

## Avance del Proyecto

### Integrantes:

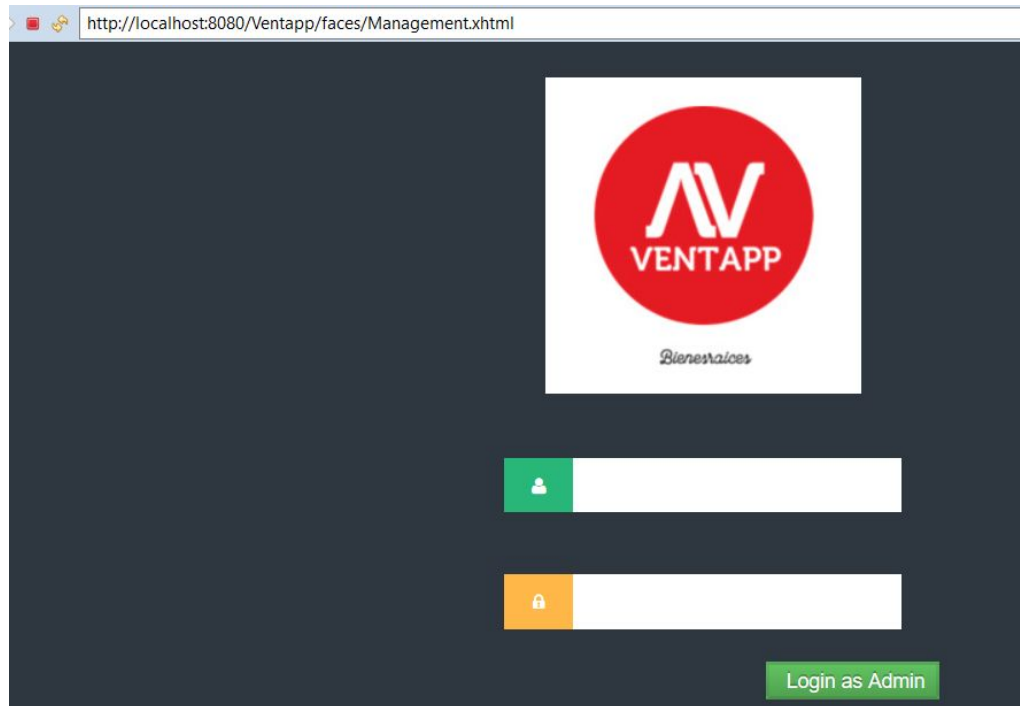
Jacqueline Maldonado

Javier Mendez


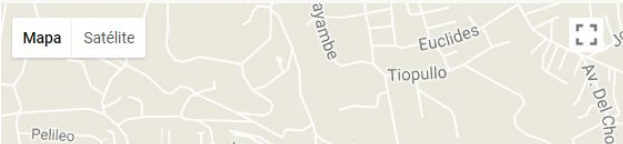
Jonnathan Chalco

Adrian Poveda

### Administracion



### Asignación de imagenes



latitud	<input type="text" value="0.0"/>
longitud	<input type="text" value="0.0"/>
Ciudad	<input type="text" value="Cuenca"/>
Sector	<input type="text"/>
Categoría	<input type="text" value="departamento"/>
Ingrese email propietario: <input type="text"/>	
Código cliente	<input type="text" value="0"/>
Imágenes	<div> <div>Descripción terreno 1</div> <div>  <input type="button" value="Seleccionar archivo"/> </div> </div> <div><input type="button" value="CARGAR IMAGEN"/></div>
Código	Descripción Eliminar
latitud	<input type="text" value="0.0"/>
longitud	<input type="text" value="0.0"/>
Selecciona la ubicación en el mapa	
<div> <input checked="" type="button" value="Mapa"/> <input type="button" value="Satélite"/> </div> 	

Una vez seleccionada la imagen con el botón seleccionar archivo se procede a cargar la imagen puesto que una propiedad puede tener una o más propiedades

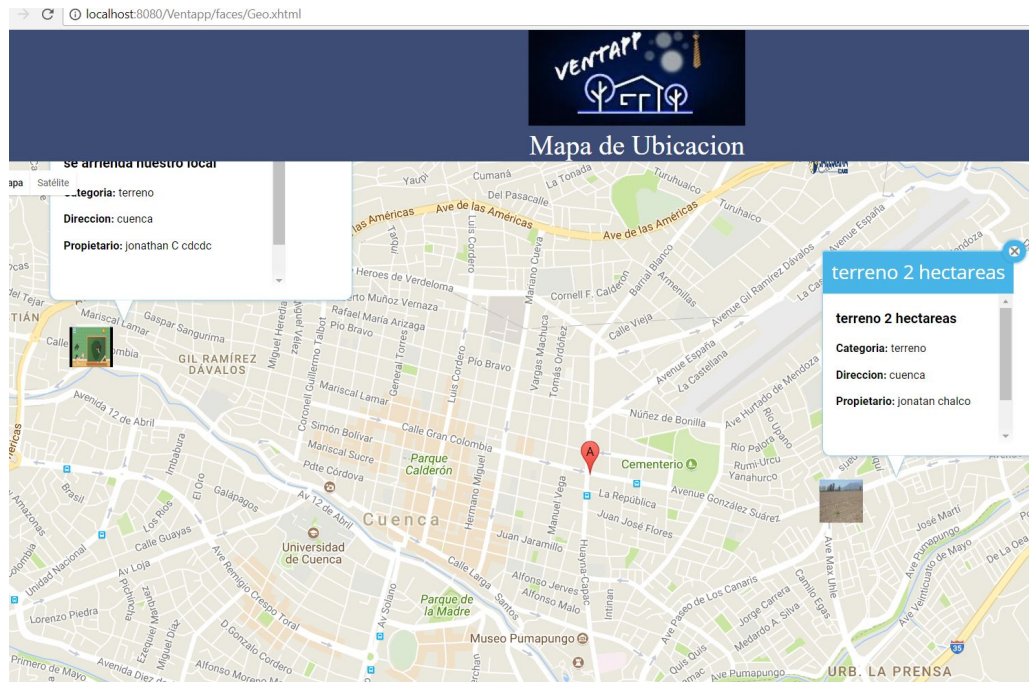
En la página de inicio se encuentra la lista de propiedades registradas las mismas tienen un detalle al que se puede acceder mediante el botón Ir



Propiedades publicadas						
Direccion	Url video	Descripcion	Estado	Tipo	Costo	
cuenca fdf	rfewf	rfew	efref	frefer	3243	lr
cqere	cecqervcq	decvefve	cecev	cceve	2312341	lr
hola	few	efw	few	ec	34	lr
ewrwe hola mijin	ewr	erw	ewr	wer	34	lr
guayaquil	frvrgrv	cdecwec	1	arriendo	32324	lr
cuenca	ffgeg	se arrienda nuestro local	en renta	local	134	lr
wedwfrew hola	cdwec	cwevc	cdwdevc	cdwvc	2312341	lr
cuenca	cecqervcq	departamento de 3 cuartos	en renta	arriendo	2312341	lr
cuenca	efvfev	se arrienda nuestro local	en renta	arriendo	2312341	lr
cuenca	efvfev	terreno 2 hectareas	1	venta	2312341	lr

Descripcion	terreno 2 hectareas
Direccion :	cuenca
# num imagenes :	4
Imágenes	
Descripcion	Imagen
foto terreno 1	
foto terreno 2	

De igual manera lo podemos encontrar en el mapa gracias a las coordenadas registradas en la propiedad.



## Creación de los web services

- ▼ Ventapp
  - ▼ JAX-RS Web Services
    - > GET /ventapp/login/user?email={String}&pass={String}
    - > GET /ventapp/propiedad/categoria?costo={int}
    - > POST /ventapp/propiedad/register
    - > GET /ventapp/propiedad/{id:[0-9][0-9]\*}
    - > GET /ventapp/user
    - > GET /ventapp/user/email?email={String}
    - > GET /ventapp/user/login?email={String}&pass={String}
    - > POST /ventapp/user/register
    - > GET /ventapp/user/{id:[0-9][0-9]\*}
  - Deployment Descriptor: Ventapp
  - ▼ JAX-WS Web Services
    - Service Endpoint Interfaces
    - Web Services
  - Web Resources
  - JPA Content
  - Java Resources
  - JavaScript Resources
  - ▼ Deployed Resources
    - webapp
      - resources
      - WEB-INF
        - comentario.xhtml
        - comentarios.xhtml

```

@POST
@Path("/register")
@Produces("application/json")
@Consumes("application/json")
public Response register(Persona persona) {
    Response rs= new Response();

    try {

        personaDao.guardar(persona);
        rs.setCodigo(405);
        rs.setMsj("datos guardados");
        return rs;
    }
    catch (Exception e) {
        // TODO: handle exception
        rs.setCodigo(402);
        rs.setMsj("error al inserar");
        return rs;
    }
}

@GET
@Path("/{id:[0-9][0-9]*}")//cualquier codigo que sea int
@Produces(MediaType.APPLICATION_JSON)
public Persona lookupUserById(@PathParam("id") int id) {
    Persona user = repository.findById(id);
    if (user == null) {
        //throw new WebApplicationException(Response.Status.NOT_FOUND);
        System.out.println("no existe");
    }
    return user;
}

```

## Creacion del Cliente

```

public void main() {

    Persona persona = guardar();
    Gson gson = new Gson();

    if(persona !=null){
        String personaJson = gson.toJson(persona);

        Client cliente= Client.create();

        //que servicio consume
        WebResource recurso = cliente.resource("http://localhost:8080/Ventapp/ventapp/user/register");

        System.err.println("conectado");

        ClientResponse respuesta = recurso.type(MediaType.APPLICATION_JSON).
            post(ClientResponse.class,personaJson);

        respuesta.bufferEntity();

        String respuestaJson =respuesta.getEntity(String.class);

        //pasr de un obj a un String
        System.err.println("Guardado" +respuestaJson);
    }
}

```

enteinsertarpersonas.CienteInsertarPersonas > guardar >

- ClienteInsertarPersonas (run) X

run:

conectado

Guardado{"codigo":405,"msj":"datos guardados"}

BUILD SUCCESSFUL (total time: 3 seconds)

## Avance aplicacion Android

### 1. Creación de ventanas de interfaces con la siguiente estructura

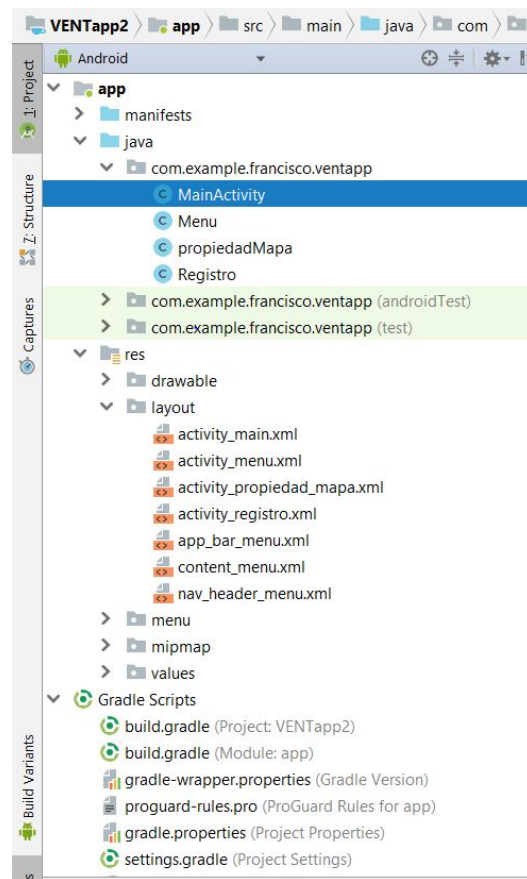


figura 1. Imagen estructura android

### 2. Creación de ventana de Login de acceso

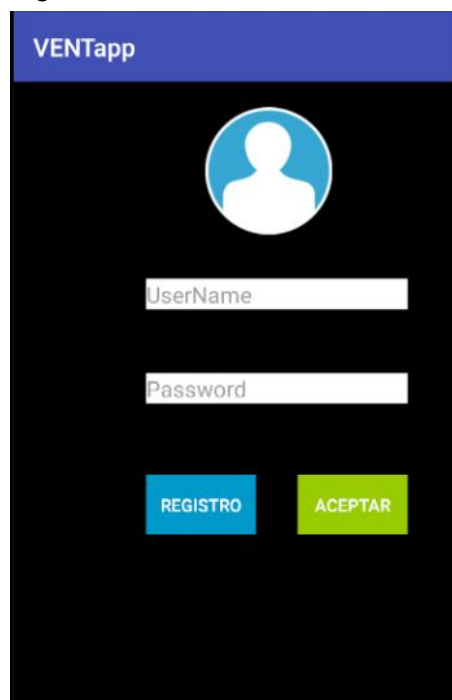


figura 2. Imagen de Login



```

public String clienteRest(String correo, String clave){
    if (android.os.Build.VERSION.SDK_INT > 9) {
        StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);
    }

    try {

        URL url = new URL( spec: "http://192.168.43.105:8080/Ventapp/ventapp/login/user?email="+correo+"&pass="+clave);
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setRequestMethod("GET");
        conn.setRequestProperty("Accept", "application/json");

        if (conn.getResponseCode() != 200) {
            throw new RuntimeException("Failed : HTTP error code : "
                + conn.getResponseCode());
        }

        BufferedReader br = new BufferedReader(new InputStreamReader(
            (conn.getInputStream())));

        String output;
        System.out.println("Output from Server .... \n");
        while ((output = br.readLine()) != null) {
            System.out.println(output);
            return output;
        }

        conn.disconnect();
    }
}

```

figura 2.1 Codigo de conexion cliente android

### 3. Creacion de ventana de Registro de usuario

VENTapp

Crear Cuenta

UserName

UserName

Date

Direction

Email

Password

Phone

Tipe

Telefono

GUARDAR

VOLVER

figura 3. Imagen Registro usuario



```

public String clienteRest(String correo, String clave) {
    if (Build.VERSION.SDK_INT > 9) {
        StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);
    }

    try {
        JSONObject ObjT = new JSONObject();
        ObjT.accumulate( name: "numero", telefono.getText().toString());
        ObjT.accumulate( name: "tipo", tipo.getText().toString());

        JSONObject Obj = new JSONObject();
        Obj.accumulate( name: "nombres", nombre.getText().toString());
        Obj.accumulate( name: "apellido", apellido.getText().toString());
        Obj.accumulate( name: "fecha", fecha.getText().toString());
        Obj.accumulate( name: "direccion", direccion.getText().toString());
        Obj.accumulate( name: "email", email.getText().toString());
        Obj.accumulate( name: "password", password.getText().toString());
        Obj.accumulate( name: "telefonos", ObjT.toString());

        System.out.print("Resultado Obj >>> "+Obj);

        URL url = new URL( spec "http://192.168.43.105:8080/Ventapp/ventapp/user/register?" + Obj);
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setRequestMethod("GET");
        conn.setRequestProperty("Accept", "application/json");

        if (conn.getResponseCode() != 200) {
            throw new RuntimeException("Failed : HTTP error code : "
                + conn.getResponseCode());
        }

        BufferedReader br = new BufferedReader(new InputStreamReader(
            (conn.getInputStream())));

        String output;
    }
}

```

figura 3.2 Código de Registro

#### 4. Creacion de menu para interacción de ventanas

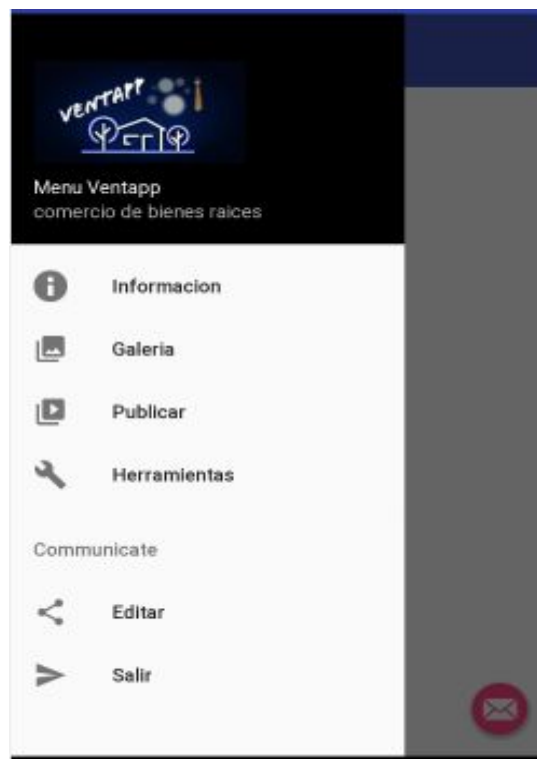
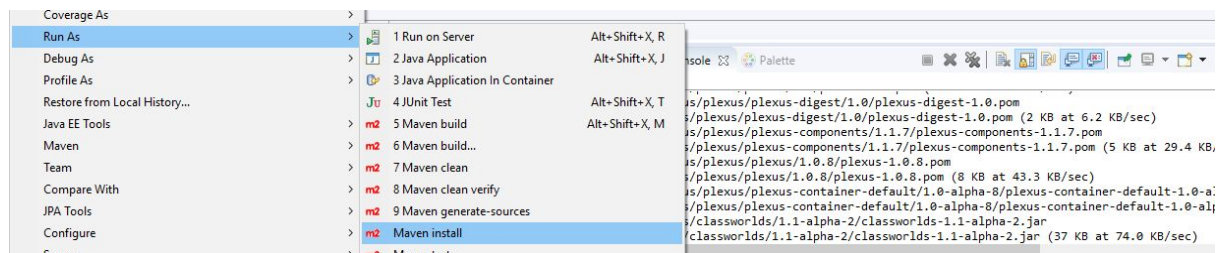


figura 4. Imagen menu

#### 5. Subida de archivos en general (Imagen,video,imagen 360)

Se utiliza dependencias Maven, para realizar la subida de un archivo. Se escoge según pop-up en vista, al escoger se obtiene ubicación de imagen, para almacenar dirección en cliente y subir a la carpeta \Resources\ en el servidor.

Para obtener dependencias, se configura el pom.xml. Y luego se instalan las correspondientes:



Vista:

File:

C:\Users\Marcelo\Documents\Scanned Documents\Digitalización de bienvenida.jpg

Upload File

File:

C:\Users\Marcelo\Pictures\Presentacion.avi

Upload File