



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

NRC:

6526

ASIGNATURA:

APLICACIONES MÓVILES

PROYECTO 2.1

TEMA:

Aplicación Calculadora en Xamarin.

INTEGRANTE:

Bryan Llangarí

Jonathan Pila

José Sailema

Lennin Calderón

Isaac Chanataxi

DOCENTE:

Ing. José Rubén Caiza Caizabuano

FECHA:

22/Julio/2022

PROYECTO 2.1

1. TEMA:

Aplicación Calculadora en Xamarin.

2. OBJETIVOS:

2.1 Objetivo General

- Realizar una aplicación móvil que me permita ejecutar operaciones matemáticas básicas como es suma, resta, multiplicación y división.

2.2 Objetivos Específicos

- Diseñar la interfaz de la aplicación móvil aspectos como ubicación de botones, tamaño, y color.
- Crear botones con el fin de que nos permita ingresar datos y obtener resultados de las operaciones de matemáticas básica mediante la aplicación.
- Realizar los cálculos necesarios de acuerdo a cada función para obtener resultados exactos.

3. MARCOTEÓRICO

VisualStudio

Visual Studio .NET es un entorno de desarrollo integrado, lo cual también es conocido como IDE (Integrated Development Environment). Por lo tanto, este comprende un conjunto de herramientas que pueden simplificar bastante el desarrollo de sitios, aplicaciones y servicios web.

Visual Studio .NET cuenta con todas las características que esperas de un IDE como la opción de autocompletado, identificación de código y resaltado de errores. Además, puede generar código de forma automática, permite arrastrar y soltar controles a un formulario y provee atajos de trabajo que facilitan compilar y construir las aplicaciones.

Por otra parte, es posible hacer uso de este ambiente de programación visual de

Microsoft para crear servicios web basados en el uso de la sintaxis XML (Extensible Markup Language). Visual Studio .NET viene con .NET Framework, incluyendo el CLR (Common Language Runtime) y varios lenguajes de programación como Visual Basic, Visual C++ y Visual C#.

El paquete del producto ofrece una interfaz visual que permite identificar un programa como servicio web, construir una interfaz de usuario (también soporta interfaces para dispositivos móviles), posee características que permiten la integración de aplicaciones de datos existentes y de depuración. (CERTUS, s.f.)

Ilustración 1

Símbolo de Visual Studio



Xamarin

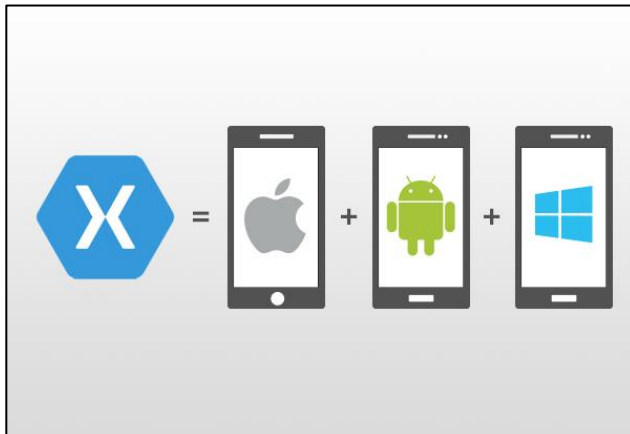
Xamarin es un software imprescindible para el desarrollo de aplicaciones móviles, puesto que permite a los desarrolladores compartir hasta un 90% del código entre plataformas distintas, utilizando un único lenguaje de programación, **C#**, y que cada vez está ganando más adeptos tanto por el rendimiento como por la posibilidad del acceso al API nativo.

Compilado de forma nativa, Xamarin es una buena herramienta para crear aplicaciones de alto rendimiento con aspecto nativo. Cuenta con **Xamarin.iOS** y **Xamarin.Android** como principales clientes, que compilan de

forma automática el código fuente con el método **Ahead-of-time** en caso de iOS y con lenguaje intermedio y posteriormente en **AOT** para Android, con la posibilidad de adaptarse para solucionar posibles problemas como la asignación de memoria. (FORMADORES, s.f.)

Ilustración 2

Símbolo de Xamarin.



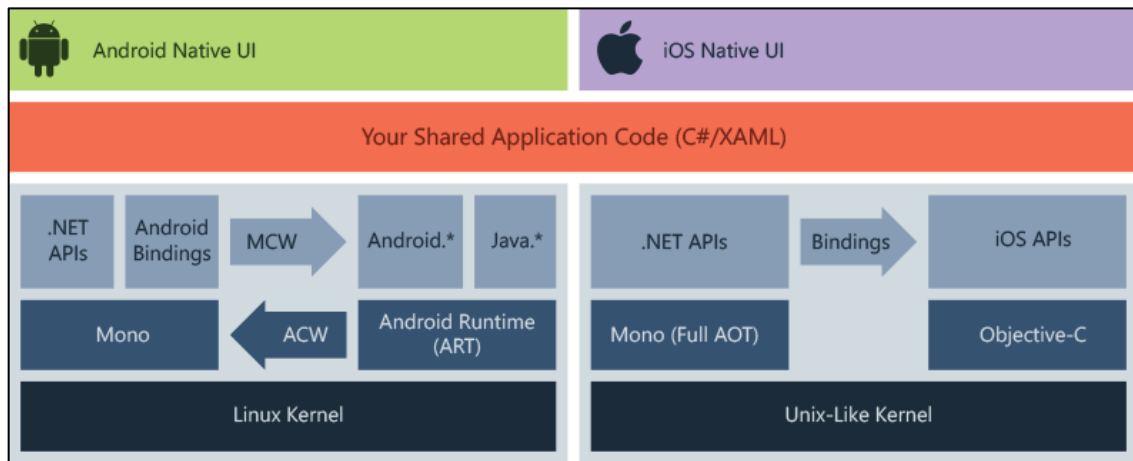
Xamarin es para desarrolladores con los siguientes objetivos:

- Compartir código, pruebas y lógica de negocios entre plataformas.
- Escribir aplicaciones multiplataforma en C# con Visual Studio.
- Funcionamiento de Xamarin
- Diagram of Xamarin architecture

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

Ilustración 3

Arquitectura de una aplicación en Xamarin.

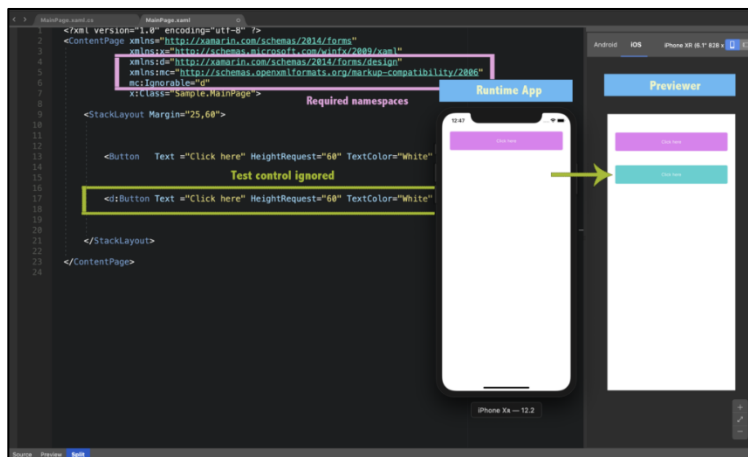


Xamarin.Forms.Xaml

El Lenguaje de marcado de aplicaciones extensible (XAML) es un lenguaje basado en XML creado por Microsoft como una alternativa al código de programación para crear instancias e inicializar objetos y organizar esos objetos en jerarquías padre-hijo. XAML se ha adaptado a varias tecnologías dentro del marco .NET, pero ha encontrado su mayor utilidad en la definición del diseño de las interfaces de usuario dentro de Windows Presentation Foundation (WPF), Silverlight, Windows Runtime y Universal Windows Platform (UWP). (Microsoft, 2021)

Ilustración 4

Arquitectura de una aplicación en Xamarin.



4. DESARROLLO

Se empezó creando un nuevo proyecto en Visual Studio se escogió la plantilla Aplicación Móvil (Xamarin.Forms) se asignó un nombre al proyecto, se eligió en blanco y crear.

Ilustración 1

Nuevo Proyecto

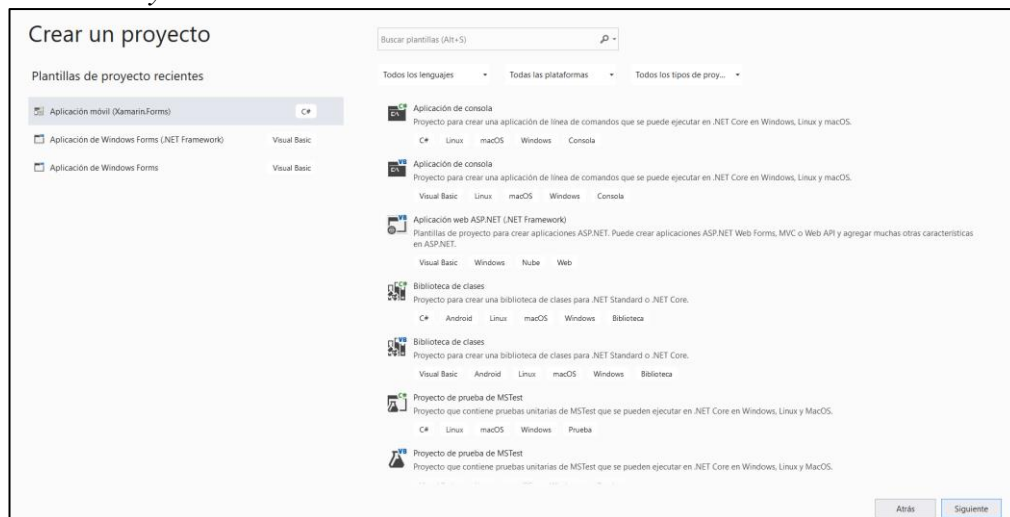


Ilustración 2

Nombre del proyecto

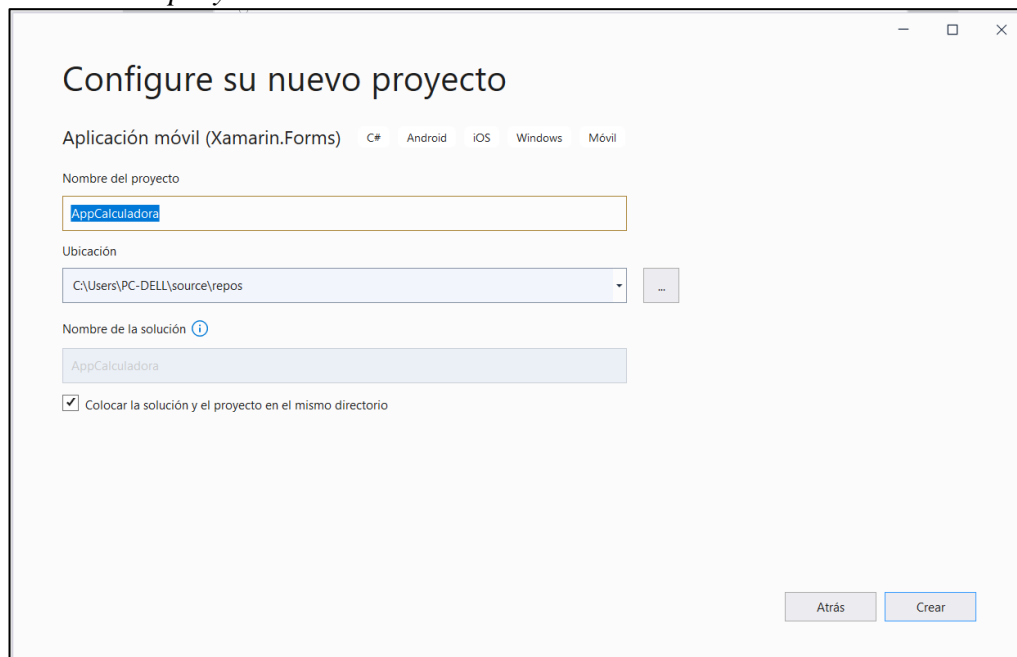
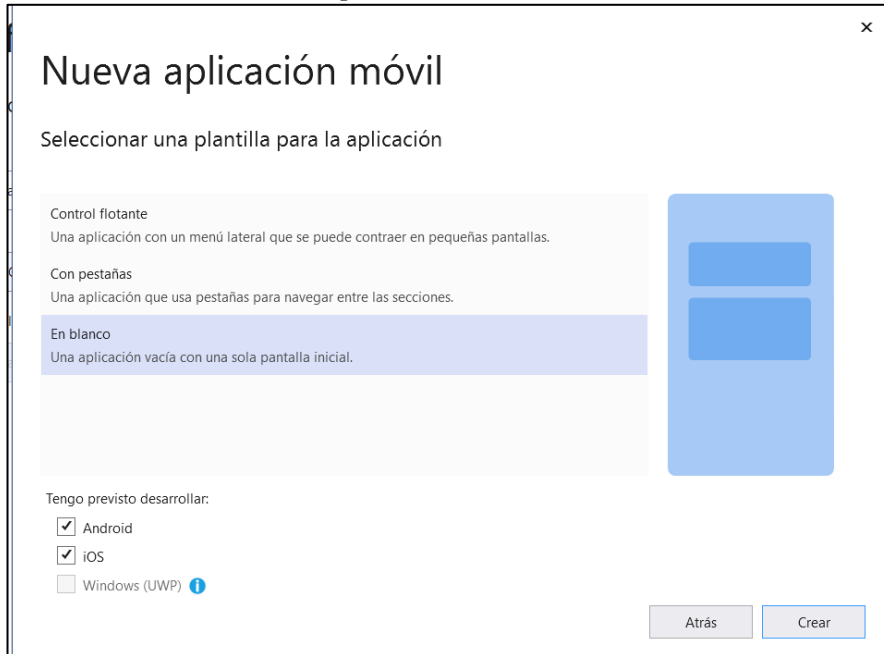


Ilustración 3

Selección en blanco de la plantilla



En la Página de contenido Menu.xaml en donde se definió el contenido visual de la página y su funcionamiento en conjunto con un archivo de código subyacente, en el archivo XAML, en el cual se definió las interfaces para el usuario en la cual se utilizó Controles (Views), Diseños (Layouts), Páginas (Pages) o clases personalizadas, también se declaro una clase que era la public partial class en la cual están ddefinidas no una sino varias clases a la vez.

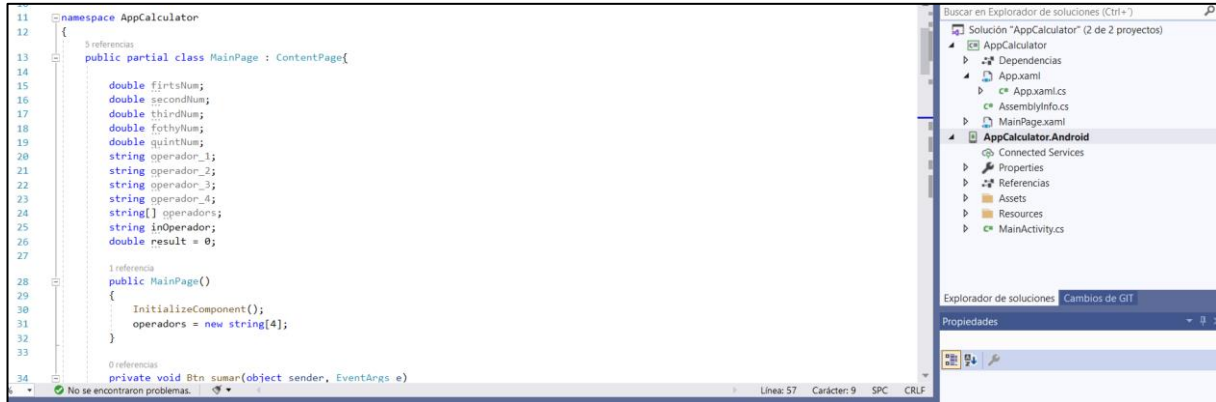
Ilustración 4

Página de contenido Menu.xaml



Ilustración 5

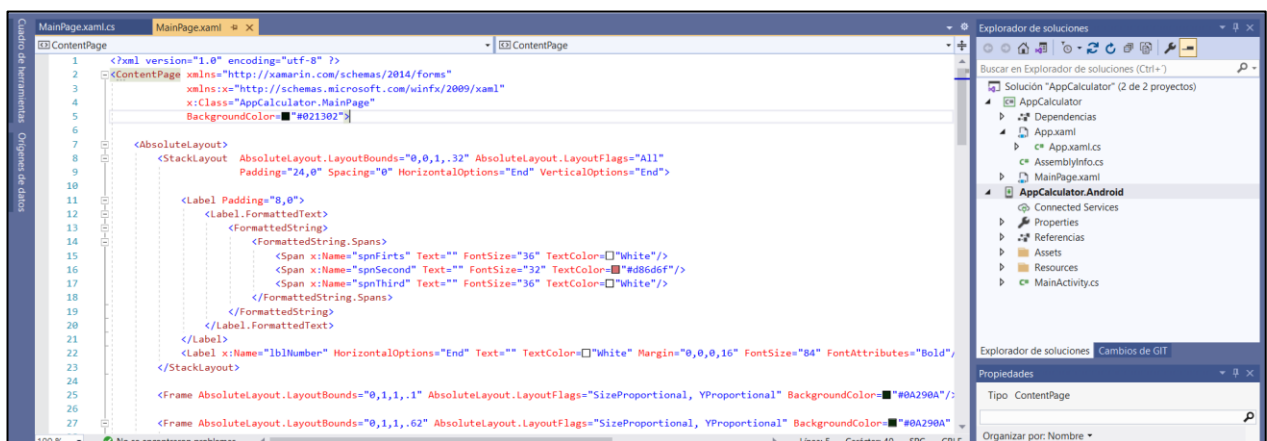
Código subyacente

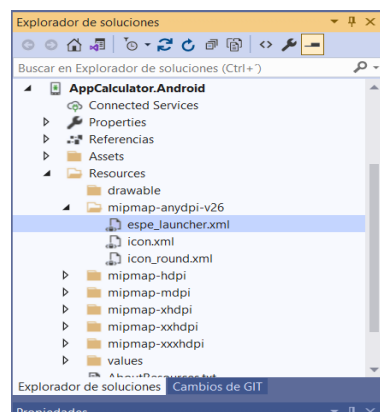


Después de esto también se procedió a la edición del fondo de la aplicación con algunos comandos como el **BackColor**, **FontSize**, **TextColor**, **VerticalOptions** entre otras líneas de codificación, las cuales nos ayudan al alineado y la personalización del texto de la aplicación a desarrollarse, se procedió a la creación de botones cada uno con su respectiva comando los mismos que se uso para el desarrollo del menú.

Ilustración 6

Página de contenido *AppCalculadora.xml*





Al terminar todos estos pasos se dio la ejecución del programa en donde se pudo visualizar que la aplicación tiene una funcionalidad cumpliendo con el objetivo de la creación de la aplicación.

Ilustración 9

Ejecución de la App

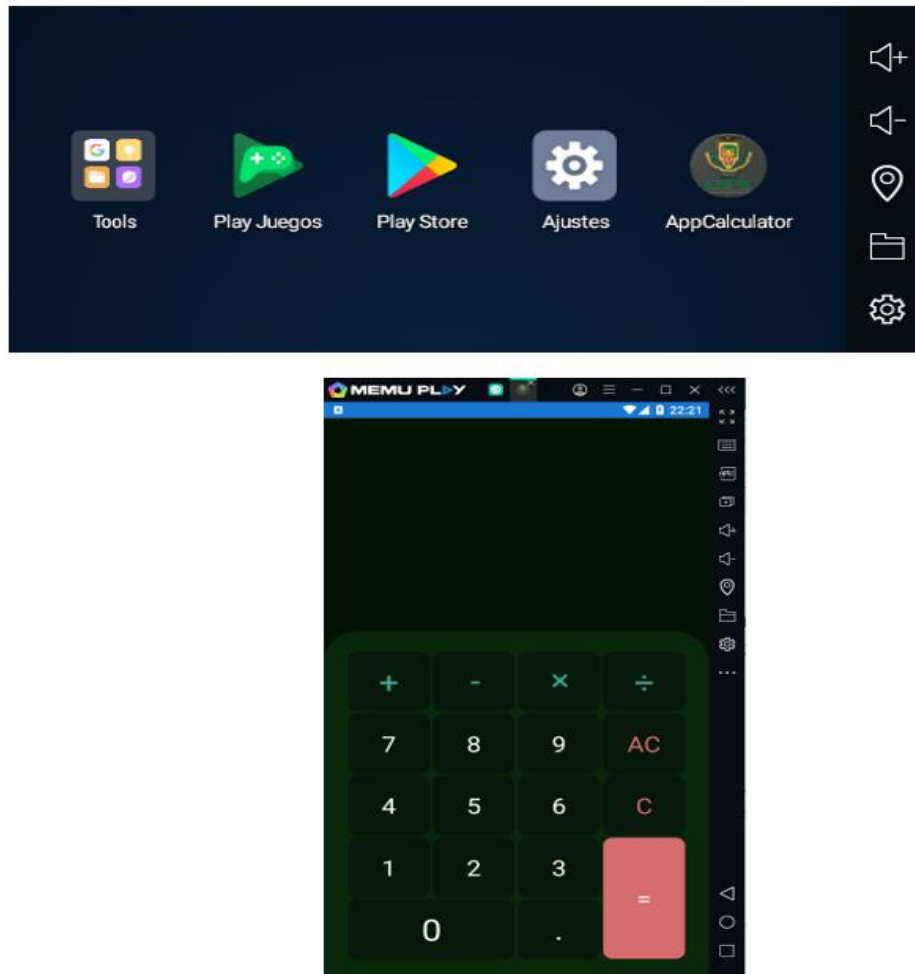
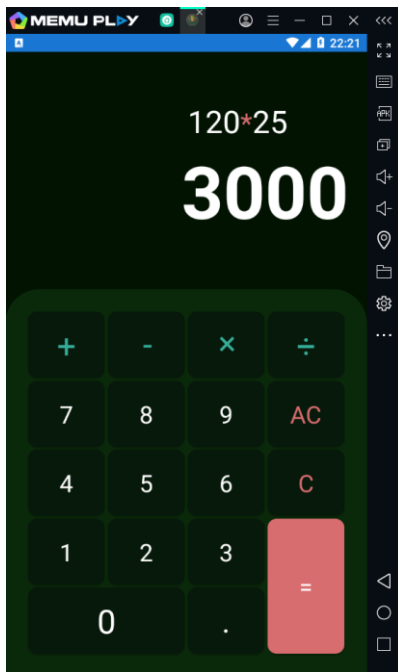


Ilustración 10

Calculadora aplicando la Operación Multiplicación



5. CONCLUSIONES

- Se concluye que en `mainpage.xml` mediante los siguientes códigos se puede modificar o cambiar el formato y estilo de contenido de la aplicación son `BackgroundColor`, `FontSize`, `TextColor`, `FontAttributes`, `VerticalOptions`, `HorizontalOption` entre otros.
- Se creó los botones los cuales nos ayudan al cálculo de cada función de matemática básica, mediante esto la aplicación es más amigable para el usuario.
- La clase que se utilizó para implementar el código de los cálculos es `public partial class` porque la clase está definida usando más de un archivo de código fuente y métodos como el `private void` porque el campo o método sólo es visible dentro de la clase donde se define.

6. RECOMENDACIONES

- Se recomienda identificar el código correcto para cambiar la interfaz gráfica de la aplicación.
- Es importante al momento de realizar la aplicación se la realiza de forma ordenada para evitar errores al momento de ejecutarla.
- Tener en cuenta la declaración de las variables, para que en el momento de ejecutar el código en la aplicación ejecute lo codificado.

7. BIBLIOGRAFÍA

CERTUS. (s.f.). Obtenido de <https://www.certus.edu.pe/blog/que-es-visual-studio-net/>

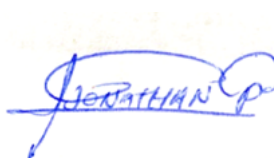
FORMADORES. (s.f.). Obtenido de <http://www.formadoresit.es/xamarin-que-es-y-para-que-sirve/>

Microsoft. (s.f.). *docs.microsoft*. Obtenido de <https://docs.microsoft.com/en-us/dotnet/api/xamarin.forms.contentpage?view=xamarin-forms>

1. LEGALIZACIÓN



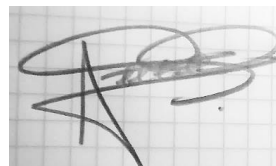
Llangarí Bryan




Pila Jonathan



Sailema José



Calderón Lenin



Chanataxi Isaac

