

# ***Práctica 5***

## ***Las cosas que no vemos***



Ana María Martínez Valdés  
David Chaparro García  
Eros Guerrero Sosa  
Iñigo Rolando García  
Ignacio Villegas de Miquel  
Johan Cruz Huertas  
Javier Gil Caballero  
Luis Enrique Barrero Peña  
Santiago Gabriel y Galán Puelles  
Shuyi Wang

# Índice

<b>Introducción</b>	2
<b>Arquitectura</b>	2
<b>Prototipo</b>	3
Configuración del entorno de administración de la configuración	3
Herramientas del prototipo	5
Desarrollo del prototipo	5
Instrucciones que indiquen cómo ejecutar y probar el prototipo	6
Para desarrolladores	6
Para usuarios	7
<b>Product Backlog</b>	8
<b>Actividad grupal</b>	11
Descripción	11
Desarrollo	11
Resultados	12

# 1. Introducción

Para la realización de esta práctica, el coordinador dividió el trabajo entre todos los miembros restantes, respetando el rol de cada uno a la hora de hacerlo. De esta manera, al equipo de desarrollo le fue asignada la parte de la memoria del prototipo, pero como el coordinador es el Scrum Master, la división del propio prototipo fue realizada en consenso entre el equipo de desarrollo. El scrum master y el product owner realizaron la exportación del PB y del User Map a Jira y rellenaron la memoria. Para que la división quedase más clara, el coordinador hizo una tabla con las tareas de cada persona (tabla 1.1).

	PARTE	actividad 1	actividad 2		DESCRIPCIÓN
Ana	prototipo		código		realizar el código del prototipo ( a la hora de la división hara un poco más que aquellos con actividad 1)
David	prototipo	rellenar intro 3	código		realizar el código del prototipo y rellenar memoria (apartado 3)
Eros	revisión	hacer encuestas PB	rellenar 4	actualizar PB	realizar revision PB y rellenar memoria
Íñigo	prototipo	rellenar 3c	código		realizar el código del prototipo y rellenar memoria (apartado 3)
Javi	prototipo		código		realizar el código del prototipo ( a la hora de la división hara un poco más que aquellos con actividad 1)
Johan	prototipo	rellenar 3b	código		realizar el código del prototipo y rellenar memoria (apartado 3)
Kike	prototipo	rellenar 3d	código		realizar el código del prototipo y rellenar memoria (apartado 3)
Nacho	prototipo		código		realizar el código del prototipo ( a la hora de la división hara un poco más que aquellos con actividad 1)
Santi	revisión	rellenar 1	rellenar 5	actualizar PB	realizar revision PB y rellenar memoria
Shuyi	prototipo	rellenar 3a	código		realizar el código del prototipo y rellenar memoria (apartado 3)
Equipo de D.	memoria	rellenar punto 2			

Tabla 1.1 “Tabla con el reparto de trabajo”

# 2. Arquitectura

Como nuestro proyecto se trata de una aplicación móvil, concretamente preparada para ejecutarse en un Sistema Operativo Android, para el prototipo hemos utilizado la herramienta de Android Studio, para la base de datos hemos utilizado Firebase Realtime Database, un servicio de Firebase que permite a los usuarios introducir datos en una BBDD de forma sencilla. En cuanto al lenguaje en Android Studio hemos utilizado Java, ya que es un lenguaje que todos los miembros del grupo conocemos hasta cierto punto y por esto es una mejor opción que kotlin para nuestra aplicación.

Android Studio es la mejor opción para realizar el prototipo ya que nuestra aplicación final se hará en la misma aplicación. Además Android Studio permite hacer las interfaces de la aplicación de manera sencilla, esto beneficia el desarrollo de nuestra aplicación porque la mayor parte de ella se basa en vistas. Por último, es más accesible ya que todo el mundo dispone de un teléfono móvil y no dependen de tener un ordenador a mano para utilizar la aplicación.

### 3. Prototipo

Uno de los objetivos principales del equipo de desarrollo durante la práctica cinco ha sido implementar un prototipo tecnológico, es decir, un prototipo en que se puedan ver las diferentes herramientas que vamos a necesitar a lo largo del proyecto, y que nos permite probarlas para poder ir obteniendo práctica en ellas. A lo largo de este punto tres de la memoria, se explica qué herramientas hemos utilizado y en qué consisten (GitHub, Android Studio y Firebase), cómo ha sido el proceso desarrollo e implementación por parte de los desarrolladores y cuales son las instrucciones de utilización del prototipo tecnológico.

#### 3.1. Configuración del entorno de administración de la configuración

Entre todos los miembros del equipo de desarrollo hemos decidido usar GitHub como herramienta para realizar el control de versiones. La razón de elegir esta herramienta está muy clara, es una herramienta muy popular y la mayoría de los miembros del equipo la conoce, además permite la vinculación con Android Studio, otra aplicación que se usará para el desarrollo de código. También cabe decir que existe la posibilidad de administrar el proyecto con la herramienta Jira, ya que mejoraría el proceso de organización en el futuro.

Dejando explicado el porqué de la elección de esta herramienta, vamos a proceder a la explicación de su uso:

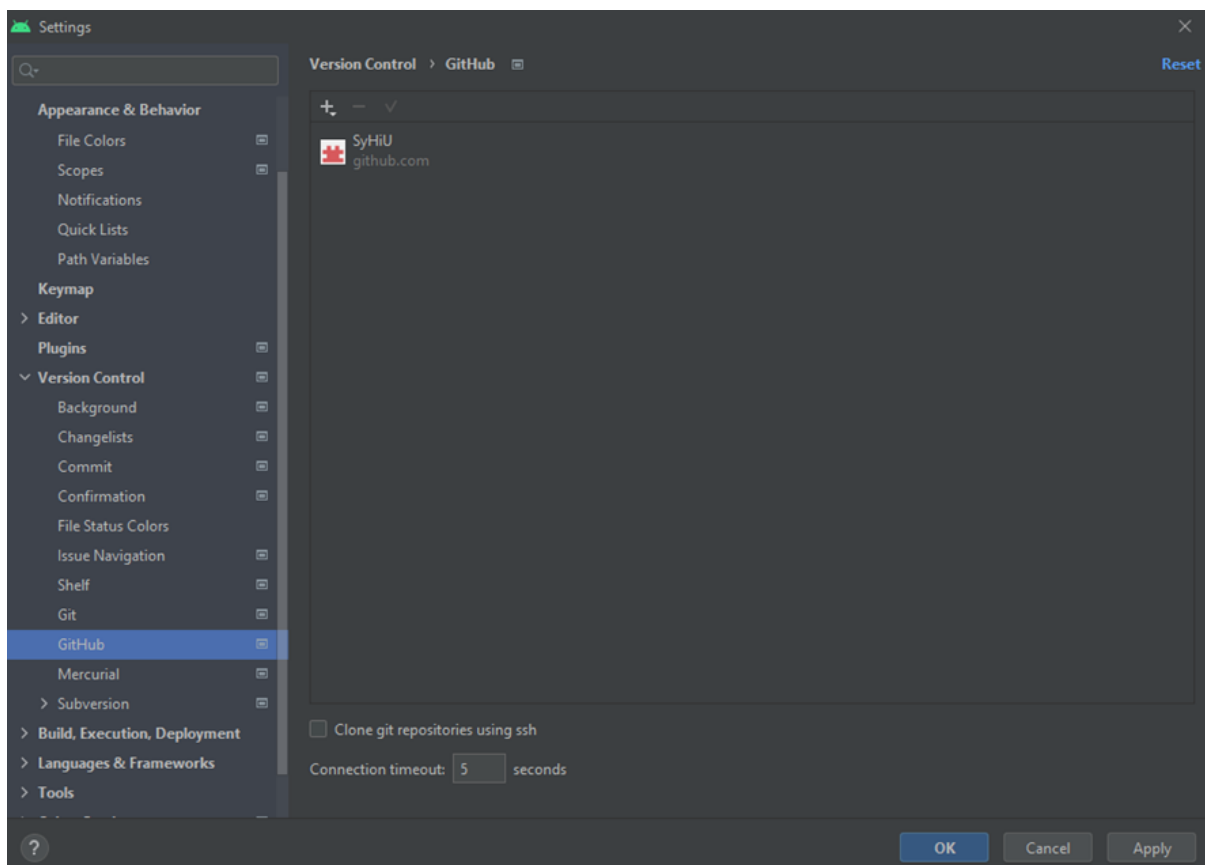
Primero, necesitamos crear una cuenta de GitHub. Para ello, entramos por el siguiente enlace [Join GitHub · GitHub](#), que es un enlace directo hacia la página de creación de cuenta. Una vez dentro, debemos seguir unos pasos sencillos para que nuestra cuenta se cree con éxito. Esto es, introducir un correo electrónico válido por donde recibiremos un código de verificación, introducir una contraseña segura y el nombre que utilizarás para identificarte.

Una vez introducido dichas informaciones, aparecerá abajo del todo el botón de crear cuenta en verde y activo. Le damos al botón y en unos instantes recibiremos en nuestro correo electrónico el mensaje de verificación de cuenta. Cabe a citar que también aparece una casilla que pregunta si deseas recibir actualizaciones, anuncios u ofertas, y también otra casilla de verificación para asegurar de que somos humanos y no somos robots. Tan solo con aceptar el mensaje de verificación, tenemos nuestra cuenta de GitHub.

Ya tenemos nuestra cuenta de GitHub, ahora necesitamos crear un repositorio. En nuestro caso solo uno de los miembros creará el repositorio e invitará al resto del equipo a colaborar en dicho repositorio.

Teniendo el repositorio creado, antes de empezar a trabajar en él, tenemos que tenerlo configurado para que se adecue a nuestro proyecto. En este caso necesitamos que sea apropiado para la implementación en Android Studio, para realizar esto es necesario tener instalado Android Studio y crear nuestro proyecto.

Hay que configurar el proyecto para que esté sincronizado con el Android Studio, para hacer esto seleccionamos en el proyecto la siguiente ruta: *Configure> Settings> Version Control > GitHub*. Nos aparecerá una ventana donde nos pedirá que introduzcamos nuestro nombre de usuario de GitHub y la contraseña. Para asegurar que estemos ya vinculados a la cuenta de GitHub, podemos presionar el botón *Test* para comprobar si los datos introducidos son correctos. Si es correcto debería mostrarnos lo siguiente , ver 3.1.1.



### 3.1.1 “Correcta configuración”

Iremos al apartado de Git y nos aseguraremos de que la ruta de la instalación es la adecuada. Podemos subir el repositorio a GitHub o importar un repositorio de GitHub, en nuestro caso vamos a importar un proyecto. Para ello solo necesitamos seleccionar el nombre del proyecto y pulsamos la opción de

*Check out project form Control Version > GitHub*, después introducimos la contraseña que pusimos en el apartado de configuración e introducimos la ruta del repositorio y el nombre. Y enhorabuena! ya tenemos sincronizado nuestro repositorio con Android Studio.

Para la base de datos hemos decidido usar la herramienta Firebase. Requiere una configuración inicial con Android Studio, y solo es necesario que una persona lo configure, porque utilizando la función *commit* del GitHub, todo los colaboradores del repositorio tendrá la misma configuración y código. Johan se encargará de la configuración e invitará a los demás a su Firebase.

### 3.2. Herramientas del prototipo

Como se menciona en el punto anterior, el prototipo se ha llevado a cabo usando Android Studio, la cuál nos aporta un gran número de facilidades para producir nuestro prototipo. Esta herramienta era nueva para la gran mayoría de nosotros, por eso tuvimos que apoyarnos en un compañero en especial para que explicara al resto cómo manejarse un poco con ella. Para ello, tuvimos una reunión en la que se encontraba todo el equipo de desarrolladores, donde se trató tanto la funcionalidad de la aplicación como el trabajo que debía hacer cada uno.

Todo el equipo estuvo muy atento a la explicación, se consiguió que todo el equipo de desarrollo fuéramos a la par siguiendo las instrucciones marcadas, de tal forma que al final de esta reunión todos pudiéramos hacer nuestra parte sin grandes dudas.

La implementación del prototipo la realizamos en el lenguaje Java, ya que es el lenguaje nativo que utiliza Android, lo que nos hizo algo más sencilla la implementación debido a que todos conocíamos ese lenguaje y solo nos íbamos a tener que familiarizar con dónde situar cada parte de la implementación, o cómo estructurarlo.

Por último, y como se menciona en el punto anterior, utilizamos GitHub para poder realizar el control de versiones y subir cada uno nuestros cambios. Al principio fue un poco complicado hacernos con esta herramienta aunque todos la conociéramos, pero tras subir al repositorio nuestros primeros commits y hacer pull para refrescar el proyecto y poder descargar lo que habían hecho los demás, nos adaptamos rápidamente a ella.

### 3.3. Desarrollo del prototipo

Para desarrollar el prototipo lo primero que se realizó fue una reunión en la que se debatió acerca de qué funcionalidades debería tener, de esta manera

tendríamos claro la funcionalidad que había que implementar y sería más fácil repartir el trabajo más tarde. En esta misma reunión se configuró el repositorio con sus correspondientes ramas para que pudiéramos desarrollar el prototipo sobre el mismo y nos aseguramos de que cada uno de los integrantes del equipo conociera las herramientas que se iban a utilizar (Android Studio y GitHub). Tras esta reunión, dado que se encontraron bastantes dificultades a la hora de delimitar las funcionalidades del prototipo (ya que queríamos estar seguros de que fueran funcionalidades que no estuvieran directamente en nuestra aplicación real pero que nos fueran útiles para que el desarrollo de la aplicación en un futuro fuera más ágil) se planteó una segunda reunión. En esta reunión se definieron las funcionalidades definitivas que queríamos implementar y, dado que muchos miembros del equipo no tenemos experiencia en la programación en Android, se hizo una explicación sobre el entorno de desarrollo, cómo funciona (cómo están organizadas las carpetas, que archivos existen y cómo se prueba) y cómo se podía subir y bajar versiones desde GitHub. Una vez finalizada esta parte, se repartieron las tres pantallas que se habían decidido crear entre el equipo de desarrollo, de tal manera que cada integrante pudiera probar distintas funcionalidades y subir sus propias versiones al repositorio correspondiente. Aunque muchos no hubiéramos usado antes Android Studio, gracias a las distintas reuniones que se realizaron explicando todo lo necesario, no hubo ningún problema en la implementación de la aplicación, siendo GitHub el entorno que más dificultades supuso (debido a que el equipo no había trabajado nunca antes con distintas ramas dentro de un repositorio). Para resolver este último problema, se realizó una reunión en la que se explicó de manera muy clara cómo había que subir el proyecto a GitHub, además de resolver los errores que habían tenido los distintos equipos al hacer “merge” entre su rama y la rama main del repositorio. Por último, para que todo el equipo de desarrollo conociera las distintas funcionalidades implementadas en el prototipo, cada grupo presentó a los demás su implementación y, además, se probó el prototipo completo.

### 3.4. Instrucciones que indiquen cómo ejecutar y probar el prototipo

En las siguientes secciones se explica cómo se debe usar la aplicación tanto si eres un usuario (solo dispones del .apk) como si eres un administrador y dispones de todas las herramientas usadas (GitHub y AndroidStudio).


#### Para desarrolladores

Para poder probar y ejecutar el prototipo tendremos que actualizar nuestros repositorios para tener la última versión sobre lo que estamos trabajando, esto lo podemos conseguir mediante la aplicación de Github. Para ello debemos


haber subido nuestro trabajo yendo a la rama de main y haciendo merge desde la nuestra, así subimos el contenido que hayamos añadido, y posteriormente vamos a la rama creada para cada uno dentro de Github y haremos merge y después push para así tener todo.

Una vez ya tenemos la última versión y todos los cambios que se hayan realizado, abriremos la aplicación Android Studio, en ella es donde anteriormente hemos trabajado todo el código necesario. Aquí podemos probar el resultado de la aplicación, para ello hay dos formas de ejecutarlo.

La primera forma de ejecutar el prototipo es en un dispositivo real, para este fin tenemos que conectar el dispositivo a la máquina de desarrollo con un cable USB, habilitar la depuración por USB en la ventana Opciones para desarrolladores (abrir app configuración, seleccionar sistema, acerca del teléfono, presionar número de compilación siete veces, volver a la pantalla anterior y presionar opciones para desarrolladores y habilitar la depuración por USB), una vez hecho esto, ejecutamos la app en nuestro dispositivo y dentro de ésta elegimos la app deseada y en la barra de herramientas seleccionamos el dispositivo de destino.

Hacer clic en **Run** , esto instala la app en el dispositivo conectado y la inicia.

La segunda opción es la de ejecutarla en un emulador. En Android Studio, creamos un dispositivo virtual de Android que el emulador pueda usar para instalar y ejecutar la app. Seleccionamos la app en la barra de herramientas desde el menú desplegable de configuraciones de ejecución y depuración. En el menú desplegable del dispositivo de destino, seleccionamos el dispositivo virtual de Android donde deseamos ejecutar la app. Para terminar, hacemos

clic en **Run** , instala la app en el dispositivo virtual y, luego, inicia el emulador.

## Para usuarios

Instalar la aplicación en un dispositivo Android mediante el .apk, una vez la tengamos instalada, tendremos que usar las distintas funcionalidades que se nos presentan en cada pantalla.

La primera pantalla permite guardar en la base de datos un nombre y una descripción asociada a este, debes registrar y el prototipo te indicará si el registro se ha completado con éxito o si ha tenido lugar algún error. Para pasar de pantalla pulsaremos continuar.

La segunda pantalla permite recuperar los datos que se han introducido en la anterior pantalla, si se introduce un dato correcto y después se pulsa el botón



de validar, se mostrará la descripción asociada al mismo y en otro caso se mostrará un mensaje de error. Esta pantalla nos permitirá desplazarnos a la anterior mediante el botón “volver hacia atrás” si queremos introducir más datos en la base de datos o ir a la siguiente pantalla para seguir probando las funcionalidades implementadas, pulsando el botón continuar

La tercera pantalla permite seleccionar checkboxes y te muestra cuales se han seleccionado o si no se ha seleccionado ninguno, además incluye un botón imagen el cual mostrará una imagen al ser pulsado (y la ocultará si se vuelve a pulsar). En esta pantalla podemos ver también un botón volver hacia atrás, que permitirá al usuario volver a la pantalla anterior.

## 4. Product Backlog

Entre los cambios fundamentales del Product Backlog destacan las nuevas historias o la eliminación de otras, un nuevo sistema de identificación, una reorganización de prioridades mediante otro método y la exportación del documento a otra herramienta.

En primer lugar, con lo aprendido en la práctica anterior hemos corregido tanto el User Story Map como el Product Backlog (PB). En consecuencia, han desaparecido varias historias. Muchas de estas historias hacían referencia a elementos que no deben reflejarse en el PB pues pertenecen al diseño de la aplicación y son cosas que no debemos plantearnos actualmente, entre ellas destacan: “ir a modo crear partida”, “Ir a modo unirse” ... También se ha modificado el sistema de creación de pantallas, ahora en la creación de un texto de teoría se le asigna una temática, luego a las preguntas se le asigna un texto y las pistas, y en la creación de pantalla se selecciona el conjunto teoría-pregunta y el título de la misma.

También hemos ampliado el alcance de algunas funcionalidades y han aparecido algunas historias como “añadir imágenes a la teoría” entre otras.

Nuestro sistema de identificación no permitía distinguir Historias de Usuario de varios proyectos y tampoco evitaba de forma eficiente que no hubiera repeticiones, es por ello que lo hemos refactorizado. Nuestra nueva forma de priorizar sigue el formato LCQNV (en referencia a las siglas de nuestro proyecto) y un número entero positivo único, solucionando así nuestros problemas de identificación.

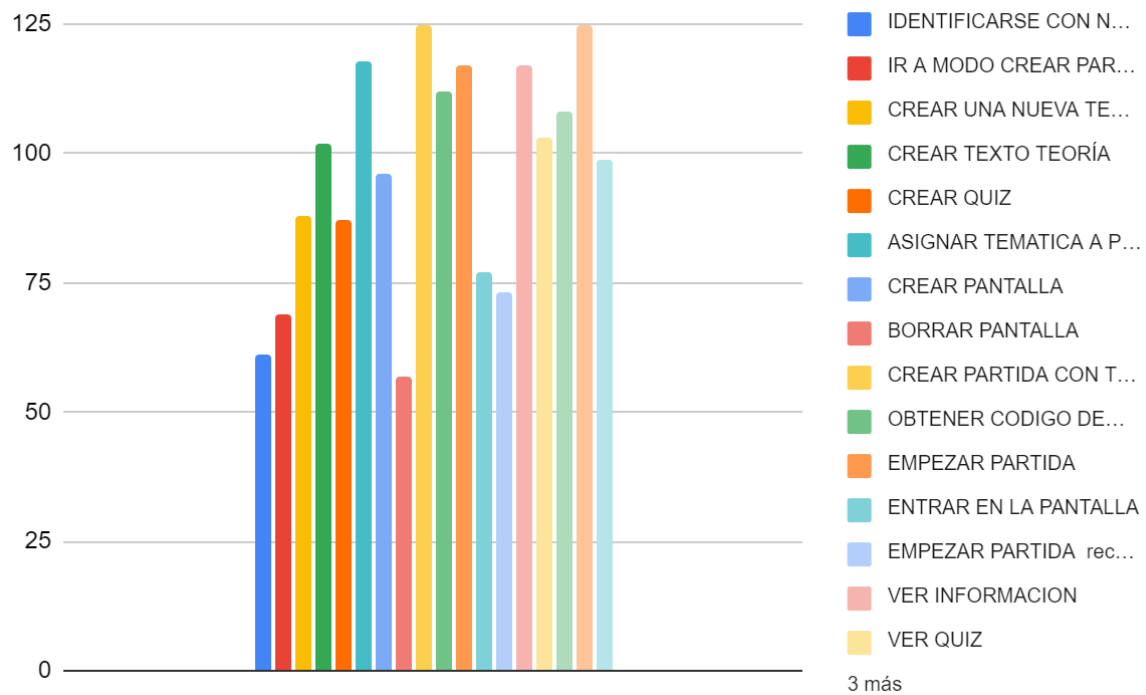
El orden de las historias en el Product Backlog representa la prioridad (ahora ya no explícita) de las mismas en orden decreciente. En esta refactorización

de prioridades hemos usado el método de los 100 puntos para romper empates entre aquellas historias que se encuentren en el mismo nivel de altura dentro de una release (en paralelo). Aquellas en un nivel de altura superior en el User Story Map son más prioritarias que aquellas en una altura inferior. Se ha usado la herramienta Google Forms para recopilar los resultados, y posteriormente una exportación a excel, tal y como muestra el recurso 4.1.1, para su procesado de forma gráfica. En el recurso 4.1.2 vemos una muestra de lo que sería la gráfica general y en el 4.13 una versión centrada solamente en una release, provocando así un gráfico mucho más manejable. Era inviable mostrar toda la información recopilada con el método de priorización en una sola hoja, por lo que se ha decidido usar muestras de los datos por releases. En consecuencia, se ha creado una nueva carpeta “Forms” en la unidad compartida, como se ve en el recurso 4.1.4, en la que las encuestas se dividen por releases.

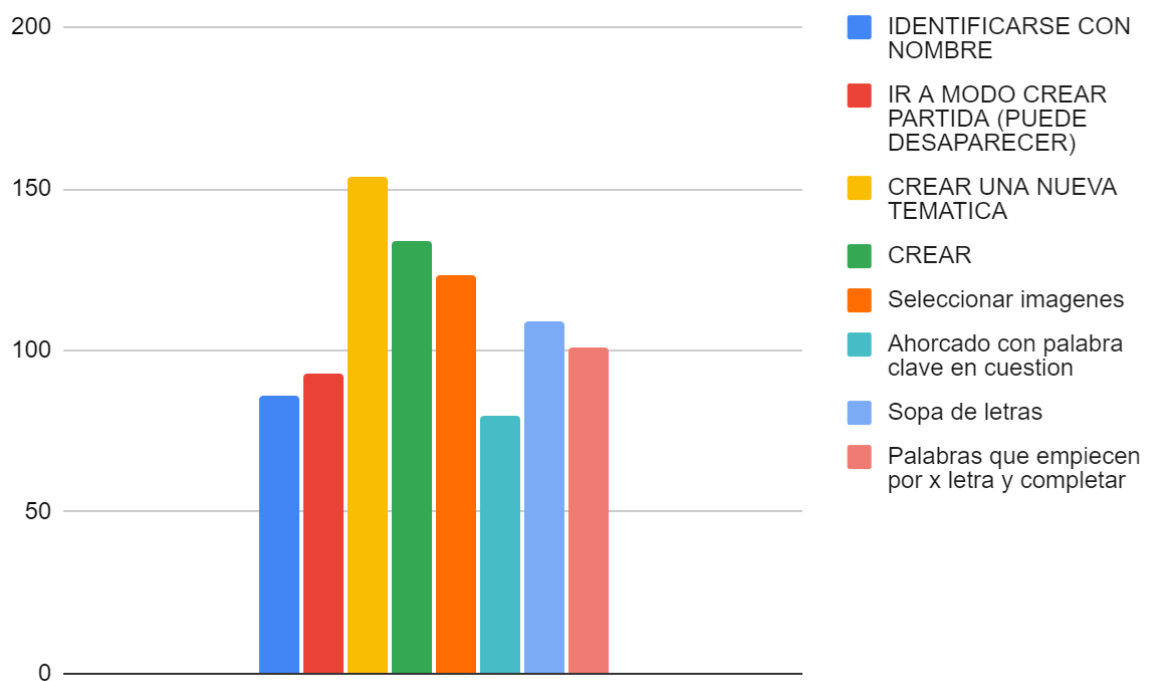
La exportación a Jira se realizó de forma satisfactoria tras un arduo trabajo. Actualmente, todas nuestras historias de usuario se encuentran almacenadas en nuestro proyecto de la página, priorizadas y con campos personalizados como “Criterios de Aceptación” que facilitan el trabajo y la lectura de las mismas. Además, se ha creado una plantilla personalizada para la exportación del Product Backlog.

Marca temporal	Dirección de correo elect	IDENTIFICARSE CON N	IR A MOD	CREAR PA	CREAR UNA NUEVA TE	CREAR	Seleccionar imagenes	Ahorcado con palabra cli	Sopa de letras	Palabras que empiecen por x letra y co	
18/02/2022 12:12:26	javig01@ucm.es	20	10	10	20	10	10	10	10	10	
18/02/2022 12:13:29	johanc01@ucm.es	0	20	10	0	30	10	20	10	10	
18/02/2022 12:17:39	sgabriel@ucm.es	10	20	30	10	5	5	15	5	5	
18/02/2022 12:17:46	anamam20@ucm.es	0	0	20	20	20	0	20	20	20	
18/02/2022 12:21:02	iville01@ucm.es	11	3	25	20	6	12	2	1	1	
18/02/2022 12:23:04	davichap@ucm.es	10	10	20	20	10	5	5	20	20	
18/02/2022 12:27:14	irolando@ucm.es	15	10	15	10	10	15	15	10	10	
18/02/2022 12:49:01	shuyiwan@ucm.es	20	15	10	10	20	10	10	5	5	
18/02/2022 13:53:01	lbarrero@ucm.es	0	5	14	24	12	13	12	20	20	
		86	93	154	134	123	80	109	101	101	
	Marca temporal	Dirección de correo elect	IDENTIFICARSE CON N	IR A MOD	CREAR PA	CREAR UNA NUEVA TE	CREAR	Seleccionar imagenes	Ahorcado con palabra cli	Sopa de letras	Palabras que
			86	93	154	134	123	80	109		

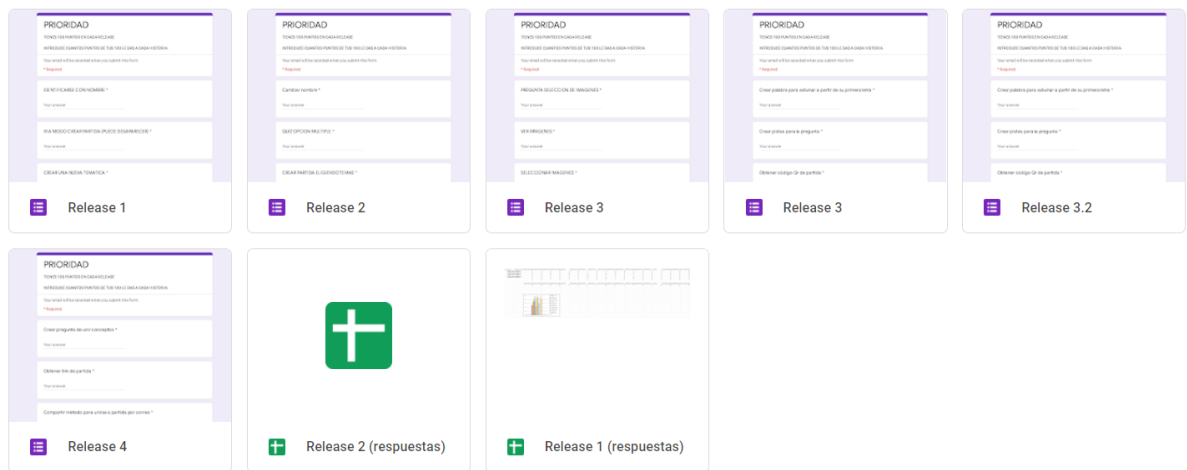
#### 4.1.1 “Resultados de la encuesta más suma”



#### 4.1.2 “Fragmento gráfica general”



#### 4.1.3 “Gráfica por release”



4.1.4 “Carpeta Forms de la unidad compartida”

## 5. Actividad grupal

Para esta práctica, decidimos hacer una actividad que nos permitiese aprender a trabajar con los distintos miembros del equipo, familiarizarse con Android y aprender a dar una explicación sencilla y clara de lo que se va haciendo.

### 5.1. Descripción

Para su realización, el PO y el SM idearon unos equipos al azar y una historia para cada equipo. Estas se repartieron al azar también y durante 15 minutos cada equipo tenía que implementar lo que se les pedía a la vez que explicaban lo que iban haciendo. De esa manera, pasado el tiempo, se repartieron otros equipos nuevos las mismas historias ya empezadas y las continuaron. Mientras tanto, el PO y el SM iban haciendo un seguimiento de esta actividad.

### 5.2. Desarrollo

El grupo inició sesión en “Kumo Space” y el PO y el SM explicaron la actividad. El Scrum Master dividió al resto del equipo en distintos grupos, dos de ellos formados por dos personas y un último por tres y el Product Owner les dio a cada uno una historia de usuario.

Durante 15 minutos, trabajaron cada grupo en su historia, buscando información y rellenando un documento con lo que hacían. Pasado ese tiempo, el SM rehizo los equipos poniendo en cada uno a gente que no había estado anteriormente trabajando en grupo y se volvieron a repartir las historias (se hizo lo posible por no repartir la historia anterior a un miembro que trabajó en ella, pero al ser solo tres equipos fue imposible, y dos personas volvieron a una historia que ya habían trabajado). Cada grupo tuvo que leer lo escrito por el anterior en su tarea y avanzar en la historia. Esto se repitió una última vez, y

pasados 45 minutos, todo el equipo volvió a reunirse a ver los resultados. (Link con el video de la actividad: <https://youtu.be/rqC1itO2jFk>).

### 5.3. Resultados

Los resultados fueron mejor de lo esperado, las anteriores actividades nos habían ayudado a conocernos mejor, por lo que el trabajo entre los distintos miembros no fue difícil.

El mayor resultado que se obtuvo fue la capacidad de realizar una pequeña síntesis clara y limpia para explicar a nuestros compañeros el trabajo anterior y que ellos puedan continuar sin problema desde donde se dejó.