

# Proyecto #1 - CineTEC

Instituto Tecnológico de Costa Rica  
Área Académica Ingeniería en Computadores  
Bases de Datos (CE3101)  
II Semestre 2021  
Valor 20%



## Objetivo general

Desarrollar una aplicación que permita manejar la descripción del caso expuesto en la sección **Requerimientos del Software**.

## Objetivos específicos

- Aplicar los conceptos del modelo conceptual y relacional.
- Crear una Base de Datos en Postgresql para que permita el almacenamiento de los datos.
- Crear un servicio API para que centralice la funcionalidad.
- Crear una página Web para que exponga la funcionalidad al usuario.
- Crear una aplicación móvil utilizando SQL Lite como Base de datos empotrada.
- Usar herramientas como Angular, Bootstrap, HTML5, CSS, Entity Framework, y Reporting Services o Cristal Reports.
- Crear un documento de instalación.
- Crear un plan de proyecto que evidencie: la participación en el equipo de trabajo asignado de acuerdo a su rol y la ejecución de lineamientos para trabajo en equipos.

## Descripción del problema

Su equipo ha sido designado con la responsabilidad del desarrollo de un sistema que permita gestionar la administración de la cadena de cines **CineTEC**.

En este periodo de pandemia y con las bajas en ventas que ha tenido el cine podría pensarse que no es tiempo de invertir en un nuevo sistema que permita realizar una mejor gestión de los cines, sin embargo la gerencia considera que es el momento oportuno debido a que cualquier imprevisto con afectación negativa afectaría a un mínimo de usuarios.

Por lo tanto la gerencia ha tomado la decisión de desarrollar una nueva aplicación de facturación y venta de boletos.

La definición de requerimientos se define en la sección **Requerimientos de Software**. Es importante mencionar: en caso de retraso de la entrega de la aplicación la empresa contratada deberá pagar la multa de \$20.000 mensuales por cada semana de atraso.

A continuación, se describen las vistas de la solución que debe ser desarrollada:

- **Vista Administrador:** esta es la plataforma que utilizarán los administradores para la configuración del sistema y la visualización de los reportes de uso del sistema por ubicación geográfica.

- **Vista Cliente:** plataforma que utilizarán los usuarios para gestionar la información correspondiente a la reservación de boletos.
- **App Móvil:** corresponde a la plataforma que utilizarán los usuarios para gestionar sus reservas.

## Requerimientos del Software

### → Vista Administrador.

- ◆ **Login:** el sistema debe permitir que los administradores puedan autenticarse e ingresar al sistema.
- ◆ **Registro de Empleados:** Permite al administrador gestionar toda la información de los empleados, donde se almacenará el nombre del empleado, cédula, número de teléfono, fecha de nacimiento, edad, fecha de ingreso a la cadena de cines, rol, usuario y contraseña.
- ◆ **Registro de clientes:** Permite al administrador gestionar toda la información de los clientes, donde se almacenará el nombre del cliente, cédula, número de teléfono, fecha de nacimiento, edad.
- ◆ **Registro de Películas:** Permite gestionar toda la información de las películas donde se almacenará el nombre original de la película, nombre, imagen, duración, **protagonistas, director, clasificación.**
- ◆ **Registro de sucursales:** Permite gestionar toda la información de las sucursales del cine, deberá poder almacenar un nombre del cine, ubicación, cantidad de salas.
- ◆ **Registro de Salas:** Permite gestionar toda la información de las salas de una sucursal. Deberá poder almacenar un identificador, nombre de la sucursal a la que pertenece, cantidad de filas, columnas y/o espacios, capacidad.
- ◆ **Asignación de proyecciones:** El sistema debe poder gestionar la lista de proyecciones de las películas. Osea debe permitir asignar a una sala, en un momento específico a una película.
- ◆ **Restricciones de capacidad Covid-19:** Debido a las restricciones sanitarias impuestas por el ministerio de salud la capacidad debe ser limitada según lo indicado por el ministerio. Este valor será variable y se utilizará para calcular el aforo máximo permitido (el despliegue de asientos disponibles).

### → Vista Cliente.

- ◆ **Login:** el sistema debe permitir que los clientes puedan autenticarse e ingresar al sistema, el sistema deberá cargar todos los datos del cliente en la pantalla inicial.
- ◆ **Selección del cine:** Inicialmente el usuario debe poder seleccionar el cine donde desea comprar boletos.
- ◆ **Selección de la película:** Una vez seleccionado el cine la aplicación deberá mostrar las películas que tiene en cartelera.
- ◆ **Selección de la proyección:** Una vez seleccionada la película el sistema deberá mostrar las proyecciones que tiene para esa película.
- ◆ **Selección de asiento:** Una vez seleccionada la proyección el sistema deberá mostrar todos los asientos (debe diferenciar entre ocupados, desocupados y restringidos) para la proyección y el usuario seleccionará los asientos deseados.
- ◆ **Generación de la Factura:** Una vez seleccionados los asientos el sistema generara 2 archivos un XML y un PDF con la factura respectiva. El XML se deberá 'enviar' Hacienda (el

XML generado debe seguir las especificaciones de hacienda) y el PDF al cliente. Revisar ejemplos anexos.

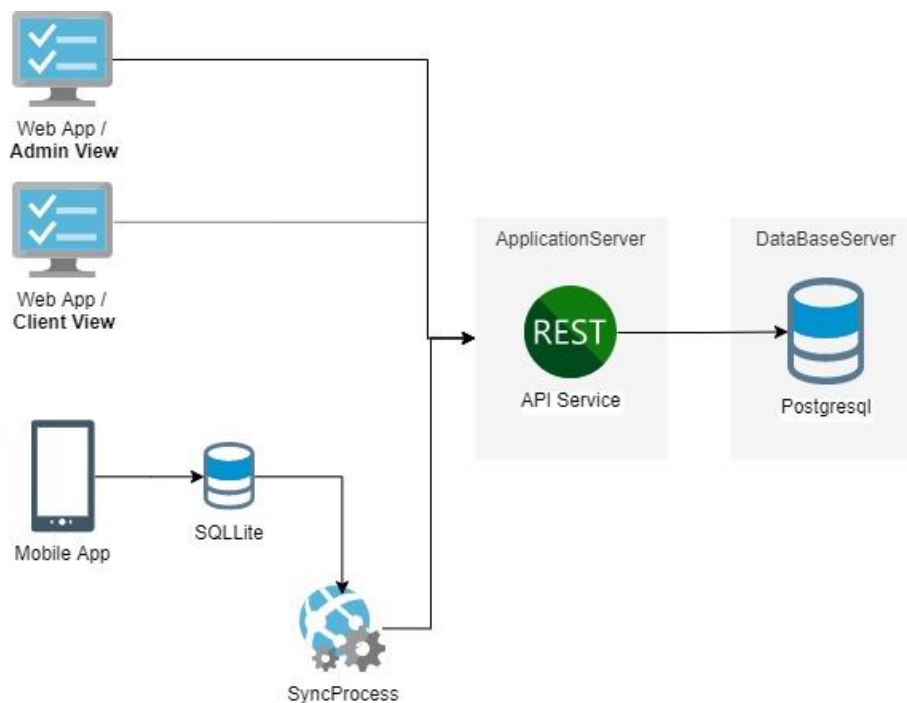
#### → Aplicación Móvil.

- ◆ La aplicación móvil debe poder utilizarse sin conexión, por tanto se debe manejar un base de datos local SQL Lite, que se sincronizará con la base de datos en la nube. Para el proceso de sincronización es indispensable conexión de datos.
- ◆ **Selección del cine:** Inicialmente el usuario debe poder seleccionar el cine elegido.
- ◆ **Selección de la película:** Una vez seleccionado el cine la aplicación deberá mostrar las películas que tiene en cartelera.
- ◆ **Selección de la proyección:** Una vez seleccionada la película el sistema deberá mostrar las proyecciones que tiene para esa película.
- ◆ **Selección de asiento:** Una vez seleccionada la proyección el sistema deberá mostrar todos los asientos (debe diferenciar entre ocupados y desocupados) para la proyección y el usuario seleccionará los asientos deseados.
- ◆ **Proceso de sincronización:** El proceso de sincronización debe ejecutarse cada cierto tiempo y debe enviar todas las reservas realizadas entre sincronizaciones. Por otro lado debe consultar y/o actualizar todos los datos de cines, sucursales y reservas realizadas.

Aspectos adicionales:

- El término gestionar corresponde a las opciones de insertar, editar, eliminar y consultar.
- Los números de factura son únicos para todos los usuarios. No pueden existir dos facturas con el mismo número de factura.

#### Arquitectura del Software



### Ilustración 3 Arquitectura deseada.

#### Requerimientos no funcionales del sistema

- El Sistema debe ser una aplicación web (utilizando Angular, Bootstrap, HTML5, CSS y Entity Framework).
- La App Móvil debe ser desarrollada utilizando Android Studio y SQL Lite como base de datos.
- La Base de Datos debe estar en Postgresql.
- **No se** permite el uso de Procedimientos Almacenados, Vistas o Triggers. Los scripts de Base de Datos deben implementarse en la capa de datos.
- La capa de servicios debe estar desarrollada en C# y debe ser desplegada en el IIS.
- El equipo de trabajo debe seleccionar a uno de sus miembros como único punto de contacto. Todas las comunicaciones y solicitudes deben ser a través de dicho punto de contacto.

#### Entregables

- Manual de Usuario.
- Documentación Técnica y del proyecto (descrita en el punto 6).
- Documento de instalación.
- Plan de Proyecto (Trabajo individual y en equipo).
  - ◆ Metas del proyecto: Describir los entregables asociados al proyecto.
  - ◆ Roles: Describir los roles y responsabilidades utilizados en el desarrollo del proyecto.
  - ◆ Reglas: Describir las reglas principales que han sido definidas para el trabajo en equipo.
  - ◆ Cronograma: Plan de proyecto con las actividades planeadas, responsables de cada actividad y fechas de entregas estimadas.
  - ◆ Minutas de las sesiones de trabajo: De manera que se evidencie la participación activa en el equipo de trabajo de acuerdo con el rol asignado y el seguimiento al plan de trabajo.
  - ◆ Bitácora: Evidencia la participación activa en el trabajo en equipo y la ejecución adecuada de los lineamientos para el trabajo en equipo, de cada uno de los integrantes del equipo. Deben describir las actividades realizadas como reuniones con el compañero de trabajo, investigaciones, consultas, entre otras. Se debe describir todo por más insignificante que sea. Esto demostrará el trabajo de cada uno de los miembros del equipo según el rol asignado.
- Script de Base de Datos.
- Script de población de Base de Datos.
- Aplicación WEB.
- Aplicación Móvil.
  - ◆ Proceso de Sincronización.
- Web API.

#### Documentación requerida

- Se deberá documentar el código fuente.
- Se deberá entregar un documento que contenga:
  - ◆ Modelo conceptual utilizando la notación de Chen.
  - ◆ Modelo relacional.

- ◆ Descripción de las estructuras de datos desarrolladas (Tablas).
  - ◆ Descripción detallada de la arquitectura desarrollada.
  - ◆ Problemas conocidos: En esta sección se detalla cualquier problema que no se ha podido solucionar en el trabajo.
  - ◆ Problemas encontrados: descripción detallada, intentos de solución sin éxito, soluciones encontradas con su descripción detallada, recomendaciones, conclusiones y bibliografía consultada para este problema específico.
- **Documentación de evidencia del trabajo en equipo.**
- ◆ Actividades planeadas, su responsable y fecha de entrega. (Plan de trabajo)
  - ◆ Minutas de sesiones de trabajo. (Seguimiento al plan de trabajo)
  - ◆ Actividades realizadas por cada estudiante. (Bitácora en digital, donde se describen las actividades realizadas, desde reuniones con el compañero de trabajo, investigaciones, consultas, etc. Se debe describir todo por más insignificante que sea, esto demostrará si ustedes están trabajando en realidad. Este es su diario de trabajo, llevan seguimiento de todo en el tiempo, imaginen que, si un compañero los releva en su trabajo, le bastaría con leer sus bitácoras para seguir el trabajo.
  - ◆ Evidencia de uso de un manejador de código (se recomienda Github).
- Conclusiones y Recomendaciones del proyecto.
- Bibliografía consultada en todo el proyecto
- Diagrama de clases y un documento que explique el porqué del diseño.

#### **Aspectos operativos y evaluación:**

1. **Fecha de entrega:** De acuerdo con el cronograma del curso y lo establecido en el TEC Digital. Se establece el siguiente plan de entregas parciales:
  - a. Plan de proyecto: 22 de Setiembre 2021
  - b. Resumen Ejecutivo Avance 1 (REA1): 29 de Setiembre 2021
    - i. Estado del Proyecto.
    - ii. Actividades realizadas en la semana.
    - iii. Actividades por realizar.
    - iv. Riesgos y plan de mitigación.
    - v. Incidentes (riesgos materializados).
  - c. Resumen Ejecutivo Avance 2 (REA2): 6 de Octubre 2021
    - i. Igual que el REA1
  - d. Funcionalidad completa: 13 de Octubre 2021
2. El proyecto tiene un valor de 20% de la nota del curso.
3. El trabajo es **en grupos de cuatro personas**.
4. La implementación tendrá un valor de un 70% de la nota final, debe estar funcional. La defensa vale un 10% y la documentación un 20%.
5. Cumplir con los requerimientos especificados en la documentación no significa que se tienen todos los puntos, se evaluará que la documentación sea coherente, acorde al tamaño del proyecto y el trabajo realizado, no escatimen en documentación.
6. Cada grupo recibirá una nota en cada uno de los siguientes apartados Código y Documentación.
7. El profesor no sólo evaluará la funcionalidad del proyecto, esto quiere decir que aunque el proyecto este 100% funcional esto no implica una nota de un 100, ya que se evaluarán aspectos de calidad de

- código, aplicación del paradigma OOP, uso de herramientas solicitadas, calidad de documentación interna y externa, trabajo en equipo.
8. No se revisarán funcionalidades parciales, ni funcionalidades no integradas.
  9. Es responsabilidad de cada miembro del grupo conocer su código, el profesor puede preguntar a cualquier miembro del grupo que le explique alguna funcionalidad/porción de código.
  10. Las citas de revisión oficiales serán determinadas por el profesor durante las lecciones o mediante algún medio electrónico.
  11. Aún cuando el código y la documentación tienen sus notas por separado, se aplican las siguientes restricciones
    - a. Si no se entrega documentación, automáticamente se obtiene una nota de 0.
    - b. Si no se utiliza un manejador de código se obtiene una nota de 0.
    - c. Si el código y la documentación no se entregan en la fecha indicada se obtiene una nota de 0.
    - d. Si el código no compila se obtendrá una nota de 0, por lo cual se recomienda realizar la defensa con un código funcional.
    - e. Si el grupo no cuenta con los equipos necesarios para realizar la revisión y no avisó al profesor de esta situación obtendrá una nota de 0.
    - f. El código debe ser desarrollado en C#, en caso contrario se obtendrá una nota de 0.
    - g. Si algún estudiante no se presenta a la revisión se obtiene nota de 0.
  12. Cada grupo tendrá como máximo 30 minutos para exponer su trabajo al profesor y realizar la defensa de éste, es responsabilidad de los estudiantes mostrar todo el trabajo realizado, por lo cual se recomienda tener todo listo antes de ingresar a la defensa.
  13. Cada excepción o error que salga durante la ejecución del proyecto y que se considere debió haber sido contemplada durante el desarrollo del proyecto, se castigará con 2 puntos de la nota final del proyecto.
  14. Cada grupo es responsable de llevar los equipos requeridos para la revisión.
  15. Durante la revisión únicamente podrán participar los miembros del grupo, asistentes, otros profesores y el coordinador del área.
  16. Las revisiones se realizan con los estudiantes matriculados en el curso, cualquier persona fuera de estos y los mencionados en el punto 15, no pueden participar en la revisión.
  17. Si no se realiza la defensa del proyecto se asignará una nota de 0 en el proyecto.

## Referencias

**AngularJS** (2018-10-04). Recuperado de: <https://angularjs.io>

**Bootstrap Themes & Templates** (2018-10-04). Recuperado de: <https://wrapbootstrap.com/>

**How to Write Doc Comments for the Javadoc Tool.** (2018-10-04). Recuperado de: <http://www.oracle.com/technetwork/articles/java/index-137868.html>

**C# Coding Conventions (C# Programming Guide).** (2018-10-04). Recuperado de: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions>