FACHADASAPP

17 de Octubre de 2014

Descripción de la metodología de trabajo

Oscar Pulido Rodríguez

Diego Martínez Botero

Tabla de Contenidos

1. Introducción 3

1.1 Propósito de este documento 3

1.2 Alcance 3

2. Descripción Del proceso 4

2.1 En qué se ha trabajado 4

2.2 Avances 4

3. Requerimientos 5

3.1 Codigo fuente 5

3.2 Librerias utilizadas 5

3.3 Disponibilidad 6

4. Comparativo 7

4.1 Planeación vs Avance 7

5. Problemas Presentados 8

2.1 Problemas encontrados 8

2.2 Atraso 8

6. Planeacion 9

2.1 ¿Qué falta? 9

2.2 Plan para continuar y completar el proyecto 9

# Introducción

Este documento describe el proceso y la planificación de trabajo proyectado en la metodología Scrum para la clase de Ingeniería de Software II, segundo semestre de 2014, para la gestión del desarrollo del proyecto FachadasApp.

## Propósito de este documento

Facilitar la información de referencia necesaria a las personas implicadas en el desarrollo del sistema FachadasApp.

## Alcance

Personas y procedimientos implicados en el desarrollo del sistema FachadasApp.

# Descripción Del Proceso

Se tiene una app que permite enviar y recibir datos, para la finalización total hace falta solo el manipular la foto que se toma en la app.

## En qué se ha trabajado

El proyecto para dispositivos Android, ha sido implementado en un sistema operativo Windows 8.1 utilizando Java y el SDK de Android. Ha sido desarrollado en Eclipse versión 23.0.2.1259578, utilizando el SDK de Android revisión 23.0.2.

El servidor ha sido implementado en un sistema operativo Linux Fedora y se ha trabajado en python, utilizando un servicio RESTful y una base de datos no relacional MongoDB.

## Avances

Se ha trabajado las piezas por separado como servicios independientes y luego se han armado para completar su funcionalidad. Las funcionalidades por separado del sistema que actualmente se han logrado son:

* El sistema es capaz de conectarse al servidor y enviarle la información para ser guardada en la base de datos.
* El sistema es capaz de solicitar la información al servidor, recibirla y mostrarla en la aplicación.
* El sistema es capaz de organizar la información y mostrarla en una lista, con nombre y descripción, acorde al diseño planteado para la aplicación.
* El sistema es capaz de permitirle al usuario tomar una foto de una fachada y agregarle su nombre y descripción y almacenar la información para enviarla al servidor.

Por aparte se ha trabajado el servidor, el cual responde acorde a las peticiones de la aplicación y es capaz de:

* Recibir la información, convertirla a formato Json y guardarla en la base de datos de la manera acordada.
* Recibir la petición de una fachada y devolver la información almacenada en la base de datos de esa fachada específica.

# Requerimientos

* Dispositivo Android:
  + Api nivel 14 o superior (4.0).
  + Acceso a internet.
  + Cámara fotográfica.
  + Memoria externa.
* Servidor:
  + Debe tener instalado Python
  + Debe tener instalado Bottle Framework para la conexión Api REST.
  + PyMongol para la conexión con la base de datos MongoDB

# Código fuente

El código fuente de la aplicación se encuentra disponible en Github en el repositorio de nombre FachadasApp.

<https://github.com/diegomtzb/FachadasApp>

https://github.com/oscarpr/FachadasAPP

## Librerías utilizadas

import java.io.BufferedReader;  
import java.io.IOException;  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.util.ArrayList;  
import java.util.List;

import org.apache.http.HttpEntity;  
import org.apache.http.HttpResponse;  
import org.apache.http.NameValuePair;  
import org.apache.http.client.HttpClient;  
import org.apache.http.client.entity.UrlEncodedFormEntity;  
import org.apache.http.client.methods.HttpGet;  
import org.apache.http.client.methods.HttpPost;  
import org.apache.http.entity.BufferedHttpEntity;  
import org.apache.http.impl.client.DefaultHttpClient;  
import org.apache.http.message.BasicNameValuePair;  
import org.apache.http.util.EntityUtils;

import android.os.Bundle;  
import android.os.StrictMode;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;

import java.net.HttpURLConnection;  
import java.net.URL;

## Disponibilidad

El código en Github es de acceso público y puede ser descargado y compilado utilizando Eclipse y Android SDK o utilizando Android Studio.

# Comparativo

Le ha logrado avanzar hasta el punto de llegar casi a la totalidad de los objetivos trazados inicialmente.

## Planeación Vs Avance

En nuestra planeación del proyecto acordamos que las semanas que transcurrimos, serían de desarrollo verificación y corrección en un ciclo continuo. Sin embargo para esta entrega, esperábamos contar con las funcionalidades que hemos logrado hasta el momento pero funcionando de manera conjunta y no por separado como lo hace hasta ahora. En el segundo avance aun no funcionaban las divisiones que se le hizo a la manera de trabajar, ya en este periodo de tiempo esas partes funcionan perfectamente, quedando faltando el proceso de mostrar y subir fotografías.

# Problemas presentados

Se presentaron varios problemas al transcurrir del semestre, ejemplo de esto fue inicialmente el ordenador de uno de los integrantes, no podía ejecutar ningún código de Eclipse, luego con el paso, otro integrante tuvo retrasos por compromisos externos a la universidad, como lo fue una capacitación en desarrollo en Android, pero el principal problema fueron los compromisos internos con la universidad de ambos integrantes del grupo puesto que otras materias, demandaron tiempo que se estaba estipulado dedicar a este Software.

## Problemas encontrados

* Problemas con los computadores utilizados: Uno de los computadores utilizados siempre presentó problemas para la ejecución del entorno de desarrollo (Eclipse y Android Studio). Lo cual incurrió en un trabajo más lento y muchas veces unilateral en el desarrollo de la aplicación.
* Falta de Drivers y conocimiento: Uno de los principales atrasos era la falta de capacidad para probar la aplicación en el celular, ya que el pc en el que se estaba desarrollando no reconocía el celular. Esto se solucionó instalando los drivers USB de Android que no se encontraban instalados en este pc y no nos habíamos percatado de la falla.
* Descuido y falta de permisos: Otro atraso importante que nos retrasó el desarrollo de la aplicación consistió en realizar la conexión con el servidor desde la aplicación, ya que en el módulo individual creado inicialmente funcionaba, y al momento de conectarlo con el sistema conjunto no. Este problema consistió en el descuido al momento de crear el nuevo proyecto, de no implementar los permisos necesarios en el archivo manifesto de la aplicación, para conectarse a internet.
* Descuido y falta de tiempo: Por supuesto en nuestra vida diaria el tiempo siempre es un riesgo inminente en el desarrollo de cualquier proyecto y mucho más aún si no se administra y organiza debidamente. En nuestro caso nos ha faltado un poco de organización para trabajar y más aún cuando sabemos que los dos desarrolladores del proyecto contamos con diversas actividades y horarios disponibles. Nada que no se pueda solucionar con un poquito más de organización y disposición.

## Atraso

A pesar de que el atraso no es mucho a nuestro parecer, pues aun creemos que podemos concluir el proyecto en el plazo estipulado y que contamos con casi todo lo estipulado para esta entrega, esto no se puede apreciar de una manera evidente. Sin embargo los problemas encontrados han sido la causa de este atraso.

# Planeación

En esta etapa del proyecto nos hemos replanteado la organización y hemos reestructurado el plan para poder cumplir con la entrega completa del proyecto en el tiempo estimado.

## ¿Qué falta?

* Hacer log in como usuario.
* Registrarse como usuario del sistema.
* Subir y mostrar fotos.

## Plan para continuar y completar el proyecto

Para poder concluir satisfactoriamente el proyecto es necesario principalmente mejorar la organización del tiempo. Para esto hemos decidido cumplir con las reuniones diarias estrictamente, dedicándole entre 5 y 15 minutos a la planeación del día y de la semana.

Con esto pretenderemos lograr una pequeña meta o corrección diaria mínimo y no dejar que se acumulen para el final de la semana.

De esta misma manera pretendemos gestionar mejor el tiempo de trabajo y avanzar mucho más rápido en lo que hace falta para completar el producto mínimo viable y luego poder proceder con los siguientes aspectos que hagan falta.