

# Plan de Proyecto

---

## LOGROLLING

Alberto Almagro  
Rubén Gómez  
Juan Carlos Llamas  
Jaime Martínez

Santiago Mourenza  
Pedro Palacios  
Adrián Sanjuán  
Pablo Torre





# Índice

1. Introducción.....	3
1.0 ¿Qué es Logrolling?.....	3
1.1 Propósito del plan .....	3
1.2 Ámbito del proyecto .....	4
1.3 Modelo de proceso .....	7
2. Estimación.....	8
2.1 Datos históricos .....	9
2.2 Técnicas de estimación.....	9
2.3 Estimación de esfuerzo, coste y duración.....	9
2.4 Ingresos.....	12
3. Gestión de riesgos .....	14
3.1 Introducción: Estudio de los riesgos .....	14
3.2 Priorización de riesgos del proyecto.....	16
3.3 Plan de gestión de riesgos: Reducción, supervisión, y plan de contingencia.....	17
3.4 Planificación temporal del control de riesgos .....	21
3.5 Resumen .....	21
4. Planificación temporal .....	23
4.1 Gráfico Gantt .....	23
4.2 Estructura de descomposición de trabajo / Planificación temporal .....	23
4.3 Red de tareas: .....	30
4.4 Tabla de uso de Recursos:.....	30
5. Organización del personal (gestión del equipo) .....	31
5.1 Estructura del equipo .....	31
6. Recursos del proyecto.....	33
6.1 Personal .....	33
6.2 Hardware.....	33
6.3 Software .....	34
7. Plan de GCS / Mecanismos de Gestión y Control .....	36
7.1 Introducción.....	36
7.2 Gestión de la configuración.....	37
7.3 Programa de la Gestión de Configuración .....	42
7.4 Control de la Configuración.....	46
7.5 Políticas, directivas y procedimientos de modificación .....	51
7.6 Informes y auditorías.....	53
7.7 Control de versiones .....	55
8. Referencias .....	57
9. Revisiones.....	58



# 1. Introducción

## 1.0 ¿Qué es Logrolling?

**Logrolling** es un concepto nacido en Norteamérica que se define como “**La acción de intercambiar favores**”, sobre todo aplicado en un contexto político.

---

*Nosotros queremos extrapolar la idea a la ayuda que pueden ofrecer diferentes integrantes de una misma comunidad, creando una sociedad entre usuarios de forma que todos se vean beneficiados.*

---

## 1.1 Propósito del plan

---

*Logrolling pondrá en contacto a una comunidad de usuarios con el objetivo de que estos se ayuden mutuamente. Una medida contra el estrés y el exceso de tareas que afecta a estudiantes, familias y otros grupos sociales.*

---

Vamos a crear una aplicación que acerque a las personas ayudándose mutuamente en la medida del tiempo del que dispone cada individuo. Cada vez más españoles sufren ansiedad, y el estrés constituye una de las principales causas de bajas laborales en España, el país europeo con más estrés en el entorno de trabajo. Según un estudio de CINFA publicado en Cadena SER (Ballesteros, 2017), **la principal causa del estrés es el exceso de actividades y la falta de tiempo**, por lo que no nos sorprende que **quienes más sufren el trastorno del estrés sean las parejas con hijos y los estudiantes, entre otros grupos**.

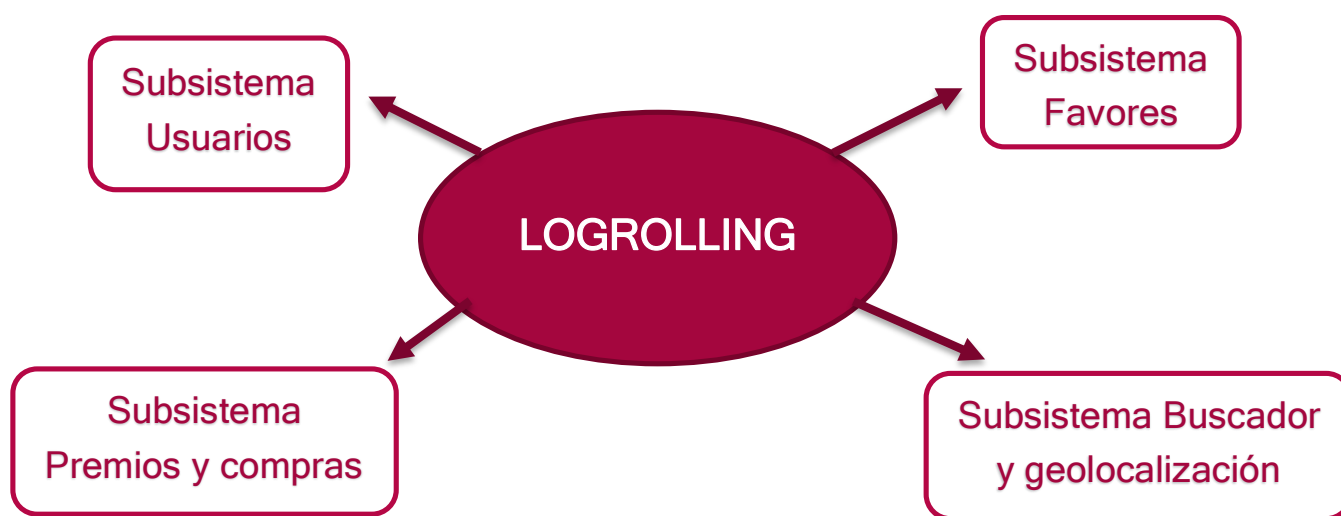
Con el objetivo de solucionar este problema vamos a desarrollar una aplicación que permita poner en contacto a diferentes integrantes de una comunidad con la intención de que **cada usuario pueda ofrecer su tiempo y sus habilidades, así como demandar las de otro, en forma de favores**. Consideramos que nuestro proyecto no solo ayuda a solucionar este **problema social**, sino que contribuye a crear un **sentimiento de comunidad** entre los usuarios que utilicen la aplicación. Entre el tipo de favores que incluirá la aplicación se encuentra la realización de recados, transporte de individuos, favores particulares (ej: cuidado de mascota) y muchos otros. Una característica importante que tendrá la aplicación es la posibilidad de que **empresas y negocios puedan generar favores realizables por varios usuarios**, lo que garantiza que siempre haya favores disponibles para los usuarios interesados y evitando posibles escenarios en el que nadie pueda solicitarlos.



## 1.2 Ámbito del proyecto

*Destinado a un público genérico, a través de la aplicación se podrán obtener beneficios por la realización de favores solicitados por otros usuarios.*

Nuestro proyecto está **orientado a la obtención de diferentes recompensas y beneficios por la realización de distintos favores** publicados por otros usuarios. Por ello **la aplicación está destinada a un público general**, en especial aquellas personas que puedan ofrecer parte de su tiempo para ayudar a otros individuos, y a aquellas necesitadas de tiempo por el exceso de actividades en su día a día. Teniendo en cuenta el público objetivo y la intencionalidad de la aplicación hemos diferenciado cuatro módulos o subsistemas:





## 1.2.1 Funciones principales

Las principales operaciones realizables en nuestra aplicación, que explican a grandes rasgos su funcionamiento, son las siguientes:

*Un concepto clave necesario para entender la mayoría de funcionalidades de la aplicación son los **grollies**, moneda virtual interna de la aplicación.*

Módulo	Función	Descripción
Usuarios y empresas	Registro	En un primer instante, se le dará la opción al usuario de registrarse en la aplicación como empresa o como usuario.
	Log-in y log-out	El usuario podrá iniciar y cerrar sesión cuando quiera (aunque esta se mantendrá abierta por defecto)
	Modificar perfil	Aspectos concretos del perfil del usuario (e-mail, foto de perfil, contraseña...) podrán ser modificados en cualquier momento.
Favores	Publicar un favor/favor múltiple	Un usuario o una empresa podrá publicar un favor (una necesidad) ofreciendo una retribución (en <i>grollies</i> ) estandarizada con el objetivo de que otro usuario lo realice.
	Seleccionar y realizar favor /favor múltiple	Un usuario puede adjudicarse un favor con la motivación de obtener una retribución (en <i>grollies</i> ) tras su realización.
	Chat	Los usuarios se podrán comunicar a través de un chat privado para concretar detalles del favor a realizar y la forma.
Premios y compras	Compra de <i>grollies</i>	Los usuarios podrán pagar en la aplicación para recibir <i>grollies</i> a cambio.
	Solicitar premios	Si el usuario tiene suficientes <i>grollies</i> podrá intercambiarlas por selectos premios que la aplicación suministrará a través de Amazon.
Buscador y geolocalización	Buscar favores por filtro	Un usuario puede buscar favores filtrando por diferentes características como su nombre, localización, etc.



## 1.2.2 Aspectos de rendimiento

La aplicación se basa en el almacenamiento de los datos de usuarios y en las operaciones y conexiones que se llevan a cabo entre estos. Tenemos la intención de que nuestra app empiece en pequeñas zonas locales, pero con posibilidad de expansión. Por lo tanto, **el público objetivo es muy amplio**, por lo que **la base de datos deberá ser capaz de almacenar la información de una cantidad superior a los 10.000 usuarios ya desde el comienzo**. El servidor deberá estar adaptado de forma que el registro de varios usuarios o la conexión simultánea de varios terminales no aumente el tiempo de respuesta de la aplicación y no la colapse.

Relacionado con el tema de pagos, destaca **el aspecto de la seguridad en las compras** de *grollies* a cambio de dinero, por lo que la aplicación deberá tener un sistema de cifrado adecuado que evite posibles pérdidas de datos y permita realizar las transacciones de manera segura.

Por último, uno de los principales aspectos a tener en cuenta es que **la aplicación depende de la conexión a la red individual de cada usuario** y la conexión de estos a los servidores de la aplicación.

## 1.2.3 Restricciones y técnicas de gestión

**Restricciones legales:** Debido a normativas de la Unión Europea no podemos crear una moneda virtual intercambiable por dinero. Para evitar este impedimento, **la aplicación incluye un sistema de recompensas** de forma que cuando quieras materializar tus *grollies*, puedas intercambiar estos por diferentes premios. También hay que tener en cuenta el aspecto legal de las transacciones monetarias que hay que llevar a cabo según indica la directiva europea de servicios de pago PSD 2 (UE, 2015).

**Restricciones técnicas:** Una de las restricciones técnicas más importantes es que trabajamos con **transacciones monetarias** y por tanto hay que tener en cuenta varios estándares. Además, por esta razón, el usuario tiene que estar dispuesto a utilizar la aplicación con este objetivo. Por otro lado, nuestra aplicación está **destinada a usuarios Android**, aunque no descartamos una versión iOS y otra de escritorio para el futuro.

**Técnicas de gestión:** Hemos optado por un **modelo descentralizado democrático** en un comienzo, pensando **evolucionar a un modelo descentralizado controlado** tras un periodo de prueba de alrededor de tres semanas, tras el cual elegimos al responsable de organizar el grupo, pero siempre al mismo nivel jerárquico que los demás integrantes del grupo. Sin embargo, dado el excelente rendimiento del equipo con el primer modelo, se ha decidido permanecer siguiendo un modelo descentralizado democrático.





## 1.3 Modelo de proceso

*Consideramos que los métodos ágiles, en especial la metodología de **Lean Software Development**, son los más aptos para que el proyecto que queremos desarrollar resulte de la mayor utilidad y mejor calidad posible.*

Se ha decidido llevar a cabo el proyecto siguiendo la metodología **Lean de desarrollo de software** incorporando algún aspecto de otros métodos ágiles, como **la figura del product owner del método scrum**.

El motivo de adoptar esta metodología es el de **realizar varias iteraciones con el objetivo de hacer un trabajo continuo que sea revisado con frecuencia, y el de trabajar eficientemente** como plantean los principios Lean de desarrollo de software:

1. Eliminar los desperdicios
2. Amplificar el aprendizaje
3. Decidir lo más tarde posible
4. Entregar tan rápido como sea posible
5. Capacitar al equipo
6. Construir integridad intrínseca
7. Visión de todo el conjunto

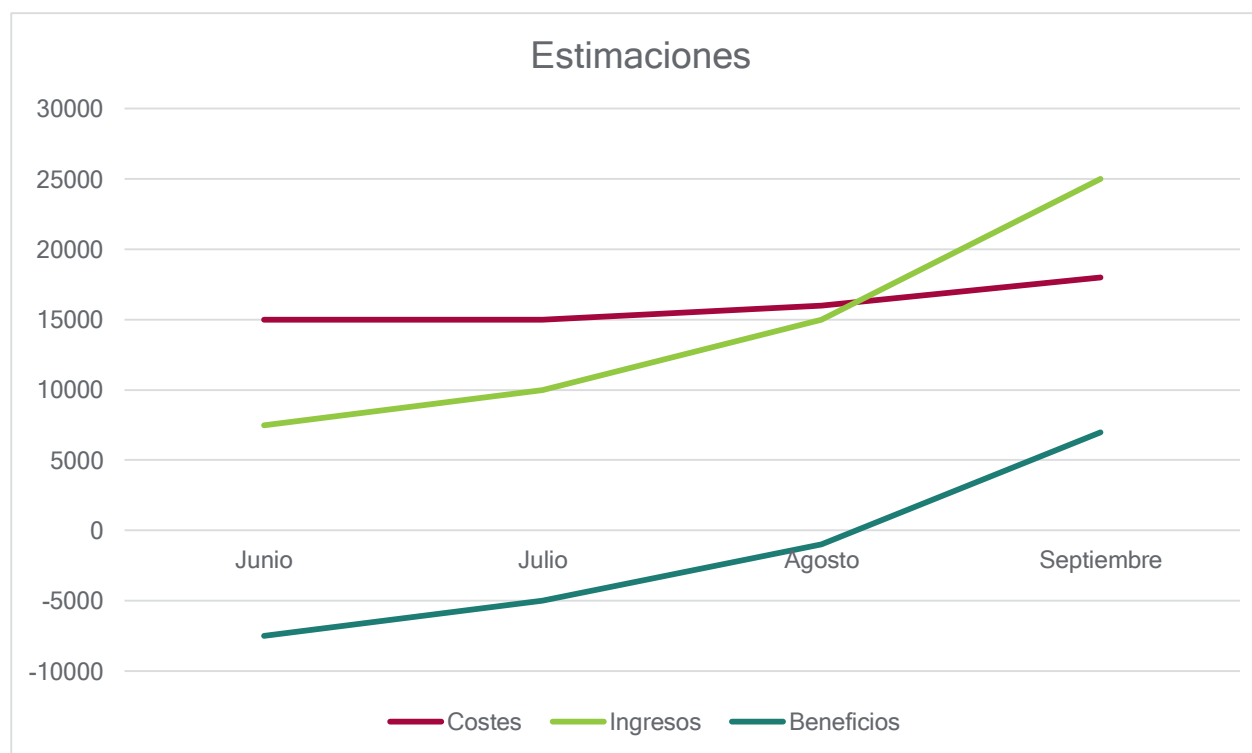
Es importante la figura del **product owner** en nuestra metodología, ya que sustituir al cliente por un integrante del equipo **nos ayuda a especificar requisitos y funcionalidades de la aplicación**, así como a **obtener feedback constante**, lo que se refleja en el gran número de reuniones que se llevan a cabo.



## 2. Estimación

*El proyecto tiene un coste total estimado de 24.000 € y se prevé poder recuperar la inversión inicial antes de finales del año 2020 (con lanzamiento inicial en mayo), pues se calcula que en septiembre ya se obtendrán beneficios de aproximadamente 7.000€ en tan solo cuatro meses tras su lanzamiento.*

Estimaciones	Costes	Ingresos	Beneficios
Junio	15000 (-)	7500 (-)	-7500 (-)
Julio	15000 (-)	10000 (↑)	-5000 (↑)
Agosto	16000 (↑)	15000 (↑)	-1000 (↑)
Septiembre	18000 (↑)	25000 (↑)	7000 (↑)







## 2.1 Datos históricos

Todas las estimaciones se han realizado ayudadas por el gran éxito de la aplicación *Wallapop*. Esto ha sido especialmente relevante a la hora de considerar el impacto que tendrá *Logrolling* sobre la población, pero también ha influido en la determinación de ingresos y, en menor medida, en la de gastos, debido a las similitudes entre ambas aplicaciones.

## 2.2 Técnicas de estimación

Se ha decidido llevar a cabo la técnica de descomposición basada en el proceso, que por tanto estará enteramente basada en la planificación temporal del proyecto, realizada previamente y que se encuentra en la sección 4.

## 2.3 Estimación de esfuerzo, coste y duración

Todos los módulos en los que se descompone el proyecto se han organizado y planificado para desarrollarse en el mismo periodo de tiempo, por lo que los cálculos del esfuerzo y el coste se realizarán del proyecto en su conjunto más que de cada módulo por separado. Dado que se realizarán varias iteraciones y evaluaciones con el cliente, se tendrán en cuenta los costes a lo largo de todo el desarrollo.

### 2.3.1 Estimación de esfuerzo

El esfuerzo necesario por parte de los trabajadores constituye una de las partes más importantes y fundamentales del desarrollo de cualquier proyecto que use las técnicas de ingeniería del software.

Dado que el proyecto se realiza de forma modular, se estimarán de manera separada el esfuerzo de cada uno los módulos. Esto queda recogido en la siguiente tabla:



AE	Com. Cliente	Plan	Análisis riesgos	Ingeniería		Construcción y adaptación		Ev. Cliente	Total
Módulo				Análisis	Diseño	Codif.	Pruebas		
Usuarios	-	-	-	0,25	1	1,25	0,5	-	3
Favores	-	-	-	0,25	0,5	1,5	0,5	-	2,75
Compras y regalos	-	-	-	0,25	0,25	0,5	0,25	-	1,25
Buscador y geolocalización	-	-	-	1	0,25	2	0,25	-	3,5
Estimación total	0	2,5	0,25	1,75	2	5,25	1,5	1	14,25
%Esfuerzo	0	17,54	1,75	12,28	14,04	36,84	10,53	7,02	100

De esta manera, el esfuerzo total resulta ser **14,25 PM** distribuido a lo largo de 38 semanas.

### 2.3.2 Estimación de coste

Los costes estimados son los siguientes:

- **Sueldos y salarios:** Varios estudios (Indeed.es, 2019) estiman que el sueldo medio de un programador junior en España ronda los 19200 € brutos al año, lo que se traduce en unos **1.600 € / PM**. En total, dado que el equipo consta de 8 integrantes, se necesitarán **12.800 € / mes**. Esta decisión se ve motivada por el hecho de que no buscamos los mejores programadores para nuestro proyecto, sino, la mejor combinación entre algo que nos salga a cuenta en un primer momento y que además cubra las necesidades que buscamos.

- **Alquiler de local:** Se ha planeado buscar oficinas alrededor de las zonas de Moncloa - Argüelles (en Madrid), esta ubicación se ve favorecida por la decisión de lanzar el producto en la zona de Ciudad Universitaria y se encuentra a una distancia razonable de todos los integrantes del grupo. Dado que el equipo solamente está formado por 8 miembros, bastará una pequeña oficina para los meses antes del lanzamiento de la aplicación. Búsquedas en páginas web (idealista.com, 2019) (belbex.com, 2019) (fotocasa.es, 2019) revelan que se pueden encontrar espacios para oficinas cuyo coste por alquiler sea de unos **400 € / mes**. Este precio incluye los gastos por agua, calefacción, limpieza y wifi, entre otros, que en ocasiones no vienen incluidos en los precios de los anuncios.

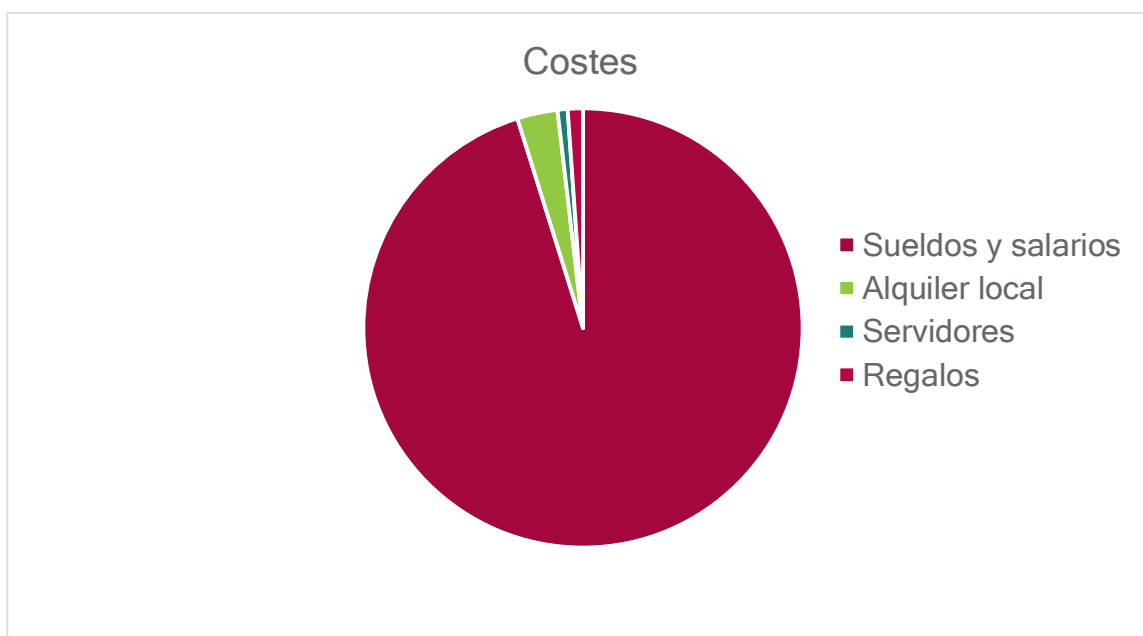


- **Servidores:** Dado que toda la aplicación es online y cuenta con cuentas de usuarios, es imprescindible una base de datos en la que guardar toda la información que vaya necesitando la aplicación. Sin embargo, esta también es necesaria desde antes de lanzar la aplicación, y especialmente durante el periodo de pruebas, pues se tendrá que asegurar que la aplicación funciona se comunica correctamente con la base de datos que se usará. La base de datos más básica, con 2 usuarios de escritorio remotos, 50GB de memoria SSD, 2GB de memoria RAM, 1 CPU y transferencia ilimitada de archivos (esto último es un factor clave para realizar las pruebas, pues muchas empresas de alquiler de servidores tienen un límite de transferencia de archivos) cuesta **100 € / mes** (clouding.com, 2019). Una vez lanzada la aplicación, es probable que se necesiten más servidores con más capacidad y potencia.

- Dado que la aplicación contará con **un sistema de regalos**, es importante estimar los gastos que se tendrán debidos a este sistema. Poco después del lanzamiento inicial, se considera que no habrá mucha gente que llegue a tener una cantidad suficiente de grollies como para canjearlos por regalos. Dado que empezaremos en Ciudad Universitaria, se estima que dedicar 1.200 € (lo que supone una media de **150 € / mes**) en el momento de lanzar la aplicación es una cantidad suficiente para los inicios de la aplicación.

En total los gastos serán **13.450 € / mes**, lo que equivale a unos **1.681,25 € / PM**.

Los costes del proyecto se pueden visualizar en el siguiente gráfico:



Por lo tanto, el coste final del proyecto es de:

$$14,25 \text{ PM} * 1.681,25 \frac{\text{€}}{\text{PM}} = 23.958 \text{ €} \approx 24.000 \text{ €}$$



## 2.4 Ingresos

La aplicación está pensada para lanzarse a finales de mayo o principios de junio de 2020. Este será un muy buen momento, pues muchos universitarios estarán muy ocupados con los exámenes finales de sus carreras y necesitarán hacer uso de la aplicación para realizar tareas cotidianas. Por lo tanto, se estima tener una buena acogida en este periodo inicial. Las estadísticas (Metro de Madrid, 2019) revelan que en el año 2018 alrededor de **15.000.000 de personas** cruzaron la boca de metro de Ciudad Universitaria. Se estima que, en el primer mes y con la propaganda adecuada, se podría llegar a las **10.000 descargas** de la aplicación.

Dado que la mayoría de la gente se instalará la aplicación con el fin de solicitar favores y que, si no se quiere realizar ningún favor, la cantidad de grollies inicial es inferior a la necesaria para pedir uno, muchos optarán por ver anuncios en los que puedan obtener fácilmente algunos. Como cada anuncio de

Google aporta una media de 0,02 € / clic (El Androide Libre, 2015), se estima que se llegarán a unos 50.000 clics en un mes, lo que se traduce en **1.000 € en el primer mes**.

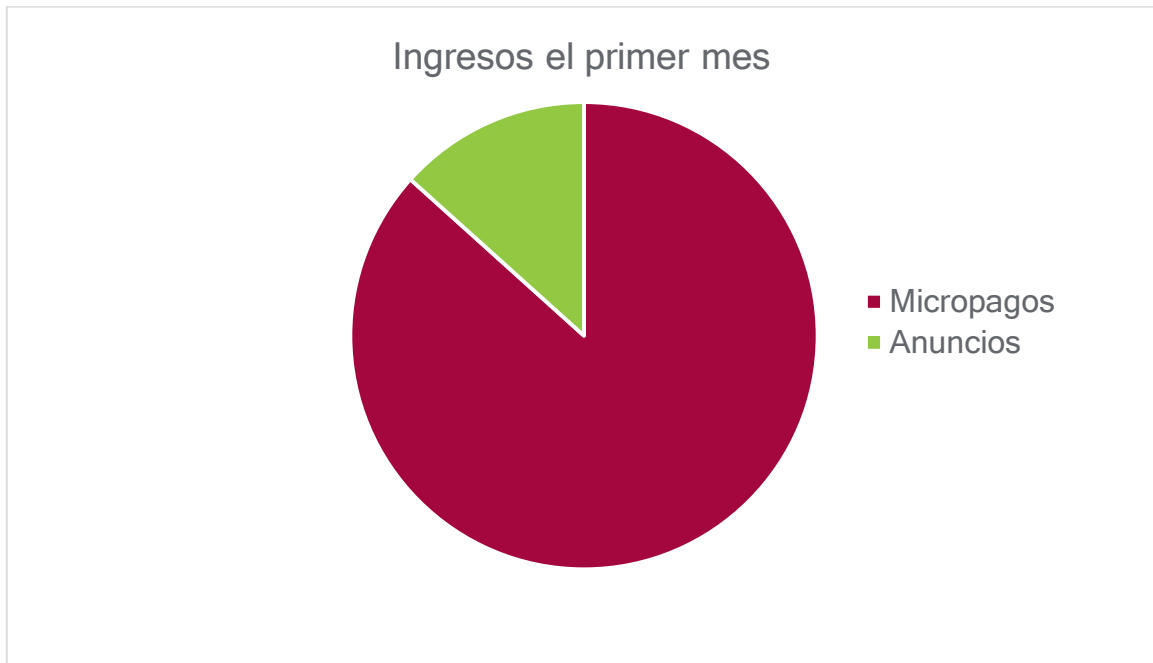
Además, la aplicación incorpora un sistema de micropagos, que sin duda será utilizado por muchos usuarios no dispuestos a ver anuncios. Una estimación razonable para el primer mes es que un 10% de los usuarios realizarán al menos una compra de 1 €, que es la mínima cantidad que se ofrece en la tienda. Sin embargo, habrá alrededor de un 5% que realizará una compra superior a 10 € a lo largo de todo el mes y un 1% que gastará más de 20 € (Pastor, 2017). Todas estas microtransacciones resultarán en unos ingresos de **6.500 € en el primer mes**.

Así pues, se calculan unos **ingresos totales de 7.500 € en el primer mes**. Sin embargo, una vez los usuarios compartan la aplicación y más usuarios nuevos se unan y basándonos en el crecimiento de Wallapop (Naveira, 2017), se estiman un total de 50.000 descargas para finales de septiembre, lo que multiplicará los ingresos obtenidos en cada mes, llegando así a unos **25.000 € / mes** en un caso relativamente poco rentable, momento en el cual la aplicación ya estará produciendo beneficios gracias al comienzo del curso universitario.

Además, una vez la aplicación crezca, será el momento en el que el equipo contactará con empresas que estén interesadas en obtener feedback a gran escala para que creen cuentas de empresa en la aplicación, para las que será necesario pagar para anunciar favores globales, lo que aumentará de forma considerable los beneficios. Sin embargo, no se estima que esta opción vaya a tener un gran éxito durante los primeros meses, por lo que no se ha considerado a la hora de estimar los ingresos.



Los ingresos estimados para el primer mes tras el lanzamiento de la aplicación pueden observarse en el siguiente gráfico:





## 3. Gestión de riesgos

*En esta sección reconoceremos los principales riesgos en el desarrollo de nuestra aplicación y haremos un plan para prevenirlos, supervisar la prevención y, en caso de que se hagan reales, crearemos un plan de contingencia para que su impacto sea mínimo.*

### 3.1 Introducción: Estudio de los riesgos

En esta sección trataremos de **identificar** adecuadamente los **riesgos** que puedan surgir durante el desarrollo del proyecto para así poder **prevenirlos** y, en caso de que se hagan reales, tener un **plan de contingencia** para que su impacto sea el mínimo.

Estos son algunos de los diversos riesgos a los que nos podemos enfrentar:

*Clasificación de riesgos:*

Nombre del riesgo	Explicación
Mala acogida (no alcanzar la masa crítica)	Para el desarrollo fluido de los intercambios de favores. Es decir, que la demanda y la oferta se mantengan activos, debemos alcanzar un número mínimo de personas que descarguen la aplicación. Es lo que se denomina masa crítica y que, en caso de no alcanzarse, presenta el intercambio de favores entre usuarios como inviable.
Mala estimación del tiempo del proyecto	Dada la magnitud del proyecto es posible que las estimaciones en cuanto a plazos no sean tan precisas como se requiere. Esto puede suponer retrasos que desagraden al cliente o impriman mucha presión y carga de trabajo a los trabajadores.



<b>Mala gestión de situaciones imprevistas</b>	Debido a la falta de experiencia de algunos trabajadores, es posible que la manera de actuar frente a situaciones imprevistas no sea la más correcta o la que de mejores resultados.
<b>Baja eficiencia de los trabajadores (no alcanzar los plazos)</b>	Debido a la falta experiencia de algunos trabajadores es posible que su esfuerzo y trabajo no se vea luego reflejado en resultados tangibles y que, en consecuencia, no se alcancen los plazos preestablecidos.
<b>Falta de comunicación entre trabajadores</b>	Al regirnos por un sistema de descentralizado democrático el consenso y comunicación entre trabajadores se puede hacer difícil y llegar a repercutir negativamente en el transcurso del proyecto,
<b>Cambio de requisitos del cliente</b>	Al depender de un cliente que financia el proyecto, también estamos ligados a sus exigencias en cuanto a requisitos, que es posible que cambien a lo largo del proyecto.
<b>Problemas legales con la moneda virtual y los premios</b>	Como estamos creando algo equivalente a una moneda virtual en nuestra aplicación pueden surgir ciertos problemas en cuanto a la gestión y al cambio que podamos realizar entre euros y nuestra moneda o nuestra moneda y premios.
<b>Falta de financiación</b>	Todo proyecto está ligado a su financiación, por eso es un riesgo a tener en cuenta.
<b>El cliente cancela el proyecto</b>	Ya sea porque no le guste como trabajamos o porque se haya quedado sin dinero, corremos el riesgo de que el cliente cancele el proyecto
<b>Algún integrante deja el proyecto</b>	Trabajamos en un grupo de 8 personas y todas las tareas están bien repartidas. Siempre es un riesgo que uno de los integrantes del equipo abandonase el proyecto.
<b>Mala estimación del coste del proyecto</b>	Al igual que con el tiempo, es posible que las estimaciones en cuanto a coste no sean tan precisas como se requiere. Esto puede suponer muchos problemas en el desarrollo del proyecto.
<b>Sacar aplicación con bugs</b>	En toda aplicación es inminente el riesgo de que aparezca un bug.
<b>Pérdida y divulgación de datos</b>	Nuestra aplicación maneja muchos datos personales de nuestros usuarios: ubicación, identificación... Es posible que nos encontremos con una pérdida o divulgación de datos.





## 3.2 Priorización de riesgos del proyecto

Vamos a clasificar la prioridad de los riesgos según el **criterio SQAS-SEI**. Este modelo establece **4 niveles de prioridad** en función de la probabilidad de que tenga lugar el riesgo y del impacto que tiene la ocurrencia del riesgo en el desarrollo del proyecto.

RIESGO	FRECUENCIA	IMPACTO	PRIORIDAD
Mala acogida (no alcanzar la masa crítica)	Ocasional	Catastrófico	Intolerable
Mala estimación del tiempo del proyecto	Ocasional	Serio	Media
Mala gestión de situaciones imprevistas	Ocasional	Serio	Media
Baja eficiencia de los trabajadores (no alcanzar los plazos)	Probable	Serio	Alta
Falta de comunicación entre trabajadores	Remoto	Crítico	Media
Cambio de requisitos del cliente	Frecuente	Menor	Media
Problemas legales con la moneda virtual y los premios	Ocasional	Menor	Baja
Falta de financiación	Ocasional	Crítico	Alta
El cliente cancela el proyecto	Improbable	Catastrófico	Media
Algún integrante deja el proyecto	Remoto	Crítico	Media
Mala estimación del coste del proyecto	Probable	Serio	Alta
Sacar aplicación con bugs	Probable	Serio	Alta
Pérdida y divulgación de datos	Improbable	Catastrófico	Media



### 3.3 Plan de gestión de riesgos: Reducción, supervisión, y plan de contingencia

A continuación, se muestran en las siguientes tablas los **planes de gestión de los riesgos** establecidos anteriormente, indicando las acciones necesarias para **prevenirlos**, **supervisarlos**, e indicando las **acciones de contingencia** si ocurriese el riesgo.

#### Nivel de prioridad 1:

RIESGO	Mala acogida (no alcanzar la masa crítica)
PREVENCIÓN	Realizar un estudio de mercado para saber cómo orientar la aplicación para que sea del agrado de los potenciales clientes
SUPERVISIÓN	Introducir la figura de un “usuario” dentro de nuestro equipo que se ponga en el papel de los futuros usuarios para guiarnos hacia los gustos del público
PLAN DE CONTINGENCIA	Recoger las críticas que recibe la aplicación y trabajar sobre ellas para sacar una nueva versión mejorada acompañada de otra campaña publicitaria más ambiciosa

#### Nivel de prioridad 2:

RIESGO	Baja eficiencia de los trabajadores (no alcanzar los plazos)
PREVENCIÓN	Incluir un sistema de incentivos económicos para premiar la productividad de los trabajadores.
SUPERVISIÓN	Control de entrega de todos los trabajadores y actualización progresiva de las cargas de trabajo adaptándolas a la productividad real de los trabajadores.



<b>PLAN DE CONTINGENCIA</b>	Sustituir a los trabajadores que no alcancen un mínimo exigible de productividad por personas de vasta experiencia laboral que adopten el rol de “veterano”, es decir, que guíe a los trabajadores más novatos a una manera de hacer el trabajo más práctica, rápida y eficiente.
<b>RIESGO</b>	<b>Falta de financiación</b>
<b>PREVENCIÓN</b>	Para prevenir este problema, hemos pensado en organizar nuestro proyecto con un margen de financiación por si el dinero que se nos ha ofrecido en un principio no es el que recibimos luego. Además, es importante mantener el contacto con el cliente por si hay algún cambio en la financiación del proyecto.
<b>SUPERVISIÓN</b>	Hablar con el cliente con frecuencia acerca del tema
<b>PLAN DE CONTINGENCIA</b>	En el caso que nos encontremos una falta de financiación en el proyecto deberemos tratar de simplificarlo o incluso llegar a pedir un préstamo
<b>RIESGO</b>	<b>Mala estimación del coste del proyecto</b>
<b>PREVENCIÓN</b>	Analizar proyectos similares y estimar por lo alto
<b>SUPERVISIÓN</b>	Actualizar periódicamente las estimaciones
<b>PLAN DE CONTINGENCIA</b>	Pedir más presupuesto o simplificar el proyecto y reducir gastos
<b>RIESGO</b>	<b>Sacar aplicación con un bug</b>
<b>PREVENCIÓN</b>	Realizar pruebas en nuestro código con mucha frecuencia para encontrar errores.
<b>SUPERVISIÓN</b>	Ver que estas pruebas se están realizando correctamente.
<b>PLAN DE CONTINGENCIA</b>	Centrarnos en arreglar ese bug y, en caso extremo, paralizar la aplicación hasta encargarnos del problema.



### Nivel de prioridad 3:

RIESGO	Mala estimación del tiempo del proyecto
PREVENCIÓN	Realizar una estimación del proyecto realista y que de cierto margen de flexibilidad en caso de acumulación de trabajo
SUPERVISIÓN	Llevar un control exhaustivo de las tareas en marcha, las ya realizadas y las que quedan por realizar comprobando si las estimaciones han sido realistas
PLAN DE CONTINGENCIA	Ajustar las estimaciones de tiempo para hacerlas más realistas y contratar más personal
RIESGO	Mala gestión de soluciones imprevistas
PREVENCIÓN	Realizar un plan específico de actuación frente a las situaciones imprevistas compuesto por un comité y crear la figura de un supervisor, encargado de comunicar al equipo la aparición de alguna
SUPERVISIÓN	Realizar reuniones semanales donde el supervisor informe de la situación actual de imprevistos.
PLAN DE CONTINGENCIA	Pedir ayuda externa para el trámite de este tipo de imprevistos
RIESGO	Falta de comunicación entre trabajadores
PREVENCIÓN	Divulgar entre los trabajadores los principios de la metodología lean, que promueve la realización de reuniones con bastante frecuencia y la comunicación entre miembros del equipo
SUPERVISIÓN	Realizar reuniones periódicas en las que se informe del trabajo realizado y se interactúe entre los distintos módulos del proyecto
PLAN DE CONTINGENCIA	Facilitar las reuniones y la comunicación entre los trabajadores sustituyendo parte de la carga de trabajo por tiempo dedicado exclusivamente a la puesta en común de información relevante y relacionada con la coordinación entre trabajadores



RIESGO	Cambio de requisitos del cliente
PREVENCIÓN	Informar al cliente de que va a tener un rol activo en el desarrollo del proyecto e instándole a fijar reuniones con cierta periodicidad donde se les vaya enseñando prototipos y nos muestre sus impresiones
SUPERVISIÓN	Realizar reuniones frecuentes con el cliente
PLAN DE CONTINGENCIA	Reorientar el proyecto a los nuevos requisitos del cliente, lo cual no debería suponer un mayor problema ya que al practicar la metodología lean las iteraciones cortas nos proporcionan una rápida capacidad de reacción
RIESGO	El cliente cancela el proyecto
PREVENCIÓN	Tener frecuentes reuniones y asegurarnos de recibir feedback por parte del cliente. Además, queremos que el cliente esté al tanto del proyecto en todo momento
SUPERVISIÓN	Comprobar que la situación económica del cliente no es mala y que las reuniones se están llevando a cabo con la frecuencia debida
PLAN DE CONTINGENCIA	Intentar usar el trabajo realizado en otro proyecto similar
RIESGO	Algún integrante deja el proyecto
PREVENCIÓN	Crear un sentimiento de responsabilidad y trabajar la solidaridad y el compañerismo dentro del equipo
SUPERVISIÓN	Ver que las relaciones entre integrantes del equipo son buenas y que hay un sentimiento de solidaridad
PLAN DE CONTINGENCIA	Sustitución de personal. Buscar a otros alumnos que puedan apoyar nuestro proyecto. Redistribuir las tareas entre los demás miembros del equipo
RIESGO	Pérdida y divulgación de datos
PREVENCIÓN	Realizar un proyecto que cumpla completamente la legislación vigente en la actualidad respecto a la protección de datos
SUPERVISIÓN	El cifrado de los datos en reposo en el dispositivo móvil se establece en el Estándar de Cifrado Avanzado. Mediante este control de datos se almacenan de forma segura para evitar la extracción maliciosa de la aplicación cuando los datos estén en reposo
PLAN DE CONTINGENCIA	Cambiar todo aquello que no cumpla la ley



*Nivel de prioridad 4:*

RIESGO	Problemas legales con la moneda virtual y los premios
PREVENCIÓN	Informarse sobre el tema y contratar un asesor legal que nos indique los límites legales que se puedan alcanzar
SUPERVISIÓN	Estar en continuo contacto con los asesores legales y preguntar cualquier duda en ese aspecto
PLAN DE CONTINGENCIA	Acudir a un bufete de abogados donde nos indiquen si nos es más conveniente dar marcha atrás en el desarrollo de algunos aspectos de la aplicación en cuyo caso habría que modificarlos o, si es mejor, ir a juicio contra el denunciante

### 3.4 Planificación temporal del control de riesgos

Haremos **frecuentes reuniones** con todos los integrantes del equipo donde se pondrán en común todos los problemas encontrados y entre todos buscaremos la **solución más óptima**. Estas reuniones servirán para acabar con los problemas de la manera más eficiente y rápida posible, de manera **que se pierda el menor tiempo** y así poder cumplir con las estimaciones y plazos establecidos.

### 3.5 Resumen

En este apartado hemos analizado los posibles riesgos del proyecto, describiéndolos y analizándolos según la **frecuencia y el impacto**, y clasificándolos en **órdenes de prioridad**. Después hemos analizado más detenidamente cada uno de los riesgos y hemos establecido un **plan de actuación** con cada uno de ellos, concretando primero cómo preverlos, luego cómo supervisarlos, y, en caso de que ocurran a pesar de haber intentado evitarlos, cómo subsanarlos. Para esto último, hemos establecido lo que llamamos plan de



contingencia. Así esperamos poder **solucionar todos los problemas** que vayan surgiendo, y **completar el proyecto con éxito**.

Las **reuniones entre integrantes del equipo** serán esenciales para abordar estos problemas y solucionarlos de la manera más eficiente posible. Para ello, la comunicación y la relación entre trabajadores debe ser la mejor posible. Así crearemos un ambiente de trabajo donde abordemos los problemas con optimismo y los solucionemos lo más rápido posible.





## 4. Planificación temporal

*El proyecto está organizado a lo largo de **7 meses y medio**, en los que desarrollaremos **5 módulos** (usuarios y empresas, favores, premios y compras, buscador y geolocalización y proyecto). El proyecto se llevará a cabo mediante **5 iteraciones con el cliente**.*

### 4.1 Gráfico Gantt

*En el [Anexo 1](#) se puede encontrar tanto el gráfico Gantt en [LogrollingGantt.png](#) como en [LogrollingGantt.gan](#), en los que se podrá observar con detalle cómo se han distribuido las tareas a lo largo de los meses que dura el proyecto.*

### 4.2 Estructura de descomposición de trabajo / Planificación temporal

*Las distintas iteraciones en las que se dividirá el proyecto están destinadas a realizar tareas como la **formación del equipo**, **generar documentos**, **reparto de tareas**, **generar el código**, **evaluarlo** y **presentarlo al cliente**. Nos organizaremos siguiendo el método de desarrollo de software **Lean**.*

El proyecto está previsto para ser desarrollado desde el 11 de septiembre de 2019 al 30 de mayo de 2020. Las primeras semanas (hasta el 15 de octubre aproximadamente) estarán orientadas a la **formación del equipo** en los conceptos técnicos de Ingeniería del Software, para más adelante (hasta finales del año 2019) proceder a la instalación de las herramientas necesarias con el fin de **generar los documentos** que definirán el proyecto, el plan de proyecto (PP) y la especificación de requisitos de software (ERS, SRS en inglés).



Un aspecto clave del proyecto es el **método de trabajo Lean**, que nos ha impulsado a realizar dos evaluaciones con el cliente antes de la entrega del PP y el ERS y tres más durante el desarrollo de la aplicación. El fin es, lógicamente, entregar **material de mejor calidad**.

Se aprovecharán también estos primeros meses para la **división y organización** del equipo, procurando así crear un entorno motivador, eficiente y cercano en el que desarrollar software más cómodamente. Los roles serán distribuidos con el fin de aprovechar las capacidades y talentos de cada integrante de la mejor manera posible, agrupando a los diferentes miembros en 4 módulos interdependientes. De esta manera, el 15 de diciembre se procederá al cierre del PP y del ERS.

Una vez generados ambos documentos, y dadas las inamovibles obligaciones que tenemos los integrantes del equipo del 22/12/2019 al 27/01/2020, hemos dedicado ese periodo para desconectar del proyecto, atender otros asuntos y traer ideas nuevas. Comenzará de esta manera la segunda parte del proyecto, en la que nos centraremos en la **generación del código** de la aplicación.

Durante todo el proyecto, se dedicará una media de **5 horas semanales** por parte de cada miembro del equipo. Se llevarán a cabo reuniones semanales, en las que, aprovechando el pequeño tamaño del equipo, se pondrán en común los avances realizados desde la reunión anterior y se establecerán los objetivos para la reunión siguiente. Gracias a la constante crítica que se recibirá del resto de integrantes y módulos, cualquier código generado estará sujeto a una **permanente revisión y mejora**. La fecha de entrega definitiva será para el 30 de mayo de 2020, momento en el que ya se habrá obtenido una versión consistente de la aplicación.



Módulo	Actividades	Clasificación
Usuarios y empresas	Documentación empresas	Formación
	Diseño de interfaz de usuario y empresas	Ingeniería
	Evaluación del diseño de IU	Evaluación con el cliente
	Codificación del IU	Construcción y adaptación
	Pruebas IU	Construcción y adaptación
Favores	Documentación bases de datos	Formación
	Diseño de gestión de favores	Ingeniería
	Evaluación del diseño de GF	Evaluación con el cliente
	Codificación del GF	Construcción y adaptación
	Pruebas GF	Construcción y adaptación
Premios y compras	Documentación premios	Formación
	Diseño premios y compras	Ingeniería
	Evaluación diseño premios y compras	Evaluación con el cliente
	Codificación premios y compras	Construcción y adaptación
	Pruebas premios y compras	Construcción y adaptación
Buscador y geolocalización	Documentación ByG	Formación
	Diseño ByG	Ingeniería
	Evaluación ByG	Evaluación con el cliente
	Codificación ByG	Construcción y adaptación
	Pruebas ByG	Construcción y adaptación
Proyecto	Análisis de proyecto	Ingeniería
	Diseño de proyecto	Ingeniería
	Opinión cliente	Evaluación con el cliente
	Revisión y cierre SRS	Evaluación con el cliente
	Revisión y cierre Plan de Proyecto	Planificación y gestión del riesgo, evaluación con el cliente
	Revisión diseños	Ingeniería
	Revisiones código	Construcción y adaptación
	Revisiones de riesgos	Planificación y gestión del riesgo
	Cierre código	Construcción y adaptación
	Entrega proyecto	Evaluación con el cliente



*Primera iteración: 11/09/2019-05/12/2019*

Primera iteración				
Módulo	Documentación y Gestión de Riesgos	Análisis	Diseño	Evaluación
Usuarios y empresas	Documentación empresas (11/ 09/ 2019 - 04/ 10/ 2019)		Diseño de la interfaz de usuario y empresas (15/ 10/ 2019 - 30/ 11/ 2019)	Evaluación del proyecto con el cliente (30/ 11/ 2019 - 05/ 12/ 2019)
Favores	Documentación de favores (11/ 09/ 2019 - 04/ 10/ 2019)		Diseño de gestión de favores (15/ 10/ 2019 - 27/ 11/ 2019)	
Premios y compras	Documentación premios (11/ 09/ 2019 - 04/ 10/ 2019)		Diseño premios y compras (15/ 10/ 2019 - 20/ 11/ 2019)	
Buscador y geolocalizador	Documentación B y G (11/ 09/ 2019 - 04/ 10/ 2019)		Diseño B y G (15/ 10/ 2019 - 30/ 11/ 2019)	
Proyecto		Análisis del Proyecto (04/ 10/ 2019 - 15/ 10/ 2019)	Diseño general del proyecto (PdP y SRS) (15/ 10/ 2019 - 30/ 11/ 2019)	



*Segunda iteración: 05/12/2019-17/12/2019*

Segunda iteración			
Módulo	Análisis	Diseño	Evaluación
Usuarios y empresas	Análisis del feedback del cliente  (05/ 12/ 2019 - 08/ 12/ 2019)	Revisión de los diseños, el PdP y el SRS según el feedback del cliente  (08/ 12/ 2019 - 14/ 12/ 2019)	Evaluación del proyecto con el cliente  (16/ 12/ 2019 - 17/ 12/ 2019)
Favores			
Premios y compras			
Buscador y geolocalizador		Cierre de los diseños, del SRS y del PdP  (14/ 12/ 2019 - 16/ 12/ 2019)	
Proyecto			



*Tercera iteración: 17/12/2019-18/04/2020*

Tercera iteración					
Módulo	Análisis	Diseño	Codificación	Prueba	Evaluación
Usuarios y empresas	Análisis del feedback del cliente (18/ 12/ 2019-21/ 12/ 2019)	Revisión de los diseños según el feedback del cliente (28/ 01/ 2020-31/ 01/ 2020)	Codificación de usuarios y empresas (01/ 02/ 2020 - 11/ 03/ 2020)	Pruebas de usuarios y empresas (12/ 03/ 2020 - 31/ 03/ 2020)	Evaluación del producto completamente funcional con el cliente (05/ 04/ 2020 - 18/ 04/ 2020)
Favores			Codificación de favores (01/ 02/ 2020 - 13/ 03/ 2020)	Prueba de favores (14/ 03/ 2020 - 02/ 04/ 2020)	
Premios y compras			Codificación de premios y compras (01/ 02/ 2020 - 03/ 03/ 2020)	Prueba de premios y compras (04/ 03/ 2020 - 18/ 03/ 2020)	
Buscador y geolocalizador			Codificación de B y G (01/ 02/ 2020 - 15/ 03/ 2020)	Prueba de B y G (16/ 03/ 2020 - 04/ 04/ 2020)	
Proyecto					



### Cuarta iteración: 18/04/2020-15/05/2020

Cuarta iteración				
Módulo	Análisis	Codificación	Prueba	Evaluación
Usuarios y empresas	Análisis del feedback del cliente  (19/ 04/ 2020 - 25/ 04/ 2020)	Revisión de la codificación según el feedback del cliente  (26/ 04/ 2020 - 10/ 05/ 2020)	Pruebas tras las modificaciones  (11/ 04/ 2020 - 13/ 05/ 2020)	Evaluación del producto final con el cliente  (14/ 05/ 2020 - 15/ 05/ 2020)
Favores				
Premios y compras				
Buscador y geolocalizador				
Proyecto				

### Quinta iteración: 15/05/2020-30/05/2020

Quinta iteración				
Módulo	Análisis	Codificación	Prueba	Evaluación
Usuarios y empresas	Análisis del feedback del cliente  (16/ 05/ 2020 - 17/ 05/ 2020)	Revisión final de la codificación según el feedback del cliente  (18/ 05/ 2020 - 19/ 05/ 2020)  Cierre del código (19/ 05/ 2020 - 20/ 05/ 2020)	Pruebas finales  (21/ 05/ 2020 - 25/ 05/ 2020)	Entrega del proyecto  (26/ 05/ 2020 - 30/ 05/ 2020)
Favores				
Premios y compras				
Buscador y geolocalizador				
Proyecto				





### 4.3 Red de tareas:

Al igual que el gráfico Gantt, en el [Anexo 1](#) se puede encontrar la red de tareas en [LogrollingRedTareas\(PERT\).png](#), para la que ha usado la técnica PERT.

### 4.4 Tabla de uso de Recursos:

En la siguiente tabla se detallan todos los recursos utilizados para el desarrollo de la aplicación, así como los que necesitarán nuestros clientes para poder ejecutarla. Se han dividido los recursos en hardware y software:

	Desarrollo	Clientes
Hardware	Ordenadores personales del equipo. Ordenadores del laboratorio.	Dispositivo móvil que disponga de conexión a Internet
Software	Modelio GanttProject Microsoft Word Eclipse Github Trello Slack Adobe XD Adobe Photoshop Android Studio Paint 3D Adobe Illustrator Sony Vegas	Android 5.0 Aplicación de Logrolling



## 5. Organización del personal (gestión del equipo)

### 5.1 Estructura del equipo

Antes de decidir el modelo a seguir en cuanto a la estructura del equipo, debemos estudiar los 7 factores definidos por **Mantei** (Santos, 2019) para determinar un tipo u otro:

<b>Dificultad del proyecto</b>	La dificultad del proyecto es media / baja. Los conocimientos necesarios no son demasiado avanzados.
<b>Tamaño del problema</b>	Nuestro proyecto es de tamaño medio/grande, ya que en la aplicación habrá muchas funcionalidades y líneas de código.
<b>Duración del equipo</b>	Somos un equipo de 8 trabajadores desde el inicio del proyecto hasta su final.
<b>Modularidad del problema</b>	La modularidad del problema es media/alta. Todas las funcionalidades tienen cierta relación entre ellas, aunque hay también partes independientes.
<b>Calidad y fiabilidad del sistema a construir</b>	Con la metodología Lean nos aseguramos de eliminar el máximo desperdicio posible antes de la entrega final, puliendo cada detalle y evitando los fallos que se puedan ocasionar.
<b>Fecha de entrega</b>	La última entrega se realizará en junio de 2020.
<b>Comunicación</b>	La comunicación tanto en el equipo como en los distintos módulos de la aplicación es vital para el correcto funcionamiento de la misma.



Teniendo en cuenta la tabla anterior, el equipo decidió su estructura de trabajo y organización que explicamos ahora.

El modelo de gestión que vamos a seguir va ser el **descentralizado democrático**, pero con cierta flexibilidad para ir adquiriendo algunas características del descentralizado, con el fin de realizar tareas en las que se precise un jefe que guíe a los demás trabajadores. Estos jefes se elegirán por consenso entre los miembros del grupo según, por ejemplo, sus conocimientos acerca de la tarea a realizar o capacidad de liderazgo. De este modo, alguien puede tomar la iniciativa en una tarea en la que esté experimentado y enseñar a los demás.

La comunicación en nuestro proyecto es clave y horizontal. Se realizan reuniones cada semana para debatir, plantear dudas y tomar decisiones. En estas tomas de decisiones, se plantean todas las opciones que aparecen y se somete a votación entre los miembros del grupo hasta llegar a un consenso entre todas las posturas. Además de las reuniones, la comunicación es diaria mediante aplicaciones como **Slack**, donde se llevan a cabo dichas votaciones y se plantean posibles dudas.

Respecto a la asignación de tareas, nos dividimos en **pequeños grupos de dos o tres personas sin jefes** para elaborarlas, según las preferencias de los miembros. De este modo se facilita la comunicación entre los integrantes del equipo y se mejora la motivación entre ellos al ser una actividad elegida y no impuesta.

Sin embargo, se ha nombrado a Santiago el **coordinador general** del proyecto, cuya función será agilizar todo el proceso de elección y asignación de tareas y asegurarse de la organización del trabajo.

Tenemos la siguiente organización que se mantendrá a lo largo de todo el proyecto:

<b>Marketing</b>	Santiago y Alberto
<b>Documentación</b>	Jaime, Juan Carlos y Pablo
<b>Diseño</b>	Rubén y Adrián
<b>Coordinador general</b>	Santiago
<b>Coordinador técnico</b>	Pedro

Para mantener en orden de qué tareas se encargan los diferentes integrantes del grupo, usamos **Trello**. En esta aplicación dividimos las tareas, fijamos los plazos de entrega, revisiones, y actualizamos diariamente el estado de nuestros trabajos. De esta forma nos aseguramos de que todos los integrantes del grupo sepamos en tiempo real cómo va el proyecto y así involucrar a todo el equipo en cada tarea.



## 6. Recursos del proyecto

*El proyecto se dividirá en cuatro subsistemas, en cada cual trabajará una pareja. Las aplicaciones y el hardware necesarios son los usuales para el desarrollo de aplicaciones móviles Android.*

### 6.1 Personal

Los 8 integrantes del proyecto serán divididos en 4 parejas. Cada pareja **se encargará de un subsistema** de la aplicación, realizando toda la documentación, diseño y desarrollo de ese subsistema y, además, se encargan de **supervisar otro subsistema**. Para llevar un mayor control, existe un **encargado de revisar** el buen funcionamiento de todos los subsistemas y la coordinación entre ellos.

### 6.2 Hardware

El hardware necesario para Logrolling se divide en tres categorías:

Tipo de hardware	Explicación
Hardware de desarrollo	Es el hardware necesario durante el <b>desarrollo de la aplicación</b> y en el que se van a ejecutar las diversas herramientas para generar código, documentación y recursos, entre otros. Cada integrante del equipo trabajará en su propio ordenador personal.
Hardware del servidor	Es el hardware en el que se ejecutará la parte de la aplicación relativa a <b>bases de datos y el backend</b> de la aplicación. Es suficiente un ordenador con una arquitectura capaz de soportar Linux y conexión a Internet. Durante las pruebas, usaremos un ordenador personal propio para simular un servidor.
Hardware del cliente	Los clientes únicamente deberán disponer de un <b>smartphone Android</b> en el que instalar y ejecutar la aplicación.



## 6.3 Software

Al igual que con el hardware, el software necesario se divide en tres categorías:

### 6.3.1 Software de desarrollo

Software	Descripción
Github.com	Sistema de control de versiones.
Slack	Sistema de comunicación vía mensajes.
Trello	Sistema de gestión de tareas.
Modelio	Entorno de modelado de aplicaciones que permite trabajar con diversos diagramas UML, entre otros, diagramas de casos de uso y de actividad.
GanttProject	Herramienta para crear diagramas de planificación temporal.
Adobe XD	Herramienta de diseño gráfico que permite realizar prototipos.
Adobe Illustrator	Herramienta de diseño gráfico para imágenes vectoriales.
Microsoft Word	Programa de edición de textos para la documentación.
MySQLWorkbench	Herramienta para la gestión de bases de datos MySQL.
Eclipse	Sistema de desarrollo integrado para Java.
Android Studio	Sistema de desarrollo oficial para Android.
Adobe Photoshop	Programa de edición de imágenes.



## 6.3.2 Software de servidor

Software	Descripción
MySQL	Servidor de bases de datos MySQL.
Java	Máquina virtual de Java que ejecuta el backend.

## 6.3.3 Software de desarrollo

El único software del que debe disponer el cliente es de **Android Lollipop 5.0** o posterior con los servicios de Google.



## 7. Plan de GCS / Mecanismos de Gestión y Control

---

*En esta sección definimos las metodologías a seguir por el equipo de Logrolling con el objetivo de trabajar en un producto responsable de manera veraz, honesta y leal obteniendo siempre la máxima eficiencia.*

---

### 7.1 Introducción

#### 7.1.1 Propósito

El plan de gestión de la configuración constituye un elemento clave para **establecer y garantizar la integridad de nuestro producto** durante el proceso de desarrollo, es por ello por lo que el objetivo del presente documento es asegurar que los cambios no se realicen de forma inapropiada, así como **describir las actividades de gestión de configuración** de software y definir los **procedimientos que seguirá el equipo** de *Logrolling*.

#### 7.1.2 Alcance

El ámbito de este documento **incluye en general toda fuente que es manejada dentro del proyecto** y establece un plan para administrar **los productos de software** en todos sus ambientes (desarrollo, pruebas y producción), así como la **documentación del proyecto** (ingeniería, gestión, calidad del producto, usuario).





### 7.1.3 Definiciones de términos clave

<b>GCS</b>	Gestión de configuración del software
<b>Línea Base</b>	Conjunto de elementos de configuración formalmente aprobados que sirve como punto de partida para futuras versiones.
<b>CCC</b>	Comité de control de la configuración
<b>CMDB</b>	Repositorio que relaciona todos elementos de la compañía que son necesarios para la prestación de servicios.
<b>ECS</b>	Elementos de configuración del software
<b>GCA</b>	Departamento de gestión del cambio
<b>SQA</b>	Rol en una organización que se encarga de revisar y auditar los productos

## 7.2 Gestión de la configuración

### 7.2.1 Organización de la GCS

La organización de la GCS se realizará a través del **Comité de control de la configuración (CCC)**, una autoridad cuyas principales funciones son:

1. **Evaluar todas las peticiones de cambio**
2. **Aceptar o rechazar los cambios propuestos**
3. **Tomar las respectivas decisiones sobre los cambios a implementar, cualquier cambio en los requerimientos, o en el diseño.**

Dicho comité estará formado por todos los integrantes del grupo, los cuales desempeñan los siguientes roles:

- Gestor de configuración
- Líder del proyecto
- Gestores de cambio
- Director económico
- Arquitecto de la aplicación
- Arquitectos técnicos y diseñadores



### Coordinador de proyecto:

Santiago Mourenza Rivero

### Gestores de cambio:

Juan Carlos Llamas Núñez  
Pablo Torre Piñana

### Gestor de configuración:

Pedro Palacios Almendros



### Director económico:

Alberto Almagro Sánchez

### Arquitecto de la aplicación:

Jaime Martínez Gamero

### Arquitectos técnicos y diseñadores:

Rubén Gómez Blanco  
Adrián Sanjuán Espejo

## 7.2.2 Descripción de roles del CCC

### 1. Coordinador del proyecto:

Funciones	
	Coordinar las interacciones con los clientes y los usuarios finales.
	Motivar y organiza el equipo de trabajo para lograr un objetivo definido
	Informar sobre el estado actual del proyecto
	Definir las prioridades de las tareas relacionadas con el proyecto



## 2. Gestores de cambio:

Funciones	
	Evaluar el impacto y riesgo de los cambios
	Asegurar que los responsables de los elementos de configuración actualizan los históricos de estos elementos con los cambios implementados
	Planificar las iteraciones
	Conseguir que la estrategia de la compañía se lleve a cabo

## 3. Gestor de configuración

Funciones	
	Gestionar la planificación, identificación, control, seguimiento y auditoría de todos los elementos de configuración en la base de datos de configuración.
	Desarrollar el plan de gestión de configuración
	Promover el uso efectivo de la CMDB
	Asegurar la consistencia e integridad de los datos de la CMDB a través de la ejecución de procedimientos de verificación y auditoría
	Aprobar cambios estructurales en la CMDB.



#### 4. Director económico:

Funciones	
	Analizar desde el punto de vista económico los cambios planteados en las reuniones del CCC.
	Determinar políticas de estrategia y operación del sistema financiero de la empresa
	Dirigir las decisiones financieras de la empresa

#### 5. Arquitecto de la aplicación:

Funciones	
	Identificar todos los ECS de la aplicación a nivel general y reportarlos
	Desarrollar especificaciones funcionales de la aplicación
	Emitir reportes sobre el alcance de los cambios planteados a nivel general en los ECS.

#### 6. Arquitectos técnicos y diseñadores:

Funciones	
	Identificar todos los ECS de la aplicación a nivel de programación y reportarlos
	Desarrollar especificaciones a nivel de programación de la aplicación
	Emitir reportes sobre el alcance de los cambios planteados a nivel de programación en los ECS
	Crear las interfaces y el diseño gráfico de la aplicación



### 7.2.3 Funcionamiento del Comité de control de configuración

El **comité de control de configuración** se reunirá todos los viernes de manera regular además de las reuniones extraordinarias convocadas por cualquier miembro del equipo si este lo considerase necesario

### 7.2.4 Herramientas, entorno e infraestructura

Durante el proceso de gestión de configuración se utilizará un **sistema de subversiones** para el control de versiones del producto.

Cuando algún miembro haga una modificación en el proyecto, deberá acceder al servidor para almacenar la parte modificada en él, de tal forma que **todos los miembros del equipo de desarrollo tendrán acceso a la última versión actualizada del proyecto** en dicho servidor.

Esta **gestión de acceso** al servidor para la actualización se hará mediante la herramienta **GitHub** tanto para los documentos como para el código fuente.

El control de cambios, así como la gestión de defectos, se llevará a cabo mediante un **sistema de revisiones** llevadas a cabo por un miembro ajeno a la realización de dicha parte, permitiendo al equipo de desarrollo ir detallando los defectos encontrados para que la persona responsable pueda corregirlos.



## 7.3 Programa de la Gestión de Configuración

### 7.3.1 Identificación de la Configuración

Identificación de los formularios

Todos los formularios serán identificados con la siguiente nomenclatura “F-TIP-XX” donde:

- F indica que se trata de un formulario
- TIP indica el tipo de formulario
- XX representa el número de petición

Nomenclatura	Tipo formulario
F-PC-XX	Formulario de petición de cambio
F-SRE-XX	Formulario de solicitud de reunión extraordinaria
F-RC-XX	Formulario de registro de cambios



## Identificación de los recursos del proyecto

Nomenclatura	Recursos
<b>S</b>	<b>Servidores</b>
RP-1	Unidad de procesamiento
RP-2	Unidad de almacenamiento
RP-3	Backups
<b>ETD</b>	<b>Entorno de trabajo del desarrollador</b>
RP-4	Unidad de procesamiento (CPU)
RP-5	Almacenamiento en disco local
RP-6	Monitor
RP-7	Teclado
RP-8	Ratón
RP-9	Tablet
RP-10	Router wifi portátil
<b>COM</b>	<b>Comunicaciones</b>
RP-11	GitHub
RP-12	Conexión a internet
RP-13	Conexión de área local
RP-14	Slack
RP-15	Trello
<b>SD</b>	<b>Software de desarrollo</b>
RP-16	Modelio
RP-17	Gantt Project
RP-18	Microsoft Word
RP-19	Eclipse
RP-20	Proto.io
RP-21	Adobe Photoshop



## Identificación de los casos de uso

Nomenclatura	Casos de uso
CU-PF	Pedir favor
CU-PTE	Poner tiempo de entrega
CU-PG	Proponer Grollies
CU-SP	Solicitar Premios
CU-CC	Cambiar configuración
CU-SG	Solicitar Grollies
CU-BF	Buscar un favor por filtro
CU-CH	Chatear con un usuario que propone un favor
CU-SF	Seleccionar favor
CU-NG	Negociar Grollies
CU-CF	Cambiar foto de perfil
CU-CE	Cambiar e-mail
CU-CI	Cambiar idioma
CU-CC	Cambiar contraseña
CU-LN	Leer normas de uso
CU-CF	Conectar con Facebook
CU-GD	Gestionar datos





## 7.3.2 Líneas base del proyecto

Las líneas base **reflejan ideas muy genéricas** debido a la reciente creación del proyecto, aunque **sus visiones de futuro de la aplicación son muy claras** ya que indican que es lo que queremos sin en un principio entrar en el cómo, no obstante, **podrán ser modificadas durante el transcurso del proyecto** si se considera que por ello la aplicación obtendrá mejoras significativas.

Líneas Base	
Línea fundacional	
Objetivo del proyecto Logrolling	Conseguir una mayor conexión entre las personas de los vecindarios de los núcleos urbanos, pueblos y ciudades a través del intercambio de favores.
Líneas esenciales	
Forma del proyecto y dispositivos	Aplicación móvil de Android
Datos que tratamos	<p>1.Recolección de datos relacionados con la posición geográfica de favores, su frecuencia y el mercado que generan para su posterior análisis y posible venta a compañías externas.</p> <p>2.Nunca trataremos, analizaremos ni venderemos de ninguna forma los datos personales de nuestros clientes.</p>
Líneas principales	
Acceso a la aplicación	El usuario debe registrarse de forma obligatoria para poder recibir y ofrecer favores.
Relación entre usuario y aplicación	Los usuarios podrán canjear los puntos obtenidos mediante el intercambio de favores en un sistema de premios implementado en una aplicación o por dinero
Relación entre usuarios	La aplicación tendrá un sistema de valoraciones y reportes que sólo podrán realizarse tras realizar una interacción para garantizar la confianza de nuestros clientes y recompensar a los mejores usuarios.



## 7.4 Control de la Configuración

### 7.4.1 Formularios de petición de cambio (FPC)

El objetivo de este documento es **recoger todas las peticiones** de cambio que surjan durante el desarrollo del producto software. En dicho documento se indicará el análisis que se realice de la **solicitud de cambio** (impacto, esfuerzo, alternativas, consecuencia del rechazo, etc.) así como la decisión final.

La petición de cambio suele venir de **clientes y usuarios** a los que se les plantean nuevas necesidades, pero también pueden venir del equipo de desarrollo por dificultades técnicas no previstas o como resultado de la **resolución de conflictos** o no aceptaciones.

Durante las primeras etapas del proyecto los formularios de petición de cambio provendrán en su gran mayoría de este último grupo.

Respecto al formato utilizaremos la siguiente plantilla:

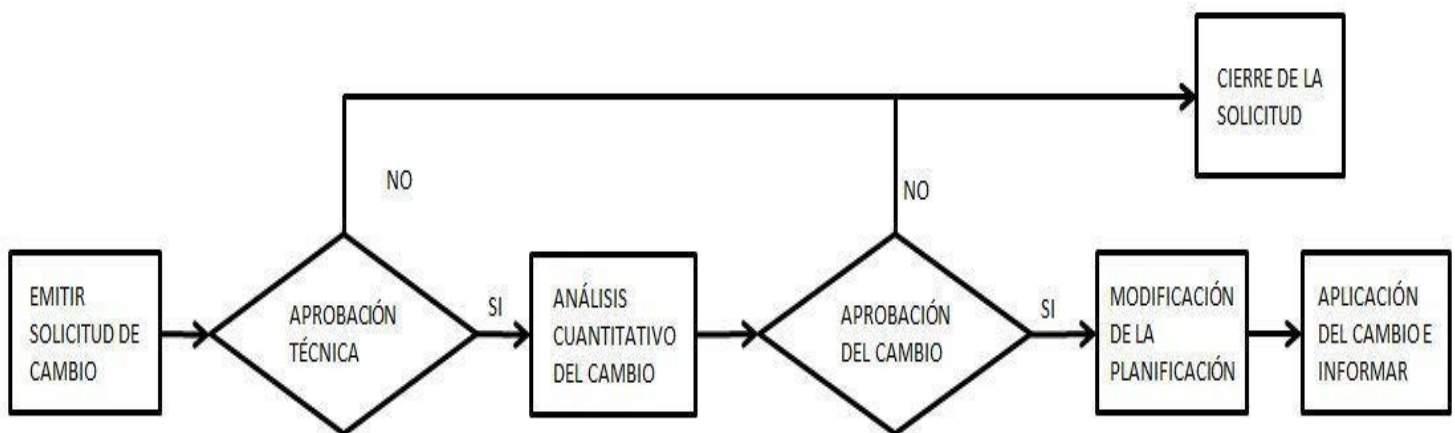
<id>-999	<nombre descriptivo>
<b>Proyecto</b>	<proyecto para el que se solicita el cambio>
<b>Fecha</b>	<fecha de la petición de cambio>
<b>Fuentes</b>	<ul style="list-style-type: none"> <li>• &lt; persona que ha identificado la necesidad del cambio&gt;</li> <li>• ...</li> </ul>
<b>Autores</b>	<ul style="list-style-type: none"> <li>• &lt; persona que formaliza la petición de cambio&gt;</li> <li>• ...</li> </ul>
<b>Descripción</b>	El cambio solicitado consiste en: <descripción del cambio>.
<b>Impacto directo</b>	Los elementos directamente afectados por el cambio son: <ul style="list-style-type: none"> <li>• &lt;elementos afectados directamente organizados por categoría&gt;</li> <li>• ...</li> </ul>
<b>Justificación</b>	El cambio solicitado se considera necesario por: <descripción de la justificación del cambio>.
<b>[Alternativas]</b>	Otras posibles alternativas para abordar la situación descrita son: <ul style="list-style-type: none"> <li>• &lt;descripción de alternativa&gt;</li> <li>• ...</li> </ul>
<b>[Consecuencias del rechazo]</b>	En caso de rechazar el cambio, las posibles consecuencias son: <ul style="list-style-type: none"> <li>• &lt;descripción de las consecuencias de rechazar el cambio&gt;</li> <li>• ...</li> </ul>
<b>[Plazo de resolución]</b>	<fecha en la que se espera la resolución de la petición de cambio>
<b>[Comentarios]</b>	<comentarios adicionales sobre la petición de cambio>



Cuando se genere un FPC, éste debe registrarse para posteriormente ser evaluado por el Comité de Control de Cambios del proyecto (CCC).

La petición debe ser tan **completa** como sea posible para facilitar las posteriores labores de análisis y evaluación. La descripción de la solicitud de cambio deberá ir acompañada por los siguientes campos, a la hora de quedar registrada, para su identificación, como son la descripción y la justificación.

En general, el proceso de solicitud y aprobación de los cambios seguirá el siguiente esquema:



### Aprobación Técnica:

Si el cambio afecta al contenido técnico del proyecto, este **debe ser analizado y aprobado por el responsable técnico del mismo**. Un cambio que desde el punto de vista técnico no sea viable quedará descartado en este punto.

### Análisis cuantitativo:

Para poder proceder con la aprobación del cambio será necesario **cuantificar el efecto que este tendrá sobre las limitaciones del proyecto** (coste, plazo, recursos, etc.) en el caso de ser aplicado. Este análisis debe ser pilotado por el coordinador del proyecto, pero realizado por el equipo del proyecto



## Aprobación del cambio:

El comité de Cambios del proyecto (CCC) decidirá la aceptación o el rechazo del cambio propuesto según el análisis del impacto del mismo.

## Modificación de la planificación:

Una vez aprobado el cambio, **este debe aplicarse a la planificación del proyecto**, lo que **podría implicar modificar las líneas base del proyecto** en base al análisis cuantitativo realizado, y volver a emitir los documentos de planificación que se hayan visto alterados.

## Informar:

**La decisión será remitida al solicitante de la petición de cambio**, indicando en caso de realizarse, en qué versión de la aplicación vendrá resuelto.

## 7.4.2 Formularios de solicitud de reunión extraordinaria (FSRE)

El objetivo de este documento es recoger todas las **peticiones de solicitudes de reuniones extraordinarias** que surjan durante el desarrollo del producto software, ya sea por necesidades de un miembro del CCC o por el equipo de proyecto.

Respecto al formato utilizaremos la siguiente plantilla:

ACTA DE REUNION DE TRABAJO			
Comité o Grupo:		Acta No	
Citada por:		Fecha:	
Coordinador:		Hora inicio:	Fin:
Secretario:		Lugar:	

PARTICIPANTES			
No.	Nombre	Cargo	Teléfono
1			
2			
3			
4			
5			



Al igual que con los FPC, cuando se genere un FSRE, éste debe registrarse para posteriormente ser evaluado por el Comité de Control de Cambios del proyecto (CCC) quien decidirá la aceptación o el rechazo de la reunión propuesta según la relevancia de la misma.

### 7.4.3 Formularios de registro de cambios (FRC)

El objetivo de este documento **es monitorear, registrar y controlar todos los cambios** que han sido tanto aplicados como no en el proyecto con el objetivo de **mantener un histórico** para poder entender el origen de las modificaciones sobre el cronograma y costes debido a los cambios, y el resultado de las diferentes solicitudes, **evitando así repetir el proceso para una solicitud que en su día fue rechazada**.

Estos documentos se generarán tras ser rechazado o aceptado un cambio, y en este último caso se actualizar de manera periódica.

Respecto al formato **utilizaremos la siguiente plantilla:**

Propuesta de Cambio:		
Solicitud N°:	Equipo / Sistema:	
Tipo de Cambio	<input type="checkbox"/> M. Correctivo	<input type="checkbox"/> M. Programado
Descripción y Justificación del cambio:		
Solicitante del cambio	Firma	Fecha
Observaciones a la Implementación del cambio:		



#### 7.4.4 Resolución de conflictos

Con respecto a los formularios mencionados anteriormente, **en caso de producirse una situación de empate** o de no alcanzar una mayoría absoluta en las votaciones que estos generan se realizará un **proceso de resolución** donde los distintos miembros expondrán sus posturas **con el fin de dirimir**.

Posteriormente, tras un plazo de dos días de reflexión se realizará una **segunda votación** dependiendo de la urgencia del conflicto.

#### 7.4.5 Introducción de cambios y reuniones extraordinarias

**Los cambios** aprobados por el CCC tras sus respectivas votaciones **serán introducidos de forma periódica en pequeñas etapas** con el fin de no perturbar a grandes rasgos la estructura del proyecto, para ello se procederá a aplicar una serie de políticas, directivas y procedimientos que indicaremos más adelante en este documento.

Como norma general las solicitudes de cambio para el proyecto se deberán complementar con un **estudio del impacto** que tendrán estos cambios sobre el Proyecto. Todas las solicitudes de cambio, tanto las aprobadas como las denegadas, actualizan el **registro de cambios del proyecto**.

De manera similar a los cambios, las **reuniones extraordinarias** aprobadas por CCC se fijarán en una determinada fecha que estudiará el comité atendiendo a la urgencia que esta supone y al calendario de reuniones.



## 7.5 Políticas, directivas y procedimientos de modificación

En esta sección, se indica como abordaremos la introducción de todos los cambios aprobados por el CCC, siempre respetando la coherencia y uniformidad de la estructura del proyecto.

### 7.5.1 Control de cambios en documentos

En la **documentación del proyecto** que se someta a cambios y modificaciones se seguirá el siguiente procedimiento:

Si el cambio **no supone a grandes rasgos una modificación importante** del documento (menos del 30%) se realizarán los cambios oportunos en los correspondientes apartados sin modificar su identificador salvo cambios de formato los cuales se reflejarán añadiendo tras el nombre del documento “con formato”.

Si el **cambio supone una modificación importante del documento** (más del 30%), se modificará el identificador del documento añadiendo la versión de la que se trata.

Los **registros de cambio** incluyen el nombre del autor y fecha y hora de su modificación además de los cambios realizados en este.

### 7.5.2 Control de cambios de recursos del proyecto

En el caso de modificación de alguno/s de los **recursos del proyecto**, se realizará el siguiente procedimiento:

**Se informará a todo el personal afectado** por dicho cambio con un **período de antelación** suficiente para no perturbar de manera agresiva la estructura y el progreso del proyecto.

En caso de tratarse de cambios relacionados con el software como el lenguaje de programación que requieran conocimientos nuevos se impartirán cursos gratuitos en la empresa para **ampliar la formación de los trabajadores**.



Se realizará un **estudio detallado** de las consecuencias que producirán en el proyecto la aplicación de dichos cambios. Si alguno de estos cambios entrase en conflicto con alguna de las líneas base o con el diseño original del proyecto se convocará una reunión del CCC donde se evaluará la situación.

Se evaluarán los **resultados económicos** del cambio y se intentará que los beneficios superen los costes.

Los detalles del cambio y de su seguimiento se mantendrán **registrados y constantemente actualizados** en su correspondiente FRC.

### 7.5.3 Control de cambios en el código o la interfaz de la app

En el caso de modificación de algún cambio en el código de la aplicación o la interfaz, se realizará el siguiente procedimiento:

Si la modificación del código (más del 20%) o los cambios en la interfaz suponen una **variación importante** para el proyecto se realizará un informe detallado del mismo donde se indicará el nombre del autor la fecha y un comentario sobre las modificaciones propuestas. Dicho documento será entregado al arquitecto de la aplicación y **se llevará al CCC donde será evaluado**.

Si la modificación del código (menos del 20%) o los cambios en la interfaz **no suponen unos cambios críticos** para la estructura del proyecto se realizará un informe detallado de los mismos que **será evaluado por el arquitecto de la aplicación**.

### 7.5.4 Control de cambios en las líneas base

Las líneas base, al tratarse de los cimientos sobre los que se construye el proyecto, su modificación será sometida a **votación en el CCC** donde será necesario obtener **mayoría absoluta** para su modificación.





## 7.6 Informes y auditorías

### 7.6.1 Revisión técnica formal

Dado que la mayoría de los errores de arquitectura y diseño se identifican en la implementación, utilizaremos la revisión técnica formal para prevenir dichos errores y garantizar que la arquitectura es factible para el proyecto.

**Se evaluará** la arquitectura del software en los siguientes aspectos:

- Cumplimiento de los **atributos de calidad**
- Distintas **alternativas de estilos arquitectónicos** que se manejaron
- Si se produjo una **evaluación de riesgos** y si están contemplados en la arquitectura
- Si a partir de ellas se puede **implementar el software requerido**

La revisión seguirá el siguiente **procedimiento**:

- **Presentación de la RTF** (responsable de SQA)
- **Presentación de la Arquitectura** (Arquitecto)
- **Dudas** (Generales)
- **Evaluación de los posibles escenarios**
- **Registro de problemas** e incumplimientos de la Arquitectura y posibles soluciones.



## 7.6.2 Informes de estado del proyecto

Los informes de estado del proyecto tienen como finalidad proporcionar una evaluación del avance y comunicar los detalles de la ejecución.

Información que deben de incluir	
	Fecha de realización del informe
	Periodo a evaluar
	Descripción sobre el informe del estado de proyecto
	Número de horas trabajadas en el periodo.
	Informe sobre el grado de ejecución del plan de riesgos en el periodo
	Informe sobre el estado de ejecución del plan de recursos humanos y comunicaciones en el periodo

## 7.6.3 Auditorías del proyecto

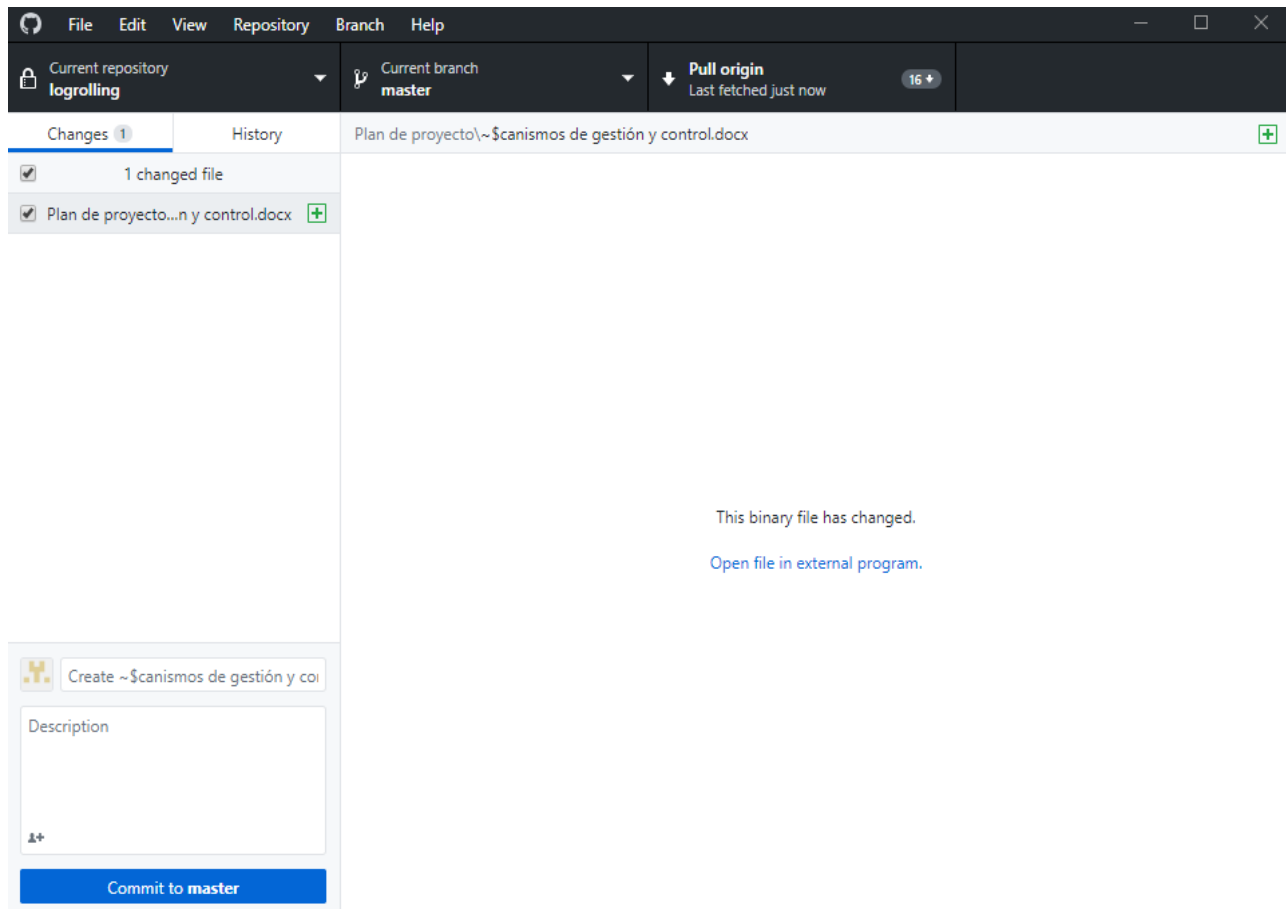
Las auditorías del proyecto se encargan de **revisar todos los ECS de forma exhaustiva e interrelacionada**; por ello, consumen una cantidad de tiempo desmesurada. Tal es así, que **se prescindirá** en este PGC de formalizar procedimientos para la realización de auditorías, pues no se realizará ninguna ni a corto ni a medio plazo



## 7.7 Control de versiones

Durante el proyecto utilizaremos GitHub como sistema de control de versiones ya que su metodología nos garantiza la eficiencia y la confiabilidad del mantenimiento de versiones.

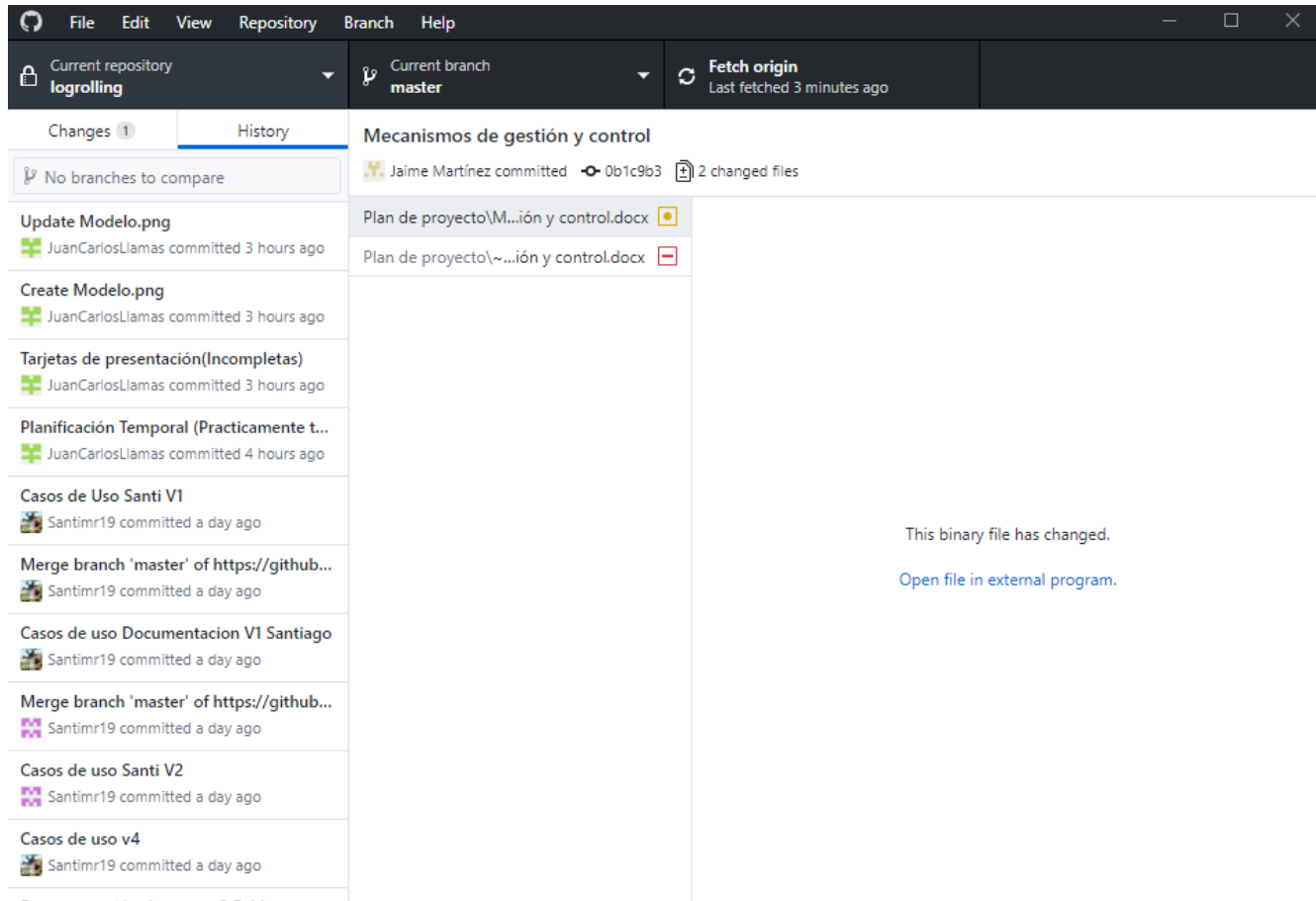
Todos los usuarios tendrán instalados en sus entornos de trabajo GitHub Desktop y dependiendo de su rol en la empresa tendrán acceso a unos repositorios u otros:



De esta forma cada vez que se realice un cambio en alguno de los archivos del repositorio se actualizara la base de datos con dichos archivos. Los usuarios deberán pulsar en “pull origin” y en “fetch origin” para descargar los archivos en sus entornos de trabajo.



Además, el programa cuenta con una pantalla donde se podrán observar los cambios y la fecha realizados por los miembros del trabajo como se muestra en la siguiente imagen:





## 8. Referencias

- Ballesteros, S. (27 de septiembre de 2017). *El 42% de los españoles sufre estrés con frecuencia*. Obtenido de Cadena SER:  
[https://cadenaser.com/ser/2017/09/27/sociedad/1506506301\\_788903.html](https://cadenaser.com/ser/2017/09/27/sociedad/1506506301_788903.html)
- belbex.com. (2019). Obtenido de belbex.com: <https://belbex.com/oficinas/moncloa-aravaca-madrid-madrid/alquiler/hasta-2000-euro-por-mes/>
- clouding.com. (2019). Obtenido de clouding.com: <https://www.clouding360.com/>
- El Androide Libre. (5 de mayo de 2015). *La publicidad en las Apps Android desde un punto de vista técnico*. Obtenido de El androide libre:  
<https://elandroidelibre.elespanol.com/2015/05/la-publicidad-en-las-apps-android-desde-un-punto-de-vista-tecnico.html>
- fotocasa.es. (2019). Obtenido de fotocasa.es:  
<https://www.fotocasa.es/es/alquiler/oficina/madrid-capital/aire-acondicionado/152568414/d?RowGrid=2&tti=3&opi=300>
- idealista.com. (2019). Obtenido de idealista.com:  
<https://www.idealista.com/inmueble/84119306/>
- Indeed.es. (5 de noviembre de 2019). *Indeed.es*. Obtenido de  
<https://www.indeed.es/salaries/programador-junior-Salaries>
- Metro de Madrid. (2019). *metromadrid.es*. Obtenido de  
<https://www.metromadrid.es/sites/default/files/documentos/Portal%20de%20transparencia/Informaci%C3%B3n%20econ%C3%B3mica/Datos%20estad%C3%ADsticos/Entradas%20y%20Utilizaciones%20por%20estaciones%20v3.xlsx>
- Naveira, A. (20 de noviembre de 2017). *Historia de Wallapop: cómo nació la líder española de las apps de segunda mano*. Obtenido de marketing3ecommerce.net:  
<https://marketing4ecommerce.net/historia-wallapop-la-app-sin-ingresos-los-millones-euros/>
- Pastor, J. (24 de marzo de 2017). *La mitad de los ingresos generados en juegos móviles provienen del 0,19% de los usuarios*. Obtenido de Xataka:  
<https://www.xataka.com/moviles/la-mitad-de-los-ingresos-generados-en-juegos-moviles-proviene-del-0-19-de-los-usuarios>
- Santos, C. d. (8 de marzo de 2019). *Organización de equipos según Mantei*. Obtenido de lecasabe.com: <http://lecasabe.com/organizacion-de-equipos-segun-mantei/>
- Unión Europea. (2015). *Directive 2015/2366/EU*.



## 9. Revisiones

Sección	Revisor(es)
Introducción	Adrián Sanjuán
Estimación	Alberto Almagro y Santiago Mourenza
Gestión de riesgos	Juan Carlos Llamas y Pablo Torre
Planificación temporal	Alberto Almagro y Juan Carlos Llamas
Organización del personal	Rubén Gómez
Recursos del proyecto	Pedro Palacios
Mecanismos de Gestión y Control	Jaime Martínez