



UNIVERSIDAD ESTATAL A DISTANCIA
VICERRECTORÍA ACADÉMICA
ESCUELA DE CIENCIAS EXACTAS Y NATURALES
Cátedra Tecnología de Sistemas



Programación Avanzada

Código: 00830

Proyecto 2. Valor 3%

Temas de Estudio

1. Tema 2 Particularidades del lenguaje C#
2. Tema 3 Manejo de excepciones en C#
3. Tema 4 Conceptos de Interfaz Gráfica
4. Tema 5 Subprocesamiento múltiple
5. Tema 6 Bases de datos y componentes ADO.NET
6. Tema 7 Colecciones
7. 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 la empresa **DESARROLLO-UNED** le ha ido muy bien con el seguimiento y gestión de las citas programadas, ya que el sistema de información desarrollados les ha brindado muchas facilidades, por lo que le han contratado nuevamente para hacer mejoras en su sistema de información y que la información sea almacenada de forma permanente en una base de datos SQL Server.

El sistema a 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 las citas dentales que requiere, mientras que el proyecto servidor será utilizado por un único administrador para dar mantenimiento a las funcionalidades de tipos de consulta, clientes, doctores y citas. 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 citas.
- 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 cita: Se debe mostrar una opción que permita al cliente registrar las citas con los doctores disponibles. Para esto el sistema debe crear la pantalla programar cita y debe realizar lo siguiente:

- Se debe enviar solicitud al servidor para consultar los doctores disponibles, esta información se debe mostrar en un control ComboBox, o bien, en un control DataGridView.
 - Se debe validar que al menos exista un doctor activo registrado en el sistema para realizar la asignación de la cita.
 - Solo debe mostrar los doctores activos.
- 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 cita del cliente que ejecuta la aplicación.
- Se debe enviar solicitud al servidor para consultar todos los tipos de consulta. La información de los tipos de Consulta se debe mostrar en un control Combobox.
 - Se debe validar que al menos exista un tipo de consulta registrada en el sistema.
- Se debe ingresar la fecha y hora de la cita. Se debe enviar una solicitud al servidor para validar si ya existe una cita para esa fecha y hora para el doctor seleccionado.

- Al hacer clic en el botón “Realizar Cita” debe enviarse al servidor, mediante la red, la información de la cita. 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.
- Mostrar en pantalla si el proceso fue exitoso.

Consultar mis citas: Está opción le permite al cliente consultar desde la aplicación cliente las citas realizadas.

- Está opción solo muestra las citas realizadas por el cliente que está ejecutando la aplicación cliente, no debe mostrar citas realizadas por otros clientes.
 - Debe enviar la solicitud de consultar las citas 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 las citas realizadas por el cliente.
 - La información para presentar debe mostrar los números de cita, fecha, hora y el doctor asignado a la cita.
 - Se debe implementar dos formas (funcionalidades) de consultar citas, una pudiendo consultar todas las citas realizadas por el cliente; la otra consultando citas específicamente mediante el número de la cita.
- **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 14100 para escuchar solicitudes.
- Programación Avanzada Código: 00830 Página 5 de 15.
- 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 citas, consulta de cita. • Siempre debe ser visible la cantidad de clientes conectados.

El sistema debe contar con un **menú principal** con las siguientes opciones:

- **Registrar Tipos de Consulta**
- **Administración de Clientes**
- **Administración de Doctores**
- **Registro de Citas**
- **Reporte Citas por Fecha**
- **Reporte Citas por Doctor**
- **Reporte Citas por Cliente**

Registrar Tipos de Consulta

Esta opción del menú le permite al usuario realizar el registro, consulta y modificación de los tipos de consultas que realizarán en el consultorio, por ejemplo: extracción de piezas dentales, limpieza, cirugías, entre otros servicios brindados.

Esta pantalla deberá con la funcionalidad de registro de la siguiente información:

- Número (int), este campo corresponde a almacenar una identificación numérica única por tipo de consulta.
- Descripción(string), este campo corresponde almacenar la descripción del tipo de consulta y debe ser requerido.

- Estado (char), debe mostrar la información en un Combobox con el detalle del control de Activo o Inactivo, pero debe almacenar a lo interno una "A" para Activo y una "I" para Inactivo, este campo también es requerido.

Así mismo, el sistema deberá mostrar un mensaje en las siguientes validaciones:

- Cuando se trate de ingresar un número de tipo de consulta que ya se encuentra registrada en el sistema.
- Se presente un error en el formato del campo numérico.
- Cuando los campos número y descripción se encuentren vacíos (nulo).
- Cuando el Combobox no tenga seleccionada una opción.
- Cualquier otro error que se presente en el sistema, debe estar manejado por una excepción.

Una vez capturada la información se debe registrar la información en la base de datos. La pantalla debe tener la funcionalidad de consultar en un DataGridView la información de los tipos de Consulta registrados la base de datos, cada vez que ingrese un dato nuevo debe actualizarse la consulta en el control del DataGridView.

Además, la pantalla debe tener la funcionalidad de seleccionar un tipo de consulta y modificar el estado del tipo de consulta (activo o inactivo). **No debe modificar el número de identificación ni la descripción.**

Se deberá crear como **mínimo** una nueva ventana (Windows Form) para resolver los requerimientos solicitados anteriormente.

Administración de Clientes

Esta opción del menú le permite al usuario realizar el registro, consulta y modificación de clientes que serán atendidos en las citas programadas, es importante registrar sus datos personales.

Esta pantalla deberá con la funcionalidad de registro de la siguiente información:

- Identificación (int), este campo corresponde a almacenar una identificación numérica y única del cliente.
- Nombre(string), este campo corresponde almacenar el nombre del cliente y debe ser requerido.
- Apellido1 (string), este campo corresponde almacenar el primer apellido del cliente y debe ser requerido.
- Apellido2 (string), este campo corresponde almacenar el segundo apellido del cliente y debe ser requerido.

- Fecha de nacimiento (DateTime), este campo corresponde la fecha de nacimiento del cliente y debe ser requerido.
- Género (char), debe mostrar la información en un Combobox con el detalle del control de Femenino, Masculino o No Especificado, pero debe almacenar a lo interno una "F" para Femenino, una "M" para Masculino o una "N" para No especificado, este campo también es requerido.

Así mismo, el sistema deberá mostrar un mensaje en las siguientes validaciones:

- Cuando se trate de ingresar una identificación que ya se encuentra registrada en el sistema.
- Se presente un error en el formato del campo, ya sea numérico o de fecha.
- Cuando los campos identificación, nombre, apellido1, apellido2 y fecha de nacimiento se encuentren vacíos (nulo).
- Cuando el Combobox no tenga seleccionada una opción.
- Cualquier otro error que se presente en el sistema, debe estar manejado por una excepción.

Una vez capturada la información se debe registrar la información en la base de datos. La pantalla debe tener la funcionalidad de consultar en un DataGridView la información de los clientes registrados en la base de datos, cada vez que ingrese un dato nuevo debe actualizarse la consulta en el control del DataGridView.

Además, la pantalla debe tener la funcionalidad de seleccionar un cliente y modificar la fecha de nacimiento y el género. **No debe modificar el número de identificación, nombre, apellido1 y apellido2.**

Se deberá crear como **mínimo** una nueva ventana (Windows Form) para resolver los requerimientos solicitados anteriormente.

Administración de Doctores

Esta opción del menú le permite al usuario realizar el registro, consulta y modificación de doctores que serán los que brindaran las consultas programadas, es importante registrar los datos relevantes del doctor.

Esta pantalla deberá con la funcionalidad de registro de la siguiente información:

- Identificación (int), este campo corresponde a almacenar una identificación numérica y única del doctor.
- Nombre(string), este campo corresponde almacenar el nombre del doctor y debe ser requerido.

- Apellido1 (string), este campo corresponde almacenar el primer apellido del doctor y debe ser requerido.
- Apellido2 (string), este campo corresponde almacenar el segundo apellido del doctor y debe ser requerido.
- Estado (char), debe mostrar la información en un Combobox con el detalle del control de Activo o Inactivo, pero debe almacenar a lo interno una "A" para Activo y una "I" para Inactivo, este campo también es requerido.

Para ello se debe crear una Clase con los atributos mencionados anteriormente. Así mismo, el sistema deberá mostrar un mensaje en las siguientes validaciones:

- Cuando se trate de ingresar una identificación que ya se encuentra registrada en el sistema.
- Se presente un error en el formato del campo, ya sea numérico o de fecha.
- Cuando los campos identificación, nombre, apellido1 y apellido2 se encuentren vacíos (nulo).
- Cuando el Combobox no tenga seleccionada una opción.
- Cualquier otro error que se presente en el sistema, debe estar manejado por una excepción.

Una vez capturada la información se debe registrar la información en la base de datos. La pantalla debe tener la funcionalidad de consultar en un DataGridView la información de los doctores registrados en la base de datos, cada vez que ingrese un dato nuevo debe actualizarse la consulta en el control del DataGridView.

Se deberá crear como **mínimo** una nueva ventana (Windows Form) para resolver los requerimientos solicitados anteriormente.

Registro de Citas

Funcionalidad que debe permitir al usuario registrar las citas que requiere el paciente (cliente) con el dentista disponible. El sistema deberá tener la funcionalidad de asignar un número de cita, poder seleccionar el cliente que ya se encuentre registrado, así como el doctor designado. El sistema debe validar que no encuentre una cita a la fecha y hora seleccionada para el doctor seleccionado. Para esto se debe realizar lo siguiente:

- Imprimir en pantalla la lista de doctores disponibles.
 - Se debe validar que al menos exista un doctor activo registrado en el sistema para realizar la asignación de la cita.
 - Solo debe mostrar los doctores activos.
- Imprimir en pantalla la lista de los clientes registrados.
 - Se debe validar que al menos exista un cliente registrado para ser asociado

- Para cada registrado de la lista se debe mostrar la siguiente información: id, nombre, apellido1, apellido2.
- Número(int), este campo corresponde a almacenar el número de cita y único.
- Fecha y Hora de la cita (DateTime), este campo corresponde a almacenar la fecha y hora de la cita y debe ser requerido.
- Tipo de Consulta, debe tener la funcionalidad de seleccionar el tipo de consulta, de acuerdo con lo almacenado en la base de datos de Tipos de Consulta.
- Cliente (Objeto Cliente), debe tener la funcionalidad de seleccionar el cliente, de acuerdo con lo almacenado en la base de datos de Clientes.
- Doctor (Objeto Doctor), debe tener la funcionalidad de seleccionar el doctor, de acuerdo con lo almacenado en la base de datos de Doctores.

Para ello se de crear una Clase con los atributos mencionados anteriormente. Así mismo, el sistema deberá mostrar un mensaje en las siguientes validaciones:

- Cuando se trate de ingresar un número que ya se encuentra registrada en el sistema.
- Se presente un error en el formato del campo, ya sea numérico o de fecha.
- Cuando los campos identificación, nombre, apellido1 y apellido2 se encuentren vacíos (nulo).
- Cualquier otro error que se presente en el sistema, debe estar manejado por una excepción.

Una vez capturada la información se debe registrar la información en la base de datos.

Reporte Citas por Fecha

Esta opción debe permita al usuario visualizar las citas por una fecha en específico, debe contar con la funcionalidad de que el usuario seleccione una fecha y el sistema muestre solamente las citas que existen para el día seleccionado. La información que debe mostrar en el control DataGridView es el siguiente:

- Número de cita.
- Fecha y Hora de la cita.
- Descripción del tipo de consulta.
- Nombre completo del cliente.
- Nombre completo del doctor.

Se deberá crear como **mínimo** una nueva ventana (Windows Form) para resolver los requerimientos solicitados anteriormente.

Reporte Citas por Doctor

Esta opción debe permitir al usuario visualizar las citas por un doctor en específico, debe contar con la funcionalidad de que el usuario seleccione un doctor que ya se encuentre registrado en la base de datos de doctores y el sistema muestre solamente las citas que existen para el doctor seleccionado. La información que debe mostrar en el control DataGridView es el siguiente:

- Número de cita.
- Fecha y Hora de la cita.
- Descripción del tipo de consulta.
- Nombre completo del cliente.

Se deberá crear como **mínimo** una nueva ventana (Windows Form) para resolver los requerimientos solicitados anteriormente.

Reporte Citas por Cliente

Esta opción debe permitir al usuario visualizar las citas por un cliente en específico, debe contar con la funcionalidad de que el usuario seleccione un cliente que ya se encuentre registrado en la base de datos de cliente y el sistema muestre solamente las citas que existen para el cliente seleccionado. La información que debe mostrar en el control DataGridView es el siguiente:

- Número de cita.
- Fecha y Hora de la cita.
- Descripción del tipo de consulta.
- Nombre completo del doctor.

Se deberá crear como **mínimo** una nueva ventana (Windows Form) para resolver los requerimientos solicitados anteriormente.

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 (Tipos de Consulta. Clientes, Doctores y Citas) no deben contener lógica para solicitar información al usuario, solo debe tener la estructura de la clase y sus propiedades.
- Las clases no deben tener métodos vacíos.
- Cada clase debe ser creada en archivo por separado.

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 combobox y DataGridView
- No use DataSet ni DataTable

Interfaz de usuario

- Debe usar interfaz de usuario con formularios GUI.

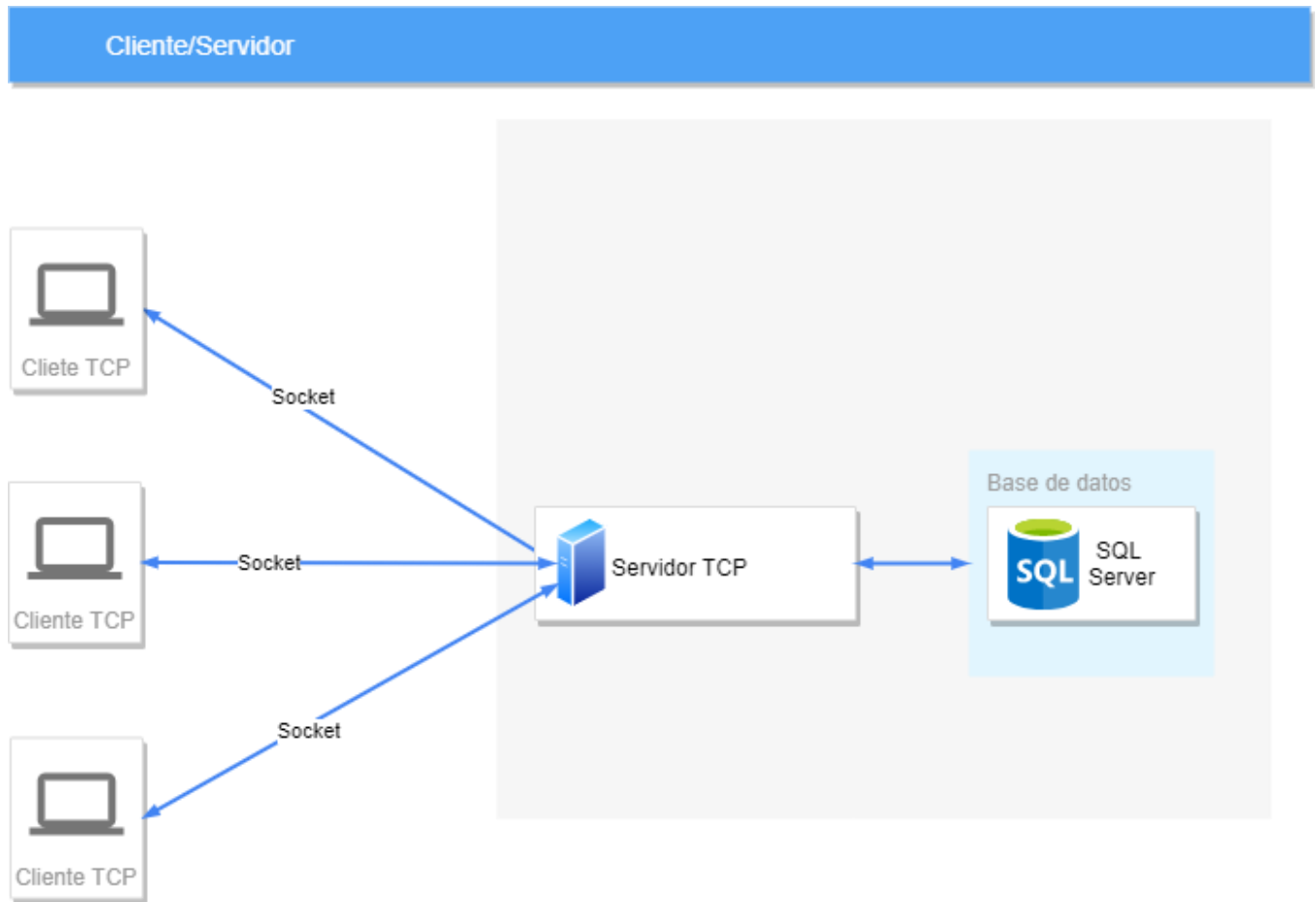
Base de datos

- 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



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 < Proyecto1>.
- El proyecto 1 debe estar desarrollado en Visual Studio Community 2019 o superior 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 del proyecto. **Ejemplo: JuanRojas-proyecto1.**
- La entrega de la <Proyecto1> en las fechas establecidas en la plataforma de aprendizaje en línea Moodle en el apartado que se indique.

- Si no concluyó a tiempo la asignación, 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	0.5
POO y creación de las clases (Tipos de Consulta, Clientes, Doctores, Citas)	7	4	1	0.5
Registrar y Modificación de Tipos Consulta en el servidor	5	6	3	0.5
Registrar y Modificación de Clientes y Doctores en el servidor	10	6	3	0.5
Registrar de Citas en el servidor	10	6	3	0.5
Reporte de Citas por Fecha, Doctor y Cliente	5	3	2	0.5
Correcto uso del manejo de excepciones	5	3	2	0.5
Conexión de la aplicación cliente con el servidor mediante TCP y administración de las conexiones en el servidor	10	6	3	0.5
Validación del usuario cliente en el cliente	5	3	2	0.5
Registrar de Citas en el cliente	10	6	3	0.5
Consultar de Citas en el cliente	5	3	2	0.5
En la aplicación servidor se muestran los eventos	5	3	2	0.5

enviados por todos los clientes conectados				
Implementación de un método para administrar la concurrencia de recursos compartidos en el servidor al realizar una venta	10	6	3	0.5
Interfaz de usuario de consola es fácil de usarse y presenta buen diseño.	5	3	2	0.5
Manual de uso de la solución	5	3	2	0.5
TOTAL	100			