

Informe Proyecto Final de Carrera



UTN Móvil

Aplicación móvil como canal de información en tiempo real para la Facultad Regional Santa Fe de la Universidad Tecnológica Nacional.

Alumnos

**Gutiérrez, Héctor Tomás
Paciuk, Sebastián Federico**

Directora del Proyecto

Dra. Gutiérrez, María de los Milagros

Año 2016

Índice

Capítulo 1: Introducción	2
Justificación	2
Problemática detectada	4
Alcance del proyecto	5
• Propuestas similares	5
• Objetivo general	7
• Objetivos específicos	7
Capítulo 2: Metodología	8
Introducción	8
Planificación	9
• Recursos humanos	9
• Estimación	9
Ejecución	10
• Reporte primera iteración	10
• Reporte segunda iteración	12
• Reporte tercera iteración	16
Conclusión sobre la metodología aplicada	17
Capítulo 3: Tecnologías e infraestructura de desarrollo	18
Tecnologías Utilizadas	18
• MongoDB	18
• NodeJS	19
• Google Cloud Messaging	20
• HTML, CSS y JavaScript	21
• Bootstrap	22
Infraestructura de Desarrollo	22
• Android Studio	22
• Atom	23
• Robomongo	23
• Git & GitHub	23
Capítulo 4: Análisis y diseño de la solución	25
Arquitectura	25
• Patrón Modelo-Vista-Adaptador (MVA)	26
Usuarios Identificados	26
Implementación	27
• Semántica y esquemas de datos	27
• Estructura de archivos del Back-end	30
• Interfaz pública y privada del servidor	31
• Despliegue del servidor	31
• Estructura de la aplicación móvil	32
• Soporte de múltiples densidades de pantalla	33
Requisitos No Funcionales	36
• a) Seguridad	36
• b) Movilidad	37
• c) Usabilidad	39
• d) Mantenibilidad	39
• e) Cuestiones legales	40
Capítulo 5: Documentación de uso	43
Usuario creador de contenidos (web)	43
Usuario que consulta contenidos (app)	48
Capítulo 6: Conclusiones y trabajos futuros	68
Bibliografía	69
Anexos	71
Diseño de pantallas (mockups)	71
Interfaz pública del servidor	78
Interfaz privada del servidor	79
Detalles de la planificación y plazos reales	82

Capítulo 1: Introducción

Justificación

Hoy en día, las tecnologías de la información y las comunicaciones (TICs) brindan la posibilidad de acceder a la información de forma instantánea. En los últimos tiempos se ha extendido el uso de smartphones, al punto de convertirlos en herramientas imprescindibles para las actividades de la vida diaria adquiriendo una gran cantidad de funcionalidades, hasta incluso se puede decir que sus usos son infinitos. Esto es así debido a que en los smartphones es posible instalar aplicaciones frecuentemente llamadas *apps* que agregan funcionalidades a estos dispositivos, aumentando notoriamente su utilidad. Así es posible usar el smartphone para recibir/enviar correo electrónico, escuchar radio, hacer videoconferencias, conectarse a un canal de televisión, jugar videojuegos, enviar mensajería instantánea, acceder a redes sociales y un inmenso conjunto de finalidades que dependerá de la personalización que cada usuario realice en su dispositivo en particular. Las aplicaciones móviles brindan uno de los mejores canales de comunicación dirigido a un público específico que existen en la actualidad. Así se comienzan a desarrollar aplicaciones dirigidas a un público en particular. Un ejemplo de ello es la aplicación *¿Dónde Curso?* [1] que fue diseñada específicamente para los alumnos de la Facultad Regional Santa Fe de la Universidad Tecnológica Nacional. Ésta aplicación les brinda información sobre el horario y aula a la que tienen que dirigirse los alumnos de acuerdo a cada materia que deben cursar. Como se aprecia, esta aplicación tiene sentido sólo para estudiantes, no así para el público en general. Otra aplicación interesante es *Cuando Pasa?* [2] que está orientada para brindar información de la hora en que pasan las distintas líneas de colectivos en las diferentes paradas de la ciudad de Santa Fe. Ésta aplicación está orientada a los usuarios de transporte público de pasajeros.

También es posible que los usuarios pueden acceder a otros contenidos a través de los sitios Web para móviles, y si bien hacerlo por este medio no requiere de ninguna descarga o instalación extra, las aplicaciones móviles tienen las ventajas de que:

- Son más fáciles de usar.
- Se ejecutan más rápido que los sitios web para móviles.
- Acceso completo a todas las características del dispositivo (sensores, cámaras, sistema de archivos, entre otros).
- Permiten mantener almacenada cierta información incluso si no se cuenta en el momento con una conexión a internet.
- Poseen interfaces de usuario más agradables, ya que permiten gráficos y efectos más innovadores.
- Le brindan la posibilidad al usuario de acceder de una manera rápida a sus preferencias para poder personalizar la aplicación a su gusto.
- Tienen mayor visibilidad en las tiendas de aplicaciones. [3]

En el contexto universitario, existe gran cantidad de información que los alumnos y docentes pueden necesitar a la hora de realizar sus actividades con el fin de hacerlas más eficientes.

Un ejemplo de esto es conocer el aula donde se dicta una determinada materia. Esta información necesita estar constantemente actualizada dado que puede cambiar por distintas circunstancias. Contar con este dato, hace que tanto profesores como alumnos no pierdan tiempo tratando de encontrar el lugar adecuado para llevar a cabo sus actividades. Sin embargo existe más información que podría ser de utilidad a los miembros de la comunidad universitaria pero no es fácilmente accesible, esto se debe a que la misma sólo existe distribuida a través de los diferentes sistemas actuales, pero la comunicación entre ellos es escasa y para consultarla es necesario accederlos individualmente. Dado que las universidades conforman un ámbito en el que hay mucha información disponible y también muchos interesados en la misma, se hace indispensable hoy en día contar con aplicaciones móviles que la procesen, agreguen y la pongan a disposición de los usuarios.

El presente proyecto tiene como principal beneficiario a la Universidad Tecnológica Nacional Facultad Regional Santa Fe pero a los fines prácticos del mismo serán consideradas su Subsecretaría de Tecnologías de la Información y Comunicaciones y su Área de Comunicación como el “cliente” en el sentido estricto de la metodología aplicada. En la actualidad la facultad posee variados sistemas generadores de contenido a los cuáles los alumnos pueden acceder y consultar información útil para el desempeño de sus actividades académicas. Algunos ejemplos de esto son el *Campus Virtual* y el sistema de distribución de aulas.

Actualmente también existe la aplicación *¿Dónde Curso?* desarrollada por la Subsecretaría de Tecnologías de la Información y Comunicaciones, pero sus funciones están acotadas a consultar el aula designada para el cursado de una o más materias según un filtro de búsqueda especificado en un día en particular. A continuación se muestra la interfaz de la aplicación en la figura 1.

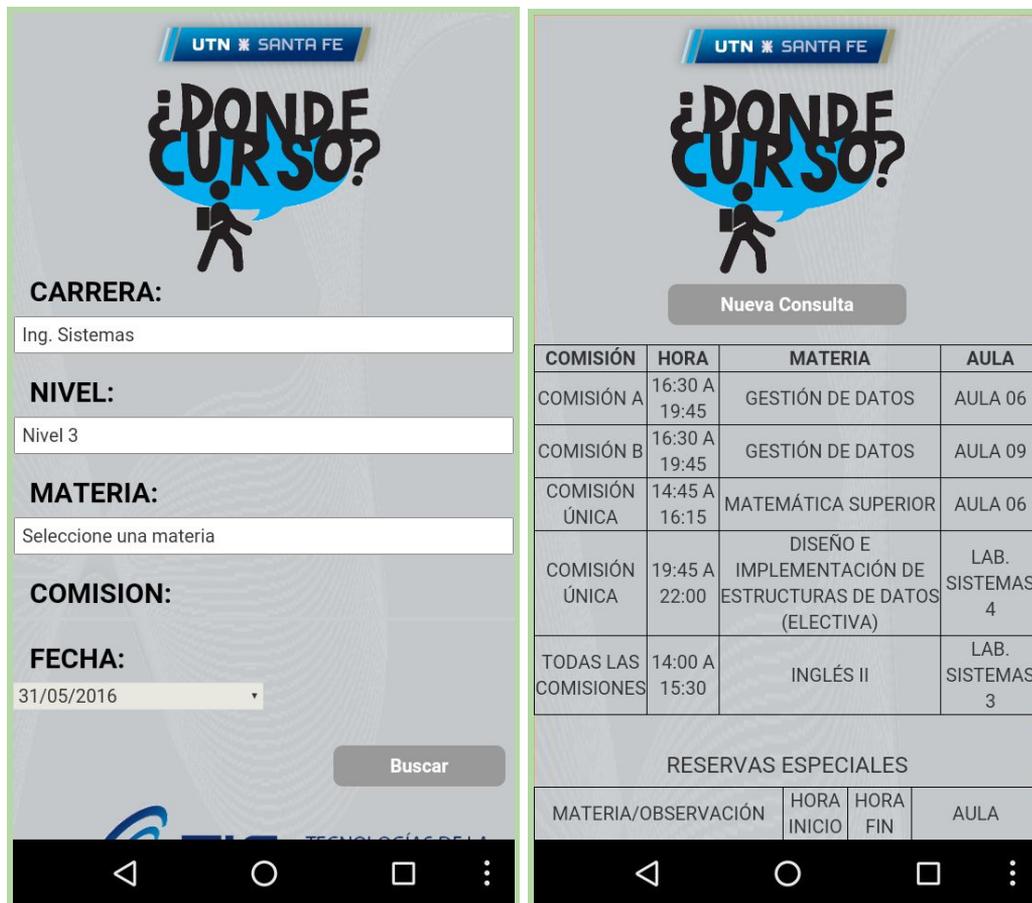


Figura 1. Interfaz de la aplicación móvil “¿Dónde Curso?”.

Para que estos sistemas puedan comunicarse y compartir información, es posible implementar aplicaciones livianas conocidas como *apps* que sean capaces de consumir información. Luego en su contraparte es necesario que se agrupen los datos e implementen los servicios correspondientes para poder intercambiarla.

Dentro de la terminología informática, esta arquitectura que promueve la separación define a sus componentes como cliente (*front-end*) y servidor (*back-end*) respectivamente [4].

Problemática detectada

Desde el Área de Comunicación surgió la necesidad de contar con un medio de comunicación exclusivamente apuntado a dispositivos móviles. La solución que se propuso fué ampliar la aplicación *¿Dónde Curso?* (figura 1), incorporando nuevas funcionalidades.

La ampliación que se propuso consiste en que los usuarios de la aplicación puedan recibir a través de la misma información pertinente de la facultad, como puede ser noticias, eventos, cursos y búsqueda laboral.

Otra preocupación existente radica en la necesidad de emitir avisos de último momento con carácter de emergencia hacia toda la comunidad educativa. Algunos ejemplos provistos por el Área de Comunicación fueron: Suspensión de actividades por desastres climáticos,

duelos y otras decisiones imprevistas de último momento. Su frecuencia es relativamente baja (entre 1 y 3 veces al año) pero son de alta importancia institucional.

La idea es aprovechar la tendencia y los beneficios de las comunicaciones móviles para alcanzar en forma más directa a los usuarios a través de notificaciones que alerten sobre los cambios anteriormente mencionados.

Alcance del proyecto

El presente trabajo aborda el diseño e implementación de una aplicación móvil que cuente con un conjunto inicial de funcionalidades y que pueda ser fácilmente extensible.

Estas funcionalidades iniciales son:

- Consultar los horarios y la asignación de aulas para el cursado de las materias (esto se encuentra actualmente desarrollado en la aplicación *¿Dónde Curso?*).
- La recepción de noticias y eventos provenientes del Área de Comunicación de la facultad.
- Proveer un calendario dinámico que muestre los horarios de cursado para cada alumno y los eventos importantes a acontecer en la facultad.

También se diseñará e implementará paralelamente un sistema web para el personal de la facultad responsable de actualizar el contenido informativo del sistema.

Propuestas similares

Se investigaron referencias de aplicaciones móviles similares para analizar algunos de sus módulos, ya que en la actualidad existen universidades de Norteamérica y Europa que hacen uso de Aplicaciones Móviles como canal de información con su comunidad educativa. Por dar algunos ejemplos se pueden nombrar las aplicaciones móviles del M.I.T (Massachusetts Institute of Technology) (figura 2.a) disponible en mobi.mit.edu y la de University of Notre Dame (figura 2.b) disponible en mobile.nd.edu/app.

Estas aplicaciones ofrecen variadas funcionalidades para sus usuarios, como consultar noticias, eventos, información relacionada con el transporte, directorio de personas, mapa del campus universitario, comedores cercanos disponibles, servicios de impresión, entre otras.

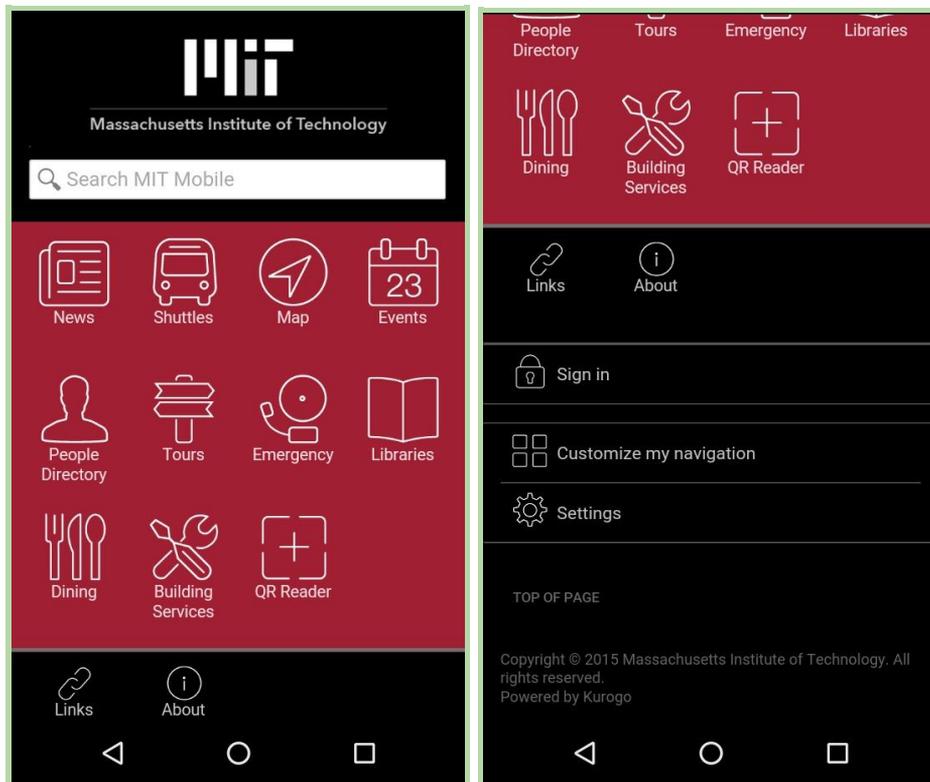


Figura 2.a. MIT Mobile.

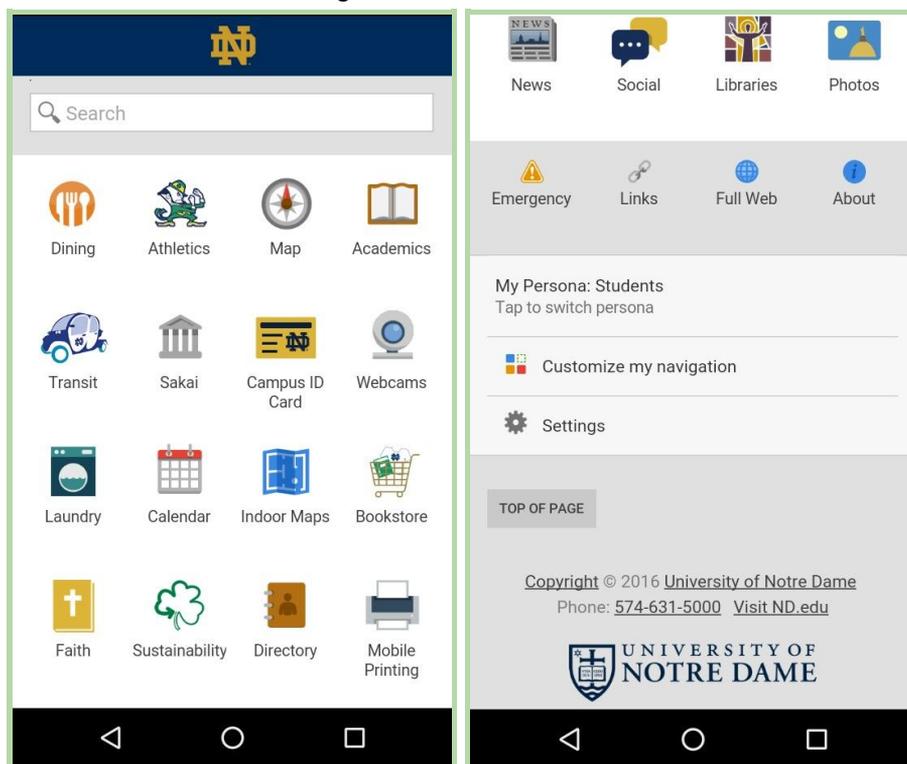


Figura 2.b. Notre Dame Mobile.

Objetivo general

El objetivo del proyecto es diseñar e implementar una infraestructura digital (framework) con énfasis en su aplicación móvil que permita registrar y consultar información acerca de las actividades académicas, institucionales y de interés general que se realizan en la Facultad Regional Santa Fe de la Universidad Tecnológica Nacional.

Objetivos específicos

- Análisis, diseño e implementación de una aplicación móvil Android.
- Análisis, diseño e implementación de un servidor de back-end para dar soporte a la aplicación móvil.
- Análisis, diseño e implementación de una base de datos.
- Definición de un modelo semántico de los datos para favorecer la interoperabilidad con otros sistemas.
- Diseño de una arquitectura que permita escalabilidad para implementar fácilmente futuras funcionalidades y soportar gran cantidad de usuarios.
- Desarrollo del módulo de noticias en tiempo real.
- Desarrollo de un módulo para notificación de eventos y su organización en el calendario.
- Integración con el sistema existente *¿Dónde Curso?*, con información sobre horarios y aulas donde se dictan las materias.
- Desarrollo de un sistema web que cuente con un mecanismo de autenticación segura para la generación de contenidos.

De ahora en adelante, el informe se encuentra estructurado de la siguiente manera: en el capítulo 2 “Metodología” se procederá a explicar los recursos humanos y tareas que se siguieron para llevar a cabo la ejecución del trabajo. En el capítulo 3 “Tecnologías e infraestructura de desarrollo” se exponen las tecnologías que se usaron y sus características, resaltando aquellas que son especialmente útiles al proyecto. En el capítulo 4 “Análisis y diseño de la solución” se muestra un conjunto de modelos y se justifican las decisiones de diseño tomadas para alcanzar los objetivos propuestos. Se describen cuestiones como la arquitectura elegida, los usuarios identificados y requisitos no funcionales. En el capítulo 5 “Documentación de uso” se registra cómo los usuarios deben hacer uso tanto la aplicación móvil como del cliente web. Al final, en el capítulo 6 “Conclusiones y trabajos futuros” se exponen los resultados obtenidos y las opiniones que surgieron al finalizar el presente trabajo.

Capítulo 2: Metodología

Introducción

Para el presente proyecto se utilizó un modelo de ciclo de vida basado en prototipos, el cual es apropiado cuando el cliente tiene en mente objetivos generales pero no identifica requerimientos detallados y características. En cada entrega se muestra un subconjunto de funcionalidades para luego obtener retroalimentación de las partes interesadas. Posteriormente se procede a realizar la siguiente iteración agregando otro subconjunto de funcionalidades y mejorando las anteriores con la nueva información disponible. Este ciclo se repite hasta tener completos los objetivos generales y la conformidad de las partes. [5]

En este proyecto, cada iteración comprende las etapas mostradas en la figura 3.



Figura 3. Etapas de cada iteración.

Además, se identificaron 3 subsistemas que fueron desarrollados de forma concurrente y que se completan al final de las iteraciones, estos son:

- **Back-end**
- **Aplicación Móvil**
- **Módulos de Contenido**

En la próxima sección se detallan las actividades que fueron desarrolladas en cada iteración.

Planificación

Recursos humanos

Se dispone de dos recursos humanos: Gutiérrez, Héctor Tomás y Paciuk, Sebastián Federico que desempeñarán las tareas de análisis, diseño, codificación y pruebas. La disponibilidad horaria son 25 horas semanales por cada uno.

La directora del proyecto es la Dra. Milagros Gutiérrez.

Además se encuentran el personal y encargados de la Subsecretaría de Tecnologías de la Información y Comunicaciones (TICs) de la facultad quienes se han comprometido a colaborar con la integración de esta aplicación con sus sistemas existentes.

También se deben mencionar al personal y encargados del Área de Comunicación de la facultad quienes harán uso del sistema web para la generación de contenido. También darán su opinión y aprobación de los aspectos gráficos del proyecto para que se corresponda con la imagen institucional. A fines prácticos de la metodología utilizada, serán considerados como clientes la Subsecretaría de TICs y el Área de Comunicación de la Universidad Tecnológica Nacional Facultad regional Santa Fe.

Estimación

Se realizó una estimación del esfuerzo en la planificación inicial, teniendo en cuenta las tareas de más alto nivel y las metas a alcanzar en cada iteración. Basado en esto, se estimaron un total de 1000 (mil) horas de trabajo, distribuidas de la forma que indica la figura 4. Para profundizar sobre esto se recomienda consultar el anexo: Detalles de la planificación.

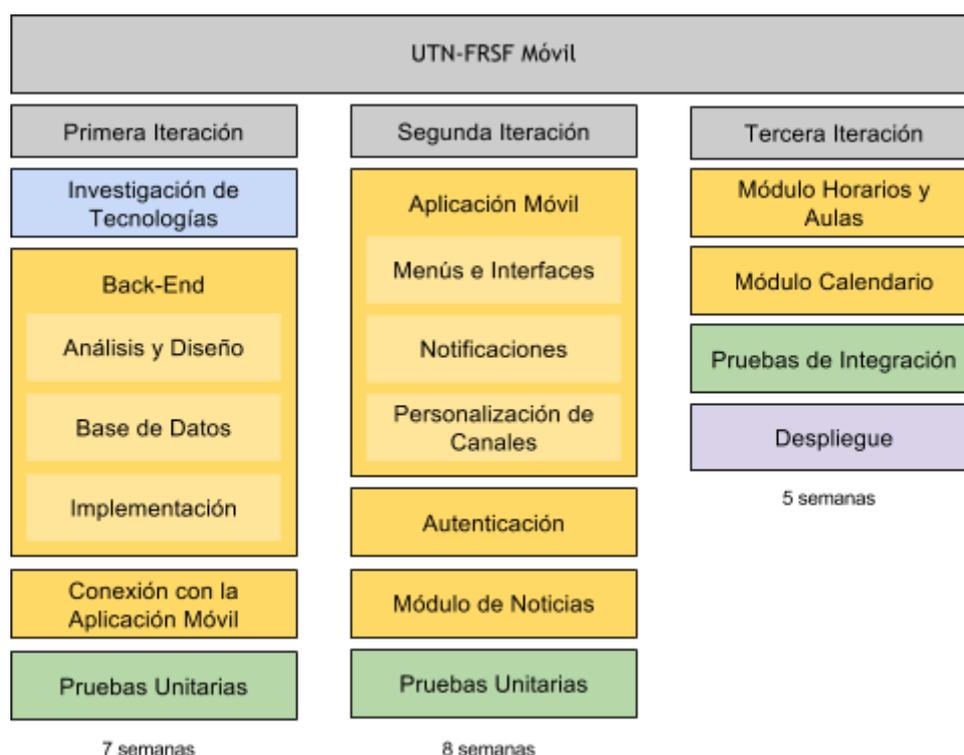


Figura 4. Estimación del esfuerzo por iteración.

Ejecución

Reporte primera iteración

En la tabla 1 se muestra un resumen y el estado de las tareas desarrolladas en la primera iteración.

Tabla 1: Tareas desarrolladas en la iteración 1.

Iteración # 1			
Tareas Desarrolladas			
Nro	Nombre	Estado	Resumen
1	Investigación de tecnologías	Completo	Se realizó la investigación en el uso de las tecnologías Android, Node.js y MongoDB.
2	Instalación y configuración del motor de base de datos	Completo	Se puso en funcionamiento el motor de base de datos (MongoDB), se crearon los usuarios con sus respectivos permisos y se instalaron los programas de gestión necesarios.
3	Instalación del entorno Android Studio (Java)	Completo	Se instaló y configuró Android Studio y su kit de desarrollo (SDK) más actualizado. Se incluyeron algunas librerías y dependencias comúnmente usadas.
4	Instalación del entorno Atom (JavaScript)	Completo	Se logró su completa instalación y configuración, además se incluyó un módulo de sugerencias y autocompletado (ternjs) para facilitar el desarrollo.
5	Definición de la arquitectura	Completo	Se definió el modelo de Arquitectura para el desarrollo del proyecto y se realizó una explicación del mismo.
6	Diseño e implementación de la base de datos	Completo	Creación de algunos documentos de prueba. Definición de un esquema de datos tentativo.
7	Diseño e implementación del servidor de back-end	Completo	Despliegue del servidor en un ambiente local, se implementaron algunos servicios básicos.
8	Diseño e implementación de la estructura de la aplicación móvil	Completo	Se comenzó con una estructura básica pero es factible que haya un reordenamiento en la próxima iteración a medida que crezca el número de clases.

9	Pruebas de la aplicación en dispositivos reales	Completo	Se logró desplegar una aplicación de prueba y conectarla para ser depurada en el entorno.
10	Conexión básica del back-end con la aplicación móvil	Completo	Se logró mediante el protocolo HTTP, desde la aplicación móvil usando la librería Volley. El estándar de intercambio de datos es JSON.
11	Diseño de interfaces para la aplicación móvil	Completo	Consistió en la creación de un documento con el diseño preliminar de las pantallas principales y sus funcionalidades. A cada una se le adjuntó una descripción para ayudar a su comprensión.
12	Implementación de cliente HTML para cargar noticias	Completo	Un cliente estático que es servido desde el back-end para facilitar la carga de noticias a las personas que no son desarrolladores.

Riesgos materializados

Falta de conocimientos sobre las tecnologías a utilizar.

Se invirtió gran cantidad de tiempo en investigar a fondo sobre tecnologías, protocolos y herramientas. Se utilizaron varias fuentes disponibles desde libros hasta videos y foros. Fueron llevadas a cabo reuniones para compartir conocimiento y hacer pruebas técnicas.

Conclusión de iteración

Esta primera iteración tuvo un gran componente de horas invertidas en investigación de tecnologías, tendencias en el mercado de aplicaciones, prácticas comunes en el desarrollo móvil y puntualmente las guías de diseño de Android en sus versiones más modernas.

Se instalaron los distintos entornos de desarrollo y se configuraron para adaptarlos a nuestras necesidades.

Se buscó apuntar primero a los riesgos técnicos más importantes y se logró la compatibilidad con todas las tecnologías involucradas.

Reunión con el cliente

Se llevó a cabo el día 6 de Junio de 2016 con los representantes de la Subsecretaría de Tecnologías de la Información y Comunicaciones, Walter Justet, Juan Carlos Ramos, Marcelo Romitti y Nicolás Mizerniuk. Se hizo una presentación de los avances logrados en cuanto al desarrollo de la aplicación móvil y el back-end. Luego los temas que se discutieron fueron:

- Qué se debería mostrar en el Calendario. Este se pensó en un inicio para mostrar las fechas de los eventos académicos de la facultad (por ej. charlas) pero se decidió que se podría agregar al mismo la información del calendario académico y además los días y horarios de cursado de las materias a las que se encuentra suscrito el alumno.

- Cuáles serían las fuentes de noticias para la aplicación. Actualmente la facultad cuenta con varios puntos que proporcionan información relevante a la comunidad, como puede ser El Emilio, el Campus Virtual y el TV-Hall. El Emilio y el TV-Hall se consideraron factibles de vincular con la Aplicación a través del Área de Comunicación, centralizando la emisión de noticias desde allí. En cuanto al campus, se planteó el inconveniente de no poder vincular directamente los cursos del mismo con las materias de la facultad, ya que en el campus los cursos son por año de cursado y comisión (por ej. Análisis Matemático I 2016 - Turno Noche) y no hay una correspondencia directa con las materias de la facultad. Este tema se descartó de implementar en un principio.
- También se planteó la necesidad de definir un formato para las noticias que tiene que recibir la Aplicación y mostrar a los usuarios. Este tema se definirá después con el Área de Comunicación.
- Se estableció que todos los diseños de pantallas y gráficos que se usen tienen que ser validados por el Área de Comunicación, y si ellos lo requieren pueden proponer sus propios diseños.
- Por último se charló sobre la posibilidad de implementar un mecanismo para que los profesores puedan enviar noticias a través de la aplicación, es decir, en vez de emitirlas a través del campus y que luego se estas se envíen a la aplicación, que la aplicación tenga la funcionalidad de escribir una noticia y enviarla a un público específico. Pero para esto se debería implementar un mecanismo de autenticación conectado a una base de datos de usuarios para definir quién puede enviar noticias y hacia quién las puede emitir. Dado que no se cuenta con esta estructura de permisos la idea se dejó para un futuro.

Reporte segunda iteración

En la tabla 2 se muestra el estado de las tareas desarrolladas en la segunda iteración.

Tabla 2: Tareas de la iteración 2.

Iteración # 2			
Tareas Desarrolladas			
Nro	Nombre	Estado	Resumen
1	Definición del modelo semántico de datos	Completo	Generación del documento "Semántica y esquemas de datos"
2	Refinamiento del diseño del servidor del back-end y base de datos	Completo	Se lo adaptó para implementar Express. Se separó la lógica de la interfaz con la aplicación de la web de carga de datos.

3	Diseño de los menús e interfaces básicas de la aplicación móvil	Completo	Generación del documento “Diseño de pantallas” con las principales
4	Implementación de los menús e interfaces básicas de la aplicación móvil	Completo	Terminadas el menú principal, la sección noticias, personalización de canales y el calendario pero esta última sin conectividad con el servidor todavía.
5	Autenticación segura para la generación de contenidos	Completo	Se implementó el mecanismo de seguridad Basic Auth sobre HTTP temporalmente. Después se migró a JSON Web Token sobre HTTPS que es más seguro.
6	Desarrollo del primer módulo de noticias	Completo	Se puso en funcionamiento paralelamente con la pantalla de selección de canales.
7	Desarrollo de las notificaciones en tiempo real	Completo	Implementación y configuración de Google Cloud Messaging, tanto en la aplicación como en el back-end.
8	Desarrollo de la sección Acerca de	Completo	Necesaria por motivos legales. Incluye las licencias de las librerías utilizadas hasta el momento y los créditos por el desarrollo. Evitamos posibles problemas futuros al momento de la distribución de la aplicación.
9	Prueba y retroalimentación	Completo	Se hicieron las pruebas y cargaron los problemas encontrados en la sección issues de los repositorios de GitHub.

Riesgos materializados

Dificultades en la implementación de actualización de información en tiempo real iniciada por el servidor hacia el cliente.

Como producto de la investigación se encontró una solución intermedia que facilita la distribución de mensajes hacia los dispositivos. Google Cloud Messaging permite el envío de hasta 4kb desde el servidor de back-end hacia grupos de clientes con una interfaz relativamente sencilla de implementar. Se detalla su funcionamiento en la sección Tecnologías Utilizadas.

Incidentes no contemplados

Google Cloud Messaging comenzó a discontinuar su funcionamiento, migró a Firebase Cloud Messaging y actualizó sus librerías.

Se tuvo que adaptar tanto el cliente como el servidor a la nueva librería con sus respectivos métodos para prolongar su compatibilidad en un futuro. Otros beneficios de implementar

Firestore Cloud Messaging son la recolección de datos anónimos de los dispositivos, es información útil para analizar posteriormente como versiones de sistema operativo y modelos.

Conclusión de la iteración

Se tuvo que rehacer parte del código varias veces, tanto por parte del cliente como del servidor. Esto principalmente porque se cambió el enfoque en algunas cuestiones como la seguridad y la compatibilidad con una mayor cantidad de dispositivos.

Se invirtió un alto porcentaje de tiempo de la iteración en el desarrollo del módulo de noticias, pero dejó valioso conocimiento y algoritmos para aplicar en los módulos siguientes.

Reuniones con los clientes

Se llevó a cabo el día 29 de Julio. El objetivo de la misma fue mostrar la aplicación para que el personal del Área de Comunicación, Lic. Mariano Bravi y Juan de Rossi, dieran su opinión sobre el diseño de la misma. Las devoluciones al respecto fueron las siguientes:

- En primer lugar mostraron conformidad con el diseño general de la aplicación, ya que esta era simple y hacía uso de los componentes nativos de Android. Expusieron que el hecho de proponer un rediseño iba a prolongar los tiempos del proyecto y no iba a aportar mejoras significativas.
- Sugirieron que se use para el fondo de la aplicación el degradado institucional que se utiliza en diferentes medios informativos de la facultad (El Emilio, afiches, etc). Y también pusieron a disposición los diseños del logo institucional para que este se visualice en la barra superior de la sección principal.
- Además ampliaron información sobre cuáles son los medios por los que actualmente se emite información referida a la facultad (El Emilio, pagina web, TV-Hall, revista La Araña) y de qué manera cargan la información en los mismos, para tener en cuenta esto y considerar la posibilidad de integrar estas plataformas con la aplicación móvil. La posibilidad de realizar esta integración fue postergada hasta la próxima reunión, para ser discutida con el personal de la Subsecretaría de TICs.
- Sugirieron incluir botones con links a las redes sociales oficiales que son mantenidas por ellos. En un principio la página de Facebook y la cuenta de Twitter. También la página web oficial.
- Por último, pidieron que se pueda enviar a la aplicación un tipo de noticia denominada “último momento” ó “urgente”, para comunicar información muy importante en el caso de que ocurra alguna catástrofe, o se suspendan las actividades en la facultad por algún motivo. Este tipo de notificación se tiene que diferenciar del resto (por ej: otro color) y no puede ser silenciada bajo ningún motivo, la misma tiene que ser recibida por todos los usuarios de la aplicación sin necesidad de haberse suscrito a algún canal en especial.

Otra reunión se llevó a cabo el día 4 de Agosto con los representantes de la Subsecretaría de Tecnologías de Información y Comunicaciones (TICs), Walter Justet, Juan Carlos Ramos, Alejandro Tassoni, Marcelo Romitti y Nicolás Mizerniuk. Se hizo una presentación de la aplicación con los últimos avances. Los temas tratados fueron los siguientes:

- La posibilidad de unificar los sistemas creadores de contenidos de la facultad con la interfaz propuesta en este proyecto. Actualmente el Área de Comunicación cuenta con un sistema donde cargan noticias y eventos que son los que se muestran cuando se ingresa a la página web de la facultad (www.frsf.utn.edu.ar). Se propuso inicialmente que la carga de los contenidos que se van a mostrar en la aplicación se haga en una interfaz web desarrollada en este proyecto para tal fin. Para que la carga de contenidos se haga sólo una vez, se planteó integrar los dos sistemas de cargas de contenidos. En el caso de unificar los sistemas, se dejó en claro que el sistema que quede tendría que tener una opción para elegir si el contenido que se está creando se publica en la aplicación, en la página de la facultad o en ambas. No se dió una respuesta al mismo, se evaluaron posibilidades, pero el tema queda a definir por el personal de la Subsecretaría de TICs.
- Se evaluó la posibilidad de incorporar el contenido que se muestra en la página de la facultad en las secciones de “Cursos” (www.frsf.utn.edu.ar/cursos) y “Búsquedas laborales” (www.frsf.utn.edu.ar/graduados/busquedas-laborales) a la aplicación móvil. Se encontró una forma de hacerlo dentro de los tiempos planificados pero con la intención de integrar mejor estas funcionalidades con su fuentes de información en un futuro.
- Se identificaron los usuarios potenciales de la aplicación, especialmente se destacan aquellos que deben tener permisos especiales para publicar noticias y eventos. Dado el carácter institucional de la aplicación, se consideró que el Área de Comunicación tendrá este rol, es decir, si alguien de alguna área de la facultad quiere publicar alguna noticia se lo informa al Área de Comunicación, quienes lo expresarán de forma adecuada y lo publicarán para que quede disponible para los usuarios.
- Se estableció como requerimiento la posibilidad de que los usuarios de la aplicación puedan silenciar las notificaciones, salvo las de carácter urgente las cuales estas nunca se deben poder silenciar.
- Se requiere resaltar de alguna forma más intuitiva el acceso al panel lateral que filtra los canales.
- Se discutió sobre qué información hay que mostrar en la sección de “Acerca de” y se sugirieron formas de organizar la misma.
- Necesidad de incluir en la aplicación íconos con links a los perfiles oficiales de la facultad en las diferentes redes sociales (Facebook, Twitter) y la página de la facultad.

Reporte tercera iteración

En la tabla 2 se muestra un resumen y el estado de las tareas desarrolladas en la tercera iteración.

Tabla 3: Tareas de la iteración 3.

Iteración # 3			
Tareas Desarrolladas			
Nro	Nombre	Estado	Resumen
1	Análisis de los resultados de las pruebas y devoluciones de los usuarios	Completo	Detallado en las reuniones con la Subsecretaría de TICs y Área de Comunicación de la facultad.
2	Rediseño según las devoluciones de los usuarios	Completo	Se refinaron los modelos y se cargaron los issues correspondientes en GitHub.
3	Desarrollo del módulo de calendario de actividades	Completo	Desde el cliente se integró la librería de código abierto Caldroid con varias modificaciones. Se implementaron los correspondientes webservices en el back-end.
4	Desarrollo del módulo que conecta con el sistema <i>¿Dónde Curso?</i>	Completo	Se hizo ingeniería inversa de la aplicación <i>¿Dónde Curso?</i> y se adaptó el cliente a las funciones disponibles en el servidor de la facultad con algunas mejoras significativas.
5	Desarrollo del módulo Cursos	Completo	Se adaptaron los encabezados de la consulta y manejo de la respuesta del servidor de la facultad. Se trabajó para mejorar el manejo de imágenes y adaptarlas a los dispositivos móviles.
6	Desarrollo del módulo Búsquedas Laborales	Completo	Se conectó con éxito al servidor de la facultad para recuperar el contenido y se agregó un ícono en el menú principal.
7	Integración con redes sociales	Completo	Se incluyeron botones en el menú principal que tienen vinculación a las redes sociales y página oficial de la facultad.
8	Pruebas de integración	Completo	Se contrató un servidor en la nube basado en linux para poner en funcionamiento el back-end. Se subió una prueba alpha cerrada a Google Play.

9	Despliegue del sistema	Completo	El despliegue final del sistema será realizado por TICs después de su revisión completa por un pedido de ellos. Se abrió la prueba alpha a algunos usuarios.
---	------------------------	----------	--

Riesgos Materializados

Aumento del esfuerzo necesario para llevar a cabo la adaptación de los Módulos de Contenido, ya sea por aumento en la cantidad a utilizar, o por complejidad de los mismos.

Después de las pruebas alpha, se realizó una reunión con la Subsecretaría de TICs y se negoció la integración de la aplicación con la información de “Búsquedas Laborales” y “Cursos”, actualmente disponibles también en la web. Pero se consideró conveniente para darle una mayor visibilidad.

También se incluyeron botones que vinculan al sitio web institucional y las principales redes sociales de la facultad (Facebook y Twitter).

Conclusión de la iteración

Se aprovechó una gran cantidad de conocimiento adquirido de las iteraciones anteriores. Se reutilizó partes del código cuando se presentaban problemas anteriormente resueltos.

Por otra parte aparecieron módulos que no estaban planificados inicialmente, pero se decidió llevarlos a cabo porque compartían estructuras similares con los que ya estaban desarrollados.

Conclusión sobre la metodología aplicada

De lo anterior vimos que el impacto de los riesgos se pudo manejar sin extender los plazos ni sacrificar funcionalidades.

Incluso se pudieron llevar adelante nuevos módulos considerados clave por la Subsecretaría de TICs y el Área de Comunicación.

Las reuniones con el cliente tuvieron lugar después de que cada iteración fue completada, esto permitió mostrar prototipos sólidos con sus nuevas funcionalidades que habían sido puestas a prueba previamente por los desarrolladores. Se buscó formar un grupo de asistentes a las reuniones que realicen tareas tanto técnicas como ejecutivas. A pesar de estar sujetas a disponibilidades horarias de un gran número de participantes solamente una reunión se demoró algunos días más de lo previsto, antes de comenzar la tercera iteración, pero se siguió avanzando normalmente con las tareas hasta el día que pudo llevarse a cabo.

Se documentó lo charlado en las reuniones para luego traducirlos en retroalimentación para los modelos y el desarrollo. Su frecuencia fue adecuada y mejoraron la calidad general de la aplicación móvil y el sistema web.

Teniendo en cuenta los factores mencionados podemos concluir que la metodología aplicada fue apropiada para el proyecto. Aportó la flexibilidad necesaria y mantuvo el orden sin agregar una sobrecarga de trabajo significativa.

Capítulo 3: Tecnologías e infraestructura de desarrollo

Tecnologías utilizadas

MongoDB



Es un sistema de base de datos orientado a documentos bajo el concepto de código abierto. En la actualidad es uno de los más populares y utilizados dentro del paradigma NoSQL. [6] La principal diferencia con las bases de datos relacionales tradicionales, radica en que no almacena los datos en tablas, sino que utiliza estructuras de datos sin esquemas BSON, que a su vez son una codificación binaria de documentos JSON (Notación de Objeto JavaScript), un formato de texto plano y reducido para el rápido intercambio de los datos y su manipulación por parte de las aplicaciones [7].

Se destacan a continuación las características que son útiles al proyecto:

Indexación: Al igual que el enfoque relacional, cualquier campo de un documento puede ser un índice. Sobre los mismos se pueden ejecutar algoritmos de *MapReduce*, que es un modelo de programación distribuida utilizado para procesar grandes lotes de datos realizando operaciones de agregación para acelerar notablemente las consultas más frecuentes [8].

Flexibilidad: Permite a los esquemas cambiar rápidamente a medida que las aplicaciones evolucionan sin sacrificar otras características útiles para los desarrolladores como la consistencia estricta de los datos.

Escalabilidad y tolerancia a fallos: Puede funcionar tanto en una sola instancia de un servidor, como en forma distribuida en arquitecturas complejas. Está diseñado para manejar grandes volúmenes de transacciones. Soporta replicación, donde un servidor secundario copia las acciones del primario. También maneja el concepto de *sharding* donde los datos son divididos por rangos de un campo clave y distribuidos horizontalmente entre las unidades de cómputo con el objetivo de balancear sus cargas dinámicamente.

NodeJS



Node.js es un entorno en tiempo de ejecución asincrónico y orientado a eventos. Diseñado para construir programas de red altamente escalables como servidores web. Es multiplataforma y de código abierto. [9]

Integra V8 que es un motor de alta performance para JavaScript desarrollado por Google.

Características que son útiles al proyecto:

Escalabilidad: A diferencia de otras soluciones similares para la capa de servidor (como implementaciones de PHP), puede manejar gran número de usuarios simultáneamente limitado únicamente por la cantidad de conexiones que soporta el sistema operativo en el que se está ejecutando.

Orientado a eventos: Las consultas de los usuarios se atienden a través de un bucle que se ejecuta en un solo hilo (thread). La figura 5 muestra el ciclo de gestión de eventos. Las operaciones de entrada/salida más lentas son puestas en cola y atendidas, a medida que son completadas la respuesta es enviada al cliente. Como resultado se reduce considerablemente la demanda de cómputo del servidor. Esto es muy eficiente para aplicaciones que tienen alta concurrencia pero no son críticas.

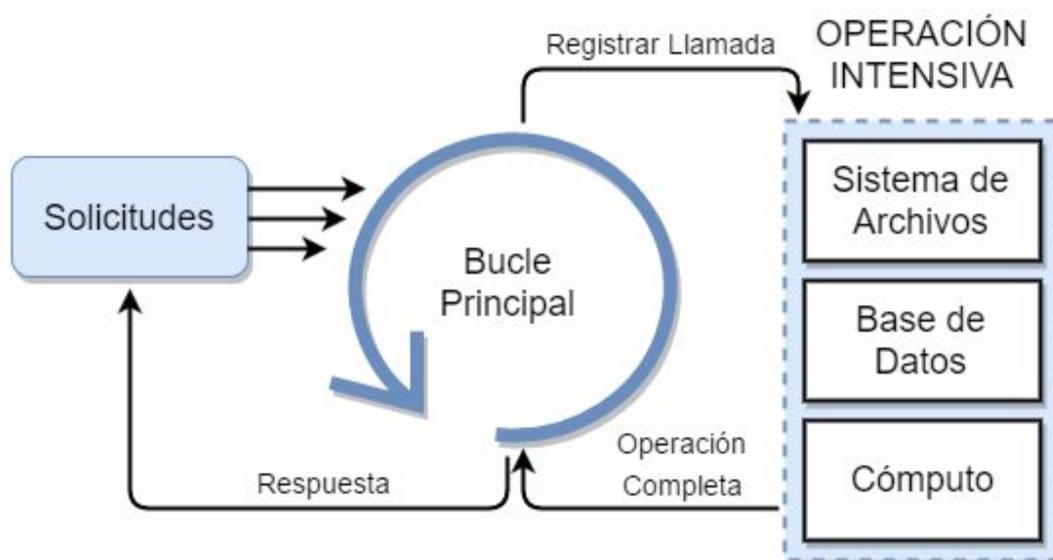


Figura 5. Ciclo de gestión de eventos de NodeJS.

Simplicidad y flexibilidad: JavaScript es un lenguaje fácil de aprender y mantener en proyectos de esta escala. Al ser de código abierto cuenta con una gran comunidad en constante crecimiento que desarrolla y mantiene librerías para la mayoría de las necesidades que tienen los servidores como gestión de clientes, acceso al sistema de archivos y conectores con los principales motores de base de datos incluido *MongoDB*.

Google Cloud Messaging



Comúnmente conocido como GCM, es un servicio de la nube para dispositivos móviles desarrollado por Google que permite a los desarrolladores intercambiar mensajes hacia instancias de sus aplicaciones en tiempo real. [10]

Soluciona de forma transparente la distribución de notificaciones a todos o a grupos de usuarios, incluso los guarda si sus dispositivos se encuentran sin conexión a internet en el momento que se emiten los mensajes para hacer la entrega cuando estén disponibles.

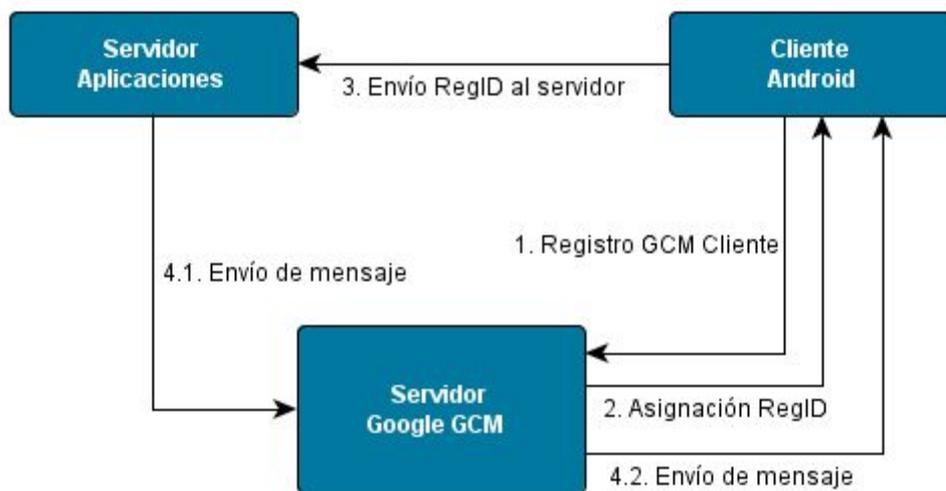


Figura 6. Diagrama del funcionamiento básico de GCM.

Como se observa en la figura 6 la interacción entre el Servidor de aplicaciones, el cliente android y el servidor GCM se describe a continuación:

1. Registro GCM Cliente: Cuando se ejecuta la aplicación, se envía al servidor GCM un identificador único de esa instancia (token).
2. Se le asigna un identificador de registro en el servidor RegID y es retornado a la aplicación móvil.

3. La aplicación móvil envía su RegID al servidor para que la tenga en cuenta para futuros envíos.
4.
 - 4.1. El servidor de aplicaciones envía un mensaje (notificación) al servidor GCM, especificando si está dirigido a todos, a un grupo de usuarios que estén suscritos un determinado tema.
 - 4.2. El servidor GCM distribuye el mensaje a todas las instancias activas de la aplicación móvil o espera a que lo estén para hacer la entrega.

Desde el lado del cliente móvil, si toda la configuración es correcta se pueden realizar las operaciones de suscripción a los canales de interés (identificados por una cadena alfanumérica) y la cancelación de la misma cuando el usuario lo indica. También se vinculan funciones responsables del manejo de los mensajes y cómo se comporta la aplicación cuando estos son entregados.

HTML, CSS y JavaScript



HTML, es la sigla en inglés de HyperText Markup Language (lenguaje de marcas de hipertexto), hace referencia al lenguaje más básico para la creación de páginas web. Es un estándar que se utiliza como referencia para sus diferentes implementaciones y cuenta con varias versiones, siendo la 5ta la más actual. Define una estructura básica y un código para la ubicación del contenido de una página web, como texto, imágenes, videos, entre otros.

Hoja de estilo en cascada o **CSS** (siglas en inglés de Cascading Style Sheets) es un lenguaje usado para definir y crear la presentación de un documento estructurado y escrito en HTML. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. La información de estilos puede ser definida en un documento separado de la lógica con el propósito de ser reutilizada.

JavaScript es un lenguaje de programación interpretado. Se utiliza principalmente en las aplicaciones del lado del cliente, implementado generalmente como parte de un navegador web. Permite mejoras en la interfaz para enriquecer la experiencia del usuario además de permitir escuchar eventos, agregar acciones y ejecutar operaciones lógicas para lograr páginas web más útiles y dinámicas. La mayoría de los navegadores modernos interpretan el lenguaje JavaScript.

Bootstrap



Twitter Bootstrap es un conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web adaptables a cualquier pantalla. Contiene plantillas de diseño con tipografías, formularios, botones, cuadros, menús de navegación y otros elementos basados en HTML y CSS. Además de extensiones y utilidades JavaScript, como funciones de validación y componentes visuales dinámicos para mejorar la experiencia del usuario. [11]

Infraestructura de Desarrollo

Android Studio



Es un entorno de desarrollo integrado (IDE) que provee edición de código, depuración y un gran conjunto de herramientas para construir y desplegar aplicaciones en todas las versiones del sistema operativo Android. [12]

Características que son útiles al proyecto:

- Editor de código inteligente, con funciones de sugerencias y autocompletado.
- *Instant Run*, mecanismo para desplegar cambios en la aplicación rápidamente y hacerlos visibles en el dispositivo o emulador.
- Gráficos vectoriales. Importante para dar soporte a múltiples densidades y proporciones de pantalla.
- Permite la compilación del proyecto para que soporte múltiples versiones del sistema operativo desde un mismo código fuente.
- Fácil integración con otras tecnologías y servicios.
- Gran cantidad de extensiones, documentación disponible y una comunidad activa.

Atom



Es un editor de texto inteligente desarrollado por GitHub. Está orientado al desarrollo de código fuente, es libre y cuenta con amplio soporte para extensiones. [13]

Entre algunas de las que se le instalaron cabe mencionar a *ternjs* que provee funciones de autocompletado al editor haciendo más rápido el desarrollo para personas con poca experiencia en el lenguaje JavaScript. Se lo utilizó para hacer el desarrollo del servidor de back-end junto con la plataforma web.

Robomongo



Robomongo es una herramienta de código abierto multiplataforma para administrar bases de datos MongoDB. Esta permite conectarse de forma remota o local al motor de base de datos permitiendo la visualización y edición de colecciones, documentos e índices de una forma amigable para el usuario. [14]

Una vez avanzada la etapa de desarrollo, el servidor que antes corría de forma local sobre nuestras computadoras se migró a la nube. Ese cambio de etapa implicó una nueva configuración pero se mantuvo el mismo software que proporciona una interfaz para crear, destruir y modificar documentos con facilidad.

Git & GitHub



git



GitHub

Para el control de versiones se utilizó Git [15], un sistema distribuido de control de versiones. Cada desarrollador cuenta con una copia completa del repositorio en forma local. Se puede realizar cambios libremente sobre el proyecto para luego sincronizar el trabajo con el resto del equipo. Esto permite autonomía y flexibilidad para trabajar en grupo.

Se prefirió utilizar un sistema distribuido por las siguientes razones:

- Se puede realizar cambios locales y mantener la copia local del repositorio actualizado sin perder los cambios realizados sobre la misma (si es que no se dan conflictos).
- No es necesario contar con un acceso a un servidor remoto, lo que permite mayor producción al no haber bloqueo de archivos debido a que otro integrante se encuentre trabajando sobre el mismo código. La desventaja es que puede ocurrir que se hagan cambio sobre un mismo archivo, lo cual requiere posteriormente resolver el conflicto manualmente.

Como repositorio central, se decidió utilizar los servicios de *GitHub* [16] que se trata de una plataforma para el desarrollo colaborativo de software. Provee alojamiento a repositorios de proyectos utilizando el sistema de control de versiones *Git*. El código, en la versión gratuita del servicio, se almacena de forma pública y ofrece seleccionar entre varios tipos de licencias para proteger el trabajo que se está desarrollando.

GitHub ofrece además una extensión de gestión de incidentes que se utilizó para descomponer las tareas principales en tareas de programación y reportar los fallos encontrados en el sistema para su posterior corrección.

Capítulo 4: Análisis y diseño de la solución

Arquitectura

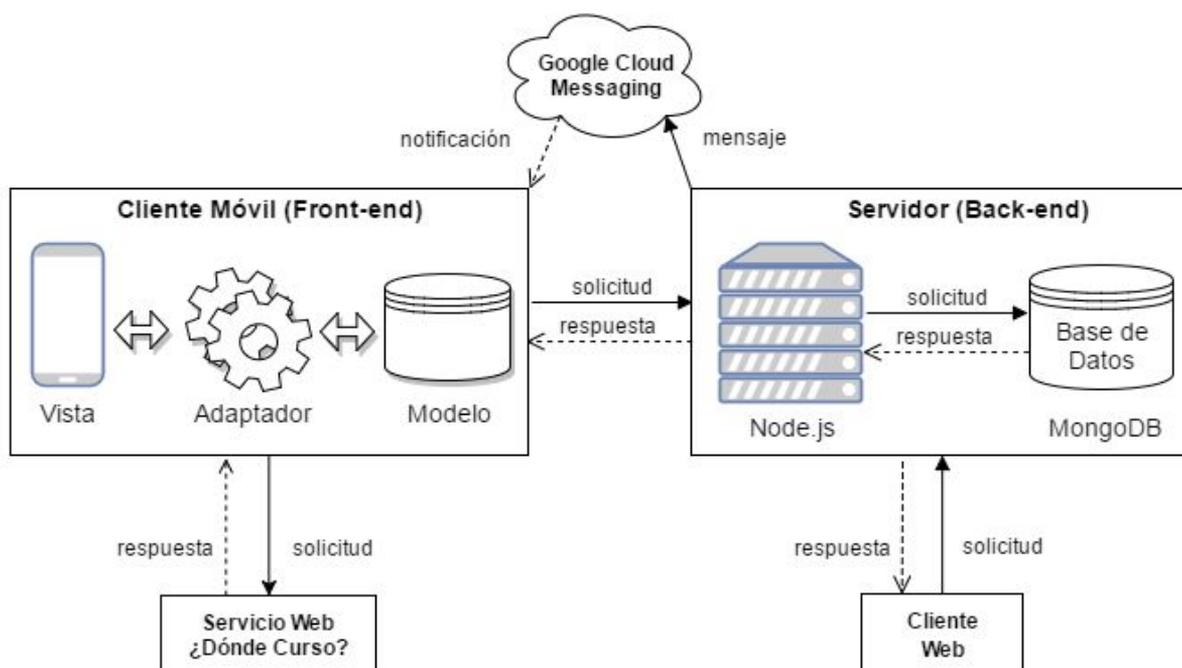


Figura 7. Diagrama de arquitectura del sistema.

El esquema de arquitectura presentado en la figura 7 funciona de la siguiente manera:

- Por un lado tenemos el cliente móvil que interactúa con el servidor y otros webservices como el actual *¿Dónde Curso?*. Además recibe mensajes desde el servicio *Google Cloud Message* para emitir notificaciones al usuario.
- Por otro lado en el servidor, tenemos alojados el motor y los scripts de Node.js. Y en la misma infraestructura la base de datos no relacional, MongoDB. El servidor además de responder las solicitudes puede enviar notificaciones a todos los dispositivos móviles que tienen instalado el cliente a través del servicio *Google Cloud Message*.
- La función del *Cliente HTML* es proveer una interfaz para que los usuarios que tengan el permiso correspondiente puedan enviar noticias, eventos al servidor, para que éste los almacene en la base de datos, o envíe notificaciones a los usuarios móviles.

- Por otro lado están los sistemas propios de la facultad como es el caso del *¿Dónde Curso?*. El cliente accede a la información de los mismos mediante webservices y un protocolo de solicitud/respuesta.

Patrón Modelo-Vista-Adaptador (MVA)

Para estructurar las clases del lado cliente de la aplicación móvil, se investigó sobre los distintos patrones que existen y cuáles son los más populares.

El patrón del que se parte es el de Modelo-Vista-Controlador (MVC) que es uno de los más usados tradicionalmente en el desarrollo de software, especialmente cuando se trata de sistemas basados en cliente y servidor. Este apunta a mejorar la reutilización de código e independencia entre los componentes del modelo y la vista. [17]

La principal característica de MVA es que se pueden tener múltiples adaptadores, permitiendo mostrar la misma información de diferentes maneras para un mismo par de modelo-vista.

Otra característica esencial es que la vista no tiene comunicación directa para manipular el modelo si no es a través de algún adaptador. [18]

Vista: es el componente que tiene interacción con el usuario que extienden de clases nativas como *Activity* o *Fragment*.

Adaptador: Se encargan de poblar de datos la vista y manejar eventos. Generalmente extienden de *BaseAdapter* y *Listener*.

Modelo: componente que tiene correspondencia con los datos como *Noticia* y *Evento*.

El motivo de elección de este patrón surge de que la mayoría de los datos en la aplicación deben ser presentados en forma de lista o celdas, a veces más de una por pantalla.

La modificación de cómo se decide mostrar la información que contienen es un cambio muy común que puede presentarse durante el desarrollo.

Las clases de modelo y vista están intencionalmente separadas porque ambas tienen restricciones inherentes a sus responsabilidades. El hecho de que estén desacopladas favorece el mantenimiento de la aplicación en un futuro.

Tener una capa independiente que procese la lógica y opere los datos de cada elemento de las listas o celdas permite una gran flexibilidad para realizar cambios sobre las mismas sin tener que modificar otras clases que ya fueron exhaustivamente probadas.

Usuarios identificados

La figura 8 muestra un diagrama de casos de usos donde se identificaron los principales actores. Se definieron tres categorías de usuarios:

- **Usuarios que consultan contenido:** Alumnos, futuros ingresantes, graduados, docentes y no docentes. También, aunque en menor medida, público en general.

- **Usuarios que generan contenido:** Personal del Área de Comunicación. Con la excepción de los horarios y aulas, que son obtenidos desde el sistema de gestión de recursos a través de una interfaz creada previamente para el sistema *¿Dónde Curso?*.
- **Usuarios que administran la base de datos:** Personal de la Subsecretaría de Tecnologías de la Información y Comunicaciones.

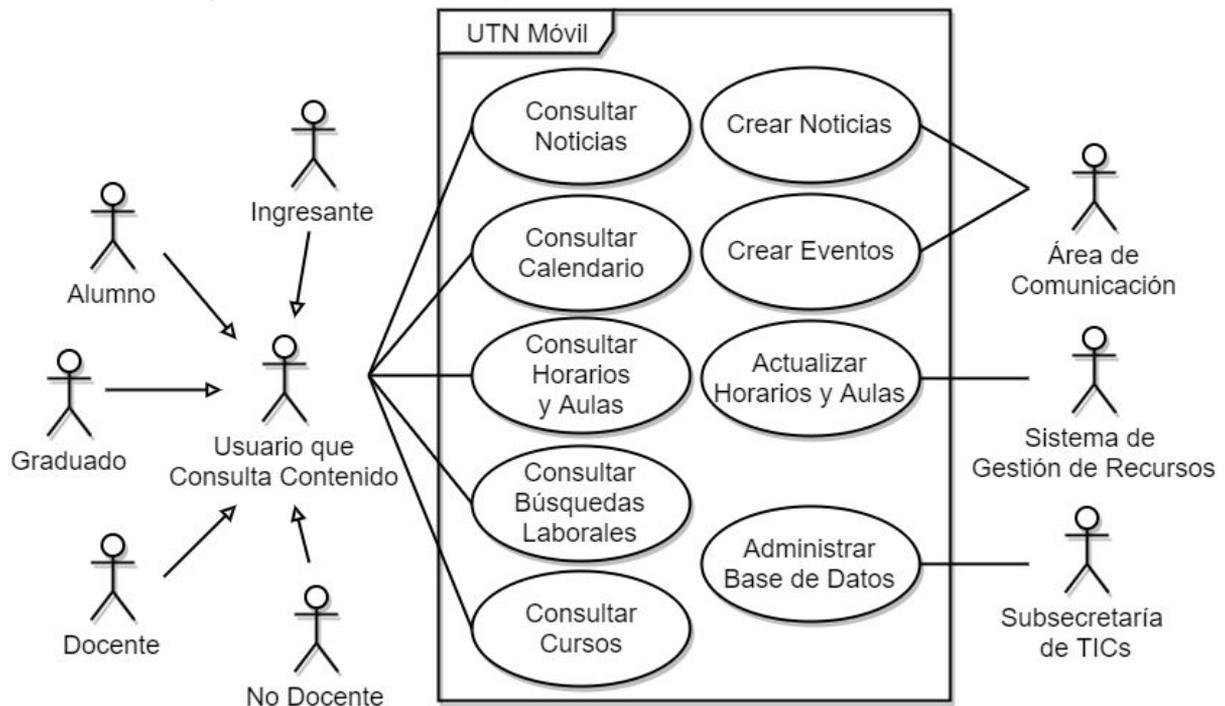


Figura 8. Diagrama de casos de uso.

Implementación

Semántica y esquemas de datos

MongoDB, así como otros motores de base de datos NoSQL, no fuerzan el uso de esquemas. Es decir que pueden convivir objetos con distintos atributos dentro de la misma colección.

Sin embargo, es recomendable darle un orden y significado a los datos para favorecer el desarrollo y la interoperabilidad con otros sistemas. También existen beneficios relacionados con la *Web Semántica* [19] y la manipulación automática de datos pero escapan al alcance de este proyecto. Se tuvieron en cuenta consideraciones de performance y simplicidad para las consultas más frecuentes iniciadas por la aplicación móvil. En la figura 9 se presenta una red semántica que contempla las entidades identificadas y sus relaciones.

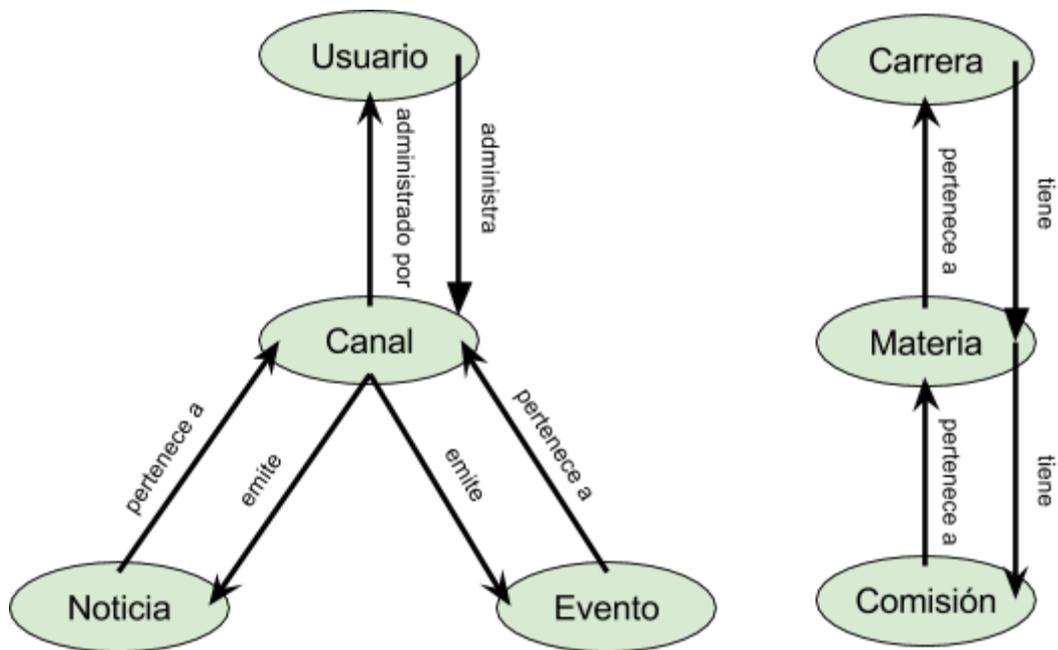


Figura 9. Esquemas de datos.

A continuación listamos los atributos que se detallaron para cada entidad en el proceso de diseño de la base de datos.

Noticia:

- **_id:** Clave única del objeto. Generada por MongoDB.
- **canal:** *_id* del canal al que pertenece. Ej: "global"
- **titulo:** Título descriptivo de la noticia.
- **cuerpo:** Contenido desarrollado de la noticia.
- **fecha:** Día y hora en el que fue publicada. Ej: "Mon, 05 Jan 2016 17:20:00 +0000"
- **imagen:** URL de una imagen representativa opcional.
- **autor:** Persona o entidad que emite la noticia.

Evento:

- **_id:** Clave única del objeto. Generada por MongoDB.
- **fecha:** Día y hora en que comienza el evento.
- **titulo:** Título descriptivo del evento.
- **lugar:** Espacio físico donde transcurre el evento.

Canal:

- **_id:** Identificador único del canal, es recomendable utilizar una cadena que pueda ser fácilmente interpretada y en minúsculas. Ejemplo: "institucional".
- **nombre:** Título que se mostrará del canal.
- **desc:** Descripción general del contenido del canal.

Usuario:

- **_id:** Identificador único del usuario. Generado por MongoDB.

- **usuario:** Nombre de usuario.
- **db:** Base de datos asociada en la que este usuario puede actuar. Esto se debe a que generalmente, por seguridad, la base de datos de administración y aplicación están disociadas.
- **credenciales:** Información encriptada acerca de la contraseña del usuario y sus correspondientes permisos.

Carrera*:

- **_id:** Clave única. Autonumérico por defecto.
- **nombre:** Nombre completo de la carrera.

Materia*:

- **_id:** Clave única. Autonumérico por defecto.
- **nombre:** Nombre que se mostrará de la asignatura.
- **nivel:** Nivel académico al que corresponde.
- **carrera:** Número entero que identifica a la carrera.
- **comisiones:** Arreglo de las comisiones disponibles.

Comisión*:

- **_id:** Clave única que identifica a la comisión.
- **nombre:** Nombre que se mostrará de la comisión. *Ej: "COMISIÓN ÚNICA"*

*Claves únicas y datos heredadas del sistema de gestión de recursos.

Estructura de archivos del Back-end

La figura 10 muestra cómo se estructuran los archivos y se distribuyen en sus correspondientes directorios. Se incluye una breve descripción del contenido de cada uno. Algunos archivos quedaron excluidos del repositorio de GitHub. Unos para evitar redundancias, como los módulos JavaScript (ubicados en el directorio *node_modules*) que pueden ser fácilmente instalados con el gestor de paquetes NPM (www.npmjs.com), la recomendación es no mantenerlos bajo control de versiones propio si no especificarlos en el archivo *packages.json*. Otros quedaron excluidos por cuestiones de seguridad, como es el ejemplo de los certificados y claves privadas que en manos de terceros comprometen a la seguridad del servidor.

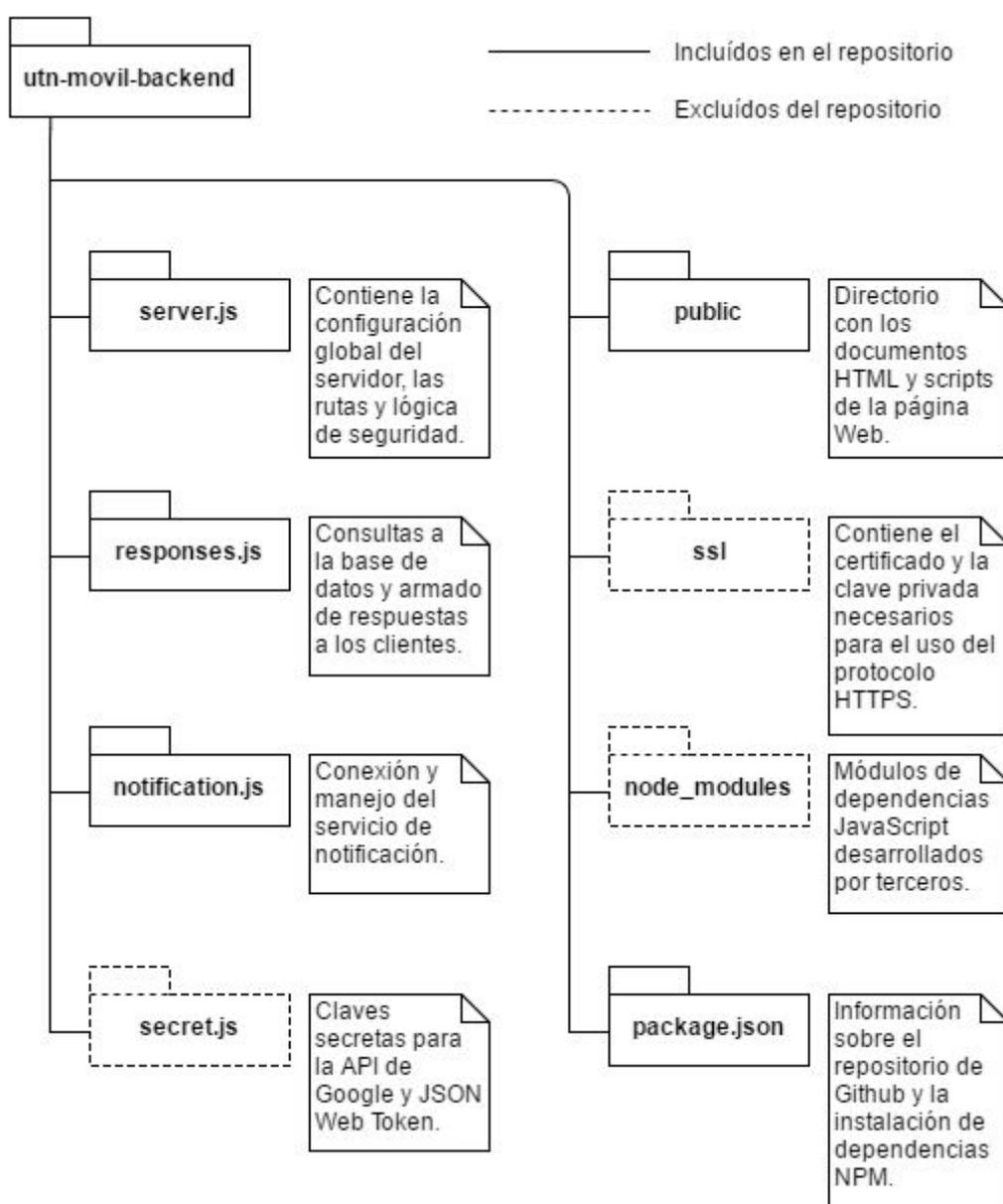


Figura 10. Diagrama de paquetes del servidor back-end.

Interfaz pública y privada del servidor

El protocolo utilizado para la interfaz pública es HTTP. Todas las solicitudes y respuestas son en formato JSON para que sean fáciles de interpretar por las partes en el intercambio de datos. Quien inicia las solicitudes es el cliente móvil a través de la librería de código abierto, *Volley*.

Se definió la interfaz pública de desarrollo del servidor. El sentido de dejar algunas consultas liberadas es alentar su uso por parte de estudiantes y programadores para crear y mejorar nuevas aplicaciones en diferentes plataformas cubriendo así las necesidades de información de una mayor cantidad de usuarios. Las mismas se pueden ver en el anexo *Interfaz pública del servidor*.

Por otro lado, están las operaciones que son exclusivas para los usuarios que crean contenido en el servidor y corresponden a la interfaz privada. Están protegidas por JSON Web Token y utilizan el protocolo HTTPS para que solamente puedan accederlas las personas que tengan las credenciales correspondientes. Estas operaciones están detalladas en tablas y se pueden consultar en el anexo *Interfaz privada del servidor*.

Despliegue del servidor



Google Cloud Platform

Durante la etapa de desarrollo, el despliegue del servidor de back-end se hizo en *Google Cloud Platform*, una plataforma que reúne todas las aplicaciones orientadas al desarrollo en la nube ofrecidas por la empresa. En particular *Google Compute Engine*, permite crear y utilizar una máquina virtual que se corre en sus servidores, la misma cuenta con el sistema operativo Debian en su versión 8.

En la imagen del servidor contratado estaban preinstaladas por defecto la mayoría de las tecnologías necesarias (NodeJS, MongoDB y Git entre otros). Para comunicarse y administrar la máquina virtual se utilizó *Putty*, que es un cliente para controlar servidores remotos a través de un intérprete de línea de comandos utilizando un protocolo seguro (SSH). [20]

Como se aprecia en la figura 11, las clases fueron organizadas en paquetes correspondiendo al patrón Modelo-vista-adaptador.

Estructura de la aplicación móvil

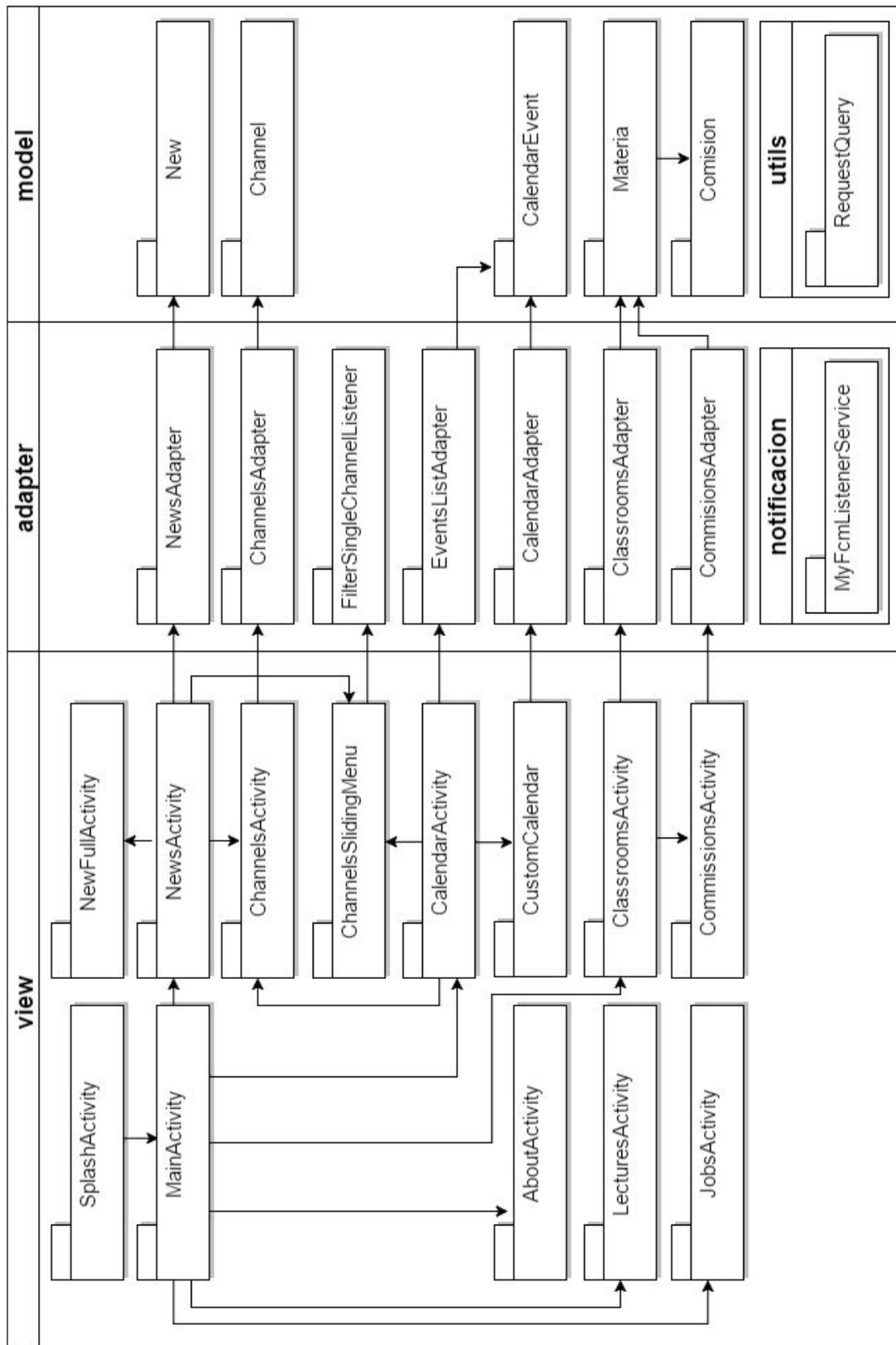


Figura 11: Diagrama de paquetes del cliente móvil.

El punto de entrada es la clase *SplashActivity* que muestra el logo de la aplicación unos segundos hasta que esta se termine de cargar por completo para luego mostrar la pantalla principal, *MainActivity*.

Desde este punto se puede acceder a las distintas secciones de la aplicación, indicadas por las flechas en la figura 11.

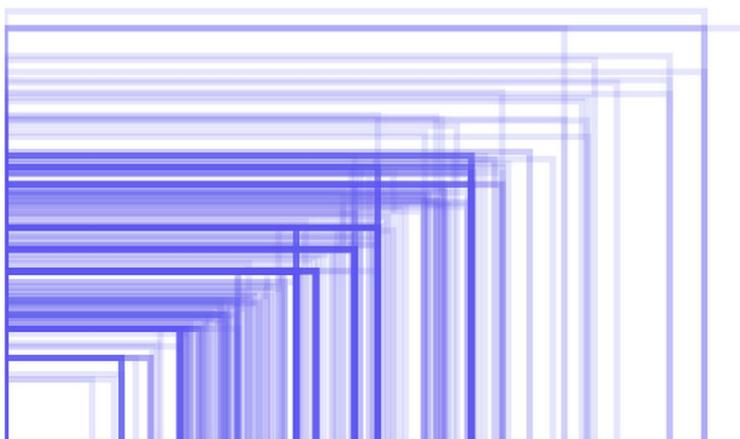
La pantalla de canales, *ChannelsActivity* puede ser accedida tanto desde la sección de noticias como la de eventos y es la misma globalmente para ambos, al igual que el menú deslizable, *ChannelsSlidingMenu* que permite filtrar de a un canal a la vez.

El paquete *notification* contiene las rutinas que se ejecutarán cuando llegue un aviso de notificación desde el servicio de mensajes en tiempo real.

El paquete *utils* contiene herramientas comunes utilizadas por las clases del modelo, como las funciones asincrónicas que construyen la conexión y recuperan información del servidor.

Soporte de múltiples densidades de pantalla

Representar la misma imagen en todos los dispositivos con una gran diversidad de tamaños de pantalla y resoluciones es uno de los problemas inevitables en el desarrollo móvil. A modo de ejemplo de esta fragmentación, se observa en la figura 12 un delineado superpuesto de resoluciones que pertenecen solo a algunos dispositivos existentes en el mercado y utilizan Android como sistema operativo.



*Figura 12. Fragmentación de resoluciones de pantalla de dispositivos Android.
Fuente: OpenSignal. [21]*

El concepto de DPI (puntos por pulgada), es una propiedad de la pantalla con forma rectangular que se mantiene desde su fabricación. Se calcula utilizando la fórmula presentada en la ecuación (1).

$$dpi = \frac{\text{resolución diagonal}}{\text{pulgadas de pantalla}} \quad (1)$$

Donde

$$\text{resolución diagonal} = \sqrt{\text{ancho}^2 + \text{alto}^2}$$

El ancho y el alto están dados en píxeles. Por ejemplo, en una pantalla de 540 por 960 píxeles y de 4,5 pulgadas, el dpi es 245, como se puede observar en la ecuación (2) al sustituir estos valores en la ecuación (1).

$$\text{dpi} = \frac{\sqrt{540^2 + 960^2}}{4,5 \text{ ''}} \approx 245 \quad (2)$$

Asociado a esto aparece el concepto de *dp*, que es una unidad abstracta independiente de los *dpi*, útil para el desarrollo, la equivalencia es mostrada en la ecuación (3). Por convención, está definida relativamente a una pantalla de 160 *dpi*. Pero el intérprete se encarga de traducirlos al tamaño en píxeles correspondiente según cada dispositivo.

$$1 \text{ dp} = 1 \text{ pixel en una pantalla de } 160 \text{ dpi} \quad (3)$$

La posición y tamaño de los componentes visuales pueden estar expresados en función de esta unidad y mantener una consistencia con todas las pantallas de los dispositivos.

A esto se suma la posibilidad de definir las propiedades de los elementos visuales en forma relativa a las dimensiones totales de pantalla y a otros componentes, el resultado es una interfaz gráfica de usuario que se adapta de la forma más óptima posible a cualquier resolución y tamaño físico de pantalla.

Tradicionalmente las aplicaciones móviles utilizan la estrategia de incluir una copia de la imagen en un tamaño relativo para cada densidad de pantalla. [22] Se pueden observar en la figura 13 las siglas más comunes utilizadas para denominar el agrupamiento de los rangos de puntos por pulgada y las relaciones entre ellos.

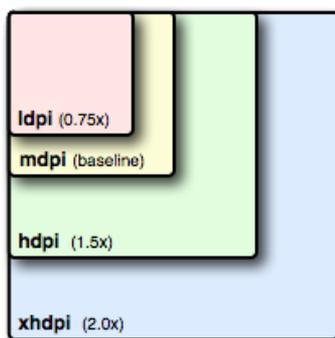


Figura 13. Grupos de densidades de pantalla tradicionalmente usados.

Esta estrategia tiene como desventajas la cantidad de archivos de imagen que hay que incluir en el paquete final, de los cuales se utilizarán solamente una pequeña cantidad de las mismas.

Esto sumado a que con el avance de las tecnologías se fabrican pantallas cada vez más grandes y con densidades más altas como el *xxhdpi*, *xxxhdpi* y *tvdpi*.

Otra desventaja es que se incluye una sola imagen para un rango amplio de densidad de pantalla, pero sucede que la coincidencia no siempre es exacta y la imagen tiene que estirarse o contraerse dando lugar al indeseable efecto denominado como pixelado que se produce cuando el ojo humano puede percibir los píxeles por los que está compuesta la imagen.

En el presente proyecto utilizamos una estrategia alternativa que utiliza un solo archivo en formato XML. El proceso completo, mostrado en la figura 14, comienza con el concepto de la imagen, tal como es creada por el diseñador.

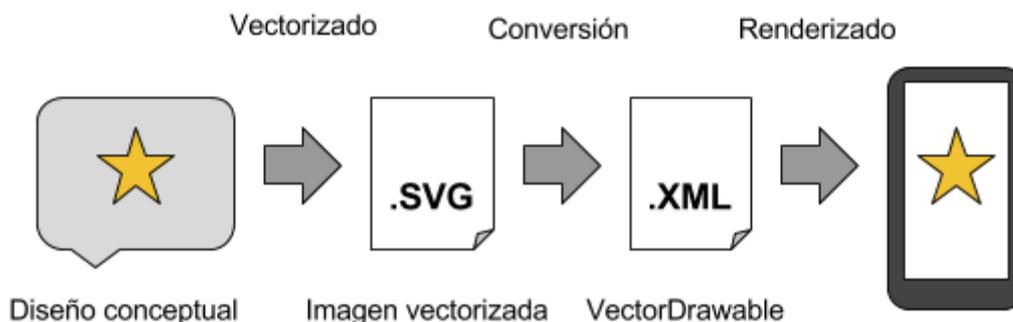


Figura 14. Cadena de una imagen vectorial multidensidad desde su concepción hasta su visualización en el dispositivo.

Luego pasa por un proceso de vectorizado, que la transforma en curvas vectoriales y al formato abierto SVG (Scalable Vector Graphics).

Luego para convertirlas en el formato que es aceptado por el entorno de desarrollo de *Android Studio* se puede usar la herramienta integrada que este incluye pero todavía se encuentra en etapas tempranas de madurez.

Se tuvieron mejores resultados con una herramienta libre desarrollada por terceros denominada *svg2android* disponible en la web. [23]

Por último la clase *VectorDrawable* y las librerías de renderizado integradas se encargan con la información provista de adaptar y mostrar la imagen al tamaño adecuado de cada dispositivo.

Como principales ventajas, el uso de esta técnica representa una considerable reducción del espacio de almacenamiento utilizado y también una calidad superior en el amplio rango de pantallas que existen. Como contrapartida, puede aumentar los tiempos de carga iniciales.

Requisitos No Funcionales

A continuación se describen los requisitos no funcionales tenidos en cuenta en el desarrollo de esta aplicación.

a) Seguridad

Se describen a continuación los protocolos utilizados para proveer comunicaciones seguras en este proyecto.

JSON Web Token



JSON web token (JWT), es un estándar abierto basado en el RFC 7519 [24] que define una forma compacta y autocontenida de representar afirmaciones seguras entre dos partes.

Existen numerosas implementaciones en distintos lenguajes, para el desarrollo de esta aplicación se usó la implementación de JavaScript para Node.js que es compatible con el servidor utilizado.

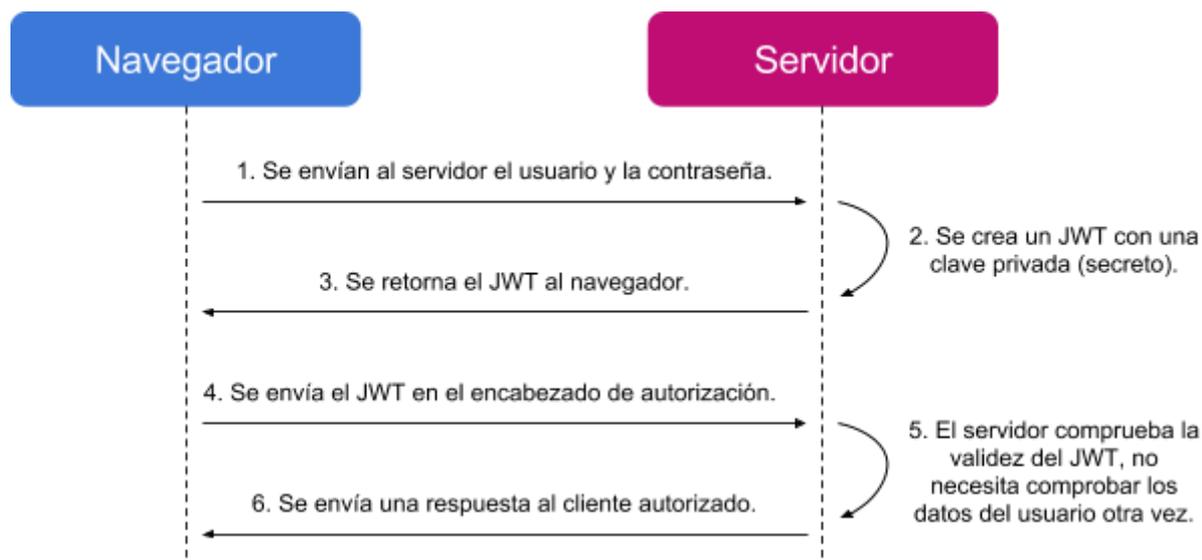


Figura 15. Diagrama de flujo del funcionamiento básico de JWT.

La figura 15 muestra un diagrama de flujo del funcionamiento básico de este protocolo. Los tres primeros pasos se realizan una sola vez al principio de la sesión. Luego se repiten los restantes en cada par de solicitud-respuesta seguro entre el navegador y el servidor.

Una consideración sobre la implementación en el presente proyecto es que el JWT se almacena en una cookie para que la sesión persista por más tiempo y sea compatible con más navegadores.

La confianza del método reside en la complejidad del *secreto* utilizado para encriptar y verificar transacciones. Puede ser una cadena alfanumérica que existe solamente del lado del servidor y no debe ser revelada bajo ninguna circunstancia.

La información extra necesaria que se intercambia es mínima y se compone de tres partes: (i) Encabezado: Donde se especifica el algoritmo de encriptación y el tipo de token, en este caso JWT. (ii) Carga útil: El contenido que se desea intercambiar de forma segura, en formato JSON. Generalmente serán datos del perfil del usuario y sus permisos. (iii) Firma: El resultado de aplicar el algoritmo de encriptación al encabezado y la carga útil, utilizando la clave privada del servidor. [25]

Características que son útiles al proyecto

La aplicación de este protocolo, soluciona en parte los posibles problemas de seguridad en los extremos, así se puede asegurar que las credenciales no sean almacenadas en texto plano independientemente del navegador que se use. También contiene (en forma cifrada) los permisos que son otorgados por el servidor al usuario y no es necesario validarlos contra la base de datos en cada consulta ya que se encuentran almacenados en el token.

HTTPS (Hypertext Transfer Protocol Secure)



El protocolo HTTPS, también conocido como HTTP sobre TLS (RFC 2818 [26]) permite cifrar una conexión en un túnel de extremo a extremo entre el cliente y el servidor. A fines de evitar posibles robos o modificaciones maliciosas de la información y suplantaciones de identidad.

Para su correcta implementación son necesarios un certificado y una clave privada.

Características que son útiles al proyecto

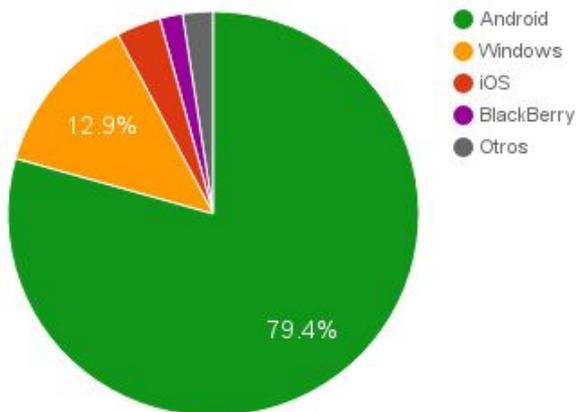
La aplicación de este protocolo soluciona la mayoría de los problemas de seguridad en los nodos intermedios de la red, ya que con su correcta implementación reduce la superficie de ataque a los puntos extremos de la conexión.

b) Movilidad

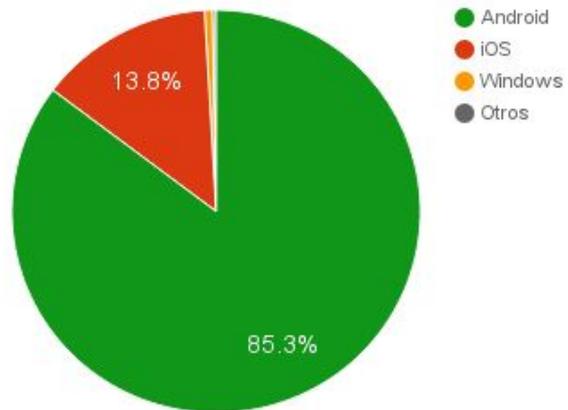
Uno de los desafíos del desarrollo móvil es lograr compatibilidad con la mayor cantidad de dispositivos. En el mundo coexisten de manera concurrente múltiples fabricantes de dispositivos móviles, quienes utilizan en ellos sistemas operativos propios o de uso libre. Se

tomó la decisión de dar prioridad a la plataforma Android por sus pocas restricciones tanto en su desarrollo como en la distribución de aplicaciones y su gran alcance a nivel mundial. En la figura 16 (a), se muestra el porcentaje de uso de cada sistema operativo en Argentina, mientras que en 16 (b) a nivel mundial. Como se observa casi el 80% de los celulares en Argentina utilizan Android.

Ventas de Smartphones por Sistema Operativo en Argentina (2015)



Ventas de Smartphones por Sistema Operativo en el mundo (2016)



(a)

(b)

Figura 16. Cuota de mercado de los sistemas operativos móviles. Fuentes: Kantar WorldPanel [27] y Gartner [28] respectivamente.

Incluso dentro del sistema operativo Android se cuenta con múltiples versiones del mismo que han sido distribuidas a lo largo de los años, las proporciones de uso de cada una sobre el total de instalaciones se muestran en la figura 17.

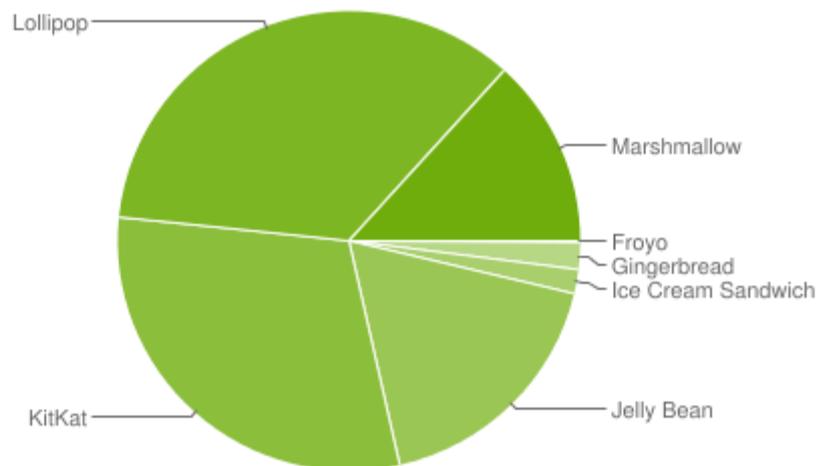


Figura 17. Distribución de las diferentes versiones del sistema operativo Android. Fuente: Android Developers Dashboards. [29]

La solución propuesta dentro del kit para desarrolladores (SDK) son clases implementadas en el lenguaje nativo de desarrollo (Java) que en conjunto con las herramientas de

compilación hacen el mejor esfuerzo para que la aplicación se vea y comporte de forma predecible a lo largo de todas sus variantes.

El cliente móvil que se desarrolló apunta al nivel de API 14 o superiores, esto implica según el gráfico una compatibilidad con más del 98% de los dispositivos Android del mundo. Lograr la compatibilidad con el 100% de los dispositivos implicaría sacrificar algunas funcionalidades que no están disponibles en versiones anteriores pero son útiles al proyecto.

Se analizaron otras alternativas como los sistemas operativos iOS y Windows, pero no se contaba con los recursos para hacer las pruebas correspondientes en cada plataforma, por lo que quedaron fuera del alcance del proyecto.

c) Usabilidad

Es importante buscar siempre que haya una experiencia de usuario agradable, que el mismo pueda acceder rápidamente a alguna funcionalidad que necesite, que la interfaz de usuario sea intuitiva, etc. Todo esto contribuye en gran parte al éxito de una aplicación. Se aplicaron algunas buenas prácticas para obtener una buena experiencia de usuario, las cuales se describen a continuación:

En la aplicación móvil, se hizo uso de los lineamientos de diseño, denominados por Google como *Material Design* [30], que intenta mantener una consistencia a través de los componentes, gestos, estilos y colores en las aplicaciones. El objetivo de esto es lograr que los usuarios se familiaricen con prácticas comunes en dispositivos móviles. Como beneficio principal la aplicación resulta más intuitiva y conserva una relación estética con las que vienen preinstaladas y otras desarrolladas por terceros. También se hizo uso de *Toasts (avisos flotantes)* los cuales aparecen por encima de la aplicación y desaparecen al cabo de unos segundos, el objetivo de los mismos es proporcionarle al usuario información que lo oriente con la acción que está realizando o debe realizar a continuación.

En el cliente web, se utilizó *Responsive Design (Diseño Adaptable)* el cual consiste en que mediante la definición de ciertas reglas, un único diseño web reacciona a la resolución de la ventana del navegador adaptándose a la misma, obteniendo una visualización adecuada.

d) Mantenibilidad

Dado que en este proyecto se desarrolla un conjunto inicial de funcionalidades pensando que estas se puedan expandir en un futuro, se hizo hincapié en el uso de una arquitectura que facilite la expansión de las funcionalidades.

En el cliente web, se utilizó el patrón *MVA (Model-View-Adapter)* para separar el código en capas, como ya se explicó con anterioridad.

En el servidor, se utilizó un archivo para que contenga las rutas de las peticiones al servidor y otro archivo para las funciones con las consultas que acceden a la Base de Datos, lo cual aumenta la legibilidad y la organización del código.

e) Cuestiones Legales

Como se trata de un sistema no solamente con fines académicos sino que también existe en un entorno de producción y abierto al público, se investigó un poco más acerca de las formalidades legales para evitar problemas futuros.

Certificados Digitales y Autoridades de Certificación

Un certificado digital válido es necesario para la correcta implementación del protocolo HTTPS, que forma parte de las tecnologías involucradas en la seguridad de nuestro servidor.

También se utiliza esta tecnología de forma similar para firmar digitalmente una aplicación como desarrollador, lo cual representa un requerimiento para publicarla en mercados digitales para su posterior distribución.

Una autoridad de certificación (CA) es una entidad de confianza, responsable de emitir y revocar certificados digitales, utilizados en la firma electrónica.

Estas autoridades tienen a su vez una jerarquía, donde las que firman sus propios certificados son conocidas como entidades raíz y estas autorizan mediante mecanismos de criptografía a las entidades subordinadas que también pueden emitir certificados a una entidad final.

Cada una de estas entidades posee un nombre distinguido (DN), que las identifica de forma exclusiva. El DN posee varios atributos como un nombre propio, el nombre de la organización, la unidad organizativa, la localidad, el estado o provincia y el país. A continuación, en la figura 18, se puede observar cómo está compuesto el certificado de cada entidad.

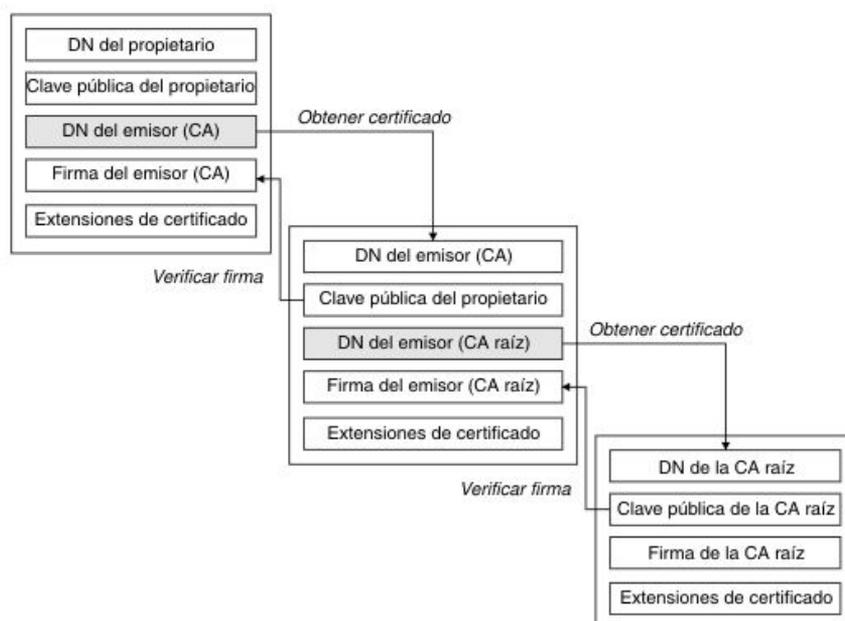


Figura 18. Cadena de certificación confiable. Fuente: IBM Knowledge Center [31].

Todas estas comprobaciones son realizadas de forma automática por todos los navegadores modernos, si todo el proceso es correcto se muestra el aviso de que se está usando una conexión segura al lado de la dirección de la web.

Licencias de código abierto

Durante el desarrollo es común encontrarse con problemas que ya fueron previamente estudiados y solucionados. Cada vez más gente en el mundo se agrupa en comunidades para abstraerse de cuestiones redundantes a través de la colaboración y así poder enfocarse en los problemas del futuro.

Existen contratos cuidadosamente elaboradas por expertos para darle un marco legal a esta actividad. Todas las librerías utilizadas en este proyecto se encuentran bajo alguna de las siguientes licencias.

Licencia Apache 2.0

Es una licencia de software libre creada por la Apache Software Foundation. La tabla 4 presenta un cuadro con un resumen de los principales puntos de la misma. [32]

Tabla 4: Resumen de puntos clave para la licencia Apache 2.0.

Permisos	Condiciones	Limitaciones
<ul style="list-style-type: none"> ● Uso comercial ● Distribución ● Modificación ● Uso bajo patente ● Uso privado 	<ul style="list-style-type: none"> ● Incluir licencia y copyright ● Incluir aviso de cambios de estado 	<ul style="list-style-type: none"> ● No se asumen responsabilidades ● No otorga permisos de marcas

El software distribuido bajo la misma es flexible con su uso, distribución y modificación. Se pueden vender copias, publicar o sublicenciar.

En particular, patentar algo que incluya alguna de estas licencias está permitido ya que explícitamente esos derechos son cedidos de por vida, excepto que se quiera litigar por la propiedad de ellos, en ese caso la concesión pierde toda validez. [33]

Se deben cumplir obligatoriamente algunas condiciones para usar las obras licenciadas como parte del nuevo software (como es el caso de las librerías), la primera es incluir un aviso de copyright que atribuya la obra a sus correspondientes autores y también una copia de la licencia.

También se debe incluir un archivo *NOTICE* en los trabajos derivados, que es un documento de texto con todos los avisos obligatorios de las porciones de software utilizadas, además de aclarar si se produjeron modificaciones de las mismas y cuales fueron en ese caso.

A diferencia de otras licencias *Copyleft*, no requiere que el software derivado esté concebido bajo la misma licencia.

Como posibles desventajas, aclara que el código fuente es provisto sin ningún tipo de garantías y nadie tomará responsabilidades por los posibles daños que su uso pueda

ocasionar. Tampoco son cedidos los derechos de marca, nombres comerciales o de producto.

Para aplicar esta licencia a este trabajo, se incluye el copyright y licencia en el archivo *LICENSE*, dentro del código fuente o en forma de documentación.

Licencia MIT (o X11)

Se origina en el Instituto Tecnológico de Massachusetts (MIT), tiene pocas restricciones y es muy utilizada en el ambiente del software libre. La tabla 5 muestra las características de esta licencia de acuerdo a los permisos, condiciones y limitaciones. [34]

Tabla 5: Resumen de puntos clave para la licencia MIT.

Permisos	Condiciones	Limitaciones
<ul style="list-style-type: none">• Uso comercial• Distribución• Modificación• Uso privado	<ul style="list-style-type: none">• Incluir licencia y copyright	<ul style="list-style-type: none">• No asumen responsabilidades

A diferencia de la licencia Apache 2.0, no es necesario incluir un informe sobre los cambios de estado realizados al software provisto. Tampoco dice nada acerca de las patentes ni el uso de las marcas. [35] Es una licencia más simple y libre, aunque no llega a ser de dominio público. También se debe atribuir el trabajo a sus correspondientes autores e incluir una copia de la licencia. Cabe aclarar que el no cumplimiento de alguna de las condiciones, invalida todos los derechos otorgados.

Suplantación de identidad

Con el aumento en los últimos años del uso de internet y las nuevas tecnologías, el delito de suplantación de identidad se ha vuelto una preocupación para la mayoría de las legislaciones. [36] En el presente proyecto se encontró en dos oportunidades esta problemática.

En el primer caso porque es importante evitarlo en la emisión de noticias y eventos, como se trata de una aplicación de impacto institucional solamente pueden hacerlo quienes estén debidamente autorizados. Se invirtió gran cantidad de tiempo en investigar e implementar medidas de seguridad actuales y confiables sin sacrificar la facilidad de uso del sistema.

La segunda fue cuando se deposita la aplicación para su revisión en *Google Play*, la primera vez fue rechazada por infringir sus políticas de uso a través de una posible suplantación de identidad. Esto se debió a la utilización de herramientas automatizadas por parte de Google y los íconos que utilizamos resultaron ser similares a los de otras aplicaciones preexistentes. El error fue aclarado cuando se presentó la documentación necesaria y se intercambiaron correos electrónicos con su departamento de soporte al desarrollador.

Capítulo 5: Documentación de uso

Si bien el proyecto consiste en el desarrollo de una aplicación móvil, también se cuenta con una interfaz web que es para la gente (personal de la facultad) que administra el contenido que se va a publicar (noticias, eventos, etc). Es decir, por un lado tenemos los usuarios que consultan contenidos mediante el uso de la aplicación móvil y por otro lado tenemos los usuarios que crean dichos contenidos a través de la página web. A continuación, se presenta la documentación de uso para los usuarios que crean contenidos (parte web), y luego para los usuarios que consultan contenidos (parte móvil).

Usuario creador de contenidos

Esta clase de usuarios es en este caso el personal del Área de Comunicación de la UTN FRSF, que es el único (por el momento) que tiene los permisos necesarios para publicar contenido en la aplicación.

Las acciones que pueden realizar son:

- Publicar Noticias
- Publicar Eventos

Para realizar esto, en primer lugar, deben loguearse en la plataforma web para publicar contenidos, presentando las credenciales indicadas.

Iniciar sesión

El usuario debe proveer las credenciales, usuario y contraseña, requeridas para loguearse. La figura 19 muestra la pantalla de login de usuarios.

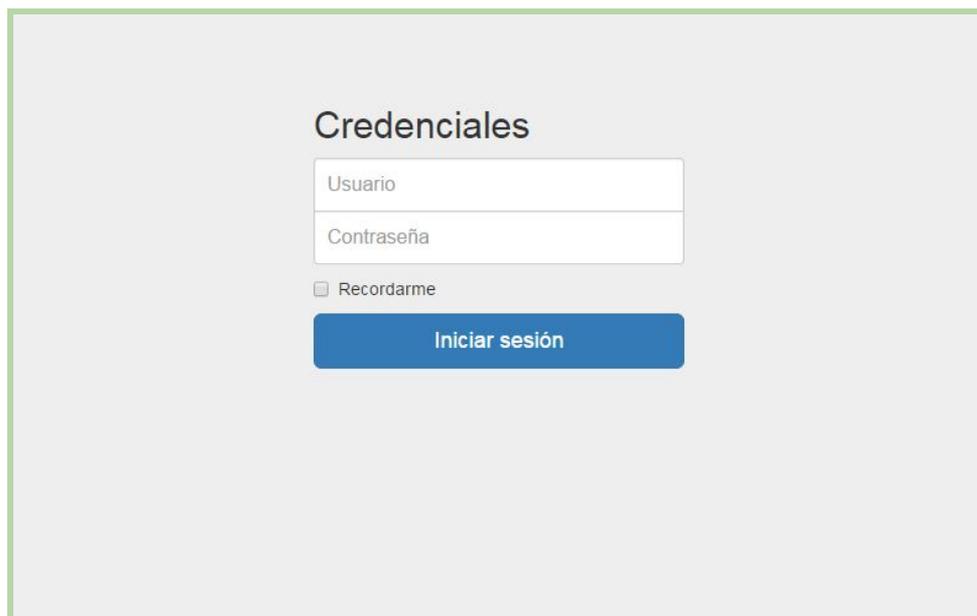
La imagen muestra una pantalla de autenticación con el título "Credenciales". Hay dos campos de entrada: "Usuario" y "Contraseña". Debajo de estos campos hay un checkbox etiquetado "Recordarme". En la parte inferior de la pantalla hay un botón azul con el texto "Iniciar sesión".

Figura 19. Pantalla de autenticación de los usuarios.

Luego de haber iniciado sesión el usuario tiene la posibilidad de realizar las 2 acciones antes mencionadas, cargar una noticia o un evento.

Encabezado

Una vez que el usuario ha proporcionado credenciales correctas, es posible usar el sistema. Un elemento compartido es el encabezado que proporciona acceso a las funciones principales, como se ve en la figura 20.

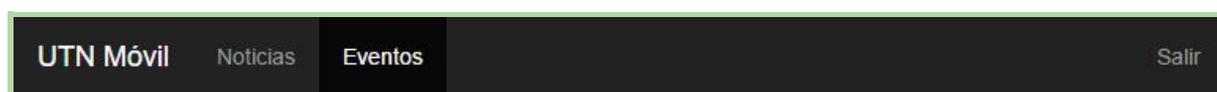


Figura 20. Encabezado de las páginas de creación de contenido.

Contiene el nombre de la aplicación, dos pestañas y un botón para salir del sistema.

La pestaña *Noticias* deriva al formulario de creación de noticias. La pestaña *Eventos* deriva al formulario de creación de eventos. Por último el botón *Salir* borra las credenciales del almacenamiento interno del navegador y retorna a la pantalla de inicio de sesión.

Crear noticia

La figura 21 muestra la ventana de diálogo para cargar una noticia.

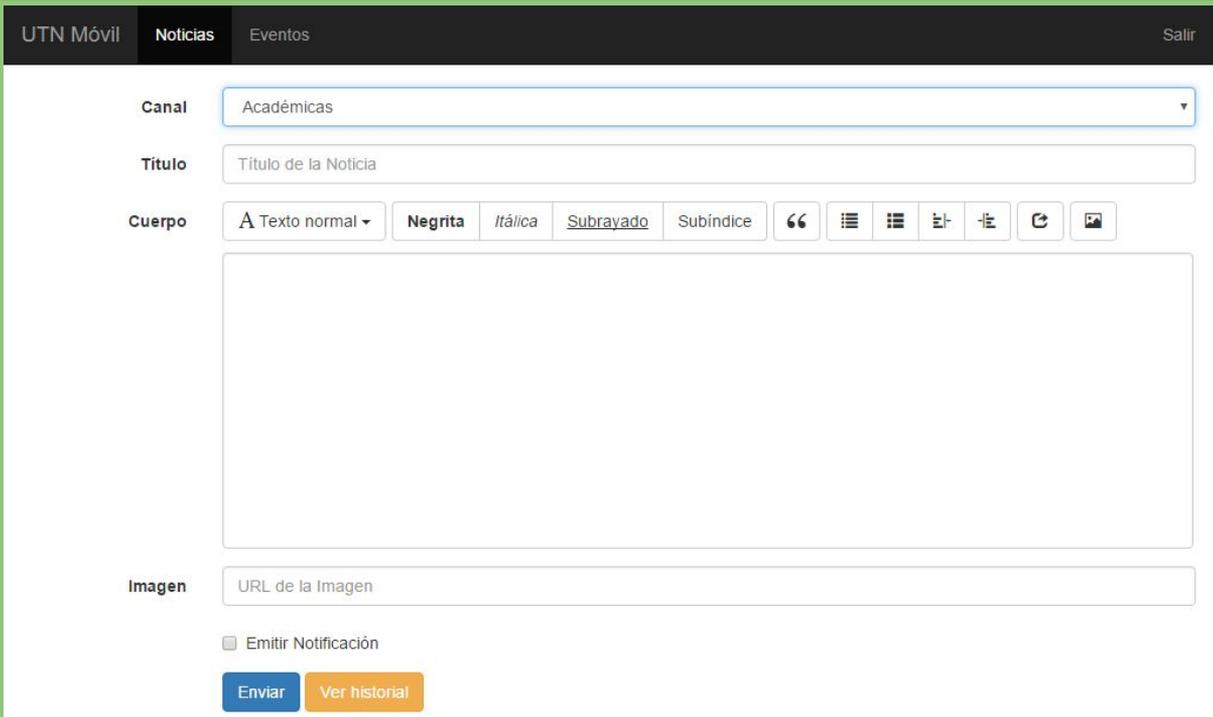


Figura 21. Pantalla para la carga de noticias.

En primer lugar, el usuario tiene que elegir por cual canal enviará la noticia, las opciones disponibles inicialmente son:

- Académicas
- Estudiantiles
- Ciencia y Tecnología
- Institucionales

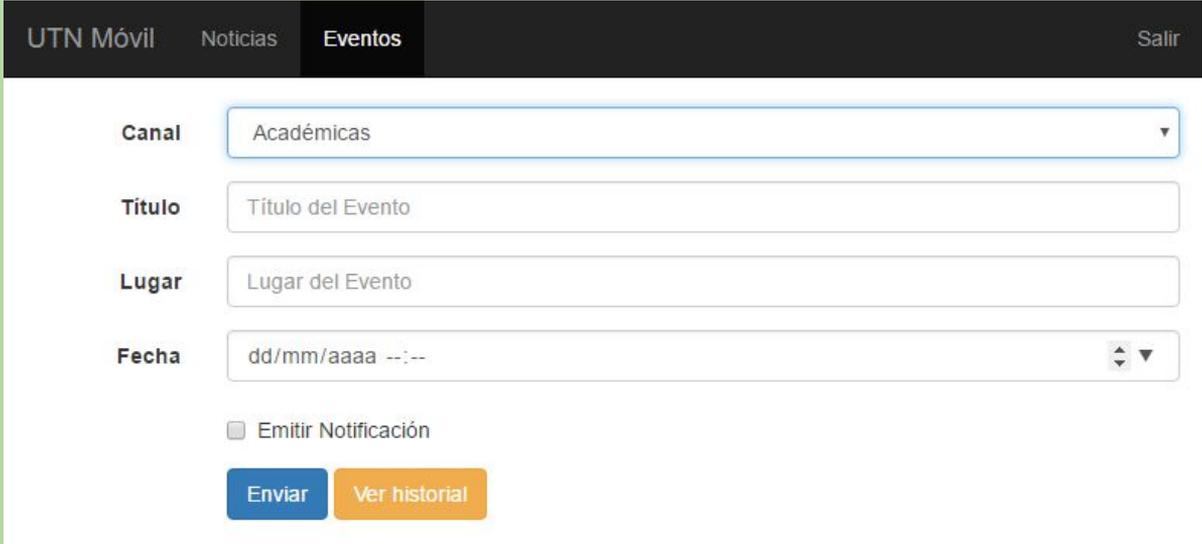
Luego tiene que escribir cuál será el título de la noticia, y a continuación el cuerpo de la misma, para esto hay un editor que permite insertar formato con distintos estilos (normal, subrayado, negrita, itálica), varios tamaños de fuente y además permite insertar imágenes y links. En el caso de querer insertar una imagen, se debe proveer un link en el campo *Imagen*.

Por último, se puede tildar la opción *Emitir notificación* para enviar notificación a los dispositivos que tiene instalada la app, avisando que hay una nueva noticia.

Al presionar *Enviar* se cargará la noticia en la base de datos y se enviará la notificación a los dispositivos, si es que fué seleccionada la opción.

Crear evento

La ventana de diálogo que se muestra al abrir la pestaña *Eventos* se muestra en la figura 22.



UTN Móvil Noticias **Eventos** Salir

Canal Académicas

Título Título del Evento

Lugar Lugar del Evento

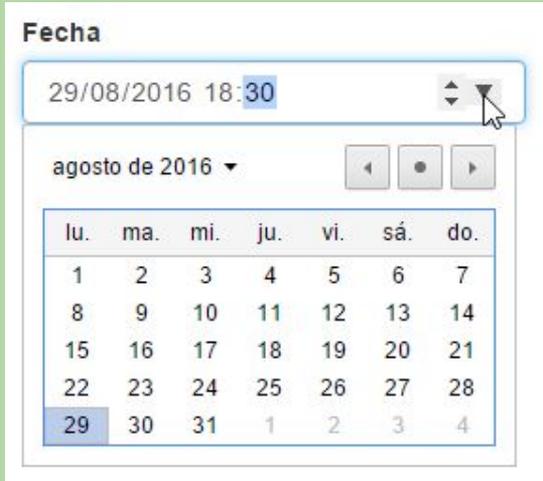
Fecha dd/mm/aaaa --:--

Emitir Notificación

Enviar Ver historial

Figura 22. Pantalla para la carga de eventos.

Para cargar un evento, el usuario tiene que elegir el canal por el que este se emitirá, siendo éstos los mismos que fueron presentados para el caso de las noticias. Luego debe ingresar un título y un lugar para el mismo. Por último se tiene que elegir la fecha. Para este caso, se muestra un cuadro desplegable con un calendario para facilitar la acción del usuario, el cual se muestra en la figura 23. También aparece la opción de emitir notificación para poder avisar de dicho evento a todos los dispositivos que tienen la *app* cargada.



Fecha

29/08/2016 18:30

agosto de 2016

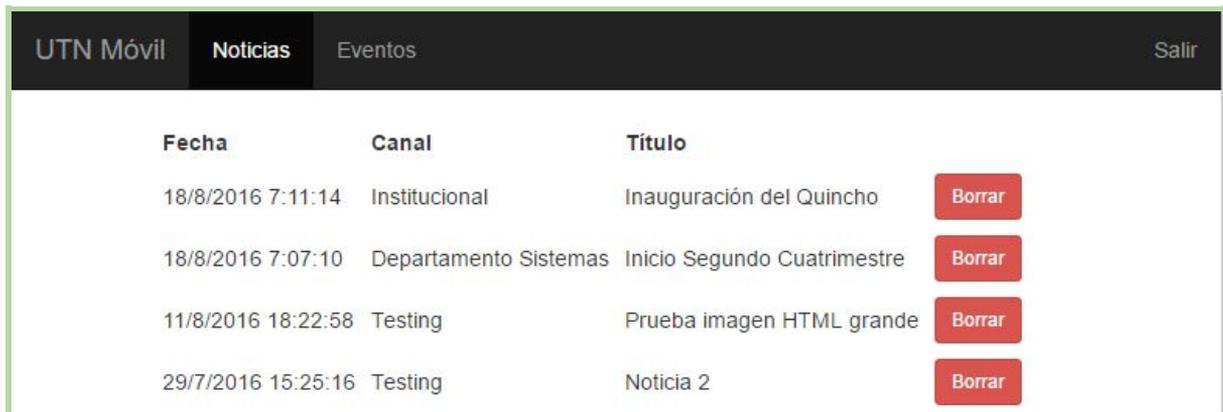
lu.	ma.	mi.	ju.	vi.	sá.	do.
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Figura 23. Selector desplegable de fecha y hora.

Al presionar *Enviar* se cargará el evento en la Base de Datos y se emitirá la notificación a los dispositivos en caso que corresponda.

Consultar y eliminar contenido

Tanto en la sección de Noticias o Eventos, se incluye un botón de “Ver historial” el cual permite consultar las entradas más recientes en la base de datos, ordenadas por fecha, como se ve en la figura 24.



Fecha	Canal	Título	
18/8/2016 7:11:14	Institucional	Inauguración del Quincho	Borrar
18/8/2016 7:07:10	Departamento Sistemas	Inicio Segundo Cuatrimestre	Borrar
11/8/2016 18:22:58	Testing	Prueba imagen HTML grande	Borrar
29/7/2016 15:25:16	Testing	Noticia 2	Borrar

Figura 24. Historial de noticias recientes.

Por este medio podemos comprobar si las operaciones que se realizaron fueron correctas. En el caso de haberse cometido un error en la creación de una noticia o evento, también existe la posibilidad de darlos de baja utilizando el botón de “Borrar” asociado a su fila en la tabla.

Usuario que consulta contenidos

Esta clase de usuarios son los que interactúan con la aplicación móvil, la cual deben tener instalada en sus celulares. Dentro de este grupo de usuarios pueden haber alumnos, profesores, personal administrativo de la facultad, y ya que la aplicación está disponible para todo el público también la pueden usar gente interesada en la facultad como pueden ser futuros ingresantes.

Identificación de la aplicación

Toda aplicación Android debe tener un nombre y un ícono que la identifique del resto de las aplicaciones, el nombre escogido fué **UTN Móvil** y el ícono es el que se muestra en la figura 25.

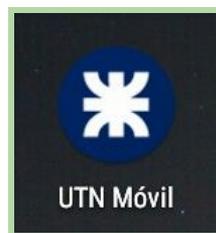


Figura 25. Ícono y nombre de la aplicación.

Pantalla principal

Al ejecutar la aplicación se mostrará el logo de la facultad durante unos segundos mientras la misma se inicializa. Al terminar esta acción aparece la pantalla principal mostrada en la figura 26.



Figura 26. Pantalla principal UTN Móvil.

Esta pantalla contiene las opciones que el usuario tiene a disposición. En la sección inferior, hay tres íconos correspondientes a los links a las páginas web oficiales de la UTN FRSF: la web www.frsf.utm.edu.ar, la cuenta oficial de Facebook y la cuenta oficial de Twitter respectivamente. Si se selecciona alguno de estos botones se abre la aplicación correspondiente, si se encuentra instalada (en el caso de Facebook y Twitter) o por defecto el navegador web predeterminado del dispositivo, como se muestra en la figura 27.

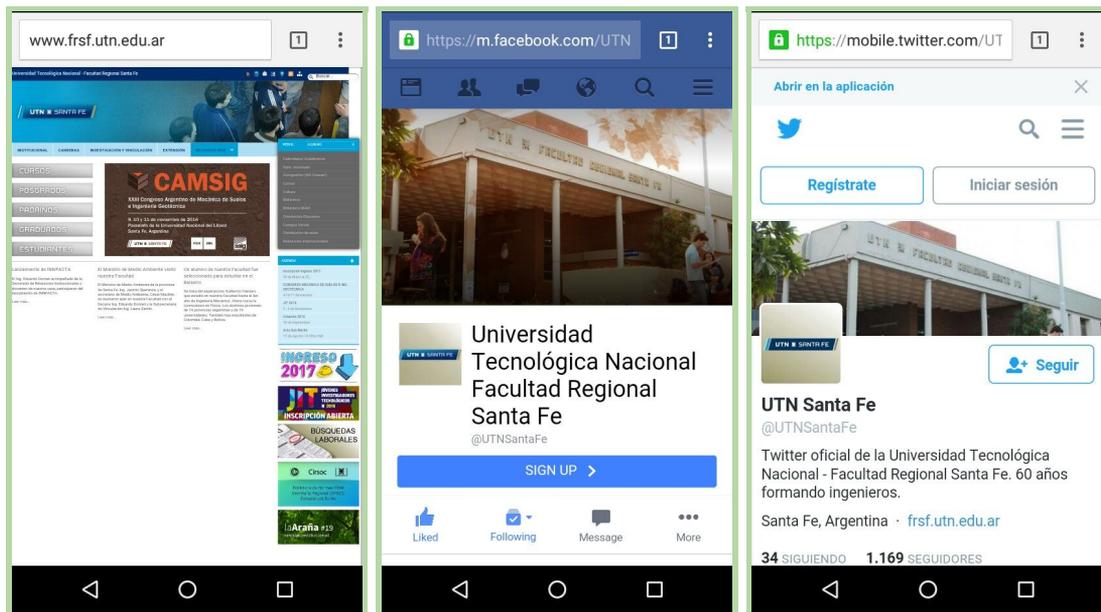


Figura 27. Página web y redes sociales oficiales de la UTN FRSF.

Sección noticias

Al acceder a la sección *Noticias* la primera pantalla que se muestra es la lista de las últimas noticias pertenecientes a los canales a los que el usuario se encuentra suscrito, ordenadas por fecha, como se muestra en la figura 28.



Figura 28. Sección Noticias.

El usuario puede ampliar el contenido de una noticia en particular pulsando sobre cualquier parte del recuadro de la que le interese, la noticia se mostrará como se ve en la figura 29.



Figura 29. Visualización de una noticia completa.

El usuario puede configurar los canales de noticias a los que está suscrito accediendo en la esquina superior derecha al botón que despliega la opción “Configurar Mis Canales”, como se ve en la figura 30.



Figura 30. Opción “Configurar Mis Canales” de noticias.

Al acceder a dicha opción, aparece la lista de canales existentes, permitiendo al usuario marcar si se quiere suscribir o cancelar la recepción de información de cada canal, como se muestra en la figura 31. Cada canal aparece con un nombre, una descripción y sobre el lado derecho un selector para activar o desactivar dicho canal.

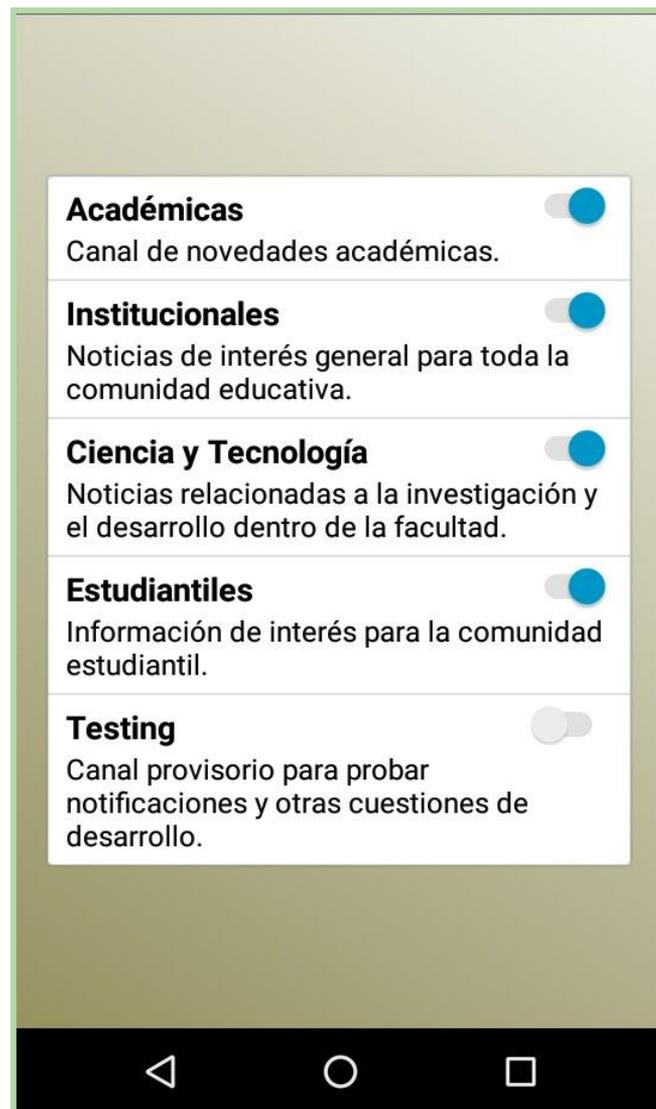


Figura 31. Configuración de suscripción a canales de noticias.

Además, en la pantalla de las últimas noticias, en la esquina superior izquierda hay un botón que si se lo selecciona aparece una barra lateral con una lista de opciones. El usuario también puede acceder a ella con un gesto desde donde comienza el extremo izquierdo de pantalla y luego deslizando hacia la derecha. La barra lateral se muestra en la figura 32.



Figura 32. Barra filtro de canales de noticias.

Aquí el usuario puede elegir si quiere visualizar las noticias de los canales a los que está suscrito actualmente (opción *Mis Canales*, la cual está por defecto), o si quiere visualizar las noticias de un canal en particular, independientemente de si se encuentra suscrito a ese canal o no. La opción seleccionada no alterará la preferencia de suscripciones del usuario, solamente va a actualizar la lista de noticias que se va a mostrar.

Calendario

Al seleccionar *Calendario* se visualiza un calendario en el cual están marcados con un color distinto los días en los que hay cargados eventos de los canales a los que el usuario se encuentra suscrito, como se ve en la figura 33.



Figura 33. Sección Calendario.

El usuario puede seleccionar un día marcado con un círculo azul para ver la lista de eventos que acontecen en ese día, como se ve en la figura 34. El número dentro del círculo indica la cantidad de eventos que acontecen simultáneamente en esa fecha del mes.



Figura 34. Visualización de un evento.

El usuario puede configurar los canales de eventos a los que está suscrito accediendo en la esquina superior derecha al botón que despliega la opción *Configurar Mis Canales*, como se muestra en la figura 35.



Figura 35. Opción "Configurar Mis Canales" de eventos.

Al acceder a dicha opción, aparece la lista de canales existentes, permitiendo al usuario indicar si se quiere suscribir o no a cada canal, en la figura 36 se muestra dicha lista.

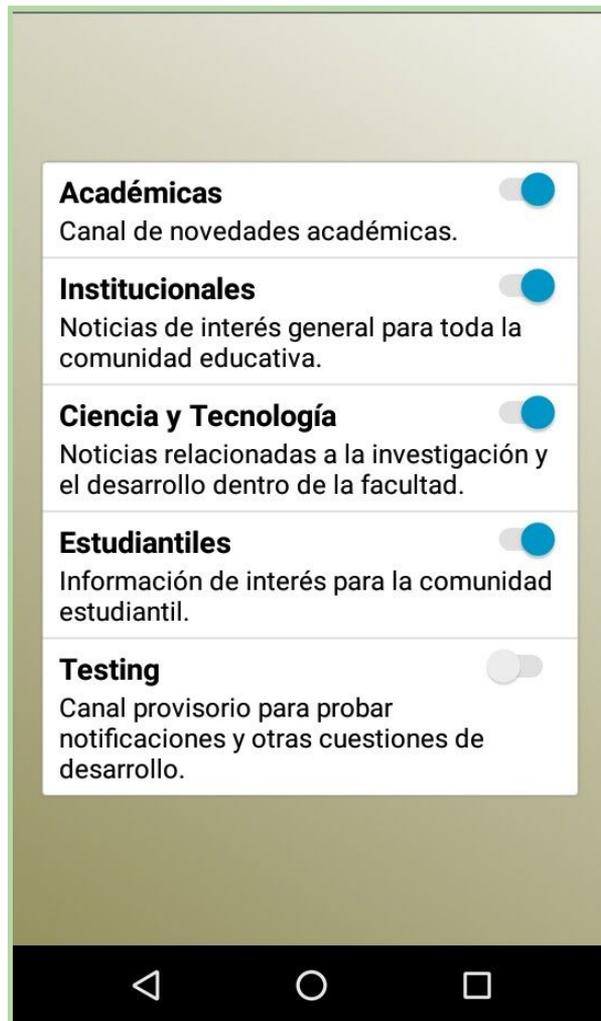


Figura 36. Configuración de suscripción a canales de eventos.

Al igual que en la sección *Noticias*, en la esquina superior izquierda hay un botón que si lo presionamos aparece una barra lateral con una lista de opciones. El usuario también puede acceder a ella arrastrando la pantalla hacia la derecha. La barra lateral se muestra en la figura 37.



Figura 37. Barra filtro de canales de eventos.

Aquí el usuario puede elegir si quiere visualizar los eventos de los canales a los que está suscrito actualmente (opción *Mis Canales*, la cual está por defecto), o si quiere visualizar los eventos de un canal en particular, independientemente de si se encuentra suscrito a ese canal o no. La opción seleccionada no altera la preferencia de suscripciones del usuario, solamente filtra los eventos que se van a mostrar del canal seleccionado hasta que la pantalla sea abandonada o se seleccione otro canal.

Horarios y aulas

Al seleccionar la opción *Horarios y Aulas* del menú principal se visualiza la lista de materias que el usuario tiene configuradas para que aparezcan de cada una el aula y horario de cursado como se muestra en la figura 38.



Figura 38. Sección Horarios y Aulas.

En la esquina superior derecha hay un botón que despliega una serie de opciones: *Cambiar Fecha*, *Forzar Actualización* y *Configurar Materias* como se muestra en la figura 39.



Figura 39. Menú opciones "Horarios y Aulas".

La opción *Cambiar Fecha* permite cambiar el día para el cual se realiza la consulta de horarios y aulas de las materias configuradas por el usuario. Por defecto siempre se muestra el día actual, si el usuario quiere puede cambiar la fecha eligiéndola desde el calendario que se aprecia en la figura 40.



Figura 40. Calendario para elegir fecha de consulta.

La segunda opción del menú en la figura 39 es *Forzar Actualización*, la cual sirve para que la aplicación haga la consulta al servidor para traer la información de las materias, esto es útil en el caso de que al entrar a la sección *Horarios y Aulas* la consulta falle por problemas de la conexión a internet u otros motivos, mediante este mecanismo el usuario elige realizar de nuevo la consulta.

Por último *Configurar Materias*” permite al usuario seleccionar de a una las materias a las que se quiere suscribir o desuscribir.

En esta pantalla para elegir una materia el usuario primero debe seleccionar la carrera a la cual pertenece la materia, y luego en el campo de texto que aparece abajo tiene que escribir el nombre de la materia, aunque con solo escribir las primeras letras se sugieren una lista de materias para su autocompletado, como se ve en la figura 41.

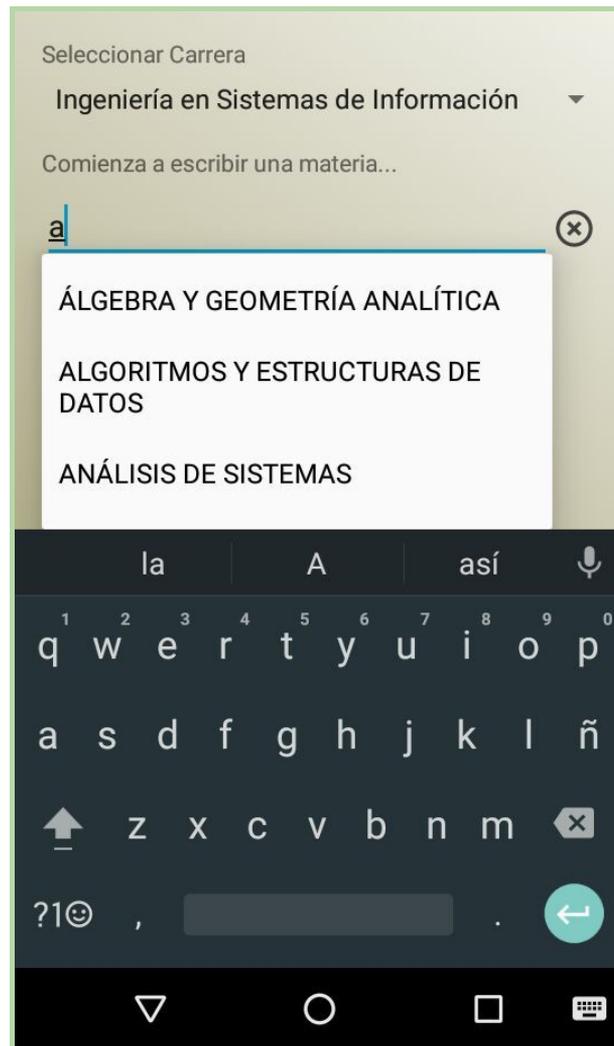


Figura 41. Pantalla de "Configurar Materias".

Luego de haber elegido la materia, se presentan las comisiones disponibles de la materia seleccionada, pudiendo elegir las comisiones que desee, como se muestra en la figura 42. En la pantalla de *Horarios y Aulas* va a figurar la materia una vez por cada comisión elegida. Si la comisión ya se encontraba seleccionada el usuario puede des-seleccionarla y así dejar de estar suscrito a esa comisión.

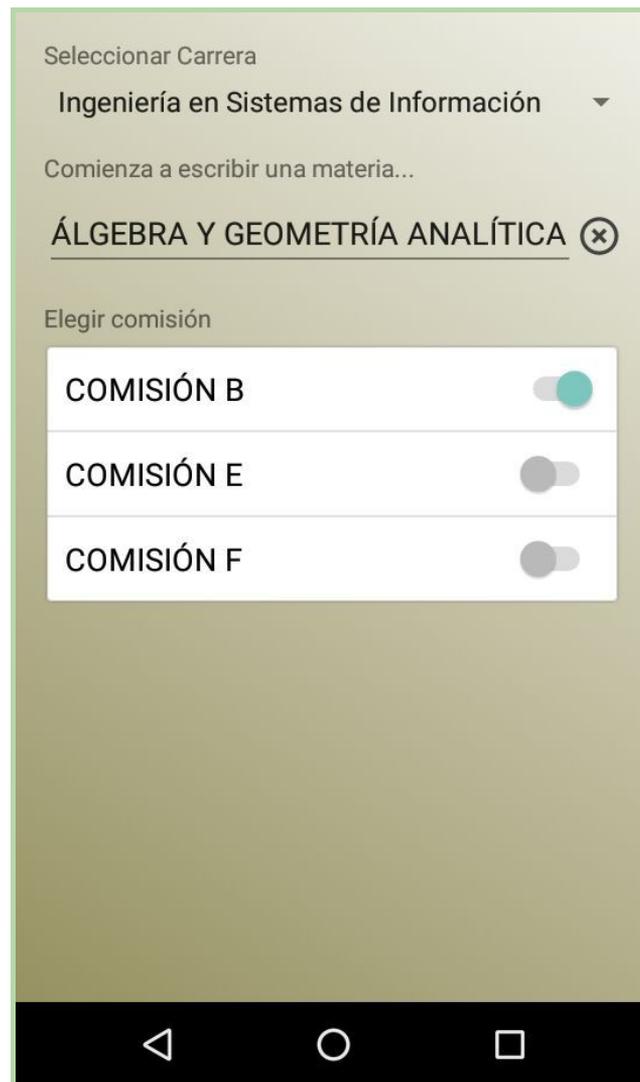


Figura 42. Selección de comisión para una materia.

Cursos

Cuando se selecciona la opción *Cursos* en el menú principal de la aplicación, se presentan los contenidos que existen en la sección *Cursos* de la web de la UTN FRSF www.frsf.utn.edu.ar/cursos, la cual contiene información sobre cursos a distancia de *Otra forma de aprender*, idiomas, etc. En la figura 43 se muestra dicha sección.

Capacitaciones de la Secretaría de Extensión Universitaria

Cursos a Distancia "Otra Forma de Aprender" - 2do. semestre de 2016 (inicios a partir de la 2da. quincena de Agosto)

OTRA FORMA DE APRENDER UTN SF DISTANCIA

CURSOS CON INICIO A PARTIR DEL 20 DE AGOSTO

Estudí a distancia, con nuestra innovadora metodología de clases en vivo por internet.

Cupos Limitados. Con el prestigio y aval de UTN Santa Fe.
<http://www.frsf.utn.edu.ar/cursos>

Area Automatismo Industrial:

- *Automatización Industrial
- *Desarrollo de sistemas de control basados en plataforma Arduino
- *Instrumentación Industrial

Figura 43. Sección "Cursos".

Búsquedas laborales

Cuando se selecciona la opción *Búsqueda laboral* del menú principal de la aplicación, se presentan los contenidos que existen en la sección *Búsquedas Laborales* de la web de la UTN FRSF www.frsf.utn.edu.ar/graduados/busquedas-laborales, la cual contiene información sobre las diversas propuestas laborales que las empresas difunden mediante la web de la facultad para que alumnos y graduados las conozcan y se puedan postular para las convocatorias. En la figura 44 se muestra dicha sección.

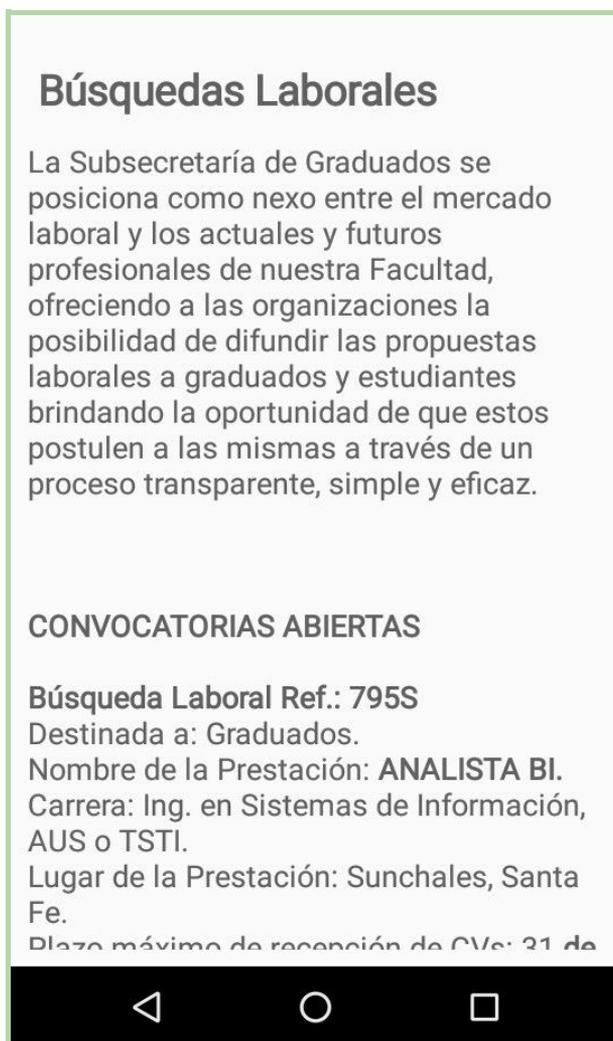


Figura 44. Sección "Búsquedas Laborales".

Acerca de

Esta sección está dividida en dos pestañas. La primera, *Créditos*, se muestra información sobre la aplicación, quienes fueron los autores y en qué ámbito se realizó (figura 45.a).

En la segunda pestaña *Licencias*, se incluye todo lo relativo al uso de licencias de código abierto que se usaron durante el desarrollo de la misma, incluirlas representa una obligación legal y además una forma de agradecimiento a sus autores (figura 45.b).

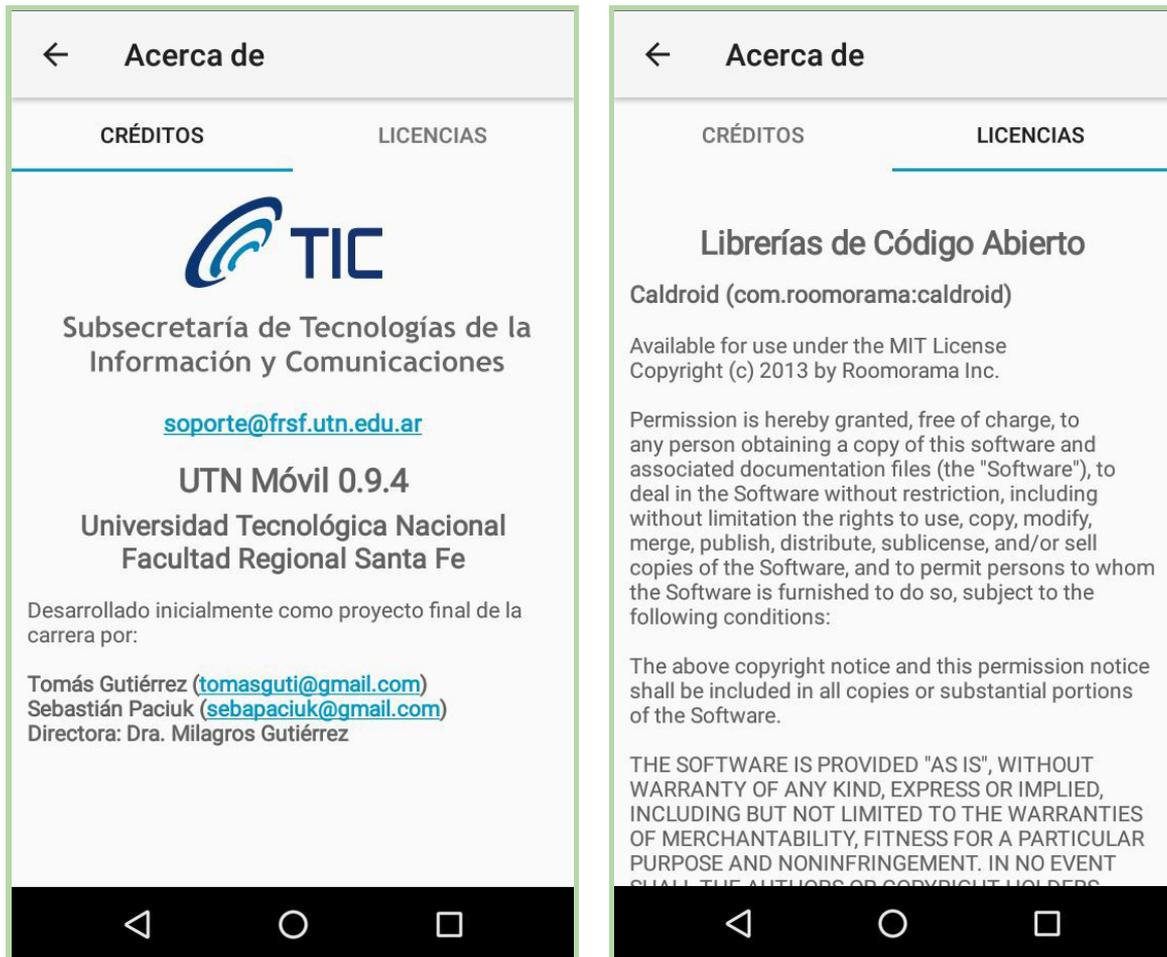


Figura 45. Sección “Acerca de”, pestaña “Créditos” y “Licencias”.

Capítulo 6: Conclusiones y trabajos futuros

Uno de las principales motivaciones de este proyecto se basó en el interés por las tecnologías móviles y se buscó durante el desarrollo del mismo avanzar en el aprendizaje tanto teórico como práctico de las mismas.

Respecto a la planificación estamos en condiciones de decir que fue adecuada al proyecto y sirvió como guía durante su transcurso. Algunos problemas generales pueden señalarse, como la sobreestimación de las etapas de análisis y diseño que luego eran compensadas por la subestimación de etapas de desarrollo. La principal causa de la extensión en algunos de los plazos fueron las nuevas características que se agregaron durante la ejecución (Sección Acerca De, Cursos y Búsquedas Laborales), que surgían de necesidades específicas y las pruebas de los prototipos. También se pueden sumar que algunos riesgos fueron materializados y se requirió algo de esfuerzo adicional en el requisito no funcional de la seguridad, pero su resolución se encontró dentro de lo esperado.

La metodología aplicada permitió ponerse al tanto del estado del arte en cuanto al desarrollo tanto de las tecnologías del servidor de back-end como las del cliente móvil. También se pudo alcanzar las expectativas de las partes interesadas, permitiendo además su participación activa dentro del proyecto.

Nuestro objetivo fue lograr una aplicación que se vea y funcione como las que se utilizan a diario en los dispositivos de uso común, que resulte intuitiva y aporte información de valor para el usuario final.

Con todo el conocimiento adquirido se puede afirmar que el equipo se encuentra en condiciones de adoptar una metodología ágil para las próximas iteraciones que puede ser más adecuada para seguir incorporando cambios y desarrollando nuevos módulos.

Algunos de los trabajos a futuros que surgieron en etapas de análisis son los siguientes:

- Módulo para reportar incidencias de mantenimiento de la facultad, como daños edilicios o de propiedad de la facultad.
- Mapa inteligente de la facultad. Que indique la ubicación aproximada del usuario pueda encontrar lugares dentro del edificio con facilidad.
- Información sobre los deportes que se practican dentro de la facultad y eventos relacionados a ellos.
- Integración con los sistemas de biblioteca. Ya existen servicios de consulta interesantes en web, se los podría adaptar a su versión móvil.
- Difusión de actividades relacionadas con la cantina de la facultad, desde precios y menús semanales hasta información sobre peñas y otros eventos de interés para la comunidad estudiantil.

La interfaz del servidor es pública porque se pretende el desarrollo de clientes móviles para otras plataformas en un futuro y extender el uso del servicio a más personas.

Bibliografía

- [1] Nueva forma de acceder al sistema de distribución de aulas [ref. agosto 2016]. Disponible en Web: <http://www.frsf.utn.edu.ar/noticias/484-cuandocurso.html>
- [2] Cuándo pasa? [ref. agosto 2016] Disponible en Web: <http://cuandopasa.efibus.com.ar/>
- [3] Tipos de aplicaciones móviles: ventajas y desventajas [ref. agosto 2016]. Disponible en Web: <http://www.avante.es/tipos-de-aplicaciones-moviles-ventajas-y-desventajas/>
- [4] Tanenbaum, Andrew. (2003). Redes de computadoras. Cuarta edición.
- [5] Pressman, Roger. (2010). Ingeniería del software, Un enfoque práctico. Séptima edición.
- [6] Reinventando la gestión de datos | MongoDB [ref. agosto 2016]. Disponible en Web: <https://www.mongodb.com/es>
- [7] BSON - Binary JSON [ref. agosto 2016]. Disponible en Web: <http://bsonspec.org/>
- [8] Google Research Publication: MapReduce [ref. agosto 2016]. Disponible en Web: <http://research.google.com/archive/mapreduce.html>
- [9] About Node.js [ref. agosto 2016]. Disponible en Web: nodejs.org
- [10] Google Cloud Messaging [ref. agosto 2016]. Disponible en Web: <https://developers.google.com/cloud-messaging/>
- [11] Bootstrap [ref. agosto 2016]. Disponible en Web: <http://getbootstrap.com/>
- [12] Android Studio y SDK Tools [ref. agosto 2016]. Disponible en Web: <https://developer.android.com/studio/index.html>
- [13] Atom [ref. agosto 2016]. Disponible en Web: <https://atom.io/>
- [14] Robomongo [ref. agosto 2016]. Disponible en Web: <https://robomongo.org/>
- [15] About - Git [ref. agosto 2016]. Disponible en Web: <https://git-scm.com/about/distributed>
- [16] GitHub [ref. agosto 2016]. Disponible en Web: github.com
- [17] Java SE Application Design with MVC [ref. agosto 2016]. Disponible en Web: <http://www.oracle.com/technetwork/articles/javase/index-142890.html>
- [18] Model-view-adapter [ref. agosto 2016]. Disponible en Web: <https://en.wikipedia.org/wiki/Model-view-adapter>
- [19] Guía Breve de Web Semántica [ref. agosto 2016]. Disponible en Web: <http://www.w3c.es/Divulgacion/GuiasBreves/WebSemantica>
- [20] Putty - SSH and telnet client for Windows [ref. agosto 2016]. Disponible en Web: <http://www.putty.org/>
- [21] Android Fragmentation Visualized - OpenSignal [ref. agosto 2016]. Disponible en Web: <https://opensignal.com/reports/fragmentation.php>
- [22] Supporting Multiple Screens | Android Developers [ref. agosto 2016]. Disponible en Web: https://developer.android.com/guide/practices/screens_support.html
- [23] Android SVG to VectorDrawable [ref. agosto 2016]. Disponible en Web: <http://inloop.github.io/svg2android/>
- [24] RFC 7519 - JSON Web Token (JWT) [ref. agosto 2016]. Disponible en Web: <https://tools.ietf.org/html/rfc7519>
- [25] JSON Web Token [ref. agosto 2016]. Disponible en Web: <https://jwt.io/>
- [26] RFC 2818 - HTTP over TLS [ref. agosto 2016]. Disponible en Web: <https://tools.ietf.org/html/rfc2818>

- [27] Gartner Press Releases Archive [ref. agosto 2016]. Disponible en Web: <http://www.gartner.com/newsroom/archive/>
- [28] Smartphone OS sales market share - Kantar World Panel ComTech [ref. agosto 2016]. Disponible en Web: <http://www.kantarworldpanel.com/global/smartphone-os-market-share/>
- [29] Dashboards | Android Developers [ref. agosto 2016]. Disponible en Web: <https://developer.android.com/about/dashboards/index.html>
- [30] Introduction - Material Design [ref. agosto 2016]. Disponible en Web: <https://material.google.com>
- [31] IBM Knowledge Center - Cómo funcionan las cadenas de certificados [ref. agosto 2016]. Disponible en Web: http://www.ibm.com/support/knowledgecenter/es/SSFKSJ_7.0.1/com.ibm.mq.csqzas.doc/sy10600_.htm
- [32] Apache License 2.0 - Choose a License [ref. agosto 2016]. Disponible en Web: <http://choosealicense.com/licenses/apache-2.0/>
- [33] Licenses - The Apache Software Foundation [ref. agosto 2016]. Disponible en Web: <https://www.apache.org/licenses/>
- [34] MIT License - Choose a License [ref. agosto 2016]. Disponible en Web: <http://choosealicense.com/licenses/mit/>
- [35] Lista de licencias con comentarios - Proyecto GNU [ref. agosto 2016]. Disponible en Web: <https://www.gnu.org/licenses/license-list.html#X11License>
- [36] Borghello, Cristian. y Temperini, Marcelo. (2012). Suplantación de Identidad Digital como delito informático en Argentina. Simposio Argentino de Informática y Derecho. ISSN: 1850-2814

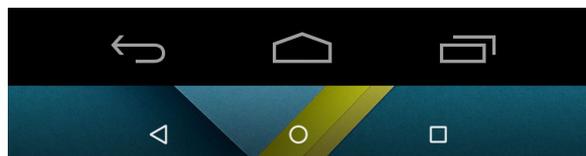
Anexos

El siguiente documento se presentó durante la primera iteración para mostrar a grandes rasgos las funcionalidades y un diseño tentativo para ser aprobado por la Subsecretaría de TICs y el Área de Comunicación. Se pueden apreciar las diferencias con la versión final, debido a las iteraciones y la retroalimentación recibida luego de cada una.

Diseño de Pantallas (mockups)

Se utilizó la herramienta Evolus Pencil (open-source) para diagramar los diseños preliminares antes de pasar a la etapa de desarrollo. Cabe destacar que pueden variar a lo largo del ciclo de vida de la aplicación.

No se tuvieron en cuenta botones de “Volver” en cada una de las pantallas, porque esta acción está contemplada en el sistema operativo de Android y todas sus implementaciones poseen esta característica. Algunos teléfonos la tienen en cuenta desde un componente físico (botón) independiente y otros más nuevos reservan una parte de su pantalla táctil para las tres acciones estándar.



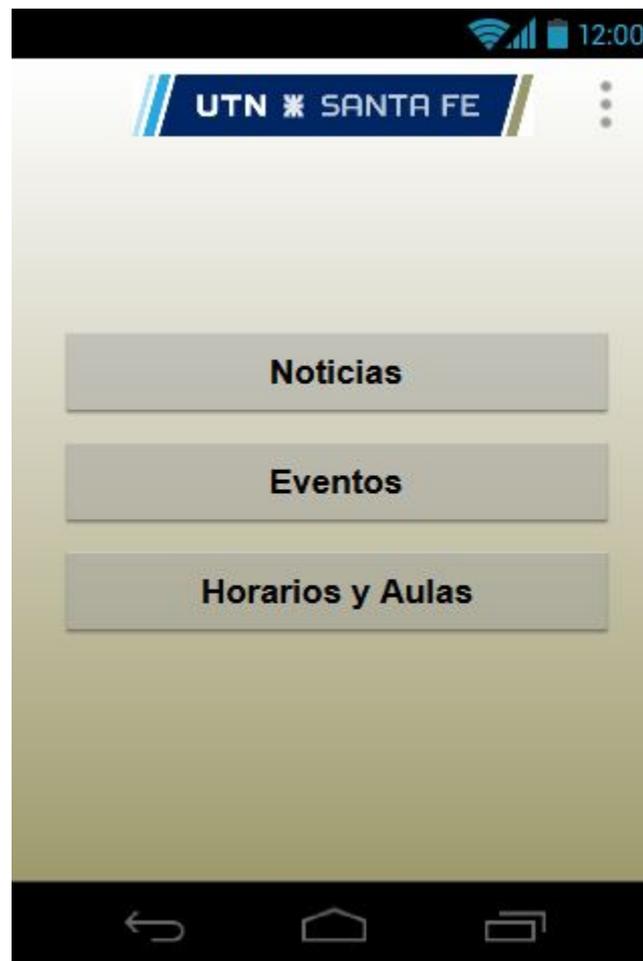
El diseño superior corresponde a versiones de Android anteriores a Lollipop.

Back: Retorna a la pantalla anterior de la aplicación en la pila de pantallas que conserva el orden en el cual fueron navegadas. Cuando se llega a la pantalla principal, se minimiza la aplicación y se muestra la pantalla principal del sistema operativo.

Home: Acceso rápido a la pantalla principal de Android

Grid: Muestra el administrador de tareas del sistema operativo, permite cambiar fácilmente de aplicación y cerrar las que no están en uso.

Pantalla Principal



Es lo primero que vemos al abrir la aplicación.

Contiene una barra de acción superior con el logo de la regional Santa Fe. A la derecha del mismo se encuentra un botón que despliega un menú contextual con más acciones relacionadas con la personalización de la aplicación y la autenticación de usuarios.

En el área principal se encuentran los botones para acceder a los módulos principales.

Noticias



Contiene un componente scrolleable con las últimas noticias ordenadas cronológicamente. Cada resumen de noticia está compuesto por el título de la misma, el nombre del canal por el que fue emitida y las primeras palabras.

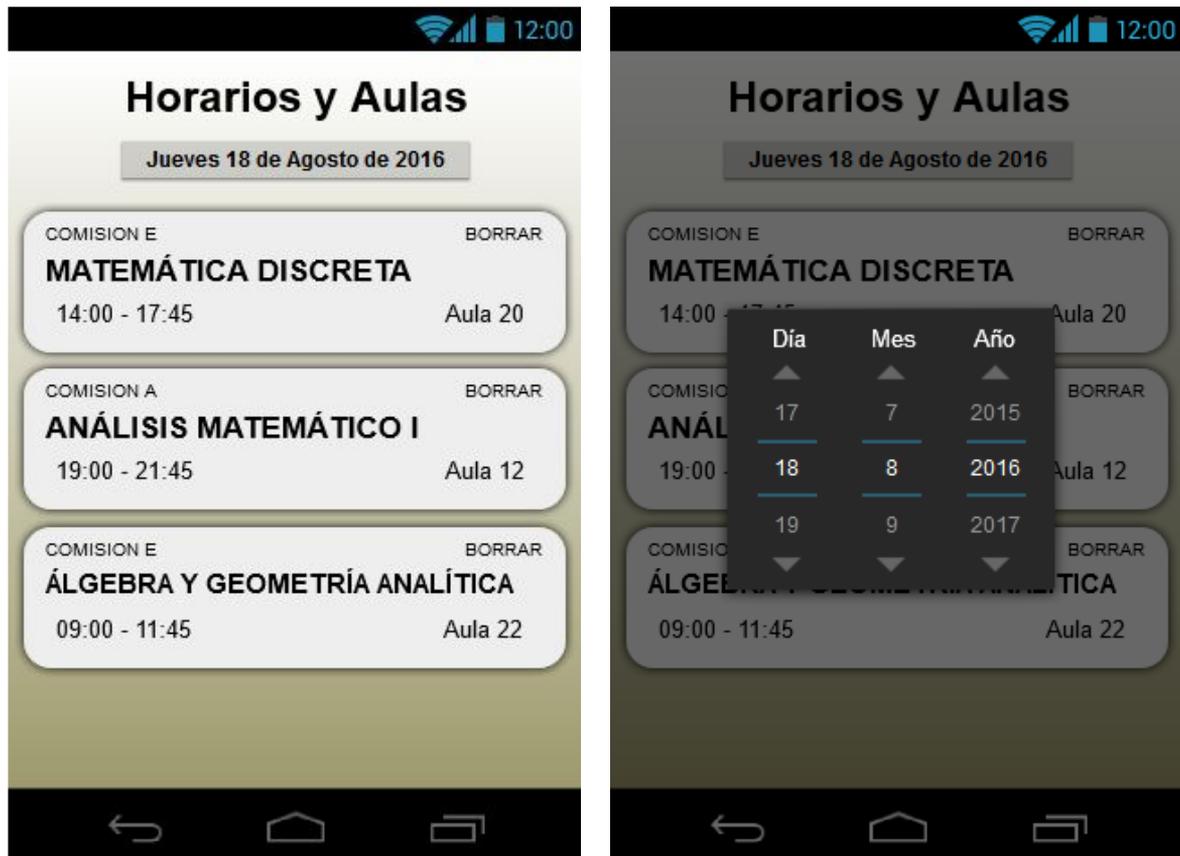
Eventos



El elemento principal de esta pantalla es el calendario con vista de todos los días del mes y con posibilidad de cambiar al siguiente mes o al previo.

Cuando seleccionamos un día nos muestra abajo el título y la hora de comienzo de los eventos asociados al mismo. Si los tiene, en ese caso estarán representados por un punto de color en el recuadro correspondiente.

Horarios y Aulas



Debajo del título vemos un botón inicializado con el texto del día de hoy, ya que la consulta más probable es preguntar por las materias de la fecha. De todas formas eso puede cambiarse a través de un selector de fecha, el cual es un componente nativo de Android. En el área principal nos muestra las respectivas comisiones de materias a las cuales nos suscribimos previamente con su horario de comienzo y finalización así también como el aula que tiene asignada. También cada contenedor tiene un botón de "BORRAR" que facilita la desuscripción de los horarios de esa comisión.

Selección de Materia



El objetivo de esta pantalla es suscribirse o dejar de hacerlo a una determinada comisión. El primer paso es seleccionar la carrera desde el componente de selección única. En el segundo paso consiste en ingresar las primeras letras de una materia en el campo de texto, lo que desplegará las sugerencias que coinciden con la entrada. Una vez seleccionada la materia nos mostrará un contenedor con las comisiones disponibles de la misma. Acto seguido podemos activar o desactivar las notificaciones relacionadas a dicha comisión. Implícitamente esta acción también nos suscribe al canal de la materia

Personalizar Canales



Desde esta pantalla podemos elegir los canales de información sobre los que queremos recibir actualizaciones. Esto le da control al usuario sobre los mismos y regula la cantidad de notificaciones que recibe a diario.

Está compuesta por una lista desplazable con los nombres de los canales y sus correspondientes interruptores que representan el estado de encendido o apagado.

Interfaz pública del servidor

Cada tabla corresponde a una función operación pública soportada por el servidor.

Operación: <i>getChannels</i>
Descripción: Retorna la lista completa de canales que existen en la base de datos. No necesita parámetros adicionales. Método: <i>GET</i>

Operación: <i>getNews?offset=<DESPLAZAMIENTO></i>
Descripción: Retorna las últimas noticias ordenadas en forma descendente por fecha filtradas por canales. Método: <i>POST</i>
Parámetros de la dirección: <DESPLAZAMIENTO> Número entero opcional que indica la cantidad de noticias que deben ser salteadas, se utiliza para poder dividir la lista de noticias en páginas. Formato: <i>URL Encoded</i>
Parámetros del cuerpo: <i>channels:</i> <CANALES>
<CANALES> Arreglo con los identificadores únicos de los canales que se quieren consultar. Formato: <i>JSON</i>

Operación: ***getEvents?month=<MES>&year=<AÑO>***

Descripción: Retorna todos los eventos de un mes especificado.

Método: ***POST***

Parámetros de la dirección:

<MES> *Número entero del mes (0 a 11).*

<AÑO> *Año completo.*

Formato: ***URL Encoded***

Parámetros del cuerpo:

channels: **<CANALES>**

<CANALES> Arreglo con los identificadores únicos de los canales que se quieren consultar.

Formato: ***JSON***

Interfaz privada del servidor

Cada tabla corresponde a una función operación privada soportada por el servidor, mantienen esta condición porque crean o modifican información en la base de datos.

<i>Operación: authenticate</i>
Descripción: Valida las credenciales para el inicio de sesión. Método: POST Parámetros del cuerpo: <i>inputUser: <USUARIO></i> <i>inputPassword: <CONTRASEÑA></i> <USUARIO> Identificador alfanumérico de la entidad autorizada. <CONTRASEÑA> Secreto conocido solamente por la entidad autorizada. Formato: JSON
<i>Operación: addNew</i>
Descripción: Agrega una noticia a la base de datos con la fecha y hora actual. Opcionalmente emite una notificación a los dispositivos que están suscritos al canal. Método: POST Parámetros del cuerpo: <i>canal: <IDENTIFICADOR DEL CANAL></i> <i>titulo: <TITULO></i> <i>cuerpo: <CUERPO DE LA NOTICIA EN FORMATO HTML></i> <i>imagen: <URL DE LA IMAGEN></i> Formato: JSON
<i>Operación: removeNew?id=<ID></i>
Descripción: Elimina una noticia de la base de datos. Método: GET <i>Parámetros de la dirección:</i> <ID> Identificador único de la noticia. Formato: URL Encoded

Operación: *addEvent*

Descripción: Agrega un evento a la base de datos. Opcionalmente emite una notificación a los dispositivos que están suscritos al canal.

Método: **POST**

Parámetros del cuerpo:

canal: <IDENTIFICADOR DEL CANAL>

titulo: <TITULO>

lugar: <LUGAR>

fecha: <FECHA DEL EVENTO>

Formato: **JSON**

Operación: *removeEvent?id=<ID>*

Descripción: Elimina un evento de la base de datos.

Método: **GET**

Parámetros de la dirección:

<ID> Identificador único del evento.

Formato: **URL Encoded**

Operación: *logout*

Descripción: Borra los cookies (información almacenada por el navegador) que sostienen la sesión segura entre el servidor y el cliente web.

Método: **GET**

Las operaciones pueden ser respondidas con alguno de los siguientes códigos que son tomados del estándar HTTP/HTTPS.

Código	Descripción
200	La operación fue completado con éxito.
401	La operación no está autorizada.
500	Ocurrió un error interno del servidor.

Detalles de la planificación

Antes de comenzar con la ejecución del proyecto, durante etapas de análisis, se listaron las metas que se esperaban cumplir en cada iteración. Las mismas están descritas en la tabla A. Actúan de guía para ordenar el trabajo pero contemplan cierta flexibilidad, pueden aparecer nuevas a medida que avanzan las iteraciones y las negociaciones con el cliente.

Tabla A. Descripción de las metas a ser alcanzadas en cada iteración.

Iteración	Metas
1	<ul style="list-style-type: none">1.1 Investigación de tecnologías a utilizar.1.2 Definir la arquitectura a utilizar.1.3 Diseño del servidor de backend.1.4 Diseño de la estructura de la aplicación móvil.1.5 Implementación del servidor de backend.1.6 Implementación de la base de datos orientada a documentos.1.7 Implementación de la estructura de la aplicación móvil.1.8 Conexión básica del backend con la aplicación móvil.1.9 Prueba y retroalimentación.
2	<ul style="list-style-type: none">2.1 Definición del modelo semántico de datos.2.2 Refinamiento del diseño del servidor del backend y base de datos.2.3 Diseño de los menús e interfaces básicas de la aplicación móvil.2.4 Implementación de los menús e interfaces básicas de la aplicación móvil.2.5 Autenticación segura para la generación de contenidos.2.6 Desarrollo del primer módulo de noticias.2.7 Desarrollo de las notificaciones en tiempo real.2.8 Prueba y retroalimentación.
3	<ul style="list-style-type: none">3.1 Análisis de los resultados de las pruebas y devoluciones de los usuarios.3.2 Rediseño según las devoluciones de los usuarios.3.3 Desarrollo del módulo de calendario de actividades.3.4 Desarrollo del módulo que conecta con el sistema <i>¿Dónde Curso?</i>3.5 Pruebas de integración.3.6 Despliegue del sistema.

La unidad de tiempo utilizada son semanas de trabajo. La disponibilidad horaria de cada persona que realizará tareas de análisis, diseño, codificación y pruebas es de 25 horas semanales. Al disponerse de dos recursos humanos equivaldría a un total de 50 horas de trabajo semanales, que es lo que aproximadamente representa cada cuadro del diagrama de Gantt en la tabla B. Notar que algunas tareas están superpuestas con otras y todavía no se las ha asignado a cada recurso, esto se dejó para el momento de la ejecución según la disponibilidad de cada recurso.

Tabla B. Diagrama de Gantt con las metas planificadas.

Iteraciones	Primera Iteración							Segunda Iteración							Tercera Iteración					
Etapas \ Semanas	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Análisis	1.1							2.1								3.1				
Diseño		1.2	1.3	1.4					2.2	2.3						3.2				
Desarrollo				1.5	1.6	1.7	1.8				2.4	2.5	2.6	2.7			3.3	3.4		
Pruebas						1.9								2.8					3.5	3.6
Informe Final																				

Plazos reales

Al cumplirse cada meta se fue contabilizando el tiempo insumido en cada una para luego determinar si estaba dentro de los plazos estipulados. En esta instancia sí se dividieron las tareas a cada recurso humano. Siendo el “Recurso 1” quien centraría sus tareas en el desarrollo de la aplicación móvil y el “Recurso 2” quien trabajaría para completar las tareas asociadas a el servidor de back-end y el sistema web para el personal del Área de Comunicaciones de la FRSF. Las tareas de análisis, diseño y pruebas se realizaron en forma conjunta en gran parte de los casos.

En las siguientes tablas se ve reflejado el cumplimiento de las metas correspondientes a cada etapa, luego se calcula la desviación en base a lo planificado con anterioridad.

Como se mencionó anteriormente, cada semana de trabajo representa 50 horas aproximadamente. Al subdividir las semana en dos, cada columna equivale a 25 horas y esto implica 12 horas y media de cada recurso por cuadro de la grilla.

Primera Iteración

Hubo sobreestimación en la etapa de análisis sobre la investigación de tecnologías y también en la implementación de la estructura general de la aplicación. Esto produjo que la iteración concluya antes y se comience la siguiente en la semana 7.

		Primera Iteración						
Etapas \ Semanas		1	2	3	4	5	6	7
Análisis	Recurso 1	1.1						
	Recurso 2	1.1						
Diseño	Recurso 1		1.3	1.4				
	Recurso 2		1.2					
Desarrollo	Recurso 1				1.7	1.8		
	Recurso 2			1.5	1.6			
Pruebas	Recurso 1						1.9	
	Recurso 2					1.9		

Metas Iteración 1 (En semanas)	Planificado	Real	Desviación
1.1 Investigación de tecnologías a utilizar.	1,5	1	-0,5
1.2 Definir la arquitectura a utilizar.	0,5	0,5	0
1.3 Diseño del servidor de backend.	0,5	0,5	0
1.4 Diseño de la estructura de la aplicación móvil.	0,5	0,5	0
1.5 Implementación del servidor de backend.	1	1	0
1.6 Implementación de la base de datos orientada a documentos.	0,5	0,5	0
1.7 Implementación de la estructura de la aplicación móvil.	1	0,5	-0,5
1.8 Conexión básica del backend con la aplicación móvil.	0,5	0,5	0
1.9 Prueba y retroalimentación.	1	1	0
Totales	7	6	-1

Segunda Iteración

Esta comienza una semana antes de lo previsto, pero luego será necesaria para completar tareas asociadas a la autenticación y módulo de noticias. También se creó la sección “Acerca De” de la aplicación móvil, la cual era una meta no contemplada en la planificación original. La fecha de terminación se extendió a la semana siguiente de la esperada.

		Segunda Iteración									
Etapas \ Semanas		7	8	9	10	11	12	13	14	15	
Análisis	Recurso 1	2.1									
	Recurso 2	2.1									
Diseño	Recurso 1		2.3								
	Recurso 2		2.2								
Desarrollo	Recurso 1			2.4			2.6			2.9	
	Recurso 2			2.5			2.7				
Pruebas	Recurso 1									2.8	
	Recurso 2							2.8			

Metas Iteración 2 (En semanas)	Planificado	Real	Desviación
2.1 Definición del modelo semántico de datos.	1,5	0,5	-1
2.2 Refinamiento del diseño del servidor del backend y base de datos.	0,5	0,5	0
2.3 Diseño de los menús e interfaces básicas de la aplicación móvil.	0,5	0,5	0
2.4 Implementación de los menús e interfaces básicas de la aplicación móvil.	1,5	1,5	0
2.5 Autenticación segura para la generación de contenidos.	0,5	1,5	1
2.6 Desarrollo del primer módulo de noticias.	1	1,5	0,5
2.7 Desarrollo de las notificaciones en tiempo real.	1	1	0
2.9 Desarrollo de la sección Acerca De.	0	0,5	0,5
2.8 Prueba y retroalimentación.	1,5	1,5	0
Totales	8	9	1

Tercera Iteración

En esta iteración aparecieron nuevas metas que surgieron de las negociaciones con el cliente. Entre las mencionadas están el módulo “Cursos”, el de “Búsquedas Laborales” y la integración con las redes sociales oficiales de la UTN FRSF.

		Tercera Iteración														
Etapas \ Semanas		16	17	18	19	20	21									
Análisis	Recurso 1	3.1														
	Recurso 2	3.1														
Diseño	Recurso 1		3.2													
	Recurso 2		3.2													
Desarrollo	Recurso 1			3.3			3.4									
	Recurso 2			3.7	3.8											
Pruebas	Recurso 1															
	Recurso 2							3.5	3.6							

Metas Iteración 3 (En semanas)	Planificado	Real	Desviación
3.1 Análisis de los resultados de las pruebas y devoluciones de los usuarios.	0,5	0,5	0
3.2 Rediseño según las devoluciones de los usuarios.	0,5	0,5	0
3.3 Desarrollo del módulo de calendario de actividades.	1	1	0
3.4 Desarrollo del módulo que conecta con el sistema ¿Dónde Curso?	1	1,5	0,5
3.7 Desarrollo de módulos Cursos y Búsquedas Laborales.	0	0,5	0,5
3.8 Integración con redes sociales.	0	0,5	0,5
3.5 Pruebas de integración.	1	0,5	-0,5
3.6 Despliegue del sistema.	1	1	0
Totales	5	6	1