

2014

Facultat d'Informàtica de
Barcelona (UPC)

Autor:

Rubén Rodríguez Alcalde

Fecha:

18/06/2014

[IMUSEUMS]

Aplicación móvil de ayuda para visitas a museos

Titulación: Ingeniería en Informática

Tribunal:

- Presidente: GUILLERMO GODOY BALIL
- Vocal: JOSÉ LUIS RUIZ MUÑOZ
- Secretario/Codirector: BORJA VALLES FUENTE
- Codirector: ALEJANDRO BALLARIN LATRE

CONTENIDO

Motivaciones.....	4
Introducción	5
Los museos	5
Las audioguías	5
Los museos y las apps.....	6
iMuseums y sus clientes potenciales	7
Android: El mercado de smartphones	8
Android: El mercado de Apps	10
Objetivos	11
Propuesta.....	12
Requisitos funcionales.....	12
Requisitos no funcionales.....	13
Arquitectura del proyecto	14
Planificación	15
Presupuesto inicial.....	15
Método de trabajo	17
Scrum	17
GIT y Bitbucket.....	19
Product backlog.....	20
Diseño técnico	21
Entorno de trabajo: Android Studio	22
Patrón Modelo Vista Controlador (MVC)	23

Arquitectura de ficheros de la app	23
Internacionalización	27
REST / JSON	27
Librerías de apoyo	28
Desarrollo	29
Sprint 1 – Versión básica	29
Sprint 2 – Añadiendo la lectura de texto	31
Sprint 3 – Reproducción de vídeos	32
Sprint 4 – El mapa	33
Sprint 5 – Refinado del mapa	35
Sprint 6 – Diseñar recorridos para los mapas	36
Sprint 7 – Pintar los recorridos del museo	37
Sprint 8 – Añadir las puertas a los mapas y mejorar el recorrido hacia ellas	39
Sprint 9 – Visibilidad del progreso de la visita y carrito de la compra	41
Sprint 10 – Toma de datos de museo real y multiidioma	44
Sprint 11 – Mis obras favoritas	46
Sprint 12 – Desplazamiento del mapa	48
Sprint 13 – Límites en el desplazamiento del mapa	49
Sprint 14 – Vídeos de museo y mejora del TTS	50
Sprint 15 – Mejora del TTS y dibujado del mapa	51
Sprint 16 – Primera integración con el Backend	52
Sprint 17 – Segunda integración con el Backend	53
Sprint 18 – Tercera Integración con el Backend	54

Sprint 19 – Cuarta integración con el Backend	55
Conclusiones	57
Presupuesto final vs Presupuesto inicial.....	57
Conclusiones sobre el proyecto	60
Conclusiones personales	60
Trabajo futuro.....	62
Glosario.....	63
Bibliografía	64
Anexos.....	65
Pantallas de la app.....	65

MOTIVACIONES

Inicialmente mi intención fue buscar un proyecto que me permitiera desarrollar alguna aplicación para Android. Antes de comenzar el proyecto, había hecho algún que otro curso de desarrollo para Android. Con ayuda del proyecto mejoraría mi experiencia con aplicaciones para Android, aumentando mi currículum profesional. Además me interesaba poder hacerlo con metodologías ágiles, como SCRUM.

Buscando un proyecto, encontré a Alex Ballarín y a Borja Valles. El proyecto consistía en implementar un SCRUM póker con una app de Android. Al ponerme en contacto con ellos, la respuesta fue negativa, había llegado tarde y otra persona se me había adelantado, pero Alex me ofreció comenzar un proyecto de similares características pero de temática distinta.

La app que se me estaba proponiendo era *iMuseums*, una guía de ayuda con información para movernos por un museo.

INTRODUCCIÓN

LOS MUSEOS

Todos conocemos lo que es un museo y, en algún momento de nuestra vida, todos hemos ido a alguno. Los hay grandes, pequeños, sobre ciencia, sobre arte, los que hablan de los dinosaurios, los que te muestran pinturas, o esculturas, hay museos de todo tipo.

A pesar de todas estas diferencias que encontramos entre ellos, todo lo que en ellos se expone necesita tener una explicación. La Mona Lisa no sería lo que es sin todas esas historias que hay detrás de ella, que no conoceríamos de no ser porque el museo en el que se expone, algunas de las curiosidades, que no todo el mundo conoce, salen a la luz en una placa al lado o nos la explica un guía o pagamos por el servicio de audioguía, que el museo pone a nuestra disposición. De la misma manera un cráneo de Triceratops, sería nada más que eso si no nos explicasen en el museo el porqué de sus cuernos, o la forma de su boca.

Como ya he dicho antes, toda esta información, suele entrar por varios canales: placas junto a los elementos en exposición, personas que nos hacen de guía, pequeños panfletos que podemos leer sin necesidad de estar junto a lo expuesto y audioguías, que nos van narrando las explicaciones según avanzamos o seleccionamos, en un mando de botones, la sala o frente a la obra en la que nos encontramos.

LAS AUDIOGUÍAS

Por el tipo de formato que utilizan, las audioguías son lo más parecido a una aplicación para smartphones o tablets que nos podemos encontrar. Aunque no sea cómoda, la audioguía es portátil y el visitante del museo puede seleccionar en todo momento lo que quiere oír en el entorno del museo. Además la información es oída y no tiene que ser leída, lo cual ayuda al visitante a relajar la vista, poder disfrutar en mayor medida de la obra y no tenerse que detener a leer la placa o la guía en papel.

LOS MUSEOS Y LAS APPS

De entre los museos que hacen uso de las nuevas tecnologías para difundir las obras que contienen nos encontramos con varios ejemplos.

El **MoMA**, por ejemplo, a 29 de agosto del 2012 había publicado cuatro apps para iOS (tres para iPad y una para iPhone), cada una de las cuales servía a un propósito distinto.

- **MoMA App (iPhone y Android):** Aplicación que sirve para que los visitantes puedan planear una visita al museo y ver las novedades del museo. Además contiene también un calendario sobre futuras exposiciones del museo.
- **Art Lab iPad App:** Esta app fue creada para permitir a sus usuarios crear dibujos, guardarlos, compartirlos...
- **AB EX NY iPad App:** Esta app permite al usuario ver información de las obras pertenecientes a la exposición *Abstract Expressionist New York*.
- **The MoMA Books iPad App:** Esta app contiene los libros publicados por el MoMA.

El museo del **Louvre** también ha publicado una app oficial con las 100 obras maestras que contiene el museo, según la descripción que tienen en su web más de 500 imágenes, con información del palacio y su historia.

Otro museo que ha publicado apps es el **Museo del Prado**, su *Second Canvas*. Esta app contiene las catorce obras maestras que expone el museo madrileño, permite contemplarlas en alta calidad desde una tableta.

El **Tate Modern** o **Museo de Arte Moderno** ha publicado también varias apps, la mayoría de pago. Entre las que son gratuitas nos encontramos con *Magic Tate Ball*, con solo agitar el Smartphone la aplicación lleva al usuario a contemplar una de las obras de arte del museo. La aplicación elige la obra a mostrar dependiendo de la posición GPS del usuario.

También el **Thyssen** tiene una app publicada que permite al usuario ver una agenda de las exposiciones y actividades del museo día a día. También ofrece rutas y propuestas temáticas a través de una selección de pinturas.

Estos son algunos de los ejemplos más conocidos de grandes museos con la capacidad suficiente para publicar, en su propio nombre, una app para dispositivos móviles, para sus visitantes

IMUSEUMS Y SUS CLIENTES POTENCIALES

La idea de crear una app en la que se pudiera incluir cualquier museo era la de llegar sobre todo a los museos más pequeños, los cuales no pueden hacer una inversión muy grande para poder publicar una app propia.

iMuseums les ofrece, entonces, una plataforma en la que pueden subir la información de sus exposiciones adaptada para la app, y así ofrecer la visita virtual que quisieran ofrecer a sus visitantes.

Los datos que se han tomado para hacer las pruebas de iMuseums son del museo Can Tinturé de Esplugues de Llobregat, un pequeño museo que expone la colección de baldosas de muestra de Salvador Miquel.

ANDROID: EL MERCADO DE SMARTPHONES

Antes de la aparición de los *smartphones*, el mercado de teléfonos móviles siempre había buscado tener dispositivos cada vez más pequeños. El teléfono móvil evolucionó de un aparato que, aunque era pequeño, no se podía llegar a guardar en un bolsillo, hasta los aparatos más pequeños en busca de la perfección que se había buscado durante tanto tiempo.

A partir del año 2000, cuando Microsoft lanzó al mercado Windows Mobile, la intención ha sido totalmente la contraria en ciertos aspectos. El mercado no busca dispositivos más grandes por el simple hecho de hacerlos más grandes, las especificaciones para los usuarios han cambiado. Las pantallas deben ser más grandes para que el usuario pueda ver bien todos los elementos que se reparten a lo largo y ancho de esta. Estos dispositivos dejan de ser teléfonos móviles a convertirse en *smartphones* (teléfonos inteligentes).

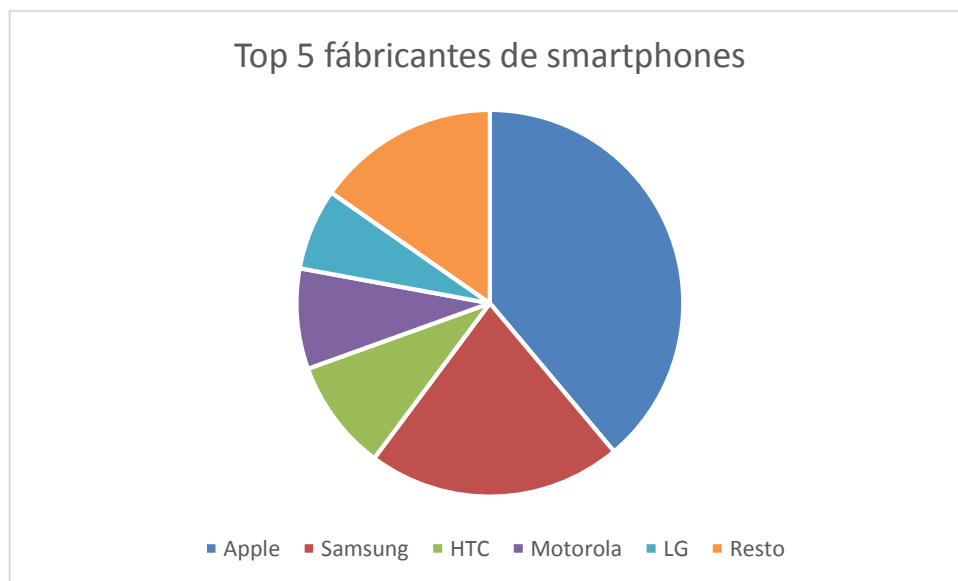


Ilustración 1 - Top 5 fabricantes de smartphones

En el panorama actual de los fabricantes de dispositivos, el mayor fabricante es Apple, seguido de Samsung y HTC. A pesar de esta clara ventaja en favor de la compañía de la manzana, este resultado se explica con los datos que se pueden observar en la ilustración.

Actualmente más de la mitad de *smartphones* del planeta tienen instalado Android en ellos, debido a que provienen de diferentes fabricantes y iOS solo se instala en *smartphones* fabricados por Apple, la suma de dispositivos de fabricantes que instalan Android es mucho mayor que los dispositivos que instalan iOS.

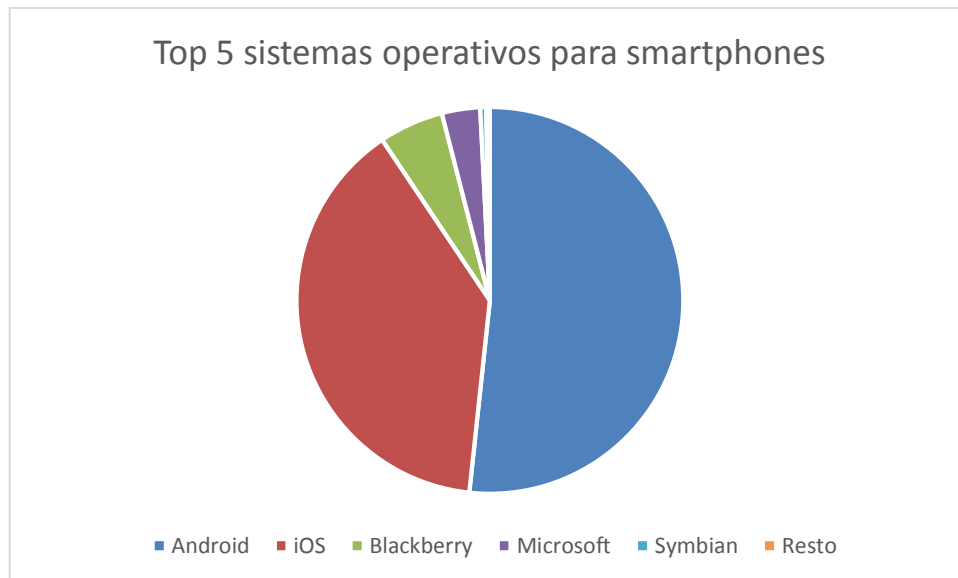


Ilustración 2 - Top 5 sistemas operativos para smartphones

Esta es una de las razones por las que, al final, decidí que Android era una plataforma mucho más acorde con el objetivo del proyecto. También ayudó en la decisión el factor precio de los dispositivos, que debido a la competencia y la gran variedad de fabricantes, los *smartphones* con Android tienen una gama de precios mucho más amplia, llegando a tener un precio mínimo bastante más reducido que los aparatos fabricados por Apple.

ANDROID: EL MERCADO DE APPS

En la gran batalla de las apps hay tres bandos en disputa, para ayudar en la elección de la plataforma sobre la que desarrollar también se hizo uso de datos relacionados con la cantidad de apps que se descargaban en cada *Marketplace*, ya que es un dato bastante significativo de la cantidad de usuarios que tiene cada una.

Los tres principales competidores a día de hoy son Google Play, App Store y Windows Marketplace.

Dado que Microsoft abandonó el desarrollo de Windows Mobile poco después que Google y Apple hubieran comenzado con los suyos, y no se volvió a meter con su Windows Phone hasta un año más tarde hace ya cuatro años, cuando fue presentado en el Mobile World Congress de 2010 en Barcelona. Por tanto Microsoft quedó muy atrasado en cuanto a evolución de su sistema operativo. Hace falta decir que, cuando iOS y Android evolucionaban favorablemente, creando sistemas más robustos, Windows Mobile se quedó estancado en la versión 6.5 llevando al sistema a caer en desuso.

Además de estos tres competidores de hoy en día muchos otros también se han querido posicionar en esta pelea, pero la mayoría han acabado cayendo irremediablemente, por diferentes motivos.

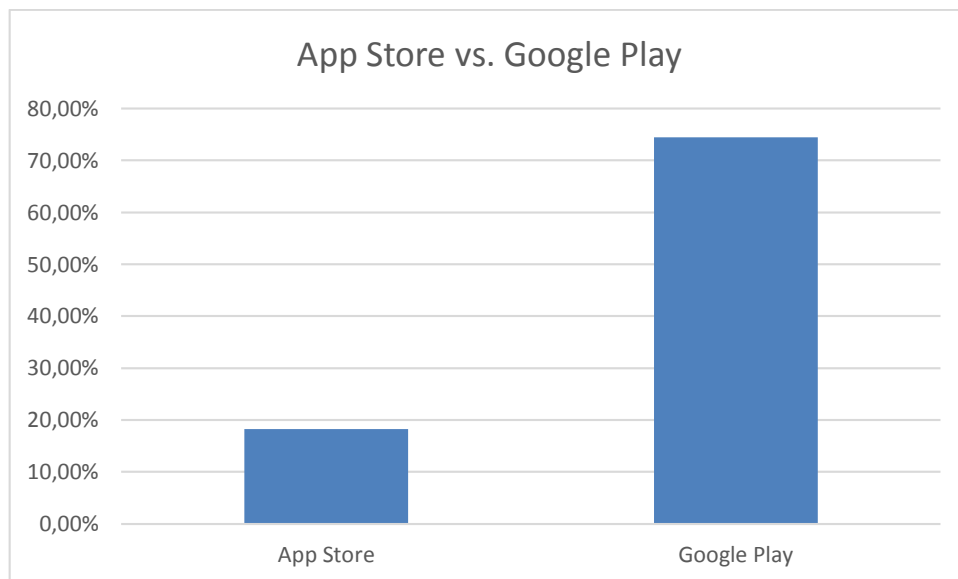


Ilustración 3 - App Store vs. Google Play

De entre los dos contrincantes dominantes que quedan en la ecuación, podemos ver por los datos registrados en los últimos años, que Google Play tiene mayor cuota de mercado en la descarga de apps, debido a que la gran mayoría de las apps que contiene son gratuitas.

OBJETIVOS

Una vez vistos algunos aspectos del mundo al que está dedicado la aplicación, y tomadas algunas decisiones técnicas, pasé a definir los objetivos de dicha aplicación.

Está claro que el objetivo principal es el desarrollo de una aplicación para móviles Android que permita hacer de guía en un museo de forma que pueda tener la información al alcance del dedo usando aparatos seguramente más ligeros, y todo centralizado en un solo dispositivo.

PROPUESTA

REQUISITOS FUNCIONALES

Los requisitos funcionales son aquellos requisitos del proyecto que definen el comportamiento de la aplicación.

Los requisitos funcionales son:

- El usuario podrá seleccionar el museo que desea visitar, y el recorrido que puede hacer en este.
- El usuario podrá descargar la información del museo, que debe poderse consultar offline.
- El usuario puede seleccionar un área del museo y ver la representación en mapa de esta.
- El usuario puede seleccionar una habitación en el mapa para ver una descripción y acceder a ella.
- El usuario puede navegar a través de los diferentes espacios que esa habitación o sala pueda tener (paredes, pedestales...).
- El usuario puede acceder a la información de una obra.
- El usuario puede ver la descripción de la obra.
- El usuario puede oír la descripción de la obra leída por su dispositivo.
- El usuario puede votar la obra y ver sus votos.
- El usuario puede ver una lista de vídeos relacionados con la obra seleccionada.
- El usuario puede añadir a un carrito las obras y gestionar ese carrito.

REQUISITOS NO FUNCIONALES

Los requisitos no funcionales son propiedades que mejoran el uso por parte del usuario sin necesidad de pertenecer al funcionamiento de la aplicación, es por esto que se tratan de manera distinta que el resto de requisitos.

Generalmente cuando se definen los requisitos no funcionales se suele usar terminología relacionada con la tecnología a usar en el proyecto.

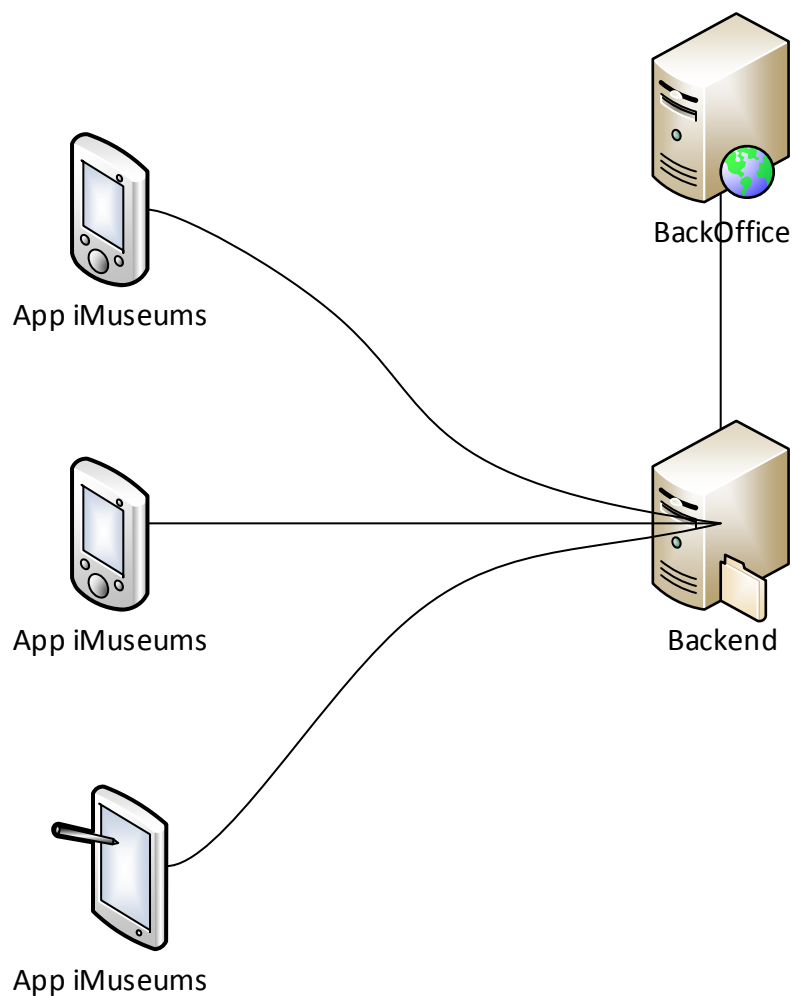
En nuestra aplicación tenemos los siguientes requisitos no funcionales:

- La aplicación estará preparada para dispositivos con Android 4.1.1.
- La aplicación permitirá guardar mediante SQLite los datos que puede gestionar el usuario en el dispositivo.
- La aplicación deberá descargar los paquetes y controlar que se puedan mantener actualizados en todo momento.
- La aplicación deberá conectarse con un servidor en la red para la descarga.
- La aplicación interpretará un modelo de datos de parte del servidor formateado en JSON.

ARQUITECTURA DEL PROYECTO

La arquitectura del proyecto consta de tres elementos principales.

- Servidor Backend, que servirá de repositorio de datos y será el encargado de ofrecer la información tanto al BackOffice y a la app.
- Servidor BackOffice, que es el que usarán los usuarios de los museos para dar de alta y modificar los datos que querrán que se vean en iMuseums, cuando los visitantes descarguen sus guías.
- App iMuseums, la app que se dotará de contenido para mostrar la guía que el usuario del museo ha dado de alta a través del BackOffice.



PLANIFICACIÓN

PRESUPUESTO INICIAL

La fase de desarrollo del proyecto se planificó según un cómputo de horas totales de trabajo, unas 500 horas, aproximadamente. Además de estas 500 horas de trabajo en el desarrollo, 250 horas adicionales han sido usadas para las fases de antes y después de la fase de desarrollo.

La fase inicial no se iba a prolongar más de una semana y media, con 30 horas era suficiente para definir la duración de un Sprint, decidir la tecnología con la que se iba a trabajar y finalmente crear una primera versión de Product Backlog.

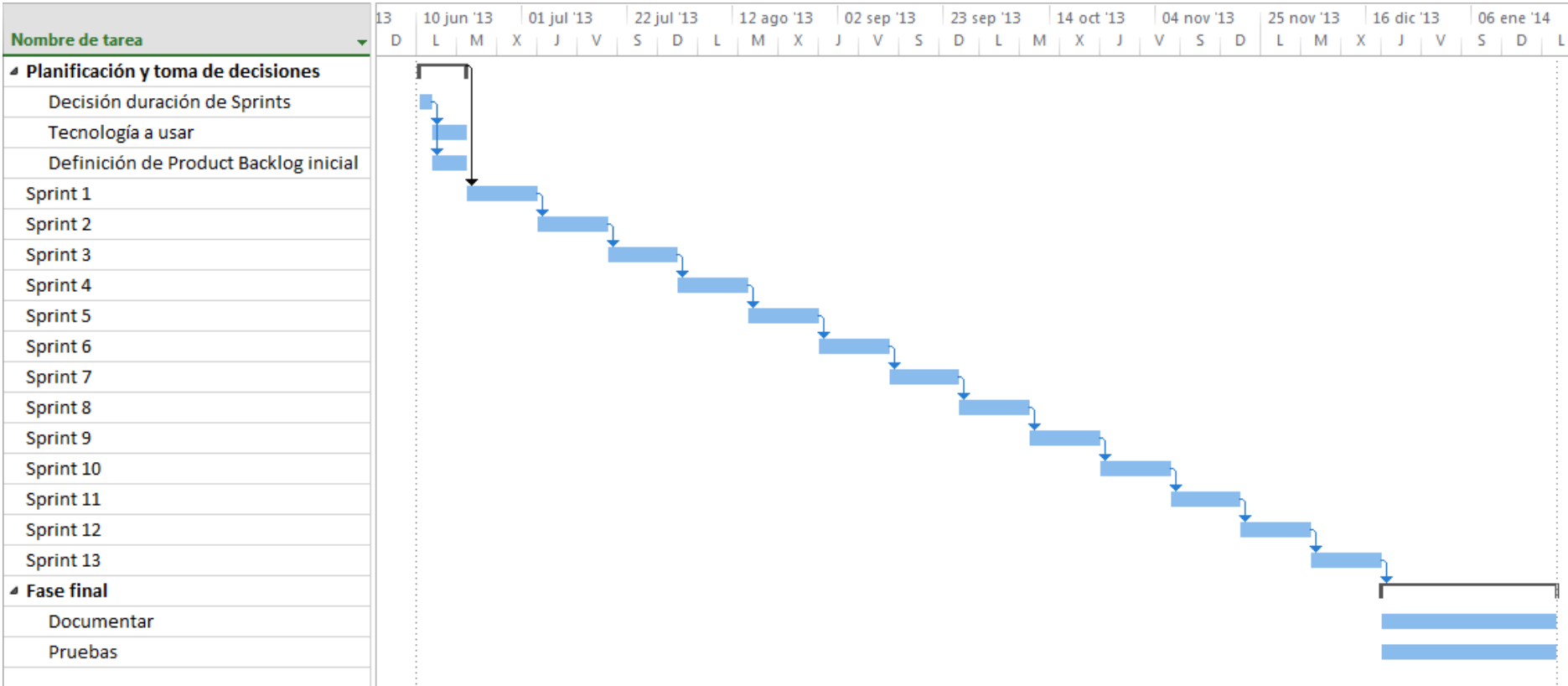
Finalmente, la fase de pruebas y documentación duraría las 200 horas restantes.

Ya que SCRUM es el método de trabajo elegido, lo primero que se tenía que hacer era definir un Sprint. En el caso de iMuseums el Sprint constaba de unas 40 horas aproximadamente, que ocupaban dos semanas, cada día se trabajaría unas 4 horas.

En un principio no se definieron tareas a hacer en cada Sprint, así que el Gantt presentado en la siguiente página mostrará un gráfico que coincide con la siguiente tabla de datos, además de las horas planificadas para las fases anterior y posterior a esta fase de desarrollo.

Sprint	Duración
Sprint 1	40 horas
Sprint 2	40 horas
Sprint 3	40 horas
Sprint 4	40 horas
Sprint 5	40 horas
Sprint 6	40 horas
Sprint 7	40 horas
Sprint 8	40 horas
Sprint 9	40 horas
Sprint 10	40 horas
Sprint 11	40 horas
Sprint 12	40 horas
Sprint 13	40 horas
Total de horas	520 horas

El inicio del proyecto fue el 17 de Junio de 2013, y estaba planificado para ser terminado el 13 de Diciembre de 2013. Faltaría entonces la escritura de la memoria y la fase de pruebas, que se resolvería a partir de ese momento.



Gantt 1 - Planificación inicial

MÉTODO DE TRABAJO

Para facilitar la toma de decisiones y la creación de los requisitos, tanto los funcionales como los no funcionales iMuseums se ha llevado a cabo con SCRUM como metodología de trabajo, el SCM usado para el desarrollo es GIT, y la plataforma GIT que se ha utilizado es Bitbucket de Atlassian.

SCRUM

Scrum es una metodología ágil de desarrollo de proyectos.

En el mundo del desarrollo de proyectos, los ingenieros de software nos encontramos día a día con un problema recurrente, que parece no tener solución, la falta de comunicación, documentaciones excesivamente extensas, que normalmente no suelen estar actualizadas, reuniones de definición de requisitos que duran varias jornadas consecutivas, en las que finalmente se acaba gastando más tiempo del necesario.

Bajo el Agile Manifesto, las metodologías ágiles pretenden solucionar muchos de estos puntos. Por supuesto, no son la solución a todos los males, ni todos los proyectos pueden ser llevados a cabo con métodos ágiles.

Scrum es uno de ellos y quizá el más popular. Mejora la comunicación entre los miembros del equipo, entre los que se encuentran un **Product Owner**, que es el que conoce cuál es el resultado final de lo que se está desarrollando; el **Development Team**, que se define como un equipo de desarrollo multidisciplinar y es el que construye el proyecto; y el **Scrum Master**, el facilitador que guía al equipo en cada paso de la metodología.

El componente principal de Scrum es la Lista de Producto (Product Backlog), que se compone de todas las Historias de Usuario (Casos de uso) que se desarrollarán en el producto de forma priorizada.

Scrum se mueve en torno al evento llamado Sprint, cada Sprint cuenta con un componente llamada Lista de Pendientes del Sprint (Sprint Backlog), que es la versión reducida del Product Backlog, con algunos de los elementos de ese que se realizarán durante la duración del Sprint.

Cada Sprint se compone de:

- Una Reunión de Planificación de Sprint (Sprint Planning Meeting), que intenta conseguir un Objetivo del Sprint (Sprint Goal).
- Diversas Reuniones de Scrum Diarias (Daily Scrum), para hacer el seguimiento día a día del Sprint.
- Finalmente se realiza una reunión que suele englobar una Revisión de Sprint (Sprint Review) y una Retrospectiva de Sprint (Sprint Retrospective).

Después de todo esto, se vuelve a iniciar un nuevo Sprint.

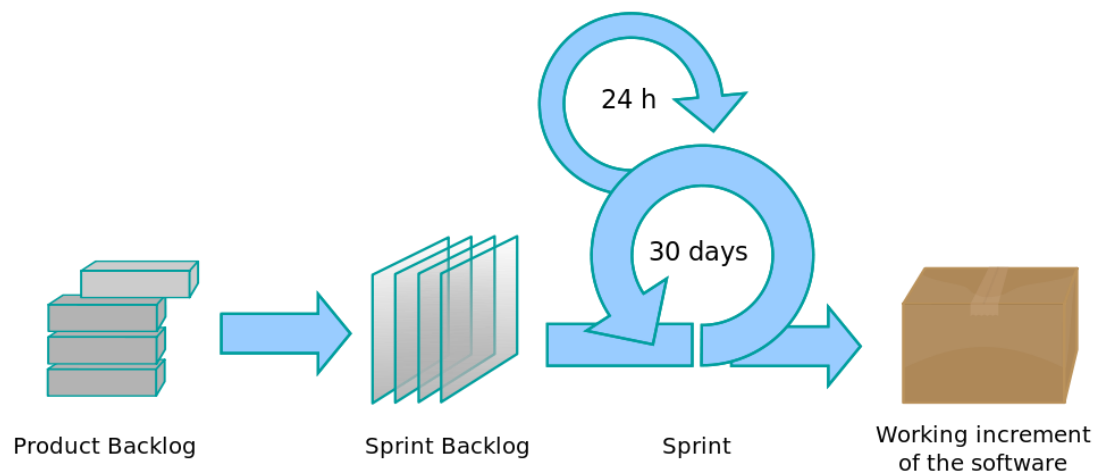


Ilustración 4 - El ciclo de vida de Scrum

Scrum incentiva la comunicación entre el Development Team con el Product Owner, la mayoría de eventos fuera del Sprint, que sirven para definir las historias de usuario, suelen tener como participantes el equipo entero de Scrum, que son el Product Owner, el Development Team y el Scrum Master. Esto facilita que el equipo de desarrollo tenga claro que es lo que el dueño del producto quiere hacer, y que el dueño del producto conozca de primera mano el proceso de creación del producto. Además, cada vez que se acaba un Sprint las tareas del Product Backlog se podrán repriorizar y/o cambiar, siempre y cuando el alcance del proyecto no supere el inicial. De esta manera el producto puede evolucionar sin problemas, en cada Sprint el dueño del producto conoce cómo evoluciona y puede aplicar cambios en las historias ya creadas.

GIT Y BITBUCKET

A la hora de mantener el código los desarrolladores de software tenemos la certeza que una buena práctica a tomar en cuenta es la de gestionar nuestro código, los SCMs nos ayudan en gran medida a cumplir este cometido.



Un SCM (Source Control Manager) también conocido como SCV (Sistemas de Control de Versiones) es un sistema que sirve para hacer este control del código. No solo nos aseguran que una copia de nuestro código quedará guardada en caso de que la estación de trabajo quede inutilizada, sino que además nos permiten volver atrás en el tiempo, pudiendo descargar una versión específica. También permite ver los cambios que han sufrido nuestros archivos. Esta técnica no es exclusiva del código y actualmente hay otros sistemas, como procesadores de texto, que incluyen un control de versiones.

El mundo de los SCMs es tan grande que no pararíamos de enumerar la gran cantidad de ellos. De entre todos, uno de los que está sobresaliendo, gracias a plataformas como GitHub o Bitbucket, es GIT.

GIT es un sistema de gestión de código distribuido. Esto facilita el trabajo entre varias personas, ya que, aunque uno de los integrantes del equipo dé de alta código, siempre lo hará en su propia máquina. Esto permite que todo el mundo pueda seguir trabajando sin tener que realizar, durante la jornada de trabajo, pesadas resoluciones de conflictos. Finalmente se tendrá que hacer aún una subida de código al servidor, para que los cambios sean totalmente aceptados. Este aspecto de los SCMs distribuidos ayuda también al trabajo cuando no se dispone de una red para entrar código al gestor de versiones, ya que se trabaja en local, y finalmente se sube todo el código una vez se vuelve a tener red disponible.

Los SCMs en general ayudan a comprender cuál ha sido la evolución de un código, y permiten volver atrás en el tiempo, para ver cómo se hicieron algunas cosas en versiones anteriores.

PRODUCT BACKLOG

Como ya he dicho antes el Product Backlog es el componente más importante de Scrum, ya que sobre el se plasma una primera versión de lo que se quiere conseguir. El Scrum Team en conjunto trabaja durante una sesión que suele ocupar una jornada de trabajo, en definir, priorizar y dar un peso, a las historias de usuario que compondrán el Product Backlog.

En esta primera reunión, el equipo establece una definición de hecho (Definition of Done) que dirá cuando una historia de usuario está acabada.

Como el Product Backlog puede ir creciendo, iMuseums comenzó con la siguiente batería de historias de usuario:

Definición estructura ficheros, JSON y base de datos interna (word)

Mapa de navegación aplicación (ppt)

Pantalla Acceder guía del museo (app compartida)

Selección de sala (versión sencilla)

Vista sencilla de sala

Vista de obra

1. Pantalla / foto / Descripción

2. Ficha con HTML

3. Video (p.e. youtube / almacenado)

2. Audio (text-to-speech)

Escoger sala de museo (versión mapa)

Pantalla preferencias de usuario

Sliding para el cambio de espacios

Activar selección de recorrido

Controles de control de reproducción de vídeo

No merecía la pena pensar en más cosas, ya que se valoró que estas historias serían suficientes para ocupar unos cuantos Sprints, este Product Backlog estaba priorizado pensando en las dependencias que había entre historias, y teniendo en cuenta cuantas historias podrían caber dentro de un Sprint.

DISEÑO TÉCNICO

Para comenzar con el diseño técnico se creó un modelo de datos basado en las clases necesarias para trabajar en la aplicación.

Por las características de Scrum, el modelo de datos no estuvo completo hasta bien entrado el proyecto, y aun yendo al final del proyecto sufrió algún que otro cambio ligero.

Finalmente el modelo de clases resultantes quedó de la siguiente manera

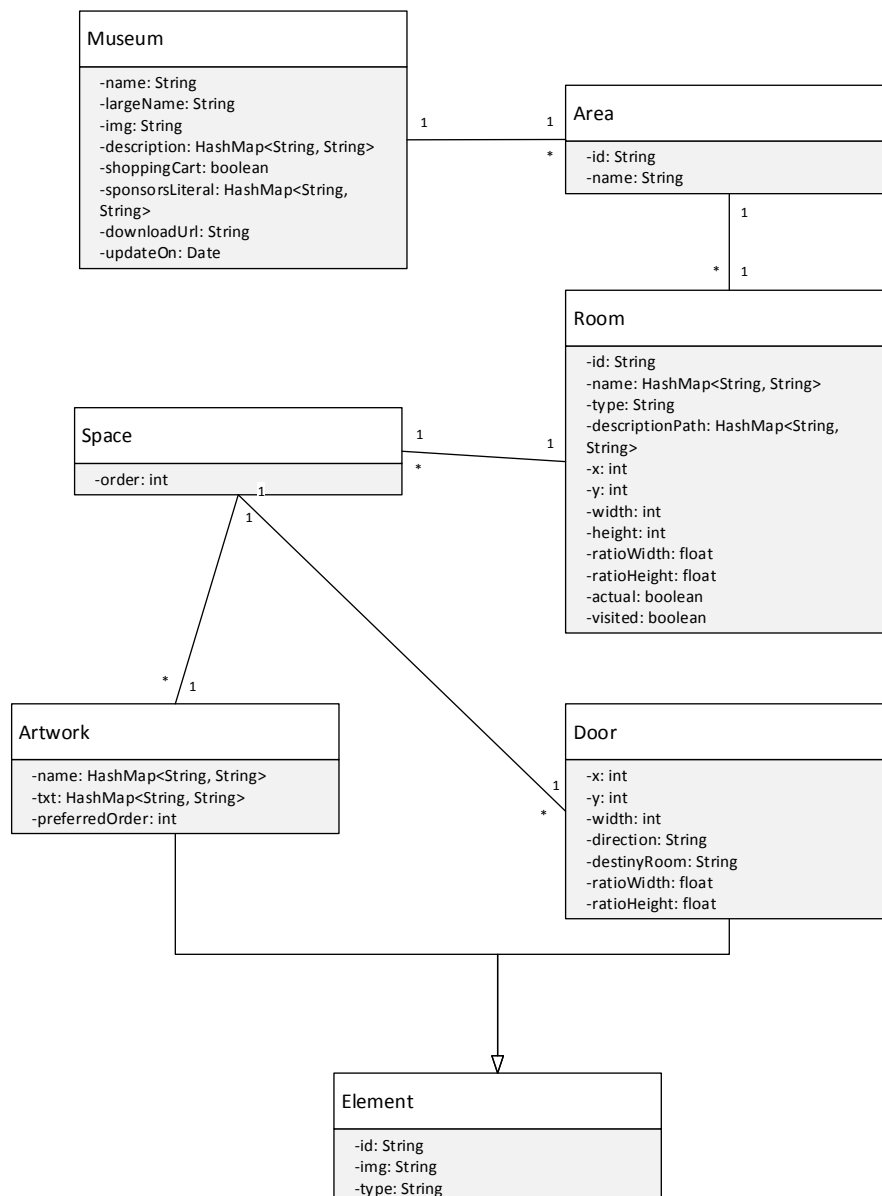


Ilustración 5 - Modelo de datos del museo

ENTORNO DE TRABAJO: ANDROID STUDIO

Al desarrollar aplicaciones que requieren cierto trabajo, hay algunas tareas del día a día del desarrollo que suelen alargar el proceso, cosas como cambiar el nombre de una variable, buscar la referencia de un nombre, ir a la definición de una función, extraer una función de un conjunto de líneas de código...

El mundo del desarrollo del software siempre ha buscado facilitar estas tareas a través de procesos automáticos, y para esto mismo se crearon los Entornos de Desarrollo Integrado (IDEs). Gracias a los IDE todas esas acciones costosas en tiempo y superficiales se pueden automatizar reduciendo el tiempo de cambiar alguna parte del código, encontrar un error en este o ejecutar la aplicación, sin necesidad de líneas de comandos.

En el mundo Android se utilizan dos IDEs, por una parte, Eclipse, como IDE para desarrollo de aplicaciones, proporciona la base sobre la que los ingenieros de Android desarrollaron las herramientas de desarrollo de Android (ADT). El ADT es un plug-in que se puede instalar en cualquier Eclipse, con el que se pueden desarrollar aplicaciones de cero sin tener que preocuparse nada más que del desarrollo. El segundo IDE en cuestión y el que se utilizó para iMuseums es Android Studio. Este IDE está desarrollado en base a IntelliJ Idea de JetBrains, un IDE alternativo a Eclipse para el desarrollo de aplicaciones. El método que emplea no es exactamente el mismo que Eclipse. Por su parte, Android Studio es un paquete atómico en sí, desarrollado por los ingenieros de Android que aún se encuentra en versión Early Access Preview. Al estar aún en desarrollo, cada versión de Android Studio puede modificar la estructura del proyecto haciendo que nuestro proyecto pueda requerir cambios manuales, uno de los problemas con los que se tuvo que lidiar en algún punto del desarrollo de iMuseums.

PATRÓN MODELO VISTA CONTROLADOR (MVC)

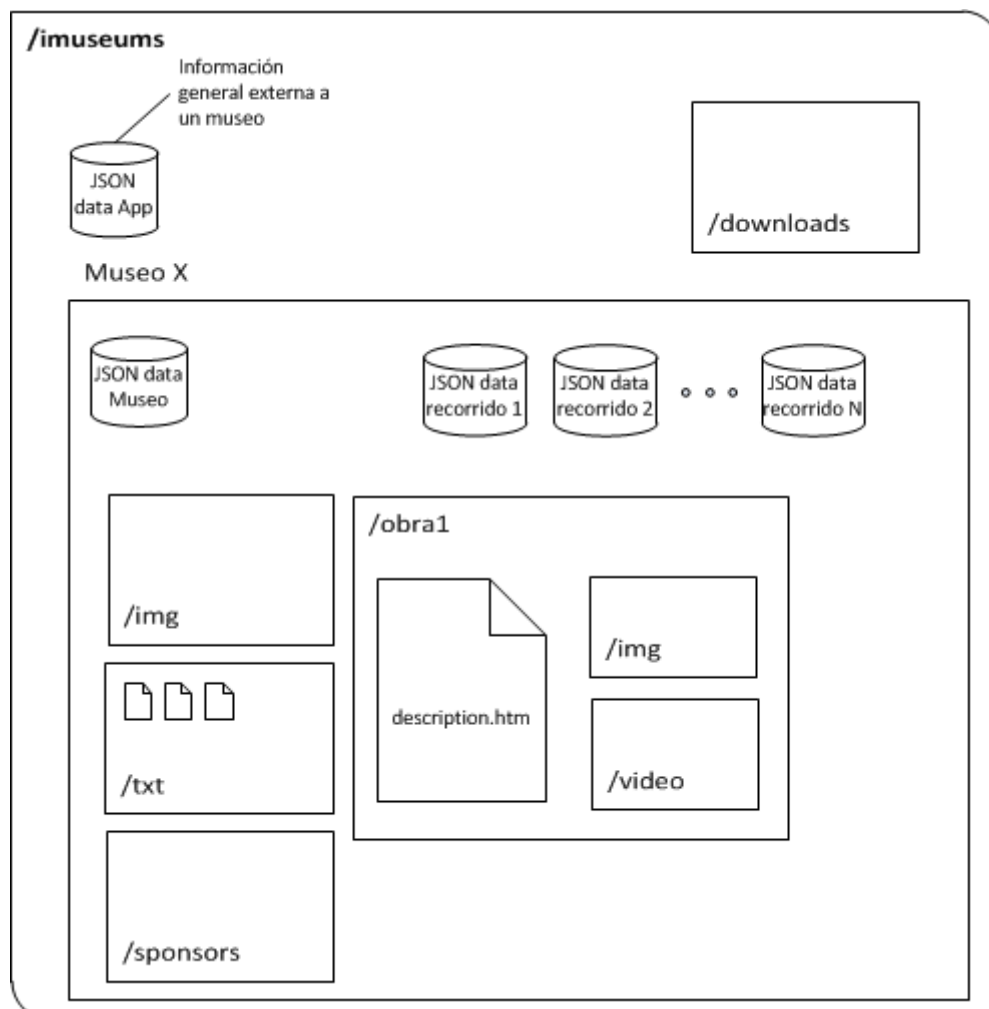
El patrón MVC pretende separar el desarrollo del proyecto en tres componentes básicos (Modelo, Vista y Controlador) desacoplando el código y facilitando su lectura.

Aunque el patrón MVC no se aplica con este nombre en Android, bien se pueden distinguir las diferentes partes de este en el desarrollo de aplicaciones. Por una parte tenemos la Activity (Controlador) que contiene toda la lógica de negocio que sirve de comunicación entre el modelo y la vista. Las vistas se crean dentro de ficheros XML que definen cómo se verá el modelo que contiene la aplicación.

ARQUITECTURA DE FICHEROS DE LA APP

En este apartado definimos la estructura de ficheros que existirá en la tarjeta SD del teléfono con el que trabajará **iMuseums** para gestionar los contenidos y descargables de los museos.

El siguiente esquema presenta la estructura de ficheros.



La jerarquía consta de un primer directorio *imuseums* que incluye lo siguiente:

Fichero *stub* JSON con datos de la aplicación, que muestra los datos que usará la aplicación al iniciarse, y que definiremos en la siguiente sección de este documento.

Directorio *downloads* que contendrá las últimas descargas comprimidas de los ficheros, para mantener los ficheros una vez se descarguen desde el servidor.

Un directorio para cada museo descargado en el dispositivo que contiene a su vez.

Un fichero *stub* JSON con los datos acerca del museo.

Varios ficheros *stub* JSON con los datos acerca de cada recorrido.

Un directorio *img* donde se almacenan las imágenes relativas al museo, como su logo, o las diferentes imágenes que se puedan utilizar en la descripción de las varias salas.

Un directorio *txt* donde se almacenan los textos descriptivos de cada museo (pej. Descripciones de los recorridos). Un directorio *sponsors* donde se almacenan las imágenes de patrocinadores del museo que se verán en el carrusel de fotos cuando se seleccione el museo en la pantalla principal.

Un directorio para cada obra en el museo que contiene los diferentes elementos.

Un fichero *.htm* que tendrá la descripción de la obra.

Un directorio *img* para almacenar las imágenes de la obra o los ficheros que puedan ser referenciados por la descripción de esta.

Un directorio *video* donde se almacenan los videos relativos al museo, que se podrán reproducir en las pantallas de cada obra.

DEFINICIÓN DE DATOS

DEFINICIÓN DE FICHEROS JSON

Se define aquí la estructura y el contenido de los ficheros JSON de los que se alimenta la aplicación. Inicialmente la aplicación se alimenta de estos ficheros de manera local, pero está pensado para que un servicio web ofrezca esta capacidad en un futuro.

FICHERO JSON APLICACIÓN

El primer fichero que definimos es el fichero que nos sirve para rellenar los *dropdown* de selección de museo en la pantalla principal.

Este JSON contiene una *array* de objetos JSON representando los museos que tenemos disponibles en la aplicación. Cada objeto contiene un nombre que lo identifica, un nombre largo, un path relativo donde se encuentra el logo del museo, una descripción corta, como un objeto con tres atributos que se refieren a los idiomas disponibles, el atributo *shoppingcart* que puede tener los valores *yes* o *no*, que nos indica si este museo tiene carrito de la compra, y un parámetro para el literal que se muestra sobre los patrocinadores para los tres idiomas en los que está disponible la aplicación.

Además, este objeto de museo incluye una ruta donde se encuentran los ficheros a descargar, la ruta donde se van a guardar, y la última fecha de actualización de los ficheros que se van a descargar.

El fichero de ejemplo que podemos encontrarnos es el siguiente.

FICHERO JSON MUSEO

Para cada museo, y dentro del directorio de cada uno, se define un fichero JSON que incluye información básica para tratar el primer paso antes de entrar a la visita guiada del museo.

En este fichero se define una lista de recorridos, con un id y su nombre en los diferentes idiomas disponibles y una lista de objetos con información de los patrocinadores del museo, con la ruta de la imagen, el nombre del patrocinador y la dirección que se abrirá al interactuar con la imagen; además de una lista de vídeos con información sobre el museo, con el título y descripción en formato multiidioma, la ruta de este, y si se reproduce en streaming o no.

Además se incluye una lista de áreas para el museo que contiene, para cada área, una lista de salas. Cada sala tiene espacios ordenados que contienen una o varias obras de arte además de las puertas que la conforman.

Cada sala se puede separar en varios tipos diferentes, en el caso de ser una sala distinta al tipo *artwork* esta sala no contiene espacios con obras de arte, solamente con puertas.

Cada elemento de un espacio (obra de arte o puerta) contiene un identificador, un tipo (si es *artwork* o *door*) y un atributo con el nombre del fichero de imagen que se visualizará para mostrarlo en el espacio.

Los elementos de tipo obra de arte identifican además un nombre para mostrar, un nombre de fichero *htm* donde se encuentra la descripción de la obra, un atributo *preferred_order* que indica cual es el orden de preferencia en el que se tiene que visitar esa obra (teniendo en cuenta que este valor solo será válido para este recorrido) y un array de objetos JSON que definen videos relacionados con la obra, con un título, descripción path relativo desde el directorio de la obra, y un booleano *internet_streaming* que indica si el video proviene de una fuente en internet o no.

Los elementos de tipo puerta contienen además de los atributos antes citados, la posición bidimensional del punto de comienzo de la puerta, lo larga que es esta y la dirección en la que se dibuja, además de la habitación de destino. Puede ser que una puerta de al exterior, con lo que el valor *outside* será el válido para este tipo de puertas.

FICHERO JSON PASOS DEL RECORRIDO

Además del fichero de recorrido nos encontramos con un fichero acabado en *_route* que contiene la ruta que sigue este recorrido. Es una *array* de objetos JSON que definen una ruta. En este caso el orden de los elementos de la *array* determina el orden en el que se visitan las salas y una *array* ordenada con los ids de obra en el orden de preferencia de visita de esas obras.

INTERNACIONALIZACIÓN

La internacionalización es una parte muy importante de las apps del estilo de la que nos ocupa. Todos los proyectos tienen como requisito no funcional la internacionalización, el poder llevar la aplicación a otros idiomas.

Esto suele conseguirse con una opción en el menú de las aplicaciones que permite elegir el idioma.

A pesar que Android y muchos otros desarrolladores de software desaconsejan esta práctica en favor de dejar al sistema operativo que seleccione el idioma en el que se va a mostrar, es decir, el desarrollador sigue definiendo textos en diferentes idiomas, pero se deja esta gestión al sistema operativo; iMuseums es un caso especial ya que puede darse el caso que un usuario con su sistema operativo en inglés, por diversas causas prefiera obtener la información en castellano o en otro idioma.

Los idiomas definidos en iMuseums son el castellano, el catalán y el inglés.

REST / JSON

La comunicación entre el servidor y la aplicación tiene que ser entendible por ambos y fluida. Actualmente en el panorama de los servicios web, REST es, hoy en día el protocolo que se usa en la comunicación entre aplicaciones cliente-servidor.

Junto con REST, el lenguaje de marcado JSON va siempre muy relacionado, y por norma general los objetos viajan a través de la red traducidos en cadenas de caracteres en formato JSON.

De la misma manera iMuseums utiliza esta capacidad y todas las comunicaciones entre el cliente y el servidor se realizan a través de REST, siendo JSON el lenguaje de marcado usado para la transferencia de objetos.

LIBRERÍAS DE APOYO

Dada la complejidad de algunos procesos dentro de la aplicación, ha sido necesario el uso de librerías ya creadas por la comunidad para realizar trabajos que pueden ocupar demasiado tiempo, y desviar el proyecto de su objetivo.

Las librerías de apoyo que se han usado para el desarrollo de iMuseums son:

- android-support-v4, que suele incluirse por defecto en los proyectos de Android para dar soporte a algunas funcionalidades que se encuentran antes de las versiones 3.0 de Android.
- iSpeech-SDK-1.4.2 como librería Text-To-Speech, para hacer que el Smartphone leyera los textos escritos en el modelo.
- json-simple-1.1.1 como intérprete de objetos JSON.

DESARROLLO

SPRINT 1 – VERSIÓN BÁSICA

SPRINT PLANNING MEETING

En el primer Sprint Planning Meeting intentamos dar un primer vistazo para crear la base de lo que sería la futura app.

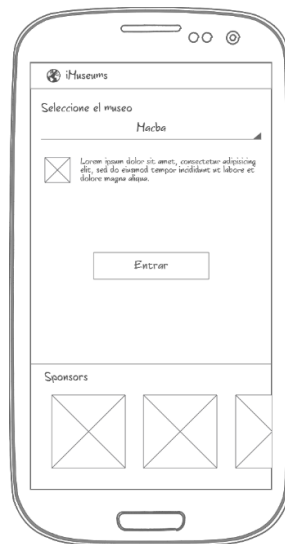


Ilustración 6 - Pantalla selección de museo

El Sprint Backlog consistía en, básicamente, casi todas las pantallas navegables que debía tener la aplicación, de esta manera aparecieron las siguientes pantallas:

- Selección de museo
- Selección de área y habitación, en modo sencillo
- Vista sencilla de sala
- Vista de obra
- Vista de descripción de obra
- Lista de vídeos de obra

Además de esto se le dedicó tiempo a una documentación inicial en vistas a la posibilidad de añadir un backend que diera soporte a la app, y poder definir el contenido de lo que se iba a tratar desde el principio.

Inicialmente el Sprint fue establecido en unas 60 horas.



Ilustración 7 - Pantallas de selección y detalle de sala y de obra

SPRINT REVIEW

En la Sprint Review pudimos ver el resultado final de este Sprint. Los puntos más reseñables sobre este primer Sprint tenían bastante que ver con las pantallas que finalmente acabarían evolucionando, la pantalla de Selección de Sala y la de Detalle de Sala.

Por una parte la vista de Selección de sala acabaría conteniendo un mapa, pero para hacer una primera versión solamente veríamos dos listas desplegables, donde seleccionaríamos el área y la sala. La pantalla de Detalle de Sala se creó con dos flechas a derecha e izquierda de la pantalla para poder desplazarse por los diferentes espacios de la sala.

Por último se observaron algunos pequeños fallos en la maquetación de las pantallas.

SPRINT 2 – AÑADIENDO LA LECTURA DE TEXTO

SPRINT PLANNING MEETING

Para el segundo Sprint, marcado en 40 horas, se quiso incluir un TTS (Text-To-Speech) para las descripciones de obras, como cambio mayor, y se priorizaron los refinamientos de las pantallas, para acabar de tener un esqueleto sólido que no sufriera demasiados cambios más adelante.

SPRINT REVIEW

Para implementar la funcionalidad del TTS se utilizaron en un principio los servicios de TTS de Google. Los otros servicios de TTS que fueron encontrados en internet ofrecían un servicio parcial, como por ejemplo, solo un idioma, o solo un número de palabras, o directamente eran de pago. Como el TTS que ofrecía Google era completamente gratuito aunque de peor calidad para el idioma castellano y peor aún para el catalán, se decidió utilizarlo pendiente de un estudio más en profundidad para encontrar un mejor proveedor.

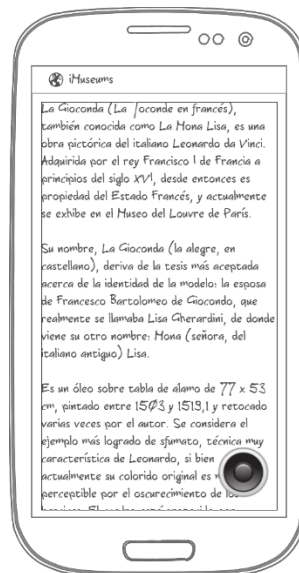


Ilustración 8 - Descripción de obra con TTS

SPRINT 3 – REPRODUCCIÓN DE VÍDEOS

SPRINT PLANNING MEETING

Aparte de reparar y mejorar algunos aspectos de la aplicación, como seguían siendo la maqueta, o parte del funcionamiento de esta, en este Sprint se planificó un cambio menor, pero significativo.

Se detectó que muchos usuarios solo usarían el Smartphone en posición vertical, además la rotación de la pantalla imprevista podría confundir a algunos usuarios. Con estas razones se decidió a fijar la orientación de todas las pantallas a vertical, a excepción de la pantalla del detalle de la obra, que podía ser apaisado, ya que había imágenes de obras que se veían mejor en forma apaisada que en vertical.

También se planificó un cambio en la estructura de ficheros, debido a una posible colisión de nombres entre ficheros.

Además, la pantalla de visualización de vídeo no contenía los controles para la reproducción del vídeo y no estaba reescalado al ratio de aspecto original del vídeo.

SPRINT REVIEW

Los ficheros se reestructuraron por obra, de esta forma, de forma que si existieran varios ficheros con el mismo nombre dentro de diferentes obras estos quedarían definitivamente separados.

El control de reproducción de vídeos cambió. Ya se había implementado con un control simple de reproducción de ficheros multimedia de la librería de Android. En este Sprint pasó a usarse otro control que también forma parte de la librería de Android. Este control incluye todo lo necesario para visualizar correctamente vídeos, e incluir los controles de reproducción de estos.

SPRINT 4 – EL MAPA

SPRINT PLANNING MEETING

En este Sprint llegaron varios cambios importantes.

Primero, se incluyó la selección de la sala del museo en versión mapa. Esta era la intención principal de la aplicación, pero como ya se ha comentado antes, para simplificar el coste de una primera versión se simplificaron algunas de las pantallas, incluida esta.

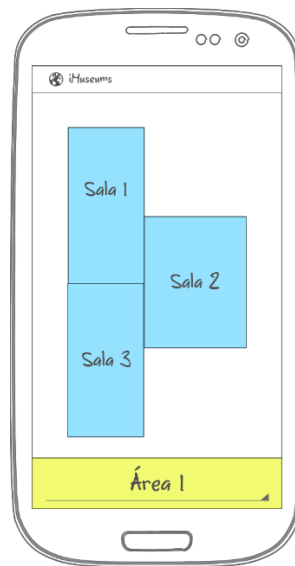


Ilustración 9 - Mapa de selección de sala

Se decidió eliminar las flechas de cambio de espacio en el detalle de la sala, por el movimiento de deslizamiento del dedo sobre la pantalla, a izquierda y derecha.

Para avisar a un usuario que la pantalla de obra era rotatable, se necesitaba un icono que indicase este hecho.

También se añadiría el botón de opciones desplegable en la barra de acción (action bar), que de momento debía permanecer inactivo.

Finalmente se debía crear la pantalla de preferencias de usuario.

SPRINT REVIEW

Los cambios significativos de este Sprint fueron el añadir el mapa a la aplicación, se cambió ligeramente la definición del modelo de datos para adaptarse a unos cambios que sucedieron durante el Sprint, y se usó el Canvas de Android para dibujar los objetos que definían las habitaciones, totalmente seleccionables. Al pulsar sobre ellos se podía ver la información de la sala, y permitía la posibilidad de entrar en ella o cerrar esa información, para seguir navegando por el mapa. Este mapa estaba completamente escalado a un ratio determinada, para que toda el área cupiera en la pantalla, y simplificar el desarrollo en el Sprint.

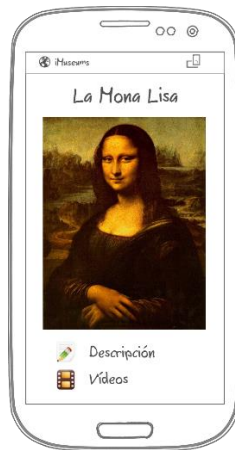


Ilustración 10 - Pantalla de detalle de obra con icono de "rotable"

SPRINT 5 – REFINADO DEL MAPA

SPRINT PLANNING MEETING

El objetivo principal de este Sprint fue acabar de refinar la pantalla de vista de mapa de las áreas. A partir de aquí también se aprovechó para definir un ejemplo de museo completo para generar un mapa completo.

El nuevo JSON también debía incluir el tamaño del mapa, de entre tres disponibles (mini, mediano y grande) para clasificarlos por ratios de tamaño, y que el cálculo del escalado a la hora de pintar no fuera tan grande.

SPRINT REVIEW

Como resultado del Sprint, las salas cambiaron de color y la maquetación de las pantallas mejoró.

Desde el servidor, se podían definir diferentes tamaños de museo. Todo esto para que, al definir los tamaños de la sala, el usuario del backend no tuviera que preocuparse por el tamaño de esta.

SPRINT 6 – DISEÑAR RECORRIDOS PARA LOS MAPAS

SPRINT PLANNING MEETING

Como principal objetivo en este Sprint se quiso añadir los recorridos a los mapas. Aunque no se buscaba dibujar el recorrido en el mapa, sí se quiso comenzar a dar forma a la definición de los datos.

Además, nos dimos cuenta que también sería importante poder tener diferentes tipos de salas, como cafetería, jardín, tienda, que pudieran mostrarse en el mapa y que el usuario pudiera tener una visión mejor de donde se encontraba.

SPRINT REVIEW

La funcionalidad para ver diferentes tipos de sala no se pudo desarrollar, aunque la definición de los recorridos estaba cerrada, además de en el modelo de datos que también se nutría de esta información.

SPRINT 7 – PINTAR LOS RECORRIDOS DEL MUSEO

SPRINT PLANNING MEETING

Este Sprint ya demandaba la inclusión de los recorridos en el mapa, una línea que fuera de sala en sala para mostrarle al usuario el camino a seguir.

Además se pensaba incluir más información en el mapa como la situación de las puertas de la sala, para orientar mejor al usuario.

También se incluyó el desarrollo de nuevos tipos de sala.

Finalmente, para trabajar mejor con los ficheros de las obras de arte, sus ficheros debían estar incluidos en una carpeta con un nombre identificativo de cada obra, para lo cual se eligió implementar un nombre con un número de 4 cifras seguido de un '_' y finalizado por un nombre de obra corto.



Ilustración 11 - Algunos tipos de sala

SPRINT REVIEW

Al finalizar el Sprint se había comenzado a definir las localizaciones de las puertas, pero aún no se podían pintar.

Los recorridos se trataban de una línea azul que iba de centro de sala a centro de sala siguiente. Como muestra para ver un recorrido estaba bastante bien, pero en una situación real podría suponer que el usuario atravesase paredes. Se sabía que este no era el último Sprint para los recorridos, y la solución inicial pareció apropiada.

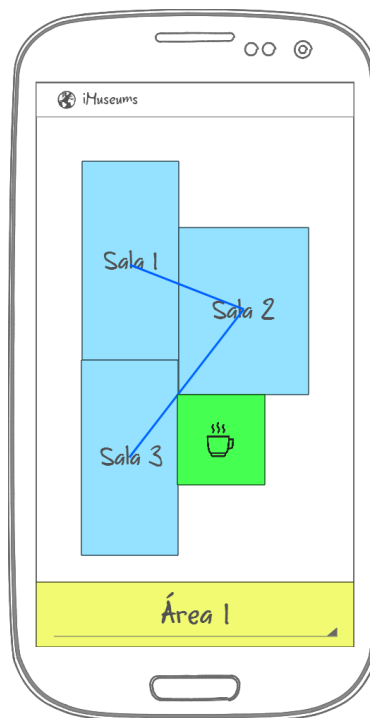


Ilustración 12 - Vista de mapa con los recorridos y un nuevo tipo de sala

Finalmente los diferentes tipos de sala aparecían en el mapa. Se optó por mostrar un icono sobre la sala de diferente tipo, cuando las salas con obras tenían un nombre pintado sobre ellas y el color cambiaba del resto de salas.

SPRINT 8 – AÑADIR LAS PUERTAS A LOS MAPAS Y MEJORAR EL RECORRIDO HACIA ELLAS

SPRINT PLANNING MEETING

Este Sprint era el momento para incluir el pintado de las puertas en las salas. Llegados a este punto, se encontró el problema de tener puertas compartidas entre salas, y que se tendría que solucionar de alguna manera, así que este Sprint también incluía dicha solución.

Además de incluir definitivamente las puertas, esto daba la posibilidad de pintar los recorridos desde el centro de la sala, a la puerta que lleva a la sala siguiente y de vuelta al centro de la sala siguiente, creando una visión más realista de los recorridos y evitando el problema inicial de atravesar paredes que se comentaba en el Sprint anterior.

Se añadieron flechas en las líneas de recorrido para indicar la direccionalidad de estas.

Puesto que los museos no muestran sus obras en cuadrículas, como se mostraban en las pantallas de salas de la aplicación, se planificó cambiar la orientación de la vista de sala a horizontal y poner en una sola fila todas las obras que hay en cada espacio de la sala.

Finalmente se eliminaría la orientación manual de la vista de obra, suprimiendo el icono que indicaba al usuario que la pantalla se podía girar. Se encontró que era bastante más sencillo orientar automáticamente la pantalla dependiendo de la orientación de la obra.

SPRINT REVIEW

Todo lo planificado en este Sprint se cumplió, a excepción de las flechas en los recorridos, después de varias implementaciones, el pintar las flechas era demasiado costoso para hacerse en ese Sprint, y se decidió seguir avanzando con otras cosas más importantes.

El usuario podía seguir el recorrido que quería hacer a través de las salas atravesando sus puertas, que podía visualizar desde el mapa como una línea más gruesa en la pared de las salas.

Las obras se veían una detrás de otra y la pantalla se colocaba en apaisado en la vista de sala, y dependiendo de si la obra tenía más ancho que alto o al revés, la pantalla rotaba en una u otra dirección.

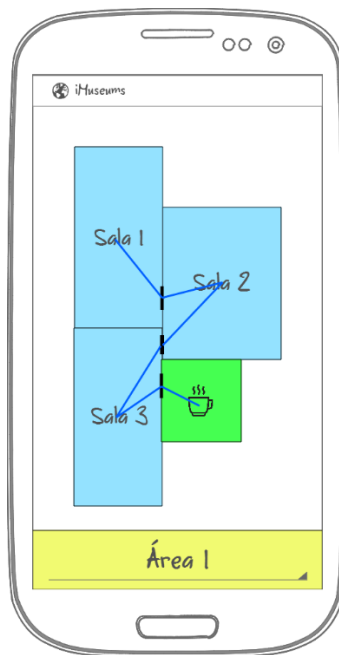


Ilustración 13 - Selección de sala con los recorridos pintados hacia las puertas

SPRINT 9 – VISIBILIDAD DEL PROGRESO DE LA VISITA Y CARRITO DE LA COMPRA**SPRINT PLANNING MEETING**

En vez de las flechas direccionales en el recorrido, se quiso ayudar al usuario de otra manera. Para cada sala se incluirían dos iconos sobre ellas, un icono que le señalaría al usuario que se encuentra en esa misma sala, y otro que le indicaría que ya ha pasado por dicha sala.

Además se definiría el orden de preferencia de visionado de las obras en la sala, a tenor que cada recorrido podría querer mostrar unas obras, en un orden específico, mientras que algunas de las obras de las salas podrían no pertenecer a ese recorrido.

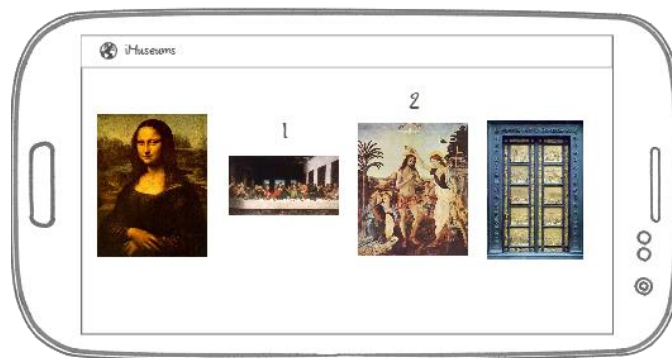


Ilustración 14 - Orden recomendado de obras

También se activaría aquí la selección del recorrido del museo.

En este Sprint se incluiría la funcionalidad del “carrito”, el usuario podría añadir a su carrito las obras del museo que quisiera, y luego podría verlas en una pantalla diferente.

SPRINT REVIEW

En la pantalla de selección de museos apareció un nuevo elemento para seleccionar el recorrido a seguir.

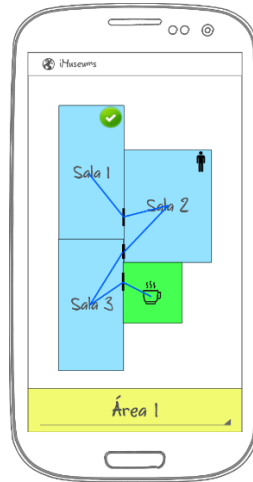




Ilustración 15 - Mapas con guía de "ya visitado"

En el mapa se podía ver un icono con forma de persona (), para mostrar la sala actualmente visitada; y un icono en forma de tick (), para mostrar las salas ya visitadas; conforme el usuario iba accediendo a la información de las salas.

Sobre las obras de arte en las salas se podía ver un número en color negro que indicaba que obras pertenecían al recorrido y el orden en el que debían ser vistas.

Durante toda la aplicación se podía ver un icono de un carrito en la action bar donde el usuario podía acceder a la vista del "carrito", y ver una lista de obras que él hubiera añadido a dicho "carrito".

En la vista de obra, además, también se podía ver otro icono en la action bar que añadía la obra en la que el usuario se encontraba al "carrito".

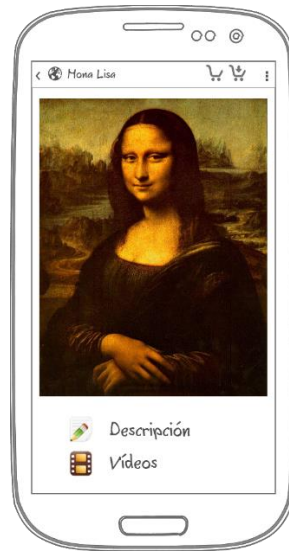


Ilustración 16 - Pantalla de detalle de obra con el carrito

SPRINT 10 – TOMA DE DATOS DE MUSEO REAL Y MULTIIDIOMA

SPRINT PLANNING MEETING

Para este Sprint se planificaron varias mejoras en la aplicación.

En primer lugar se debía mejorar el sistema de TTS, se comenzó por incluir un cambio en el icono de lectura del Text-To-Speech. Cuando se estuviera reproduciendo, el icono debía cambiar para indicar eso mismo, y de nuevo, cuando estuviera detenido debía volver a su estado inicial.

También se permitiría la reproducción de vídeos en 3GP.

En vistas a pruebas más reales se decidió incluir un museo real, como muestra se utilizó el museo Can Tinturé, que se encuentra en Esplugues del Llobregat.

Los iconos del carrito debían cambiar, ya que eran bastante parecidos entre ellos, y debían diferenciarse.

La opción del carrito debía mostrarse u ocultarse según la definición del museo permitiera o no ver esta opción.

El literal de los patrocinadores debía ser configurable desde la definición del museo.

También debía incluirse un cambio mayor en la aplicación, a partir de ese mismo instante se comenzaría a usar la opción de cambio de idioma.

SPRINT REVIEW

Al finalizar el Sprint, el usuario podría ver el cambio del icono para el TTS.

La información del museo Can Tinturé se fue cargando, haciendo necesario otro Sprint más adelante, ya que la cantidad de información a cargar era muy grande, aunque en este Sprint ya se podía ver el mapa del museo.

En las opciones de la aplicación se podía seleccionar el idioma con el cual visualizar la aplicación, como se indica al principio, en castellano, catalán e inglés.

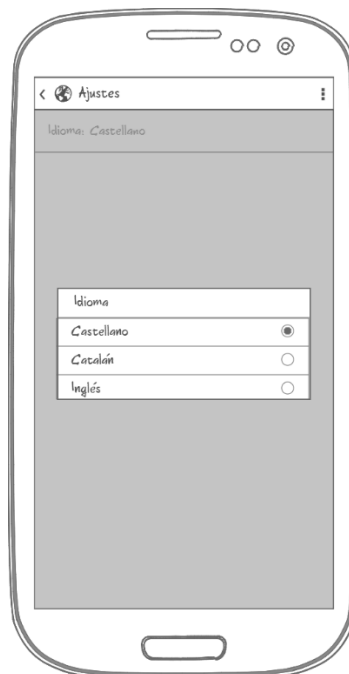


Ilustración 17 - Pantalla de ajustes y selección de idioma

SPRINT 11 – MIS OBRAS FAVORITAS

SPRINT PLANNING MEETING

Además de alguna reparación de errores del anterior Sprint, entre las historias de usuario a implementar se encontraban las siguientes.

Como antes se ha dicho, en este Sprint se seguirían cargando datos del museo de prueba.

El número con el orden de obra ahora se incluiría con un icono por debajo para ofrecerle más visibilidad y ayudar mejor al usuario.

Se decidió eliminar los tamaños en la definición de las áreas, de esta manera el ratio de tamaño quedaba hecho de forma automática por la aplicación, a veces los museos no estaban bien definidos en tamaño y quedaban espacios en blanco demasiado grandes, con lo que era difícil definir, ~~a veces,~~ el tamaño del museo.

El carrito sufría varios cambios, el número de copias añadidas se reducía a una, ya no se podrían añadir más veces la misma obra al carrito, y debía de existir la funcionalidad para eliminar la obra del carrito.

Las áreas se mostrarían con un nombre y no con un identificador, como hasta ahora, y se debían ordenar en el selector.

Se planeó añadir una clasificación para las obras. En la nueva versión aparecería un icono en el detalle de la obra que abriera una pantalla para seleccionar la puntuación que le asignaba al usuario.

Además esta lista de obras puntuadas aparecería en la selección del museo para que el usuario pudiera ver sus obras favoritas.

SPRINT REVIEW

Todos los temas hablados en el Sprint Planning se cumplieron.

Como temas más significativos, el botón que se incluyó en la vista de obra fue una estrella que abría una ventana modal con cinco estrellas que se pintaban de amarillo según la selección del usuario.

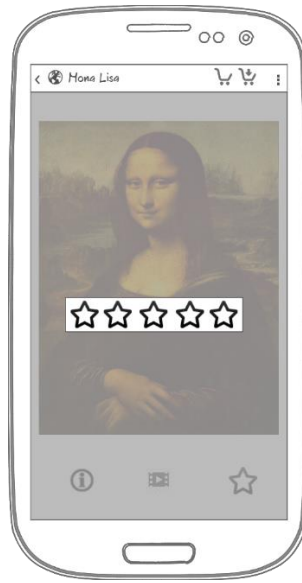


Ilustración 18 - Puntuación de obra

SPRINT 12 – DESPLAZAMIENTO DEL MAPA

SPRINT PLANNING MEETING

En este Sprint se continuaban añadiendo datos de Can Tinturé.

Además de esto y algunas mejoras en las funcionalidades ya implementadas, se planificaron cambios mayores en el mapa.

Cuando el museo era demasiado grande, la aplicación escalaba este a una medida máxima quedando este más grande que la pantalla. La idea era implementar el scroll del mapa con el dedo.

SPRINT REVIEW

El desplazamiento con el dedo se implementó, pero sin tener en cuenta los límites del mapa. El usuario podía desplazarse por todo el museo sin problemas.

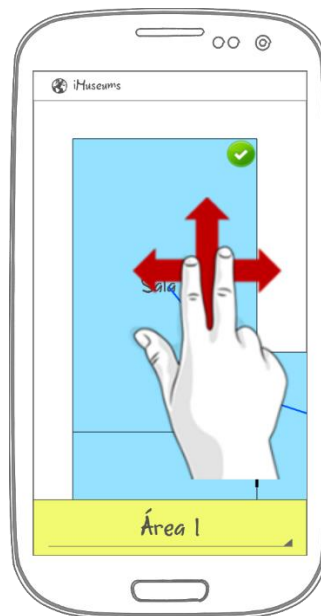


Ilustración 19 - El usuario puede navegar a través del mapa

SPRINT 13 – LÍMITES EN EL DESPLAZAMIENTO DEL MAPA

SPRINT PLANNING MEETING

Este Sprint se dedicaría principalmente a añadir los límites de desplazamiento de los mapas, teniendo en cuenta el ancho y alto máximo del área dibujada, el usuario necesitaba unos límites para evitar que se desorientase y no pudiera volver a la zona del mapa que había dibujada.

El resto de cambios que se planificaron fueron enfocados a arreglar errores encontrados, como por ejemplo el funcionamiento del icono del Text-To-Speech, que no volvía a su estado inicial cuando se acababa de reproducir la lectura.

Además también se dedicaría a mejorar la documentación del modelo de datos por razón de todos los cambios que había ido sufriendo durante los anteriores doce Sprints.

SPRINT REVIEW

El añadir los límites al mapa llevó más del doble en este Sprint, incluso obligando a llevarlo de nuevo al siguiente Sprint.

Los límites superiores quedaron sin funcionar debido a un problema de cálculo al trasladar la imagen de un punto a otro, que no permitía establecer bien el punto donde el mapa debía detener el movimiento.

Debido al sobrecoste en tiempo que supuso este cambio, hubo mejoras menores que tuvieron que dejarse de lado.

SPRINT 14 – VÍDEOS DE MUSEO Y MEJORA DEL TTS

SPRINT PLANNING MEETING

Siguiendo con el Sprint anterior, este estuvo dirigido en acabar las historias de usuario que no habían podido ser terminadas.

Acabar de arreglar los límites máximos de desplazamiento de los mapas era una prioridad, y seguir trabajando con las mejoras del anterior Sprint.

También se decidió comenzar a trabajar la mejora del Text-To-Speech. Se buscaría una nueva librería que permitiera una mejor lectura del texto en los tres idiomas que trabaja la aplicación, eliminando el uso de la librería de Google a tal efecto que se había implementado en principio.

Finalmente también se incluyó una sección de vídeos del museo, que podría ser accesible desde la pantalla principal, y que solo estaría activa si el museo tuviera vídeos que mostrar.

SPRINT REVIEW

Trabajando un poco más la idea inicial de los límites del mapa, el usuario ya podía moverse a través de un mapa que ocupase más espacio que la pantalla deslizando su dedo sobre ella, como se haría con un mapa de cualquier otra aplicación.

Para el correcto funcionamiento del Text-To-Speech, se cambió la librería de Google por una iSpeech que parecía mejorar bastante en los tres idiomas de la app. El catalán dejó de oírse robotizado y, junto con el resto de idiomas, el habla mejoró en fluidez.

Aunque parecía todo solucionado en este punto, surgió un problema con las voces en castellano, la voz de España no leía correctamente algunas palabras con tilde gramatical como es el caso de “claveles”, palabra llana que la voz leía como esdrújula. Aprovechando que existían otras voces en castellano, y probando cual podría quedar mejor, se decidió optar por la voz de castellano de Méjico por ser la que mejor leía todas las palabras con los textos que se hizo la prueba, y la que más se asemejaba al castellano de España.

El usuario también podía seleccionar la opción de vídeos del museo para ver una lista de vídeos del museo, no asociados a ninguna obra que contuviera este.

SPRINT 15 – MEJORA DEL TTS Y DIBUJADO DEL MAPA

SPRINT PLANNING MEETING

Teniendo un sistema de Text-To-Speech, muy mejorado en cuando a la forma de leer el texto, este Sprint estaba dedicado especialmente a encontrar mejoras para ello.

Para evitar tiempos de espera al usuario, a la hora de cargar el audio que se reproducía del texto se quiso hacer la descarga del fichero mp3 y guardarla en el dispositivo de manera que si el usuario quería volver a oír el texto, la aplicación tuviera un lugar en local donde obtener el audio a reproducir.

Aunque el desarrollo de los límites del mapa estaban ya correctos, el mapa se movía con lentitud debido al pintado constante de las salas. También se quiso cambiar esto para que no pintase todas las salas sino que obtuviera una imagen que se desplazase.

SPRINT REVIEW

Durante el Sprint se gastó mucho tiempo en investigar la forma de descargar un MP3 del audio que se iba a reproducir para luego evitar tantas llamadas a la red.

Según las condiciones de uso, y el nivel del servicio que se había obtenido en el registro, el servicio gratuito no permitía la descarga del audio en formato MP3, con lo que era imposible montar el sistema. Se intentó hacerlo de una manera distinta sin éxito, ya que el grabar el audio a medida que se reproducía iba a ocupar más de un Sprint, y no quedaban más Sprints sobre los que trabajar en ello, así que la funcionalidad se quedó para futuras mejoras.

SPRINT 16 – PRIMERA INTEGRACIÓN CON EL BACKEND

SPRINT PLANNING MEETING

En este Sprint, el objetivo principal fue poder establecer un estándar para el JSON, y lograr una primera integración con el Backend que permitiera obtener datos de prueba desde un servidor remoto, como era la intención del proyecto al principio.

Se propusieron entonces varios cambios en la definición del JSON, que suscitaron algunos cambios más sobre el modelo. Todos ellos se iban a abordar en este Sprint.

SPRINT REVIEW

Al finalizar este Sprint la aplicación estaba preparada para hacer una primera integración, al menos a nivel de lectura de datos.

SPRINT 17 – SEGUNDA INTEGRACIÓN CON EL BACKEND

SPRINT PLANNING MEETING

Este Sprint seguía teniendo como prioridad la integración con el Backend.

Además de esto, se aprovechó para comenzar a hacer algunas pruebas sobre el mapa, y a empezar a solucionar problemas para evitar que salieran en un futuro.

Debido al desplazamiento del mapa, la posición en pantalla de las salas no se modificaba, esto resultó en un error que se quiso solucionar en este Sprint. Al no refrescarse las posiciones, cada vez que el usuario pulsaba sobre una zona de la pantalla, la sala seleccionada no cambiaba con lo que siempre aparecía la misma información de sala, independientemente de que hubiera movido el mapa.

Como anteriormente ya se habían añadido las puertas al detalle de la sala, para mejorar la navegabilidad, las puertas que se veían en las salas, podrían ser pulsadas y éstas abrirían la sala a la que diera paso.

SPRINT REVIEW

Finalmente la versión que salía de este Sprint Review tenía integrado todo el modelo de datos con el Backend, el paso siguiente era acabar con el salto, y comenzar a hacer la descarga de los ficheros y la visualización de datos del servidor.

El usuario ahora podía seleccionar la sala correcta que seleccionaba, incluso habiendo movido el mapa.

La funcionalidad asociada a las puertas de las salas quedó colgada debido a un error que se arrastraba aunque estaba implementada por completo.

SPRINT 18 – TERCERA INTEGRACIÓN CON EL BACKEND

SPRINT PLANNING MEETING

Para este Sprint era prioritario poder leer el fichero JSON del servidor y mostrar la información en la pantalla de selección de museo.

Además también se quiso reparar el error sobre el paso entre salas a través de las puertas.

SPRINT REVIEW

Al acabar el Sprint la pantalla de selección de museo/recorrido mostraba la información extraída del JSON que se obtenía del servidor.

No se pudo arreglar el error que había en las puertas ya que hasta que no descargásemos el museo y se pudiera tener acceso a los ficheros de este, no se podría comenzar con la implementación y reparación de errores, ya que lo prioritario era poder obtener datos del backend.

SPRINT 19 – CUARTA INTEGRACIÓN CON EL BACKEND

SPRINT PLANNING MEETING

Para este Sprint, el usuario tenía que poder descargar el zip que contenía los datos del museo, se debía mostrar el botón para la descarga cuando el museo no estuviera descargado y finalmente hacer la descarga del fichero y la descompresión.

SPRINT REVIEW

El usuario a estas alturas podía entonces descargar el fichero zip de un museo, y la app lo extraía de forma que el usuario no tuviera que preocuparse de nada más.

El primer problema que apareció fue el del tamaño de las imágenes, con imágenes pequeñas la descarga y la descompresión iban bien y rápido, pero con imágenes muy grandes, la descarga se ralentizaba.

A pesar de esto, no se iba a tomar ninguna medida, ya que el servidor debería servir imágenes en tamaño reducido, ya que los dispositivos no podrían mostrar correctamente imágenes grandes.

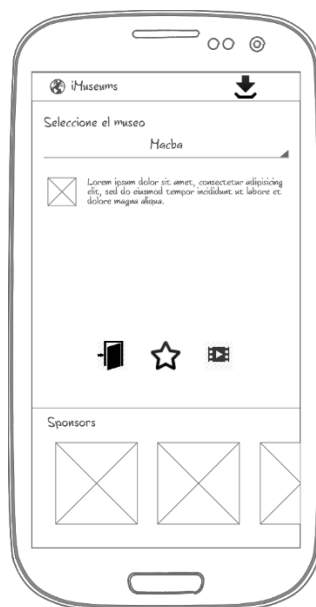


Ilustración 20 - Pantalla principal con el botón de descarga y el resto de botones antes creados

Una vez acabado el Sprint 19 la app queda completa, en cuanto a las funcionalidades que se podían hacer en el tiempo disponible. El resto de funcionalidades se podrían seguir desarrollando siguiendo este mismo método. Sencillamente se seguirían desarrollando Sprints, y haciendo crecer el Product Backlog de manera que la app consiguiera evolucionar en un proceso de mejora continua

CONCLUSIONES

PRESUPUESTO FINAL VS PRESUPUESTO INICIAL

El desvío existente del proyecto en la fecha de final se debe, principalmente, a la complejidad de algunos problemas que se han tenido que resolver, ya que había algunas historias de usuario muy costosas que no se podían añadir en Sprints en los que ya se había rellenado más de la mitad del tiempo máximo del Sprint. Esto ha obligado a fragmentar más las tareas, creando Sprints que a veces eran muy pequeños.

El número de Sprints iniciales de 13 Sprints se vio aumentado en 6, llegando hasta los 19. A pesar de este aumento del número de Sprints realizados, la cantidad de horas no se ha visto aumentada. Algunos de estos Sprints estaban alargados en tiempo debido a periodos de vacaciones.

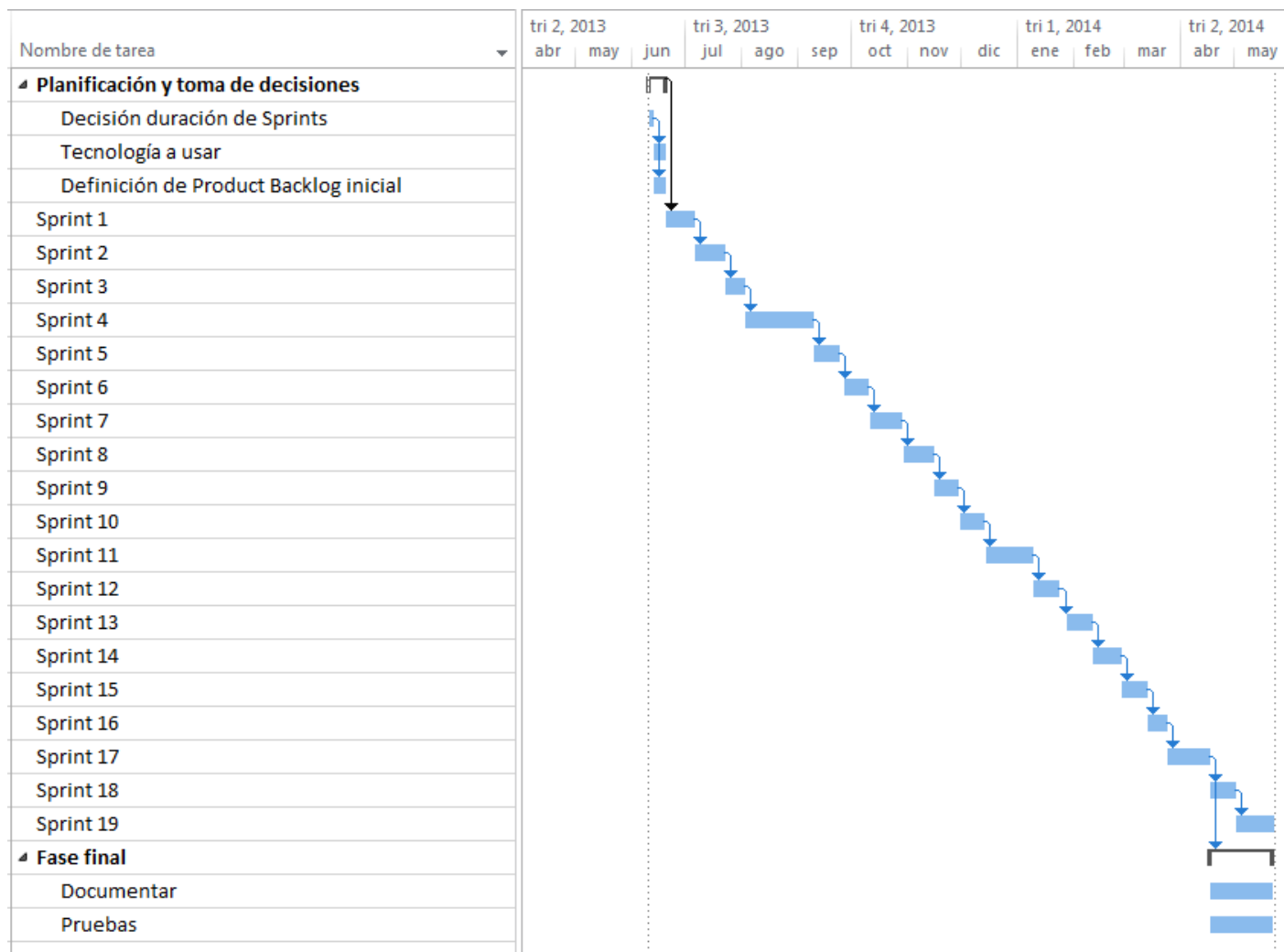
El coste del proyecto se ha mantenido intacto ya que la duración en horas del proyecto ha sido la misma, quedando de la siguiente manera.

Sprint	Duración
Planificación y toma de decisiones	30
Sprint 1	36
Sprint 2	23
Sprint 3	20,5
Sprint 4	25
Sprint 5	20
Sprint 6	12,5
Sprint 7	17
Sprint 8	29
Sprint 9	29
Sprint 10	35
Sprint 11	45
Sprint 12	50
Sprint 13	36
Sprint 14	32
Sprint 15	24
Sprint 16	28
Sprint 17	20
Sprint 18	14
Sprint 19	24
Documentar	100
Pruebas	100
TOTAL	750

El presupuesto final del proyecto quedaría de la siguiente manera:

Total de horas Equipo de Desarrollo	720
Coste laboral por hora Equipo de Desarrollo	30 €
Coste en horas Equipo de Desarrollo	21.600 €
Total de horas de Scrum Master	30
Coste laboral por hora de Scrum Master	50 €
Coste en horas de Scrum Master	1.500 €
Licencia de desarrollador Android	18,43 €
Subtotal	23.118,43 €
Margen comercial (15%)	3.467,76 €
TOTAL	26.586,19 €

En el presupuesto final también se ha tenido en cuenta el precio del Scrum Master que le ha tenido que dedicar pocas horas al proyecto ya que solamente se encargaba de facilitar las reuniones de seguimiento que componen un Sprint.



Gantt 2 - Presupuesto final del proyecto

CONCLUSIONES SOBRE EL PROYECTO

Los objetivos del proyecto han sido cubiertos al final de este. Incluso hemos llegado a añadir funcionalidad adicional como la integración con un backend, funcionalidad que se había comentado al principio, pero que no se contemplaba como funcionalidad a final del proyecto.

Los museos de hoy en día se encuentran bastante encerrados en su propio sistema, son pocos los que se aventuran a la era digital y muchos de ellos siguen usando el mismo sistema que anteriormente.

iMuseums es una herramienta que engloba varios métodos de difusión de los museos, sin obligarles a estos a deshacerse de los antiguos sistemas, ya que no son incompatibles, y no todos los usuarios están adaptados para el uso de smartphones, o simplemente prefieren un formato más físico.

El proyecto enseña al desarrollador a pulir su trabajo, ya que tiene que incluir mucha información en una pantalla pequeña, lo que hace que su trabajo sea más fino, que se estudié mucho más el contenido de las pantallas, ya que mucha información debe ser mostrada en forma de icono, no con demasiado texto.

CONCLUSIONES PERSONALES

El proyecto iMuseums me ha dado la posibilidad de trabajar con Android.

Aprovechando que están en desarrollo de su IDE Android Studio he tenido la oportunidad de desarrollar el proyecto entero con ese IDE. Ya que en la mayoría de empresas se desarrolla bajo Eclipse con Android Developer Tools, esta es una buena oportunidad de aprender a usar algo novedoso en el mundo del desarrollo de apps.

También me ha permitido ver como es el desarrollo de un IDE de estas características, he sufrido los más y los menos del IDE. Me he encontrado que el IDE va especialmente bien, ya que está desarrollado directamente para desarrollo de apps Android, pero también me he encontrado que, en cada cambio mayor, me llegaba a encontrar con problemas, ya que se había aumentado la versión del Gradle, o que habían cambiado la estructura de ficheros dentro del proyecto.

Trabajar con Scrum me ha facilitado, primero de todo, la forma de trabajar. Al acotar el trabajo a hacer durante un periodo de tiempo es posible concentrarse en solo ese grupo de tareas.

En la mayoría de las empresas de desarrollo de software el trabajo suele sacarse desde el principio con un gran conjunto de tareas a hacer, sueles tener del orden de la veintena de tareas a realizar durante dos o tres meses. Trabajar como lo he hecho con iMuseums me ha permitido no sobrecargar el análisis de la aplicación al inicio, ya que en la mayoría de proyectos en los que he trabajado, la especificación de requisitos tendía a alargarse más de lo necesario. Esto es debido a que esa especificación la haces al inicio. Con Scrum los requisitos van tomando forma conforme se avanza en los Sprints, no es necesario comenzar sabiendo todo lo que vas a hacer hasta el final del proyecto.

TRABAJO FUTURO

El trabajo del desarrollo de apps es constante, en una primera versión de una app es imposible determinar que todo está correcto y que no va a sufrir ningún cambio.

El mundo de los smartphones evoluciona a pasos agigantados, y el trabajo del desarrollador de aplicaciones móviles es constante.

En el caso específico de iMuseums, el trabajo que vendría después de esto incluiría varios puntos.

- Mejora del pintado de mapas: se podría incluir pintar salas con formas geométricas diferentes de cuadrados, lo que aumentaría la sensación de realidad para el usuario.
- Mapas en 3D: esto proporcionaría al usuario una mejor orientación, ya que podrían situarse mejor dentro de la sala.
- Geolocalización indoor: ofrecer al usuario la posibilidad de ir narrándole o de acceder a cada sala sin necesidad de pulsar sobre ella sería el objetivo final de esta mejora. Actualmente los sistemas de geolocalización de interiores hacen un uso excesivo de la batería puesto que necesitan que el Smartphone tenga, constantemente, conectados Wi-Fi y Bluetooth.
- Archivos del audio del Text-To-Speech: para mejorar la accesibilidad, y evitar tiempos de espera muy largos. La aplicación debería guardar, como copia local los archivos de audio, en formato MP3 de lo leído por iSpeech u otro sistema de TTS.

GLOSARIO

Android: Sistema operativo basado en el kernel de Linux diseñado para dispositivos móviles.

Scrum: Metodología ágil de desarrollo de proyectos basada en el desarrollo por iteraciones o Sprints.

Scrum Master: Dentro del equipo de Scrum, el Scrum Master se encarga de facilitar el seguimiento de la metodología.

Product Owner: Como su nombre indica, es el propietario del producto. En un equipo de Scrum es el encargado de asegurarse que el producto desarrollado cumpla con las funcionalidades acordadas antes de cada Sprint.

Development Team: El equipo de desarrollo es el encargado de construir el producto a través del ciclo de vida de Scrum.

Eclipse: Entorno de desarrollo, principalmente diseñado para proyectos en Java.

IDE (Integrated Development Environment): Es el entorno que dispone de diversas herramientas para facilitar el desarrollo de aplicaciones.

SCM (Source Control Management): Es el proceso dentro de un Proyecto de software que se encarga de gestionar el control del código, para facilitar el acceso a este y el tratado de versiones.

GIT: Gestor de versiones de código distribuido.

Smartphone: Dispositivo informático móvil que amplía las funcionalidades estándar de los teléfonos móviles.

BIBLIOGRAFÍA

Adrián Fariñas. *Así se reparte el mercado entre las compañías del sector smartphone en Estados Unidos*. [Consulta: 17/05/2014] <http://andro4all.com/2013/04/reparto-mercado-movil-estados-unidos>

John Koetsier. *Comparing Apples and Googles: The App Store vs. Google Play (infographic)*. [Consulta: 17/05/2014] <http://venturebeat.com/2013/07/17/comparing-apples-and-googles-the-app-store-vs-google-play-infographic/>

Emma Llensa. *Aplicaciones para museos: casos de estudio (i)*. [Consulta: 16/06/2014] <http://www.ubicuostudio.com/es/resenas-de-apps/apps-iphone-ipad-museos/>

ZOOMNEWS. *Las apps de museos más originales*. [Consulta: 16/06/2014] <http://www.zoomnews.es/236089/estilo-vida/arte/mejores-originales-apps-museos/>

Museo Thyssen-Bornemisza. *App Museo Thyssen*. [Consulta: 16/06/2014] http://www.museothyssen.org/thyssen/app_thyssen/

Louvre. *Musée du Louvre official app*. [Consulta: 16/06/2014] <http://www.louvre.fr/en/mediaapps/musee-du-louvre-official-app/>

Wikipedia. *Scrum (software development)*. [Consulta: 17/06/2014] [http://en.wikipedia.org/wiki/Scrum_\(software_development\)](http://en.wikipedia.org/wiki/Scrum_(software_development))

Android. *Android Reference*. [Consulta: 18/06/2014] <http://developer.android.com/reference/packages.html>

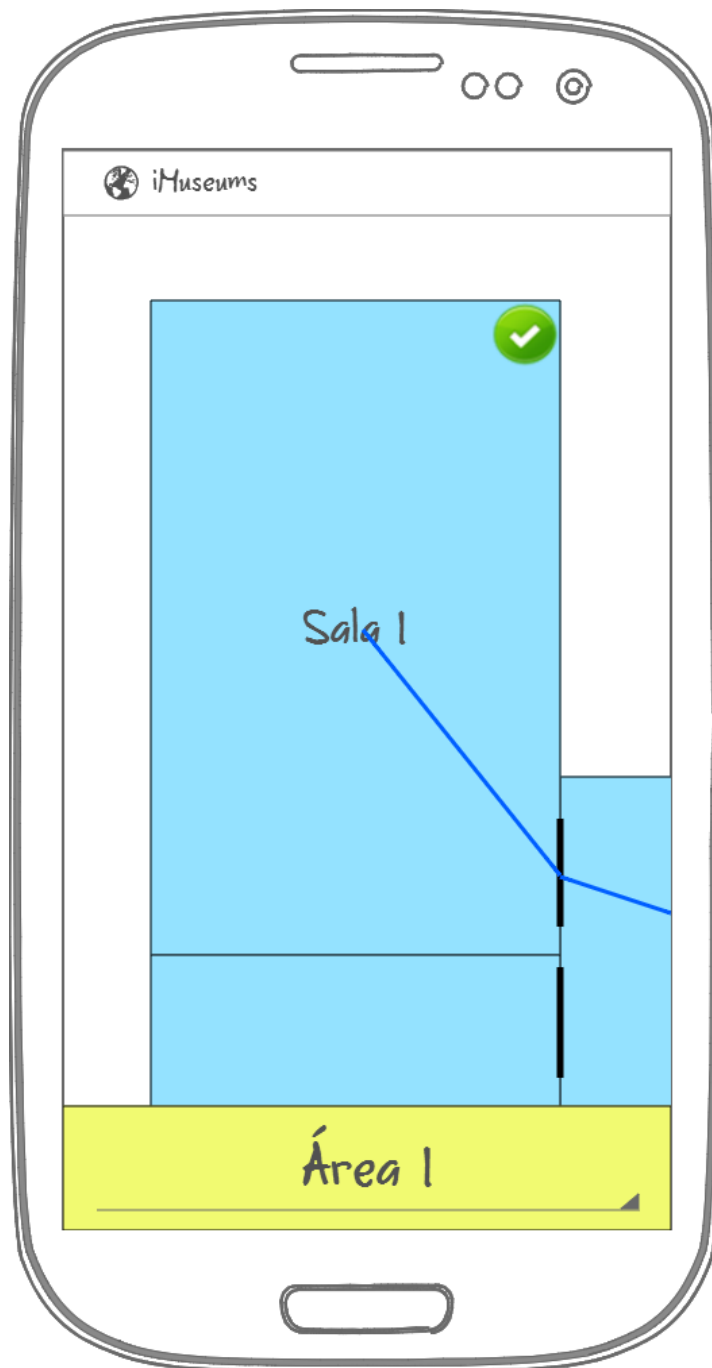
Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas. *Manifesto for Agile Software Development*. [Consulta: 19/06/2014] <http://agilemanifesto.org/>

ANEXOS

PANTALLAS DE LA APP



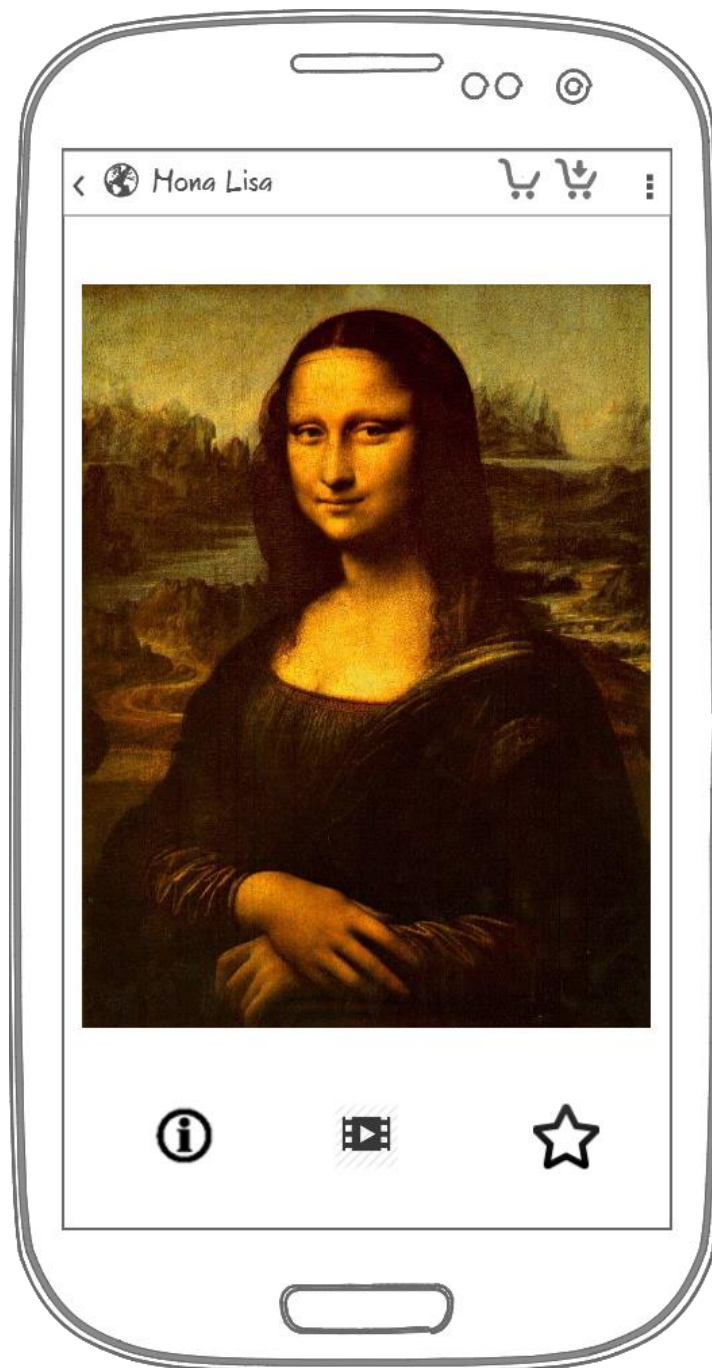
Pantalla 1 - Selector de museo y recorrido



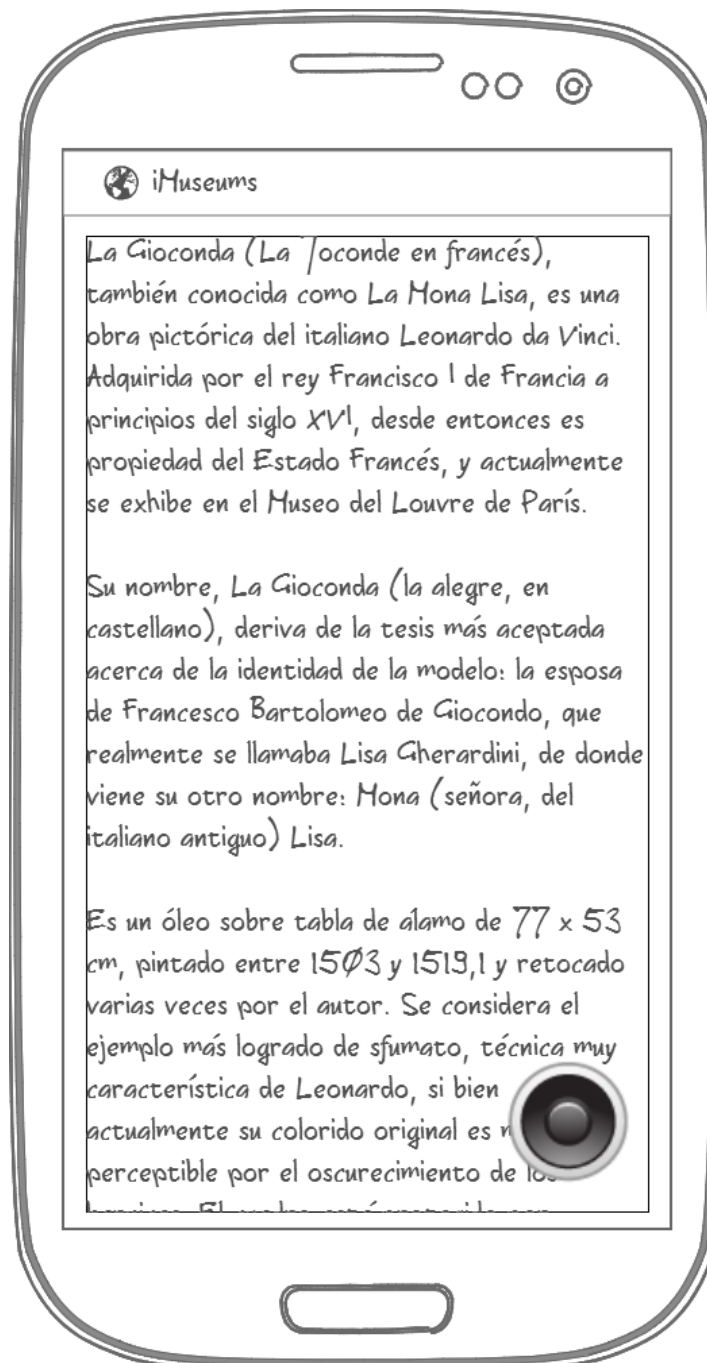
Pantalla 2 - Selección de sala en mapa



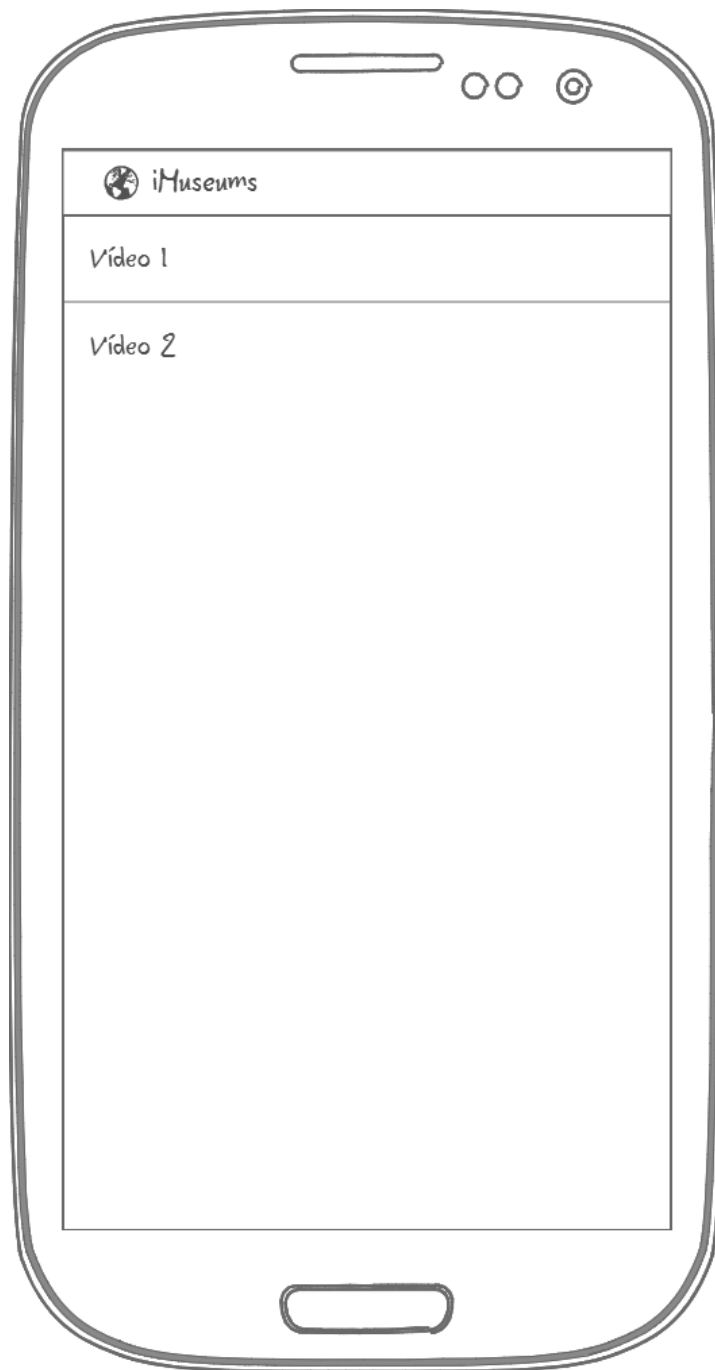
Pantalla 3 - Vista de detalle de sala



Pantalla 4 - Detalle de obra



Pantalla 5 - Descripción de obra



Pantalla 6 – Vídeos



Pantalla 7 - Ajustes