UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

MANUAL TÉCNICO

TEMA:

"DESARROLLO DE UN APLICATIVO INFORMÁTICO PARA DIFUNDIR LOS BENEFICIOS Y USOS DE LAS PLANTAS MEDICINALES MEDIANTE DISPOSITIVOS MÓVILES"

AUTOR: GEOVANNY RAÚL CÁCERES ARÉVALO
DIRECTOR: ING. DIEGO TREJO

IBARRA - ECUADOR
2015

TABLA DE CONTENIDO

1 MANUAL TÉCNICO DEL SISTEMA ADMINISTRADOR Y APLICATIVO MÓVIL DE PLANTAS MEDICINALES CLASIFICADAS POR TIPO DE ENFERMEDAD.

1.1 INTRODUCCIÓN

En presente de este documento constituye una herramienta de soporte y ayuda para los administradores, operadores y en general para cualquier persona interesada en el concepto técnico del Sistema Administrador y Aplicativo Móvil de Plantas Medicinales y la estructura del mismo.

Este sistema fue desarrollado con herramientas de software libre y publicado en la nube.

A continuación las herramientas utilizadas para el desarrollo del sistema Administrador.

- ✓ Desarrollado en lenguaje Java con JPA y JSF tomando como framework Primefaces 5.1 y un IDE de programación Eclipse Luna.
- ✓ Servidor de aplicaciones Tomcat 7.
- ✓ Motor de base de datos PostgreSQL 9.2
- ✓ Plataforma como servicio (PaaS) Openshift para la publicación del sistema administrador.
- ✓ Se utilizó Java Server Faces que es un framework de desarrollo basado en el patrón MVC (Modelo Vista Controlador), porque normaliza y estandariza el desarrollo de aplicaciones web.

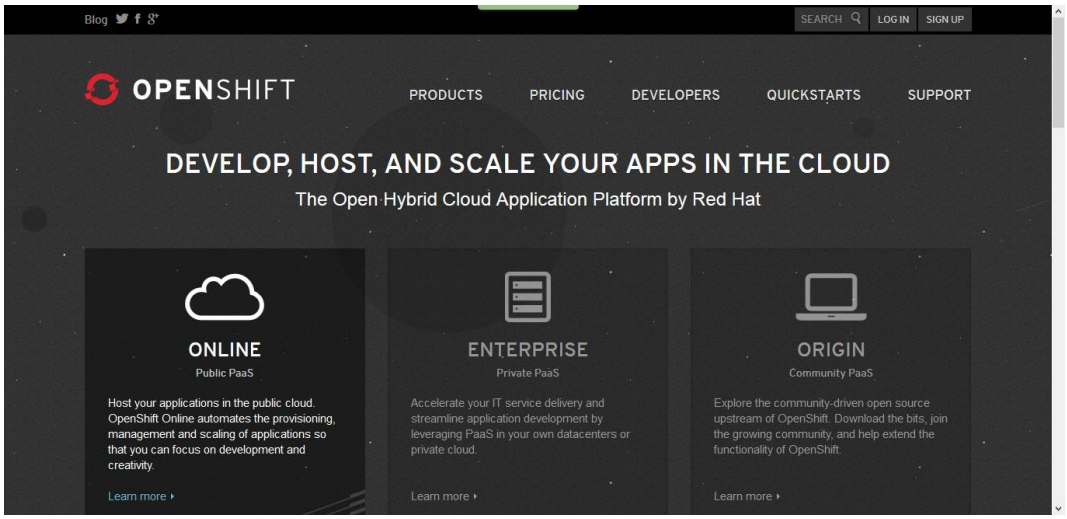
- ✓ Java Resources aquí están todos los archivos *.java y eclipse se encargará automáticamente de publicar los *.class en donde corresponda.
- ✓ Web aquí estarán los archivos *.xhtml, *.jsf, dentro de esta carpeta podemos ver la estructura estándar de JEE para un proyecto Web con la carpeta WEB-INF, el archivo web.xml.

1.2 OPENSHIFT

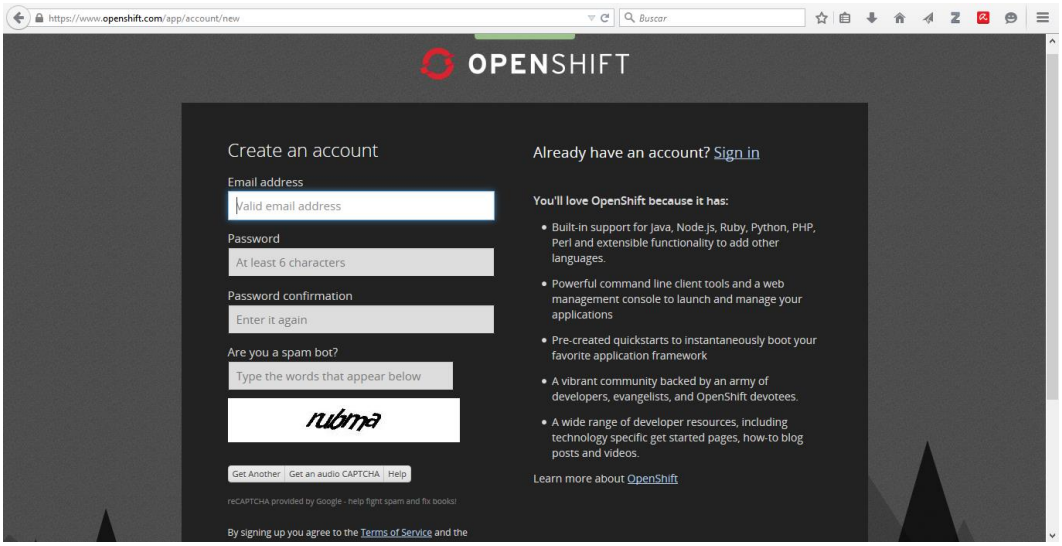
OpenShift es la plataforma como servicio (PaaS) de Red Hat, que permite a los desarrolladores crear, gestionar y desplegar rápidamente aplicaciones escalables en un entorno de nube.

1.1 CREACIÓN DE UNA CUENTA

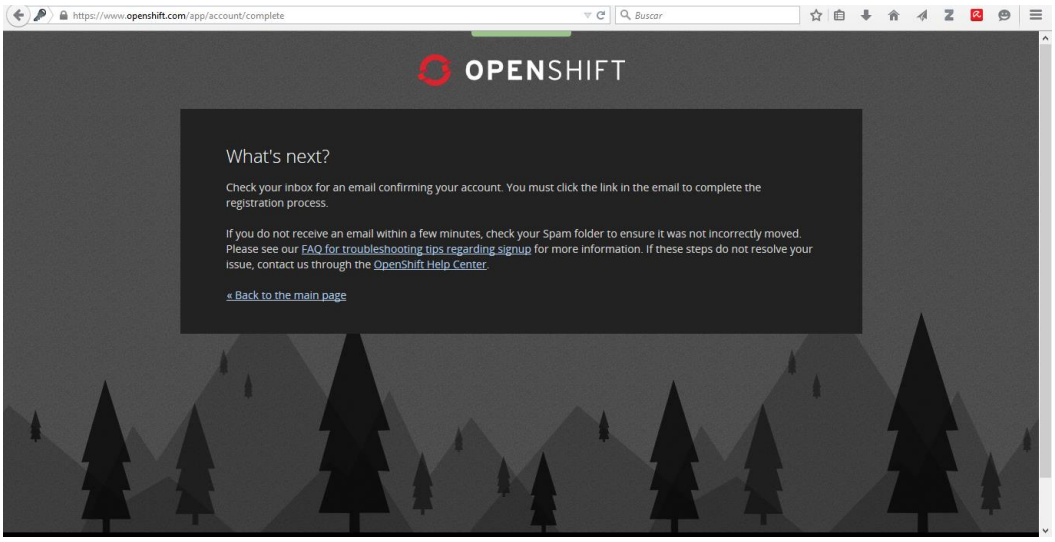
Para crear una cuenta nos dirigimos a https://www.openshift.com/, y pulsamos el botón **Sign Up**, que se encuentra en la parte superior derecha**.**



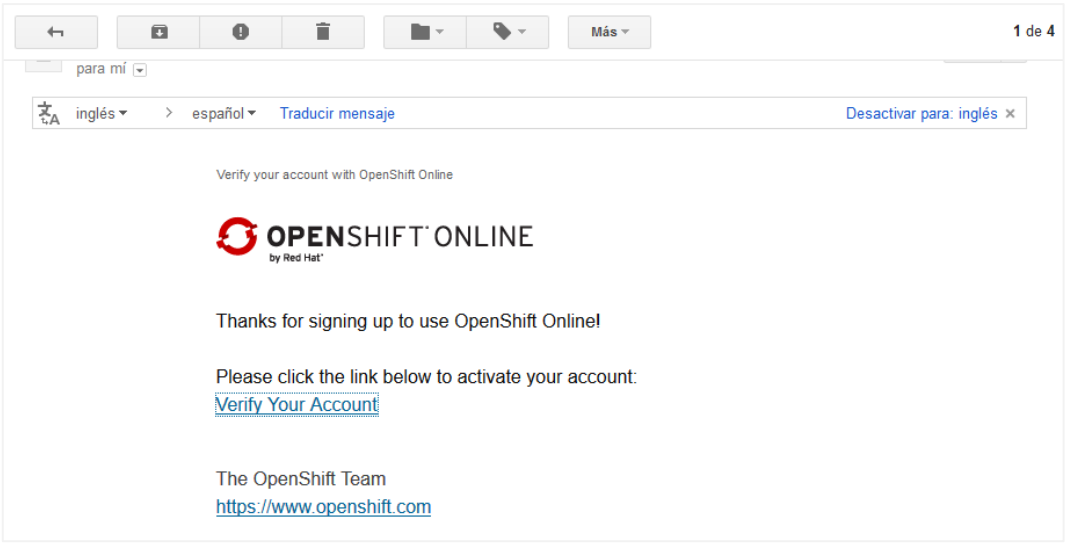
Luego se mostrara un formulario de registro donde ingresaremos los datos de nuestra cuenta, y pulsaremos el botón **Sign Up**.



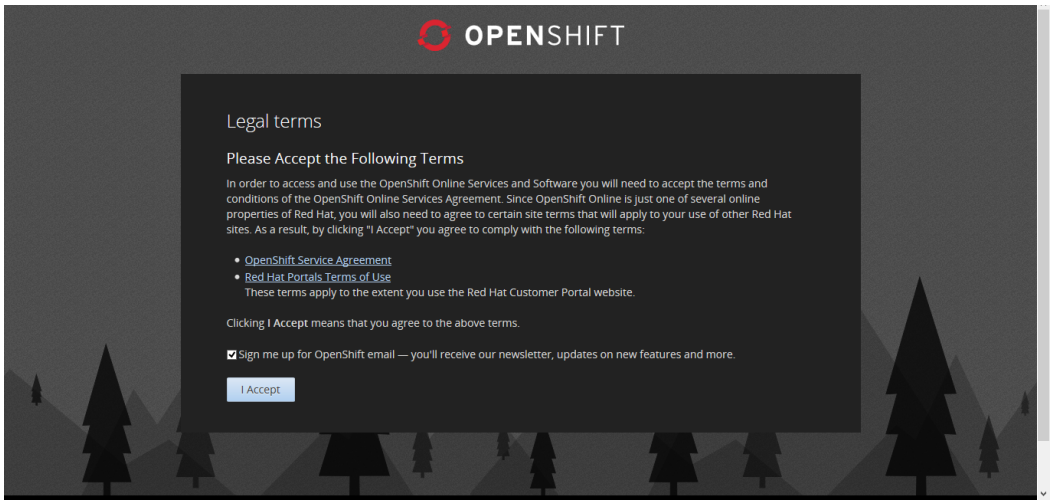
A continuación nos aparecerán las indicaciones para activar nuestra cuenta ya creada, en resumen no indican que debemos realizar este proceso mediante nuestro correo electrónico.



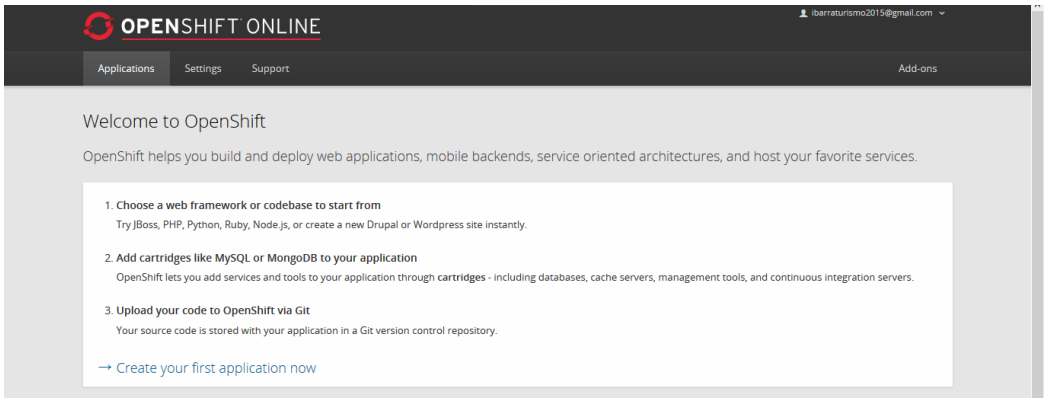
Al abrir nuestro mensaje dentro del correo electrónico, nos aparece un enlace de activación de cuenta.



Al dar clic en este nos aparece la aceptación de términos legales de uso de OpenShift.

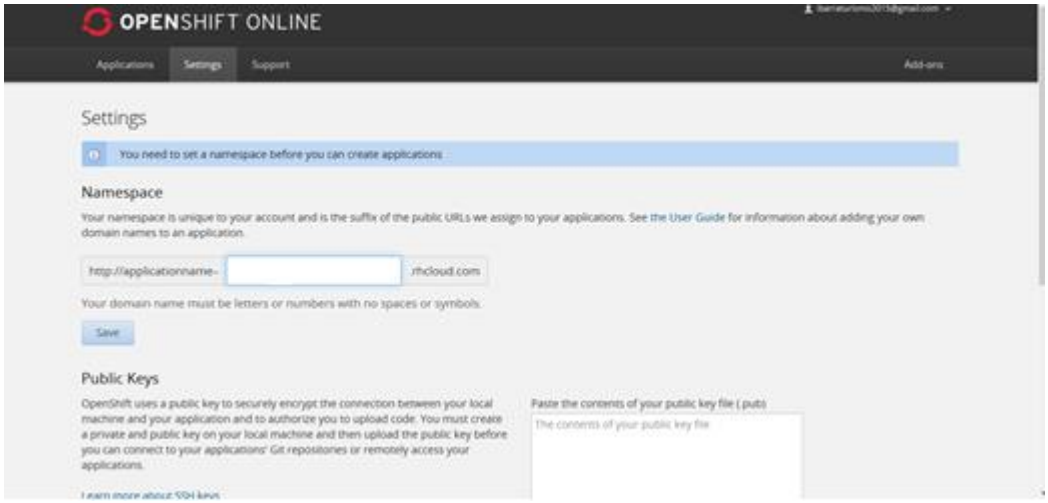


Finalmente se muestra nuestro escritorio de trabajo virtual de OpenShift.

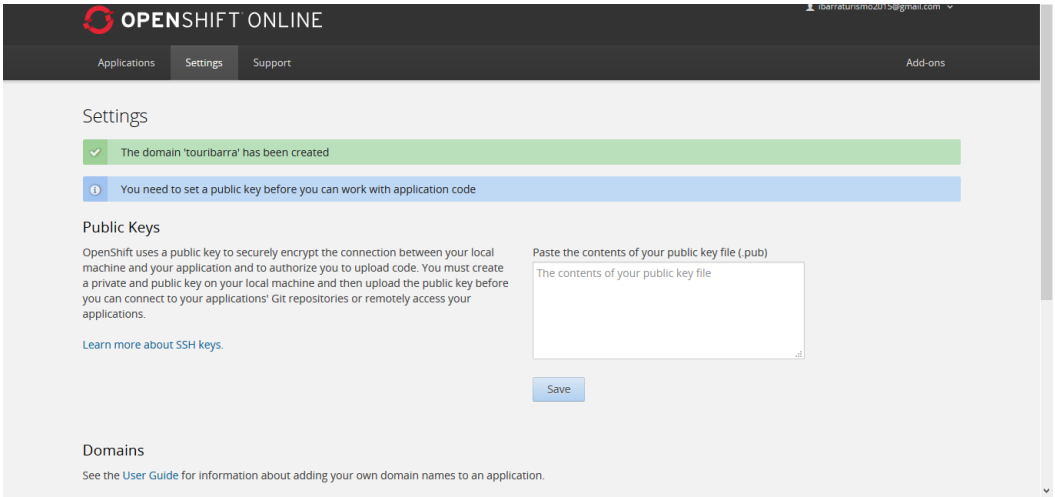


1.2 REGISTRO DE DOMINIO

El primer paso dentro de nuestra cuenta es la creación de un dominio, para esto nos dirigimos a la pestaña de **Settings** y en la sección **Namespace** nos aparece el cuadro de texto donde escribiremos el nombre del dominio, el cual será verificado para su disponibilidad.

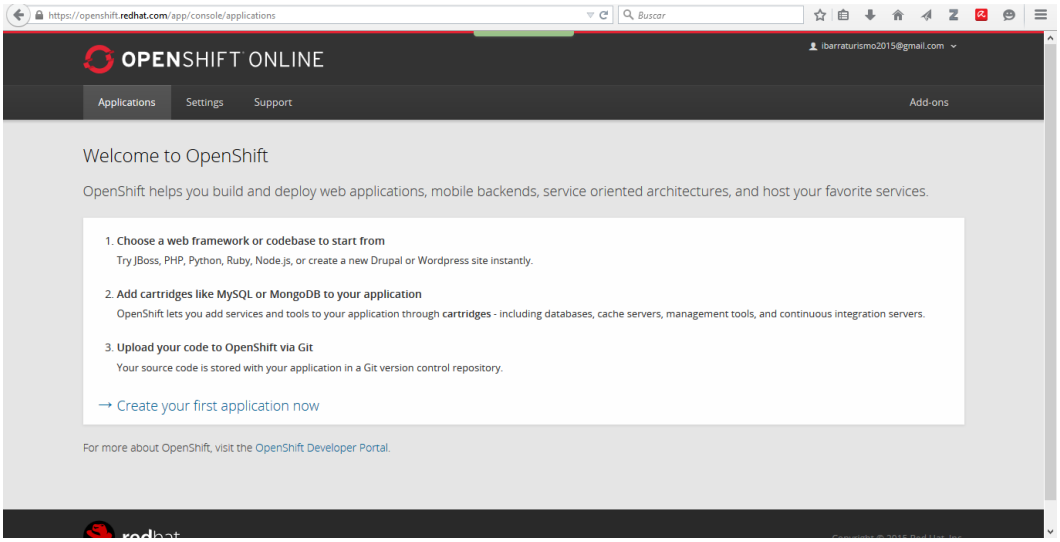


Al presionar el botón **Save** finalizaremos el proceso de creación de nuestro dominio y nos aparecerá un mensaje de éxito.

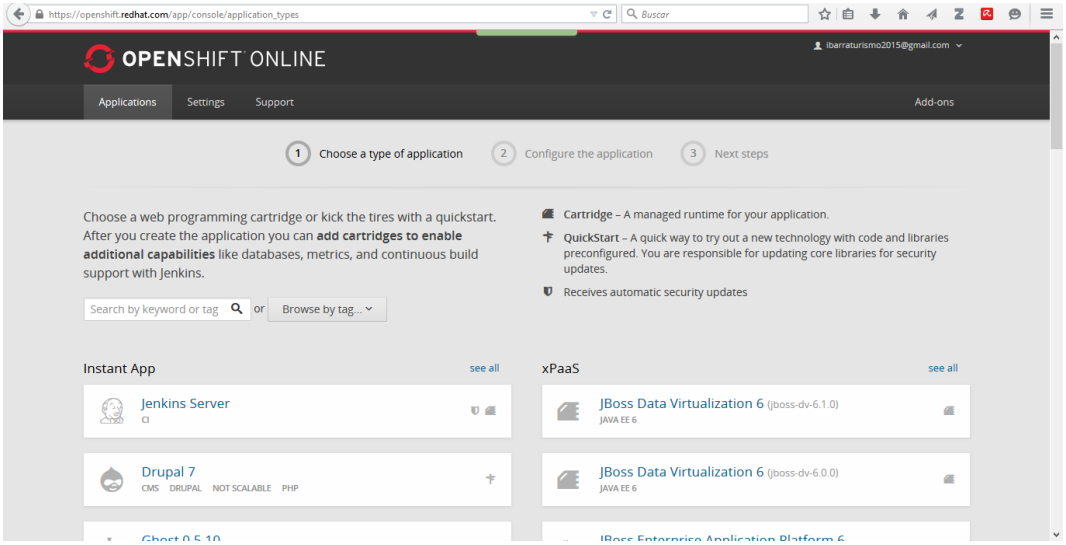


1.3 CREACIÓN DE UNA APLICACIÓN

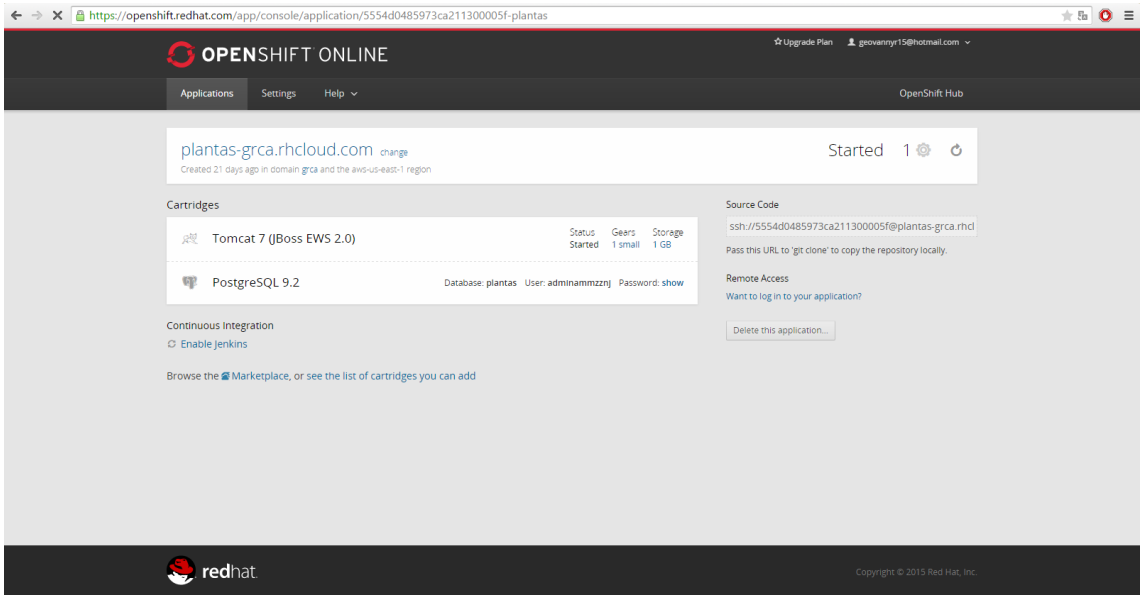
Dentro de nuestra cuenta tenemos la posibilidad de crear nuestras aplicaciones para modificarlas posteriormente. Para crear una aplicación nos dirigimos a la pestaña **Applications** y pulsamos **Create application.**



OpenShift ofrece varias posibilidades a los desarrolladores de aplicaciones como son JAVA, PHP, PHYTON, RUBY, entre otras. Basta con seleccionar la que necesitamos y seguir los pasos de forma intuitiva.

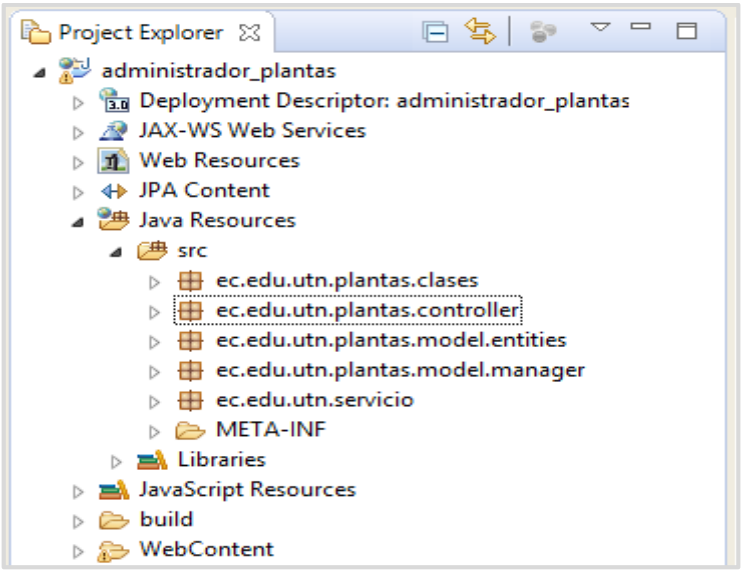


Ahora configuraremos nuestra aplicación y procederemos a crearla.



Una vez creada podemos verificar en el navegador haciendo click en el enlace

1.3 ESTRUCTURA BASE DEL SISTEMA ADMINISTRADOR



1.3.1 Java Resources

Paquete que contiene todo el código fuente, fue distribuido de la siguiente manera siguiendo el estándar modelo - vista - controlador.

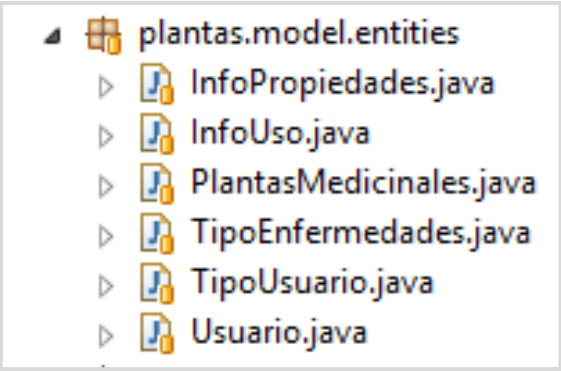
1.3.2 Controladores (Lógica de Negocios)

A) Estructura básica de una clase controladora

InfoUsoBean.java

```
1  package ec.edu.utn.plantas.controller;
2
3  import java.util.List;
12
13 @SessionScoped
14 @ManagedBean
15 public class InfoUsoBean {
16
17    private Integer id;
18    private String nombre;
19    private String descripcion;
20
21    private InfoUsoDao UsoDao;
22    private List<InfoUso> listado;
23
24    public List<InfoUso> getListado() {
25       listado = UsoDao.findAllInfoUso();
26       return listado;
27    }
28
29    public InfoUsoBean() {
30       UsoDao = new InfoUsoDao();
31    }
32    public Integer getId() {
33       return id;
34    }
```

1.3.3 Paquete entidades JPA

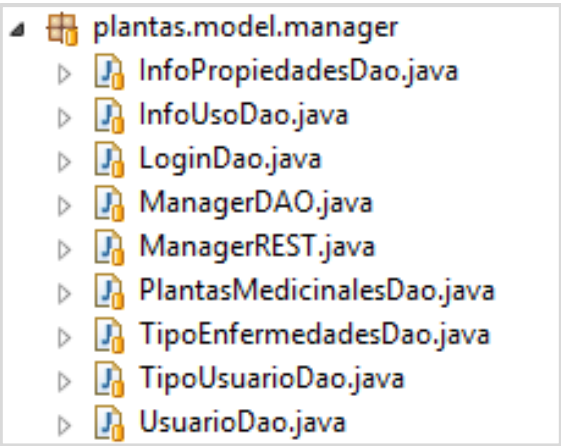


A) Estructura base de una entidad JPA

InfoUso.java

```java
package plantas.model.entities;

import java.io.Serializable;


/**
 * The persistent class for the info_uso database table.
 *
 */
@Entity
@Table(name="info_uso")
@NamedQuery(name="InfoUso.findAll", query="SELECT i FROM InfoUso i")
public class InfoUso implements Serializable {
	private static final long serialVersionUID = 1L;

	@Id
	@SequenceGenerator(name="INFO_USO_IDUSO_GENERATOR", sequenceName="SEC_USO",allocationSize=1)
	@GeneratedValue(strategy=GenerationType.SEQUENCE, generator="INFO_USO_IDUSO_GENERATOR")
	@Column(name="id_uso")
	private Integer idUso;

	private String descripcion;

	@Column(name="nombre_uso")
	private String nombreUso;

	public InfoUso() {
	}

	public Integer getIdUso() {
```

1.3.4 Paquetes ManagerDAO



A) Estructura básica del ManagerDAO

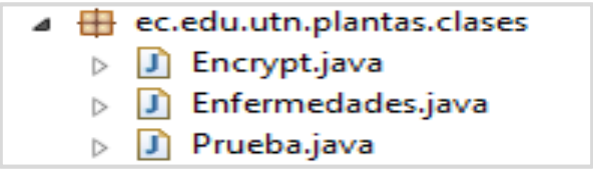
InfoPropiedadesDao.java

```
11
12 public class InfoPropiedadesDao {
13
14    private static EntityManagerFactory factory;
15    private static EntityManager em;
16    private ManagerDAO mDAO;
17
18    /**
19     * Constructor de la clase ManagerDAO. Se encarga de crear los objetos
20     * factory y entity manager utilizando el patron de diseño singleton.
21     */
22    public InfoPropiedadesDao() {
23
24       if (factory == null)
25          factory = Persistence
26                .createEntityManagerFactory("administrador_plantas");
27
28       if (em == null)
29          em = factory.createEntityManager();
30       mDAO = new ManagerDAO();
31    }
32
33    // METODO FINDER PARA CATEGORIAS
34
35    @SuppressWarnings("unchecked")
36    public List<InfoPropiedades> findAllInfoPropiedades() {
37       List<InfoPropiedades> listado;
38       Query q;
39       em.getTransaction().begin();
40       q = em.createQuery("SELECT u FROM InfoPropiedades u ORDER BY u.nombrePropiedad");
41       listado = q.getResultList();
42       em.getTransaction().commit();
43       return listado;
```

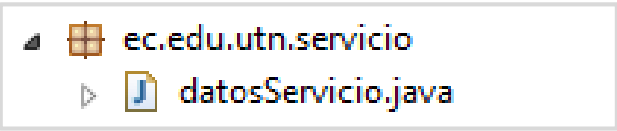
InfoPropiedadesDao.java

```
48
49  public void crearInfoPropiedades(String nombre, String descripcion) {
50      em.getTransaction().begin();
51      InfoPropiedades t = new InfoPropiedades();
52
53      t.setNombrePropiedad(nombre);
54      t.setDescripcionPropiedad(descripcion);
55      em.persist(t);
56      em.getTransaction().commit();
57
58  }
59
60  // METODO FINDER PARA ENCONTRAR UN OBJETO CATEGORIA ESPECIFICO
61
62  public InfoPropiedades findByCodInfoPropiedades(Integer id) {
63      em.getTransaction().begin();
64      InfoPropiedades t = em.find(InfoPropiedades.class, id);
65      em.getTransaction().commit();
66      return t;
67
68  }
69
70  // METODO PARA BORRAR UNA CATEGORIA
71
72  public void eliminarInfoPropiedades(Integer id) {
73      // PRIMERO BUSCAMOS EL OBJETO QUE DEBE SER BORRADO
74      InfoPropiedades t = findByCodInfoPropiedades(id);
75      em.getTransaction().begin();
76      em.remove(t);
77      em.getTransaction().commit();
```

1.3.5 Paquete de clases extras



1.3.6 Paquete Servicio REST

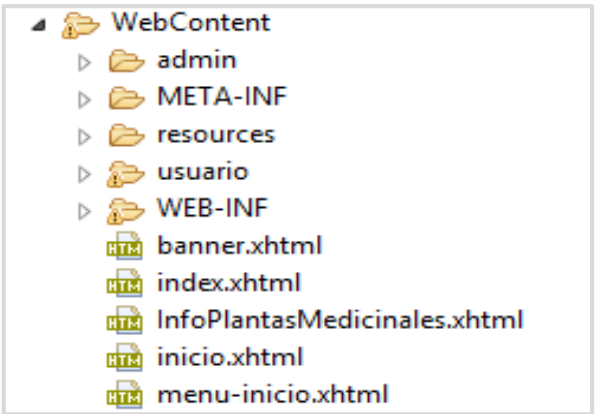


A) Estructura básica de un servicio REST

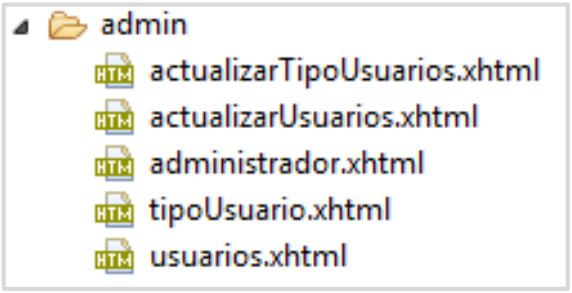
datosServicio.java

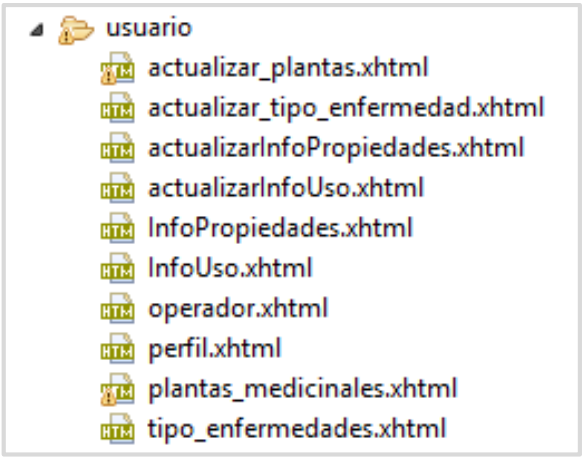
```java
@WebServlet("/datosServicio")
public class datosServicio extends HttpServlet {
  private static final long serialVersionUID = 1L;

  /**
   * @see HttpServlet#HttpServlet()
   */
  private ManagerREST mngRest;
  public datosServicio() {
    super();
    mngRest=new ManagerREST();
  }

  /**
   * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
   */
  protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    response.addHeader("Access-Control-Allow-Origin", "*");
    response.addHeader("Access-Control-Allow-Methods", "GET, PUT, POST, OPTIONS, DELETE");
    response.addHeader("Access-Control-Allow-Headers", "Content-Type");
    response.addHeader("Access-Control-Max-Age", "86400");

    String JSON = allData();

    response.getWriter().write(JSON);
    response.getWriter().close();
  }

  private String allData(){
    return mngRest.datosAppString();
```

1.3.7 Web Content (vista)

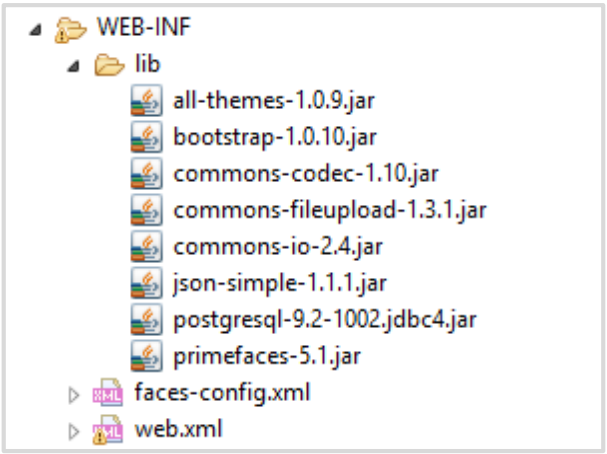


1.3.8 Paquete páginas xhtml





1.3.9 Paquete WEB-INFO (librerías)



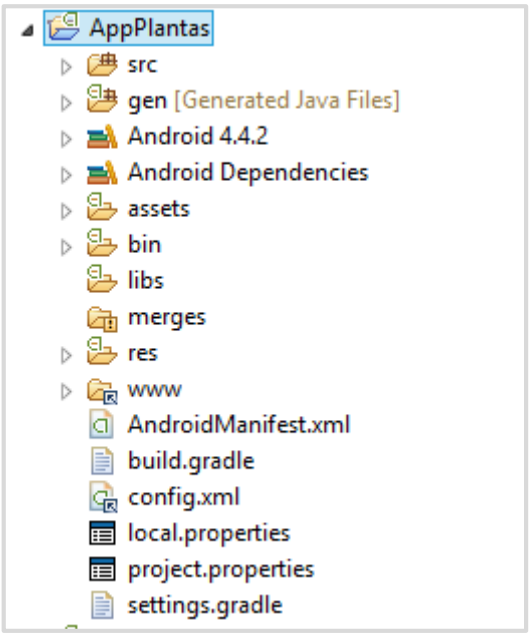
1.3.10 Web.xml

web.xml

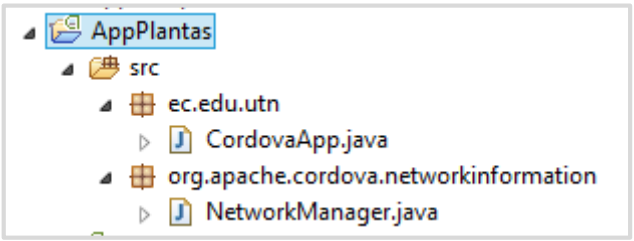
```
  </welcome-file-list>
  <servlet>
    <servlet-name>Faces Servlet</servlet-name>
    <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
    <url-pattern>/faces/*</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
    <url-pattern>*.xhtml</url-pattern>
  </servlet-mapping>
  <context-param>
    <description>State saving method: 'client' or 'server' (=default). See JSF Specification 2.5.2</description>
    <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
    <param-value>client</param-value>
  </context-param>
  <context-param>
    <param-name>javax.servlet.jsp.jstl.fmt.localizationContext</param-name>
    <param-value>resources.application</param-value>
  </context-param>
  <filter>
    <filter-name>FiltroSesion</filter-name>
    <filter-class>plantas.controller.FilterSesion</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>FiltroSesion</filter-name>
    <url-pattern>/faces/usuario/*</url-pattern>
    <url-pattern>/faces/admin/*</url-pattern>
  </filter-mapping>
  <context-param>
    <param-name>primefaces.THEME</param-name>
    <param-value>south-street</param-value>
  </context-param>
  <listener>
    <listener-class>com.sun.faces.config.ConfigureListener</listener-class>
  </listener>
</web-app>
```

1.4 ESTRUCTURA BÁSICA DE LA APLICACIÓN MÓVIL ANDROID CON PHONEGAP

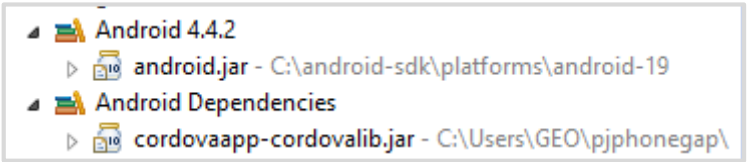
1.4.1 Paquete src (Código Fuente)



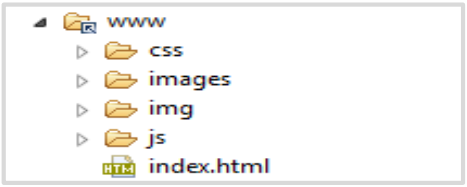
1.4.2 Actividades (controladores)



1.4.3 Librerías



1.4.4 Paquete www



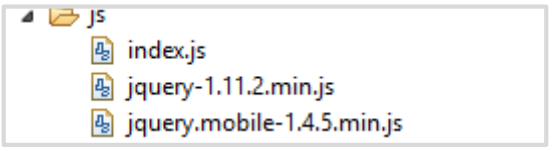
1.4.5 Paquete Css (estilos)

A) Estructura básica de una hoja de estilos Css

plantas.css

```
html {
    font-size: 100%;
}
body,
input,
select,
textarea,
button,
.ui-btn {
    font-size: 1em;
    line-height: 1.3;
     font-family: sans-serif /*{global-font-family}*/;
}
legend,
.ui-input-text input,
.ui-input-search input {
    color: inherit;
    text-shadow: inherit;
}
/* Form labels (overrides font-weight bold in bars, and mini font-size) */
.ui-mobile label,
div.ui-controlgroup-label {
    font-weight: normal;
    font-size: 16px;
}
/* Separators
------------------------------------------------------------------------------------------*/
/* Field contain separator (< 28em) */
.ui-field-contain {
    border-bottom-color: #828282;
    border-bottom-color: rgba(0,0,0,.15);
    border-bottom-width: 1px;
    border-bottom-style: solid;
}
/* Table opt-in classes: strokes between each row, and alternating row stripes */
/* Classes table-stroke and table-stripe are deprecated in 1.4. */
.table-stroke thead th,
.table-stripe thead th,
.table-stripe tbody tr:last-child {
    border-bottom: 1px solid #d6d6d6; /* non-RGBA fallback */
```

1.4.6 Paquete JavaScript



A) Estructura básica de un archivo JavaScript

index.js

```
 *
 * Unless required by applicable law or agreed to in writing,
 * software distributed under the License is distributed on an
 * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
 * KIND, either express or implied.  See the License for the
 * specific language governing permissions and limitations
 * under the License.
 */
var app = {
    // Application Constructor
    initialize: function() {
        this.bindEvents();
    },
    // Bind Event Listeners
    //
    // Bind any events that are required on startup. Common events are:
    // 'load', 'deviceready', 'offline', and 'online'.
    bindEvents: function() {
        document.addEventListener('deviceready', this.onDeviceReady, false);
    },
    // deviceready Event Handler
    //
    // The scope of 'this' is the event. In order to call the 'receivedEvent'
    // function, we must explicitly call 'app.receivedEvent(...);'
    onDeviceReady: function() {
        app.receivedEvent('deviceready');
    },
    // Update DOM on a Received Event
    receivedEvent: function(id) {
        var parentElement = document.getElementById(id);
        var listeningElement = parentElement.querySelector('.listening');
        var receivedElement = parentElement.querySelector('.received');

        listeningElement.setAttribute('style', 'display:none;');
        receivedElement.setAttribute('style', 'display:block;');

        console.log('Received Event: ' + id);
    }
};

app.initialize();
```

1.4.7 Acceso a Servicio Web

```
$.ajax({
    url: "http://plantas-grca.rhcloud.com/datosServicio",
    success: function(data) {
      alert(data);
    },
    error: function() {
        alert(error);
    }
});

//CARGA DE PAGINA
$('#pageMain').on('pageinit', function () {
  setTimeout(function () {
    $(':mobile-pagecontainer').pagecontainer('change', '#home', {
      transition: 'pop',
      changeHash: false,
      reverse: true
    });
  }, 1000);
});
//CARGA LISTA ENFERMEDADES
$(document).on('pagecreate', '#enfermedades-page', function(){
  var li="";

    $.each(jsonfile.enfermedades, function (i, enfermedades) {
        li += '<li><a href="#" id="' + i + '" class="ui-li-icon cat-go">'+
       '<font style="text-transform: capitalize;">'+ enfermedades.enfermedad + '</font></a></li>';
    });

    $('#lista-enfermedades').append(li).promise().done(function () {
        $(this).on('click', '.cat-go', function(e){
          e.preventDefault();
          $('#plantas-enfermedad-page').data('jsonfile', jsonfile.enfermedades[this.id]);
          $(':mobile-pagecontainer').pagecontainer('change','#plantas-enfermedad-page');
        });

        $(this).listview('refresh');
    });
});
```

1.4.8 Archivo Manifest

Archivo principal de Android en donde se pone en manifiesto: versión soportada mínima y máxima, permisos de red, actividades principales y secundarias.

AppPlantas Manifest

```
<?xml version='1.0' encoding='utf-8'?>
<manifest android:hardwareAccelerated="true" android:versionCode="1" android:versionName="0.0.1" package="ec.edu.utn" xmlns:android="http://schemas.android.com/apk/res/android">
    <supports-screens android:anyDensity="true" android:largeScreens="true" android:normalScreens="true" android:resizeable="true" android:smallScreens="true" android:xlargeScre
    <uses-permission android:name="android.permission.INTERNET" />
    <application android:hardwareAccelerated="true" android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:configChanges="orientation|keyboardHidden|keyboard|screenSize|locale" android:label="@string/activity_name" android:launchMode="singleTop" android:name
            <intent-filter android:label="@string/launcher_name">
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-sdk android:minSdkVersion="10" android:targetSdkVersion="19" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
</manifest>
```