

Rubrica

Universidad
politécnica de
Tecámac



Unidad I

Grupo: 2822 IS

Octavo cuatrimestre

Ingeniería en software

Asignatura: Programación móvil I

Profesor: Emmanuel Torres Servín

Alumnos:

- Fernando Manuel Galván – 1320114003
- Fernando Brayan Mejía Gómez – 1320114054
- Eduardo Antonio Méndez Sandoval – 1320114070
- Miguel Ángel Martínez Castro - 1319104806

Índice

Descripción del problema.....	3
Justificación de selección de sistema operativo móvil.....	3
Justificación de lenguaje de programación móvil.	7
Funcionamiento de .NET MAUI.....	8
Proyecto único	10
Conclusión	13

Descripción del problema.

Juan tiene una empresa de patitos de hule, dicha empresa tiene 3 años y ya cuenta con 40 empleados, actualmente se encuentra en crecimiento constante, sin embargo el presupuesto sigue siendo limitado para cuestiones de inversión y mejora, por lo que ha perdido control de sus empleados en ciertas áreas de la empresa, por lo que Juan toma la decisión de contratar unos programadores para realizar una aplicación móvil en la cual pueda administrar de forma general su empresa, sin embargo no tiene conocimiento alguno sobre el mundo del desarrollo de software.

Justificación de selección de sistema operativo móvil.

El proyecto que se desarrolla para la empresa *patitos de hule S.A. de C.V.* se despliega en el sistema operativo distribuido por Google denominado Android específicamente en la versión *Android 10* lanzada el 3 de septiembre de 2019 como versión mínima para la aplicación móvil, permite una mayor compatibilidad a futuro con posteriores versiones de Android sin malgastar la compatibilidad con dispositivos con versiones antiguas. La razón por la que se desarrolla en la versión mínima de Android 10 es debido a una serie de factores importantes que se enlistan en lo siguiente:

- Mejora la gestión de notificaciones: Permite a los usuarios clasificarlas o priorizarlas.
- Mejor gestión en la batería: Permite ajustar las configuraciones de energía produciendo mayor rendimiento en la batería.
- Mejora de seguridad: Incluye protección contra ataques y verificación de identidad.
- Live Caption: Esta opción añadirá subtítulos de manera automática a cualquier tipo de contenido multimedia en vídeo o audio sin necesidad de conexión a Internet.
- Bubbles: Integrada dentro del sistema de notificación, estas bubbles o burbujas, esencialmente, flotan sobre otro contenido en la pantalla, permitiendo al usuario hacer cosas rápida y fácilmente.

- Soporte nativo de reconocimiento facial 3D.
- Gestión de permisos mejorada.
- Nuevos permisos de acceso en fotos de fondo, video y archivos de audio.
- Las aplicaciones de fondo ya no pueden saltar al primer plano.
- Panel de configuración flotante, que permite cambiar la configuración del sistema directamente desde las aplicaciones.

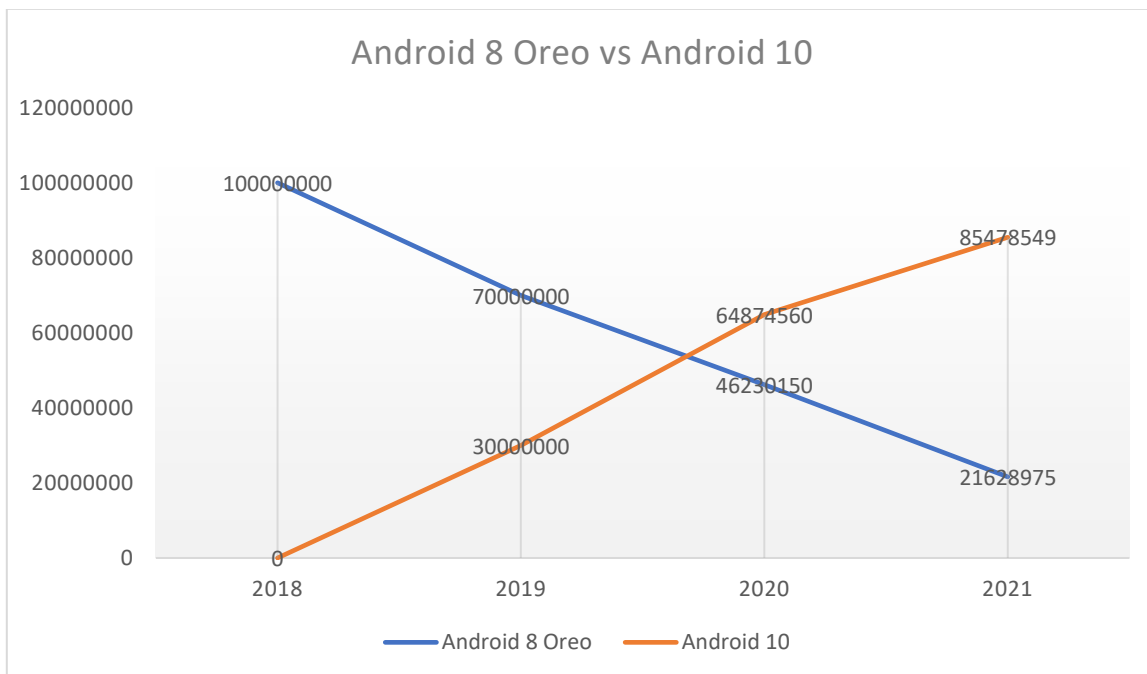
La aplicación móvil que se desarrolla para la empresa patitos de hule S.A. de C.V. fue impuesta en la versión Android 10 como requerimiento para la aplicación por las siguientes características que se muestran en la tabla comparativa Android 8 (Oreo) vs Android 10:

Tabla 1 Android 8 (Oreo) vs Android 10

Característica	Android 8 (Oreo)	Android 10
Interfaz de usuario	La interfaz de usuario de Android 8 se basa en Material Design 2.0 y es intuitiva y fácil de usar. Ofrece un diseño elegante y una experiencia de usuario coherente en todo el sistema.	La interfaz de usuario de Android 10 se basa en Material Design 2.5 y ofrece mejoras en la personalización y la accesibilidad. Además, incluye un modo oscuro incorporado.
Gestión de permisos	Android 8 ofrece permisos a nivel de aplicación, lo que significa que los usuarios deben dar permiso a una aplicación para acceder a determinados datos y funciones.	Android 10 ofrece permisos a nivel de tiempo de ejecución, lo que significa que los usuarios pueden controlar y revocar los permisos en cualquier momento.

Privacidad	Android 8 ofrece permisos limitados y opciones de privacidad básicas.	Android 10 incluye mejoras significativas en la privacidad y la seguridad, como la gestión de permisos a nivel de tiempo de ejecución, el cifrado de dispositivos por defecto y la detección de seguridad mejorada.
Compatibilidad con foldables (plegables)	Android 8 no admite dispositivos foldables.	Android 10 admite dispositivos foldables y ofrece una experiencia optimizada para ellos.
IA y asistencia	Android 8 incluye Google Assistant, un asistente virtual basado en IA.	Android 10 incluye una versión mejorada de Google Assistant con mejoras en IA y funciones adicionales.
Rendimiento	Android 8 ofrece un rendimiento general bueno, pero puede ser limitado en dispositivos antiguos o de gama baja.	Android 10 ofrece un rendimiento general mejorado, pero requiere dispositivos más actualizados para obtener la mejor experiencia.
Disponibilidad	Android 8 está disponible en una amplia variedad de dispositivos, pero no en todos ellos.	Android 10 está disponible en una amplia variedad de dispositivos, pero algunas características pueden no estar disponibles en todos ellos.

Además, que la mayoría de los dispositivos se encuentran en la versión Android 10 como versión mínima, ya que los dispositivos con versión Android 8 Oreo se encuentran en mínimos dispositivos como se muestra en la siguiente gráfica:



Debido a que se encuentra en una mejor versión y más estable Android 10, se ocupara como sistema operativo base para la aplicación que se desarrollara.

Justificación de lenguaje de programación móvil.

La aplicación que se desarrollara para la empresa *patitos de hule S.A. de C.V.* se desarrollara en el entorno de MAUI correspondiente al lenguaje de programación C#. Debido a que un marco de multiplataforma para crear aplicaciones móviles y de escritorio nativas con C# y XMAL.

Puede desarrollar aplicaciones que se pueden ejecutar en Android, iOS, macOS, y Windows desde una sola base de código compartida como se muestra en la imagen .NET Multi-plataform App UI.

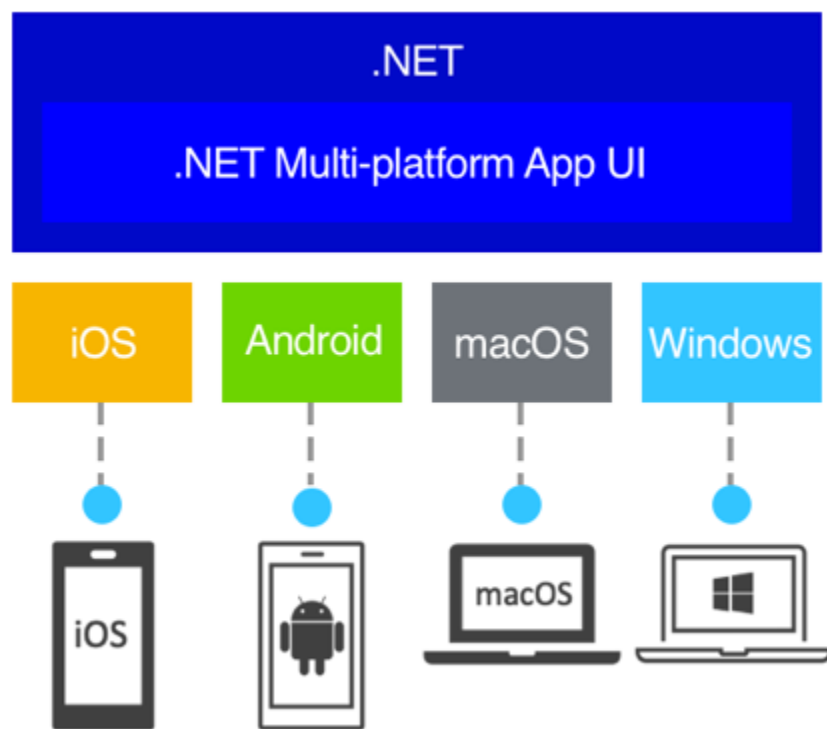


Figura 1 .NET Multi-plataform App UI

.NET MAUI es de código abierto y es la evolución de Xamarin.Forms, extendida desde escenarios móviles a escritorio, con controles de interfaz de usuario recompilados desde cero para el rendimiento y la extensibilidad. Uno de los objetivos clave de .NET MAUI es permitirle implementar la mayor parte de la lógica de la aplicación y el diseño de la interfaz de usuario en una única base de código.

.NET MAUI es para desarrolladores que quieren:

- Escriba aplicaciones multiplataforma en XAML y C#, desde una base de código compartida única en Visual Studio.
- Compartir el diseño de la interfaz de usuario entre plataformas.
- Comparta código, pruebas y lógica de negocios entre plataformas.

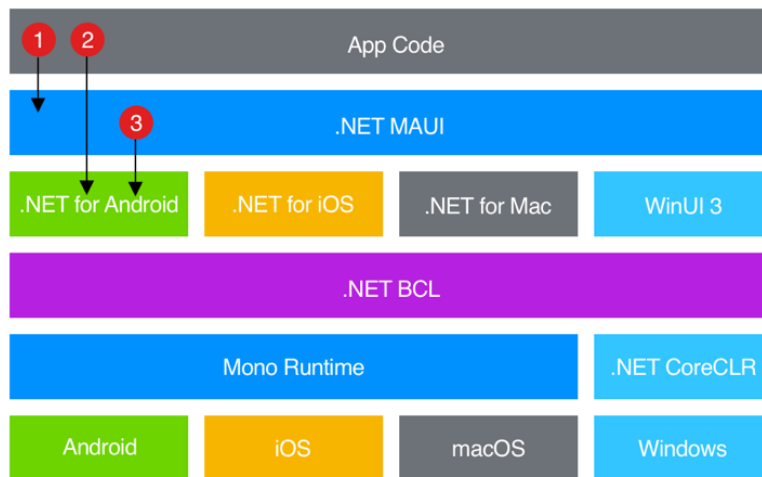
Funcionamiento de .NET MAUI

.NET MAUI unifica las API de Android, iOS, macOS y Windows en una sola API que permite una experiencia de desarrollador que se ejecuta en cualquier lugar y de solo una escritura, a la vez que proporciona acceso profundo a todos los aspectos de cada plataforma nativa.

.NET 6 o posterior proporciona una serie de marcos específicos de la plataforma para crear aplicaciones: .NET para Android, .NET para iOS, .NET para macOS y biblioteca de interfaz de usuario de Windows 3 (WinUI 3). Todos estos marcos tienen acceso a la misma biblioteca de clases base (BCL) de .NET. Esta biblioteca abstrae los detalles de la plataforma subyacente fuera del código. La BCL depende del entorno de ejecución de .NET para proporcionar el entorno de ejecución para el código. Para Android, iOS y macOS, mono implementa el entorno, una implementación del entorno de ejecución de .NET. En Windows, .NET CoreCLR proporciona el entorno de ejecución.

Aunque la BCL permite que las aplicaciones que se ejecutan en distintas plataformas compartan lógica de negocios común, las distintas plataformas tienen diferentes formas de definir la interfaz de usuario para una aplicación y proporcionan distintos modelos para especificar cómo se comunican e interoperan los elementos de una interfaz de usuario. Puede crear la interfaz de usuario para cada plataforma por separado mediante el marco específico de la plataforma adecuado (.NET para Android, .NET para iOS, .NET para macOS o WinUI 3), pero este enfoque requiere que mantenga una base de código para cada familia individual de dispositivos.

.NET MAUI proporciona un único marco para compilar las interfaces de usuario para aplicaciones móviles y de escritorio. En el diagrama siguiente se muestra una vista de alto nivel de la arquitectura de una aplicación MAUI de .NET:



Que proporciona .NET MAUI

.NET MAUI proporciona una colección de controles que se pueden usar para mostrar datos, iniciar acciones, indicar actividad, mostrar colecciones, seleccionar datos, etc. Además de una colección de controles, .NET MAUI también proporciona:

- Un motor de diseño elaborado para diseñar páginas.
- Varios tipos de páginas para crear tipos de navegación enriquecidos, como los cajones.
- Compatibilidad con el enlace de datos, para patrones de desarrollo más elegantes y fáciles de mantener.
- La capacidad de personalizar controladores para mejorar la forma en que se presentan los elementos de la interfaz de usuario.
- API multiplataforma para acceder a las características nativas del dispositivo. Estas API permiten a las aplicaciones acceder a características del dispositivo, como el GPS, el acelerómetro y los estados de la batería y la red. Para obtener más información, consulte API multiplataforma para las características del dispositivo.

- Funcionalidad de gráficos multiplataforma, que proporciona un lienzo de dibujo que admite formas e imágenes de dibujo y pintura, operaciones de redacción y transformaciones de objetos gráficos.
- Un único sistema de proyecto que usa varios destinos para dirigirse a Android, iOS, macOS y Windows. Para obtener más información, vea [.NET MAUI Single project \(Proyecto único de .NET MAUI\)](#).
- Recarga activa de .NET, para que puedas modificar el CÓDIGO FUENTE XAML y el código fuente administrado mientras se ejecuta la aplicación y, a continuación, observar el resultado de tus modificaciones sin volver a generar la aplicación. Para más información, consulte [Recarga activa de .NET](#).

Proyecto único

El único proyecto de .NET MAUI toma las experiencias de desarrollo específicas de la plataforma que normalmente se encuentran al desarrollar aplicaciones y las abstrae en un único proyecto compartido que pueda tener como destino Android, iOS, macOS y Windows.

Un único proyecto de .NET MAUI proporciona una experiencia de desarrollo multiplataforma simplificada y coherente, independientemente de las plataformas destinadas. El proyecto único de .NET MAUI proporciona las siguientes características:

- Un único proyecto compartido que pueda tener como destino Android, iOS, macOS y Windows.
- Selección simplificada de destino de depuración para ejecutar las aplicaciones .NET MAUI.
- Archivos de recursos compartidos dentro del único proyecto.
- Un único manifiesto de aplicación que especifica el título, el identificador y la versión de la aplicación.
- Acceso a las API y herramientas específicas de la plataforma cuando sea necesario.
- Un único punto de entrada de aplicación multiplataforma.

En el siguiente cuadro comparativo se diferencian algunas características importantes del entorno de MAUI y Android Studio, y justificar por qué la utilización de MAUI:

	CARACTERISTICAS	VENTAJAS	DESVENTAJAS
MAUI	<ul style="list-style-type: none"> • Acceso a sensores, como el acelerómetro, la brújula y el giroscopio en los dispositivos. • Capacidad de comprobar el estado de conectividad de red del dispositivo y detectar cambios. • Proporcione información sobre el dispositivo en el que se ejecuta la aplicación. • Copie y pegue texto en el Portapapeles del sistema, entre aplicaciones. • Elija uno o varios archivos del dispositivo. • Almacene los datos de forma segura como pares clave-valor. • Use motores de texto a voz integrados para leer texto del dispositivo. • Inicie flujos de autenticación basados en explorador que escuchen una devolución de llamada a una dirección URL registrada de aplicación específica. 	<ul style="list-style-type: none"> • Experiencia en un único proyecto de desarrollo: Con otros frameworks, los desarrolladores tienen que trabajar múltiples proyectos para cada una de las plataformas. .NET Maui soluciona este problema, ofreciendo trabajar en un proyecto único con una experiencia de desarrollo multiplataforma simplificada y coherente • Garantiza el mayor rendimiento posible: Al poder ser utilizada sobre cualquier plataforma no existe penalización por multidispositivo, algo muy frecuente en las aplicaciones híbridas. • Ahorro de tiempo y costes: Al poder ser utilizada en cualquier dispositivo no es necesario duplicar proyectos para adaptarlos a cada una de las plataformas. 	<p>En primer lugar, es un producto nuevo que acaba de salir y, si bien la versión publicada es estable, aún faltan algunos controles imprescindibles como los mapas. Aunque están próximos en el roadmap, no están disponibles cuando escribo este post.</p> <p>A veces, crear interfaces compartidas para Mobile y escritorio puede resultar complejo, por ser mundos diferentes. MAUI está pensado para conseguirlo, pero, a veces, crear interfaces para escritorio y Mobile, con las interacciones del usuario, es más difícil que crearlas separadas.</p>

ANDROID STUDIO

- | | | |
|--|---|---|
| <ul style="list-style-type: none"> • Soporte para programar aplicaciones para Android Wear (sistema operativo para dispositivos corporales como por ejemplo un reloj). • Herramientas Lint (detecta código no compatible entre arquitecturas diferentes o código confuso que no es capaz de controlar el compilador) para detectar problemas de rendimiento, usabilidad y compatibilidad de versiones. • Utiliza ProGuard para optimizar y reducir el código del proyecto al exportar a APK (muy útil para dispositivos de gama baja con limitaciones de memoria interna). • Integración de la herramienta <u>Gradle</u> encargada de gestionar y automatizar la construcción de proyectos, como pueden ser las tareas de testing, compilación o empaquetado. • Posibilita el control de versiones accediendo a un repositorio desde el que poder descargar Mercurial, Git, Github o Subversion. • Alertas en tiempo real de errores sintácticos, compatibilidad o rendimiento antes de compilar la aplicación | <ul style="list-style-type: none"> • Android Studio ha pasado a ser el entorno recomendado para el desarrollo de aplicaciones en Android, al tratarse de un IDE oficial de Google en colaboración con JetBrains (compañía de desarrollo software especializada en diseño de IDEs). • Android Studio permite la creación de nuevos módulos dentro de un mismo proyecto, sin necesidad de estar cambiando de espacio de trabajo para el manejo de proyectos, algo habitual en Eclipse. • Con la simple descarga de Android Studio se disponen de todas las herramientas necesarias para el desarrollo de aplicaciones para la plataforma Android. • Su nueva forma de construir los paquetes .apk, mediante el uso de Gradle, proporciona una serie de ventajas más acorde a un proyecto Java | <ul style="list-style-type: none"> • Aunque ya ha se ha lanzado la primera versión estable, la v1.0, al estar en una fase inicial, siempre es susceptible de introducirse más cambios que puedan provocar inestabilidad entre proyectos de diferentes versiones. • Curva de aprendizaje más lenta para nuevos desarrolladores de Android. • El sistema de construcción de proyectos Gradle puede resultar complicado inicialmente. • En comparativa con Eclipse, menor número de plugins. |
|--|---|---|

Conclusión

La plataforma de desarrollo MAUI es una nueva tecnología que ha surgió en el año 2022 por lo que es una buena opción para desarrollo de proyectos, ya que cuenta con el respaldo de una de las empresas mas importantes del mundo Microsoft, quien es la distribuidora de este software para la construcción de aplicaciones, además de que ofrece mejores ventajas sobre Android studio ya que al ser una aplicación multiplataforma con un único entorno de desarrollo se desenvuelve un desarrollo de código fuente que se ocupa en un único proyecto para cada una de las plataformas. Así mismo dejando en claro una de que para la empresa de patitos de hule S.A. de C.V. se requieren funciones de servicios que ofrece Xamarin, por lo que el entorno de C# ofrece estas funciones y servicios para la utilización en el proyecto.

Juan, dueño de una empresa de patitos de hule en crecimiento constante, contrata programadores para desarrollar una aplicación móvil para mejorar el control de su empresa. La aplicación se desarrollará en el sistema operativo Android 10 para aprovechar sus mejoras en gestión de notificaciones, batería, seguridad, subtítulos en tiempo real, permisos de acceso, y otros aspectos. Además, se utilizará el entorno de MAUI y el lenguaje de programación C# para desarrollar la aplicación móvil y multiplataforma. .NET MAUI es un marco de código abierto que permite a los desarrolladores implementar la mayor parte de la lógica de la aplicación y el diseño de la interfaz de usuario en una única base de código.