

**1. Objetivo General**

Desarrollar un **Prototipo** que permita manejar la descripción del caso expuesto en el punto 4. **Funcionalidades que se encuentran resaltada en amarillo deben considerarse únicamente para el prototipo y toda la descripción del caso para el modelo conceptual y relacional.**

**2. Objetivos Específicos**

- 2.1. **Conocer las herramientas que se utilizarán en el proyecto I y II.**
- 2.2. Crear un API/REST Service (**C#-.Net**) y realizar la conexión con el Front End de al menos para un post, un Get de un valor y un Get multivalor.
- 2.3. Crear un **prototipo** de una página web (con múltiples vistas).
- 2.4. Usar herramientas como Angular/React, Bootstrap, HTML5, CSS3, *Crystal Report, Reporting Services o alguna herramienta de reporte.*
- 2.5. Instalar localmente una aplicación Web en un ambiente Windows en el IIS (Front End y Back End).

**3. Datos Generales**

- 3.1. El valor del proyecto: 6%
- 3.2. **Nombre código: CineTec.**
- 3.3. La tarea debe ser implementada grupos de 5 personas.
- 3.4. La **fecha de entrega:**
  - 3.4.1. Plan de trabajo: 30/Agosto/2023.
  - 3.4.2. Resumen Ejecutivo Avance 1: 6/Setiembre/2023.
  - 3.4.3. Resumen Ejecutivo Avance 2: 13/ Setiembre /2023.
  - 3.4.4. Prototipo completo: 20/ Setiembre /2023.
- 3.5. Cualquier indicio de copia será calificado con una nota de 0 y será procesado de acuerdo con el reglamento.

**4. Descripción del caso de estudio**

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

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 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, desde la configuración del cine hasta las películas que se proyectarán.
- **Vista Cliente:** plataforma que utilizarán los usuarios para realizar reservaciones de las películas que desean ver.
- **App Móvil:** Igual que la vista cliente pero desde un dispositivo móvil.

#### 4.1. Requerimientos funcionales del Sistema

→ Vista Administrador.

- ◆ **Login:** el sistema debe permitir que los administradores puedan autenticarse e ingresar al sistema.
- ◆ **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 comercial, 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 CoTec-23:** 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 para el despliegue de asientos disponibles.

- Vista Cliente.
- ◆ **Selección del cine:** Inicialmente el usuario debe poder seleccionar el cine de su preferencia.
  - ◆ **Selección de la película:** Una vez seleccionado el cine la aplicación deberá mostrar las películas que tiene en cartelera.

¿ Si ocupa login donde inicia solo en el admin?

Una  
diferencia  
(real)

◆ **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.

→ **Aplicación Móvil.**

◆ **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.

¿Operaciones de cinema?

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.

4.2. **Requerimientos NO funcionales del Sistema**

- **SOLO debe implementarse la funcionalidad resaltada en amarillo para la Tarea Corta. Para la TC1 y TC2 se debe contemplar todo el caso.**
- El Sistema debe ser una aplicación web (utilizando Angular/React, Bootstrap, HTML5 y CSS3).
- La Base de Datos debe ser un archivo XML, Jason, Txt. \*¿SQL Server?
- La capa de servicios y la aplicación web deben estar desarrollados en C# y debe ser desplegada en Internet Information Services IIS (Windows).

5. **Entregables**

- 5.1. Manual de Usuario.
- 5.2. Documentación Técnica y del proyecto (descrita en el punto 6).
- 5.3. Documento de instalación.
- 5.4. Plan de Proyecto.
- 5.5. Aplicación WEB. ✓
- 5.6. API/Web Service/REST Service. ✓
- 5.7. Minutas. Bitacora ....

## 6. Documentación

6.1. Se deberá documentar el código fuente.

6.2. Se deberá entregar un documento que contenga:

6.2.1. Modelo conceptual utilizando la notación de Chen (**Tarea Corta 1**)

6.2.2. Modelo relacional (**Tarea Corta 2**).

6.2.3. Descripción de las estructuras de datos desarrolladas (**Entidades**).

6.2.4. Descripción detallada de la arquitectura desarrollada (**Tarea Corta 2**).

6.2.5. Problemas conocidos: En esta sección se detalla cualquier problema que no se ha podido solucionar en el trabajo.

6.2.6. Documentación de evidencia del trabajo en equipo (**Plan de proyecto**).

6.2.6.1. Metas del proyecto: Describir los entregables asociados al proyecto.

6.2.6.2. Roles: Describir los roles y responsabilidades utilizados en el desarrollo del proyecto.

6.2.6.3. Reglas: Describir las reglas principales que han sido definidas para el trabajo en equipo.

6.2.6.4. Cronograma: Plan de proyecto con las actividades planeadas, responsables de cada actividad, duración estimada y fechas de entregas estimadas.

6.2.6.5. *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.*

6.2.6.6. *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.*

6.2.7. Conclusiones

6.2.8. Recomendaciones del proyecto.

6.2.9. Bibliografía consultada en todo el proyecto.

6.3. Diagrama de clases y un documento que explique el porqué del diseño (**Tarea Corta 1**).

## 7. Evaluación

1. Las funcionalidades implementadas corresponderán a un valor de 70%, la documentación 20% y la defensa 10%. Cumplir con los requerimientos especificados en la documentación no significa que se tienen todos los puntos, dado que se evaluará que la documentación sea coherente, acorde al tamaño del proyecto y el trabajo realizado, no escatimen en documentación.
2. Cada grupo recibirá una nota en cada uno de los siguientes apartados Código, Documentación y Defensa.
3. 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 POO, uso de herramientas solicitadas, calidad de documentación interna y externa, trabajo en equipo.

4. No se revisarán funcionalidades parciales, ni funcionalidades no integradas.
5. 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.
6. De las notas mencionadas en el punto 1 se calculará la Nota Final del Proyecto.
7. Las citas de revisión oficiales serán determinadas por el profesor durante las lecciones o mediante algún medio electrónico.
8. Aún cuando el código y la documentación tienen sus notas por separado, se aplican las siguientes restricciones
  - 8.1. Si no se entrega documentación, automáticamente se obtiene una nota de 0.
  - 8.2. Si no se utiliza un manejador de código se obtiene una nota de 0.
  - 8.3. Si el código y la documentación no se entregan en la fecha indicada se obtiene una nota de 0.
  - 8.4. Si el código no compila se obtendrá una nota de 0, por lo cual se recomienda realizar la defensa con un código funcional.
  - 8.5. 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.
  - 8.6. El código debe ser desarrollado en C#, en caso contrario se obtendrá una nota de 0.
  - 8.7. No presentarse a la defensa se obtendrá una nota de 0.
9. 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.
10. 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.
11. Cada grupo es responsable de llevar los equipos requeridos para la revisión y estar listos con los ejecutables corriendo.
12. Durante la revisión únicamente podrán participar los miembros del grupo, asistentes, otros profesores y el coordinador del área.
13. Las revisiones se realizan con los estudiantes matriculados en el curso, cualquier persona fuera de estos y los mencionados en el punto 12, no pueden participar en la revisión.
14. Después de enviada la nota final del proyecto el estudiante tendrá un máximo de 3 días hábiles para presentar un reclamo siempre y cuando la funcionalidad esté completa.

## 8. Referencias

*Es obligatorio de una?*

**Bootstrap Themes & Templates** (2022-02-25). Recuperado de: <https://wrapbootstrap.com/>

**How to Write Doc Comments for the Javadoc Tool.** (2022-02-25). Recuperado de: <http://www.oracle.com/technetwork/articles/java/index-137868.html>

**C# Coding Conventions (C# Programming Guide).** (2022-02-25). Recuperado de: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions>

Anexos y estructuras (Ministerio de Hacienda). (2022-02-25). Recuperado de:  
<https://www.hacienda.go.cr/ATV/ComprobanteElectronico/frmAnexosyEstructuras.aspx#>

¿Para qué?

## Tarea Corta #2

Instituto Tecnológico de Costa Rica  
Área Académica Ingeniería en Computadores  
Bases de Datos (CE3101)



### Objetivo General

- Elaborar el modelado relacional de un caso utilizando **mapeo** del modelo Entidad-Relación al modelo relacional.

### Objetivos Específicos

- Describir el procedimiento mediante el cual se establece un modelo relacional haciendo un mapeo del modelo Entidad-Relacional.
- Elaborar un modelo relacional de base de datos utilizando como referencia el caso expuesto en el caso del Prototipo de la Tarea Corta y el modelo Entidad-Relación elaborado en la Tarea Corta #1.
- Elaborar el diagrama de la arquitectura utilizada para la implementación del prototipo de la Tarea Corta.

### Descripción de la tarea corta

Para el caso expuesto en el Prototipo de la Tarea Corta deben elaborar un modelo relacional utilizando como referencia el modelo Entidad-Relacional elaborado en la Tarea Corta #1.

Dado lo anterior, para esta tarea deben entregar:

- Descripción del mapeo realizado para obtener el Modelo Relacional a partir del Modelo Entidad-Relación elaborado en la Tarea Corta #1. Utilizar los pasos vistos en clase. Debe describir cómo aplicaron cada uno de los pasos a su modelo.
- Modelo Relacional resultante del punto anterior.

*→ mapeo conotoj*

*→ ¿Qué?*

Además, deben elaborar el diagrama de la arquitectura utilizado en la implementación del prototipo realizado en la Tarea Corta.

*→ arquitectura*

### Aspectos operativos y evaluación:

1. **Fecha de entrega:** [6/13]-Setiembre-2023.
2. **Valor:** 2%.
3. **Deben entregar un documento PDF** que contenga lo solicitado en la especificación de esta tarea.
4. El trabajo es **en grupos de cinco personas**. Los mismos grupos de la Tarea Corta #1 para darle continuidad al trabajo realizado.

## Tarea Corta #1

Instituto Tecnológico de Costa Rica  
Área Académica Ingeniería en Computadores  
Bases de Datos (CE3101)



### Objetivo General

- Elaborar el modelado conceptual de un caso mediante un modelo Entidad-Relación para la base de datos y un diagrama de clases para la descripción del contexto del problema.

### Objetivos Específicos

- Elaborar un modelo Entidad-Relación utilizando la notación de Chen que permita la descripción del modelo conceptual de la base de datos del caso expuesto en el prototipo de la Tarea Corta.
- Elaborar un diagrama de clases que permita la descripción del contexto del problema descrito en el prototipo de la Tarea Corta.

### Descripción de la tarea corta

Para el caso expuesto en el prototipo de la Tarea Corta deben elaborar un modelo Entidad-Relación utilizando la notación de Chen para describir el modelo conceptual de la base de datos del caso expuesto en el Prototipo.

También deben elaborar un diagrama de clases que permita describir el contexto del problema planteado en el caso descrito en el prototipo.

### Aspectos operativos y evaluación:

1. **Fecha de entrega:** 30-Agosto-2023.
2. **Valor:** 2%
3. **Deben entregar un documento PDF** que contenga lo solicitado en la especificación de esta tarea.
4. El trabajo es **en grupos de cinco personas**. Los mismos grupos del prototipo para darle continuidad al trabajo realizado.



