



UNIVERSIDAD ESTATAL DEL SUR DE MANABÍ
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

PROYECTO INTEGRADOR DE SABERES

TEMA DEL PROYECTO:
IMPLEMENTACIÓN DE UN APlicativo MÓVIL
INFORMATIVO DE CURSOS CERTIFICADOS MEDIANTE LA
HERRAMIENTA VISUAL STUDIO (MAUI) PARA ESCUELA DE
PESCA-MANTA

AUTORES:
ANCHUNDIA ANCHUNDIA JORDY ABRAHAM
CASTRO VALENCIA MISHELLE FERNANDA
GARCIA PANCHANA GEMA JULISSA
MARCILLO TUMBACO CRUZ MARIA
ZAMBRANO FIGUEROA OSCAR JESUS

TUTOR:
ING. KLEBER MARCILLO PARRALES. MG.

JIPIJAPA - MANABÍ – ECUADOR

PI 2023

INDICE

Contenido

1. INTRODUCCIÓN.....	4
2. PROBLEMATIZACION.....	5
 2.1 CONTEXTUALIZACION DEL PROBLEMA.....	5
 2.2 FORMULACION DEL PROBLEMA.....	5
 2.3 OBJETIVOS.....	5
2.3.1 General	5
2.3.1 Específicos	5
 2.4 JUSTIFICACIÓN	5
 2.5 HIPÓTESIS	6
3. METODOLOGÍA	7
3.1 Métodos Empíricos.....	7
3.2 Métodos de Inspección	7
3.2.1 Evaluación heurística	7
3.3 Método bibliográfico	7
4. MARCO TEORICO.....	9
4.1 ¿Qué es aplicación móvil?	9
4.1.1 Plataformas móviles para bibliotecas (webs móviles)	9
4.1.2 Métodos aplicados al desarrollo de aplicaciones móviles.....	10
4.1.3 Fases de los proyectos de desarrollo de aplicaciones móviles	10
4.1.4 Usabilidad	11
4.1.5 Usabilidad de las aplicaciones móviles	13
4.2 Lenguajes de Programación	14
4.2.1 ¿Qué es un lenguaje de programación?	14
4.3 Clasificación.....	16
4.3.1 Lenguaje máquina	16
4.3.2 Lenguajes de bajo nivel.....	16
4.3.3 Lenguaje de alto nivel	16
4.4 ¿Qué es C#?	17
4.4.1 C# dónde se utiliza	18
4.4.2 Desarrollo de aplicaciones de escritorio con WPF	19
4.5.1 Historia y evolución de Visual Studio	19
4.5.2 Características de Visual Studio.....	20
4.5.3 Ventajas de Visual Studio	21

4.6 ¿Qué es un IDE?	22
4.7 ¿Qué es .NET MAUI?	23
4.7.1 Conocimientos necesarios para desarrollar aplicaciones en .NET MAUI.....	23
4.7.2 Herramientas	23
4.7.3 Consejos prácticos para comenzar en el desarrollo de aplicaciones en .NET MAUI	23
4.8 Diseño de apps o aplicaciones móviles	25
4.8.1 Ventajas.....	25
4.8.2 Proceso de diseño de la aplicación	25
4.9 Diseño y Esquematización del Aplicativo Móvil a desarrollar	31
5. RESULTADOS.....	35
6. CONCLUSIONES Y RECOMENDACIONES	43
6.1 Conclusión.....	43
6.2 Recomendaciones	43
7. BIBLIOGRAFÍA	45
8. ANEXOS	47

1. INTRODUCCIÓN

La formación y capacitación en la industria pesquera son elementos fundamentales para el desarrollo sostenible de esta actividad. La oferta de cursos certificados y oportunidades de formación continua juega un papel crucial en la preparación de futuros pescadores y en el crecimiento profesional de quienes ya están involucrados en la sección pesquera. Sin embargo, a menudo se observa una problemática en cuanto al acceso y la difusión efectiva de estos programas debido a la falta de herramientas tecnológicas educativas adecuadas. En este sentido, la implementación de un aplicativo móvil informativo de cursos certificados brinda acceso a una amplia variedad de cursos especializados en pesca, diseñados para cualquier persona mayor de edad interesada en adquirir habilidades y conocimientos en esta apasionante disciplina. A través de este proyecto abarcaremos la creación y desarrollo de esta aplicación educativa, que busca conectar a entusiastas de la pesca con una oferta educativa de calidad, el objetivo es proporcionar una herramienta interactiva y accesible que facilite la búsqueda de información detallada sobre los cursos disponibles, sus contenidos, requisitos y fechas de inicio.

La aplicación permitirá a los usuarios explorar una amplia gama de programas de formación, desde cursos básicos para principiantes hasta cursos avanzados para pescadores experimentados que deseen mejorar sus habilidades y obtener certificaciones reconocidas, por medio de una aplicación intuitiva y amigable para que cualquier persona pueda acceder a información relevante y tomar decisiones informadas sobre su educación en pesca, con un enfoque especial en la seguridad de la información.

Esta aplicación contribuirá al fortalecimiento de la comunidad pesquera en la ciudad de Manta, al proporcionar una vía para adquirir habilidades especializadas y certificadas en esta industria vital para la región. Además, la seguridad de la información se considera un aspecto fundamental de esta plataforma, garantizando que los datos de los usuarios, incluyendo su progreso y registros de cursos, estén protegidos de manera sólida contra posibles amenazas ciberneticas.

Al unir estos elementos, la implementación exitosa del aplicativo móvil informativo puede convertirse en un valioso recurso educativo que beneficiará tanto a la comunidad pesquera de Manta como a aquellos que buscan adquirir conocimientos y habilidades especializadas en esta actividad, sin comprometer la confidencialidad y la integridad de sus datos personales y de formación. El presente proyecto tiene el propósito de aportar a

la mejora del acceso y difusión efectiva de los programas de formación de la Escuela de Pesca, apoyando así el desarrollo de la industria pesquera en la región y protegiendo la información crítica de los usuarios.

2. PROBLEMATIZACION

2.1 CONTEXTUALIZACION DEL PROBLEMA

La formación y capacitación en la industria pesquera son elementos fundamentales para el desarrollo sostenible de esta actividad. En una Escuela de Pesca, la oferta de cursos certificados y oportunidades de formación continua juega un papel crucial en la preparación de futuros pescadores y en el crecimiento profesional de quienes ya están involucrados en la sección pesquera. Sin embargo, a menudo se observa una problemática en cuanto al acceso y la difusión efectiva de estos programas debido a la falta de herramientas tecnológicas educativas adecuadas. En este sentido, la implementación de un aplicativo móvil informativo podría ser una solución potencial para abordar este desafío.

2.2 FORMULACION DEL PROBLEMA

¿De qué manera la calidad del software se relaciona con la formación de cursos a futuros pescadores en la Escuela de Pesca Manta?

2.3 OBJETIVOS

2.3.1 General

Implementar un aplicativo móvil informativo acerca de cursos certificados mediante la herramienta visual studio (Maui) para la escuela de pesca Manta.

2.3.1 Específicos

- Analizar los requisitos y objetivos del aplicativo móvil dentro de la Escuela de Pesca Manta.
- Determinar las herramientas tecnológicas a utilizar para el funcionamiento del aplicativos.
- Desarrollar una interfaz de usuario intuitiva dentro del aplicativo que permita acceder a la sección pesquera y recursos necesarios sobre la Escuela de Pesca.

2.4 JUSTIFICACIÓN

La justificación de este proyecto se fundamenta en la importancia de la formación y capacitación en la industria pesquera como elementos clave para el desarrollo sostenible

de esta actividad. Siendo la oferta de cursos certificados y oportunidades de formación continua fundamentales para la preparación de futuros pescadores y el crecimiento profesional de quienes ya están involucrados en la sección pesquera, es evidente la necesidad de abordar la problemática existente en cuanto al acceso y la difusión efectiva de estos programas debido a la falta de herramientas tecnológicas educativas adecuadas.

Por lo tanto, la implementación de un aplicativo móvil informativo de cursos certificados representa una solución estratégica para abordar esta problemática y superar las barreras existentes. Al desarrollar esta aplicación educativa, se pretende crear una plataforma interactiva y accesible que permita a los usuarios explorar una amplia gama de programas de formación, brindando información detallada sobre cada curso de manera clara y concisa. La interfaz intuitiva y amigable garantizará que cualquier persona, sin importar su nivel de experiencia o formación previa, pueda acceder a información relevante y tomar decisiones informadas sobre su educación en el ámbito pesquero.

2.5 HIPÓTESIS

La instrucción de cursos pesqueros basado en un aplicativo móvil para facilitar la interacción con los usuarios.

Variable dependiente

La interacción con los usuarios

Variable independiente

La instrucción de los cursos pesqueros basado en un aplicativo móvil

3. METODOLOGÍA

3.1 Métodos Empíricos

Este tipo de pruebas se desarrollan en laboratorios que intentan simular las condiciones reales bajo las cuales se usa la aplicación. Se le solicita al usuario que realice tareas previamente definidas, relacionadas con la funcionalidad de la aplicación. La interacción del usuario con la aplicación es evaluada para detectar los problemas que experimenta el usuario con el uso de la misma. Para realizar la evaluación se necesita analizar la información de usabilidad que se desprende de la interacción. Para capturar esa información, por un lado, están las métricas que se van obteniendo en tiempo real al interactuar el usuario con la aplicación. Por otro lado, se puede registrar la forma de realizar las tareas por parte del usuario mediante una grabación de video y de audio; también se pueden efectuar entrevistas o cuestionarios.

3.2 Métodos de Inspección

Inspección de usabilidad es el nombre genérico para un conjunto de formas rentables de evaluación de las interfaces de usuario para encontrar problemas de usabilidad, son muy informales y fáciles de usar. Este método consiste en formar un grupo de expertos en usabilidad que analizan o inspeccionan la aplicación considerada. Estos realizan un informe comentando sobre distintos aspectos de usabilidad de la aplicación, basándose en su experiencia en el área. Este informe es utilizado para realizar los cambios o ajustes necesarios en la aplicación, para resolver los problemas indicados. Dos de los métodos más utilizados de este tipo son: evaluación heurística y recorrido cognitivo

3.2.1 Evaluación heurística

Un grupo de evaluadores inspecciona el diseño de la interfaz basándose en un conjunto de heurísticas de usabilidad (principios de usabilidad establecidos). La evaluación heurística es fácil de realizar, económica y capaz de encontrar varios problemas de usabilidad. Sin embargo, puede no encontrar problemas específicos del dominio. Es por eso por lo que el uso de apropiadas heurísticas es muy significativo.

3.3 Método bibliográfico

La investigación bibliográfica o documental consiste en la revisión de material bibliográfico existente con respecto al tema a estudiar. Se trata de uno de los principales pasos para cualquier investigación e incluye la selección de fuentes de información

bibliográfica y documental. Se le considera un paso esencial porque incluye un conjunto de fases que abarcan la observación, la indagación, la interpretación, la reflexión y el análisis para obtener bases necesarias para el desarrollo del estudio de sobre la temática de dar una explicación clara y precisa de todo lo que se ha relacionado al tema tratado.

4. MARCO TEORICO

4.1 ¿Qué es aplicación móvil?

Una aplicación móvil consta esencialmente de dos partes: las aplicaciones nativas y las webs móviles. Sin importar el tipo de aplicación que se decida usar, ambas deben proporcionar la misma calidad de información a los usuarios que las utilizan. Sin embargo, es importante considerar algunos aspectos a la hora de evaluar la calidad en el funcionamiento de las aplicaciones, como los espacios restringidos de navegación y el elevado costo de su desarrollo. Cuando se desea desarrollar aplicaciones se debe tomar en cuenta que los lenguajes de programación para el desarrollo de aplicaciones varían de un sistema operativo a otro; esto conlleva que los diseñadores deban plantear varias veces la misma programación para ser utilizada en distintos dispositivos. (Garita-Araya, 2013)

Las aplicaciones móviles han ido incursionando a partir de inserción de lo que se denomina Smartphone a nuestra cotidianidad, donde cada vez es más frecuente su utilización para comunicarnos y ejecutar tareas de distinta índole. Para entender que es una aplicación móvil, se puede decir que las aplicaciones móviles son programas diseñados para ejecutarse en teléfonos, tablets y otros dispositivos móviles, y que, permiten al usuario realizar actividades profesionales, acceder a servicios, mantenerse informado, entre otras posibilidades y usos, según sea el fin para el cual se desarrollan.

Con el ingreso al mercado de empresas como iPhone, Google, Samsung, Huawei se dio un cambio en el modelo de negocio donde las aplicaciones móviles se hicieron rentables, tanto para los desarrolladores como para el mercado de aplicaciones en las tiendas App Store, Google Play entre otras tiendas. En la actualidad se han desarrollado tres tipos de aplicaciones las cuales son: app nativa, aplicaciones web y aplicaciones híbridas, son el resultado de la combinación entre las aplicaciones nativas y las webs apps que se desarrollan usando tecnologías web tales como: HTML, CSS, JavaScript que se compila y empaqueta de tal forma que el resultado final es una app para dispositivos móviles. (Puetate, 2020)

4.1.1 Plataformas móviles para bibliotecas (webs móviles)

El paradigma en las bibliotecas ha cambiado en los últimos años. Antes de esta evolución para tener acceso a los productos y servicios de las unidades de información los usuarios debían solicitarlos de manera presencial; sin embargo, y en la actualidad, las

bibliotecas deben salir a la caza de sus usuarios, haciendo uso de varias plataformas disponibles en la Internet.

Esos productos y servicios pueden ser clasificados en varias categorías:

- Los servicios básicos, estos incluyen las llamadas telefónicas, información de la institución y resolución de consultas de referencia en mostrador.
- Los servicios en red, el acceso a los catálogos públicos, la referencia virtual y la formación de usuarios.
- Los servicios virtuales, como las guías informativas, las inducciones virtuales y el uso de las redes sociales.

4.1.2 Métodos aplicados al desarrollo de aplicaciones móviles

En el mundo del desarrollo de software existen muchos métodos de desarrollo, cada uno con sus puntos fuertes y sus puntos débiles. En el caso del desarrollo de aplicaciones móviles sucede lo mismo, y cuando os planteéis qué método elegir deberéis saber escoger en función de vuestras necesidades. Algunos de los métodos más conocidos son los siguientes:

- Modelo waterfall
- Desarrollo rápido de aplicaciones
- Desarrollo ágil (cualquiera de sus variantes)
- Mobile-D

4.1.3 Fases de los proyectos de desarrollo de aplicaciones móviles

Hemos visto que hay varios métodos existentes en el mercado para el desarrollo de aplicaciones móviles y que todos ellos se dividen en diferentes fases. Cada uno de estos métodos especifica lo que se debe hacer en cada fase, así como el nivel o los resultados que se requieren. Estas fases del desarrollo de aplicaciones móviles tendrán problemas y soluciones comunes, de modo que vamos a repasarlas.

Planificación

En la fase de planificación se intenta distribuir el tiempo y los recursos necesarios para poder llevar a cabo el proyecto, ya sea en una única planificación completa o en planificaciones más divididas. Durante las fases de planificación siempre se intenta conseguir la máxima precisión de las estimaciones, para minimizar los riesgos asociados al proyecto. (Vique, 2018)

Dificultades por el desconocimiento de la tecnología. Suele tratarse de tecnologías nuevas, en ocasiones desconocidas para los desarrolladores. Eso repercute en el tiempo de aprendizaje y puede ocasionar desviaciones por contratiempos no conocidos

Disponer de dispositivos reales. Para poder realizar un buen proyecto de desarrollo de aplicaciones móviles, es necesario poder probarlo en dispositivos reales. Para ello, se debe planificar una fase de pruebas reales.

Time to market. A la hora de planificar hay que tener muy en cuenta el que suele tardar una idea en llegar al mercado e intentar reducirlo al máximo, pues la competencia es muy grande.

Prototipado. Es importante planificar cuando se va a conseguir el primer prototipo (y, tal vez, la primera salida en beta), pues el prototipado rápido puede ser muy útil para este tipo de aplicaciones.

4.1.4 Usabilidad

Usabilidad en general tiene que ver con la forma en que se usa algún elemento (herramienta, dispositivo electrónico, etc.), es la facilidad con que se usa y si permite hacer lo que se necesita. Particularmente la usabilidad de una aplicación de software se refiere a la facilidad con que los usuarios pueden utilizar la misma para alcanzar un objetivo concreto. Este nivel de usabilidad no puede medirse o ser evaluado directamente, debido a que depende de diferentes factores

Efectividad: Está relacionada con la precisión y completitud con la que los usuarios utilizan la aplicación para alcanzar objetivos específicos. La calidad de la solución y la tasa de errores son indicadores de efectividad.

Eficiencia: Es la relación entre efectividad y el esfuerzo o los recursos empleados para lograr esta. Indicadores de eficiencia incluyen el tiempo de finalización de tareas y tiempo de aprendizaje. A menor cantidad de esfuerzo o recursos, mayor eficiencia.

Satisfacción: Es el grado con que el usuario se siente satisfecho, con actitudes positivas, al utilizar la aplicación para alcanzar objetivos específicos. La satisfacción es un atributo subjetivo, puede ser medido utilizando escalas de calificación de actitud. (Isabe, 2013)

Diferentes percepciones de usabilidad resultan en una variedad de estructuras de medidas, estas estructuras definen los atributos y sus métricas asociadas. Por ejemplo, Jakob Nielsen, define usabilidad en términos de cinco atributos: facilidad de aprendizaje, eficiencia, memorabilidad, errores y satisfacción. Nielsen puntuiza que la importancia de estos atributos variará en función del contexto y de los usuarios finales. Por ejemplo en un cajero automático, la facilidad de aprendizaje podría ser considerado el atributo principal a evaluar, mientras que en un sistema aeroportuario, la tolerancia y prevención de errores pasa a ser primordial. El término atributo de usabilidad es bastante ambiguo, esto genera ciertas diferencias, algunos autores usan diferentes nombres para el mismo atributo, tales como memorabilidad y facilidad de aprendizaje. Existen diferentes opiniones sobre que atributos pueden ser considerados y la manera de combinarlos para componer la usabilidad. Los siguientes son algunos de los atributos utilizados para medir el grado de usabilidad de una aplicación de software:

Facilidad de Aprendizaje: La facilidad con la que los usuarios alcanzan objetivos específicos la primera vez que utilizan la aplicación. La primer experiencia que tiene los usuarios con un nuevo sistema es la de aprender a usarlo.

Memorabilidad: La facilidad para memorizar la forma de utilizar la aplicación y alcanzar objetivos específicos, y la facilidad con que vuelven a utilizar la aplicación después de un tiempo. La curva de aprendizaje debe ser significativamente menor para un usuario que ya utilizó el sistema, que para uno que es la primera vez que lo va a utilizar.

Errores: Los errores que comete el usuario al utilizar la aplicación y la gravedad de estos. La aplicación debe producir la menor cantidad de errores posibles. Si se producen, es importante que se den a conocer al usuario de forma rápida y clara, además de ofrecer algún mecanismo para recuperarse de ese error.

Contenido: Aspectos relacionados a la distribución del contenido y de los formatos utilizados para mostrar información al usuario.

Accesibilidad: Consideraciones tenidas en cuenta por posibles limitaciones físicas, visuales, auditivas o de otra índole de los usuarios.

Seguridad: Capacidad para alcanzar niveles aceptables de riesgo. Disponibilidad de mecanismos que controlan y protegen la aplicación y los datos almacenados.

Portabilidad: Capacidad de la aplicación de ser transferida de un entorno a otro (diferentes plataformas).

Contexto: Relacionado a los factores o variables del entorno de uso de la aplicación

4.1.5 Usabilidad de las aplicaciones móviles

Las aplicaciones o productos de software cuando son lanzados al mercado se espera que tengan cierto grado de aceptación entre los usuarios, ese grado va a depender de las características particulares que cada usuario considere importantes. Desde el punto de vista de la Ingeniería de Software (SE: Software Engineering), una de las principales características que tiene que tener una aplicación para ser exitosa entre los usuarios es que sea de calidad. Resulta relevante para los desarrolladores de software poder medir esa calidad o realizar pruebas de calidad a las aplicaciones construidas, pero para poder medir se necesita saber qué es lo que hay que medir y cómo. La calidad del software es el grado que el software posee de una combinación deseada de atributos, esta combinación de atributos deberá ser claramente especificada. Definir calidad de software para un sistema es equivalente a definir una lista de atributos de calidad del software requeridos para ese sistema [1]. Utilizando esta definición se puede afirmar que lo que hay que medir más puntualmente son ciertos atributos del software relacionados a la calidad. Dentro de estos atributos uno de los considerados más importantes es la usabilidad, que indica la facilidad con la que un usuario puede usar una aplicación de software. Por lo tanto resulta de interés poder obtener una medida del grado de usabilidad que tiene una aplicación.

Los beneficios que se obtienen al realizar pruebas de usabilidad a las aplicaciones son los siguientes:

Mejorar el diseño del producto de software mediante la información obtenida de la retroalimentación.

Reducir los costos de desarrollo, al disminuir el número de cambios posteriores requeridos.

Reducir los costos de mantenimiento y soporte: Los sistemas que son fáciles de usar requieren menos entrenamiento, menos soporte para el usuario y menos mantenimiento.

Mejorar el uso: los sistemas que mejor se ajustan a las necesidades del usuario reducen el esfuerzo y mejoran la productividad y la calidad de las acciones.

Mejorar la calidad del producto: productos de mayor calidad de uso, son más competitivos en un mercado que demanda productos de fácil uso.

4.2 Lenguajes de Programación

4.2.1 ¿Qué es un lenguaje de programación?

Un lenguaje de programación es una forma de comunicarnos con una computadora, tablet o celular e indicarle qué queremos hacer.

Existen distintos tipos de lenguaje: principalmente de bajo nivel y de alto nivel. La diferencia se encuentra en lo cerca o lejos que estemos del hardware de nuestro equipo. Esta cercanía tiene que ver con el control que tengamos sobre el dispositivo, placa o controlador.

Encontramos diferentes lenguajes como C, C++, Java, PHP, Python, C#, ASP, entre otros.

Los programas habitualmente se diseñan o escriben a través de un entorno de desarrollo integrado (IDE por sus siglas en inglés). Un IDE habitualmente cuenta con un editor de texto (donde escribiremos nuestro código), el compilador y el depurador para el lenguaje que estemos utilizando y la correspondiente plataforma en la que luego correrá. Muchas veces cuenta con un constructor de interfaz gráfica (en inglés GUI). (2017)

En términos generales, un *lenguaje de programación* es una herramienta que permite desarrollar *software* o programas para computadora. Los lenguajes de programación son empleados para diseñar e implementar programas encargados de definir y administrar el comportamiento de los dispositivos físicos y lógicos de una computadora. Lo anterior se logra mediante la creación e implementación de algoritmos de precisión que se utilizan como una forma de comunicación humana con la computadora. A grandes rasgos, un lenguaje de programación se conforma de una serie de símbolos y reglas de sintaxis y semántica que definen la estructura principal del lenguaje y le dan un significado a sus elementos y expresiones. Programación es el proceso de análisis, diseño, implementación, prueba y depuración de un algoritmo, a partir de un lenguaje que compila y genera un código fuente ejecutado en la computadora.

La función principal de los lenguajes de programación es escribir programas que permiten

la comunicación usuario-máquina. Unos programas especiales (compiladores o intérpretes) convierten las instrucciones escritas en código fuente, en instrucciones escritas en lenguaje máquina (0 y 1). Los intérpretes leen la instrucción línea por línea y obtienen el código máquina correspondiente.

En cuanto a los compiladores, traducen los símbolos de un lenguaje de programación a su equivalencia escrita en lenguaje máquina (proceso conocido como compilar). Por último, se obtiene un programa ejecutable.

Para entender mejor la forma como se estructura un lenguaje de programación, observa la siguiente imagen (en este apunte se utilizará el lenguaje C). En particular, este lenguaje está caracterizado por ser de uso general, de sintaxis compacta y portable. Así, un lenguaje de programación es una herramienta informática que permite desarrollar programas para computadoras (Ceballos, 2004)

Antecedentes

Profesor de matemáticas e inventor en la universidad de Cambridge, Inglaterra, a mediados del siglo XIX, Charles Babbage fue el primero en concebir la idea de un lenguaje de programación, al predecir varias de las teorías en las que se basan las computadoras actuales.

Babbage desarrolló la idea de una máquina analítica programable que, por limitaciones tecnológicas de su época, no pudo ser construida. Junto con él, su colaboradora Ada Lovelace es considerada como la primera programadora de la historia, ya que escribió los primeros programas para la máquina concebida por Babbage en tarjetas perforadas, siguiendo una lógica de programación muy similar a la empleada en nuestros días. Estos programas nunca pudieron verse ejecutados debido a que la máquina no fue construida. Las técnicas empleadas por Babbage y Ada fueron seguidas por los primeros programadores de computadoras, quienes se valieron de tarjetas perforadas para introducir sus programas en las computadoras.

En 1823, con el apoyo del gobierno británico, se aprobó el proyecto de construcción de una máquina de diferencias. Esta máquina era un dispositivo mecánico diseñado para realizar sumas de forma repetitiva. Babbage abandonó el proyecto para dedicarse a su máquina analítica, influenciado por la creación de un fabricante de telas francés, Joseph

Marie Jacquard, que había desarrollado una máquina tejedora con la capacidad de reproducir patrones de tejidos, leyendo información codificada en tarjetas perforadas de

papel rígido.

Desde entonces, Babbage se propuso construir una máquina que efectuara cálculos matemáticos de precisión, empleando 20 dígitos, y que pudiera ser programada mediante tarjetas perforadas. Aun cuando esta idea quedó sólo en el proyecto, fue una contribución muy importante para el diseño y funcionamiento de las computadoras actuales.

4.3 Clasificación

Los circuitos microprogramables son sistemas digitales, lo que significa que trabajan con dos únicos niveles de tensión simbolizados con el cero (0) y el uno (1). Por eso, el lenguaje de máquina utiliza sólo dichos signos.

Un lenguaje de bajo nivel es trasladado fácilmente a lenguaje de máquina (la palabra *bajo* se refiere a la abstracción reducida entre el lenguaje y el *hardware*). Y los lenguajes de programación de alto nivel se caracterizan por expresar los programas de una manera sencilla.

4.3.1 Lenguaje máquina

Es el sistema de códigos interpretable directamente por un circuito microprogramable, como el microprocesador de una computadora. Este lenguaje se compone de un conjunto de instrucciones que determinan acciones que serán realizadas por la máquina. Y un programa de computadora consiste en una cadena de estas instrucciones de lenguaje de máquina (más los datos). Normalmente estas instrucciones son ejecutadas en secuencia, con eventuales cambios de flujo causados por el propio programa o eventos externos. El lenguaje máquina es específico de cada máquina o arquitectura de la máquina, aunque el conjunto de instrucciones disponibles pueda ser similar entre ellas.

4.3.2 Lenguajes de bajo nivel

Un lenguaje de programación de bajo nivel es el que proporciona poca o ninguna abstracción del microprocesador de una computadora. Consecuentemente, su trasladado al lenguaje máquina es fácil. El término *ensamblador* (del inglés *assembler*) se refiere a un tipo de programa informático encargado de traducir un archivo fuente, escrito en un lenguaje ensamblador, a un archivo objeto que contiene código máquina ejecutable directamente por la máquina para la que se ha generado.

4.3.3 Lenguaje de alto nivel

Los lenguajes de programación de alto nivel se caracterizan porque su estructura semántica es muy similar a la forma como escriben los humanos, lo que permite codificar

los algoritmos de manera más natural, en lugar de codificarlos en el lenguaje binario de las máquinas, o a nivel de lenguaje ensamblador.

Ejemplos de lenguajes de alto nivel

C++

Fue diseñado a mediados de la década de 1980 por Bjarne Stroustrup. Abarca dos paradigmas de la programación: la estructurada y la orientada a objetos.

Fortran

Fue desarrollado en la década de 1950 y es empleado activamente desde entonces. Acrónimo de *formula translator*, Fortran se utiliza principalmente en aplicaciones científicas y análisis numérico.

Java

Lenguaje de programación orientado a objetos desarrollado por *Sun Microsystems* a principios de 1990. Las aplicaciones Java están típicamente compiladas en un *bytecode*, aunque la compilación en código máquina nativo también es posible.

PHP

Es empleado frecuentemente para la creación de contenido para sitios web con los cuales se pueden programar las páginas HTML y los códigos de fuente. PHP es un acrónimo que significa PHP *hypertext pre-processor* (inicialmente, PHP *tools* o *personal home page tools*), y se trata de un lenguaje interpretado que se aprovecha para la creación de aplicaciones para servidores o contenidos dinámicos de sitios web. Últimamente también sirve para crear otro tipo de programas que incluyen aplicaciones con interfaz gráfica, librerías Qt o GTK+.

Python

Fue hecho por Guido van Rossum en 1990. En la actualidad se desarrolla como un proyecto de código abierto administrado por la *Python Software Foundation*. La última versión estable del lenguaje es la 2.5 (septiembre 2006).

4.4 ¿Qué es C#?

Es un lenguaje de programación multiparadigma desarrollado por Microsoft, que evoluciona de la familia de lenguajes C -como su nombre indica- tomando lo mejor de los

lenguajes C y C++ y que se asemeja mucho a lenguajes de alto nivel de abstracción como Java y JavaScript. Form parte de la plataforma .NET de Microsoft, una API que se ha convertido en una de las principales plataformas de desarrollo debido a la facilidad que ofrece para la construcción de todo tipo de aplicaciones multiplataforma sólidas y duraderas.

Se caracteriza principalmente por ser un lenguaje de programación orientado a objetos (POO) polivalente y fácil de aprender. Sin perder potencia original de C, da la posibilidad de acceder a bajo nivel al núcleo de los sistemas operativos, trabajar con punteros a memoria e interactuar con elementos físicos de los dispositivos.

La POO se basa en estructurar un programa en piezas simples y reutilizables llamadas clases para crear instancias individuales a partir de estas también conocidas como objetos. Buscando, de esta manera, dejar de centrarnos en la lógica pura de los programas, todo esto nos permite empezar a pensar en objetos y las relaciones o interacciones que existen entre ellos.

4.4.1 C# dónde se utiliza

C# es uno de los lenguajes más usados hoy en día por grandes y pequeñas compañías debido a su versatilidad y sencillez para el desarrollo de aplicaciones de cualquier índole que resultan eficientes y seguras.

Es importante decir que C# está diseñado para la plataforma .NET de Microsoft, pero puede permitir escribir programas para otras plataformas como Unix, Android, iOS, Windows Phone, Mac OS y GNU/Linux.

C# se puede utilizar para desarrollar todo tipo de aplicaciones: desde videojuegos hasta aplicaciones web como Stack Overflow. Algunas de los usos de C# más destacados son:

- Aplicaciones de escritorio, en consola o con interfaz gráfica usando WinForms, WPF, UWP o la nueva .NET MAUI.

- Aplicaciones para dispositivos móviles con Xamarin.
- Servicios de API y páginas web con ASP.NET.
- Microservicios con ASP.NET Core.
- Software embebido con .NET Nanoframework

- Servicios Web XML.
- Desarrollo de videojuegos con Unity

4.4.2 Desarrollo de aplicaciones de escritorio con WPF

En Tribalyte Technologies somos expertos en el desarrollo de aplicaciones de escritorio utilizando Windows Forms o WPF (Windows Presentation Foundation) para desarrollar la interfaz de usuario y C# para implementar el CodeBehind (Código ejecutable que implementa los requisitos funcionales) sobre la plataforma .NET de Microsoft.

Para el desarrollo de aplicaciones de escritorio utilizamos el patrón MVVM (modelo-vista-modelo de vista), un esquema de arquitectura que se caracteriza por tratar de desacoplar lo máximo posible la interfaz de usuario de la lógica de la aplicación. Esto se consigue separando la aplicación en tres componentes:

Modelo: representa la capa de datos donde se almacena la información necesaria.

Vista: componente encargado de mostrar la información al usuario y permitir a este último interactuar con ella.

Modelo de Vista: contiene toda la lógica existente por detrás de la interfaz de usuario.

Estas tecnologías ofrecen infinitas posibilidades ya que permiten desarrollar todo tipo de aplicaciones ejecutables en Windows de manera sencilla, personalizable, escalable y segura. Todo ello ha permitido que se introduzca en todo tipo de sectores dando muy buenos resultados. (GARCÍA, 2021)

4.5 ¿Qué es Visual Studio?

Visual Studio es un entorno de desarrollo integrado (IDE) desarrollado por Microsoft que permite a los programadores crear aplicaciones para distintas plataformas (incluyendo Windows, Android, iOS y Linux). Este software ofrece diversas herramientas para facilitar el proceso de desarrollo, como depuración de código, integración con sistemas de control de versiones, pruebas automatizadas, y asistencia en la escritura de código.

4.5.1 Historia y evolución de Visual Studio

Visual Studio es una de las herramientas más populares y utilizadas por los desarrolladores de todo el mundo. Este entorno de desarrollo integrado (IDE) ha evolucionado

significativamente desde su lanzamiento en 1997 y ha cambiado drásticamente la forma en que los desarrolladores crean aplicaciones.

La historia de Visual Studio comenzó en 1989, cuando Microsoft lanzó su primer entorno de desarrollo integrado llamado Visual Basic. Esta herramienta se convirtió rápidamente en una de las más populares para el desarrollo de aplicaciones de Windows y estableció los cimientos para lo que se convertiría en Visual Studio.

El primer lanzamiento de Visual Studio fue en 1997 y se denominó Visual Studio 97. Esta versión incluía un entorno de desarrollo integrado para varias plataformas, incluyendo Windows y el lenguaje de programación Visual Basic. Desde entonces, Microsoft ha lanzado regularmente nuevas versiones de Visual Studio, cada una con nuevas características y capacidades significativas para ayudar a los desarrolladores a crear aplicaciones de manera más rápida y eficiente.

4.5.2 Características de Visual Studio

Visual Studio cuenta con múltiples ediciones, incluyendo una versión gratuita llamada «Visual Studio Community» y versiones de pago con características adicionales. Además, es compatible con distintos lenguajes de programación como C++, C#, F#, Visual Basic, y Python, entre otros (2022)

Algunas de las características más relevantes de este software incluyen:

- **Entorno Integrado:** Visual Studio proporciona un entorno de desarrollo integrado que permite a los programadores escribir, depurar y probar el código en un mismo lugar, lo que facilita el proceso de desarrollo y permite ahorrar tiempo.
- **Lenguajes de Programación:** Este software es compatible con múltiples lenguajes de programación, lo que lo hace flexible y adaptable a diferentes necesidades. Algunos de los lenguajes de programación compatibles son C++, C#, F#, Visual Basic y Python.
- **Integración de Herramientas:** Visual Studio cuenta con integración de herramientas de terceros, lo que permite a los desarrolladores utilizar herramientas adicionales para complementar el proceso de desarrollo, como sistemas de control de versiones o pruebas automatizadas.

- **Depuración de Código:** El depurador de Visual Studio ofrece una amplia variedad de herramientas y características para ayudar a los desarrolladores a detectar y solucionar errores en el código de manera rápida y eficiente.
- **Plantillas y Emuladores:** Visual Studio viene con una variedad de plantillas predefinidas y emuladores que permiten a los desarrolladores crear aplicaciones de manera rápida y sin tener que empezar desde cero.

4.5.3 Ventajas de Visual Studio

Entre los beneficios de utilizar Visual Studio para desarrollo web podemos citar los siguientes:

- **Mayor productividad:** Visual Studio ofrece una amplia gama de herramientas y características integradas que permiten a los desarrolladores crear aplicaciones de manera más rápida y eficiente. Además, al ser un entorno de desarrollo integrado, ofrece una interfaz de usuario intuitiva y optimizada que ayuda a ahorrar tiempo en tareas repetitivas.
- **Mayor facilidad de uso:** El entorno integrado de Visual Studio hace que las tareas de desarrollo sean más simples y fáciles de administrar. Los programadores pueden integrar varias actividades relacionadas con el desarrollo, como escribir código, depuración y pruebas, todo en un mismo lugar.
- **Multiplataforma:** Visual Studio es compatible con múltiples plataformas, lo que permite a los desarrolladores trabajar en una variedad de sistemas operativos y dispositivos. Esto es especialmente útil para aquellos que desean crear aplicaciones para diferentes sistemas operativos como Windows, Android, iOS y Linux.
- **Lenguajes de programación:** Visual Studio admite múltiples lenguajes de programación, lo que lo hace más flexible y adaptable a diferentes necesidades. Algunos de los lenguajes de programación compatibles son C++, C#, F#, Visual Basic y Python, entre otros.
- **Integración con otras herramientas de desarrollo:** Visual Studio puede integrarse con otras herramientas de desarrollo de terceros, como sistemas de control de versiones, pruebas automatizadas y administración de tareas, lo que facilita el uso de múltiples herramientas en un solo entorno.

- **Depuración de código:** La herramienta de depuración de Visual Studio ofrece muchas características útiles para detectar y solucionar errores en el código, lo que ayuda a los programadores a encontrar y solucionar problemas en el código de manera rápida y eficiente.
- **Plantillas predefinidas y emuladores:** Visual Studio viene con una variedad de plantillas predefinidas y emuladores que permiten a los desarrolladores crear aplicaciones de manera rápida y sin tener que comenzar desde cero. Estas plantillas y emuladores se actualizan regularmente para agregar nuevas características y funcionalidades.

4.6 ¿Qué es un IDE?

Un entorno de desarrollo integrado (en inglés Integrated Development Environment) no es más que una plataforma compuesta por un grupo de herramientas de programación que *facilitan* al programador el desarrollo de un software determinado.

¿De qué forma el IDE cumple esto? Pues el programador evita introducir comandos excesivos para tareas como las siguientes:

- Compilar.
- Depurar.
- Interpretar.

¿Y cuáles son esas herramientas que utiliza el IDE para esas tareas? Por lo general, esta plataforma está hecha con *componentes* como los siguientes:

- Un editor de texto.
- Un compilador.
- Un intérprete.
- Un cliente.
- Un depurador.

Usualmente, un IDE también posee una gran interfaz gráfica de usuario además de abarcar un único lenguaje de programación. O puede abarcar varios, depende de la clase de IDE.

4.7 ¿Qué es .NET MAUI?

.NET MAUI es un framework multiplataforma para crear aplicaciones móvil nativas y de escritorio utilizando los idiomas de programación C# y XAML. Fue lanzado el 23 de mayo de 2022 con la versión de .NET 6 y hace poco ha sido actualizado con la salida .NET 7 el pasado martes 8 de noviembre.

Usando .NET MAUI puedes desarrollar aplicaciones para Android, iOS, macOS y Windows a partir del mismo código, compartiendo la mayor parte de este para las distintas plataformas.

4.7.1 Conocimientos necesarios para desarrollar aplicaciones en .NET MAUI

Si ya has trabajado previamente con Xamarin ya dispones de los conocimientos necesarios para empezar a utilizar .NET MAUI, ya que este es la evolución de Xamarin y tienen muchas similitudes.

En caso de no haber trabajado nunca con Xamarin deberás familiarizarte primero con las tecnologías .NET y XAML. En el caso de .NET, hablamos de un framework desarrollado por Microsoft basado en el lenguaje de Programación C#, es multiplataforma y permite desarrollar aplicaciones web, de escritorio e incluso móviles. XAML es un idioma para la creación de interfaces de usuarios basado en XML. Ya era usado en WPF, el framework de Microsoft para crear aplicaciones de escritorio, y también en Xamarin, por lo que por supuesto lo tenemos también en .NET MAUI.

4.7.2 Herramientas

La herramienta que principalmente necesitamos es el entorno de desarrollo (IDE) Visual Studio, disponible tanto para Windows como para Mac, que nos permitirá desarrollar nuestras aplicaciones en .NET MAUI. También necesitaremos un MAC, al que conectaremos remotamente (por ssh) desde el Visual Studio, para poder compilar y generar la versión de iOS de nuestra aplicación. Opcionalmente, pero altamente recomendado, necesitaremos un teléfono Android/iPhone para probar nuestra app, o podemos usar el emulador que viene integrado con el Visual Studio.

4.7.3 Consejos prácticos para comenzar en el desarrollo de aplicaciones en .NET MAUI

- **Patrón MVVM**

El patrón de diseño MVVM (model – view –.viewmodel) es una arquitectura de software que define como estructurar el código del proyecto de forma que la interfaz quede, en lo máximo posible, desacoplada de la lógica de la aplicación. A efectos prácticos esto se resume en separar el código del proyecto en tres apartados distintos.

5. El modelo es la capa de dominio de la aplicación, en esta capa solo se encuentra la información y no debería contener ningún tipo de lógica.
6. La capa de la vista contiene la interfaz gráfica, en esta capa se encuentran todas las vistas de la aplicación y es la que se encarga de mostrar y recibir información del usuario.
7. La capa vista-model (.viewmodel) es la que conecta la vista con el modelo y contiene toda la lógica de presentación.

Una de las principales ventajas de este patrón de diseño es que, al estar toda la lógica en el.viewmodel, se pueden crear test unitarios para probar la app sin tener que pasar por la interfaz.

- **Community toolkit**

Como ya fue el caso con Xamarin, .NET MAUI tiene una gran comunidad que ha dejado a nuestra disposición una serie de elementos que suelen usarse comúnmente en el desarrollo de aplicaciones y que están disponibles para que los usemos en nuestros proyectos.

Los podemos descargar instalando el paquete NuGet “CommunityToolkit.Maui” y podemos encontrar los elementos que contiene y su documentación en el siguiente enlace: Documentación NET MAUI Community Toolkit

- **Servicios**

Una variante del patrón MVVM es MVVMS, esta variante consiste en sacar trozos de código que se repiten en varios.viewmodel a otra capa del proyecto llamada Servicios, donde guardamos esta lógica que luego puede ser usada por distintos.viewmodel.

Usar servicios es también bastante cómodo en pantallas con una gran cantidad de funcionalidad, ya que reducen considerablemente el tamaño de los.viewmodel.

- **Usar pruebas unitarias**

Las pruebas unitarias nos ayudan a encontrar los posibles errores en nuestra aplicación, aprovechando una de las ventajas de usar el patrón MVVM, podemos probar gran parte de la lógica de nuestra aplicación de forma más sencilla.

Las pruebas unitarias deben de ir en un proyecto separado dentro de la misma solución y siempre deben probar partes de código independientes.

- **Sigue aprendiendo**

.NET, XAML, C#, ... son varias las tecnologías que se utilizan con .NET MAUI sin contar la gran cantidad de complementos que tenemos disponibles como el community toolkit, todos ellos tienen una gran cantidad de cosas interesantes que puedes utilizar en tus aplicaciones y se van añadiendo nuevas constantemente, solo tienes que descubrirlas.

4.8 Diseño de apps o aplicaciones móviles

El diseño de aplicaciones móviles consiste en desarrollar una aplicación para el uso en smartphones, con sistemas operativo Android y/o iOS. Va desde la conceptualización de la idea, al esbozo, los primeros diseños, la programación y el diseño final. En estructurar la navegación de la aplicación a partir de la lista funcionalidades de la solución y definir las directrices visuales que se aplicarán a los distintos elementos y pantallas.

Por lo tanto, garantiza por un lado que la información está bien estructurada y comprueba que la navegación y funcionalidad de la app fluye correctamente, asegurando una experiencia de usuario adecuada. Y por otro, se encarga de que el look and feel de la app vaya acorde con la guía de estilo de la marca y sea atractivo para el tipo de usuario al que se dirige (Castillo, 2022).

4.8.1 Ventajas

- Ofrecer novedades e información general al usuario en tiempo real
- Aumentar las probabilidades de fidelización hacia tu marca, producto o servicio
- El usuario accede a la app con mayor rapidez y comodidad que a la web, por lo que fortalecemos la unión con el usuario al ofrecerle la información y/o el catálogo de productos/servicios de una forma directa, fácil, intuitiva y adaptada a su dispositivo (htt138).

4.8.2 Proceso de diseño de la aplicación

Paso 1: Encontrar la idea de tu aplicación

El primer paso del proceso de diseño de una aplicación móvil es encontrar una idea de aplicación. Este paso permite a los diseñadores de aplicaciones especificar el motivo que puede añadir valor al proceso empresarial existente. Lo que se nota es que las aplicaciones móviles se han convertido en una parte crucial de la vida de los usuarios, pero hay que hacer que la aplicación móvil se distinga de la competencia añadiendo características distintas. Elegir la idea correcta de la aplicación y añadirle características útiles le ayudará a crear una aplicación de éxito. Antes de especificar el nicho de la aplicación, los diseñadores de aplicaciones pueden hacerle preguntas sobre sus clientes potenciales, sus actividades, los problemas a los que se enfrentan en sus vidas y la posible solución a ese problema a través de su aplicación. Además, los diseñadores de aplicaciones realizarán una investigación en profundidad sobre:

- Diseño UX

Después de obtener la idea correcta de la aplicación, es el momento en que un diseñador de aplicaciones busca el mejor diseño UI/UX para tener una idea de cómo la gente va a interactuar con su aplicación. Además, los diseñadores de aplicaciones llevarán a cabo un estudio de usuarios y de mercado para aclarar los requisitos de los usuarios y el aspecto y el funcionamiento de la aplicación para su público potencial. Esta investigación en profundidad ayudará a los diseñadores a obtener los comentarios de los usuarios y a añadir características al prototipo de acuerdo con los requisitos de diseño específicos de los usuarios. Para la investigación de mercado, puede utilizar los datos existentes, pero la práctica más beneficiosa es tomar las opiniones de sus clientes potenciales para un proceso de desarrollo sin problemas.

- Opiniones de los usuarios

Los comentarios de los usuarios son otro aspecto crucial que hay que tener en cuenta en el proceso de diseño de la aplicación. En esta fase, los diseñadores de aplicaciones realizarán encuestas y entrevistas para aclarar los requisitos de diseño de los usuarios. Un creador de aplicaciones debe tener en cuenta los comentarios de los usuarios antes de comenzar el diseño de una aplicación.

- Diseño de Experiencia de Usuario (UX)

Se centra en cómo los usuarios interactuarán con la aplicación, cómo navegarán por ella y cómo se sentirán al usarla. Implica la creación de flujos de usuario intuitivos y la

consideración de cómo las diferentes funciones se integrarán para brindar una experiencia coherente y satisfactoria.

- Principios de Diseño

En el diseño de aplicaciones se aplican principios de diseño visual y de interacción, como la jerarquía visual, la consistencia, la retroalimentación instantánea, la simplicidad y otros, para asegurar que la aplicación sea fácil de entender y usar (htt139).

- Compatibilidad con Dispositivos

En la actualidad, las aplicaciones deben ser diseñadas para funcionar en una variedad de dispositivos y tamaños de pantalla, como smartphones, tabletas y computadoras de escritorio. El diseño debe adaptarse a diferentes resoluciones y características técnicas de estos dispositivos (htt140).

- Consideraciones Técnicas

El diseño de la aplicación también debe tener en cuenta las limitaciones y capacidades técnicas, como la velocidad de carga, la eficiencia en el uso de recursos y la integración con sistemas externos.

- Estudio de mercado

La investigación de mercado es crucial en el proceso de desarrollo para el análisis de la competencia y el seguimiento de las últimas tendencias de diseño. A través del análisis de la competencia, puedes averiguar cuáles son tus competidores, su cuota de mercado y las características destacadas de sus aplicaciones móviles. Por ejemplo, los diseñadores de aplicaciones pueden rediseñar su aplicación móvil o web tras obtener ideas de las aplicaciones de sus competidores.

Paso 2: Valida tu idea

Ahora es el momento de validar tu idea para comprobar la viabilidad de tu aplicación. Para ello, la mejor práctica es presentar tu discurso de ascensor frente a un espejo. Si tiene sentido para ti, también lo tendrá para los demás. Puedes hacer que tu idea tenga una comunicación clara o arruinarla con una explicación confusa. Así que, antes de presentar tu idea a un creador de aplicaciones, te recomendamos que la expliques delante de tu familia y amigos y les pidas sus sugerencias. Tus allegados son las personas que realmente te orientarán sobre tu app. Es posible que encuentres algunos puntos que no habías

considerado antes. Además, puedes presentar tu idea a desconocidos para tener una visión clara de lo que estás creando.

Paso 3: Elige las características de tu aplicación

La elección de las características de tu aplicación te ayudará a decidir el diseño. Este paso es el primer paso hacia tu sueño. Ya puedes imaginar cómo quieras que funcione tu aplicación. La elección de las características de su aplicación móvil le proporcionará una visión clara de su aplicación. En esta fase, estás plasmando tu idea en papel para darle vida. Lo que se nota es que en los dos primeros pasos, tienes que centrarte en la viabilidad de tu aplicación en lugar de pensar en la monetización de la misma. Al principio, la monetización de la aplicación no es tu preocupación. En esta etapa, debes pensar en las mejores opciones para hacer la mejor experiencia de usuario a través de las características de tu aplicación. Digamos que estás interesado en crear una aplicación Android para redes sociales, y puedes empezar a escribir las siguientes cosas en el papel:

- Cuenta de usuario

Diferentes formas de acceder a la cuenta de usuario, como contraseña, Google o Facebook

Opciones para que los usuarios añadan su nombre de usuario, su foto de perfil y su biografía Opciones para añadir una foto de perfil desde la biblioteca del dispositivo o directamente desde la cámara Notificaciones push

Paso 4: Haz un prototipo de diseño para tu aplicación

Después de escribir las características de su aplicación móvil, el siguiente paso en el desarrollo de la aplicación es planificar el diseño de la aplicación móvil con los mejores elementos de la interfaz de usuario. Este paso te ayudará a especificar cómo interactuarán tus usuarios con las características de tu aplicación. En esta fase, los diseñadores utilizan un lápiz para dibujar la aplicación. Al dibujar los bocetos, tienen en cuenta los siguientes elementos de diseño de la aplicación:

- Pantalla

Al dibujar el diseño de tu aplicación, es crucial dividir la pantalla principal de tu aplicación en secciones. Centrarse en la pantalla de la aplicación es crucial, ya que es el componente principal para el funcionamiento de tu aplicación. Las direcciones de la pantalla de la aplicación establecidas por los diseñadores son generales, pero puedes

imaginar el funcionamiento de tu aplicación con la ayuda de estos bocetos de pantalla. Cuando se diseña una pantalla de aplicación, el diseñador de aplicaciones se preocupa por el número de secciones para ajustar todas las características. Digamos que tienes una aplicación de comida, y habría una pantalla para la lista de todos los artículos de comida disponibles, otra pantalla para los artículos de comida pedidos, y una pantalla para la configuración general de la aplicación.

Una vez que tengas una hoja de ruta clara para la pantalla de la aplicación, es el momento de pasar al siguiente paso. Lo que hay que tener en cuenta es que este boceto de pantalla es definitivo, y hay un 100% de posibilidades de que el diseñador de la aplicación móvil pueda hacer los cambios necesarios cuando el proceso avance.

- Barra de navegación

Después de tener una idea sobre la pantalla de la aplicación, es el momento de centrarse en la navegación principal de su aplicación para moverse dentro de la misma. El diseñador de tu app móvil puede decidir añadir una barra de pestañas en la parte inferior, o puede añadir una opción lateral deslizante para la navegación de las diferentes secciones.

En este sentido, también puedes sugerir tus ideas de navegación favoritas siguiendo la barra de navegación de tus aplicaciones móviles favoritas. Deberá elegir una opción de navegación intuitiva y fácil de usar para su aplicación para que parezca más natural y fácil de usar. Te recomendamos que evites un estilo de navegación que dificulte el desplazamiento por las pantallas de la app.

- Combinación de colores

El color es un elemento crucial del diseño de la aplicación que puede hacer o deshacer su aplicación. Por lo tanto, tenga cuidado al seleccionar una combinación de colores para su aplicación. Al seleccionar una combinación de colores, asegúrate de elegir colores que sean adecuados para las personas con problemas de daltonismo.

- Tamaño de la fuente

Es crucial tener en cuenta el tamaño de la fuente para que tu aplicación parezca más profesional. Por lo tanto, mantenga el tamaño de la fuente proporcional a las imágenes de la aplicación para evitar el desorden en el diseño de su aplicación. Una aplicación con un tamaño de fuente adecuado creará una interfaz atractiva para tus usuarios.

- Icono o logotipo de la aplicación

El ícono o logotipo de la aplicación es crucial para un diseño intuitivo. Por lo tanto, debes diseñar un ícono de aplicación autoexplicativo y memorable que pueda atraer a tus usuarios y que explique claramente el propósito de la aplicación (2022).

- Diseño de aplicaciones para una mejor experiencia de usuario

Un diseño intuitivo te ayudará a ganar más usuarios para tu móvil. Una aplicación con opciones de diseño pobres acabará reduciendo el número de usuarios de la aplicación. Después de recorrer los pasos del proceso de diseño de la app, te desvelamos los consejos de diseño de la app para que el diseño de tu app sea más intuitivo para tus usuarios.

- Mantén el diseño de tu aplicación simple

El primer consejo para el diseño es mantener el diseño de tu aplicación lo más simple posible. La razón es que la mayoría de los usuarios de aplicaciones sienten dudas al utilizar aplicaciones Android con elementos de diseño complicados. Por lo tanto, la mejor práctica para desordenar el diseño de tu aplicación es mantenerlo lo más simple posible. La magia de la simplicidad radica en que un diseño sencillo de la aplicación móvil ayudará a los usuarios a reducir la carga cognitiva de sus mentes.

- Céntrese en su público objetivo

A continuación, te recomendamos que te centres en tus usuarios objetivo a la hora de diseñar una aplicación. Digamos que quieres desarrollar una app de planificación de eventos, por lo que es una app que los usuarios utilizarán ocasionalmente. Entonces, para una aplicación móvil que los usuarios utilizan raramente, la mejor práctica es mantener el diseño lo más formal posible. La razón es que los colores brillantes con elementos visuales desordenados harán que estas aplicaciones Android parezcan poco profesionales. Pero si estás diseñando una aplicación de uso diario, te recomendamos que la mantengas informal y atractiva para los usuarios.

- Diseñar una interfaz de usuario indulgente

El tercer consejo es crucial para ganarse el corazón de los usuarios. Es un fenómeno común que la gente cometa errores y quiera revertirlos. Por lo tanto, es importante que los diseñadores diseñen una interfaz de usuario indulgente que permita a los usuarios de la aplicación saltar o volver a la pantalla anterior. Así, una interfaz de usuario flexible con

elementos de interfaz de usuario intuitivos ayudará al equipo de desarrollo a ganarse la confianza y la fidelidad de los usuarios.

- Conozca los hábitos de los usuarios

Antes de diseñar una barra de navegación para tu aplicación, tienes que centrarte en los hábitos de los usuarios respecto a cómo se desplazan por sus dispositivos móviles. La mayoría de los usuarios móviles sólo se desplazan hacia arriba y hacia abajo para acceder a las funciones de la aplicación. Por lo tanto, añadir muchas funciones hará que tu aplicación sea complicada para los usuarios.

Consejo 5: Utilice una combinación de colores distinta

Utiliza una combinación de colores distinta a la de otros teléfonos y diseña una pestaña para que el diseño de la aplicación sea sencillo.

- Añade botones de llamada a la acción

Diseña opciones para los usuarios de la aplicación cuando accedan a ella por primera vez. Para ello, puedes añadir botones de llamada a la acción como "Regístrate" o "Entra" para acceder a la app. Pero lo que se nota es que al crear una aplicación para la App Store de Apple, estas opciones harán que la aplicación sea desaprobada. La razón es que la mayoría de los usuarios no están interesados en revelar sus datos para acceder a las funciones de tu app.

- Añade animaciones al diseño de tu aplicación

Añadir animaciones al diseño de tu aplicación hará que ésta sea más atractiva para tus usuarios. Así que no te preocupes por el alcance del desarrollo de tu aplicación, y añade animaciones y características interesantes para que tu aplicación sea más atractiva para tus usuarios (2022).

4.9 Diseño y Esquematización del Aplicativo Móvil a desarrollar

Diseño de la aplicación

Para el diseño de la aplicación móvil se lleva a cabo un proceso fundamental que va más allá de crear una simple interfaz visual atractiva. Implica una cuidadosa planificación y ejecución para garantizar que la aplicación no solo sea visualmente agradable, sino también funcional, intuitiva y capaz de brindar una experiencia excepcional a los usuarios.

En primer lugar, se debe tomar en cuenta la usabilidad. El diseño de nuestra aplicación debe priorizar la usabilidad centrada en la educación. Los usuarios deben tener una plataforma que les permita acceder fácilmente a cursos, recursos y contenido relacionado con la pesca. El diseño de la navegación debe reflejar la estructura de cursos, categorías y contenido, lo que facilita que los usuarios encuentren lo que necesitan sin dificultad.

Los usuarios esperan que las aplicaciones sean fáciles de entender y utilizar desde el primer momento. El diseño de la aplicación debe crear una experiencia inmersiva de aprendizaje queatraiga a los usuarios y los mantenga comprometidos. Esto puede lograrse mediante el uso de elementos visuales relacionados con la pesca, como imágenes de escenarios de pesca, equipos y especies acuáticas. Animaciones sutiles y transiciones suaves pueden mejorar la experiencia del usuario y hacer que el proceso de aprendizaje sea más atractivo. Un diseño coherente y atractivo puede generar confianza, aumentar el compromiso del usuario y fortalecer el reconocimiento de la marca. La accesibilidad puede incorporarse mediante una elección cuidadosa de colores y contrastes, tamaños de fuente legibles y la implementación de características de lectura en voz alta y navegación por voz. Un diseño optimizado tiene en cuenta las limitaciones de hardware y ancho de banda de los dispositivos móviles para garantizar una experiencia de usuario fluida y rápida. El diseño debe considerar la minimización de imágenes pesadas y recursos que puedan ralentizar el tiempo de carga. Además, las animaciones y las interacciones deben ser suaves y responsivas para garantizar una experiencia sin interrupciones.

Desarrollo de la aplicación

El desarrollo de una aplicación móvil es una actividad esencial en la era digital actual, ya que las aplicaciones móviles se han convertido en una parte integral de la vida cotidiana de las personas. El desarrollo de aplicaciones móviles permite abordar las necesidades y demandas de los usuarios. Las aplicaciones móviles brindan una plataforma para innovar y ofrecer soluciones prácticas y convenientes, además de resolver problemas cotidianos, facilitar tareas y mejorar la eficiencia, todo mientras se adaptan a las preferencias cambiantes de los usuarios, fomentando el desarrollo de aplicaciones móviles fomenta la interconexión y la comunicación global. Las aplicaciones permiten a las personas conectarse entre sí sin importar su ubicación física.

La aplicación móvil ofrece contenido educativo adaptado a las necesidades de los entusiastas de la pesca, desde principiantes hasta pescadores experimentados. Ya sea que

estén interesados en técnicas de pesca, conservación de especies o equipo especializado, la aplicación puede proporcionar información detallada y cursos diseñados para mejorar sus habilidades y conocimientos.



Ilustración 1 Diseño Página principal del App



Ilustración 2 Diseño Ventanas de navegación del App



Ilustración 3 Diseño Ventana de opciones

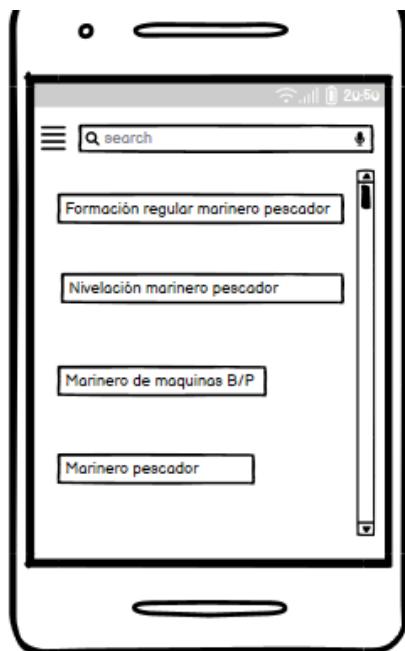


Ilustración 4 Diseño Ventana de cursos

5. RESULTADOS

El proyecto se propuso abordar la problemática de la falta de acceso y difusión efectiva de programas de formación en la industria pesquera mediante una solución tecnológica. Mejorando la accesibilidad mediante una plataforma interactiva y amigable que permite a los usuarios explorar una amplia gama de programas de formación. La interfaz intuitiva garantiza que cualquier persona, sin importar su nivel de experiencia, pueda acceder a información detallada sobre cada curso de manera clara y concisa sobre cursos certificados en pesca. Esto beneficia tanto a futuros pescadores como a aquellos ya involucrados en la pesca. Dado que los dispositivos móviles son vulnerables a ataques, es fundamental implementar medidas de seguridad robustas en la aplicación móvil. Los controles de seguridad de las aplicaciones, como el manejo adecuado de entradas inesperadas, son esenciales para reducir las vulnerabilidades. Esto garantiza que la aplicación móvil sea resistente a intentos de explotación por parte de ciberdelincuentes que puedan buscar debilidades en el código.

La seguridad de las aplicaciones móviles también se relaciona con la protección de los datos de los usuarios. La aplicación contiene información personal de los estudiantes, como registros académicos y detalles de contacto, al implementar medidas de seguridad, como la encriptación de datos y la autenticación segura, es esencial para proteger esta información sensible. Se ha seguido un enfoque de planificación detallada y seguimiento constante del proyecto para asegurarse de que se cumplan los objetivos y se minimicen los riesgos asociados. Se ha destacado la importancia de la seguridad de la información en el contexto del desarrollo de la aplicación móvil protegiendo los datos sensibles de los usuarios que es prioridad y se han implementado cortafuegos para aplicaciones móviles y pruebas de seguridad (fuzzing).

El Fuzzing garantiza la robustez de la aplicación móvil y es importante realizarlas las pruebas que implica examinar cómo la aplicación responde a entradas inesperadas que podrían explotarse. Al identificar y abordar estos puntos débiles, se fortalece la seguridad en toda la infraestructura educativa. La implementación de estas medidas de seguridad implica una atención constante a la seguridad de las aplicaciones móvil para proteger los datos de los usuarios y garantizar un entorno de formación seguro y confiable. La seguridad es esencial para mantener la integridad de la información académica y personal, así como para prevenir posibles amenazas ciberneticas.

```

13     <Label
14         Text="CURSOS PARA APRENDER A PESCAR"
15         SemanticProperties.HeadingLevel="Level1"
16         FontSize="32"
17         HorizontalOptions="Center" />
18
19     <Label
20         Text="Podras visualizar los diferentes cursos que tenemos para ti."
21         SemanticProperties.HeadingLevel="Level2"
22         SemanticProperties.Description="Welcome to dot net Multi platform App U I"
23         FontSize="18"
24         HorizontalOptions="Center" />
25     <Button Text="Siguiente"
26         VerticalOptions="Center"
27         HorizontalOptions="Center"
28         Clicked="Button_Clicked" />
29     <Label x:Name="pagina2"
30         Text=""
31         FontSize="18"
32         VerticalOptions="Center" />
33

```

100 % No se encontraron problemas. Línea: 34 Carácter: 45 SPC CRLF

Ilustración 5 Desarrollo ventana principal de cursos

```

1    <?xml version="1.0" encoding="UTF-8" ?>
2    <Shell
3        x:Class="MauiApp1.AppShell"
4        xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
5        xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
6        xmlns:local="clr-namespace:MauiApp1"
7        Shell.FlyoutBehavior="Flyout">
8
9    <FlyoutItem FlyoutDisplayOptions="AsMultipleItems" Route="ParentRoute">
10       <ShellContent
11           Title="INICIO"
12           ContentTemplate="{DataTemplate local: MainPage}"
13           Route="MainPage" />
14
15       <ShellContent
16           Title="DATOS"
17           ContentTemplate="{DataTemplate local: MainPage}"
18           Route="MainPage" />
19
20   </FlyoutItem>
21

```

100 % No se encontraron problemas. Línea: 25 Carácter: 28 SPC CRLF

Ilustración 6 Desarrollo de barra de navegación

```

1    <?xml version="1.0" encoding="utf-8" ?>
2    <ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
3        xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
4        x:Class="MauiApp1.NewPage6"
5        Title="">
6        <VerticalStackLayout>
7            <Image
8                Source="marineromaquina.JPG"
9                SemanticProperties.Description="Cute dot net bot waving hi to you!"
10               HeightRequest="400"
11               HorizontalOptions="Center" />
12            <Label
13                Text="Marinero de maquina B/P"
14                SemanticProperties.HeadingLevel="Level1"
15                FontSize="20"
16                HorizontalOptions="Center" />
17            <Label
18                Text="Tripulante del buque que forme parte de la guardia en una sala de máquinas"
19                VerticalOptions="Center"
20                SemanticProperties.HeadingLevel="Level2"
21                SemanticProperties.Description="Welcome to dot net Multi platform App U I"
22

```

100 % No se encontraron problemas. Línea: 10 Carácter: 35 SPC CRLF

Ilustración 7 Desarrollo ventana de curso Marinero de maquina

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="MauiApp1.MainPage">

    <ScrollView>
        <VerticalStackLayout
            Spacing="25"
            Padding="30, 0"
            VerticalOptions="Center">

            <Image
                Source="brinquillo.gif"
                SemanticProperties.Description="Cute dot net bot waving hi to you!"
                HeightRequest="200"
                HorizontalOptions="Center" />

            <Label
                Text="Bienvenidos!"
                SemanticProperties.HeadingLevel="Level1"
                FontSize="32" />
        </VerticalStackLayout>
    </ScrollView>
</ContentPage>
```

Ilustración 8 Desarrollo ventana principal

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="MauiApp1.NewPage1"
    Title="">

    <VerticalStackLayout>
        <Image
            Source="logo.png"
            SemanticProperties.Description="Cute dot net bot waving hi to you!"
            HeightRequest="300"
            HorizontalOptions="Center" />

        <Label
            Text="Escuela de Pesca, para transformar tu estilo de trabajo "
            SemanticProperties.HeadingLevel="Level1"
            FontSize="32"
            HorizontalOptions="Center" />

        <Label
            Text="Un paso al éxito del mundo pesquero"
            SemanticProperties.HeadingLevel="Level2"
            SemanticProperties.Description="Welcome to dot net Multi platform App U I"
            FontSize="18"
            HorizontalOptions="Center" />
        <Button Text="Siguiente"
            VerticalOptions="Center"
            HorizontalOptions="Center"
            Clicked="Button_Clicked" />
        <Label x:Name="pagina3"
            Text=""
            FontSize="18"
            VerticalOptions="Center"
            HorizontalOptions="Center" />
    </VerticalStackLayout>
</ContentPage>
```

Ilustración 9 Desarrollo ventana de información de la escuela de pesca



Ilustración 10 Pantalla principal



Ilustración 11 Pantalla de introducción a los cursos



Ilustración 12 Pantalla de información de escuela de pesca



INICIO DATOS CURSOS ...



Aquí tendrás accesos a nuestros cursos

Somos una institución académica de primer orden para servir al Sector Pesquero y Acuícola nacional

[Siguiente](#)

Ilustración 13 Pantalla para acceder a los cursos

INICIO DATOS CURSOS ...

Ilustración 14 Barra de navegación

— □ ×

≡

INICIO DATOS CURSOS ...



Marinero de maquina B/P

Tripulante del buque que forme parte de la guardia en una sala de máquinas

[Abrir](#)



Marinero pescador

Ejercer de tripulante subalterno para cualquier servicio en los buques dedicados a la pesca o auxiliar de acuicultura

[Abrir](#)



Nivelacion Marinero pescador

Ilustración 15 Pantalla de cursos

6. CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusión

La implementación de un aplicativo móvil informativo de cursos certificados mediante la herramienta Visual Studio (MAUI) para la Escuela de Pesca-Manta aborda una problemática crítica en el ámbito de la formación en la industria pesquera. La falta de acceso y difusión efectiva de programas educativos había sido un obstáculo para el desarrollo sostenible de esta actividad. Este proyecto se presenta como una solución estratégica para superar estas barreras. Con el aplicativo móvil educativo busca democratizar el acceso a la información sobre cursos certificados, proporcionando una plataforma interactiva y accesible. Su interfaz intuitiva garantiza que cualquier persona, independientemente de su nivel de experiencia o formación previa, pueda acceder a información detallada sobre los cursos disponibles. Esto no solo beneficia a quienes buscan capacitarse en la pesca, sino que también fortalece la comunidad pesquera de Manta al proporcionar oportunidades de crecimiento profesional.

A lo largo de este proyecto, se han identificado y abordado diversas fases del desarrollo de aplicaciones móviles, desde la planificación hasta la implementación. La planificación precisa ha sido fundamental para distribuir eficientemente los recursos y minimizar los riesgos asociados al proyecto.

En conclusión, la implementación de este aplicativo móvil informativo no solo responde a una necesidad crítica en la industria pesquera, sino que también demuestra cómo la tecnología puede ser una aliada poderosa en la formación y capacitación de profesionales. Este proyecto tiene el potencial de impulsar el desarrollo sostenible de la pesca en la región, asegurando al mismo tiempo la accesibilidad y seguridad de la información para los usuarios.

6.2 Recomendaciones

- Mantener un proceso constante de evaluación de las necesidades de formación de la comunidad pesquera.
- Implementar medidas sólidas de seguridad de datos para proteger la información personal y de formación.
- Mantener la aplicación actualizada con las últimas prácticas de seguridad cibernética.

- Asegurar que la aplicación siga siendo intuitiva y fácil de usar, mejorando la interfaz de usuario y la experiencia de usuario.

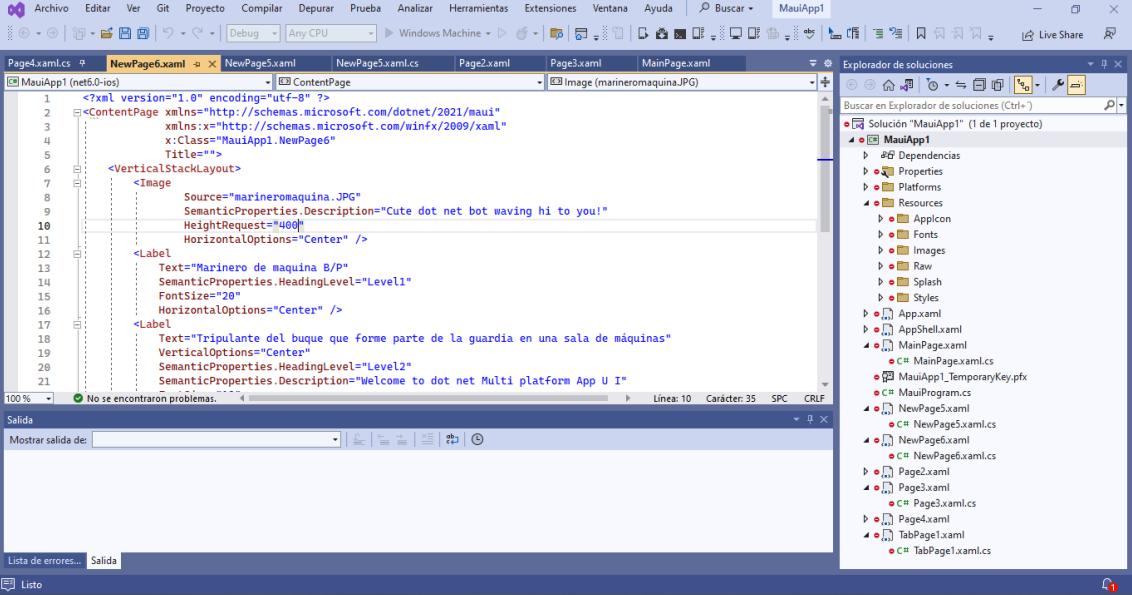
7. BIBLIOGRAFÍA

- (s.f.). <https://ccfprosario.com.ar/que-es-el-diseno-de-aplicaciones-moviles-2/#:~:text=El%20dise%C3%B1o%20de%20una%20app,programaci%C3%B3n%20y%20el%20dise%C3%B1o%20final>.
- (s.f.). <https://www.applinet.es/blog/principios-diseno-aplicacion-movil-app-2/#:~:text=Principios%20del%20dise%C3%B1o%20de%20una%20APP%20m%C3%B3vil&text=Lo%20m%C3%A1s%20importante%20a%20la,necesita%20este%20perfil%20de%20usuario>.
- (s.f.). <https://developer.android.com/guide/practices/compatibility?hl=es-419>.
- (26 de Octubre de 2022). <https://appmaster.io/es/blog/como-disenar-una-aplicacion>.
- (26 de Octubre de 2022). <https://appmaster.io/es/blog/como-disenar-una-aplicacion>.
- Arimetrics. (2022). Obtenido de <https://www.arimetrics.com/glosario-digital/visual-studio>
- Castillo, A. d. (10 de 03 de 2022). <https://mobivery.com/diseno-de-apps/#:~:text=El%20dise%C3%B1o%20de%20aplicaciones%20m%C3%B3viles,los%20distintos%20elementos%20y%20pantallas>.
- Ceballos, F. (2004). *Unidad de Apoyo Para el Aprendizaje*. Obtenido de https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/1023/mod_resource/content/1/contenido/index.html
- Cilsa. (2017). Obtenido de <https://desarrollarinclusion.cilsa.org/tecnologia-inclusiva/que-es-un-lenguaje-de-programacion/>
- GARCÍA, J. (20 de Octubre de 2021). *Tribalyte*. Obtenido de <https://tech.tribalyte.eu/blog-c-sharp-que-es-para-que-sirve>
- Garita-Araya, R. A. (2013). *Ciencias de la informacion* . Obtenido de <https://dialnet.unirioja.es/servlet/articulo?codigo=5511036>
- Isabe, L. E. (2013). *USABILIDAD EN APLICACIONES MÓVILES*. Obtenido de <https://publicaciones.unpa.edu.ar/index.php/ICTUNPA/article/view/581/560>

Puetate, M. G. (2020). *APLICACIONES*. Obtenido de
<https://www.pucesi.edu.ec/webs2/wp-content/uploads/2021/02/Aplicaciones-Móviles-Híbridas-2020.pdf>

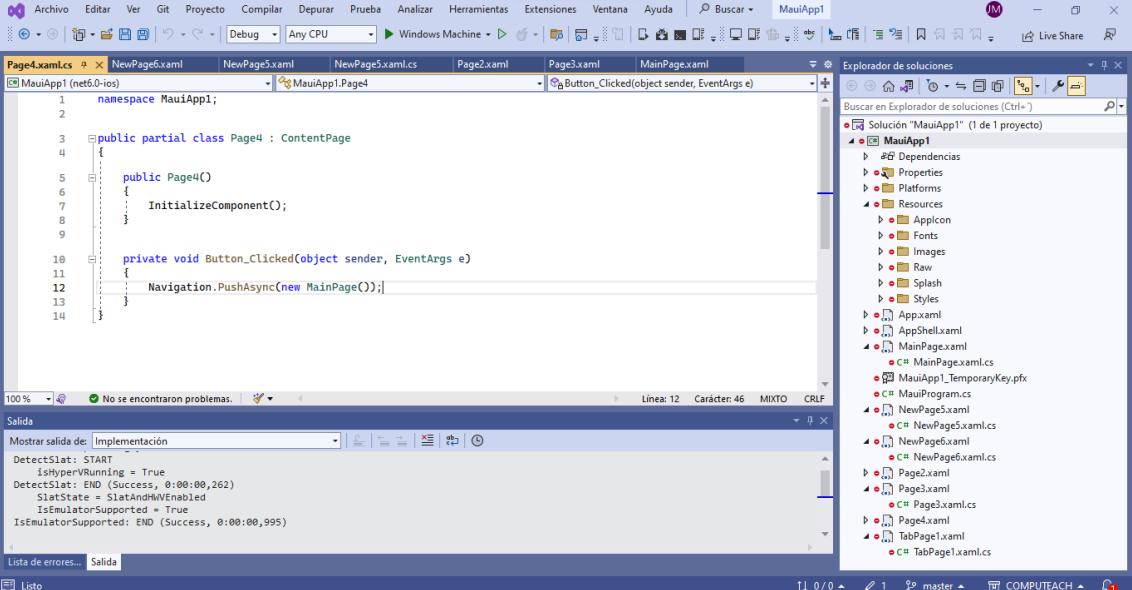
Vique, R. R. (6 de Enero de 2018). *Métodos para el desarrollo de aplicaciones móviles*. Obtenido de
https://d1wqtxts1xzle7.cloudfront.net/55462359/Tecnologia_y_desarrollo_en_dispositivos_moviles-libre.pdf?1515267422=&response-content-disposition=inline%3B+filename%3DMetodos_para_el_desarrollo_de_aplicacion.pdf&Expires=1691547158&Signature=bW~xVH7GagQi2K

8. ANEXOS



The screenshot shows the Visual Studio IDE interface with the following details:

- Menu Bar:** Archivo, Editar, Ver, Git, Proyecto, Compilar, Depurar, Prueba, Analizar, Herramientas, Extensiones, Ventana, Ayuda.
- Solution Explorer:** Shows the solution "MauiApp1" with files like MainPage.xaml, MainPage.xaml.cs, App.xaml, AppShell.xaml, and various Page files.
- Code Editor:** Displays the XAML code for Page4.xaml.cs, which includes an image and a label.
- Output Window:** Shows the message "No se encontraron problemas." (No errors found).
- Status Bar:** Shows the current file is Page4.xaml.cs, and the status "Linea: 10 Carácter: 35 SPC CRLF".



The screenshot shows the Visual Studio IDE interface with the following details:

- Menu Bar:** Archivo, Editar, Ver, Git, Proyecto, Compilar, Depurar, Prueba, Analizar, Herramientas, Extensiones, Ventana, Ayuda.
- Solution Explorer:** Shows the solution "MauiApp1" with files like MainPage.xaml, MainPage.xaml.cs, App.xaml, AppShell.xaml, and various Page files.
- Code Editor:** Displays the C# code for Page4.xaml.cs, which contains a constructor and a Button_Clicked event handler.
- Output Window:** Shows the message "No se encontraron problemas." (No errors found).
- Status Bar:** Shows the current file is Page4.xaml.cs, and the status "Linea: 12 Carácter: 46 MIXTO CRLF".

The screenshot shows the Visual Studio IDE interface. The top menu bar includes Archivo, Editar, Ver, Git, Proyecto, Compilar, Depurar, Prueba, Analizar, Herramientas, Extensiones, Ventana, Ayuda, Buscar, and MauiApp1. The toolbar has icons for file operations like Open, Save, and Print. The main window displays the XAML code for MainPage.xaml:

```
<ScrollView>
    <ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
        xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
        x:Class="MauiApp1.MainPage">

        <ScrollView>
            <VerticalStackLayout
                Spacing="25"
                Padding="30, 0"
                VerticalOptions="Center">

                <Image
                    Source="brinquillo.gif"
                    SemanticProperties.Description="Cute dot net bot waving hi to you!"
                    HeightRequest="200"
                    HorizontalOptions="Center" />

                <Label
                    Text="Bienvenidos!"
                    SemanticProperties.HeadingLevel="Level1"
                    FontSize="32" />
            </VerticalStackLayout>
        </ContentPage>
    </ScrollView>

```

The Solution Explorer on the right shows the project structure for "MauiApp1":

- Solución "MauiApp1" (1 de 1 proyecto)
 - MauiApp1
 - Properties
 - Platforms
 - Resources
 - AppIcon
 - Fonts
 - Images
 - Raw
 - Splash
 - Styles
 - App.xaml
 - AppShell.xaml
 - MainPage.xaml
 - MainPage.xaml.cs
 - MauiApp1_TemporaryKey.pfx
 - NewPage5.xaml
 - NewPage6.xaml
 - NewPage6.xaml.cs
 - NewPage6.xaml.csfx
 - Page2.xaml
 - Page3.xaml
 - Page4.xaml
 - Page5.xaml
 - TabPage1.xaml
 - TabPage1.xaml.cs

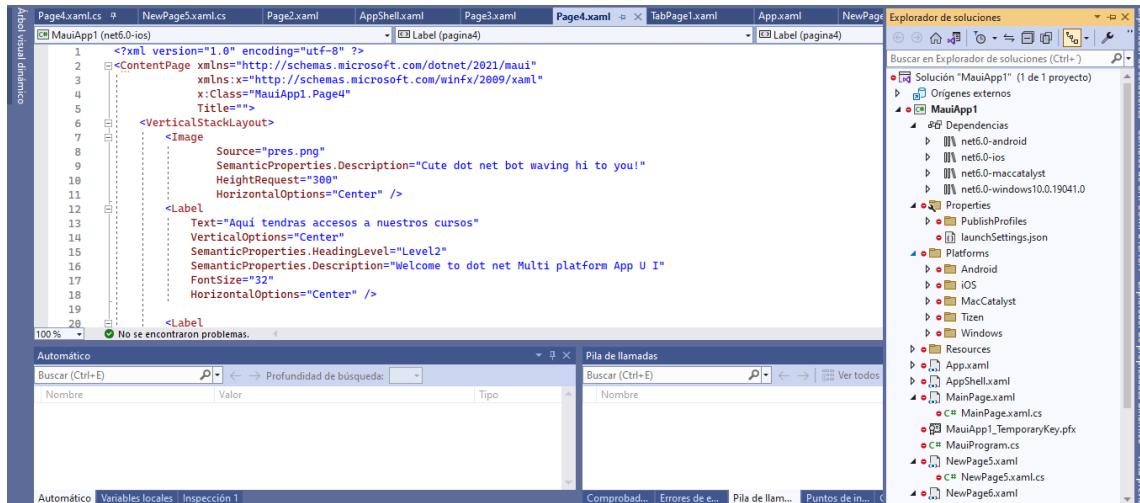
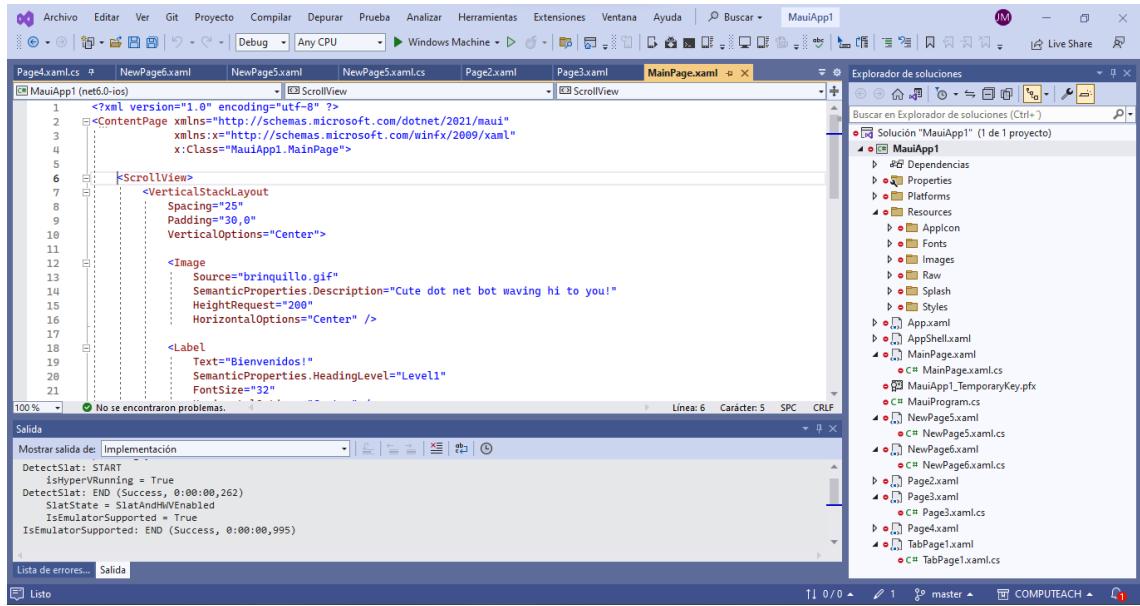
The screenshot shows the Visual Studio IDE interface. The top menu bar includes Archivo, Editar, Ver, Git, Proyecto, Compilar, Depurar, Prueba, Analizar, Herramientas, Extensiones, Ventana, Ayuda, Buscar, and MauiApp1. The toolbar has icons for file operations like Open, Save, and Print. The main window displays the XAML code for MainPage.xaml:

```
<Label>
    <Text>"Podras visualizar los diferentes cursos que tenemos para ti."</Text>
    <SemanticProperties.HeadingLevel>Level2</SemanticProperties.HeadingLevel>
    <SemanticProperties.Description>Welcome to dot net Multi platform App U I</SemanticProperties.Description>
    <FontSize>18</FontSize>
</Label>
<Button>
    <Text>"Siguiente" </Text>
    <VerticalOptions>Center</VerticalOptions>
    <HorizontalOptions>Center</HorizontalOptions>
    <Clicked>button_Clicked</Clicked>
</Button>
<Label x:Name="pagina2">
    <Text>=</Text>
    <FontSize>18</FontSize>
    <VerticalOptions>Center</VerticalOptions>
    <HorizontalOptions>Center</HorizontalOptions>
</Label>
</VerticalStackLayout>

```

The Solution Explorer on the right shows the project structure for "MauiApp1":

- Solución "MauiApp1" (1 de 1 proyecto)
 - MauiApp1
 - Properties
 - Platforms
 - Resources
 - AppIcon
 - Fonts
 - Images
 - Raw
 - Splash
 - Styles
 - App.xaml
 - AppShell.xaml
 - MainPage.xaml
 - MainPage.xaml.cs
 - MauiApp1_TemporaryKey.pfx
 - NewPage5.xaml
 - NewPage6.xaml
 - NewPage6.xaml.cs
 - NewPage6.xaml.csfx
 - Page2.xaml
 - Page3.xaml
 - Page4.xaml
 - TabPage1.xaml
 - TabPage1.xaml.cs



```
namespace MauiApp1;
public partial class NewPage5 : ContentPage
{
    public NewPage5()
    {
        InitializeComponent();
    }

    private void Button_Clicked(object sender, EventArgs e)
    {
        Navigation.PushAsync(new NewPage6());
        Navigation.PushAsync(new NewPage6());
    }
}
```

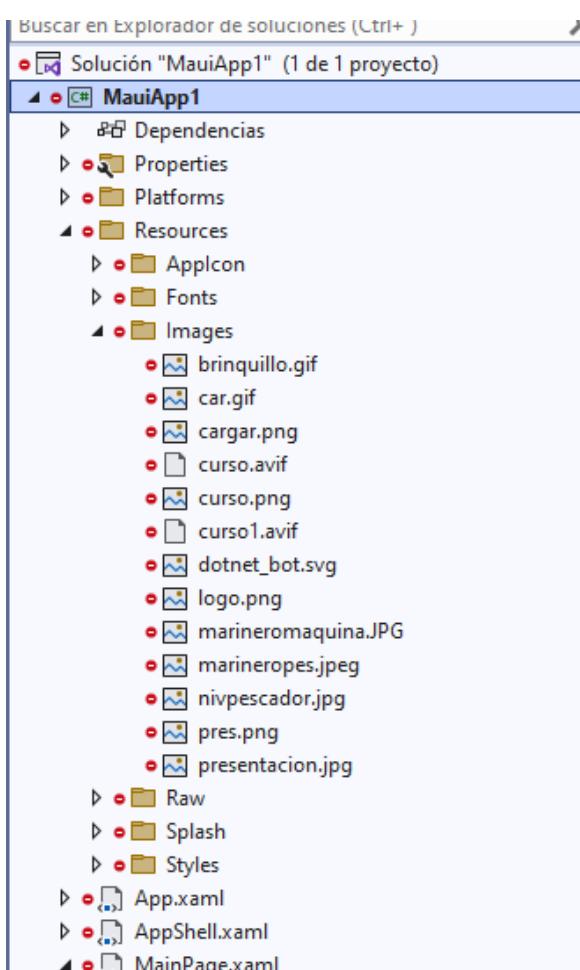
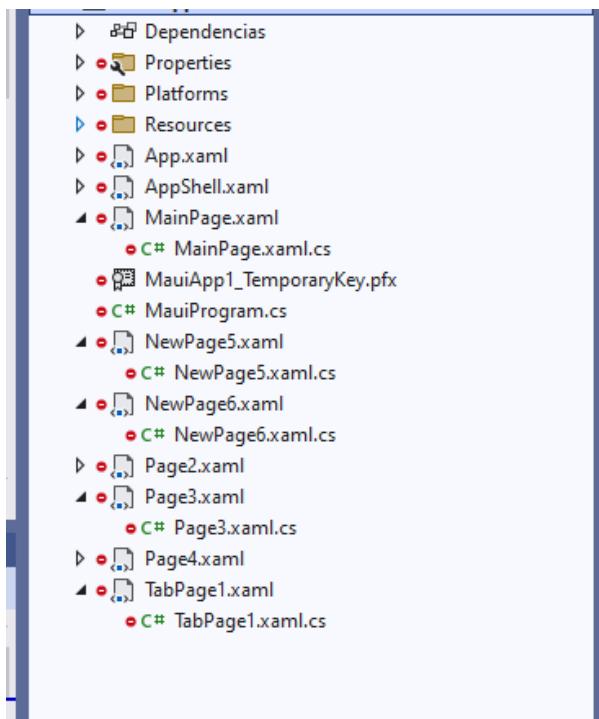
Explorador de soluciones

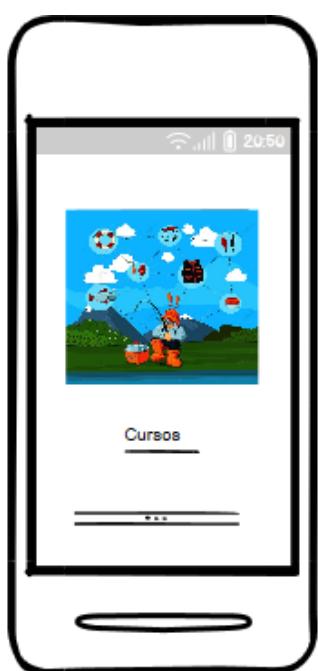
- Solución "MauiApp1" (1 de 1 proyecto)
 - Orígenes externos
 - MauiApp1
 - Dependencias
 - net6.0-android
 - net6.0-ios
 - net6.0-mac-catalyst
 - net6.0-windows10.0.19041.0
 - Properties
 - PublishProfiles
 - launchSettings.json
 - Platforms
 - Android
 - iOS
 - MacCatalyst
 - Tizen
 - Windows
 - Resources
 - App.xaml
 - AppShell.xaml
 - MainPage.xaml
 - MainPage.xaml.cs
 - MauiApp1.TemporaryKey.pfx
 - MauiProgram.cs
 - NewPage5.xaml
 - NewPage5.xaml.cs
 - NewPage6.xaml

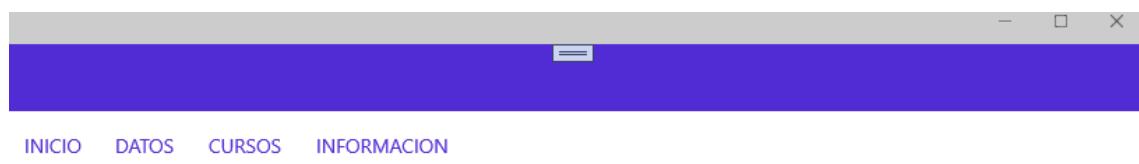
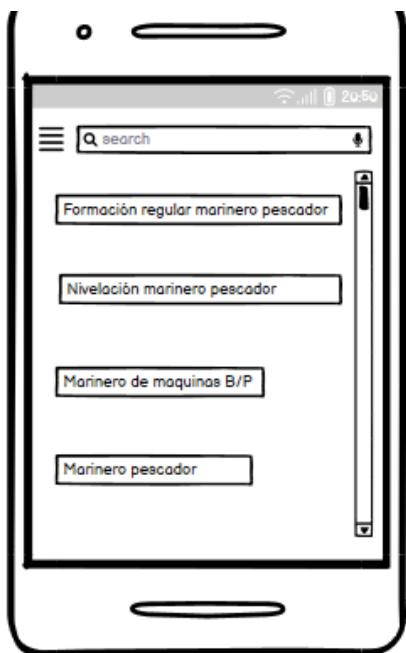
```
<xaml version="1.0" encoding="utf-8" ?>
<TabbedPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:local="clr-namespace:MauiApp1"
    x:Class="MauiApp1.TabPage1"
    Title="TabPage1">
    <local: MainPage/>
    <local: NewPage1 />
    <local: Page2 />
</TabbedPage>
```

Explorador de soluciones

- Solución "MauiApp1" (1 de 1 proyecto)
 - MauiApp1
 - Dependencias
 - Properties
 - Platforms
 - Resources
 - App.xaml
 - AppShell.xaml
 - MainPage.xaml
 - MainPage.xaml.cs
 - MauiApp1.TemporaryKey.pfx
 - MauiProgram.cs
 - NewPage5.xaml
 - NewPage5.xaml.cs
 - NewPage6.xaml
 - NewPage6.xaml.cs
 - TabPage1.xaml
 - Page2.xaml
 - Page3.xaml
 - Page4.xaml
 - TabPage1.xaml.cs





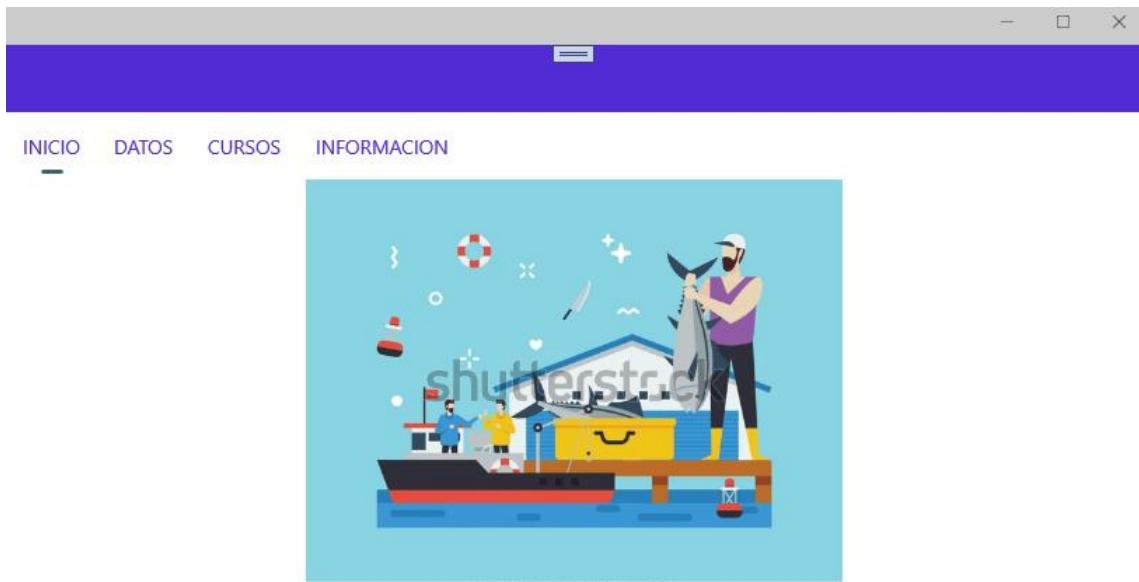


Bienvenidos!

Una escuela de pesca que te brindará nuevas oportunidades.



[Siguiente](#)



CURSOS PARA APRENDER A PESCAR

Podras visualizar los diferentes cursos que tenemos para ti.

[Siguiente](#)





INICIO DATOS CURSOS INFORMACION



Aquí tendrás accesos a nuestros cursos

Somos una institución académica de primer orden para servir al Sector Pesquero y Acuícola nacional

[Siguiente](#)



INICIO DATOS CURSOS INFORMACION



Marinero de maquina B/P

Tripulante del buque que forme parte de la guardia en una sala de máquinas

[Abrir](#)



Marinero pescador

Ejercer de tripulante subalterno para cualquier servicio en los buques dedicados a la pesca o auxiliar de acuicultura

[Abrir](#)