



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY



Sistema de aprendizaje autónomo para niños -Uso de periféricos-

Tutores:

Raquel Sosa
Christian Clark

Alumnos:

Farid Elías
Diego Rodríguez

Cliente:

Arsen Sahakian



Resumen

A nivel mundial las nuevas tecnologías están transformando las actividades educativas y habilitando nuevas formas de aprendizaje de contenidos que eran impartidos únicamente en formatos tradicionales. En Uruguay, Plan Ceibal distribuyó laptops y tabletas, permitiendo entre otras cosas una movilidad de las actividades de aprendizaje fuera del aula y el involucramiento del entorno del estudiante en el uso de la tecnología.

La organización XPrize propone desafíos abiertos de interés mundial en el que pueden participar pequeños equipos para proponer soluciones a un problema específico. Este proyecto se enmarca en la competencia “Global Learning XPrize”. El objetivo de la misma es el desarrollo de aplicaciones educativas para tabletas que permitan el aprendizaje autónomo de niños. Particularmente, niños en zonas del mundo donde la infraestructura tecnológica no esté desarrollada y el acceso a educación de calidad no esté dado. En esta competición los temas incluidos son lecto-escritura y matemáticas a niveles equivalentes a los primeros años escolares. El público propuesto son niños, no necesariamente escolarizados, en zonas específicas de África.

En este contexto se creó un equipo uruguayo para ingresar en esta competición que incluye dos proyectos de grado de Ingeniería en Computación. Por un lado, el proyecto “Sistema para Aprendizaje Autónomo de Niños - Aspectos de Interfaz e Interacción” se enfoca en las distintas interfaces de usuario, usabilidad y el carácter pedagógico que las aplicaciones deben poseer en función del público objetivo.

Por otra parte, y dada la complejidad que implica el desarrollo de aplicaciones educativas, el presente proyecto: “Sistema para Aprendizaje Autónomo de Niños - Uso de Periféricos” apunta a la simplificación del uso de los distintos periféricos disponibles en tabletas por parte de un desarrollador.

Dado que la plataforma sobre la cual deben correr estas aplicaciones fue definida en las bases del concurso (tabletas con sistema operativo Android), el proyecto tiene como

objetivo la investigación de tecnologías a utilizar para lograr una correcta solución al problema.

En este contexto se plantea el desarrollo de un framework que tenga como objetivo facilitar el trabajo de los desarrolladores de las mencionadas aplicaciones.

Por un lado, exponiendo el uso de los periféricos de manera simplificada, abstrayendo a los usuarios del framework de las complejidades inherentes al uso de los mismos. En concreto, desde el framework se proponen funcionalidades como la reproducción de sonido y video, captura de imágenes utilizando la cámara, conectividad entre dispositivos sin infraestructura de red y la transformación de texto en audio sintetizado.

Por otro lado, detectando e implementando algunas funcionalidades base de estas aplicaciones educativas, presentes en todas ellas. Para esto se propone el desarrollo de un generador de aplicaciones que facilita la inicialización del desarrollo de las mismas. El usuario del framework se encuentra entonces con un proyecto que ya tiene definidas las actividades educativas, las relaciones de precedencia entre unas y otras, la pantalla de selección de actividades y la lógica necesaria para definir qué actividades están disponibles.

Se realizó la verificación funcional que permitió encontrar y solucionar fallas. Se presentan a su vez dos casos de estudio, uno generado por nuestro proyecto que se trata de una Batalla Naval donde se utiliza la mayoría de los periféricos mencionados. Otro creado desde el proyecto “Sistema para Aprendizaje Autónomo de Niños - Aspectos de Interfaz e Interacción” utilizando el framework.

Se concluye que los objetivos del proyecto fueron alcanzados, desarrollando todas las funcionalidades propuestas e inclusive un generador de proyectos que facilita la inicialización del desarrollo para los usuarios del framework.

Palabras clave: aprendizaje, android, framework, cordova

Tabla de Contenido

1	Introducción	1
1.1	Contexto	1
1.2	Motivación	2
1.3	Objetivos del proyecto	3
1.4	Resultados esperados	3
1.5	Resultados obtenidos	4
1.6	Organización del documento	4
2	Marco de trabajo	5
2.1	Contexto XPrize	5
2.2	Detalles del concurso XPrize Global Learning	7
2.2.1	Hitos	8
2.2.2	Cronograma	9
2.3	Equipo uruguayo XPrize	10
2.4	Desarrollo de aplicaciones en Android	11
2.4.1	Desarrollo de aplicaciones Nativas	11
2.4.2	Desarrollo de aplicaciones Híbridas	11
2.4.3	Android SDK	11
2.4.4	Corona SDK	12
2.4.5	Apache Cordova	12
2.5	Tecnologías web	12
2.5.1	Angular.js	13
2.5.2	Ionic	13
2.5.3	Node.js	13
2.5.4	Yeoman	13
2.5.5	Underscore.js	14
2.6	Otras tecnologías	14
2.6.1	Gradle	14
3	Requerimientos y Análisis	15
3.1	Introducción	15
3.1.1	Aplicación educativa y actividad educativa	15
3.1.2	Proceso de creación de la aplicación educativa	16
3.2	Actores	16
3.3	Requerimientos específicos del generador	17
3.4	Requerimientos específicos del framework	17
3.5	Requerimientos no funcionales	18
3.6	Requerimientos de documentación	19
3.6.1	Manual de Usuario	19
3.6.2	Guías de instalación, configuración y archivo Léame	19
3.7	Decisiones tomadas	19
4	Arquitectura y Diseño	21
4.1	Introducción	21
4.2	Diseño del generador	23
4.3	Diseño del framework	26
5	Implementación	29
5.1	Ambiente de desarrollo	29
5.1.1	Ambiente de desarrollo Android	29
5.1.2	Ambiente de desarrollo node.js	29
5.1.3	Herramientas	29
5.2	Componentes de implementación a destacar	31
5.2.1	Verificación de ciclos	31
5.2.2	Wifi Direct	31
6	Verificación	35
6.1	Metodología de verificación	35

6.2	Casos de prueba.....	37
6.3	Resultado Verificación	39
7	Casos de estudio	41
7.1	Batalla Naval.....	41
7.1.1	Descripción	41
7.1.2	Flujo	42
7.1.3	Interfaz de usuario.....	43
7.1.4	Plugins utilizados	44
7.1.5	Análisis y Conclusiones de utilización del framework.....	44
7.2	“Aspectos de Interfaz e Interacción”	45
7.2.1	Introducción	45
7.2.2	Listado de actividades	46
8	Gestión	51
8.1	Planificación.....	51
8.2	Ejecución	52
9	Conclusiones y trabajo a futuro	53
9.1	Conclusiones.....	53
9.1.1	Restricciones	54
9.2	Trabajo a Futuro	55
9.2.1	Soporte Wifi-Direct para tres o más dispositivos	55
9.2.2	Políticas de rotación de logs.....	55
9.2.3	Soporte a otros periféricos	55
9.2.4	Colaboración con Plan Ceibal.....	55
9.2.5	Soporte de nuevos idiomas en plugin “text to speech”	56
9.2.6	Extensión de plugins en ambiente con conexión a redes	56
9.2.7	Soporte para múltiples plataformas.....	56
9.2.8	Mejora en la sucesión de actividades presentadas	56
10	Referencias.....	57

1 Introducción

En este capítulo se presenta una descripción global del proyecto, sus motivaciones y objetivos, así como también los resultados esperados y alcanzados. Por último, se indica cual es la organización del documento.

1.1 Contexto

La Fundación XPrize [1] es una organización sin fines de lucro que diseña y gestiona concursos públicos destinados a fomentar el desarrollo tecnológico que podría beneficiar a la humanidad. La misión de la Fundación XPrize es lograr "avances radicales en beneficio de la humanidad" [2] a través de la competencia incentivada. Fomenta las competiciones de alto perfil que motivan a las personas, las empresas y organizaciones en todas las disciplinas a desarrollar ideas y tecnologías innovadoras que ayuden a resolver los grandes retos que limitan el progreso de la humanidad.

Es por esto que XPrize propone un concurso donde se pretende lograr aplicaciones educativas que permitan a los niños aprender de forma autónoma. Además, estas aplicaciones serán de código libre para que puedan ser utilizadas, compartidas y mejoradas por todos. Esta tecnología se podría desplegar a nivel global, logrando experiencias de aprendizaje para niños que de otra forma serían inalcanzables. Niños que no tienen acceso a la educación de calidad o complementando las experiencias de niños que si lo tienen.

Más específicamente, el concurso se centra en el uso de tecnología para lograr que niños de todo el mundo aprendan a leer, escribir y realizar operaciones básicas de aritmética. Además, apuesta a sistemas que permitan que los niños puedan aprender solos, sin que sea necesario tener acceso a la infraestructura de una escuela. El objetivo es que un niño pueda acceder a un dispositivo como una tableta y empezar a aprender.

El cliente del proyecto se propuso conformar un equipo uruguayo para participar en este concurso. Dado que el desarrollo de aplicaciones educativas para este contexto es una tarea multidisciplinaria se detectaron problemas interesantes para ser trabajados en proyectos de grado de computación. Es por esto que se proponen dos proyectos diferentes:

- “*Sistema para Aprendizaje Autónomo de Niños - Aspectos de Interfaz e Interacción*”: orientado al trabajo en lo que respecta a interacción de usuario y diseño de actividades educativas. Los mayores desafíos de este proyecto están del lado de los aspectos pedagógicos de las aplicaciones a construir, logrando con actividades educativas, que los niños tengan contacto con el dispositivo, les resulte atractivo su uso y aprendan con el mismo.
- “*Sistema de aprendizaje autónomo para niños – uso de periféricos*”: orientado a facilitar el acceso a los distintos periféricos disponibles en los dispositivos y exponer funcionalidades comunes a todas las aplicaciones para alivianar el proceso de implementación de las mismas.

1.2 Motivación

Por medio de este enfoque de separación en dos proyectos se logra enfocar a ambos grupos en tareas bien distintas. El grupo “Interfaz e Interacción” puede enfocarse en aspectos pedagógicos y diseño de las actividades para lograr los mejores resultados en términos de aprendizaje. Otro punto importante es la interfaz en sí, generando una experiencia de usuario atractiva, logrando un mayor vínculo entre el usuario y la aplicación y de esta forma también aportando a los resultados en aprendizaje.

El trabajo de este proyecto: “Uso de periféricos” en el marco del aprendizaje autónomo de niños, cobra importancia al tener que allanar el camino en cuanto al uso del hardware de las tabletas, encapsulando complejidades inherentes a las APIs del sistema operativo y exponiendo una capa de funciones sencillas de utilizar.

A continuación, alguna de las definiciones y restricciones, tomadas de las directrices del concurso [3]:

- El hardware objetivo son tabletas con sistema operativo Android.
- El uso de la tableta puede ser en un aula o no.
- Los equipos que participan no tendrán acceso a las poblaciones donde se ejecutan las pruebas de campo, de todas formas, XPrize dará reportes obtenidos de las tabletas acerca del uso de las aplicaciones.
- La tableta contará con una cámara, micrófono y parlantes. Además, también contará con un adaptador de red.
- Las tabletas no contarán con conexión a internet.

1.3 Objetivos del proyecto

El objetivo general del proyecto es desarrollar un framework que simplifique el desarrollo de aplicaciones educativas permitiendo el acceso sencillo a periféricos de las tabletas Android a utilizar.

Los objetivos específicos del proyecto son:

- Investigar las alternativas tecnológicas existentes que provean una solución a la problemática planteada.
- Análisis, diseño e implementación de un framework para facilitar la creación de aplicaciones de aprendizaje autónomo para niños con interfaces de usuario ricas y animadas, en un entorno sencillo de implementar y conocido por los programadores encargados de dicha tarea. El framework deberá incorporar:
 - Acceso facilitado a los distintos periféricos presentes en las tabletas.
 - Funcionalidades que se consideren base, presente en cada una de las aplicaciones educativas, que, en lugar de ser definidas en cada aplicación, sean provistas por el framework.
- Diseño e implementación de un caso de estudio que muestre las ventajas de la utilización del Framework.

1.4 Resultados esperados

El principal resultado del proyecto deberá ser el framework brindando acceso sencillo a distintos periféricos, así como también a funcionalidades base ya implementadas por el mismo. Además, se desea generar una aplicación que lo utilice para probarlo.

Otros resultados esperados:

- Documentación respecto de tecnologías afines al problema planteado
- Documento de arquitectura y diseño de la solución.
- Manual de usuario del framework para el desarrollador.

1.5 Resultados obtenidos

El framework provee un mecanismo sencillo de generación de la configuración de las actividades educativas y proporciona la lógica necesaria para mostrar cuáles de estas actividades están disponibles, favoreciendo y simplificando la implementación de las aplicaciones por parte de los usuarios desarrolladores.

El framework cumple con los objetivos presentados, pudiendo utilizar distintos recursos de hardware de la tableta dentro del sistema operativo sin la necesidad de conocer en detalle las distintas APIs nativas. En su lugar, se utiliza la capa de abstracción propuesta por el framework, particularmente siendo capaz desde el mismo, de reproducir sonido y video, tomar fotos con la cámara y conectividad sin necesidad de infraestructura de red; también mecanismos para generar reportes de uso de la aplicación.

Se realiza también una batería de tests funcionales sobre todo el framework, haciendo mejoras e iterando hasta lograr un nivel de calidad satisfactorio para los objetivos propuestos. Por último, se implementa un caso de estudio para corroborar la facilidad de uso y el funcionamiento en general de la solución.

Se documenta el resultado de la investigación inicial sobre tecnologías existentes, relevantes a una posible solución del problema.

Dentro de los entregables también se encuentra el Documento de Arquitectura y Diseño, junto al Manual de Usuario, con guías para la instalación de la solución junto a la referencia completa de las funcionalidades expuestas por el framework.

1.6 Organización del documento

El documento se compone de 10 capítulos principales. Al comienzo se introduce el contexto en el que ocurre el proyecto junto con los objetivos planteados. Se definen a su vez los resultados que se esperan y los alcanzados. En el capítulo 2 se presenta el marco teórico, entrando más en detalle en lo referente al concurso XPrize y a las tecnologías que fueron investigadas.

En el capítulo 3 se presentan los requerimientos relevados en la etapa de análisis y algunas decisiones tomadas en cuanto al uso de tecnologías. El capítulo 4 presenta la Arquitectura general de la solución junto al diseño de los componentes. A continuación, se detallan elementos de la implementación y verificación en los capítulos 5 y 6 respectivamente.

Dos casos de estudio basados en el uso de la solución propuesta se presentan en el capítulo 7. En el capítulo 8 se encuentra el cronograma estimado y el ejecutado. Luego, en el capítulo 9, se presentan las conclusiones y mejoras al framework que se podrían realizar a futuro. Por último, en el capítulo 10 se encuentran las principales referencias consultadas durante el presente trabajo.

2 Marco de trabajo

En este capítulo se presenta el concurso XPrize, el cual enmarca este proyecto. Además, se muestra un resumen del marco de trabajo de las tecnologías evaluadas en el proyecto.

2.1 Contexto XPrize



Imagen 1 - Contexto XPrize

“Se estima que 250 millones de niños en el mundo no pueden leer, escribir o demostrar habilidades básicas de aritmética. Muchos de estos niños están en países en vías de desarrollo y no cuentan con acceso a escuelas de calidad o profesores. UNESCO estima que el mundo necesita 1.6 millones más de profesores a nivel global para el año 2015. Ese número se duplica para el año 2030.

Más allá de la existencia de aplicaciones para las escuelas y el entrenamiento de maestros, los modelos tradicionales de la educación no son capaces de escalar lo suficientemente rápido para satisfacer la demanda. Simplemente no se pueden crear suficientes escuelas o entrenar a suficientes profesores para enfrentar la demanda actual.

Según XPrize la humanidad se encuentra en un punto clave donde una alternativa o un enfoque radical es necesario, donde las actuales barreras para acceder a una educación de calidad sean eliminadas y donde las semillas de la innovación sean impartidas a todos los niños sin importar su ubicación geográfica o su estatus económico” [4].

La Fundación XPrize [1] es una organización sin fines de lucro que diseña y gestiona concursos públicos destinados a fomentar el desarrollo tecnológico que podría beneficiar a la humanidad. La misión de la Fundación XPrize es lograr "avances radicales en beneficio de la humanidad" [2] a través de la competencia incentivada. Fomenta las competiciones de alto perfil que motivan a las personas, las empresas y organizaciones en todas las disciplinas a

desarrollar ideas y tecnologías innovadoras que ayuden a resolver los grandes retos que limitan el progreso de la humanidad.

En este contexto surge el concurso XPrize Global Learning [5] que pretende lograr aplicaciones educativas que permitan a los niños aprender de forma autónoma. Además, estas aplicaciones serán de código libre para que puedan ser utilizadas, compartidas y mejoradas por todos. Esta tecnología se podría desplegar a nivel global, logrando experiencias de aprendizaje para niños que de otra forma serían inalcanzables. Niños que no tienen acceso a la educación de calidad o complementando las experiencias de niños que si lo tienen.

Desde XPrize se considera que hay un enorme potencial sin explotar para el aprendizaje autónomo basado en tecnología para transformar las oportunidades de educación de los niños que viven incluso en los lugares más pobres y remotos de la Tierra. La tecnología promete transformar el ambiente de aprendizaje para estos niños.

Si bien ha habido prisa por introducir tecnología en las aulas de las comunidades ricas, se ha hecho escaso esfuerzo para enfocar la tecnología donde más se necesita, las aldeas rurales de todo el mundo donde los recursos son escasos. [3]

En concreto, el concurso se centra en el uso de tecnología para lograr que niños de todo el mundo aprendan a leer, escribir y realizar operaciones básicas de aritmética. Además, apuesta a sistemas que permitan que los niños puedan aprender solos, sin que sea necesario tener acceso a la infraestructura de una escuela.

Aunque el desafío apunta a lograr aplicaciones que se puedan usar en todo el mundo, el concurso tiene como público objetivo para sus pruebas niños de países de África donde el inglés es el idioma oficial. También se considerará que las aplicaciones deben ser bilingües para considerar un idioma nativo y traducibles para poder ser luego reusadas en todo el mundo.

2.2 Detalles del concurso XPrize Global Learning

Las aplicaciones deben ser capaces de ser ejecutadas en una tableta Android. La tableta que se utilizará en las pruebas de campo estará equipada con un micrófono, una cámara y altavoces. Se anima a los equipos a hacer uso de estas características, pensando en actividades educativas interactivas con propuestas lúdicas y utilizando multimedia.

Por otra parte, si bien las tabletas cuentan con dispositivo de red, no existirá conectividad con internet, pudiendo no contar con infraestructura de red al ejecutar la aplicación.

Debido a que esta competencia se basa en la presunción de que el software debe ser diseñado para el aprendizaje auto-dirigido, XPrize proporcionará el hardware a las comunidades, pero no proporcionará instrucciones explícitas sobre cómo usar el software. Los equipos no tendrán acceso directo a los usuarios de las aplicaciones. Sin embargo, XPrize proporcionará a los equipos informes periódicos que contengan datos sobre el uso del software por parte de los niños involucrados en las pruebas de campo, así como también, dará a los equipos la oportunidad de actualizar sus soluciones en puntos específicos durante el período de prueba.

Respecto a los aspectos educativos, se desea mejorar las capacidades de lecto-escritura básicas de niños, entre las cuales se encuentran:

- vocabulario
- identificación inicial de sonido
- conocimiento de grafemas
- lectura de palabras conocidas
- lectura de palabras inventadas
- lectura oral de un pasaje
- comprensión lectora
- comprensión auditiva
- escritura de una oración completa

En lo que respecta a matemática y aritmética, los aspectos a ser relevados son:

- Identificación de números
- Número faltante
- Suma
- Resta
- Multiplicación
- Identificación de formas
- Nombrado de formas

Los niños, luego de utilizar las aplicaciones, en el período de prueba de campo serán evaluados en base a las pruebas EGRA (Early Grade Reading Assessment) [6] y EGMA (Early Grade Math Assessment) [7] que se comparan con un nivel de segundo grado escolar en Uruguay. Los resultados obtenidos por los niños al utilizar las distintas aplicaciones, serán tenidos en cuenta para decidir cuál es la aplicación ganadora.

Un componente esencial para el desarrollo de las aplicaciones es el compromiso a que sean de código libre. Para poder maximizar el crecimiento de las aplicaciones inclusive más allá del concurso, las aplicaciones tendrán licencias de software permisivas permitiendo uso comercial y no comercial de las mismas. De esta forma se favorece la colaboración.

2.2.1 Hitos

XPrize propone un proceso de concurso, el cual incluye los siguientes hitos:

1. Registro de participantes
 - Etapa donde individuos y equipos se anotan para participar del concurso. Al comenzar esta fase del concurso, existen 651 equipos pre-registrados (registrados como usuarios en el sitio de Global Learning XPrize) y terminan registrados 198 equipos.
2. Período de desarrollo de aplicaciones
 - Esta es la principal etapa, donde cada participante se encarga de desarrollar la aplicación a presentar en el certamen. Al momento de escribir este documento, se encuentran participando activamente 135 equipos.
3. Entrega de aplicaciones
 - Se finaliza el desarrollo y se entregan las aplicaciones
4. Ronda de juicios y anuncio de semifinalistas
 - Luego de aproximadamente 2 meses de deliberación, los juicios son anunciados junto con los semifinalistas
5. Período de pruebas piloto de aplicaciones semifinalistas
 - Si bien esta etapa es parte del itinerario, las guías del concurso [3] no especifican detalles sobre estas pruebas
6. Juicios de aplicaciones semifinalistas para anunciar finalistas
 - Lo interesante de estos juicios es que se hacen en vivo y personalizado para cada uno de los participantes semifinalistas. Los 5 finalistas son anunciados, quienes además ganan 1 millón de dólares por equipo.
7. Período de Prueba de campo con aplicaciones finalistas
 - Se someten las 5 aplicaciones finalistas a pruebas de campo durante 18 meses con niños pertenecientes a aldeas rurales de algún país perteneciente a la región de África Subsahariana (a ser determinado por XPrize).
8. Juicio final y anuncio del ganador del gran premio
 - Luego de que los niños hayan utilizado las aplicaciones, son evaluados con pruebas internacionales para medir las mejoras en el aprendizaje en lectura, escritura y aritmética básica. El equipo ganador será acreedor de la suma de 10 millones de dólares.

2.2.2 Cronograma

A continuación se muestra el cronograma general de Xprize Global Learning:

Fecha	Descripción
Setiembre 22-2014	Se abre el registro de equipo
Abril 30-2015	Registro de equipo cerrado
Marzo - Abril 2016	Reuniones de cada Equipo realizadas en múltiples ubicaciones. Para obtener detalles exactos, envíe un correo electrónico al equipo Global Learning a global.learning@xprize.org .
Mayo 1-2016	Las especificaciones de la tableta serán enviadas a los equipos.
Octubre 1-2016	Las instrucciones detalladas de la presentación serán enviadas a los equipos.
Enero 10-2017	Fecha de presentación de todos los proyectos y la primera ronda de juzgamiento comienza.
Marzo 1-2017	La primera ronda de juicios concluye y los semifinalistas serán anunciados.
Abril - Mayo 2017	Se llevará a cabo una prueba piloto para las soluciones semifinalistas.
Junio 12-16 2017	Se llevará a cabo un juicio "en vivo" en persona con cada uno de los equipos semifinalistas.
Julio 10-2017	Semifinalista juzgando conclusiones y finalistas se anuncian (premios finalistas se distribuyen).
Setiembre 1-2017	Comienza la prueba de campo.
Febrero 1-2019	La prueba de campo finaliza y comienza el juicio final. Los resultados se analizan para determinar el Ganador del Gran Premio.
Abril 1-2019	El juicio final y la evaluación concluyen. Se anuncia el ganador del Gran Premio. Evento de premiación tendrá lugar durante este tiempo.

2.3 Equipo uruguayo XPrize

Para desarrollar una aplicación educativa con los requisitos de XPrize es necesario un equipo multi-disciplinario. Algunos perfiles involucrados en este proyecto son:

- Diseñadores pedagógicos
 - se cuenta con la ayuda de personal de la facultad de psicología para facilitar la investigación realizada por el equipo de interacción en lo que respecta a aspectos pedagógicos de la aplicación
- Lingüistas
 - se contacta a un lingüista para entender aspectos del swahili, lenguaje utilizado por los niños que participarán en las pruebas de campo
- Locutor
 - El locutor Joaquín Bianco aporta grabaciones en inglés, creadas para la aplicación.
- Diseñador gráfico
 - María Alcaraz, diseñadora gráfica, coopera con el equipo de diseño de interacción aportando recursos gráficos creados específicamente para la aplicación
- Diseñadores de interacción
 - integrantes del proyecto de grado *Sistema para Aprendizaje Autónomo de Niños - Aspectos de Interfaz e Interacción*
- Desarrolladores de actividades educativas
 - integrantes del proyecto de grado *Sistema para Aprendizaje Autónomo de Niños - Aspectos de Interfaz e Interacción*
- Selección y desarrollo de tecnologías Android
 - integrantes del proyecto de grado *Sistema para Aprendizaje Autónomo de Niños – Uso de periféricos*
- Líder
 - Arsen Sahakian cliente del proyecto

Los distintos perfiles presentados, colaboran con esta iniciativa de forma honoraria, en la medida que su disponibilidad personal le permite.

Cabe también destacar que desde el Plan Ceibal se mostró interesado en participar y cooperar con los proyectos de grado, así como en el uso a futuro de las soluciones presentadas.

2.4 Desarrollo de aplicaciones en Android

Las aplicaciones sobre la plataforma Android pueden ser desarrolladas utilizando distintos enfoques:

2.4.1 Desarrollo de aplicaciones Nativas

El tradicional, conocido como desarrollo nativo, es el uso de las distintas APIs ofrecidas por el sistema operativo para crear las aplicaciones. En el caso de Android, se trata del uso de Java como lenguaje de programación y el SDK de Android como framework base para el desarrollo. El desarrollo nativo hace uso directo de las APIs provistas por el sistema operativo lo que significa el uso óptimo de los distintos recursos (lógicos, físicos) disponibles. A su vez, implica el conocimiento específico de la plataforma objetivo. Si se desea desarrollar para múltiples plataformas, el grado de reutilización es ínfimo.

2.4.2 Desarrollo de aplicaciones Híbridas

Otro enfoque, conocido como desarrollo híbrido, se define como la implementación de una aplicación web, conectada a una capa de abstracción por sobre las APIs del sistema operativo. Como desventaja, no siempre resulta en el uso óptimo de recursos, pero ofrece la ventaja de poder reutilizar una gran parte de la implementación en distintas plataformas, solo es necesario cambiar la capa de abstracción de una plataforma a la otra.

2.4.3 Android SDK



El kit de desarrollo de software “Android SDK” [8] provisto por google, es un conjunto de herramientas de desarrollo entre las cuales se incluyen “debuggers”, librerías, emuladores, documentación y tutoriales. También se provee un editor (IDE, Integrated Development Environment) que soporta el mencionado kit, conocido como Android Studio el cual es una versión personalizada de “Intelij IDEA”.

El SDK de Android tiene varias ventajas, entre ellas, que es el framework realizado por los mismos creadores que el sistema operativo lo que implica un grado de compatibilidad con las APIs del S.O. muy alto. El lenguaje de programación que utiliza es JAVA. La interfaz gráfica se define por medio de archivos XML (aunque se puede también definir a nivel de código). Por otra parte, existe una curva de aprendizaje importante en lo que refiere al conocimiento de los distintos componentes gráficos y formas de definir el diseño. La documentación que ofrece es muy completa y con muchos ejemplos. Además, las aplicaciones implementadas con el SDK de Android son nativas, favoreciendo el rendimiento pero generando una aplicación capaz de correr únicamente en esta plataforma. Fuente: “Android Developers Portal: Application Fundamentals” [9].

2.4.4 Corona SDK



Corona SDK

Otro de los frameworks utilizados para aplicaciones es el SDK de Corona Labs: “Corona” [10]. Este es un framework multi-plataforma: no es necesario definir una aplicación distinta dependiendo de la plataforma destino. El lenguaje de programación que utiliza es Lua y accede a APIs programadas sobre C++/OpenGL por lo que se puede lograr un buen rendimiento a nivel gráfico, siempre y cuando se implemente sobre las mencionadas APIs. Por otra parte, tiene la desventaja de que el framework depende fuertemente del código cerrado generado por Corona Labs, dado que el código de las aplicaciones Corona no accede directamente a las APIs del sistema operativo sino por medio del propio framework. Fuente: “Introduction to Corona” [11].

2.4.5 Apache Cordova



CORDOVA™

Cordova, al igual que Corona es un framework multi-plataforma. Por otra parte, es de código abierto y de la fundación Apache. El desarrollo de aplicaciones en cordova se realiza por medio de las tecnologías web conocidas: HTML como lenguaje declarativo para definir el diseño de la interfaz gráfica, CSS para definir los estilos y Javascript, el lenguaje de programación para definir el comportamiento de la aplicación. Esto implica conjuntamente la mayor ventaja y el mayor inconveniente. Cordova permite definir aplicaciones que corren en múltiples plataformas móviles con un único código fuente y con tecnologías conocidas con una comunidad enorme por detrás: las tecnologías de la web. Tiene la ventaja de que se puede utilizar el conocimiento adquirido en desarrollo de aplicaciones web. Por otro lado, las aplicaciones corren sobre la implementación nativa de una vista web (web-view), sujeta a las restricciones de rendimiento de éstas. Por otro lado, estas vistas web implementan los últimos estándares. También ofrecen la ventaja adicional de evitar el inconveniente del desarrollo web orientado a múltiples navegadores; utilizando Cordova, siempre se apunta a una única implementación de la vista web, en el caso de Android, basada en el proyecto de código libre “Chromium”. [12]

Cordova a su vez ofrece distintas funcionalidades propias de las plataformas en las que corre por medio de “plugins”. Estos plugins tienen una implementación nativa por plataforma y la funcionalidad es expuesta por medio de javascript a la capa web. Fuente: “Apache Software Foundation: Cordova introduction” [13].

2.5 Tecnologías web

Luego de estudiadas las distintas opciones para la implementación de aplicaciones móviles Android y ante la posibilidad de ir por el camino de aplicaciones híbridas (definidas en la sección 2.4.2) con tecnologías web, se tornó necesario investigar más en detalle el estado del arte de las mismas.

2.5.1 Angular.js



Angular.js. es un framework orientado a aplicaciones SPA (single page apps) y se adecúa muy bien al desarrollo híbrido en plataformas móviles. Algunas de sus características:

- Desarrollo modular (cada controlador, servicio, componente se definen en módulos separados)
- inyección de dependencias
- encapsulamiento de modificaciones del DOM (Document Object Model) por medio de directivas o componentes
- Patrón MVC (Model-View-Controller) del lado del cliente

2.5.2 Ionic



Ionic es un framework o librería que hace uso a su vez de otros frameworks aportando una gama de componentes web orientados a los dispositivos móviles. También provee un conjunto de herramientas de desarrollo que permiten inicializar proyectos, compilarlos, desplegarlos en las plataformas objetivo y depurarlos.

A su vez Ionic se apoya en otros frameworks como son Apache Cordova o Angular.js definiendo los componentes web como directivas o componentes de Angular. Fuente: “*The oficial Ionic Blog: Where does the Ionic Framework Fit In?*” [14]

2.5.3 Node.js



Node es muy popular y joven (debutó en el 2009). Es el segundo proyecto más observado de la plataforma git (<https://github.com/joyent/node>), con muchos seguidores en el grupo de google (<http://groups.google.com/group/nodejs>) y su canal IRC <http://webchat.freenode.net/?channels=node.js>.

Además, tiene más de 15000 módulos creados por la comunidad, publicados en NPM (Node Package Manager). Todos estos detalles enfatizan la atracción que ejerce en la comunidad. El sitio oficial (<http://www.nodejs.org>) define node como "una plataforma construida sobre el ambiente de ejecución Javascript de Chrome para construir aplicaciones de red escalables. Node.js usa un modelo de entrada/salida orientado a eventos, no bloqueante que lo hace liviano y eficiente, perfecto para aplicaciones de tiempo real, con manejo intensivo de datos y corriendo en sistemas distribuidos". [15]

2.5.4 Yeoman



YEOMAN

Dentro de las herramientas mencionadas en el punto anterior, vale la pena destacar Yeoman. Muchos frameworks requieren de varios archivos html, css, javascript, contenido a su vez bastante repetitivo (conocido como “Biolerplate”). Existen herramientas que se encargan de generar estos archivos para que la inicialización de un proyecto sea más rápida y fácil. Yeoman es una herramienta generadora de proyectos. Estos generadores facilitan la producción de los primeros archivos necesarios para inicializar los proyectos,

aliviando al desarrollador con la carga de esta tarea repetitiva. Los generadores de Yeoman se encuentran en su mayoría en github y la herramienta tiene una forma sencilla de indexarlos y obtenerlos automáticamente desde la web. Existen generadores para proyectos de diversas tecnologías como angular, ionic, bootstrap, backbone, etc. Fuente: “Getting started with yeoman” [16].

2.5.5 Underscore.js

UNDERScore.JS “Underscore” es una librería que ofrece alrededor de 100 funciones categorizadas según los tipos de datos que manejan:

1. Funciones de manipulación de arreglos
2. Funciones de manipulación de objetos
3. Funciones para la manipulación de colecciones
4. Funciones para manipulación de funciones

2.6 Otras tecnologías

2.6.1 Gradle



Gradle es el paso siguiente en herramientas de construcción basadas en la Máquina Virtual de Java (JVM). Su especificación está basada en lecciones aprendidas de herramientas de construcción de proyectos anteriores como Ant o Maven y lleva las mejores ideas de estas al próximo nivel. A partir de un enfoque "construcción por convención" (build-by-convention), Gradle permite modelar de manera declarativa el dominio del problema utilizando un lenguaje de dominio específico (DSL) expresivo y potente. Como gradle es nativo en la Máquina Virtual de Java, permite al usuario escribir lógica personalizada en un lenguaje conocido por el mismo, por ejemplo: Java o Groovy. [17]

3 Requerimientos y Análisis

El presente capítulo presenta dos conceptos esenciales en el desarrollo de las aplicaciones: “Aplicación educativa” y “Actividad educativa”. Además, se definen las especificaciones funcionales, no funcionales y del sistema.

3.1 Introducción

Los requerimientos se definen por un lado con el cliente del proyecto y por otro con el grupo encargado de desarrollar las actividades educativas. Dado que el grupo cuenta con conocimientos de tecnologías web, de las alternativas de desarrollo para Android presentadas en el capítulo 2, se decidió tempranamente el diseño de un framework para aplicaciones híbridas.

3.1.1 Aplicación educativa y actividad educativa

A continuación, se presentan dos conceptos que son esenciales en el desarrollo de las aplicaciones objetivo del framework.

3.1.1.1 Aplicación educativa

Una aplicación educativa es una aplicación de Android que se encarga de cubrir los objetivos propuestos por el concurso XPrize. La misma contiene un conjunto de actividades en forma de juegos. Estas actividades pueden tener como condición para su ejecución, la completitud de otras actividades previas. La aplicación educativa entonces se constituye por un número n de actividades educativas junto con una pantalla de selección de actividades, la cual tiene en cuenta el criterio mencionado de previas para desplegar cuales son las actividades disponibles en un momento dado.

3.1.1.2 Actividad educativa

Una actividad educativa constituye un juego particular dentro de la aplicación. La actividad educativa se enfoca en uno o varias áreas de la educación como ser lectura, escritura o matemáticas; además tiene un objetivo a cumplir para ser considerada como terminada y de esta forma dar lugar a las actividades educativas que le suceden.

3.1.2 Proceso de creación de la aplicación educativa

En esta sección se detallan las distintas etapas que hacen al proceso de creación de una aplicación educativa.

1. Diseño

Durante el diseño se especifican cuáles son las distintas actividades que conformaran a la aplicación, un identificador para cada una y las relaciones de precedencia entre las mismas. A modo de ejemplo, supongamos el siguiente escenario, representado en la imagen 2 donde tenemos 3 actividades “Juego 1”, “Juego 2” y “Juego 3” y donde “Juego 1” es previa a “Juego 2” y esta es previa a “Juego 3”.

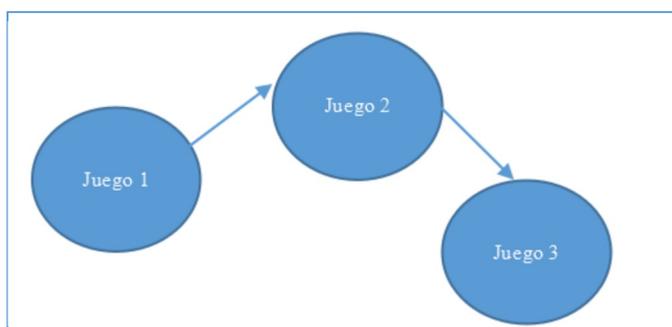


Imagen 2 - Actividades y sus precedencias

Luego de definida esta configuración se procede a la etapa 2

2. Implementación

Una vez definidas las distintas actividades, se procede a implementar las mismas. Esto es, definir las distintas interfaces de usuario, la lógica de la actividad y las condiciones para que la misma llegue al estado “finalizada”. Es en esta etapa donde los desarrolladores de actividades educativas deben considerar los distintos aspectos pedagógicos para poder lograr los objetivos de aprendizaje.

3. Empaquetado

Luego de implementadas las actividades se procede al empaquetado y creación del ejecutable para luego desplegar el mismo dentro de un dispositivo compatible. Esta etapa crea un archivo de tipo “apk” (aplicación Android) incluyendo sonidos, imágenes, código javascript y html de cada una de las actividades.

3.2 Actores

Al tratarse del desarrollo de un framework, el único actor que se maneja en los requerimientos es el usuario del framework, desarrollador con conocimientos web (HTML, CSS, Javascript) y el uso de frameworks y librerías como son Ionic y Angular.js.

3.3 Requerimientos específicos del generador

- El generador deberá crear la estructura inicial del proyecto web.
- Dicho proyecto web utilizará componentes de “Ionic” y empleará como framework MVC de la vista a angular.js
- El generador proveerá un mecanismo interactivo de configuración de las distintas actividades, así como las relaciones de precedencia entre las mismas.
- El generador será capaz de agregar nuevas actividades y sus respectivas relaciones de precedencia modificando el mencionado proyecto.
- El generador validará que no existan ciclos entre actividades.
- El generador proveerá una API para marcar actividades como finalizadas.
- El generador proveerá la lógica dentro del proyecto web para definir cuales actividades deben ser mostradas en el lobby.
- El generador creará servicios angular que envuelvan las distintas APIs de los plugins disponibles en el framework para facilitar el acceso desde el proyecto web angular a los mismos.

3.4 Requerimientos específicos del framework

- El Framework brindará la posibilidad de crear actividades, cada actividad será identificada por un nombre.
- El Framework brindará la posibilidad de agregar a una actividad, la lista de actividades previas de la primera.
- El Framework controlará que no se generen ciclos entre las actividades.
- El Framework brindará la posibilidad de utilizar la cámara de la tableta, para tomar fotos y filmar.
- El Framework permitirá el uso del micrófono incorporado en la tableta para la grabación de audio.
- El Framework permitirá la reproducción de videos.
- El Framework permitirá la reproducción de audio.
- El Framework permitirá comunicar con otros dispositivos utilizando el adaptador de red inalámbrica con el protocolo Wifi-Direct [18].
- El Framework brindará la posibilidad de transformar texto a audio en español e inglés.
- El framework permitirá que las aplicaciones guarden información referente al uso que luego podrá ser recolectada.
- El framework creará la pantalla de “lobby” donde se podrán ver las actividades que el usuario final podrá acceder.
- El framework gestionará las actividades que se muestran en el “lobby”. Para ello ofrecerá una API que permita marcar a las actividades como finalizadas y creará la lógica necesaria para comparar la configuración de actividades con las actividades finalizadas, definiendo así cuales son mostradas en la pantalla inicial.
- Toda solución desarrollada deberá depender únicamente de otras herramientas/frameworks/utilidades de tipo “open source”.

3.5 Requerimientos no funcionales

- El framework deberá ajustarse a los siguientes límites:
 - memoria primaria 512 MB
 - memoria secundaria 2 GB
- El framework tendrá que ser soportado en tabletas con el sistema operativo Android a partir de la versión 4.0.
- El framework deberá facilitar la creación de interfaces de usuario ricas, animadas y sencillas de implementar.

3.6 Requerimientos de documentación

3.6.1 Manual de Usuario

- El proyecto contará con un manual de usuario en el que se mostrará cada una de las funcionalidades del framework, información sobre el generador y referencia de las APIs con ejemplos.

3.6.2 Guías de instalación, configuración y archivo Léame.

- Por medio del manual de usuario, los desarrolladores contarán con guías necesarias para instalar el framework, configurar e iniciar el proyecto web así como hacer uso de las distintas funcionalidades provistas por el framework.

3.7 Decisiones tomadas

Uno de los principales requerimientos del framework y por ende del proyecto en sí, es facilitar la creación de aplicaciones móviles y sus distintas interfaces de usuario orientadas al aprendizaje autónomo del público objetivo (niños), proporcionando a los usuarios del framework un entorno de implementación sencillo y con una curva de aprendizaje baja o nula.

Un posible enfoque para este tipo de aplicaciones sería el desarrollo nativo. Para esto se debe contar con conocimiento específico de las distintas APIs, así como también del modelo de interfaces de usuario provisto por el SDK de Android. Este modelo se basa en la configuración de componentes de la interfaz por medio de archivos xml. Su uso requiere de una curva de aprendizaje un tanto elevada.

Otro posible enfoque es el desarrollo híbrido. Se resuelve seguir este camino por dos fundamentales razones:

- Abstracción de las APIs del sistema operativo por medio de una capa simplificada de acceso desde la aplicación web
- Definición de la interfaz de usuario utilizando tecnologías web conocidas y dominadas por desarrolladores web

Dentro del desarrollo híbrido, se decide utilizar el framework Cordova.

4 Arquitectura y Diseño

En este capítulo se muestra una visión general de la arquitectura del sistema a construir.

4.1 Introducción

De acuerdo al proceso de creación de aplicaciones educativas (sección 3.1.2) visto en el capítulo anterior, se definen dos grandes componentes que atacan a dos etapas diferentes del mencionado proceso. A continuación, se presenta en la imagen 3 la relación entre estos componentes y las etapas en el proceso de creación de aplicaciones educativas.

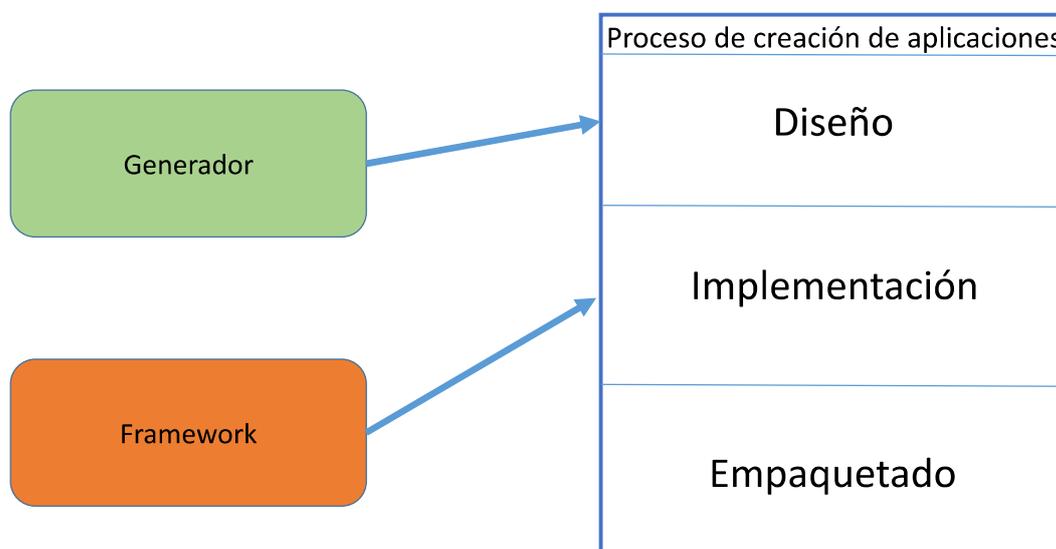


Imagen 3 - Relación entre componentes y etapas de proceso

El generador es un componente de gran utilidad en la etapa de diseño de la aplicación. Se encarga de generar los archivos iniciales del proyecto web y en particular de volcar la información sobre las distintas actividades y sus relaciones de precedencia. De esta forma los usuarios del framework no tienen que considerar los detalles de esta configuración ni la implementación de la pantalla principal ya que la misma es parte del proyecto inicial creado por el generador.

Luego de contar con el proyecto inicial y a la hora de implementar las distintas actividades educativas, los usuarios del framework pueden utilizar las distintas funcionalidades o plugins provistos para acceder a los distintos periféricos.

Aplicación Educativa construida con framework

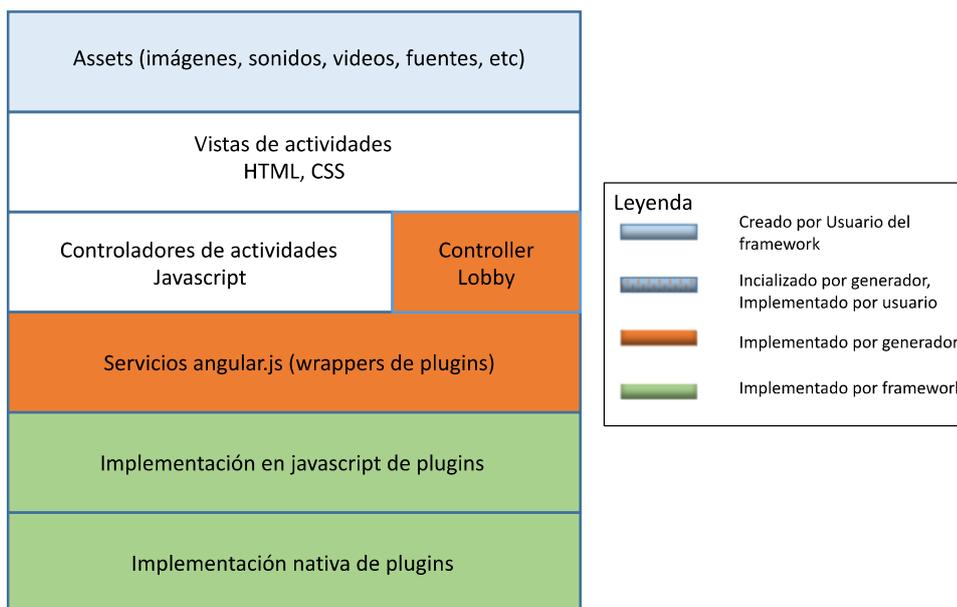


Imagen 4 - Aplicación educativa tipo

La imagen 4 muestra una aplicación creada con el framework y el generador. Podemos notar las dos capas inferiores en las que están implementados los distintos plugins del framework, código nativo y javascript. Luego están las capas que tienen contenido implementado por el generador, como los servicios angular que envuelven a los distintos plugins. Luego vienen los controladores angular, uno ya implementado por el generador y los demás inicializados por el mismo e implementados por los usuarios del framework. Esto también sucede con las vistas de cada actividad educativa. Por último, tenemos los “assets” de las vistas, todos introducidos a la aplicación por los usuarios del framework.

Con las imágenes 3 y 4 se explica cuál es la participación del generador y framework en las distintas etapas de creación y en las distintas capas de la aplicación.

4.2 Diseño del generador

Dado que el propósito principal del framework es abstraer al desarrollador web (usuario del framework) de los detalles de la plataforma Android, exponiendo funcionalidades útiles para las aplicaciones generadas por el framework, se decide no sólo proveer funcionalidades nativas de la plataforma Android, sino también funcionalidades a nivel web. En otras palabras, se considera de utilidad para el usuario del framework, poder arrancar con un proyecto web que contenga ciertas funcionalidades ya implementadas.

En la plataforma node.js existen herramientas conocidas como generadores, encargados de crear los archivos iniciales en un proyecto. Desde el generador se puede obtener información del usuario. En función de la misma, se genera el proyecto con detalles personalizados a las necesidades del usuario. [19]

Partiendo de la base de que toda aplicación tendrá un conjunto de “actividades” y que además algunas actividades deben ser completadas para poder realizar las que le siguen, se implementa un generador que obtiene información sobre el número de actividades, nombre de las mismas e información acerca del grafo de previas.

Durante el proceso de obtención de la información, se verifica constantemente que no haya ciclos en el grafo.

Luego se genera un archivo de configuración del proyecto con el grafo de actividades. Se crea una vista y un controlador de angular por cada actividad educativa, se crea un conjunto de servicios angular que facilitan el acceso a los distintos plugins y se agregan las dependencias a los frameworks ionic y angular.js.

A continuación, se muestra un diagrama con los principales componentes del generador:

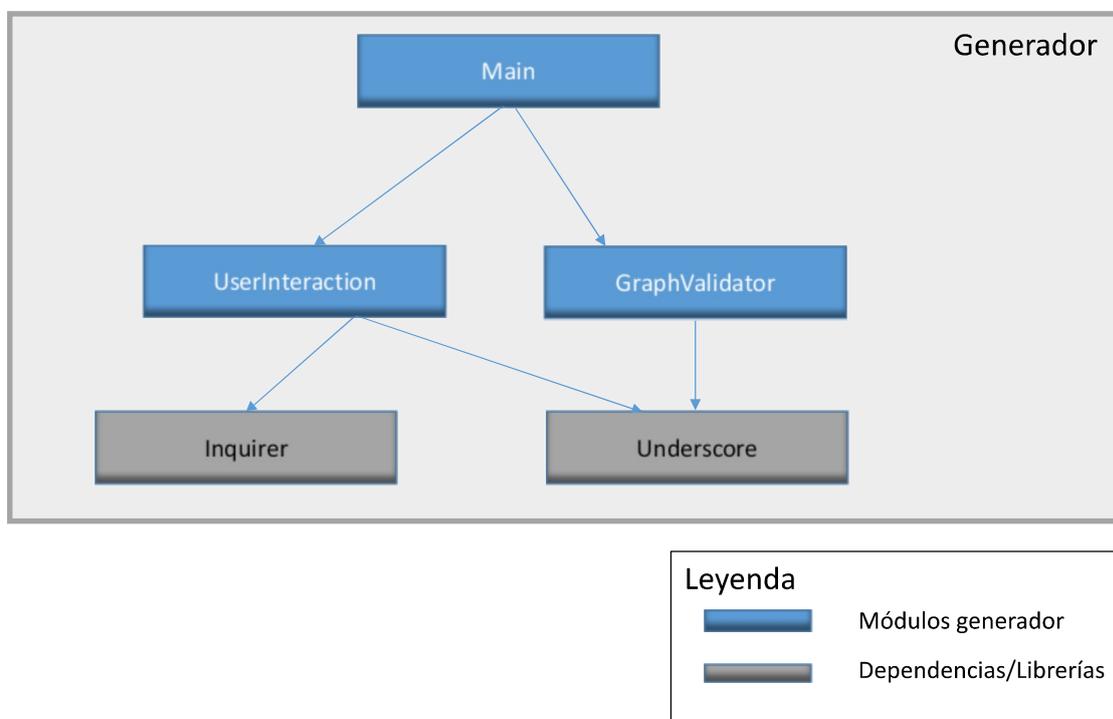


Imagen 5 - Módulos del generador

El módulo “main” es el principal y utiliza funcionalidades del módulo “userInteraction” para definir las preguntas y respuestas que se le hacen al usuario del generador. En todos los casos de asincronismo (por ejemplo, al hacer una pregunta, la respuesta retorna de forma asincrónica al generador) se hace un fuerte uso de promesas. [20]

El módulo “graphValidator” verifica para un grafo si posee ciclos.

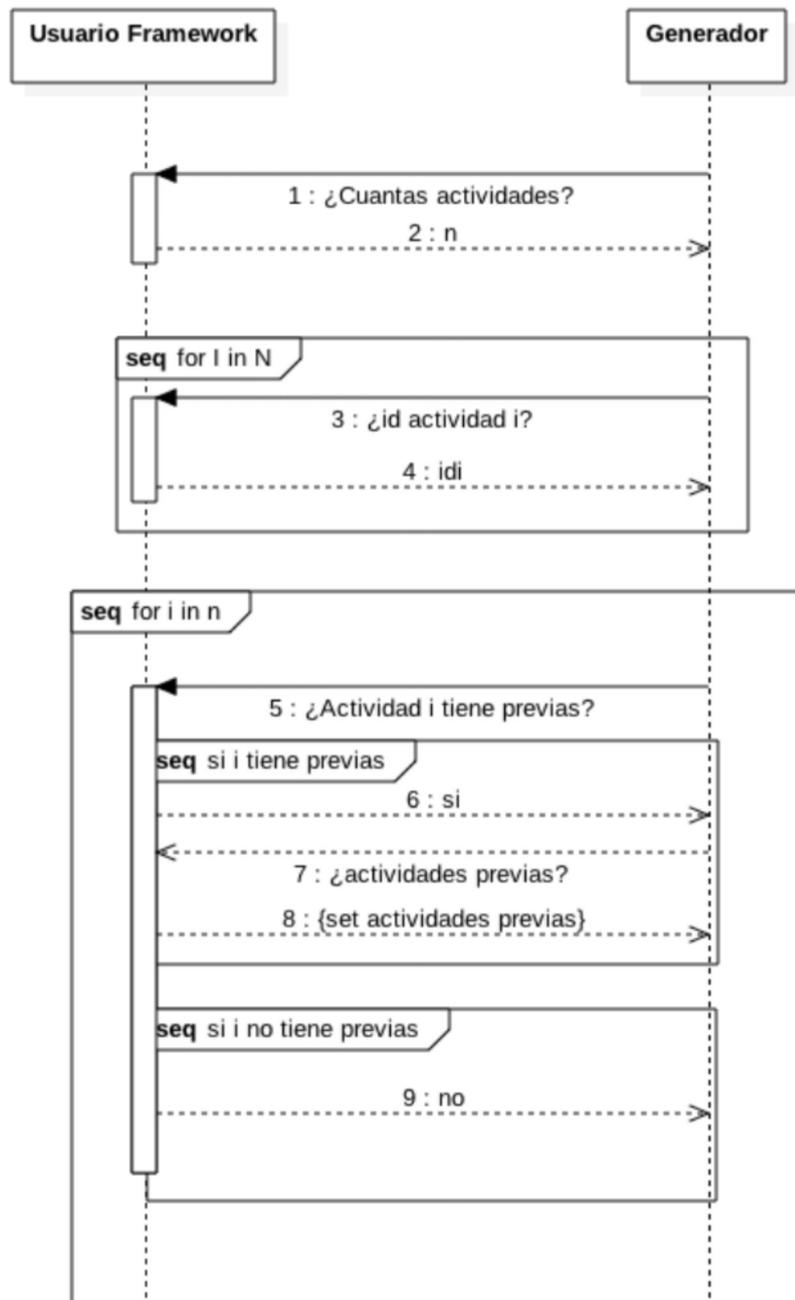


Imagen 6 - Diagrama de secuencia generador

En la imagen 6 podemos ver un diagrama de secuencia del generador, evidenciando las distintas preguntas que el usuario del framework debe responder para poder generar la configuración y los archivos iniciales del proyecto.

4.3 Diseño del framework

Dada la decisión de utilizar Apache Cordova, la arquitectura de nuestro framework está fuertemente basada en la arquitectura de Cordova. Dentro de Cordova, se definen un conjunto de componentes lógicos llamados “plugins” que se encargan de exponer funcionalidades nativas de los dispositivos a la capa web.

Cada uno de los requerimientos funcionales del framework se mapean a un plugin cordova diferente.

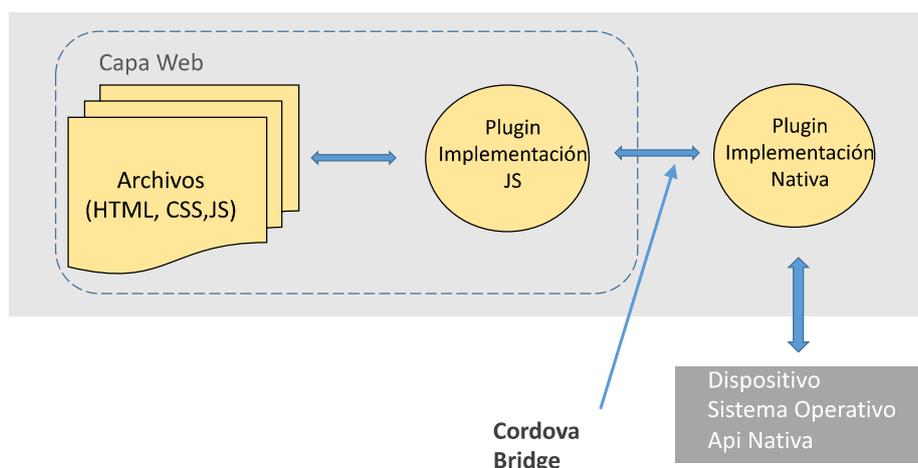


Imagen 7 : Modelo de plugin cordova

El diseño que se describe en la imagen 7, aprovecha las ventajas que brinda Cordova que resuelve la comunicación entre la capa web y la capa nativa sobre Android. De esta manera los usuarios del framework pueden concentrarse en brindar una buena experiencia de usuario con vistas atractivas para el público objetivo, utilizando tecnologías conocidas de la web y abstraerse de las APIs nativas de los dispositivos.

Dentro de la capa web, se decide utilizar la librería Ionic, que presenta componentes visuales orientados a aplicaciones móviles. Además, Ionic se apoya fuertemente sobre Angular.js.

Este último define una forma de trabajo con componentes de diseño bien identificados: vistas, controladores, directivas (componentes de la vista) y servicios, los cuales son los encargados de encapsular toda interacción con los sistemas backend en aplicaciones web o, en caso de una aplicación móvil híbrida (como lo son las aplicaciones cordova) toda comunicación con la plataforma nativa. Es por esto que además de exponer a nivel javascript los distintos plugins, estos se encuentran encapsulados dentro de servicios Angular para facilitar su utilización por parte de los usuarios del framework.

SISTEMA DE APRENDIZAJE AUTÓNOMO PARA NIÑOS – USO DE PERIFÉRICOS

A continuación, en la imagen 8 se muestran los distintos componentes del framework y sus sub-componentes: servicio angular, implementación javascript e implementación nativa. Todo separado en dos grandes capas: capa web y capa nativa.

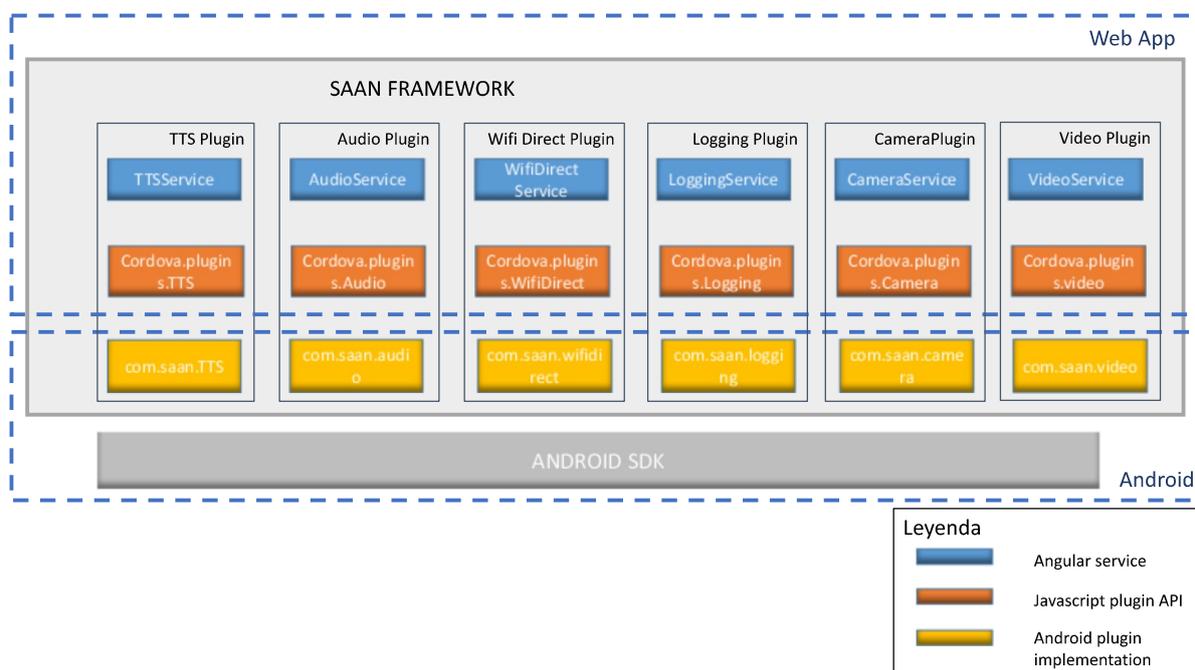


Imagen 8 - Diagrama de componentes

5 Implementación

Este capítulo describe aspectos referentes a la implementación de la solución, como ser ambiente de desarrollo, herramientas utilizadas y algunos componentes a destacar de la implementación en sí misma.

5.1 Ambiente de desarrollo

Dado que la solución propuesta implica dos piezas de software diferentes, por un lado el generador y por otro la implementación de los plugins, se definen dos ambientes separados de desarrollo.

5.1.1 Ambiente de desarrollo Android

Para la implementación de plugins, se utiliza la IDE proporcionada por android conocida como Android Studio:

Android Studio proporciona las herramientas más rápidas para la creación de aplicaciones en todos los tipos de dispositivos Android. La edición de códigos de primer nivel, la depuración, las herramientas de rendimiento, un sistema de compilación flexible y un sistema instantáneo de compilación e implementación te permiten concentrarte en la creación de aplicaciones únicas y de alta calidad. [21]

Dado que además Android Studio es compatible con la herramienta de construcción “gradle” y que cordova utiliza gradle para la implementación de los plugins Android, es que Android Studio se presenta como la opción ideal para nuestro proyecto debido a la compatibilidad del mismo con Apache Cordova.

A modo de ejemplo, para crear un plugin android, primero se utiliza el cliente de terminal de Cordova para crear la estructura del plugin. Luego en Android Studio se importa la misma como proyecto gradle utilizando el archivo build.gradle que define el proyecto.

5.1.2 Ambiente de desarrollo node.js

Para la implementación del generador, se utiliza node.js como plataforma de desarrollo. Como IDE, se emplea el editor de texto sublime text que provee la facilidad de destacar lenguajes como los utilizados en el desarrollo web, ejemplos: HTML, CSS, Javascript.

5.1.3 Herramientas

5.1.3.1 Cordova command line tool

La herramienta de línea de comando de cordova permite crear la estructura inicial de un plugin, así como inicializar un proyecto cordova, configurar las plataformas destino y agregar/quitar plugins ya sea desde internet o desde el sistema de archivos local. Además provee mecanismos para la construcción e instalación de la aplicación en los dispositivos o emuladores.

5.1.3.2 NPM

La herramienta NPM (“Node Package Management”) [22] se encarga de manejar las dependencias de aplicaciones node.js de módulos provistos por la comunidad. Por medio de esta, se puede instalar dependencias a nivel local de proyectos o a nivel global del sistema operativo donde se desarrolla. También permite la instalación de paquetes personalizados a nivel global, ejemplo: instalación global del generador para ser utilizado para la creación de proyectos web.

5.1.3.3 Gradle

Gradle [23] es la herramienta utilizada para construir la parte android (java) de las aplicaciones híbridas creadas. Por medio de la misma se pueden entre otras cosas configurar las dependencias de los plugins sobre otras librerías utilizadas.

5.1.3.4 Genymotion

Luego de probar el emulador de dispositivos provisto por Android SDK, optamos por utilizar “Genymotion” [24] que ofrece una mayor velocidad de emulación e inicialización.

5.1.3.5 Control de versionado

Para el control de versionado utilizamos Git dado que se cuenta con el repositorio provisto por la Facultad de Ingeniería “Gitlab” [25]. Para poder utilizar este repositorio solo se debe contar con las credenciales de facultad por lo que nos resultó muy útil. Como cliente git utilizamos el cliente por defecto de línea de comando “git” así como el cliente windows “Git Extensions” [26] y el cliente OSX “Source tree” [27].

5.2 Componentes de implementación a destacar

En esta sección se resaltan dos componentes de la implementación:

- Algoritmo de verificación de ciclos
- Implementación del plugin Wifi Direct

5.2.1 Verificación de ciclos

Para verificar la existencia de ciclos en el grafo de actividades, se implementa un algoritmo de backtracking recursivo de búsqueda en amplitud sobre el grafo. El orden del algoritmo es $O(|V|+|E|)$ peor caso, siendo $|V|$ la cantidad de nodos y $|E|$ la cantidad de aristas. A continuación el pseudocódigo del mismo.

```
funcion detectarCiclos(grafo, nodo, caminoActual){
  si (caminoActual contiene nodo){
    //CICLO DETECTADO
    retorno caminoActual
  } sino {
    caminoActual + nodo
    para cada nodo.adjacentes adj{
      retorno detectarCiclos(grafo, adj, caminoActual)
    }
  }
}
```

5.2.2 Wifi Direct

5.2.2.1 Introducción

Wifi direct es un protocolo relativamente nuevo que nació de la necesidad de poder conectar dispositivos próximos utilizando tecnologías WiFi sin la necesidad de contar con infraestructura (access points). Este protocolo de comunicación resulta adecuado para los requerimientos de las aplicaciones que son ejecutadas en dispositivos no conectados a redes wifi pero si con dispositivos cercanos con compatibilidad WiFi Direct [18].

Dentro de este protocolo, se definen dos roles principales:

- Group Owner
Dispositivo que maneja la conexión del grupo. Se define por sorteo en el momento de la negociación de la conexión [28].
- Group Member
Dispositivo miembro del grupo, no tiene ninguna connotación relevante, salvo por el hecho de no ser el “Group Owner”.

5.2.2.2 API Android para Wifi-Direct

A partir de la versión 14 del SDK de Android, existe una API para este protocolo. [28] La misma expone dos formas de encontrar o descubrir dispositivos:

- ◆ Peer Discovery
- ◆ Service Discovery

La diferencia principal entre peer Discovery y service Discovery es que el primero sólo encuentra dispositivos mientras que el segundo encuentra dispositivos que exponen servicios específicos. En el contexto de una clase, llena de dispositivos cada uno pudiendo ejecutar distintas actividades (juegos), se considera el “Service Discovery” como mecanismo más adecuado para la conexión entre dispositivos: los jugadores de un juego específico podrían conectarse únicamente con otros jugadores del mismo juego. La alternativa es conectar a todos contra todos, validar luego de conectados si se está utilizando la misma actividad y en caso negativo cancelar la conexión y probar con otros dispositivos.

Lamentablemente, el “Service Discovery” resultó inestable de acuerdo a las pruebas realizadas: aproximadamente en la mitad de los casos, los dispositivos exponiendo servicios no son encontrados. Las pruebas se hicieron con distintos dispositivos como Nexus 5, Nexus 7, Galaxy S6 y la Tablet de XPrize Google Pixel C, obteniendo siempre resultados similares.

Otro inconveniente al tratar de conectar dispositivos, se presenta cuando ya existe una conexión entre dos y un tercero entra en modo “Discovery”. En algunos casos los dispositivos no son encontrados y en otros, al ser encontrados existen dos opciones:

- O bien el tercero intenta conectarse al group owner y la conexión es exitosa.
- O bien el tercero intenta conectarse al dispositivo que no es owner. Esta conexión falla y además interrumpe la conexión existente entre los otros dos dispositivos.
- Al intentar crear conexiones entre 3 dispositivos, la tasa de fallos es de aproximadamente 50%.

Debido a estos resultados, la funcionalidad del plugin Wifi Direct se vio reducida a la conexión entre dos dispositivos únicamente y utilizando el mecanismo “Peer Discovery”.

5.2.2.3 Implementación Wifi-Direct

A continuación, se muestra un diagrama de secuencia para un caso típico de conexión entre dos dispositivos.

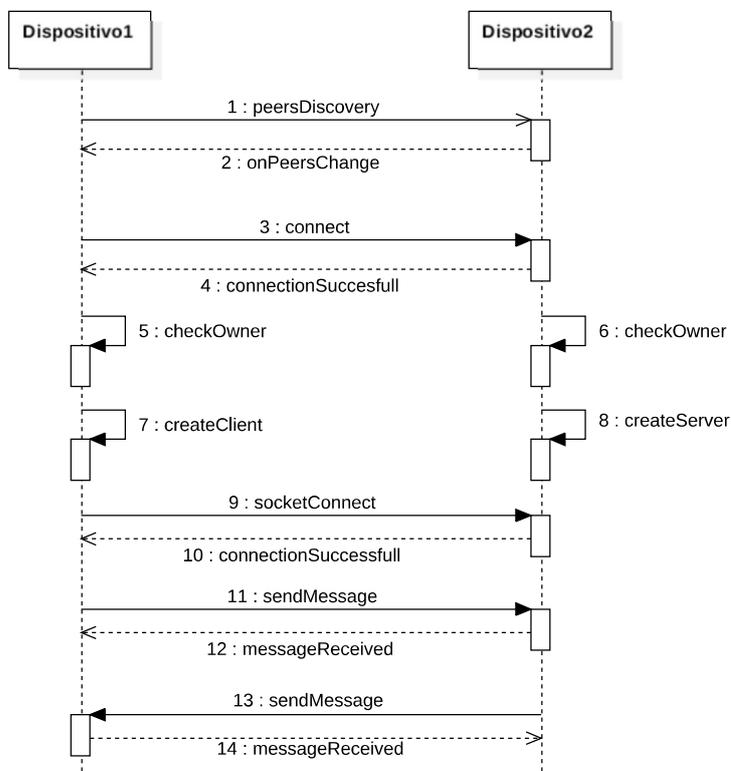


Imagen 9 - Diagrama de secuencia Wifi Direct

Cabe destacar que, en esta secuencia, podemos ver primero la conexión a nivel de capa de red entre dispositivos y luego la conexión por socket a nivel de capa de transporte (ambas capas pertenecientes al modelo de capas del conjunto de protocolos TCP/IP de redes).

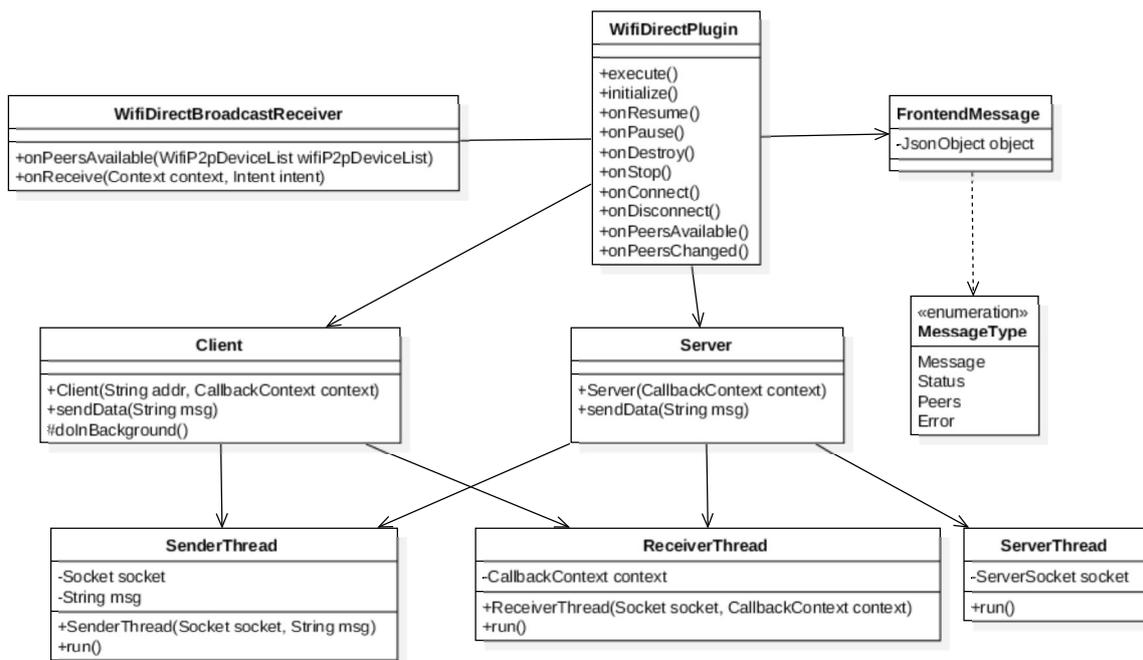


Imagen 10 - Wifi Direct - Diagrama de clases

La clase principal es WifiDirectPlugin, que extiende de CordovaPlugin. Como en todo plugin cordova, esta clase se encarga de recibir los llamados desde la vista web y procesarlos. También tiene control de eventos de la aplicación Android como: onResume, onPause, onStop, onDestroy y recibe notificaciones de los “intents” en la clase WifiDirectBroadcastReceiver.

Dado el funcionamiento de Wifi Direct, una vez conectados los dispositivos, uno recibe el rol de “GroupOwner” (GP). Los demás dispositivos al conectarse reciben la IP del GP y tienen la responsabilidad de abrir una conexión por socket en un puerto pre-especificado con el GP. Como el rol de “Owner” (dueño) es distinto al del resto de los dispositivos, esto se tiene que reflejar en el diseño del plugin. Es por eso que la clase WifiDirectPlugin tiene una instancia de Client y una de Server. Tanto Client como Server instancian SenderThread y ReceiverThread. Estos hilos son responsables de enviar y recibir información respectivamente. Además, existe la clase ServerThread que es el hilo con el ServerSocket esperando conexiones nuevas en el servidor.

6 Verificación

Este capítulo describe todo lo relevante al testing de la aplicación y sus componentes. Primero se presenta la metodología y el plan de verificación elaborado y posteriormente se presentan los resultados y conclusiones de la verificación.

6.1 Metodología de verificación

Fueron evaluadas distintas formas de realizar la verificación de la solución:

- Test unitarios: debido al funcionamiento de los plugins de cordova, la mayoría de los métodos no retornan un resultado. Existe la posibilidad de crear objetiso de prueba y verificar que desde un método A se esté llamando otro método B pero se consideró que este tipo de test no generaba suficiente valor para la verificación.
- Test de interfaz de usuario web: presentan los siguientes inconvenientes
 - El plugin wifi direct necesita el dispositivo de red y este no funciona sobre un emulador.
 - El plugin de audio o text to speech hacen imposible la validación de si se reproduce el audio desde la interfaz web.

La opción escogida es la de pruebas funcionales sobre el framework y el generador, usando un enfoque de “caja negra” [29]. En base a los requerimientos, se verifican los resultados obtenidos, comparados con los resultados esperados.

Se propone un conjunto de casos de prueba para cada una de las funcionalidades definidas. Para esto se utilizan distintas técnicas, tratando de abarcar la mayor cantidad de escenarios significativos. Algunas de las técnicas utilizadas:

- Pruebas de partición de equivalencia [30]

Dado que no es posible probar las funcionalidades sobre todo el dominio de valores de entrada posibles, se definen clases de equivalencia de los mismos.
- Pruebas de valor límite

En lugar de seleccionar cualquier elemento como representativo de la clase de equivalencia, el análisis de valor límite requiere que se seleccionen uno o más elementos de forma tal que cada margen de la clase de equivalencia sea sujeto a una prueba [31]

Ejemplo: al probar la funcionalidad de “texto a audio” se prueban valores límite para el parámetro de velocidad de reproducción
- Pruebas en base al estado [32]

Este tipo de pruebas se definen en base a los distintos estados que puede atravesar una aplicación. Ejemplo de uso: esta técnica de prueba fue muy útil a la hora de probar el plugin de Wifi-Direct. En esta funcionalidad se recorren distintos estados y las pruebas fueron basadas en los mismos.

Además del conjunto de pruebas definidas, para cada uno de los componentes del framework se genera una aplicación particular, con el objetivo de probar cada una de las funcionalidades de la misma forma que lo haría una aplicación final. Para este tipo de pruebas se utiliza la técnica de test exploratorio [33]. Las aplicaciones generadas para las pruebas fueron ejecutadas con distintas configuraciones del emulador variando versiones del sistema operativo, en dispositivos físicos (celulares y tabletas del equipo) y también en la tableta entregada por XPrize en el mes de agosto 2016. A continuación se presentan las características de este dispositivo:

Especificaciones del dispositivo donde corren las aplicaciones

Plataforma	OS	Android OS, v6.0.1 (Marshmallow), actualizable a v7.0 (Nougat)
	Chipset	Nvidia Tegra X1
	CPU	Quad-core 1.9 GHz
	GPU	Nvidia Maxwell
Memoria	Interna	32/64 GB, 3 GB Ram
Cámara	Primaria	8 MP, f/2.4
	Features	Si
	Video	Si
	Secundaria	2 MP
Sonido	Tipos de Alertas	Vibración; MP3, WAV ringtones
	Parlantes	Si, Stereo
	3.5 mm jack	Si
Comunicación	WLAN	Wi-Fi 802.11 a/b/g/n, dual-band, Wi-Fi Direct, hotspot
	Bluetooth	V4.1, A2DP
	GPS	
	USB	V3.0, Type-C conector reversible
Features	Sensores	Acelerómetro, giro, proximidad, compás
	Mensajería	Email, Push mail, IM
	Browser	HTML5
		MP4/H.264 player , MP3/WAV/eAAC+B21/Flac player Photo/video editor Document viewer
Batería		No removible 34,2 Wh

Para la verificación de las funcionalidades también se contó con la participación activa de integrantes del proyecto “Aspectos de Interfaz e Interacción” implementando las actividades educativas sobre el framework. De esta manera se obtuvo feedback en etapas tempranas del desarrollo, sobre las distintas funcionalidades expuestas.

6.2 Casos de prueba

A continuación, se detalla para cada uno de los componentes (generador, plugins), una breve descripción de los casos de prueba utilizados en el proceso de verificación junto con las técnicas empleadas. La totalidad de pruebas realizadas se encuentra documentada en el anexo Pruebas [34]

- Casos de prueba del generador

Para probar el generador, la mayoría de los casos consisten en la creación de proyectos variando las relaciones de precedencia, incluyendo un caso borde sin actividades. Técnicas utilizadas: partición de equivalencia, valor límite.

- Casos de prueba para plugin “Text to speech”

Con respecto a la verificación del plugin de transformación de texto a audio, se varían los parámetros de entrada verificando los resultados, tanto para los casos donde los parámetros son válidos como cuando no lo son. Se utilizan también las opciones por defecto. Técnicas utilizadas: partición de equivalencia, valor límite.

- Casos de prueba para plugin “Logging”

Se prueban varios casos donde la fecha varía o donde los parámetros lo hacen, incluyendo algunos casos borde con parámetros vacíos, validando que en todos los casos los archivos con los logs son generados correctamente.

- Casos de prueba para el plugin “Media”

Dada la cantidad de operaciones distintas que ofrece el plugin para la reproducción de audio como ser play, seek, getDuration, se realizan pruebas variadas con cada una de las mencionadas funcionalidades. Para definir las pruebas también se tiene en cuenta el estado del plugin al momento de llamar las funcionalidades. Técnicas utilizadas: partición de equivalencia, pruebas en base al estado.

- Casos de prueba para plugin “Video”

Para la validación del plugin de video, se varían los formatos de video a probar y se valida que los recursos utilizados por el reproductor sean correctamente liberados. Dado que esta funcionalidad también maneja algunos estados, se definen los casos en función de los mismos. Técnicas utilizadas: partición de equivalencia, valor límite, pruebas en base al estado.

- Casos de prueba para plugin “Camera”

Se prueban las dos funcionalidades “takePhoto” y “choosePhoto” variando los parámetros y verificando opciones por defecto en los casos que corresponde. Técnicas utilizadas: partición de equivalencia, valor límite.

- Casos de prueba para plugin “Wifi-direct”

En particular se prueban las distintas funcionalidades y los distintos estados por los cuales pasa el plugin, incluyendo algunos casos borde. Técnicas utilizadas: valor límite, pruebas en base al estado.

6.3 Resultado Verificación

Con respecto a los componentes se cumplió con la verificación funcional en su totalidad. A continuación se presentan los resultados de las pruebas funcionales. Tal como se estableció en el plan de verificación, estas pruebas se realizaron utilizando un enfoque de caja negra sobre cada una de las funcionalidades provistas por los componentes para probar diferentes combinaciones de datos de entrada de la misma forma que lo haría un sistema externo.

En total se planificaron 76 casos de prueba. En la imagen siguiente se muestran los resultados de las pruebas por funcionalidad detallando la cantidad de casos de prueba con resultado positivo (columna OK), los casos de prueba que fallaron (columna Falla) y el total por componente.

	OK	FALLA	Total
Generador	9	1	10
TTService	6	2	8
LogService	3	5	8
MediaService	9	6	15
VideoService	7	0	7
CameraService	12	5	17
WifiDirectService	8	3	11

Imagen 11 – Resumen Fallas

Estos resultados arrojan un total de fallas detectadas del 29% sobre los casos de prueba planteados como se puede apreciar siguiente imagen.

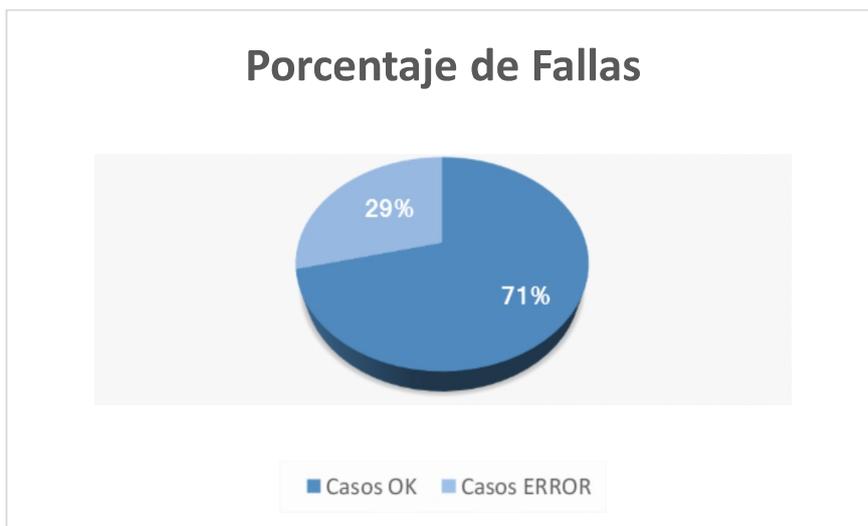


Imagen 12 – Porcentaje de Fallas

Dentro de las fallas detectadas, la mayor cantidad están asociadas con las funcionalidades relacionadas al plugin MediaService con un total del 27% como se puede apreciar en la siguiente imagen.



Imagen 13 – Fallas por Componente

Las fallas detectadas fueron corregidas en su totalidad.

7 Casos de estudio

En este capítulo se presentan dos casos de estudio que hacen uso de la solución propuesta por el proyecto.

7.1 Batalla Naval

Para probar las funcionalidades provistas por el framework se implementa una actividad educativa que simula el clásico juego “Batalla Naval”.

7.1.1 Descripción

Cada jugador maneja dos tableros divididos en casillas. Cada tablero representa una zona diferente del mar abierto: la propia y la contraria. En uno de los tableros, el jugador coloca sus barcos y registra los «tiros» del oponente; en el otro, se registran los tiros propios. Al comenzar, cada jugador posiciona sus barcos en el primer tablero, de forma secreta, invisible al oponente. Cada quien ocupa, según sus preferencias, una misma cantidad de casillas, horizontal y/o verticalmente, las que representan sus naves. Una vez todas las naves han sido posicionadas, se inicia una serie de rondas. En cada ronda, cada jugador en su turno "dispara" hacia la flota de su oponente indicando una posición (las coordenadas de una casilla). El juego termina cuando uno de los dos jugadores destruya primero todas las naves de su oponente.

Se decide mantener el enfoque sobre los aspectos funcionales y no en los aspectos gráficos ya que el objetivo de la implementación de esta prueba de concepto es probar la facilidad en el uso de los periféricos, así como también la utilidad del generador en la inicialización del proyecto. Resulta interesante este juego ya que permite el uso de varios de los periféricos que fueron tenidos en cuenta dentro de los requerimientos del proyecto.

7.1.2 Flujo

En el siguiente diagrama, se muestra la secuencia de mensajes que se comparten entre los dos dispositivos durante una partida de Batalla Naval.

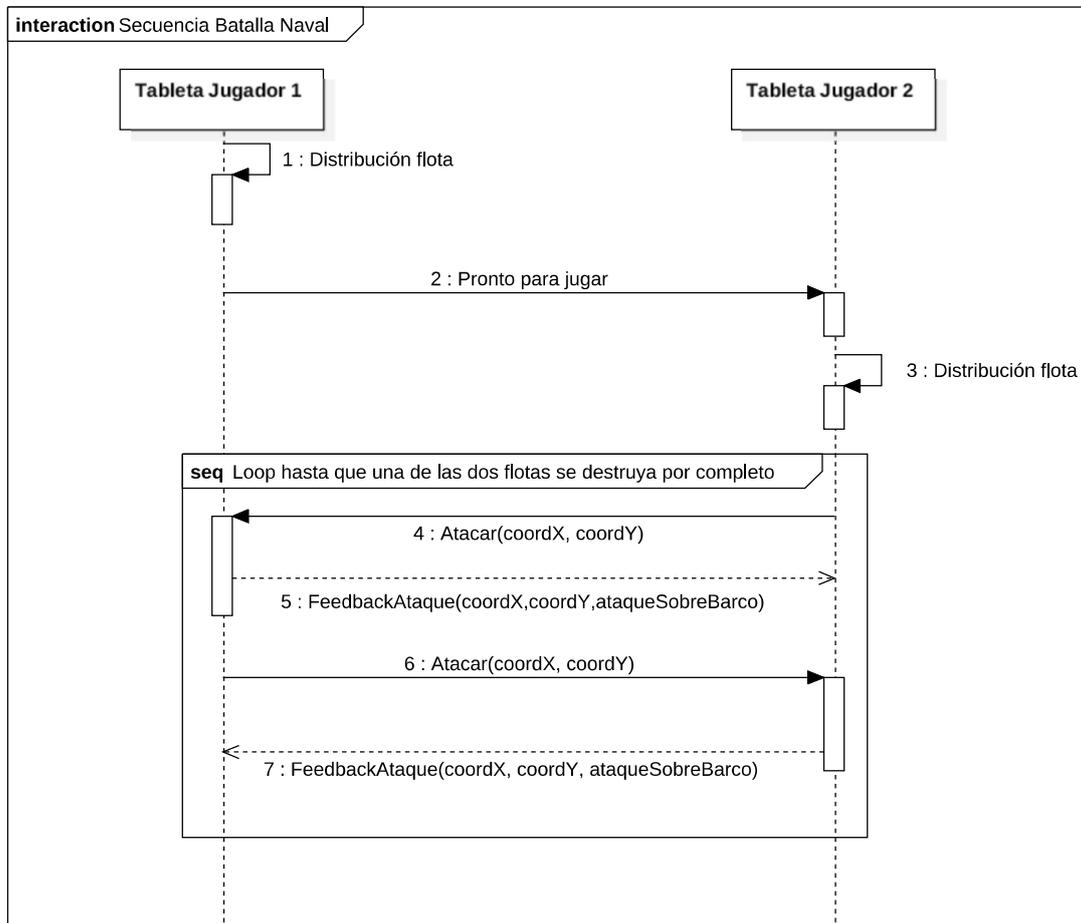


Imagen 14 - Diagrama de secuencia Batalla Naval

7.1.3 Interfaz de usuario

Las siguientes imágenes son capturas de pantalla de la interfaz de usuario del juego Batalla Naval:

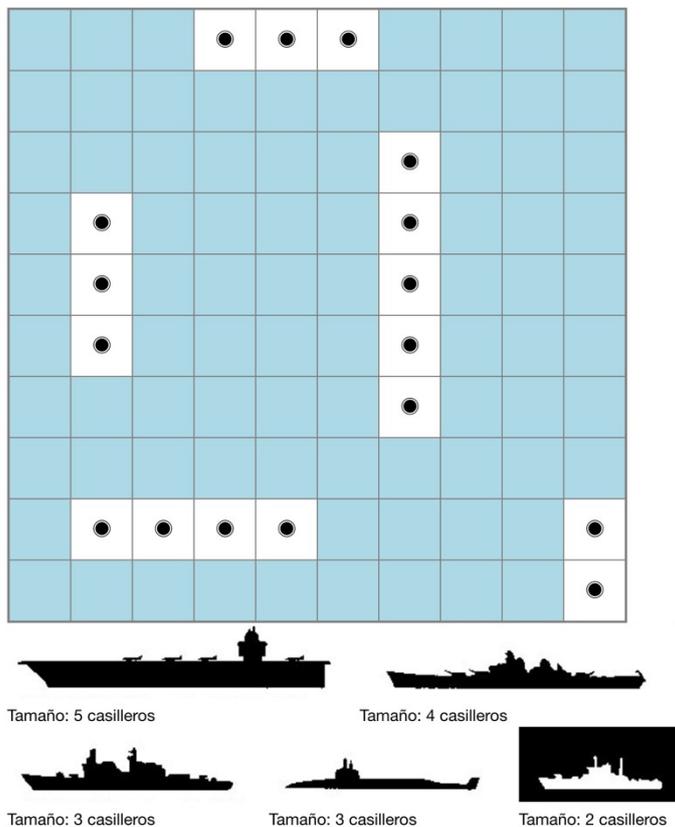


Imagen 15 - Disposición inicial de flota

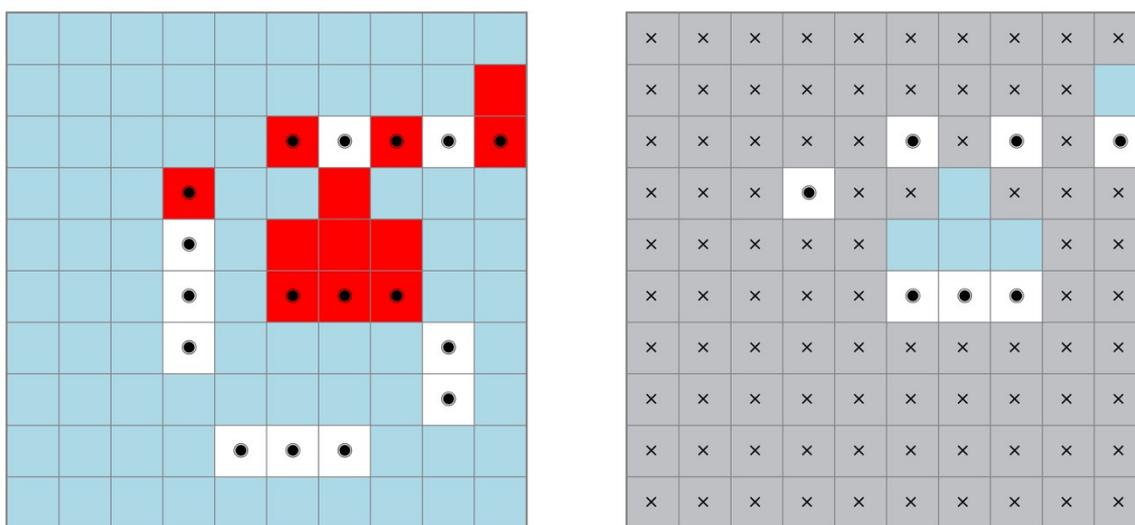


Imagen 16 - Izquierda: tablero propio jugador 1, derecha: tablero jugador 2 mostrando situación de jugador 1

7.1.4 Plugins utilizados

A continuación, se detalla el uso de los distintos plugins dentro de la actividad:

- Reproductor de video para reproducir el video al inicio del juego (portada)
- Reproductor de sonido para los distintos efectos de sonido del juego
- Plugin de Logging para poder persistir en el sistema de archivos, información sobre la utilización de la actividad
- Plugin wifi-direct para poder enviar y recibir todos los mensajes y que el juego se pueda desarrollar entre dos dispositivos diferentes.

7.1.5 Análisis y Conclusiones de utilización del framework

La siguiente tabla muestra las líneas de código implementadas exclusivamente para esta actividad y las líneas de código de la aplicación en total:

	Lineas de código
Javascript actividad "Batalla Naval"	462
Javascript total	1442
Java total	4247
Total Javascript + Java	5689

De las 462 líneas de código para la batalla naval, menos de 20 líneas hacen llamadas a los distintos plugins para reproducir sonido, video, conexión entre dispositivos o creación de registros (“logging”).

Luego de la implementación de esta prueba de concepto, se puede concluir lo siguiente:

- La mayor complejidad en la implementación de esta actividad educativa, se encuentra del lado de la interacción con el usuario y las interfaces gráficas. Por otra parte, el uso de periféricos para reproducir sonido o enviar mensajes entre dispositivos es resuelto en menos del 5% de las líneas de código. Esto prueba que el objetivo de enfocar al usuario del framework en el diseño e interacción se logra de manera exitosa.
- El uso de servicios angular envolviendo a las APIs javascript de los distintos plugins, prueba ser un mecanismo sencillo de integración entre los controladores angular y las funcionalidades expuestas desde cordova.
- El uso de tecnologías web para el desarrollo de las interfaces e interacción resulta ser una buena elección, no solo por la situación del equipo que utiliza esta solución (ya que son conocedores de las tecnologías mencionadas) sino también por el soporte de las comunidades de cordova, node.js, angular.js e Ionic.
- El manual de usuario beneficia al usuario del framework, siendo una forma rápida y clara de verificar pasos de inicialización o aspectos de las distintas funcionalidades expuestas.

7.2 “Aspectos de Interfaz e Interacción”

7.2.1 Introducción

A continuación, se encuentran las actividades educativas generadas por el grupo “Aspectos de Interfaz e Interacción” utilizando el framework. Estas actividades educativas componen a la aplicación educativa que se presenta en el concurso “Global Learning XPrize”, por parte del equipo uruguayo XPrize y es implementada sobre las bases del framework de este trabajo.

Cada actividad define qué área trabaja (Lengua o Matemáticas), el objetivo a nivel educativo y se describe de qué trata la actividad en sí misma (consigna) así como también que aspectos educativos que se manejan.

Para la creación de estas actividades se contó con la colaboración de María Alcaraz en el diseño de recursos gráficos. Sus aportes ayudan a crear una única línea de diseño entre las distintas actividades educativas, teniendo cada una, recursos creados específicamente para ser utilizados en esta aplicación educativa.

Es importante destacar que al comienzo de ambos proyectos (“Interfaz e Interacción” junto al presente “Uso de periféricos”), para las voces de los juegos se decide utilizar la sintetización de voz. Por esto, el plugin “Text to speech” es parte de la funcionalidad ofrecida. Durante las siguientes etapas y en función de generar aplicaciones educativas con mayor capacidad de atraer al público objetivo, se cambia el uso de la sintetización de voz en favor de grabaciones reales. En este momento se decide agregar al alcance del proyecto, mecanismos para reproducir audio y video.

El equipo cuenta también con el aporte del locutor Joaquín Bianco, quien se encarga de grabar audio en inglés y español para ser utilizado en estas actividades educativas.

Los integrantes del grupo “Aspectos de Interfaz e interacción” comienzan la implementación de la aplicación educativa utilizando el generador. Introducen nombres para las distintas actividades educativas a implementar y configuran las relaciones de precedencia entre las mismas.

Luego de este paso, cuentan con una aplicación constituida por actividades educativas sin implementar y la pantalla de inicio donde se muestran las actividades disponibles. Durante la implementación de actividades, se utilizan las distintas funcionalidades del framework para la reproducción de audio y video, persistencia de registros sobre patrones de uso de la aplicación, etc. Además, es responsabilidad de los desarrolladores de actividades, definir cuando el usuario final logra finalizar una actividad educativa. Utilizando mecanismos provistos por el framework, la actividad se marca como finalizada y con esto se hacen visibles en la pantalla inicial, las actividades que la suceden.

7.2.2 Listado de actividades

A continuación, se listan algunas actividades que componen la aplicación implementada por el grupo “Aspectos de Interfaz e Interacción”. La totalidad de las actividades se encuentran en el documento anexo “Actividades Educativas”

7.2.2.1 Actividad 1

Área de trabajo: Lengua

Objetivo: Trabajar la decodificación de los sonidos de una palabra, así como también asociar el sonido de la letra con su símbolo.

Consigna: Seleccionar en cada columna la letra correcta para formar la palabra reproducida.

Aspectos educativos:

- Identificación de los sonidos que conforman una palabra
- Reconocimiento de sonido de letras
- Decodificación de sonido palabra

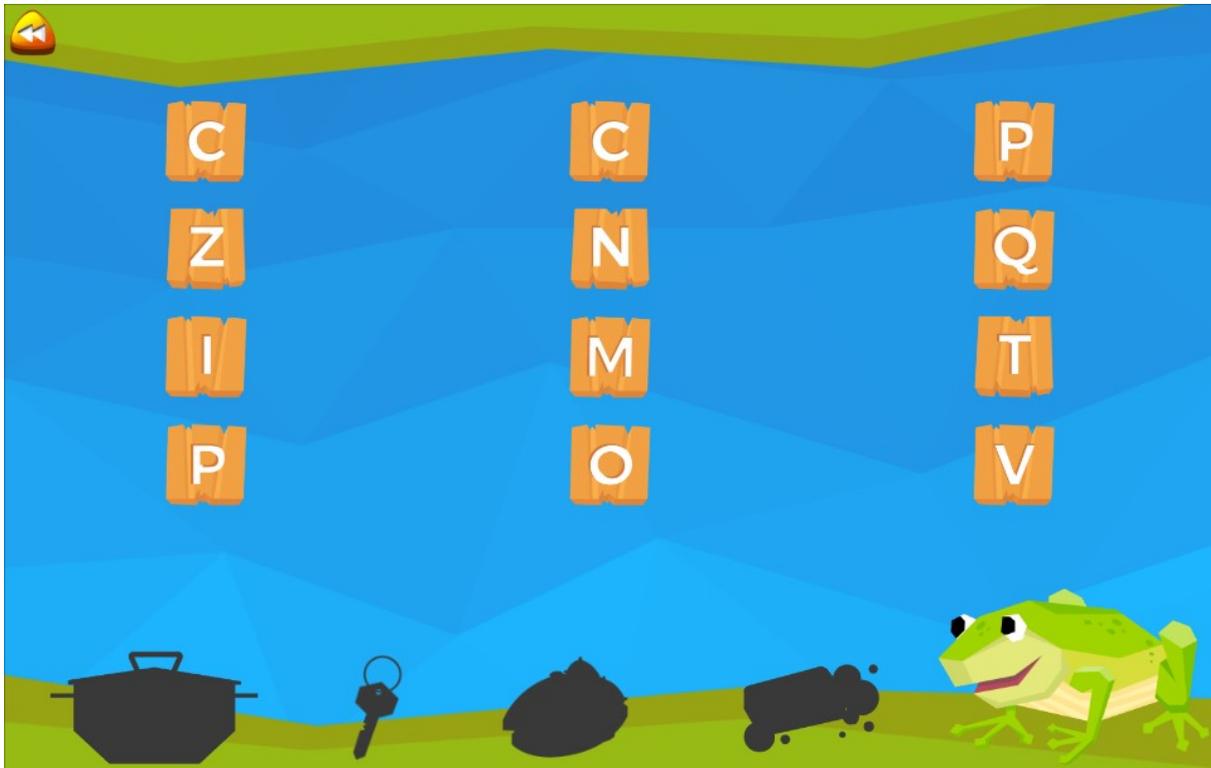


Imagen 17 - Diseño actividad 1

7.2.2.2 Actividad 2

Área de trabajo: Matemática

Objetivo: Identificar y reproducir patrones correctamente. Trabajar en el modelado mental de secuencias y órdenes.

Consigna: Seleccionar el color correcto para cada pluma, basándose en el patrón presentado

Aspectos educativos:

- Detección de patrones y secuencias

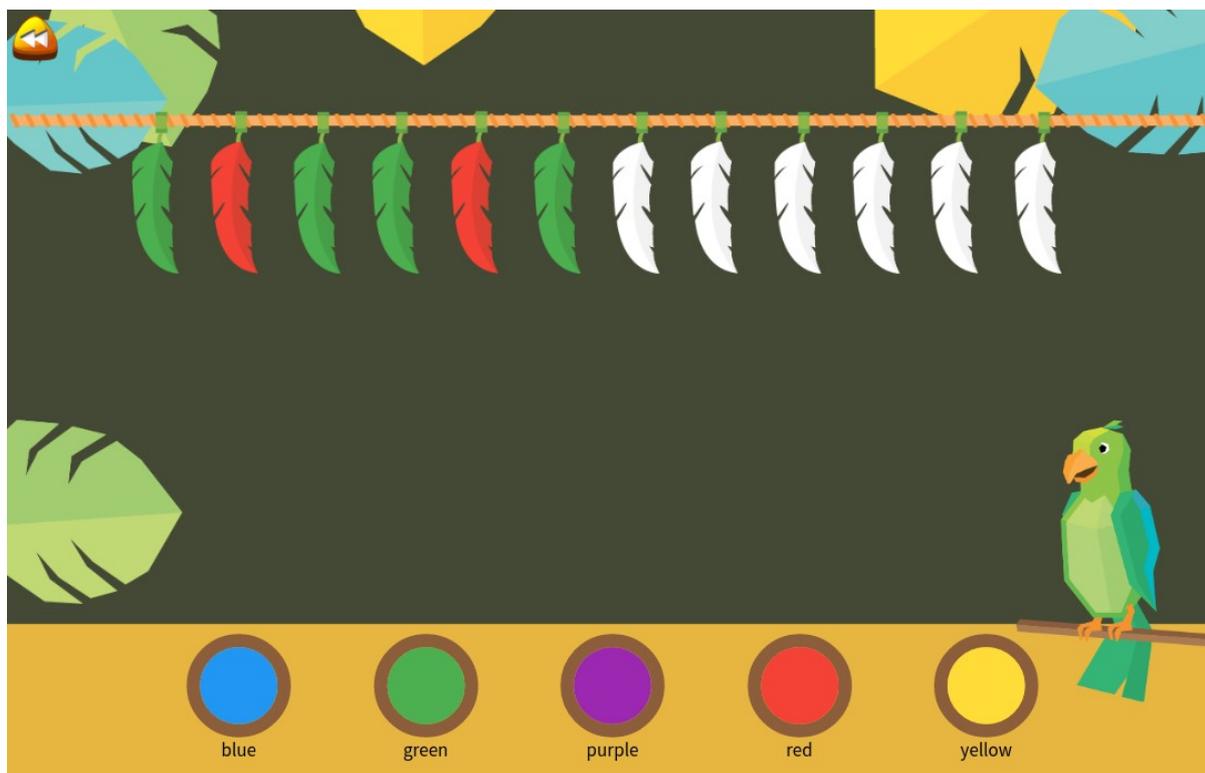


Imagen 18 - Diseño actividad 2

7.2.2.3 Actividad 3

Área de trabajo: Lengua

Objetivo: Reconocer la primera letra de una palabra y asociarla al símbolo.

Consigna: Arrastrar los objetos hacia la repisa correspondiente según la primer letra del mismo.

Aspectos educativos:

- Identificar letras
- Identificación de sonidos de las letras

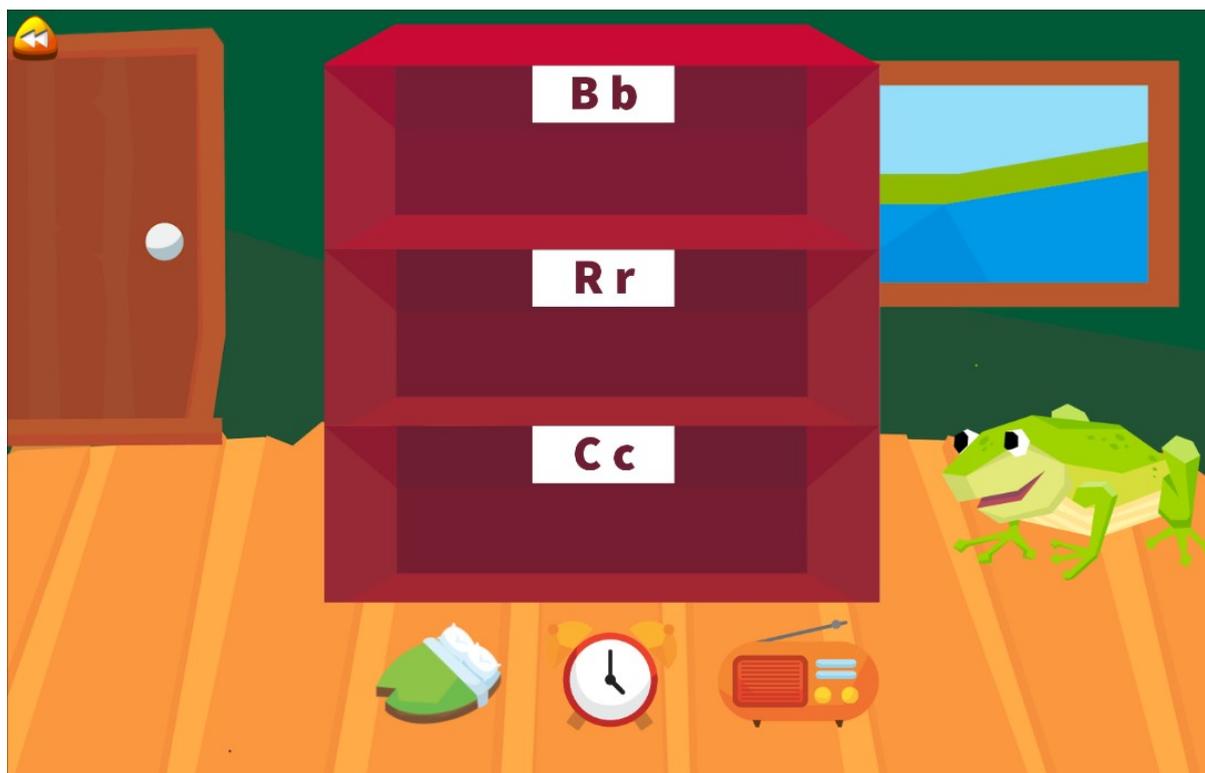


Imagen 19 - Diseño actividad 3

7.2.2.4 Actividad 4

Área de trabajo: Matemática

Objetivo: Asociar símbolos numéricos con cantidades

Consigna: Arrastrar la etiqueta con el número a la caja con la cantidad correcta

Aspectos educativos:

- Identificación de números
- Discriminación de números
- Alcanzar la precisión
- Sumas

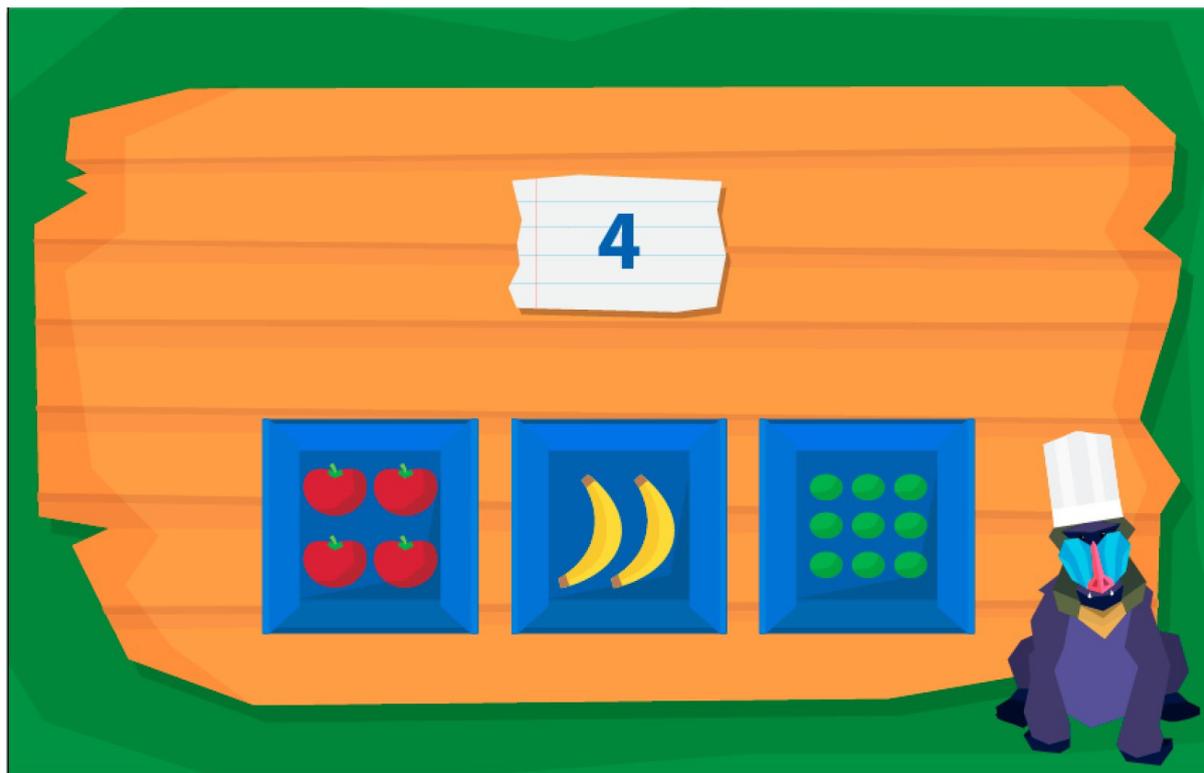


Imagen 20 - Diseño actividad 4

7.2.2.5 Actividad 5

Área de trabajo: Matemática

Objetivo: Identificar una secuencia numérica y detectar el número que falta para completarla

Consigna: Se presenta una secuencia numérica incompleta. El niño deberá analizar la secuencia e identificar el número que la completa de forma correcta entre las opciones.

Aspectos educativos:

- Identificación de números
- Detección de patrones y secuencias
- Sumas
- Modelado con matemática

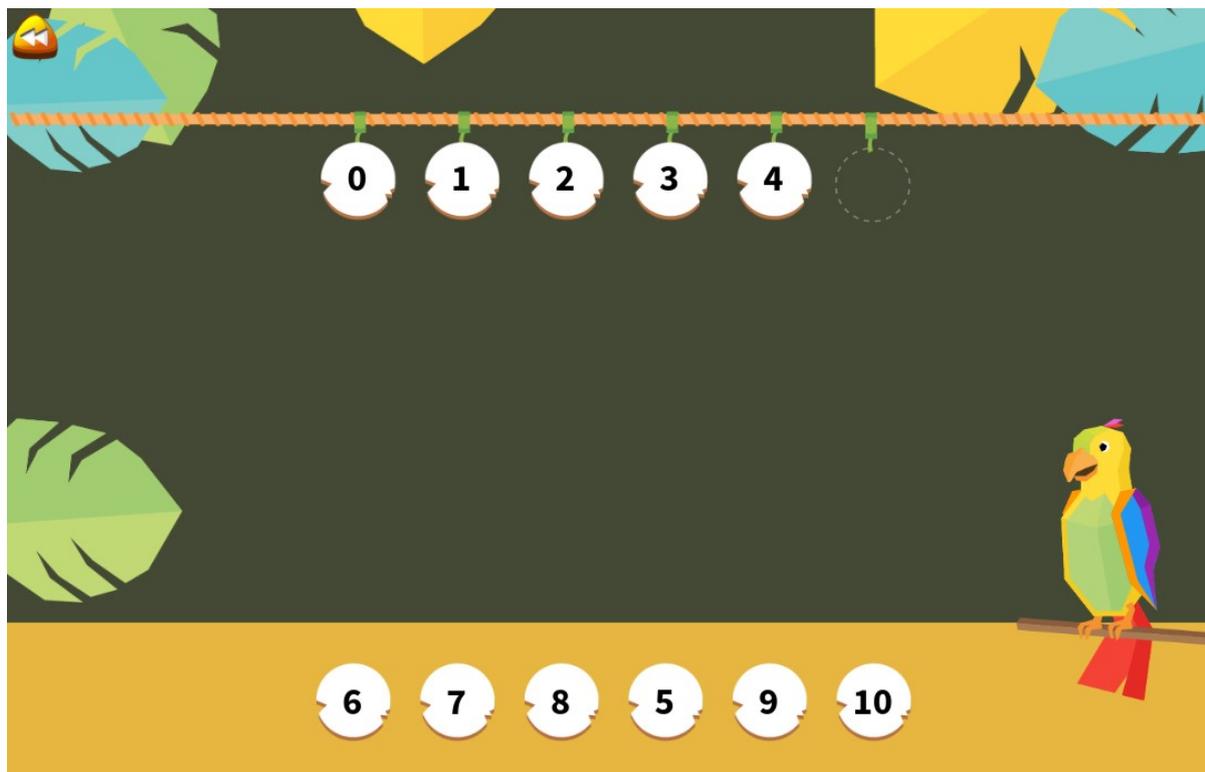


Imagen 21 - Diseño actividad 16

8 Gestión

En este capítulo se tratan los puntos relacionados a cómo se desarrolló el proyecto en general.

8.1 Planificación

La planificación del proyecto se divide en seis actividades:

- **Investigación:** se corresponde con el estudio del estado de desarrollo actual de las distintas tecnologías disponibles.
- **Relevamiento y Análisis de Requerimientos:** durante esta actividad, se realizan reuniones con el equipo formado por el cliente para definir las inquietudes, restricciones y requerimientos.
- **Arquitectura y Diseño:** en esta actividad se definen la arquitectura y el diseño. Se detectan riesgos técnicos, se estudian tecnologías prototipando posibles soluciones.
- **Implementación:** durante esta actividad se lleva a cabo el desarrollo de la solución diseñada.
- **Verificación:** en esta actividad se detectan y corrigen defectos de la implementación. Por lo tanto, habrá actividades de desarrollo que serán ejecutadas luego de comenzada la verificación. Por este motivo, se solapan las actividades de implementación y verificación.
- **Documentación:** esta tarea se ejecuta durante todo el proyecto y tiene como fin documentar todas las etapas anteriores.

En la siguiente imagen, se pueden apreciar la distribución y duración estimada de cada una de las etapas así como la duración total estimada del proyecto

id.	Nombre de tarea	T2 15			T3 15			T4 15			T1 16		
		Abr.	May.	Jun.	Jul.	Ago.	Set.	Oct.	Nov.	Dic.	Ene.	Feb.	Mar.
1	Investigación	■											
2	Relevamiento y Análisis de Requerimientos		■										
3	Arquitectura y Diseño				■								
4	Implementación						■						
5	Verificación								■				
6	Documentación	■											

Imagen 22 - Vista trimestral de la planificación

8.2 Ejecución

En la imagen siguiente se puede apreciar la distribución y duración real de cada una de las etapas del proyecto.

id.	Nombre de tarea	T2 15			T3 15			T4 15			T1 16			T2 16			T3 16			T4 16		
		Abr.	May.	Jun.	Jul.	Ago.	Set.	Oct.	Nov.	Dic.	Ene.	Feb.	Mar.	Abr.	May.	Jun.	Jul.	Ago.	Set.	Oct.	Nov.	Dic.
1	Investigación	█																				
2	Relevamiento y Análisis de Requerimientos	█			█			█									█					
3	Arquitectura y Diseño							█									█					
4	Implementación										█			█			█			█		
5	Verificación										█						█			█		
6	Documentación	█			█			█			█			█			█			█		

Imagen 24 – Vista trimestral Real – En azul el cambio de alcance

Como se ve en la imagen 24, la tarea de investigación se ejecuta en el tiempo estipulado. Por otra parte, se extiende la duración de la tarea de relevamiento y análisis de requerimientos debido a que los usuarios del framework (el grupo “Aspectos de Interfáz e Interacción”) debieron extender su fase de investigación. Además, en esta fase se definen requerimientos adicionales como la necesidad de un generador de proyectos y sus funcionalidades, lo que también consume tiempo adicional. El resultado del análisis de requerimientos estuvo bien definido a fines de Octubre. Esto supondría un problema al inicio del proyecto debido a las fechas estipuladas por Xprize para participar del concurso (presentación de aplicación a principios de 2016). Sin embargo, el itinerario de Xprize fue modificado durante el transcurso de este proyecto relajando las fechas, haciendo posible el logro de objetivos del cliente (la participación en el concurso). El calendario final del concurso se encuentra en capítulo “Marco de trabajo”, sección 2.2.2 “Contexto Xprize - cronograma” del presente documento.

Debido a motivos personales y a sabidas de que el calendario del concurso había sido modificado, el equipo realiza una pausa durante el periodo de Marzo a Mayo de 2016.

Durante el mes de Julio se re-evalúa el alcance del proyecto. Se entiende que, además de ser necesaria la funcionalidad de transformación de texto a audio, en algunas casos es también imprescindible utilizar grabaciones reales para lograr un mayor impacto sobre los usuarios finales de la aplicación. Esto se traduce a una nueva funcionalidad: la reproducción de audio y de video, lo cual implica una extensión en el relevamiento de requerimientos, diseño e implementación de la solución.

Durante la implementación, se realizan pruebas sobre dispositivos android personales. En agosto de 2016, se recibe un dispositivo por parte de XPrize, el cual será utilizado en las pruebas de campo de las aplicaciones que lleguen a esa instancia. Es por esto que se decide extender a un mes más de lo planificado las pruebas sobre este dispositivo.

Por ultimo, las tareas relacionadas a la generación de la documentación fueron realizadas en 3 períodos, el primero en base a la investigación, el segundo en base a los requerimientos y el tercero y último para cerrar el resto de los entregables.

9 Conclusiones y trabajo a futuro

Esta sección describe las conclusiones del trabajo, lo que se planteó hacer y lo que se hizo realmente, los resultados alcanzados, dificultades encontradas y posibles extensiones al trabajo.

9.1 Conclusiones

Se realizó una investigación de las distintas alternativas tecnológicas existentes que proveen una solución a la problemática planteada, las cuales se describen en el capítulo dos “Marco de Trabajo”. Se diseñó e implementó una solución que facilita la creación de aplicaciones de aprendizaje autónomo para niños, que permite a los usuarios de la solución (generador-framework) enfocarse en la creación de interfaces ricas y animadas, utilizando de manera sencilla los distintos periféricos y recursos existentes en las tabletas. Se logró un diseño que como característica principal permite extender las funcionalidades existentes o incluso agregar nuevas de forma simple. Por medio de la funcionalidad expuesta por el framework, se logró la abstracción de las APIs nativas del sistema operativo Android para un uso de los periféricos más sencillo, obviando varios detalles que son implementados dentro de cada plugin.

La plataforma cordova, resultó ser confiable y su documentación suficiente y clara. Además, cuenta con una comunidad de usuarios que hace más fácil la implementación de plugins específicos como fue nuestro caso.

Con respecto a las APIs nativas de Android, en general resultaron fiables y la documentación de las mismas son extensas y claras. La única excepción fue la API de Wifi-Direct que por un lado su uso resultó innecesariamente complejo y además no confiable, ya que algunos de los métodos expuestos no funcionan como lo prevé la especificación. En particular no se logró crear conexiones de red entre más de 2 dispositivos de forma sostenida. En otras palabras, los casos de conexiones exitosas de 3 o más dispositivos no superan el 50% de los casos totales.

Se generó un manual de usuario, siendo la guía inicial para un desarrollador web que comienza a utilizar el framework. En este documento se describen requisitos para la utilización del framework y generador, pasos de instalación, información del proyecto web creado e información detallada del uso de los distintos plugins que conforman al framework.

Se logró implementar una prueba de concepto utilizando la solución propuesta. Esta tarea permitió comprobar la utilidad del framework, facilitando el acceso a periféricos y enfocando al usuario de la solución a la resolución de problemas de interfaz e interacción.

Por otra parte, la solución es utilizada por el equipo de “Aspectos de Diseño e Interacción” obteniendo una respuesta positiva frente al uso de los distintos plugins y en particular de la funcionalidad de la pantalla inicial donde se maneja desde la solución, la lógica de aplicaciones terminadas y aplicaciones disponibles. Los conflictos de configuración, si bien existen, se ven reducidos al mínimo gracias al uso del manual de usuario.

La documentación técnica a ser presentada en el concurso Global Learning XPrize fue aprobada por el cliente del proyecto. Además, se logró el objetivo de exponer las funcionalidades del framework a tiempo, para que puedan ser utilizadas por el equipo de Diseño e Interacción. El objetivo de presentar la aplicación en el concurso, aún no fue alcanzado, pero al momento de la creación de este documento (diciembre de 2016), están dadas todas las condiciones para que en enero del 2017 el equipo XPrize Uruguayo pueda entregar la aplicación al concurso. Es este hito, el principal objetivo de ambos proyectos, definido por el cliente.

9.1.1 Restricciones

Dado que, desde el inicio, el objetivo es la implementación de aplicaciones que corren sobre Android, el framework no es compatible con otras plataformas móviles como iOS (iPhone).

Otra restricción que presenta en particular el plugin “Text to speech” es la compatibilidad únicamente con idiomas soportados por la API Android, con la ventaja de ser compatible con idiomas soportados a futuro por esta API. Es decir, no es necesario ningún cambio en la implementación para soportar nuevos idiomas que sean soportados por el sistema operativo en un futuro.

Con respecto al plugin “Wifi-Direct”, existe la restricción de no poder conectar más de dos dispositivos debido a la inestabilidad de la API ofrecida por el sistema operativo.

9.2 Trabajo a Futuro

En base a las restricciones presentadas en la sección 9.1.1 y a ideas durante las distintas etapas del proyecto, se presentan a continuación algunas extensiones que podrían llevarse a cabo en el futuro.

9.2.1 Soporte Wifi-Direct para tres o más dispositivos

Si bien esto depende del desarrollo y estabilidad de la API Wifi-Direct de Android, en un futuro se podrían implementar mejoras en el plugin, basándose a su vez en mejoras de la API original. En particular, la implementación de sockets realizada tiene en cuenta la restricción de dos dispositivos por lo que necesita un cambio no solo en la implementación sino también a nivel de diseño para poder soportar conexiones de más de dos dispositivos.

9.2.2 Políticas de rotación de logs

Con respecto al plugin de logging, se podría mejorar la separación entre archivos y el manejo adecuado y eficiente del espacio disponible. Luego de cierta cantidad definida de tamaño de almacenamiento, se podrían aplicar políticas de rotación, por ejemplo, eliminando registros más viejos y manteniendo una ventana más restringida de registros recientes.

9.2.3 Soporte a otros periféricos

La arquitectura definida hace sencilla la extensibilidad de la funcionalidad provista por el framework por lo que se podría agregar la utilización de otros periféricos que hoy en día no están disponibles o no fueron tenidos en cuenta por este proyecto. Algunos de los periféricos existentes que se podrían incluir son:

- acelerómetro, sensor de movimiento que detecta el cambio en movimiento en relación con la orientación actual del dispositivo.
- brújula, sensor que detecta la dirección o el rumbo al que apunta el dispositivo.
- vibración, plugin que proveerá la funcionalidad necesaria para que el dispositivo vibre.
- geolocalización, facilitara información sobre la ubicación del dispositivo, como la latitud y la longitud.
- notificaciones push, para poder recibir mensajes, alertas u otras notificaciones, incluso cuando la aplicación esté cerrada.

9.2.4 Colaboración con Plan Ceibal

Durante el desarrollo del proyecto, el cliente mantuvo contactos con Plan Ceibal, de donde se mostró interés en la creación de aplicaciones educativas. En un futuro se podría realizar un proyecto de características similares, donde el público objetivo fueran niños uruguayos.

9.2.5 Soporte de nuevos idiomas en plugin “text to speech”

La implementación actual del mencionado plugin, soporta únicamente los lenguajes de la API Android de transformación de texto a audio. Utilizando otras librerías de sintetización de voz, se puede lograr a futuro soporte a nuevos idiomas.

9.2.6 Extensión de plugins en ambiente con conexión a redes

Dadas las restricciones de conexión a internet definidas en las bases del concurso, todos los plugins fueron implementados teniendo en cuenta esto. En caso de extender el uso de las aplicaciones o otros escenarios donde haya conectividad, algunas funcionalidades pueden ser mejoradas. A modo de ejemplo:

- logging: en lugar de persistir durante meses archivos de log, se podrían implementar políticas de respaldo donde los desarrolladores de aplicaciones podrían obtener feedback en etapas más tempranas del uso
- actualización “over the air” (OTA) de aplicaciones educativas: en base al feedback de uso, se podrían realizar cambios sobre las aplicaciones ya en uso, mejorando la calidad de las aplicaciones (solucionando fallas) o incluso la experiencia del usuario en base a patrones encontrados en logs. También se podría extender el contenido de las mismas con el transcurso del tiempo, rotando las distintas actividades educativas, pero manteniendo la misma aplicación
- conectividad entre dispositivos: extender la capacidad de envío de mensajes entre dispositivos conectados por wifi-direct a dispositivos conectados a un router o Access Point

9.2.7 Soporte para múltiples plataformas

La implementación de los distintos plugins del framework cuenta únicamente con soporte Android. A futuro, implementando únicamente la parte nativa de los plugins, se podría ejecutar las mismas aplicaciones educativas en otras plataformas como iOS.

9.2.8 Mejora en la sucesión de actividades presentadas

En la implementación actual, las actividades se presentan en función de la configuración de sus previas y el estado de las mismas (finalizadas o no). Como punto a mejorar, se podrían plantear distintos caminos entre actividades, en base al desempeño de cada usuario, ofreciendo una experiencia personalizada. Ejemplo: si el usuario exhibe dificultades para terminar un tipo específico de actividad, ofrecer actividades extra para reforzar ese tipo de conocimientos. Por el contrario, si presenta una facilidad no esperada al resolver otras, se incrementa la dificultad de manera más rápida, ofreciendo actividades que a otros usuarios le aparecerían más adelante en el futuro.

10 Referencias

- [1] «Xprize Foundation,» [En línea]. Available: <http://www.xprize.org/>.
- [2] «XPrize - Who we are,» [En línea]. Available: <http://www.xprize.org/about/who-we-are>.
- [3] «Global Learning XPrize Guidelines,» [En línea]. Available: http://learning.xprize.org/sites/default/files/global_learning_xprize_guidelines_v4_03_11_2015.pdf.
- [4] «Global Learning XPrize - Perspectiva general,» [En línea]. Available: <http://learning.xprize.org/about/overview>.
- [5] «Global Learning XPrize,» [En línea].
- [6] «Early Grade Reading Assessment,» [En línea]. Available: <https://www.eddataglobal.org/reading/>.
- [7] «Early Grade Math Assessment,» [En línea].
- [8] «Android Development,» [En línea]. Available: <https://developer.android.com/develop/index.html>.
- [9] «Application Fundamentals,» [En línea]. Available: <https://developer.android.com/guide/components/fundamentals.html>.
- [10] «Corona SDK,» [En línea]. Available: <https://coronalabs.com/corona-sdk/>.
- [11] «Introduction to Corona,» [En línea]. Available: <https://docs.coronalabs.com/guide/programming/intro/index.html>.
- [12] «Android Webview Overview,» [En línea]. Available: <https://developer.chrome.com/multidevice/webview/overview>.
- [13] «Cordova Introduction,» [En línea]. Available: <https://cordova.apache.org/docs/en/latest/>.
- [14] A. Bradley, «Where does the ionic framework fit in,» [En línea]. Available: <http://blog.ionic.io/where-does-the-ionic-framework-fit-in/>.
- [15] M. Cantelon, M. Harter, T. Holowaychuk y N. Rajlich, «Welcome to node.js,» de *Node.js in action*.
- [16] «Getting started with Yeoman,» [En línea]. Available: <http://yeoman.io/learning/index.html>.
- [17] B. Muschko, «Next generation builds with Gradle,» de *Gradle in action*.
- [18] «Wifi Direct,» [En línea]. Available: <http://www.wi-fi.org/discover-wi-fi/wi-fi-direct>.
- [19] «Yeoman,» [En línea]. Available: <http://yeoman.io/>.
- [20] «Promesas,» [En línea]. Available: https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Global_Objects/Promise.
- [21] «Android Studio,» [En línea]. Available: <https://developer.android.com/studio/index.html>.
- [22] «Npm,» [En línea]. Available: <https://www.npmjs.com>.
- [23] «Gradle,» [En línea]. Available: <https://gradle.org/>.
- [24] «Genymotion,» [En línea]. Available: <https://www.genymotion.com/>.
- [25] «Gitlab,» [En línea]. Available: <https://gitlab.fing.edu.uy>.

- [26] «Git Extensions,» [En línea]. Available: <https://gitextensions.github.io/>.
- [27] «Source Tree,» [En línea]. Available: <https://www.sourcetreeapp.com/>.
- [28] «Referencia API Wifi Direct Android,» [En línea]. Available: <https://developer.android.com/guide/topics/connectivity/wifip2p.html>.
- [29] R. Patton, Software Testing, Indianapolis: Sams Publishing, 2005.
- [30] «Proyecto de ingeniería de software - Fing - Guías de Verificación,» [En línea]. Available: <https://www.fing.edu.uy/inco/cursos/ingsoft/pis/memoria/dvd01/experiencia2005/MUM/protest/guias/guias/particionE.htm>.
- [31] «Proyecto de ingeniería de software - Fing - Guías de Verificación,» [En línea]. Available: <https://www.fing.edu.uy/inco/cursos/ingsoft/pis/memoria/dvd01/experiencia2005/MUM/protest/guias/guias/valorLimite.htm>.
- [32] «Proyecto de ingeniería de software - Fing - Guías de verificación,» [En línea]. Available: <https://www.fing.edu.uy/inco/cursos/ingsoft/pis/memoria/dvd01/experiencia2005/MUM/protest/guias/guias/tecnicas/cubrimiento.htm>.
- [33] J. Bach, «Exploratory Test Explained,» [En línea]. Available: <http://www.satisfice.com/articles/et-article.pdf>.
- [34] «Node.js,» [En línea]. Available: <https://nodejs.org>.
- [35] «Apache Cordova,» [En línea]. Available: <https://cordova.apache.org/>.
- [36] «Git Downloads,» [En línea]. Available: <https://git-scm.com/downloads>.
- [37] «Ionic instalation,» [En línea]. Available: <http://ionicframework.com/docs/guide/installation.html>.
- [38] «Underscore.js,» [En línea]. Available: <http://underscorejs.org/>.
- [39] Farid Elías, Diego Rodríguez, «Manual de usuario».
- [40] Farid Elías, Diego Rodríguez, «Documento de Requerimientos».
- [41] Farid Elías, Diego Rodríguez, «Documento de Arquitectura».
- [42] «XPrize Foundation,» [En línea]. Available: <http://www.xprize.org/>.



La organización XPrize propone desafíos abiertos de interés mundial en el que pueden participar pequeños equipos para proponer una solución a un problema específico. Este proyecto se enmarca en la competencia "Global Learning Xprize". El objetivo de la misma es el desarrollo de aplicaciones educativas para tabletas que permitan el aprendizaje autónomo de niños.

Dada la complejidad que implica el desarrollo de aplicaciones educativas, el presente proyecto: "Sistema para Aprendizaje Autónomo de Niños - Uso de Periféricos" apunta a la simplificación del uso de los distintos periféricos disponibles en tabletas.



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

