



GracoSoft

ALCANCE DE PROYECTO

LABORATORIO DE C# II

Instructor: Carmelo Yonso

Email: carmelo.yonso@outlook.com

Telegram: @FritoX

Fecha: Abril - Mayo 2023

Laboratorio II

A lo largo de este laboratorio, diseñaremos e implementaremos una aplicación de escritorio en la cual pondremos en práctica todos los conocimientos recibidos a lo largo de los cursos del segundo nivel, aplicando todo lo que hemos estudiado entre Lógica de Programación II, Bases de Datos I y, por supuesto, C# II. El proyecto se resume en el desarrollo de una aplicación de escritorio que implemente un control de acceso con usuario y contraseña, roles de usuario y con persistencia de datos en SQL Server a través del uso del ORM Entity Framework Core.

Instrucciones Generales

- Este proyecto será estrictamente individual.
- El proyecto comenzará a ser desarrollado en la mañana del día lunes 24 de Abril del 2023, con fecha de finalización el día viernes 12 de Mayo del 2023. Este día se llevará a cabo una exposición individual del programa, en la cual los estudiantes mostrarán sus aplicaciones.
- Los estudiantes contarán con apoyo, tutorías y asesorías del instructor de forma presencial durante las semanas del 24 de Abril 12 de Mayo, en el horario comprendido de las 8 am a las 12 pm.
- Para el desarrollo del proyecto, los estudiantes deberán utilizar los elementos estudiados hasta ahora. Esto implica, pero no se limita, a:
 - Diseño de Bases de Datos Relacionales.

- SQL Server.
 - Entity Framework Core.
 - Interfaces MDI con WinForms.
 - Excepciones.
 - Funciones Lambda.
 - Enumerable.
 - Estructuras de datos: pilas, colas, listas.
- El proyecto debe implementar una Base de Datos en el manejador SQL Server. La misma será diseñada e implementada por el estudiante, y deberá cumplir con las tres formas normales y tener una cobertura apropiada para el funcionamiento de la aplicación.
- La aplicación de escritorio resultante debe satisfacer completamente lo expresado en el enunciado del proyecto. Cualquier característica extra que el estudiante desee implementar, correrá por su propia cuenta individual, y se sugiere que se proceda con su implementación *sólo* tras finalizar lo requerido por el instructor.
- El instructor realizará una serie de pruebas con un conjunto de datos bien estructurado para verificar que la aplicación funcione como es debido. En este sentido, se sugiere que el estudiante organice su propio conjunto de datos de prueba que garanticen la funcionalidad de la aplicación.

Entregables e instrucciones de entrega

El proyecto contará con dos entregas:

1) Primer Punto de Control. En este, se realizará la entrega de la primera parte del proyecto, y está conformado por:

- Modelo de Objetos de Negocio.
- Diagrama MER.
 - Implementación usando EF Core.
- Listado de Requisitos Funcionales.
 - Del listado, especificar dos requisitos.
- Diagrama de Casos de Uso.

Todo esto (excepto la implementación de los modelos) se deberá desarrollar en un informe, el cual será entregado en formato PDF al aula virtual **hasta el martes 25/04/2022** a las 11.59 p.m.

La implementación de los modelos se deberá entregar en un archivo comprimido .rar, .zip o .7z, en el cual se encuentre la carpeta Modelos de su proyecto, y dentro de ella, todos sus modelos junto a la clase DbContext correspondiente.

2) Entrega final. En esta entrega, se deberá enviar el informe corregido del primer punto de control, y junto a él, el código fuente de la aplicación ya finalizada.

Al igual que con el punto de control, en esta entrega final se solicitará que el informe esté en formato pdf, y que el código fuente del

proyecto esté en un archivo comprimido. Su fecha tope de entrega será el **viernes 12/05/2023** a las 11:59 p.m.

Evaluación

El proyecto será evaluado por el instructor, y las entregas tendrán la siguiente puntuación:

- 1) El punto de control consistirá de 20 puntos, divididos en 15 puntos para el informe y 5 puntos para el código de las entidades.
- 2) La entrega final consiste de 80 puntos, cuyo detalle se especifica en la rúbrica ubicada al final del documento.

Modalidad

A diario, cada clase se dividirá en dos bloques bien definidos:

- De 8 am a 10 am, se procederá a realizar un diagnóstico individual para evaluar el estado actual del proyecto. En este, se solicitará a cada estudiante sus avances del día, qué problemas presentaron y cuál será su objetivo diario. Todo esto quedará registrado en un diario de actividades que llevará el instructor.
- De 10 am a 10:15 am habrá un descanso.
- De 10:15 am a 12:00 pm, el instructor estará disponible para contestar cualquier duda, tanto teórica como técnica, que surja en los estudiantes.

Predacons Appointment Management

La Empresa **Predacons Appointment Management**, se encarga de la instalación de ventanas, persianas electrónicas y cortinas térmicas, esta empresa necesita un software para agendar todas sus citas de instalación, este software le debe permitir a la empresa la facilidad de ver citas diarias para que sus trabajadores (**Instaladores**) puedan gestionar mejor sus instalaciones.

El sistema de permitir hacer las siguientes acciones:

1. Menú: EL sistema debe mostrar las opciones que se pueden realizar dentro del sistema dependiendo del tipo de usuario.

2. Gestión de Citas: El sistema debe permitir que un operador pueda registrar una cita con la siguiente información: código de la cita, cedula del cliente, nombre del cliente, dirección del cliente, número de teléfono del cliente, debe especificar quien será el instalador (trabajador), fecha de instalación de posible instalación, fecha de la instalación y servicio requerido.

El sistema debe permitir cambiar los estados de las citas (Pendiente, Instalada o Cancelada). Ejemplo:

- i. Pendiente: Estado que toma la cita al crearse.
- ii. Instalada: Cuando un trabajador concrete una instalación. Aquí el sistema debe permitir almacenar los materiales y cantidades utilizados.
- iii. Cancelada: Estado que toma la cita si el cliente cancela la cita.

3. Listar las Citas: El sistema debe ser capaz de listar todas las citas registradas en la base de datos por estados. (Reporte).

4. Gestión de acceso. La aplicación debe mantener una gestión de acceso, para lo cual se debe solicitar antes de iniciar un usuario y contraseña, lo que brindará un servicio de autenticación para el control de acceso. Así, el programa también debe permitir listar, agregar, modificar y eliminar los usuarios, y además de ello, definir sus roles.

- a. **Gestión de roles.** La aplicación debe permitir definir distintos roles de usuario, para lo cual se debe especificar su nombre y

una breve descripción del mismo. Estos roles podrán ser listados, creados, modificados y eliminados. Existen solo 2 tipos de roles: Operador y Administrador. El Administrador tendrá acceso a todo y operador no tendrá acceso a los reportes.

5. Gestión de Materiales: El sistema debe gestionar materiales con la siguiente información: código del material, descripción, precio de compra, precio de venta (25% más del precio de compra), cantidad en existencia.

6. Listar Materiales: El programa debe listar la lista de materiales por cada cita. (Reporte).

7. Gestión de Trabajadores: Opción del menú que permite las operaciones de actualización de Trabajadores. Los trabajadores contarán con los siguientes datos: cédula, nombre, apellido, teléfono, fecha de nacimiento, dirección, sexo.

8. Gestión de Servicio: Opción del menú en donde se registran los servicios previstos por la empresa. Consta de los siguientes datos: código, descripción, precio.

9. Gestión de Facturación: Opción en el sistema que permite almacenar la información que se genere al momento de facturar los materiales y el servicio de instalación prestado. La facturación estará desglosada con los siguientes datos: código de la factura, código de la cita, fecha, iva. **Nota:** Todos los precios se manejarán en \$.

Extra:

Esta opción es adicional, si la desea desarrollar y lo hace completamente, tendrá una puntuación adicional de 10 pts, en dado caso la requiera.

Listar Facturas con montos en Bs. : Esta opción debe permitir listar las facturas emitidas, pero con los montos en \$ según la tasa del día en que se hicieron.

Gestión de tasas: Opción del menú en donde se registran los cambios diarios de la tasa según el Banco Central de Venezuela, esto con fines de mostrar la información necesaria de pago en moneda extranjera (\$).

Nota: Todas las tablas maestras, deben manejar status y el sistema debe proveer los métodos de actualización de las mismas.

Recomendaciones finales

1. Las interfaces del programa deben ser acordes a la realidad conceptual del mismo y a las características necesarias por este modelo de negocio: es así que se requiere que sean unas interfaces limpias, cómodas y rápidas de utilizar.
2. Recordemos que **todos** los formularios, controles y contenedores deben estar debidamente identificados. Esto es: no deben existir formularios llamados Form1, o botones llamados button1. Más aún, ya que se trata de una solución compleja, se recomienda encarecidamente que el estudiante utilice la arquitectura de código MVC modificada dada en el curso.
3. Todas las interfaces, deben mostrar en tiempo real la fecha y hora. No olvide que el programa debe estar hecho con mensajes, manejo de variables, entre otros, en el idioma inglés.
4. Tener presente la rúbrica de evaluación.
5. El código siempre debe ser legible, y de preferencia, estar comentado, de forma tal que cualquier programador sea capaz de entender sin problemas el código escrito por el estudiante. Del mismo modo, el estudiante es totalmente libre de consultar, basarse y utilizar los materiales, guías, recursos, videos, tutoriales, etc, que considere necesario para desarrollar su aplicación. **El internet es una fuente inagotable de conocimiento**, y todos somos libres de aprovecharlo al máximo.