

Samplers: Framework para construir aplicaciones
Android para recolectar muestras en proyectos de
Ciencia Ciudadana

Laura Lus y Javier Ramírez

17 de noviembre de 2017

Resumen

Algunos proyectos de investigación requieren la recolección de muestras en el campo de estudio. AppEAR y CazaMosquitos son ejemplos de aplicaciones Android que permiten recolectar muestras de campo para investigaciones de monitoreo de estuarios y poblaciones de especies de mosquitos respectivamente. La recolección de muestras consiste en que los usuarios de la aplicación respondan preguntas, saquen fotos y envíen datos de geolocalización, entre otros como pueden ser fecha y hora de la muestra o comentarios que quieran hacer.

Samplers es un framework para generar aplicaciones para Android que ayuden a recolectar muestras en proyectos de investigación que incluyan Ciencia Ciudadana. Samplers es una herramienta que permite, definiendo un protocolo de toma de la muestra, recolectar

Índice general

1. Introducción	2
1.1. Algunas palabras sobre este documento	3
2. Trabajo Relacionado	5
3. Estrategia general	6
4. Capítulo específico	8
5. Evaluación	9
6. Conclusiones y trabajo futuro	10

Capítulo 1

Introducción

La introducción responde cinco preguntas:

¿en que tema están trabajando? Introduce en contexto en el que están trabajando (p.e., el ámbito en el que se dá el problema). Da definiciones (brevemente) de conceptos que aparecen en el contexto.

¿cual es el problema? ¿es interesante? Presenta el problema que se quiso resolver (o la necesidad que se quiso cubrir), y explica por que es relevante y no es trivial resolverlo. Introduce definiciones (brevemente) que ayuden a entender el problema. Esta parte (en conjunto con la anterior) dan la motivación del trabajo.

¿cual es la estrategia general de solución? Presenta en términos generales la estrategia de solución (p.e., cuenta que se desarrolló un sistema que permite... y que por su característica... resuelve el problema planteado.

¿cuales son las contribuciones del trabajo? Enumera que es lo que obtiene quien lee el trabajo. Ejemplos de contribuciones son: un análisis detallado del problema X en el contexto Y, una comparación de tecnologías de servidores ZZZ, una revisión de literatura del estado actual del tema J, una arquitectura y diseño de un sistema que permite..., un prototipo que muestra como la arquitectura Z puede implementarse de forma B, evidencia experimental que muestra que es posible WW, un framework/libreria web para GGG, un algoritmo eficiente para ...

¿Cómo esta organizado el trabajo? El ultimo párrafo de la Introducción explica como esta organizado el documento; como se lee, que es lo que se va a encontrar en cada capítulo y como se relacionan.

Dependiendo de lo larga que quede puede tener varias secciones. No estaría mal que esas secciones se llamen “Motivación”¹ (que habla del contexto y el problema), “Enfoque” (que explica como lo van a hacer), “Contribuciones” y “Organización”.

Muchos dicen que la intro se puede escribir al final. De hecho, las contribuciones y la organización es algo que está claro solo al terminar el proyecto de tesis. Sin embargo, creo que tener un borrador corto de la intro (un párrafo respondiendo a cada una de las preguntas) es bueno para utilizarlo como referencia cada vez que nos olvidamos que es lo que queríamos hacer.

1.1. Algunas palabras sobre este documento

Mi idea al escribir este documento es que sirva como punto de partida para organizar la tesis (el proyecto y el documento) y como punto de partida para editar el documento en LaTeX. Hay muchas referencias en la web con guías similares. Todas tienen algo bueno. La página web de Willilam Shoaff Shoaff [1] habla sobre como organizar una tesis de maestría (lo cual creo que es aplicable a la tesis de Licenciatura). Si se imaginan un futuro escribiendo artículos técnicos y están con tiempo para leer, les recomiendo el libro de Justin Zobel “Writing for Computers Science”[2] que está en biblioteca de Informática de UNLP. En la Figura 1.1 se pude ver un extracto de libro en el que discute el poco valor que tienen palabras como ”muy.”^{en} la escritura técnica.

¹prestá atención a como se ponen las comillas en LaTeX

Qualifiers such as “very” and “quite” should be avoided altogether, because they are in effect meaningless. If an algorithm is “very fast”, is an algorithm that is merely “fast” deficient in some way? Writing is invariably more forceful without “very”.

× There is very little advantage to the networked approach.

✓ There is little advantage to the networked approach.

Likewise, “simply” can often be deleted.

× The standard method is simply too slow.

✓ The standard method is too slow.

Other words of this kind are “totally”, “completely”, “truly”, “highly”, “usually”, “accordingly”, “certainly”, “necessarily”, and “somewhat”.

Figura 1.1: Extracto del libro Writing for Computers Science de Justin Zobel

Capítulo 2

Trabajo Relacionado

Hacer una tesis implica encontrar una pregunta que valga la pena responder o un problema que valga la pena resolver y darle respuesta o solución. Es una tarea de investigación que tiene como aspecto muy importante conocer lo que ya existe alrededor de la pregunta o problema que se elige.

Al llegar a este capítulo, el lector tiene una idea de cual es el problema. Seguramente se imagina problemas similares o soluciones al problema. El objetivo, en este momento, es convencerlo de que conocemos el problema y otros similares; que conocemos las formas en las que se lo ha intentado resolver (o a problemas similares); y que aún después de saber todo eso sigue siendo un problema importante, difícil y que nadie resolvió o nadie resolvió tan bien como nosotros).

Para escribir este capítulo hay que leer. Hay que buscar soluciones a problemas similares y compararlas con lo que nosotros queremos hacer. Si sabemos que la nuestra es mejor, ya podemos marcar cuales son los puntos débiles de las existentes. También se puede escribir un poco sobre otras investigaciones, que si bien no atacaron problemas parecidos, pueden ser aplicadas a resolver parte de este.

Este capítulo es bueno ir escribiéndolo en borrador cada vez que se lee algo (un artículo por ejemplo). Por lo menos hay que escribir un resumen de un párrafo de lo leído (registrando la referencia en el archivo bibliografía.bib y citando desde acá), y dar nuestra opinión al respecto en términos de su relación con el problema de nuestra tesis.

Capítulo 3

Estrategia general

Un tipo de tesis común en Sistemas es la que propone una solución a un problema ¹. Puede ser que el problema todavía no haya sido resuelto (poco probable); o puede ser que se proponga una solución que es mejor a las existentes en algún aspecto. Una forma interesante de imaginar el documento de tesis es como un espiral, que da cuatro vueltas, de adentro para afuera. En cada vuelta da mas detalles.

- Vuelta 1 (el resumen): se cuenta toda la tesis (problema, estrategia de solución, resultado obtenido) en un solo párrafo.
- Vuelta 2 (la introducción): En la introducción, se vuelve a contar el problema (ahora se introduce el contexto, se explica por que es un problema relevante y difícil, se dan algunas definiciones), se adelanta cual es la estrategia de solución aunque todavía no se puede explicar mucho, se listan las contribuciones principales.
- Vuelta 3 (varios capítulos): Ahora se puede dedicar un capítulo completo a contar bien cual es la estrategia general (que método se aplica, que arquitectura, que tecnologías, que pasos tiene la solución, etc), y se puede dedicar un capítulo completo a cada parte interesante de la solución (esto depende mucho de lo que resuelvas y que partes importantes tenga).

El capítulo de estrategia general tiene como objetivo contar cual es la estrategia/método de solución al problema elegido. Por ejemplo, ¿se propone

¹hay otros tipos, por ejemplo aquellas que demuestran experimentalmente alguna cualidad de algún fenómeno

una metodología? ¿que pasos tiene? ¿Se construye un sistema? ¿que arquitectura tiene? ¿que partes importantes tiene? ¿que funcionalidad provee?

Con este capítulo le debería alcanzar al que lee para entender como se resolvió el problema. Los capítulos que siguen a este pueden dar mas detalle sobre aquellos aspectos/partes que valga la pena detallar. De alguna forma, este capitulo es el mapa que ordena los capítulos que siguen.

Capítulo 4

Capítulo específico

Al capítulo 3, que describe la estrategia general, lo siguen varios capítulos que entran en detalle en distintas partes de la solución. Uno, por ejemplo, puede describir el modelo de datos que utiliza la aplicación, otro puede describir el front-end de la aplicación, otro puede describir el algoritmo de recomendación de nuevos contenidos, etc. Por lo general hay que explicar en detalle aquellas cosas que no son obvias para quien no hizo la tesis y que las necesitaría si quiere reproducir lo que ustedes hicieron.

No es necesario escribir mucho. Simplemente es cuestión de preguntarse que le podría resultar interesante o novedoso a un compañero de estudio que no conoce el tema.

Capítulo 5

Evaluación

Supongamos que se quiso atacar el problema de la dificultad en el desarrollo de aplicaciones móviles multi-plataforma. Y que lo que se hizo fue desarrollar una librería de clases. ¿Cómo demuestro que la librería de clases resuelve el problema?

Antes que nada, deberíamos haber dejado claro, en alguna sección del capítulo 1 cuales son los indicadores que miro para decir que hay "dificultad en el desarrollo de aplicaciones móviles". Por ejemplo, ¿cantidad de bugs específicos de la plataforma? ¿tiempo que lleva traducir los aspectos específicos?. Conocer esos indicadores (o aspectos) es importante para decidir a cuales de ellos voy a apuntar en mi solución (porque tal vez no puedo ser mejor en todos los aspectos). Es importante para poder comparar mi solución con otras. Y es importante porque en este capítulo tengo que demostrar que mi solución es mejor en términos del/los aspectos elegidos.

La evaluación se puede hacer de muchas formas y depende del caso en particular. Por ejemplo, podrías poner a varios compañeros a hacer la misma aplicación demo con tu librería y otras que ellos quieran. Y luego les hacés preguntas para saber si con tu librería fué mejor. O podés contar la cantidad de bugs que se hicieron usando tu librería vs los que se hicieron sin ella. Hacer un experimento es complejo, pero hay muchas alternativas intermedias para que puedas demostrar que tu propuesta resuelve el problema planteado.

Capítulo 6

Conclusiones y trabajo futuro

En el apartado de conclusiones se hace un breve resumen de lo que uno quiso hacer y lo que pudo hacer. Se sacan conclusiones respecto a la efectividad de la estrategia aplicada en resolver el problema. Se sacan conclusiones sobre la complejidad del problema y su importancia. Hasta se puede hablar de alternativas que probamos y no funcionaron. Esta sección es nuestra oportunidad para recordarle al lector que era un problema difícil y que pudimos resolverlo. Y de paso le recordamos la lista de contribuciones.

Antes de terminar, hay que dejar algunas líneas para quien quiera continuar investigando el tema (trabajo futuro). ¿Qué hubiese sido bueno hacer y no se hizo porque no hubo tiempo? ¿Qué no funciona del todo bien y se podría mejorar? ¿Qué otras alternativas de solución se te ocurren ahora que podrían ser mejores?

Bibliografía

- [1] William D. Shoaff. How to Write a Master's Thesis in Computer Science.
- [2] Justin Zobel. *Writing for computer science*. Springer-Verlag, second edition, 2004.