**MANUAL TÉCNICO**

**SOFTWARE CODE QR**

**APLICACIÓN PARA DISPOSITIVOS MÓVILES CON TECNOLOGÍA MICROSOFT EN WINDOWS PHONE QUE PERMITA RECIBIR INFORMACIÓN DE LOS CONTENIDOS DE UN MUSEO A PARTIR DE CÓDIGO E IMÁGENES QR**

**ELABORADO POR**

**JOSÉ DAVID GÓMEZ CRUZ**

**UNIVERSIDAD MANUELA BELTRÁN**

**FACULTAD DE INGENIERÍAS**

**PROGRAMA INGENIERÍA DE SISTEMAS**

**BOGOTÁ D.C.**

**ÍNDICE DE CONTENIDO**

[1. INTRODUCCIÓN 7](#_Toc486091569)

[2. REQUERIMIENTOS TECNICOS 8](#_Toc486091570)

[2.1. Requerimientos de hardware Computadora 8](#_Toc486091571)

[2.2. Requerimientos de hardware de Dispositivo Móvil 8](#_Toc486091572)

[2.3. Requerimientos de software computadora 8](#_Toc486091573)

[2.4. Requerimientos de software Dispositivo Móvil 8](#_Toc486091574)

[3. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO 8](#_Toc486091575)

[3.1. Microsoft Visual Studio 2012 for Windows Phone 8](#_Toc486091576)

[3.2. SDK 8 de Windows Phone 9](#_Toc486091577)

[3.3. Nokia Lumia 9](#_Toc486091578)

[4. INSTALACIÓN DE VISUAL STUDIO EXPRESS FOR WINDOWS PHONE 2012 9](#_Toc486091579)

[5. PLAN DE PUBLICACIONES (RELEASE PLANNING): 11](#_Toc486091580)

[6. ITERACIONES 12](#_Toc486091581)

[6.1. Iteración 1: 12](#_Toc486091582)

[6.1.1. Requerimientos 12](#_Toc486091583)

[6.1.1.1. Funcionales: 12](#_Toc486091584)

[6.1.1.2. No funcionales 13](#_Toc486091585)

[6.1.2. Diagrama de Interacción 13](#_Toc486091586)

[6.1.2.1. Documentación: 13](#_Toc486091587)

[6.1.2.2. Diagrama de actividad por carril de Escanear Código QR 16](#_Toc486091588)

[6.1.2.3. Diagrama de estado de Escanear Código 17](#_Toc486091589)

[6.2. Iteración 2: 19](#_Toc486091590)

[7. MUERTE DEL PROYECTO 26](#_Toc486091591)

[8. CONCLUSIONES 27](#_Toc486091592)

[9. BIBLIOGRAFÍA 28](#_Toc486091593)

**ÍNDICE DE TABLAS**

[Tabla 1: Plan De Publicaciones 12](#_Toc486091594)

[Tabla 2: Iteración 1. 12](#_Toc486091595)

[Tabla 3: Documentación Escanear. 14](#_Toc486091596)

[Tabla 4: Obtener Información. 15](#_Toc486091597)

[Tabla 5: Iteración 2. 19](#_Toc486091598)

**ÍNDICE DE FIGURAS**

[Figura 1: Descarga De SDK DE Windows Phone 9](#_Toc486091599)

[Figura 2: Instalación De Visual Studio 10](#_Toc486091600)

[Figura 3: Interfaz De Visual Studio 11](#_Toc486091601)

[Figura 4: Diagrama Caso De Uso 13](#_Toc486091602)

[Figura 5: Diagrama De Actividad 16](#_Toc486091603)

[Figura 6: Diagrama De Estado 17](#_Toc486091604)

[Figura 7: Diagrama De Secuencia 18](#_Toc486091605)

**OBJETIVOS**

**General:**

Brindar toda la información requerida para poder realizar de forma íntegra y seguro el proceso de creación, instalación y configuración del software a desarrollar.

**Específicos:**

* Representar la funcionalidad técnica de la estructura, diseño y definición del aplicativo.
* Detallar la especificación de los requerimientos de Hardware y Software para la construcción de la aplicación y su posterior funcionamiento.
* Describir las herramientas utilizadas para el diseño y desarrollo del software.

# INTRODUCCIÓN

Este manual va dirigido a aquellas personas que desean saber el funcionamiento técnico del aplicativo “Code QR” con el fin de saber cómo se creó, implemento y puso en ejecución. Así mismo intenta detallar de la manera más completa, la estructura de su funcionamiento y como su ejecución produce un servicio final.

Es importante, aclarar los requerimientos que se necesitan para desarrollar la aplicación y a su vez, para la ejecución desde un dispositivo móvil, más exactamente en Windows Phone.

# 2. REQUERIMIENTOS TECNICOS

## 2.1. Requerimientos de hardware Computadora

* **Procesador:** Quad Core equivalente o Superior
* **Memoria RAM:** 4GB (Preferible 8 o superior)
* **Disco Duro:** 50Gb mínimo

## 2.2. Requerimientos de hardware de Dispositivo Móvil

* **Procesador:** dualQuad Core o superior
* **Memoria RAM:** 1Gb mínimo
* **Almacenamiento:** 8Gb mínimo
* **Pantalla:** 4 pulgadas o superior
* **Cámara:** 2 Megapíxeles o superior

## 2.3. Requerimientos de software computadora

* Permisos de administrador
* **Sistema Operativo:** Windows 8 o superior

## 2.4. Requerimientos de software Dispositivo Móvil

* **Sistema Operativo:** Windows Phone 8 o superior

# 3. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO

## 3.1. Microsoft Visual Studio 2012 for Windows Phone

Visual Studio es un gran conjunto de herramientas, el cual apoya el desarrollo de software bajo las tecnologías de Microsoft donde se crean de manera eficaz y de alto rendimiento, aplicaciones web, servicios y cualquier entorno que soporte dicha plataforma. Para ser más específicos, se puede generar aplicaciones web ASP.NET, servicios bajo protocolos web de XML, aplicaciones que sean en entornos de escritorio y en este caso las aplicaciones móviles. [1]

Lenguajes como Visual Basic, C# y C++ y utilizan este entorno de desarrollo integrado que permite la interacción compartida de herramientas y facilita a su vez la creación de soluciones bajo varios lenguajes y del mismo modo, ayudan a que se simplifique tecnologías de desarrollo. [1]

## 3.2. SDK 8 de Windows Phone

El SDK (Software Development Kit) es un conjunto de herramientas que ayuda a programar aplicaciones para un entorno tecnológico de manera particular a ciertas plataformas; por esto se debe entender que las aplicaciones acá desarrolladas están destinadas a algún sistema operativo o plataforma en específico. [2]

Dentro de lo que se puede encontrar como recurso del SDK se destaca [2]:

* Entorno de desarrollo integrado: Es aquel que permite escribir de manera fácil el código del programa a crear, donde además de eso, generalmente proporciona una interfaz para dos aplicaciones fundamentales:
  + Debugger: Testea el programa
  + Compilador: Traduce el código escrito por el usuario a lenguaje máquina.
* Interfaz de programación de aplicaciones: es un conjunto de funciones, rutinas, estructuras de datos, clases y variables que manipulan el mecanismo de la plataforma.
* Emulador de entorno: permite visualizar de una manera virtualizada a la real el cómo lo vería el usuario final.

## 3.3. Nokia Lumia

Nokia es uno de los grandes comercializadores de celulares de baja y alta gama en el Mercado mundial, también se centra en el sector de la tecnología para ampliar sus productos. Uno de sus productos es el Nokia lumia el cual es un dispositivo móvil “Smartphone” que viene con cámara integrada, memoria RAM, procesador y Tecnología Touch para soportar los procesos que requiere las aplicaciones que usa el usuario. [3]

# 4. INSTALACIÓN DE VISUAL STUDIO EXPRESS FOR WINDOWS PHONE 2012

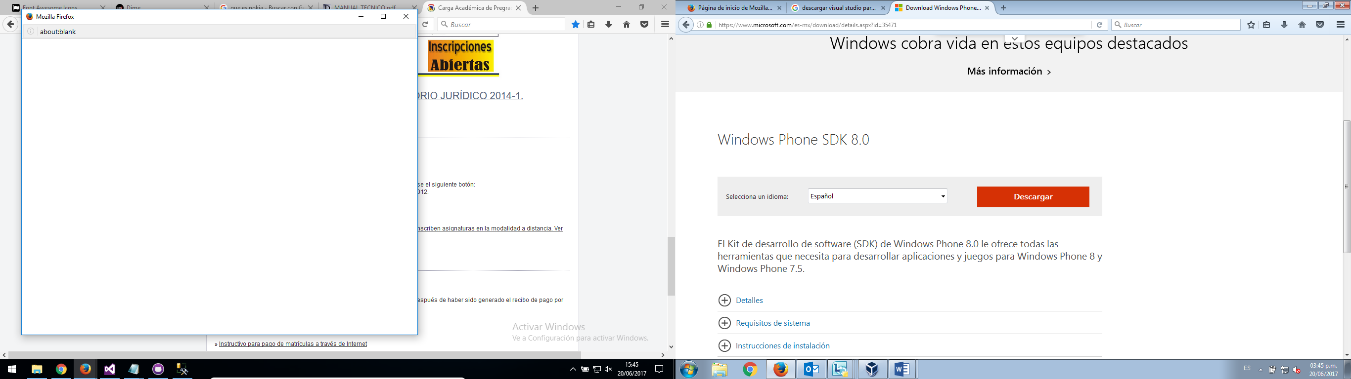
**Se debe descarga directamente desde las páginas de Microsoft la versión de Visual Studio para Windows Phone 2012 o superior, y de ese ejecutable realizar la instalación que se realizara de forma automática.

Figura 1: Descarga De SDK DE Windows Phone.

La descarga se inicia y se intala todo de forma automatica; al finalizar reinicia el equipo para que la computadora se adapte a los cambios:

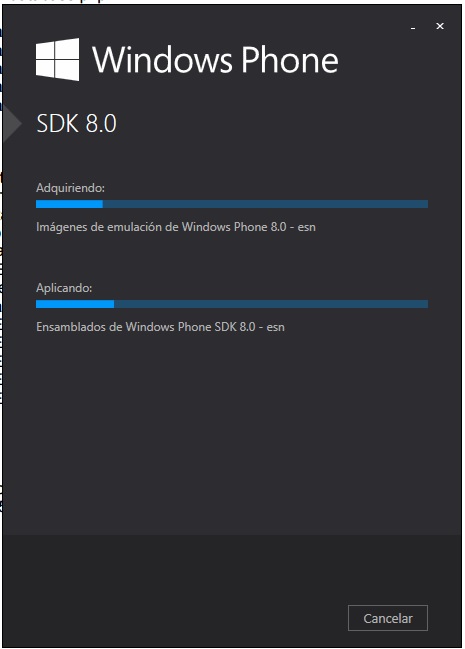


Figura 2: Instalación De Visual Studio.

Una vez instalado, se procede iniciar la interfaz donde se programara dicho código

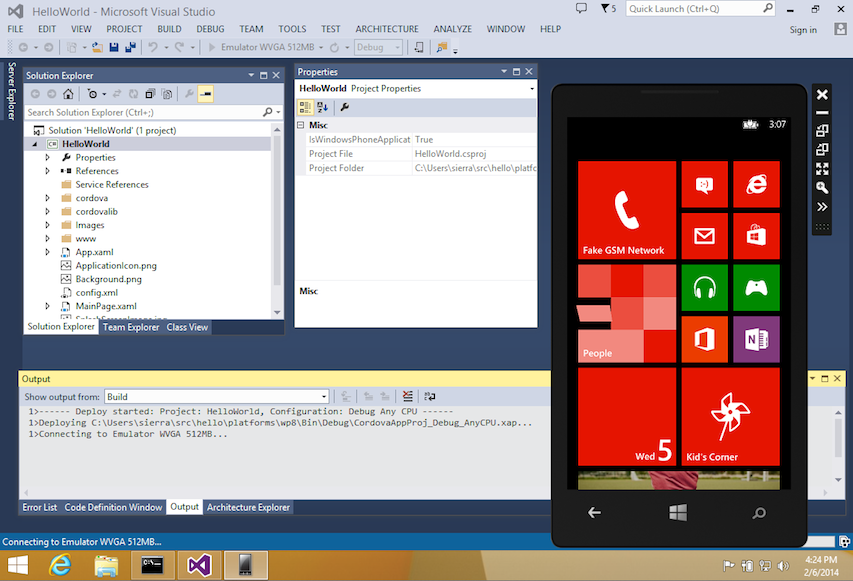


Figura 3: Interfaz De Visual Studio.

# 5. PLAN DE PUBLICACIONES (RELEASE PLANNING):

Para iniciar el desarrollo, se realizó el siguiente cuadro con las historias de usuario a realizar para que se tenga una visual clara de lo que se va a realizar dentro de la metodología aplicada:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Numero** | **Historia de usuario** | **Fecha de publicación** | **Tiempo de desarrollo (estimado)** | **Número de personas** | **Pruebas** |
| H1 | Realizar la arquitectura en la que va a funcionar el lector de códigos QR | 20/08/2016 | 1 mes | 1 | No aplica |
| H2 | Desarrollar la aplicación funcional para que lea e interprete códigos QR | 30/09/2016 | 1 mes | 1 | Funcionales |
| H3 |  |  |  |  |  |

Tabla 1: Plan De Publicaciones.

# 6. ITERACIONES

Para las iteraciones se tendrá en cuenta nada más que dos (2), ya que el desarrollo de dicha aplicación es bastante sencillo y las pruebas se pueden generar de manera fácil para terminar el objetivo final de construir una aplicación para Windows Phone.

## 6.1. Iteración 1:

|  |  |  |  |
| --- | --- | --- | --- |
| **Descripción** | **Historias/Tareas** | **Resultado** | **Fecha de entrega** |
| Historias realizadas | H1 | Finalizada | 30/09/2016 |
| Tareas de la historia H1 | Diseñar requerimientos | Realizadas | 30/09/2016 |
| Tareas de la historia H1 | Diseñar diagramas que muestren el funcionamiento de la aplicación | Realizadas | 30/09/2016 |
|  |  |  |  |

Tabla 2: Iteración 1.

## 6.1.1. Requerimientos

### 6.1.1.1. Funcionales:

* **RF1.** El usuario Ingresa en la aplicación.
* **RF2.** El usuario Escanea el código.
* **RF3.** El usuario consulta la información obtenida del código.

### 6.1.1.2. No funcionales

* **RNF1** La aplicación se desarrollara para arquitectura Microsoft Windows Phone en el lenguaje de programación XAML.
* **RNF2** La interfaz de usuario será integrada o unida con la cámara del dispositivo móvil.
* **RNF3** La aplicación necesitara uso de internet con datos móviles o WI-FI.

## 6.1.2. Diagrama de Interacción

****

Figura 4: Diagrama Caso De Uso.

### 6.1.2.1. Documentación:

* 1. 001 – Escanear

|  |  |  |
| --- | --- | --- |
| **Nombre** | Escanear. Cód.: 001 | |
| **Fecha y Versión** | 12/05/2015 – V\_001 | |
| **Requerimientos** | RF1. RF2. RNF1. RNF2 | |
| **Actor(es)** | Usuario | |
| **Descripción** | Esta opción permite Escanear un código o imagen QR establecida en el museo para consultar mayor información. | |
| **Precondiciones** | Instalar aplicación. | |
| **Flujo Normal** | **Usuario**  1 Ingresa a la aplicación.  3 Escoge la opción de escanear un código nuevo | **Aplicación**  2 Permite ingreso del usuario y muestra las opciones a realizar  4 Valida opción.  5 Escanea código en el medio o museo. |
| **Flujo Alternativo** | **Usuario**  1  3  6 Vuelve a la elección de opciones | **Aplicación**  2  4 Valida opción.  5 Error en la cámara. |
| **Post Condiciones** | Mira información del código o consultar nuevo código otra vez. | |

Tabla 3: Documentación Escanear.

* 1. 002 – Obtener información

|  |  |  |
| --- | --- | --- |
| **Nombre** | Obtener información. Cód.: 002 | |
| **Fecha y Versión** | 12/05/2015 – V\_001 | |
| **Requerimientos** | RF1. RF2. RF3. RNF3 | |
| **Actor(es)** | Usuario | |
| **Descripción** | Esta opción permite obtener información de un código o imagen QR establecida en el museo. | |
| **Precondiciones** | Escanear Código. | |
| **Flujo Normal** | **Usuario**  1 Ingresa a la aplicación.  3 Escoge la opción de escanear un código nuevo | **Aplicación**  2 Permite ingreso del usuario y muestra las opciones a realizar  4 Valida opción.  5 Escanea código en el medio o museo.   1. Recupera y muestra información del código |
| **Flujo Alternativo** | **Usuario**  1  3  6 Vuelve a la elección de opciones | **Aplicación**  2  4 Valida opción.  5 Error en la cámara.  7. Error en la recuperación de información. |
| **Post Condiciones** | Mira información del código o consultar nuevo código otra vez. | |

Tabla 4: Obtener Información.

### 6.1.2.2. Diagrama de actividad por carril de Escanear Código QR



Figura 5: Diagrama De Actividad.

### 6.1.2.3. Diagrama de estado de Escanear Código



Figura 6: Diagrama De Estado.

**6.1.2.4. Diagrama de secuencia de Escáner código**



Figura 7: Diagrama De Secuencia.

## 6.2. Iteración 2:

|  |  |  |  |
| --- | --- | --- | --- |
| **Descripción** | **Historias/Tareas** | **Resultado** | **Fecha de entrega** |
| Historias realizadas | H2 | Finalizada | 05/11/2016 |
| Tareas de la historia H1 | Codificar en lenguaje C# para Windows Phone | Realizadas | 05/11/2016 |
| Tareas de la historia H1 | Pruebas funcionales del código | Realizadas | 05/11/2016 |
|  |  |  |  |

Tabla 5: Iteración 2.

**Clase PhotoCameraLuminanceSource**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using ZXing;

namespace Tesis

{

public class PhotoCameraLuminanceSource: LuminanceSource

{

public byte[] PreviewBuferY { get; private set; }

public PhotoCameraLuminanceSource(int Width, int Heigh): base(Width, Heigh)

{

PreviewBuferY = new byte[Width \* Heigh];

}

public override byte[] Matrix

{

get { return (byte[])(Array)PreviewBuferY; }

}

public override byte[] getRow(int y, byte[] row)

{

if (row == null || row.Length < Width)

{

row = new byte[Width];

}

for (int i = 0; i < Height; i++)

{

row[i] = (byte)PreviewBuferY[i \* Width + y];

}

return row;

}

}

}

**MainPage.xaml**

<phone:PhoneApplicationPage

x:Class="Tesis.MainPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"

xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d"

FontFamily="{StaticResource PhoneFontFamilyNormal}"

FontSize="{StaticResource PhoneFontSizeNormal}"

Foreground="{StaticResource PhoneForegroundBrush}"

SupportedOrientations="Portrait" Orientation="Portrait"

shell:SystemTray.IsVisible="True">

<!--LayoutRoot es la cuadrícula raíz donde se coloca todo el contenido de la página-->

<Grid x:Name="LayoutRoot" Background="Transparent">

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<!-- NOTA PARA LA TRADUCCIÓN:

Para traducir las cadenas mostradas, copia sus valores a las claves con el nombre

correspondiente en el archivo de recursos del idioma neutro (AppResources.resx) de la aplicación y

reemplaza el valor de texto codificado de forma rígida entre las comillas de los atributos

con la cláusula de enlace cuya ruta de acceso apunte a ese nombre de cadena.

Por ejemplo:

Text="{Binding Path=LocalizedResources.ApplicationTitle, Source={StaticResource LocalizedStrings}}"

Este enlace apunta al recurso de cadena de la plantilla denominado "ApplicationTitle".

<!--TitlePanel contiene el nombre de la aplicación y el título de la página-->

<StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="12,17,0,28">

<TextBlock Text="Universidad Manuela Beltran" Style="{StaticResource PhoneTextNormalStyle}" Margin="12,0"/>

<TextBlock x:Name="te" Text="Lector QR" Margin="9,-7,0,0" Style="{StaticResource PhoneTextTitle1Style}"/>

</StackPanel>

<!--ContentPanel. Colocar aquí el contenido adicional-->

<Grid x:Name="ContentPanel" Background="CornflowerBlue" Grid.Row="1" Margin="12,0,12,0">

<Button x:Name="ret" Content="Escanear" Background="Black" Margin="30,44,122,459" Click="Button\_Click"/>

</Grid>

<!--Quitar la marca de comentarios para ver una cuadrícula de alineación que

ayuda a comprobar que los controles están alineados en los límites normales. La imagen tiene un margen superior de -32px para

tener en cuenta la bandeja del sistema. Establécelo en 0 (o quite el margen)

si la bandeja del sistema está oculta.

Antes de enviarla, quita este código XAML y la propia imagen.-->

<!--<Image Source="/Assets/AlignmentGrid.png" VerticalAlignment="Top" Height="800" Width="480" Margin="0,-32,0,0" Grid.Row="0" Grid.RowSpan="2" IsHitTestVisible="False" />-->

</Grid>

</phone:PhoneApplicationPage>

**MainPage.xaml.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Navigation;

using Microsoft.Phone.Controls;

using Microsoft.Phone.Shell;

using Tesis.Resources;

namespace Tesis

{

public partial class MainPage : PhoneApplicationPage

{

// Constructor

public MainPage()

{

InitializeComponent();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new Uri("/Escaner.xaml", UriKind.Relative));

}

}

}

**Escaner.xaml**

<phone:PhoneApplicationPage

x:Class="Tesis.Escaner"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"

xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

FontFamily="{StaticResource PhoneFontFamilyNormal}"

FontSize="{StaticResource PhoneFontSizeNormal}"

Foreground="{StaticResource PhoneForegroundBrush}"

SupportedOrientations="Portrait" Orientation="Portrait"

mc:Ignorable="d"

shell:SystemTray.IsVisible="True">

<!--LayoutRoot es la cuadrícula raíz donde se coloca todo el contenido de la página-->

<Grid x:Name="LayoutRoot" Background="Transparent">

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<!--TitlePanel contiene el nombre de la aplicación y el título de la página-->

<StackPanel Grid.Row="0" Margin="12,17,0,28"/>

<!--ContentPanel. Colocar aquí el contenido adicional-->

<Grid x:Name="ContentPanel" Background="Red" Grid.Row="1" Margin="10,0,12,10">

<Rectangle Margin="0,0,0,62">

<Rectangle.Fill>

<VideoBrush x:Name="videoBrush">

<VideoBrush.RelativeTransform>

<CompositeTransform x:Name="compositeTransform" CenterX="0.5" CenterY="0.5">

</CompositeTransform>

</VideoBrush.RelativeTransform>

</VideoBrush>

</Rectangle.Fill>

</Rectangle>

<ListBox Margin="10,10,10,62" x:Name="\_matchetList" FontSize="30" FontWeight="ExtraBold"></ListBox>

<Grid x:Name="ContentPanel2" Grid.Row="1" Margin="10,656,12,10"/>

</Grid>

</Grid>

</phone:PhoneApplicationPage>

**Escaner.xaml.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Navigation;

using Microsoft.Phone.Controls;

using Microsoft.Phone.Shell;

using Tesis.Resources;

using Microsoft.Devices;

using System.Windows.Threading;

using ZXing;

using ZXing.QrCode;

using ZXing.Common;

namespace Tesis

{

public partial class Escaner : PhoneApplicationPage

{

private PhotoCamera photoCamera;

private readonly DispatcherTimer dispatcherTimer = new DispatcherTimer();

private QRCodeReader codeReader;

private HyperlinkButton Link = new HyperlinkButton();

public Escaner()

{

InitializeComponent();

}

protected override void OnNavigatedTo(NavigationEventArgs e)

{

photoCamera = new PhotoCamera(CameraType.Primary);

photoCamera.Initialized += (s, a) =>

{

codeReader = new QRCodeReader();

Dispatcher.BeginInvoke(() => { compositeTransform.Rotation = photoCamera.Orientation; });

};

videoBrush.SetSource(photoCamera);

dispatcherTimer.Interval = TimeSpan.FromMilliseconds(250);

dispatcherTimer.Tick += (o, arg) => ScanPreviewBuffer();

dispatcherTimer.Start();

base.OnNavigatedTo(e);

}

protected override void OnNavigatedFrom(NavigationEventArgs e)

{

dispatcherTimer.Stop();

base.OnNavigatedFrom(e);

}

private void ScanPreviewBuffer()

{

try

{

int width = Convert.ToInt32(photoCamera.PreviewResolution.Width);

int height = Convert.ToInt32(photoCamera.PreviewResolution.Height);

PhotoCameraLuminanceSource \_luminance = new PhotoCameraLuminanceSource(width, height);

photoCamera.GetPreviewBufferY(\_luminance.PreviewBuferY);

var binBitmap = new BinaryBitmap(new HybridBinarizer(\_luminance));

var result = codeReader.decode(binBitmap);

if (result != null)

{

MessageBox.Show("Codigo Atrapado");

Link.Content = "Abrir Link";

Link.NavigateUri = new Uri(""+result);

Link.TargetName = "\_blank";

ContentPanel2.Children.Add(Link);

//Link.Click += new RoutedEventHandler(URLink\_Click);

}

}

catch { }

}

}

# 7. MUERTE DEL PROYECTO

Una vez no existen más historias de usuario para realizar entonces se hace entrega final del desarrollo y se da muerte al proyecto con una reunión final entregando todos los detalles de lo que se hizo y este manual técnico.

# 8. CONCLUSIONES

* Se explicó de manera clara y simple la forma de instalación de las herramientas para la construcción del software y su posterior desarrollo sobre las mismas, dando como resultado un manual con el paso a paso de lo que se realizó.
* Se desarrolló por medio de una metodología la aplicación en plataforma Windows Phone la cual permite escanear un código y retornar información para darle al usuario una experiencia más enriquecedora acerca de los elementos de un museo.

# 9. BIBLIOGRAFÍA

[1] ¿Qué es y para qué sirve Visual Studio 2017?. Msn Noticias. 2017. [En línea]: https://www.msn.com/es-cl/noticias/microsoftstore/%C2%BFqu%C3%A9-es-y-para-qu%C3%A9-sirve-visual-studio-2017/ar-AAnLZL9

[2] ¿Qué es un kit de desarrollo de software (SDK)?. 28 de Febrero de 2013. [En línea]: http://www.4rsoluciones.com/blog/que-es-un-kit-de-desarrollo-de-software-sdk-2/

[3] Que es Nokia. Instituto Español de Marketing Digital. [En línea]: https://iiemd.com/nokia/que-es-nokia