



Cátedra Tecnología de Sistemas

Programación Avanzada

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), SQL Server, SQL Management Studio

Desarrollo

A la empresa **RESORTS-UNED** le ha ido muy bien con el seguimiento y gestión de sus hoteles, ya que el sistema de información desarrollado les ha brindado muchas facilidades, por lo que nuevamente le han contratado para hacer mejoras en su sistema de información, y que la información sea almacenada de forma permanente en una base de datos SQL Server. A la fecha los usuarios no pueden realizar pedidos de artículos de acuerdo con los hoteles. Por lo anterior, se requiere incorporar nuevas funcionalidades que permitan realizar el registro de solicitudes de alquiler de artículos. 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 registrar los pedidos de artículos.

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 pedidos de artículos, mientras que el proyecto servidor será utilizado por un único administrador en las oficinas centrales para dar mantenimiento a las funcionalidades de hotel, clientes, categoría de artículo, artículo y artículo por hotel. Toda la información debe ser almacenada en una base de datos SQL Server.

El proyecto cliente debe cumplir con los siguiente:

- Debe ser utilizado por los clientes para conectarse con el servidor central y realizar reservas.
- Debe tener una opción para conectarse y desconectarse del servidor.
- Al iniciar el cliente se debe solicitar el número de identificación del cliente
 - 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 identificación del usuario cliente que ingresó.
 - El cliente debe haber sido registrado previamente por el administrador en el servidor.
 - Si la identificación existe, es decir, es un cliente registrado 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, esta información debe ser siempre visible en la aplicación cliente. Además, una vez que el cliente ha sido validado, en la aplicación cliente se muestran desbloqueadas las funcionalidades de la aplicación.

Realizar pedido de artículos: Se debe mostrar una opción que permita al cliente realizar el pedido de artículos. Para esto el sistema debe crear el pedido de artículos (PedidoArtículos). Para esto se debe realizar lo siguiente:

- Se debe enviar solicitud al servidor para consultar los hoteles, esta información se debe mostrar en un control ComboBox, o bien, en un control DataGridView.
 - Solo se deben mostrar los hoteles activos
- Una vez seleccionado el hotel se debe enviar solicitud al servidor para consultar todos los artículos asociados al hotel seleccionado. La información de artículos disponibles se debe mostrar en un control DataGridView.
- La aplicación debe permitir que el cliente seleccione más de un artículo.
- Finalmente, de acuerdo con los artículos seleccionados se debe indicar en pantalla la información del pedido, es decir: artículo (id artículo, nombre, precio y costo total del pedido; así como, permitir que el usuario realice el pedido.
- 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 pedidos del cliente que ejecuta la aplicación.
- Al hacer clic en el botón “Realizar Pedido” debe enviarse al servidor, mediante la red. El servidor es el encargado de hacer los cambios en base de datos.
- En el servidor se debe implementar semáforos u otro mecanismo de sincronización para evitar que dos clientes ingresen al mismo tiempo al recurso compartido, es decir, la base de datos.
- Se debe validar los datos en todos los pasos y las posibles excepciones.
- La información por registrar es la siguiente:
 - Id Pedido (int) Se debe validar que no existan id's repetido. Se debe generar de forma automática
 - Id Artículo (int)
 - Id Cliente (int)
 - Fecha pedido (DateTime)
- Mostrar en pantalla si el proceso fue exitoso.

Consultar mis pedidos: Está opción le permite al cliente consultar desde la aplicación cliente los pedidos realizados.

- Está opción solo muestra los pedidos realizados por el cliente que está ejecutando la aplicación cliente, no debe mostrar pedidos realizadas por otros clientes.

- Debe enviar la solicitud de consultar pedidos 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 cliente muestra en pantalla los pedidos realizados por el cliente.
- La información por presentar no debe ser el código de cliente, sino el nombre y apellidos del cliente, Id Artículo, nombre y precio del artículo, el id de pedido y la fecha de pedido, monto total del pedido
- Se debe implementar dos formas (funcionalidades) de consultar pedidos, una pudiendo consultar todos los pedidos realizados por el cliente; la otra consultando pedido específicamente mediante el id del pedido.
- **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 de cumplir con lo 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 13200 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 los usuarios 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 principal 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 pedido, consulta de pedido.
- Siempre debe ser visible la cantidad de clientes conectados.

El servidor debe contar con un menú principal con las siguientes opciones:

- **Registrar Hotel**
- **Registrar Categoría Artículo**
- **Registrar Artículos**
- **Registrar Clientes**
- **Registrar Artículo por Hotel**
- **Consultar Hotel**
- **Consultar Categoría Artículo**
- **Consultar Artículo**
- **Consultar Clientes**
- **Consultar Artículos por Hotel**

Registrar Hotel

Esta opción del menú le permite al usuario registrar o agregar los datos relacionados con los hoteles con que cuenta la empresa. Se debe poder registrar la siguiente información:

- Id de hotel (int), este valor debe ser único por lo que se debe validar en el sistema que no existan valores repetidos
- Nombre del hotel (string)
- Dirección (string)
- Estado (bool), activo o inactivo, (mostrar en Combobox)
- Teléfono (string)

Una vez capturada la información se debe agregar en tabla "Hotel" de la base de datos

Registrar Clientes

Debe permitir al usuario registrar los clientes del restaurante, pudiendo registrar la siguiente información:

- Identificación (string) se debe validar que no existan identificaciones repetidas
- Nombre (string)

- Primer apellido (string)
- Segundo apellido (string)
- Fecha de nacimiento (DateTime)
- Género (char)

Una vez capturada la información se debe guardar en tabla "Cliente" de la base de datos

Registrar Categoría de Artículo

Debe permitir al usuario registrar las categorías de artículos con que cuenta la empresa, pudiendo registrar la siguiente información:

- Id categoría (int), este valor debe ser único por lo que se debe validar en el sistema que no existan valores repetidos
- Descripción (string)
- Estado (bool), activo o inactivo, (mostrar en Combobox)

Una vez capturada la información se debe guardar en tabla "CategoriaArticulo" de la base de datos

Registrar Artículos

Debe permitir al usuario registrar los artículos con que cuenta la empresa, pudiendo registrar la siguiente información:

- Id artículo (int), este valor debe ser único por lo que se debe validar en el sistema que no existan valores repetidos
- Nombre (string)
- Precio (int)
- Id Categoría (int)

Una vez capturada la información se debe guardar en tabla "Articulo" de la base de datos

Registrar Artículos por Hotel

Funcionalidad que debe permitir al usuario registrar (asociar) los artículos que están disponibles en cada hotel. Para ello se debe crear una clase "ArticuloHotel", la cual debe contener además de los atributos de id asignación, fecha de afiliación, hotel, debe contener un arreglo de los artículos asignados a un hotel. Para esto se debe realizar lo siguiente:

- Mostrar en pantalla en un ComboBox los hoteles disponibles.

- Se debe validar que al menos exista un hotel registrado para realizarle asignación
 - Para cada hotel registrado debe mostrar los siguientes datos: Id hotel, Nombre, Dirección
 - Solo debe mostrar los hoteles activos
- Permitir al usuario seleccionar desde ComboBox el hotel
- Mostrar en un control DataGridView los artículos disponibles para asignar al hotel
 - Se debe validar que al menos exista un artículo registrado para ser asociado
 - Para cada artículo de la lista se debe mostrar la siguiente información: id artículo, nombre, precio
- El usuario puede seleccionar uno o más artículos del control DataGridView para asociar al hotel.
- Se debe preguntar al usuario si desea agregar más artículos al hotel. Si la respuesta es afirmativa, entonces se vuelve a mostrar la lista de artículos para que el usuario nuevamente seleccione e indique un id.
- La clase ArtículoHotel debe tener un arreglo de 10 posiciones donde se agreguen los artículos
- En la parte inferior de esta pantalla, se debe mostrar en un control DataGridView los artículos (id artículo, nombre, precio) que han sido asociados al hotel seleccionado inicialmente, es decir, cada vez que se asocia un artículo al hotel, estos deben ser mostrada en la parte inferior.

Para esta opción se debe poder registrar la siguiente información para Artículo Hotel:

- Id asignación (int)
- Fecha (DateTime)
- Id Hotel (int)
- Id Artículo(int)

La asociación de artículos a hoteles se debe guardar en tabla "ArticuloHotel" de la base de datos

Consultar Hoteles

En esta opción debe mostrar en pantalla en un control DataGridView todos los hoteles registrados, para cada hotel se debe mostrar todos los atributos

Consultar Clientes

En esta opción debe mostrar en pantalla en un control DataGridView los clientes registrados, para cada cliente se debe mostrar todos los atributos

Consultar Categoría de Artículo

Esta opción permite al usuario mostrar en pantalla en un control DataGridView las categorías de artículos registrados, se debe mostrar todos los atributos

Consultar Artículos

Esta opción permite al usuario mostrar en pantalla en un control DataGridView todos los artículos registrados, para cada artículo se debe mostrar todos los atributos

Consultar Artículos por Hotel

Esta opción permite consultar los artículos que tiene asociados un hotel, es decir, consulta los artículos que están asignados a un hotel. Debe imprimir en pantalla la información de cada asociación, es decir, id asignación, fecha afiliación, datos de hotel (id hotel, nombre, dirección), la información de los artículos (id artículo, nombre, precio) que están asignados al hotel. Puede usar un o más DataGridView o el control de interfaz gráfica que permita mostrar la información de la asociación de hotel y el listado de artículos asociados.

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 (-Hotel, Cliente, Categoría Artículo, Artículo, Artículo Hotel) no deben contener lógica para solicitar información al usuario, 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: Capa de Entidades, Capa de Lógica Negocio y Capa de Presentación (Interfaz del Usuario).
- 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.

Base de datos

- 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.
- **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

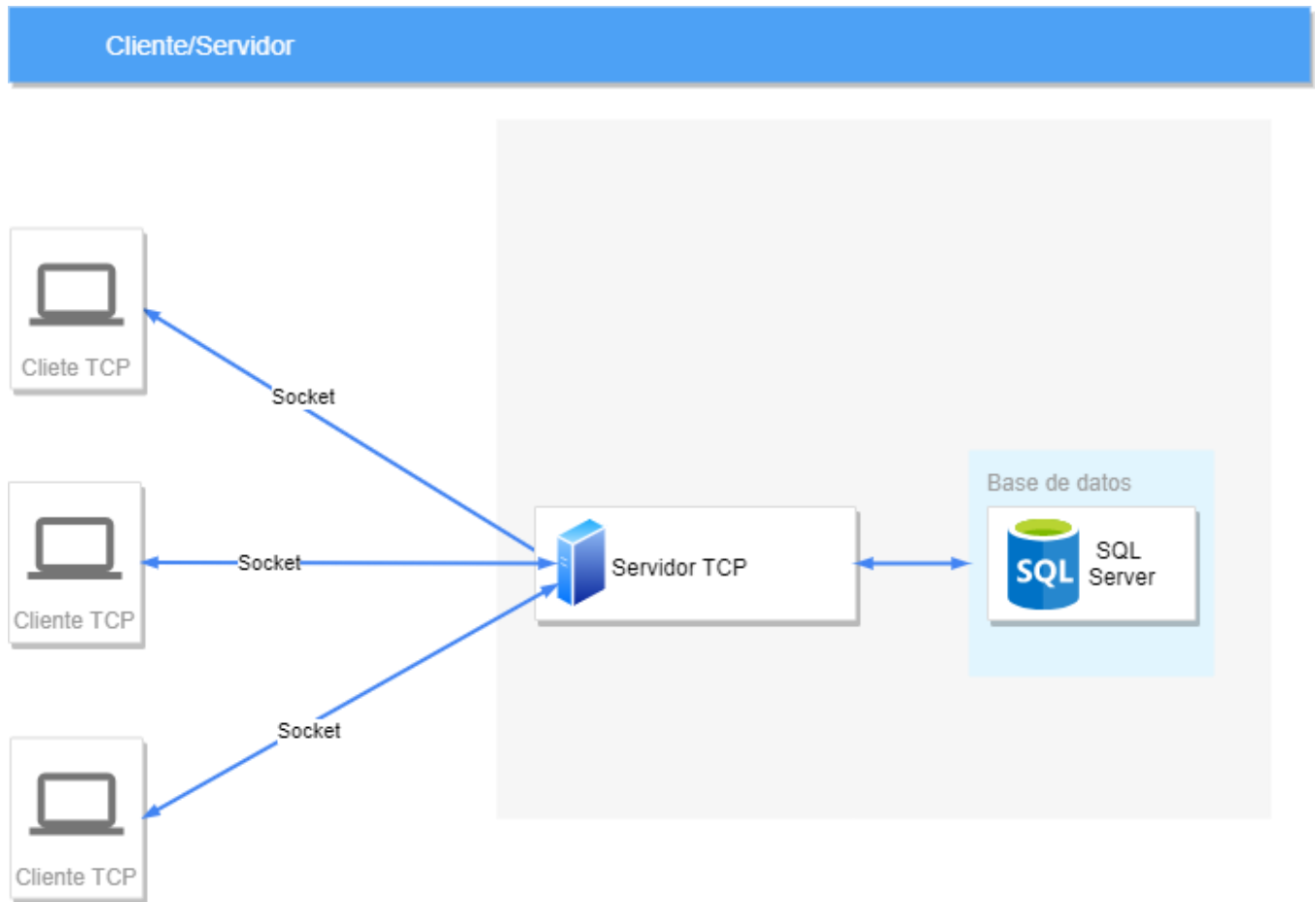
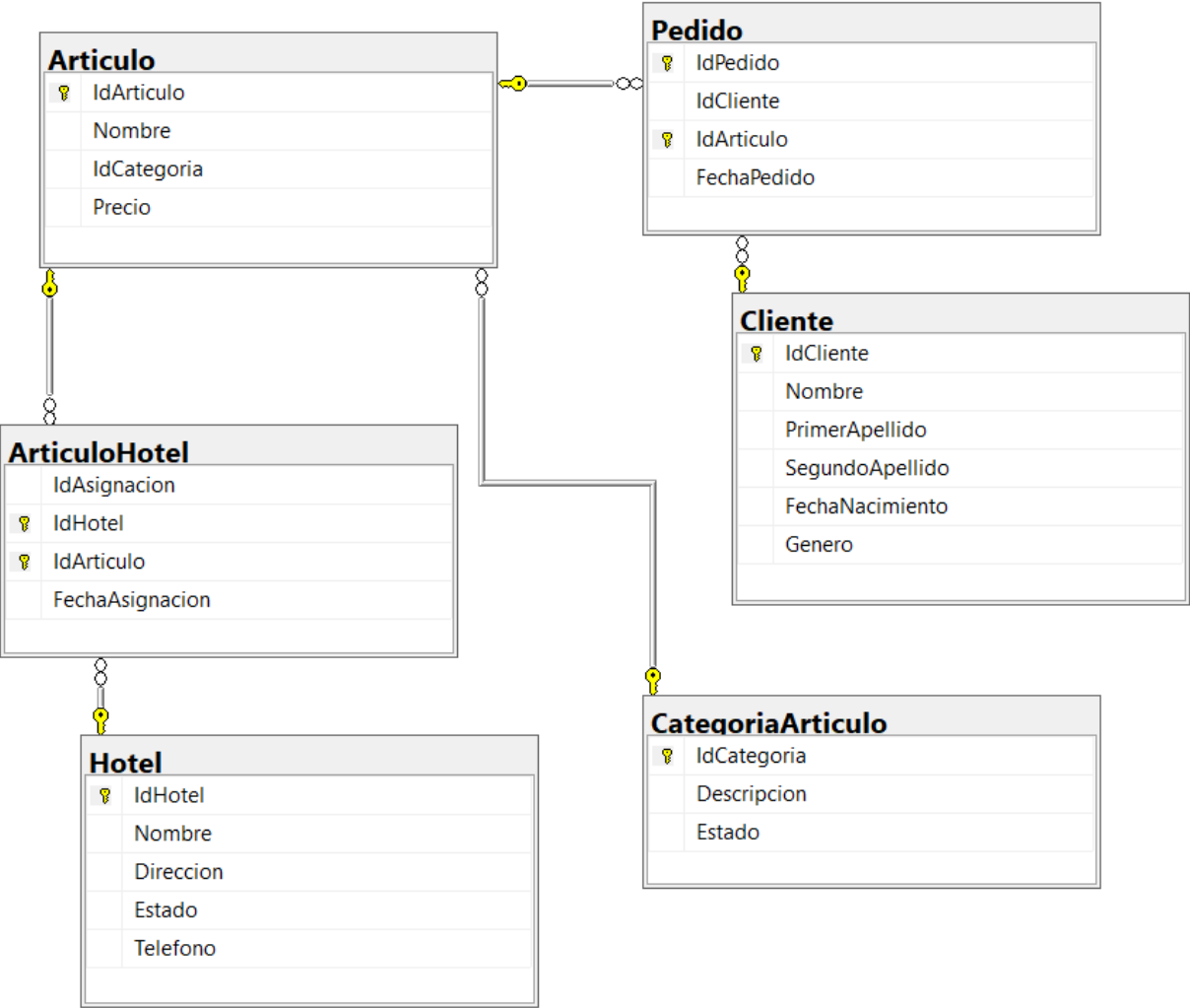


Diagrama base de datos



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 expone a las sanciones indicadas 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 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 en 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

Criterio	Cumple a satisfacción lo indicado en la evaluación	Cumple medianamente en lo indicado en la evaluación	Cumple en contenido y formato, pero los aportes no son significantes	No cumple o no presenta lo solicitado
Presentación del código: Correcto uso de variables, nombres de métodos significativos, no presenta métodos vacíos.	3	2	1	0
Registrar hotel y clientes en el servidor	5	3	1	0
Registrar categoría de artículo y artículos en el servidor.	10	5	2	0
Registrar artículos por hotel; en el servidor	5	3	1	0
Consultar hotel, clientes y artículo por hotel; en el servidor	10	3	1	0
Conexión de la aplicación cliente con el servidor mediante TCP y administración de las conexiones en el servidor	10	5	2	0
Validación del usuario cliente en la aplicación cliente	5	3	1	0
Realizar pedido de artículos en la aplicación cliente	10	5	2	0
Consultar pedidos de artículos en la aplicación cliente	10	5	2	0
En la aplicación servidor se muestran los eventos enviados por todos los clientes conectados	5	3	1	0

Correcto uso de subprocesamiento múltiple en servidor	5	3	1	0
Implementación de un método para administrar la concurrencia de recursos compartidos en el servidor al realizar una venta	5	3	1	0
Correcto uso del manejo de excepciones	3	2	1	0
Interfaz de usuario GUI es fácil de usarse y presenta buen diseño	5	3	1	0
Manual de uso de la solución	4	2	1	0
TOTAL	100			