**Title**: **EcoVital** - A Mobile Application for Health and Wellness with Intelligent Chatbot

**Introduction**: Good day, everyone. Today, I am excited to present my project, EcoVital, a mobile application designed to enhance health and wellness. EcoVital stands out by incorporating an advanced chatbot powered by artificial intelligence, specifically utilizing the OpenAI API. This intelligent chatbot provides personalized health-related advice, answering users' questions and offering recommendations suited to their individual needs. In addition, the app includes features like personalized reminders for important health-related actions and a system to track progress. Furthermore, EcoVital ensures user data security with secure login and registration processes. The project was chosen to leverage the latest AI technology to address the growing demand for accessible and personalized health management solutions. By combining AI technology with user-centric design, EcoVital aims to create a seamless and supportive experience for all users, promoting overall well-being.

**Explicación Detallada de Casos de Uso, Diagramas y Planificación del Proyecto (2.5 Minutos)**

**Planificación del Proyecto**

**Utilizando el método de diagrama de Scrum, el proyecto se dividió en seis fases principales:**

1. **Estudio de Viabilidad:**
   * Evaluación de la viabilidad técnica y de mercado: Se analizó la capacidad técnica del equipo y la demanda del mercado, considerando la API de OpenAI y .NET MAUI.
2. **Análisis:**
   * Creación de diagramas de casos de uso y clases: Visualización de interacciones y estructura interna.
   * Especificación de características del chatbot: Detalle de las funciones y capacidades del chatbot.
3. **Diseño:**
   * Diseño de la arquitectura de la aplicación y del chatbot: Estructura técnica y cohesión de componentes.
   * Creación de mockups y prototipos: Diseños preliminares de la interfaz de usuario para feedback temprano.
4. **Implementación:**
   * Codificación de la aplicación e integración con la API de OpenAI: Desarrollo del código base y funcionalidad del chatbot.
   * Desarrollo de funcionalidades de recordatorios y seguimiento de metas: Implementación de funciones clave.
5. **Pruebas:**
   * Pruebas unitarias, de integración y de experiencia de usuario: Asegurar el correcto funcionamiento y una buena experiencia de usuario.
   * Refinamiento basado en el feedback: Mejoras según las opiniones de los usuarios.
6. **Despliegue y Documentación:**
   * Configuración del entorno de demostración: Preparación para mostrar la funcionalidad completa.
   * Documentación del proyecto y preparación para la defensa: Compilación de material y preparación de la presentación.

**Casos de Uso**: EcoVital ofrece varias funcionalidades clave orientadas a mejorar la salud y el bienestar de los usuarios:

1. **Chatbot Inteligente**: Proporciona consejos personalizados de salud, responde a preguntas de los usuarios y mejora continuamente su interacción basado en IA.
2. **Recordatorios Personalizados**: Los usuarios pueden configurar recordatorios para actividades como tomar medicamentos, beber agua o hacer ejercicio.
3. **Seguimiento de Metas de Salud**: Permite a los usuarios establecer y seguir sus metas de salud y progreso.
4. **Autenticación de Usuarios**: Sistema seguro de inicio de sesión y registro para garantizar la privacidad y seguridad de los datos del usuario.

**Diagramas**:

1. **Diagrama de Casos de Uso**: Muestra cómo los usuarios interactúan con las funcionalidades principales de la aplicación, incluyendo la interacción con el chatbot, configuración de recordatorios, seguimiento de metas…
2. **Diagrama de Clases**: "El diagrama de clases muestra la estructura de nuestra aplicación, destacando las principales clases y sus relaciones. Permite visualizar cómo interactúan entre sí las diferentes partes del sistema."

**Dependencias en Models**

"Comencemos con la sección de Models:

UserInfo: Esta clase almacena la información básica del usuario y se relaciona con otras clases como SecurityQuestion y ChangePasswordRequest."

"UserInfo también se relaciona con UserGoals para seguir las metas de los usuarios y ActivityRecords para registrar las actividades realizadas. Además, hemos añadido Feedback, que permite a los usuarios enviar sus opiniones y sugerencias sobre la aplicación."

 **Dependencias en Services**

"En la sección de Services, tenemos:

ILoginRepository: Una interfaz que define métodos como Login, Register, y ChangePassword, y es implementada por LoginService, lo que permite abstraer la lógica de acceso a datos para la autenticación del usuario."

OpenAIService: Se conecta con Message para gestionar las interacciones del chatbot."

LoadingService: Proporciona métodos para mostrar u ocultar una interfaz de carga utilizada durante operaciones asíncronas."

**Dependencias en ViewModels**

"Todos los ViewModels heredan de BaseViewModel, que es la clase que proporciona funcionalidades comunes a todos los ViewModels. Esto incluye:

**Notificación de Cambios**: Implementa INotifyPropertyChanged para que las vistas se actualicen automáticamente cuando cambian las propiedades en el ViewModel."

**Gestión de Carga**: Incluye una propiedad \_isBusy para indicar si el ViewModel está realizando una operación que toma tiempo."

**Título de la Página**: Proporciona una propiedad \_title que puede ser usada para establecer el título de la vista correspondiente."

**Comandos Comunes**: Puede incluir comandos que son comunes a múltiples ViewModels, centralizando la lógica de comandos reutilizables."

"Cada ViewModel específico interactúa con ILoginRepository para la autenticación y la recuperación de información del usuario. Ejemplos incluyen ChangePasswordViewModel, AppShellViewModel, ForgotPasswordPageViewModel, entre otros."

 **Dependencias en Views**

* "Finalmente, en la sección de Views:
  + Las vistas (como AboutPage, ChangePasswordPage, ChatBotPage, etc.) se relacionan con BaseViewModel a través del patrón MVVM (Model-View-ViewModel), lo que permite que la interfaz de usuario se actualice dinámicamente en respuesta a los cambios de estado en los ViewModels."
  + ChatBotPage interactúa directamente con Message para mostrar y gestionar las respuestas del chatbot."

1. **Esquema de Base de Datos (MER)**: Representa la estructura de la base de datos, incluyendo tablas para información de usuarios, preguntas de seguridad, feedback y registros de actividades.

**Funcionalidad Detallada de la App (Aprox. 7 Minutos)**

**Descripción General de la Funcionalidad**:

1. **Control y Autenticación de Usuarios**:
   * **Registro e Inicio de Sesión**: Sistema seguro de registro e inicio de sesión con cifrado SHA-256.
   * **Recuperación de Contraseña**: Implementado mediante preguntas de seguridad para asegurar la facilidad y seguridad de uso.
2. **Desarrollo del Chatbot**:
   * **Interacción**: Los usuarios pueden hacer preguntas relacionadas con la salud, y el chatbot proporciona respuestas detalladas y personalizadas.
   * **Integración con la API de OpenAI**: El chatbot utiliza el modelo "gpt-3.5-turbo" para generar respuestas.
   * **Sistema de Relevancia de Palabras Clave**: Evalúa la relevancia de las entradas de los usuarios para asegurar respuestas precisas y útiles.
3. **Sistema de Recordatorios**:
   * **Recordatorios Personalizados**: Los usuarios pueden configurar recordatorios para diversas actividades de salud, asegurando que se mantengan en sus objetivos de bienestar.
   * **Integración de TimePicker**: Permite a los usuarios elegir horas específicas para sus recordatorios.
4. **Seguimiento de Metas**:
   * **Establecimiento y Seguimiento de Metas**: Los usuarios pueden establecer metas de salud y seguir su progreso con el tiempo.
   * **Indicadores Visuales de Progreso**: Barras de progreso y estadísticas para visualizar el logro de metas.
5. **Interfaz de Usuario**:
   * **Página Principal**: Mensaje de bienvenida, acceso rápido al chatbot y registro de actividades.
   * **Registro de Actividades**: Los usuarios pueden registrar sus actividades diarias y ver su progreso.
   * **Consejos de Salud**: Consejos diarios de salud presentados en la página principal.
6. **Funcionalidad de Administrador**:
   * **Gestión de Usuarios**: Los administradores pueden gestionar cuentas de usuarios y actualizar configuraciones del chatbot.
7. **Implementación Técnica**:
   * **.NET MAUI y Rider para el Desarrollo**: Elegido por sus capacidades multiplataforma y facilidad de integración.
   * **Azure para Hosting**: Asegura escalabilidad y seguridad para la aplicación.
   * **SQL Server**: Utilizado para gestionar y almacenar datos de usuarios de forma segura.

**Pruebas y Garantía de Calidad**:

* **Pruebas Unitarias**: Aseguraron que los componentes individuales funcionen según lo esperado.
* **Pruebas de Integración**: Verificaron las interacciones entre diferentes componentes.
* **Pruebas de Experiencia de Usuario**: Recogieron feedback de usuarios para refinar la usabilidad de la aplicación.

**Conclusión**: EcoVital está diseñado para proporcionar una plataforma interactiva y fácil de usar para que los usuarios gestionen su salud y bienestar de manera eficiente. Aprovechando tecnologías avanzadas de IA y manteniendo un fuerte enfoque en la experiencia del usuario, EcoVital tiene como objetivo promover hábitos saludables y ofrecer apoyo continuo a sus usuarios.