

# **Geolocalización de Sitios de Interés Para Aplicaciones Móviles**

## **G-SIAM**



## **Estudio de Factibilidad**

**VERSIÓN**

1.2



## Contenido

<b>1</b>	<b><u>FACTIBILIDAD ECONÓMICA.....</u></b>	<b><u>3</u></b>
<b>2</b>	<b><u>FACTIBILIDAD TÉCNICA.....</u></b>	<b><u>3</u></b>
<b>2.1</b>	<b><u>TECNOLOGÍA INVOLUCRADA.....</u></b>	<b><u>3</u></b>
2.1.1	SPRINGANDROID.....	3
2.1.2	RESTEASY .....	4
2.1.3	FACEBOOK API.....	4
<b>3</b>	<b><u>CONCLUSIÓN.....</u></b>	<b><u>6</u></b>



# 1 Factibilidad Económica

No existe un factor de costo económico, ya que se trata de un proyecto de grado universitario, por ende, la mano de obra involucrada no es considerada en los costos del proyecto. A demás dicho proyecto será lanzado a la comunidad Open Source por lo que todas las herramientas utilizadas respetan esta filosofía.

## 2 Factibilidad Técnica

El estudio de factibilidad técnica se realiza mediante una prueba de concepto en una etapa temprana del proyecto. Considerando la definición del alcance del proyecto y el conocimiento técnico del equipo, se identifican tres funcionalidades que se presentan como determinantes para evaluar la factibilidad técnica:

- Llamadas a servicios externos mediante REST
- Visualizar un mapa de google maps con una posición determinada
- Conexión e integración con la red social Facebook.

Cabe destacar que los conocimientos técnicos del tutor son de gran aporte y sirven como guía a la hora de afrontar este tipo de problemas técnicos.

### 2.1 Tecnología Involucrada

#### 2.1.1 SpringAndroid

Para la llamadas a servicios Web REST desde el dispositivo Android se utilizo un framework de Spring llamado SpringAndroid que facilita la llamada y parseo de los mensaje devueltos por el servicio.

Abajo se muestra una simple llamada a un servicio con una determinada URL que devuelve el resultado como String.

```
RestTemplate restTemplate = new RestTemplate(new  
HttpComponentsClientHttpRequestFactory());  
String result = restTemplate.getForObject(url, String.class);
```



## 2.1.2 RestEasy

Para realizar y publicar los servicios Rest del lado del servidor de aplicaciones se utilizó el framework RestEasy de Jboss que se integra fácilmente a Jboss Application Server que es el servidor elegido por el equipo.

A continuación se muestra como definir el servicio con annotation de una forma fácil e intuitiva.

```
@GET
@Path("/login/{email}/{pass}")
@Produces("application/json")
public UsuarioDTO login(@PathParam("email") String email, @PathParam("pass")
String pass)
```

Esta firma indica que este servicio acepta solicitudes GET, la URL en la cual estará publicado y que la salida de método será un usuario con formato JSON que es un formato ligero para el intercambio de información parecido al XML pero más liviano y fácil de leer. Cabe aclarar que la seguridad no fue un punto a considerar en el proyecto.

Desde el cliente móvil la llamada a este servicio y el parseo de la respuesta se realiza de una forma muy sencilla.

```
Map<String, String> parms = new HashMap<String, String>();
parms.put("email", email);
parms.put("pass", pass);
UsuarioDTO user = restTemp.getForObject(
Constantes.LOGIN_SERVICE_URL, UsuarioDTO.class, parms);
```

De esta forma se está llamando al servicio antes descrito pasándole el email y contraseña para el login en el sistema. El parámetro del UsuarioDTO.class está diciendo que el retorno será de este tipo y el parseo lo realizará SpringAndroid de forma transparente.

## 2.1.3 Facebook API

Facebook proporciona una librería Open Source, para la integración desde Android con los servicios que este publica. Dicha librería, permite entre otras cosas, publicar información, sea un texto o una imagen, en el muro de facebook de forma sencilla. Este es uno de los requerimientos que se plantearon como importantes en el proyecto, por lo tanto es de prioridad su realización.

Las siguientes líneas permiten la autenticación en facebook para realizar las llamadas a los distintos servicios que ofrece.

EL APP\_ID es un código de autenticación proporcionado por facebook al registrar la aplicación.



```
FaceBook facebook = new Facebook(APP_ID);  
facebook.authorize(this, getResources().getStringArray(R.array.permissions),  
new FaceBookDialog());
```

El siguiente código, permite publicar un mensaje en el muro del usuario autenticado.

```
Bundle parameters = new Bundle();  
parameters.putString("message", msg);  
String response = facebook.request("me/feed", parameters, "POST");
```

También se puede publicar una foto de esta manera:

```
Bundle parameters = new Bundle();  
parameters.putByteArray("picture", byte[] foto);  
String response = facebook.request("me/photos", parameters, "POST");
```

A continuación se muestra un fragmento de código el cual permite obtener los amigos del usuario en Facebook. Estos son devueltos en formato JSON y luego son parseados para desplegarlos en el dispositivo móvil. Los datos devueltos son, el identificador del usuario en Facebook, el nombre y la foto de perfil.

```
String respuesta = InvitarAmigosActivity.facebook.request("me/friends", params);  
JSONArray = new JSONObject(respuesta).getJSONArray("data");  
AmigoFacebook amigo;  
if (JSONArray.length() > 0) {  
    for (int i = 0; i < JSONArray.length(); i++) {  
        JSONObject jo = JSONArray.getJSONObject(i);  
        amigo = new AmigoFacebook();  
        amigo.setId(jo.getString("id"));  
        amigo.setNombre(jo.getString("name"));  
        amigo.setFotoUrl(jo.getString("picture"));  
        amigo.setSeleccionado(false);  
        list.add(amigo);  
    }  
}
```

Además de las funcionalidades anteriores el equipo evaluó e investigó la posibilidad de implementar el login de la aplicación vía Facebook, es decir, utilizar el usuario de Facebook en vez de crear uno propio de Gsiam. Esto es muy utilizado actualmente por muchas aplicaciones, para que el usuario no tenga que perder tiempo al crear una nueva cuenta en otro sistema y tener que recordar varias contraseñas.

El equipo prefirió en esta etapa del proyecto tener más control sobre la aplicación y no depender de ningún servicio externo que pudiera dejar el sistema fuera de servicio. Sin embargo, se entiende que esta es una funcionalidad importante y se agregará a los trabajos futuros.



### 3 Conclusión

Como se menciono anteriormente, la factibilidad económica no fue considerada por tratarse de un proyecto académico y Open Source.

Con respecto a la factibilidad técnica, las tres problemáticas planteadas fueron resueltas de manera exitosa recogiendo los siguientes resultados:

- La comunicación entre, el servidor de aplicaciones donde serán publicados los servicios webs y la aplicación Android corriendo en un dispositivo móvil.
- Visualización de un mapa con una marca en un punto geográfico determinado.
- Publicación de mensajes e imágenes en Facebook así como también el acceso a la lista de amigos de esta red.

En base a estos resultados, el equipo cree que técnicamente es factible la realización del proyecto. Se puede concluir que los conocimientos adquiridos en esta etapa del proyecto serán de vital importancia para la correcta ejecución del mismo, así como también el código generado para esta prueba servirá como base para el comienzo de la implementación de Gsiam.