

# Instituto Tecnológico de Costa Rica



## Bases De Datos

II Semestre 2021

Armando Fallas Garro.  
Carmen Araya Chacon.  
Fatima Leiva Chinchilla.  
Harold Espinoza Matarrita.

### **Profesor:**

Marco Rivera Meneses

Documentación Técnica

## **Descripción de las estructuras de datos desarrolladas**

Para este caso se utilizaron tablas de tal manera que se crearon a partir del modelo conceptual y relacional donde:

Se realiza una tabla para las sucursales de CineTec que contiene un id único de la sucursal, el nombre de esa sucursal, la ubicación donde se encuentra y la cantidad de salas que posee.

Se crea una tabla para las Salas de cada sucursal, esta contiene el id único asociado a la sala, la cantidad de filas, la cantidad de columnas, la capacidad que tiene esa sala, y una referencia hacia el id de la sucursal a la que se encuentra asociada.

Se realiza una tabla para los asientos de las salas, esta tabla contiene el id para el asiento, la disponibilidad del asiento y una referencia hacia el id de la sala a la que está asociado ese asiento.

Se realiza una tabla para las películas que se presentarán en el cine, esta tiene el nombre original de la película, un nombre con el que se conoce, la duración, los precios para niños, adultos y de la tercera edad, además se guarda el enlace a la imagen de portada de la película

Se crea una tabla para los directores que contiene el nombre del director, y el nombre de la película que dirigió.

Se crea una tabla para los protagonistas que contiene el nombre del protagonista y el nombre de la película que interpreta.

Se crea una tabla para la clasificación, la cual contiene el tipo de clasificación del contenido y el nombre de la película que tiene esa clasificación.

Se realiza una tabla `pelicula_por_sala` que funcionara de intermediario entre la tabla `pelicula` y la tabla `sala` para las proyecciones, esta tiene un id que se da para

cada proyección, la hora de la película, el id asociado a la sucursal, el id de la sala en la que se dará y el nombre de la película que se proyectará.

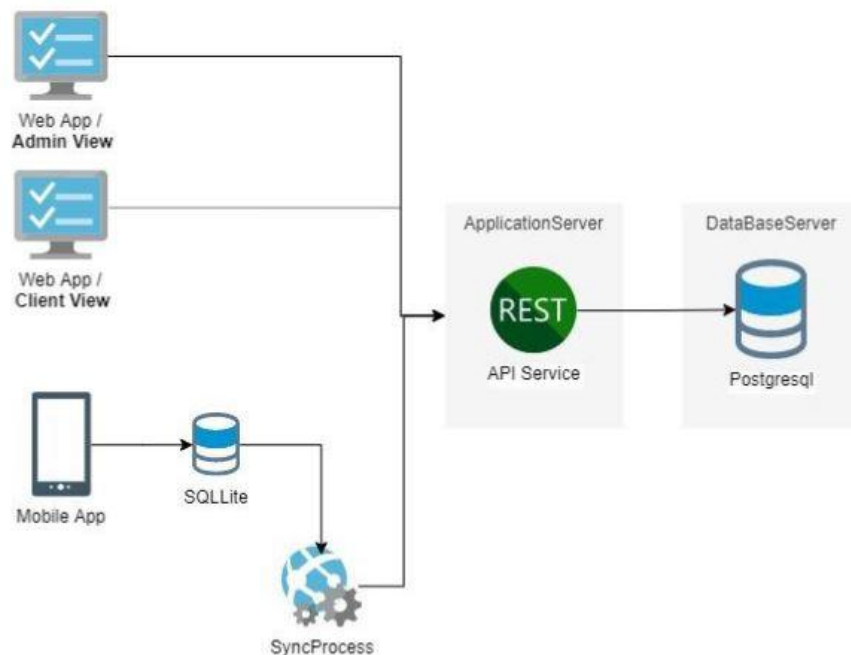
Se crea una tabla empleado para almacenar, el primer y segundo nombre así como los dos apellidos del trabajador, la cédula, la fecha de nacimiento, el número de teléfono, la fecha en la que ingresó a la empresa, un usuario y contraseña para el trabajador, así como el id de la sucursal donde trabaja el empleado y el nombre del rol que desempeña.

Se crea una tabla para los roles de los empleados, esta contiene el nombre del rol que desempeña y una breve descripción para el rol.

Se realiza una tabla para los clientes que contiene, la cédula, el nombre completo, la fecha de nacimiento, el número de teléfono, un usuario y contraseña para el acceso a la plataforma y el id de la sucursal en la que se registró el cliente

Se realiza una tabla para las facturas que almacena una clave, consecutivo e id para la factura, con el fin de garantizar que cada factura sea única, además tiene un breve detalle de la compra, la fecha en la que se realizó y la cédula del cliente que realizó la compra.

## Descripción detallada de la arquitectura desarrollada



La rest api montada en c# funciona como intermediario entre la base de datos montada en Postgresql y las aplicaciones de los clientes y administrador.

Las aplicaciones web deben realizar las peticiones necesarias directamente con la API Service y este a su vez les retorna la información necesaria que se encuentra en la base de datos a las aplicaciones web. Pero de forma contraria la aplicación móvil no mantiene una conexión directa con la API Service ya que este cuenta con su propia base de datos montada en SQLite, por lo cual la aplicación obtiene toda información directamente de la base.

Debido a que la aplicación no mantiene una conexión directa con la API Service debe de haber un proceso de sincronización con la base de datos SQLite y la de Postgresql, así que la API Service le envía toda la información necesaria de cada una de la tablas a la base de datos SQLite y este la agrega, cuando esta primera sincronización termina este misma base le envía los datos de las nuevas facturas

agregadas, además de la información de los asientos de cada sala que ya no están disponibles a la API Service.

### **Problemas conocidos**

1. La aplicación móvil tiene un bug en la vista de los asientos, debido a que si se selecciona un asiento y se quita la selección del mismo, no se logra seleccionar de nuevo si así se quiere, el error puede estar en el content description de cada asiento, posiblemente se agregó un nombre incorrecto.

### **Problemas encontrados**

1. La aplicación móvil dejaba de compilar en Android Studio sin razón alguna, a partir de algunos foros se intentó reiniciar el proyecto, se pasó de carpeta todo el documento y se daba al botón de sincronizar pero sin poder lograr solucionarlo, al final el error era provocado por lo antivirus instalados en la computadora, en este caso Avast y Eset, por lo que se deben desactivar mientras se trabajaba en la programación.
2. Se obtuvieron varios errores al realizar la conexión de la aplicación con la api, esto se dio por varios errores principalmente errores de los estudiantes, ya que se colocaba una dirección incorrecta para las peticiones, se colocaban nombres distintos a los atributos que nos llegaba en el json y se olvidaba actualizar la api en el IIS.
3. Al cargar los asientos de discapacitados en la aplicación móvil cargaban las imágenes incorrectas, esto debido a que en las condiciones if para poder diferenciarlos se realizó de manera incorrecta causando el error.
4. Al realizar cambios en la base de datos de la aplicación móvil, estos no se veían reflejados al compilar, esto era causado a que se debía borrar los datos de la aplicación para así borrar la base de datos antigua.

5. Al realizar la conexión del API con la base de datos esta fallo debido a que no se configuró correctamente el “connectionString” que contiene los datos necesarios para conectarse con la base de datos, este se soluciona modificando el appsettings.json del API ingresando la parte faltante del “connectionString”.
6. Al intentar realizar una petición de la vista Administrador/Cliente al API se da un error de permisos de “Cors” el cual impide ejecutar los métodos, por lo que se arregla configurando los permisos de “Cors” en el API para permitir el acceso a los métodos solicitados
7. Al hacer un update del estado de los asientos a la base de datos los asientos se reordenaban, ocasionando que a la hora de presentarlos en la plataforma se desordenaban, para esto se aplicó un método de sorting de javascript para reordenar la lista de asientos que ingresaba de la base de datos.

## **Documentación de evidencia del trabajo en equipo**

### **Conclusiones**

1. El uso de un manejador de código facilita el control de la programación en general del proyecto, además que funciona como copia de seguridad del trabajo por si algo sucediera con las computadoras usadas para realizar dicho trabajo.
2. El uso de Internet Information Services da la comodidad de poder montar nuestra rest api en este caso en una red local y así poder acceder a ella desde cualquier dispositivo conectado a esta red, de igual manera facilita el manejo de las distintas imágenes usadas, ya que solo se coloca la dirección de cada una de las imágenes en el IIS y se carga en las distintas interfaces.
3. Android Studio simplifica la elaboración de una aplicación android, ya que cuenta con gran cantidad de métodos y herramientas que facilita la

programación, así como previsualizar la interfaz de la aplicación antes de compilar.

4. Angular permite manejar servicios para obtener información de un servidor y permite almacenar información útil para utilizarla en distintos componentes sin relación alguna.
5. El uso de Angular para la creación de plataformas front-end resulta muy conveniente dado la cantidad de métodos que contiene como extensión, así como otros módulos que contienen más funciones como Angular Material.

## **Recomendaciones**

1. Para evitar fallos de la nada al compilar en android studio es recomendable no mantener activadas cualquier antivirus instalado, solo se debe dejar el que trae windows por defecto ya que este no genera errores.
2. Estar conectado a la red en la cual se montó la API para que así pueda visualizar las imágenes de cada película en la aplicación móvil.
3. La aplicación solo fue hecha para dispositivos Android.
4. El dispositivo android donde se corre la aplicación móvil debe tener como mínimo Android 6.0.
5. No recargar la página de la vista cliente una vez que se ha ingresado porque se puede borrar la data guardada en los servicios de Angular.
6. Cerrar y abrir sesión nuevamente en la vista cliente si aparece algún error de recargar la página.

## Bibliografía

*Cómo guardar datos con SQLite | Desarrolladores de Android.* (s. f.). Android Developers.  
<https://developer.android.com/training/data-storage/sqlite?hl=es-419>

*Introducción a Android Studio | Desarrolladores de Android.* (s. f.). Android Developers.  
<https://developer.android.com/studio/intro?hl=es-419>

*Retrofit.* (s. f.). Retrofit. <https://square.github.io/retrofit/>

Robles, V(2021) *Master en Javascript: Aprender Js, jQuery, Angular, NodeJs.*  
<https://www.udemy.com/course/master-en-javascript-aprender-js-jquery-angular-nodejs-y-mas/>