
Cicchinelli Fernández Joaquín Augusto
Laboratorio de Desarrollo de Software, 2016
Universidad Nacional de la Patagonia Austral
Unidad Académica Caleta Olivia

**Aplicación móvil para el Sistema de Votación
Electrónica Universitaria**

Informe final

Tabla de Contenidos

1. Resumen	2
2. Introducción	2
3. Objetivos	2
4. Dificultades encontradas	3
5. Resultado final	4
6. Lecciones aprendidas	6
7. Comentarios y conclusiones finales	8

1. Resumen

Los aspectos relevantes de este documento se centran en señalar las variables que incidieron en pro o en contra del desarrollo del proyecto, así como también las metodologías y actividades utilizadas provenientes de las distintas áreas de estudio y asignaturas previas de la carrera.

En una primera instancia se desarrollará el propósito del proyecto y los objetivos ligados a su realización. Luego, las dificultades que surgieron al inicio y durante el periodo de trabajo. Finalmente un balance de los resultados obtenidos, lecciones aprendidas y comentarios

2. Introducción

El proyecto surgió como una extensión de otro sistema implementado por un alumno del año anterior sumado a la iniciativa de querer desarrollar una aplicación móvil cuya función principal fuera el poder llevar a cabo los procesos eleccionarios de consejeros de la universidad utilizando un dispositivo móvil cualquiera (extendiendo el alcance del sistema base original que se implementa en una plataforma web). Debido a esto, este sistema móvil no debe tomarse como un sistema aislado, sino que es parte de un sistema distribuido más complejo, y que requiere de su contraparte Web para poder realizar todas sus funciones correctamente.

Adicionalmente este proyecto tuvo como finalidad secundaria el poder aprender a trabajar con un nuevo lenguaje de programación (Android).

3. Objetivos

- Llevar a la práctica los conceptos teóricos vistos en clase y aplicando aquellos necesarios provenientes de otras disciplinas estudiadas años anteriores (ingeniería de requerimientos, manejo de bases de datos, modelados/diagramas, análisis y diseño, planificación, etc.).
- Aprender a programar en un lenguaje nuevo (Android).
- Generar código siguiendo pautas y buenas prácticas propias del lenguaje.
- Producir una aplicación móvil funcional.
- Desarrollar una estrategia de trabajo para la implementación del sistema.
- Adoptar, a lo largo del proyecto, los distintos roles que se ven involucrados (líder de requerimientos, de diseño, de implementación, etc.).
- Generar la documentación propia del desarrollo de un proyecto de software (plan de pruebas, documento de especificación de requerimientos, manual de usuario, especificación suplementaria, modelo de datos, etc.).

4. Dificultades encontradas

Durante el desarrollo se presentaron algunos percances que afectaron negativamente al proyecto, sin embargo, no fueron lo suficientemente graves como para no poder resolverlos. La constante motivación y apoyo por parte de los involucrados (docentes y compañeros) fue una de las fortalezas más significativas. Uno de los pasos más difíciles de realizar fue el análisis y realización de la documentación relacionada a la estrategia de seguimiento del proyecto ya que, al no tener una base conceptual ni de referencia del desarrollo de aplicaciones móviles en Android, se dificultó el poder proyectar una propuesta de plan de proyecto, por lo que se decidió hacer énfasis en la formulación de la estrategia e implementación del trabajo.

La principal dificultad vino dada por el hecho de querer trabajar con un lenguaje totalmente nuevo y que, además, uno en el que ninguno de los docentes de la carrera se había especializado. Esto hizo que las interrogantes que iban surgiendo a lo largo del proceso de desarrollo fuesen difíciles de solucionar en periodos cortos de tiempo.

En lugar de poder acudir rápidamente a un profesor para salvar las dudas, debí apoyarme en la documentación oficial del lenguaje, tutoriales escritos o audiovisuales por internet, y bibliografía relacionada. Esto hacía del proceso algo mucho más lento que lo normal, ya que no solo debía invertir tiempo en el proceso de desarrollo del sistema, sino que también debía dedicar una gran carga horaria a aprender “cómo” producir el código que necesitaba y respetando pautas/prácticas propias de la programación en Android.

Debido a lo anterior tuvimos que acordar con los docentes concentrarnos solo en la implementación del sistema y dejar en un segundo plano la producción de la documentación propia del proceso de desarrollo de software.

Un inconveniente menor fue el hecho de que mi computadora tenía la potencia de hardware justa para ejecutar el entorno de programación Android Studio, y al momento de utilizar el simulador (virtualmente se emula un dispositivo móvil para poder probar la aplicación sin necesidad de depender de un dispositivo físico) para realizar las pruebas pertinentes, la computadora se ralentizaba demasiado o dejaba de responder.

La comunicación entre una aplicación Android y una Base de Datos SQL no es posible de manera directa, por lo que además de aprender a programar en Android y generar la aplicación, debí aprender sobre el funcionamiento de los Web Service y programar uno para que hiciera de nexo entre la BD en el servidor y la Aplicación en el dispositivo cliente.

5. Resultado final

El resultado final puede dividirse y ser analizado en dos puntos cronológicos ya que al concluir el periodo de cursado de la asignatura continué trabajando en mejorar el sistema y completar la documentación:

- Al final de la cursada: la mayoría de los documentos quedaron relegados a un segundo plano ya que se priorizó el aprendizaje del lenguaje y el desarrollo de la aplicación. Se generó un prototipo funcional que cumplía con el 90% de las funcionalidades estipuladas; aún así, se requerían algunas pequeñas iteraciones más sobre el proceso de desarrollo para completar algunos módulos aislados y realizar las pruebas pertinentes, como así también generar y completar la documentación faltante.
- Al momento del examen final: en este punto la aplicación se encuentra en un estado final, cumpliendo el 100% de las funciones estipuladas, habiéndose realizado las pruebas pertinentes que verifiquen y validen el comportamiento del sistema. Además se produjo toda la documentación que debía acompañar al producto final:
 - Documento de especificación de requerimientos: provee definiciones, alcance, suposiciones y restricciones del sistema. Especifica las funciones que deberá cumplir y ayudar a comprender al cliente los beneficios que obtendrá del mismo.
 - Especificación suplementaria: capturar los requerimientos del sistema que no se reflejan en el documento *Especificación de Requerimientos de Software* ni en el documento *Modelo de Casos de Uso*. Estos requerimientos refieren a los atributos de calidad que el sistema debe cumplir. Entre ellos se destacan: usabilidad, performance y soporte. También se incluyen otros relacionados con la plataforma, sistemas operativos, compatibilidad e interoperabilidad.
 - Modelo de casos de uso: provee una descripción escrita y detallada del comportamiento del sistema al afrontar una tarea (cada tarea corresponde a un requerimiento funcional). Otorga al cliente una vista acerca de la funcionalidad del sistema y la manera en la que se interactúa con el mismo. Incluye los actores y una descripción de la secuencia de pasos a seguir para utilizar las funcionalidades del sistema.
 - Modelo de datos: describe las estructuras de datos de la base de datos que aloja la información utilizada y generada por el sistema, así como las relaciones que existen entre ellas.

- Plan de proyecto: engloba la información relativa a la gestión del proyecto. Comunicar el ámbito y recursos al personal de desarrollo y al cliente. Definir los riesgos y sugerir técnicas de control. Definir los costes y planificación. Describir cómo se garantizará la calidad y se gestionarán los cambios.
- Casos de prueba: tiene como finalidad entregar las pautas y definir la estrategia que se seguirá para llevar a cabo la certificación del software. El objetivo general es establecer las condiciones para la aplicación de las pruebas de manera de obtener un sistema que cumpla con la totalidad de las funcionalidades requeridas.
- Requerimientos funcionales e interfaces de usuario: brinda tanto al cliente como al usuario, capturas de pantalla del sistema relacionadas con las funcionalidades del mismo. Permitiendo tener una idea de cómo están hechas las pantallas (interfaces de usuario), los elementos que hay en cada una, colores, tipografía, funcionalidades, formas de interactuar y su conexión con los requerimientos funcionales.
- Manual de usuario: explica cómo hacer uso de todas las funcionalidades del sistema por parte del usuario. Además de brindar información extra sobre características, datos, funciones realizadas en segundo plano, etc.

La entrega final de los artefactos producidos durante el proyecto se realizará de la siguiente manera:

- Dos (2) copias del presente “Informe Final”, cada una acompañada de un CD conteniendo el resto de la documentación en formato digital y la aplicación en formato “.apk” para su instalación en dispositivos móviles.
- Un (1) juego de copias de toda la documentación producida, acompañada de un (1) CD conteniendo la misma documentación en formato digital y la aplicación en formato “.apk” para su instalación en dispositivos móviles.

6. Lecciones aprendidas

Durante la realización del proyecto apliqué conceptos y metodologías vistos a lo largo de toda la carrera:

- Requerimientos de software: aplicar ingeniería de requerimientos y técnicas para la recolección de datos e identificación de requerimientos del sistema, pautas para la construcción del documento de especificación de requerimientos (DER), definición de casos de uso, etc.
- Diseño de bases de datos: si bien la base de datos ya había sido construida por el equipo web, para poder desarrollar la aplicación móvil y que ésta manipule la información de manera correcta fue estrictamente necesario comprender como estaba distribuida y relacionada la información contenida en la base de datos. Conceptos de algebra relacional, SQL, diagrama de clases UML, modelo entidad-relación fueron conceptos esenciales.
- Análisis y diseño de software: complementando los conceptos de requerimientos de software (casos de uso, modelos estáticos y dinámicos, roles, actores), aplicación de patrones.
- Programación orientada a objetos: conceptos que se vieron en esta asignatura y que resultaron ser muy relevantes en la programación Android, como lo son: objeto, alcance, visibilidad, encapsulamiento, modularización, herencia, polimorfismo, entre otros.
- Laboratorio de programación: fue la primera asignatura donde me enfrenté al desarrollo de una aplicación (pero en este caso, una aplicación web). Para éste proyecto en particular, me sirvió todo lo aprendido sobre el lenguaje PHP, con el que programé el Web Service para comunicar la aplicación Android en el cliente móvil, con la base de datos en el servidor.
- Validación y verificación de software: conceptos y técnicas para asegurar que el software cumple con sus especificaciones y satisface las expectativas de los usuarios. Técnicas de testing (estructural y funcional) y documentación.
- Gestión de proyectos de software: estrategias y técnicas de desarrollo, habilidades de gestión, errores clásicos, trabajo de alta calidad, definición de objetivos, alcance y tiempo de desarrollo, etc.

Aprendí lo básico de la programación en el lenguaje Android. Semántica y sintáctica del lenguaje, estructuración del código, buenas prácticas y el manejo del entorno de desarrollo integrado específico (Android Studio).

Simulación completa de un proceso de desarrollo de software, además de experimentar los distintos roles (programador, QA, tester, líder de proyecto, etc). Mi experiencia cumpliendo estos roles puede resumirse de la siguiente manera:

- Gerente de proyecto: fui responsable de la planificación del proyecto y de mantenerlo dentro de los plazos establecidos. Comunicar al cliente el progreso del proyecto mediante entregables parciales (prototipos con funcionalidades limitadas) en fechas previamente acordadas. Una observación es que, durante el desarrollo del proyecto, las docentes a cargo de la asignatura tomaron el rol

de cliente para definir los requerimientos funcionales y no funcionales que el sistema debería satisfacer.

- Líder de planificación y seguimiento: controlar el avance del proyecto y la realización de las fases del proyecto según estándares acordados. Establecer un plan de trabajo y su cumplimiento.
- Líder de requerimientos: recabar información del cliente y plasmarla en un documento que sirva de nexo entre él y el equipo de desarrollo (Documento de especificación de requerimientos). Definir y validar los requerimientos.
- Líder de aseguramiento de calidad: aplicar técnicas y prácticas que aseguren la calidad del software.
- Líder de diseño: diseñar las interfaces de usuario y una arquitectura interna limpia y simple. La definición de la base datos relacional del sistema fue realizada por el equipo de desarrollo encargado de la implementación del entorno web (S.V.E.U).
- Líder de implementación/Ingeniero de desarrollo: convertir la especificación del sistema en código fuente ejecutable.
- Ingeniero de control de calidad: diseñar pruebas que permitan validar requerimientos, eliminar diferentes tipos de errores y requiriendo la mínima cantidad de esfuerzo y tiempo.

7. Comentarios y conclusiones finales

A lo largo del proyecto surgieron dudas, problemas, contratiempos que me hicieron dudar de si iba a poder si quiera aprender a programar en el lenguaje. A pesar de todo, el proyecto fue avanzando a pasos pequeños pero constantes y sin darme cuenta tuve en mis manos la aplicación funcionando perfectamente en distintos dispositivos (celulares de distintos tamaños e incluso en una tablet).

La base de datos producida por el equipo de desarrollo de la versión web poseía algunos errores, pero por motivos de simplicidad se decidió trabajar sobre ella sin modificarla ya que, si se le realizaban cambios, toda la parte web del sistema iba a quedar obsoleta y a perder la mayoría de sus funcionalidades, y el tener que revisar todo el código del entorno web excedía el alcance de mi proyecto, además de que hubiese ampliado aún más la complejidad, demanda y tiempos de desarrollo que ya de por si nos resultaban ajustados debido a incursionar con una nueva tecnología sin previos conocimientos. Sería recomendable que ambos equipos (web y móvil) trabajaran en conjunto para pulir estas imperfecciones manteniendo ambas plataformas funcionales y compatibles para así dar como resultado un sistema completo y de alta calidad.

Respecto al documento “Modelo de casos de uso”, se decidió eliminar una de sus secciones (la que especifica cada caso de uso). Una mejora a futuro de la documentación podría ser reincorporar esta sección y completarla.

Finalmente, una aclaración que hay que hacer en cuanto al entregable final (la aplicación en formato “.apk” lista para instalarse) es la siguiente:

- Como esto es un sistema distribuido, una parte del sistema se encuentra del lado del cliente (la aplicación móvil) y otra del lado del servidor (la base de datos y el web service).
- Para que la aplicación se comunice con el web service se necesita configurarla con una IP (que debe ser estática) hacia donde la aplicación va a enviar todas sus peticiones (envío y solicitud de información).
- Como yo no disponía de un servidor propio, utilice mi computadora para emular el servidor, pero esto tiene una contra: la IP de mi computadora cambiaba todo el tiempo, porque era asignada por la red, y cada vez que me conectaba-desconectaba se me asignaba una IP distinta.
- Esto me obligaba a que para poder probar la comunicación entre la aplicación y la computadora emulando el servidor, tuviese que editar en Android Studio la dirección IP que la aplicación iba a utilizar para contactar al servidor, compilar una nueva versión y reinstalarla en el dispositivo móvil de prueba.
- Esto obviamente hace imposible que pueda dejarse un prototipo entregable 100% funcional, ya que hasta que no se disponga de un servidor dedicado con una IP estática, siempre será necesario modificar una parte del código de la aplicación para que sepa hacia donde debe enviar sus solicitudes. O forzar a la computadora servidor a adoptar la IP que posee configurada la aplicación.