



## Cátedra Tecnología de Sistemas

[Programación Avanzada con C#]

Código: [00830]

### Proyecto 2. Valor 20%

#### Temas de Estudio

1. Tema 1 Introducción al lenguaje de programación
2. Tema 2 Particularidades del lenguaje C#
3. Tema 3 Manejo de excepciones en C#
4. Tema 4 Conceptos de Interfaz Gráfica
5. Tema 5 Subprocesamiento múltiple
6. Tema 6 Bases de datos y componentes ADO.NET
7. Tema 7 Colecciones
8. Tema 8 Redes

#### Objetivo

Poner en práctica los conceptos subprocesamiento múltiple y redes, así como retomar los conceptos ya aprendidos de base de datos y colecciones.

#### Software de Desarrollo

Visual Studio Community 2022 (C#, .Net Framework 4.8 o Net 6.0)

#### Desarrollo

A los encargados de la videoteca de la Universidad Estatal a Distancia están contentos con la primera etapa de la automatización de sus procesos, sin embargo, han decidido hacer algunos cambios como almacenar de manera permanente la información en una base de datos SQL Server, además quieren poner en funcionamiento lo antes posible el préstamo de las películas en línea. Ahora bien, para lograr la incorporación de esta nueva funcionalidad debe implementarse una solución cliente/servidor que permita a los clientes conectarse mediante una aplicación cliente al servidor central para solicitar los préstamos.

El sistema por desarrollar debe estar compuesto mínimo por dos proyectos; un cliente y un servidor, ambos deben comunicarse a través de la red mediante el protocolo TCP. El proyecto cliente será utilizado por los clientes para registrar los préstamos de películas, mientras que el proyecto servidor será utilizado por un único administrador en las oficinas centrales para dar mantenimiento a las

funcionalidades de Categoría de Película, Película Encargado, Sucursal, Película por Sucursal y Cliente. Toda la información debe ser almacenada en una base de datos SQL Server.

**El proyecto cliente debe cumplir con los siguiente:**

#### **Conexión y Validación.**

- Debe ser utilizado por los clientes para conectarse con el servidor central y realizar préstamos.
- Debe tener una opción para conectarse y desconectarse del servidor.
- Al iniciar el cliente se debe solicitar el Id de cliente o el número de identificación.
  - Se debe validar que el cliente exista en base de datos antes de mostrar las funcionalidades de la aplicación cliente.
  - La aplicación cliente envía al servidor la solicitud de verificación de la persona usuaria cliente que ingresó (que se encuentre registrada en la Base de Datos) y además que se encuentre activa, eso quiere decir que el cliente debió haber sido registrado previamente por el administrador en el servidor.
  - Si la identificación existe, es decir, es un cliente registrado y activo se permite el ingreso a la aplicación cliente.
- Una vez que el cliente ha sido validado, en la aplicación cliente se debe mostrar el nombre del cliente con los dos apellidos, Además, una vez que el cliente ha sido validado, en la aplicación cliente se muestran desbloqueadas las funcionalidades de la aplicación.

#### **Préstamo de películas.**

- Se debe mostrar una opción que permita al cliente realizar una solicitud de préstamo de película. Para esto el sistema debe crear dicha solicitud (Préstamo). Para esto se debe realizar lo siguiente:
  - Se debe enviar solicitud al servidor para consultar las sucursales, esta información se debe mostrar en un control "ComboBox", o bien, en un control "DataGridView".
    - Solo se deben mostrar las sucursales activas.
  - Una vez seleccionada la sucursal se debe enviar solicitud al servidor para consultar todas las películas asociadas a esta. La información de dichas películas disponibles se debe mostrar en un control "DataGridView".

- La aplicación debe permitir que el cliente seleccione solo una película por préstamo.
- Finalmente, se debe permitir que la persona usuaria solicite el préstamo.
- En esta funcionalidad no se selecciona el cliente, dado que el cliente seleccionado es el que está ejecutando el programa. La aplicación cliente solo puede registrar préstamos del cliente que ejecuta la aplicación.
- Al realizar la “Solicitud de Préstamo” debe enviarse al servidor, mediante la red. El servidor es el encargado de hacer las validaciones de negocio y los cambios en base de datos.
  - Debe validar que el cliente no tenga actualmente un préstamo de la misma película pendiente de devolver, caso contrario indicarle a la persona usuaria que no se le podrá realizar el préstamo.
  - Debe validar que la película seleccionada en la sucursal seleccionada tenga la “cantidad” de inventario mayor a 0, caso contrario indicarle a la persona usuaria que no se le podrá realizar el préstamo.
  - Si se realiza el préstamo se debe **actualizar** el inventario de la película en la sucursal seleccionada restando en 1 la “cantidad” actual.
- En el servidor se debe implementar semáforos u otro mecanismo de sincronización para evitar que dos clientes ingresen al mismo tiempo al recurso compartido en el momento de hacer el préstamo (es decir a la Base de Datos), esto impedirá hacer un sobre préstamo y que la “cantidad” del inventario quede en negativo.
- Se debe validar los datos en todos los pasos y las posibles excepciones.
- La información por registrar es la siguiente:
  - Id Préstamo (int) Se generará de forma automática en la Base de Datos
  - Id Cliente (int)
  - Id Sucursal (int)
  - Id Pelicula (int)
  - Fecha Préstamo (datetime) Fecha actual al momento de realizar el préstamo.
  - Pendiente devolución (bool) Se asignará el valor “false”
- Mostrar en pantalla si el proceso fue exitoso.

**Consultar préstamos.**

- Se debe mostrar una opción que permita al cliente visualizar la información de los préstamos que ha realizado.
  - El programa solo mostrará los préstamos realizados al cliente que está ejecutando la aplicación cliente, no deberá mostrar préstamos realizados por otros clientes.
  - Debe enviar la solicitud de consultar préstamos del cliente hacia el servidor a través de la red.
  - El servidor es quien consulta la base de datos y devuelve la información al cliente a través de la red.
  - El programa muestra en pantalla los préstamos realizados por el cliente.
  - La información por presentar debe ser los datos completos del préstamo (Id préstamo, Fecha préstamo, Pendiente devolución, Id Película, Título, Nombre de la categoría, Año de Lanzamiento, Idioma).

**La aplicación cliente no tiene conexión directa a la base de datos, todas sus solicitudes deben pasar a través de la aplicación servidor, este último es el que tiene conexión hacia la base de datos.**

**El proyecto servidor debe cumplir con los siguiente:**

- Debe implementar un servidor que escuche y responda las solicitudes de red de múltiples clientes TCP de forma simultánea utilizando subprocesamiento múltiple.
- Debe utilizar sockets en la dirección 127.0.0.1 y el puerto 15500 para escuchar solicitudes.
- La aplicación cliente debe almacenar toda la información en base de datos SQL Server
- El servidor almacena en la base de datos toda la información indicada desde la aplicación cliente por las personas usuarias clientes.
- El servidor consulta y almacena en la base de datos la información solicitada por los clientes.
- **Solo la aplicación servidor se comunica con la base de datos, la aplicación cliente no tiene conexión con la base de datos.**
- El servidor debe ser utilizado por un único usuario administrador en las oficinas centrales, el cual no requiere registrarse, tampoco requiere usuario y contraseña.
- Debe tener una pantalla donde muestre en bitácora todos los eventos enviados por todos los clientes conectados en tiempo real, como, por ejemplo: conexión al servidor, desconexión, registro de préstamo, consulta de préstamo.

El servidor debe contar con las opciones de:

- Registrar y consultar Categoría de Película.
- Registrar y consultar Película.
- Registrar y consultar Encargado.
- Registrar y consultar Sucursal.
- Registrar y consultar Película por Sucursal.
- Registrar y consultar Cliente.

### **Registrar Categoría Película**

Esta opción le permitirá a la persona usuaria registrar o agregar los datos de la categoría a la que puede pertenecer una película, por ejemplo: podcast, documental, acción, etc.

Se debe poder registrar la siguiente información:

- Id de categoría (int), este valor debe ser único por lo que se debe validar en el sistema que no existan valores repetidos.
- Nombre de la categoría(string)
- Descripción (string)

Una vez capturada la información se debe agregar en la tabla "CategoriaPelícula".

### **Registrar Película**

Esta opción le permitirá a la persona usuaria registrar o agregar los datos de una película.

Se debe poder registrar la siguiente información:

- Id de película (int), este valor debe ser único por lo que se debe validar en el sistema que no existan valores repetidos.
- Título(string)
- Categoría película (Objeto Categoría Película) Debe validar que por lo menos exista registrado una categoría de película.
- Año de lanzamiento (int)
- Idioma (string)

Una vez capturada la información se debe agregar en la tabla "Película".

### **Registrar Encargado**

Esta opción le permitirá a la persona usuaria registrar o agregar los datos de una persona encargada de administrar una sucursal.

Se debe poder registrar la siguiente información:

- Id Encargado (int) este valor debe ser único por lo que se debe validar en el sistema que no existan valores repetidos.
- Identificación(string), este valor debe ser único por lo que se debe validar en el sistema que no se vaya a registrar un cliente con la misma identificación y distinto Id Encargado.
  - Debe verificar que la persona con esta identificación exista en la base de datos en la tabla "Persona" antes de registrarlo como Encargado, caso contrario deberá ingresarlo primero a la tabla "Persona".
- Nombre(string)
- Primer apellido(string)
- Segundo apellido(string)
- Fecha de nacimiento (Datetime)
- Fecha de ingreso (Datetime)

Una vez capturada la información se debe agregar en la tabla "Persona" y "Encargado".

### **Registrar Sucursal**

Esta opción le permitirá a la persona usuaria registrar o agregar los datos de una sucursal.

Se debe poder registrar la siguiente información:

- Id de sucursal (int), este valor debe ser único por lo que se debe validar en el sistema que no existan valores repetidos.
- Nombre(string)
- Encargado (Objeto Encargado) Debe validar que por lo menos exista registrado un Encargado.
- Dirección (string)
- Teléfono (string)
- Activo (bool), Sí o No, (mostrar en "Combobox")

Una vez capturada la información se debe agregar en la tabla "Sucursal".

### **Registrar Cliente**

Esta opción le permitirá a la persona usuaria registrar o agregar los datos de una persona cliente.

Se debe poder registrar la siguiente información:

- Id Cliente (int) este valor debe ser único por lo que se debe validar en el sistema que no existan valores repetidos.
- Identificación(string), este valor debe ser único por lo que se debe validar en el sistema que no se vaya a registrar un cliente con la misma identificación y distinto Id Cliente.
  - Debe verificar que la persona con esta identificación exista en la base de datos en la tabla "Persona" antes de registrarlo como cliente, caso contrario deberá ingresarlo primero a la tabla "Persona".
- Nombre(string).
- Primer apellido(string).
- Segundo apellido(string).
- Fecha de nacimiento (Datetime).
- Fecha de registro.
- Activo (bool), Sí o No, (mostrar en "Combobox")

Una vez capturada la información se debe agregar en la tablas "Persona" y "Cliente".

### **Registrar Película por Sucursal**

Funcionalidad que debe permitir al usuario registrar (asociar) las películas que estarán disponibles en cada Sucursal. Para ello se debe crear una clase "PeliculaxSucursal".

Para esta opción se debe poder registrar la siguiente información para Película por Sucursal:

- Sucursal (Objeto Sucursal), este valor en combinación con Película debe ser únicos por lo que se debe validar en el sistema que no existan valores repetidos.
- Película (Objeto Película), este valor en combinación con Sucursal debe ser únicos por lo que se debe validar en el sistema que no existan valores repetidos.
- Cantidad (int)

Una vez capturada la información se debe agregar en la tabla "PeliculaxSucursal".

Para esto se debe realizar lo siguiente:

- Mostrar en pantalla en un “ComboBox” las sucursales disponibles.
  - Se debe validar que al menos exista una sucursal registrada para realizarle asociación.
  - Para cada sucursal registrada debe mostrar los siguientes datos: Nombre
  - Solo debe mostrar las sucursales que se encuentran activas.
- Permitir a la persona usuaria seleccionar desde “ComboBox” la sucursal.
- Mostrar en un control “DataGridView” las películas disponibles para asignar a la sucursal.
  - Se debe validar que al menos exista una película registrada para ser asociada.
  - Para cada película de la lista se debe mostrar toda la información de los objetos: Id, título, categoría, año de lanzamiento, idioma.
- La persona usuaria puede seleccionar uno o más películas del control “DataGridView” para asociar a la sucursal.
- La persona usuaria debe contar con un control de entrada de datos “TextBox”, donde podrá definir la cantidad de inventario que contará de una película.
  - Se debe validar que la cantidad ingresada sea mayor a cero.
  - Si se selecciona varias películas a asociar a todas se les asignará la misma cantidad de inventario.
- Al momento de realizar la asociación deberá validar que no se haya ingresado en la base de datos anteriormente la “combinación” de una sucursal y una película.

### **Consultar Categoría Película**

Esta opción le permitirá a la persona usuaria visualizar en pantalla todas las categorías de películas registradas, para tal fin debe hacer uso del control “DataGridView”, se debe mostrar todos los atributos.

### **Consultar Película**

Esta opción le permitirá a la persona usuaria visualizar en pantalla todas las películas registradas, para tal fin debe hacer uso del control “DataGridView”, se debe mostrar todos los atributos incluso los datos de la categoría a la que pertenece.



**Consultar Encargado**

Esta opción le permitirá a la persona usuaria visualizar en pantalla todos los encargados registrados, para tal fin debe hacer uso del control “DataGridView”, se debe mostrar todos los atributos.

**Consultar Sucursal**

Esta opción le permitirá a la persona usuaria visualizar en pantalla todas las sucursales registradas, para tal fin debe hacer uso del control “DataGridView”, se debe mostrar todos los atributos incluso los datos del encargado asignado.

**Consultar Cliente**

Esta opción le permitirá a la persona usuaria mostrar en pantalla todos los clientes registrados, para tal fin debe hacer uso del control “DataGridView”, se debe mostrar todos los atributos.

**Consultar Película por Sucursal**

Esta opción le permitirá a la persona usuaria visualizar en pantalla todas las asociaciones registradas, para tal fin debe hacer uso del control “DataGridView”, se debe mostrar todos los atributos de la sucursal y las películas.

## Consideraciones técnicas

Uso de comentarios:

Se deberá implementar el uso de comentarios cuando se realiza la codificación del programa, en la parte superior indicar la universidad, el cuatrimestre, el nombre y la descripción del proyecto, el nombre del estudiante y la fecha en que está realizando la programación. Lo anterior debe estar en cada archivo cs (clase, program, etc) que se realiza el código de programación; como se puede observar en el siguiente ejemplo:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  /* UNED III Cuatrimestre
8   * Proyecto XX: Descripción del Proyecto
9   * Estudiante:
10  * Fecha:
11
12  */
13
14  namespace CapaEntidades
15  {
16  }
```

Debe utilizar POO (Programación orientada a objetos) para resolver el problema.

- Las clases de objetos que usted considere como entidades (Categoría Película, Película, Encargado, Sucursal, Cliente, Película por Sucursal, Préstamo, etc.) no deben contener lógica para solicitar información a la persona usuaria, solo debe tener la estructura de la clase y sus propiedades.
- Implementar en la Solución proyectos independientes con el fin de tener las distintas capas que usted considere: Capa de Entidades, Capa de Acceso a Datos, Capa de Lógica Negocio, Capa de Presentación (Interfaz del Usuario) etc.
- Cada clase debe ser creada en archivo por separado
- Las clases no deben tener métodos vacíos y recuerde utilizar nombres significativos a sus variables.

Debe implementar el manejo de excepciones

- Si ocurre una excepción, el sistema no debe cerrarse, se debe mostrar un mensaje al usuario y manejar la excepción de forma apropiada.

### Uso de colecciones

- Se deben usar colecciones genéricas (List<>) cuando se requiera consultar de base de datos y retornar más de un registro.
- Las colecciones genéricas se usan como "DataSource" en los controles "combobox" y "DataGridView".
- **No use "DataSet" ni "DataTable".**

### Interfaz de usuario

- Debe usar interfaz de usuario con formularios GUI (Proyecto Windows Form).

### Base de datos

- Debe hacer el manejo de la tabla "persona" de la manera adecuada, lo cual permitirá el registro de una persona una única vez independientemente si es un encargado o un cliente o podría llegar hacer ambos.
- Debe almacenar y consultar la información en una base de datos SQL Server. No se permite otro motor de base de datos.
- Para la cadena de conexión use seguridad integrada de Windows y un archivo de configuración "App.config".
- **El script de creación de la base de datos se publicará en la plataforma. No de be modificar el script de base de datos**

### Otras consideraciones

- La solución será probada utilizando al menos 5 aplicaciones clientes simultáneamente y una única instancia de la aplicación servidor.
- Se recomienda crear un tercer proyecto de entidades tipo biblioteca de clases para las entidades a utilizar en la solución, de esa manera no tendrá que duplicar las clases de entidades en el proyecto cliente y servidor, en estos solo se agrega la referencia o dll.
- Revise constantemente los foros de consulta y el foro de interés, ahí se aclaran dudas del proyecto o se realizan anuncios que le pueden ser de mucha utilidad en el desarrollo del proyecto.
- Los proyectos en donde se identifique una copia textual o similar, tanto del material digital, como de código fuente, serán anulados y se aplicará el reglamento para dichos casos. Si usa código fuente que no es de su autoría debe indicarlo, aunque sea código del tutor.

- Debe presentar un pequeño manual con los pasos para probar la solución entregada.

### Diagrama cliente/servidor

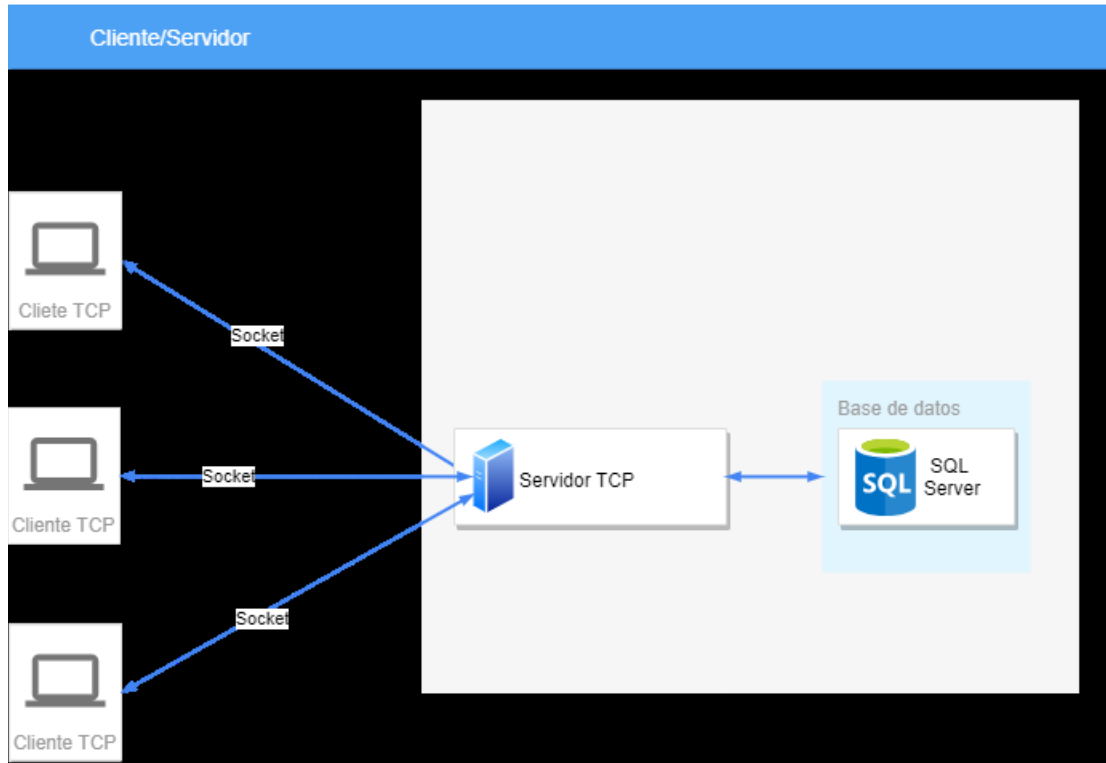
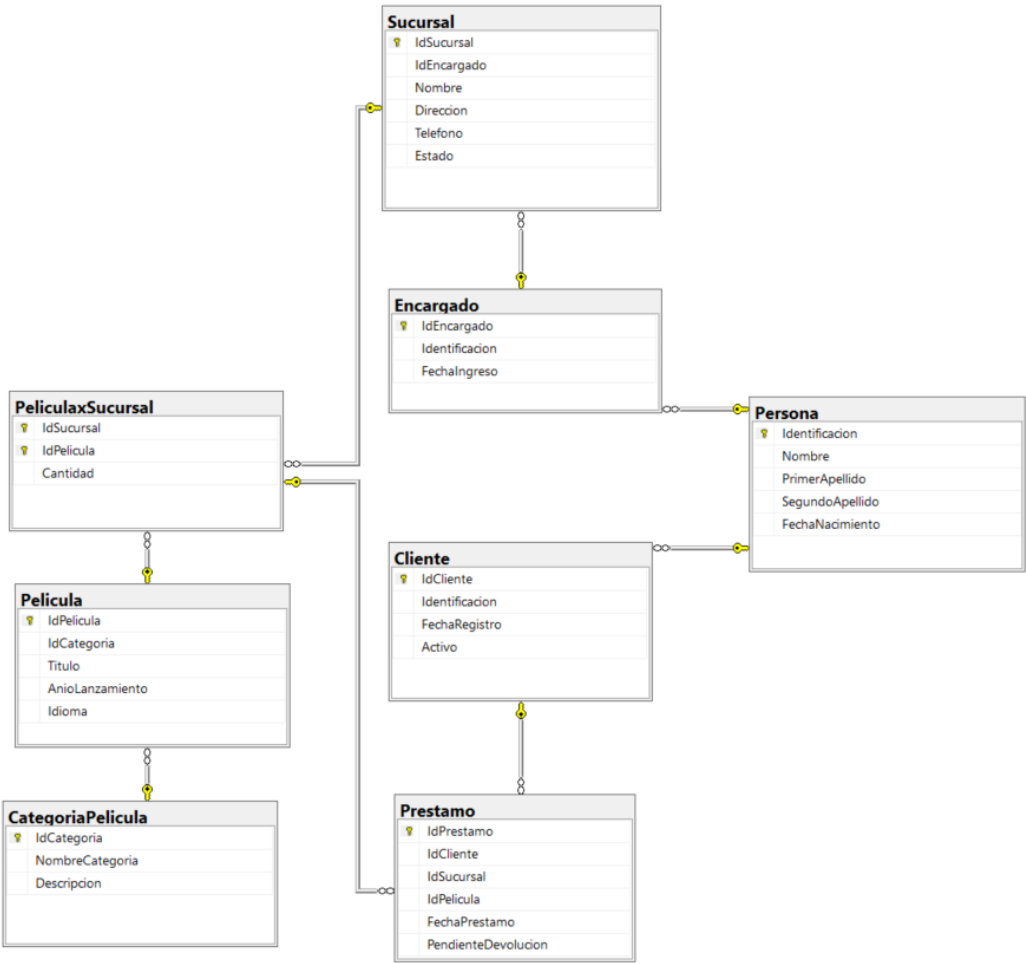


Diagrama base de datos



## Honestidad Académica



<https://audiovisuales.uned.ac.cr/play/player/23048>

### Nota Importante

Cada estudiante es responsable del contenido que entrega, si no es el archivo correcto, no podrá entregarlo posterior a la fecha establecida.

Si el contenido del archivo coincide con algún otro estudiante, o se comprueba que no es de su autoría, se aplicaría lo indicado en la plataforma en el documento [Lineamientos ante casos de plagio](#)

## Indicaciones Importantes

- Es obligatorio que incluya todo el directorio donde se encuentra el proyecto 2.
- El **proyecto 2** debe estar desarrollado en **Visual Studio Community 2022** que es la herramienta oficial del curso.
- El programa debe ser modular, utilizando de la mejor manera funciones definidas por usted.
- **Los trabajos deben realizarse en forma individual. Dentro del código del programa debe de indicar la documentación que explique cómo fue realizado el programa.**
- **Si utiliza código de algún ejemplo del libro, del algún profesor o de otra fuente que no sea de su autoría, debe de indicarlo.**
- Comprima todos los archivos en un solo archivo .zip o .rar.
- **Nombre del archivo que envía:** debe ser nombre y primer apellido del estudiante, y nombre de la tarea. **Ejemplo: JuanRojas-proyecto2.**
- La entrega del proyecto las fechas establecidas en la plataforma de aprendizaje en línea Moodle en el apartado que se indique.
- Si no concluyó a tiempo la tarea, debe entregar lo que pudo hacer e incluir una carta explicando las razones por las cuales no finalizó.

**Rúbrica de Evaluación**

<b>Criterio</b>	<b>Cumple a satisfacción lo indicado en la evaluación</b>	<b>Cumple medianamente en lo indicado en la evaluación</b>	<b>Cumple en contenido y formato, pero los aportes no son significantes</b>	<b>No cumple o no presenta lo solicitado</b>
Presentación del código: Correcto uso de variables, nombres de métodos significativos, no presenta métodos vacíos.	5	3	2	1
Registra Categoría de película y película en el servidor	5	3	2	1
Registra Encargado y Sucursal en el servidor	5	3	2	1
Registrar Cliente y Películas por Sucursal en el Servidor	10	7	3	1
Consultar Sucursal, Películas y Películas por Sucursal; en el servidor.	10	7	3	1
Conexión de la aplicación cliente con el servidor mediante TCP y administración de las conexiones en el servidor.	10	7	3	1
Validación del usuario cliente en la aplicación cliente.	5	3	2	1
Realizar solicitud de préstamo en la aplicación cliente	10	7	2	1
Consulta de préstamos en la aplicación cliente	10	7	2	1
En la aplicación servidor se muestran los eventos enviados por todos los clientes conectados.	5	3	2	1

Correcto uso de subprocesamiento múltiple en servidor.	5	3	2	1
Implementación de un método para administrar la concurrencia en el servidor al realizar un préstamo y evitar un sobre préstamo.	5	3	2	1
Correcto uso del manejo de excepciones.	5	3	2	1
Interfaz de usuario es fácil de usarse y presenta buen diseño.	5	3	2	1
Manual de uso de la solución	5	3	2	1
TOTAL	100			