

PROYECTO DE USAID

PUENTES PARA EL EMPLEO

Fundación Gloria de Kriete

Centro de Desarrollo de Software

Diseño y Maquetado de Formularios con XAML

MANUAL DEL PARTICIPANTE

Nivel 1

2018

San Salvador, El Salvador, Centroamérica



La reproducción de este material ha sido posible gracias al generoso apoyo del pueblo de los Estados Unidos a través de la Agencia de los Estados Unidos para el Desarrollo Internacional (USAID). El contenido es responsabilidad exclusiva del Universidad Salvadoreña Alberto Masferrer; no refleja necesariamente la opinión de USAID o del Gobierno de los Estados Unidos

Contenido

[1. INFORMACIÓN DEL MÓDULO 2](#_Toc518731204)

[1.1 Campo de Especialidad 2](#_Toc518731205)

[1.2 Código del Módulo 2](#_Toc518731206)

[1.3 Duración del módulo 3](#_Toc518731207)

[2. DESCRIPCIÓN GENERAL DEL MÓDULO 4](#_Toc518731208)

[2.1 Descripción 4](#_Toc518731209)

[2.2 Competencia del módulo 4](#_Toc518731210)

[2.3 Criterios de evaluación 4](#_Toc518731211)

[2.4 Criterios de promoción 4](#_Toc518731212)

[3. DESARROLLO DE UNIDADES DE APRENDIZAJE 5](file:///C:\Users\SACDS03\Desktop\Manual%20XAML\M.1.15%20Diseño%20y%20Maquetado%20de%20Formularios%20con%20XAML%20(1).docx#_Toc518731213)

[3.1 Unidad 1: XAML 5](#_Toc518731214)

[3.1.1 Identificar conceptos principales del lenguaje XAML. 5](#_Toc518731215)

[Actividad Propuesta1 7](#_Toc518731216)

[3.2 Unidad 2: Formularios XAML. 8](#_Toc518731217)

[3.2.1 Diseñar formularios con etiquetas XAML. 8](#_Toc518731218)

[Actividad Propuesta 2 16](#_Toc518731219)

[3.3 Unidad 3: Estilo XAML. 17](#_Toc518731220)

[3.3.1 Aplicar diseño y estilo a formularios en Xamarin Form. 17](#_Toc518731221)

[Actividad Propuesta 3 21](#_Toc518731222)

[4. GLOSARIO 22](#_Toc518731223)

[5. PLAN DE EVALUACIONES 23](#_Toc518731224)

[6. JORNALIZACIÓN DEL MÓDULO 24](#_Toc518731225)

[7. FUENTES DE INFORMACIÓN Y MATERIALES DE APOYO 25](#_Toc518731226)

# 

# 1. INFORMACIÓN DEL MÓDULO

## 1.1 Campo de Especialidad

* Desarrollador de Aplicaciones Web usando C# ASP.net con MVC
* Desarrollador de Aplicaciones Móviles con C# Xamarin
* Programador Analista en Lenguaje PHP
* Programador Analista en Lenguaje JAVA.

## 1.2 Código del Módulo

DMFX

## 1.3 Duración del módulo

2 Semanas





# 2. DESCRIPCIÓN GENERAL DEL MÓDULO

## 2.1 Descripción

El módulo Diseño y Maquetado de Formularios con XAML realiza una inmersión teórica y práctica sobre las etiquetas de XAML más utilizados en el diseño de interfaces para dispositivos móviles. Se enseñará las distintas páginas, controles y propiedades que pueden tener dichas etiquetas en los formularios.

## 2.2 Competencia del módulo

Hacer uso de etiquetas XAML para el diseño de formularios en dispositivos móviles.

## 2.3 Criterios de evaluación

* Utiliza etiquetas XAML para el maquetado de formularios
* Aplica propiedades para cambiar el aspecto de los controles
* Usa distintas paginas para el diseño de los formularios

## 2.4 Criterios de promoción

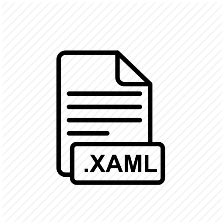
Para dar por aprobado un módulo el participante deberá:

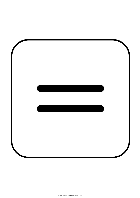
* Haber asistido al 100% de las clases del módulo.
* Desarrollar las actividades que se presentan en el manual.
* Obtener un promedio final mayor o igual a 7.0 en las actividades del módulo.

## 3.1 Unidad 1: XAML

# 3. DESARROLLO DE UNIDADES DE APRENDIZAJE

## 3.1.1 Identificar conceptos principales del lenguaje XAML.

XAML (acrónimo pronunciado xammel del inglés eXtensible Application Markup Language, Lenguaje Extensible de Formato para Aplicaciones en español) es el lenguaje de formato para la interfaz de usuario para la Base de Presentación de Windows (WPF por sus siglas en inglés) y Silverlight(wpf/e), el cual es uno de los "pilares" de la interfaz de programación de aplicaciones .NET en su versión 3.0 (conocida con anterioridad con el nombre clave WinFX).

****XAML es un lenguaje declarativo basado en XML, optimizado para describir gráficamente interfaces de usuarios visuales ricas desde el punto de vista gráfico, tales como las creadas por medio de Adobe Flash. XUL y UIML son otros ejemplos de lenguajes de interfaz basados en XML. SVG es un estándar de la organización W3C, el cual soporta gráficos, animaciones, audio y video integrados, eventos y comportamiento descrito por medio de escritura y puede ser utilizado como lenguaje de interfaz basado en XML.



XAML: historia y similitudes:

***XAML*** fue inicialmente diseñado como parte de ***Windows Presentation Foundation (WPF)***, donde les permite a los desarrolladores de aplicaciones Windows construir interfaces de usuario ricas. Usted usará el mismo concepto para contruir las interfaces de usuarios para sus aplicaciones Silverlight. Conceptualmente, *XAML* juega un papel muy parecido a *HTML* o mejor dicho es más cercano al más estricto *XHTML.* *XHTML* le permite definir los elementos que forman una página web común. De manera similar, *XAML* le permite definir los elementos que conforman un área de contenido *XAML*. Para manipular elementos *XHTML* en una página *XHTML*, usamos JavaScript del lado del cliente. Para manipular elementos *XAML,* escribimos código *VB* o *C#* del lado del cliente. *XAML* and *XHTML* comparten muchas convenciones de sintáxis, ya que como *XHTML, XAML* es un lenguaje basado en *XML* que consiste en elementos que pueden ser anidados de la manera que quiera.

Imagen que contiene texto

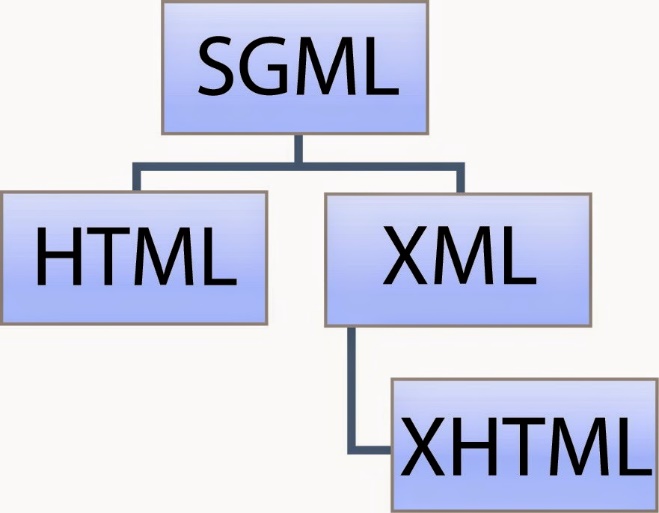
Descripción generada con confianza alta

**MainPage.xaml:**MainPage.xaml es el archivo que abre automáticamente Visual Studio en el diseñador. Aquí es donde podemos declarar todos los elementos visuales que requiera la apliacion ya que actua como la raíz visual para la aplicación de manera predeterminada.  
  
  
**MainPage.xaml.cs:**MainPage.xaml.cs es el archivo code-behind de MainPage.xaml, que hace uso del lenguaje de programación C# (de ahí su extensión .cs ya que si fuera una aplicación de Visual Basic seria .vb). En este archivo es posible escribir todo el código y funcionabilidad que deseemos para MainPage.xaml.

Imagen que contiene captura de pantalla

Descripción generada con confianza muy alta

***Diferencias de XAML con otros lenguajes.***

******

XML deriva de SGML, que es el lenguaje de marcado maestro. XML tiene un esquema configurable, en el que se pueden definir las etiquetas y los atributos, mientras que HTML tiene un esquema definido con etiquetas que describen la presentación del contenido para un navegador web, o cualquier programa que use HTTP y analice HTML.

Hay diferentes versiones de HTML, como 1.0, 2.0, 3.0, 4.0 y 5.0. Cada nueva versión del esquema se vuelve mas estructurado y bien-formado. HTML no es exactamente XML, ese sería XHTML, pero ahora con HTML 5 es mejor.

XAML es otro esquema que deriva de XML que describe la presentación de contenido para analizadores que producen WPF, Silverlight y otros productos de Microsoft. Es uno de los "pilares" de la API de .NET en su versión 3.0.

Ejercicio:

1. ¿Cuáles son las similitudes de XAML con otros lenguajes?
2. ¿Qué significa XAML?
3. ¿Cuál es la característica o las características principales de XAML?
4. Investigar cuales son las ventajas y desventajas de XAML
5. ¿Qué es una Content Page?

### Actividad Propuesta1

ACTIVIDAD #1

Identificar conceptos principales del lenguaje XAML.

**OBJETIVO:**

Que el alumno sea capaz de identificar la importancia que tiene el legunaje XAML, tanto asi como sus conceptos basicos.

**INTRUCCIONES:**

1. Lea detenidamente cada una de las interrogantes planteadas a continuacion.
2. Debe contestar cada una de las preguntas planteadas en la prueba escrita.
3. Al finalizar la prueba escrita debe llamar al entrenador y esperar indicaciones.

**Preguntas:**

1. **Cuáles son las similitudes de XAML con otros lenguajes?**
2. **Que significa XAML**
3. **Cual es la característica o las características de XAML**
4. **Investigar cuales son las ventajas y desventajas de XAML**
5. **Que es un countent Page**
6. **Controles mas utilizados en XAML**



**TIEMPO ESTIMADO:**

50 minutos



**PARTICIPANTES:**

Individual



**MATERIALES**

Prueba escrita   
lapicero azul

## 3.2 Unidad 2: Formularios XAML.

## 3.2.1 Diseñar formularios con etiquetas XAML.

***Etiquetas XAML:***

(Tag). Marca o marcas que se dejan en un texto para que luego sean interpretadas, generalmente para realizar alguna acción sobre el mismo texto marcado. Por ejemplo, el lenguaje XAML se basa en la aplicación de múltiples etiquetas para dar formato y darle características a un texto y a diversos elementos. Este texto es interpretado y mostrado por un Emulador.

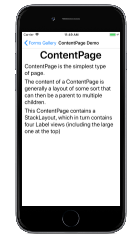
**Principales etiquetas XAML:**

***Paginas***

**ContentPage**

ContentPage es el tipo de página más simple y común. Establezca la propiedad de contenido en un único objeto de vista, que a menudo es un diseño como StackLayout, Grid o ScrollView.

<ContentPage> </ContentPage>



***MasterDetailPage***

Un *MasterDetailPage* maneja dos paneles de información. Establezca la propiedad principal en una página que generalmente muestra una lista o menú. Establezca la propiedad ***Detalle(Detail****)* en una página que muestre un elemento seleccionado de la Master Page.

<MasterDetailPage.Master>

</MasterDetailPage.Master>

<MasterDetailPage.Detail>

</MasterDetailPage.Detail>



**NavigationPage**

NavigationPage gestiona la navegación entre otras páginas utilizando una arquitectura basada en pila. Al utilizar la navegación de página en su aplicación, una instancia de la página de inicio debe pasarse al constructor de un objeto NavigationPage.

Ejemplo de Modificacion en el código .cs

***Contenedores***

**StackLayout***StackLayout* coloca los elementos secundarios en una pila, ya sea horizontal o verticalmente, en función de la propiedad *Orientation* . La propiedad *Spacing* rige el espaciado entre los elementos secundarios y tiene un valor predeterminado de 6.

Ejemplo:   
<StackLayout> </StackLayout>

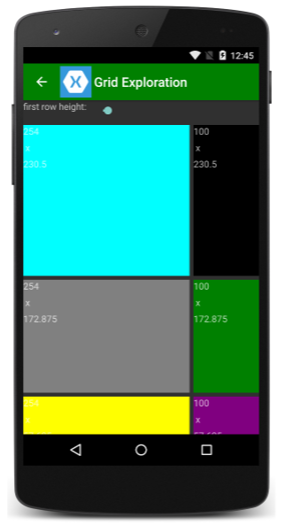


**Grid**

Grid admite organizar vistas en filas y columnas. Las filas y las columnas se pueden configurar para que tengan tamaños proporcionales o tamaños absolutos. El diseño de la cuadrícula no se debe confundir con las tablas tradicionales y no está destinado a presentar datos tabulares. Grid no tiene el concepto de formateo de filas, columnas o celdas. A diferencia de las tablas HTML, Grid está destinado exclusivamente a diseñar contenido.

<Grid> </Grid>





**ScrollView**  
  
ScrollView contiene diseños y les permite desplazarse fuera de pantalla. ScrollView también se usa para permitir que las vistas se muevan automáticamente a la parte visible de la pantalla cuando se muestra el teclado.

Ejemplo:   
<ScrollView> </ScrollView>



**ContentView**

*ContentView* contiene un único elemento secundario que está configurado con la propiedad Content. La propiedad de contenido se puede establecer en cualquier derivado de vista, incluidos otros derivados de diseño. *ContentView* se usa principalmente como un elemento estructural y sirve como una clase base para *Frame.  
  
Ejemplo:*<ContentView> </ContentView>

******

***Vistas***

* **Label**: Muestra cadenas de texto de una sola línea o bloques de texto de varias líneas, ya sea con formato constante o variable.

Imagen que contiene captura de pantalla

Descripción generada con confianza muy altaEjemplo:

<Label> </Label>

* **Entry:** permite al usuario ingresar y

editar una sola línea de texto.

Ejemplo:   
<Entry> </Entry>

Imagen que contiene captura de pantalla

Descripción generada con confianza alta

**Editor:** permite al usuario ingresar y editar múltiples líneas de texto.   
  
Ejemplo:   
<Editor> </Editor>

Imagen que contiene monitor, electrónica, teléfono, captura de pantalla

Descripción generada con confianza alta

**Nota Importante:** Para poder colorcar tu elemento tipo Imagen debes colocar tu imagen previamente en tu proyecto, para tu proyecto Android debes agregarlo en carpeta Resources > Drawable.  
para tu proyecto IOS debes agregalos en carpeta Resources.

* **Image**: Es un control que muestra una imagen. Puede usar el objeto

Imagen o el objeto ImageBrush. Un objeto Imagen muestra una imagen, mientras que un objeto ImageBrush pinta otro objeto con una imagen.

La fuente de la imagen se especifica al hacer referencia a un archivo de imagen que utiliza varios formatos compatibles.   
  
Ejemplo:

Imagen que contiene electrónica, captura de pantalla, iPod

Descripción generada con confianza muy alta <image source=” [Nombre]”></Image>

• **BoxView:** muestra un rectángulo sólido. BoxView tiene una solicitud de tamaño predeterminada de 40x40.  
  
Ejemplo:

<BoxView> </BoxView>



**• Button:** es un objeto rectangular que muestra texto y que activa un evento Clicked cuando se lo presiona.

Ejemplo:  
<Button> </Button>

Imagen que contiene captura de pantalla, electrónica

Descripción generada con confianza muy alta

**• SearchBar:** muestra un área para que el usuario escriba una cadena de texto y un botón (o una tecla del teclado) que indica a la aplicación que realice una búsqueda.  
  
Ejemplo:  
<SearchBox> </ SearchBox>

Imagen que contiene captura de pantalla

Descripción generada con confianza muy alta

* **Slider**: es una barra horizontal que puede ser manipulada por el usuario para seleccionar un valor doble de un rango continuo.

Ejemplo:

Imagen que contiene captura de pantalla, electrónica

Descripción generada con confianza muy alta <Slider> </Slider>

* **Switch:** toma la forma de un interruptor de encendido / apagado para permitir al usuario seleccionar un valor booleano.

Ejemplo:

<Switch> </Switch>

Imagen que contiene electrónica, iPod, captura de pantalla

Descripción generada con confianza alta

* **DatePicker:** permite al usuario seleccionar una fecha con el selector de fecha de la plataforma.

Ejemplo:

<DatePicker> </DatePicker>

Imagen que contiene monitor, electrónica

Descripción generada con confianza muy alta

* **TimePicker:** permite al usuario seleccionar una hora con el selector de tiempo de la plataforma.

Imagen que contiene monitor

Descripción generada con confianza muy altaEjemplo:   
<TimePicker> </TimePicker>

* **ActivityIndicator:** usa una animación para mostrar que la aplicación está involucrada en una actividad prolongada sin dar ninguna indicación de progreso.   
    
  Ejemplo:  
  <ActivityIndicator> </ActivityIndicator>

Imagen que contiene electrónica

Descripción generada con confianza alta

* Imagen que contiene captura de pantalla

  Descripción generada con confianza altaImagen que contiene captura de pantalla, electrónica, monitor

  Descripción generada con confianza alta**ProgressBar:** usa una animación para mostrar que la aplicación está progresando a través de una actividad prolongada.  
    
  Ejemplo:   
  <ProgressBar> </ProgressBar>
* Imagen que contiene captura de pantalla

  Descripción generada con confianza alta**Picker:** muestra un elemento seleccionado de una lista de cadenas de texto, y permite seleccionar ese elemento cuando se toca la vista.  
    
  Ejemplo:  
  <Picker> </Picker>
* **ListView:** muestra una lista desplazable de elementos de datos seleccionables.  
    
  Ejemplo:  
  <ListView> </ListView>  
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
    
  **Pagina de Referencias:**  
  Xamarin forms Pages.  
    
  **Ejercicio:**  
  Investigar en parejas haciendo uso del internet:  
    
  1- Controles que se pueden agregar (aparte de los anteriormente vistos)  
  2- ¿para que sirve un TableView?

3- ¿Qué es un StackLayout?  
4-tipos de contenedores. listar almenos 5  
5- ¿Se puede introducir , un StackLayout sobre otro StackLayout?

Ejercicio 2:  
  
1- desarollar una interfaz con los diferentes controles explicados por el entrenador.

Imagen que contiene captura de pantalla

Descripción generada con confianza muy alta2- desarrollar la siguiente interfaz

### Actividad Propuesta 2

ACTIVIDAD #2

Diseñar formularios con etiquetas XAML.

**OBJETIVO:**

Aplicar los diferentes tipos de controles en XAMARIN.

**INTRUCCIONES:**

1. Debe tener a su disposición la prueba con los Ejercicios Practicos.
2. Tendra 100 minutos para realizar la prueba Practica.
3. Al finalizar la prueba Practica debe llamar al entrenador para verificación del trabajo realizado.



**TIEMPO ESTIMADO:**

100 minutos



**PARTICIPANTES:**

Individual



**MATERIALES**

Computadora personal.

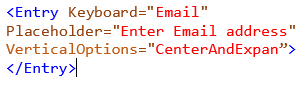
Pagina de Ejercicios practicos.

## 3.3 Unidad 3: Estilo XAML.

## 3.3.1 Aplicar diseño y estilo a formularios en Xamarin Form.

**¿Qué es un estilo?**

Dentro del contexto del diseño de interfaces de usuario, un estilo consiste en aquellacaracterística disponible en una herramienta de desarrollo, IDE, lenguaje, diseñador visual, etc., que permite definir unas reglas comunes de aspecto, aplicables

***Aplicando diferentes estilos a   
algunos controles.***

**Etiqueta Label**

A nuestro label podemos aplicarles estilo a sus diversas propiedades: entre ellas, el tamaño de la letralos aplicamos por medio de *FontSize*, aplicamos estilo a la letra por medio de la propiedad *FontAttributes,* podríamos aplicarle un estilo *Bold,* y en la para colocarlo en una posición especifica utilizamos la propiedad *HorizontalOptions*, y para cambiarle el texto que esta etiqueta va a tener , utilizamos la propiedad *Text* tal , al igual si deseamos cambiarle el color a dicha letra podríamos utilizar el atributo *TextColor* y lo podemos ver tal y como lo aplicamos en el siguiente ejemplo: <Label   
FontSize="50" FontAttributes="Bold"

HorizontalOptions="Center"  
Text="Label"

TextColor="Green">

</Label>

**Etiqueta Entry**

Propiedades a la etiqueta Entry:   
podemos aplicarle diversos estilos a nuestra etiqueta, el colocar el tipo de teclado lo cambiamos con la propiedad *Keyboard*, tambien podemos ser indicativos con lo que le solicitamos al usuario, en este caso para guiar al cliente podemos indicarle que debe colocar en nuestro Entry, con la propiedad *Placeholder*, y para adecuarlo a la pantalla de nuestra interfaz podríamos aplicarlo con la propiedad *VerticalOptions.*

**Etiqueta Image**

La etiqueta Image tiene propiedades que podemos cambiarle o modificarle, la primera parte es indicarle el nombre de la imagen que deseamos colocar con la propiedad *Source*, y luego para las dimensiones de la imagen utilizamos el *WidthRequest* y *HeightRequest,*y para adecuarlo a la pantalla de nuestra interfaz podríamos aplicarlo con la propiedad *VerticalOptions.*

<Image Source="ejemplo"

HeightRequest="46"

HorizontalOptions="Center"

VerticalOptions="CenterAndExpand"

WidthRequest=”46”>

</image>

**Etiqueta BoxView**

A la etiqueta BoxView al igual que las propiedades que hemos estado visualizando anteriormente se aplican a nuestro BoxView, para las dimensiones como hemos verificado utilizamos *WidthRequest* y *HeightRequest*, y dado que nos crea un cuadro de texto m podemos aplicarle el color que deseamos que posea, con la propiedad *Color*, y para su orientación en su pantalla tenemos las propiedades *HorizontalOptions* y *VerticalOptions*.

<BoxView   
Color="Accent"

WidthRequest="150"

HeightRequest="150"

HorizontalOptions="Center"

VerticalOptions="CenterAndExpand">  
</BoxView>

**Etiqueta Button**

La etiqueta Button tiene muchas propiedades de las cuales podemos hacer uso de ellas, con la etiqueta *Text*, le colocamos el texto que contendrá el botón, con el *BorderColor* aplicamos un color al borde del botón, luego con la propiedad *BorderRadius*, hacemos que el borde que le hemos colocado lo haga radial para mayor vistosidad, con *Font* le aplicamos un estilo predeterminado largo a la letra dentro de nuestro botón, con el *BorderWidth* establecemos el ancho de nuestro borde.

<Button Text="ColocarTexto"

BorderColor="Blue"

BorderRadius = "5"

Font="Large"

BorderWidth="1"

HorizontalOptions="Center"

VerticalOptions="CenterAndExpand">

**Etiqueta SearchBar**La etiqueta *SearchBar* tiene propiedades de las cuales podemos hacer uso de ellas, entre ellas la propiedad*Placeholder* ***,*** con la que podemos ser indicativos con lo que le solicitamos al usuario, en este caso para guiar al cliente podemos indicarle que debe colocar en nuestro SearchBar, con esta propiedad.

<SearchBar Placeholder="Buscar ">

</SearchBar>

***Algunas Propiedades Aplicativas para etiquetas:***

***Placeholder:*** *Indicacion de lo que se debe colocar ya sea en un Entry, searchBar , entre otras.* ***Margin:*** *Esta propiedad sirve para dejar una separacion entre el elemento al que se la pongamos y otro. Es una propiedad muy sencilla de usar y tiene mucha utilidad*

***Title:*** *esta propiedad es utilizada para colocar títulos, en dado caso, podríamos utilizar esta propiedad en una NavigationPage para poder ir verificando en que pagina estamos situados.*

***Color:*** *La propiedad color establece el color con el que se muestra el texto contenido en un elemento. El color de la letra no puede ser transparente, ya que se debe indicar explícitamente un color.*

***BackgroundColor:*** *con**la propiedad background se puede establecer de forma directa el color de fondo a un elemento predeterminado.*

***HorizontalTextAlignment:*** *En la posición que estamos ubicados con nuestro elemento podemos indicarle con este tipo de propiedad hacia que lado deseamos ubicar nuestro control horizontalmente, generalmente se maneja nde manera centrada es decir* ***“Center”.***

***VerticalTextAlignment:*** *En la posición que estamos ubicados con nuestro elemento podemos indicarle con este tipo de propiedad hacia que lado deseamos ubicar nuestro control en parte vertical.*

***Spacing:*** *Obtiene o establece un valor que indica la cantidad de espacio entre cada elemento.*

***Padding:*** *Obtiene o establece el relleno interior del control.*

***FontSize:*** *Obtiene o establece el tamaño de fuente para el contenido del elemento.*

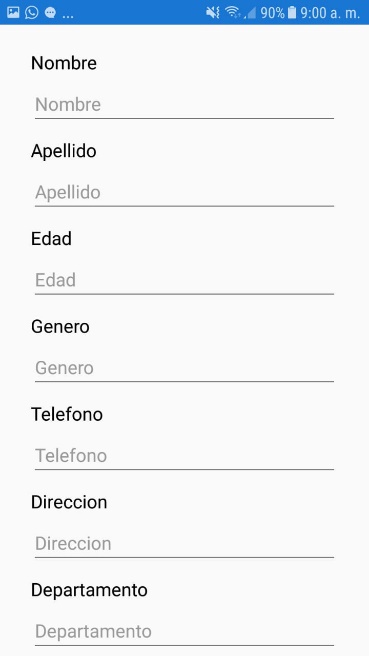
***Text:*** *se utiliza para mostrar un texto en un elemento predeterminado.*

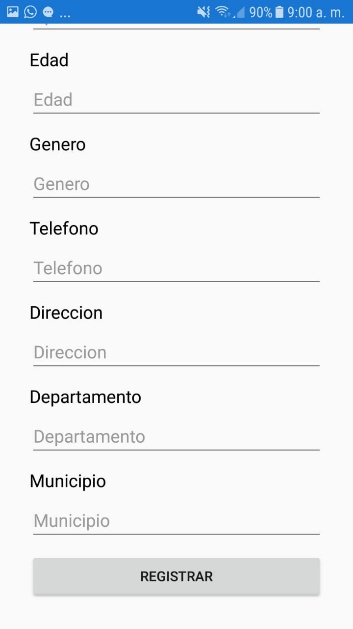
***Font:*** *Obtiene o establece la fuente del texto que muestra el control. un valor que indica la cantidad de espacio entre cada elemento.*

Nota: Las diversas propiedades de estilo pueden ser aplicadas a muchos controles

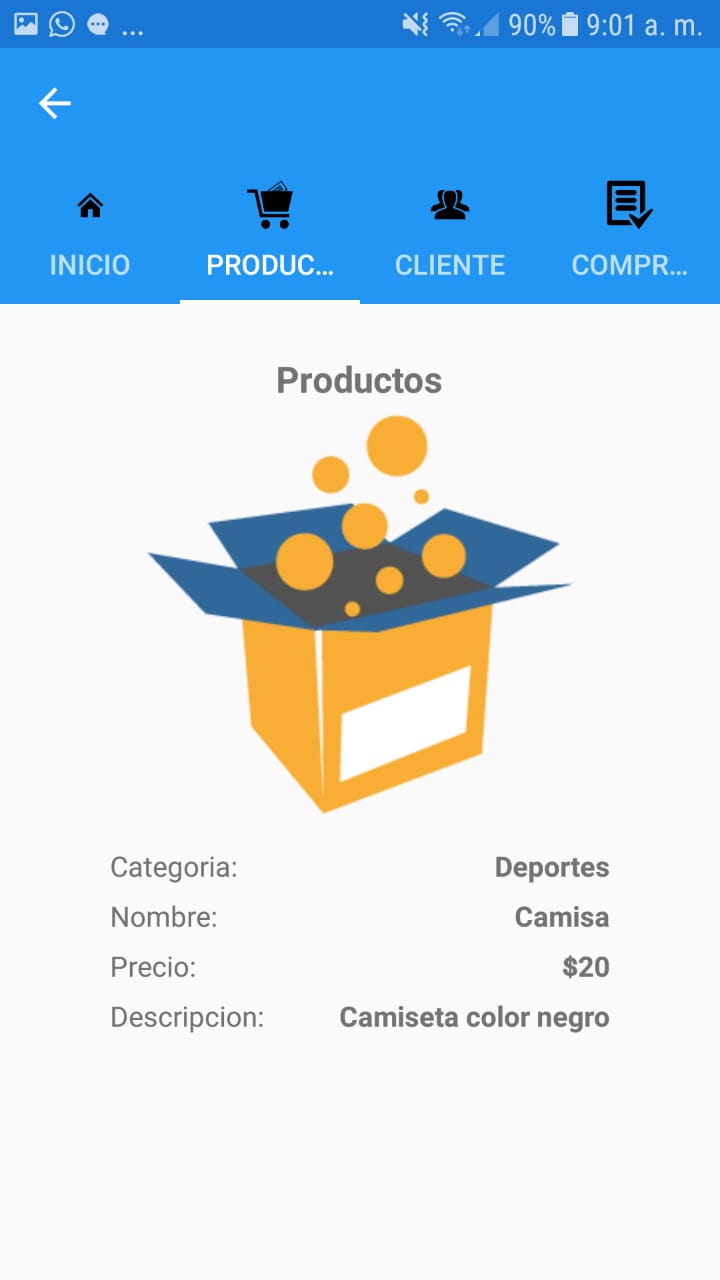
Ejercicios Practicos  
  
**Indicacion:** Realizar las siguientes pantallas como se muestra en las imágenes, se tomara como actividad evaluada .  
  
Guia de ejercicios:

1-

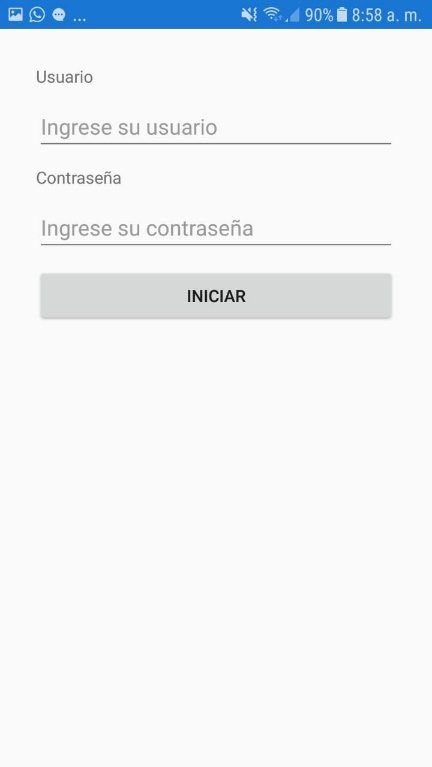




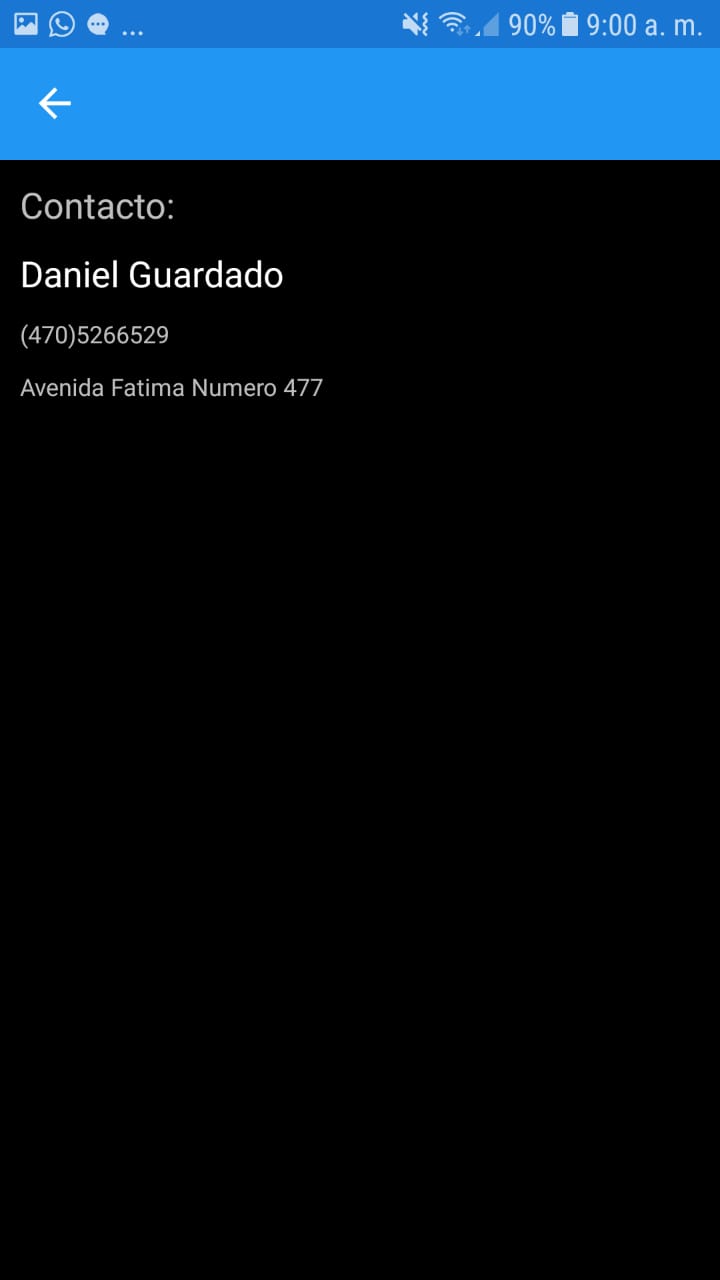
2 – desarrollar la siguiente Interfaz



3- Diseñar la siguiente pantalla



4-diseñar la siguiente interfaz



### Actividad Propuesta 3

ACTIVIDAD #3

Aplicar diseño y estilos a formularios en Xamarin Form.

**OBJETIVO:**

Aplicar diferentes estilos a diversos controles en XAMARIN.

**INTRUCCIONES:**

1. Debe tener a su disposición la prueba con los Ejercicios Practicos.
2. Tendra 100 minutos para realizar la prueba Practica.
3. Al finalizar la prueba Practica debe llamar al entrenador para verificación del trabajo realizado.



**TIEMPO ESTIMADO:**

100 minutos



**PARTICIPANTES:**

Individual



**MATERIALES**

Computadora personal.

Entorno de desarrollo integrado

# 4. GLOSARIO

**Interfaz**

El Interfaz es lo que conocemos en inglés como interface (“superficie de contacto”).se utiliza para nombrar a la conexión funcional entre dos sistemas, programas, dispositivos o componentes de cualquier tipo, que proporciona una comunicación de distintos niveles permitiendo el intercambio de información.

**Form**

Etiqueta de HTML que representa un formulario. En este formulario se agregan los diferentes campos de entrada de datos o de confirmación, así como los botones, que como mínimo ha de haber uno, el de envío. También pueden agregarse más botones como el de Restablecer que pone el formulario en blanco.

**HTML**

es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet. Se trata de la sigla que corresponde a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto, que podría ser traducido como Lenguaje de Formato de Documentos para Hipertexto.

**IOS:**

es un sistema operativo móvil de la multinacional Apple Inc. Originalmente desarrollado para el iPhone (iPhone OS), después se ha usado en dispositivos como el iPod touch y el iPad. No permite la instalación de iOS en hardware de terceros.

**Emulador**

Es En informática, un emulador es un software que permite ejecutar programas o videojuegos en una plataforma (sea una arquitectura de hardware o un sistema operativo) diferente de aquella para la cual fueron escritos originalmente. A diferencia de un simulador, que solo trata de reproducir el comportamiento del programa, un emulador trata de modelar de forma precisa el dispositivo de manera que este funcione como si estuviese siendo usado en el aparato original.

**IDE**

Un entorno de desarrollo integrado12 o entorno de desarrollo interactivo, en inglés Integrated Development Environment (IDE), es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

**XAMARIN**

Xamarin es una herramienta para los desarrolladores de aplicaciones móviles, y la novedad de esta herramienta es la capacidad que tiene para que el desarrollador escriba su app en lenguaje C# y el mismo código sea traducido para ejecutarse en iOS, Android y Windows Phone.

**Android**

Es Android es el nombre de un sistema operativo que se emplea en dispositivos móviles, por lo general con pantalla táctil, este sistema perativo esta basado en el SO linux.

# 5. PLAN DE EVALUACIONES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Actividad y  Tema evaluado | Tipo de evaluación | Semana | Ponderación |
| 1 | Actividad 1  Subtema 3.1 | Trabajo Individual | 1 | 20% |
| 2 | Actividad 2  Subtema 3.2 | Trabajo Individual | 1 | 30% |
| 3 | Ejercicio 3  Subtema 3.3 | Trabajo Individual | 1 | 20% |
| 4 | Actividad 4  Subtema 3.3 | Trabajo Individual | 2 | 30% |
| **TOTAL** | | | | **100%** |

# 6. JORNALIZACIÓN DEL MÓDULO

|  |  |  |  |
| --- | --- | --- | --- |
| SEMANA | SUB COMPETENCIA | ESCENARIO DE APRENDIZAJE | DURACIÓN EN HORAS |
| 1 | Tema 1: Identificar conceptos principales del lenguaje XAML. | TIC | 4 horas |
| 2 | Tema 1: Diseñar formularios con etiquetas XAML. | TIC | 12 horas |
| 3 | Tema 1: Aplicar diseño y estilo a formularios en Xamarin Form.. | TIC | 8 horas |
| **TOTAL DE HORAS DEL MÓDULO** | |  | **24 HORAS** |

# 7. FUENTES DE INFORMACIÓN Y MATERIALES DE APOYO

*Microsoft Docs.* (2018). Xamarin form pages. https://docs.microsoft.com/en-us/xamarin/xamarin-forms/user-interface/controls/pages

Layout. (2018). Microsoft Docs. Retrieved 5 https://docs.microsoft.com/en-us/xamarin/xamarin-forms/user-interface/controls/layouts

Estandares de Programacion. (2013). https://estandaresdeprogramacion.wordpress.com/2013/02/12/guia-de-sintaxis-xaml-

Wikipedia. (2018). Comunicado 4 julio xaml. Retrieved 4 Julio, 2018, https://es.wikipedia.org/wiki/Wikipedia:Comunicado\_4\_julio\_2018http://respag.net/xaml.aspx

Lablogera.net. (2007). La bloguera http://labloguera.net/cesar/2007/05/08/xaml-qu-es-y-como-utilizarlo/