FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

Proyecto Final

ISWZ2103 - Programación IV

Ingeniería de Software | udla

RESULTADO DE APRENDIZAJE DE LA CARRERA:

RC4 Reconoce responsabilidades éticas y profesionales en situaciones de ingeniería y emite juicios informados, que deben considerar el impacto de las soluciones de ingeniería en contextos globales, económicos, ambientales y sociales.

INDICADORES DE DESEMPEÑO:

- Análisis de un dilema ético.
- Evaluación del impacto de las soluciones.
- Planteamiento de conclusiones.

OBJETIVO PROPUESTO DE LA CONSIGNA:

El objetivo de la actividad es que el estudiante sea capaz de identificar los posibles dilemas éticos a considerar en el diseño de una solución informática que procesa y almacena datos de forma permanente.

INDICACIONES:

El equipo de estudiantes debe trabajar en una solución de software para resolver el problema propuesto al inicio del semestre, mediante la construcción de una aplicación web que permita procesar y almacenar los datos relacionados con el tema del proyecto propuesto.

El equipo de estudiantes debe identificar los principales dilemas éticos relacionados con la confidencialidad de los datos que se manejan dentro de la aplicación web, para luego escoger uno y plantear una solución a dicho dilema.

Fecha de entrega: **26/01/2025**

Autores: **Paulo Olivo, Santiago Conlago, Julián Torres**

Título del proyecto: **KartHub**

Descripción técnica de la propuesta de proyecto, cumpliendo las indicaciones técnicas señaladas a lo largo del semestre.

1. Introducción

KartHub es una aplicación diseñada para la compra y venta segura sobre vehículos, integrando funcionalidades de gestión local y acceso a datos externos. Este proyecto busca satisfacer las necesidades de usuarios interesados en consultar especificaciones técnicas y realizar operaciones de CRUD sobre datos vehiculares, todo desde una aplicación móvil intuitiva y funcional. La solución se desarrolla cumpliendo con los principios de arquitectura moderna y patrones de diseño como MVC y MVVM.

2. Arquitectura del Sistema

El sistema utiliza una arquitectura modular basada en:

- **Backend:** Desarrollado en ASP.NET Core, incluye una aplicación web MVC para la gestión de la base de datos y una API local para exponer los datos.
- **Frontend:** Una aplicación móvil en .NET MAUI que utiliza el patrón MVVM para separar la lógica de negocio, la presentación y los datos.
- **API pública:** Se integra con la API de Ninja Cars (https://api-ninjas.com/api/cars) para obtener información externa sobre vehículos.

El siguiente diagrama representa la interacción entre los componentes principales:

1. **Base de datos SQL Server:** Contiene los datos principales creados por la aplicación MVC.
2. **API local:** Expuesta por el backend para que la aplicación móvil acceda a los datos.
3. **SQLite:** Base de datos local utilizada en la aplicación móvil para persistencia y acceso offline.

3. Backend

3.1. Aplicación Web MVC La aplicación web utiliza el enfoque 'Code First' para la creación de la base de datos en SQL Server. Entre las entidades principales destacan:

- **Vehículo:** Contiene información básica como marca, modelo, año y tipo de motor.
- **Usuario:** Gestiona los datos de los propietarios o interesados en vehículos.

Relaciones entre entidades:

- Un usuario puede estar asociado con varios vehículos.
- Los vehículos pueden tener especificaciones técnicas adicionales derivadas de la API Ninja Cars.

3.2. Web API La API local, construida en ASP.NET Core, proporciona endpoints para realizar operaciones CRUD sobre la base de datos. Ejemplo de endpoints:

- `GET /api/vehiculos`: Devuelve una lista de vehículos almacenados.
- `POST /api/vehiculos`: Agrega un nuevo vehículo a la base de datos.
- `PUT /api/vehiculos/{id}`: Actualiza los datos de un vehículo existente.
- `DELETE /api/vehiculos/{id}`: Elimina un vehículo.

4. Frontend

4.1. Estructura MVVM en .NET MAUI La aplicación móvil sigue el patrón MVVM:

- **View:** Diseño de interfaces gráficas responsivas para mostrar información de vehículos y permitir interacciones.
- **ViewModel:** Contiene la lógica de negocio, gestiona los datos obtenidos de la API local y de la base de datos SQLite.
- **Model:** Define las clases y estructuras para representar los datos vehiculares.

4.2. Consumo de APIs La aplicación móvil integra:

- **API local:** Para obtener y actualizar información gestionada por el backend.
- **API pública Ninja Cars:** Para consultar información externa sobre especificaciones técnicas de vehículos, como consumo de combustible y potencia.

4.3. Base de Datos Local (SQLite) Se utiliza SQLite para persistir datos relevantes en el dispositivo. Ejemplo:

- Al consultar un vehículo mediante la API Ninja Cars, los datos pueden almacenarse localmente y estar disponibles en modo offline.
- Operaciones CRUD en la base de datos local:
    - Crear un registro de vehículo.
    - Leer registros existentes.
    - Actualizar información almacenada.
    - Eliminar registros innecesarios.

5. Flujo de Trabajo

El flujo de datos entre los componentes se describe de la siguiente manera:

1. El usuario realiza una búsqueda de vehículos en la aplicación móvil.
2. La aplicación consulta la API pública Ninja Cars para obtener datos detallados.
3. Los datos relevantes se almacenan en SQLite para uso offline.

4. Si se desea agregar un vehículo, la aplicación envía los datos al backend a través de la API local.
5. La información se almacena en la base de datos SQL Server.
6. La aplicación permite consultar, editar o eliminar vehículos, sincronizando los cambios con el backend.

6. Tecnologías y Herramientas Utilizadas

- **Lenguajes y Frameworks:** C#, ASP.NET Core, .NET MAUI, Entity Framework Core.
- **Bases de datos:** SQL Server (backend), SQLite (frontend).
- **APIs:** API local (backend), API Ninja Cars (externa).
- **Herramientas:** Visual Studio, GitHub para control de versiones y colaboración.

7. Conclusiones

KartHub representa una solución robusta y escalable para la gestión de información vehicular, cumpliendo con los principios del diseño MVC/MVVM. La integración de APIs y bases de datos asegura una experiencia fluida tanto online como offline. Además, el uso de tecnologías modernas permite una evolución futura del sistema para adaptarse a nuevas necesidades de los usuarios.

URL AL REPOSITORIO: https://github.com/JulianDTA2/SPJ_ProyectoMVC

DESCRIPCIÓN DETALLADA DE:

- **Identificación y análisis del dilema ético en el caso planteado. ¿Cuáles son las responsabilidades profesionales y éticas relacionadas a su proyecto de software?**

En el desarrollo del proyecto KartHub, surgen varios dilemas éticos relacionados con el manejo de datos, la privacidad de los usuarios, y el impacto social del software. El análisis de los principales dilemas éticos incluye:

1. **Privacidad y seguridad de los datos:**
    - **Dilema:** La aplicación utiliza datos proporcionados por la API pública de Ninja Cars y datos personales ingresados por los usuarios. Existe el riesgo de exposición o uso indebido de esta información.
    - **Resolución:** Implementar medidas de seguridad como cifrado de datos, autenticación segura y cumplimiento de regulaciones como el GDPR (General Data Protection Regulation).
2. **Veracidad y transparencia de la información:**
    - **Dilema:** Los datos obtenidos de la API pública podrían contener errores o ser incompletos, lo que podría inducir a los usuarios a tomar decisiones incorrectas.
    - **Resolución:** Informar claramente que algunos datos provienen de fuentes externas, y establecer mecanismos de validación para garantizar la precisión de la información más relevante.
3. **Impacto social del software:**
    - **Dilema:** Si KartHub promueve el consumo excesivo de vehículos, podría contribuir negativamente al impacto ambiental.
    - **Resolución:** Incorporar funcionalidades que fomenten la sostenibilidad, como mostrar datos sobre eficiencia energética de los vehículos y opciones ecológicas.

Responsabilidades profesionales y éticas relacionadas al proyecto

1. **Cumplir con las regulaciones legales:** Asegurarnos que la aplicación cumple con leyes de privacidad y protección de datos, como la GDPR o leyes locales.
2. **Garantizar la seguridad del usuario:** Proteger la información personal de los usuarios contra ataques cibernéticos o usos no autorizados.
3. **Fomentar la honestidad y la transparencia:** Proporcionar datos precisos y advertir sobre posibles limitaciones o errores en la información obtenida de la API.

4. **Compromiso con la calidad:** Asegurar que el software esté bien diseñado, probado y sea confiable para evitar inconvenientes o riesgos para los usuarios.

- **Evaluación del impacto de la solución de ingeniería en su proyecto de software. ¿Cuáles serían las implicaciones si existieran problemas entre los actores involucrados en su proyecto?**

El impacto de la solución en el proyecto KartHub es significativo, ya que afecta tanto a los usuarios finales como a los colaboradores técnicos y socios externos. Un análisis del impacto y las posibles implicaciones de problemas entre los actores involucrados se describe a continuación:

Impactos:

1. **Privacidad y seguridad de los datos:** Vulnerabilidades podrían exponer información personal de los usuarios, generando problemas legales y éticos.
2. **Dependencia de la API externa (Ninja Cars):** Fallos en la API pública podrían dejar la aplicación sin acceso a datos clave, afectando su funcionalidad principal.
3. **Falta de comunicación en el equipo:** Malentendidos o documentación insuficiente podrían generar retrasos en el desarrollo y aumento de costos.

Recomendaciones:

1. **Implementar medidas de seguridad:** Usar cifrado de datos, autenticación segura y cumplir con regulaciones como el GDPR para proteger la privacidad de los usuarios.
2. **Diseñar mecanismos de respaldo:** Incluir cachés locales o notificaciones claras en caso de fallos en la API externa, asegurando una experiencia de usuario fluida.
3. **Establecer reuniones y documentación claras:** Mantener al equipo alineado mediante reuniones regulares, actas y documentación detallada para evitar malentendidos.

- **Su conclusión ética y profesional sobre el manejo de datos en el contexto de su proyecto de software.**

En el contexto del proyecto KartHub, el manejo de datos requiere un enfoque ético y profesional que priorice la privacidad, seguridad y transparencia. Como desarrolladores, tenemos la responsabilidad de garantizar que los datos de los usuarios y de las fuentes externas sean tratados de forma ética, cumpliendo con normativas legales y estándares de la industria.

La confidencialidad de la información personal de los usuarios debe estar protegida mediante el uso de técnicas de cifrado y controles de acceso. Además, se debe informar de manera clara a los usuarios sobre cómo se recopilan, procesan y almacenan sus datos, asegurando su consentimiento informado.

En conclusión, un manejo de datos ético y profesional no solo garantiza el cumplimiento de leyes y normativas, sino que también fortalece la confianza de los usuarios y el éxito a largo plazo del proyecto.