

DP2 2021-2022

D04 – Edición de datos Informe de progreso

URL Github: [Repositorio de Github](#)

Miembros:

- Miguel Ángel Gómez Gómez (miggomgom1@alum.us.es)
- Dámaris Gómez Serrano (damgomser@alum.us.es)
- Mariano Martín Avecilla (marmarave@alum.us.es)
- Iván Moreno Granado (ivamorgra@alum.us.es)
- Miguel Ángel Rivas Rosado (migrivros@alum.us.es)
- Rafael Sanabria Espárrago (rafasana9@gmail.com)

GRUPO E3.04

Versión 1.0.7

08-05-2022

Tabla de contenidos

Tabla de contenidos.....	2
Historial de versiones	3
Introducción.....	5
Resumen ejecutivo	5
Contenido	5
Presupuesto	13
Amortización.....	14
Coste Total	14
Conclusiones	14
Bibliografía	15

Histórial de versiones

Fecha	Versión	Descripción de los cambios	Sprint
08/05/2022	V1.0.0	<ul style="list-style-type: none"> • Creación del documento • Introducción • Resumen ejecutivo • TASK-062: Operaciones por todos los usuarios sobre las cuentas de usuario. • TASK-067: Operaciones por inventores sobre patrocinios. • TASK-070: Crear un anuncio como administrador • TASK-073: Intercambio de dinero según la divisa. • TASK-076: Informe de planificación • TASK-077: Informe de progreso. 	4
10/05/2022	V1.0.1	<ul style="list-style-type: none"> • TASK-063: Operaciones por todos los usuarios sobre susurros (creación) • TASK-071: Actualizar la configuración del sistema • TASK-072: Internacionalización de la cantidad de dinero y fechas. 	4
11/05/2022	V1.0.2	<ul style="list-style-type: none"> • TASK-064: Operaciones por inventores sobre componentes • TASK-065: Editar mis herramientas como inventor 	4
13/05/2022	V1.0.3	<ul style="list-style-type: none"> • TASK-063: Operaciones por todos los usuarios sobre susurros (creación) 	4
19/05/2022	V1.0.4	<ul style="list-style-type: none"> • TASK-068: Operaciones por inventores sobre informes de patrocinio. • TASK-069: Operaciones de patrones sobre patrocinios. 	4
20/05/2022	V1.0.5	<ul style="list-style-type: none"> • TASK-069: Operaciones de patrones sobre patrocinios. 	4
21/05/2022	V1.0.6	<ul style="list-style-type: none"> • TASK-066: Operaciones por todos los inventores en cajas de herramientas 	4
23/05/2022	V1.0.7	<ul style="list-style-type: none"> • TASK-074: Implementación de la detección de spam en la aplicación. • TASK-075: Reusabilidad del detector de spam (implementación como proyecto independiente). • TASK-078: Informe de análisis Lint. • TASK-079: Informe de rendimiento. • TASK-080: Test sobre la funcionalidad de Anuncios. • TASK-081: Test sobre la funcionalidad de Susurros. • TASK-082: Test sobre la Configuración del Sistema. • TASK-083: Tests sobre la funcionalidad de Componentes. 	4

		<ul style="list-style-type: none">• TASK-084: Test sobre la funcionalidad de Herramientas.• TASK-085: Test sobre la funcionalidad de Informes de Patrocinio	
--	--	--	--

Introducción

Este documento ha sido elaborado por todos los miembros del equipo de trabajo incluyendo el manager, y supervisado estrictamente por este último, ya que es el que posee una perspectiva sobre todo el grupo y el foco de los desarrolladores es más local a las tareas que se les encargan.

El contenido de este documento describe las diversas tareas y documentos que el equipo de trabajo ha ido realizando según el avance la asignatura. Por lo tanto, el contenido del documento estará estructurado en un orden fiel a la cronología de trabajo del equipo.

Además, no solo presenta el trabajo realizado por el equipo y su progreso, sino que aporta una valoración a posteriori de este, discutida y alcanzada por todos los miembros del equipo.

Resumen ejecutivo

La necesidad de elaboración de este documento radica en detallar de manera explícita la valoración del funcionamiento del trabajo en grupo, teniendo en cuenta las decisiones previas tomadas por cada uno de los miembros del equipo en relación a la planificación del trabajo.

Para el desarrollo de este documento, cada participante que desarrolle un nuevo informe o tarea será el encargado de anotar y describir cómo la ha realizado, junto con la revisión y valoración del manager del proyecto.

De esta forma, siempre tendremos actualizado dicho informe de progreso, y, además, será validado por la persona con la visión más general sobre el proyecto.

Para finalizar, el crecimiento y desarrollo de este documento permitirá al equipo conocer el progreso de su trabajo, tanto individual como colectivo y así adecuar poco a poco su esfuerzo para otorgar mayor eficiencia al equipo. Esto último es de gran importancia, ya que la base del crecimiento de un proyecto en equipo se da en el buen trabajo de sus integrantes y en el conocimiento de la valoración de sus tareas.

Contenido

El trabajo ha sido distribuido entre los distintos miembros del equipo de trabajo a través de los tableros de proyectos de GitHub, de forma que el manager ha asignado a cada miembro una serie de tareas intentando que el trabajo estuviera equilibrado (en tiempo y esfuerzo) entre cada uno de los desarrolladores.

También ha sido el manager el encargado de crear cada rama para cada tarea, de forma que la persona que tuviera que hacer una tarea la pudiera realizar en la rama asignada para dicha tarea. Además, se ha asegurado de que los cambios fuesen adecuados y realizar la pull request (resolviendo los conflictos en caso de que existiesen) para que dichas modificaciones fueran incorporadas a la rama principal (master).

Para llevar a cabo el trabajo en equipo y mantener la comunicación en tiempo real entre los diferentes miembros, ha sido utilizada la plataforma “Discord”, mediante la cual se han llevado a cabo tanto resolución de dudas, como desarrollo en paralelo y reuniones necesarias para una mejor organización del equipo.

A continuación, se muestra un listado de las diferentes tareas que ha realizado el equipo de desarrollo hasta la fecha:

- **TASK-062: Operaciones por todos los usuarios sobre las cuentas de usuario.**

07/05/2022. Mariano Martín Avecilla

Para esta tara se ha utilizado como referencia Acme Jobs y se han modificado las siguientes clases: AuthenticatedInventorCreateService y los formularios authenticated/inventor/form.jsp y authenticated/patronage/form.jsp. En ellas se han modificado los métodos authorise y validate y los formularios se han editado con los campos necesarios para convertir un usuario autenticado en patron o inventor.

La tarea llevó un tiempo total de 1 hora y 30 minutos.

- **TASK-063: Operaciones por todos los usuarios sobre susurros (creación).**

10/05/2022. Dámaris Gómez Serrano

Esta tarea se ha realizado creando un servicio adicional “AnyChirpCreateService” y en la vista del formulario “form.jsp” hemos añadido un botón de submit de creación que aparezca solo cuando la ruta sea “any/chirp/create”. Además, hemos añadido un campo obligatorio de tipo checkbox al formulario para confirmar que queremos crear dicho objeto. El momento de creación lo imponemos desde el servicio como la fecha actual y lo mostramos de solo lectura en la vista.

13/05/2022. Dámaris Gómez Serrano

Se han realizado modificaciones en la funcionalidad tras la revisión en clase. Se ha eliminado el campo creationMoment con readOnly del formulario para no confundir al usuario y se han cambiado un text-area por un text-box y un text-box por un input-email.

La tarea llevó un tiempo total de 1 hora y 30 minutos.

- **TASK-064: Operaciones por inventores sobre componentes.**

11/05/2022. Iván Moreno Granado

Esta tarea se ha realizado junto con la tarea posterior, ya que internamente, los componentes y las herramientas las tratamos como items. Se han creado las clases “InventorItemCreateService”, “InventorItemUpdateService” y “InventorItemDeleteService”, a través de los cuales, se permiten crear, actualizar y eliminar herramientas. Por otro lado, hemos añadido en la vista de form.jsp un botón de submit de creación que aparezca solo el comando que se quiera realizar sea create (para que no aparezca en el show), así como dos botones de submit de edición y eliminación. Por último, aunque hemos intentado separar los botones de componentes y herramientas para que sólo aparezca uno de ellos en sus respectivas vistas de listado, hemos tomado la decisión de crear un único botón que lleve al usuario a la creación de una herramienta o componente, dependiendo de la vista de listado donde se encuentre.

La tarea llevó un total de 2 horas y 15 minutos.

- **TASK-065: Operaciones por todos los inventores sobre herramientas.**

11/05/2022. Miguel Ángel Rivas Rosado

Esta tarea se ha realizado creando unos servicios adicionales “InventorItemCreateService”, “InventorItemUpdateService” y “InventorItemDeleteService”, a través de los cuales, se permiten crear, actualizar y eliminar herramientas (la eliminación y actualización solo se podrán realizar antes de publicar definitivamente la herramienta). Además, en la vista del formulario “form.jsp” hemos añadido un botón de submit de creación que aparezca solo cuando la ruta sea “inventor/item/create”, así como dos botones de submit de edición y eliminación, los cuales aparecen solo cuando la ruta es “inventor/item/show” y dicho item visualizado no se encuentra publicado (atributo published=false). También se ha creado un botón en la vista del listado “list.jsp”, que nos permite dirigirnos a la vista “form.jsp” para crear una nueva herramienta, usando como ruta destino del botón. “inventor/item/create”.

La tarea llevó un tiempo total de 2 horas y 15 minutos.

- **TASK-066: Operaciones por todos los inventores sobre cajas de herramientas.**

21/05/2022. Miguel Ángel Gómez Gómez

Esta tarea se ha realizado atendiendo a las explicaciones y directrices dadas por los profesores en las clases de follow up y en el foro de la asignatura y buscando ideas en el framework de ejemplo Acme-Jobs-22.5. La realización de esta tarea ha resultado bastante dura ya que requería tratar con la entidad intermedia ItemQuantity y no solo con Toolkit. Además, requería de algo sin precedentes hasta el momento en las tareas realizadas, hacer dos métodos unbind en un servicio.

Para la consecución de la tarea ha sido necesario mucho trabajo de prueba y error ya que aunque se disponía de ejemplos similares, eran simplemente similares y la sintaxis en muchas ocasiones variaba.

A la tarea se le dedicó un total de 24 horas.

- **TASK-067: Operaciones por inventores sobre patrocinios.**

04/05/2022. Rafael Sanabria Espárrago

Esta tarea se ha realizado estudiando primero el funcionamiento para aceptar o denegar aplicaciones de Acme Jobs y aplicándolo a nuestro contexto. Durante la realización de esta tarea, nos dimos cuenta de que el período de Patronage no estaba bien implementado y se procedió a solucionarlo.

La tarea llevó un tiempo de 35 minutos y arreglar el período de Patronage unos 40 minutos.

- **TASK-068: Operaciones por inventores sobre informes de patrocinio.**

19/05/2022. Mariano Martín Avecilla

Para esta tarea ha sido necesario se ha creado un nuevo servicio. Un nuevo controlador y un formulario. Un inventor puede crear un nuevo reporte sobre un patronage a través del listado de patronage si el patronage está publicado. También se han creado los archivos de traducción correspondientes. El mayor impedimento ha sido asociar el InventorId al patrocinio y acceder a él para crear el report correspondiente. Además del authoise, que ha sido un poco más complejo que los anteriores.

La tarea llevó un tiempo total de 14 horas.

- **TASK-069: Operaciones de patrones sobre patrocinios.**

19/05/2022. Dámaris Gómez Serrano

Para esta tarea ha sido necesario se han crear nuevos servicios: PatronPatronageCreateService, PatronPatronageDeleteService, PatronPatronageUpdateService y PatronPatronagePublishService. Además, ha habido que modificar el form.jsp y los archivos de traducción .i18n. El mayor obstáculo encontrado ha sido el de asociar el InventorId al patrocinio, ya que no se estaba haciendo de forma correcta tanto en el create como en el update y nos saltaban excepciones.

20/05/2022. Dámaris Gómez Serrano

Tras la revisión, se han incluido las restricciones de que la fechas no pueden ser anteriores al momento de creación ni de que la fecha final pueda ir antes que la fecha de inicio, además de la restricción de que no se pueda cambiar el código al actualizar un patrocinio. También se eliminó el campo Period de la vista form.jsp, que estaba en readonly.

La tarea llevó un tiempo total de 14 horas.

- **TASK-070: Operaciones por administradores sobre anuncios.**

06/05/2022. Iván Moreno Granado

Para realizar esta tarea se ha creado nuevos jsp en la subcarpeta “administrator” de la carpeta de vistas. Uno de estos jsp se trata del form vacío que se utiliza para crear un nuevo anuncio y el otro se trata de un list (al igual que el list creado anteriormente en la subcarpeta authenticated) con la particularidad del botón de crear un nuevo anuncio (necesario para llevar al administrador al form vacío).

Además, se ha creado otra carpeta de features donde contiene un controlador nuevo con los comandos create, show y list (para el administrador), junto con los servicios de create, list y show, además del repositorio necesario para dichos servicios.

El principal problema encontrado ha sido el paso del modelo a la vista y de la vista al modelo, donde es necesario al hacer el submit de dicho formulario pasarle todos los atributos de la entidad announcement. Sin embargo, la mayoría del tiempo dedicado a esta tarea se ha llevado a cabo para decidir cómo permitir al administrador insertar datos en los campos del form vacío y cómo hacer que el botón de crear un nuevo anuncio aparezca sólo en el listado de un usuario registrado como administrador y no en cualquier usuario registrado. Para ello se han creado prácticamente las mismas vistas que había en la subcarpeta authenticated con la opción del botón de creación.

Finalmente, esta tarea se ha llevado a cabo en un total de 2 horas y 19 minutos.

- **TASK-071: Operaciones por administrador sobre la configuración.**

10/05/2022. Miguel Ángel Rivas Rosado

Esta tarea se ha realizado creando un servicio adicional “AdministratorSystemConfigurationUpdateService” y en la vista del formulario “form.jsp” hemos añadido un botón de submit de actualización que aparezca solo cuando la ruta sea “administrator/system-configuration/show”, de forma que permite a los administradores editar los campos de la conofiguración del sistema y usando el botón creado confirmar dichos cambios.

La tarea llevó un tiempo total de 1 hora y 35 minutos.

- **TASK-072: Internacionalización de la cantidad de dinero y fechas.**

10/05/2022. Miguel Ángel Gómez Gómez

Esta tarea viene ya implementada por el propio framework.

- **TASK-073: Intercambio de dinero según la divisa.**

04/05/2022. Rafael Sanabria Espárrago

Para realizar esta tarea se ha cogido la base de AcmeJobs de MoneyExchange. Se ha creado una entidad MoneyExchangeCache que guarda el rate y la fecha de la primera visita de un día para cada cambio. Esta caché se utiliza para no llamar continuamente a la API. Solo se llama a la API una vez al día para el primer cambio de una divisa a otra. Se creó también otra clase auxiliar con los métodos que llaman a la API, y se aplicaron estos métodos en todos los servicios necesarios, create y update de Item. También se decidió aplicarlos en los show de Item, para así actualizar el rate cuando muestre un Item. Por último, se aplicó la conversión para calcular el precio total de un toolkit.

El tiempo total fue de unas 4 horas.

- **TASK-074: Implementación de la detección de spam en la aplicación.**

23/05/2022. Mariano Martín Avecilla

Para realizar esta tarea se ha creado una clase llamada Spam detector en el cual se le pasa como parámetro un texto, una lista de palabras consideradas como spam y un threshold. El método comprueba que el texto una vez adaptado contiene alguna palabra de la lista de Spam y luego valida que la cantidad de letras del spam dividido por el número total de letras no supera el threshold .

Luego esto se les pasa a los validates de los create, update y published y hace una comprobación tanto para el weak spam como el strong spam, los cuales se obtienen a partir del system configuration. El spam está hecho así para que no diese problemas en la url, porque pensamos en calcular el threshold a través del número de palabras pero nos daría errores en la url ya que la interpretaría como una sola palabra y no consideraría el threshold.

El tiempo total transcurrido fue de 12 horas.

- **TASK-075: Reusabilidad del detector de spam (implementación como proyecto independiente).**

23/05/2022. Dámaris Gómez Serrano

A través de la búsqueda de información de algunos foros se ha podido crear una clase específica que sirva para identificar el spam. A través del contenido publicado en EV, como se vió al inicio del proyecto, se ha podido exportar dicha clase como archivo .jar e importarla en el proyecto.

El tiempo total transcurrido fue de 1 hora y 30 minutos.

- **TASK-076: Informe de planificación.**

03/05/2022. Todos

Creación del informe de planificación, asignación de nuevo manager, asignación y estimación de cada tarea independiente.

El tiempo total entre reunión y redacción del documento ha sido de 1 hora y 30 minutos.

- **TASK-077: Informe de progreso.**

08/05/2022. Todos

Este documento se ha ido desarrollando a medida que cada uno iba completando sus tareas.

El tiempo total ha sido de 2 horas.

- **TASK-078: Informe de análisis Lint.**

23/05/2022. Miguel Ángel Rivas Rosado

Se ha realizado un análisis a través de la herramienta Lint y hemos resuelto algunos bad smells que aparecían, relacionados con la accesibilidad de la página web. A partir de esto, se ha redactado el informe.

El tiempo total ha sido de 30 minutos.

- **TASK-079: Informe de rendimiento.**

23/05/2022. Iván Moreno Granado y Mariano Martín Avecilla

Para ello se ha seguido las instrucciones proporcionadas por las diapositivas: se han ejecutado los tests JUnit desde Eclipse, se han sacado los tiempos en Excel con dos ordenadores y se han hecho comparaciones. Se ha usado el 95% de intervalo de confianza, según nos pedían.

El tiempo total ha sido de 1 hora.

- **TASK-080: Tests sobre la funcionalidad de Anuncios (Task-070).**

23/05/2022. Dámaris Gómez Serrano

Para la realización de esta tarea se ha creado una clase java llamada "AdministratorAnnouncementCreateTest" y dos archivos CSV. En uno de ellos se ha rellenado los datos correspondientes a los anuncios que se crean en dicho test para los casos de prueba positivos y en el segundo de ellos lo mismo para los casos de prueba negativos. En la clase java se han creado dos test parametrizados uno para el csv de los casos de prueba positivos y el otro para los tests de casos de prueba negativos, además de un caso de prueba hacking que comprueba que no puedas entrar a la URL /administrator/announcement/create si no eres administrador.

El tiempo total dedicado a esta tarea ha sido de 1 hora y 30 minutos.

- **TASK-081: Tests sobre la funcionalidad de Susurros (Task-063).**

23/05/2022. Iván Moreno Granado

Para la realización de esta tarea se ha creado una clase java llamada “AnyChirpCreateTest” y dos archivos csv. En uno de ellos se ha rellenado los datos correspondientes a las chirps que se crean en dicho test para los casos de prueba positivos y en el segundo de ellos lo mismo para los casos de prueba negativos. En la clase java se han creado dos test parametrizados uno para el csv de los casos de prueba positivos y el otro para los tests de casos de prueba negativos. La principal dificultad encontrada fue que no se sabía qué hacer con el atributo “creationMoment”, ya que es un atributo que no aparece en el form pero si aparece luego en el listado. Finalmente se decidió no indicarlo en ninguna parte del test.

El tiempo total dedicado a esta tarea ha sido de 1 hora y 30 minutos.

- **TASK-082: Tests sobre la Configuración del Sistema (Task-071).**

23/05/2022. Miguel Ángel Rivas Rosado

Para la realización de esta tarea se ha creado una clase java llamada “AdministratorSystemConfigurationUpdateTest” y dos archivos CSV. En uno de ellos se ha rellenado los datos correspondientes a la configuración del sistema que se actualiza en dicho test para los casos de prueba positivos y en el segundo de ellos lo mismo para los casos de prueba negativos (que deben fallar). En la clase java se han creado dos test parametrizados uno para el CSV de los casos de prueba positivos y el otro para los tests de casos de prueba negativos. Debido a que no existen muchas restricciones en los atributos, se ha limitado a probar en los test negativos que no se dejen campos vacíos al actualizar o que el porcentaje del filtro spam (debe estar entre 0% o 100%) no sea menor que 0 o mayor que 1.

La tarea llevó un tiempo total de 1 hora y 40 minutos.

- **TASK-083: Tests sobre la funcionalidad de Componentes (task 064).**

23/05/2022. Mariano Martín Avecilla

Para la realización de esta tarea se ha creado una clase java llamada “InventorComponentCreateTest” y dos archivos csv. En uno de ellos se ha rellenado los datos correspondientes a los components que se crean en dicho test para los casos de prueba positivos y en el segundo de ellos lo mismo para los casos de prueba negativos. En la clase java se han creado dos test parametrizados uno para el csv de los casos de prueba positivos y el otro para los tests de casos de prueba negativos. El principal problema es que la creación requería de muchos campos por lo tanto se han tenido que probar

diversos campos extremos, como atributos con mínimo 1 en descripción y título o un formato incorrecto en el retailPrice o una cadena que no se correspondiera con una URL en el caso de moreInfo.

El tiempo total dedicado a esta tarea ha sido de 1 hora.

- **TASK-084: Test sobre la funcionalidad de Herramientas (task 064).**

23/05/2022. Miguel Ángel Gómez Gómez

Para la realización de esta tarea se ha creado una clase java llamada “InventorToolCreateTest” y dos archivos CSV. En uno de ellos se ha llenado los datos correspondientes a las tools que se crean en dicho test para los casos de prueba positivos y en el segundo de ellos lo mismo para los casos de prueba negativos. En la clase java se han creado dos test parametrizados uno para el CSV de los casos de prueba positivos y el otro para los tests de casos de prueba negativos. El principal problema es que la creación requería de muchos campos por lo tanto se han tenido que probar diversos campos extremos, como atributos con mínimo 1 en descripción y título o un formato incorrecto en el retailPrice o una cadena que no se correspondiera con una URL en el caso de moreInfo.

El tiempo total dedicado a esta tarea ha sido de 1 hora.

- **TASK-085: Test sobre la funcionalidad de Informes de Patrocinio (task 068).**

23/05/2022. Rafael Sanabria Espárrago

Para la realización de esta tarea se ha creado una clase java llamada “InventorPatronageReportCreateTest” y dos archivos csv. En uno de ellos se ha llenado los datos correspondientes a los reports que se crean en dicho test para los casos de prueba positivos y en el segundo de ellos lo mismo para los casos de prueba negativos. En la clase java se han creado dos test parametrizados uno para el csv de los casos de prueba positivos y el otro para los tests de casos de prueba negativos. También se ha creado un test de hacking que comprueba los permisos.

El tiempo total dedicado a esta tarea ha sido de 40 minutos.

Presupuesto

	Número de horas utilizadas	Precio por hora	Precio total
Desarrollador	82,15	15 EUR	1232,25
Manager	1	25 EUR	25
Analista	22,5	25 EUR	562,5
Tester	7,33333	15 EUR	110
		Total	1.929,75 EUR

Amortización

	Coste de amortización	Amortización por semana	Precio en las cinco semanas
Miguel Ángel Gómez Gómez	2200 EUR	14,10 EUR/semana	70,50 EUR
Dámaris Gómez Serrano	1300 EUR	8,33 EUR/semana	41,65 EUR
Mariano Martín Avecilla	950 EUR	6,09 EUR/semana	30,45 EUR
Iván Moreno Granado	1000 EUR	6,41 EUR/semana	32,05 EUR
Miguel Ángel Rivas Rosado	2000 EUR	12,82 EUR/semana	64,10 EUR
Rafael Sanabria Espárrago	1200 EUR	7,69 EUR/semana	38,45 EUR
		Total 55,44 EUR	277,20 EUR

Coste Total

El coste total de este Sprint (Presupuesto + Amortización) ha sido de **2.944,45 EUR.**

Conclusiones

En este Sprint se han implementado las funcionalidades para editar los datos que se muestran en la aplicación de los usuarios, así como las operaciones de crear, editar y eliminar para la mayoría de las entidades. Además, se han realizado pruebas funcionales para hacernos ver el éxito de la aplicación, y pruebas de rendimiento, para poder ver cómo funciona el sistema.

El equipo de trabajo ha tenido dificultades para llevar un ritmo de trabajo constante a lo largo del Sprint, dado que se ha juntado semanas de vacaciones como la de feria y tras esta semana, hemos tenido numerosos exámenes y entregas de otras asignaturas.

Es por ello que nos ha quedado una tarea sin poder implementar, relativa a las toolkits, ya que el miembro del grupo encargado de ella tuvo que enfrentarse a una tarea difícil y tuvo la mala suerte de estropearse su ordenador días antes de la entrega.

Aun así, el progreso del resto de tareas se ha llevado con unos tiempos acordes a los estipulados.

Gracias a los seguimientos (follow ups) al equipo le ha parecido más sencillo enfrentarse a algunas dificultades que se presentaban, localizando errores, no sólo nuestros sino también de otros equipos de desarrollo, y las alternativas para solventarlos, además de tomar algunos consejos del profesor como guía a la hora de desarrollar dichas funcionalidades.

El conjunto de los desarrolladores es consciente de la importancia de la comunicación y la proactividad para la consecución exitosa del proyecto. Es por ello que pretende seguir trabajando de

manera constante y atendiendo a las directrices e indicaciones del equipo docente para asegurarse de ello.

La diferencia entre el presupuesto estimado en el informe de planificación y el real se debe a que los desarrolladores han necesitado más horas para la implementación de sus tareas, mientras se iban familiarizando con la nueva actualización del framework, incluyendo sus nuevas prestaciones.

Bibliografía

Intencionadamente en blanco.