|  |
| --- |
|  |
| **EVALUACIÓN 2 - 20%.**  UNIDAD I:  Sistema de prestamo de libros (WEB API) Puntaje Obtenido: Nota: |
| FECHA: El 02-04-2025 |

# NOMBRE ALUMNO:

**CARRERA**: Ingeniería en Ejecución Informática

**CÓDIGO**: IEI- 089

**ASIGNATURA:** Programación .NET **DOCENTE:** Maximiliano Moraga

**MODALIDAD:** Presencial Diurna

Lea atentamente las instrucciones y cada pregunta antes de responder.

**Estructura:** La evaluación consta de 24 puntos. Requiere 15 puntos para aprobar. La evaluación se subdivide en 4 ejercicios con 8 puntos cada uno.

**Entregables:** El estudiante luego del tiempo establecido entregara el proyecto con la solución.

**Tiempo asignado:** Usted dispone de 7 días, para desarrollar esta evaluación.

**Entregables**

* Código fuente en GitHub (con README) o comprimido en zip por el aula virtual.
* Capturas de pruebas de cada funcionalidad requerida en este documento o video demostrativo.
* Nombre del comprimido: NombreApellido\_Eva\_2.zip

# Caso de Estudio: Desarrollo de una Web API RESTful en ASP.NET Core

La Biblioteca Municipal “Letras Libres” desea modernizar su sistema de préstamos y gestión de libros. Actualmente usan hojas de cálculo y registros manuales. Tu equipo de desarrollo ha sido contratado para construir una **API RESTful** que permita digitalizar este proceso y sentar las bases para futuras aplicaciones móviles y web.

Desarrollar una **Web API** utilizando **ASP.NET Core** y **Entity Framework Core**, que permita gestionar libros, usuarios y préstamos de forma eficiente, segura y escalable.

# Requisitos Funcionales

**Libros**

* GET /api/libros – Lista todos los libros.
* GET /api/libros/{id} – Muestra los detalles de un libro.
* POST /api/libros – Crea un nuevo libro.
* PUT /api/libros/{id} – Edita la información de un libro.
* DELETE /api/libros/{id} – Elimina un libro (solo si no está prestado).

**Usuarios**

* GET /api/usuarios – Lista de usuarios registrados.
* POST /api/usuarios – Registrar un nuevo usuario.
* GET /api/usuarios/{id}/prestamos – Ver el historial de préstamos del usuario.

**Préstamos**

* POST /api/prestamos – Registrar un nuevo préstamo (libro → usuario).
* POST /api/devoluciones – Registrar la devolución de un libro.

# Entity Framework Core (OBLIGATORIO)

* Uso de **EF Core como ORM** con SQL Server (LocalDB, SQLite o InMemory para pruebas).
* Configuración de DbContext e inyección en Program.cs.
* Uso de **migraciones** para crear y actualizar la base de datos.
* Relación entre entidades bien definida (uno-a-muchos, etc.).
* Validaciones en modelos con **Data Annotations**. (Opcional)
* Uso de Include() para obtener datos relacionados en consultas (como historial de préstamos).

**Documentación**

* Configurar Swagger para probar y documentar los endpoints.

# Validaciones

Las pruebas de las validaciones son con respecto a funcionalidad, entregar captura del programa funcionando mal con su código y captura del sistema funcionando bien con su código corregido.

# Criterios de evaluación

|  |  |
| --- | --- |
| Criterio | Puntaje |
| Configuración de proyecto, limpieza y estructura | 10 pts |
| Implementación de CRUD para libros, usuarios y préstamos | 20 pts |
| Uso correcto de Entity Framework Core | 20 pts |
| Relaciones entre entidades correctamente modeladas | 10 pts |
| Validaciones y control de errores | 10 pts |
| Pruebas funcionales (Swagger/Postman) | 10 pts |
| Código limpio y documentado | 5 pts |

# Rubrica de evaluación

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Criterio** | **Excelente (100%))** | **Bueno (80%)** | **Aceptable (60%)** | **Insuficiente (40%)** | **Deficiente (20% o menos)** | **Puntaje Máx.** |
| **Configuración de proyecto, limpieza y estructura** | Estructura de carpetas correcta, sin archivos innecesarios, Program.cs y Startup.cs/builder bien organizados. | Estructura mayormente correcta, 1-2 archivos innecesarios. | Estructura con errores o carpetas mal ubicadas. | Código confuso, archivos mezclados, sin organización clara. | Proyecto desordenado, difícil de ejecutar o entender. | 10 pts |
| **CRUD para libros, usuarios y préstamos** | CRUD completo (Create, Read, Update, Delete) para las 3 entidades. Total: 12 endpoints funcionales. | CRUD completo para 2 entidades y parcial para 1. (8–11 endpoints funcionales) | Solo operaciones de lectura o escritura. (4–7 endpoints) | CRUD incompleto para todas las entidades. (1–3 endpoints) | CRUD inexistente o no funcional. | 20 pts |
| **Uso correcto de Entity Framework Core** | Usa DbContext, DbSet, migraciones, LINQ y async/await correctamente en 90%+ de los casos. | Uso correcto en 70–89% de los casos. Migraciones usadas. | EF usado pero con errores de sintaxis o mal manejo de relaciones. (50–69%) | Uso incorrecto en mayoría de los casos. No aplica buenas prácticas de EF. | No usa EF Core o lo usa erróneamente. | 20 pts |
| **Relaciones entre entidades correctamente modeladas** | Relaciones 1:N y N:M definidas en modelos y en DbContext. Navegación funcional. | Relaciones definidas pero sin navegación clara o con omisiones menores. | Relaciones presentes pero con errores de clave foránea o navegación. | Relaciones ausentes o mal implementadas. | No hay relaciones o están rotas. | 10 pts |
| **Validaciones y control de errores** | Validaciones en DTOs/Controllers. Manejo de excepciones y mensajes claros. (90%+ cubierto) | Validaciones presentes pero sin manejo completo de errores. (70–89%) | Validaciones en algunos campos, control limitado de errores. (50–69%) | Muy pocas validaciones. Errores no controlados. | No hay validaciones ni manejo de errores. | 10 pts |
| **Pruebas funcionales (Swagger/Postman)** | Documentación completa en Swagger. Pruebas en Postman evidencian 90%+ de endpoints. | Swagger funcional, 70–89% probado. Capturas/Postman entregadas. | Swagger básico, Postman con 50–69% de pruebas. | Swagger con errores o rutas incompletas. Postman deficiente. | No hay Swagger ni evidencia de pruebas. | 10 pts |
| **Código limpio y documentado** | Código indentado, sin redundancias, con comentarios explicativos en métodos y clases. | Código claro, pocos comentarios, pero entendible. | Código funcional, con comentarios mínimos o desorganizados. | Código difícil de seguir, muchas líneas redundantes. | Código sucio, sin comentarios, sin seguir convenciones. | 5 pts |

# Escala de notas:

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

# Validaciones

Al agregar libros me mostraba el json de prestamos juntos con los de libros lo que ocasionaba confusión:  
Texto

El contenido generado por IA puede ser incorrecto.

Texto

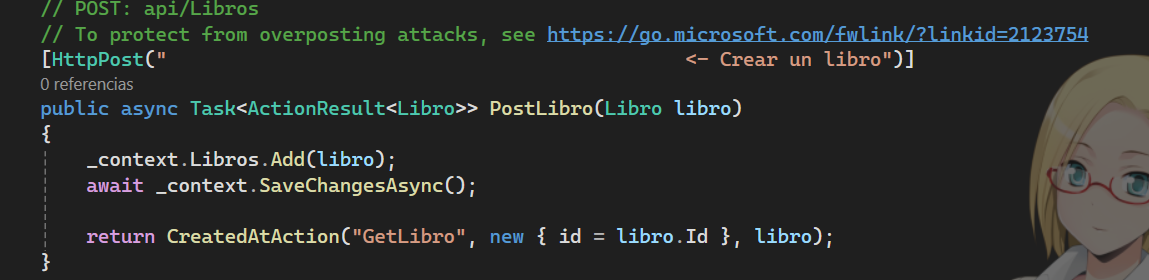
El contenido generado por IA puede ser incorrecto.

¿Cómo se soluciono?

R: Utilizando [JsonIgnore] que es un atributo de c# que nos permite ocultar campos del Json

Un reloj digital en la pantalla

El contenido generado por IA puede ser incorrecto.

Agregando libros sin validaciones;  


Como se ve en json:  
Texto

El contenido generado por IA puede ser incorrecto.

Las validaciones que se agregaron fueron que el libro siempre se debe de crear con el estado de préstamo false ya que al estar recién creado no se existe un préstamo hacia este libro , No se añaden libros que estén publicados antes del año 868 ya que se supone que este es el mas antiguo y no se puede agregar libros que estén publicados después de el año en que estamos, los campos no se pueden enviar como string o campos vacíos.

Texto

El contenido generado por IA puede ser incorrecto.

Pantalla negra con letras blancas

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Eliminando Libros sin validaciones:  
Texto

El contenido generado por IA puede ser incorrecto.

Forma, Patrón de fondo, Rectángulo

El contenido generado por IA puede ser incorrecto.

Borrando Libros con validaciones; Para borrar libro se valida de la siguiente manera, que la id exista en la base de datos, que el libro no este en estado de prestamos en true ya que significa que esta en préstamo y no se puede borrar asta que el libro sea devuelto a por el usuario, se muestra mensaje de que libro se borro correcta mente.

Patrón de fondo

El contenido generado por IA puede ser incorrecto.

Patrón de fondo

El contenido generado por IA puede ser incorrecto.

Actualizar libro sin validacionesTexto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Actualizar libro con validaciones; se valida que la ID del libro exista , no se puede editar el EstadoPrestado si esta en True porque el libro tiene que ser devuelto para poder editarlo, se agrego mensaje de que libro fue actualizado con éxito y un mensaje en caso de que pase lo contrario.

Texto

El contenido generado por IA puede ser incorrecto.

Patrón de fondo

El contenido generado por IA puede ser incorrecto.

Traer Uno sin validaciones:

Texto

El contenido generado por IA puede ser incorrecto.

Patrón de fondo

El contenido generado por IA puede ser incorrecto.

Traer Uno con validaciones: se agrego mensaje de libro encontrado exitosamente:  
Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Listar Libros sin validaciones:

Texto

El contenido generado por IA puede ser incorrecto.

Forma, Rectángulo

El contenido generado por IA puede ser incorrecto.

Listar libros con validaciones; se agrega mensaje de que se encontraron libros las cantidad de libros encontrados y uno de no hay registros en la base de datos.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Al agregar usuarios me mostraba el json de prestamos juntos con los de usuarios lo que ocasionaba confusión:  
Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto. Texto

El contenido generado por IA puede ser incorrecto. Texto

El contenido generado por IA puede ser incorrecto.

Forma, Rectángulo

El contenido generado por IA puede ser incorrecto.

Agregar usuario con validaciones: Texto

El contenido generado por IA puede ser incorrecto.

Forma, Rectángulo

El contenido generado por IA puede ser incorrecto.

Eliminar Usuario sin validación:  
Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Patrón de fondo

El contenido generado por IA puede ser incorrecto. Texto

El contenido generado por IA puede ser incorrecto.

Eliminar Usuario con validacion: se añade mensaje de usuario eliminado correctamente e incorrectamente , al igual que un mensaje en el caso de que se quiera eliminar un usuario con un prestamo, tiene que devolver primero el libro para ser eliminado.

Texto

El contenido generado por IA puede ser incorrecto. Patrón de fondo

El contenido generado por IA puede ser incorrecto.

Actualizar usuario sin validaciones

Texto

El contenido generado por IA puede ser incorrecto.

Actualizar usuario con validaciones:  
Texto

El contenido generado por IA puede ser incorrecto. Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Patrón de fondo, Rectángulo

El contenido generado por IA puede ser incorrecto. Texto

El contenido generado por IA puede ser incorrecto.

Traer un usuario sin validaciones: Patrón de fondo

El contenido generado por IA puede ser incorrecto.Texto

El contenido generado por IA puede ser incorrecto.

Traer un usuario con validaciones: se agrega mensaje de usuario encontrado exitosamente y uno de usaurio no encontrado:  
Texto

El contenido generado por IA puede ser incorrecto. Texto

El contenido generado por IA puede ser incorrecto.

Trear a todos los usuarios sin validaciones:  
Texto

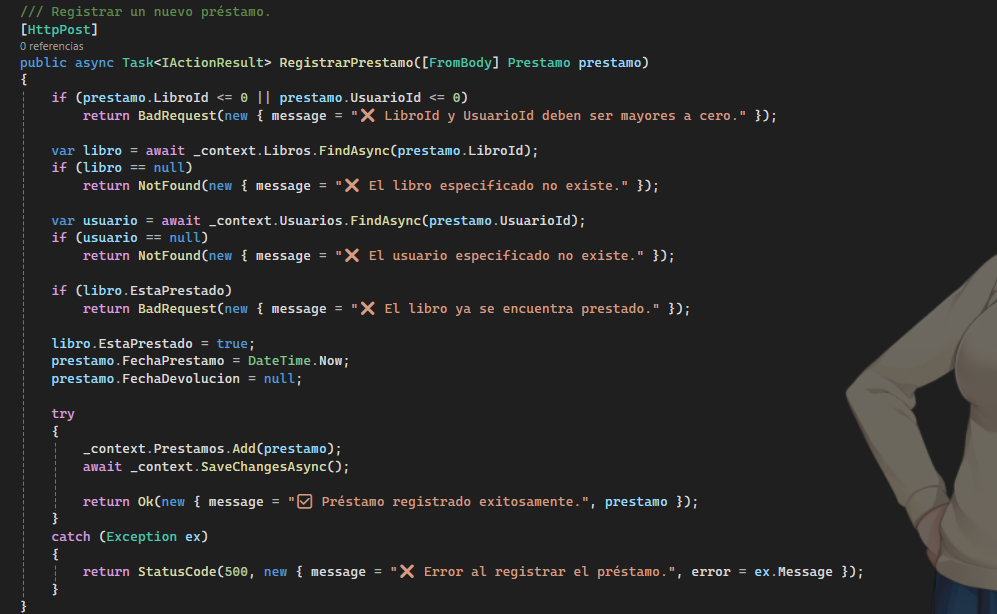
El contenido generado por IA puede ser incorrecto. Forma, Rectángulo

El contenido generado por IA puede ser incorrecto.

Traer a los usuarios con validaciones: se agrega mensaje en caso de que no encuentre usuarios en el sistema y un mensaje en caso de que enncuentre Texto

El contenido generado por IA puede ser incorrecto. Forma, Rectángulo

El contenido generado por IA puede ser incorrecto.

Agregar Prestamo con sus validaciones:  Texto

El contenido generado por IA puede ser incorrecto.

Devolucion de prestamo con validaciones: Texto

El contenido generado por IA puede ser incorrecto.

Pantalla de computadora con letras

El contenido generado por IA puede ser incorrecto.

Eliminar prestramos con validaciones: Texto

El contenido generado por IA puede ser incorrecto. Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Obtener un prestamo por el ID del prestamo con validaciones: Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto. Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Obtener un prestamo por ID del usuario con validaciones: Texto

El contenido generado por IA puede ser incorrecto. Escala de tiempo

El contenido generado por IA puede ser incorrecto. Texto

El contenido generado por IA puede ser incorrecto.

Obtener todos los prestamos con validaciones:  
Texto

El contenido generado por IA puede ser incorrecto. Aplicación, Teams

El contenido generado por IA puede ser incorrecto. Texto, Forma, Rectángulo

El contenido generado por IA puede ser incorrecto.