Citying

Pla de Desenvolupament de Software

1. Organización y equipo

· Gestor de proyectos de Software (1 persona)

. Arquitecto de Software (1 persona)

· Desarrollador de back-end (2 personas -> 1 junior + 1 senior)

· Desarrollador de front-end (1 persona)

· Especialista en Marketing (1 persona - atracción empresas)

· Tester (1 persona)

Nuestro equipo estará compuesto por un total de siete personas. Para empezar, tenemos al gestor de proyectos, que se encargará a tiempo completo de gestionar tanto los recursos como al equipo. A continuación, tendremos también a un arquitecto de software, que se encargará de construir el sistema de forma correcta, en particular la fase de especificación. Luego, repartimos la tarea de programación e integración con 3 personas, 2 back-end (un senior y un junior) y un front-end (senior), todos a tiempo completo. Para finalizar, tendremos también un especialista en marketing, que ayudará a realizar las campañas de marketing y publicidad a tiempo completo, y un tester a jornada completa, que ayudará a mejorar la aplicación.

2. Plan de Proyecto

2.1. Estimación del esfuerzo

Actores		
Nombre	Pesos	UAW
Proveedor de login	1	19
Sistema de pagos	1	
Ubicación de servidor	2	
Proveedor mapas	1	
Servidor de ubicación (GPS)	2	
Equipo externo de reportes	3	
Usuario	3	
Organizador	3	
Participante	3	

UAW = ∑Pesos

Para la justificación de pesos de los actores nos hemos basado en el siguiente criterio. En caso de ser una API le corresponde el peso 1, tendríamos la API de Google Maps o el sistema de Login. Si es un servicio diferente a una API y no requiere interacción humana el peso es 2, como por ejemplo el servidor de ubicación o el propio servidor principal en el que se alojará el sistema. Por último, en todos los actores que conlleven interacción humana su peso es de 3.

Casos de uso		
Nombre	**Pesos**	**UUCW**
UC001	15	230
UC002	10	
UC003	10	
UC004	5	
UC005	5	
UC006	5	
UC007	5	
UC008	5	
UC009	5	
UC010	5	
UC011	5	
UC012	15	
UC013	10	
UC014	10	
UC015	5	
UC016	5	
UC017	5	
UC018	5	
UC019	5	
UC020	10	
UC021	5	
UC022	10	
UC023	15	
UC024	5	
UC025	5	
UC026	5	
UC027	5	
UC028	15	
UC029	15	
UC030	5	

UUCW = ∑Pesos

Para determinar los pesos de los distintos casos de uso, los hemos clasificado por su complejidad en el trabajo.

Los casos de uso con peso 5 son aquellos que sus actividades implican modificar datos dentro del sistema o indicar a otro sistema externo que haga una acción por ejemplo, indicar al sistema de notificaciones de enviar un aviso a los participantes de una actividad.

Los casos de uso con peso 10 son aquellos que su trabajo es intermedio por ejemplo, eliminar la cuenta de un usuario provoca desapuntarlo de todas las actividades que selecciona, y eliminar todas las actividades las cuales fuera organizador teniendo que avisar a muchos usuarios, o hacer un pago a través de un sistema externo de pagos para añadir saldo.

Los casos de uso con peso 15 son aquellos que su trabajo es complejo por ejemplo, pedir al sistema GPS la ubicación del usuario, contactar al proveedor de mapas para que nos dé la información para construir el mapa o pedir al sistema de autentificación que nos verifique el documento oficial de un usuario.

UAW + UUCW
249

Complejidad técnica				
Nombre	Pesos	Prioridad	(PxP)/100	TCF
Distributed System	2	5	0,1	1,12
Performance	1	4	0,04	
End User Efficiency	1	4	0,04	
Complex Internal Processing	1	2	0,02	
Reusability	1	5	0,05	
Easy to Install	0,5	1	0,005	
Easy to Use	0,5	3	0,015	
Portability	2	2	0,04	
Easy to Change	1	2	0,02	
Concurrency	1	3	0,03	
Special Security Features	2	5	0,1	
Provides Direct Access for Third Parties	1	5	0,05	
Special User Training Facilities Are Required	1	1	0,01	

TCF = 0.6 + ∑(Pesos x Prioridad)/100

Para completar la tabla de complejidad técnica hemos analizado cada uno de los puntos de una forma realista, viendo qué implicación tiene en nuestro proyecto y la relevancia que tiene cada uno. Por una parte hemos seleccionado un parámetro de complejidad que puede ser 0.5, 1 y 2 (contando 2 como máxima complejidad), y por otro lado hemos indicado en una escala del 1 al 5 (siendo 5 la prioridad máxima) la prioridad que supone cada apartado en el desarrollo de nuestro proyecto.

Aquellos puntos con complejidad 0.5 son aquellos que tendrán la menor complejidad técnica, aquellos que van a implicar un desarrollo más sencillo y que no requieren de una formación complementaria o una excesiva dificultad en el desarrollo. En nuestro caso tenemos 2 apartados con valor 0.5: Easy to Install and Easy to Use. En el caso de Easy to Install usaremos un sistema proveedor externo, como Google Play o Apple Store, donde el proceso de instalación es realizado automáticamente y no supone una dificultad para el usuario. En el caso Easy to Use, nuestra aplicación no va a tener un sistema de interfaces y funcionamiento muy complicado, estará basado en modelos

estudiados y que se encuentran en uso en miles de aplicaciones hoy en día, así que no va a tener una curva de aprendizaje complicada para el usuario.

En el caso de los apartados con complejidad técnica 1, hemos tenido en cuenta la complicación que supone el desarrollo y las dependencias en otros servicios que los vuelven más complejos. Son aquellos puntos que pueden suponer una carga de trabajo de nivel medio para nuestros programadores y son aspectos básicos que ha de cumplir nuestro software. Estos puntos han de estar bien trabajados ya que asientan en muchos casos la base de nuestro programa, facilitando su reusabilidad, escalabilidad y mantenimiento en caso que se desee.

Por último, aquellos apartados dotados de complejidad técnica 2 son aquellos que nos van a causar más problemas durante el desarrollo, aquellos donde habrá que invertir más tiempo y poner más atención ya que son críticos para el correcto funcionamiento del sistema. Como se puede observar, hemos dado especial importancia a la seguridad en nuestra aplicación, debido a que manejamos datos sensibles de nuestros usuarios y dependemos de ciertos servicios externos que nos ayuden a potenciar la seguridad general del sistema. Existirá una gran complejidad en este punto debido a la integración de todos los servicios que vamos a implementar y sobretodo la atención al detalle para evitar brechas de seguridad que puedan acabar en problemas graves. Así mismo, hemos tenido en cuenta el apartado de portabilidad, debido a que en nuestro caso, nos interesa que nuestro programa pueda ser porteado a la mayor cantidad de dispositivos posibles de una forma rápida, sencilla y económica, así que habrá que diseñar con mucho cuidado nuestra arquitectura de aplicación.

En el caso de las prioridades, hemos tenido en cuenta la relevancia que tiene para nosotros cada uno de los apartados de la tabla, debido a que si tenemos valores bajos de importancia, destinaremos menos recursos en fases importantes de nuestro desarrollo dejando algo más de lado dichos puntos. Pero en el caso de aquellas de prioridad 5, indicará que son puntos críticos para nuestro desarrollo y será conveniente poner a trabajar lo antes posible a nuestro personal para dejar todo listo y cumplir el objetivo lo antes posible, siendo estos los puntos más refinados y trabajados para mejorar al máximo la experiencia del usuario final.

Factores de entorno				
Nombre	Pesos	Evaluación	PxE	ECF
Familiarity with UML/UP	1,5	5	7,5	0,68
Part-Time Workers	-1	0	0	
Analyst Capability	0,5	4	2	
Application Experience	0,5	3	1,5	
Object-Oriented Experience	1	4	4	
Motivation	1	3	3	
Difficult Programming Language	-1	2	-2	
Stable Requirements	2	4	8	

ECF = 1.4 + (-0.03) x ∑(Pesos x Evaluación)

Para los pesos de los factores de entorno hemos usado los estándares[1]. Nuestra aplicación depende de la orientación de objetos por ello, hemos evaluado con 5 a la familiaridad con UML además, es un método que permite especificar al equipo de desarrollo los requisitos del sistema sin depender de la tecnología. No tendremos trabajadores a tiempo parcial por lo tanto, la evaluación 0. Nos interesa que la evaluación de la capacidad de análisis sea 4 porque debe recoger los requisitos del cliente, y los conocimientos y datos relacionados con las actividades que se desarrollan en Barcelona. La experiencia en la aplicación del dominio es 3 porque el equipo debe tener experiencia en softwares que se relacionen con proveedores de mapas, sistemas GPS, sistemas de pagos, sistemas de autentificación de datos personales delicados. La experiencia en Orientación a Objetos es 4 porque nuestro sistema depende de ello. La motivación del equipo es 3 porque nos interesa que el equipo tenga interés en desarrollar las actividades asignadas ya que, podrían beneficiarse de ello porque viven en la misma ciudad. La Dificultad de programación es 2 porque nos interesa que el programador Junior aprenda el lenguaje SQLs en un mes además, de dar espacio a otros miembros del desarrollo a repasar otros lenguajes que se usarán. Los requerimientos estables tienen evaluación 4 porque no consideramos que los requerimientos del sistema sufran muchos cambios frecuentes.

UCP
189,6384

UCP = (UUCW + UAW) x TCF x ECF

Temps esforç
3792,768

Temps Esforç = UCP x PF

Hemos escogido un PF de 20 ya que nuestro equipo será nuevo.

2.2. Presupuesto

Rol	Coste para la empresa
Gestor de proyectos de Software	30.67 €/h
Arquitecto de Software	28.21 €/h
Desarrollador de back-end junior	14.14 €/h
Desarrollador de back-end senior	26.50 €/h
Desarrollador de front-end senior	24.25€/h
Especialista en Marketing	17.25 €/h
Tester	14.26 €/h

3792,77	**Trabajo sobre 7 meses (Teniendo en cuenta festivos para las fechas límite)**			
	Inception	**Elaboration**	**Construction**	**Transition**
Días	14 días	30 días	90 días	25 días
Inicio - Final	12 Febrero - 1 Marzo	4 Marzo - 12 Abril	15 Abril - 23 Agosto	26 Agosto - 1 Octubre
Esfuerzo (en h)	190	759	2.465	379

Tabla Esfuerzo en porcentaje				
Partes equipo	Inception	Elaboration	Construction	Transition
Gestor p.	60%	30%	15%	40%
Arquitecto s.	30%	20%	10%	10%
Back-end jr.	0%	15%	20%	10%
Back-end sr.	10%	15%	25%	15%
Front-end	0%	15%	20%	10%
Marketing	0%	0%	0%	15%
Tester	0%	5%	10%	0%

Tabla Esfuerzo en horas (UCPA = 3793)				
Partes equipo	Inception	Elaboration	Construction	Transition
Gestor p.	114	228	370	152
Arquitecto s.	57	152	247	38
Back-end jr.	0	114	493	38
Back-end sr.	19	114	616	57
Front-end	0	114	493	38
Marketing	0	0	0	57
Tester	0	38	247	0

Totales	Núm. meses	Núm. meses(Redondeado meses naturales laborales)
863	5,39284	7
493	3,08163	5
645	4,02982	6
806	5,03727	7
645	4,02982	6
57	0,35557	2
284	1,77786	4

Trabajadores	Meses	Precio/mes	Precio total meses + ss
Gestor p.	7	4.600 €	47.810 €
Arquitecto s.	5	4.250 €	30.436 €
Back-end jr.	6	2.150 €	18.350 €
Back-end sr.	7	4.025 €	39.855 €
Front-end	6	2.645 €	22.528 €
Marketing	2	2.625 €	8.857 €
Tester	4	2.200 €	11.836 €
Instalaciones/Equipos	Meses	Precio/mes	Precio total meses
Agua	7	24 €	168 €
Luz	7	85€	595€
Internet	7	36€	252€
Local	7	600€	4.200€
Equipos informáticos	7	350€	2.450€
Servidores AWS	7	70€	490€
Call center	7	125 €	125 €

Precio Total (sin costes estruct ni riesgos)	Precio Total (Sin riesgos)	Precio Total
187.951 €	216.144 €	237.758 €

Para empezar a realizar el presupuesto de forma coherente, hubo que determinar el tiempo de trabajo. Para ello, se usó el cálculo de esfuerzo UCPA, dando cómo resultado unas 3793 horas, que se repartieron en porcentajes en cada fase dependiendo del trabajo realizado y el personal requerido. Al realizar los cálculos, obtuvimos que el empleado que con más trabajo tendría que trabajar 5 meses enteros. De ese resultado hay que tener en cuenta que solo se trabajan 5 días a la semana, por lo tanto tendríamos que añadir 2 meses extra, que obtenemos a partir del siguiente cálculo: 4 fines de semana*2 días = 8 días al mes → 5*7 meses = 35 días → 35 / 20 días mensuales laborales = 2 meses extra (aproximadamente, redondeado hacía arriba). Consecuentemente, el total de tiempo de trabajo total real es de 7 meses.

Teniendo en cuenta el tiempo total, el presupuesto consiste en estimar, con la máxima precisión posible, los costes por mes de la aplicación teniendo en cuenta la logística a usar, así como el personal contratado.

Por un lado, en la logística tenemos en cuenta el alquiler del local dónde tiene que trabajar todo el equipo, el alquiler de ordenadores y los servicios de luz, agua, internet y

servidores. Para todo lo anterior se calcula el coste a partir del precio mensual multiplicado por los 7 meses de trabajo, teniendo cómo única excepción el call center, que sólo contamos una tasa fija de instalación (el call center y las apis no las contamos durante los siete meses ya que su precio depende mayormente de sus tasas por uso).
Por otro lado, el coste de los trabajadores se calculó de la siguiente forma: primero, como se puede apreciar en las tablas de arriba, estimamos el tiempo de trabajo de cada uno de ellos en base a los porcentajes de trabajo de cada fase y, a continuación, añadimos a este resultado los 2 meses extra laborales explicados anteriormente. Para finalizar, al igual que con los servicios, se realizó el cálculo de *sueldo mensual * meses de trabajo * extra seguridad social + costes fijos,* con el que obtuvimos el coste total de los empleados.
Para terminar, se realizó un sumatorio de los precios totales de cada apartado y luego, al resultado, se le añadió un 15% de costes estructurales para cubrir averías o cualquier cambio en los precios del apartado de logística, así como un último 10% sobre el total para cubrir cualquier tipo de riesgo al que se pueda someter al proyecto.

3. Plan de fases

3.1. Estado de los casos de uso al final de cada iteración y fase

Habiendo revisado los diferentes patrones del plan de fases presentados en teoría, establecimos que un patrón incremental adaptado a nuestro proyecto sería la mejor opción. En vez de tener seis fases, cómo el incremental estándar, tendría siete.

En *Inception*, al no tratarse de una parte dónde el esfuerzo a realizar sea muy arduo, consideramos que una iteración sería suficiente y se podría concentrar todo el trabajo en ella. Esta fase se centraría en definir bien la visión y la mayoría de los casos de uso, completar el caso de negocio y estimar los costes y por último tener claro los riesgos más importantes de los cuales nos tendríamos que preparar.

A continuación, en *Elaboration* acordamos dos iteraciones para diferenciar dos partes del trabajo de forma clara. Por un lado, en la primera iteración se buscará implementar los casos de uso críticos para el sistema y hacer una primera instalación para probar la

arquitectura. Por el otro lado, en la segunda iteración se comprobarán los riesgos arquitectónicos encontrados en la iteración anterior y se buscará implementar la siguiente tanda de casos de uso principales.

Seguidamente, en *Construction* de igual manera se mantienen 2 iteraciones, que buscan separar el diseño e implementación de los componentes del sistema e implementar el máximo número de casos de uso posibles en la primera iteración y, en la segunda, el completar la implementación de los casos de uso y preparar la beta para ser lanzada en la última fase.

Para finalizar, planteamos realizar una fase de *Transition* con dos iteraciones en vez de una. La primera estará centrada en liberar una beta del producto, para recoger la mayor cantidad de feedback posible y realizar una primera campaña de marketing. Para finalizar, en la última iteración se buscará aplicar el feedback recibido y perfeccionar la aplicación para acabar realizando la entrega al cliente.

	Inception	Elaboration 1	Elaboration 2	Construction 1	Construction 2	Transition 1	Transition 2
UC001	Refinat	Complet	Complet	Complet	Complet	Complet	Complet
UC002	Refinat	Complet	Complet	Complet	Complet	Complet	Complet
UC003	Esbossat	Refinat	Analitzat	Complet	Complet	Complet	Complet
UC004	Refinat	Complet	Complet	Complet	Complet	Complet	Complet
UC005	Identificat	Esbossat	Refinat	Analitzat	Complet	Complet	Complet
UC006	Identificat	Identificat	Refinat	Analitzat	Complet	Complet	Complet
UC007	Identificat	Esbossat	Refinat	Analitzat	Complet	Complet	Complet
UC008	Identificat	Identificat	Refinat	Analitzat	Complet	Complet	Complet
UC009	Identificat	Esbossat	Refinat	Analitzat	Complet	Complet	Complet
UC010	Refinat	Complet	Complet	Complet	Complet	Complet	Complet
UC011	Refinat	Complet.	Complet	Complet	Complet	Complet	Complet
UC012	Refinat	Complet	Complet	Complet	Complet	Complet	Complet
UC013	Esbossat	Refinat	Analitzat	Complet	Complet	Complet	Complet

UC014	Esbossat	Refinat	Analitzat	Complet	Complet	Complet	Complet
UC015	Identificat	Esbossat	Refinat	Analitzat	Complet	Complet	Complet
UC016	Identificat	Esbossat	Refinat	Analitzat	Complet	Complet	Complet
UC017	Identificat	Esbossat	Refinat	Analitzat	Complet	Complet	Complet
UC018	Refinat	Complet	Complet	Complet	Complet	Complet	Complet
UC019	Esbossat	Refinat	Analitzat	Complet	Complet	Complet	Complet
UC020	Refinat	Complet	Complet	Complet	Complet	Complet	Complet
UC021	Identificat	Esbossat	Refinat	Analitzat	Complet	Complet	Complet
UC022	Esbossat	Refinat	Analitzat	Complet	Complet	Complet	Complet
UC023	Refinat	Complet	Complet	Complet	Complet	Complet	Complet
UC024	Identificat	Esbossat	Refinat	Analitzat	Complet	Complet	Complet
UC025	Identificat	Esbossat	Refinat	Analitzat	Complet	Complet	Complet
UC026	Identificat	Identificat	Refinat	Analitzat	Complet	Complet	Complet
UC027	Identificat	Identificat	Refinat	Analitzat	Complet	Complet	Complet
UC028	Refinat	Complet	Complet	Complet	Complet	Complet	Complet
UC029	Refinat	Complet	Complet	Complet	Complet	Complet	Complet
UC030	Identificat	Identificat	Refinat	Analitzat	Complet	Complet	Complet

Para determinar el estado de los casos de uso se han seguido los siguientes criterios:

1. El peso que tienen los casos de uso en nuestro proyecto, centrándonos en priorizar aquellos que resulten críticos en la aplicación, como el caso “buscar actividad”.
2. Siguiendo las directrices de trabajo en el apartado 3.2, donde establecemos de forma clara la carga de trabajo respecto los casos de uso que queremos realizar en cada fase.
3. Buscando cierta correlación con los ejemplos vistos y explicados en teoría.

3.2. Planificación de cada fase

Inception	1a iteración	- Determinar el ámbito del proyecto y las condiciones de frontera. - Definir la visión. - Completar el caso de negocio.

		- Inicio del documento de riesgos aportando información, prevención y mitigación. - Identificar gran parte de casos de uso del sistema y describirlos brevemente. - Refinar los casos de uso críticos para el sistema y aquellos con alta complejidad. - Establecer una arquitectura candidata. - Estimar el coste i planificación temporal - Preparar el entorno de trabajo del proyecto (Aprendizaje de lenguaje de programación en caso necesario, comprar ordenadores, etc.)
Elaboration	1a iteración	- Demostrar que la arquitectura propuesta cumple con la visión con un coste y tiempo razonable - Instalar y probar arquitectura - Implementar casos de uso prioritarios, sean claves del sistema y/o que se comuniquen con sistemas externos. - Avanzar los estados de los casos de uso - Completar los diseños de mockups e UI de la aplicación
	2a iteración	- Añadir los riesgos arquitectónicos no manejables al documento de riesgos con prevenciones y mitigaciones - Implementar la siguiente ronda de casos de uso prioritarios - Avanzar estados de casos de uso - Completar documento de riesgos - Primera versión de la implementación de la UI completada
Construction	1a iteración	- Diseñar e implementar los componentes del sistema - Avanzar estados de casos de uso - Implementar y validar casos de uso - Primera mitad de la implementación de la UI completada - Aprobación de cumplimiento de requisitos no funcionales - Segunda versión de la implementación de la UI completada
	2a iteración	- Todos los casos de uso están completados - Preparación versiones alfa y beta - Completada la implementación de la UI - Aprobación de cumplimiento de requisitos no funcionales
Transition	1a iteración	- Despliegue versión beta - 1a versión campaña de marketing - Obtener y procesar feedback
	2a iteración	- Modificar el sistema según el feedback - Versión final de la campaña de marketing - Entrega proyecto al Cliente

3792,77	Trabajo sobre 7 meses (Teniendo en cuenta festivos para las fechas límite)			
	Inception	**Elaboration**	**Construction**	**Transition**
Días	14 días	30 días	90 días	25 días
Inicio - Final	12 Febrero - 1 Marzo	4 Marzo - 12 Abril	15 Abril - 23 Agosto	26 Agosto - 1 Octubre
Esfuerzo (en h)	190	759	2.465	379

Tabla Esfuerzo en porcentaje				
Partes equipo	Inception	Elaboration	Construction	Transition
Gestor p.	60%	30%	15%	40%
Arquitecto s.	30%	20%	10%	10%
Back-end jr.	0%	15%	20%	10%
Back-end sr.	10%	15%	25%	15%
Front-end	0%	15%	20%	10%
Marketing	0%	0%	0%	15%
Tester	0%	5%	10%	0%

Justificación esfuerzo de cada rol:

En cuanto al esfuerzo por rol, dado que Inception es una fase en la que hay que mantener una comunicación activa con los stakeholders para delimitar los objetivos y requisitos del proyecto, hemos decidido que el gestor realizará el 60% del trabajo, ya que además de gestor realiza la función de analista de software. El resto se repartirá principalmente entre el arquitecto de software y el programador senior. Ya que, el programador senior realizará funciones de asesoría y consultoría para delimitar los requisitos que puede cumplir el sistema, y el arquitecto de software deberá realizar una búsqueda de cual es el mejor patrón a seguir para la arquitectura de nuestro proyecto.

Para la fase de Elaboration el gestor de software sigue teniendo la mayor carga de trabajo porque debe mantener a los stakeholders al día de los cambios y asegurarse de que el proyecto avanza correctamente. El arquitecto de software instalará y comprobará que la arquitectura base que decidió en la fase de inception es la correcta, el equipo de programadores Back-End trabajarán junto con el tester, comenzarán a programar los

casos de uso prioritarios, y por último, el programador Front-End realizará mockups del diseño final de nuestra aplicación.

Para la fase de Construction el gestor de proyecto reduce sus funciones y pasa a mantener comunicaciones puntuales con las partes interesadas y supervisar que el equipo cumpla los plazos. El equipo toma más autonomía, el arquitecto de software acabará la instalación y diseño de los componentes del sistema, el front-end se encarga de implementar el diseño de la interfaz, el equipo de back-end implementan los casos de uso que falten y el tester comprobará que las implementaciones funcionan correctamente.

Por último, en la fase de Transition, cuando se despliega la beta, el gestor de proyecto vuelve a tomar protagonismo porque es quien mantendrá las comunicaciones con los stakeholders e interesados que utilizaran la beta de la aplicación. Y trasladará el feedback al equipo de trabajo para que realicen las modificaciones pertinentes. El encargado de marketing, tendrá su aporte en esta fase, planificará el mejor método para que nuestra aplicación sea bien publicitada y llegue a nuestro público objetivo.

4. Recursos

[1] *Project Estimation with use case points using Enterprise Architect (EA) - PDF free download*..
https://docplayer.net/7144645-Project-estimation-with-use-case-points-using-enterprise-architect-ea.html

[2] Página de información sobre los precios medios mensuales de las posiciones de trabajo del equipo. http://www.glassdoor.es

[3] Documento google con el cálculo UCPA. https://docs.google.com/spreadsheets/d/1A4F6-qkW6aK9p3BPSmuJP_dbh0sM3WJbz7fEOhE3nRs/edit?usp=sharing