

“AÑO DE LA DIVERSIFICACION PRODUCTIVA Y DEL FORTALECIMIENTO DE LA EDUCACION”



**UNIVERSIDAD NACIONAL**

**JOSE MARIA ARGUEDAS**

**FACULTAD DE INGENIERÍA**

**Escuela Profesional Ingeniería de Sistemas**

APLICACIÓN PARA WINDOWS PHONE

«JUEGO DE AJEDREZ».

**Proyecto :**

**Curso :** Ingeniería de Software II

**Docente :** Ing. Iván Soria Solís

**Proyectista :** ROMANI FLORES, Rogelio

Andahuaylas, 30 de Julio del 2015

Tabla de contenido

[1. INTRODUCCION 3](#_Toc425957472)

[2. HERRAMIENTAS DE DESARROLLO 3](#_Toc425957473)

[2.1. SDK (Software Development Kit) 3](#_Toc425957474)

[2.2. VISUAL STUDIO PARA WINDOWS PHONE 3](#_Toc425957475)

[2.3. BLEND PARA VISUAL STUDIO 2013 3](#_Toc425957476)

[2.4. WINDOWS PHONE EMULADOR 4](#_Toc425957477)

[3. ANALISIS 4](#_Toc425957478)

[3.1. IDENTIFICACION Y VALIDACION PREVIA DE REQUERIMIENTOS 4](#_Toc425957479)

[3.1.1. REQUERIMIENTOS FUNCIONALES 4](#_Toc425957480)

[3.1.2. REQUERIMIENTOS NO FUNCIONALES 5](#_Toc425957481)

[3.2. OBJETIVOS DE LA APLICACIÓN 6](#_Toc425957482)

[4. DISEÑO 6](#_Toc425957483)

[4.1. ARQUITECTURA DE LA APLICACIÓN 6](#_Toc425957484)

[4.1.1. CASOS DE USO 6](#_Toc425957485)

[4.1.2. DIAGRAMA DE CLASES 7](#_Toc425957486)

[5. IMPLEMENTACION 8](#_Toc425957487)

[5.1. MainPage.xaml 8](#_Toc425957488)

[5.2. CLASE Tablero.cs 9](#_Toc425957489)

[5.3. CLASE Jugador.cs 10](#_Toc425957490)

[5.4. CLASE Peon.cs 11](#_Toc425957491)

[6. CONCLUSIONES 12](#_Toc425957492)

[7. BIBLIOGRFIA 12](#_Toc425957493)

# INTRODUCCION

# HERRAMIENTAS DE DESARROLLO

La cuestión de Windows Phone, de ser la más reciente plataforma operativa para dispositivos móviles de Microsoft, y dado su retraso en optar por entrar a competir en un negocio donde gran parte del mercado móvil está custodiado por Apple y Google; Apura al mismo Microsoft a brindar beneficios de acceso al paquete de herramientas de desarrollo de manera gratuita, para aquellos interesados en optar por el desarrollo de App para esta plataforma.

## SDK (Software Development Kit)

El Kit de desarrollo de software SDK para Windows Phone, es un conjunto de herramientas de desarrollo de software que permite al desarrollador crear aplicaciones para un sistema en concreto. El SDK de Windows Phone proporciona las herramientas que son necesarias para desarrollar Apps y juegos para Windows Phone 8 y Windows Phone 7.x. El paquete completo contiene las siguientes herramientas:

* Visual Studio Express 2012-2010 edition para Windows Phone.
* Blend para Visual Studio 2012-2010.
* Windows Phone Developer Registration tool.
* Windows Phone Connect tool.
* Emulators de Windows Phone 7.1 y 8.0.
* Windows Phone Application Analysis tool.
* Simulation Dashboard para Windows Phone.

## VISUAL STUDIO PARA WINDOWS PHONE

Visual Studio para Windows Phone es la herramienta que proporciona los mecanismos para ingresar código y construir las App, proporciona exclusivamente características requeridas por el desarrollo de Windows Phone. El SDK de Windows Phone puede adaptarse a cualquier versión de Visual Studio 2012. (Pahkala, 2012).

## BLEND PARA VISUAL STUDIO 2013

Blend para Visual Studio es un editor de XAML por excelencia que normalmente se usa para diseñar Interfaces gráficas para aplicaciones en Silverlight. La clave principal de Blend es permitir a los diseñadores de UI crear interfaces, mientras que los desarrolladores escriben código de Code-Behind. Blend ofrece un flujo grafico muy dinámico, con posicionamiento relevante de elementos sobre layouts, especificaciones de valores mínimos y máximos de Altura y anchura en cada uno de los elementos, y manejadores de enlaces de datos que le dan al diseñador descripción exacta y en tiempo real de elementos XAML enlazados con datos. (Pahkala, 2012).

## WINDOWS PHONE EMULADOR

El emulador de Windows Phone es una aplicación de escritorio que emula un dispositivo real que tenga como sistema operativo Windows Phone. El emulador propicia un ambiente virtualizado en la cual los desarrolladores tienen la posibilidad de probar sus aplicaciones sin la necesidad de usar un dispositivo real.

El emulador es totalmente diseñado para ofrecer un rendimiento comparable al de un dispositivo real. Microsoft recomienda a los desarrolladores, probar sus aplicaciones en un dispositivo real antes de publicar la aplicación en la tienda de aplicaciones. (msdn Microsoft).

# ANALISIS

A continuación, se lleva a cabo la generación y la recopilación de la información sobre los objetivos, requerimientos, restricciones, intereses, aspectos de calidad y motivación arquitecturales de la aplicación del caso de estudio, cuya estructura de documentación está basada en gran medida por el estándar internacional ISO/IEC/IEEEE 29148-2011[11].

## IDENTIFICACION Y VALIDACION PREVIA DE REQUERIMIENTOS

Se toma un conjunto de requerimientos previamente identificados, siendo la naturaleza, objetivos y propósito general del proyecto.

### REQUERIMIENTOS FUNCIONALES

* **La aplicación inicializa el tablero y las piezas**

Al momento de abrir la aplicación en los móviles el software debe cargar inmediatamente el tablero y las piezas.

* **Poder interpretar las jugadas realizadas.**

Cada pieza del ajedrez tiene movimientos distintos. La aplicación debe reconocer esos movimientos.

* **Distinguir entre una jugada válida y una incorrecta.**

Reconocer las jugadas correctas e incorrectas atreves de mensajes de alerta.

* **Reconocer la finalización de un juego (Jaque Mate).**

Un partida de juego de ajedrez tiene reglas de juego como es el caso de jaque mate el software debe reconocer la jugada y debe finalizar.

* **Identificar cada pieza de ajedrez y su posición.**

La aplicación debe identificar cada posición y movimiento de cada pieza el tablero.

* **Validación de movimientos**

Se debe permitir la validación del movimiento de cada una de las fichas según su estado en el tablero, esta debe ejecutarse previamente cada que un jugador realice una jugada. La validez de los movimientos debe ser evaluada de acuerdo a las reglas y no normas del juego.

* **Turno para cada jugador**

Los jugadores no pueden ejecutar órdenes de movimientos al mismo tiempo, a cada jugador le asigna un turno para realizar su jugada. El cambio de turno se establece cada que ocurre un intercambio de tiempo en el temporizador doble del reloj.

### REQUERIMIENTOS NO FUNCIONALES

* **Portabilidad**

La aplicación será diseñada para ejecutarse sobre el sistema operativo Windows 8.1 para móviles como Windows phone utilizando el lenguaje C# en el visual studio 2013.

* **Integridad**

La aplicación deberá presentar una alta integridad y facilidad de instalación.

* **Normas técnicas y reglas del juego**

Este requerimiento es importante para saber si al momento de jugar se realiza las jugadas correctamente, según sus técnicas y reglas.

* **Apariencia**

El juego es una interfaz realmente de uso intuitiva, sencilla y gráfica. Debe usar los colores adecuados. Como se trata de un juego de ajedrez.

* **Modularidad**

La aplicación del juego deberá presentar un diseño modular para permitir las posteriores evoluciones del juego incorporándole nuevas funcionalidades o añadiendo nuevas características al código existente. Dependiendo del tipo de modificación que se desee realizar, podría consistir simplemente en sustituir un módulo por otro sin que los demás se vieran afectados.

* **Fácil comprensión del código fuente**

La aplicación del juego deberá tener un código fuente fácilmente comprensible por lo que debe ir acompañado de comentarios que aclaren la funcionalidad del mismo, sobre todo en las líneas que pueden resultar más confusas.

## OBJETIVOS DE LA APLICACIÓN

Los objetivos la aplicación que describen el conjunto de intereses de los involucrados, se presentan a continuación:

* Permitir que el tablero se comunique con el usuario haciendo uso de sus sentidos del tacto y el audio, apoyando su discapacidad visual a partir del potenciamiento de sus demás capacidades sensoriales.
* Informar de manera gráfica y/o a través de mensajes de audio a los jugadores sobre el estado actual de la partida, como: posición de las fichas, tiempo en el reloj, movimientos del contrincante, jaques, movimientos inválidos y demás alertas que generalmente están asistida por un tercero.
* Reducir la complejidad con la que el jugador interactúa con el tablero de juego, determinando un conjunto mínimo de órdenes de fácil acceso que le permitan realizar la totalidad de las acciones permitidas.
* Se debe poder configurar la parametrización de las variables de juego, de diversas formas para permitir el uso del sistema en diversos contextos (profesional, casual, entrenamiento y etc.)
* Permitir a dos personas videntes y/o invidentes, jugar una partida de ajedrez con normalidad y en igualdad de oportunidades.
* Brindar a dos usuarios las garantías para llevar a cabo una partida de ajedrez bajo las normas internacionales FIDE.

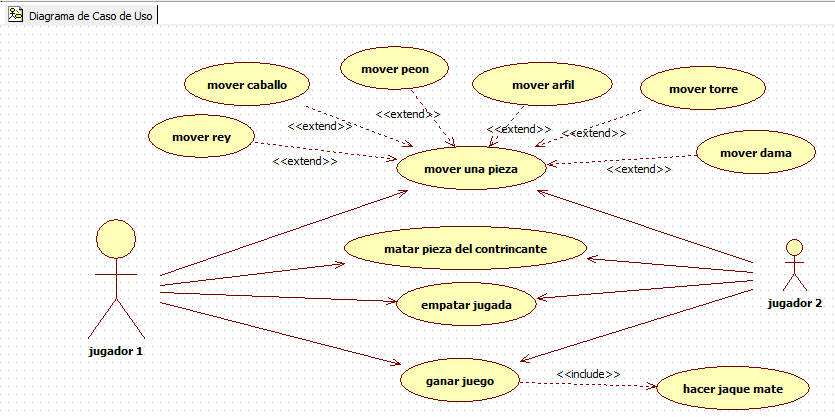
# DISEÑO

## ARQUITECTURA DE LA APLICACIÓN

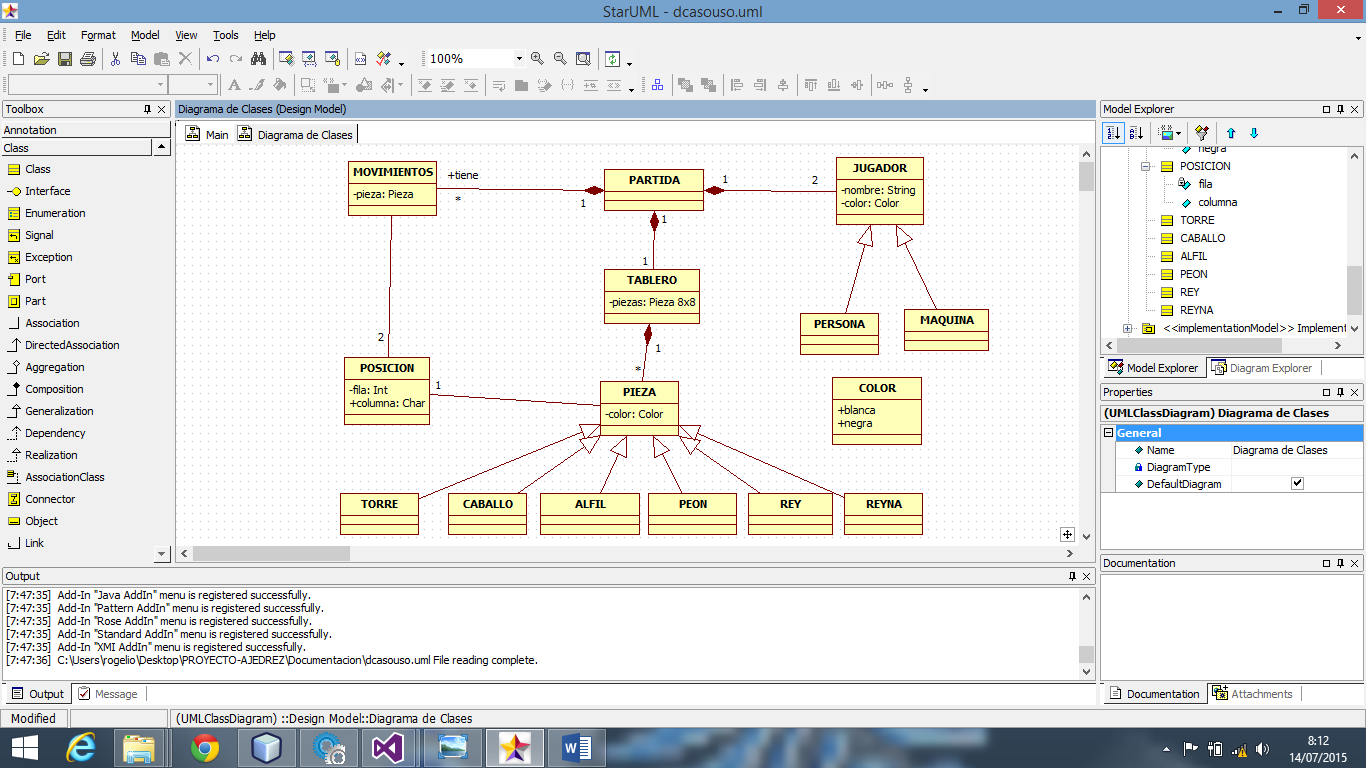
La arquitectura que utilizaremos en nuestra aplicación es el “Modelo de Orientado a Objetos”. A continuación detallaremos el proceso de diseño de la arquitectura de la aplicación.

### CASOS DE USO

Cada uno de los requerimientos funcionales mencionados en la sección 3.1.1 se asocian a un mismo sistema, el cual el usuario interactúa de manera dinámica y amigable, un diagrama de casos de uso es el mejor ejemplo para ilustrar dicha interacción. La siguiente imagen instruye el diagrama de casos de uso de la aplicación a realizar.



### DIAGRAMA DE CLASES



# IMPLEMENTACION

En este apartado intentaremos plasmar los detalles más importantes de la implementación de la aplicación. Resaltando la funcionalidad de los atributos y métodos de las clases más importantes podremos hacernos una idea general sobre el funcionamiento interno de la aplicación, sin la necesidad de explicar el código fuente en su totalidad.

En los siguientes apartados detallaremos los aspectos más relevantes de cada clase, así como sus atributos y métodos, teniendo en cuenta que todo el código que haya sido generado por el diseñador de Windows Phone de Visual Studio no será comentado, por carecer de interés.

## MainPage.xaml

Este en archivo que contiene el diseño de la interfaz gráfica de la aplicación. Este código se genera automáticamente atraves de inserción de las herramientas.

<phone:PhoneApplicationPage

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"

xmlns:i="clr-namespace:System.Windows.Interactivity;assembly=System.Windows.Interactivity" xmlns:el="clr-namespace:Microsoft.Expression.Interactivity.Layout;assembly=Microsoft.Expression.Interactions"

x:Class="Ajedrez.MainPage"

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="LawnGreen" RenderTransformOrigin="0.487,0.492">

<Grid.RowDefinitions>

<RowDefinition Height="Auto" MinHeight="752"/>

<RowDefinition/>

</Grid.RowDefinitions>

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

<StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="10,0,0,440">

<TextBlock FontSize="46" Foreground="Black" x:Name="txtB" Text="AJEDREZ- UNAJMA"/>

</StackPanel>

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

<Grid x:Name="ContentPanel" Margin="20,317,20,0">

<Button FontSize="46" Foreground="Red" x:Name="ayuda" Content="Ayuda" Height="100" Margin="0,25,0,310" Background="Aqua" Click="ayuda\_Click"/>

<Button FontSize="46" Foreground="Red" x:Name="partida" Content="Crear Partida" Height="100" Margin="0,125,0,210" Background="Aqua" Click="partida\_Click"/>

<Button FontSize="46" Foreground="Red" x:Name="salir" Content="Salir" Height="100" Margin="0,225,0,110" Background="Aqua" Click="salir\_Click"/>

</Grid>

</Grid>

</phone:PhoneApplicationPage>

## CLASE Tablero.cs

Contiene atributos estáticos constantes que definen las posiciones de las piezas dentro de la estructura que las guarda.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Ajedrez.Piezas;

namespace Ajedrez.Ajedrez

{

public class Tablero:ITablero

{

private Pieza pieza;

private Posicion posicion = new Posicion(0,0);

private Peon[] peon = new Peon[16];

private Alfil[] alfil = new Alfil[4];

private Torre[] torre = new Torre[4];

private Caballo[] caballo = new Caballo[4];

private Rey[] rey = new Rey[2];

private Reyna[] reyna = new Reyna[2];

public Tablero() {

// Declaramos los peones

for(int i=0; i<8; i++){

peon[i] = new Peon(new Posicion(1,i), Ajedrez.Color.negra);

peon[i+8] = new Peon(new Posicion(6,i), Ajedrez.Color.blanca);

}

// Declaramos los caballos

caballo[0] = new Caballo(new Posicion(0,1), Ajedrez.Color.negra);

caballo[1] = new Caballo(new Posicion(0,6), Ajedrez.Color.negra);

caballo[2] = new Caballo(new Posicion(7,1), Ajedrez.Color.blanca);

caballo[3] = new Caballo(new Posicion(7,6), Ajedrez.Color.blanca);

// Declaramos los alfiles

alfil[0] = new Alfil(new Posicion(0,2), Ajedrez.Color.negra);

alfil[1] = new Alfil(new Posicion(0,5), Ajedrez.Color.negra);

alfil[2] = new Alfil(new Posicion(7,2), Ajedrez.Color.blanca);

alfil[3] = new Alfil(new Posicion(7,5), Ajedrez.Color.blanca);

// Declaramos las torres

torre[0] = new Torre(new Posicion(0,0), Ajedrez.Color.negra);

torre[1] = new Torre(new Posicion(0,7), Ajedrez.Color.negra);

torre[2] = new Torre(new Posicion(7,0), Ajedrez.Color.blanca);

torre[3] = new Torre(new Posicion(7,7), Ajedrez.Color.blanca);

// Declaramos las reyes

rey[0] = new Rey(new Posicion(0,4), Ajedrez.Color.negra);

rey[1] = new Rey(new Posicion(7,4), Ajedrez.Color.blanca);

// Declaramos los reynas

reyna[0] = new Reyna(new Posicion(0,3), Ajedrez.Color.negra);

reyna[1] = new Reyna(new Posicion(7,3), Ajedrez.Color.blanca);

}

## CLASE Jugador.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Ajedrez.Ajedrez

{

public class Jugador:IJugador

{

private String nombre;

private Color color;

private Movimientos movimiento;

public Jugador(String nombre, Color color){

this.nombre = nombre;

this.color = color;

}

public void preguntarNombre() {

}

public void preguntarColorPiezas() {

}

public Movimientos pedirMovimiento()

{

// Devuelvo un movimiento dumy para poder compilar

return movimiento;

}

public Boolean puedeMover()

{

// Devuelve verdadero para poder compilar

return true;

}

public void anotarPiezaPerdida(Pieza pieza){

}

}

}

## CLASE Peon.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Ajedrez.Ajedrez;

namespace Ajedrez.Piezas

{

public class Peon:Pieza

{

Movimientos mov = new Movimientos();

//Posicion pos;

int fila\_actual;

int columna\_actual;

char[] filas = {'0','1','2','3','4','5','6','7','8'};

char[] columnas = {'a','b','c','d','e','f','g','h'};

Posicion[] result = new Posicion[64];

public ArrayList resultado;

Color color;

public Peon(Posicion pos, Color col)

:base(pos, col)

{

this.fila\_actual = pos.getFila();

this.columna\_actual = pos.getColumna();

this.color = col;

}

public void MostrarTodas()

{

for (Posicion palabra ; resultado) {

System.Windows.MessageBox.Show(palabra +" ");

}

System.Windows.MessageBox.Show("");

resultado.clear();

}

# CONCLUSIONES

En conclusión puedo decir que el desarrollo parcial de este software fue una experiencia nueva para mi formación profesional. Porque al investigar nuevas teorías y practicas sobre el desarrollo móvil enriquece el conocimiento.

El desarrollo progresivo del software fue en versiones, este control se hizo gracias a Git. Este es una genial herramienta que permite el control de versiones distribuidas.

# BIBLIOGRFIA