Proyecto

Questionfinder

**Historial de revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 17/03/2017 | 1.0 | Se realiza el diseño de la propuesta de negocios y el diseño de los instrumentos de estudio de mercado de la propuesta elegida | José Ferrer |
| 20/03/2017 | 1.1 | Se realiza la aplicación de los instrumentos diseñados, y se documentan los cambios que se deben de realizar en la aplicación. | José Ferrer |
| 22/03/2017 | 1.2 | Se incluyen los hiperenlaces a los documentos y modelos utilizados para el estudio de mercado entre los potenciales usuarios. | José Ferrer |
| 13/04/2017 | 1.3 | Se desarrolla la vista de la aplicación en conjunto con una vista secundaria | José Ferrer |
| 16/04/2017 | 1.4 | Se documentan y completan los informes | José Ferrer |
| 17/04/2017 | 1.5 | Se finalizan los documentos y se realiza la Retrospective de la metodología | José Ferrer |

**Contenido**

[1.-Introducción 4](#_Toc480375207)

[1.1.- Propósito de este documento 4](#_Toc480375208)

[1.2.- Alcance 4](#_Toc480375209)

[2.- Descripción de metodología de trabajo 4](#_Toc480375210)

[2.1.- Fundamentos 4](#_Toc480375211)

[2.2.- Valores de trabajo 5](#_Toc480375212)

[3.- Distribución de Miembros y de Actividades 5](#_Toc480375213)

[3.1.- Personas involucradas y roles 5](#_Toc480375214)

[3.2.- Responsabilidades de los roles 5](#_Toc480375215)

[3.3.- Product Backlog 6](#_Toc480375216)

[4.- Artefactos (Primera iteración) 7](#_Toc480375217)

[4.1.- Sprint Backlog 7](#_Toc480375218)

[4.2.- Incremento 8](#_Toc480375219)

[4.3.- Cálculo de tiempos 15](#_Toc480375220)

[4.4.- Gráfico de Burn-up 16](#_Toc480375221)

[4.5.- Gráfico de Burn-down 16](#_Toc480375222)

[5.- Artefactos (Segunda Iteración) 16](#_Toc480375223)

[5.1.- Sprint Backlog 16](#_Toc480375224)

[5.2.- Incremento 18](#_Toc480375225)

[5.3.- Cálculo de tiempos 23](#_Toc480375226)

[5.4.- Gráfico de Burn-up 24](#_Toc480375227)

[5.5.-Gráfico de Burn-down 24](#_Toc480375228)

[6.- Artefactos (Tercera Iteración) 25](#_Toc480375229)

[6.1.- Sprint Backlog 25](#_Toc480375230)

[6.2.- Incremento 25](#_Toc480375231)

[6.3.- Cálculo de tiempos 30](#_Toc480375232)

[6.4.- Gráfico de Burn-up 31](#_Toc480375233)

[6.5.- Gráfico de Burn-down 31](#_Toc480375234)

[7.- SCRUM Retrospective 32](#_Toc480375235)

[7.1.- Cálculo de tiempos 32](#_Toc480375236)

[7.2.- Gráfico de Burn-up 33](#_Toc480375237)

[7.3.- Gráfico de Burn-down 33](#_Toc480375238)

# 1.-Introducción

En este documento se desarrolla la aplicación de la metodología SCRUM en el grupo Vertex Caster Studios para la gestión del desarrollo del proyecto Questionfinder. Se incluye el alcance del proyecto, la metodología de trabajo a usar, con sus valores y fundamentos, la distribución de roles entre las personas involucradas, la lista de elementos a desarrollar para el proyecto (Product Backlog), y, por cada iteración realizada, una lista de tareas realizadas, el incremento, los gráficos de burn-up y burn-down en cada iteración.

## 1.1.- Propósito de este documento

Documentar el desarrollo del proyecto en un único archivo, en el cual se ilustre el incremento de características del proyecto Questionfinder.

## 1.2.- Alcance

Todas las personas y procedimientos relacionados con el proyecto Questionfinder.

# 2.- Descripción de metodología de trabajo

## 2.1.- Fundamentos

Las razones por las que se decidió utilizar la metodología iterativa e incremental SCRUM son las siguientes:

* La metodología utiliza un Sistema Modular que permite crear del proyecto Questionfinder una base mínima, e ir mejorando sus características a lo largo del tiempo.
* Lo anterior permite entregas frecuentes al cliente de las características en desarrollo del proyecto, poniendo en servicio características en poco tiempo para su posterior depuración.
* Debido a la naturaleza cambiante de las características del proyecto, se necesita una metodología que no se atasque en caso de que una de las características cambie o se elimine, o una nueva sea añadida.

## 2.2.- Valores de trabajo

Los valores de trabajo que se deben de practicar son los siguientes:

* Autonomía entre los miembros del equipo.
* Respeto en todo momento.
* Dedicación a las tareas
* Visibilidad y transparencia en la información

# 3.- Distribución de Miembros y de Actividades

## 3.1.- Personas involucradas y roles

Las personas involucradas y sus roles son:

|  |  |  |
| --- | --- | --- |
| **Persona** | **Contacto** | **Rol** |
| José Ferrer | [jgfr2009@gmail.com](mailto:jgfr2009@gmail.com) / 0416-0777220 | Coordinador del equipo |
| José Urbina | [urbinaj110@gmail.com](mailto:urbinaj110@gmail.com) / 0426-7291655 | Equipo Técnico / Documentador |
| Ángelo Tarascio | [tarasciomorenoa@uvm.edu.ve](mailto:tarasciomorenoa@uvm.edu.ve) / 0414-7110761 | Equipo Técnico / Diseñador |
| Miguel Colmenares | [colmenaresmiguel192@gmail.com](mailto:colmenaresmiguel192@gmail.com) / 0414-0786451 | Equipo Técnico / Desarrollador |
| Jonhmer Bencomo | [bjohnmer@gmail.com](mailto:bjohnmer@gmail.com) / 0424-7047809 | Gestor del producto |

## 3.2.- Responsabilidades de los roles

El Coordinador del equipo tiene las siguientes responsabilidades:

* Supervisar que las actividades asignadas para cada una de las iteraciones están siendo realizadas oportunamente.
* Comunicarse con el Product Owner en caso de deficiencias durante el curso de alguna de las iteraciones de desarrollo del proyecto.

El Documentador del equipo tiene las siguientes responsabilidades:

* Registrar cada uno de los cambios realizados en el proyecto.
* Documentar el uso de aquellas características que pudieran presentar algún problema para el trabajo del equipo, por desconocimiento de estas o por su mal uso.

El resto del equipo técnico tiene como responsabilidades lo siguiente:

* Desarrollar las tareas definidas en cada una de las iteraciones de desarrollo del proyecto.
* Tomar nuevas tareas en todo momento, priorizando aquellas que pertenecen a su capacidad.

El Gestor del Producto tiene las siguientes responsabilidades:

* Definir qué características se van a incluir para su planificación y desarrollo del producto en el backlog.
* Mantener registro de qué elementos del backlog se han desarrollado, que cambios hacer en los elementos existentes, o bien quitarlos o eliminarlos, durante las reuniones semanales.

## 3.3.- Product Backlog

El registro de elementos a desarrollar es el siguiente:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridad** |  | | **Nueva estimación en Sprint…** | | | |
| **Elemento** | **Estimación inicial (H)** | **1** | **2** | **3** | **SR** |
| 1 | Determinar la viabilidad económica del proyecto Questionfinder | 80 | 77 | 0 | 0 | - |
| 2 | Desarrollar los elementos de diseño y los prototipos de la aplicación | 80 | 80 | 77 | 0 | - |
| 3 | Programar y probar la prueba de concepto de la aplicación diseñada | 100 | 100 | 100 | 100 | - |
| 4 | Implementar la sección de registros e inicio y cierre de sesión de los usuarios | - | - | - | - | - |
| 5 | Implementar el mecanismo de referencias para el uso de la aplicación | - | - | - | - | - |
| 6 | Diseñar una sección para permitirles a los usuarios realizar cuestionarios personalizados | - | - | - | - | - |
| 7 | Integrar los mecanismos para poder evaluar el uso de la aplicación por parte de los usuarios | - | - | - | - | - |
| 8 | Diseñar e implementar la sección que permite registrar el progreso de los usuarios a lo largo del tiempo | - | - | - | - | - |
| 9 | Implementar todas las secciones que no se han logrado implementar en el transcurso del desarrollo | - | - | - | - | - |
| 10 | Documentar y probar todas las secciones desarrolladas de la aplicación desarrollada | - | - | - | - | - |
|  | **Horas totales de desarrollo** | **260** | **257** | **177** | **100** | **0** |

Se desglosan los elementos del backlog de la siguiente manera:

* En el elemento “Determinar la viabilidad económica del proyecto Questionfinder”, se propone determinar los elementos económicos del proyecto, es decir, como va a funcionar la aplicación, como se monetizará, cuáles serán sus soportes, asimismo se diseñarán el posible ícono para la aplicación los instrumentos para las pruebas de aplicación, , que son específicamente una encuesta y un prototipo de aplicación utilizando LibreOffice Impress, se aplicarán dichos instrumentos, y se adaptará el diseño acorde a los hallazgos obtenidos con los instrumentos.
* En el elemento “Desarrollar los elementos de diseño y los prototipos de la aplicación”, se desarrollará la interfaz de la aplicación (considerando los elementos de diseño obtenidos en las encuestas), y se presentarán dichos prototipos al Product Owner para que determine las modificaciones que deberán de realizarse antes de iniciar la codificación.
* En el elemento llamado “Programar y probar la prueba de concepto de la aplicación diseñada” se creará usando la biblioteca AngularJS en conjunto con HTML la interfaz de la aplicación, la cual está diseñada específicamente para usarse en dispositivos móviles, en este elemento, se debe de considerar que, para que la aplicación sea útil, se necesita una API, la cual es desarrollada por otro equipo de desarrollo, y que esta aplicación, por ser híbrida, deberá adaptarse a los elementos del diseño responsivo y la filosofía Mobile-First.
* En el elemento llamado “Implementar la sección de registros e inicio y cierre de sesión de los usuarios” se crearán las vistas y la lógica con la cual se le permitirá a los usuarios registrarse en la aplicación, en conjunto con las vistas para iniciar y cerrar su sesión, con la finalidad de poder hacer un uso más eficiente de las secciones que requieren del uso de datos personales
* En el elemento llamado “Implementar el mecanismo de referencias para el uso de la aplicación”, se realizarán las adaptaciones para que la aplicación pueda usar imágenes de referencias, asimismo, se añadirá la lógica necesaria para que, en caso de que una respuesta tenga una referencia, esta sea desplegada luego de ser verificados los cuestionarios.
* En el elemento llamado “Diseñar una sección para permitirles a los usuarios realizar cuestionarios personalizados” se creará una serie de vistas que le permitirán a los usuarios diseñar sus propios cuestionarios, con la única limitación de que son solo para su uso por la cuenta con la que se creó el cuestionario.
* En el elemento llamado “Integrar los mecanismos para poder evaluar el uso de la aplicación por parte de los usuarios” se implementará la lógica en la aplicación que documentará cuanto tiempo pasa el usuario en qué vistas en un lapso de tiempo semanal, y de esta forma saber qué componentes pueden necesitar mejoras.
* En el elemento llamado “Diseñar e implementar la sección que permite registrar el progreso de los usuarios a lo largo del tiempo” se creará la lógica de la aplicación que le permite saber cuántas preguntas ha respondido el usuario, cuantos cuestionarios ha llenado desde que está registrado, cuantas preguntas ha respondido correctamente, en cuantas se ha equivocado, su porcentaje de aciertos y de fallo, y el tiempo dedicado a responder cuestionarios.
* En el elemento llamado “Implementar todas las secciones que no se han logrado implementar en el transcurso del desarrollo”, se corregirán todas las secciones que estén implementadas, y se implementarán aquellas que no se han logrado implementar, considerando que se tienen elementos nuevos en la aplicación.
* En el elemento llamado “Documentar y probar todas las secciones desarrolladas de la aplicación desarrollada”, se crearán y documentarán las pruebas para validar el correcto funcionamiento de la aplicación en diferentes dispositivos, pasando por los antiguos teléfonos Blackberry, hasta las computadoras más potentes a las que se tiene acceso.

# 4.- Artefactos (Primera iteración)

## 4.1.- Sprint Backlog

Las tareas a realizar en esta iteración son las siguientes, y empieza el día 14 de marzo de 2017, y termina el 21 de marzo de 2017:

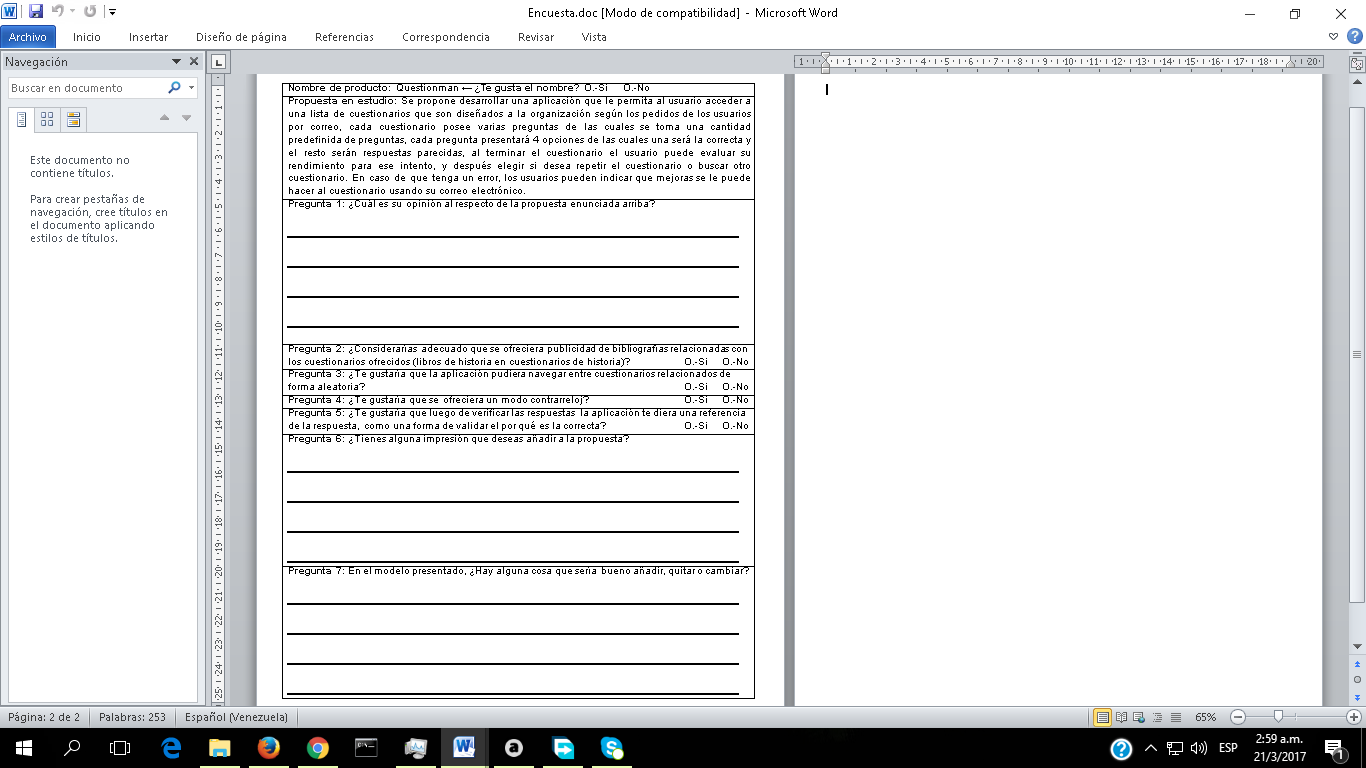


## 4.2.- Incremento

El modelo CANVAS de la aplicación propuesta se encuentra en la página siguiente, y describe de forma detallada, cómo funciona el proyecto:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Fecha:** 21-3-17 | **Propuesta:** Questionfinder | **Empresa:** Vertex Caster Studios. | | | **Creadores:** José Ferrer, José Urbina, Ángelo Tarascio, Edgar Tarascio, Miguel Colmenares | **Iteración:** 1 |
| **8.- Socios Clave:**  - Usuarios quienes serán los que velarán por la calidad del contenido de la plataforma, y lo moderarán usando sus reportes.  - Librerías tanto físicas como digitales, las cuales ofrecerán las obras que tengan en existencia para su publicitación a los usuarios luego de resolver cuestionarios que tienen asociados diferentes bibliografías. | **6.- Actividades clave:**  - Mantenimiento y desarrollo de la plataforma web.  - Publicidad.  - Administración.  - Revisión de cuestionarios.  - Corrección de cuestionarios.  - Gestión de usuarios. | | | **1.- Propuesta de valor:**  Desarrollar una aplicación web adaptable que le permita a una persona acceder a diferentes cuestionarios relacionados con diversos temas, la persona puede ver una de últimos cuestionarios publicados o buscar una coincidencia de títulos, con la finalidad de aportar una herramienta más para el aprendizaje de muchos temas existentes. | **4.- Relación con los clientes:**  - Un usuario solicita la creación de un cuestionario dando el tema del cual quiere que se cree el cuestionario, o el conjunto de preguntas que quiere que se carguen.  - El moderador revisa que todo el contenido sea apto para la plataforma, y revisa aquellos cuestionarios que tengan muchos reportes, y carga los cuestionarios, verificando que respuestas se pueden asociar entre sí.  - Al cargar un cuestionario, el moderador informa al usuario que está listo, indicando el nombre del cuestionario creado.  - Los usuarios informan en caso de que los cuestionarios tengan errores. | **2.- Segmento de Mercado:**  -Estudiantes de educación media y superior, especialmente aquellos cuyos estudios los ponen en contacto con mucho conocimiento teórico.  -Profesores de educación media y superior, cuyas áreas contienen muchos conocimientos teóricos. |
| **7.- Recursos Clave:**  - Personal: Programador Web, Diseñador Web, Publicista, Docente especializado en Metodología del aprendizaje, Contador, Moderador, Gerente.  - Infraestructura: Plataforma Web, un local físico.  - Tecnología: Servidor, Dominio Web, Bases de datos, Telefonía.  - Capital: Computadores, Teléfonos, Equipo de Oficina y Bibliografía en Digital. | | | **3.- Canales:**  - Online, a través de la aplicación web adaptable a cualquier dispositivo que soporte el motor Webkit.  - Correo Electrónico, por el cual los usuarios solicitan la creación de los cuestionarios, el cual puede incluir un pago por publicación priorizada. |
| **9.- Estructura de costos:**  Sueldos y salarios, adquisición y mantenimiento de equipos tecnológicos, servicios públicos e impuestos, hosting de la plataforma, publicidad en línea, material bibliográfico en los casos que sea estrictamente necesario.  Modelo de negocio orientado al valor. | | | **5.- Fuente de Ingresos:**  - Promoción de cuestionarios: Los usuarios interesados por la rápida publicación de sus cuestionarios pueden pagar para que el o los moderadores publiquen antes sus cuestionarios.  - Publicidad de editoriales: Las tiendas que venden libros electrónicos y físicos podrán publicar obras que sean acordes a los cuestionarios que un usuario esté resolviendo.  Se aceptarán pagos vía transferencias y depósitos tanto para las publicaciones priorizadas como para la publicidad, que serán verificados antes de ser realizados. | | | |

El siguiente es el formulario utilizado para la ejecución de las encuestas, se encuentra en el archivo Modelo\_Encuesta.doc:

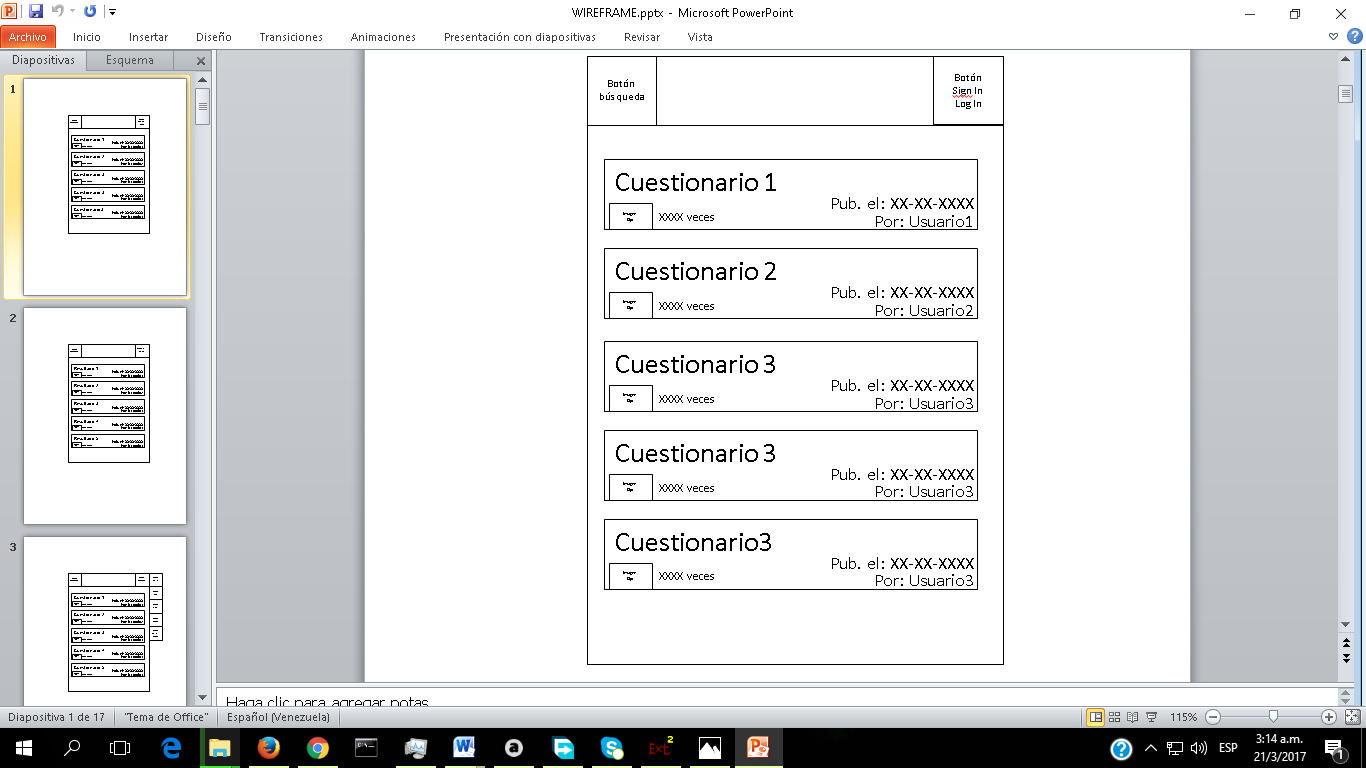
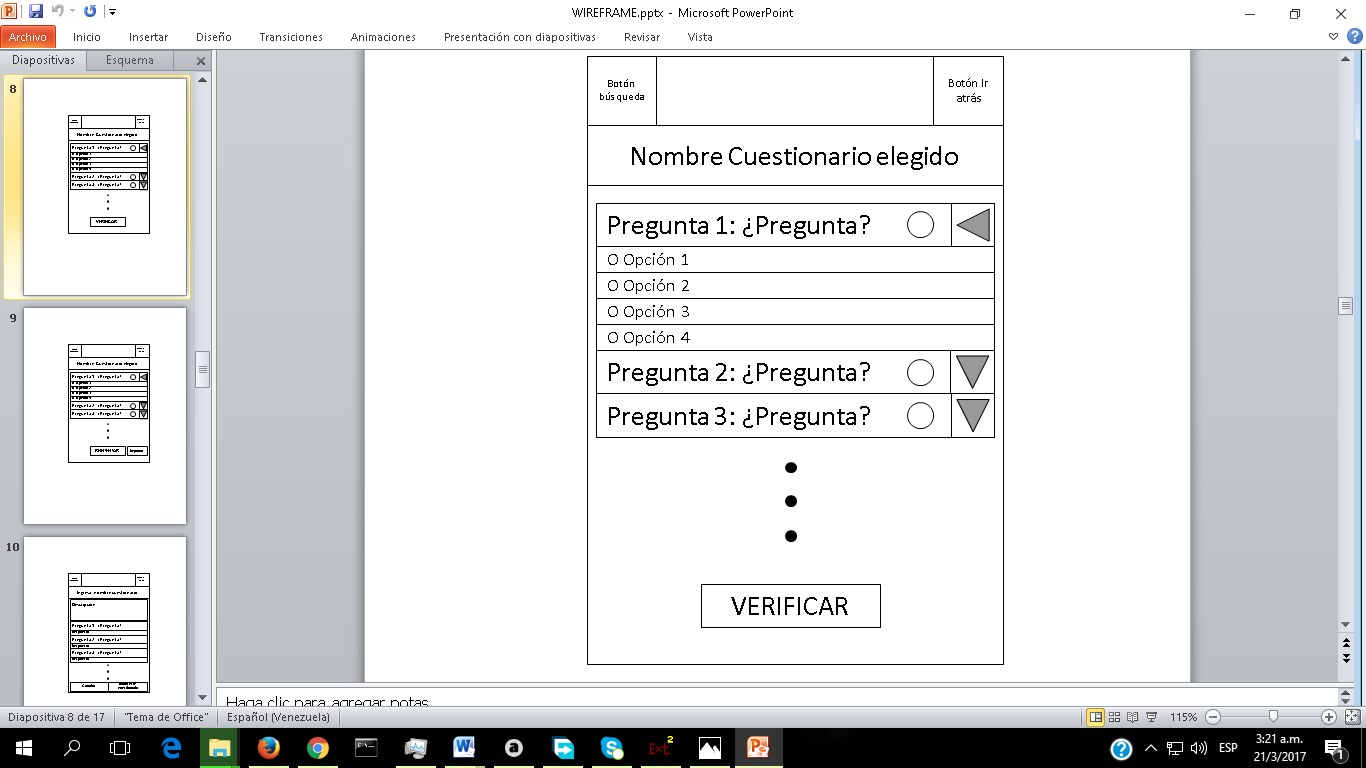


Nótese que se usa el nombre “Questionman”, cuyo logotipo es el siguiente:



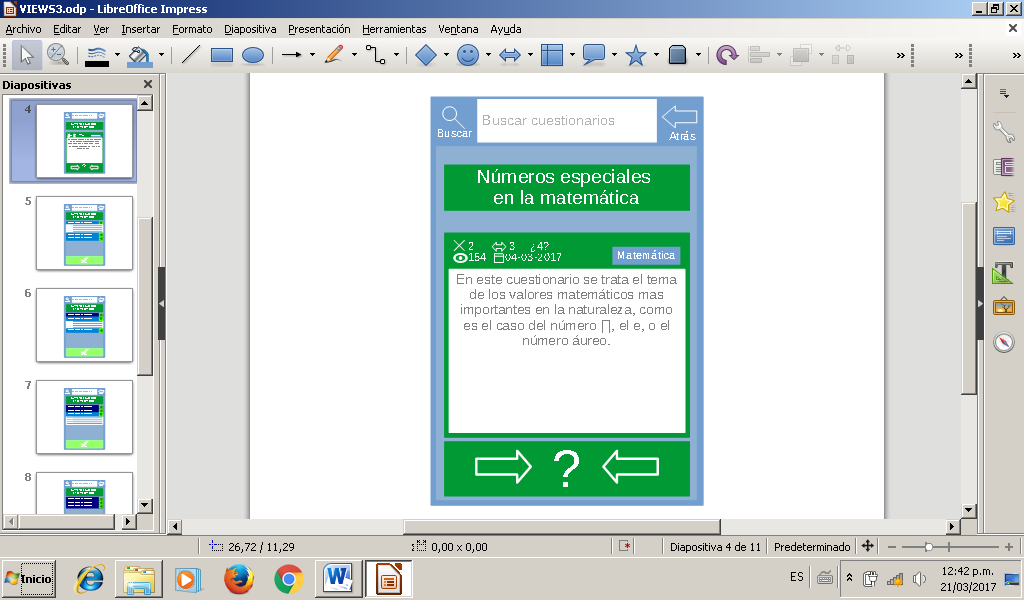
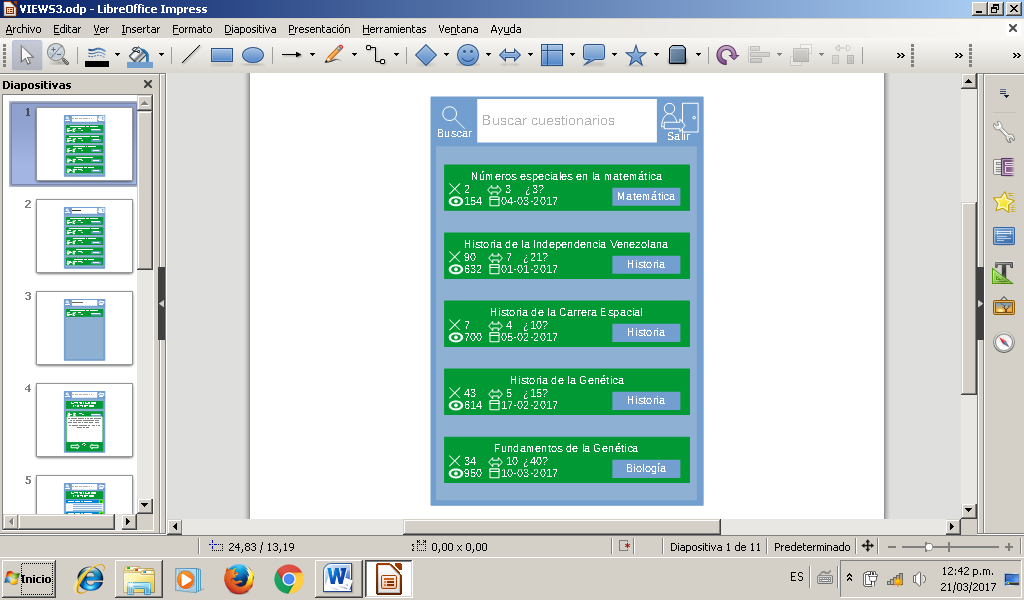
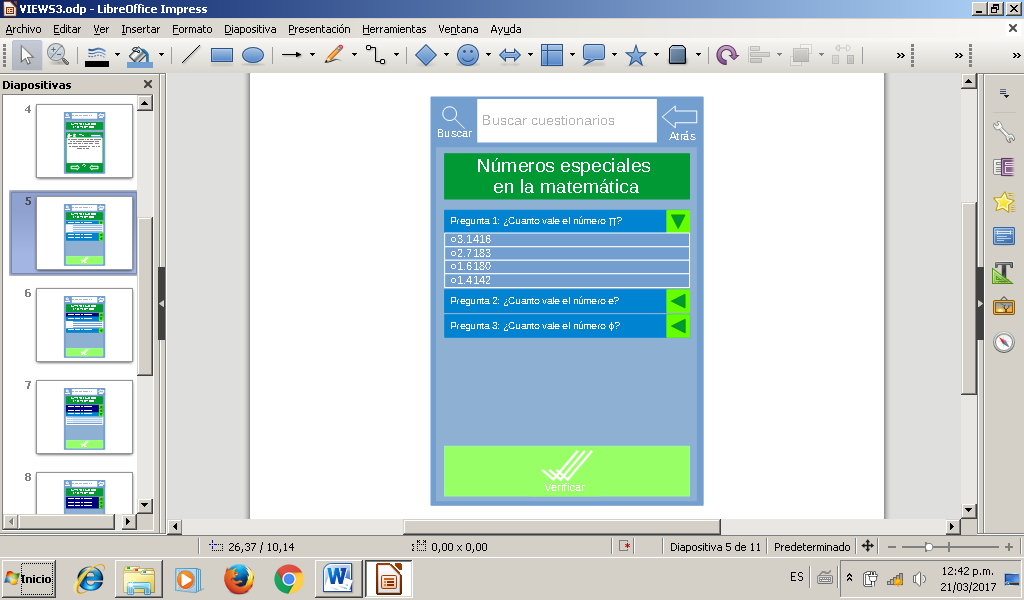
Pero deberá de ser modificado debido a la mala recepción del nombre por parte de los usuarios en las entrevistas y en la evaluación de los modelos realizada.

El diseño de la malla de la aplicación es el siguiente, la mayoría de las vistas se asemejan a esta, que se encuentran en el archivo WIREFRAME.pptx:

Sin embargo, la característica de Signin/Login/Logout fue removida por no aportar suficiente valor a los usuarios, pues las únicas razones por la que existía esta función es permitirles a los usuarios crear por su cuenta los cuestionarios, cosa que podría provocar un desastre en la base de datos en línea de la aplicación, por lo que dicha funcionalidad fue movida a la organización, que también hace las veces de filtro para los cuestionarios, y así evitar que ideas que no son moralmente correctas lleguen a las personas, como es el caso de las ideas xenofóbicas y afines.

La aplicación de prueba en Impress tiene la siguiente apariencia, y se puede encontrar en VIEWS\_pretest.odp:

Con este modelo, se desarrolló una prueba de usabilidad con 6 usuarios diferentes, tomados de los 9 potenciales usuarios entrevistados.

Los resultados de la encuesta fueron los siguientes:

* De los 9 usuarios consultados, a 6 no les gustó el nombre de “Questionman” y otro consideró atractivo el nombre de “Questiongun”, a 4 les gustó el estilo del logotipo, por este resultado adverso, el nombre del proyecto cambia a “Questionfinder”.
* De los usuarios consultados 7 indicaron que no le veían la lógica a la aplicación, por diferentes razones, las más enumeradas eran que las personas no siempre gustan de usar cuestionarios para aprender, y que la lógica de la aplicación no era clara, para aquellas dos a las cuales les gustó, indicaron que la idea es interesante y diferente.
* De las personas encuestadas, todas ellas estuvieron de acuerdo en que la aplicación tuviese publicidad de libros acordes a los cuestionarios que se ofrecen en dicha aplicación.
* De las personas entrevistadas, todas estuvieron de acuerdo en añadir una modalidad de navegación aleatoria, en la cual los usuarios pasan a un cuestionario con contenido parecido luego de terminar un cuestionario.
* De las personas entrevistadas, solo dos se oponen a la existencia de un modo contrarreloj, el resto, se siente favorable a la idea de crear dicho modo.
* De las personas consultadas, solo 2 se oponen a la existencia de un modo contrarreloj, el resto, se encuentra favorable a la idea.
* En lo que respecta a posibles añadidos, uno de los entrevistados propuso añadir un mecanismo de acumulación de experiencia a medida que se van resolviendo los cuestionarios, otros 2 indicaron que se debe de especificar aún más la propuesta en desarrollo, así como el uso de imágenes en las preguntas que la aplicación muestre (esto último incrementaría el consumo de datos de la aplicación).
* En la prueba de usabilidad los usuarios realizaron las siguientes recomendaciones de diseño:
  + Permitir la edición de los cuestionarios (el problema con esta historia de usuario es que entonces se deja la puerta abierta a una inundación de “troles” que hagan cuestionarios basura), por lo cual se descarta por el momento.
  + Colocar un manual de acceso a la aplicación ya sea como un menú desplegable o una pantalla deslizante de arriba abajo pues la aplicación necesita una forma de manual, a lo cual se le deberá de añadir una serie de instrucciones acerca de cómo utilizar la aplicación.
  + Los íconos de Salir, Iniciar cuestionario, Reportar y enviar reporte, están demasiado cargados, deben de ser más sencillos en términos de gráficos.
  + El ícono usado para indicar la cantidad de reportes que tiene un cuestionario confunde al usuario en el sentido de que en lugar de representar la cantidad de errores, representa algo parecido a una cantidad de solicitudes de eliminación, se debe cambiar por una señal de atención.
  + En los reportes, cambiar el tamaño y fuente de la letra pues no son legibles y su tamaño es excesivo.
  + Uno de los usuarios consideró que en lugar de una serie de elementos, se debería de usar una lista desplegable para seleccionar los elementos, sin embargo, esto daría lugar a que el usuario no pueda ver sus opciones para responder una pregunta, pudiendo frustrar al usuario cuando al terminar de marcar todas las preguntas de algún cuestionario (más si es largo), el botón de verificar no se activa pues le faltó una pregunta por responder.
  + El alto de la barra de búsqueda es excesivo, por lo que se deberá cambiar el tamaño de esta barra en particular.
  + El tamaño de algunos botones es excesivo, debe de ser reducido.
  + Algunos usuarios encontraron que la aplicación era ambigua en el sentido que no se definen de qué temas son los cuestionarios, se debe de describir en la sección de instrucciones cual es la finalidad de la aplicación, incluyendo la realidad de que, cualquier tema que pueda ser convertido en preguntas y respuestas, puede ser cargado en la aplicación.
* En la misma prueba de usabilidad, los usuarios destacaron los siguientes aspectos:
  + Sencillez en términos de manejo de información.
  + Valor de la propuesta, en el sentido de que se combina con muchas otras actividades no pensadas (como por ejemplo, puede usarse como un examen académico, o como parte de un juego de conocimiento).
  + Práctico, en el sentido de que la aplicación no se centra tanto en detalles sino en el funcionamiento.

Todos estos resultados se encuentran documentados en Resultado\_Encuestas.docx, que se encuentra en el mismo directorio que este informe.

## 4.3.- Cálculo de tiempos

El tiempo invertido para cada una de las actividades desarrolladas fue el siguiente:

 La diferencia se justifica por las siguientes razones:

1. La elaboración del prototipo en Impress presentó el inconveniente de que no se pueden crear hiperenlaces sobre un grupo, por lo que se utilizó parte del tiempo para determinar la solución más adecuada, que fue sobreponer un objeto transparente sobre el objeto que hace las veces de botón.
2. Aparte, realizar las entrevistas y las pruebas se demoró más tiempo del esperado porque se entrevistaron más personas de las esperadas a entrevistar, esto debido a que las personas entrevistadas originalmente no respondieron los formularios adecuadamente.
3. Finalmente, la prueba de usabilidad ameritó usar una instalación de GNU/Linux pues el computador utilizado para capturar las pruebas carecía de la potencia suficiente para ejecutar un programa de captura de video con interfaz gráfica.

## 4.4.- Gráfico de Burn-up

## 4.5.- Gráfico de Burn-down

# 5.- Artefactos (Segunda Iteración)

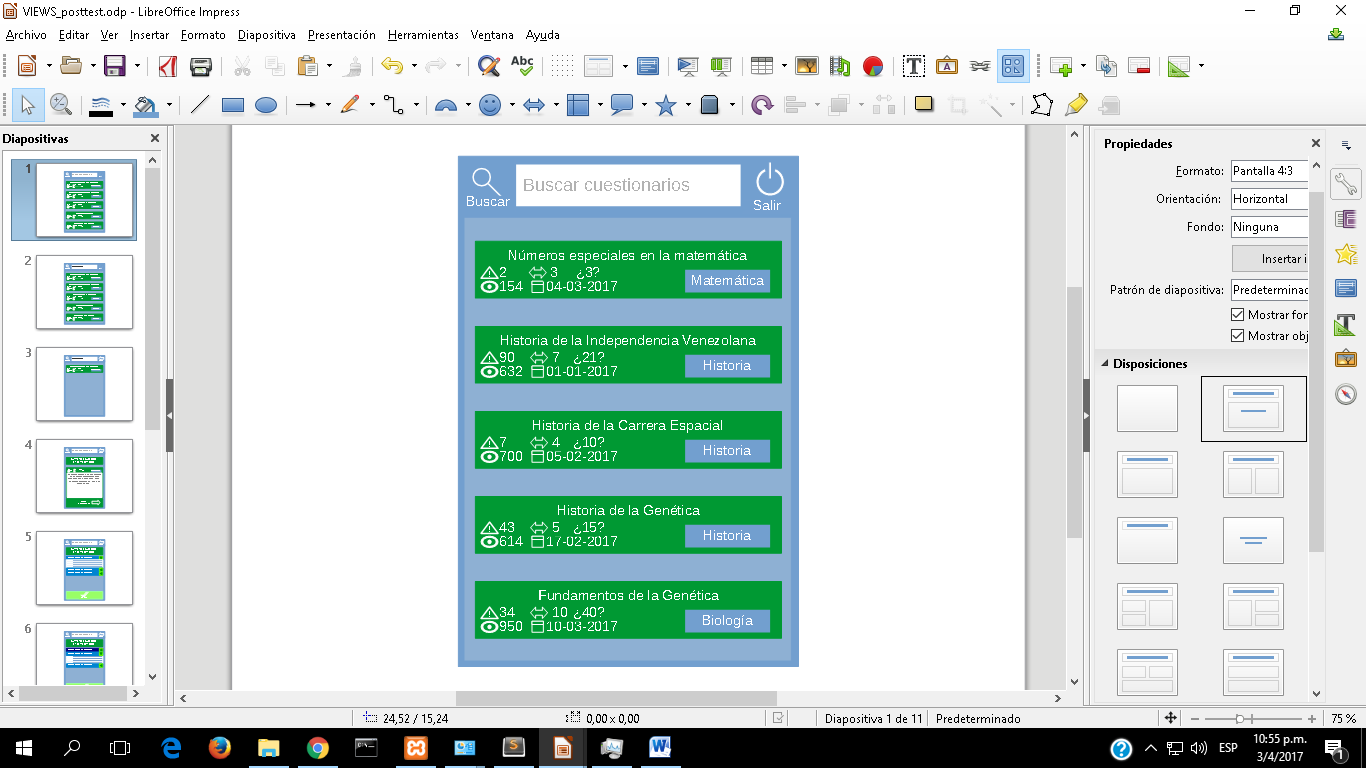
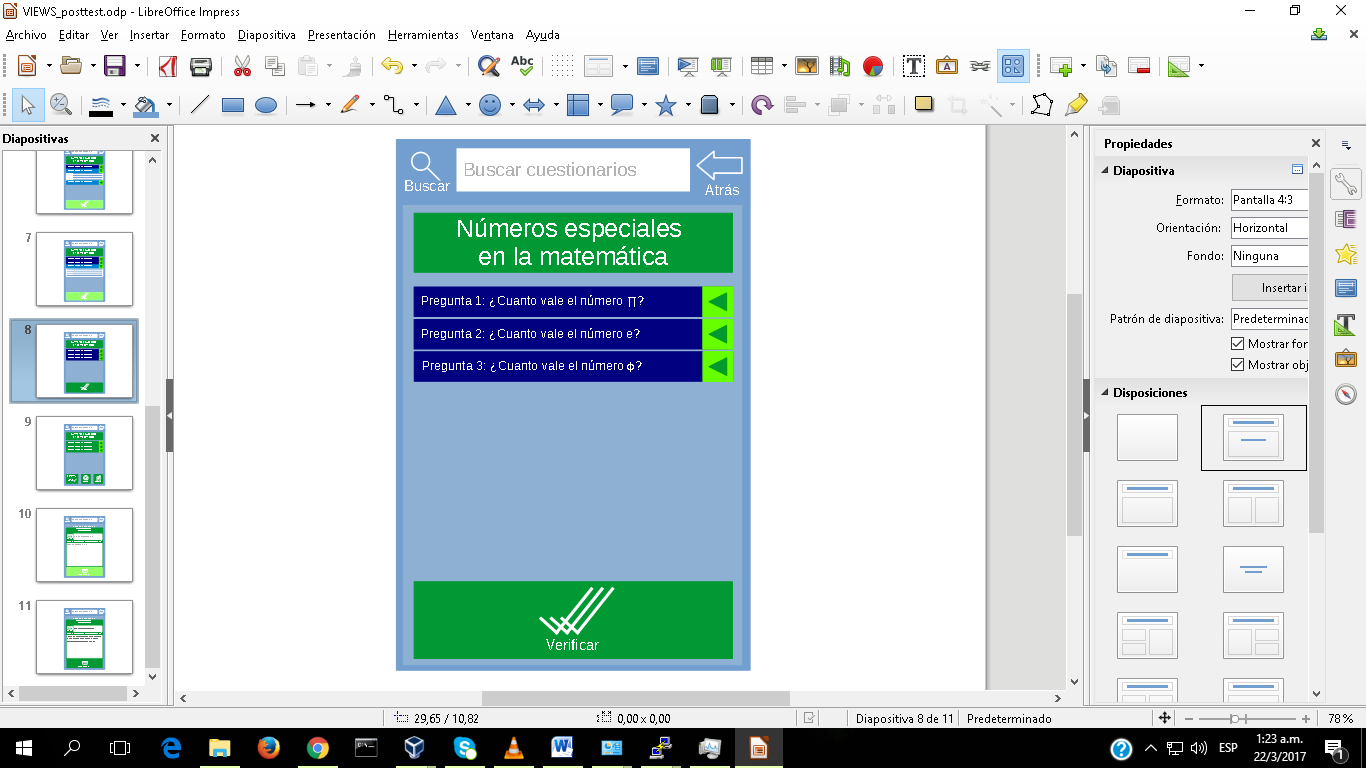
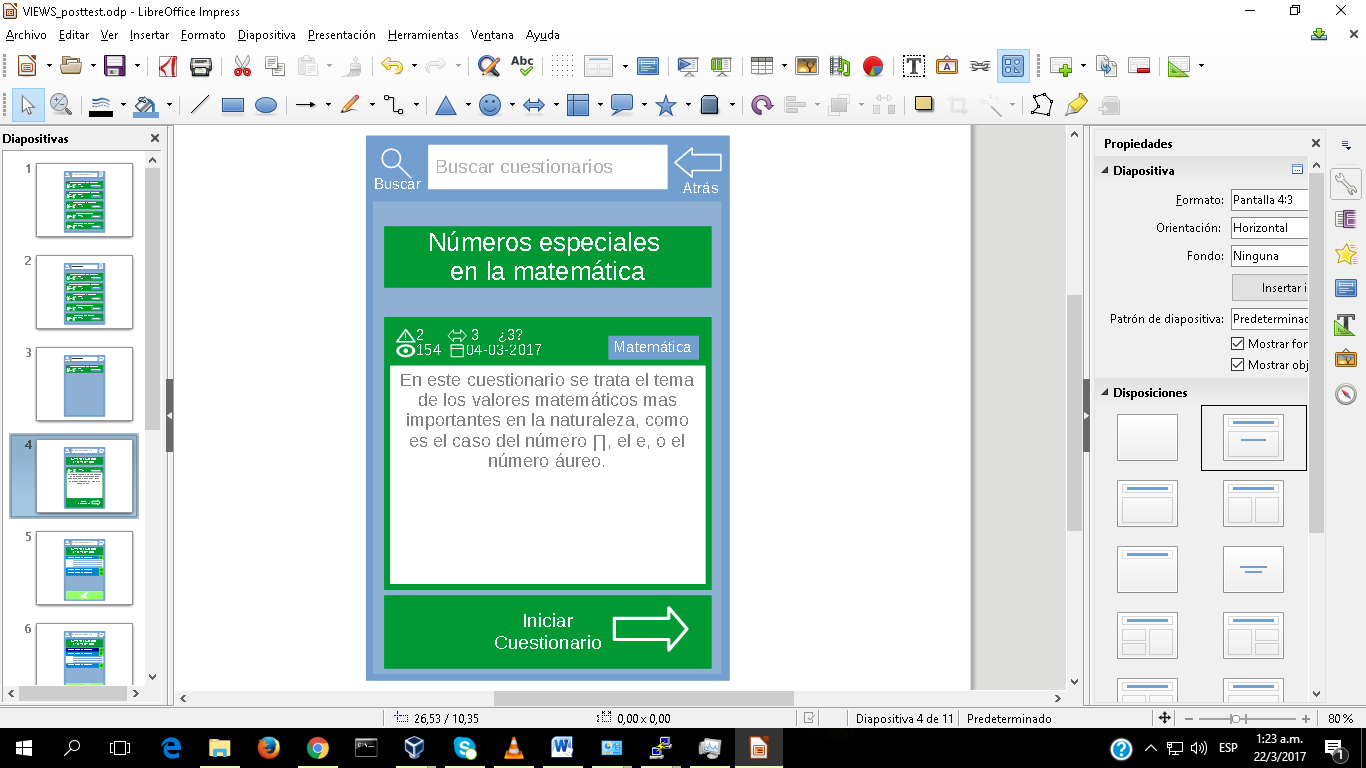
## 5.1.- Sprint Backlog

Este ciclo inicia el 21 de marzo de 2017, y termina el 4 de abril de 2017, realizándose las siguientes actividades:



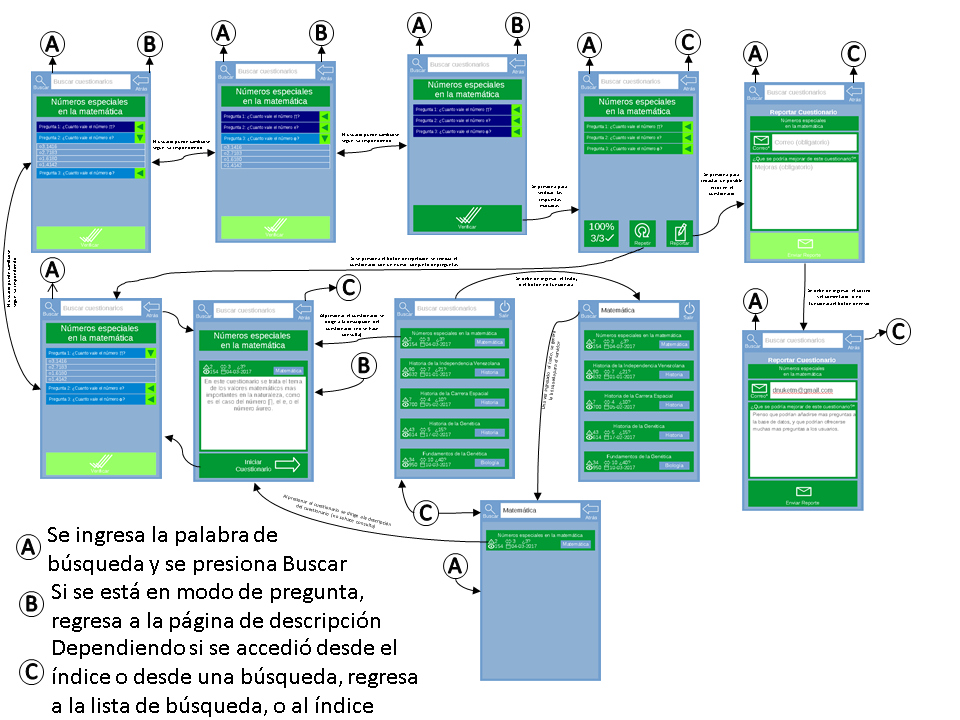
## 5.2.- Incremento

Estas son las nuevas vistas de la aplicación en desarrollo, como se aprecia, los cambios fueron mínimos:

Se puede manipular el modelo accediendo al archivo VIEWS\_posttest.odp.

El diagrama de rutas es el siguiente:



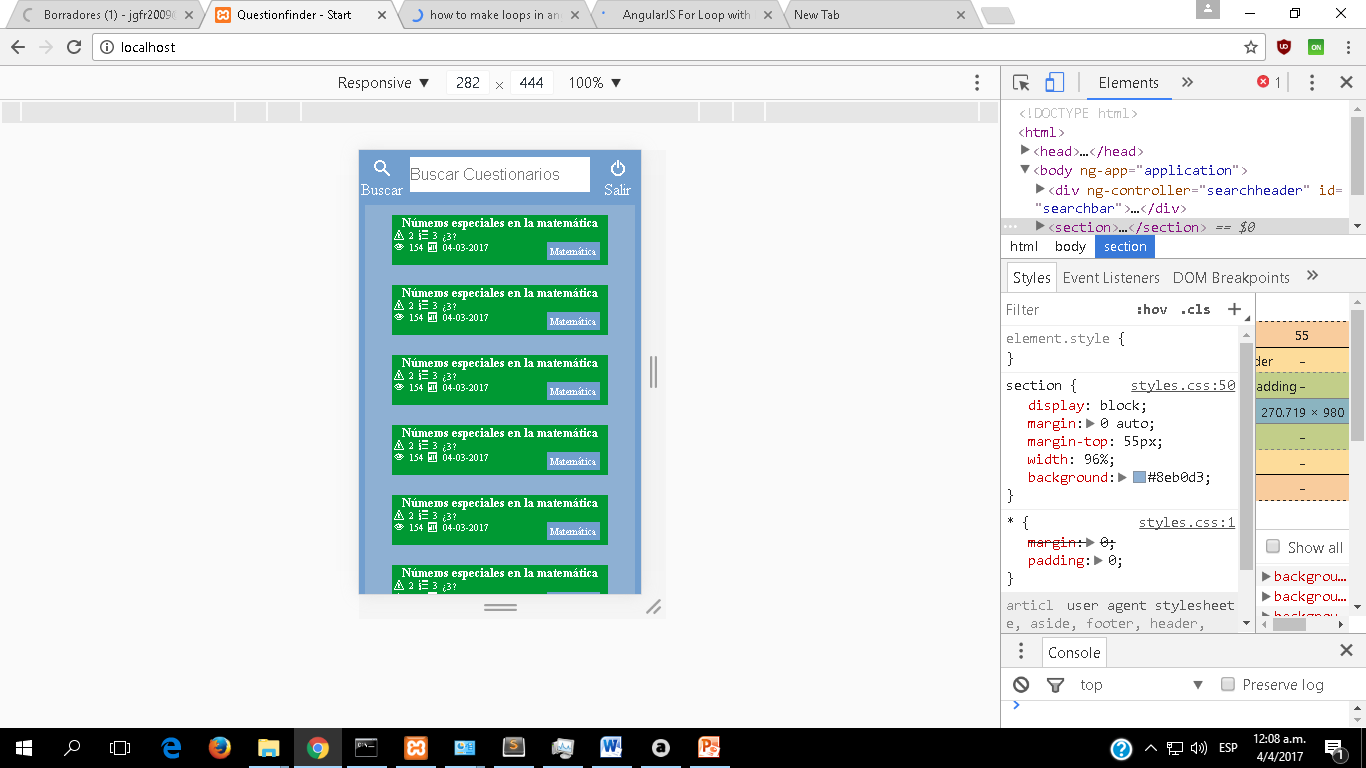
Se tiene más información si se accede al archivo routes.pptx, en el cual se aprecia mejor la funcionalidad de cada uno de los botones y la forma en que los usuarios pueden navegar por la aplicación.

El nuevo logotipo de la aplicación es el siguiente:

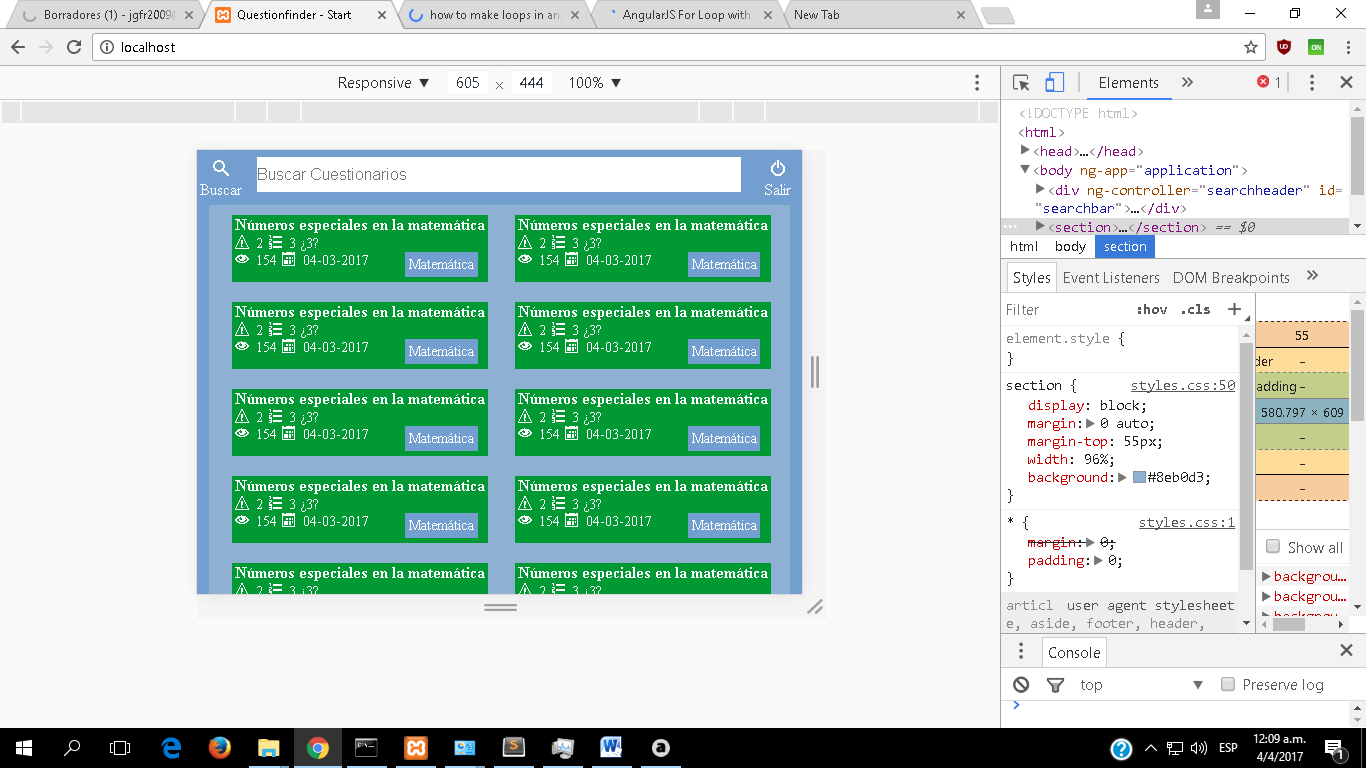


El prototipo también fue realizado, aunque solo en su parte visual, faltando por implementar la lógica de control de la aplicación, una captura del prototipo es el siguiente, mostrando el principio de responsividad al tamaño de la pantalla:

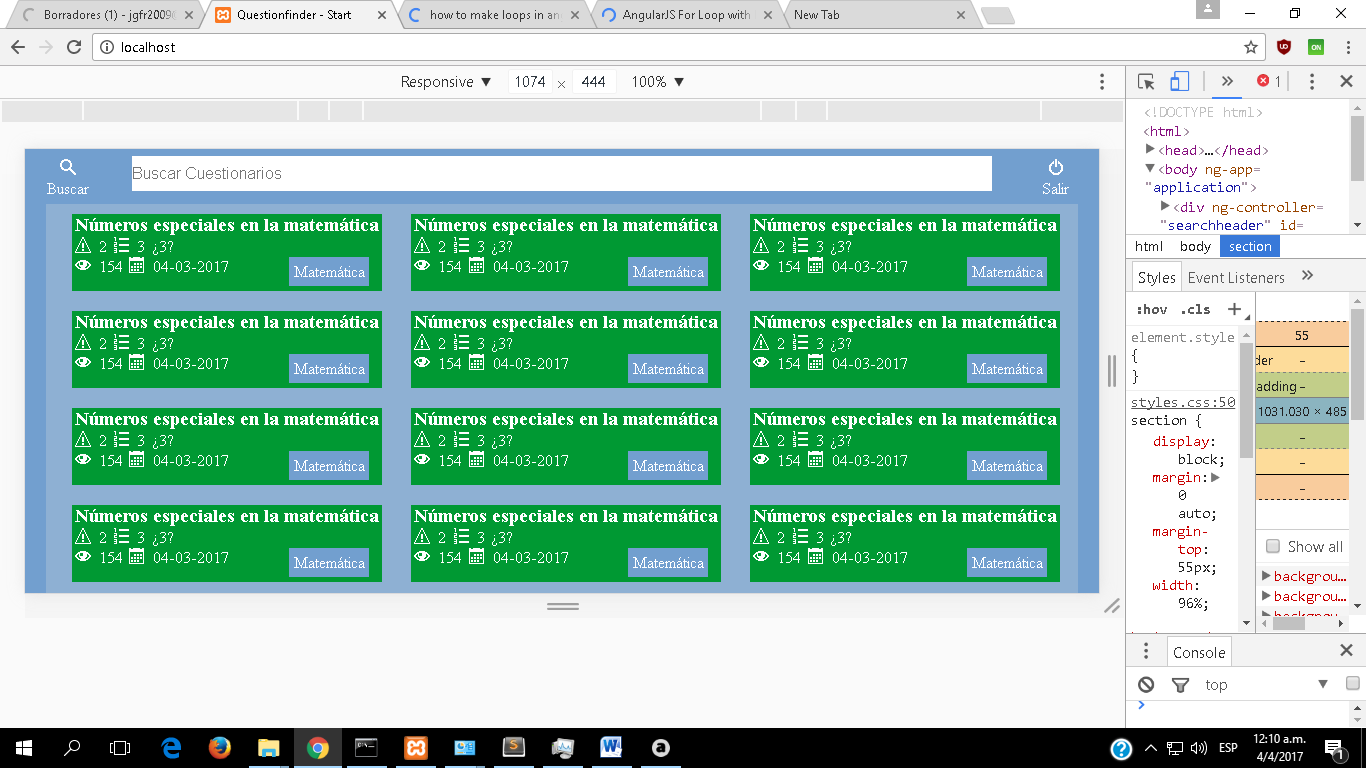
Móvil primero:



Pantallas medianas:



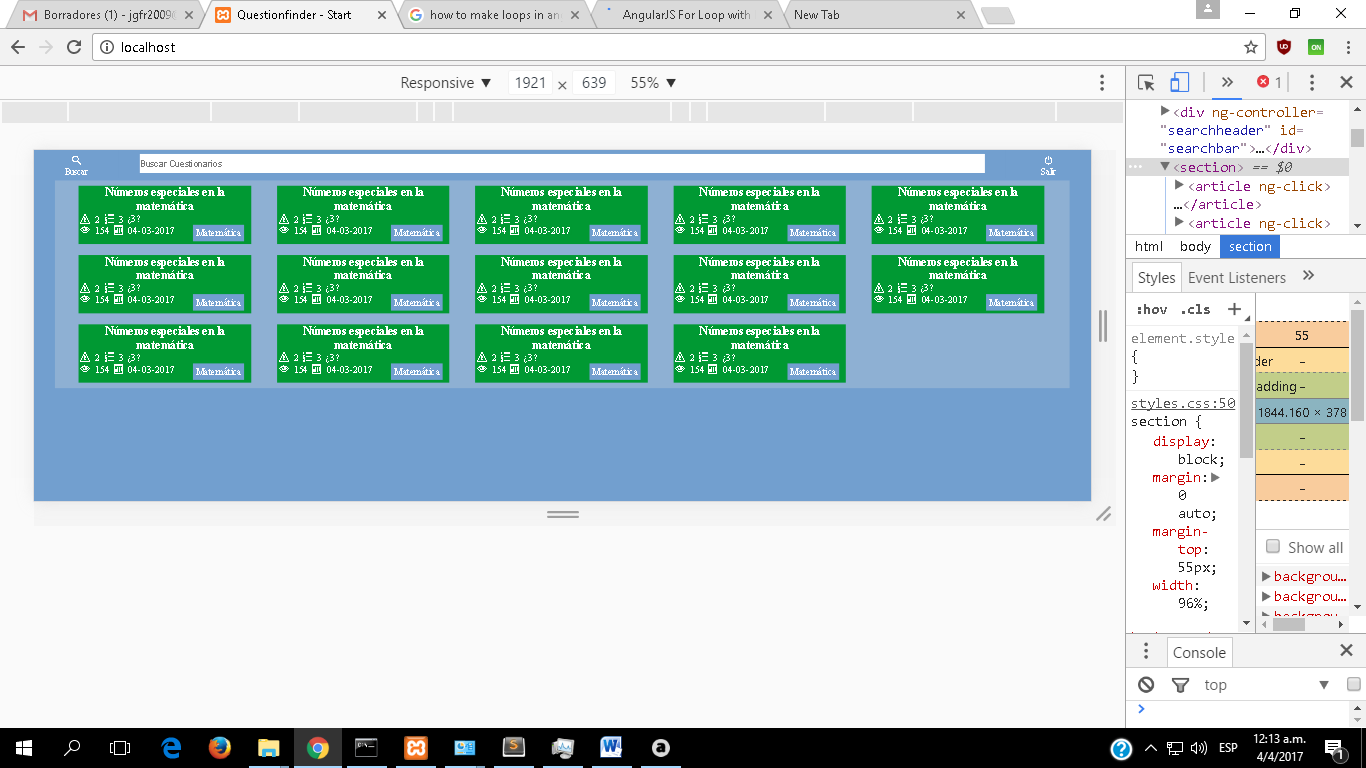
Luego para pantallas de tableta:



Escritorio Tradicional:



Pantallas FullHD:



La lista de requisitos para el desarrollo de la aplicación es la siguiente:

1. Debe de crearse una barra de búsqueda, donde el botón de búsqueda genera una petición con los términos de entrada, y el de retroceso aplica el patrón de diseño Memento, con el cual se espera implementar el retroceso entre las vistas, si no hay más puntos para retroceder, se cierra la aplicación.
2. Cada uno de los cuestionarios mostrados, ya sea del índice o de una búsqueda debe de mostrar la descripción del cuestionario cuando son seleccionados, y debe de ofrecerle al usuario la posibilidad de retroceder, buscar otros de cuestionarios, o retroceder a la búsqueda realizada.
3. Si se inicia el modo de preguntas de un cuestionario en particular, se deben de cargar las preguntas y respuestas de cada cuestionario, y deben de asignarse según su tipo de pregunta hasta 4 respuestas diferentes (y si solo tiene hay una respuesta parecida, no presentarla nunca en los cuestionarios), para que así, preguntas con respuestas similares sean agrupadas en un mismo elemento de interrogación.
4. En el modo de preguntas, si una pregunta está sin responder se debe de marcar con un color claro, sino, debe de ir con un color oscuro.
5. Cuando todas las preguntas cargadas tengan una respuesta, es cuando se debe de habilitar el botón de verificación.
6. Al verificar, revisar si el identificador de la respuesta, si el identificador de la pregunta que responde coincide con el de la pregunta realizada.
7. Luego de verificar, permitirle a l usuario reiniciar el cuestionario, cambiando las preguntas realizadas.
8. Luego de verificar un cuestionario, permitirle al usuario reportar algún error, indicando este su correo y la falla detectada en el cuestionario que acaba de resolver.
9. En pleno modo de preguntas y respuestas, si el usuario retrocede a la descripción y luego vuelve a iniciar el cuestionario, deben de cambiarse las preguntas mostradas al usuario, sin generar una nueva consulta a la base de datos.

## 5.3.- Cálculo de tiempos

El tiempo invertido para cada una de las actividades desarrolladas fue el siguiente:



## 5.4.- Gráfico de Burn-up

## 5.5.-Gráfico de Burn-down

# 6.- Artefactos (Tercera Iteración)

## 6.1.- Sprint Backlog

Esta iteración empezó el 4 de abril de 2017 y terminó el 18 de abril de 2017. Las tareas a realizar en esta iteración fueron las siguientes:

## 6.2.- Incremento

Si se usa la biblioteca Angular, creada por Google, esta utiliza diferentes servicios que extienden las capacidades de la versión pura de la biblioteca, entre ellas se tienen: Llamados asíncronos por HTTP, Rutas, Recursos, Animaciones, entre otras características que se pueden implementar con angular, pero que no vienen por defecto con la finalidad de disminuir el tamaño de archivo de la biblioteca base, en la prueba de concepto (que es el nombre dado a la aplicaciones que solo implementan algunas características), se implementan solo 2 características principales, el listado con búsqueda, y la carga de los detalles de un solo cuestionario, el resto de las funciones permanecen bloqueadas por razones de simplicidad. La razón por la que no se utiliza la versión 2 de la biblioteca es que dicha biblioteca es aún más compleja que su versión anterior, y el tiempo es oro en la elaboración de proyectos.

El servicio que provee el enrutamiento se llama ngRoute, y se usa para poder crear aplicaciones SPA, es decir, aplicaciones de una sola página, en las cuales se le da al usuario la apariencia de que la aplicación no recarga, pero la realizada es que hace el cambio en segundo plano, para crear una ruta se define dicho módulo de angular cuando se crea la aplicación, y luego se usa una estructura de código similar a la siguiente:

*app.config(function($routeProvider){*

*$routeProvider.*

*when("/details/:id/",{*

*templateUrl: "pages/details.html",*

*controller: "detailsController"*

*}).*

*when("/formlist",{*

*templateUrl: "pages/formlist.html"*

*}).*

*otherwise({*

*redirectTo: "formlist"*

*});*

*});*

El servicio que provee la funcionalidad de enrutamiento se llama $routeProvider, y bajo el capó lo que hace es valerse de plantillas las cuales son cargadas a demanda según las usa el usuario, para pasar un parámetro a una ruta, se utiliza el nombre de la variable a pasar como parámetro, antepuesta con dos puntos, para añadir variables adicionales, se deben separar con barras inclinadas, y todas deben de tener dos puntos antes de la definición, para acceder a los parámetros pasados a una ruta, se utiliza el servicio $routeParams, el cual contiene todos los parámetros pasados a una ruta. Las plantillas que se usan para implementar el enrutamiento pueden estar en el mismo código, o separada de esta, la dirección de estas plantillas se coloca en el parámetro *templateUrl*, y, en caso de que se necesite, un controlador a ejecutar en caso de que se necesite ejecutar dicho controlador, en la página principal, se debe de crear un par de etiquetas div, que tengan la propiedad ng—view, ya que esta etiqueta indica cual será el viewport en la aplicación, donde se verán las plantillas, que se incluyen según se llamen en las rutas.

Ahora bien, al tratarse de una aplicación móvil, ella debe de obtener los datos que va a desplegar desde alguna parte, en este caso, se usa como proveedor la dirección “qfapi.herokuap.com”, y para poder hacer pedidos asíncronos (recordar que se usa enrutamiento para generar el efecto de no recarga), se usa el servicio $http (es posible usar $resource, pero dicho servicio es aún más complejo que $http, y encima no es parte de la biblioteca estándar de Angular), el servicio $http implementa las siguientes funciones relevantes para el desarrollo de la prueba de concepto de la aplicación:

* .get(URL): Este método realiza una petición a un servidor remoto usando el verbo GET a la dirección especificada por el parámetro URL, se usa para solicitar datos a un servicio.
* .post(URL,[objeto]): Este método realiza una petición a un servidor remoto usando el verbo POST, este verbo se utiliza para crear registros en el servidor remoto.
* .delete(URL): Este método realiza una petición al servidor remoto usando el verbo HTTP DELETE, este verbo se usa para eliminar un registro en particular de la aplicación.
* .put(URL,[objeto]): Este método realiza una petición al servidor remoto usando el verbo HTTP PUT, este verbo se usa para modificar un registro en particular de la aplicación, indicándose además el objeto que reemplazará en el servidor los registros que coincidan con el parámetro pasado.

Una vez llamado el verbo con su ruta (y el objeto opcional), se debe de implementar el retorno, para estos retornos se debe considerar si se recibe un código de operación exitosa y uno de operación fallida, los códigos de estado HTTP de éxito van desde el 200 al 299, y cualquier cosa por debajo y por encima de este rango se considera fallida la petición. Esto se hace de la siguiente forma:

*$http.get("ruta")*

*.then(function(response){*

*/\*Función a ejecutar si se tiene éxito\*/*

*},function(response){*

*/\*Función a ejecutar si falla la petición\*/*

*});*

Cabe remarcar que el método GET en el código de ejemplo, se puede cambiar por cualquiera de los otros métodos ilustrados previamente, pero se deben de implementar tanto la función de éxito como la de error.

En el caso de los filtros, estos retornan una función, que a su vez retorna datos, los cuales se pueden usar para generar listas a partir de algún resultado, la implementación básica de un filtro es la siguiente:

*aplicacion.filter('searchforms', function(){*

*return function(data, searchterm){*

*return datos;*

*};*

*});*

Se tiene una definición de nombre del filtro, y una función anónima, que retorna una función, que es la que se encarga de retornar los datos de la aplicación, el filtro se llama generalmente dentro de un ng-repeat=”” dentro del cual se tiene tanto la estructura repetitiva seguido del filtro a utilizar, separado con una tubería (el carácter |).

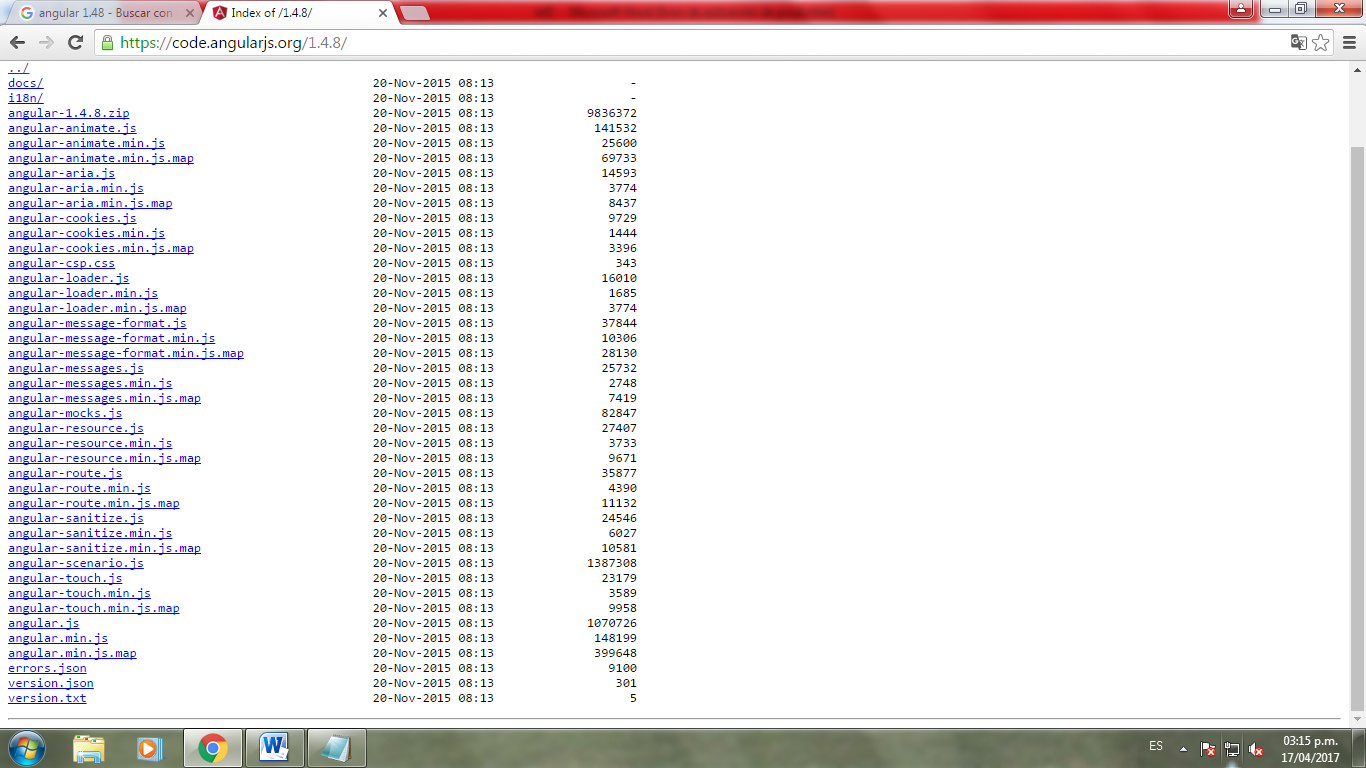
Un controlador de Angular se construye de la siguiente forma:

*app.controller('controladorController', function ($servicios){*

*})*

Se tiene que primero se debe de indicar el nombre del controlador (el cual se llama con ng-controller), seguido de una función a la cual se le pasan como parámetros los servicios que se desean utilizar dentro de esta, entre los que se cuentan el $http, el $routeProvider, $routeParams, y el $scope, el cual es el que permite asociar las variables con la lógica de la aplicación, utilizando el atributo ng-model y ng—bind, el primero se usa para que una variable sea capturada, y con el segundo se puede desplegar dicha variable.

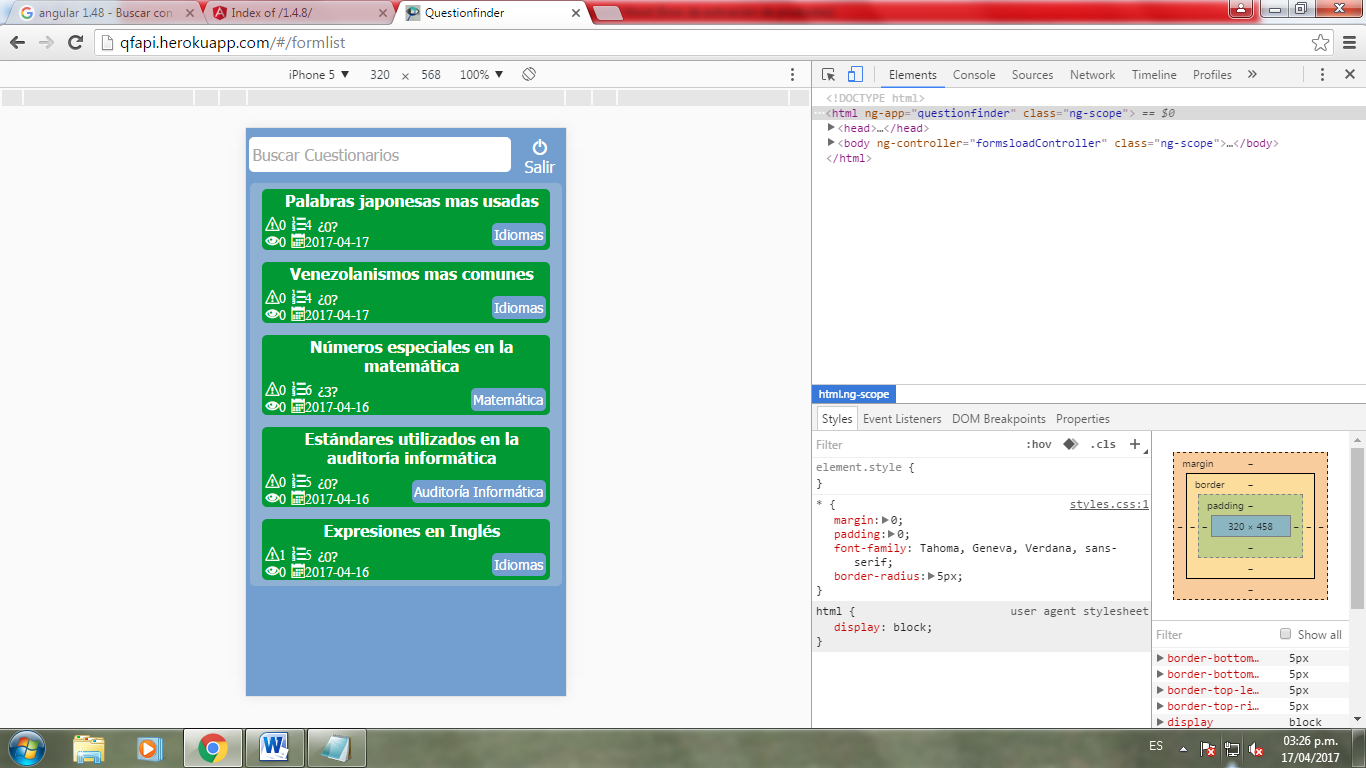
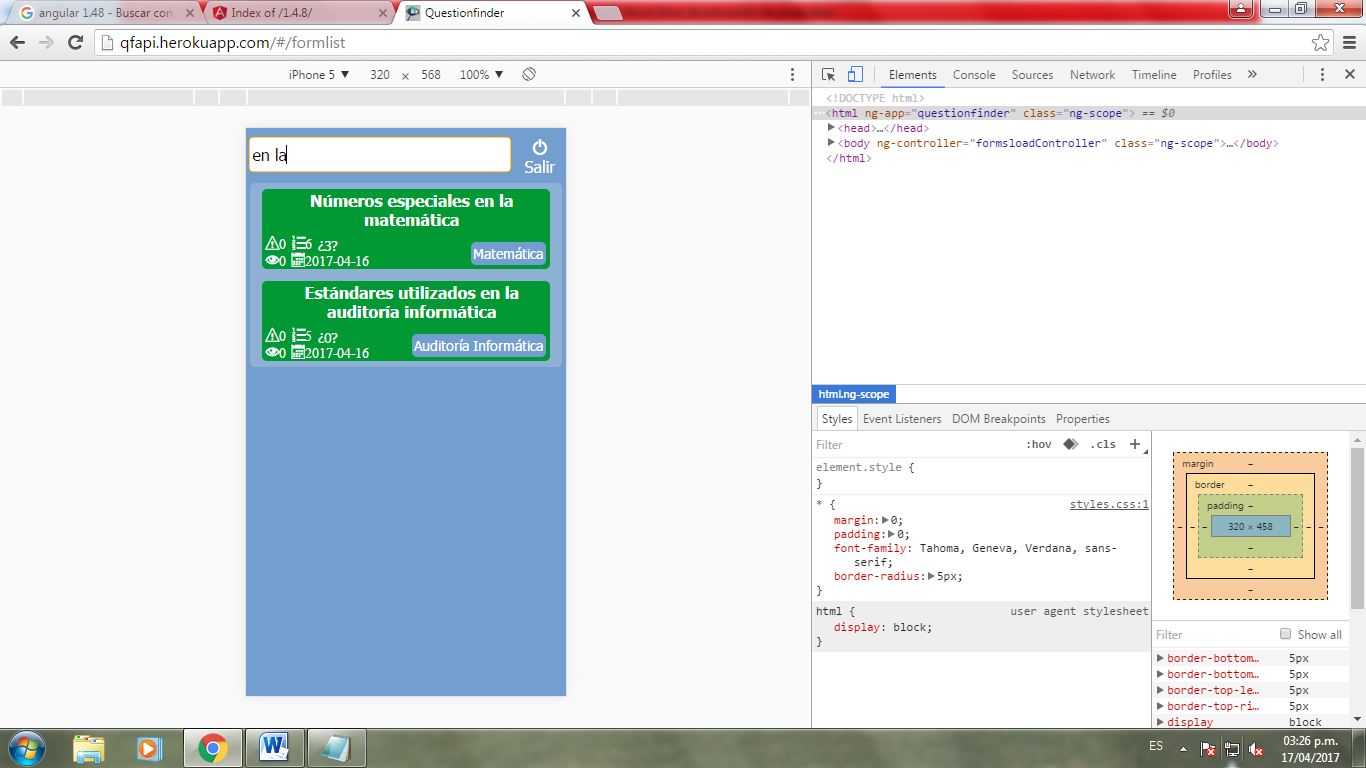
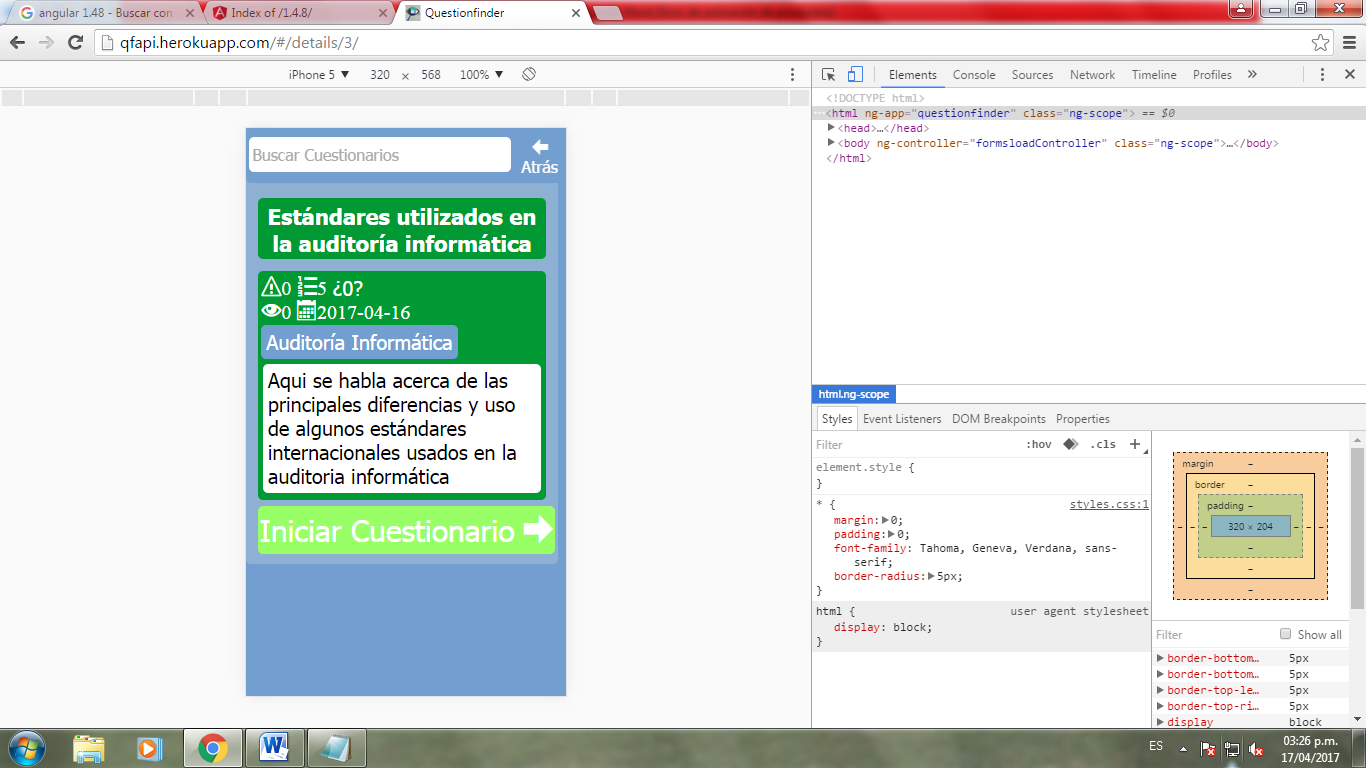
La biblioteca de Angular se puede encontrar en la siguiente dirección: <https://code.angularjs.org/1.4.8/>, el cual se puede enlazar como una biblioteca, o se puede descargar para ser usado en el servidor directamente, sin depender de un servidor externo (y así acelerar el funcionamiento de las aplicaciones), a costa de que, si el código se actualiza, se deberán actualizar las bibliotecas usadas de forma manual.



La aplicación es la siguiente: Se implementaron 5 funciones:

* El filtrado de cuestionarios.
* El listado de cuestionarios.
* La salida.
* Retroceso del detalle de cuestionario.
* El detalle de un cuestionario.

Estas son las vistas de pantalla

## 6.3.- Cálculo de tiempos

El tiempo invertido para cada una de las actividades desarrolladas fue el siguiente:



## 6.4.- Gráfico de Burn-up

## 6.5.- Gráfico de Burn-down

**7.- SCRUM Retrospective**

**7.1.- Cálculo de tiempos**

El tiempo invertido en todo el proyecto es el siguiente:



**7.2.- Gráfico de Burn-up**

**7.3.- Gráfico de Burn-down**