



FUNDAMENTOS DE SISTEMAS DE INFORMACION (SI393)
Ciclo 2023-02
Segunda Práctica Calificada

Profesor: Orestes Ramirez
Duración: 100 minutos
Sección: SS51

Instrucciones:

- a) Resuelva la prueba utilizando C# y SQL Server. Recuerde que es su responsabilidad guardar adecuadamente sus programas y enviarlos correctamente.
- b) Al finalizar el examen, suba un archivo comprimido (extensión .ZIP) a la actividad PC2. El archivo comprimido debe contener la carpeta que contiene la solución.
- c) El nombre del archivo comprimido debe seguir el formato: NOMBRE_APELLIDO_PC2.zip. Ejemplo: Pedro_Martinez_PC2.zip.

Pregunta 1: Sistema para Gestión de Proyectos de Desarrollo de Software (20 puntos)

Una empresa de desarrollo de software necesita un sistema para gestionar sus proyectos. La empresa tiene equipos de programadores distribuidos en diferentes ubicaciones, y cada equipo está asignado a un proyecto específico. Cada proyecto se caracteriza por un código único, un nombre, una fecha de inicio y una fecha de finalización estimada.

Cada programador en el equipo tiene un código único, un nombre, una especialización (por ejemplo, front-end, back-end, full-stack) y una experiencia en años. Los programadores pueden estar asignados a uno o varios proyectos.

El administrador solicita implementar el mantenimiento de las tablas **Proyectos** y **Programadores**, con las siguientes funcionalidades:

1. Agregar un nuevo proyecto.
2. Eliminar un proyecto.
3. Modificar el nombre y la fecha de finalización estimada de un proyecto según su código.
4. Mostrar los proyectos según la especialización del programador y/o la fecha de inicio del proyecto.
5. Diseñar las interfaces necesarias que materialicen estas funcionalidades.

Instrucciones:

Crear en SQL Server la base de datos **dbGestionProyectos** que contenga las siguientes tablas.

Tabla: Proyectos

- CodigoProyecto	int
- Nombre	varchar
- FechaInicio	date
- FechaFinalizacionEstimada	date

Tabla: Programadores

- CodigoProgramador	int
- Nombre	varchar
- Especializacion	varchar
- ExperienciaEnAnios	int

Crear una tabla intermedia para manejar la relación muchos a muchos entre Proyectos y Programadores, ya que un programador puede estar asignado a varios proyectos y un proyecto puede tener varios programadores. Llamarla **AsignacionProyectosProgramadores** y debe tener al menos los campos CodigoProyecto y CodigoProgramador.

Desarrollar el proyecto por capas en C# en un entorno visual, que permita implementar el mantenimiento de la tabla Proyectos y Programadores, utilizando las funcionalidades mencionadas anteriormente.

Se solicita:

1. Base de Datos y Estructura de Tablas (3 puntos):

- Correcta creación de la base de datos **dbGestionProyectos**.
- Inclusión de las tablas **Proyectos**, **Programadores**, y **AsignacionProyectosProgramadores**.
- Definición adecuada de campos con el uso correcto de tipos de datos.
- Creación de la relación muchos a muchos con la tabla intermedia.

2. Implementación en C# y Entity Framework (4 puntos):

- Desarrollo de capas en C# (presentación, negocio, datos).
- Uso efectivo de Entity Framework para interactuar con la base de datos.
- Implementación exitosa de las funcionalidades solicitadas (Agregar, Eliminar, Modificar, Mostrar).

3. Manejo de Relaciones en Entity Framework (2 puntos):

- Uso adecuado de relaciones entre las entidades en Entity Framework.
- Asegurar la consistencia y la integridad referencial.

4. Diseño de Interfaces (3 puntos):

- Diseño de interfaces visuales que reflejen las funcionalidades de mantenimiento solicitadas.
- Interfaz de usuario intuitiva y fácil de usar.

5. Claridad y Legibilidad del Código (3 puntos):

- Utilización de nombres de variables, funciones y clases descriptivos y significativos.
- Estructura del código que facilita la comprensión y el mantenimiento.
- Evitar código redundante y mantener un estilo de codificación coherente.

6. Excepciones y Manejo de Errores (2 puntos):

- Implementación efectiva de manejo de excepciones y errores.
- Identificación y gestión adecuada de situaciones inesperadas.

7. Eficiencia y Rendimiento (3 puntos):

- Uso eficiente de consultas y operaciones en la base de datos.
- Optimización del rendimiento del código en términos de velocidad y recursos.

Nota: Usar Entity Framework para facilitar la interacción con la base de datos y mantener el código C# más limpio y organizado. Además, recuerda manejar adecuadamente las relaciones entre las entidades y las operaciones de base de datos.