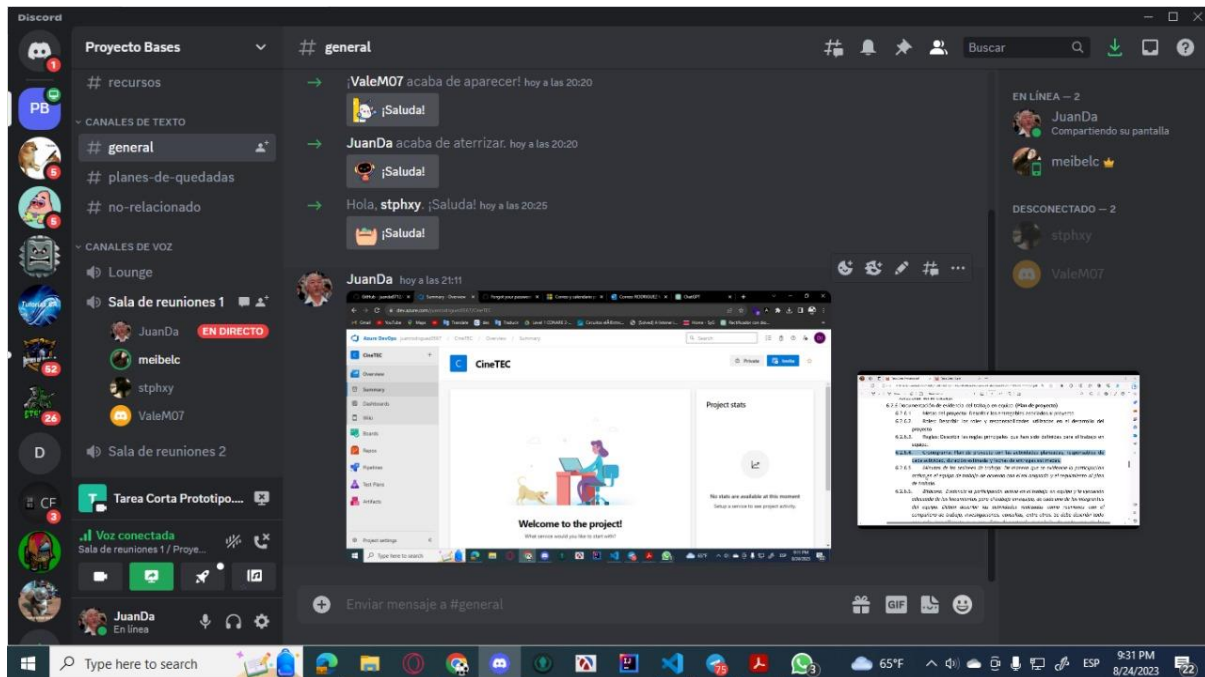
### Minuta 1:

24 de agosto de 2023

Participantes: Meibel, Juan, Estefany y Valeria

Actividades: En esta reunión se conversó sobre los procesos de organización dentro del grupo, así como la división del primer entregable. Entre estos aspectos iniciales se tomó la decisión de crear un proyecto en azure y su respectivo repositorio para tener una base sobre la cual trabajar en los siguientes avances de las diferentes etapas de las tareas. Finalmente, se generaron algunas preguntas sobre aspectos relacionados a la tarea 1 y el documento de trabajo en general, las cuales se consideraron para aclarar en la próxima clase de bases de datos.

Evidencia:



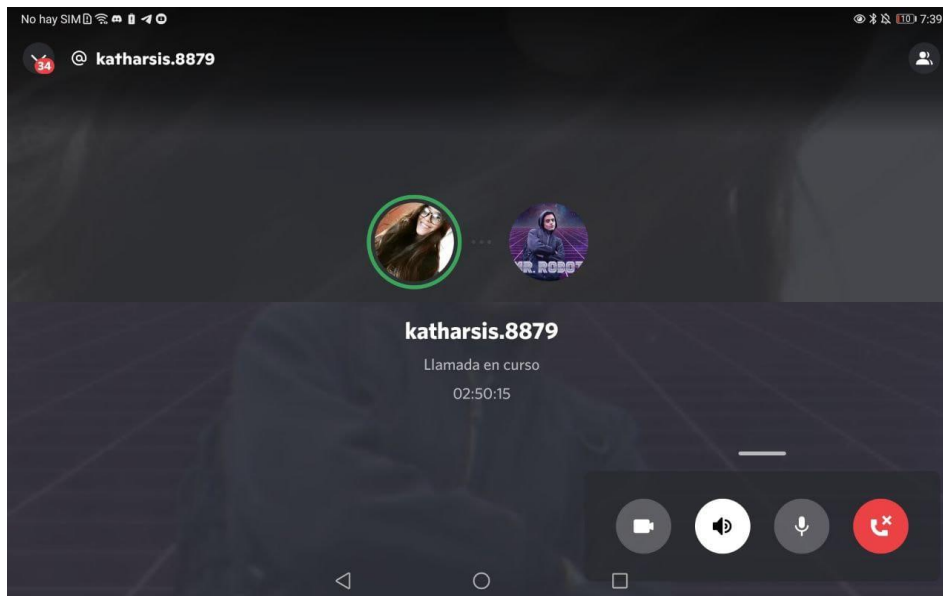
## Minuta 2:

27 de agosto de 2023

Participantes: Meibel, Sebastián

Actividades: La reunión tuvo como objetivo elaborar el diagrama de clases del proyecto, el cual pretende representar visualmente las clases que serán utilizadas a futuro para el código, en cada recuadro se pueden ver los objetos y métodos, así como la correspondencia entre ellas.

Evidencia:



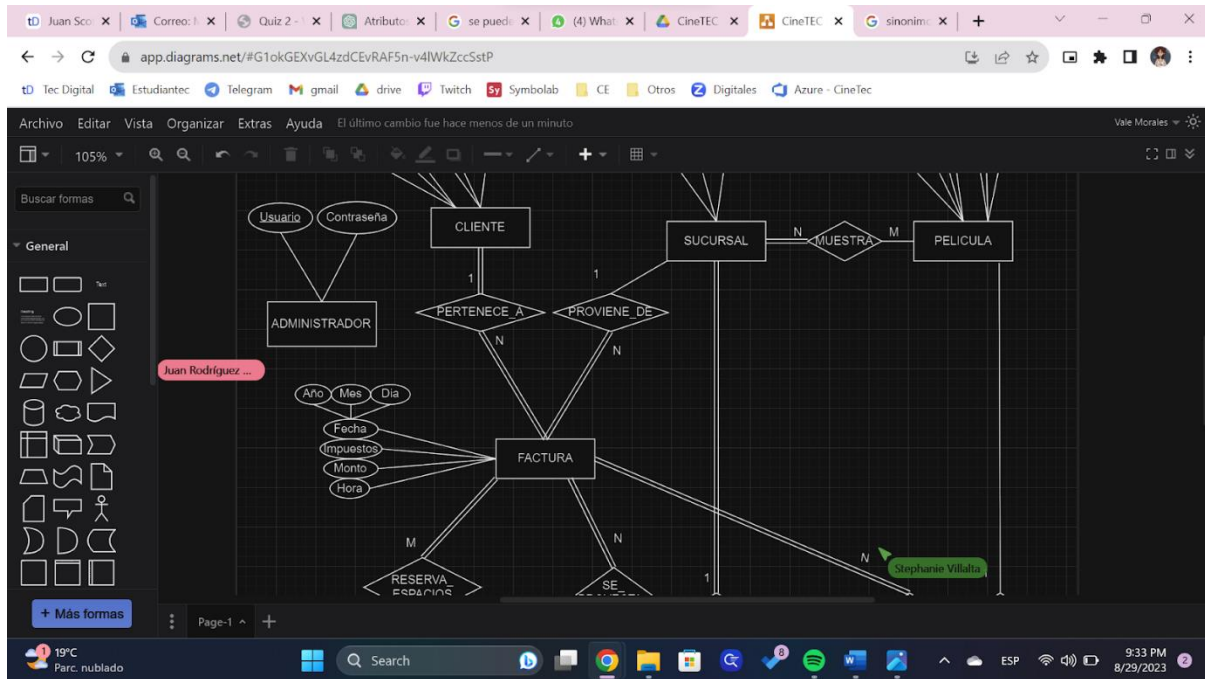
## Minuta 3:

29 de agosto de 2023

Participantes: Estefany, Juan y Valeria

Actividades: Se realizó la reunión con el propósito de finalizar la elaboración del diagrama conceptual, de manera que se analizaron las entidades, atributos y relaciones determinantes para la base de datos. Cada participante revisó y contribuyó para la resolución de problemas en el diseño inicial, para concluir se realizaron diversos cambios necesarios completando así la tarea.

Evidencia:



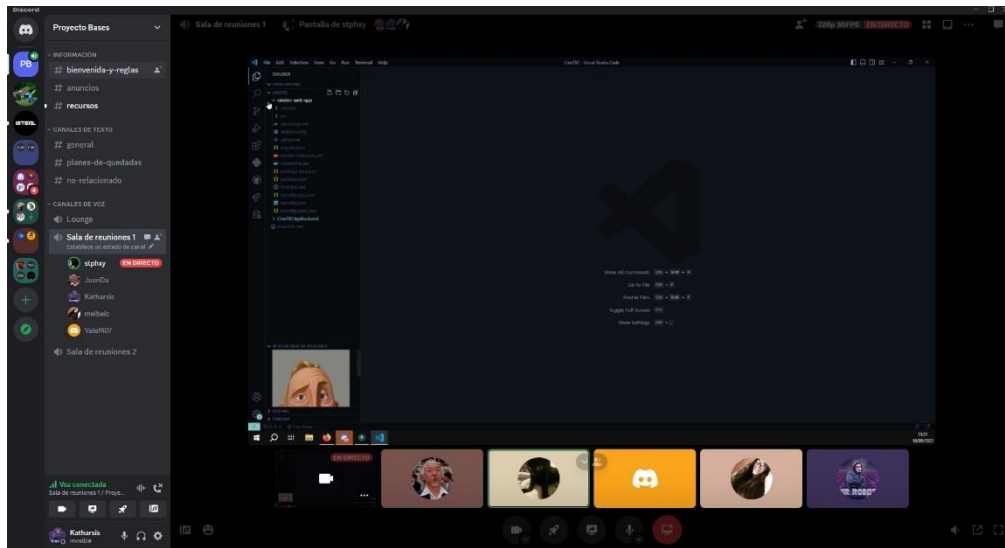
## Minuta 4:

10 de setiembre de 2023

Participantes: Estefany, Juan, Valeria, Meibel, Sebastián

Actividades: La reunión fue realizada con fin de hacer la distribución de tareas para el prototipo, se llegó al acuerdo de que Juan Daniel y Meibel estarán a cargo del BackEnd, Estefanny y Valeria elaborarán el FrontEnd y Sebastián desarrollar la aplicación móvil. Este mismo día se instalaron los entornos necesarios para empezar a trabajar, como docker, Visual Studio Code, Microsoft Visual Studio

Evidencia:



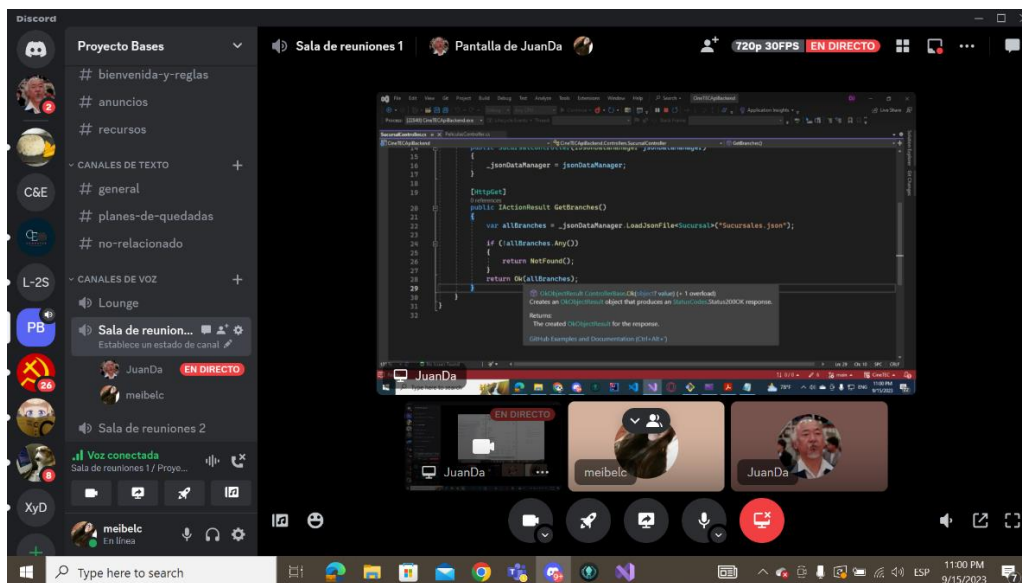
## Minuta 5:

15 de setiembre de 2023

Participantes: Meibel, Juan Daniel

Actividades: La reunión fue llevada a cabo con el fin de revisar los endpoints realizados para cada entidad individual, este mismo día fueron elaborados los modelos y controladores de las tablas intermedias, relaciones como SalaProyeccion, SucursalPelicula y el atributo multivaluado protagonistas.

Evidencia:



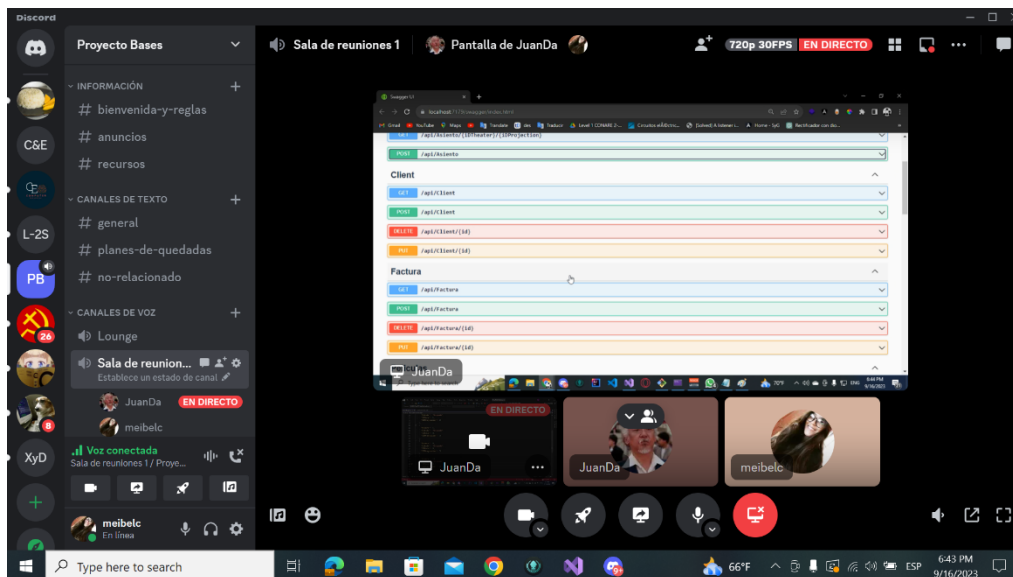
## Minuta 6:

16 de setiembre de 2023

Participantes: Meibel, Juan Daniel

Actividades: Se solucionan algunos errores detectados como el poder filtrar las proyecciones de cada película por cine, además de la generación de la factura, se hicieron algunas modificaciones en el modelo conceptual y relacional que no se ajustaban a la lógica del BackEnd como el estado de los asientos.

Evidencia:



## Minuta 7:

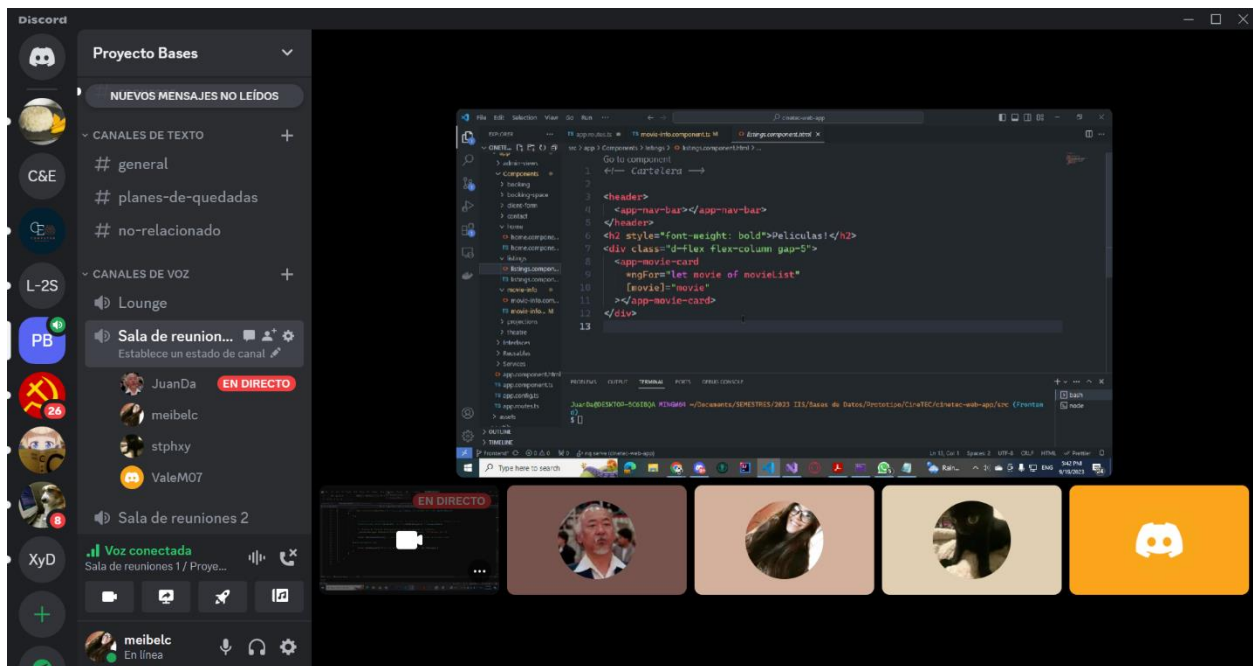
18 de setiembre de 2023

Participantes: Meibel, Juan Daniel, Valeria, Estefanny

Actividades: Se planea una reunión con el fin de poder ir enlazando los endpoints hechos en el BackEnd con las páginas elaboradas en FrontEnd, surge la necesidad de realizar un endpoint de GetSpecificMovie, el cual obtiene la información de una película en específico al enviar su OriginalName.

Evidencia:





## Bitácoras

Bitacora Meibel Ceciliano			
Fecha	Actividad realizada	Duración	Estado
27/8/2023	Elaboración del diagrama de clases	2 horas	Completado
29/8/2023	Realizar metas, cronograma, formato de tablas de bitácora	1 hora	Completado
30/8/2023	Trabajar en la elaboración del diagrama de clases para la Tarea Corta 1	1 h, 30 min	Completado
12/9/2023	Generar el mapeo para pasar del modelo conceptual al modelo relacional	2 horas	Completado
13/9/2023	Investigar sobre cómo funciona la interfaz y estructura de un api rest service	2 horas	Completado

14/9/2023	Se inician con pruebas sobre la creación de los diferentes endpoints para cada entidad, delete, put, post y get	3 horas	Completado
15/9/2023	Junto a Juan Daniel se elaboran los modelos y controladores de las tablas intermedias, relaciones como SalaProyeccion, SucursalPelicula y el atributo multivaluado protagonistas.	5 horas	Completado
17/9/2023	Crear el endpoint de GetSpecificMovie, el cual fue requerido por el equipo de FrontEnd, se trabaja también en el controlador de Sala realizando el post, put y delete.	2 horas	Completado
19/9/2023	Endpoints de modificar y eliminar para controladores Sucursal-Pelicula, Sala-Proyeccion, Asiento. Generar la documentación técnica	6 horas	Completado

Bitácora Sebastian Hidalgo			
Fecha	Actividad realizada	Duración	Estado
27/8/2023	Elaboración del diagrama de clases	2 horas	Completado
10/9/2023	Se empezó la tarea del desarrollo de la vista móvil	10 días	Pendiente
12/9/23	Se investigó acerca de cómo desarrollar la aplicación móvil con los mismos componentes web que se usaron en el	4 horas	Completado
21/9/23	Despliegue de la aplicación móvil	4h	Pendiente

Bitácora Valeria Morales			
Fecha	Actividad realizada	Duración	Estado

29/8/2023	Se participó en una reunión con Juan y Estefanny para la elaboración del diagrama conceptual	1 hora y media	Completado
16/9/2023	Se investigó sobre el uso de Angular y se inició con pruebas para establecer las rutas. Se inició la homepage.	2 horas y media	Completado
17/9/2023	Se participó en dos reuniones. Una con Estefanny, la cual se realizó para hablar sobre los avances en respectivos a algunas de las páginas principales y el ruteo. La segunda fue en conjunto con los demás integrantes del grupo. Para ver la conexión entre Angular y los endpoints del API. Además de esto se trabajó en la reestructuración del flujo de la vista del cliente.	1 día	Mayormente completado, se pretende completar las páginas faltantes el día 18/9/2023
18/9/2023	Se finalizó con la estructuración del flujo de la vista del cliente.	1 hora	Completado
18/9/2023	Se participo en varias reuniones con el proposito de entender la conexión entre el frontend y los endpoints mediante los servicios de API.	8 horas	Se completo la conexion de la cartelera e informacion de una pelicula particular
19/9/2023	Se trabajo en la conexión de selección de la proyección y selección de la sala con el backend  Se trabajo en la selección de asientos y el formulario de usuario, sin embargo, no se consiguió hacerlo funcional en conjunto con el backend  Se participio en reuniones durante el transcurso del día para solucionar dudas respectivas a la conexión	8 horas	Se completo la seleccion de la proyeccion y de la sala.

20/9/2023	Se trabajo en la documentación, manual de usuario y manual de instalación	3 horas	Completado
-----------	---	---------	------------

Bitácora Juan Daniel Rodriguez			
Fecha	Actividad realizada	Duración	Estado
25/8/2023	Creación de los Sprints para las entregas del prototipo y tareas. Creación del Backlog (Epics, Features, User Stories and Tasks) para el modelo conceptual, diagrama de clases y plan de trabajo. Creación de documento (Word compartido) para documentar el modelo conceptual y agregar links importantes: proyecto Azure DevOps y diagrama ER (notación de Chen).	2 horas	Completado
26/8/2023	Análisis del caso CineTEC para extraer entidades y atributos preliminares.	3 horas	Completado
27/8/2023	Actualizar la lista de entidades y atributos. Creación de las entidades, atributos y relaciones utilizando la lista actualizada del modelo conceptual.	4 horas	Completado

5/9/2023	Documentacion sobre modelado de bases de datos con JSON y creacion de modelos para las relaciones diagrama relacional	6 horas	Completado
15/9/2023	Desarrollo de las funciones CRUD de los archivos JSON	8 horas	Completado
18/9/2023	Desarrollo de endpoints que consultan la base de datos con jsons y utiliza las funciones CRUD para la logica de cada endpoint	12 horas	Completado

20/9/2023	Desarrollo de las vistas de administrador de la página web	15 horas	Completado
-----------	--	----------	------------

Bitacora Stephanie Villalta			
Fecha	Actividad realizada	Duración	Estado
29/8/2023	Participación en la revisión y contribución de ideas para elaborar el diagrama conceptual en conjunto con Valeria y Juan.  Revisión y aportes de las minutas grupales.	2 horas y media	Completado
5/9/2023	Aprender sobre las herramientas por utilizar en el frontend	2 horas	Completado
14/9/2023	Investigación sobre angular y relacionado al desarrollo web	4 horas	Completado
15/9/2023	Investigación sobre angular, relacionado al desarrollo web y bootstrap	4 horas	Completado
16/9/2023	Se dió seguimiento al avance del frontend por la compañera Valeria, realizando cambios en la organización del proyecto para tener una base más sólida, al igual que avanzar con la construcción de la página.	9 <u>horas</u>	Completado
17/9/2023	Se completó una reunión con Valeria para hablar sobre los cambios realizados en la página así como el uso de herraminetas de angular y bootstrap, tambien se siguió con el avance de la página.	1 Día	Completado
18/9/2023	Se realizaron reuniones para unir de la forma más adecuada el backend con el frontend por los endpoints.	4 horas	Se requería tiempo para realizar los endpoints nuevos y seguir con el trabajo que

			nos correspondía
19/9/2023	Se siguió con el enrutamiento de las páginas así como el diseño de la página en general en conjunto con la compañera Valeria.	10 horas	Para este punto solo faltaba el enrutamiento de la página de selección de asiento a reservar la entrada
20/9/2023	Se realizaron avances en la documentación, específicamente en el manual de usuario y manual de instalación.	3 horas	Completado
21/9/2023	Se realizaron las últimas actualizaciones de la bitacora	30 min	Completado

## Bibliografía

Angular. (s. f.). Recuperado de <https://angular.io/docs>

Code Skills. (2018, 4 abril). *Servicios Web - Web API REST - Parte 1* [Vídeo]. YouTube.

<https://www.youtube.com/watch?v=PwIHGCeC6-I>

Code Skills. (2018b, abril 11). *Servicios Web - consumo API REST - parte 3* [Vídeo]. YouTube.

<https://www.youtube.com/watch?v=HtBX2j2befA>

*Bootstrap templates and more on WrapBootstrap*. (s. f.). WrapBootstrap.

<https://wrapbootstrap.com/>

Macias, S. (2018). Construyendo una web API REST segura con JSON Web token en .NET

(Parte II). *Enmilocalfunciona*. [https://www.enmilocalfunciona.io/construyendo-una-web-](https://www.enmilocalfunciona.io/construyendo-una-web-api-rest-segura-con-json-web-token-en-net-parte-ii/)

[api-rest-segura-con-json-web-token-en-net-parte-ii/](https://www.enmilocalfunciona.io/construyendo-una-web-api-rest-segura-con-json-web-token-en-net-parte-ii/)

Solís, N(2015). Manual del Guerrero AngularRJS. Recuperado de [enlace](#)