



SECCIÓN TEC. DEL DEPARTAMENTO DE
ELECTRONICA Y COMPUTACIÓN
CARRERA DE REDES Y TELECOMUNICACIONES

NRC:

9831

ASIGNATURA:

Aplicaciones Móviles

PRUEBA SEGUNGO PARCIAL

TEMA:

Desarrollar una aplicación en Xamarin que tenga una evaluación de 5 preguntas de la teoría de Xamarin de selección multiple.

INTEGRANTES:

Moran Santo Alexander Adair

Chicaiza Sarabia Bryan Israel

DOCENTE:

Ing. José Caiza

FECHA:

11 de julio de 2023



INFORME DE LABORATORIO

1. TEMA:

Desarrollar una aplicación en Xamarin que tenga una evaluación de 5 preguntas de la teoría de Xamarin de selección múltiple.

2. OBJETIVOS:

2.1 Objetivo General

Desarrollar una aplicación móvil multiplataforma en C# con Visual Studio que permita a los usuarios responder preguntas y evaluar sus respuestas para mejorar su conocimiento en un determinado tema o área de interés

2.2 Objetivos Específicos

- Diseñar una interfaz de usuario intuitiva y fácil de usar para mejorar la experiencia del usuario al responder las preguntas.
- Implementar la funcionalidad de evaluación automática de las respuestas de los usuarios al responder cada pregunta utilizando C# y Visual Studio.
- Garantizar la compatibilidad multiplataforma de la aplicación en dispositivos iOS y Android para llegar a la mayor cantidad de usuarios posible.



3. MARCO TEÓRICO

¿Qué es Xamarin?

Xamarin es una plataforma de código abierto para crear aplicaciones modernas y de alto rendimiento para iOS, Android y Windows con .NET. Xamarin es una capa de abstracción que administra la comunicación del código compartido con el código de la plataforma subyacente. Xamarin se ejecuta en un entorno administrado que brinda comodidades como la asignación de memoria y la recolección de elementos no utilizados.

Xamarin permite a los desarrolladores compartir un promedio del 90 % de su aplicación entre plataformas. Este patrón permite a los desarrolladores escribir toda su lógica comercial en un solo idioma (o reutilizar el código de la aplicación existente) pero lograr un rendimiento, una apariencia y una sensación nativos en cada plataforma.

Las aplicaciones de Xamarin se pueden escribir en PC o Mac y compilar en paquetes de aplicaciones nativas, como un archivo .apk en Android o un archivo .ipa en iOS.

Para quién es Xamarin

Xamarin es para desarrolladores con los siguientes objetivos:

- Comparta código, pruebas y lógica empresarial entre plataformas.

- Escriba aplicaciones multiplataforma en C# con Visual Studio.

Cómo funciona Xamarin

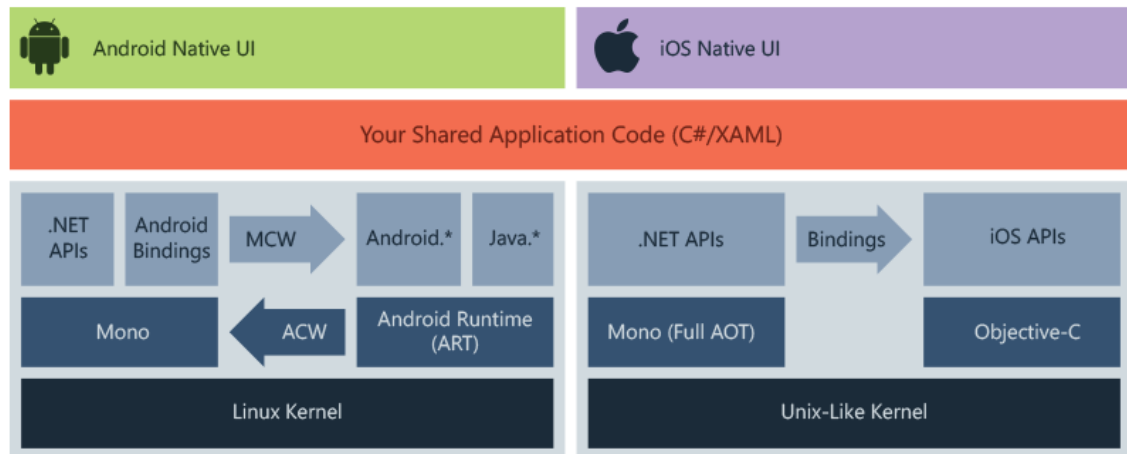


Imagen 1: Diagrama de la arquitectura Xamarin

El diagrama muestra la arquitectura general de una aplicación Xamarin multiplataforma. Xamarin le permite crear una interfaz de usuario nativa en cada plataforma y escribir lógica empresarial en C# que se comparte entre plataformas. En la mayoría de los casos, el 80 % del código de la aplicación se puede compartir con Xamarin.

Xamarin se basa en .NET, que maneja automáticamente tareas como la asignación de memoria, la recolección de elementos no utilizados y la interoperabilidad con las plataformas subyacentes.

Xamarin.Android

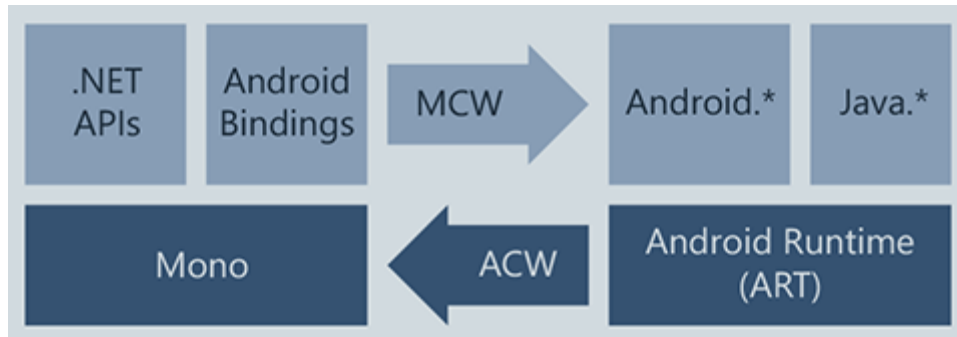


Imagen 2: Xamarin Android

Las aplicaciones de Xamarin.Android se compilan desde C# en **lenguaje intermedio (IL)** , que luego se compila **Just-in-Time (JIT)** en un ensamblado nativo cuando se inicia la aplicación. Las aplicaciones de Xamarin.Android se ejecutan en el entorno de ejecución Mono, junto con la máquina virtual Android Runtime (ART). Xamarin proporciona enlaces .NET a los espacios de nombres Android.* y Java.*. El entorno de ejecución Mono llama a estos espacios de nombres a través de **Managed Callable Wrappers (MCW)** y proporciona **Android Callable Wrappers (ACW)** al ART, lo que permite que ambos entornos invoquen código entre sí.

Xamarin.iOS

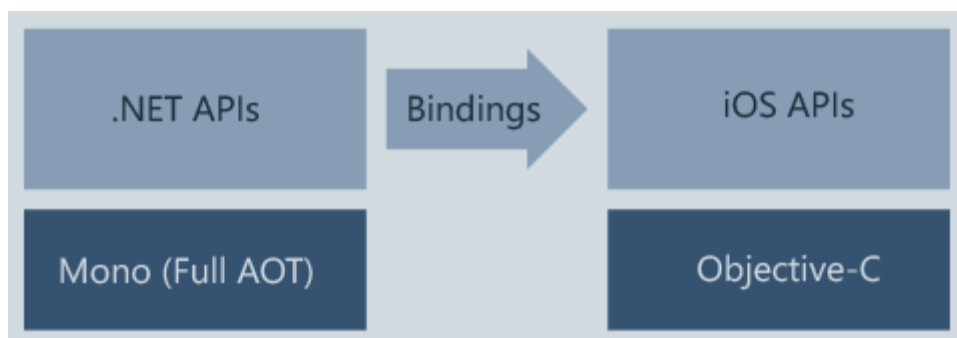


Imagen 3: Xamarin IOS



Las aplicaciones de Xamarin.iOS se compilan completamente **Ahead-of-Time (AOT)** desde **C#** en **código ensamblador ARM nativo**. Xamarin usa **selectores** para exponer Objective-C a C# administrado y registradores para exponer código de C# administrado a Objective-C. Los selectores y los registradores se denominan colectivamente "enlaces" y permiten que Objective-C y C# se comuniquen.

Xamarin.Forms

Xamarin.Forms es un marco de interfaz de usuario de código abierto. Xamarin.Forms permite a los desarrolladores crear aplicaciones de Xamarin.iOS, Xamarin.Android y Windows a partir de una única base de código compartida. Xamarin.Forms permite a los desarrolladores crear interfaces de usuario en XAML con código subyacente en C#. Estas interfaces de usuario se representan como controles nativos de alto rendimiento en cada plataforma. Algunos ejemplos de características proporcionadas por Xamarin.Forms incluyen:

- Lenguaje de interfaz de usuario XAML
- El enlace de datos
- Gestos
- Efectos
- Estilismo



¿Qué son las aplicaciones híbridas?

Las aplicaciones híbridas, a diferencia de las nativas, son aquellas capaces de funcionar en distintos sistemas operativos móviles. Entre ellos: Android, iOS y Windows Phone. De esta manera, una misma app puede utilizarse en cualquier smartphone o tablet, indistintamente de su marca o fabricante.

Para ello, estas aplicaciones tienen componentes que permiten la adaptabilidad de un mismo código a los requerimientos de cada sistema.

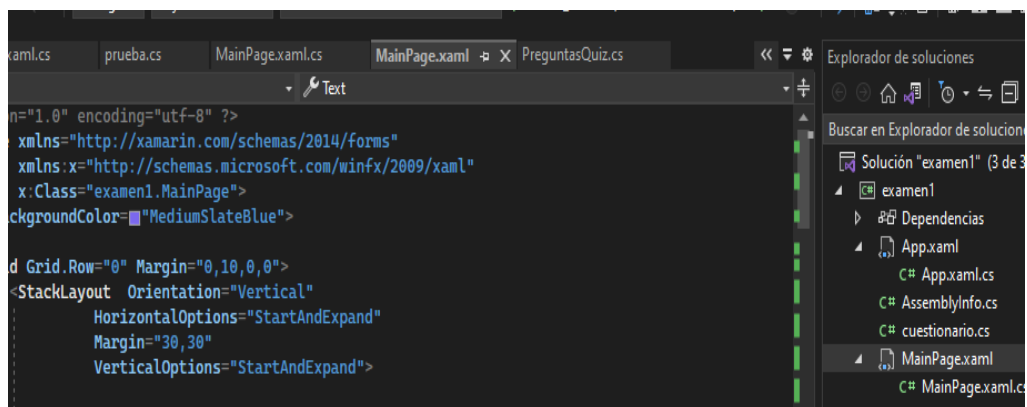
¿Qué son las aplicaciones nativas?

Las aplicaciones nativas son aquellas que solo funcionan en un único sistema operativo, así se diferencian de las aplicaciones híbridas. Por lo tanto, estas apps son dependientes al 100% de la plataforma de la cual es nativa.

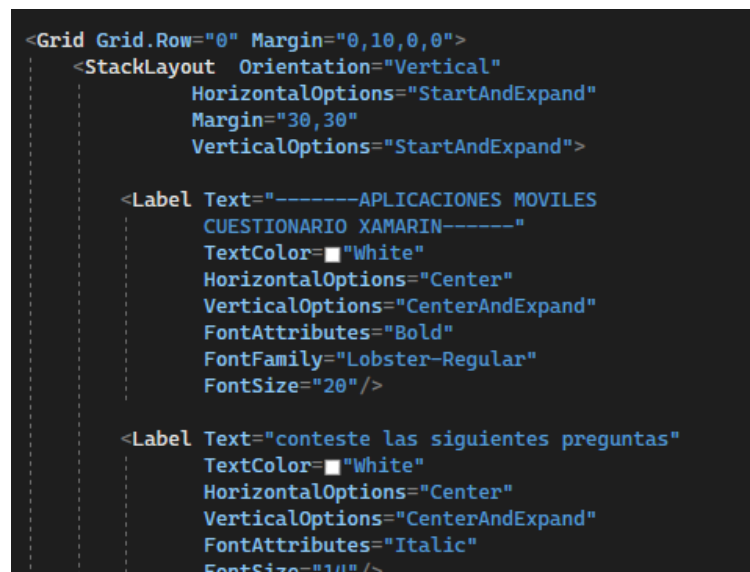
En este sentido, si se desea utilizar una app nativa en sistemas distintos, entonces se deben crear versiones distintas de esta. Cada una de las versiones tiene que estar desarrollada bajo los estrictos guidelines de cada sistema operativo móvil.

4. DESARROLLO

4.1. Como primer punto en el desarrollo de nuestro examen se creó un nuevo proyecto de xamarin forms en visual studio, ya una vez dentro del proyecto se procede a ir al mainpage.xaml en el cual editaremos la interfaz grafica que se mostrara en el dispositivo o emulador



4.2. En el mainpage se procede a crear los stacklayouts que contendrán la información, en este caso se le dio las distintas informaciones de encabezado y las preguntas



4.3. En el mismo mainpage se procede a colocar las características y la distinta información que contendrá cada uno de las respuestas las cuales cierran almacenadas dentro de un botón

```
<StackLayout Margin="20, 150,20,20">
  <Button Text="es un framework de desarrollo web" x:Name="btnresp1"
    TextColor="White"
    BackgroundColor="Gray"
    BorderColor="CadetBlue"
    BorderWidth="3"
    CornerRadius="10"
    Clicked="Button1_Clicked" ></Button>

  <Button Text="un lenguaje de programacion" x:Name="btnresp2"
    TextColor="White"
    BackgroundColor="Gray"
    BorderColor="CadetBlue"
    BorderWidth="3"
    CornerRadius="10"
    Clicked="Button2_Clicked" ></Button>

  <Button Text="una plataforma para crear aplicaciones multiplataforma" x:Name="btnresp3"
    TextColor="White"
    BackgroundColor="Gray"
    BorderColor="CadetBlue"
    BorderWidth="3"
    CornerRadius="10"
    Clicked="Button3_Clicked" ></Button>

  <Button Text="un emulador" x:Name="btnresp4"
    TextColor="White"
    BackgroundColor="Gray"
    BorderColor="CadetBlue"
    BorderWidth="3"
    CornerRadius="10"
    Clicked="Button4_Clicked" ></Button>
</StackLayout>
```

4.4. Se selecciona la clase de mainpage.xaml.css en la cual se almacenarán todas las preguntas y con la cual nos ayudara a verificar cual es la respuesta correcta y que al final de esta se muestre el correspondiente resultado



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

```
public static List<Preguntas> listapreguntas = new List<Preguntas>
{
    new Preguntas
    {
        Id = 1,
        Quest_ = "1. QUE ES XAMARIN",
        resp1_ = "es un framework de desarrollo web",
        resp2_ = "un lenguaje de programacion",
        resp3_ = "una plataforma para crear aplicaciones multiplataforma",
        resp4_ = "un emulador",
        Correcto = "una plataforma para crear aplicaciones multiplataforma",
    },
    new Preguntas
    {
        Id = 2,
        Quest_ = "2. CUAL DE LAS SIGUIENTES PLATAFORMAS ES COMPATIBLE CON XAMARIN",
        resp1_ = "android",
        resp2_ = "IOS",
        resp3_ = "windows",
        resp4_ = "todas las anteriores",
        Correcto = "todas las anteriores",
    },
    new Preguntas
    {
        Id = 3,
        Quest_ = "3. QUE LENGUAJES DE PROGRAMACION SE PUEDEN UTILIZAR CON XAMARIN",
        resp1_ = "solo C#",
        resp2_ = "solo F#",
        resp3_ = "C# y F#",
        resp4_ = "C#, F# y JAVA",
        Correcto = "C# y F#"
    },
    new Preguntas
    {
        Id = 4,
        Quest_ = "4. CUAL ES LA VENTAJA DE USAR XAMARIN PARA DESARROLLAR APLICACIONES",
        resp1_ = "mayor velocidad de desarrollo",
        resp2_ = "mayor control sobre el hardware del dispositivo",
        resp3_ = "desarrollo mas sencillo",
        resp4_ = "ahorro en los costos de desarrollo",
        Correcto = "ahorro en los costos de desarrollo"
    }
};
int point = 1, score = 0;
string RespuestaCorrecta;
1 referencia
public MainPage()
{
    InitializeComponent();
}

0 referencias
protected void OnNavigatedTo(NavigationEventArgs e)
{
}
```



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

0 referencias

```
private void Button1_Clicked(object sender, EventArgs e)
{
    if (btnresp1.Text == RespuestaCorrecta)
    {
        score++;
    }
    point++;
    SetPregunta(point);
}
```

0 referencias

```
private void Button2_Clicked(object sender, EventArgs e)
{
    if (btnresp2.Text == RespuestaCorrecta)
    {
        score++;
    }
    point++;
    SetPregunta(point);
}
```

0 referencias

```
private void Button3_Clicked(object sender, EventArgs e)
{
    if (btnresp3.Text == RespuestaCorrecta)
    {
        score++;
    }
    point++;
    SetPregunta(point);
}
```

0 referencias

```
private void Button4_Clicked(object sender, EventArgs e)
{
    if (btnresp4.Text == RespuestaCorrecta)
    {
        score++;
    }
    point++;
    SetPregunta(point);
    btnPuntuacion.IsVisible = true;
}
```

0 referencias

```
async void MostrarAlerta(object sender, EventArgs e)
{
    await DisplayAlert("Puntuación", "Su Puntuación final es de: " + score + " / 4 puntos.",
    );
}
```

4.5. Se procede a crear una nueva clase que almacenará la prueba y todas las respuestas de la misma

```
16 referencias
public class Preguntas
{
    private int id;
    private string Quest;
    private string resp1;
    private string resp2;
    private string resp3;
    private string resp4;
    private string correcto;

    5 referencias
    public int Id { get { return id; } set { id = value; } }
    5 referencias
    public string Quest_ { get { return Quest; } set { Quest = value; } }
    5 referencias
    public string resp1_ { get { return resp1; } set { resp1 = value; } }
    5 referencias
    public string resp2_ { get { return resp2; } set { resp2 = value; } }
    5 referencias
    public string resp3_ { get { return resp3; } set { resp3 = value; } }
    5 referencias
    public string resp4_ { get { return resp4; } set { resp4 = value; } }
    5 referencias
    public string Correcto { get { return correcto; } set { correcto = value; } }

    5 referencias
    public Preguntas() { }

    1 referencia
    public Preguntas setting_pregunta(Preguntas quest) { return quest; }
}
```

4.6. Se procede a crear las distintas clases que se encargaran de crear las preguntas y el respectivo formulario

```
namespace examen1
{
    0 referencias
    public class PreguntasQuiz
    {
        protected PreguntasBuilder preguntas;

        0 referencias
        public void SetPreguntasBuilder(PreguntasBuilder question) { preguntas = question; }

        0 referencias
        public Preguntas Gettingpreguntas() { return preguntas.gettingQuestion(); }

        0 referencias
        public void BuildingQuestions()
        {
            preguntas.CreandoInstancia();
        }
    }
}
```



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

```
using System;
using System.Collections.Generic;
using System.Text;

namespace examen1
{
    3 referencias
    public abstract class PreguntasBuilder
    {
        1 referencia
        protected Preguntas QuestionQuiz;

        1 referencia
        public Preguntas gettingQuestion() { return QuestionQuiz; }

        1 referencia
        public void CreandoInstancia() { QuestionQuiz = new Preguntas(); }

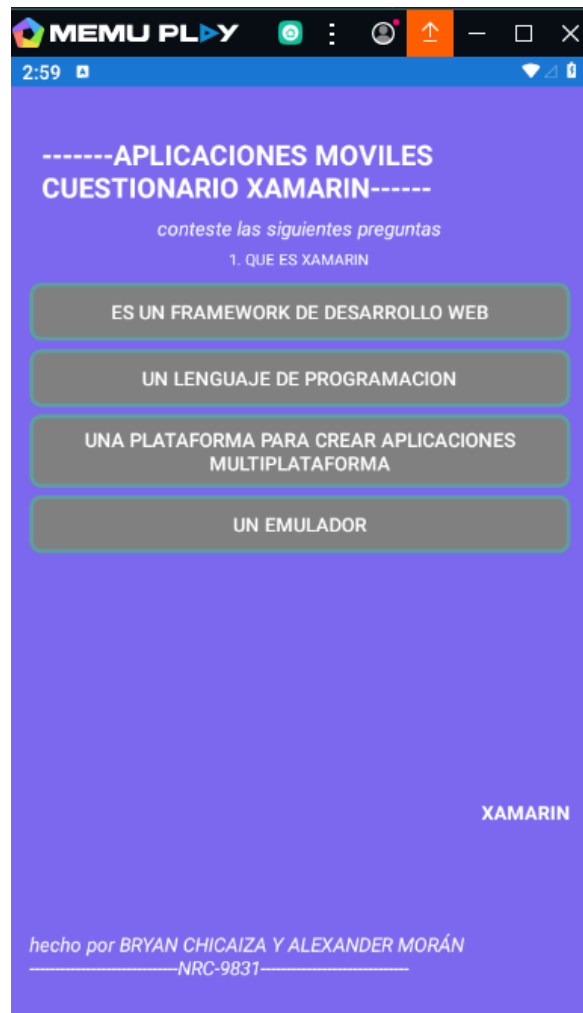
        1 referencia
        public abstract void Building(Preguntas pregunta);
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Text;

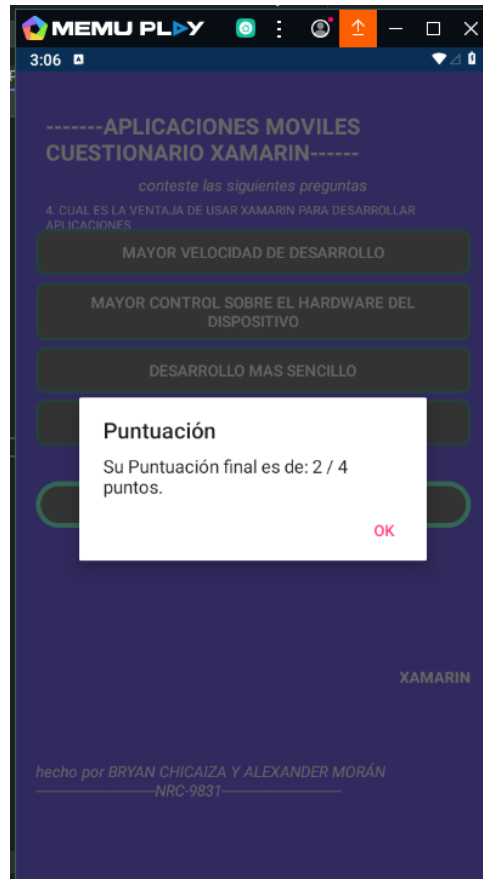
namespace examen1
{
    0 referencias
    public class Cuestionario : PreguntasBuilder
    {
        1 referencia
        public override void Building(Preguntas p)
        {
            QuestionQuiz.setting_pregunta(p);
        }
    }
}
```



4.7. Ya terminadas todas las codificaciones se procede a emular el aplicativo en MEMU



4.8. Ya terminada la encuesta se procede a mostrar el total que se haya obtenido



5. CONCLUSIONES

- El desarrollo de aplicaciones móviles en Xamarin puede proporcionar una solución eficaz para ofrecer contenido interactivo a los usuarios.
- La funcionalidad de evaluación automática de respuestas puede mejorar la experiencia de aprendizaje del usuario al proporcionar retroalimentación inmediata.
- Es importante llevar a cabo pruebas exhaustivas y controles de calidad para garantizar la estabilidad y fiabilidad de la aplicación.

6. RECOMENDACIONES

- Se debe prestar atención al diseño de la interfaz para mejorar la usabilidad y la experiencia del usuario.
- Es recomendable utilizar patrones de diseño como el patrón Modelo-Vista-Controlador (MVC) para una mejor organización y mantenibilidad del código.
- Se deben considerar las diferentes plataformas y dispositivos móviles para garantizar la compatibilidad, la escalabilidad y el rendimiento de la aplicación.

7. BIBLIOGRAFÍA

- profexorgeek. (2022, September 20). *What is Xamarin?* - *Xamarin*. Microsoft.com. <https://learn.microsoft.com/en-us/xamarin/get-started/what-is-xamarin>
- *Aplicaciones híbridas: qué son, frameworks, ejemplos y ventajas* - *ADN Cloud*. (2019, August 4). #ADNCLOUD. <https://blog.mdcloud.es/aplicaciones-hibridas-frameworks-ejemplos-y-ventajas/#:~:text=Las%20aplicaciones%20h%C3%ADbridas%2C%20a%20diferencia,de%20su%20marca%20o%20fabricante>.
- *Xamarin*. (2023). *Imagina Formación*. <https://imaginaformacion.com/tutoriales/que-es-xamarin-como-crear-un-proyecto>



- Colaborador de TechTarget. (2021). *Aplicación híbrida o app híbrida*.

ComputerWeekly.es; TechTarget.

<https://www.computerweekly.com/es/definicion/Aplicacion-hibrida-o-app-hibrida>

8. LEGALIZACION:

MORAN SANTO ALEXANDER ADAIR	
CHICAIZA SARABIA BRYAN ISRAEL	