



Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

DOCUMENTO DE DISEÑO - PARTE MÓVIL

Nombre del proyecto:
Organización

SmartBoard Project Manager
Pontificia Universidad Católica de
Chile

Fecha:
Versión:

27 de noviembre, 2013
2.0

Historia del Documento

Versión	Fecha	Autor(es)	Razón del Cambio
0.1	31/10/13	José Tomás Marquinez	Primer borrador
1.0	01/11/13	José Tomás Marquinez	Versión terminada
2.0	27/11/13	José Tomás Marquinez	Versión actualizada al Sprint 3, terminada

Equipo de Desarrollo

Nombres y Apellidos	Rol	Contacto
Nicolás Risso	Administrador del Proyecto	narisso@puc.cl (56 9) 8818-6497
Valentina Ibaseta	Desarrollador/Analista	vjibaset@puc.cl (56 9) 9497-5956
José Tomás Marquinez	Desarrollador/Analista	jtmarquinezv@puc.cl (56 9) 9020-0720
Thomas Büchi	Desarrollador/Diseñador	tbuchi@puc.cl (56 9) 9599-6990
Santiago Larraín	Desarrollador/Diseñador	slarrain@puc.cl (56 9) 8248-2759
Nicolás Escobar	Desarrollador/Tester	niescoba@puc.cl (56 9) 8824-6141
Fernando González	Desarrollador/Tester	fagonza6@puc.cl (56 9) 6727-1956

Contraparte del Proyecto

Nombres y Apellidos	Rol	Contacto
José Ignacio Benedetto	Estudiante	jibenedettoc@gmail.com (56 2) 2354-2000
Andrés Chacón	Estudiante	afchacon2@gmail.com (56 2) 2354-2000

Tabla de Contenidos

Historia del Documento	i
Equipo de Desarrollo	ii
Contraparte del Proyecto	ii
1. Descripción General	1
1.1 Propósito del Sistema.....	1
1.2 Alcance del Proyecto	2
1.3 Contexto	3
1.4 Definiciones, Acrónimos y Abreviaturas	4
1.5 Referencias.....	5
2. Vista Lógica.....	6
2.1.1 Arquitectura del Sistema.....	6
2.1.2 Arquitectura Lógica	8
2.1.2.1 Interfaz de Usuario	8
2.1.2.2 Servicios de Sistema	9
2.1.2.3 Infraestructura	9
2.1.3 Arquitectura de los Módulos.....	10
2.1.3.1 Interfaz de Usuario	10
2.1.3.2 Servicios de Sistema	20
3. Vista de Implementación	23
3.1.1 Estructura de la Aplicación.....	23
3.1.2 Arquitectura de Implementación.....	23
4. Vista de Datos.....	24
5. Vista de Deployment	25
5.1.1 Tecnología requerida.....	25
5.1.2 Deployment.....	25

Tabla de Ilustraciones

Ilustración 1: Protocolo estrella utilizado para la arquitectura de red	6
Ilustración 2: Relación entre los módulos de la parte móvil de la aplicación	7
Ilustración 3: Relación de los módulos de la parte móvil con la API de la parte Web, que entrega WebServices	7
Ilustración 4: Dependencias entre los módulos de la aplicación móvil	8
Ilustración 5: Dependencia entre los casos de uso móvil	9
Ilustración 6: Interfaces de la aplicación móvil para el sistema de conexión y de dibujo.	9
Ilustración 7: Controlador Actividades No Samsung	9
Ilustración 8: Controlador del editor de dibujo	10
Ilustración 9: Flujo regular de la aplicación	11
Ilustración 10: Interfaz de la pantalla de logueo	12
Ilustración 11: Interfaz para escoger proyecto	13
Ilustración 12: Interfaz de la pizarra de dibujo	14
Ilustración 13: Opciones de módulos de la parte del canvas	14
Ilustración 14: Interfaz de la creación de red	15
Ilustración 15: Flujo del módulo de crear red	16
Ilustración 16: Interfaz de dibujo, instanciado en un ejemplo	16
Ilustración 17: Diagrama de opciones del módulo de dibujar	17
Ilustración 18: Presentación de las funcionalidades alternativas de los botones	17
Ilustración 19: Interfaz del módulo de sincronización con Dropbox, de cargar un archivo (izquierda) o guardar un archivo (derecha)	18
Ilustración 20: Opciones adicionales abiertas, luego de apretar el botón de opciones del Tablet. .	19
Ilustración 21: Módulo de Comentarios de dibujos, de ver comentarios (izquierda) o editar comentario (derecha)	20
Ilustración 22: Comunicación entre aplicación móvil y los WebServices	21
Ilustración 23: Comunicación entre el módulo de Sincronizar con Dropbox con el WebService	22
Ilustración 24: Diagrama de módulos y su dependencia de los archivos xml	23

1. Descripción General

Actualmente la mayoría de los organismos utilizan herramientas computacionales para su operación. Esto, con el objetivo de optimizar procesos, facilitar la comunicación y las tareas dentro de la institución, entre otros. Del mismo modo, las personas han comenzado a utilizar varias aplicaciones y software para organizar personalmente su día a día, lo que ha tenido como consecuencia un constante surgimiento de nuevas necesidades tecnológicas en los usuarios. Es por esto que se requiere crear nuevas herramientas y aplicaciones para satisfacer la creciente demanda.

No obstante, la creación de nuevas aplicaciones no es un trabajo sencillo. La mayoría de los casos requiere un equipo de desarrollo para ello, el que, a su vez, requiere de mucha coordinación y cooperación para poder realizar un buen trabajo en los plazos estimados. El desarrollo de un software está constituido de varias tecnologías y etapas: definición de requisitos de usuario, casos de uso, estructura de la aplicación, modelo de datos, tareas individuales de cada una de las personas en el desarrollo, entre otras cosas. Para abordar cada una de estas etapas en el desarrollo, existen distintas herramientas como GitHub, Dropbox, Kanbanery, Gmail; que permiten la comunicación, coordinación y trabajo confluente de cada una de las partes. Sin embargo, cada una funciona por separado. Esto entrega una dificultad adicional al momento de usarlas de manera integrada para la coordinación y comunicación del equipo de trabajo y con el cliente.

El objetivo de este proyecto es crear una herramienta que permita y facilite el desarrollo de software. Integrará aplicaciones para el manejo de versiones, compartición de archivos, asignación y monitoreo de proyectos, y creación de documentos de caso de uso y requisitos, entre otros.

1.1 Propósito del Sistema

El sistema busca proveer una herramienta open source que permita al equipo de desarrollo llevar un seguimiento constante de su trabajo, de forma que pueda manejar los distintos aspectos de la construcción de su proyecto. El software, aparte de permitir el manejo de versionamiento que es indispensable para un grupo de trabajo, debe integrar secciones para el manejo de documentos como los casos de usos que estén asociados a diferentes requisitos del usuario. Así, se podrá generar constantemente tareas en base a dichos archivos.

Se pretende que la aplicación sea capaz de facilitar la coordinación y ejecución del equipo de desarrollo, así como también la comunicación con el cliente. Estos aspectos son muy relevantes, pues si estos aspectos no logran manejarse de manera adecuada, pueden ser la principal causa de retrasos en la implementación, o incluso de la reestructuración de un proyecto. Se busca generar valor en las reuniones de los clientes de los proyectos que vayan a usar el nuevo sistema, de manera que se le permita al cliente expresar sus ideas sin limitaciones computacionales, a través de un ambiente de trabajo de tablets sincronizados. De esta manera, los clientes, aprovechando las características de los dispositivos, podrán generar requisitos y casos de uso extraídos de lo que el lenguaje computacional exige.

SmartBoard Project Manager debe ser capaz de integrar todos los aspectos que son relevantes para el desarrollo de un proyecto, para que sea un medio de gestión y desarrollo. Ésta debe estar apuntada a cualquier equipo de desarrollo y debe ser simple en su utilización, pues debe facilitar el trabajo y no convertirse en un obstáculo más. Además, se desea que la arquitectura sea tal que pueda ser usada por equipos de estudiantes como medio de aprendizaje y experiencia de trabajo en un desarrollo de software real.

En resumen, se generará un ambiente que busca aumentar la usabilidad para el seguimiento del avance de cualquier proyecto. En particular, estará enfocado en proyectos de desarrollo de software, al buscar estar integrado con diversas plataformas y servicios webs tecnológicos que favorecen la integración de códigos, archivos, calendarios, entre muchos otros. Por otro lado, su ambiente móvil estará integrado fuertemente con el área web, de manera que las reuniones tendrán un valor agregado para el cliente y para el equipo del proyecto.

1.2 Alcance del Proyecto

Las principales funcionalidades que se buscan integrar dentro de la aplicación son las siguientes:

- Sistema de versionamiento de código
- Bug Tracking.
- Project Tracking.
- Testing Management.
- Task Management.
- Agenda y calendario.
- Sistema de administración de mensajes, que permita enviar mails.
- Sistema de archivos en la nube.
- Wiki interna.
- Módulo de documentos, con la posibilidad de exportar a PDF.
- Storyboard sobre el flujo de documentos.
- Herramienta móvil para diagramar.
- API para el login con redes sociales.

Así, el proyecto consiste en integrar cada uno de estos sistemas en una aplicación. Ésta estará constituida por una parte web y otra parte móvil. La primera contempla la parte principal, pues tendrá la mayoría de las funcionalidades. De esta manera, la aplicación móvil estará enfocada en tareas de diagramación para documentos o mock-ups, en las reuniones con clientes.

La primera versión de la herramienta constará de un avance integrado entre las dos partes mencionadas. En lo que respecta de la parte web, ésta consistirá en una plataforma de administración de proyectos que facilite un seguimiento de las tareas o actividades realizadas,

realizándose y por realizar. De esta manera, sus usuarios en sus distintos roles podrán trabajar en la gestión de proyectos para la facilitación de la visualización del grado de avance del proyecto. Permitirá personalizar un tablero de tareas que dispondrá de las distintas actividades registradas en los diversos grados de avance o columnas que dispongan. De esta forma, los usuarios podrán informarse sobre las horas dedicadas en cada tarea, el flujo que han tenido las tareas en el transcurso del tiempo, los comentarios realizados para cada una de las tareas, los responsables de llevar a cabo las tareas, los archivos utilizados o generados para respaldar una actividad. Como la herramienta busca facilitar el trabajo en proyectos de desarrollo, la aplicación estará integrada con un sistema de administración de código fuente como Git. A su vez, el manejo de archivos estará sincronizado y respaldado por la integración que tendrá con un sistema para guardar archivos en la nube.

Para futuros alcances, la sección web de la herramienta podrá incluir módulos de bug tracking, ambientes de testing management, integración con calendarios como Google Calendar, sistema de administración de mensajería, un módulo de documentación de proyectos de la forma de una wiki interna, una API adecuada para el login con redes sociales como Google+, entre otras.

Por otro lado, la primera versión de la parte móvil estará dedicada únicamente a agregarle valor a la generación de casos de uso y requerimientos de los clientes. De esta manera, las reuniones entre el grupo de desarrollo y los clientes contarán con un ambiente de trabajo integrado en que cada integrante de la reunión podrá respaldar, a través de un dispositivo móvil, lo que se esté conversando y dibujando en las reuniones. En otras palabras, la aplicación no será una herramienta para la administración de proyectos. Contará con un ambiente de dibujo sincronizado entre todas las tablets, que permitirá generar diagramas libres y UML. A su vez, se podrá conectar al ambiente de sincronización un Smart TV, pues se sincronizará con un dispositivo conectado a un televisor o proyector que mostrará un lienzo general de lo conversado en las reuniones. Por último, el ambiente estará sincronizado mediante un sistema para guardar archivos en la nube con la parte web, por lo que el trabajo de cada reunión quedará respaldado para su trabajo y modificación en el ambiente web de la herramienta.

1.3 Contexto

SmartBoard Project Manager (SPM) busca apoyar y facilitar la manera en que actualmente se planifican los proyectos de desarrollo de software. Actualmente, los distintos software y servicios existentes en el mercado satisfacen sólo de forma parcial todas las necesidades que pueden surgir a la hora de gestionar un proyecto. Esto exige trabajar con más de uno para lograr tener un gran abanico de características a utilizar. Por ejemplo, para coordinar tareas que llevan a cabo un proyecto más general se deben utilizar herramientas como Kanbanery o Trello; pero para realizar storyboarding se deben utilizar servicios como Fieldtest y Luzmy.

Con la integración de diversos servicios que ayudan en el desarrollo de un proyecto y con la adición de una gran variedad de características que le agregan usabilidad, se pretende construir un sistema que satisfaga gran parte de las necesidades que surgen en el desarrollo de cualquier proyecto, principalmente de desarrollo de software. Así, se busca que sea eficiente en cuanto a la cantidad de plataformas que deben ser utilizadas para manejar correctamente el funcionamiento y avance de un proyecto.

Es por esta razón que los clientes buscan crear un sistema web y móvil que ayude la forma actual de llevar a cabo este tipo de proyectos. Buscan desarrollar desde cero una plataforma de forma sistemática, planificada y que cumpliera con características mínimas para un correcto grado de usabilidad e integración entre las funcionalidades que se deseen, con la finalidad de ocupar la menor cantidad de software disponibles en el mercado posible. Además, que sea escalable a otros proyectos fuera de lo que ellos actualmente realizan, a futuro.

Así, este nuevo sistema funciona a través de una plataforma web que asegura compatibilidad entre varios navegadores y permite organizar un proyecto desde la nube. Esta plataforma se construye sobre el framework RubyOnRails, el más utilizado actualmente para el desarrollo de sistemas por su gran variedad de gemas que permite integrar un gran número de funcionalidades requeridas. A su vez, la sección móvil de SPM trabaja sobre Android nativo aprovechando el SDK de los lápices que los tablets a utilizar incluyen. Así, se asegura una integración fácil con los dispositivos Android Mini PC, que permitirá una pantalla compartida en los ambientes de trabajo. La aplicación móvil que se desarrollará significará un gran avance tecnológico en el ambiente de reuniones que generan los requisitos de las distintas reuniones. Como se busca la integración con la pantalla compartida, el lenguaje utilizado tiene todas las bondades y ventajas que el ambiente genera, permitiendo un entorno sincronizado de trabajo en que todo queda registrado digitalmente de manera casi instantánea. Otra de las razones que favorecen la utilización de dispositivos Android son su popularidad y economía frente a otras alternativas.

1.4 Definiciones, Acrónimos y Abreviaturas

- **Administrador:** Es quien maneja muchos proyectos, y tiene interés en ver el avance y trabajo realizado por cada equipo de trabajo.
- **API:** Interfaz de programación de aplicaciones, en sus siglas en inglés.
- **Bug Tracking:** Sistema de seguimiento de errores de una aplicación informática diseñado para ayudar a asegurar la calidad de software.
- **Cliente:** Es la parte interesada en que el proyecto se lleve a cabo, porque le traerá valor una vez terminado.
- **Desarrollador:** Es quien implementa la aplicación o proyecto. Puede ser tester, desarrollador de código, analista o arquitecto.
- **Git:** Software de sistema de gestión de código fuente.
- **Github:** una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git
- **Jefe de Proyecto:** Es el responsable de que un proyecto se lleve a cabo adecuadamente.
- **Kanban:** Se entiende como el sistema de información que muestra todas las tareas necesarias para llevar a cabo un proyecto, y el estado en el que se encuentra cada tarea, para controlar el avance de dicho proyecto. Está formado por varias columnas que indican el estado de avance de que tiene cada tarea o actividad
- **SDK:** Kit de desarrollo de software, en sus siglas en inglés.

- **Storyboard:** Se entiende como un dibujo libre, que podría utilizarse para mostrar el flujo de un web.

1.5 Referencias

1. Heroku (www.heroku.com/features)
2. Ruby on Rails: Documentation (www.rubyonrails.org/documentation)
3. Guía de estilo Rails (www.github.com/bbatsov/rails-style-guide)
4. The Elements of UML 2.0 Style, Scott W. Amber (2005), Primera edición.
5. IIC2154 2013-2 (Grupo 2), Documento de Perfil de Proyecto de Especialidad (v1.0)

2. Vista Lógica

Esta vista presenta tres niveles de arquitectura para la aplicación móvil del Sistema de SmartBoard Project Manager. Cada nivel corresponde a un refinamiento del nivel anterior.

2.1.1 Arquitectura del Sistema

La aplicación móvil se encuentra instalada sobre sistema operativo Android y utilizando la librería SPen de Samsung. Ésta considera una aplicación de sistema Android 4.1 API 16. El sistema funciona de dos modos distintos: uno que permite dibujar si es que contiene la librería SPen incluido en el dispositivo móvil, y otro en que sólo permite visualizar el trabajo.

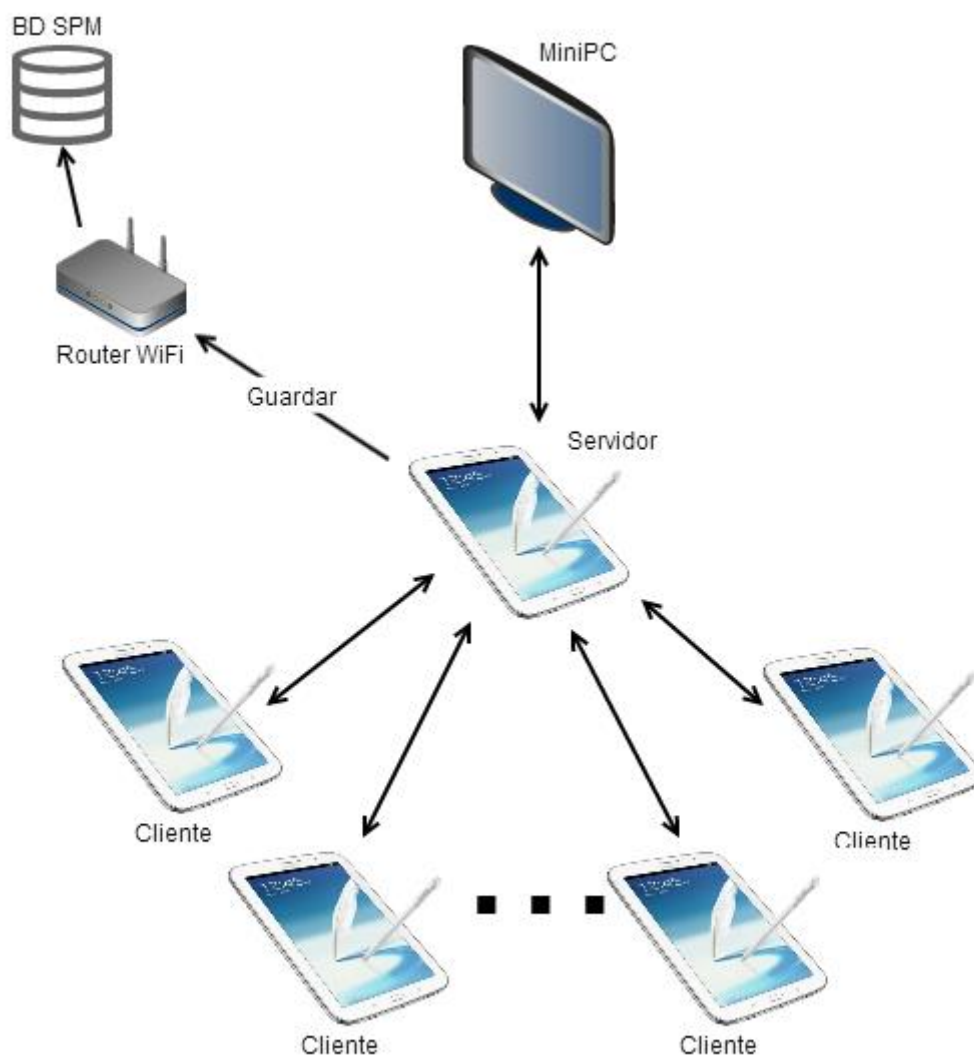
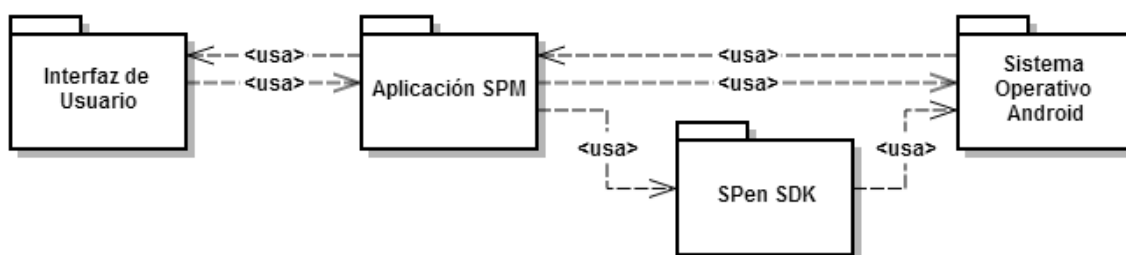


Ilustración 1: Protocolo estrella utilizado para la arquitectura de red
Fuente: Elaboración propia.

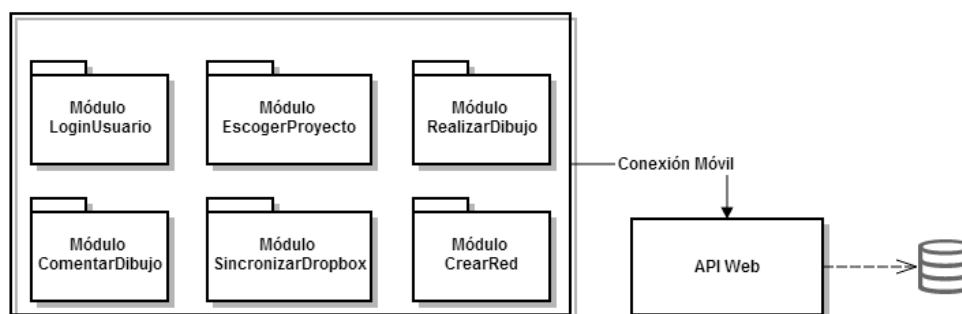
El sistema funciona con una sola aplicación, con el que interactúan diversos tablets. Todos ellos se pueden o no conectar entre sí para generar dibujos o storyboards. La arquitectura de red empleada para la conexión sigue el protocolo estrella, en que un Tablet actúa como servidor y el resto como cliente. Es el mismo servidor quien se encarga de la sincronización también de las imágenes con un dispositivo MiniPC opcional. Por otro lado, la arquitectura del software está implementada usando el patrón de diseño MVC (Modelo, Vista, Controlador), que provee la plataforma Android.



*Ilustración 2: Relación entre los módulos de la parte móvil de la aplicación.
Fuente: Elaboración propia.*

La aplicación presenta una serie de módulos que se relacionan entre sí, presentados en la Ilustración 2. Estos módulos siguen el patrón MVC, por lo que se presentan 3 capas que separan las funcionalidades del sistema, permitiendo mayor flexibilidad al encapsular funciones distintas. Las capas utilizadas son las siguientes:

- **Modelo:** Esta capa contiene la información con la que opera la aplicación. En Java, el modelo tiene la particularidad de ser reutilizable y permite hacer la aplicación escalable.
- **Vista:** Es la interfaz de la aplicación con la que interactúa el usuario. La interfaz fue construida en XML.
- **Controlador:** Es la capa encargada de la comunicación entre las dos capas anteriores. Presenta la información en la capa Vista, utilizando la información que, junto con la capa de Modelo, son obtenidas de WebServices disponibles en la API de la parte web de SmartBoard Project Manager. Los controladores son programados en lenguaje Java y son el core de la aplicación.



*Ilustración 3: Relación de los módulos de la parte móvil con la API de la parte Web, que entrega WebServices.
Fuente: Elaboración propia.*

2.1.2 Arquitectura Lógica

2.1.2.1 Interfaz de Usuario

La Vista de Casos de Uso muestra el front-end del sistema. El mismo es generado dinámicamente utilizando tecnología de contenido web dinámico. Desde el punto de vista del back-end se tiene un conjunto de páginas dinámicas generadas a partir de los procesos llevados a cabo por el sistema.

Los módulos identificados y sus interdependencias se presentan en el siguiente diagrama.

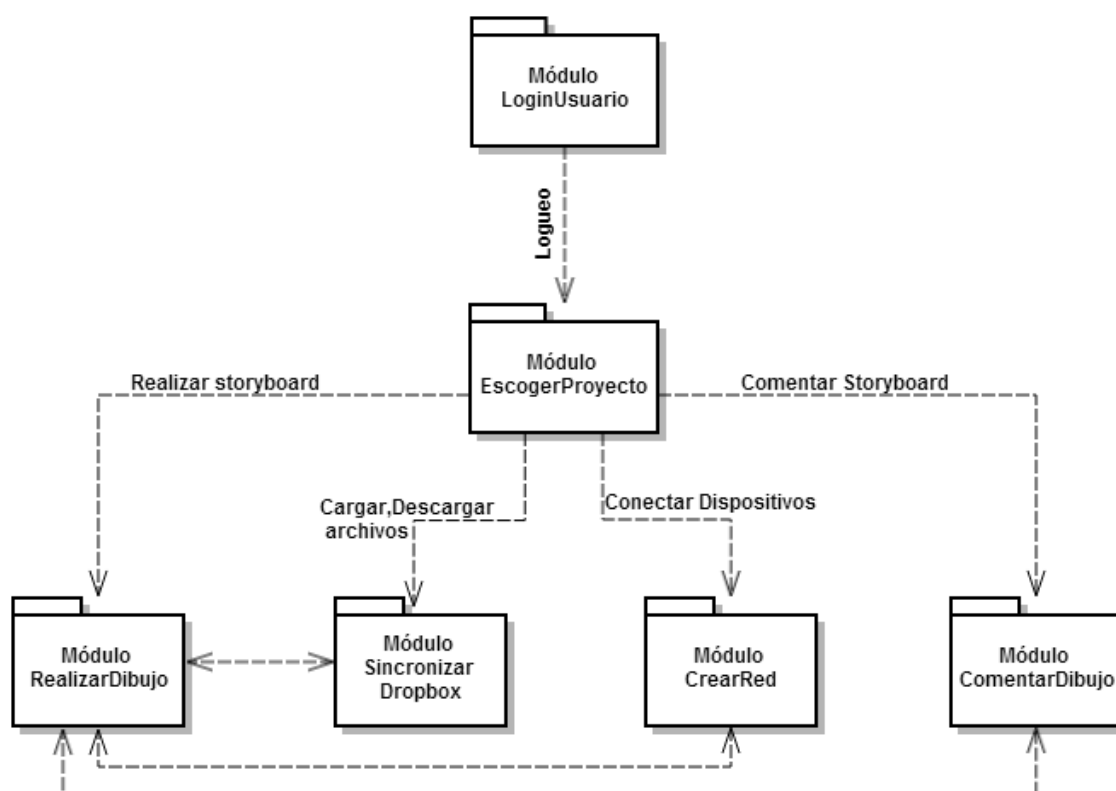
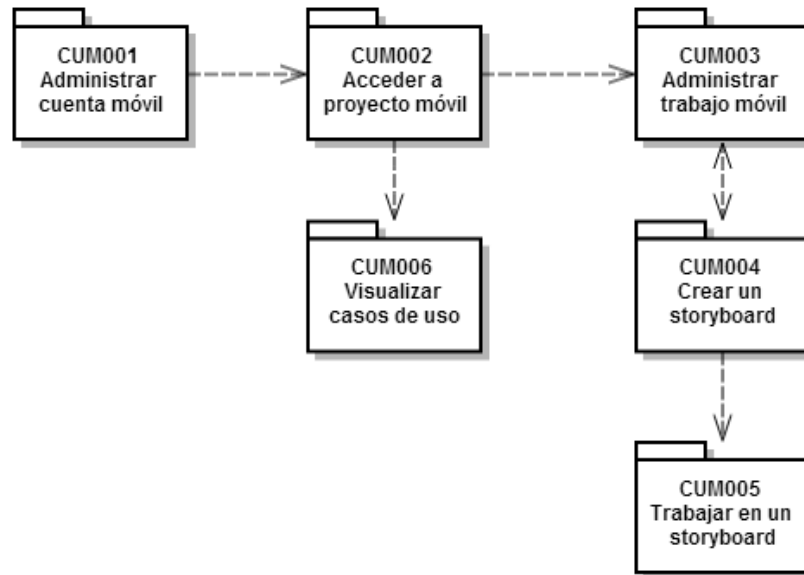


Ilustración 4: Dependencias entre los módulos de la aplicación móvil
Fuente: Elaboración propia



*Ilustración 5: Dependencia entre los casos de uso móvil.
Fuente: Elaboración Propia*

2.1.2.2 Servicios de Sistema

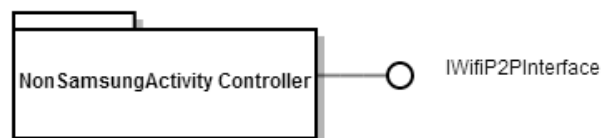
La aplicación requiere de la comunicación entre dos dispositivos utilizando la tecnología Samsung de WiFiDirect. Debido a lo anterior, se ha creado la interfaz que es base para la conexión entre dispositivos. A su vez, para reconocer el SPen de Samsung, se han utilizado las interfaces provistas por el sistema operativo, que permite reconocer el lápiz. A continuación se presentan dichas interfaces.



*Ilustración 6: Interfaces de la aplicación móvil para el sistema de conexión y de dibujo.
Fuente: Elaboración propia.*

2.1.2.3 Infraestructura

Los módulos localizados en esta capa son los controladores de la aplicación, que se encuentran disponibles para todos los módulos en las otras capas. A continuación se presentan los diagramas que relacionan los controladores con los módulos del sistema.



*Ilustración 7: Controlador Actividades No Samsung
Fuente: Elaboración Propia*

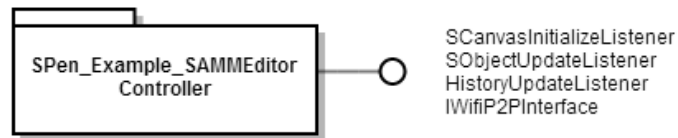


Ilustración 8: Controlador del editor de dibujo
Fuente: Elaboración Propia

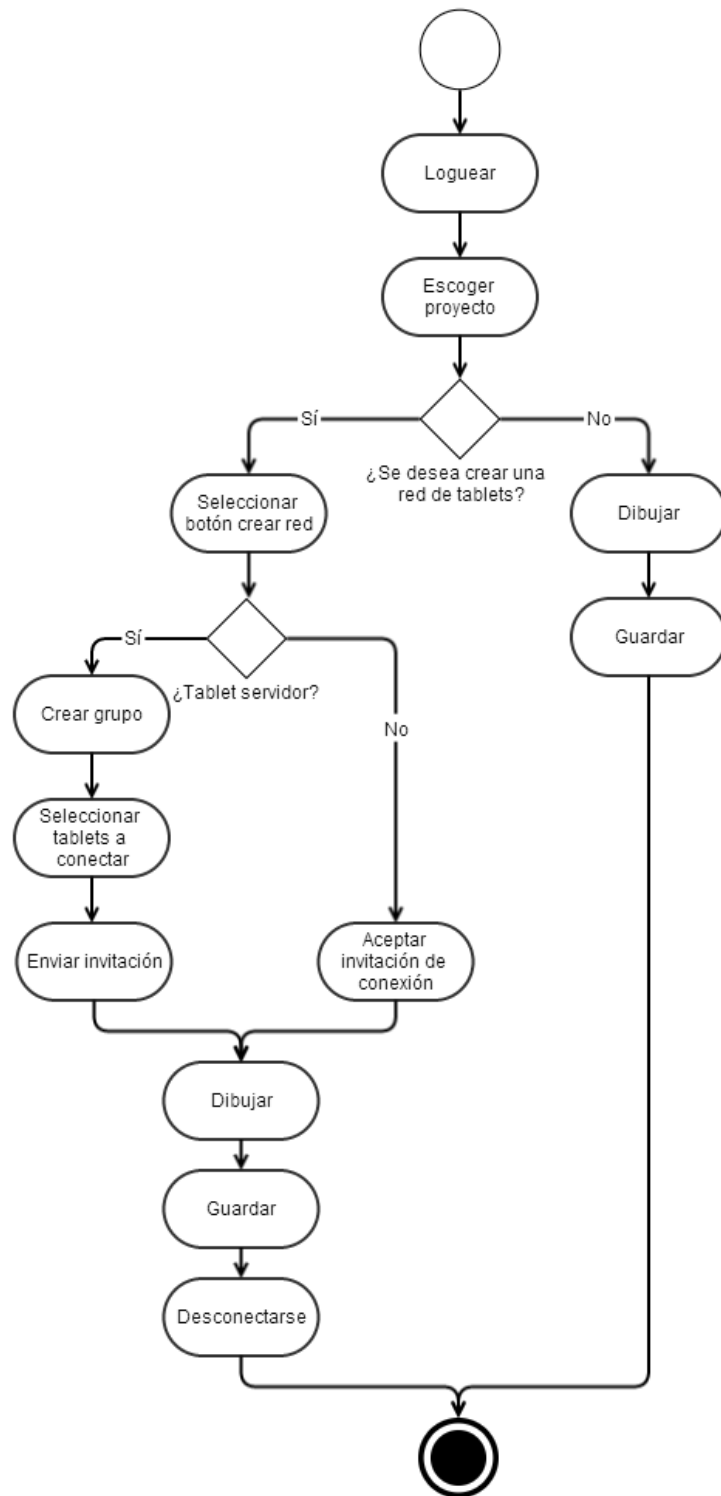
2.1.3 Arquitectura de los Módulos

La Arquitectura de los Módulos presenta un refinamiento de la Arquitectura Lógica. Esta incluye, para cada módulo, una vista que mejor define su diseño.

2.1.3.1 Interfaz de Usuario

Modelo de Navegación de la Aplicación

A continuación se presenta el modelo de navegación de un usuario tipo, que muestra el flujo regular que debería tener la aplicación y que representa una reunión. Se puede considerar esto como un ciclo, dado que esto representa la realización de un solo storyboard, pero en una reunión pueden hacerse más de uno. Adicionalmente, se muestra la interfaz de usuario para cada módulo desde el inicio de sesión hasta el guardado de un archivo.



*Ilustración 9: Flujo regular de la aplicación.
Fuente: Elaboración propia.*

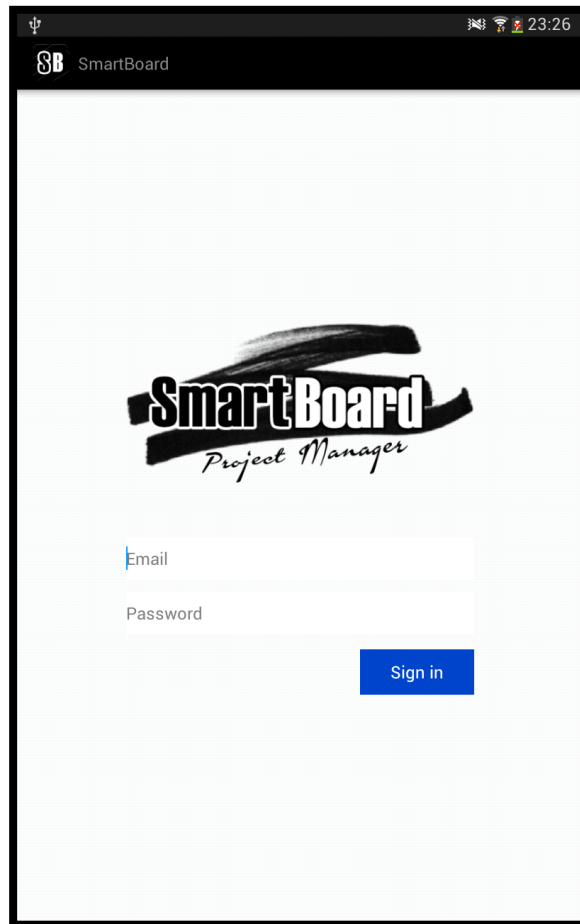
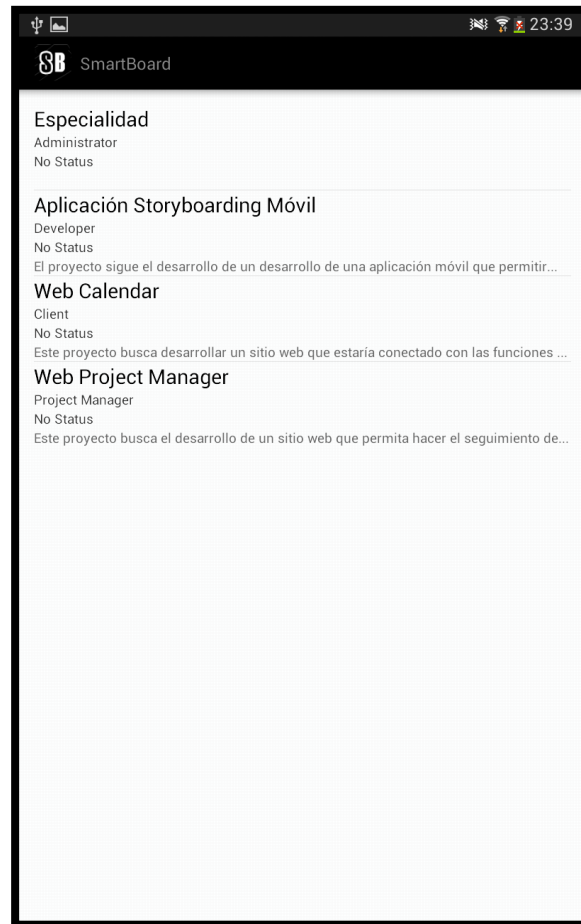


Ilustración 10: Interfaz de la pantalla de logueo
Fuente: Elaboración propia

La ilustración 10 muestra la primera imagen que surge cuando un usuario abre la aplicación sin haber iniciado sesión. Si el usuario ya inició anteriormente sesión, y no ha eliminado las cookies, esta pantalla se considera obviada y salta directamente a la ilustración 11.



*Ilustración 11: Interfaz para escoger proyecto.
Fuente: Elaboración propia*

Una vez iniciada la sesión, se enlistan todos los proyectos en los que está asignado el usuario, independiente del rol que posea. En esta ventana, se puede observar el nombre, el rol que tiene el usuario, el estado y un extracto de la descripción del proyecto.



Ilustración 12: Interfaz de la pizarra de dibujo
Fuente: Elaboración Propia

Una vez seleccionado el proyecto, el usuario verá la interfaz mostrada en la ilustración 12. En ésta, se presenta un canvas vacío, sobre el que el usuario puede trabajar. A continuación, se presenta las opciones de actividades que pueden realizarse desde el canvas, en cuanto a dibujo.

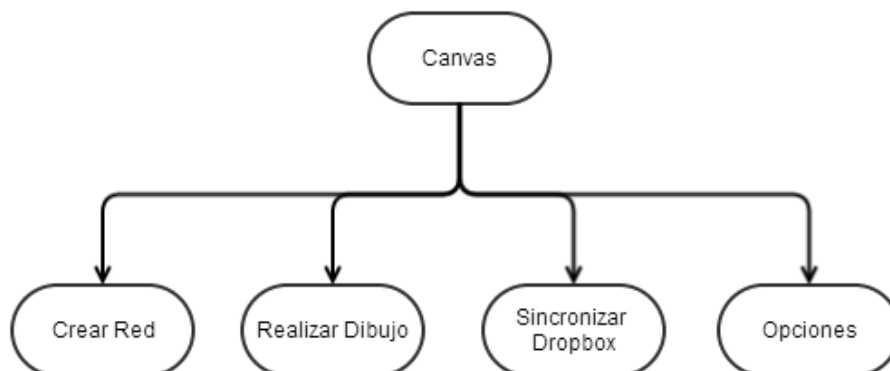
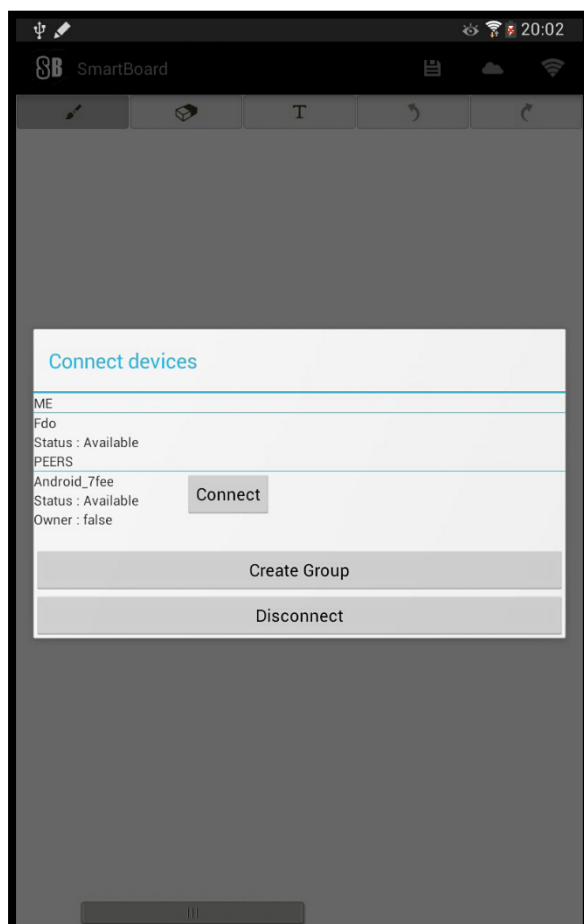


Ilustración 13: Opciones de módulos de la parte del canvas
Fuente: Elaboración propia

Módulo Crear Red

El usuario tiene la posibilidad de crear una red para que todos los tablets conectados puedan presenciar el mismo trabajo y trabajar simultáneamente sobre él. Para ello, este módulo permite que el Tablet servidor crear el grupo para luego se comiencen a enviar invitaciones de conexión para los que deseen conformar el grupo.



*Ilustración 14: Interfaz de la creación de red
Fuente: Elaboración propia*

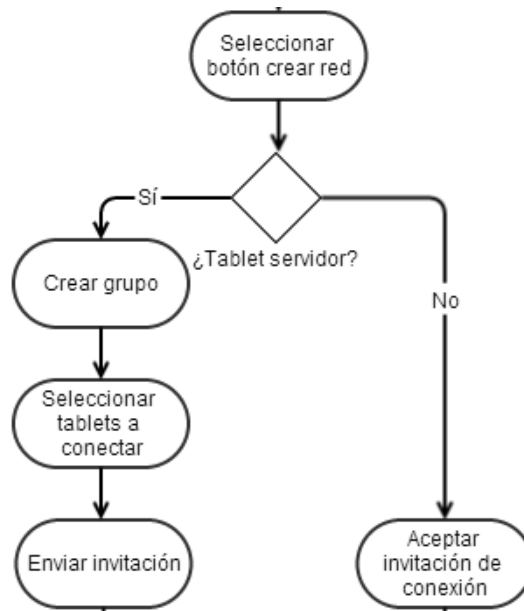


Ilustración 15: Flujo del módulo de crear red
Fuente: Elaboración propia

Módulo Realizar Dibujo

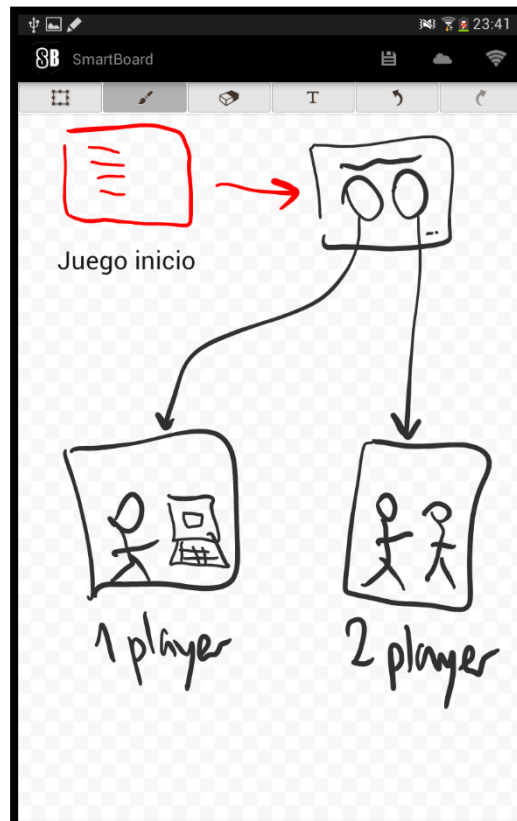
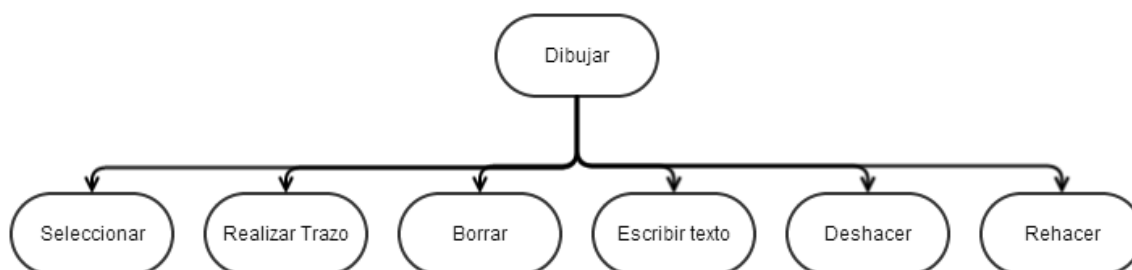


Ilustración 16: Interfaz de dibujo, instanciado en un ejemplo
Fuente: Elaboración propia

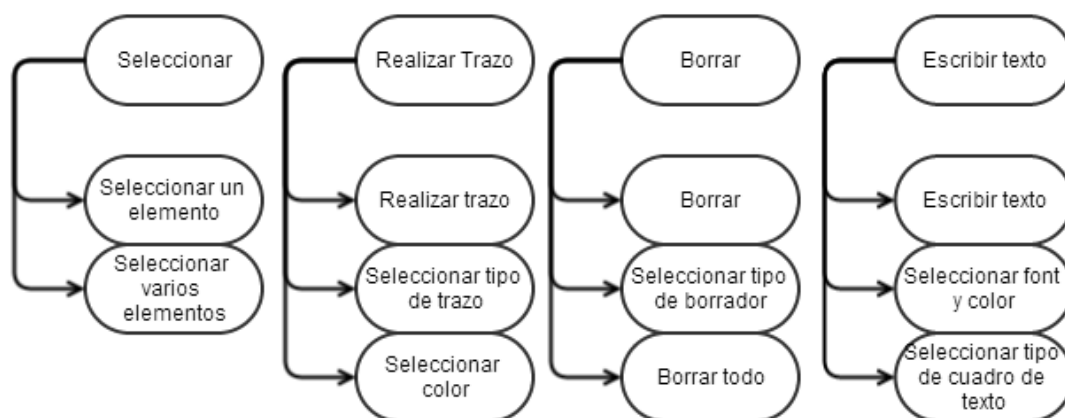
El usuario, una vez iniciada la sesión de trabajo en red (o no, si así se prefirió), puede comenzar a realizar un dibujo. También puede cargar un dibujo guardado, sin embargo, esto será explicado en el módulo de Sincronizar con Dropbox.

Existen distintas maneras de interactuar con el canvas, como se muestra en la imagen a continuación.



*Ilustración 17: Diagrama de opciones del módulo de dibujar.
Fuente: Elaboración propia*

Más en detalle, algunos botones del canvas presentan diversas funcionalidades según la interacción del usuario con este. Se puede presionar una vez, presionar dos veces, y mantener presionado, lo que permitirá ofrecer al usuario opciones de formato de la actividad seleccionada. A continuación se detallan las actividades que presentan estas funcionalidades alternativas.



*Ilustración 18: Presentación de las funcionalidades alternativas de los botones
Fuente: Elaboración Propia*

Módulo Sincronizar con Dropbox

Este módulo presenta dos opciones: cargar un archivo ya subido al repositorio de documentos en la cuenta de Dropbox asociada al proyecto, o subir un archivo realizado. Es necesario que el proyecto esté suscrito a una cuenta de Dropbox, dado que, de esa manera, se

logrará la sincronización. En caso contrario, solamente se comunicará con la memoria interna de cada Tablet en particular.

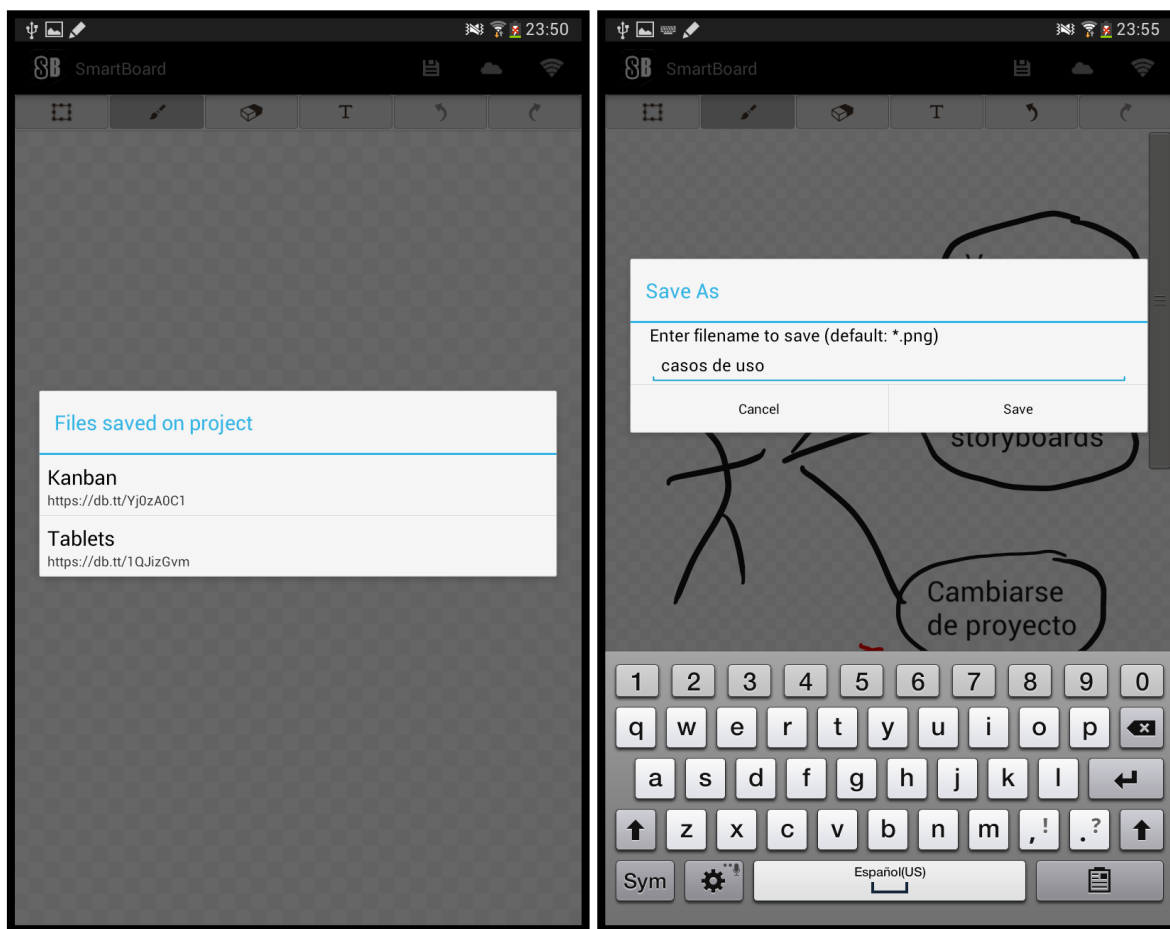


Ilustración 19: Interfaz del módulo de sincronización con Dropbox, de cargar un archivo (izquierda) o guardar un archivo (derecha)

Fuente: Elaboración propia.

Opciones adicionales

Estas opciones se obtienen al apretar el botón de opciones de los tablets. Está configurado para que permita crear un canvas desde cero, para cargar un archivo local, o para cerrar sesión. Estas opciones se muestran en la imagen a continuación.

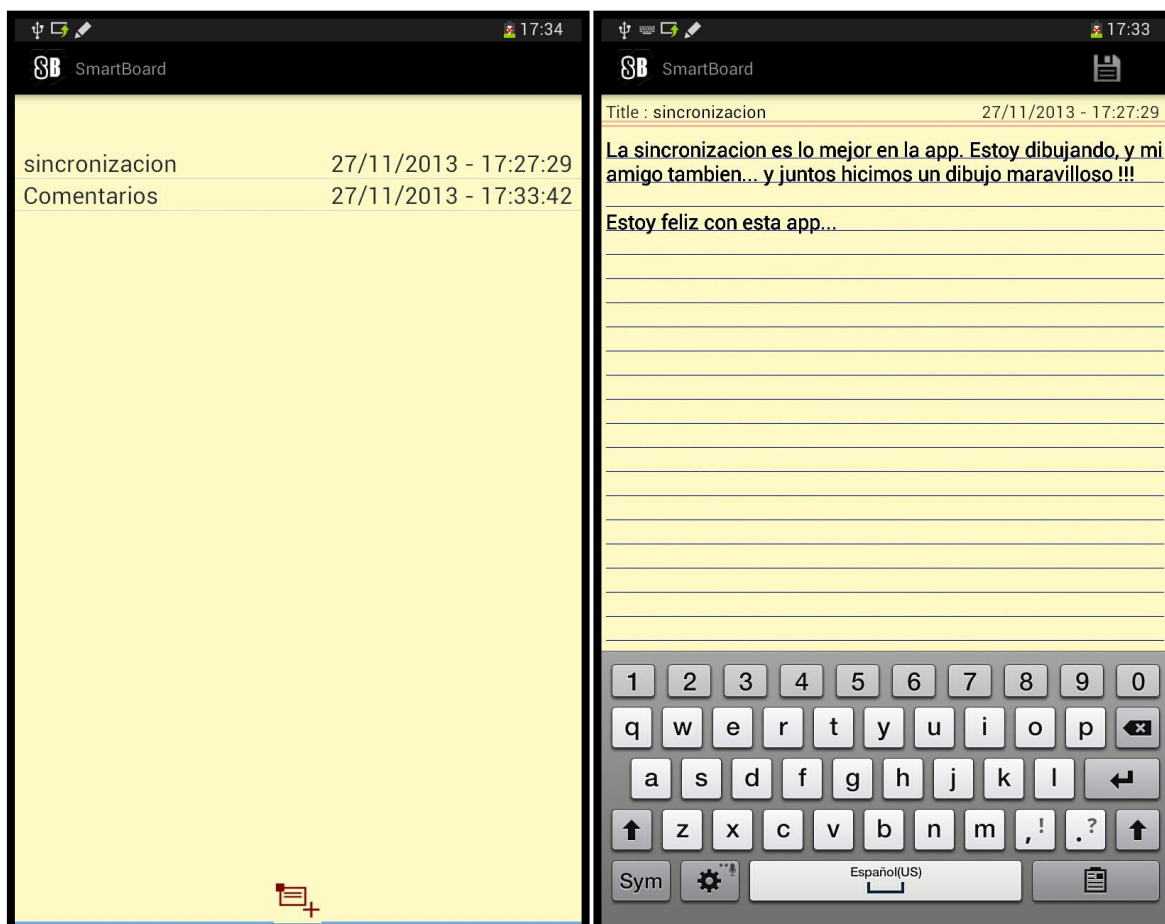


*Ilustración 20: Opciones adicionales abiertas, luego de apretar el botón de opciones del Tablet.
Fuente: Elaboración propia.*

Como la Ilustración 20 indica, se pueden comentar los storyboards realizados. Si se selecciona esa opción, es posible llegar al módulo de comentarios.

Módulo Comentar Dibujo

Este módulo permite tanto observar los comentarios realizados sobre el dibujo como crear un nuevo comentario al respecto. Si se desea crear un comentario nuevo, se abrirá una interfaz de texto de Android que permitirá ingresar el comentario, el cual será guardado en la metadata de la imagen. Por otro lado, si se desea modificar un comentario, al seleccionarlo se puede modificar el texto con el teclado. A continuación se presenta la interfaz que presenta este módulo.



*Ilustración 21: Módulo de Comentarios de dibujos, de ver comentarios (izquierda) o editar comentario (derecha)
Fuente: Elaboración propia.*

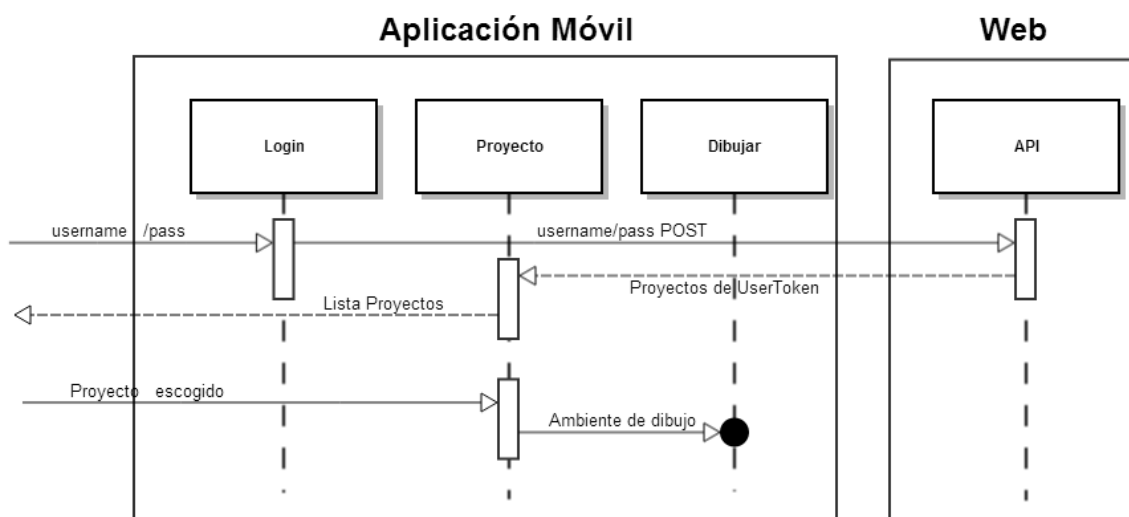
2.1.3.2 Servicios de Sistema

Existen instancias de la aplicación que harán que los distintos módulos se comuniquen con los servicios implementados en la parte web. A continuación se presenta la secuencia lógica que ocurre entre los distintos módulos y la API web de la aplicación, que permite el flujo adecuado de información para el trabajo. Estos servicios están explicados en detalle en el Documento de Diseño – Parte Web.

Para cada módulo se presentan los servicios utilizados por cada una de las acciones relevantes del usuario.

Módulo de logeo y selección de proyecto

Este módulo trata el inicio de sesión y elección de proyecto, con la ayuda del Webservice disponible en la API de la versión web.



*Ilustración 22: Comunicación entre aplicación móvil y los WebServices
Fuente: Elaboración propia*

Módulo de sincronización con Dropbox.

Este módulo trata la carga y descarga de archivos desde la cuenta de Dropbox asociada al proyecto.

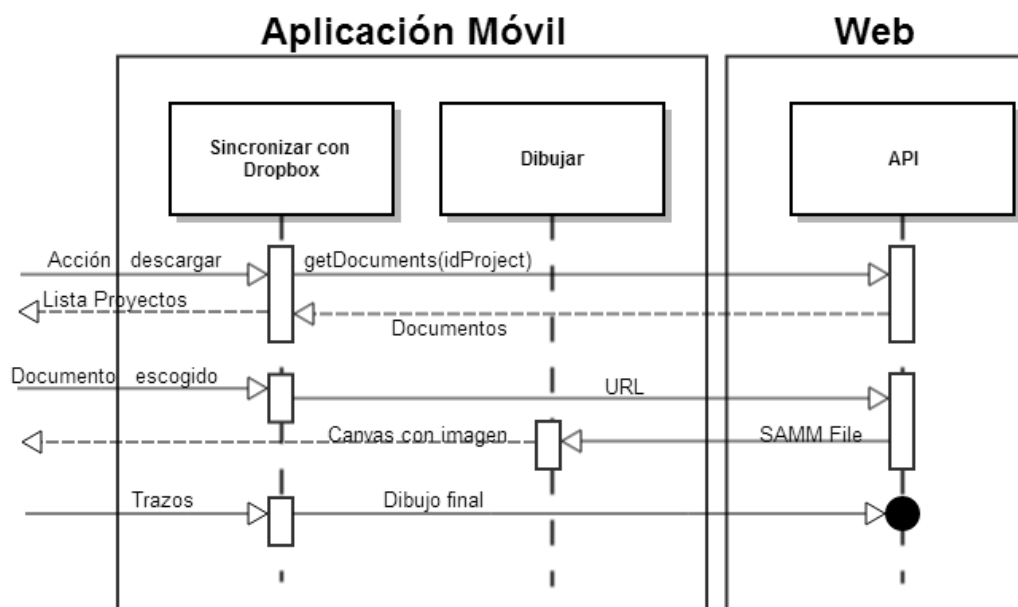


Ilustración 23: Comunicación entre el módulo de Sincronizar con Dropbox con el WebService

Fuente: Elaboración propia

3. Vista de Implementación

3.1.1 Estructura de la Aplicación

La aplicación móvil ha sido desarrollada en lenguaje java, utilizando el kit de desarrollo Android SDK. Dentro del único assembly se han separado las distintas funcionalidades y, debido al modelo que utiliza Android, la vista se encuentra totalmente independiente de las funcionalidades de la aplicación, en la sección *layout* de ésta. Todas las vistas son archivos xml, utilizando la notación definida por el sistema Android.

3.1.2 Arquitectura de Implementación

Front-End

Como se mencionó antes, el front-end de la aplicación se basa en archivos xml que definen la interfaz que ve el usuario. Cada módulo utiliza uno o más de estos archivos para manejar la información de forma visualmente agradable para el usuario. El siguiente diagrama presenta los detalles de las dependencias de estos formatos con los módulos.

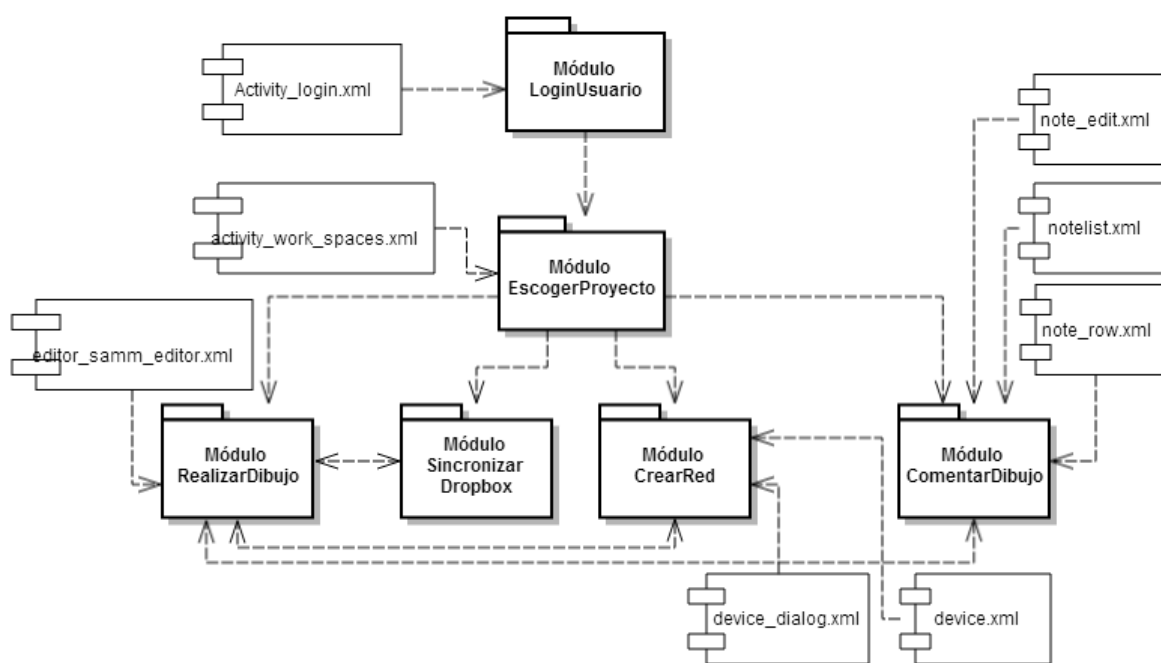


Ilustración 24: Diagrama de módulos y su dependencia de los archivos xml.
Fuente: Elaboración propia

4. Vista de Datos

Android provee numerosas opciones para la ayuda del almacenamiento de datos de aplicaciones de manera persistente. Sin embargo, SmartBoard Project Manager en su versión móvil no ocupa una gran cantidad de datos. De hecho, ocupa sólo datos de sesión y la información de proyectos. Como la cantidad de datos son bastante pocos, no se justifica hacer una BD.

Por lo mismo, luego de una investigación respectiva, se decidió utilizar la forma Shared Preferences entregada por Android, que guarda los datos en archivos ocultos en la memoria de los dispositivos y tiene las características de privacidad que concuerdan con la sensibilidad de estos datos.

Esta clase provee una estructura de datos primitiva que permite guardar datos en pares clave-valor. Por lo tanto, se usa para guardar la información respectiva al usuario y a los proyectos. Lo interesante es que esta forma de guardar la data persistirá entre diferentes sesiones de otras aplicaciones, incluso si la aplicación ha sido cerrada.

5. Vista de Deployment

A continuación se presenta la infraestructura móvil necesaria para el despliegue adecuado de la versión móvil de SmartBoard Project Manager.

5.1.1 Tecnología requerida

Para la parte móvil de la aplicación, es necesario que el dispositivo Android que la utilice posea una versión de Android 4.1, ya que las versiones anteriores no presentan la tecnología WiFiDirect, necesaria para sincronizar dispositivos y crear una sesión síncrona de trabajo.

Por otra parte, el dispositivo debe ser Samsung con SPen para poder dibujar, ya que la librería que reconoce el SPen viene incluido sólo en dispositivos de esta marca. Sin embargo, es posible que se quiera utilizar un dispositivo sólo para observar el trabajo, como lo es el Android MiniPC. En este caso, el dispositivo conectado a la red podrá sincronizar trabajo sólo desplegando imágenes actuales del trabajo.

5.1.2 Deployment

Para el correcto funcionamiento de la aplicación se requiere el SPen de Samsung, para permitir dibujar y lograr la máxima resolución y desempeño de la aplicación. Por otro lado, se requiere que la Tablet que trabaje tenga memoria suficiente en su memoria interna.