

**PROTOTIPO DE SISTEMA DE CONTROL DOMÓTICO POR MEDIO DE  
DISPOSITIVOS ANDROID, UTILIZANDO PROCESSING**

**VALENTINA AGUIRRE MUÑOZ**

**UNIVERSIDAD CATÓLICA DE MANIZALES  
FACULTAD DE INGENIERÍA Y ARQUITECTURA  
INGENIERÍA TELEMÁTICA  
MANIZALES  
2013**

**PROTOTIPO DE SISTEMA DE CONTROL DOMÓTICO POR MEDIO DE  
DISPOSITIVOS ANDROID, UTILIZANDO PROCESSING**

**VALENTINA AGUIRRE MUÑOZ**

**TRABAJO DE GRADO**

**JAIME ALBERTO SEPÚLVEDA GÓMEZ  
DIRECTOR PROGRAMA INGENIERÍA TELEMÁTICA  
ING. ELECTRÓNICO – ESPECIALISTA EN TELECOMUNICACIONES**

**UNIVERSIDAD CATÓLICA DE MANIZALES  
FACULTAD DE INGENIERÍA Y ARQUITECTURA  
INGENIERÍA TELEMÁTICA  
MANIZALES  
2013**

## **DEDICATORIA**

Este trabajo lo quiero dedicar a Dios por llenar mi vida de cosas maravillosas, permitiéndome día a día crecer intelectual, emocional y espiritualmente; a mis padres que han sido mi apoyo y son quienes me han formado en la persona que soy; y a mi sobrina, que es la persona a la cual quiero dar ejemplo, mostrándole que todos los esfuerzos valen la pena, y que todas las metas que se proponga, con esfuerzo y dedicación llegara a ellas.

## **AGRADECIMIENTO**

Quiero expresarle mi agradecimiento al director de programa de Ingeniería Telemática, Jaime Alberto Sepúlveda Gómez, por asesorarme no solo en este proyecto sino en las diferentes etapas de la carrera, y a los diferentes docentes que me ayudaron a culminar con éxito este proceso

## RESUMEN

Con este proyecto, se pretende abarcar de manera teórica y práctica el control domótico de luces y alarmas por medio de una aplicación desarrollada en *Processing* ejecutada en dispositivos con sistema operativo *Android*.

El desarrollo teórico tendrá una revisión bibliográfica, que permitirá establecer las ventajas y el estado actual en que se encuentra el desarrollo de proyectos de control domótico, estableciendo las ventajas y el grado de innovación que posee este proyecto. También se establecerá y se justificará por qué no se usó *Wakanda* como plataforma de desarrollo, sino que se optó por utilizar *Processing*.

En el desarrollo práctico, se mostrará la aplicación en las diferentes plataformas de desarrollo, *Wakanda*, *Processing* y *Android*, donde cada plataforma contará con un menú, el cual tendrá como opciones los diferentes espacios con los que cuenta la casa, al igual que las diferentes operaciones que se pueden realizar en dicho espacio.

## CONTENIDO

	pág.
INTRODUCCIÓN	8
1. INVESTIGACIÓN	9
1.1. ANTECEDENTES	9
1.2. JUSTIFICACIÓN	10
1.3. OBJETIVOS	11
1.3.1. OBJETIVOS GENERALES	11
1.3.2. OBJETIVOS ESPECÍFICOS	11
2. MARCO TEÓRICO	12
2.1. DOMÓTICA	12
2.1.1. HISTORIA	12
2.1.2. ESTADO ACTUAL	13
2.1.3. DOMÓTICA EN COLOMBIA	15
2.2. LENGUAJE UNIFICADO DE MODELADO (UML)	15
2.3. WAKANDA	22
2.3.1. INSTALACIÓN Y CONFIGURACIÓN	22
2.3.2. WAKANDA STUDIO	38
2.3.3. WAKANDA SERVER	38
2.3.4. LENGUAJE DE PROGRAMACIÓN	39
2.3.5. VENTAJAS	39
2.3.6. DESVENTAJAS	39
2.4. HTML5	39
2.5. JAVASCRIPT	42
2.6. CSS3	43
2.7. JQUERY	44
2.8. PROCESSING	45
2.8.1. PROCESSING PARA ARDUINO	46
2.8.2. PROCESSING PARA ANDROID	46
3. ARDUINO	48
3.1. PINES DE ENTRADA Y SALIDA	49
3.2. BIBLIOTECAS	50
3.3. MÓDULOS	51
3.4. LENGUAJE DE PROGRAMACIÓN	54
3.5. INSTALACIÓN	58
3.6. APLICACIONES	59

4. ANDROID	60
4.1. HISTORIA	60
4.2. CARACTERÍSTICAS	60
4.2.1. VERSIONES	62
4.2.2. USOS	63
4.2.3. DESARROLLO	63
4.3. APLICACIONES	63
5. ANÁLISIS Y DISEÑO	64
5.1. DIAGRAMA DE FLUJO	64
6. IMPLEMENTACIÓN EN WAKANDA	69
6.1. DESARROLLO DE LA APLICACIÓN	69
6.1.1. CONEXIÓN CON ARDUINO	75
6.1.2. RESULTADOS	76
7. IMPLEMENTACIÓN EN PROCESSING	77
7.1.1. DESARROLLO DE LA APLICACIÓN	77
7.1.2. CONEXIÓN CON ARDUINO	77
7.1.3. RESULTADOS	78
8. IMPLEMENTACIÓN EN PROCESSING PARA ANDROID	79
8.1. DESARROLLO DE LA APLICACIÓN	79
8.2. CONEXIÓN CON ARDUINO	79
8.3. RESULTADOS	80
9. APLICACIÓN FINAL	81
10. CONCLUSIONES	82
BIBLIOGRAFÍA	83

## INTRODUCCIÓN

Es un hecho que en la actualidad la tecnología ha tenido un avance considerable a comparación de años anteriores, lo que ha permitido el desarrollo de diferentes aplicaciones y dispositivos, donde muchas de las actividades cotidianas son realizadas por medio de dispositivos electrónicos.

Es un hecho que la domótica está generando gran impacto en la sociedad, y que en un futuro no muy lejano, gracias a las ventajas que ofrece, será la gran tendencia mundial. Y no solo la domótica se está posicionando fuerte, en nuestra sociedad hay un boom tecnológico por así decirlo, en cuanto al uso de celulares inteligentes y tabletas.

Los *Smartphone*, como se conocen los celulares inteligentes, tienen gran impacto a nivel mundial, donde una parte considerable de la humanidad posee este tipo de dispositivo electrónico. En países como estados unidos se tiene cifras de que alrededor de más del 50% de los usuarios de *Smartphone*, poseen celulares con sistema operativo *Android*, el sistema operativo que sigue en la lista es *iOS* de *iPhone*, con un 15%, mostrando una alta tendencia por los dispositivos con el sistema operativo que provee Google.

Basándose en esta tendencia en los *Smartphones*, en cuanto a su preferencia por sistema operativo, y llegando a la conclusión que los usuarios también prefieren el sistema operativo *Android* en las *Tablets*, se decide crear una aplicación para los dispositivos que utilizan este sistema operativo.

La aplicación tiene como fin combinar el uso de estos dispositivos electrónicos con la domótica, creando un prototipo de sistema de control domótico basado en *Android*, el cual permitirá controlar, luces, alarmas, puertas, entre otros.



# 1. INVESTIGACIÓN

A continuación se hablara de los sucesos que antecieron el proyecto, al igual que se justificara la razón por la cual se realiza el proyecto, donde a partir de estos antecedentes y esta justificación, se procederá a plantear los objetivos del proyecto.

## 1.1 ANTECEDENTES

La domótica, es uno de los efectos de la globalización, donde en países desarrollados está teniendo un gran avance. En sus inicios presento grandes inconvenientes, debido al costo elevado que presentaba para los usuarios, hoy en día es mucho más asequible que en sus primeras fases, y las ventajas que ofrecen son mucho más evidentes hoy en día que hace unos años; entre estas ventajas se tiene el ahorro energético, confort, seguridad, accesibilidad, entre otras.

El avance de la domótica en cada país es diferente, en unos países ha tenido gran impacto y desarrollo, mientras que en otros países el desarrollo es más bien escaso. Colombia no se queda atrás en cuanto a este tema, en nuestro país el avance ha sido grande, tanto así que los proyectos de esta índole se está llevando hasta los estratos 3, con precios acordes asequibles y facilidades de pago.

Debido a la gran cantidad de empresas que están ejerciendo en este campo, cada proyecto que se relaciona con la domótica, para poder ser exitoso debe tener un valor agregado, para de esta manera sobresalga en el mercado.

El uso de dispositivos como lo son los *Smartphones* y las *Tablets*, ha ido generando un consumismo excesivo en la sociedad, donde un gran porcentaje de esta, posee uno o ambos dispositivos. La preferencia por uno u otro dispositivo varía de las funcionalidades que quiera el usuario, o simplemente de la condición económica que se posea.

Existen diversos sistemas operativos que corren sobre estos dispositivos, entre los destacables se tienen *Android* de Google, *iOS* de Apple, *WindowsMobile* y *BlackBerryOS*. Los sistemas operativos que más se usan son el *Android* y el *iOS*, donde alrededor de más del 50% de los usuarios prefiere utilizar *Android* que *iOS*, debido a las diferentes ventajas que posee el sistema operativo de Google.

## 1.2 JUSTIFICACIÓN

Analizando el estado actual de la domótica, y de los diferentes tipos de sistemas de control domótico que se han desarrollado hasta el momento, se evidencia claramente el avance tecnológico que ha tenido este tipo de sistemas. Aunque son infinitos los proyectos de domótica que se han creado, se ha escuchado o en otras palabras se ha visto muy poco la implementación de sistemas de control domótico manejados por medio del celular. Este tipo de sistemas de control, podrían generar un auge, ya que permiten una mayor interacción entre el usuario y el sistema, ya a la vez mayor comodidad para la persona, debido a que puede controlar su casa por medio de su celular.

El mercado actual de los Smartphones y las Tablets, está presentando un crecimiento acelerado, donde las tendencias y las preferencias de los usuarios cada vez es más notoria, donde los dispositivos con sistema Android están marcando la diferencia.

El presente trabajo, tiene como fin integrar la domótica con los dispositivos Android, creando un prototipo de sistema de control domótico, manejado ya sea por un Smartphone o una Tablet, con sistema operativo Android.

## 1.3 OBJETIVOS

### 1.3.1 OBJETIVO GENERAL

Desarrollar un prototipo de control domótico por medio de Arduino y *Processing*, a través de una aplicación móvil ejecutada por dispositivos que cuenten con Android, que transmite los datos a ser procesados por el micro-controlador mediante Bluetooth.

### 1.3.2 OBJETIVOS ESPECÍFICOS

- Desarrollar una aplicación móvil en *Processing* para *Android*.
- Establecer las desventajas de Wakanda en la ejecución del proyecto frente a las ventajas ofrecidas por Android.
- Controlar alarmas, luces y motores por medio del dispositivo móvil celular a través del Bluetooth.
- Recibir datos por medio de Bluetooth y procesarlos a través de Arduino.

## 2. MARCO TEÓRICO

### 2.1 DOMÓTICA

Se entiende por domótica el *“Conjunto de sistemas que automatizan las diferentes instalaciones de una vivienda.”*(Española). Donde la automatización se desarrolla mediante el uso de diferentes elementos, lo que ofrece ciertas ventajas como un mayor ahorro energético, mayor comodidad y seguridad, entre otras.

Hay muchas funciones en el hogar que se pueden automatizar, lo que generaría una mayor comodidad en el hogar; como ejemplo podríamos hablar de un sistema de riego automático, el cual, actualmente está implementado en muchos hogares, otro ejemplo claro, son los sistemas de seguridad. Llegando más a fondo se podría hablar de sistemas de control que por ejemplo se encarguen de controlar una llave de agua, es decir si por alguna circunstancia se queda una llave abierta, el sistema estaría programado para cerrar la llave automáticamente sin tener que recibir una orden. *“Por tanto, los sistemas de un hogar digital realizan funciones útiles para los usuarios que viven dentro de ellas, estas funciones pueden adquirir mayor relevancia si un tercero proporciona un servicio de valor añadido.”*(Junestrand, Passaret, & Vázquez, 2005).

La domótica, ofrece diferentes ventajas a los usuarios, entre las que cabe destacar:

- Al automatizar ciertos lugares del hogar, se puede obtener un ahorro energético, sin necesidad de sustituir los aparatos electrónicos presentes.
- El confort, es una característica y una ventaja que se obtiene al automatizar el hogar.
- La seguridad en un hogar automatizado es un punto que satisface a cualquier usuario, ya que por medio de alarmas y otros sistemas, se protege no solo la integridad del hogar sino la integridad de la persona.
- Debido a que muchos servicios pueden controlarse remotamente, se crea una accesibilidad que favorece a cualquier tipo de persona, ya sea adulto o niño, o simplemente personas con alguna discapacidad.

Los elementos que por lo general se utilizan en cualquier sistema domótico, son los controladores, actuadores y sensores. Un controlador, es un sistema que tiene como fin manejar un hardware, proporcionando una interfaz amigable para el usuario. Los actuadores, son dispositivos capaces de utilizar cualquier tipo de energía para activar un elemento final, como un motor, una válvula, entre otros. Y

por último, los sensores, son dispositivos electrónicos, capaces de detectar señales físicas o químicas, y transfórmalas en señales eléctricas para ser procesadas e interpretadas.

### **2.1.1 HISTORIA**

Se podía hablar que la domótica inicio alrededor de los años 70, más exactamente en 1975, donde después de muchas investigaciones se inició la automatización de edificios basados en la tecnología X-10, que actualmente se utiliza; este protocolo se extendió no solo por Estados Unidos, sino también por Europa, donde Reino Unido y España, fueron los países que más se acoplaron a este estándar. A partir de este protocolo, se derivaron gran cantidad de aplicaciones, y aun en la época actual se siguen creando empresas alrededor de este protocolo, las cuales aportan novedades que mejoran la experiencia del usuario final: aunque este protocolo también evidencia ciertos inconvenientes, entre los cuales se destaca la vulnerabilidad que se presenta debido a las corrientes portadoras.

Al mismo tiempo que se expandía el protocolo anteriormente nombrado, algunas empresas del sector eléctrico decidieron unirse y crear KNX, el cual está basado en EIB, BatiBus y EHS, y que en sus inicios, recibió el nombre de *Konnex*. Este protocolo tuvo tal aceptación en el mercado, que se convirtió en un estándar mundial, el ISO/IEC 14543-3.

Un protocolo que surgió fue LonTalk, el cual es un protocolo de comunicación, que en 1999 fue normalizado como estándar de control de redes según la norma ANSI/CEA-799.1-B, después de esta normalización vinieron muchas otras a las aplicaciones que ofrecía el protocolo, hasta el punto de convertirse en un estándar europeo para domótica en el año 2005, con referencia 14908, y un estándar mundial ISO/IEC-14908.

A partir del año 2000, las pequeñas empresas se especializan como fabricantes especializados de productos basados en el estándar KNX, evidenciando un fuerte crecimiento, cubriendo nuevos mercados con precios competitivos. En gran parte de la década del 2000, las nuevas incursiones en desarrollos se basaban en estos protocolos; no fue hasta el 2006 que la domótica tomo un nuevo cambio.

En el año 2006, nacen los sistemas domótico inalámbricos RF, algunos de los cuales son compatibles con el primer protocolo de domótica creado y el resto a los diferentes protocolos creados hasta la fecha.

## 2.1.2 ESTADO ACTUAL

Para analizar el estado actual de la domótica, indirectamente se tendría que hablar del futuro de esta, debido al modo que día a día avanza la tecnología. Actualmente se están automatizando diferentes espacios, como lo es el hogar, los centros comerciales, los edificios, entre otros. Esta automatización se da debido al fuerte impacto y al auge que está teniendo esta tendencia en el mercado, para que de una manera u otra se cree más confort y más comodidad para las personas.

Las características que gobiernan este tipo de sistema y que hacen que se esté convirtiendo en una tendencia mundial son:

- La implementación de una interfaz sencilla y agradable para el usuario, hace que sea más fácil de usar y aceptar.
- Se debe ofrecer flexibilidad al usuario, lo que implica posibilidad de ampliar la oferta y/o modificarla sin que suponga un aumento elevado para el usuario final.
- La implementación de los sistemas domótico se efectúa por medio de módulos independientes, lo que genera mayor fiabilidad en el sistema, ya que un fallo en un módulo no va a afectar a otro, y la implementación de módulos nuevos, no perjudicara los anteriores.
- A pesar de que el sistema está dividido por modulo, este debe poder integrarse con el resto de subsistemas para facilitar la comunicación y permitir el intercambio de información.

Los servicios que nos ofrece actualmente la domótica, que anteriormente se habían mencionado como ventajas, son los siguientes:

- Racionalización de cargas eléctricas.
- Climatización de lugares.
- Programación de equipos domésticos.
- Programación y detección de apertura de puertas y ventanas.
- Zonas de control de iluminación.
- Regulación automática de la iluminación.

- Accionamiento automático de sistemas de riego, de cortinas, persianas, entre otros.
- Automatización de los equipos para manejarlos de la manera más cómoda y obteniendo un control eficiente de estos.
- Climatización y ventilación auto regulable.
- Control de acceso con diferentes sistemas de reconocimiento.
- Control de forzado de puertas y/o ruptura de cristales.
- Control de lugares, es decir control de presencia de persona en los lugares.
- Video vigilancia.
- Teleasistencia y telemedicina.
- Automatización para enfermos.
- Detección de daños (Agua, gas, luz, entre otros).
- Detección de incendios.
- Activación de alarmas.
- Gestión de las comunicaciones.

Se puede hablar que este tipo de servicios es un abre bocas a la cantidad de servicios que la domótica ofrecerá en un futuro, integrando diferente dispositivo al igual que diferentes herramientas

### **2.1.3 DOMÓTICA EN COLOMBIA**

En Colombia existen diferentes empresas dedicadas al negocio de la domótica, las cuales ofrecen diversos servicios, entre los que cabe destacar la automatización de hogares, edificios y sobretodo los sistemas de seguridad y de acceso a las instalaciones. Se dice que actualmente las casas inteligentes llegan a los estratos 3, donde se ofrecen precios competitivos y facilidades de pago.

*“El costo promedio de automatizar una vivienda con acceso con huella, control de iluminación, sonido de teatro en casa, zona social y sistema de seguridad ronda el 5 por ciento del valor del inmueble. Si quiere implementar cortinas y más zonas de*

*la vivienda, o agregar pantallas táctiles, el costo puede ascender al 10 por ciento”, comenta Hernán Castro, especialista de Technoimport. La gran ventaja es que la gran mayoría de los sistemas son inalámbricos y por eso se reduce mucho el costo de la obra civil. Los estudios a nivel mundial corroboran que por cada dólar invertido en automatización el inmueble se valoriza.” (La domótica sigue ganando terreno, 2010).*

A continuación se mencionaran algunas empresas que se dedican a esta actividad en Colombia:

- Hometech el Hogar Digital SAS.
- Ebingel.
- C.A.S.A constructores.
- Domoworld.

Cabe destacar que la mayoría de las empresas que practican esta actividad, están ubicadas tanto en Medellín como en Bogotá.

## **2.2 LENGUAJE UNIFICADO DE MODELADO (UML)**

*“El lenguaje UML es un estándar OMG diseñado para visualizar, especificar, construir y documentar software orientado a objetos” (Booch, Rumbaugh, & Jacobson). Este lenguaje permite visualizar, construir, documentar y especificar un sistema, a partir de un modelado, incluyendo funciones del sistema. En otras palabras “se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir.” (Wikipedia, 2013).*

Para que un sistema sea eficiente, es necesario que cuente con un modelado para su construcción, ya que este permite:

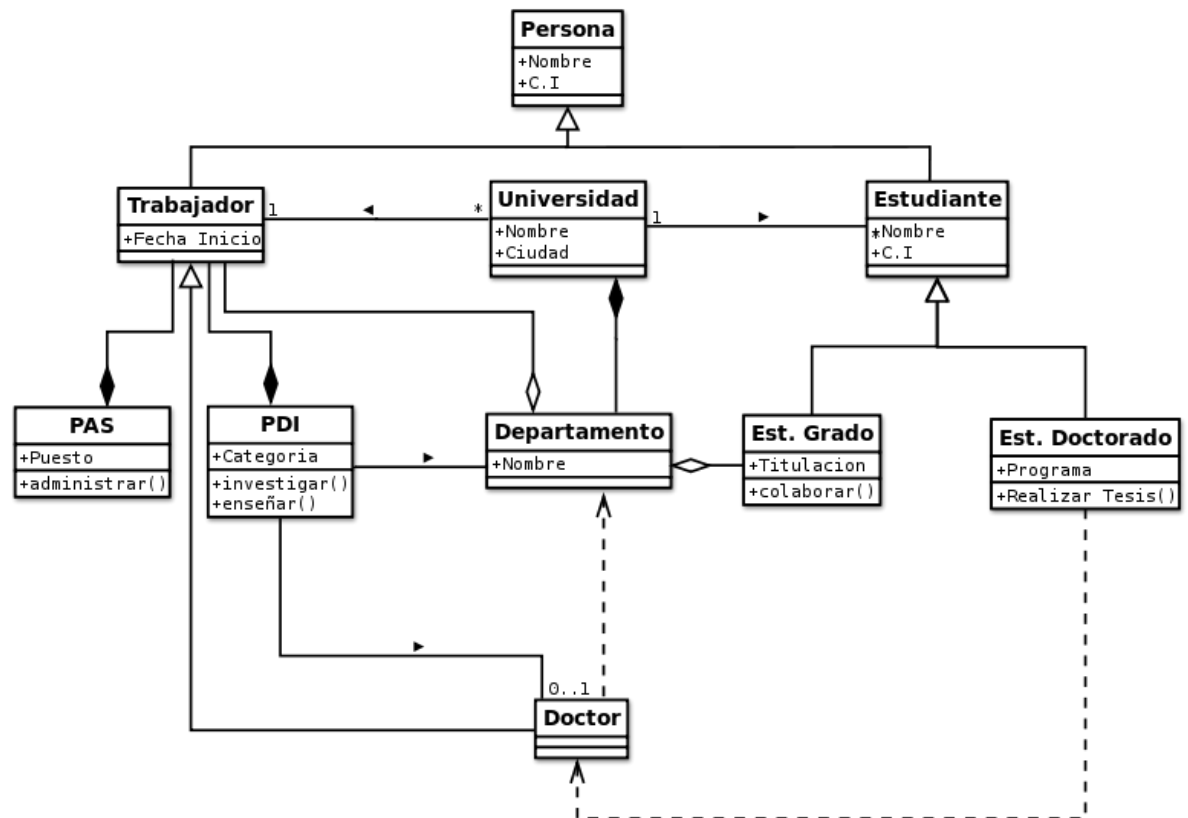
- Establecer la simplificación y reutilización.
- Comprender de una manera analítica y conceptual lo que se está construyendo.
- Especificar el comportamiento del sistema.
- Comunicar los módulos que posea el sistema.

Existen diversos tipos de diagramas de UML, los cuales se podrían clasificar de acuerdo a estructura, comportamiento e interacción. De acuerdo a su estructura, se tienen los siguientes tipos de diagrama:

- **DIAGRAMA DE CLASES**

Los diagramas de clases, son diagramas de tipo estático, que proporciona la estructura del sistema, mostrando las clases, los atributos y la relación que existe entre ellos. Este diagrama tiene como fin crear un diseño conceptual que muestre los componentes que se encargaran del funcionamiento del sistema. A continuación se puede ver un ejemplo de diagrama de clases:

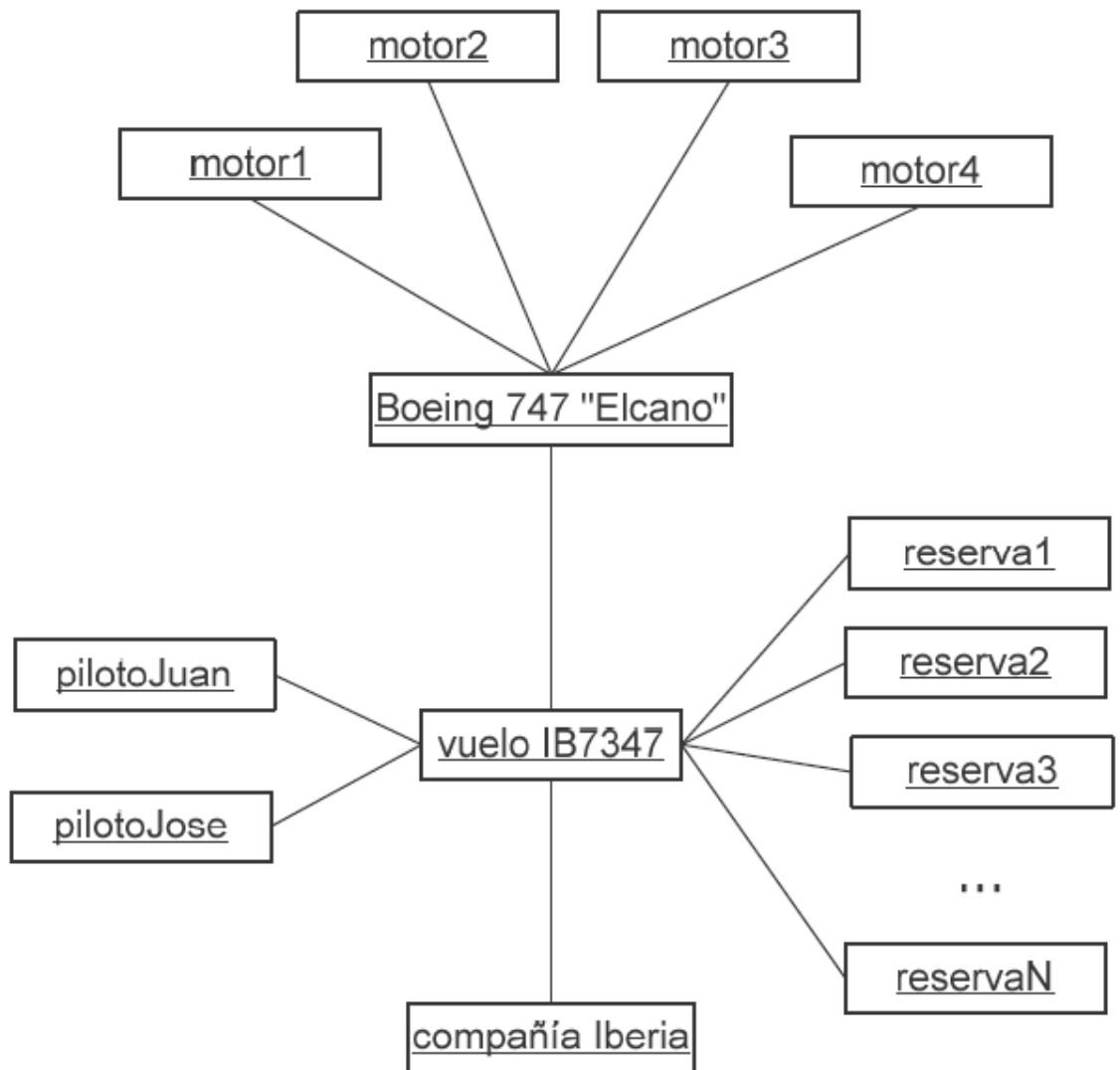
Diagrama de Clases



- **DIAGRAMA DE OBJETOS**

Los diagramas de objetos representan instancias de los elementos que aparecen en los diagramas de clases en un momento particular del sistema. La siguiente imagen es un ejemplo de un diagrama de objetos:





- **DIAGRAMA DE COMPONENTES**

Este tipo de diagrama muestra como el software se divide en componentes, mostrando las dependencias que existen entre esto. Este tipo de diagramas son utilizados con frecuencia en el campo de la arquitectura de software, aunque por lo general pueden ser usados para modelar cualquier sistema.

- **DIAGRAMA DE ESTRUCTURA COMPUESTA**

Este tipo de diagrama, muestra la estructura interna de una clase, y las colaboraciones que esta estructura permite; en otras palabras, una estructura compuesta es una serie de elementos que están

interconectados, y que a su vez colaboran en tiempo y ejecución para lograr un propósito, donde cada elemento que actúa tiene un rol definido de colaboración.

- *DIAGRAMA DE PAQUETES*

El diagrama de paquetes, se encarga de mostrar como se encuentra dividido el sistema en agrupaciones lógicas, evidenciando también las dependencias que existen entre estas agrupaciones.

- *DIAGRAMA DE DESPLIEGUE*

Estos diagramas se utilizan para modelar el hardware que utilizarán los sistemas a implementar. Algunos de los sistemas en los cuales se emplea este modelado son los empotrados, cliente – servidor, y sistemas completamente distribuidos, entre otros.

De acuerdo a su comportamiento, los diagramas se clasifican en:

- *DIAGRAMA DE CASOS DE USO*

Estos diagramas se utilizan en el modelado de los sistemas, desde el punto de vista del usuario, para representar las acciones que puede realizar cada tipo de usuario. A continuación se muestra la estructura que posee este tipo de diagrama:



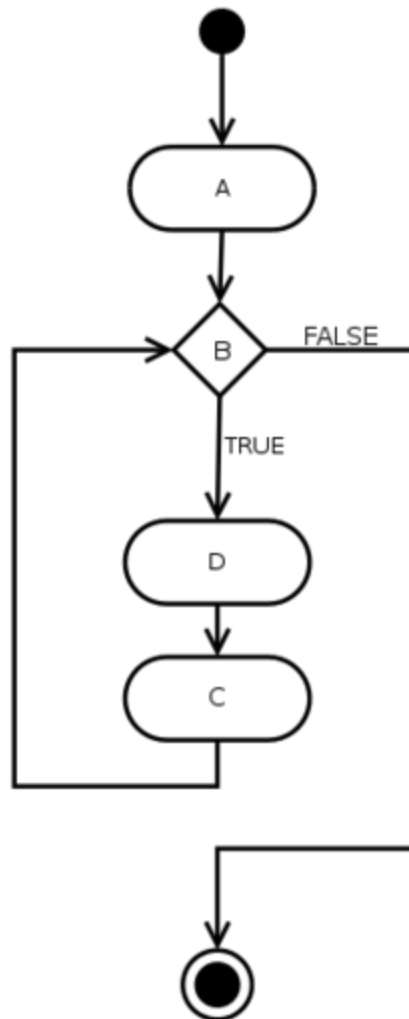
- *DIAGRAMA DE ACTIVIDADES*

El diagrama de flujo o diagrama de actividades, representa gráficamente el orden que va a tener el proceso. Estos diagramas utilizan símbolos con un significado preestablecido y representan la ejecución mediante flechas que conectan los puntos de inicio y fin del proceso. Los símbolos más utilizados son los siguientes:

- El ovalo o elipse, representa el inicio y fin del diagrama.
- El rectángulo, representa una actividad, es decir la ejecución de la actividad.
- El rombo, indica la toma de decisiones, formulando una pregunta.
- El círculo, es un conector, es decir, es el enlace de actividades dentro del diagrama.

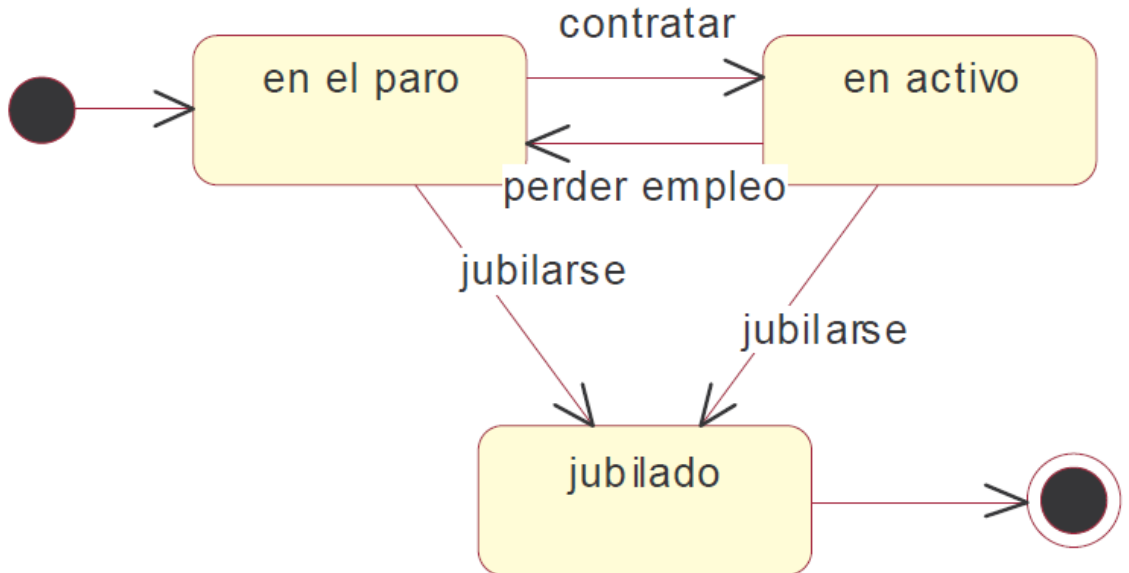
El siguiente ejemplo es el diagrama de actividades para un ciclo:

```
for(A;B;C)  
D;
```



- *DIAGRAMA DE ESTADO*

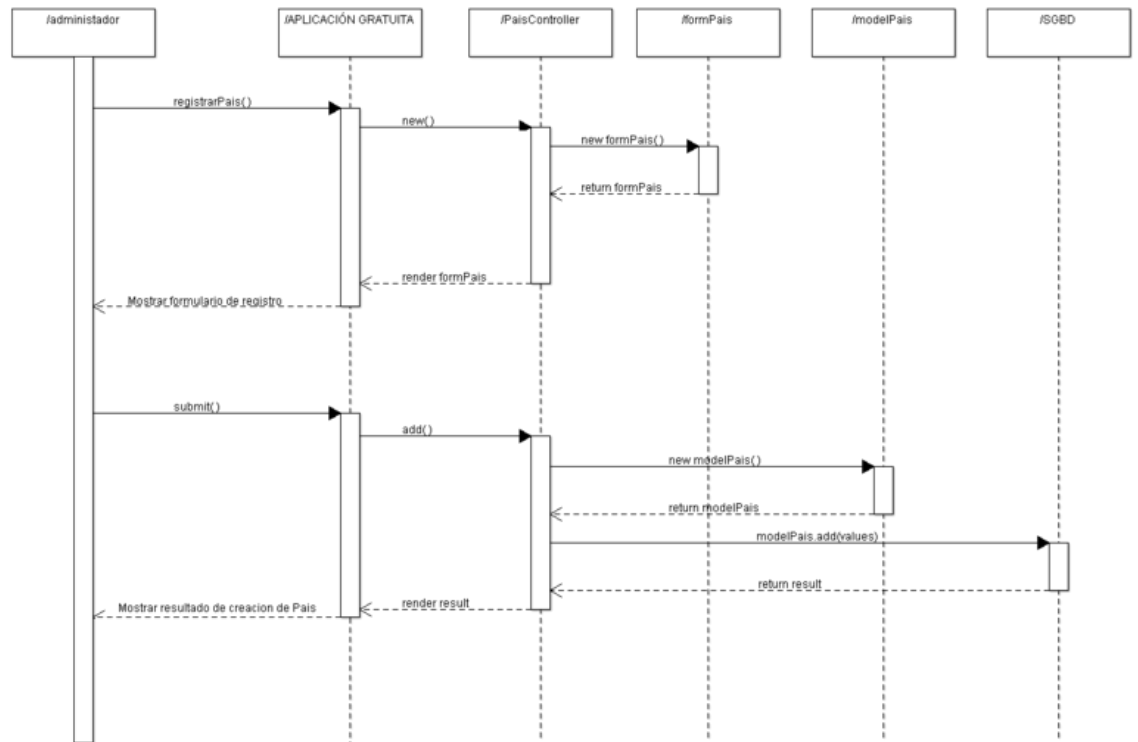
Los diagramas de estado son utilizados para describir el comportamiento de un sistema que está dirigido por eventos; es decir, muestra los estados por los que debe pasar un objeto durante una aplicación en respuesta a ciertos eventos, evidenciando sus respuestas y acciones. Un ejemplo de diagrama de estados es el siguiente:



Y de acuerdo a su interacción, se encuentran los siguientes tipos de diagramas:

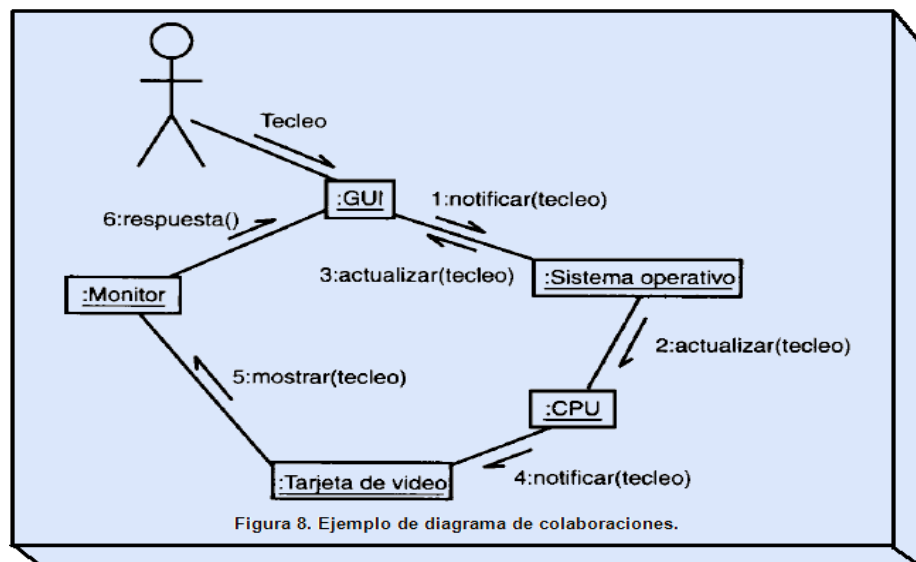
- *DIAGRAMA DE SECUENCIA*

Este tipo de diagrama se emplea para modelar la interacción que poseen los objetos del sistema, resaltando el orden temporal de los mensajes que se intercambian. Un ejemplo de diagrama de secuencia es el siguiente:



- **DIAGRAMA DE COLABORACIÓN**

Los diagramas de colaboración o comunicación permiten observar la interacción de un objeto respecto a los otros. La siguiente imagen nos describe el uso y funcionamiento de estos diagramas:



## 2.3 WAKANDA

“Wakanda es una solución completa cliente web/base de datos todo en uno que ofrece al usuario una serie de widgets muy funcionales que permiten la creación de interfaces web realmente elegantes”(4D, 2012), que incluye 3 componentes principales, entre los que se encuentra, Wakanda server, Wakanda Studio y Wakanda Framework.

El Framework de Wakanda posee un proveedor de datos que permite comunicarse con el servidor, también se compone de una interfaz y un control sobre esta, al igual que una capa de fuente de datos.

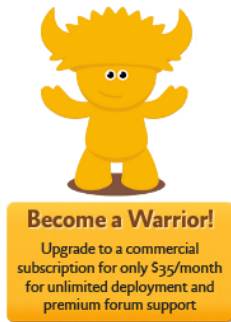
### 2.3.1 INSTALACIÓN Y CONFIGURACIÓN

Para descargar Wakanda, se debe entrar a la página principal: Wakanda: <http://www.wakanda.org/>. Luego se debe dar clic en Downloads:

The screenshot shows the Wakanda website homepage. The header includes the Wakanda logo and navigation links: Home, Overview, Features, Videos, Learning Center, and Downloads (circled in red). A yellow banner in the top right corner says "Wakanda 4 Just released!". The main content area features three yellow buttons labeled "Server", "Framework", and "Studio" on the left. In the center, there's a large image of a laptop displaying the Wakanda Studio interface, with a tablet and two smartphones below it. To the right, the text reads "The data-driven JavaScript development stack for business apps" and a large yellow button says "Download Wakanda now!". Below this, it says "What's new in Wakanda 4?" and "New build (4.132351) released 2013-05-02". At the bottom, there's a blue button that says "TRY IT NOW!". Below the button, the text reads "Play with 1,000,000 database records from your browser" and "Manipulate data with pure server-side JavaScript code".

Se debe seleccionar el sistema operativo y la versión del sistema operativo, para el Wakanda server:

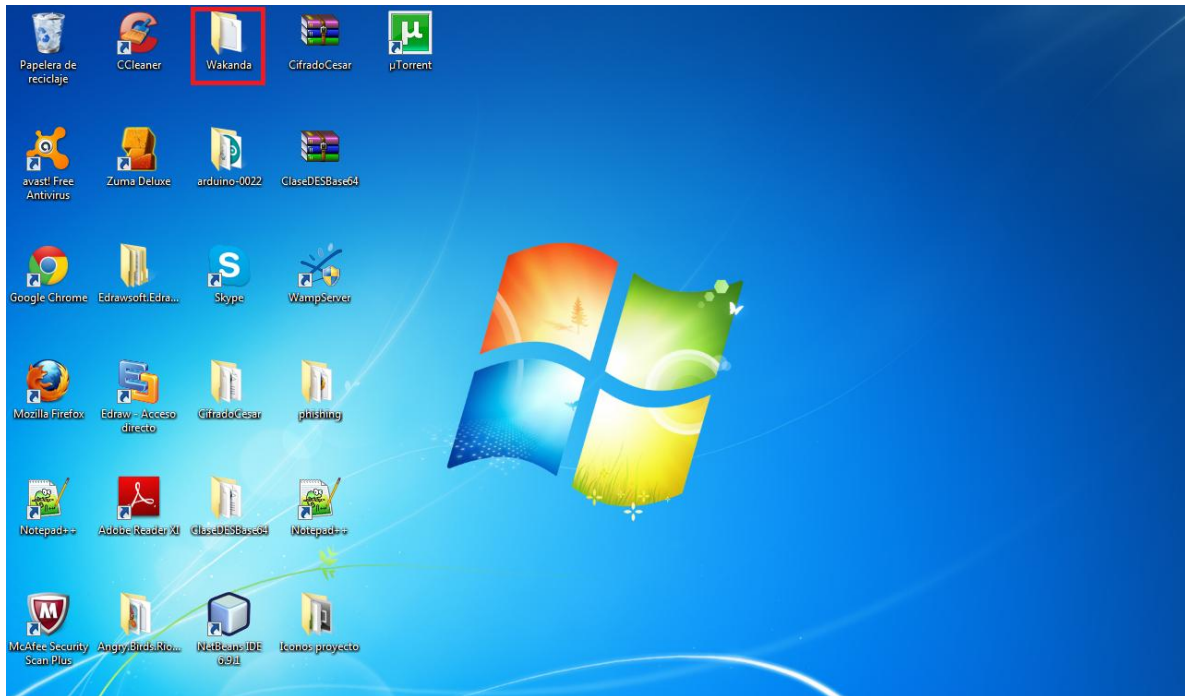
**Current version: Wakanda Production Release**



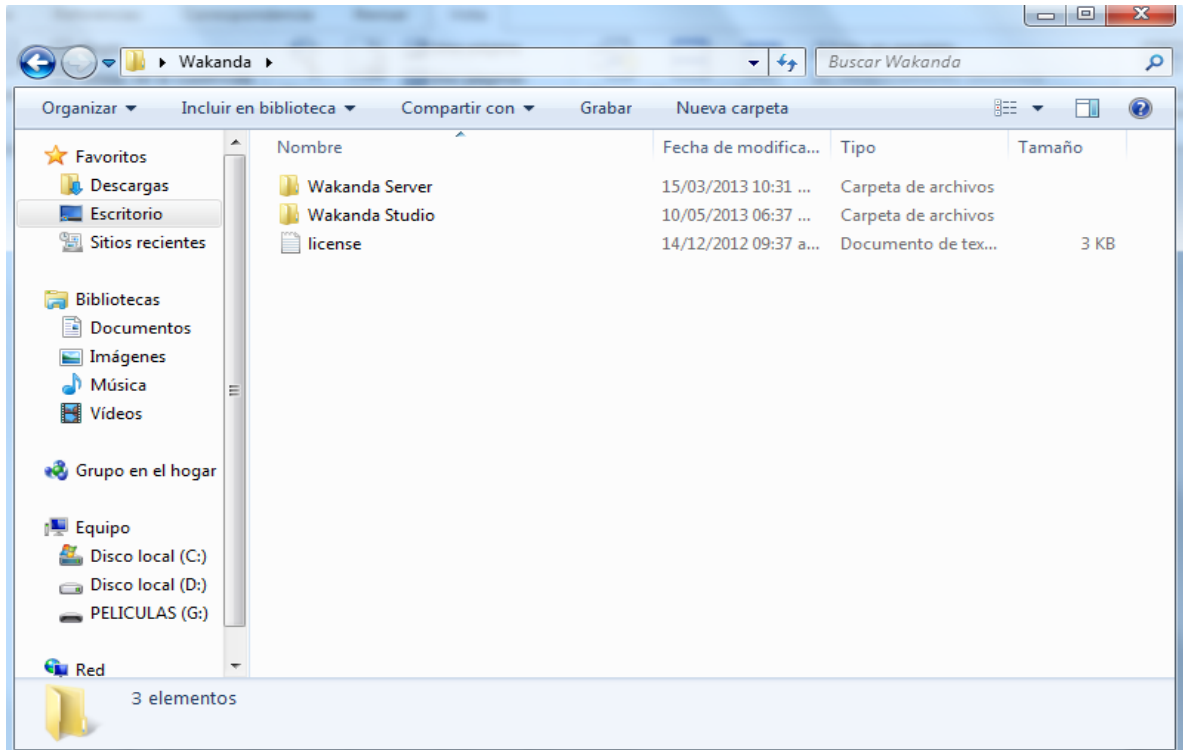
- Wakanda 4**  
 Product Number: **4.0**  
[What's new?](#)  
[Release notes](#)  
[Online Documentation](#)  
[License info](#)  
[System requirements](#)

<b>Wakanda for Windows</b> Build Number: 4.132351 - Issued on: 2 MAY 2013 <a href="#">History</a>			
All-in-One 32-bit Installer		All-in-One 64-bit Installer	
Individual ZIP archives	All-in-one 32 All-in-one 64	Studio	Server 32-bit Server 64-bit
<b>Wakanda for Mac</b> Build Number: 4.132351 - Issued on: 2 MAY 2013 <a href="#">History</a>			
All-in-One Installer			
Individual ZIP archives	All-in-one	Studio	Server 64-bit
<b>Wakanda for Linux</b> Build Number: 4.132351 - Issued on: 2 MAY 2013 <a href="#">History</a>			
All-in-One 32-bit Installer		All-in-One 64-bit Installer	
Individual ZIP archives	N/A	N/A	Server 32-bit Server 64-bit

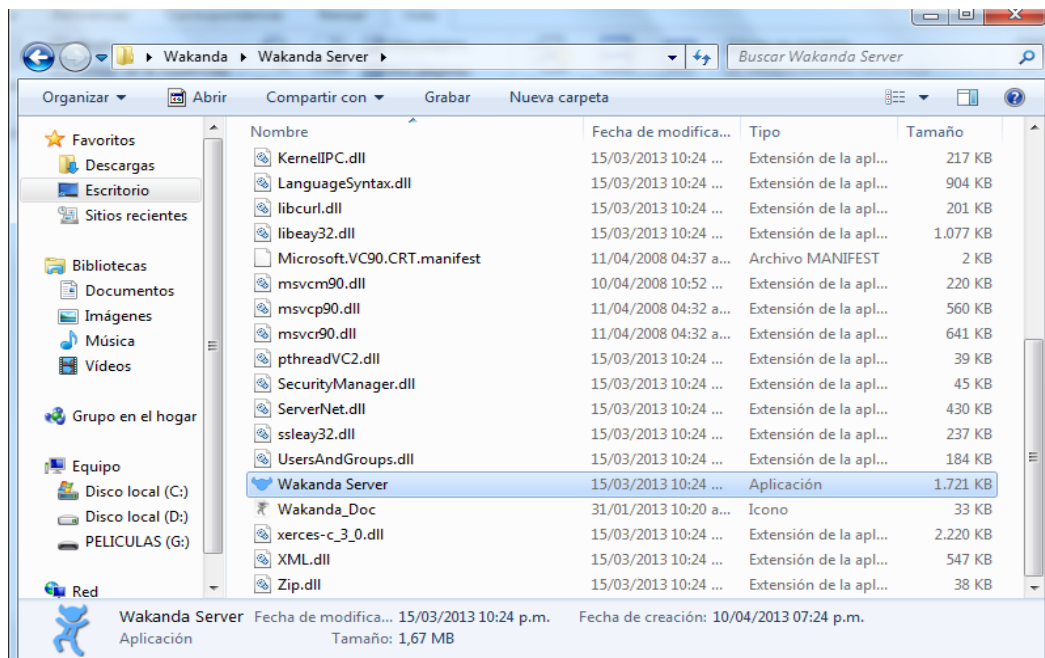
Después de tener los archivos descargados, es decir, el Wakanda server y el Wakanda Studio, estos se deben descomprimir en una carpeta llamada Wakanda:



Ambos archivos quedaran guardados en dicha carpeta:



Para iniciar el desarrollo de la primera aplicación, se debe ejecutar primero el Wakanda Server, debido a que este es el encargado de ejecutar los proyectos:

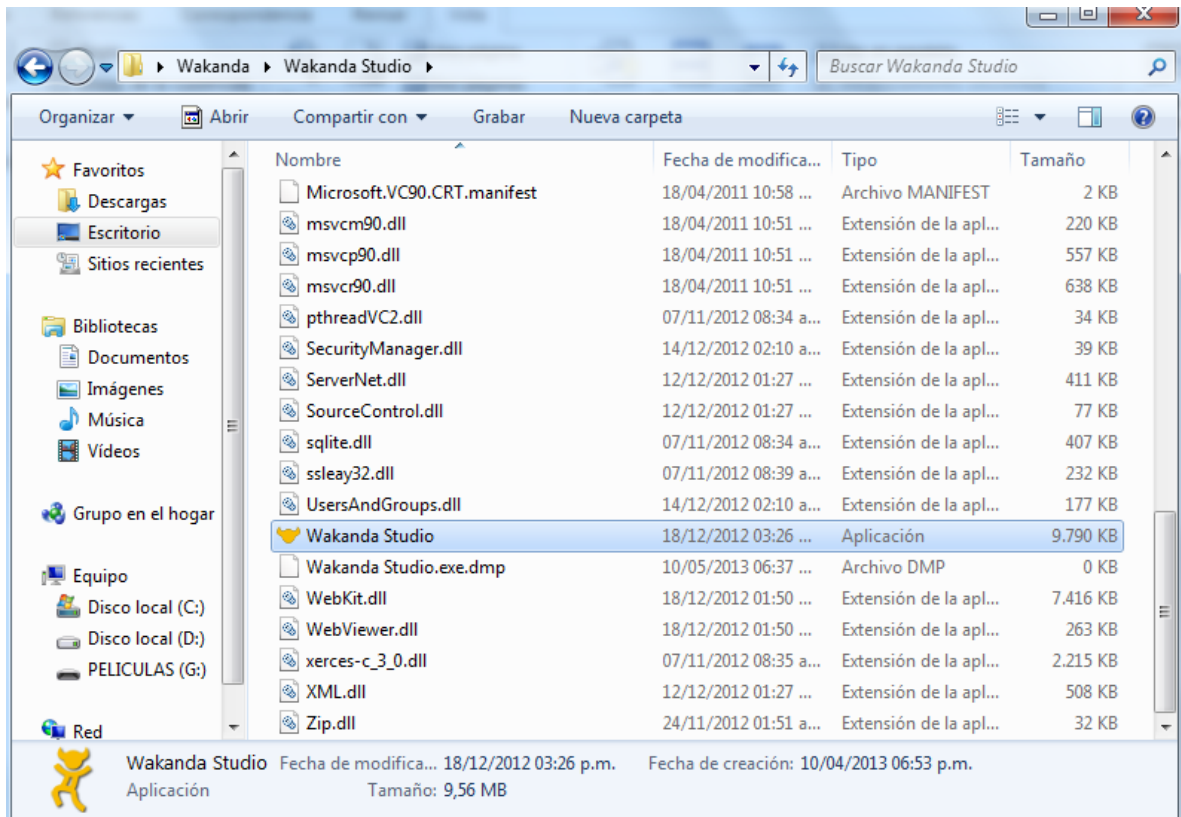




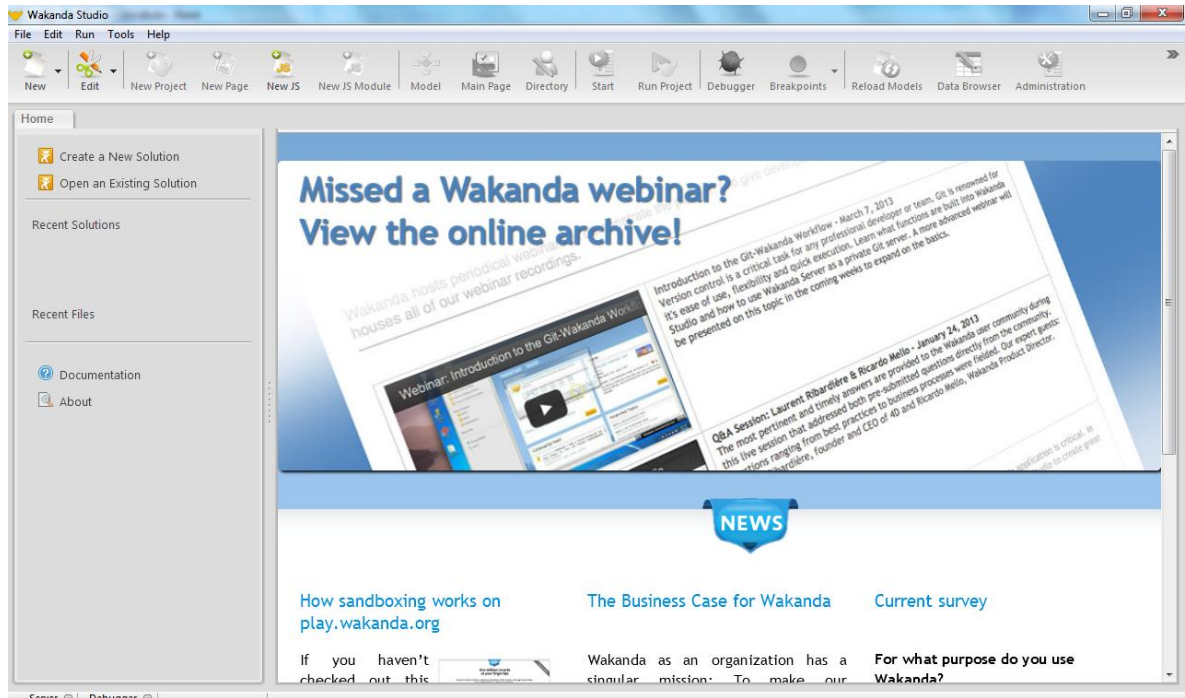
Al momento de ejecutar el Wakanda Server, debe aparecer la siguiente ventana:

```
C:\Users\Valentina\Desktop\Wakanda\Wakanda Server\Wakanda Server.exe
Welcome to Wakanda Server version 4 build 4.129628
Publishing "DefaultSolution" solution
The solution's log file will be stored in the "C:\Users\Valentina\AppData\Local\
Wakanda Server\DefaultSolution-5352\Logs\" folder
- The Administration Web Server listens for connections on port 8080 and secure
port 4433 on all IP addresses
```

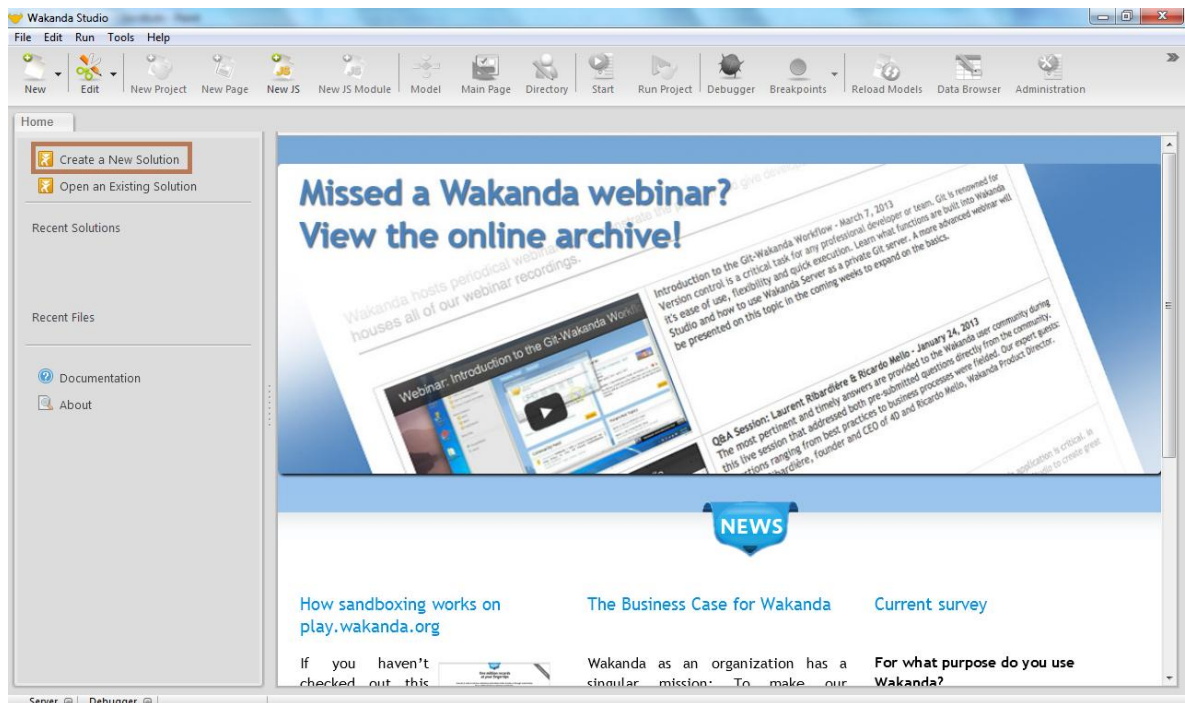
Luego de esto, se procede a ejecutar el Wakanda Studio:



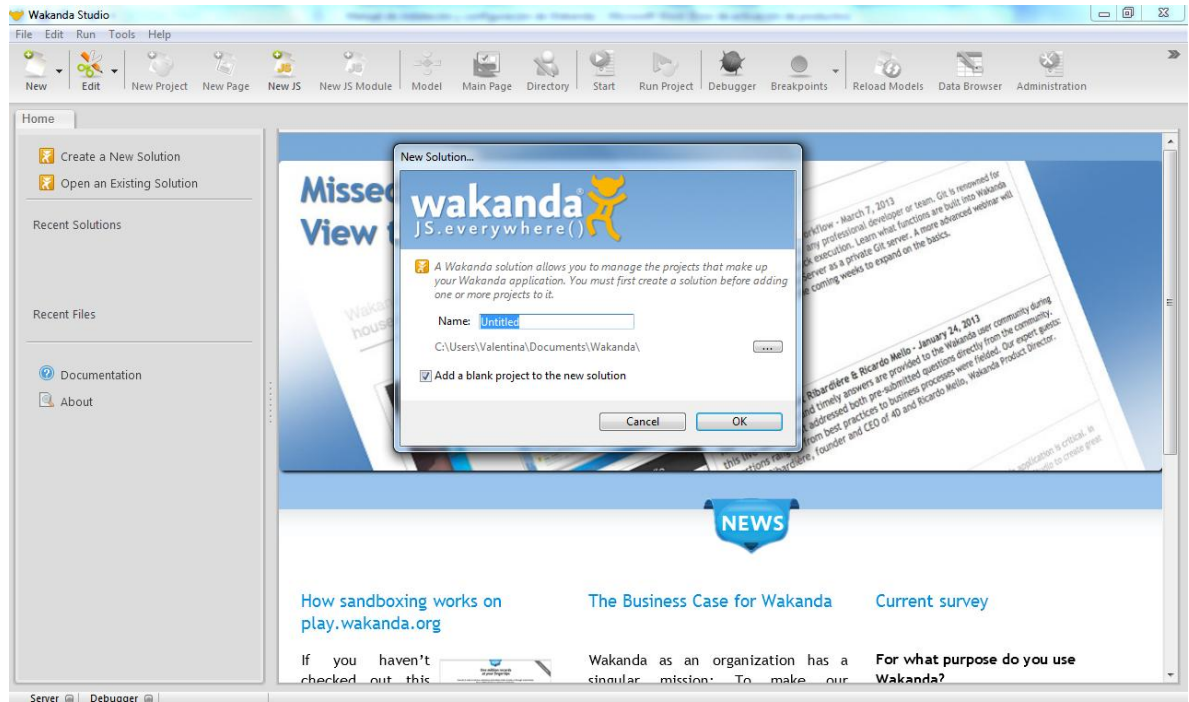
Al momento de ejecutarlo, debe aparecer la siguiente ventana:



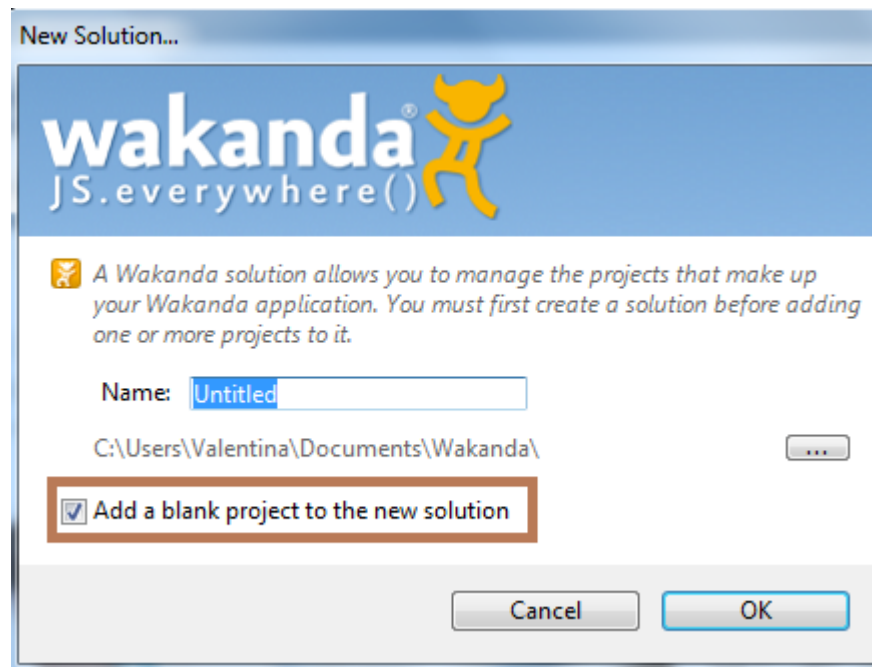
Wakanda trabaja con soluciones, para iniciar cualquier proyecto, primero se debe crear una solución:



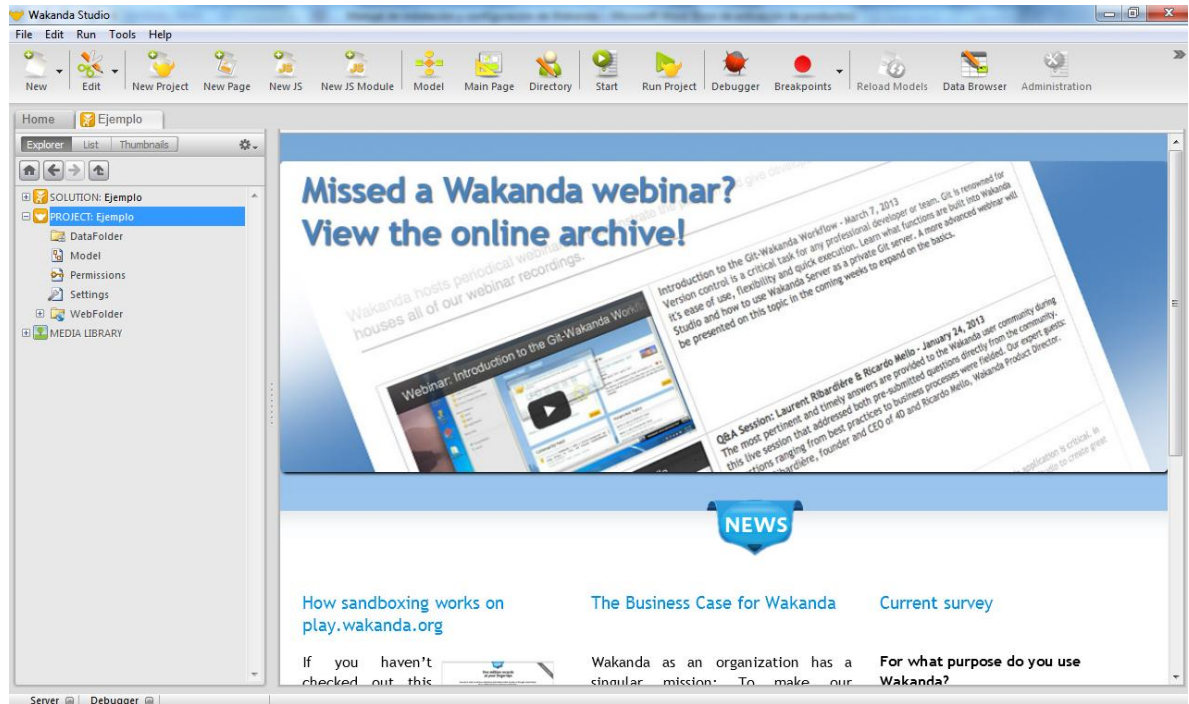
Al momento de crear la nueva solución, debe aparecer la siguiente ventana:



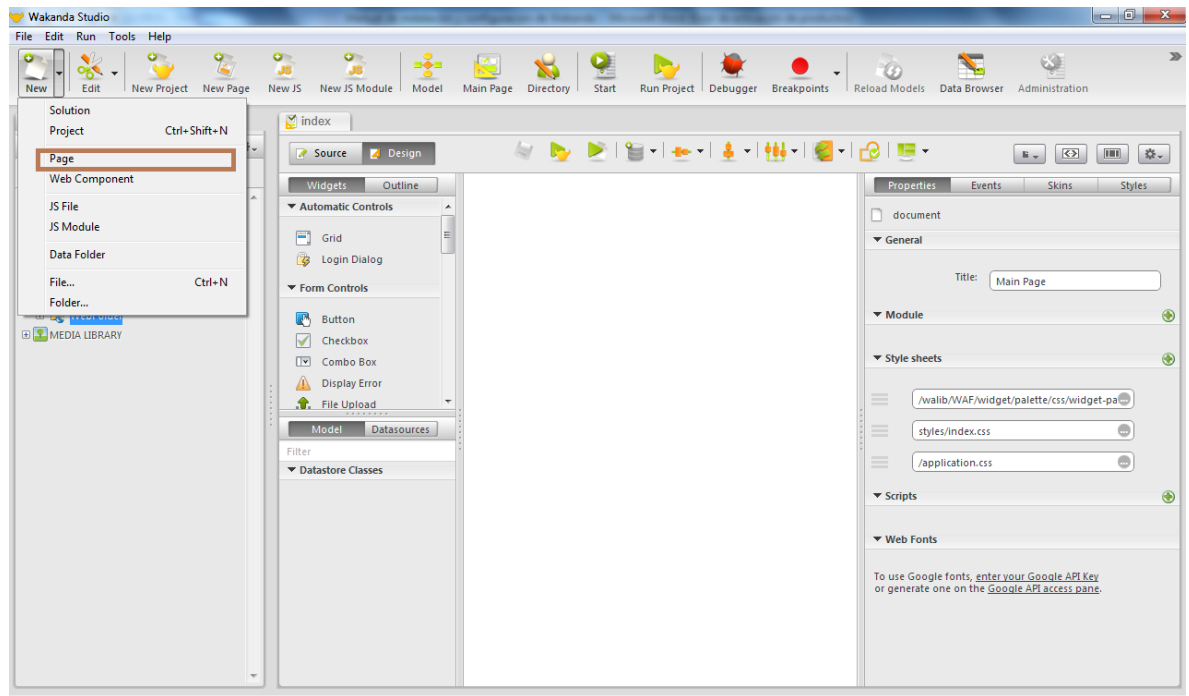
En esta ventana se da el nombre a la solución, y se añade un proyecto en blanco a la solución:



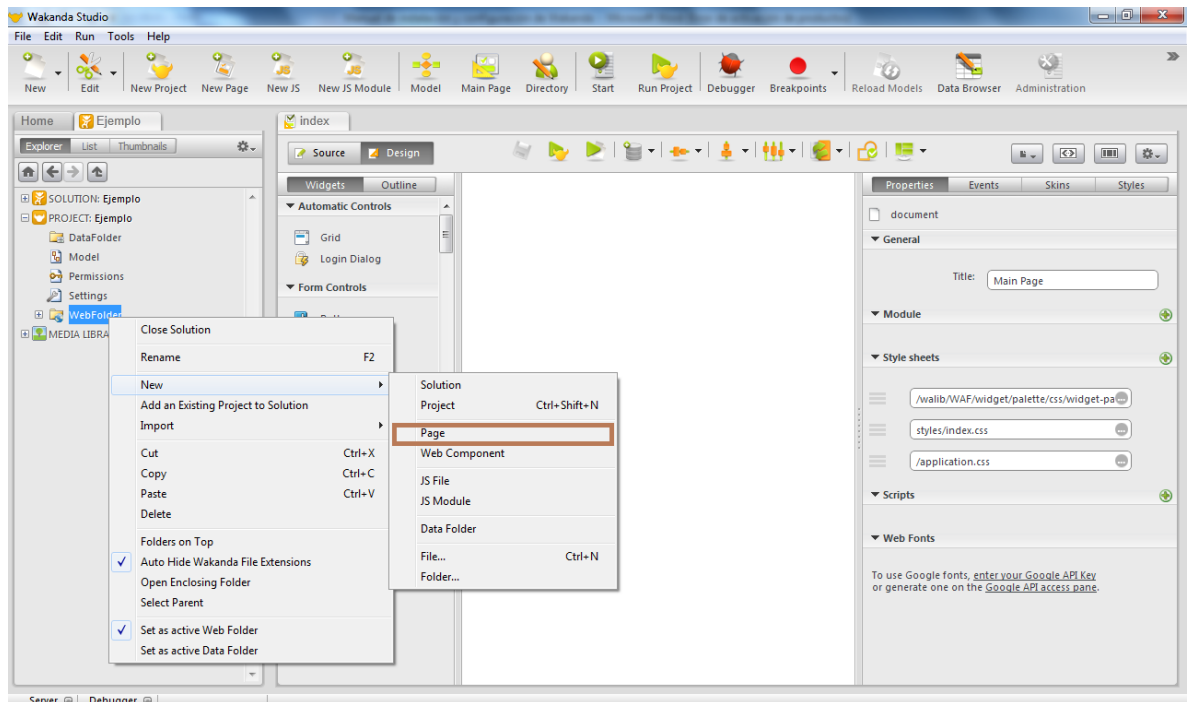
Al crear la solución, se crea también el proyecto:



En Wakanda, se pueden crear aplicaciones web, aplicaciones para móviles y aplicaciones para Tablets. Para crear una aplicación, primero se debe crear una nueva página, se puede hacer dando clic en New, y luego en Page:



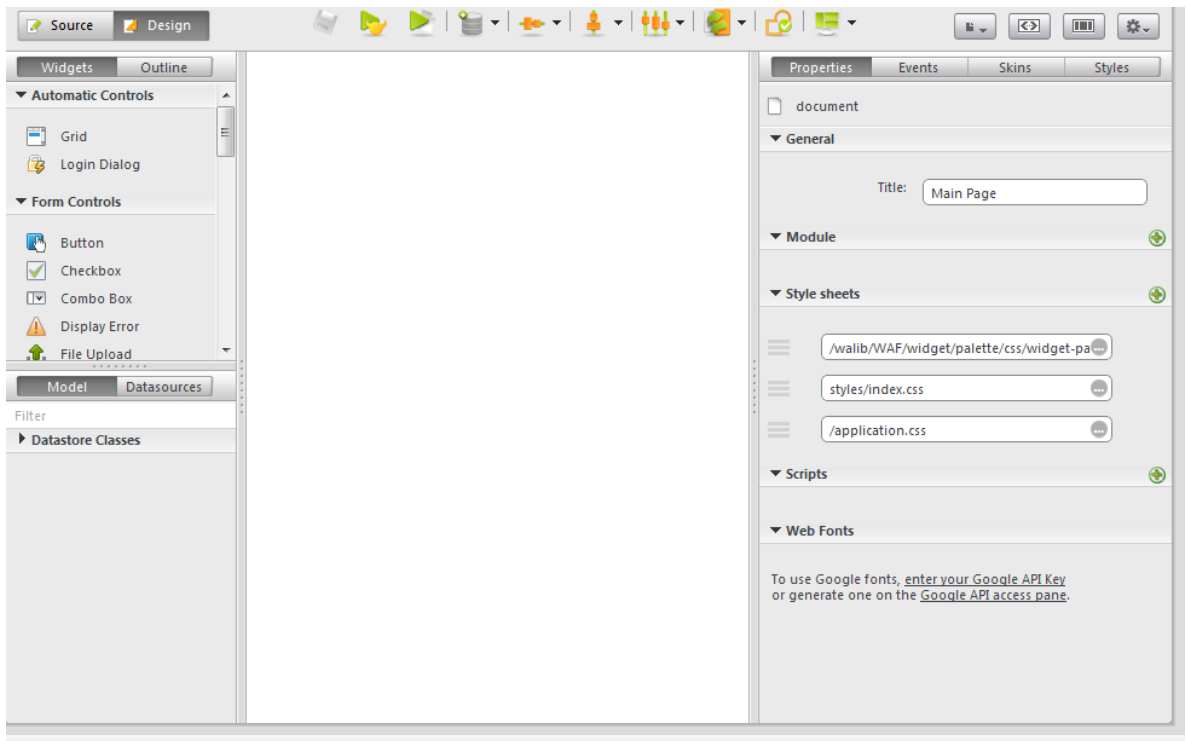
O simplemente, se ubica en WebFolder, se da clic derecho, se selecciona New, y luego Page:



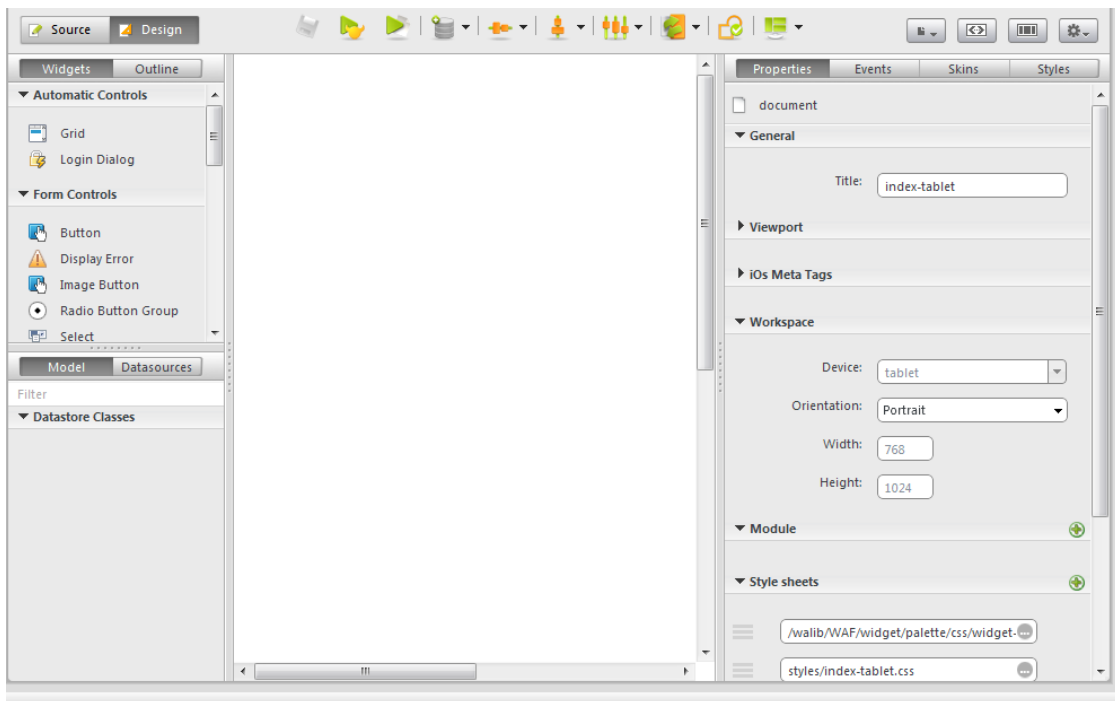
Después de esto, aparece la siguiente ventana, en la cual se coloca el nombre de la página, la cual debe llevar la extensión .waPage, y también se selecciona el tipo de página que se desea crear:



Si es una página web normal, se selecciona la opción Desktop, y debe aparecer lo siguiente:

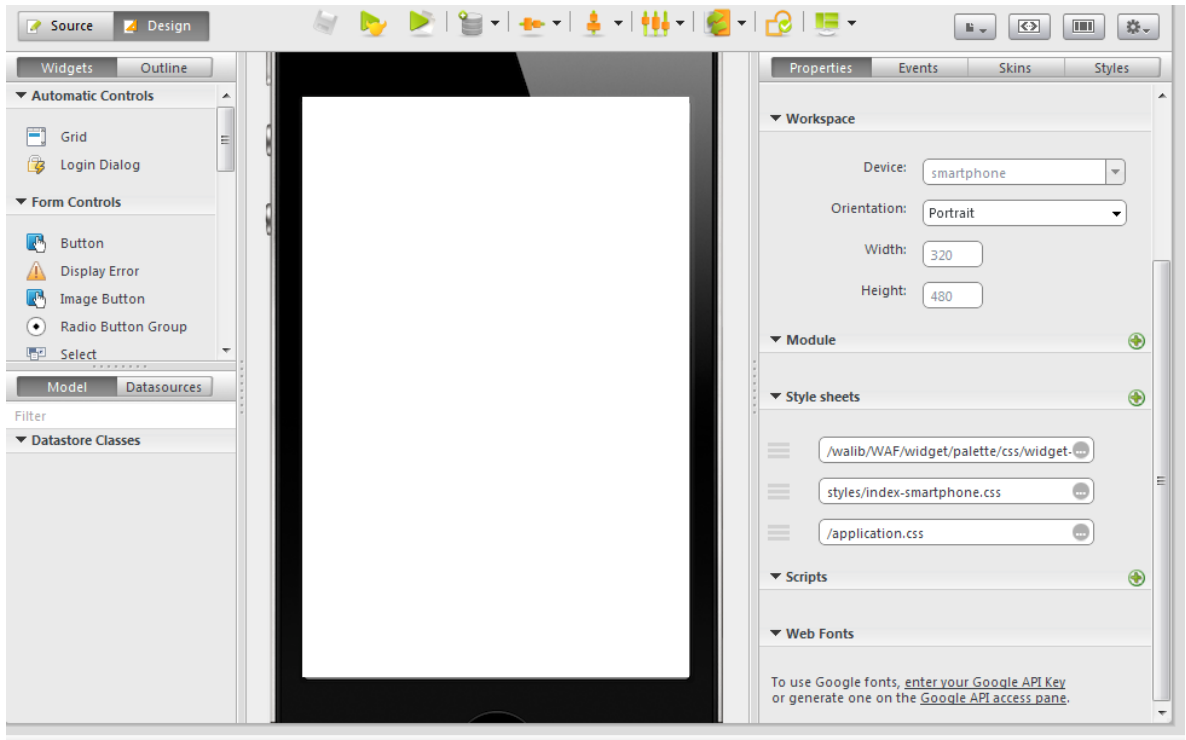


Si se selecciona la opción Tablet, aparece de la siguiente manera:

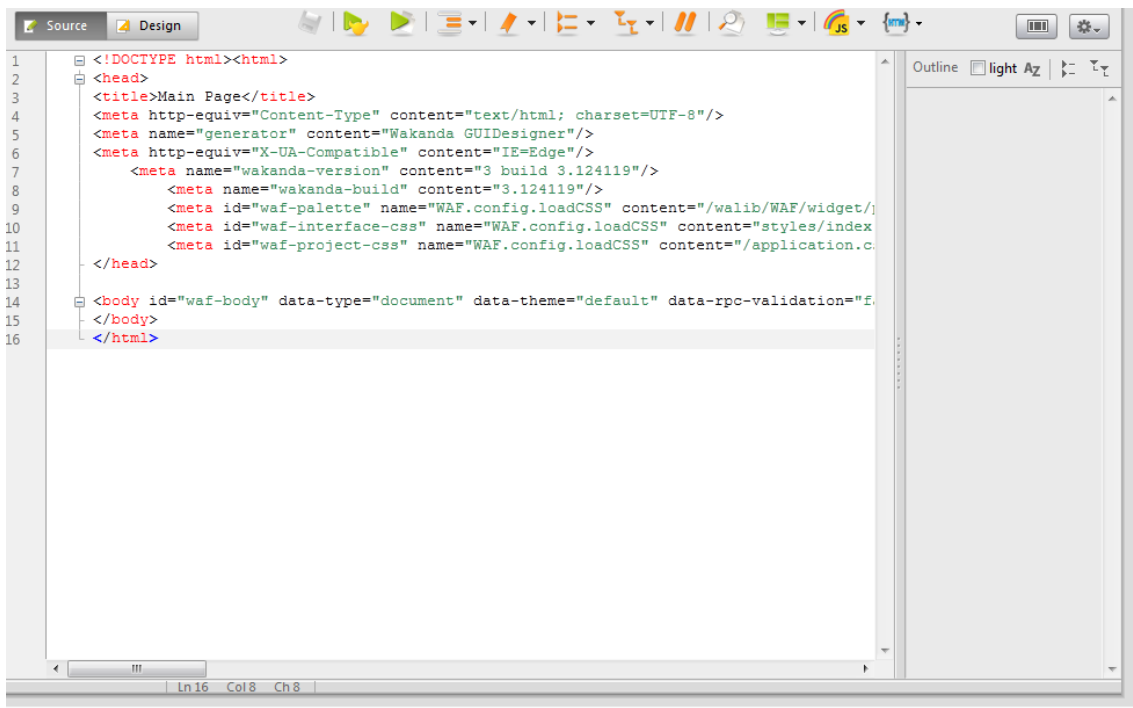




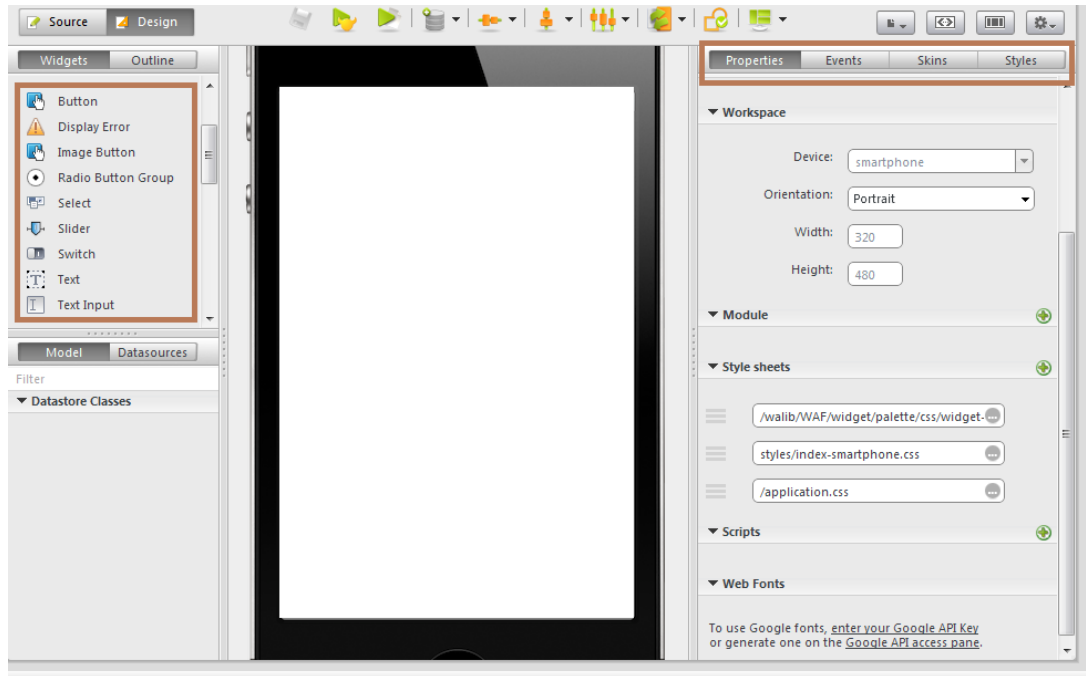
Y si se selecciona la opción Smartphone, aparece de la siguiente manera:



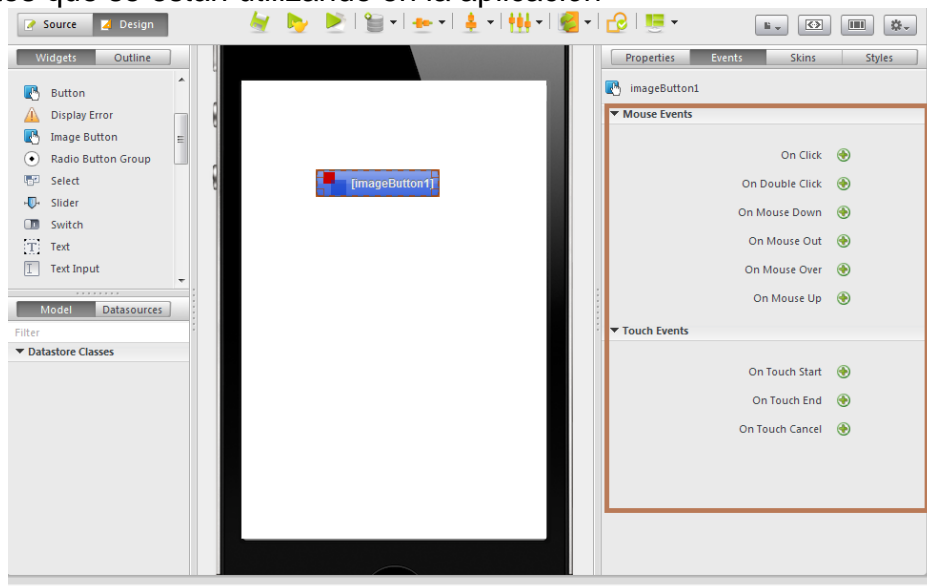
En la opción Source de cada página, se encuentra el código HTML del aplicativo:



En la opción Design, se encuentra el diseño de toda la interfaz, como lo son los Backgrounds, los botones, las cajas de texto, entre otros, es decir, en esta opción, se encuentran todas las herramientas para crear la interfaz del aplicativo:

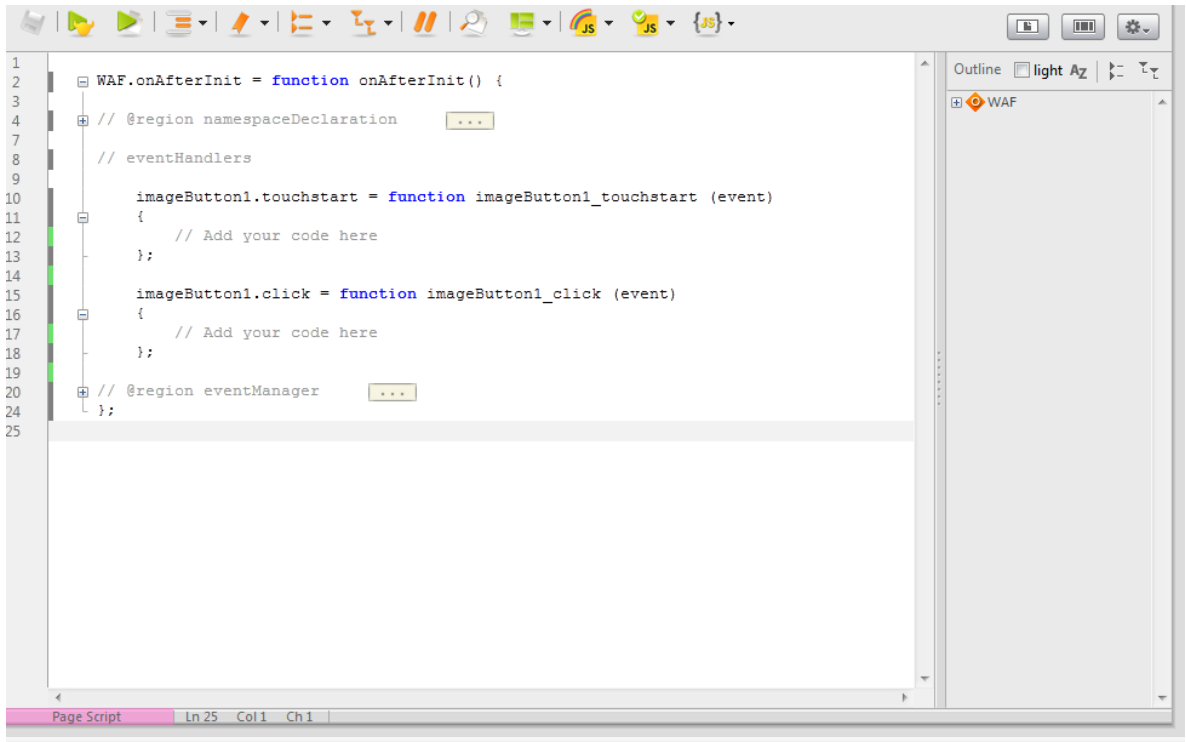


En la parte derecha, se encuentran las propiedades, los eventos y los estilos de los objetos que se están utilizando en la aplicación



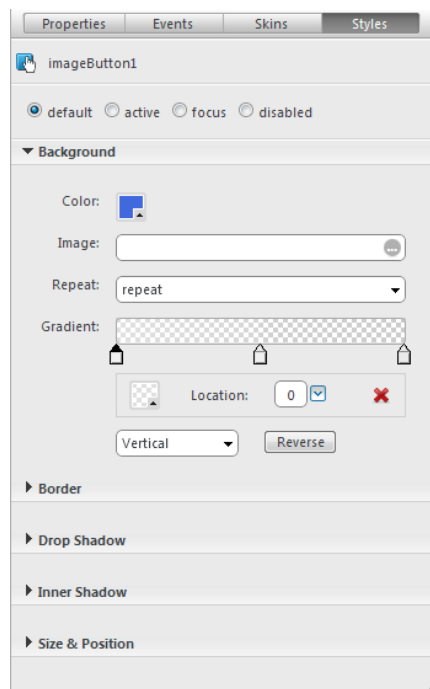


Al añadir un nuevo evento se crea un script:

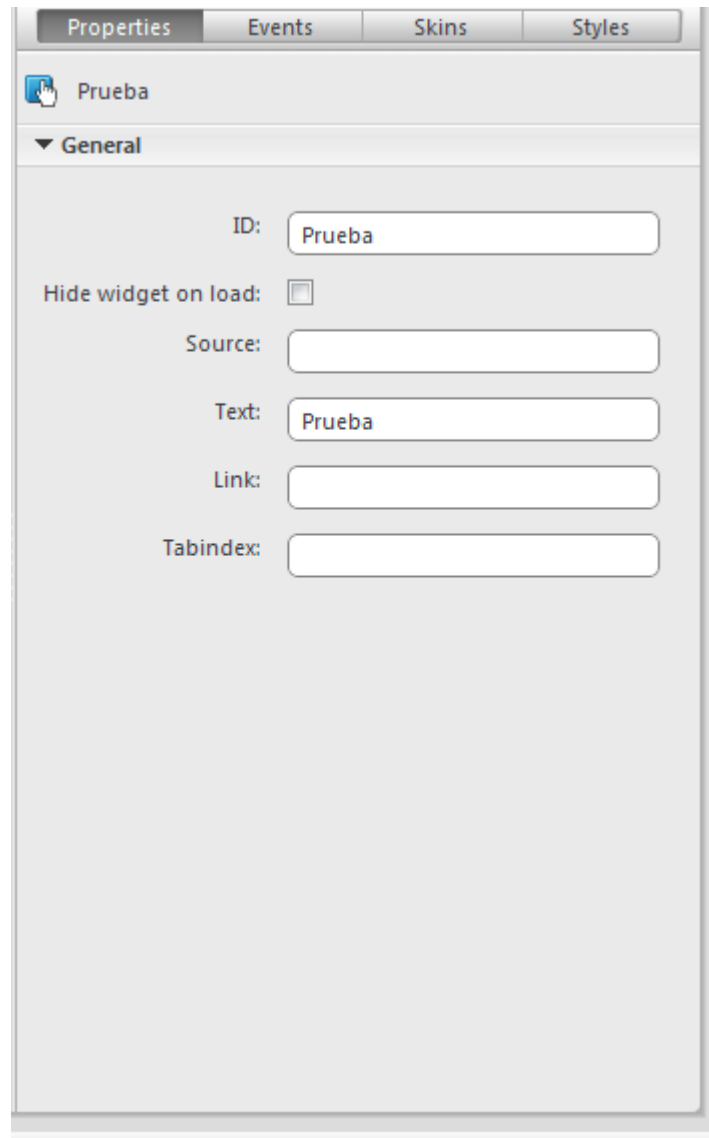


```
1 WAF.onAfterInit = function onAfterInit() {
2
3
4 // @region namespaceDeclaration ...
5
6
7 // eventHandlers
8
9
10 imageButton1.touchstart = function imageButton1_touchstart (event)
11 {
12 // Add your code here
13 };
14
15 imageButton1.click = function imageButton1_click (event)
16 {
17 // Add your code here
18 };
19
20 // @region eventManager ...
21 };
22
23
24
25
```

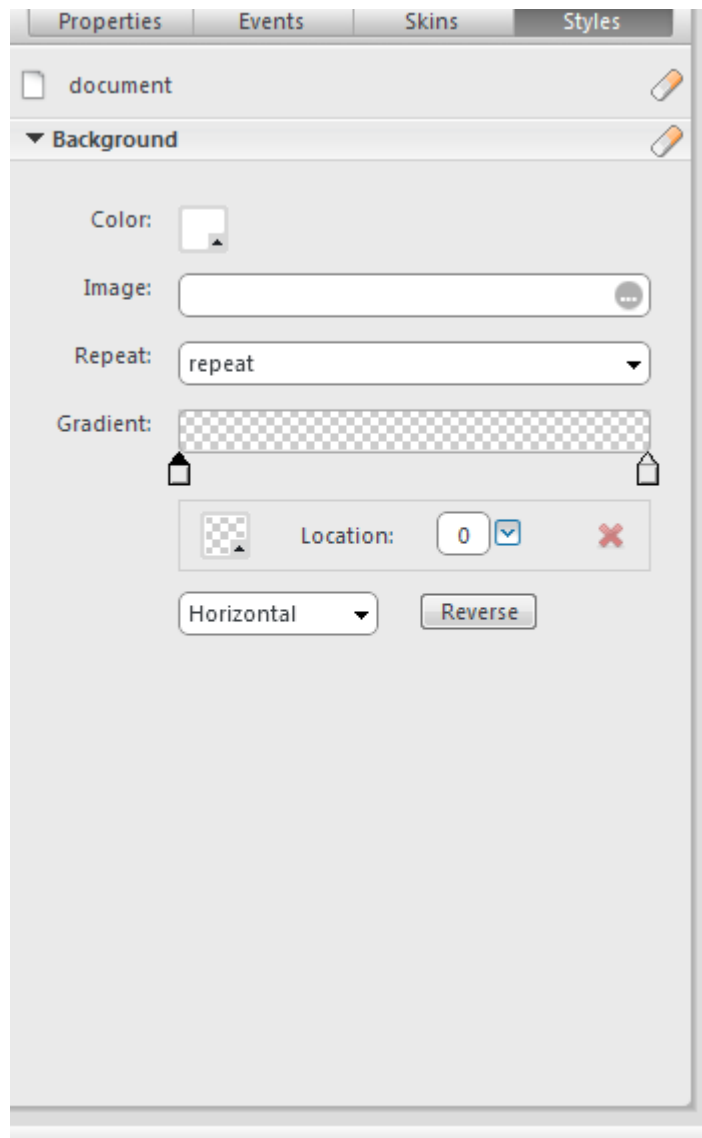
A los objetos, se les puede cambiar el fondo, el color, la imagen, para ello, se debe dar clic en la pestaña Style:



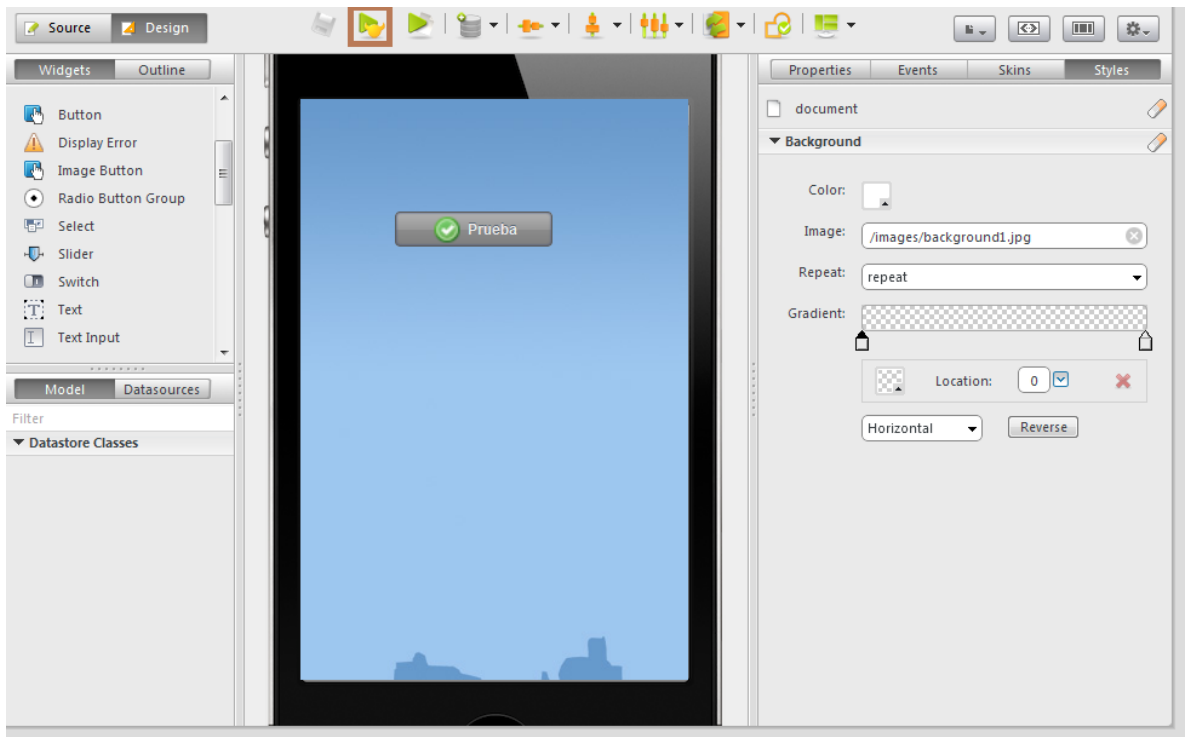
Para cambiarle, la etiqueta y el nombre al objeto, se debe ubicar en la pestaña properties:



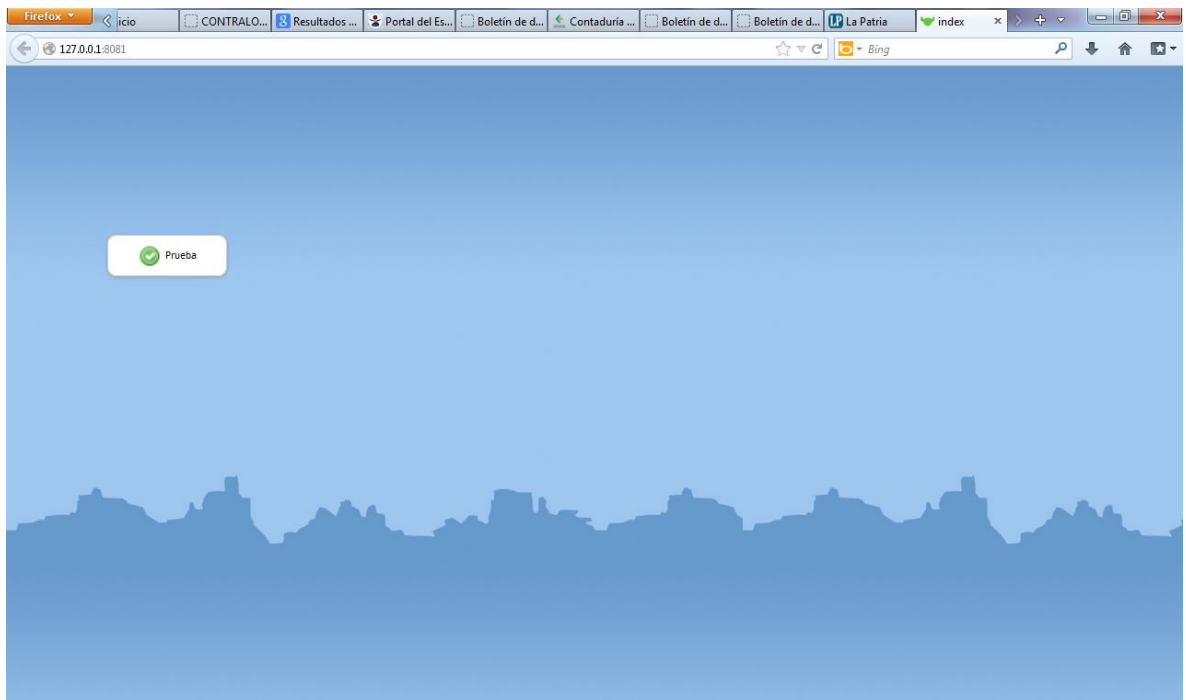
Para cambiar el fondo de la aplicación, se da clic en él, posteriormente se ubica en Styles y por último se selecciona la imagen que se desea:



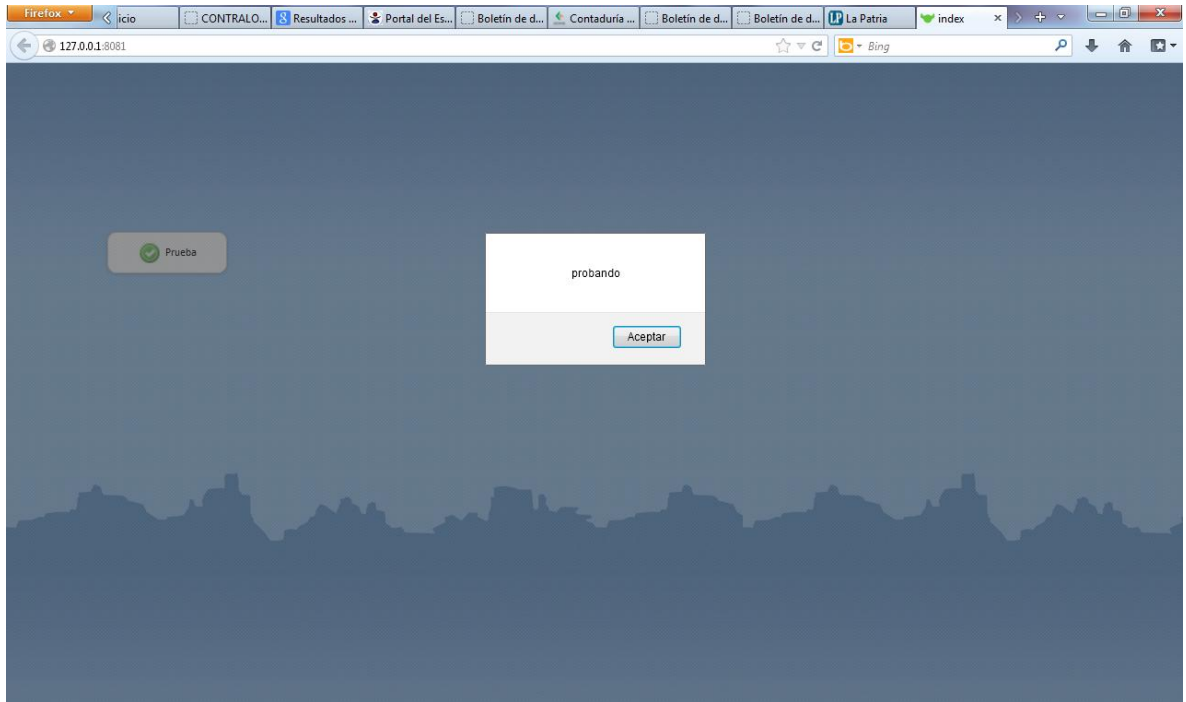
Cuando ya se ha creado la aplicación, y se desea ejecutarla, se debe dar clic en el icono run:



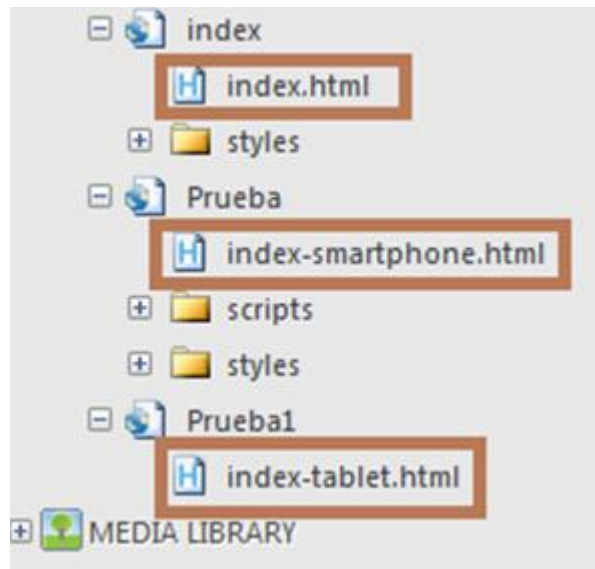
Al ejecutar el proyecto debe aparecer algo similar a la siguiente página:



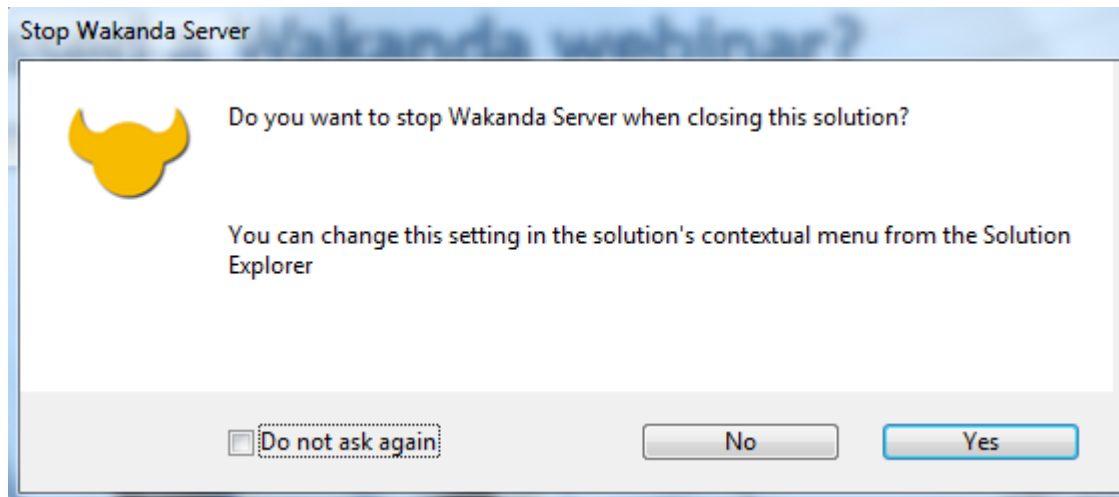
Al dar clic en el botón, se ejecuta el siguiente alert:



Sabe aclarar que toda página en Wakanda, crea una página HTML llamada index, index-smartphone o index-tablet, todo dependiendo del tipo de aplicación que se decidió crear:



Cuando ya se ha finalizado el proyecto, y se desee salir de Wakanda, se deben cerrar todas las ventanas, al momento de cerrar Wakanda Studio, este va a preguntar si desea parar el Wakanda server, en esta ventana, se selecciona la opción si, y se finaliza el proyecto:



### 2.3.2 WAKANDA STUDIO

Este es el editor visual de Wakanda, el cual es un editor gráfico, que permite almacenamiento de datos, una interfaz gráfica y un editor de código. Wakanda Studio, permite diseñar las clases de almacenamiento de datos al igual que la interfaz de usuario, proporcionando siempre la posibilidad de manejar los aspectos de seguridad.

Para desarrollar la interfaz gráfica, Wakanda cuenta con HTML5 y CSS3, del mismo modo que cuenta con la opción de arrastrar y soltar widgets.

### 2.3.3 WAKANDA SERVER

Wakanda Server se compone de dos partes, un almacén de datos y un servidor HTTP. En otras palabras *“Wakanda Server es un entorno de múltiples subprocesos, centrada en los datos JavaScript (JS) en tiempo de ejecución, que incluye WakandaDB, un NoSQL objetos del almacén de datos basado en modelos, y un servidor HTTP para aplicaciones Web basadas en la nube escalables.”*(4D, Wakanda, 2013).

El almacén de datos de Wakanda server, está dividido en varias partes. Una parte del almacén de datos, es el encargado de almacenar todos los datos de la aplicación y los modelos, el cual está dirigido por WakandaDB. WakandaDB es un NoSQL, donde su Datamodel se compone de DataClasses, donde se definen los atributos y métodos de los objetos, al igual que utiliza el código JavaScript que se ha escrito en Wakanda Studio.

El servidor HTTP de Wakanda, se encarga de ejecutar todo lo que se estableció en Wakanda Studio.

### **2.3.4 LENGUAJE DE PROGRAMACIÓN**

Se puede hablar de 2 instancias en cuanto a la programación en Wakanda, la primera instancia se trata de la interfaz, la cual se crea utilizando HTML5 y CSS3. Y la segunda instancia es la programación como tal de la aplicación, donde se define su funcionalidad y acciones a realizar; esta instancia utiliza JavaScript.

### **2.3.5 VENTAJAS**

- Facilidad de crear interfaces agradables con el usuario.
- Implementación de aplicaciones web y móviles.
- Construcción de aplicaciones nativas para móviles.
- Construcción de widgets.

### **2.3.6 DESVENTAJAS**

- Imposibilidad de conectar con hardware externos.
- No posee módulos para crear aplicaciones más complejas.
- Problemas al crear las aplicaciones nativas para los móviles.

## **2.4 HTML5**

HML5, como su nombre lo dice es la quinta versión de HTML, donde se integra el HTML y el XHTML, se creó con el fin de implementar aplicaciones más operables, identificándolo así como un candidato fuerte para el desarrollo de aplicaciones móviles multiplataforma, capaces de funcionar en Smartphones y Tablets.

HTML5, también se escribe en forma de etiquetas, y esta nueva versión, brinda la posibilidad de estructurar el aplicativo, estableciendo secciones y lo que es cada sección, eliminando el DIV, utilizado anteriormente en HTML que dividía la aplicación en secciones. Las webs, se pueden dividir en los siguientes elementos con las siguientes etiquetas:

- `<section>...</section>`: Esta etiqueta, es utilizada para representar un sección general por así decirlo dentro de la aplicación, puede tener subsecciones, lo que permite crear jerarquías de contenidos, lo que ayudaría al posicionamiento web de la página.
- `<article>...</article>`: Esta etiqueta representa una parte autónoma de la página, que puede ser reutilizada y repetida.
- `<aside>...</aside>`: Esta etiqueta se utiliza para representar un contenido de la página independiente del contenido general de la misma.
- `<header>...</header>`: Esta etiqueta representa la parte introductoria de la página o el aplicativo.
- `<nav>...</nav>`: Esta etiqueta es utilizada para crear links de navegación dentro de la página, ya sea a otra página o secciones del documento.
- `<footer>...</footer>`: esta etiqueta representa el pie de página de una sección o página, y contiene información de la página o la sección, como lo es el autor, el año, el copyright, entre otros.

La siguiente imagen, representa las principales diferencias entre HTML y HTML5:



Los formularios, son unos de los elementos más utilizados en las páginas web, HTML5, añade mejoras al elemento *input type*, ampliando la variedad de tipos que se pueden tener, entre estas mejoras, se destacan:

- `<input type="search">`: Esta etiqueta se utiliza para crear cajas de búsqueda.



- `<input type="number">`: Esta etiqueta se utiliza para adicionar o restar números mediante botones.
- `<input type="range">`: Esta etiqueta se utiliza para seleccionar un valor entre dos valores predeterminados.
- `<input type="color">`: Esta etiqueta se utiliza para seleccionar un color.
- `<input type="tel">`: Esta etiqueta se utiliza para números telefónicos.
- `<input type="url">`: Esta etiqueta se utiliza para direcciones web.
- `<input type="email">`: Esta etiqueta se utiliza para direcciones de email.
- `<input type="date">`: Esta etiqueta se utiliza para seleccionar un día en un calendario.
- `<input type="month">`: Esta etiqueta se utiliza para seleccionar meses.
- `<input type="week">`: Esta etiqueta se utiliza para seleccionar semanas.
- `<input type="time">`: Esta etiqueta se utiliza para seleccionar fechas.
- `<input type="datetime">`: Esta etiqueta se utiliza para seleccionar una fecha exacta.
- `<input type="datetime-local">`: Esta etiqueta se utiliza para seleccionar fechas locales.

Otros elementos que cabe destacar de HTML5 son:

- *Modernizr*: Es una librería de JavaScript, que permite determinar que elementos de HTML5 son compatibles con el navegador. Cuando se ejecuta, crea un objeto global que recibe el mismo nombre y le añade propiedades booleanas, dependiendo si es aceptado por el navegador o no. Un ejemplo del código utilizado para ejecutar esta librería es:

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Dive Into HTML5</title>
  <script src="modernizr.min.js"></script>
</head>
<body>
  ...
</body>
</html>

```

- *Canvas: HTML5*, define este elemento como un rectángulo en la página donde por medio de JavaScript, se puede dibujar lo que se desee.
- *Video*: Esta etiqueta (`<video>`), como su nombre lo dice permite integrar video en la página web, sin la necesidad de que el navegador tenga que identificar ningún script, esto debido a que se especifican múltiples ficheros de video y los navegadores escogen uno basado en el formato que soporten.
- *Texto como PLACEHOLDER*: Es un nuevo atributo de *HTML5* que se utiliza dentro de los campos input, cuando el campo está vacío o no está señalado, ya que cuando se hace clic dentro del campo, el texto desaparecerá. Un ejemplo de *PLACEHOLDER* es el siguiente:



El siguiente código, muestra un ejemplo de cómo se puede introducir un *PLACEHOLDER* en un formulario:

```

<form>
  <input placeholder="Buscar en la base de datos">
  <input value="Buscar">
</form>

```

## 2.5 JAVASCRIPT

JavaScript, es un lenguaje de programación interpretado orientado a objetos, el cual es implementado como parte de un navegador web, permitiendo mejoras en la interfaz de usuario, al igual que páginas web más dinámicas. También se usa en documentos PDF y widgets. Cabe aclarar que Java y JavaScript, no están relacionados ya que poseen semánticas y propósitos diferentes.

A continuación, se destacan los usos más comunes de JavaScript:

- Cargar nuevo contenido a la página sin necesidad de recargarla.
- Animación de los elementos que posee la página, los cuales se pueden hacer desaparecer, cambiar su tamaño, moverlos, entre otros.
- Manejo de contenido interactivo, como lo son juegos, videos y audio.
- Validación de formularios web, lo que permite comprobarlos antes de enviarlos.
- Mostrar mensajes de alerta, notificación, entre otros, sin necesidad de abandonar la página en la cual se encuentra el usuario.

Una de las ventajas que posee JavaScript, es que sus desarrolladores lo estandarizaron, lo que permite que pueda ser ejecutado de una manera correcta en la mayoría de los navegadores, en caso de que no sea ejecutado correctamente en algún navegador, tiene la capacidad de comprobar ciertas funcionalidades, de ahí que una misma característica se comporte de manera diferente en 2 navegadores.

## **2.6 CSS3**

Las hojas de estilo en cascada, son utilizadas para dar estilos a páginas web, CSS3, es una nueva versión de CSS, y está dividido en varios módulos, que permiten desarrollar webs más elaboradas y más dinámica, permitiendo tener un control mucho más preciso de la página. La novedad más importante que ofrece CSS3, es la incorporación de nuevos mecanismos para mantener un mayor control sobre el estilo que muestran y manejan las páginas web.

Algunas de las novedades de las propiedades que tiene esta nueva versión de las hojas de estilo en cascada son:

- Bordes
- Fondos
- Color
- Texto

- Interfaz
- Selectores
- Modelos de cajas básicos
- Degradados
- Creación de múltiples columnas de texto
- Animaciones

## **2.7 JQUERY**

JQuery, es una librería de JavaScript que permite una interacción más fácil con HTML, manejando eventos, desarrollando animaciones, agregando también una interacción con la técnica AJAX. Entre las características principales se tienen:

- Selección de elementos DOM
- Soporte y manipulación para CSS (1-3)
- Promueve la captura de eventos
- Efectos y animaciones
- Animaciones personalizadas
- AJAX
- Soporta extensiones
- Obtiene información del navegador
- Maneja objetos y vectores
- Compatible con la mayoría de navegadores
- Permite cambiar el contenido de la página sin necesidad de recargarla
- Ahorra muchas líneas de código

## 2.8 PROCESSING

Processing es un lenguaje de programación basado en Java, al estar basado en java tiene la posibilidad de heredar todas las funcionalidades que java ofrece. Inicialmente este lenguaje de programación, se creó para servir de prototipo en la creación de proyectos de software, y de esta manera enseñar los fundamentos de programación de una forma visual; actualmente Processing se ha convertido en una herramienta de desarrollo.

Entre las características que caben destacar de este lenguaje de programación, se encuentran las siguientes:

- Descarga gratuita y de código abierto.
- Posibilidad de crear programas con salida 2D, 3D o PDF.
- OpenGL integración de aceleración 3D.
- Distribución bajo licencia GNU.
- Posee más de 100 bibliotecas que amplían las posibilidades del software.
- Existe variedad de documentación.

Processing se puede ejecutar tanto en Windows, como Mac OS y Linux, donde las pruebas más extensivas sobre estabilidad y rendimiento se hacen sobre Windows y Mac OS. Como se mencionó anteriormente en una de las características de este lenguaje, Processing trabaja bajo la licencia GPL.

La licencia publica general de GNU, es la licencia más utilizada en el mundo del desarrollo de software, debido a que garantiza al usuario final del software que tiene total libertad de usar, estudiar, compartir y modificar el software, lo que indica que el software que es creado bajo esta licencia es software libre.

El logo de Processing es el siguiente, aunque dependiendo del tipo de proyecto a realizar y al entorno de desarrollo este puede cambiar:



### **2.8.1 PROCESSING PARA ARDUINO**

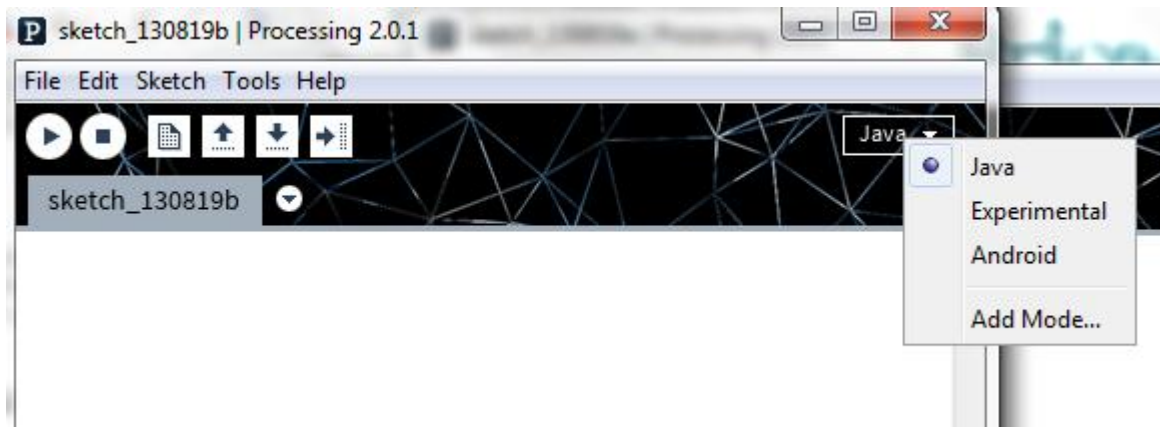
Arduino es una plataforma que permite la creación de prototipos fáciles de usar, a partir de la programación de un microcontrolador que posee la placa Arduino. Este microcontrolador es programado mediante el lenguaje de programación Arduino, el cual está basado en Wiring; pero como todo lenguaje de programación, este también necesita un entorno de desarrollo, el cual está basado en Processing.

Al entorno de desarrollo de Arduino estar basado en Processing, se crea una afinidad entre ambos lenguajes, hasta el punto en que se pueden comunicar entre sí. Processing es una herramienta que permite crear interfaces intuitivas para el usuario, lo que permite tener una experiencia más cerca sobre Arduino, ya que por medio de este se pueden visualizar datos obtenidos por Arduino, al igual que mandar órdenes al microcontrolador. Y todo a través de una interfaz intuitiva para el usuario.

Al programar en processing, se puede hacer uso del puerto serial, el cual es la comunicación de Arduino, por medio de este puerto es que se realiza la conexión entre Processing y Arduino.

### **2.8.2 PROCESSING PARA ANDROID**

En el entorno de desarrollo de Processing, se puede seleccionar el tipo de desarrollo que se desea realizar, teniendo la opción de elegir entre el modo Java, modo Experimental y modo Android:



Al desarrollar aplicaciones en el modo Android, estas pueden ser ejecutadas en cualquier teléfono que maneje este sistema operativo. Las aplicaciones se crean de igual manera como se crea otro programa en Processing, donde la única restricción que presenta este modo es el uso de algunas librerías, como es la librería serial que si se encuentra en el modo Java.

### 3. ARDUINO

Arduino, es una plataforma de hardware y software libre, flexible y fácil de usar, creado para cualquier persona que desee crear proyectos con entornos y objetos interactivos. Arduino, se basa en una placa con un microcontrolador, al igual que un entorno de desarrollo basado en processing. Existen diferentes placas, entre las que se encuentran Arduino UNO, Arduino BT, Arduino Cookbook, Arduino Due, Arduino Duemilanove, Arduino Ethernet, Arduino FIO, entre otros.

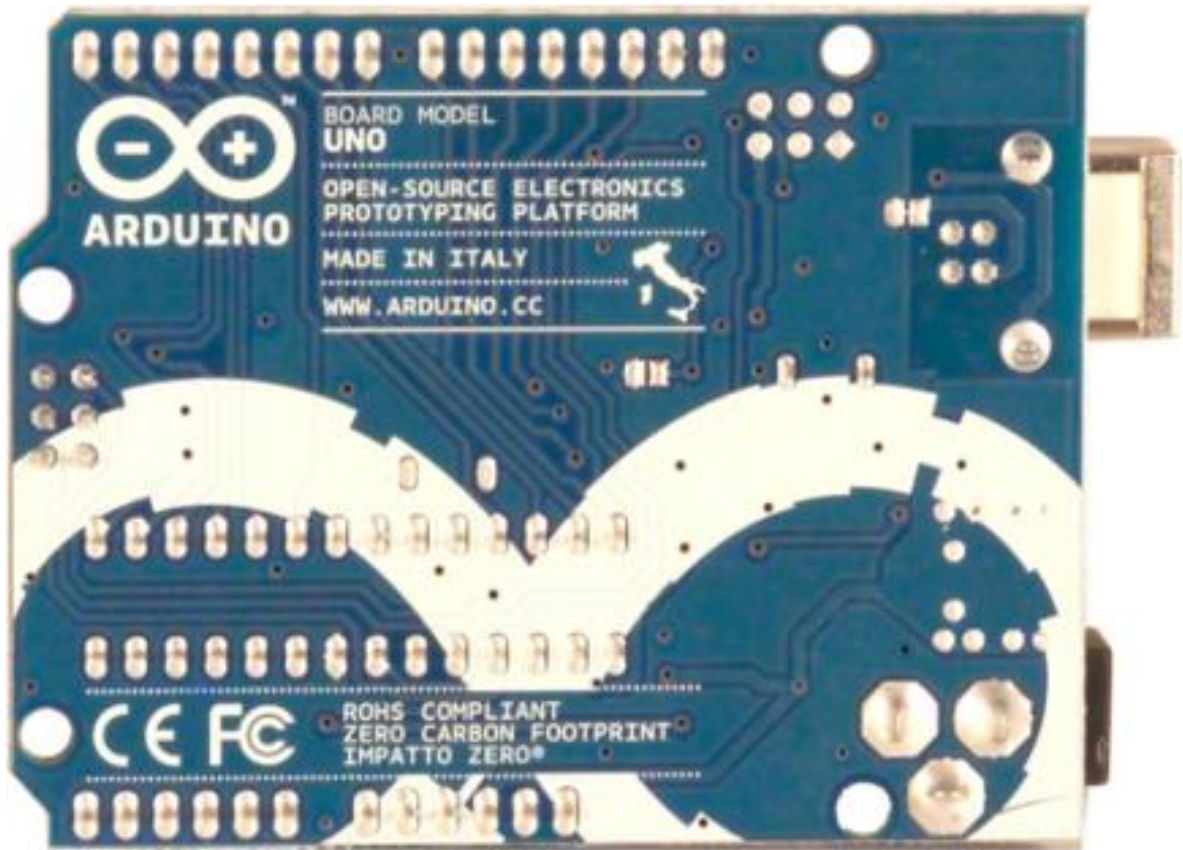
Esta placa posee 3 tierras y un botón de reinicio, una entrada de alimentación y conexión USB.

A continuación se muestra la vista frontal y vista posterior de la placa:



*Vista frontal*





*Vista posterior*

Arduino UNO, funciona con un voltaje de 5V, y proporciona este mismo voltaje de salida, cuenta con una memoria de 32 KB, de los cuales, 0,5 son utilizados por el gestor de arranque, 2 corresponden a SRAM 1 de EEPROM.

### **3.1 PINES DE ENTRADA Y SALIDA**

El Arduino UNO, tiene 14 pines digitales de entrada o salida, de los cuales, 6 se pueden utilizar como PWM, al igual que cuenta con entradas digitales, también cuenta con 6 entradas analógicas.

La siguiente es una descripción del uso y funcionalidad de cada pin que posee esta placa:

- VIN: Este pin funciona como voltaje de entrada a la placa, cuando se pretende utilizar una fuente de alimentación externa.
- 5V: Este pin, da una salida de 5V, es poco recomendado su uso, ya que no pasa por un regulador y puede dañar la placa.

- 3,3V: Este pin, genera una salida de 3,3V.
- GND: Estos son los pines de tierra.
- 0 (RX) y 1 (TX): Estos pines se pueden utilizar para recibir o transmitir datos series TTL.
- 2 y 3: Estos pines pueden ser utilizados para activar una interrupción en un valor bajo, o un cambio de valor.
- 3, 5, 6, 9, 10 y 11: Estos pines identificados con el símbolo ~, proporcionan una salida PWM de 8 bits.
- 10, 11, 12 y 13: Estos pines apoyan la comunicación SPI. SPI, es un protocolo de datos en serie síncrono que utilizan los microcontroladores para comunicarse con uno o más dispositivos en distancias cortas, al igual que es utilizado para establecer la comunicación entre 2 microcontroladores.
- 13: En este pin se encuentra conectado un LED, que cuando el pin está en un valor alto, se mantiene encendido, y cuando está en un valor bajo, se mantiene apagado.
- AREF: Este pin se utiliza como voltaje de referencia para las entradas analógicas.
- Reset: Este pin se utiliza para reiniciar el microcontrolador.

## 3.2 BIBLIOTECAS

Las bibliotecas de Arduino, o más bien conocidas como librerías, son las encargadas de dar más funcionalidad a las aplicaciones que se creen. Hay librerías que vienen junto con Arduino, pero otras que no, por lo que se deberán instalar.

Entre las bibliotecas estándar que contiene Arduino, cabe destacar:

- *EEPROM*: Esta librería sirve para leer y escribir en la memoria de forma permanente.
- *Ethernet*: Permite conectarse a internet por medio del módulo Ethernet Shield.

- *Firmata*: Permite comunicarse con otros lenguajes de programación por medio del uso de un protocolo.
- *LiquidCrystal*: Permite controlar LCD.
- *Servo*: Esta librería permite controlar Servomotores.
- *SoftwareSerial*: Permite la comunicación serial de cualquier pin digital.
- *Stepper*: Permite controlar motores paso a paso.
- *Wire*: Esta librería permite enviar y recibir datos a través de una red de dispositivos y sensores

Existen gran variedad de librerías que se pueden instalar en Arduino, todo dependiendo de la funcionalidad que se desee obtener, como lo puede ser:

- Comunicación
- Sensores
- Displays y LEDs
- Generación de frecuencias y Audio
- Motores y PWM
- Medición de tiempo
- Utilidades

### **3.3 MÓDULOS**

La ventaja de este microcontrolador, es la gran variedad de proyectos que se pueden realizar, al igual que la variedad de módulos que se pueden integrar con este microcontrolador, entre los que cabe destacar:

- Adaptador mini USB



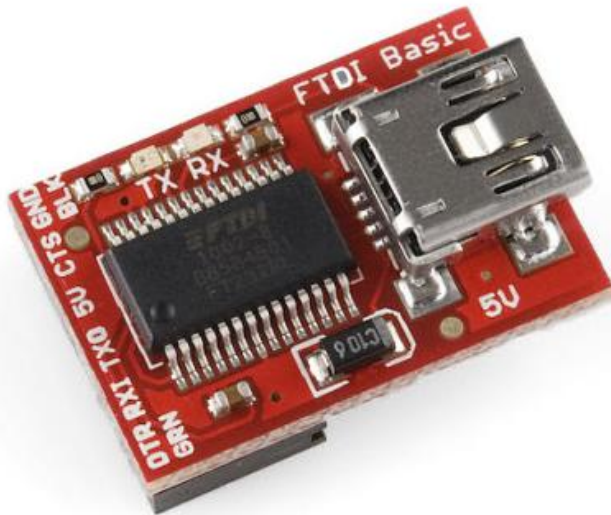
- Bluetooth:



- Celular



- Serial:



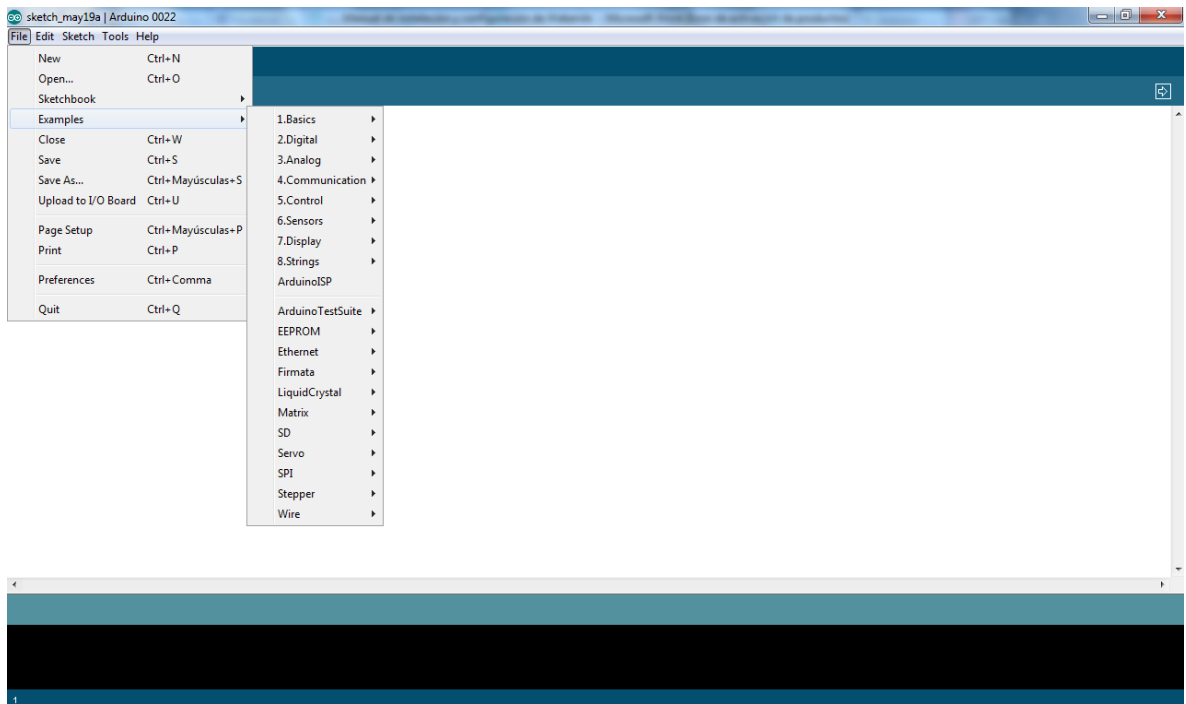
- Wifi



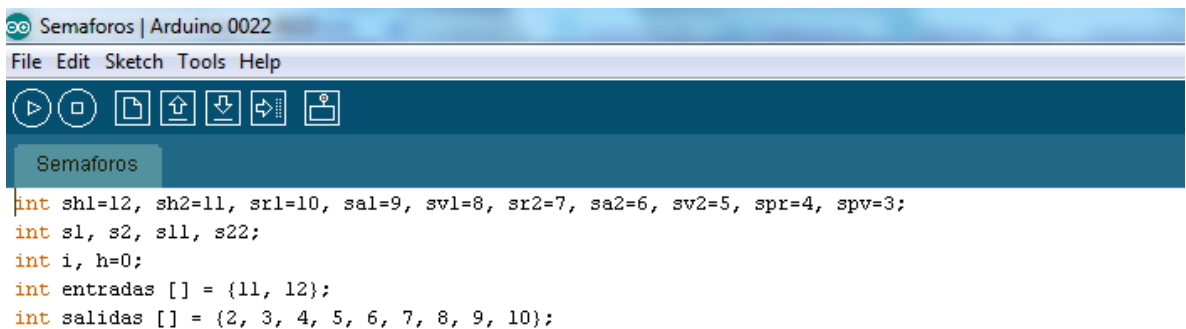
### 3.4 LENGUAJE DE PROGRAMACIÓN

Dependiendo de la versión de Arduino que se utilice, se descarga el software para programar de la siguiente dirección: <http://arduino.cc/en/Main/Software>, cuando se descargue el archivo, este se descomprime en el escritorio, para tener un acceso más fácil y cómodo. Después de descargado el software, se procede a la instalación de los drivers de la placa, que es diferente en cada sistema operativo y en cada versión. Luego de esto, se procede a la programación.

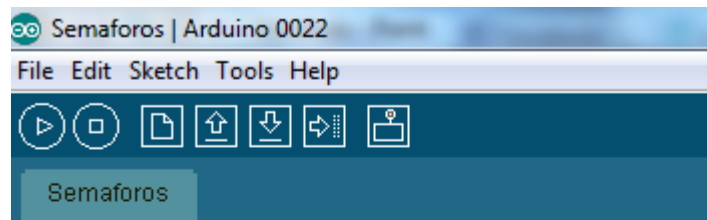
El software de Arduino, posee diferentes tipos de ejemplos para las personas principiantes, al igual que para personas que desean conocer el manejo, utilización e implementación de las diferentes librerías que posee Arduino.



Lo primero que se hace cuando se va a crear un programa, es declarar las variables, en este caso se puede dar nombre a cada pin que se va a utilizar, al igual que se declaran las banderas, los sensores, switches, botones o demás componentes a utilizar junto con las entradas y salidas definidas, estas últimas en vectores.



Después de tener definidas las variables a utilizar se procede a crear el programa, donde hay que tener en cuenta que se divide en 2 secciones void setup y void loop. En la sección void setup, se asignan los pines, se inicializa la comunicación en serie, y también se declaran variables.



```
void setup (){\n  for (i=0; i<=1; i++){  
    pinMode (entradas[i], INPUT);  
  }  
  for (i=0; i<=7; i++){  
    pinMode (salidas[i], OUTPUT);  
  }  
}
```

El termino pinMode, define si el pin se va a utilizar como entrada (INPUT) o salida (OUTPUT), y lleva la siguiente sintaxis: pinMode (NumeroPin, INPUT/OUTPUT).

En la sección void loop, se escribe el código que se ejecuta continuamente, es decir se leen las entradas, se activan las salidas, se llaman funciones etc. Cuando se lee una entrada, su valor se guarda en una de las variables declaradas con anterioridad, la sintaxis para leer una entrada digital es la siguiente: variable=digitalRead(Numeropin);

```
s1 = digitalRead(s1);
```

Las funciones se crean de la siguiente manera:

```
tipoFuncion Nombre (parámetros) {\n  Sentencias;\n}
```



```

void semaforop (int estado){
  if (estado == 1){
    digitalWrite (spr, HIGH);
    digitalWrite (spv, LOW);
  }
  else{
    if (estado == 2){
      digitalWrite (spr, LOW);
      digitalWrite (spv, HIGH);
    }
    else {
      digitalWrite (spr, LOW);
      digitalWrite (spv, LOW);
    }
  }
}

```

Y se llaman con la siguiente sintaxis: NombreFuncion (parámetros);

```
semaforop (2);
```

Como en todos los lenguajes de programación, Arduino posee también la opción de generar comentarios, los comentarios de línea se generan con // y los bloques de comentarios con /\*...\*/. Los operadores de comparación, y los operadores lógicos utilizados, son los siguientes:

- == : Es igual a...
- != : Es diferente a...
- < : Menor que...
- > : Mayor que...
- <= : Menor o igual que...
- >= : Mayor o igual que...
- && : AND Lógico
- || : OR Lógico

En este lenguaje de programación se manejan ciertas constantes, para determinar entradas, salidas, valores altos y bajos, estas constantes son:

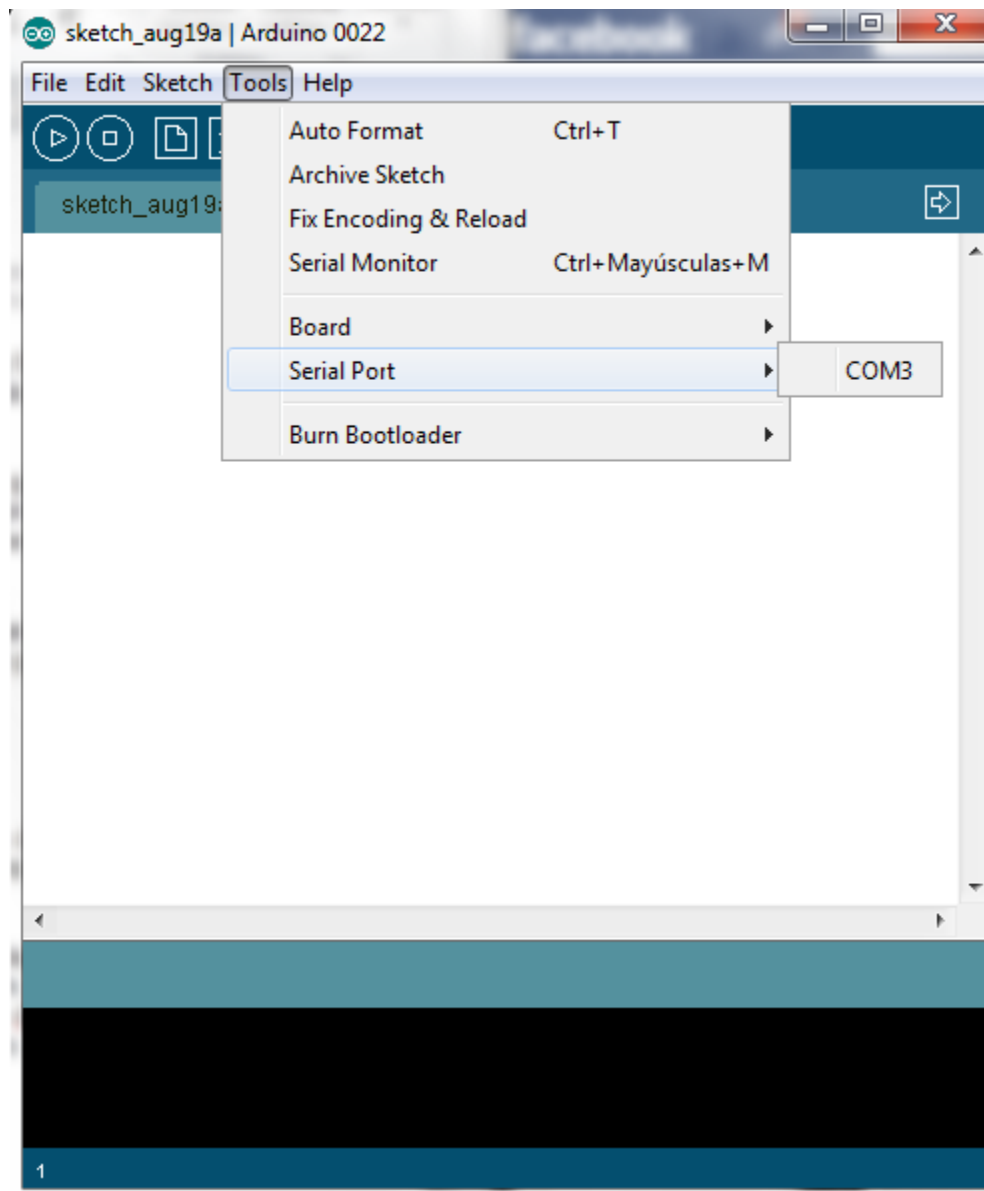
- *TRUE/FALSE*: Estas constantes son booleanas y brindan valores lógicos, un valor *FALSE*, significa 0, mientras que un valor *TRUE* significa 1 o un valor diferente de 0.
- *HIGH/LOW*: Estas constantes son las que definen qué valor tiene un pin, se usan cuando se lee o se escribe un valor en un pin digital. Un valor *HIGH*, está definido como un 1 lógico, un on o simplemente 5V; y un valor *LOW*, es el valor lógico 0, un off o un 0V.
- *INPUT/OUTPUT*: Como se había dicho anteriormente, estas constantes son utilizadas con la función *pinMode* para definir un pin digital como entrada (*INPUT*) o salida (*OUTPUT*).

### **3.5 INSTALACIÓN**

Para la correcta instalación de la placa Arduino, se debe entrar a la página principal de Arduino y descargar la versión que corresponde a la placa que se posee.

Posteriormente se debe conectar la placa Arduino, dependiendo del sistema operativo los drivers se instalaran automáticamente, si los drivers no se encuentran, lo que se procede a hacer es buscar los drivers en los archivos anteriormente descargados desde la página de Arduino.

Cuando los drivers están instalados, se procede a seleccionar el puerto serial con el que trabajar la placa y por medio del cual se subirán los programas a la placa:



### 3.6 APLICACIONES

Arduino ofrece diversidad de aplicaciones, todo depende de lo que se pretenda realizar, ya que a partir de sensores, se pueden realizar desde prototipos hasta proyectos más elaborados que inclusive pueden ser utilizados en la industria.

## 4. ANDROID

Android es un sistema operativo para dispositivos móviles, basado en Linux. Está especialmente diseñado para teléfonos inteligentes y tabletas, que poseen pantalla táctil. La ventaja de Android, es que permite tener todas las aplicaciones e ingeniería de Google.

### 4.1 HISTORIA

Inicialmente, Android fue desarrollado por Android Inc., bajo el apoyo económico de Google; aunque en 2005 fue adquirida por Google. En 2007 fue presentado junto al Open Handset Alliance, este último, es un consorcio de compañías de hardware, software y telecomunicaciones, que se encargan de desarrollar estándares para dispositivos móviles. El primer dispositivo móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008.

Las características principales del HTC Dream, eran que incorporaba una pantalla táctil, al igual que funciones completas de conectividad y navegabilidad.

Ha sido tal la aceptación de Android en el mercado que en el cuarto trimestre de 2011, alcanzó una cuota de mercado del 50,9%, más que el doble que tenía el sistema operativo iOS de Apple, su principal competencia.

### 4.2 CARACTERÍSTICAS

Entre las características que caben destacar de este sistema operativo se tienen:

- Pantalla adaptable a pantallas de mayor resolución.
- Biblioteca de gráficos 2D y 3D.
- Almacenamiento a través de SQLite, la cual es una base de datos liviana.
- Soporta las siguientes tecnologías de conectividad:
  - GSM/EDGE
  - IDEN
  - CDMA

- EV-DO
  - UMTS
  - Bluetooth
  - Wi-Fi
  - LTE
  - HSDPA
  - HSPA+
  - WiMAX
- Navegador Web basado en WebKit.
  - Soporte Java.
  - Soporte de distintos formatos multimedia
  - Soporte para Streaming
  - Soporte para cámaras de fotos y videos, al igual que depantallas táctiles, GPS, acelerómetros giroscopios, magnetómetros, sensoresdeproximidad y de presión, sensores de luz, GamePad, termómetro y aceleración por GPU 2D y 3D, entre otros.
  - Incluye herramientas de análisis del rendimiento del sistema.
  - Catálogo de múltiples aplicaciones a través de Google Play.
  - Soporte multi-táctil.
  - Soporte de Video llamada.
  - Búsqueda de contenido a través de voz

### **4.2.1 VERSIONES**

A partir de su lanzamiento en 2007, Google ha sacado distintas versiones del sistema operativo Android, cabe destacar que cada versión recibe el nombre de un postre en inglés, y cada nombre sigue un orden alfabético:

- Apple Pie – Versión 1.0
- Banana Bread – Versión 1.1
- Cupcake – Versión 1.5
- Donut – Versión 1.6
- Éclair – Versión 2.0 y 2.2
- Froyo – Versión 2.2
- Gingerbread – Versión 2.3
- Honeycomb – Versión 3.0, 3.1 y 3.2
- Ice Cream Sandwich – Versión 4.0
- Jelly Bean – Versión 4.1, 4.2 y 4.3

### **4.2.2 USOS**

Este sistema operativo tiene gran variedad de usos debido a la versatilidad que ofrece. Esta versatilidad, permite que no solo pueda ser usado en teléfonos inteligentes o en tabletas, sino también en computadores portátiles, televisores, relojes de mano, auriculares, entre otros.

Android, puede ser utilizado por dispositivos que contengan arquitectura ARM, aunque también puede ser utilizado por hardware x86, pero estos usos ya dependen de versiones especiales de este sistema operativo.

### **4.2.3 DESARROLLO**

Ha sido tal éxito de Android, que hoy en día es considerado un modelo de negocios, debido a que en su desarrollo posee factores estratégicos, que ha dado pie a que muchos desarrolladores de tendencias y negocios de alto impacto opten por seguir este estilo.

Android a diferencia de otros sistemas operativos se desarrolla de forma libre, es decir se puede acceder a l código fuente de forma gratuita, todo con el fin de encontrar y resolver problemas, mejorando la estabilidad de dicho sistema, este descubrimiento lo puede hacer cualquier persona interesada, y esto es lo que marca la diferencia con otros sistemas operativos.

### **4.3 APLICACIONES**

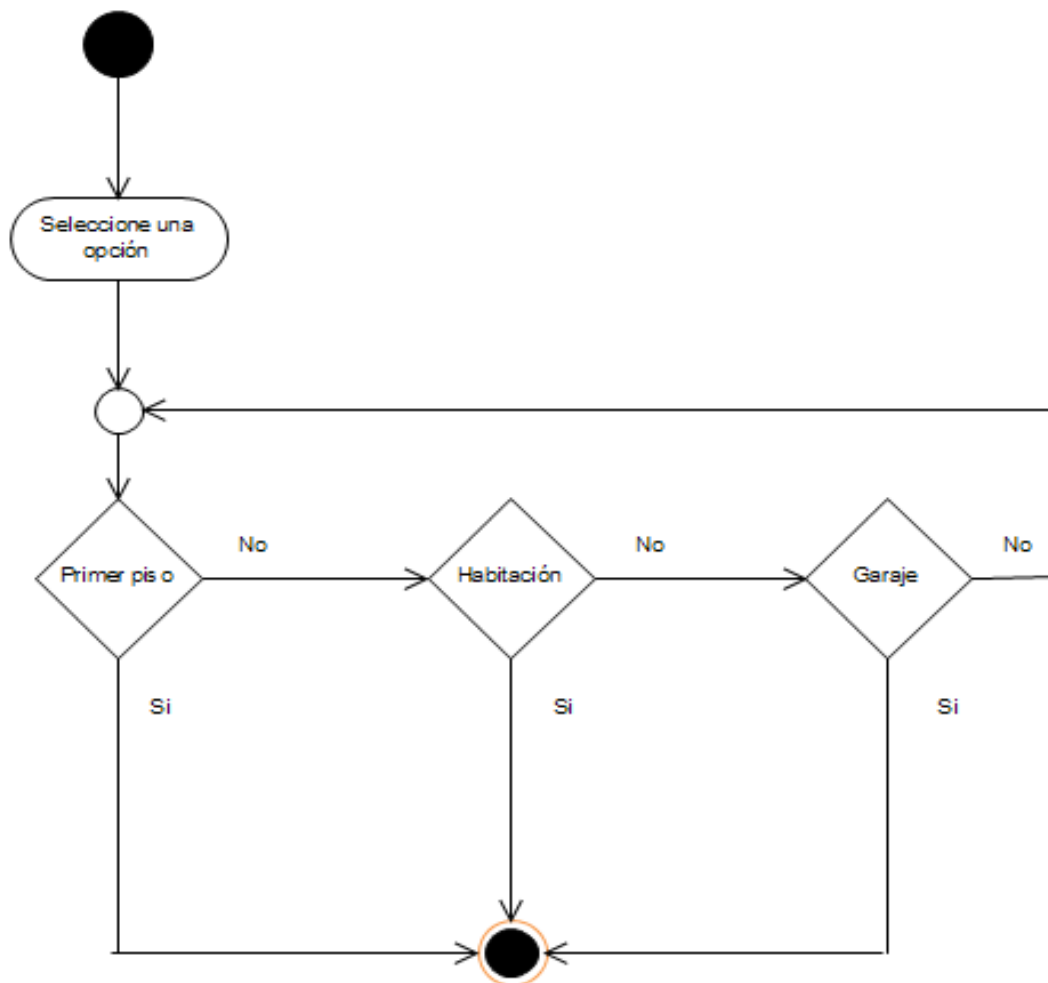
Existen gran variedad de herramientas para desarrollar aplicaciones para Android, aunque comúnmente estas aplicaciones se desarrollan en Java. Cabe aclarar que toda aplicación desarrollada para Android debe estar comprimida en formato APK, para ser instaladas de la manera más sencilla en los dispositivos

## 5. ANÁLISIS Y DISEÑO

Como todo desarrollo de software requiere de un análisis y un diseño, al ser este un prototipo de control domótico el único diagrama UML que se realizó fue el diagrama de flujo. Se optó por realizar este diagrama, ya que permite unificar el desarrollo del software con el funcionamiento del microcontrolador, teniendo una perspectiva paso a paso de lo que realizara el microcontrolador al recibir la orden desde el software.

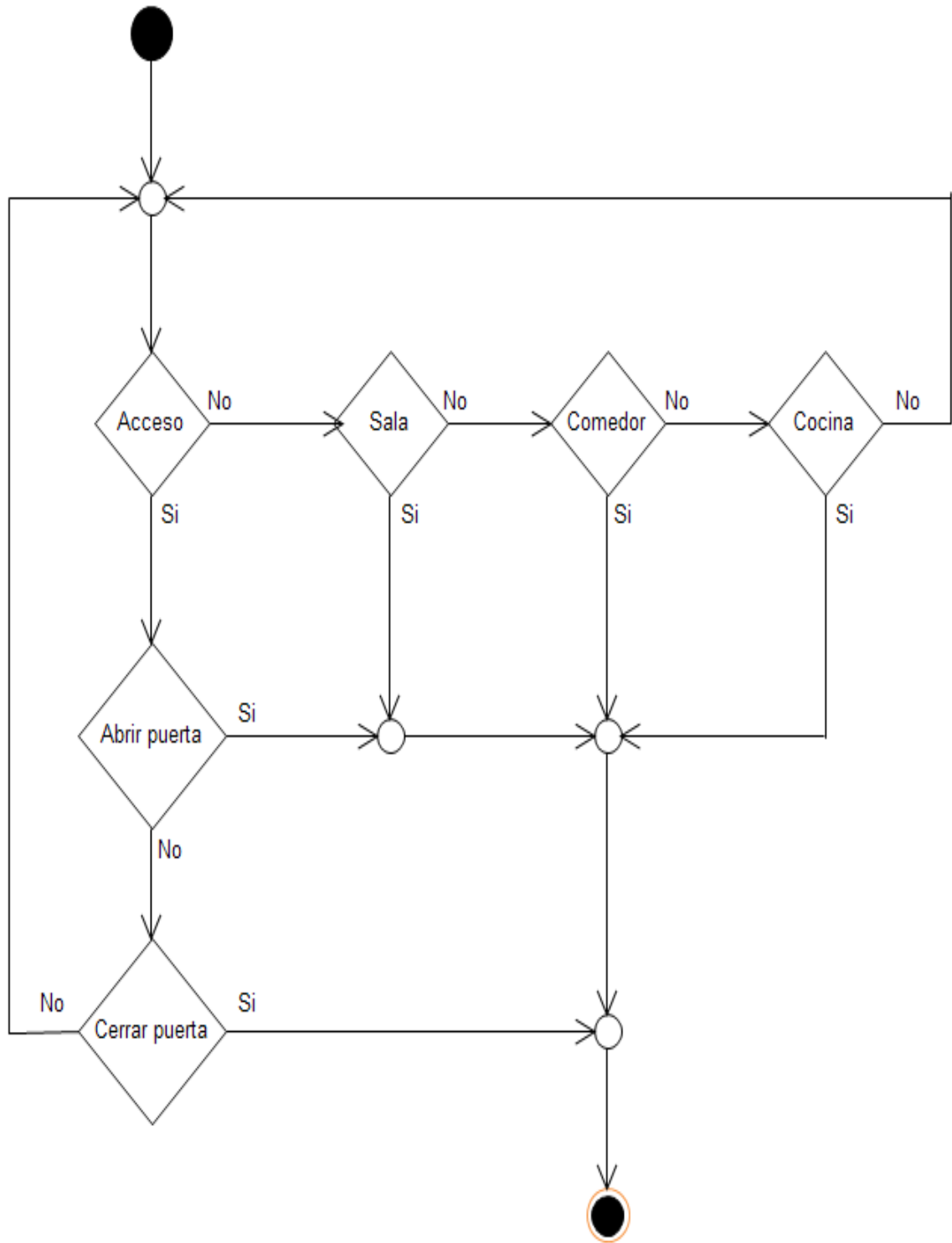
### 5.1 DIAGRAMA DE FLUJO

Debido a la estructura del proyecto y el tamaño del software, se deciden realizar varios diagramas de flujo. El primer diagrama consta de la selección del lugar de la casa al que se desea acceder:

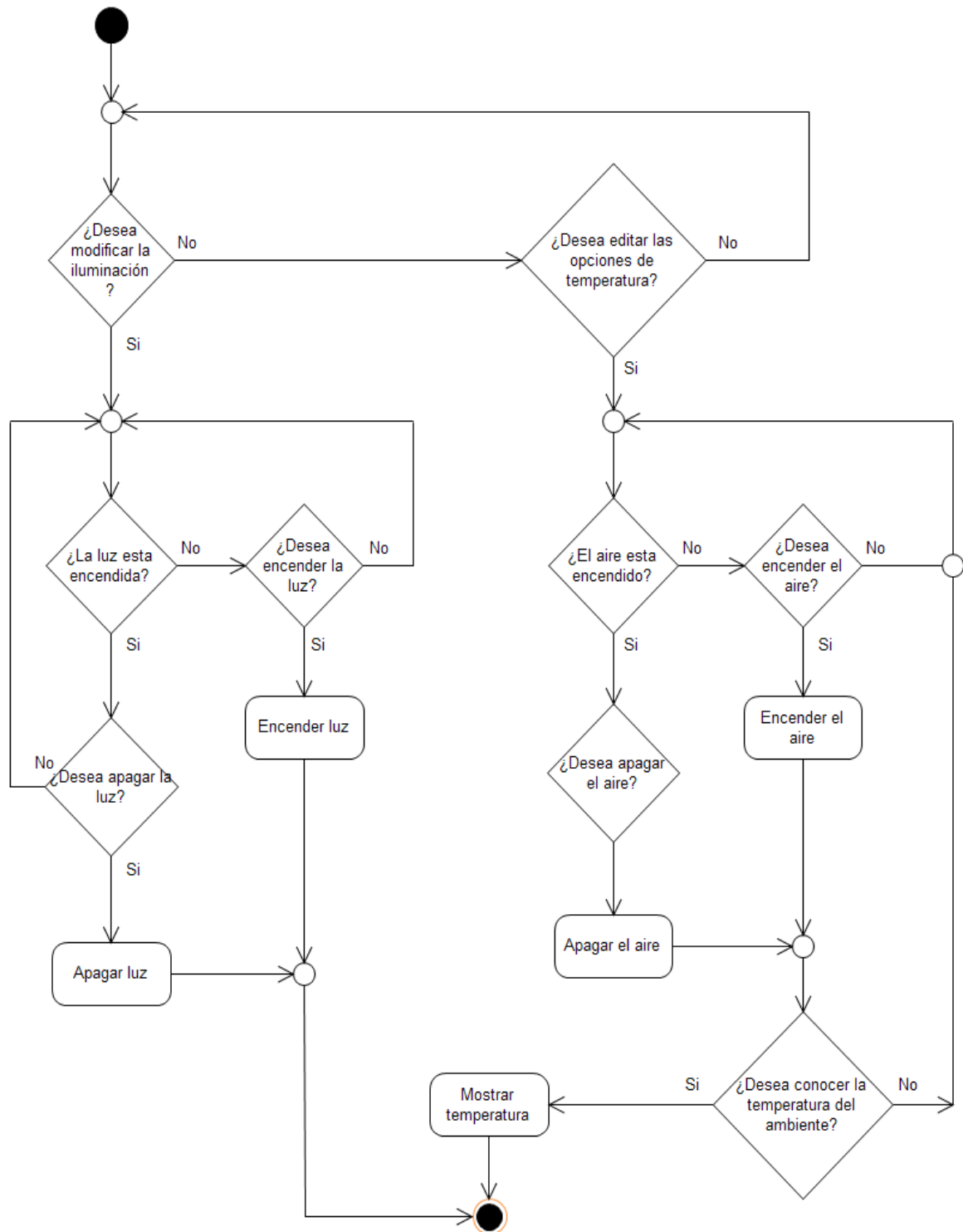




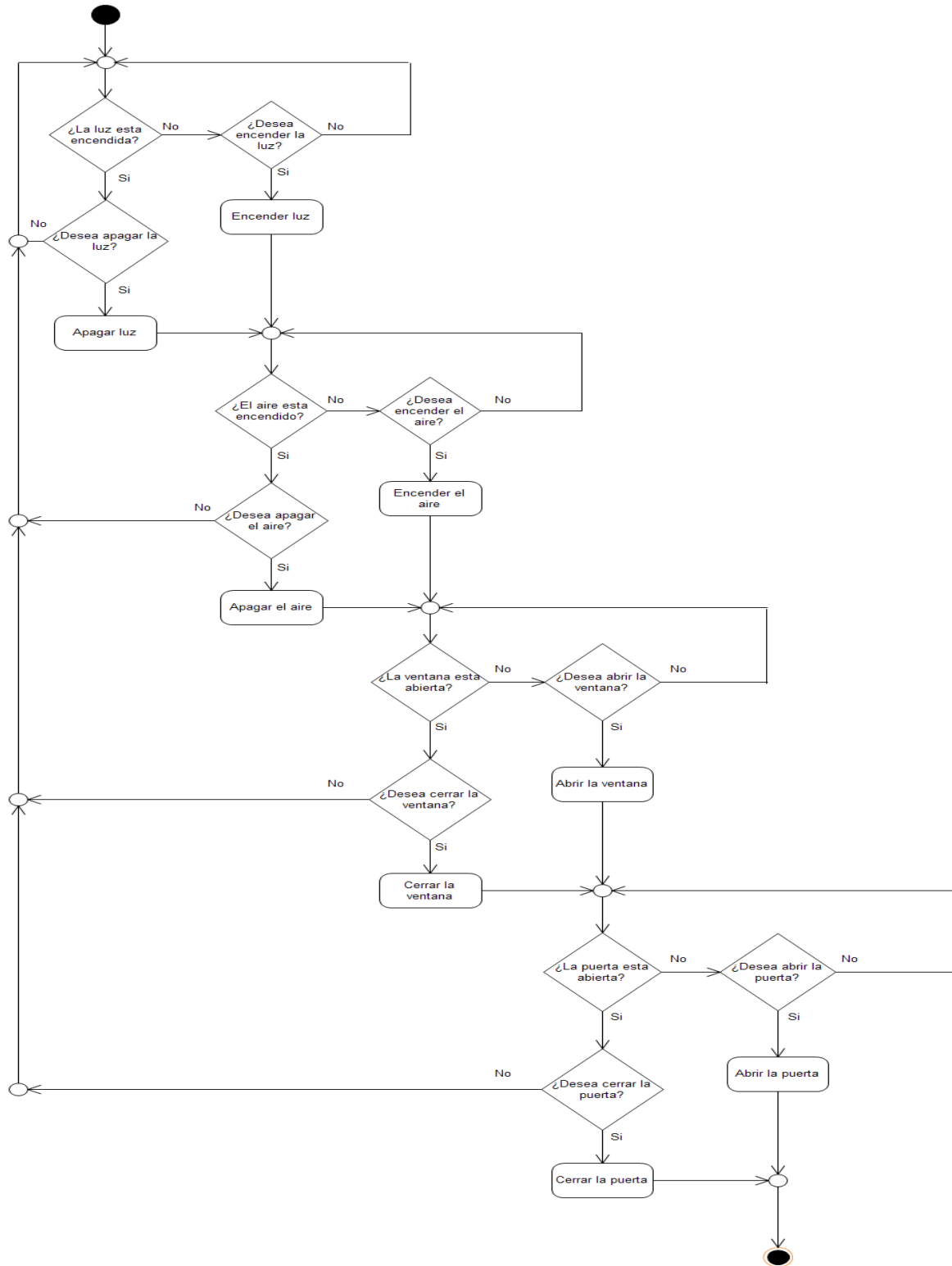
El diagrama de flujo para las opciones del primer piso, es el siguiente:



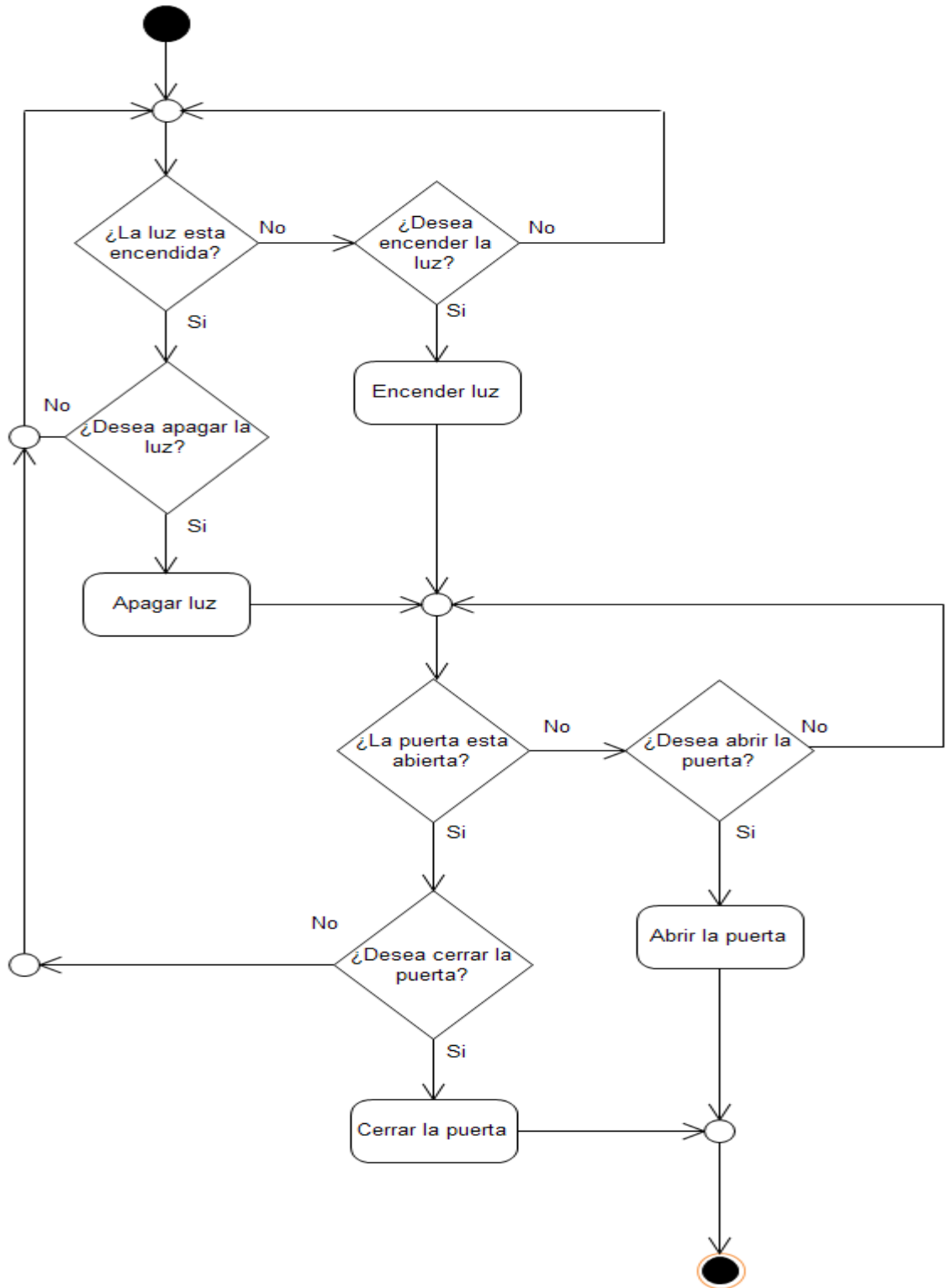
El diagrama de flujo que rige tanto la sala como el comedor y la cocina, es el siguiente:



El diagrama de flujo para la habitación, es el siguiente:



El diagrama de flujo para el garaje es el siguiente:



## 6. IMPLEMENTACIÓN EN WAKANDA

La implementación en Wakanda del prototipo de control domótico, presentaba grandes ventajas, debido a la estabilidad y versatilidad que ofrece esta plataforma, pero en el momento en que el desarrollo llevaba un grado avanzado y se deseaba implementar algunas conexiones, surgieron una serie de inconvenientes que hicieron que esta plataforma se descartara y no se utilizara, como se tenía previsto inicialmente.

A continuación se procederá a hablar del desarrollo de la aplicación como tal, de la conexión con Arduino, y de los resultados obtenidos con esta herramienta.

### 6.1 DESARROLLO DE LA APLICACIÓN

La aplicación cuenta con una pantalla de inicio, la cual tiene una selección de 3 opciones, primer piso, habitación y garaje:



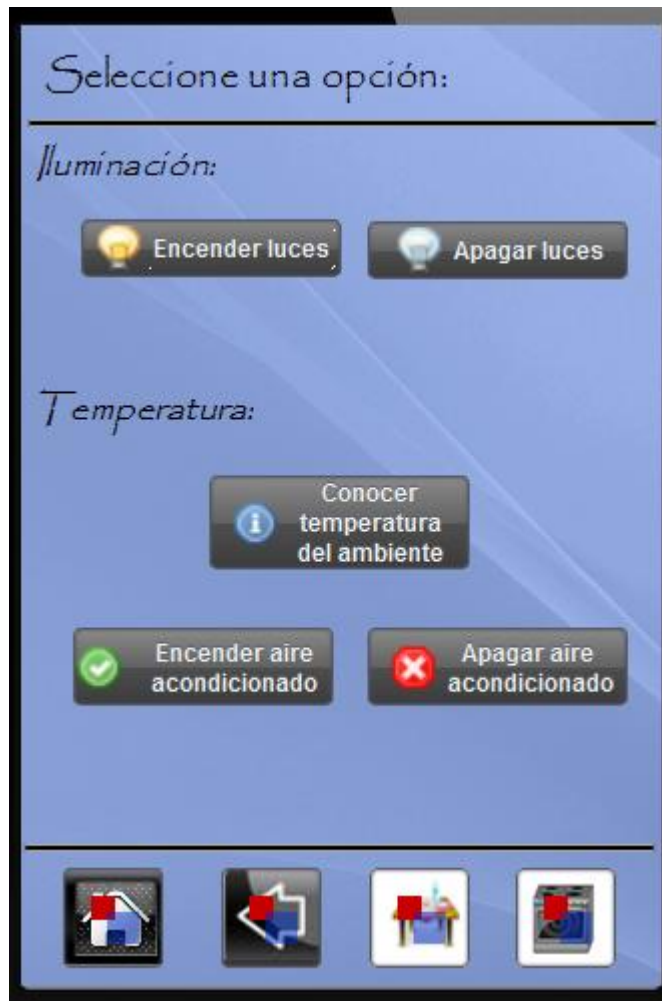
- *Primer piso*

Al seleccionar la opción del primer piso, se mostrara la siguiente interfaz:



En esta interfaz, lo que se puede observar, es una opción de acceso, la cual permite escoger entre abrir y cerrar la puerta. En la parte inferior de la interfaz, hay una serie de botones, los cuales permiten la navegación entre los espacios del primer piso, al igual que el regreso a la página inicial del aplicativo. El botón con el icono de una casa, envía directamente a la página principal de la aplicación, y aparece en todas las interfaces del aplicativo, excepto la de inicio.

El botón con el icono de un mueble, permite ingresar a la sala. Al seleccionar este botón, se mostrara una interfaz que posee las opciones de iluminación y temperatura, en las cuales se podrá decidir si se desea encender o apagar las luces y el aire acondicionado, al igual que permite determinar si se desea conocer la temperatura del ambiente. La interfaz es la siguiente:



El icono con la flecha de atrás, indica la opción de retornar a la página de inicio del primer piso. El botón con el icono de un comedor, permitirá el ingreso a las opciones del comedor. Al seleccionar este botón, se muestra una interfaz que posee las opciones de iluminación, donde se puede decidir si se desea encender o apagar la luz. La interfaz es la siguiente:



El botón con el icono de una estufa, permitirá el ingreso a las opciones de la cocina. Al seleccionar este botón, se muestra una interfaz que posee las opciones de iluminación y ventilación, donde se puede decidir si se desea encender o apagar la luz, al igual que si se desea encender o apagar el aire acondicionado. La interfaz es la siguiente:





- *Habitación*

Al seleccionar la opción de habitación, se mostrara la siguiente interfaz:



En la anterior interfaz, aparecen opciones para la iluminación, la temperatura, la vista y el acceso; al igual que el botón para retornar al inicio de la aplicación. En la opción de iluminación se puede seleccionar entre apagar y encender las luces, en la opción de temperatura, se puede determinar entre encender y apagar la calefacción. En la Vista, se puede seleccionar si se desea abrir o cerrar la ventana, y en acceso puede determinar si desea abrir o cerrar la puerta.

- *Garaje*

Al seleccionar la opción de garaje, se mostrara la siguiente interfaz:



Esta interfaz está dividida en 3 partes: acceso, iluminación, e inicio. En la parte de acceso, se puede determinar si se desea abrir o cerrar la puerta; en la opción de iluminación, se puede determinar si se desea encender o apagar la luz; y en la parte de inicio, se tiene la posibilidad de retornar a la página principal de la aplicación.

## 6.2 CONEXIÓN CON ARDUINO

La conexión entre Wakanda y Arduino, se hace por medio de JavaScript, en JavaScript se debe definir el puerto que utilizara Arduino para mandar las acciones, cada que se seleccione una opción.

En Arduino, se establece la programación de los pines, teniendo en cuenta los datos que se envían desde el programa en realizado en JavaScript.

### **6.3 RESULTADOS**

La conexión entre JavaScript y Arduino, no presento mayores inconvenientes, por ese lado, la aplicación funcionaria correctamente desde un pc. Los inconvenientes se presentaron al momento de establecer la conexión, por medio de una herramienta externa, en este caso el Bluetooth, que era la herramienta que se tenía previsto en un inicio para el desarrollo del proyecto.

En la documentación de Wakanda no se encontraba información de cómo establecer conexiones con herramientas como lo es el Bluetooth, limitando la implementación total del proyecto; por tal motivo, se terminó por descartar esta herramienta y utilizar otra que permita el uso de Bluetooth, para así poder tener un control domótico, a través de un dispositivo móvil, sin necesidad de un pc como intermediario.

## **7. IMPLEMENTACIÓN EN PROCESSING**

Se decidió optar por Processing para Android como herramienta de desarrollo, debido en primer lugar a la tendencia actual del mercado por dispositivos con este sistema operativo, y en segundo lugar por la versatilidad que ofrece este lenguaje de programación para desarrollar diversos tipos de aplicaciones, al igual que la posibilidad de implementar diferentes herramientas.

Para poder desarrollar una aplicación en Processing para Android, primero se debe desarrollar una aplicación para Processing y probar la compatibilidad que presenta con Arduino.

A continuación se procederá a hablar del desarrollo de la aplicación como tal, de la conexión con Arduino, y de los resultados obtenidos con esta herramienta.

### **7.1 DESARROLLO DE LA APLICACIÓN**

La implementación de la aplicación en Processing, es la misma implementación que se realizó en Wakanda, donde el funcionamiento de la aplicación y la estructura de las interfaces, se puede encontrar en la página 69 del presente documento.

### **7.2 CONEXIÓN CON ARDUINO**

La conexión entre Processing y Arduino, se hace por medio del puerto Serial, en ambos programas, se establece la utilización de este puerto. En el programa hecho en Processing, se establecerá una línea de código, que envíe determinado dato por el puerto serial, con el fin de establecer el estado de algún pin del Arduino.

En el programa realizado en el entorno de desarrollo de Arduino, se establecerá una línea de código que lea el valor del puerto serial, y dependiendo de este valor se procederá a enviar una orden a alguno de los pines del Arduino.

### **7.3 RESULTADOS**

La conexión entre Processing y Arduino, no presento mayores inconvenientes, donde por medio de unas cortas líneas de código, se establece una conexión exitosa entre ambos.

A partir de una implementación, y una conexión exitosa entre Processing y Arduino, se establece la viabilidad de desarrollar una aplicación para Android y conectarla con Arduino por medio de Bluetooth.

## **8. IMPLEMENTACIÓN EN PROCESSING PARA ANDROID**

Al tomar a Processing para Android como herramienta de desarrollo, se establece la funcionalidad del prototipo con este lenguaje de programación. La sintaxis cambia un poco de Processing (Java) a Processing (Arduino), debido a que algunos procesos cambian un poco la manera en realizarse, como lo es la inserción de imágenes, entre otros; otro factor importante que se destaca entre estas 2 funcionalidades de Processing, es que se pierden algunas librerías (Serial) y se agregan unas otras (Bluetooth).

A continuación se procederá a hablar del desarrollo de la aplicación como tal, de la conexión con Arduino, y de los resultados obtenidos con esta herramienta.

### **8.1 DESARROLLO DE LA APLICACIÓN**

La implementación de la aplicación en Processing, es la misma implementación que se realizó en Wakanda, donde el funcionamiento de la aplicación y la estructura de las interfaces, se puede encontrar en la página 69 del presente documento.

### **8.2 CONEXIÓN CON ARDUINO**

La conexión entre Processing para Android y Arduino, a diferencia de Processing y Wakanda, no se hace por medio del puerto serial, ya que en Processing para Android no está implementada esta opción. En este caso se hace por medio de Bluetooth.

Debido a que Arduino cuenta con un módulo de Bluetooth que se puede implementar, en el entorno de desarrollo de Arduino, se establece la acción a realizar sobre cada pin, dependiendo del dato que se reciba por Bluetooth.

Y como en Processing para Android, existe una librería para implementar el uso del Bluetooth, lo único que resta es establecer que dato se envía por medio del Bluetooth al Arduino dependiendo de la opción que se seleccione en la interfaz previamente establecida.

### **8.3 RESULTADOS**

La conexión entre Processing para Android y Arduino, no presento mayores inconvenientes, donde por medio de unas cortas líneas de código, se establece una conexión exitosa entre ambos.

Se enviaron datos desde el celular al Arduino por medio de Bluetooth, y la respuesta del Arduino en la activación o no activación de los pines se dio de manera exitosa.



## 9. APLICACIÓN FINAL

La aplicación final se desarrolló en Processing para Android, debido a que permitió la incorporación de la transmisión de datos por medio de Bluetooth. La interacción que permite Processing con los objetos, hace más interesante el desarrollo del proyecto, ya que permite un uso más cómodo y confiable para los usuarios.

La conexión entre Android y Arduino, se hace por medio del módulo Bluetooth de Arduino, esta conexión, es una conexión fiable y tiene una transmisión exitosa de datos.

Otro factor importante que vale la pena recalcar, es que la aplicación no necesita tener un instalador en el dispositivo móvil, ya que por medio del cable USB, se transmite la aplicación desde el IDE de Processing al dispositivo con Android, y posteriormente a esto se crea un ejecutable en el dispositivo, que permite el uso de la aplicación en cualquier momento, ya sin necesidad de tener el USB conectado al pc.

Entre las ventajas que cabe destacar, se encuentran:

- Fiabilidad en la transmisión de datos.
- Transmisión de órdenes sin necesidad de un intermediario físico.
- Interfaz cómoda y agradable para el usuario.
- Posibilidad de implementar otros módulos o funcionalidades al prototipo.
- Utilización de dispositivos móviles, independientemente de que sean Smartphone o Tableta.
- Versatilidad de la aplicación.
- Compatibilidad con la mayoría de Dispositivos móviles.
- Prototipo basado en las tendencias actuales del mercado.

## 10. CONCLUSIONES

- El desarrollo del prototipo de control domótico por medio de una aplicación para Android tuvo resultados exitosos, debido a las pruebas realizadas a Arduino a través de dispositivos móviles con sistema operativo Android.
- La recepción de datos por medio del módulo Bluetooth del Arduino se realizó de manera exitosa.
- El desarrollo de la aplicación móvil, se ejecutó de manera exitosa tanto en un Smartphone como en una Tablet.
- La implementación en Wakanda, presento varios inconvenientes, el más significativo y notorio, se evidencio, al momento de tratar de unificar una herramienta como el Bluetooth en el aplicativo.
- La documentación existente hasta el momento sobre Wakanda y sus herramientas, no permite el desarrollo de proyectos que integren herramientas externas. Este Framework de HTML5 es adecuado para el uso de aplicaciones Web que no requieran ningún tipo de transmisión de datos a través de herramientas como el Bluetooth.
- El control de alarmas, luces y motores por medio de la aplicación desarrollada se realizó de manera exitosa, ya que todas las ordenes enviadas a través de Bluetooth por este dispositivo, son recibidas y traducidas por el modulo Bluetooth del Arduino, permitiendo la correcta activación o desactivación de los pines.
- Los datos recibidos por medio del módulo Bluetooth del Arduino, procesan los datos recibidos, y permite la ejecución de las instrucciones establecidas con anterioridad en el microcontrolador.

## BIBLIOGRAFÍA

- [1] La domótica sigue ganando terreno. (30 de Enero de 2010). *EL Tiempo*.
- [2] *Wikipedia*. (08 de Agosto de 2013). Recuperado el Agosto de 2013, de Wikipedia.
- [3] 4D. (28 de Junio de 2012). *4D*. Obtenido de <http://www.4d.com/la/blog/4d-launches-wakanda.html>
- [4] 4D. (2013). *Wakanda*. Obtenido de Wakanda: <http://www.wakanda.org/faq>
- [5] Booch, G., Rumbaugh, J., & Jacobson, I. (s.f.). *UML - El Lenguaje Unificado de Modelado*.
- [6] Española, R. A. (s.f.). *Diccionario de la lengua española - Vigésima segunda edición*. Recuperado el 2013, de Diccionario de la lengua española - Vigésima segunda edición: <http://lema.rae.es/drae/>
- [7] Junstrand, S., Passaret, X., & Vázquez, D. (2005). Domótica y hogar digital. En S. Junstrand, X. Passaret, & D. Vázquez, *Domótica y hogar digital* (pág. 7). Madrid: Thomson Ediciones Spain Paraninfo S.A.
- [8] *Wikipedia*. (10 de Junio de 2013). Recuperado el Agosto de 2013, de Wikipedia: [http://es.wikipedia.org/wiki/Diagrama\\_de\\_clases](http://es.wikipedia.org/wiki/Diagrama_de_clases)
- [9] *Wikipedia*. (12 de Mayo de 2013). Obtenido de Wikipedia: [http://es.wikipedia.org/wiki/Diagrama\\_de\\_objetos](http://es.wikipedia.org/wiki/Diagrama_de_objetos)
- [10] *Wikipedia*. (23 de Marzo de 2013). Obtenido de Wikipedia: [http://es.wikipedia.org/wiki/Diagrama\\_de\\_componentes](http://es.wikipedia.org/wiki/Diagrama_de_componentes)
- [11] *Wikipedia*. (14 de Marzo de 2013). Obtenido de Wikipedia: [http://es.wikipedia.org/wiki/Diagrama\\_de\\_estructura\\_compuesta](http://es.wikipedia.org/wiki/Diagrama_de_estructura_compuesta)
- [12] *Wikipedia*. (13 de Marzo de 2013). Obtenido de Wikipedia: [http://es.wikipedia.org/wiki/Diagrama\\_de\\_paquetes](http://es.wikipedia.org/wiki/Diagrama_de_paquetes)
- [13] *Wikipedia*. (30 de Abril de 2013). Obtenido de Wikipedia: [http://es.wikipedia.org/wiki/Diagrama\\_de\\_despliegue](http://es.wikipedia.org/wiki/Diagrama_de_despliegue)
- [14] *Wikipedia*. (2013 de Agosto de 2013). Obtenido de Wikipedia: [http://es.wikipedia.org/wiki/Diagrama\\_de\\_secuencia](http://es.wikipedia.org/wiki/Diagrama_de_secuencia)

- [15] *Wikipedia*. (23 de Abril de 2013). Obtenido de Wikipedia: [http://es.wikipedia.org/wiki/Diagrama\\_de\\_colaboraci%C3%B3n](http://es.wikipedia.org/wiki/Diagrama_de_colaboraci%C3%B3n)
- [16] *Wikipedia*. (10 de Agosto de 2013). Obtenido de Wikipedia: [http://es.wikipedia.org/wiki/Diagrama\\_de\\_casos\\_de\\_uso](http://es.wikipedia.org/wiki/Diagrama_de_casos_de_uso)
- [17] *Wikipedia*. (12 de Agosto de 2013). Obtenido de Wikipedia: [http://es.wikipedia.org/wiki/Diagrama\\_de\\_actividades](http://es.wikipedia.org/wiki/Diagrama_de_actividades)
- [18] *Wikipedia*. (15 de Marzo de 2013). Obtenido de Wikipedia: [http://es.wikipedia.org/wiki/Diagrama\\_de\\_estado](http://es.wikipedia.org/wiki/Diagrama_de_estado)
- [19] *Wikipedia*. (14 de Agosto de 2013). Obtenido de Wikipedia: <http://es.wikipedia.org/wiki/HTML5>
- [20] *Wikipedia*. (10 de Agosto de 2013). Obtenido de Wikipedia: <http://es.wikipedia.org/wiki/JavaScript>
- [21] *Wikipedia*. (05 de Agosto de 2013). Obtenido de Wikipedia: [http://es.wikipedia.org/wiki/Hojas\\_de\\_estilo\\_en\\_cascada](http://es.wikipedia.org/wiki/Hojas_de_estilo_en_cascada)
- [22] *Wikipedia*. (21 de Julio de 2013). Obtenido de Wikipedia: <http://es.wikipedia.org/wiki/JQuery>
- [23] *Wikipedia*. (09 de Agosto de 2013). Obtenido de Wikipedia: <http://es.wikipedia.org/wiki/Processing>
- [24] *Wikipedia*. (16 de Agosto de 2013). Obtenido de Wikipedia: <http://es.wikipedia.org/wiki/Android>
- [25] *Android*. (s.f.). Obtenido de Andorid: <http://www.android.com/>
- [26] *Arduino*. (s.f.). Obtenido de Arduino: <http://arduino.cc/>
- [27] Control, C. (28 de Febrero de 2011). *Casa Control*. Recuperado el Agosto de 2013, de Casa Control: <http://casa-control.com/blog/?p=25>
- [28] *Desarrollo Web*. (s.f.). Obtenido de Desarrollo Web: <http://www.desarrolloweb.com/manuales/css3.html>
- [29] *Engibel*. (s.f.). Obtenido de [http://www.ebingel.com/index.php?option=com\\_content&view=article&id=68&Itemid=56](http://www.ebingel.com/index.php?option=com_content&view=article&id=68&Itemid=56)

- [30] *HomeTech el Hogar Digital SAS.* (s.f.). Obtenido de <http://www.hometechcolombia.com/empresa.html>
- [31] *HTML5 Rocks.* (s.f.). Obtenido de HTML5 Rocks: <http://www.html5rocks.com/es/>
- [32] *jQuery.* (s.f.). Obtenido de jQuery: <http://jquery.com/>
- [33] maji. (s.f.). *Domótica práctica paso a paso.* Recuperado el Agosto de 2013, de Domótica práctica paso a paso: <http://www.domoprac.com/protocolos-de-comunicacion-y-sistemas-domoticos/historia-de-la-domotica-pasado-presente-y-futuro/4.-la-llegada-de-los-protocolos-propietarios.html>
- [33] Pallarés, M. M. (27 de Junio de 2007). *Estudio de los sistemas domóticos y diseño de una aplicación.* Obtenido de <http://upcommons.upc.edu/pfc/bitstream/2099.1/11715/1/Memoria%20T%C3%A9cnica.pdf>
- [34] *Processing 2.* (s.f.). Obtenido de Processing 2: <http://processing.org/>
- [35] Wakanda. (2013). *Wakanda.* Obtenido de Wakanda: <http://www.wakanda.org>
- [36] Wikipedia. (01 de Agosto de 2013). *Wikipedia.* Recuperado el 2013 de Agosto, de <http://es.wikipedia.org/wiki/Controladores>
- [37] Wikipedia. (2 de Agosto de 2013). *Wikipedia.* Recuperado el Agosto de 2013, de Wikipedia: <http://es.wikipedia.org/wiki/Sensor>
- [38] Wikipedia. (22 de Julio de 2013). *Wikipedia.* Recuperado el Agosto de 2013, de Wikipedia: <http://es.wikipedia.org/wiki/Dom%C3%B3tica>
- [39] Wikipedia. (09 de Julio de 2013). *Wikipedia.* Recuperado el Agosto de 2013, de Wikipedia: <http://es.wikipedia.org/wiki/Actuador>