DISEÑO E IMPLEMENTACIÓN DE UNA PÁGINA WEB Y UN APLICATIVO MÓVIL PARA EL CONTROL DE VARIABLES EN EL INVERNADERO DE LA UNIVERSIDAD SURCOLOMBIANA

FABIAN STEVENS VARON VALENCIA

CÓDIGO: 20122112891

JOAN SEBASTIAN ROA SANCHEZ

CÓDIGO: 20111101762

UNIVERSIDAD SURCOLOMBIANA

FACULTAD DE INGENIERÍA

TECNOLOGÍA EN DESARROLLO DE SOFTWARE

NEIVA - HUILA

2014

DISEÑO E IMPLEMENTACIÓN DE UNA PÁGINA WEB Y UN APLICATIVO MÓVIL PARA EL CONTROL DE VARIABLES EN EL INVERNADERO DE LA UNIVERSIDAD SURCOLOMBIANA

FABIAN STEVENS VARON VALENCIA

CÓDIGO: 20122112891

JOAN SEBASTIAN ROA SANCHEZ

CÓDIGO: 20111101762

DIRECTOR SUGERIDO

ESAU SILVA LARA

INGENIERO DE SISTEMAS

UNIVERSIDAD SURCOLOMBIANA

FACULTAD DE INGENIERÍA

TECNOLOGÍA EN DESARROLLO DE SOFTWARE

NEIVA - HUILA

2014

TABLA DE CONTENIDO

[INDICE DE FIGURAS 6](#_Toc404733244)

[INDICE DE TABLAS 7](#_Toc404733245)

[INTRODUCCIÓN 9](#_Toc404733246)

[1. FORMULACIÓN DEL PROBLEMA 10](#_Toc404733247)

[2. JUSTIFICACIÓN 11](#_Toc404733248)

[3. OBJETIVOS 13](#_Toc404733249)

[3.1 GENERAL 13](#_Toc404733250)

[3.2 ESPECÍFICOS 13](#_Toc404733251)

[4. ANTECEDENTES 14](#_Toc404733252)

[4.1 INVERNADERO CONTROLADO POR ARDUINO 14](#_Toc404733253)

[4.2 MONITORIZACIÓN Y SEGUIMIENTO DE INVERNADERO CON ANDROID 15](#_Toc404733254)

[4.3 FARMPAD 16](#_Toc404733255)

[4.4 UNA PLANTA ONLINE 16](#_Toc404733256)

[5. MARCO DE REFERENCIA 18](#_Toc404733257)

[5.1 MARCO TEÓRICO 18](#_Toc404733258)

[5.1.1 Smartphone o Teléfono Inteligente 18](#_Toc404733259)

[5.1.2 Internet de la Cosas (IOT) 19](#_Toc404733260)

[5.1.3 Lenguajes De Programación 21](#_Toc404733261)

[5.1.4 Entorno de Desarrollo Integrado IDE 23](#_Toc404733262)

[5.1.5 Servidor de Aplicaciones 25](#_Toc404733263)

[5.1.6 Lenguajes De Marcado 26](#_Toc404733264)

[5.1.7 Sistemas operativos 28](#_Toc404733265)

[5.1.8 Bases De Datos 29](#_Toc404733266)

[5.1.9 Lenguaje de Modelado Unificado (UML) 32](#_Toc404733267)

[5.1.10 Software Libre y Código Abierto 34](#_Toc404733268)

[5.1.11 Hardware Libre 35](#_Toc404733269)

[5.1.12 Control de Versiones 35](#_Toc404733270)

[5.1.13 Computación en la nube (Cloud Computing) 36](#_Toc404733271)

[5.1.14 Responsive Design 36](#_Toc404733272)

[5.1.15 Mockup 37](#_Toc404733273)

[5.2 MARCO LEGAL 37](#_Toc404733274)

[6. ALCANCES Y LIMITACIONES 40](#_Toc404733275)

[7. METODOLOGÍA 41](#_Toc404733276)

[7.1 TIPO DE INVESTIGACIÓN 42](#_Toc404733277)

[7.2 MÉTODO 42](#_Toc404733278)

[8. DISEÑO METODOLOGICO 43](#_Toc404733279)

[8.1 METODOLOGÍA DE DESARROLLO 43](#_Toc404733280)

[8.2 ANALISIS 45](#_Toc404733281)

[8.2.1 Documentación y Diagrama de Casos de Uso 45](#_Toc404733282)

[8.2.2 Diagramas de Clase 48](#_Toc404733283)

[8.2.3 Modelado de la Base de Datos 49](#_Toc404733284)

[8.3 DISEÑO 53](#_Toc404733285)

[8.3.1 Página web 59](#_Toc404733286)

[8.3.2 Aplicación para dispositivo móvil Android 63](#_Toc404733287)

[8.3.3 Librerías adicionales 64](#_Toc404733288)

[8.4 DESARROLLO 67](#_Toc404733289)

[8.4.1 Java 68](#_Toc404733290)

[8.4.2 Seguridad y Autenticación 82](#_Toc404733291)

[8.5 PRUEBAS 85](#_Toc404733292)

[8.5.1 Responsive Design 85](#_Toc404733293)

[8.6 INTEGRACIÓN 88](#_Toc404733294)

[9. CRONOGRAMA DE ACTIVIDADES 92](#_Toc404733295)

[10. RECURSOS DISPONIBLES 93](#_Toc404733296)

[10.1 HUMANOS 93](#_Toc404733297)

[10.2 MATERIALES 93](#_Toc404733298)

[10.3 INSTITUCIONALES 94](#_Toc404733299)

[10.4 FINANCIEROS 94](#_Toc404733300)

[11. CONCLUSIONES 94](#_Toc404733301)

[12. RECOMENDACIONES 94](#_Toc404733302)

[BIBLIOGRAFÍA 95](#_Toc404733303)

[ANEXOS 99](#_Toc404733304)

# INDICE DE FIGURAS

[Ilustración 1 Smartphones 2014 19](#_Toc404733305)

[Ilustración 2 Internet De las Cosas IOT 20](#_Toc404733306)

[Ilustración 3 Eclipse Luna es la versión actual de este IDE 24](#_Toc404733307)

[Ilustración 4 Logo Oficial de HTML5 27](#_Toc404733308)

[Ilustración 5 Logo de UML 32](#_Toc404733309)

[Ilustración 6 Diagrama de la clase Avión con sus Atributos y Funciones. 33](#_Toc404733310)

[Ilustración 7 Sistema en UML 33](#_Toc404733311)

[Ilustración 8 Casos de Uso en UML 34](#_Toc404733312)

[Ilustración 9 Actor en UML 34](#_Toc404733313)

[Ilustración 10 Normatividad Colombiana en Software 40](#_Toc404733314)

[Ilustración 11 Desarrollo en Cascada 44](#_Toc404733315)

[Ilustración 12 Diagrama de Casos de Uso 46](#_Toc404733316)

[Ilustración 13 Diagrama de Clases 48](file:///D:\Stevens\Github-Proyects\Document-Inverna-SURCO\Anteproyecto_Grado.docx#_Toc404733317)

[Ilustración 14 Modelo original de la Base de Datos 49](#_Toc404733318)

[Ilustración 15 Modelo Actual de la Base de Datos 50](#_Toc404733319)

[Ilustración 16 Modelo propuesto de la Base de Datos 50](#_Toc404733320)

[Ilustración 17 Diseño final Activity\_main.xml 54](#_Toc404733321)

[Ilustración 18 Diseño final Consulta offline y online.xml 58](#_Toc404733322)

[Ilustración 19 Sketch Login– vista administrador e Invitado 59](#_Toc404733323)

[Ilustración 20 Sketch Pagina web – vista administrador. Variables 60](#_Toc404733324)

[Ilustración 21 sketch Pagina web – vista administrador. Graficas 60](#_Toc404733325)

[Ilustración 22 Sketch Pagina web – vista Invitado. Variables 61](#_Toc404733326)

[Ilustración 23 Diseño Final Login – vista Administrador e Invitado 61](#_Toc404733327)

[Ilustración 24 Diseño Final Pagina web – vista administrador. Variables 62](#_Toc404733328)

[Ilustración 25 Diseño Final Pagina web – vista administrador. Graficas 62](#_Toc404733329)

[Ilustración 26 Diseño Final Pagina web – vista Invitado. Variables 63](#_Toc404733330)

[Ilustración 27 Sketch Main Activity y Activity Variables 63](#_Toc404733331)

[Ilustración 28 Diseño Final Main Activity y Activity Variables 64](file:///D:\Stevens\Github-Proyects\Document-Inverna-SURCO\Anteproyecto_Grado.docx#_Toc404733332)

[Ilustración 29 Alert tradicional 65](#_Toc404733333)

[Ilustración 30 Alert con SweetAlert 65](#_Toc404733334)

[Ilustración 31 Ejemplo de Gráficas conseguidas con Highcharts 66](#_Toc404733335)

[Ilustración 33 Vista Responsive Desgin en Iphone 6 86](#_Toc404733336)

[Ilustración 32 Vista Responsive Design en Samsung Galaxy S4 86](file:///D:\Stevens\Github-Proyects\Document-Inverna-SURCO\Anteproyecto_Grado.docx#_Toc404733337)

[Ilustración 34 Vista Responsive Design en Htc 87](#_Toc404733338)

[Ilustración 35 Vista Responsive Design en Tablet 87](#_Toc404733339)

[Ilustración 36 Servidor en Funcionamiento. 88](#_Toc404733340)

[Ilustración 37 Login en funcionamiento 89](#_Toc404733341)

[Ilustración 38Implementación de SweetAlert al Digitar Usuario o Contraseña Errónea 89](#_Toc404733342)

[Ilustración 39 Página de Inicio para el Administrador con Información traída desde la Base de Datos 90](#_Toc404733343)

[Ilustración 40 Graficas para Administrador con Información traída desde la Base de Datos 90](#_Toc404733344)

[Ilustración 41Página de Inicio para el Invitado con Información traída desde la Base de Datos 90](#_Toc404733345)

[Ilustración 42 Aplicación Móvil en Funcionamiento 91](#_Toc404733346)

[Ilustración 43 SweetAlert Implementado en la Aplicación Android 91](#_Toc404733347)

# INDICE DE TABLAS

[Tabla 1 Actor 1 Administrador 45](#_Toc404733348)

[Tabla 2 Actor 2 Invitado 45](#_Toc404733349)

[Tabla 3 Caso de Uso Login y Consulta Web - Administrador 46](#_Toc404733350)

[Tabla 4 Caso de Uso 2 Descarga de Estadísticas - Administrador 47](#_Toc404733351)

[Tabla 5 Caso de Uso 3 Consulta web - Invitado 47](#_Toc404733352)

[Tabla 6 Caso de Uso 4 Consulta Móvil - Administrador e Invitado 47](#_Toc404733353)

[Tabla 7 Diccionario de Datos - Tabla Variables 51](#_Toc404733354)

[Tabla 8 Diccionario de Datos - Tabla Perfil 52](#_Toc404733355)

[Tabla 9 Diccionario de Datos - Tabla Usuario 52](#_Toc404733356)

[Tabla 10 Diccionario de Datos - Tabla Usuario-Perfil 52](#_Toc404733357)

[Tabla 11 Cronograma de Actividades 92](#_Toc404733358)

# INTRODUCCIÓN

El uso de las tecnologías en la realización de diferentes tareas de nuestro día a día, la optimización o mejoramiento a nuevos problemas que se presentan y la oportunidad de estar innovando con soluciones creativas, es lo que hace de la ingeniería el medio ideal para generar conocimiento, el cual en este proyecto se dará a conocer con un tema en específico que aqueja a nuestra región; la garantía de tener un cultivo de invernadero que sea lo más eficiente posible, que aplique tecnologías de uso cotidiano, para hacer de tareas que hace un tiempo requerían de que personas pasaran horas y días al pendiente de estas, sean realizadas a distancia por medio de dispositivos móviles y monitoreadas a través de estos y de una página web, revolucionando así, el proceso de cultivo y obtención de diferentes plantas en un invernadero.

El cultivo bajo invernadero ha permitido obtener producciones de mejor calidad y mayores rendimientos, en cualquier estación del año, y a su vez permiten alargar el ciclo de cultivo, garantizando la producción de alimento aun en las épocas más difíciles y a precios asequibles. Este incremento del valor de los productos permite que el agricultor pueda invertir tecnológicamente en su explotación mejorando la forma en la que actualmente realiza sus cultivos, los sistemas de riego localizado por goteo, el uso de sensores para monitorear el clima, la humedad, temperatura y tomar decisiones adecuadas frente a estos factores consiguiendo posteriormente una mejora en el rendimiento y la calidad del producto final.

Con este proyecto se pretende implementar una manera eficiente para monitorear un sistema de invernadero con la ayuda de tecnologías de software libre, las cuales permitan al agricultor obtener el máximo rendimiento de su cultivo sin tener preocuparse por factores externos que puedan afectar su producción.

# FORMULACIÓN DEL PROBLEMA

Se define un invernadero como un ambiente diseñado para mantener en un espacio específico y con condiciones adecuadas de temperatura, riego, humedad, Luminosidad entre otras, el cultivo de plantas destinadas a diversos propósitos. Sin embargo la fumigación con pesticidas y los cambios abruptos de clima son potencialmente dañinos y pueden afectar seriamente el producto final y esto generar pérdidas para los agricultores.

Una alternativa para solucionar estos problemas ha sido la automatización de los invernaderos con el fin de mejorar cada proceso, aplicando de manera adecuada condiciones fisicoquímicas óptimas para el adecuado cuidado y desarrollo que requiere cada producción. Este proyecto estará en la capacidad de mostrar en la pantalla de un dispositivo móvil en tiempo real y en una página web, diferentes variables de un invernadero como lo son, temperatura, humedad, iluminación entre otras. Con este aplicativo móvil y página web el agricultor podrá reducir el tiempo que debe permanecer en el invernadero ya que el software que se implementará realizará este trabajo obteniendo directamente la información la cual es subida a un servidor en internet gracias a Arduino (Arduino, 2014), Hardware que realizará esta dispendiosa labor.

Las condiciones de cultivo y clima varían de acuerdo a la región donde se ubique el invernadero y es por esta sencilla razón que surge la problemática de encontrar un aplicativo móvil o una página web personalizada, que permita tener un control de los diferentes aspectos que un cultivo requiere para su optima producción.

# JUSTIFICACIÓN

La agricultura es una actividad de gran importancia estratégica como pilar fundamental para el desarrollo autosuficiente y riqueza de los países en los cuales esta labor se lleva acabo. Todas las actividades económicas que abarca la agricultura, tienen su fundamento en la explotación del suelo o de los recursos que este origina tanto en su forma natural como por la acción del hombre.

El clima terrestre es caótico y complejo, esto se debe a múltiples factores en los que el hombre; a pesar de que ha tratado de controlar el clima, no tiene influencia alguna, esto afecta de manera directa a los diferentes tipos de cultivos los cuales varían de acuerdo a factores climáticos de la zona en la que se encuentran.

El uso de la tecnología cada vez es más común verlo en las actividades diarias del ser humano, como apoyo a las tareas que normalmente realiza, esto hace cada vez más atractivo desarrollar proyectos en los cuales estén involucrados estos tres factores (Tecnología, medioambiente y los seres humanos).

A raíz de esto surge la inquietud de los integrantes del grupo de buscar proyectos donde puedan aplicar sus conocimientos en el desarrollo de aplicativos móviles y diseño de páginas web para mejorar u optimizar labores agrícolas, y es así como resultado de esa búsqueda se encuentran con dos estudiantes de Ingeniería electrónica, los cuales están desarrollando la automatización y control de un invernadero a escala real en la granja de la Universidad Surcolombiana y que en búsqueda de una mejor calidad en su trabajo final de grado, como también en ampliar y crear lazos de amistad con otros programas de la misma facultad, se llega a una idea para realizar en conjunto un aplicativo móvil desarrollado bajo el sistema operativo Android (Android, 2014), el cual estará encargado de visualizar todas las diferentes variables que sean objeto de cuidado en el invernadero y de una página web desarrollada con todos los estándares y normas internacionales como lo son HTML5, CSS3 (Css3, 2014) y JavaScript que estará en la capacidad además de visualizar las variables al igual que el aplicativo en Android, permitir el manejo de diferentes perfiles de usuario los cuales tendrán acceso a diferentes privilegios según el rol que desempeñen. Usuario invitado, podrá solamente consultar las variables del invernadero, mientras que el usuario Administrador además de tener las funciones del invitado, podrá tener acceso a reportes de actividades a lo largo de los días.

La recolección y procesamiento de los datos estará a cargo de una plataforma de hardware libre como lo es Arduino UNO (UNO, 2014) , y Android Como sistema operativo nativo en el cual se desarrollará un aplicativo para dispositivos móviles que recopilará la información y que luego los usuarios de la misma, podrán mantenerse informados a través de la misma.

Este proyecto tendrá un gran impacto en la forma como se controlan y administran los invernaderos, ya que las personas no tienen que estar pendientes de cuando tienen que regar sus plantas, o cuánta agua deben suministrarles, simplemente revisará con su aplicación el estado de las variables y dado el caso en donde no pueda contar con un dispositivo móvil Android, tanto el administrador como el usuario invitado, podrán realizar la consulta mediante una página web destinada para tal fin.

# OBJETIVOS

## 3.1 GENERAL

DISEÑAR E IMPLEMENTAR UN APLICATIVO MOVIL Y UNA PÁGINA WEB PARA CONTROLAR VARIABLES EN EL INVERNADERO DE LA UNIVERSIDAD SURCOLOMBIANA

## 3.2 ESPECÍFICOS

MONITOREAR VARIABLES DEL AMBIENTE QUE SE MANEJAN DENTRO DEL INVERNADERO COMO LO SON LA TEMPERATURA, LA HUMEDAD, RADIACIÓN SOLAR, ENTRE OTRAS.

TRABAJAR CON DATOS EN TIEMPO REAL MEDIANTE LA SINCRONIZACIÓN DE LAS VARIALES ALMACENADAS EN UN SERVIDOR CON EL APLICATIVO MOVIL Y LA PÁGINA WEB.

PROTEGER EL ACCESO A INFORMACIÓN PRIVADA A LOS USUARIOS QUE NO TENGAN LOS PRIVILEGIOS PARA TAL FIN MEDIANTE EL USO DE MÉTODOS DE AUTENTICACIÓN Y VALIDACIÓN JAAS.

# ANTECEDENTES

Después de realizar un rastreo bibliográfico sobre el proyecto de investigación que se adelanta, se puede decir que se encontraron algunos trabajos de grado, artículos de revista, páginas de innovación tecnológica, aplicaciones desarrolladas en otros países e informes de investigación, que hacen referencia al desarrollo tecnológico para el control y mantenimiento de invernaderos, a continuación se relacionan algunos de ellos:

## INVERNADERO CONTROLADO POR ARDUINO

(FUNDACIÓN SANVALERO)

Enlace al proyecto y su documentación

<https://docs.google.com/folderview?id=0B-PoupdpoRSPdHJxbWgxX3FFOU0>

En este proyecto se trabajó con lenguaje de programación C, módulos bluetooth y dispositivo Android como método para el control y administración de las variables del entorno.

El trabajo a realizar consistirá en la domotización de un invernadero convencional, para que a través de un teléfono móvil Android conectado por Bluetooth al invernadero, podamos saber distintos parámetros como:

Humedad, temperatura y velocidad a la que sopla el viento. Y además, podamos encender y apagar luces, conectar y desconectar el riego y activar o desactivar los ventiladores. (Marquez, 2012)

## MONITORIZACIÓN Y SEGUIMIENTO DE INVERNADERO CON ANDROID

(ESCUELA TEC. SUPERIOR DE INGENIERÍA Y DISEÑO INDUSTRIAL – UPM)

Enlace al proyecto y su documentación

<http://icra.blogspot.es/categoria/proyecto-icra/>

Este proyecto de tesis en maestría tiene también muy buenos aportes para tomar como referencia para futuras mejoras que pueden llegar a tener el invernadero de la Universidad Surcolombiana.

El siguiente proyecto tiene como objetivo la **monitorización** y **supervisión** de un invernadero mediante el uso de la tecnología de teléfonos inteligentes Android.  
  
Los invernaderos permiten tener un mayor control del entorno donde crecen plantas. Son utilizados para aumentar la calidad y rendimiento de las plantas en ciertas ubicaciones, debido a estaciones cortas de crecimiento o bien malas condiciones de iluminación debido a las localizaciones geográficas, por lo que permite mejorar la producción de alimentos vegetales en entornos extremos.  
  
Para una correcta supervisión del sistema es fundamental conocer el estado de sus variables en tiempo real, para incluso poder tomar decisiones sobre el invernadero con base en los resultados que nos arroje el invernadero.  
  
Las nuevas tecnologías móviles nos brindan la oportunidad de monitorizar y supervisar de manera práctica, sencilla y eficaz la evolución de la plantación. Permitiendo al usuario poder conocer e intervenir en el estado del invernadero, independientemente de la localización geográfica. (Industrial., 2014)

## FARMPAD

(FARMLOGIC COMAPNY)

Enlace a la página web de la Empresa (Ingles)

<http://www.farmlogic.com/>

Debido a que la posibilidad de internacionalizar nuestros productos y que todo ahora debe tener soporte en idioma inglés, se decide también indagar por proyectos o empresas que realicen trabajos relacionados con este proyecto de grado.

Por tal razón presentamos a FARMPAD, una empresa que reside en Kentucky, Estados unidos; la cual se dedica al desarrollo de varias aplicaciones para el control de cultivos y herramientas tecnológicas para facilitar el trabajo de los granjeros.

FarmPAD provides mobility to your FarmLogic Web Headquarters when installed on your tablet or smartphone. The sync process along with a robust permissions package puts the right functions in the right hands. (FarmLogic, 2014)

## UNA PLANTA ONLINE

(PROYECTO GANADOR RETO CAMPUS PARTY)

Enlace a la página web del proyecto y cuenta en Twitter:

[@unaplantaonline](http://unaplantaonline.iqbusiness.mx/)

[Una planta Online (unaplantaonline) en Twitter](https://twitter.com/unaplantaonline)

Esta es una propuesta ganador muy interesante la cual surgió a partir de un concurso realizado por Campus Party (CampusParty, 2014) y su plataforma de retos tecnológicos para la comunidad denominado Campus Labs (lab, 2014)

¿Qué hace este dispositivo?

Permite el monitoreo y control del ambiente de un invernadero a través de la información recibida de sensores instalados en el lugar, puede funcionar de 2 maneras:

Modo Automático

El dispositivo toma muestras de los sensores cada 30 min (el periodo de las muestras puede variar dependiendo de la hora del día) y dependiendo de los valores obtenidos toma decisiones automáticamente para asegurar su óptimo cuidado.

Modo Manual

El usuario puede pedir información acerca del estado del dispositivo, por medio de un tweet, la información a requerir pueden ser los niveles de humedad, luz o temperatura. De esta manera y con los valores obtenidos, el usuario puede tomar una decisión e indicarle al dispositivo por medio de un tweet que acciones debe de tomar, estas acciones pueden ser dispensar agua, ventilar o iluminar el entorno para elevar la temperatura. (Tamayo, 2014)

# MARCO DE REFERENCIA

## MARCO TEÓRICO

A la hora de leer o revisar un proyecto de grado es muy importante conocer todas las terminologías que bien para los expositores y afines son de uso cotidiano, para jurados o personas ajenas a la carrera universitaria que cursan estos; no lo son y por este pequeño detalle en el uso de palabras específicas o de términos que puedan tener varios significados según el ámbito donde se mencionen, se pueden generar problemas a la hora tratar de transmitir las ideas por parte de los estudiantes.

Por esta razón es que se crea este marco teórico, aquí se introducirán los conceptos más relevantes sobre las metodologías, métodos y herramientas utilizadas para el desarrollo del presente proyecto de grado, pero no se puede dar una teoría completa es así que se tratará de presentar una base para su fácil comprensión.

### Smartphone o Teléfono Inteligente

Los Smartphones o teléfonos inteligentes surgieron a finales de los años noventa, combinan las funciones de un teléfono celular tradicional con otras características, como la posibilidad de instalar en el dispositivo un sistema operativo completo, con aplicaciones para realizar diversas tareas y trabajar con grandes cantidades de datos, enviar correos electrónicos, conectarse a Internet, tomar fotos, comunicarse a través de wifi y Bluetooth, etc. Por tanto tienen aplicaciones similares a las de un computador portátil o una agenda electrónica, a las que se suman las características propias de los teléfonos celulares. (Universidad Del Rosario Smartphone, 2012)

Se puede decir que con el término inteligente se hace referencia a un terminal con teclado QWERTY en miniatura, una pantalla táctil (lo más habitual, denominándose en este caso “teléfono móvil táctil”), un administrador de aplicaciones PIM (personal information manager), hardware para servicios multimedia y un sistema operativo que soporte el desarrollo de tareas de mediana complejidad similares a las que se realizan en un equipo de informático de escritorio. En síntesis, para algunos se trata de un teléfono que corre un sistema operativo completo e identificable y provee una interfaz estándar y una plataforma que incluye aplicaciones de elevada complejidad. Para otros, es simplemente un teléfono móvil con funcionalidades avanzadas como correo electrónico, Internet o un teclado integrado. (Belatrixsf, 2008)



Ilustración 1 Smartphones 2014

### Internet de la Cosas (IOT)

Internet de las cosas (IOT), algunas veces denominado "Internet de los objetos", lo cambiará todo, incluso a nosotros mismos. Si bien puede parecer una declaración arriesgada, hay que tener en cuenta el impacto que Internet ha tenido sobre la educación, la comunicación, las empresas, la ciencia, el gobierno y la humanidad. Claramente Internet es una de las creaciones más importantes y poderosas de toda la historia de la humanidad. (Cisco, 2011.)

El IOT es el mundo en el que cada objeto tiene una identidad virtual propia y capacidad potencial para integrarse e interactuar de manera independiente en la Red con cualquier otro individuo, ya sea una máquina (M2M) o un humano. Estamos sufriendo una transformación donde “things” hablan con otros “things” y estas conversaciones están creando nuevos modelos de negocio, productos y compañías. Hace 20 años, Internet se usaba principalmente como herramienta para buscar información. En los últimos 10 años hemos vivido una nueva forma de uso de Internet, donde todo se ha convertido en social, transaccional y móvil. (Pastor, 2013)

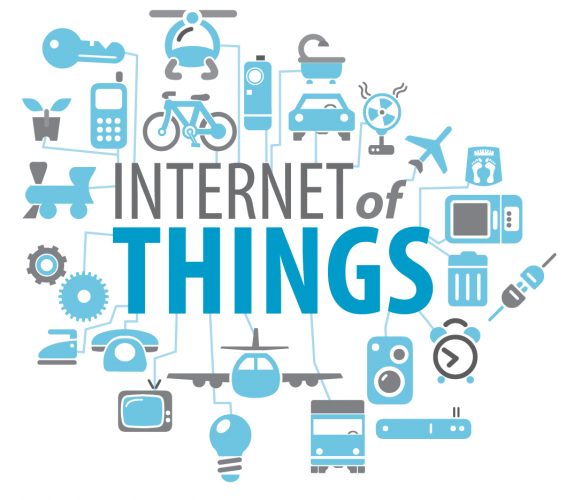


Ilustración 2 Internet De las Cosas IOT

### Lenguajes De Programación

Un Lenguaje de Programación es un conjunto de reglas, notaciones, símbolos y/o caracteres que permiten a un programador poder expresar el procesamiento de datos y sus estructuras en la computadora. Cada lenguaje posee sus propias sintaxis. También se puede decir que un programa es un conjunto de órdenes o instrucciones que resuelven un problema específico basado en un Lenguaje de Programación. (CELAYA, 2014)

Existen varios lenguajes de programación, los que este proyecto usará son los que a continuación se citan junto a una breve descripción e historia.

#### JAVA

El lenguaje Java™ fue creado por Sun Microsystems Inc. en un proceso por etapas que arranca en 1990, año en el que Sun creó un grupo de trabajo, liderado por James Gosling, para desarrollar un sistema para controlar electrodomésticos e incluso PDAs o Asistentes Personales (pequeños ordenadores) que, además, permitiera la conexión a redes de ordenadores. Se pretendía crear un hardware polivalente, con un Sistema Operativo eficiente (SunOS) y un lenguaje de desarrollo denominado Oak (roble), el precursor de Java. El proyecto finalizó en 1992 y resultó un completo fracaso debido al excesivo coste del producto, con relación a alternativas similares, tras lo cual el grupo se disolvió.

Por entonces aparece Mosaic y la World Wide Web. Después de la disolución del grupo de trabajo, únicamente quedaba del proyecto el lenguaje Oak. Gracias a una acertada decisión de distribuir libremente el lenguaje por la Red de Redes y al auge y la facilidad de acceso a Internet, propiciado por la WWW, el lenguaje se popularizó y se consiguió que una gran cantidad de programadores lo depurasen y terminasen de perfilar la forma y usos del mismo. A partir de este momento, el lenguaje se difunde a una velocidad vertiginosa, añadiéndosele numerosas clases y funcionalidad para TCP/IP. El nombre del lenguaje tuvo que ser cambiado ya que existía otro llamado Oak. El nombre “Java” surgió en una de las sesiones de “brainstorming” celebradas por el equipo de desarrollo del lenguaje. Buscaban un nombre que evocara la esencia de la tecnología (viveza, animación, rapidez, interactividad) Java fue elegido de entre muchísimas propuestas. No es un acrónimo, sino únicamente algo humeante, caliente y que a muchos programadores les gusta beber en grandes cantidades: una taza de café (Java en argot Inglés americano[[1]](#footnote-1)). De esta forma, Sun lanzó las primeras versiones de Java a principios de 1995.

Desde entonces, Sun ha sabido manejar inteligentemente el éxito obtenido por su lenguaje, concediéndose licencias a cualquiera sin ningún problema, fomentando su uso entre la comunidad informática y extendiendo las especificaciones y funcionalidad del lenguaje. (JAVA, 2014)

#### JAVASCRIPT

JavaScript es un lenguaje de script multiplataforma [cross-platform] orientado a objetos. Es un lenguaje pequeño y ligero; no es útil como un lenguaje independiente, más bien está diseñado para una fácil incrustación en otros productos y aplicaciones, tales como los navegadores Web. Dentro de un entorno anfitrión, JavaScript puede ser conectado a los objetos de su entorno para proveer un control programable sobre éstos.

El núcleo de JavaScript contiene un conjunto central de objetos, tales como Array (arreglos), Date (fechas) y Math (objetos matemáticos), además de un conjunto central de elementos del lenguaje tales como los operadores, estructuras de control y sentencias. El núcleo de JavaScript puede ser extendido para una variedad de propósitos complementándolo con objetos adicionales; por ejemplo:

* *JavaScript del lado Cliente* extiende el núcleo del lenguaje proporcionando objetos para el control del navegador (Navigator o cualquier Web browser) y su Modelo Objeto Documento [Document Object Model] (DOM). Por ejemplo, las extensiones del lado del cliente permiten a una aplicación ubicar elementos en un formulario HTML y responder a los eventos de usuario tales como los clics del mouse, entradas del formulario y navegación de páginas.
* *JavaScript del lado Servidor* extiende el núcleo del lenguaje proporcionando objetos relevantes para la ejecución de JavaScript en un servidor. Por ejemplo, las extensiones del lado del servidor permiten que una aplicación se comunique con una base de datos relacional, proporcionar continuidad de la información desde una invocación de la aplicación a otra o efectuar la manipulación de archivos en un servidor.

Netscape inventó JavaScript y JavaScript fue usado por primera vez en los navegadores Netscape. (Mozilla, 2014)

### Entorno de Desarrollo Integrado IDE

Un entorno integrado de desarrollo (IDE), es un tipo de software compuesto por un conjunto de herramientas de programación. En concreto, el IDE se compone de:

* Editor de código de programación.
* Compilador.
* Intérprete.
* Depurador.
* Constructor de interfaz gráfico.

Los primeros entornos de desarrollo integrados nacen a principios de los años 70, y se popularizan en la década de los 90. Tienen el objetivo de ganar fiabilidad y tiempo en los proyectos de software. Proporcionan al programador una serie de componentes con la misma interfaz gráfica, con la consiguiente comodidad, aumento de eficiencia y reducción de tiempo de codificación.

Normalmente, un IDE está dedicado a un determinado lenguaje de programación. No obstante, las últimas versiones de los IDEs tienden a ser compatibles con varios lenguajes (por ejemplo, Eclipse, NetBeans, Microsoft Visual Studio...) mediante la instalación de plugins adicionales.

Eclipse será nuestro entorno de desarrollo predeterminado y con el que desarrollaremos las diferentes etapas tanto del aplicativo móvil, como de la página web.

#### ECLIPSE (IDE)

Eclipse es una plataforma de desarrollo de código abierto basada en Java. Por si misma, es simplemente un marco de trabajo y un conjunto de servicios para la construcción del entorno de desarrollo de los componentes de entrada. Afortunadamente, Eclipse tiene un conjunto de complementos, incluidas las Herramientas de Desarrollo de Java (JDT). (ibm, 2012)



Ilustración 3 Eclipse Luna es la versión actual de este IDE

### Servidor de Aplicaciones

Un *servidor de aplicaciones* no es más que un cambio de nombre, para algunos servidores Web de nueva generación que proporcionan la lógica de negocio sobre la que construir aplicaciones. Suelen asociarse con servidores de alto rendimiento pensados para dar servicio a sitios Web (*Web sites*) con grandes necesidades: afluencia de visitas, movimiento de datos, atención de transacciones hacia bases de datos, etc. Generalmente los fabricantes del sector tienen a disposición del público un servidor Web básico y otro con multitud de extensiones fuertemente integradas al que llaman servidor de aplicaciones.

Como servidor de aplicaciones, se trabajará con JBoss, una tecnología desarrollada por RedHat; a continuación se explicará más detallado que es y en que consiste JBoss.

#### JBOSS

¿QUÉ ES?

JBoss Enterprise Application Platform es la plataforma para aplicaciones Java líder en la industria, integrada, simplificada y proporcionada por el líder en software de código abierto para empresas.

¿PARA QUÉ SIRVE?

Integrando las tecnologías punteras del mercado en una única y sencilla solución, la Plataforma de Aplicaciones JBoss Enterprise facilita el desarrollo, la implantación y la gestión de las aplicaciones Java. Incluye las tecnologías de código abierto más avanzadas para la creación, despliegue y alojamiento de aplicaciones Java de empresas y de servicios en una arquitectura orientada a servicios. Con la Plataforma de Aplicaciones JBoss Enterprise, las aplicaciones Java para empresa son sencillas, abiertas y asequibles.

### Lenguajes De Marcado

En el mundo de los documentos en papel, el término marcado hace referencia a la manera en la que el editor anota los manuscritos con especificaciones tipográficas y otros datos sobre su presentación. En los documentos electrónicos, el marcado es el término empleado para describir los códigos, denominados también etiquetas, añadidos al texto electrónico que definen la estructura y el formato en el que tiene que aparecer. Puede ser utilizado, además, para propósitos muy diferentes como son la escritura, la impresión, el intercambio, la presentación de pantallas, etc.

La gran variedad de lenguajes de marcado y su patente incompatibilidad constituyen la causa de los problemas que se plantean al intercambiar un documento entre plataformas heterogéneas. Los lenguajes estándar proporcionan una manera de solventar este hecho, ya que son independientes de la aplicación y de la plataforma hardware, empleando para marcar el documento, en la mayoría de los casos, código ASCII.

Cuando se habla de lenguajes de marcado, es importante distinguir entre la estructura lógica y física del documento. (Castro, 2014)

A lo largo de este proyecto se trabajará con 2 lenguajes de marcado los cuales son los más utilizados para este tipo de proyectos tecnológicos, HTML y XML.

#### HTML5

El lenguaje en el cual se realiza la Navegación Web lleva las siglas en el idioma inglés de HTML (es decir, Hyper Text Markup Language) que en nuestro idioma significa Lenguaje de Marcado de Hipertexto, siendo empleado para la creación, el diseño y el desarrollo de las páginas web que visitamos cotidianamente. (MasterMagazine, 2014)

HTML5 es la quinta revisión relevante que sufre el lenguaje HTML, y es más que importante resaltar que esta versión ya es un estándar.

Después de varios años de cambios y desarrollo, W3C acaba de anunciar la recomendación para utilizar HTML5 como estándar, esto gracias a la colaboración de desarrolladores y empresas, que juntos han resuelto problemas, bugs y hecho las pruebas necesarias. (Laister, 2014)

La W3C nos invita a defender la web abierta programando, compartiendo o diseñando junto a HTML5, un lenguaje que lleva años permitiendo acceder a cualquier sitio web y ahora nos permite tener mejores tecnologías en cualquier navegador web, sin la necesidad de instalar complementos o parecidos. (W3C, 2014)



Ilustración 4 Logo Oficial de HTML5

#### XML

XML es el acrónimo de eXtensible Markup Language, o lo qué es lo mismo: Lenguaje Extensible de Marcado.

Estas son un conjunto de reglas que describen cómo podemos escribir metainformación en un texto, es un lenguaje relativamente moderno (Febrero de 1998), y al igual que su homólogo HTML, está basado un lenguaje maduro de marcado llamado SGML: Standard Generalized Markup Languaje. (Martínez, 2014)

### Sistemas operativos

El sistema operativo de un ordenador se puede definir, de una manera muy burda, como el software (programa) encargado de gestionar y manejar el hardware (piezas) del equipo. Crea una capa de abstracción sobre la complejidad de los circuitos y conexiones eléctricas que componen un ordenador proporcionando una interfaz amigable. Por ejemplo, cuando copiamos archivos sólo le tenemos que indicar al sistema operativo los ficheros de origen y dónde los queremos pegar; no tenemos que saber que por debajo de eso hay un disco duro compuesto de platos, éstos a su vez de sectores, y éstos de bloques y de un cabezal con el que ir recorriéndolos y realizando la escritura.

Como ejemplos de sistemas operativos tenemos las distintas versiones de Windows de la compañía Microsoft, el Mac OS de Apple y GNU/Linux.

GNU/Linux es un sistema operativo libre desarrollado por voluntarios de todo el mundo. (Group, 2014)

Android está basado en Linux y se dará a continuación una breve reseña de que es este sistema operativo y cuáles son sus ventajas frente a los utilizados para ordenadores tanto de escritorio como de mesa.

#### Android

En los últimos años los teléfonos móviles han experimentado una gran evolución, desde los primeros terminales, grandes y pesados, pensados sólo para hablar por teléfono en cualquier parte, a los últimos modelos, con los que el término “medio de comunicación” se queda bastante pequeño.

Es así como nace Android. Android es un sistema operativo y una plataforma software, basado en Linux para teléfonos móviles. Además, también usan este sistema operativo (aunque no es muy habitual), tablets, netbooks, reproductores de música e incluso PC’s. Android permite programar en un entorno de trabajo (framework) de Java, aplicaciones sobre una máquina virtual Dalvik (una variación de la máquina de Java con compilación en tiempo de ejecución). Además, lo que le diferencia de otros sistemas operativos, es que cualquier persona que sepa programar puede crear nuevas aplicaciones, widget[[2]](#footnote-2), o incluso, modificar el propio sistema operativo, dado que Android es de código libre, por lo que sabiendo programar en lenguaje Java, va a ser muy fácil comenzar a programar en esta plataforma. (Sanz)

### Bases De Datos

Una base de datos es una colección de archivos relacionados que permite el manejo de la información de alguna compañía. Cada uno de dichos archivos puede ser visto como una colección de registros y cada registro está compuesto de una colección de campos. Cada uno de los campos de cada registro permite llevar información de algún atributo de una entidad del mundo real. (Chávez, 2014)

#### Modelo de datos

Bajo la estructura de la base de datos se encuentra el modelo de datos: una colección de herramientas conceptuales para describir los datos, las relaciones, la semántica y las restricciones de consistencia. Para ilustrar el concepto de un modelo de datos, describimos el modelo que usará en este proyecto de grado, que corresponde al modelo entidad-relación o (E-R) aunque también existe el relacional, si se quiere ahondar más en este tema, puede consultar la bibliografía correspondiente a Modelos de Bases de datos (Silberschatz, 2011)

##### Modelo Entidad-Relación

El modelo de datos entidad-relación (E-R) está basado en una percepción del mundo real consistente en objetos básicos llamados entidades y de relaciones entre estos objetos. Se desarrolló para facilitar el diseño de bases de datos permitiendo la especificación de un esquema de la empresa que representa la estructura lógica completa de una base de datos. El modelo de datos E-R es uno de los diferentes modelos de datos semánticos; el aspecto semántico del modelo yace en la representación del significado de los datos. El modelo E-R es extremadamente útil para hacer corresponder los significados e interacciones de las empresas del mundo real con un esquema conceptual. Debido a esta utilidad, muchas herramientas de diseño de bases de datos se basan en los conceptos del modelo E-R.

#### Sistema de Gestión de Base de Datos

Un sistema gestor de bases de datos (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos.

La colección de datos, normalmente denominada base de datos, contiene información relevante para una empresa. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente. (Silberschatz, 2011)

##### Posgresql

PostgreSQL es un gestor de bases de datos orientadas a objetos (SGBDOO o ORDBMS en sus siglas en inglés) muy conocido y usado en entornos de software libre porque cumple los estándares SQL92 y SQL99, y también por el conjunto de funcionalidades avanzadas que soporta, lo que lo sitúa al mismo o a un mejor nivel que muchos SGBD comerciales. (Ginestá, 2014)

##### Mysql

MySQL es un sistema gestor de bases de datos (SGBD, DBMS por sus siglas en inglés) muy conocido y ampliamente usado por su simplicidad y notable rendimiento. Aunque carece de algunas características avanzadas disponibles en otros SGBD del mercado, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha. Esto y su libre distribución en Internet bajo licencia GPL le otorgan como beneficios adicionales (no menos importantes) contar con un alto grado de estabilidad y un rápido desarrollo. (Ginestá, 2014)

#### SQL

Los lenguajes formales proporcionan una notación concisa para la representación de consultas. Sin embargo, los sistemas de bases de datos comerciales necesitan un lenguaje de consultas cómodo para el usuario. SQL usa una combinación de álgebra relacional y construcciones del cálculo relacional.

Aunque el lenguaje SQL se considere un lenguaje de consultas, contiene muchas otras capacidades además de la consulta en bases de datos. Incluye características para definir la estructura de los datos, para la modificación de los datos en la base de datos y para la especificación de restricciones de seguridad. (Silberschatz, 2011)

### Lenguaje de Modelado Unificado (UML)

UML (Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se ha convertido en el estándar de facto de la industria, debido a que ha sido concebido por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh. Estos autores fueron contratados por la empresa Rational Software Co. para crear una notación unificada en la que basar la construcción de sus herramientas CASE. En el proceso de creación de UML han participado, no obstante, otras empresas de gran peso en la industria como Microsoft, Hewlett-Packard, Oracle o IBM, así como grupos de analistas y desarrolladores. (Grau, 2014)



Ilustración 5 Logo de UML

#### Diagrama de Clases

Los diagramas de clases describen la estructura estática de un sistema. Las cosas que existen y que nos rodean se agrupan naturalmente en categorías. Una clase es una categoría o grupo de cosas que tienen atributos (propiedades) y acciones similares. (UML, 2014)

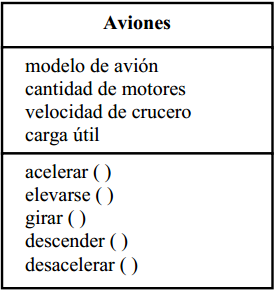


Ilustración 6 Diagrama de la clase Avión con sus Atributos y Funciones.

#### Diagramas de Casos de Uso

Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Es una herramienta valiosa dado que es una técnica de aciertos y errores para obtener los requerimientos del sistema, justamente desde el punto de vista del usuario. Los diagramas de caso de uso modelan la funcionalidad del sistema usando actores y casos de uso. Los casos de uso son servicios o funciones provistas por el sistema para sus usuarios. (UML, 2014)

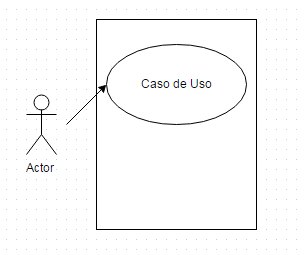


Ilustración 7 Sistema en UML

El rectángulo representa los límites del sistema que contiene los casos de uso. Los actores se ubican fuera de los límites del sistema.



Ilustración 8 Casos de Uso en UML

Se representan con Óvalos. La etiqueta en el óvalo indica la función en el sistema.



Ilustración 9 Actor en UML

Los Actores son los usuarios de un sistema.

### Software Libre y Código Abierto

El software de código abierto se desarrolla -en general- a través del esfuerzo colectivo, aunque no necesariamente coordinado, de cientos o miles de programadores aunados por una participación voluntaria, no formalizada y no siempre remunerada, que utilizan las facilidades de comunicación que ofrece Internet para actuar en forma masiva y paralela.

Conforman así comunidades virtuales, cuyos miembros operan en forma cooperativa, interactiva y descentralizada, con permanente revisión y realimentación entre pares o iguales (peer review), creando software como bien público. (Saroka, 2014)

### Hardware Libre

#### Arduino

Un Arduino contiene un microchip, que es un ordenador muy pequeño que podemos programar. Se le pueden conectar sensores para que pueda medir condiciones (como cuanta luz hay en la habitación). Puede controlar otros objetos para que reaccionen a esas condiciones (si la habitación está a oscuras enciende el led). (Playground, 2014)

### Control de Versiones

Se llama control de versiones a los métodos y herramientas disponibles para controlar todo lo referente a los cambios en el tiempo de un archivo.

Difícilmente un archivo de código o un documento de texto están terminado con la primera escritura; necesita cambios o reescrituras para corregir errores, modificar su contenido. A medida que el documento cambia existen dos opciones, mantener un historial de cambios o dejar que evolucione sin memoria. El control de versiones es un método estándar para mantener esta memoria haciendo además que sea útil para el desarrollo futuro. (Versiones, 2006)

#### Git

Git es el sistema de control de versiones distribuido de fuente abierta que facilita las actividades de Github en su computadora portátil o de escritorio. Git es el sistema de control de versiones distribuido de fuente abierta que facilita las actividades de Github en su computadora portátil o de escritorio. (Github, 2014)

### Computación en la nube (Cloud Computing)

Nos enfrentamos al nuevo paradigma de la computación en nube o cloud computing, según el cual, cualquier cosa que pueda hacerse en informática puede trasladarse a la nube o lo que es lo mismo, a la Red. Este modelo implica el uso de recursos informáticos como un suministro más, igual que si se tratara de la electricidad o el teléfono. Estos recursos son ofrecidos por proveedores de cloud, que los gestionan en grandes centros de datos remotos y prestan servicio a múltiples clientes que acceden a ellos a través de cualquier dispositivo conectado a Internet. (Computing, 2010)

#### Openshift

Openshift Express es una plataforma (PaaS) gratuita para el despliegue de aplicaciones en la nube proporcionada por RedHat. En ella podremos desplegar aplicaciones Java, Perl, PHP, Python y Ruby.  Además, permite la instalación (también gratuita) de un servidor de bases de datos como MySQL, PostgreSQL o MongoDB. (RedHat, 2014)

### Responsive Design

El término Responsive Web Design se acuña en la búsqueda de estas soluciones, concretamente la idea nació en el año 2008 por el consorcio W3C en su recomendación “Mobile Web Best Practices” [4] que definieron el concepto “OneWeb”, que se basa en la idea de realizar un diseño web para todos y accesible desde cualquier dispositivo, buscando que la experiencia del usuario se enriquezca.

“OneWeb” significa proporcionar, en la medida de lo posible, la misma información y servicios disponibles para todos los usuarios, independientemente del dispositivo que estos empleen. (Vega, 2014)

### Mockup

Un wireframe o Mockup básicamente es un **boceto básico y de baja calidad** del desarrollo de una página web o el diseño de una interfaz, la finalidad de este es el mostrar al cliente un diseño o boceto rápido y facilitar la comunicación **entre cliente y desarrollador.**

## MARCO LEGAL

La política (Resolución 305, Anexo 18) se puede enmarcar dentro del derecho de autor, en principio, dentro de la promoción del absoluto respeto al derecho de autor; Dado que las licencias de Software Libre, es decir, aquellas licencias que predican las cuatro libertades permiten el estudio, la copia y ejecución, la modificación y la distribución del software licenciado, protegiendo esas libertades. Se puede decir que el ejercicio de las libertades que otorga en software libre se lleva a cabo bajo el marco de la absoluta legalidad, ya que desde un principio se brinda al usuario la posibilidad de copiarlo, modificarlo y distribuirlo, lo contrario a lo que ocurre con el software propietario, ya que simplemente no otorga esas libertades

Esto es claramente obvio, sin embargo tiene sentido cuando se contextualiza, es decir cuando se revisan las prácticas comunes sobre el software que instalan los usuarios.

Por eso es algo complicado referirse a aspectos generales, a excepción del absoluto respeto por los derechos de autor, cuando a políticas de promoción y uso de software libre y su contexto en esas políticas. Creo que lo que se debe hacer es desarrollar puntos específicos sobre esas políticas en el derecho de autor, por ejemplo, el artículo 26 de la resolución 305 de 2008 habla de la interoperabilidad y homogenización de la información, sin profundizar mucho sobre el tema, puede decirse que, desde el derecho de autor, la promoción del uso de software libre propone una solución a la interoperabilidad en cuanto a formatos se refiere, dado que al usarse formatos libres el acceso a ellos puede predicarse de cualquier persona, toda vez que éstos son de libre acceso, no puede exigírsele, ni a la administración, ni a la ciudadanía que pretende el acceso a los servicios de la administración el uso de una aplicación privativa para poder tener ese acceso, en pocas palabras, no se le debería tener que exigir a un ciudadano el uso de Word para poder entregar un documento en formato digital.

De alguna forma, si esto fuese así, se le estará negando el acceso a los servicios de la administración a un ciudadano o incitando la infracción de los derechos de autor, al obligar a un ciudadano a conseguir software privativo, esto sin considerar muchísimas otras problemáticas que surgen con una exigencia, que aunque en principio parece nimia (insignificante), propone grandes obstáculos para el acceso a los servicios de la administración.

También se puede hablar de los derechos de autor en lo que tiene que ver con “las posibilidades y limitaciones del modelo de licenciamiento de la solución seleccionada” a la que se refiere el numeral tres de los Lineamientos Generales de los que habla el anexo 18, ya que claramente el uso de Software Libre propone una solución y una ventaja frente a éste aspecto contrapuesto al uso de software propietario.

Debe abordarse el tema también desde lo que tiene que ver con las libertades mismas del uso de software libre, y éste es un asunto que como tal no es plasmado tácitamente en la Resolución 305 ni en el anexo 18, es fundamental abordar los temas de la distribución y redistribución de versiones modificadas, sobre todo cuando se habla en el contexto de lo educativo. Éste aspecto debe mencionarse respecto de sus ventajas como generador de conocimiento, un asunto, que puede ser de absoluta y vital importancia.

Todo éste asunto debe considerarse desde un punto de vista muy técnico, ya que cuando a la implementación de las tics en la administración pública se refiere, en el fondo todo se reduce a la aplicación de determinadas políticas en una plataforma tecnológica. Por eso si se quiere contextualizar el derecho de autor como un elemento de consideración en la implementación de políticas de promoción y uso de software libre, se deben precisar claramente los elementos técnicos que tiene un punto de encuentro con los derecho de autor o bien, cualquiera de las materias de la propiedad intelectual (por ejemplo: Las patentes, la propiedad marcaría, entre otros), finalmente de eso se trata. Si esto no se hace adecuadamente se corre el riesgo de desnaturalizar uno de los principios mismos que buscar el uso de software libre: el uso permitido de las aplicaciones licenciadas. (Observatoio Software Libre, 2014)

En este mapa mental se observa como en la normatividad colombiana se agregó el software a la propiedad intelectual como una obra literaria y a su vez como el software libre entra dentro de ese tipo de obra.

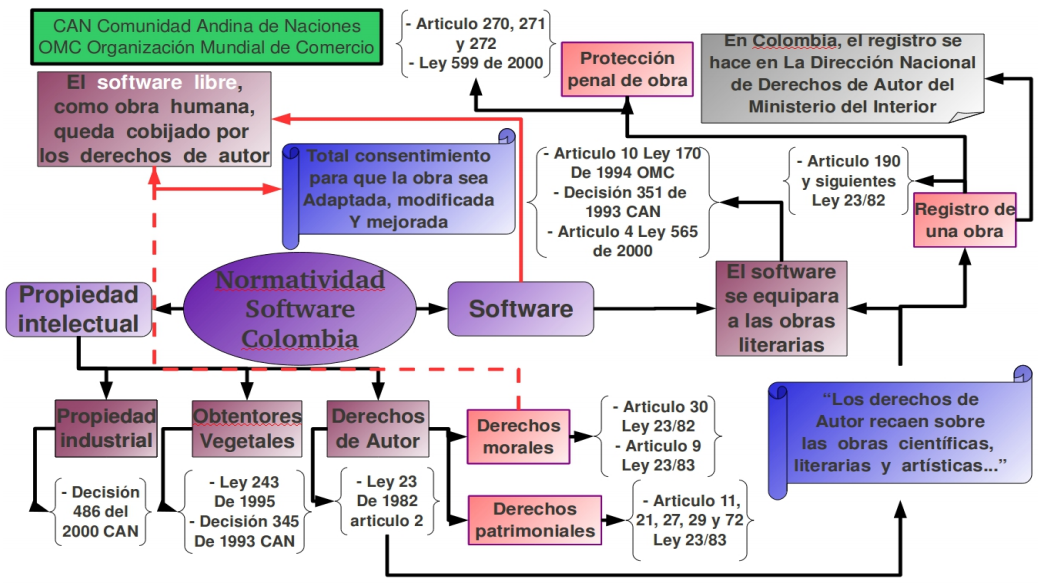


Ilustración 10 Normatividad Colombiana en Software

# ALCANCES Y LIMITACIONES

Con la creación de esta aplicación para dispositivo móvil y de su correspondiente página web, se pretende crear un impacto en la forma como hasta ahora se monitorean los invernaderos en la región, existirá una primera limitación que será el servicio de internet, ya que en la granja de la Universidad Surcolombiana no se cuenta con este activo, para esto se plantea como solución implementar en el computador donde se recibirán los datos un modem que provea los servicios de internet y de esta manera garantizar el acceso a este y poder tener los datos en tiempo real. Pero se dejará la sugerencia de que se implemente el servicio de internet para lograr que esos datos sean subidos directamente al servidor, en vez de que sean almacenados en una base de datos local como actualmente ocurre.

# METODOLOGÍA

Las fronteras entre estos tipos de proyecto no son totalmente nítidas ni están inequívocamente delimitadas, y en consecuencia, suele haber procedimientos, metodologías, recursos técnicos y derroteros comunes en la realización de cualquiera de ellos. Hay, sin embargo, rasgos distintivos que se exponen a continuación:

a. Proyectos de intervención

b. Proyectos de evaluación

c. Proyectos de desarrollo tecnológico

d. Proyectos de investigación

e. Proyectos de investigación-acción.

En este caso y para este proyecto, la metodología que se aplicará será de desarrollo tecnológico, El sello distintivo de este tipo de proyecto es que se orienta hacia la obtención de productos tangibles: un medio diagnóstico, un software para empresas, un modelo para la predicción del rendimiento académico, un currículo, maquetas, modelos experimentales o medios auxiliares de enseñanza, aplicaciones web, entre otros tantos que puedan aplicar como productos tangibles generados a partir del conocimiento tecnológico.

La obtención del producto se acompaña, casi inevitablemente, de la evaluación de sus propiedades. Si es un medio diagnóstico evaluado `in vivo', su sensibilidad, su especificidad, su valor predictivo, sus atributos de costo-beneficio, su inocuidad, sus riesgos o sus eventuales efectos adversos; si es un modelo predictivo, su ajuste, su capacidad predictiva, la relevancia de los predictores que lo componen, etc. La importancia o el peso relativo que se atribuya en el proyecto a la obtención del producto o a la evaluación de sus atributos, determina si se trata de un proyecto de desarrollo, de un proyecto de investigación, o inclusive, aunque menos frecuentemente, de un proyecto de evaluación, de acuerdo a los plazos y al tipo de diseño. (tesis, 2006)

## TIPO DE INVESTIGACIÓN

De todos los tipos de investigación que pueden ser aplicados para un proyecto de grado, se decide; luego de analizar el fin de este proyecto, sus alcances y limitaciones utilizar un tipo de investigación Aplicada la cual se define en este concepto:

INVESTIGACIÓN APLICADA: Es utilizar los conocimientos obtenidos en las investigaciones en la práctica, y con ello traer beneficios a la sociedad. Un ejemplo es el protocolo en la investigación médica. O el desarrollo de software para diferentes medios. (Investigacion, 2014)

## MÉTODO

La observación directa será el método principal con el que se obtendrán los datos y planearán los pasos a seguir en las posteriores etapas del proyecto.

La observación es participante cuando para obtener los datos el investigador se incluye en el grupo, hecho o fenómeno observado, para conseguir la información "desde adentro".

Al nosotros como equipo de trabajo, ir hasta la granja, ver el estado en el que estaba, tomar la información y las sugerencias de mejora por parte de los compañeros que estaban trabajando en proyectos similares, detectar los posibles problemas que se pudiesen presentar a la hora de trabajar en las mejoras, crear una idea y posteriormente luego de analizar si era viable, comenzar a trabajar en ella para luego tener unos productos tangibles que luego puedan ser aplicados por parte de nosotros como grupo de investigación o de compañeros que decidan acoger nuestra idea con apoyo de la universidad, estamos implícitos en la investigación directa, y esto fundamenta por qué se ha escogido esta opción frente a otras alternativas o formas de adquirir información.

# DISEÑO METODOLOGICO

## METODOLOGÍA DE DESARROLLO

La aplicación se desarrolló bajo los lenguajes de programación java, JavaScript, soportando un acceso a base de datos MySQL y PostgreSQL, gracias a los lenguajes aprendidos durante el desarrollo de la Carrera de Tecnología se logró concretar la viabilidad del proyecto y llegar a su culminación.

En cuanto a la metodología utilizada se optó por aquella que se adapta más a nuestro medio, conocida como Waterfall Model o modelo en cascada, también consignado en algunos libros como desarrollo en cascada.

A continuación se describirá la conceptualización de la misma, con el fin de justificar su aplicación en éste Proyecto.

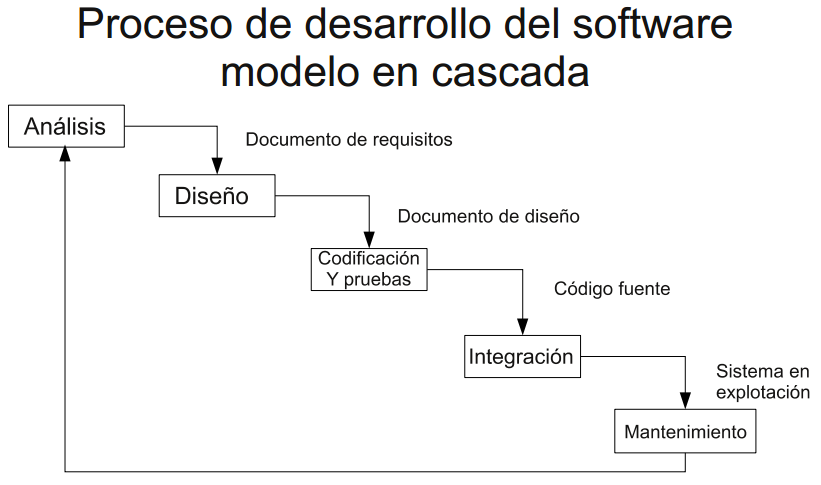


Ilustración 11 Desarrollo en Cascada

La**metodología** en cascada, también llamado modelo en cascada, es el enfoque metodológico que ordena rigurosamente las etapas del [proceso para el desarrollo de software](http://es.wikipedia.org/wiki/Proceso_para_el_desarrollo_de_software), de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior.

* Análisis: Necesidades del usuario. (especificaciones)
* Diseño: Descomposición en elementos que puedan desarrollarse por separado. (especificaciones de cada elemento)
* Codificación: Programación de cada elemento por separado. (pruebas aisladas)
* Integración: Se juntan los elementos y se prueba el sistema completo.
* Mantenimiento: Cambios ocasionales. (errores o mejoras) (cascada, 2014 )

Existen pros y contras a la hora de utilizar el modelo en cascada como método para el desarrollo de un software, se trabajará por encima de los otros modelos principalmente por qué es vital ir terminando una parte del trabajo para luego comenzar con la otra; claro está que esto también es una ventaja, puesto que si al estar en alguna fase y se detecta algún error, se puede trabajar en su corrección puntual, evitando parar toda la etapa para centrarse en la solución del mismo.

Luego de tener un concepto básico de que modelo se trabajará y sus razones de uso, se procederá a mencionar cada etapa y lo que se ha realizado en cada una de ellas.

## ANALISIS

### Documentación y Diagrama de Casos de Uso

|  |  |
| --- | --- |
| ACT–01 | Administrador |
| Descripción | Este Actor tendrá privilegios para realizar consultas detalladas. Además podrá consultar desde la aplicación Móvil. |
| Comentarios | ninguno |

Tabla 1 Actor 1 Administrador

|  |  |
| --- | --- |
| ACT–02 | Invitado |
| Descripción | Este Actor podrá acceder a una consulta simple desde la web. Además podrá consultar desde la aplicación Móvil. |
| Comentarios | Ninguno |

Tabla 2 Actor 2 Invitado

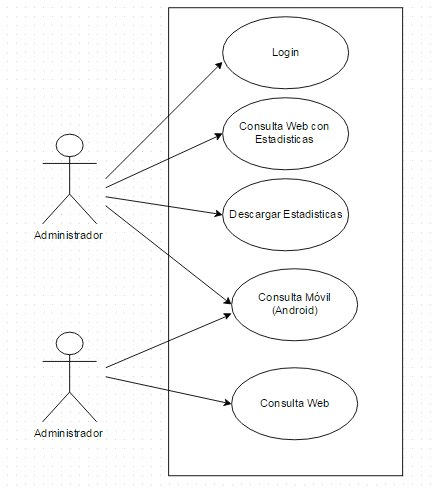


Ilustración 12 Diagrama de Casos de Uso

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de Uso** | **Login- Consulta Web** | | |
| **Actor** | **Administrador** | | |
| **Curso Normal** | | | **Alternativa** |
| 1. El Usuario Administrador ingresa sus correspondientes datos de login. | |  | |
| 1. El login lo redireccionará a la página correspondiente, donde encontrará las consultas detalladamente. | | 1. ) Si los datos de login son incorrectos, se informará al usuario que ingrese nuevamente los datos. | |

Tabla 3 Caso de Uso Login y Consulta Web - Administrador

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de Uso** | **Descargar Estadística** | | |
| **Actor** | **Administrador** | | |
| **Curso Normal** | | | **Alternativa** |
| 1. El Usuario Administrador se ubica sobre una de las estadísticas y en su respectivo menú selecciona el formato en que desea la descarga. | |  | |

Tabla 4 Caso de Uso 2 Descarga de Estadísticas - Administrador

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de Uso** | **Consulta Web** | | |
| **Actor** | **Invitado** | | |
| **Curso Normal** | | | **Alternativa** |
| 1. El usuario ingresa desde el botón de invitado. | |  | |
| 1. La página lo redireccionará a su respectiva página donde encontrará una consulta sencilla. | |  | |

Tabla 5 Caso de Uso 3 Consulta web - Invitado

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de Uso** | **Consulta Móvil** | | |
| **Actor** | **Administrador-Invitado** | | |
| **Curso Normal** | | | **Alternativa** |
| 1. El usuario descargará la aplicación desde el Google Play. | |  | |
| 1. El usuario ingresará en modo online, para cargar los datos desde la Web. | | 1. ) Si se ha agotado el tiempo de espera, la aplicación informará que lo intente de nuevo más tarde. | |
| 1. La aplicación informará que los datos están disponibles de forma Offline. | |  | |
| 1. El usuario ingresa en modo Offline, donde consultará los datos de la última consulta Online que realizó. | |  | |

Tabla 6 Caso de Uso 4 Consulta Móvil - Administrador e Invitado

### Diagramas de Clase

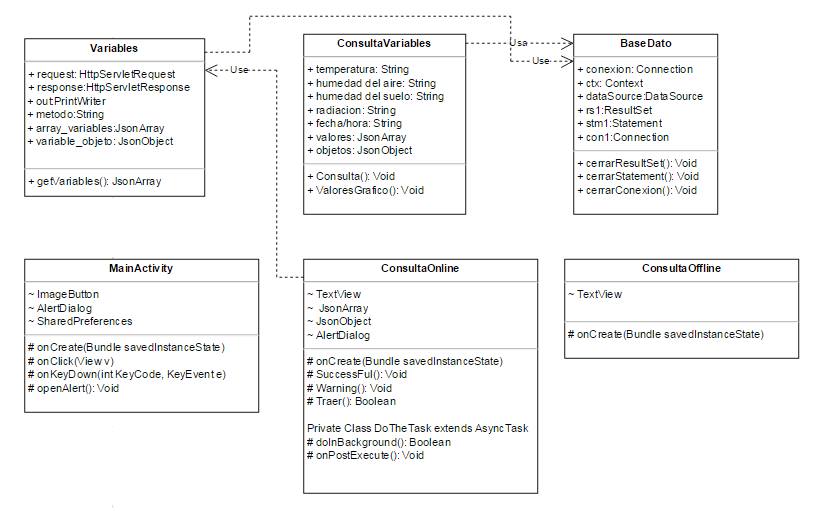


Ilustración 13 Diagrama de Clases

### Modelado de la Base de Datos

#### Modelo E-R

Cuando se llegó al invernadero a recolectar la información, existía un modelo de base de datos, el cual cumplía con su objetivo, pero esto no quiere decir que cumpliese con todas las normas o estándares que a lo largo de la carrera hemos estudiado.

Se presenta a continuación tres modelos de base de datos, el primero modelo es el original ya mencionado, el segundo es el que para este proyecto de grado se ha implementado el cual es una mejora que nace a partir del primer modelo de base de datos y el tercero es el ideal, puesto que está abierto a posibles mejoras y esto evita que a futuro se tenga que desmontar la base de datos existente para hacer cambios.

##### Modelo Original



Ilustración 14 Modelo original de la Base de Datos

##### Modelo Implementado

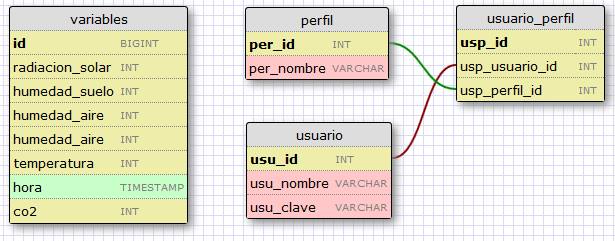


Ilustración 15 Modelo Actual de la Base de Datos

##### Modelo Propuesto

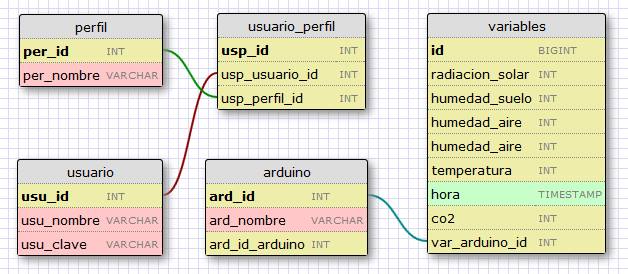


Ilustración 16 Modelo propuesto de la Base de Datos

#### Diccionario De Datos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TABLA VARIABLES** | | | | |
| **N°** | **Nombre del campo** | **Descripción** | **Longitud** | **Tipo** |
| 1 | Id | Identificador único para cada registro. |  | Bigserial |
| 2 | radiacion\_solar | Radiación solar que se produce en el invernadero | 2 | Integer |
| 3 | humedad\_suelo | Humedad que presenta el suelo del invernadero | 3 | Integer |
| 4 | humedad\_aire | Humedad del aire que se produce en el invernadero | 3 | Integer |
| 5 | Temperatura | Temperatura del invernadero | 3 | Integer |
| 6 | Co2 | Dióxido de carbono que se produce en el invernadero | 3 | Integer |
| 7 | Hora | Fecha y hora en que se ingresa el registro |  | Timestamp |

Tabla 7 Diccionario de Datos - Tabla Variables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TABLA PERFIL** | | | | |
| **N°** | **Nombre del campo** | **Descripción** | **Longitud** | **Tipo** |
| 1 | Id | Identificador único para cada registro. |  | Bigserial |
| 2 | per\_nombre | Nombre del tipo de usuario. | 10 | Character Varying |

Tabla 8 Diccionario de Datos - Tabla Perfil

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TABLA USUARIO** | | | | |
| **N°** | **Nombre del campo** | **Descripción** | **Longitud** | **Tipo** |
| 1 | Usu\_Id | Identificador único para cada registro. |  | Serial |
| 2 | usu\_nombre | Nombre del Usuario | 10 | Character Varying |

Tabla 9 Diccionario de Datos - Tabla Usuario

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TABLA USUARIO PERFIL** | | | | |
| **N°** | **Nombre del campo** | **Descripción** | **Longitud** | **Tipo** |
| 1 | Usp\_Id | Identificador único para cada registro. |  | Serial |
| 2 | Usp\_usuario\_id | Id del usuario. |  | Integer |
| 3 | Usp\_perfil\_id | Id del tipo de perfil |  | Integer |

Tabla 10 Diccionario de Datos - Tabla Usuario-Perfil

## DISEÑO

Cumplida la parte de análisis, se recoge la información y con base en esta se da inicio a la creación de, en primera medida a los bosquejos tanto de la página web como de la aplicación móvil; y con estos ya estructurados se pasa a los diseños finales.

Primero se mostrará un poco el código utilizado para realizar la parte gráfica de la aplicación o los también llamados Layouts.

Activity\_main.xml

<RelativeLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

xmlns:tools=*"http://schemas.android.com/tools"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"*

tools:context=*"com.example.proyectogrado.MainActivity"*

android:background=*"@android:color/transparent"*>

<ImageView

android:id=*"@+id/imageView1"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentTop=*"true"*

android:adjustViewBounds=*"true"*

android:background=*"@android:color/transparent"*

android:src=*"@drawable/main\_header"* />

<ImageView

android:id=*"@+id/btn\_temperatura"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_below=*"@+id/imageView1"*

android:layout\_centerHorizontal=*"true"*

android:layout\_marginTop=*"10dp"*

android:src=*"@drawable/intro\_texto"* />

<ImageButton

android:id=*"@+id/btn\_offline"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignRight=*"@+id/btn\_temperatura"*

android:layout\_below=*"@+id/btn\_temperatura"*

android:layout\_marginRight=*"41dp"*

android:layout\_marginTop=*"20dp"*

android:background=*"@android:color/transparent"*

android:src=*"@drawable/intro\_btn\_offline"* />

<ImageButton

android:id=*"@+id/btn\_online"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignLeft=*"@+id/text\_temperatura"*

android:layout\_alignTop=*"@+id/btn\_offline"*

android:layout\_marginLeft=*"34dp"*

android:background=*"@android:color/transparent"*

android:src=*"@drawable/intro\_btn\_online"* />

<ImageView

android:id=*"@+id/text\_temperatura"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentBottom=*"true"*

android:layout\_centerHorizontal=*"true"*

android:layout\_marginBottom=*"20dp"*

android:src=*"@drawable/intro\_fondo"* />

</RelativeLayout>

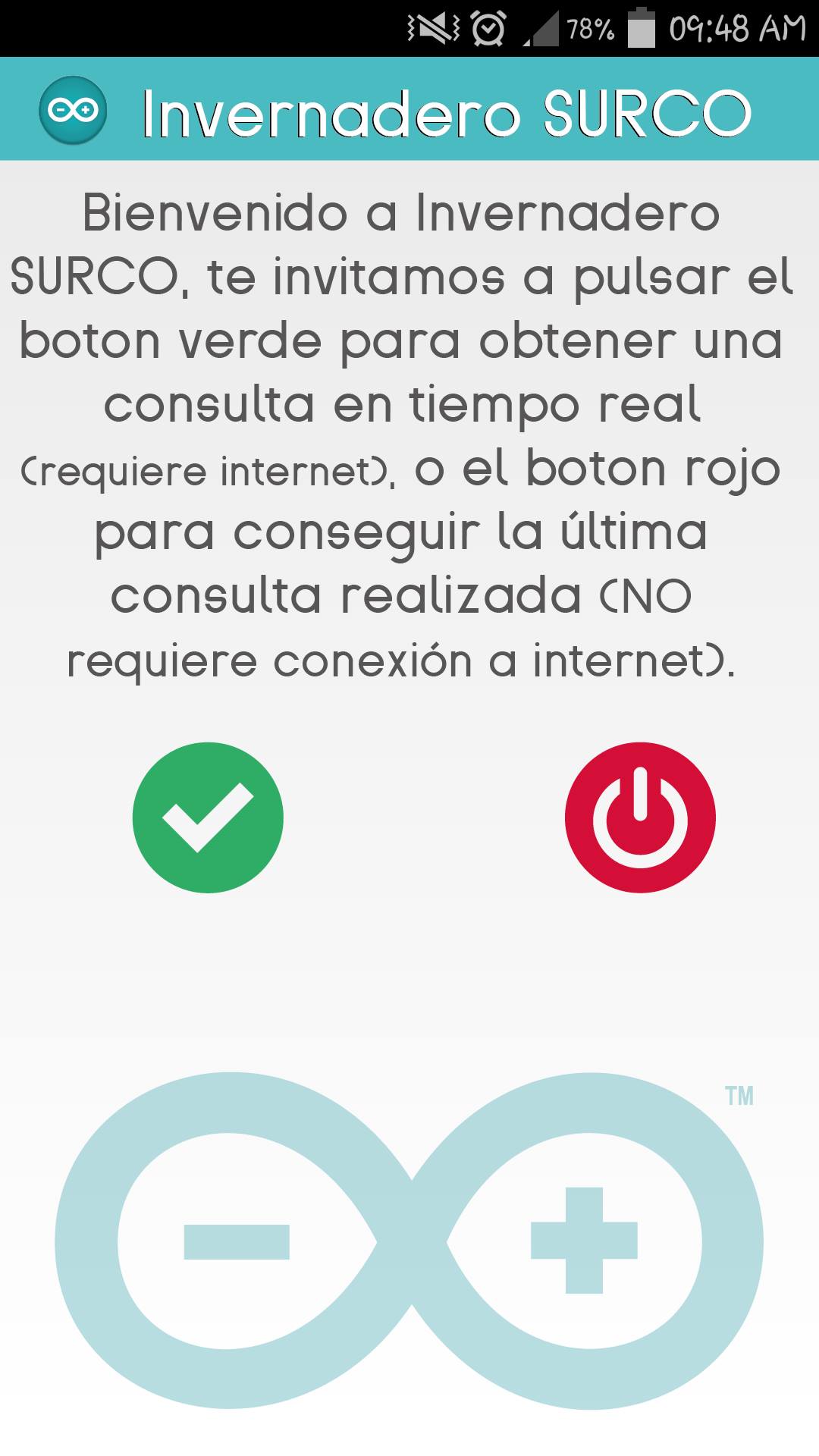


Ilustración 17 Diseño final Activity\_main.xml

El layout de consultaOnline es prácticamente el que el de ConsultaOffline por lo que se coloca solamente uno.

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<RelativeLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:id=*"@+id/RelativeLayout1"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"*

android:orientation=*"vertical"*

android:background=*"@drawable/main\_fondo"* >

<ImageView

android:id=*"@+id/imageView1"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentTop=*"true"*

android:layout\_centerHorizontal=*"true"*

android:src=*"@drawable/main\_header"* />

<ImageButton

android:id=*"@+id/btn\_temperatura"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentLeft=*"true"*

android:layout\_below=*"@+id/imageView1"*

android:layout\_marginLeft=*"50dp"*

android:layout\_marginTop=*"30dp"*

android:src=*"@drawable/main\_btn\_temp"*

android:background=*"@android:color/transparent"* />

<ImageView

android:id=*"@+id/text\_temperatura"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignTop=*"@+id/btn\_temperatura"*

android:layout\_toRightOf=*"@+id/btn\_temperatura"*

android:src=*"@drawable/main\_txt\_temp"* />

<ImageView

android:id=*"@+id/text\_radiacion"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignLeft=*"@+id/text\_temperatura"*

android:layout\_alignTop=*"@+id/btn\_radiacion"*

android:src=*"@drawable/main\_txt\_radiacion"* />

<ImageButton

android:id=*"@+id/btn\_radiacion"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_below=*"@+id/btn\_temperatura"*

android:layout\_marginTop=*"28dp"*

android:layout\_toLeftOf=*"@+id/text\_radiacion"*

android:src=*"@drawable/main\_btn\_radiacion"*

android:background=*"@android:color/transparent"* />

<ImageButton

android:id=*"@+id/btn\_humedad\_aire"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_below=*"@+id/btn\_radiacion"*

android:layout\_marginTop=*"28dp"*

android:layout\_toLeftOf=*"@+id/text\_radiacion"*

android:src=*"@drawable/main\_btn\_humsu"*

android:background=*"@android:color/transparent"* />

<ImageView

android:id=*"@+id/imageView2"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignLeft=*"@+id/text\_radiacion"*

android:layout\_alignTop=*"@+id/btn\_humedad\_aire"*

android:src=*"@drawable/main\_txt\_humsu"* />

<ImageButton

android:id=*"@+id/imageButton1"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignLeft=*"@+id/btn\_humedad\_aire"*

android:layout\_below=*"@+id/btn\_humedad\_aire"*

android:layout\_marginTop=*"28dp"*

android:src=*"@drawable/main\_btn\_humai"*

android:background=*"@android:color/transparent"*/>

<ImageView

android:id=*"@+id/imageView3"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignLeft=*"@+id/imageView2"*

android:layout\_alignTop=*"@+id/imageButton1"*

android:src=*"@drawable/main\_txt\_humai"* />

<ImageButton

android:id=*"@+id/imageButton2"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_below=*"@+id/imageButton1"*

android:layout\_toLeftOf=*"@+id/imageView3"*

android:layout\_marginTop=*"28dp"*

android:background=*"@android:color/transparent"*

android:src=*"@drawable/main\_btn\_fecha"* />

<ImageView

android:id=*"@+id/imageView4"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignLeft=*"@+id/imageView3"*

android:layout\_alignTop=*"@+id/imageButton2"*

android:src=*"@drawable/main\_txt\_fecha"* />

<TextView

android:id=*"@+id/text\_valor\_radiacion"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_below=*"@+id/text\_radiacion"*

android:layout\_marginLeft=*"35dp"*

android:layout\_toRightOf=*"@+id/btn\_radiacion"*

android:textAppearance=*"?android:attr/textAppearanceMedium"*

android:textColor=*"#575756"* />

<TextView

android:id=*"@+id/text\_valor\_temperatura"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignLeft=*"@+id/text\_temperatura"*

android:layout\_below=*"@+id/text\_temperatura"*

android:layout\_marginLeft=*"40dp"*

android:textAppearance=*"?android:attr/textAppearanceMedium"*

android:textColor=*"#575756"* />

<TextView

android:id=*"@+id/text\_valor\_humedad\_aire"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_below=*"@+id/imageView2"*

android:layout\_marginLeft=*"40dp"*

android:layout\_toRightOf=*"@+id/btn\_humedad\_aire"*

android:textAppearance=*"?android:attr/textAppearanceMedium"*

android:textColor=*"#575756"* />

<TextView

android:id=*"@+id/text\_valor\_humedad\_suelo"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignLeft=*"@+id/imageView3"*

android:layout\_below=*"@+id/imageView3"*

android:layout\_marginLeft=*"40dp"*

android:textAppearance=*"?android:attr/textAppearanceMedium"*

android:textColor=*"#575756"* />

<TextView

android:id=*"@+id/text\_valor\_hora"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_below=*"@+id/imageView4"*

android:layout\_marginLeft=*"10dp"*

android:layout\_toRightOf=*"@+id/imageButton2"*

android:textAppearance=*"?android:attr/textAppearanceMedium"*

android:textColor=*"#575756"* />

</RelativeLayout>



Ilustración 18 Diseño final Consulta offline y online.xml

### Página web

El sketch y el diseño final se dividen en dos vistas, la primera es la que tiene el administrador con sus respectivas opciones y la segunda es la que pertenece al usuario invitado.

#### Sketch

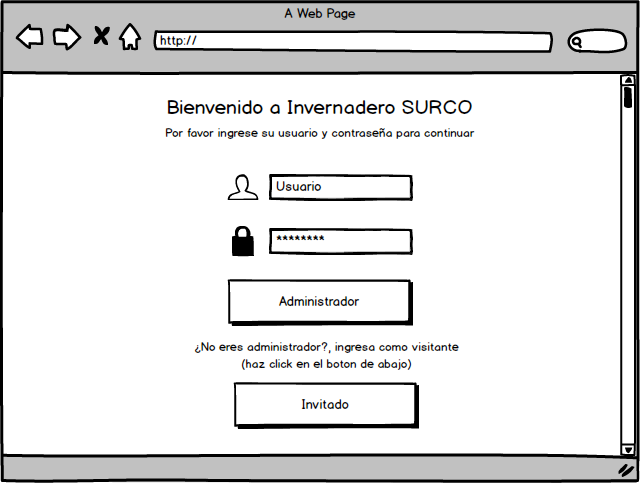


Ilustración 19 Sketch Login– vista administrador e Invitado

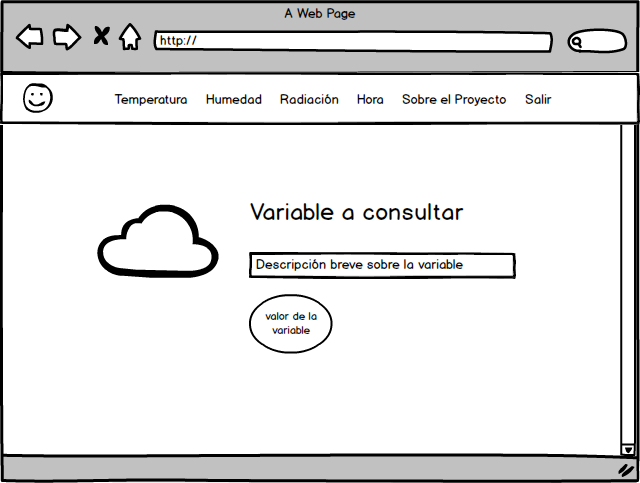


Ilustración 20 Sketch Pagina web – vista administrador. Variables



Ilustración 21 sketch Pagina web – vista administrador. Graficas



Ilustración 22 Sketch Pagina web – vista Invitado. Variables

#### Diseño final

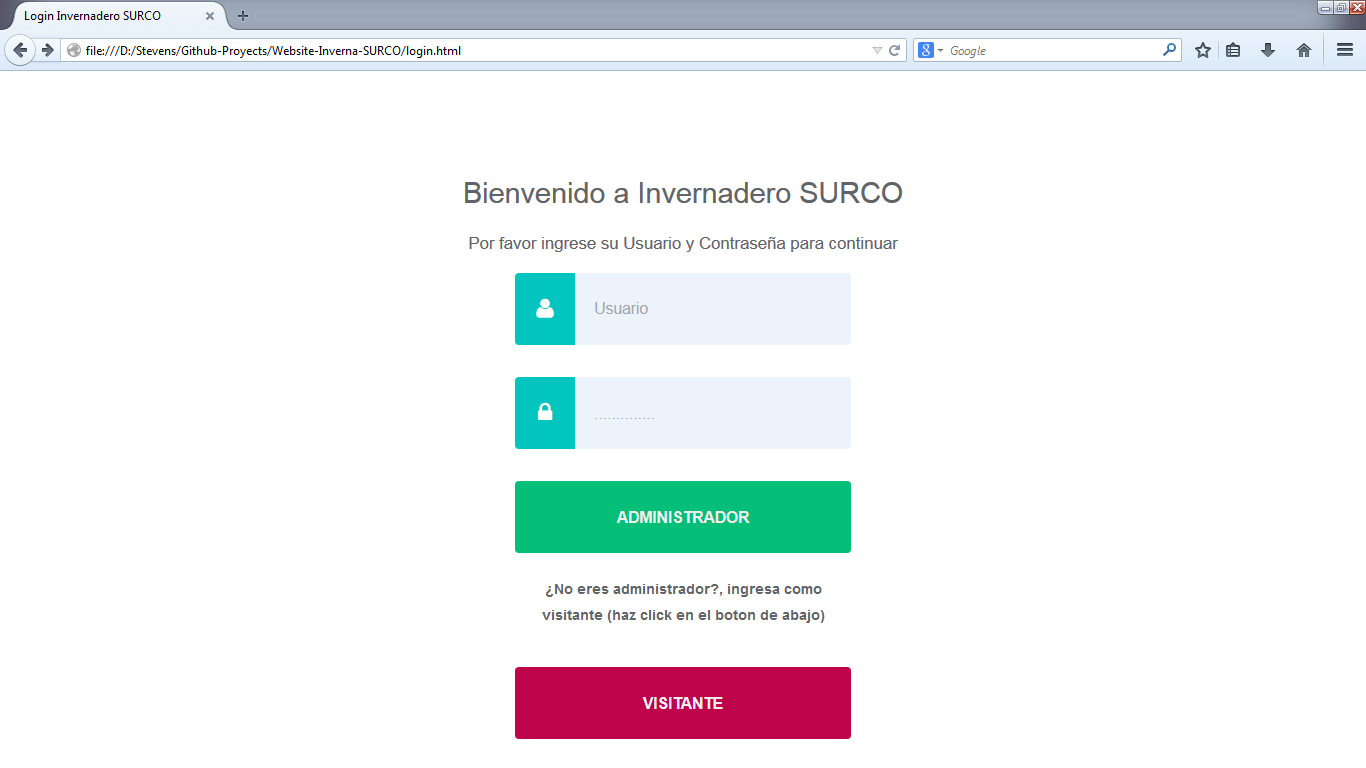


Ilustración 23 Diseño Final Login – vista Administrador e Invitado



Ilustración 24 Diseño Final Pagina web – vista administrador. Variables

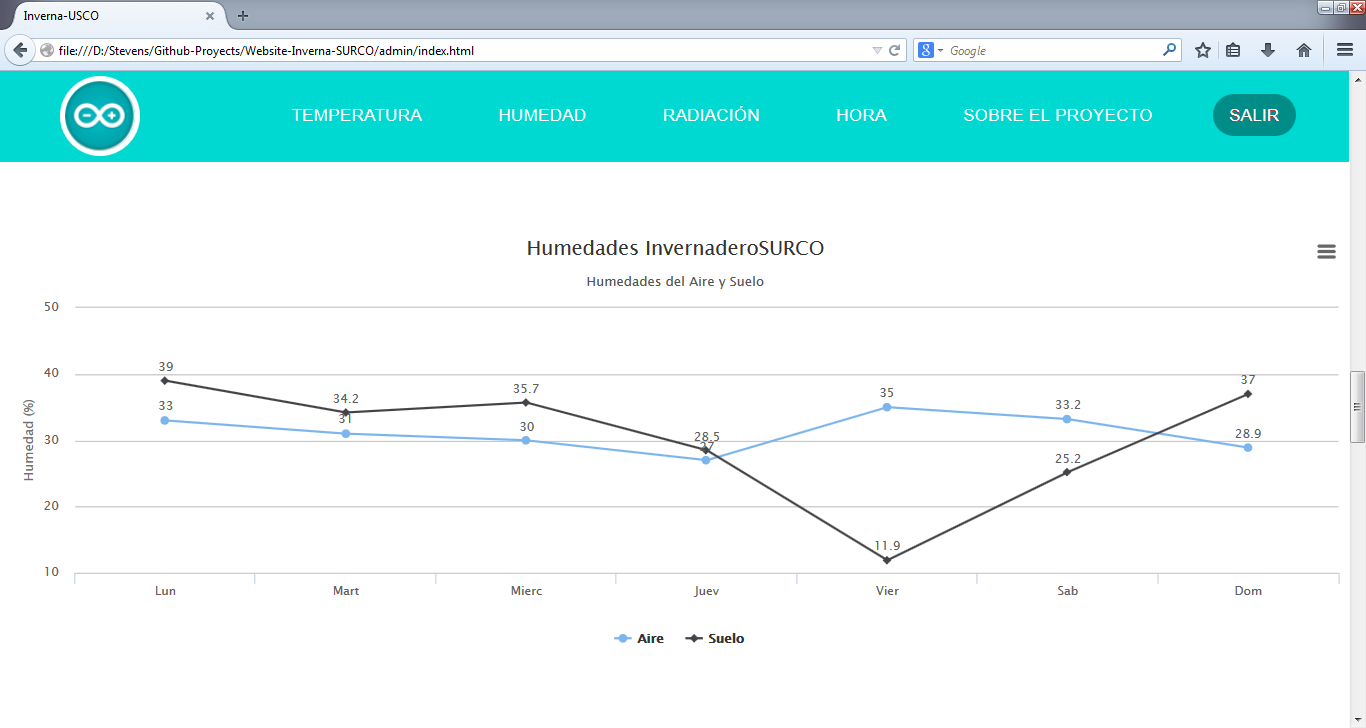


Ilustración 25 Diseño Final Pagina web – vista administrador. Graficas



Ilustración 26 Diseño Final Pagina web – vista Invitado. Variables

### Aplicación para dispositivo móvil Android

El sketch y el diseño final tienen las mismas vistas para todos dos actores, puesto que esta aplicación en su primera versión solo servirá como método de consulta en tiempo real.

#### Sketch

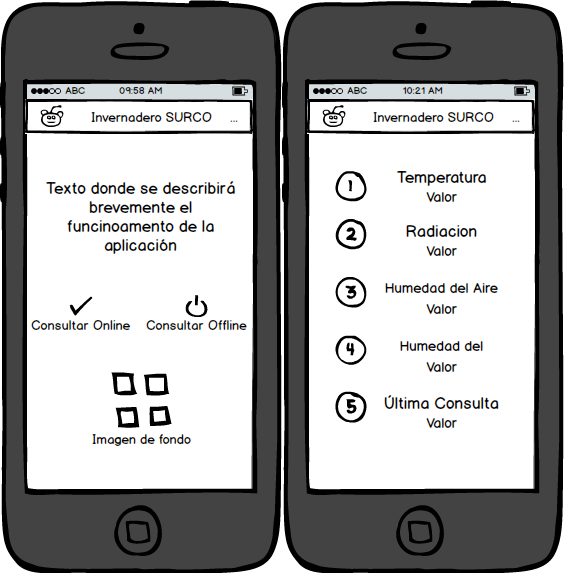


Ilustración 27 Sketch Main Activity y Activity Variables

#### Diseño final



Ilustración 28 Diseño Final Main Activity y Activity Variables

### Librerías adicionales

Las **librerías** en programación son aquellos paquetes creados por terceros que podemos agregar a nuestros proyectos para enriquecer el sistema o aprovechar ciertas funcionalidades.

Adicional a las librerías que se implementan en el trabajo de desarrollo, las cuales permiten ampliar las posibilidades a la hora de crear software.

#### Sweet Alert

Si utilizas alertas JavaScript  seguramente alguna vez te has preguntado cómo cambiar el diseño de estas, lamentablemente no es posible debido a que le corresponde a la   UI de cada navegador.

Pero no todo está perdido, sweetAlert es una herramienta con la que puedes sustituir el horrible diseño por algo más agradable visualmente, sin alterar el resultado.

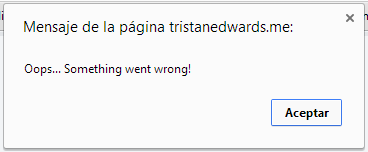


Ilustración 29 Alert tradicional

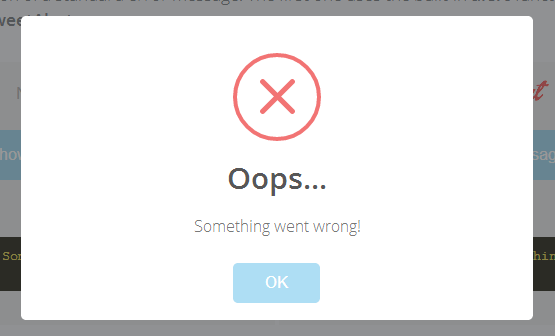


Ilustración 30 Alert con SweetAlert

Lo que aparte de esto es realmente útil es que cuenta con soporte para ser implementado en Android, y esto es realmente agradable para el usuario, que puede disfrutar de una interfaz agradable en las plataformas donde estará disponible el producto final.

Enlace a la página web y proyecto en Github.

[Página Oficial](http://tristanedwards.me/sweetalert)

[Repositorio en Github](https://github.com/t4t5/sweetalert)

#### Highcharts

Highcharts es una librería desarrollada en JavaScript que permite crear fácil y rápidamente gráficos interactivos.

Trae de serie un nutrido pack de gráficos: área, columnas, barras, pasteles (o quesitos), etc. Simplemente necesitas añadir un set de datos (acepta JSON, CSV, XML) y ya puedes incorporarlo en cualquier proyecto web. Además puedes configurar prácticamente toda la interfaz de usuario con la imagen corporativa que desees (colores, fuentes, tamaños, idiomas, etc.).

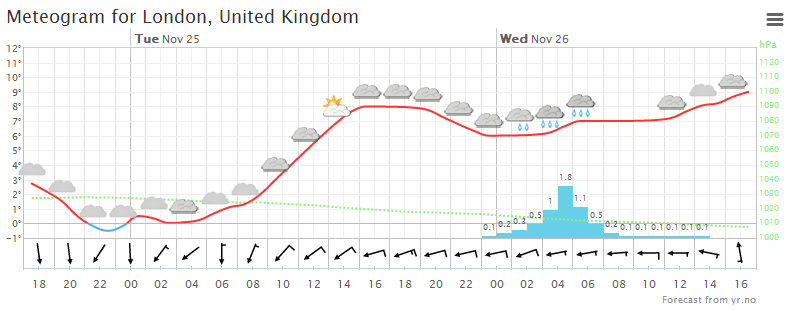


Ilustración 31 Ejemplo de Gráficas conseguidas con Highcharts

Una ventaja que tiene Highcharts frente a sus similares, es que permite la descarga de los gráficos en diferentes formatos, (pdf, Jpg, cvs, etc.). Brindándole al usuario la posibilidad de guardar su información o descargarla para su posterior análisis.

Enlace a la página web y proyecto en Github.

[Página Oficial](http://www.highcharts.com/)

[Repositorio en Github](https://github.com/highslide-software/highcharts.com/)

## DESARROLLO

Esta es la parte clave de todo el proyecto de grado, en esta sección se hablará y explicará paso a paso toda la estructura interna del proyecto, las clases, los Servlet, los métodos, su función en el proyecto, su uso e importancia tanto en Web como Android.

Otro aspecto importante del que se tratará en esta etapa de desarrollo es la seguridad de la información, esto es posible gracias a JAAS que nos permite accede a servicios de autenticación y acceso. De igual manera el cifrado de las contraseñas es vital que se realice en la base de datos para evitar fraudes por robo o intentos de ataques por métodos de inyección o fuerza bruta. Esto se logra gracias a una extensión del Gestor de Base de Datos de PostgreSQL llamada PgCrypto.

El proceso de pasar de un servidor local (LocalHost) el cual está delimitado a la red local en la cual es creado, a un servidor en la nube o Cloud Computing, que en este caso es Openshift será tratado en dos etapas, la primera será una breve explicación de lo que se trabajó y de su importancia; tanto en esta, como en toda las etapas faltantes del proyecto. Y la segunda contará con un manual detallado en los anexos, de su creación y su respectiva implementación.

### Java

Una de las Clases claves en este desarrollo es la clase BaseDato, La cual se encarga de invocar un DataSource del Pool de Conexiones del Servidor JBoss.

Esto se realiza con el fin de implementar las buenas prácticas en el desarrollo de un software, en este caso sería lograr que la conexión sea reutilizable y al mismo tiempo más segura.

**public** **class** BaseDato {

**public** Connection getConexion(String perfil) {

Connection conexion = **null**;

String datasource = "";

System.*out*.println("Perfil: " + perfil);

/\* el Operador if se encarga de validar

\* si el usuario de donde se está invocando el método

\* tenga los respectivos privilegios

\* para realizar una consulta

\*/

**if** (perfil.equals("ADMINISTRADOR")) {

datasource = "java:jboss/gradoDS";

**try** {

Context ctx = **new** InitialContext();

DataSource dataSource = **null**;

dataSource = (DataSource) ctx.lookup(datasource);

conexion = dataSource.getConnection();

}

**catch** (Exception e) {

e.printStackTrace();

System.*out*.println("getConexion(1) - ERROR: " + datasource + " :: " + e.toString());

}

}

**return** conexion;

}

Los métodos de la clase BaseDato tienen una función simple pero vital, esta clase es la encargada de cerrar todas las conexiones que se encuentran abiertas de tal manera que al realizar varias consultas, la Base de Datos no colapse.

**public** **class** BaseDato {

// Método que cierra un Objeto de la clase ResultSet

**public** **void** cerrarResultSet(ResultSet rs1){

**if**(rs1 != **null**){

**try** {

rs1.close();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

rs1 = **null**;

}

// Método que cierra una Objeto de la Clase Statement

**public** **void** cerrarStatement(Statement stm1){

**if**(stm1 != **null**){

**try** {

stm1.close();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

stm1 = **null**;

}

// Método que cierra una Objeto de la Clase Connection

**public** **void** cerrarConexion(Connection con1){

**if**(con1 != **null**){

**try** {

con1.close();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

con1 = **null**;

}

}

La clase ConsultaVariables, es creada con el objetivo de obtener valores desde la base de datos. Como se puede observar ésta tiene objetos de la Clase String en los cuales se almacenarán los datos. En éstos objetos se implementan los Métodos Get y Set para que así se pueda acceder y modificar su valor tanto adentro como afuera de la Clase.

**public** **class** ConsultaVariables {

// Objetos de la Clase String

**private** String temp;

**private** String humedad\_aire;

**private** String humedad\_suelo;

**private** String radiacion;

**private** String hora;

// Métodos Get And Set

**public** String getHumedad\_aire() {

**return** humedad\_aire;

}

**public** **void** setHumedad\_aire(String humedad\_aire) {

**this**.humedad\_aire = humedad\_aire;

}

**public** String getHumedad\_suelo() {

**return** humedad\_suelo;

}

**public** **void** setHumedad\_suelo(String humedad\_suelo) {

**this**.humedad\_suelo = humedad\_suelo;

}

**public** String getRadiacion() {

**return** radiacion;

}

**public** **void** setRadiacion(String radiacion) {

**this**.radiacion = radiacion;

}

**public** String getHora() {

**return** hora;

}

**public** **void** setHora(String hora) {

**this**.hora = hora;

}

**public** String getTemp() {

**return** temp;

}

**public** **void** setTemp(String temp) {

**this**.temp = temp; }

En ésta clase también se implementaron los objetos de la Clase JSONArray y JSONObject los cuales están destinados para almacenar más de un dato, también se aplicaron los métodos Get y Set para acceder o modificar su valor tanto adentro como afuera de la clase.

**public** **class** ConsultaVariables {

// Objetos de la Clase JSONArray & JSONObject

**private** JSONArray valoresArray = **new** JSONArray();

**private** JSONObject objetoValor = **new** JSONObject();

// Métodos Get y Set

**public** JSONArray getValoresArray() {

**return** valoresArray;

}

**public** **void** setValoresArray(JSONArray valoresArray) {

**this**.valoresArray = valoresArray;

}

**public** JSONObject getObjetoValor() {

**return** objetoValor;

}

**public** **void** setObjetoValor(JSONObject objetoValor) {

**this**.objetoValor = objetoValor;

}

A continuación se observan los métodos que se encargan de almacenar los datos que se extraen de la Base de Datos. Los cuales pueden ser invocados desde otras clases.

Método Consulta:

Éste método está encargado de crear y establecer una conexión con la Base de Datos de tal modo que se pueda realizar una consulta SQL y así obtener datos a partir de ella.

**public** **class** ConsultaVariables {

**public** **void** Consulta() {

//Se crea un Objeto de la clase BaseDato el cual es Llamado para acceder al DataSource del Pool de Conexiones.

BaseDato basedato = **new** BaseDato();

Connection conexion1 = **null**;

Statement sentencia1 = **null**;

ResultSet rs1 = **null**;

String sql = "";

**try** {

conexion1 = basedato.getConexion("ADMINISTRADOR");

sentencia1 = conexion1.createStatement();

sql = "SELECT humedad\_suelo,humedad\_aire,radiacion\_solar, temperatura,hora FROM agosto2 ORDER BY id DESC LIMIT 1;";

rs1 = sentencia1.executeQuery(sql);

**while** (rs1.next()) {

String temperatura = rs1.getString("temperatura");

setTemp(temperatura);

String hum\_suelo = rs1.getString("humedad\_suelo");

setHumedad\_suelo(hum\_suelo);

String hum\_aire = rs1.getString("humedad\_aire");

setHumedad\_aire(hum\_aire);

String rad = rs1.getString("radiacion\_solar");

setRadiacion(rad);

String hora = rs1.getString("hora");

setHora(hora);

}

} **catch** (Exception e) {

e.printStackTrace();

System.*out*.println("Error: " + e.toString());

} **finally** {

basedato.cerrarResultSet(rs1);

basedato.cerrarStatement(sentencia1);

basedato.cerrarConexion(conexion1);

}

Método VariablesGrafico:

Éste método al igual que el anterior, crea una conexión para realizar una consulta SQL. La diferencia entre estos métodos es su consulta SQL, Ya que en éste método se utiliza para crear Objetos de la clase JSONArray y JSONObject.

**public** **class** ConsultaVariables {

**public** **void** VariablesGrafico() {

// Se crea un Objeto de la clase BaseDato el cual es Llamado para

// acceder al DataSource del Pool de Conexiones.

BaseDato basedato = **new** BaseDato();

Connection conexion1 = **null**;

Statement sentencia1 = **null**;

ResultSet rs1 = **null**;

String sql = "";

**try** {

conexion1 = basedato.getConexion("ADMINISTRADOR");

sentencia1 = conexion1.createStatement();

sql = "SELECT humedad\_suelo,humedad\_aire,radiacion\_solar, temperatura,hora FROM agosto2 ORDER BY id DESC LIMIT 10;";

rs1 = sentencia1.executeQuery(sql);

**while** (rs1.next()) {

objetoValor.put("temperatura", rs1.getString("temperatura"));

objetoValor.put("humedad\_suelo", rs1.getString("humedad\_suelo"));

objetoValor.put("humedad\_aire", rs1.getString("humedad\_aire"));

objetoValor.put("radiacion\_solar",rs1.getString("radiacion\_solar"));

objetoValor.put("hora", "'"+rs1.getString("hora")+"'");

valoresArray.add(objetoValor);

}

} **catch** (Exception e) {

e.printStackTrace();

System.*out*.println("Error: " + e.toString());

} **finally** {

basedato.cerrarResultSet(rs1);

basedato.cerrarStatement(sentencia1);

basedato.cerrarConexion(conexion1);

}

}

}

La clase GetVariables que extiende de HttpServlet, permite imprimir en Web los datos que se capturan de la Base de Datos, de tal manera que la Aplicación en Android pueda acceder a los valores en formato "application/json" y así se puedan dar en un formato legible para su lectura.

El método por el cual se ejecuta la consulta es el método Post, el cual permite una mayor seguridad a la hora de establecer una consulta.

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

// **TODO** Auto-generated method stub

PrintWriter out = response.getWriter();

JSONArray respuesta;

String metodo = request.getParameter("metodo");

**if**(metodo.equals("getVariables")){

response.setContentType("application/json");

respuesta = getVariables();

out.println(respuesta);

}

}

El método que se encarga de devolver los valores para ser impresos en pantalla es el siguiente:

**public** JSONArray getVariables(){

JSONArray respuesta = **new** JSONArray();

JSONObject variable = **new** JSONObject();

BaseDato basedato = **new** BaseDato();

Connection conexion1 = **null**;

Statement sentencia1 = **null**;

ResultSet rs1 = **null**;

String sql = "";

**try**{

conexion1 = basedato.getConexion("ADMINISTRADOR");

sentencia1 = conexion1.createStatement();

sql = "SELECT TO\_CHAR(hora, 'YYYY/MM/DD HH24:MI:SS') AS hora, temperatura,radiacion\_solar, humedad\_aire,humedad\_suelo FROM

agosto2 ORDER BY id DESC LIMIT 1 ;";

rs1 = sentencia1.executeQuery(sql);

**while**(rs1.next()){

String radiacion\_solar = rs1.getString("radiacion\_solar");

String humedad\_suelo = rs1.getString("humedad\_suelo");

String humedad\_aire = rs1.getString("humedad\_aire");

String temperatura = rs1.getString("temperatura");

String hora = rs1.getString("hora");

variable.put("radiacion\_solar", radiacion\_solar);

variable.put("humedad\_suelo", humedad\_suelo);

variable.put("humedad\_aire", humedad\_aire);

variable.put("temperatura", temperatura);

variable.put("hora", hora);

respuesta.add(variable);

}

}

**catch**(Exception e){

e.printStackTrace();

System.*out*.println("Error: " + e.toString());

}

**finally**{

basedato.cerrarObjeto(rs1);

basedato.cerrarObjeto(sentencia1);

basedato.cerrarObjeto(conexion1);

}

**return** respuesta;

De esta manera concluye un repaso de todas las clases y métodos usados en la parte web para lograr establecer las conexiones y las consultas en general que se realizarán hacia la base de datos.

Ahora se explicarán las clases que se utilizaron para la creación de la Aplicación Móvil en Android.

Éstas clases extienden de Activity Y su método principal es el onCreate que es el

Encargado de crear la inicialización normal de la aplicación, crear vistas, crear Objetos, además de otras funciones importantes.

La Clase MainActivity:

**public** **class** MainActivity **extends** Activity {

// La clase SharedPreferences se encargará de almacenar Datos.

**public** **static** SharedPreferences *preferencias*;

// Definicion de los Widgets

**private** ImageButton btn\_online;

**private** ImageButton btn\_offline;

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

// Definición del contenido visual

setContentView(R.layout.*activity\_main*);

// SharedPreferences. Acá guardaremos los datos que se actualizan a partir de la Base de Datos.

*preferencias* = getSharedPreferences("Variables", Context.*MODE\_PRIVATE*);

// Creacion de las vistas o widgets

btn\_online = (ImageButton)findViewById(R.id.*btn\_online*);

btn\_offline = (ImageButton)findViewById(R.id.*btn\_offline*);

btn\_online.setOnClickListener(**new** OnClickListener() {

@Override

**public** **void** onClick(View v) {

// **TODO** Auto-generated method stub

Intent i = **new** Intent(MainActivity.**this**,ConsultaOnline.**class**);

startActivity(i);

}

});

btn\_offline.setOnClickListener(**new** OnClickListener() {

@Override

**public** **void** onClick(View v) {

// **TODO** Auto-generated method stub

Intent i = **new** Intent(MainActivity.**this**,ConsultaOffline.**class**);

startActivity(i);

}

});

Los métodos que se encuentran en ésta actividad son básicamente para controlar la finalización de la aplicación en caso de que el usuario presiona el botón “atrás”.

// Método que muestra al usuario una alerta antes de cerrar la aplicación

**private** **void** openAlert() {

AlertDialog.Builder alertDialogBuilder = **new** AlertDialog.Builder(MainActivity.**this**);

alertDialogBuilder.setTitle(**this**.getTitle()+ " Salir?");

alertDialogBuilder.setMessage("Seguro que deseas salir de la Aplicación?");

alertDialogBuilder.setPositiveButton("Si",**new** DialogInterface.OnClickListener() {

**public** **void** onClick(DialogInterface dialog,**int** id) {

finish();

}

});

alertDialogBuilder.setNegativeButton("No",**new** DialogInterface.OnClickListener() {

**public** **void** onClick(DialogInterface dialog,**int** id) {

dialog.cancel();

}

});

AlertDialog alertDialog = alertDialogBuilder.create();

// Mostrar alerta

alertDialog.show();

}

// Método que ejecúta otro método cuando se ha presionado el boton "atrás".

@Override

**public** **boolean** onKeyDown(**int** keyCode, KeyEvent event)

{

**if** ((keyCode == KeyEvent.*KEYCODE\_BACK*))

{

openAlert();

}

**return** **super**.onKeyDown(keyCode, event);

}

}

La clase ConsultaOnlie, se encarga de enviar una petición HttpGet al servidor para cargar los widgets con la información apropiada.

**public** **class** ConsultaOnline **extends** Activity {

// Definición de Objetos y Widgets.

**boolean** Succes;

ProgressDialog progressDialog = **null**;

**private** TextView txt\_temperatura;

**private** TextView txt\_radiacion;

**private** TextView txt\_humedad\_aire;

**private** TextView txt\_humedad\_suelo;

**private** TextView txt\_hora;

JSONArray jArray;

JSONObject valores;

String str = "";

// Método onCreate, se encarga de cargar las vistas y los widgets.

@Override

**protected** **void** onCreate(Bundle savedInstanceState) {

// **TODO** Auto-generated method stub

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*first\_main*);

progressDialog = **new** ProgressDialog(**this**);

progressDialog.show();

progressDialog.setContentView(R.layout.*custom\_progressdialog*);

// se ppdrá cerrar simplemente pulsando back

progressDialog.setCancelable(**true**);

txt\_temperatura = (TextView) findViewById(R.id.*text\_valor\_temperatura*);

txt\_humedad\_aire = (TextView) findViewById(R.id.*text\_valor\_humedad\_aire*);

txt\_humedad\_suelo = (TextView) findViewById(R.id.*text\_valor\_humedad\_suelo*);

txt\_radiacion = (TextView) findViewById(R.id.*text\_valor\_radiacion*);

txt\_hora = (TextView) findViewById(R.id.*text\_valor\_hora*);

// Se ejecuta una clase que extiende de AsyncTask

**new** DoTheTask().execute("");

}

El método Traer() De tipo Boolean, está encargado de enviar una petición HttpGet al servidor con el fin de realizar una consulta a través de una Url que especifica un Servlet. En éste se implementa el método de tipo Boolean para validar en caso de que haya ocurrido un error en el servidor o el tiempo de espera a la petición se haya agotado.

//Este método está encargado de darle Formato a la respuesta y cargar los widgets con su respectiva información.

runOnUiThread(**new** Runnable() {

@Override

**public** **void** run() {

**try** {

jArray = **new** JSONArray(str.toString());

valores = jArray.getJSONObject(0);

String temperatura = valores.getString("temperatura");

String radiacion = valores.getString("radiacion\_solar");

String humedad\_aire = valores.getString("humedad\_aire");

String humedad\_suelo = valores.getString("humedad\_suelo");

String hora = valores.getString("hora");

/\*Editamos las preferencias en las cuales vamos a almacenar la consulta que se realiza.

\* Con ésto podriamos conocer los resultados de la ultima consulta realizada sin necesidad de tener una conexión a internet \*/

SharedPreferences.Editor editor = MainActivity.*preferencias*

.edit();

editor.putString("temperatura", temperatura);

editor.putString("radiacion\_solar", radiacion);

editor.putString("humedad\_aire", humedad\_aire);

editor.putString("humedad\_suelo", humedad\_suelo);

editor.putString("hora", hora);

editor.commit();

txt\_temperatura.setText(temperatura + "°");

txt\_radiacion.setText(radiacion + " W/m²");

txt\_humedad\_aire.setText(humedad\_aire + "%");

txt\_humedad\_suelo.setText(humedad\_suelo + "%");

txt\_hora.setText(hora);

Succes = **true**;

} **catch** (JSONException e) {

Succes = **false**;

e.printStackTrace();

}

}

});

} **catch** (Exception e) {

Succes = **false**;

e.printStackTrace();

// **TODO**: handle exception

System.*out*.println("Error");

}

**return** Succes;

}

La clase ConsultaOnline tiene una clase Interna que extiende de AsyncTask, La cual, mediante unos métodos especiales de la clase, le permite realizar tareas asíncronas y ser controladas desde sus respectivos métodos.

**private** **class** DoTheTask **extends** AsyncTask<String, Void, Boolean> {

**protected** Boolean doInBackground(String... params) {

**return** Traer();

}

**protected** **void** onPostExecute(Boolean result) {

progressDialog.dismiss();

**if** (result == **true**) {

System.*out*.println("Post" + result);

SuccessFul();

} **else** {

System.*out*.println("Post" + result);

Warning();

}

System.*out*.println("Post" + result);

}

}

**public** **void** Warning() {

SweetAlertDialog dialogo= **new** SeetAlertDialog(**this**,SweetAlertDialog.*WARNING\_TYPE*);

dialogo.setTitleText("Ops! Ocurrió un Error")

.setContentText("Parece que tienes problemas de conexión.\n Intentalo mas tarde!")

.setConfirmText("Ok!")

.setConfirmClickListener(**new** SweetAlertDialog.OnSweetClickListener() {

@Override

**public** **void** onClick(SweetAlertDialog sDialog) {

sDialog.dismiss();

Intent i = **new** Intent(ConsultaOnline.**this**,MainActivity.**class**);

startActivity(i);

}

}).setOnDismissListener(**new** OnDismissListener() {

@Override

**public** **void** onDismiss(DialogInterface dialog) {

// **TODO** Auto-generated method stub

Intent i = **new** Intent(ConsultaOnline.**this**,MainActivity.**class**);

startActivity(i);

}

});

dialogo.show();

}

**public** **void** SuccessFul() {

**new** SweetAlertDialog(**this**, SweetAlertDialog.*SUCCESS\_TYPE*)

.setTitleText("Éxito")

.setContentText("Ya puedes consultar el último registro Sin Conección!")

.show();

}

}

Se tiene ahora la clase ConsultOffline que nos permite cargar los datos que se cargaron en la última consulta. Básicamente ésta clase solo hace uso del método onCreate() que está encargado de cargar las vistas y los widgets con sus respectivos datos.

**public** **class** ConsultaOffline **extends** Activity{

// Definición de los Widgets

**private** TextView txt\_temperatura;

**private** TextView txt\_radiacion;

**private** TextView txt\_humedad\_aire;

**private** TextView txt\_humedad\_suelo;

**private** TextView txt\_hora;

@Override

**protected** **void** onCreate(Bundle savedInstanceState) {

// **TODO** Auto-generated method stub

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*second\_main*);

//Se crea una alerta avisando al usuario que está viendo los ultimos datos consultados en la clase ConsultaOnlie

**new** SweetAlertDialog(**this**, SweetAlertDialog.*SUCCESS\_TYPE*)

.setTitleText("Offline")

.setContentText("Estás viendo el Ultimo registro sin conexión.")

.setConfirmText("Ok").show();

txt\_temperatura = TextView)findViewById(R.id.*text\_valor\_temperatura\_off*);

txt\_humedad\_aire=TextView)findViewById(R.id.*text\_valor\_humedad\_aire\_off*);

txt\_humedad\_suelo=(TextView)findViewById(R.id.*text\_valor\_humedad\_suelo\_off*);

txt\_radiacion = (TextView)findViewById(R.id.*text\_valor\_radiacion\_off*);

txt\_hora = (TextView)findViewById(R.id.*text\_valor\_hora\_off*);

//En ésta parte invocamos el objeto de la clase SharedPreferenes el cual fué utilizado para almacenar la información.

//De ésta manera podemos sacar la información y cargarlas en los Widgets (TextView)

String temperatura = MainActivity.*preferencias*.getString("temperatura","");

String radiacion\_solar=MainActivity.*preferencias*.getString("radiacion\_solar","");

String humedad\_aire = MainActivity.*preferencias*.getString("humedad\_aire","");

String humedad\_suelo = MainActivity.*preferencias*.getString("humedad\_suelo","");

String hora = MainActivity.*preferencias*.getString("hora","");

txt\_temperatura.setText(temperatura+"°");

txt\_radiacion.setText(radiacion\_solar+" W/m²");

txt\_humedad\_aire.setText(humedad\_aire+"%");

txt\_humedad\_suelo.setText(humedad\_suelo+"%");

txt\_hora.setText(hora);

### Seguridad y Autenticación

JAAS (Java Authentication and Authorization Service) es una interfaz que permite a las aplicaciones Java acceder a servicios de control de autenticación y acceso.

Puede usarse con dos fines:

* la autenticación de usuarios: para conocer quién está ejecutando nuestro código java,
* la autorización de usuarios: para garantizar que quién lo ejecuta tiene los permisos necesarios para hacerlo.

Para ello, fue necesario crear un DataSource, que se implementa en el Pool de Conexiones y nos permite comunicar la aplicación web con la base de datos.

DataSource:

<datasource jndi-name=*"java:jboss/gradoDS"* pool-name=*"gradoDS\_Pool"* enabled=*"true"* jta=*"true"* use-java-context=*"true"* use-ccm=*"true"*>

<connection-url>

jdbc:postgresql://localhost:5432/invernausco

</connection-url>

<driver>

postgresql-9.0-801.jdbc4

</driver>

<security>

<user-name>

invernausco

</user-name>

<password>

1234

</password>

</security>

</datasource>

Después de haber creado el DataSource, se crea un Módulo de Login, que está encargado de comunicar a la base de datos mediante una consulta con los datos del Login para así poder identificar a los Usuarios que deseen ingresar en la aplicación.   
  
En éste Módulo de Login se aplica el cifrado de claves, para conseguir una mayor seguridad a la hora de acceder a la base de datos y garantizar seguridad a los usuarios.

<security-domain name=*"jBossJaasMysqlRealm"*>

<authentication>

<login-module code=*"Database"* flag=*"required"*>

<module-option name=*"dsJndiName"* value=*"java:jboss/gradoDS"*/>

<module-option name=*"principalsQuery"* value=*"select usuario.usu\_clave from usuario where usu\_nombre=?"*/>

<module-option name=*"rolesQuery"* value=*"select perfil.per\_id,'Roles' from usuario, usuario\_perfil, perfil where usuario.usu\_nombre=? and usuario.usu\_id=usuario\_perfil.usp\_usuario\_id and usuario\_perfil.usp\_perfil\_id=perfil.per\_id"*/>

<module-option name=*"hashAlgorithm"* value=*"SHA1"*/>

<module-option name=*"hashEncoding"* value=*"hex"*/>

</login-module>

</authentication> </security-domai>

Para el cifrado de claves se usó un script en PostgreSQL el cual se usó para la encriptación de la clave del administrador.

El script utilizado es el siguiente: CREATE EXTENSION pgcrypto;

Después de creado toda la funcionalidad del JAAS, en el proyecto web se debe crear un archivo XML que invoca el respectivo módulo de Login.

Jboss-web.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<jboss-web>

<security-domain>java:/jaas/jBossJaasMysqlRealm</security-domain>

</jboss-web>

Luego se declaran atributos en el archivo web.xml del proyecto para definir que usuarios tienen acceso a ciertos archivos.

<security-constraint>

<display-name>Seguridad JAAS</display-name>

<web-resource-collection>

<web-resource-name>Protected Area</web-resource-name>

<url-pattern>/admin/\*</url-pattern>

<http-method>DELETE</http-method>

<http-method>GET</http-method>

<http-method>POST</http-method>

<http-method>PUT</http-method>

</web-resource-collection>

<auth-constraint>

<role-name>1</role-name>

</auth-constraint>

<user-data-constraint>

<transport-guarantee>NONE</transport-guarantee>

</user-data-constraint>

</security-constraint>

<login-config>

<auth-method>FORM</auth-method>

<realm-name>jBossJaasMysqlRealm</realm-name>

<form-login-config>

<form-login-page>/login.jsp</form-login-page>

<form-error-page>/error.jsp</form-error-page>

</form-login-config>

</login-config>

<security-role>

De ésta manera queda implementado el servicio de JAAS y de seguridad en el proyecto web.

## PRUEBAS

Esta es la etapa donde se probará la funcionalidad del software creado, se obtienen excelentes resultado en cuanto a lo realizado en programación tanto para web como para móvil.

Se destaca a continuación un añadido que se quiso implementar como parte de una serie de buenas prácticas que todo programador y diseñador debe tener a la hora de realizar un software para algún motivo.

Se decide implementar en el diseño web algo que en el marco teórico se ha definido previamente y es el Responsive Design.

### Responsive Design

El Responsive Design es básicamente que el diseño de la página se adapte a cualquier resolución a la que sea sometido. Esto con el fin de garantizar a los usuarios una excelente visualización de la página, sin importar el dispositivo que utilicen para acceder a esta.

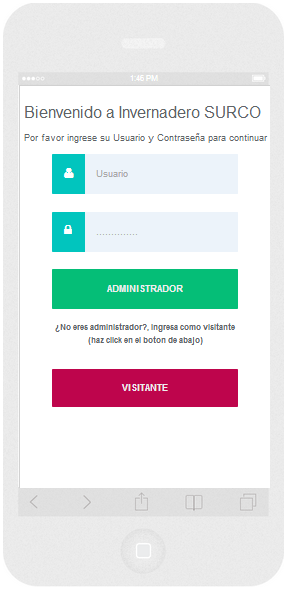
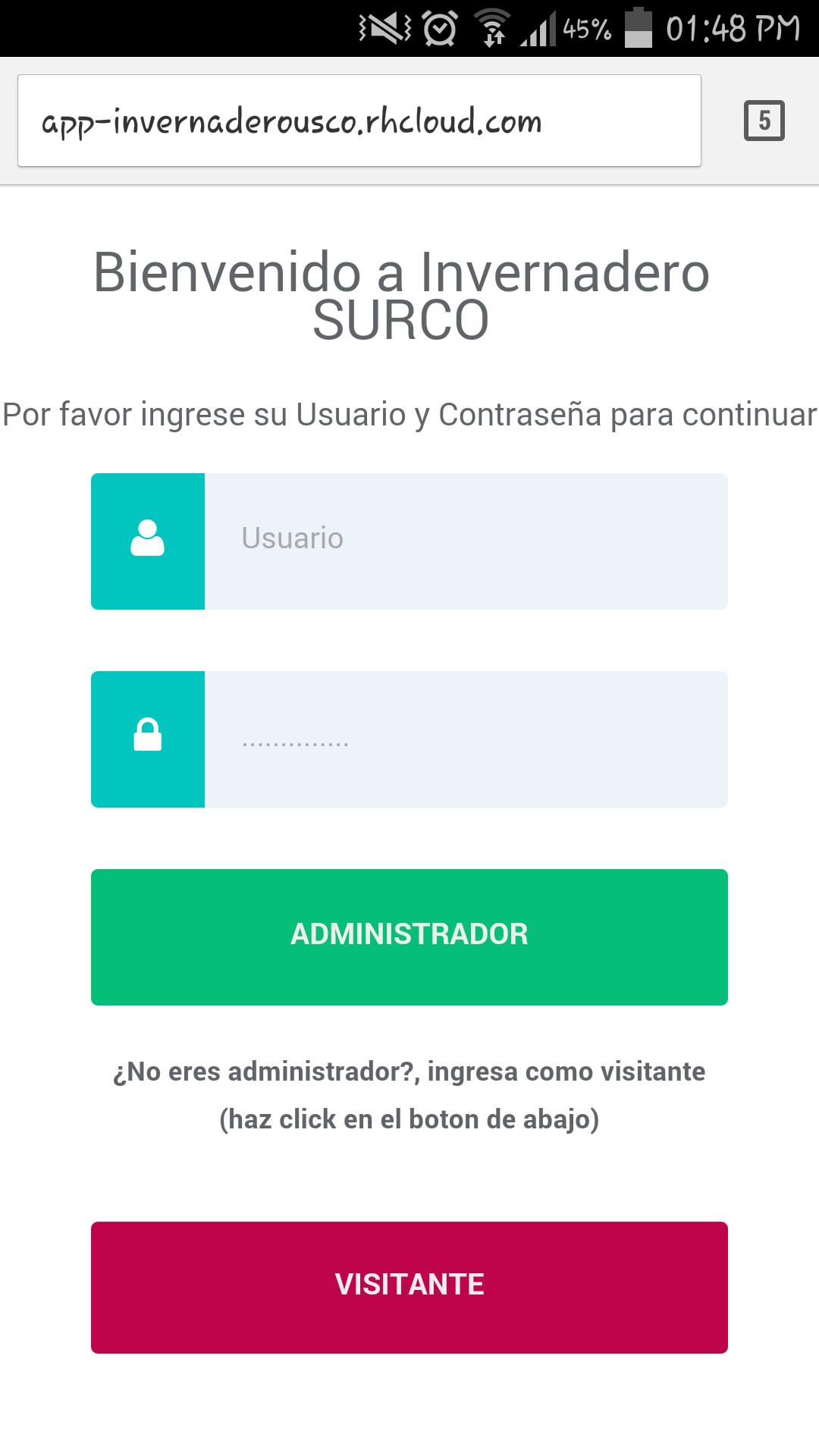


Ilustración 33 Vista Responsive Desgin en Iphone 6

Ilustración 32 Vista Responsive Design en Samsung Galaxy S4



Ilustración 34 Vista Responsive Design en Htc



Ilustración 35 Vista Responsive Design en Tablet

## INTEGRACIÓN

Luego de pasar por todas las fases previas de diseño, análisis y pruebas se llega al producto final, la aplicación móvil y su respectiva página web en funcionamiento.

Se cuenta con dos maneras para acceder a esta

[Página Web Oficial](http://app-invernaderousco.rhcloud.com/)

Pueden ingresar a la página web para efectos de prueba con:

Usuario: Administrador

Contraseña: proyectodegrado

y probar tanto la versión para administrador, como la de invitado.

También si cuenta con dispositivo Android puede acceder a la aplicación descargándola de Google play como *Invernadero SURCO.*

Se mostrará como evidencias, capturas de pantalla de la aplicación ya configurada en el servidor y de sus diferentes vistas en su etapa final tanto en la parte web como en su versión para móviles.

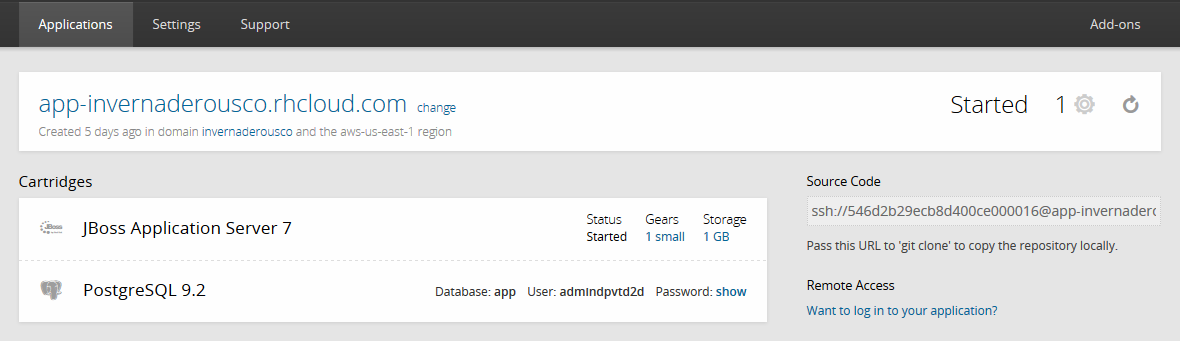


Ilustración 36 Servidor en Funcionamiento.

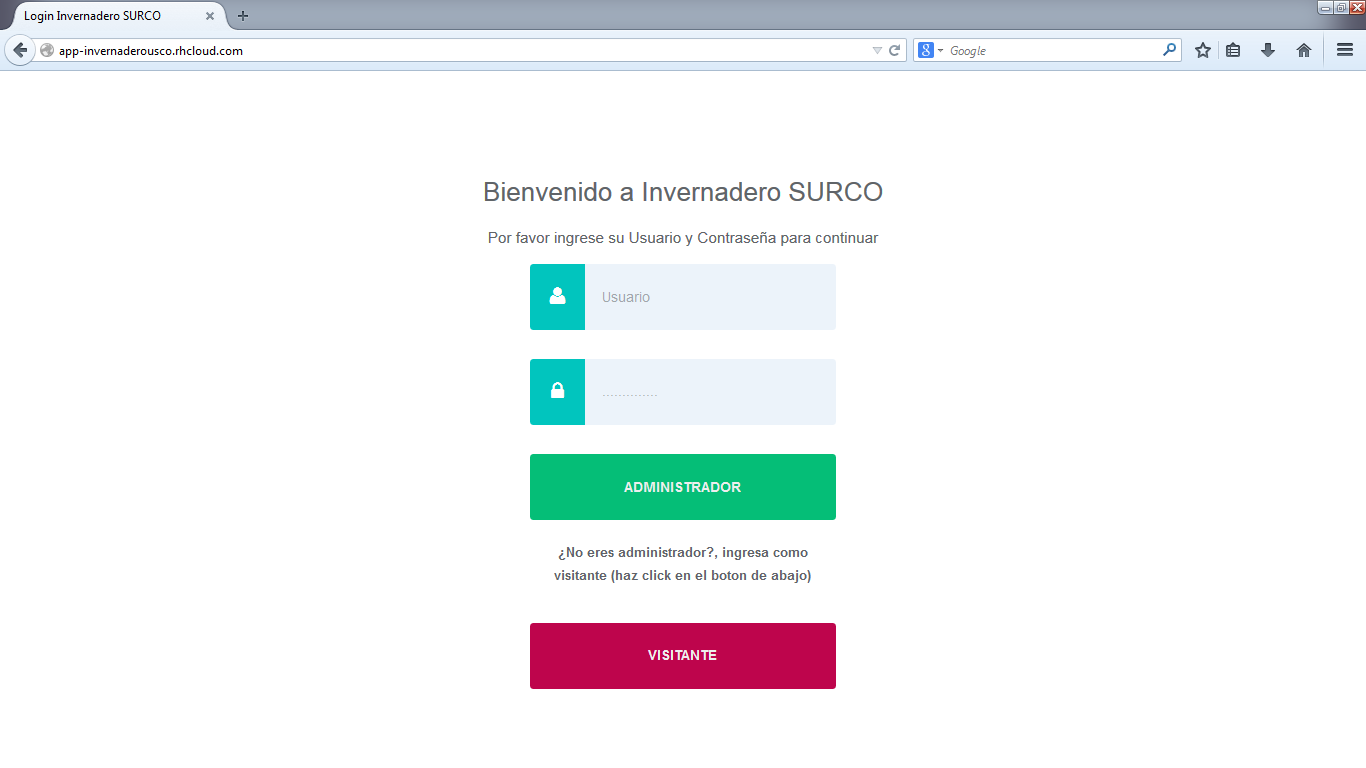


Ilustración 37 Login en funcionamiento



Ilustración 38Implementación de SweetAlert al Digitar Usuario o Contraseña Errónea



Ilustración 39 Página de Inicio para el Administrador con Información traída desde la Base de Datos

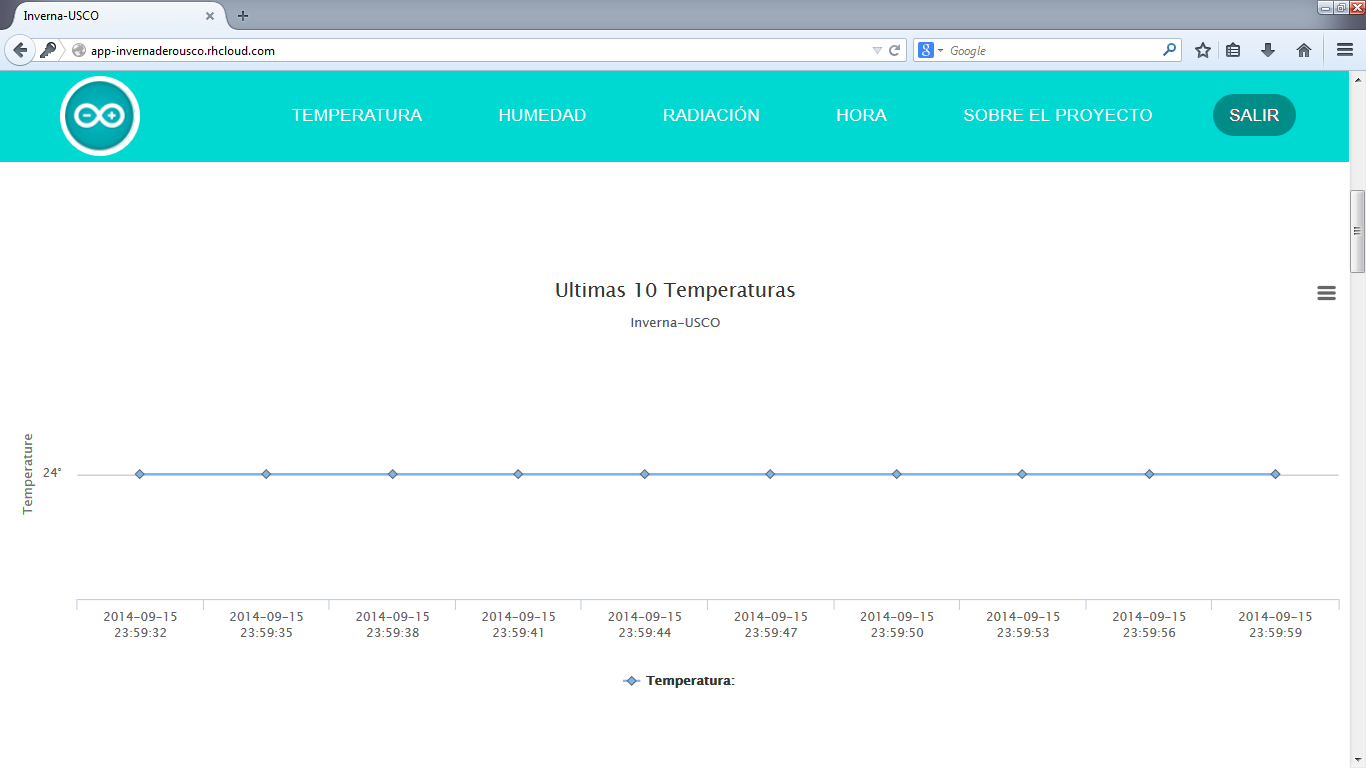


Ilustración 40 Graficas para Administrador con Información traída desde la Base de Datos



Ilustración 41Página de Inicio para el Invitado con Información traída desde la Base de Datos



Ilustración 42 Aplicación Móvil en Funcionamiento



Ilustración 43 SweetAlert Implementado en la Aplicación Android

# CRONOGRAMA DE ACTIVIDADES

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ETAPAS | ACTIVIDADES | MESES | | | | |
| AGOSTO | SEPTIEMBRE | OCTUBRE | NOVIEMBRE | DICIEMBRE |
| 1 | Elaboración del anteproyecto | X | X |  |  |  |
| 2 | Elaboración de la metodología |  | X |  |  |  |
| 3 | Diseño de la aplicación |  | X | X |  |  |
| 4 | Pruebas de control |  | X | X | X |  |
| 5 | Elaboración de los diferentes modelos de diseño del software |  |  | X | X |  |
| 6 | Publicación de la aplicación móvil |  |  | X | X |  |
| 7 | Publicación de la página web |  |  |  | X |  |
| 8 | Evaluación y validación del proyecto |  |  |  | X | X |
| 9 | Presentación final de la aplicación |  |  |  |  | X |

Tabla 11 Cronograma de Actividades

# RECURSOS DISPONIBLES

## HUMANOS

El proyecto será realizado por los estudiantes FABIAN STEVENS VARON VALENCIA y JOAN SEBASTIAN ROA SANCHEZ, del programa de Tecnología en Desarrollo de Software de la Universidad Surcolombiana, se sugiere como director al INGENIERO ESAU SILVA, jefe del programa de Tecnología en Desarrollo de Software de la Universidad Surcolombiana. Se contara con el apoyo del estudiante de ingeniería electrónica JUAN MANUEL GONZALES ALVEREZ que junto a su compañero de tesis realizaron el montaje del invernadero ubicado en SURCO (granja de la Universidad Surcolombiana). Además la ayuda y el apoyo del semillero de REUS de la Universidad Surcolombiana.

## MATERIALES

Como materiales se necesitará en primera medida el invernadero que estará ubicado en la granja de la Universidad Surcolombiana, Arduino UNO como plataforma Hardware de desarrollo, Computador portátil para trabajar la parte de código, un modem de internet para dar solución a un problema que es la falta de internet en las instalaciones de la granja, un servidor para subir tanto la base de datos como la página web, dispositivos móviles con sistema operativo Android OS, para realizar las pruebas de la aplicación móvil, herramientas informáticas como los son eclipse para la parte de desarrollo, JBoss como servidor de aplicaciones, MySQL y PostgreSQL como motores de bases de datos, Adobe Ilustrator como Herramienta de diseño para la interfaz web y móvil.

## INSTITUCIONALES

Como material institucional, se hará uso del espacio otorgado por la Universidad Surcolombiana para la construcción y adecuación de la granja, adicional a eso varios de los sensores de precisión que sirven para aumentar la veracidad de los datos obtenidos pero que serán solo de prueba para la obtención de los datos ya que este proyecto de granja ya está desarrollado e implementado por estudiantes de Ingeniería electrónica de la Universidad Surcolombiana.

## FINANCIEROS

Se trabajarán con recursos propios para todos los posibles gastos los cuales serán mencionados al final del trabajo en una tabla de costos, no se necesitará pagar por el software ya que JAVA – Eclipse trabaja con código abierto al igual que HTML5, CSS3, JavaScript y Openshift que son plataformas de código abierto, lenguajes de maquetación y lenguajes de programación de libre uso y acceso.

# CONCLUSIONES

# RECOMENDACIONES

# BIBLIOGRAFÍA

Android. (11 de 09 de 2014). *Android.* Obtenido de Android: http://www.android.com/

Arduino. (11 de 09 de 2014). *Arduino cc.* Obtenido de Arduino cc: http://arduino.cc/

Belatrixsf. (5 de 12 de 2008). *Belatrixsf*. Obtenido de Belatrixsf: http://www.belatrixsf.com/downloads/Belatrix\_PlataformasMoviles\_SP.pdf

CampusParty. (11 de 22 de 2014). *CampusParty*. Obtenido de CampusParty: www.campus-party.org/

cascada, D. e. (24 de 11 de 2014 ). *ele-mariamoliner*. Obtenido de ele-mariamoliner: http://ele-mariamoliner.dyndns.org/~fperal/proy/ingenieriaSW.pdf

Castro, M. (22 de 11 de 2014). *ocw.Innova.* Obtenido de ocw.Innova: http://ocw.innova.uned.es/mm2/tm/contenidos/pdf/tema2/lenguajes\_marcado.pdf

CELAYA, I. t. (22 de 11 de 2014). *iqcelaya.* Obtenido de iqcelaya: http://www.iqcelaya.itc.mx/~vicente/Programacion/Lenguajes.pdf

Chávez, M. A. (23 de 11 de 2014). *Universidad Autónoma del Estado de Morelos* . Obtenido de Universidad Autónoma del Estado de Morelos : http://www.gridmorelos.uaem.mx/~mcruz//cursos/miic/bd1.pdf

Cisco. (04 de 2011.). *Cisco.* Obtenido de Cisco: http://www.cisco.com/web/LA/soluciones/executive/assets/pdf/internet-of-things-iot-ibsg.pdf

Computing, C. (2010). *Cloud Computing.* Bankinter.

Css3. (08 de 11 de 2014). *w3c*. Obtenido de http://www.w3.org/TR/2001/WD-css3-roadmap-20010523/

FarmLogic. (11 de 22 de 2014). *FarmLogic*. Obtenido de FarmLogic: http://www.farmlogic.com/

Ginestá, M. G. (23 de 11 de 2014). *Open courseWare*. Obtenido de Open courseWare: http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06\_M2109\_02152.pdf

Github. (23 de 11 de 2014). *github*. Obtenido de github: https://training.github.com/kit/downloads/es/github-git-cheat-sheet.pdf

Grau, X. F. (23 de 11 de 2014). *Desarrollo Orientado a* . Obtenido de uv.mx: http://www.uv.mx/personal/maymendez/files/2011/05/umlTotal.pdf

Group, S. P. (23 de 11 de 2014). *Statistical Physics Group*. Obtenido de Statistical Physics Group: http://ergodic.ugr.es/cphys/LECCIONES/linux/00.introduccion\_a\_linux.pdf

ibm. (26 de 11 de 2012). *IBM.* Obtenido de IBM: http://www.ibm.com/developerworks/ssa/library/os-ecov/

Industrial., E. T. (16 de 10 de 2014). *ICRA*. Obtenido de http://icra.blogspot.es/categoria/proyecto-icra/

Investigacion, T. d. (19 de 11 de 2014). *Tipos de investigacion*. Obtenido de http://www.tiposdeinvestigacion.com/

JAVA, E. l. (22 de 11 de 2014). *uned.* Obtenido de uned: http://wainu.ii.uned.es:8081/WAINU/ing\_informatica/cuarto/SiI1/apuntes/Libro\_Java-SP.pdf

lab, C. (11 de 22 de 2014). *Campus Lab*. Obtenido de Campus Lab: http://www.campus-labs.com/es/index.php

Laister, J. (30 de 10 de 2014). *Ubuntunizando.* Obtenido de Ubuntunizando: http://www.ubuntizando.com/2014/10/30/html5-ya-es-un-estandar-recomendado-por-w3c/

Marquez, L. m. (2012). *Invernadero Controlado por móvil Android.* Zaragoza, España: https://docs.google.com/folderview?id=0B-PoupdpoRSPdHJxbWgxX3FFOU0.

Martínez, I. B. (23 de 11 de 2014). *Universidad de Murcia*. Obtenido de Universidad de Murcia: http://www.um.es/linux/xml/ponencia/XML\_paratodos.pdf

MasterMagazine. (23 de 11 de 2014). *MasterMagazine.* Obtenido de MasterMagazine: http://www.mastermagazine.info/termino/5286.php

Mozilla. (03 de 07 de 2014). *developers Mozilla.* Obtenido de developers Mozilla: https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Obsolete\_Pages/Gu%C3%ADa\_JavaScript\_1.5/Concepto\_de\_JavaScript

*Observatoio Software Libre*. (19 de 11 de 2014). Obtenido de http://observatorio.cds.gov.co/index.php?option=com\_content&view=article&id=28%3Aanalisis-marco-legal-software-libre&catid=34%3Aanalisis-marco-legal-software-libre&Itemid=41

Pastor, A. E. (2013). *carriots.* Obtenido de Carriots.com: https://www.carriots.com/newFrontend/img-carriots/press\_room/Construyendo\_un\_proyecto\_de\_IOT.pdf

Playground, A. (23 de 11 de 2014). *Arduino*. Obtenido de Arduino: http://playground.arduino.cc/uploads/Main/arduino\_comic\_es.pdf

RedHat. (2 de 11 de 2014). *RedHat*. Obtenido de RedHat: http://www.redhat.com/es/technologies/cloud-computing/openshift

Sanz, D. (s.f.). *Introducción a Android.* Universidad Computense de Madrid: E.M.E. Editorial ©.

Saroka, R. H. (23 de 11 de 2014). *Universidad de Buenos Aires*. Obtenido de Universidad de Buenos Aires: http://www.econ.uba.ar/www/departamentos/sistemas/plan97/tecn\_informac/rota/Zimerman/Saroka.pdf

Silberschatz, A. (2011). *Fundamentos de Bases de Datos.* España: Mc Graw Hill.

Tamayo, J. M. (14 de 05 de 2014). *unaplantaonline*. Obtenido de unaplantaonline: http://unaplantaonline.iqbusiness.mx/

tesis, C. e. (2006). *Biblioteca UCV.* Obtenido de http://biblioteca.ucv.cl/site/servicios/documentos/como\_escribir\_tesis.pdf

UML, D. d. (23 de 11 de 2014). *club del suran*. Obtenido de club del suran: http://www.clubdelsuran.com.ar/site/materiales/proyecto/diagramas\_del\_uml.pdf

Universidad Del Rosario Smartphone, A. e. (03 de 2012). *Universidad del Rosario.* Obtenido de Universidad del Rosario: http://www.urosario.edu.co/urosario\_files/91/91f9aa5e-9f6f-480a-96e4-c554f9c6ab3d.pdf

UNO, A. (08 de 11 de 2014). *Arduino cc*. Obtenido de Arduino: http://arduino.cc/en/Main/arduinoBoardUno

Vega, A. A. (23 de 11 de 2014). *UNIVERSIDAD DE ALCALÁ*. Obtenido de UNIVERSIDAD DE ALCALÁ: http://dspace.uah.es/dspace/bitstream/handle/10017/19972/Memoria.pdf?sequence=1

Versiones, C. d. (13 de 04 de 2006). *Torroja dmt*. Obtenido de Guillem Borrell: http://torroja.dmt.upm.es/media/files/cversiones.pdf

W3C. (28 de 10 de 2014). *Html5 recommendation.* Obtenido de Html5 Recommendation: http://www.w3.org/blog/news/archives/4167

# ANEXOS

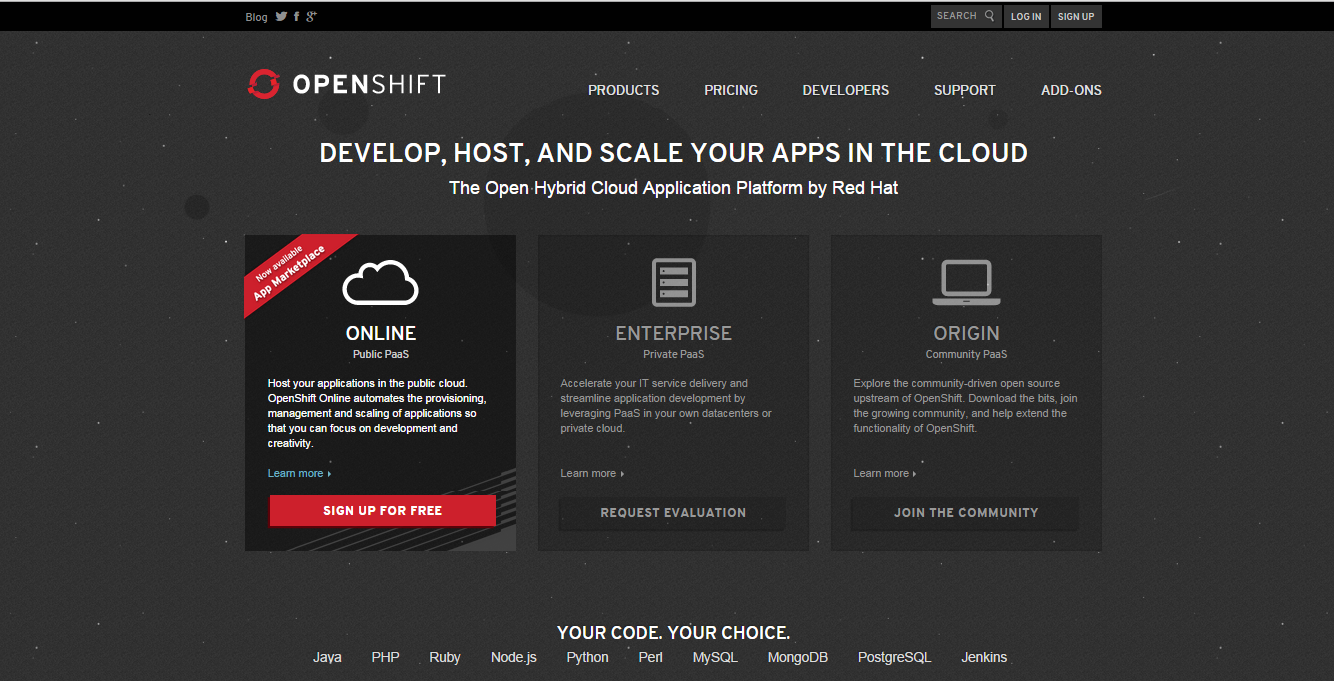
ANEXO A

Creación e implementación del proyecto web local en Openshift (Nube)

El primer paso para crear una aplicación en Openshift será crear una cuenta gratis. Las cuentas gratis en Openshift permiten crear un máximo de 3 aplicaciones con 1 respectivo cartucho, en nuestro caso el motor de Base de Datos PostgreSQL que se explicará mas adelante.

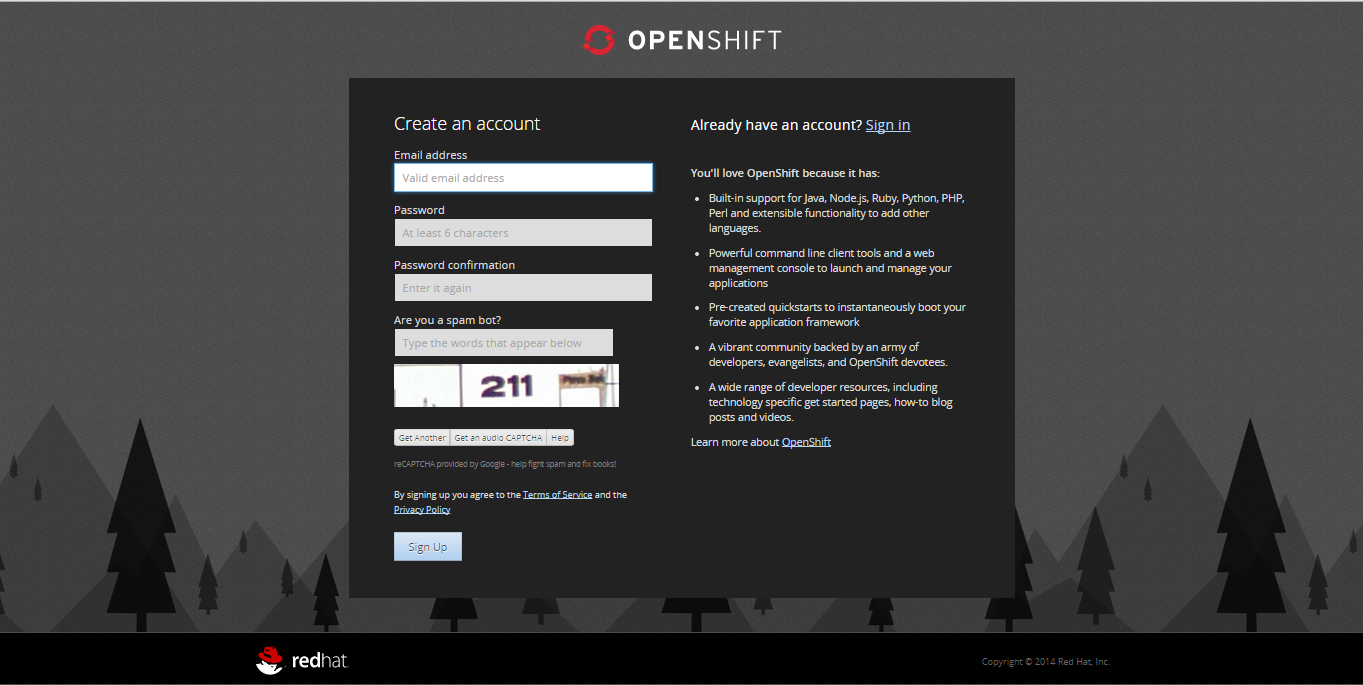
Ingresar a la página web de Openshift

<https://www.openshift.com/>

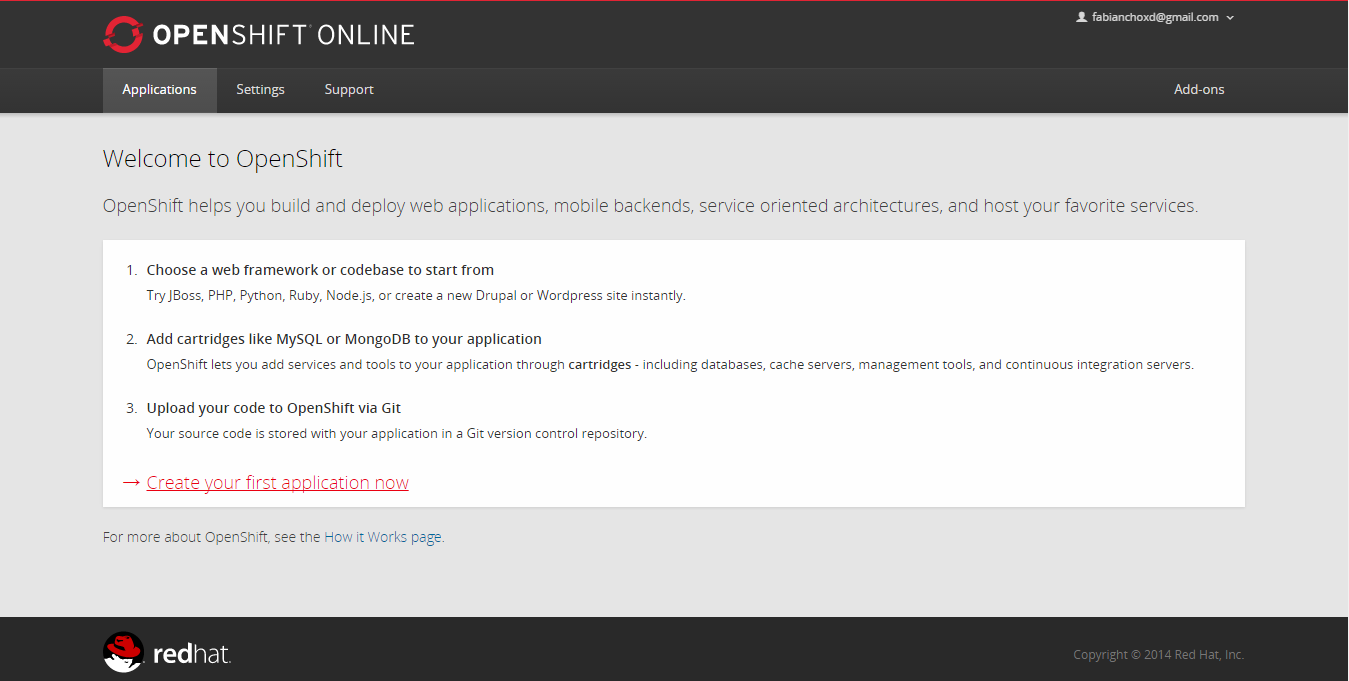


En la siguiente imagen, seleccionar “SIGN UP FOR FREE” para empezar el registro.

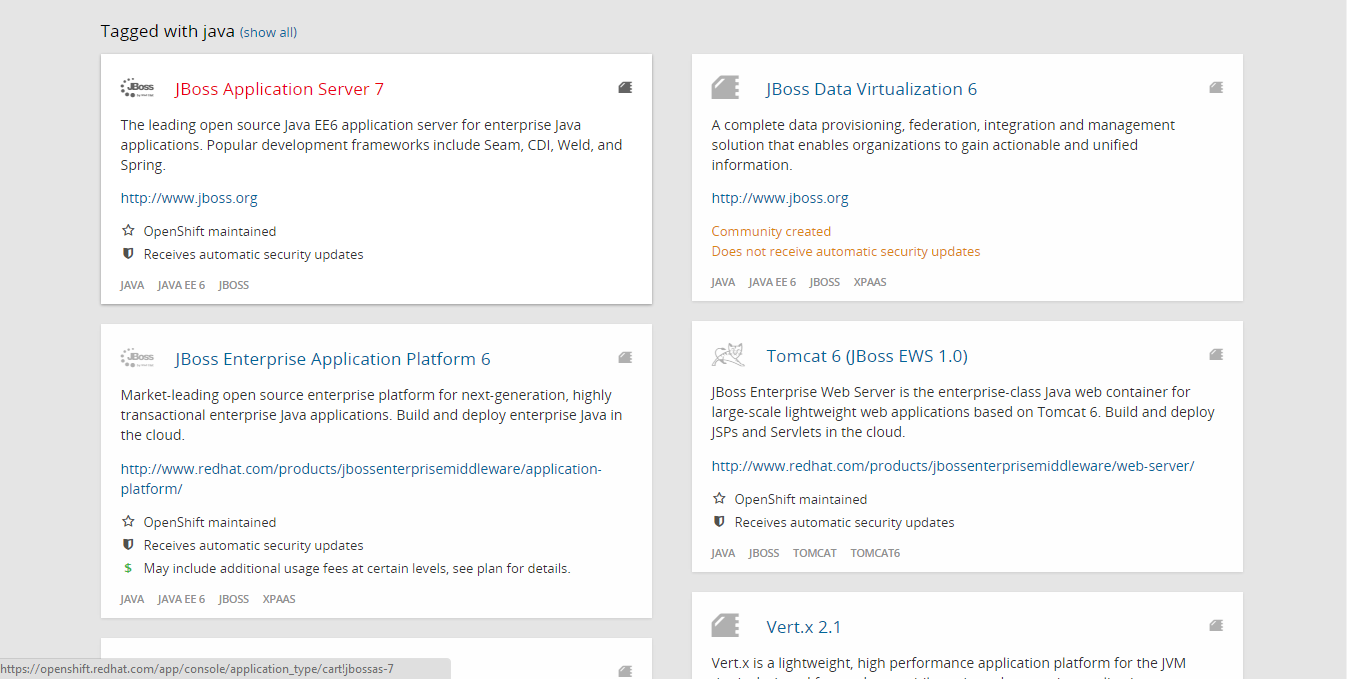
Luego se completa el formulario con los respectivos datos.



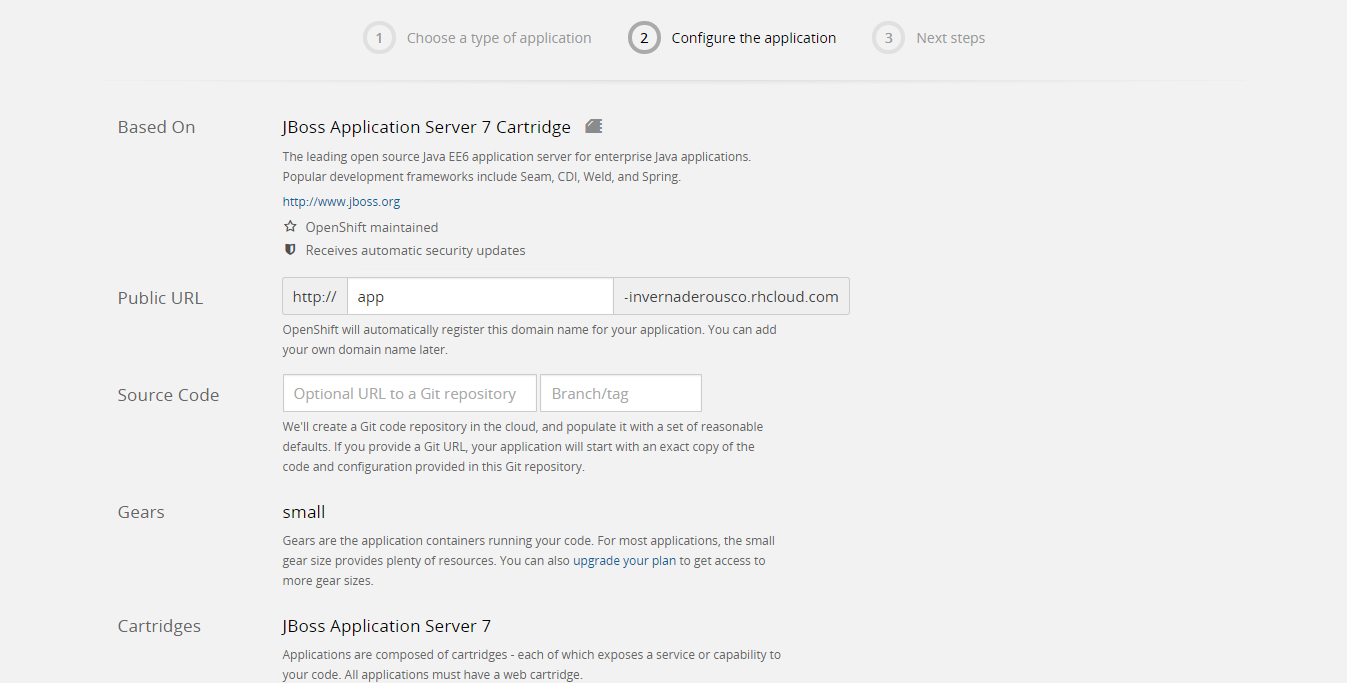
Después del registro, se verifica con un link que es enviado al correo electrónico.

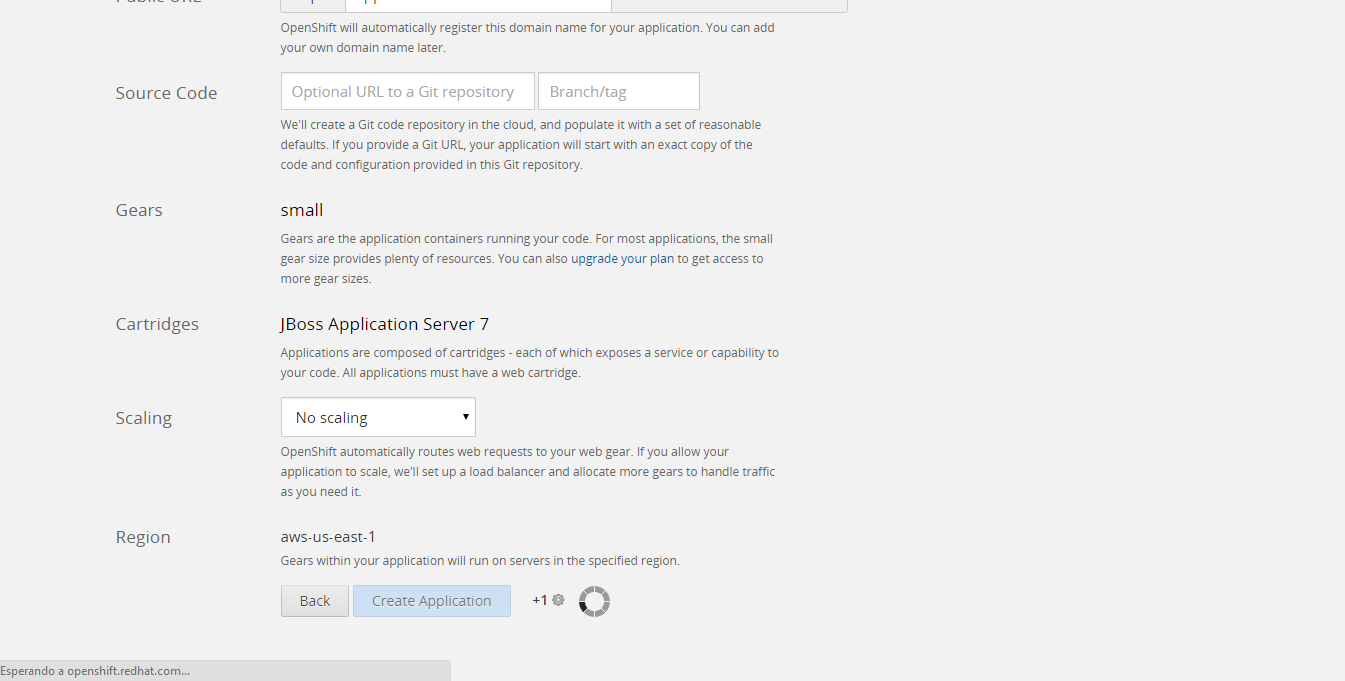
Después de verificada la cuenta, se procede a ver la consola de aplicaciones, la cual se encuentra vacía y seguido de esto a crear la primera aplicación

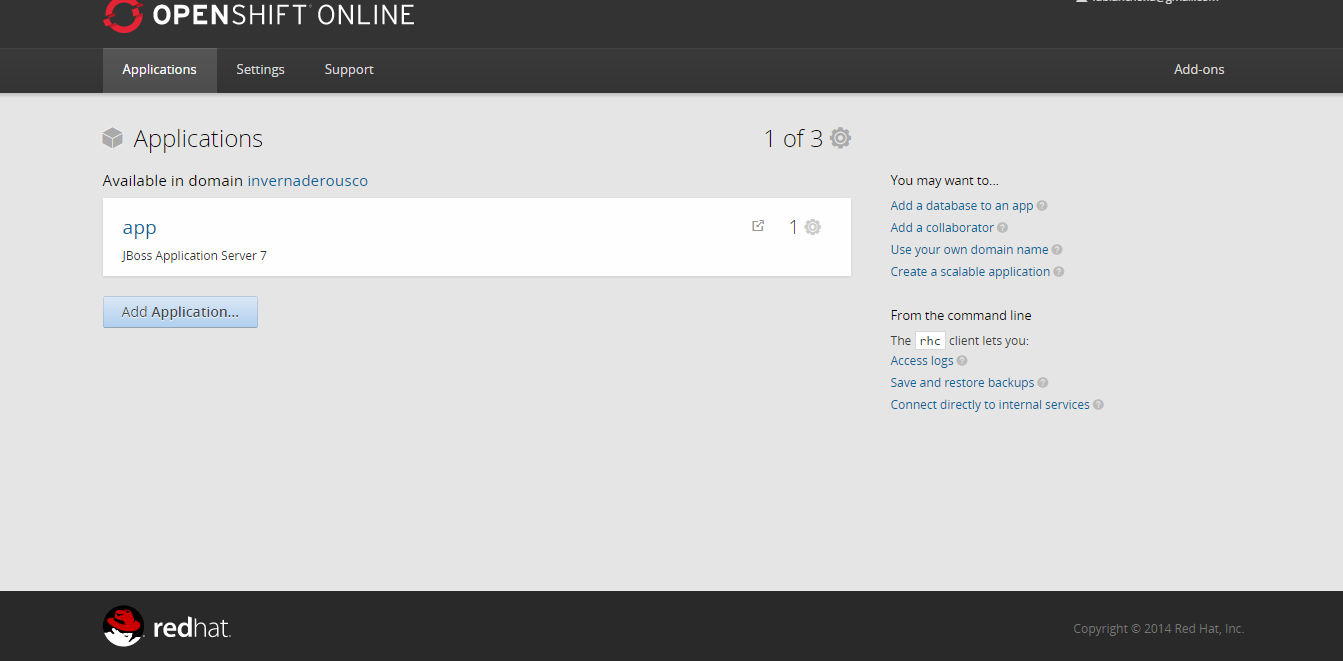
En el proceso de creación de la aplicación, se ofrecen varios servidores en los cuales se puede trabajar. Para este caso se escoge el servidor JBoss Application Server 7.



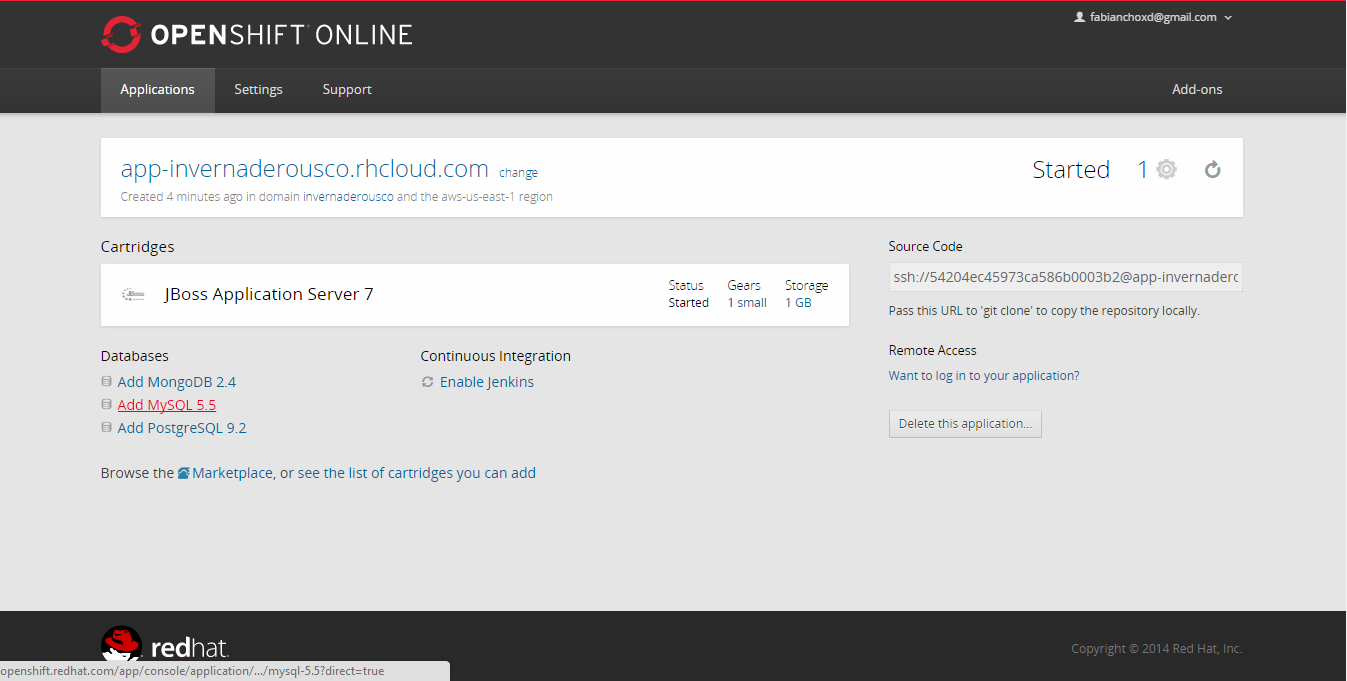
En ésta sección se asigna el nombre de espacio de trabajo, de la aplicación y opcionalmente se podrá asignar una url donde tengamos algún repositorio para almacenar el proyecto allí.



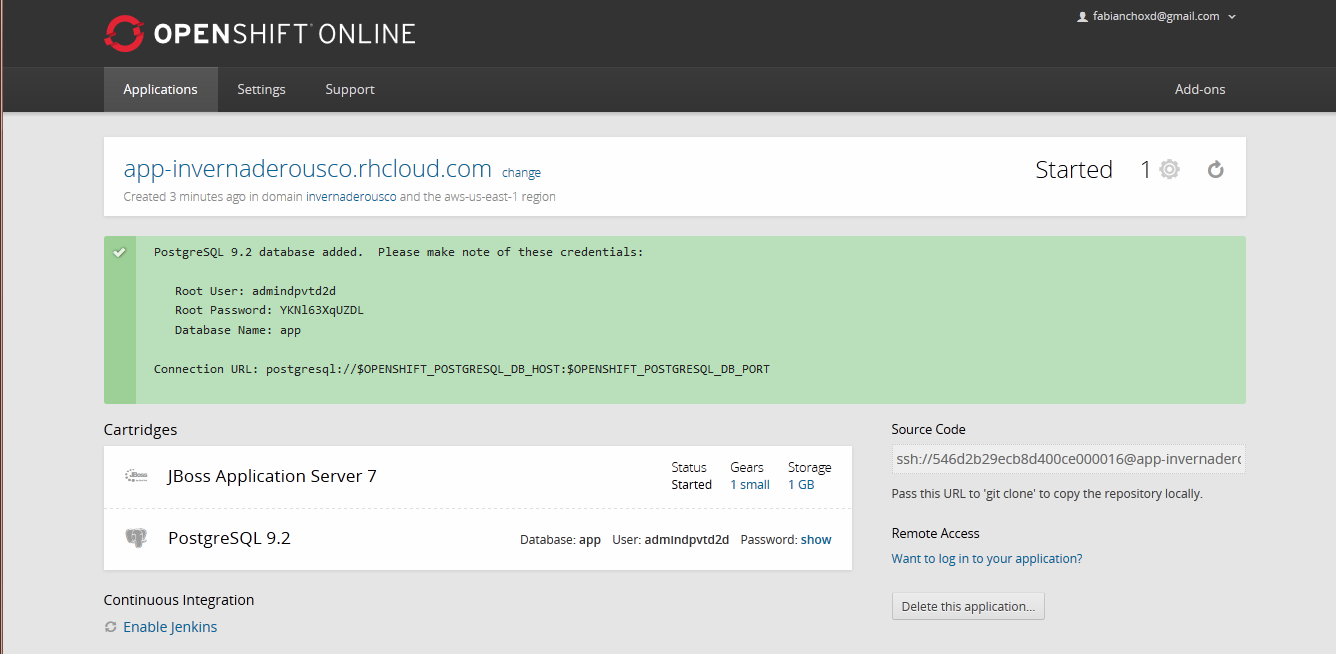


Como se puede observar, la aplicación ha sido creada. Ahora se puede acceder a ella para ver más detalles.

Como se obverva en la siguiente imagen, se encuentra la Url de la aplicación, el nombre del servidor, la url para descargar la aplicación y los Cartuchos que se pueden agregar. Para este caso se usará el cartucho de PostgreSQL.



Después de haber seleccionado el Motor de Base de Datos. Se visualizaran los detalles de la base de datos asignada a la aplicación.



Con todo esto ya se puede proceder a subir el proyecto web, una vez realizado el trabajo por completo o inclusive puede ser subido por versiones y así implementar pruebas.

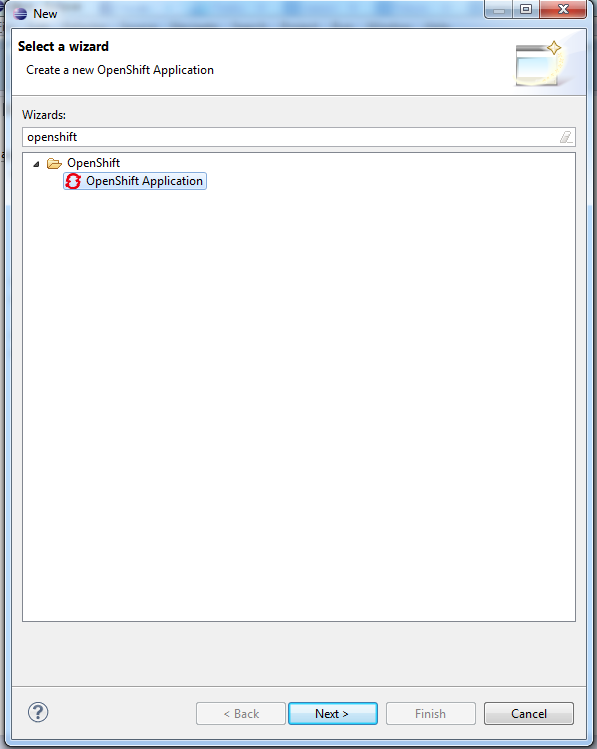
ANEXO B

Proceso de descarga y modificación del proyecto en Openshift (nube).

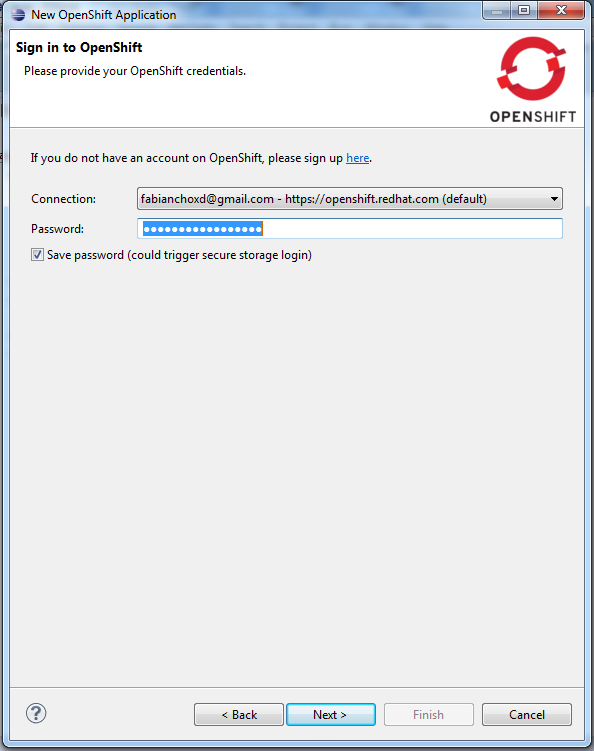
A continuación se explicará detalladamente el proceso de descarga y modificación del proyecto en la Nube.

En el respectivo IDE de desarrollo, en este caso Eclipse, Es necesario contar con JBoss Tools para realizar los respectivos pasos.

1. Seleccionar en Eclipse Nuevo proyecto.
2. Buscar y seleccionar la opción de “New Openshift Application”.



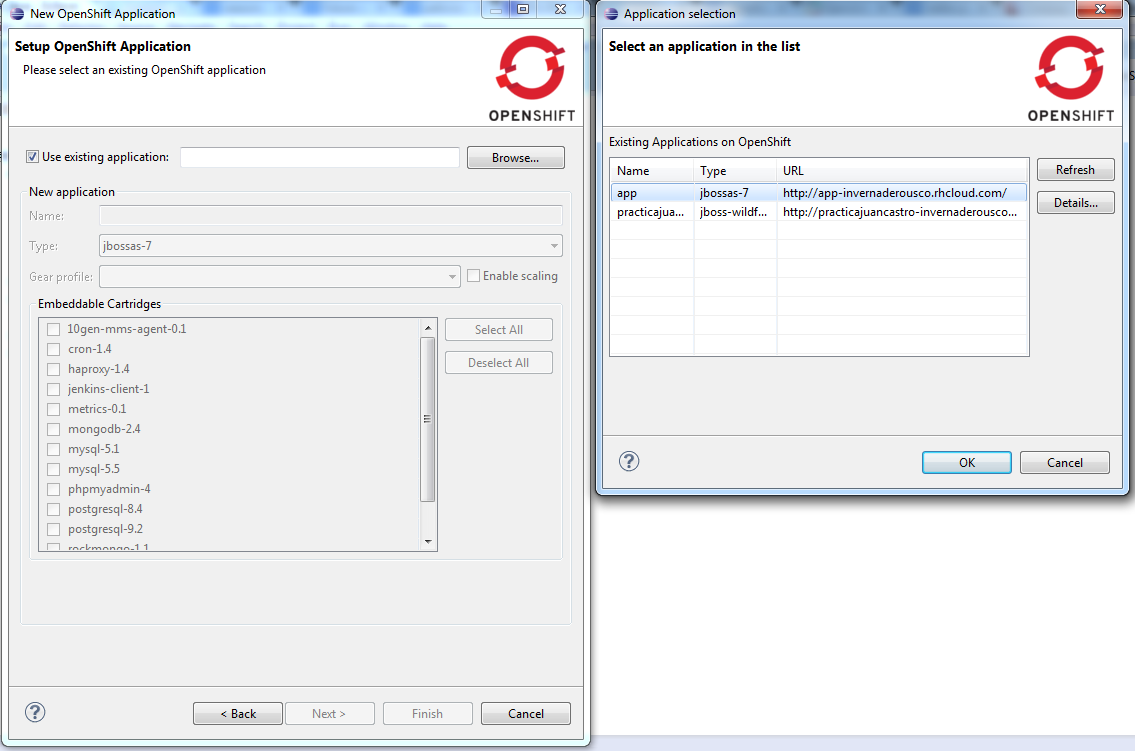
1. A continuación se pedirán las credenciales con las cuales se registraron en la Página Web de Openshift.

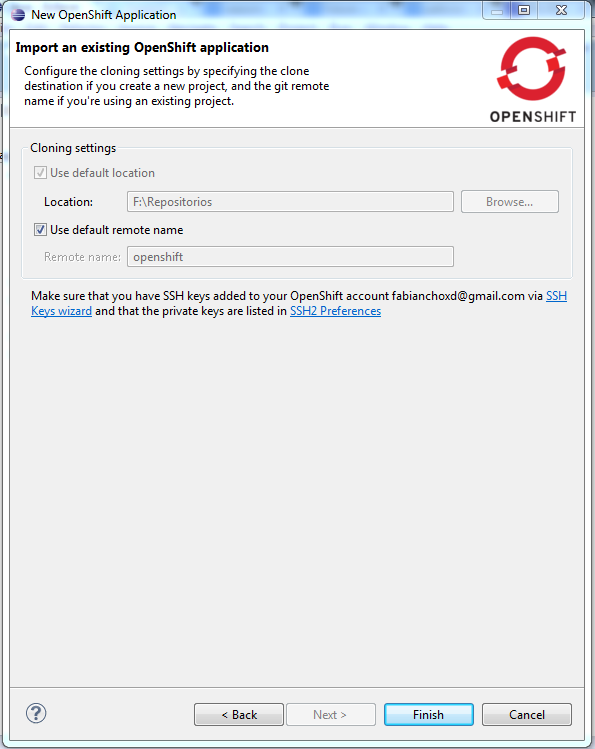


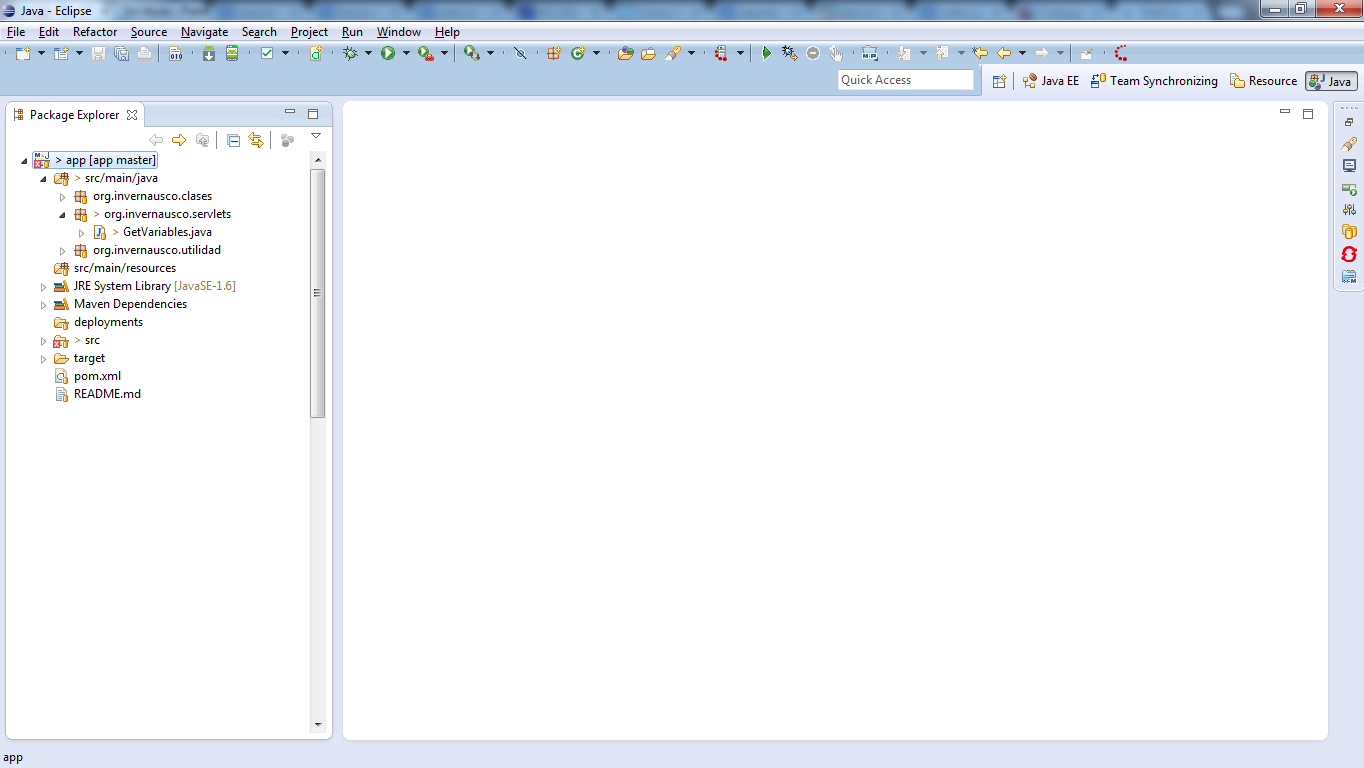
1. Luego se mostrarán dos opciones:  
   1. Si se quiere crear una aplicación nueva.

2. Si se quiere usar una aplicación existente.

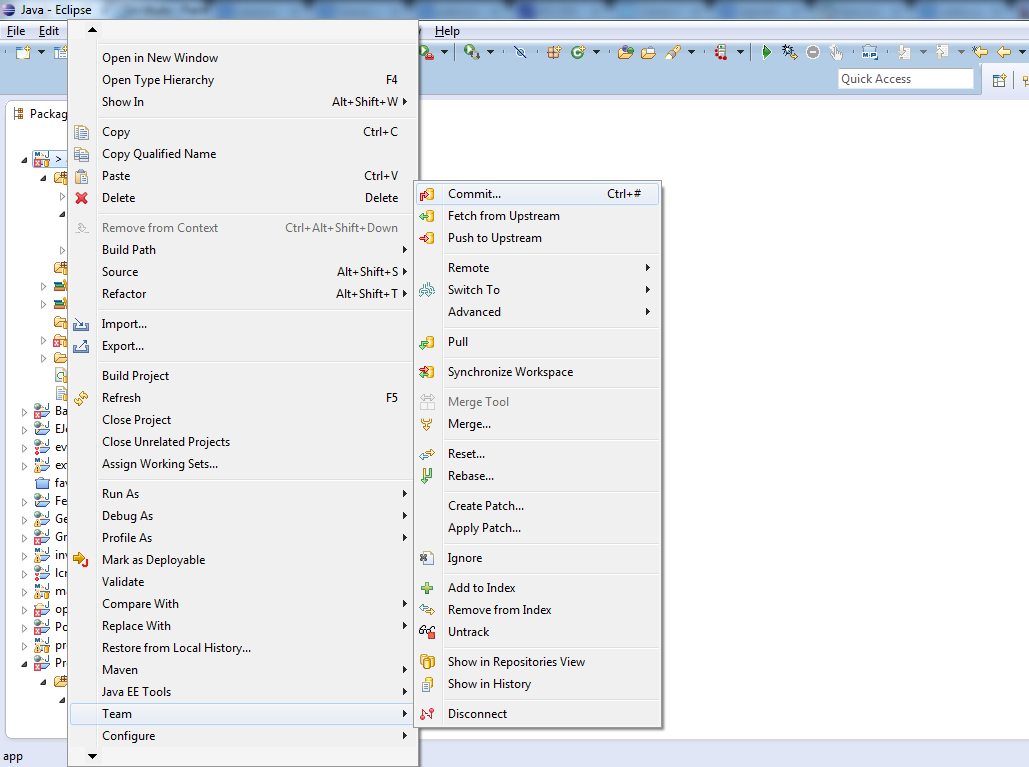
En este caso seleccionar una aplicación existente ya que esta aplicación se ha creado desde la página web. Y a continuación seleccionar la aplicación.

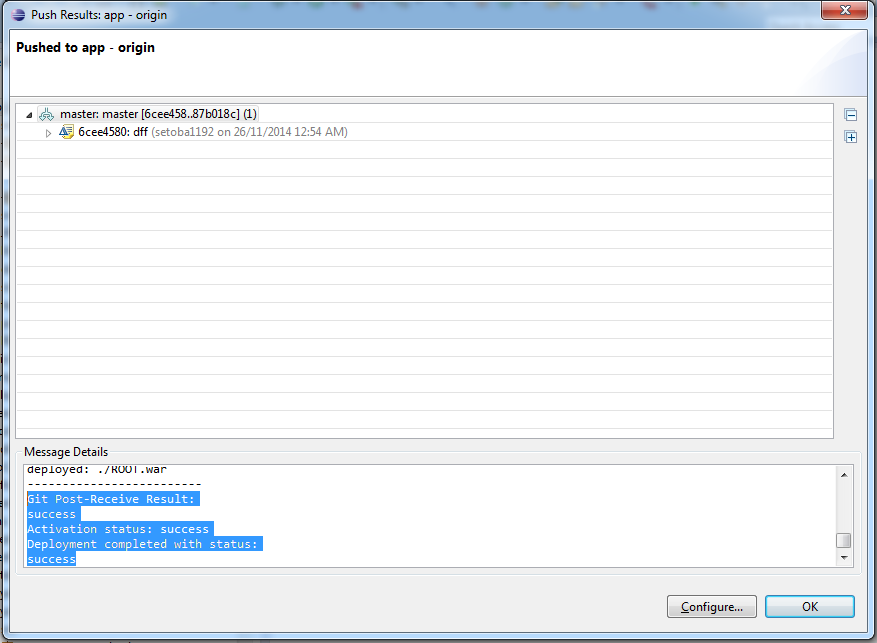


1. Luego se presiona en Finish y la aplicación será creada y descargada Localmente para proceder con sus respectivas modificaciones.
2. A continuación veremos el proyecto descargado y listo para ser trabajado.



1. Una vez realizada todas las respectivas modificaciones podremos emitir los cambios al proyecto en la Nube.



1. Después de subir los cambios a la Nube el IDE Eclipse informará de que la aplicación ha sido actualizada. Y ya se podrá acceder a la aplicación en la Nube.

ELABORADO

NOVIEMBRE 26 DE 2017

PROYECTO DE GRADO

FABIAN STEVENS VARON VALENCIA CÓDIGO: 20122112891

JOAN SEBASTIAN SANCHEZ CÓDIGO: 20111101762

UNIVERSIDAD SURCOLOMBIANA

NEIVA-HUILA

COLOMBIA

1. Casualmente, la pronunciación en inglés de este término es “yava”, que puede entenderse fuera de contexto como “¡ya va!”. [↑](#footnote-ref-1)
2. Un widget es una pequeña aplicación que facilita el acceso a funciones frecuentes. [↑](#footnote-ref-2)