



PLANNING REPORT

9 DE MARZO DE 2022

E1.07

<https://github.com/JaimeBorregoConde/Acme-Toolkits>

Jaime Borrego Conde: jaiborcon@alum.us.es

Antonio Campos Gil: antcamgil@alum.us.es

Ana Conde Marrón: anaconmar@alum.us.es

Gonzalo Martínez Fernández: gonmarfer2@alum.us.es

Jaime Moscoso Bernal: jaimosber@alum.us.es

Enrique Muñoz Pérez: enrmunper@alum.us.es

Tabla de contenido

Resumen ejecutivo 2

Tabla de revisión 3

Introducción 4

Contenido 5

 Tareas por desarrollar 5

 Asignaciones de las tareas 11

 Presupuesto 12

Conclusiones 13

Bibliografía 14

Resumen ejecutivo

El cometido de este documento es una primera gestión y organización de las tareas que se llevarán a cabo para afrontar el incremento siguiente. Para ello, se detallarán los roles de cada uno de los miembros con el objetivo de que siempre se tengan claras las responsabilidades individuales a cubrir.

Para poder conocer como se está realizando la gestión del proyecto (no se está haciendo lo suficiente rápido, no se han realizado ciertas tareas...) es necesaria una estimación a la que aferrarse. Además, para una primera valoración de costes será indispensable un estudio previo del tiempo mínimo para cada actividad y cada rol.

Tabla de revisión

Número de revisión	Fecha	Descripción
<i>1</i>	<i>9/03/2022</i>	<i>Desarrollo del documento previo al inicio de las tareas</i>

Introducción

Este documento detalla las tareas a realizar para la obtención del primer incremento. Además, indica a qué miembro se le asigna y el tiempo estimado para la realización de la misma.

Por otra parte, se realiza la consignación de los roles a los distintos miembros, cabe recalcar que todos los miembros tienen 2 roles siendo todos desarrolladores.

Por último, se desglosan los gastos previstos para el desarrollo del incremento.

Contenido

Tareas por desarrollar

Incluimos las tareas que el equipo tendrá que desarrollar para la generación del incremento. La estructura de la tarea es:

<identificador> - <título>: <descripción>.

Task-029: Create UML domain model. Produce a UML domain model.

Task-030: Create inventor and patron roles. The principals may have the following project-specific roles: inventor and/or patron (in addition to the default anonymous, authenticated, and administrator roles provided by the development framework). The project-specific roles must have a profile with the following data: a company (not blank, shorter than 101 characters), a statement (not blank, shorter than 256 characters), and an optional link with further information. The system must be delivered with an account for an administrator principal with credentials “administrator/administrator”.

Task-031: Create chirp class. A chirp is an informal short message. The system must store the following data about them: a creation moment, a title (not blank, shorter than 101 characters), an author (not blank, shorter than 101 characters), a body (not blank, shorter than 256 characters), and an optional email address. The author is not required to be the name of an actual principal.

Task-032: Create announcement class. An announcement is a formal piece of news. The system must store the following data about them: a creation moment (in the past), a title (not blank, shorter than 101 characters), a body (not blank, shorter than 256 characters), a flag to indicate whether they are critical or not, and an optional link with further information.

Task-033: Create component class. A component is an invention that is not expected to work standalone, but in conjunction with other components. The system must store the following data about them: a name (not blank, shorter than 101 characters), a code (pattern “^[A-Z]{3}-[0-9]{3}(-[A-Z])?\$”, unique), a technology (not blank, shorter than 101 characters), a description (not blank, shorter than 256 characters), a retail price (positive), and an optional link with further information.

Task-034: Create tool class. A tool is an artefact that allows to work with components. The system must store the following data about them: a name (not blank, shorter than 101 characters), a code (pattern “^[A-Z]{3}-[0-9]{3}(-[A-Z])?\$”, unique), a technology (not blank, shorter than 101 characters), a description (not blank, shorter than 256 characters), a retail price (zero or positive), and an optional link with further information.

Task-035: Create toolkit class. A toolkit is a bundle with components and tools that are expected to work as a whole. The system must store the following data about them: a code (pattern “^[A-Z]{3}-[0-9]{3}(-[A-Z])?\$”, unique), title (not blank, shorter than 101 characters), description (not blank, shorter than 256 characters), assembly notes (not blank, shorter than 256 characters), and an optional link with further information. A

toolkit may have several instances of the same component, but only one instance of a given tool.

Task-036: Create Currency Class. Currency is a class to control the monetary system of our system. We will have to store the following data: Amount, which cannot be negative and currency type which will be an enumerated.

Task-037: Create patronage class. A patronage is a financial sponsorship provided by a patron to an inventor. The system must store the following data about them: a status (proposed, accepted, or denied), a code (pattern “[A-Z]{3}-[0-9]{3}(-[A-Z])?\$”, unique), legal stuff (not blank, shorter than 256 characters), a budget (positive), a period of time (at least one month long, starting at least one month after the patronage is created), and an optional link with further information.

Task-038: Create patronageReport class. A patronage report consists of a series of messages exchanged between an inventor and a patron regarding a particular patronage. The system must store the following data about them: an automatic sequence number (not blank, matches pattern “⟨patronage-code⟩:⟨serial-number⟩”, where “⟨patronage-code⟩” denotes the code of corresponding patronage and “⟨serial-number⟩” denotes a sequential number that starts at “0001” and gets increased with every new patronage report), a creation moment (in the past), a memorandum (not blank, shorter than 256 characters), and an optional link with further information.

Task-039: Create spam class. Create the spam class that must handle the following data: Name, which should be not blank, and type, which indicates whether it is critical or not.

Task-040: Write the planning report. Write down all the tasks, the assignments, the estimated time it will take, what it will cost...

Task-041: Write progress report. This report should contain an assessment of performance throughout the development of the increment as agreed in the group report agreements.

Task-042: The system must handle patron dashboards. The system must handle patron dashboards with the following data: total number of pro-posed/accepted/denied patronages; average, deviation, minimum, and maximum budget of pro-posed /accepted/denied patronages grouped by currency...

Task-043: The system must handle administrator dashboards. The system must handle administrator dashboards with the following indicators: total number of components; average, deviation, minimum, and maximum retail price of components, grouped by technology and currency; total number of tools; average, deviation, minimum, and maximum retail price of tools, grouped by currency; total number of proposed/accepted/denied patronages; average, deviation, minimum, and maximum budget of proposed/accepted/denied patronages.

Task-044: Produce assorted sample data (methodologically). Produce assorted sample data (methodologically). The credentials in the sample user accounts must be set after

the pattern “inventor1/inventor1”, “patron1/patron1”, “inventor2/inventor2”, “patron2/patron2”, and the like.

Task-045: System configuration. The system configuration must include the following initial data:

- A system currency, which must be “EUR” by default.
- A list of accepted currencies, which must be initialised to “EUR”, “USD”, and “GBP”.
- A list of strong spam terms, which must include “sex”, “hard core”, “viagra”, “cialis”, and their Spanish translations by default.
- A strong spam threshold, which must be 10% by default.
- A list of weak spam terms, which must include “sexy”, “nigeria”, “you’ve won”, “one million”, and their corresponding Spanish translations by default.
- A weak spam threshold, which must be 25% by default.

Task-046: Deliver second increment. The operator is in charge of preparing the increment and delivering it to the Delivery Centre.

Task-047: Fix some errors after follow up.

Task-048: Create Items and Quantity Class. Create an item class in order to improve and refactor tool and component class. Furthermore, it necessary to create Quantity class due to Follow ups.

Task-049: Create Configuration Class. Create a configuration class to control some system parameter:

- Strong Spam Terms: String with all strong spam terms, separated by commas.
- Weak Spam Terms: String with all weak spam terms, separated by commas.
- Accepted Currencies: String with all accepted currencies, separated by commas.
- Default Currency: Default currency in the system
- Weak Threshold
- Strong Threshold

Task-050 : Populate Initial Data. Populate initial data with the following data:

- A system currency, which must be “EUR” by default.
- A list of accepted currencies, which must be initialised to “EUR”, “USD”, and “GBP”.
- A list of strong spam terms, which must include “sex”, “hard core”, “viagra”, “cialis”, and their Spanish translations by default.
- A strong spam threshold, which must be 10% by default.
- A list of weak spam terms, which must include “sexy”, “nigeria”, “you’ve won”, “one mil-lion”, and their corresponding Spanish translations by default.
- A weak spam threshold, which must be 25% by default.

Task-051: Populate items and quantity. Populate items class and quantity class with some examples.

Task-052: Populate with more examples. Populate database with more examples to test the app.

A continuación, añadimos otras tareas relacionadas con reuniones de coordinación, clases de teoría, clases de laboratorio y los diversos seguimientos:

Theory Lecture 1/03/2022: Se enseñaron los contenidos de la presentación S01 - Entity, relations, constraints.

Theory Lecture 8/03/2022: Se enseñaron los contenidos de la presentación S03 – Sample data.

Theory Follow Up 15/03/2022: Se realizaron los siguientes comentarios sobre el proyecto:

- En el document de planificación, las tareas van en inglés aunque el document esté en español.
- Para poblar la base de datos es mejor que cada uno cree un csv de una entidad del modelo java que no haya implementado.
- Configuración del sistema:
 - El tipo de moneda por defecto debe quedar patente en el diagram UML.
 - La entidad Spam es una alternativa aunque puede causar conflictos futuros con los umbrales.
 - Sería ideal crear una clase de Parámetros de configuración del Sistema.
- Añadir las anotaciones @Valid y @NotNull al tipo de datos Money y Patronage.

Laboratory Lecture 4/03/2022: Se enseñaron los contenidos de la presentación S02 - Entity, relations, constraints.

Laboratory Lecture 11/03/2022: Se enseñaron los contenidos de la presentación S04 – Sample data.

Laboratory Follow Up 18/03/2022: Se realizaron los siguientes comentarios sobre el proyecto:

- Quantity:
 - Quitar NotNull por el Min(1).
- Dashboards: hay que tener los 2 separados
 - Los tipos no los hemos puesto.
 - Tiene sentido que sean tipos primitivos no nulos.
 - Crearlos mediante Map
- Roles: volver a poner los que sean útiles para la aplicación.
- Money no hace falta ponerlo en el diagrama UML.
- El Periodo, ponerlo directamente en la entidad.
- *Component* y *Tool* se consideran la misma entidad, entonces el inventor gestiona ambas.
- Configuración del sistema:
 - Ver las diapositivas sobre los tipos para List<String>.
 - Una opción es un string separado por comas y tratarlos posteriormente.
- El umbral debe ser double y quitar NotNull al estar ya Range().
- Modelo Java:
- Patronage y PatronageReport: quitar los NotNull.

Coordination meeting 10/03/2022: Se definieron las tareas a cubrir y se realizó el reparto inicial de las mismas.

Coordination meeting 12/03/2022: Se realizó un planteamiento inicial conforme a cómo se iba a abordar el diseño del diagrama UML.

Coordination meeting 13/03/2022: Se comentaron las inconvenientes percibidos durante el diseño del diagrama previamente mencionado y comenzó a plantear la implementación del mismo.

Coordination meeting 15/03/2022: Se discutieron las observaciones del primer *follow-up*.

Coordination meeting 18/03/2022: Se decidió cómo se iban a aplicar las observaciones de los *follow-ups*.

Asignaciones de las tareas

Cada tarea será asignada a una única persona y se realizará en una única rama creada específicamente para el desarrollo de esta.

Miembro	Rol
Jaime Borrego Conde	Manager-Desarrollador
Jaime Moscoso Bernal	Operador-Desarrollador
Ana Conde Marrón	Analista-Desarrollador
Enrique Muñoz Pérez	Desarrollador-Tester
Gonzalo Martínez Fernández	Desarrollador-Tester
Antonio Campos Gil	Desarrollador-Tester

Tabla 1: Miembros-Roles

Tarea	Miembro	Tiempo Empleado
Task 029	Ana Conde Marrón	2 h
Task 030	Jaime Moscoso Bernal	15 min
Task 031	Enrique Muñoz Pérez	15 min
Task 032	Jaime Borrego Conde	15 min
Task 033	Ana Conde Marrón	15 min
Task 034	Gonzalo Martínez Fernández	15 min
Task 035	Ana Conde Marrón	15 min
Task 036	Jaime Borrego Conde	15 min
Task 037	Antonio Campos Gil	15 min
Task 038	Antonio Campos Gil	15 min
Task 039	Enrique Muñoz Pérez	15 min
Task 040	Jaime Borrego Conde	2 h 30 min
Task 041	Jaime Borrego Conde	2 h 45 min
Task 042	Jaime Borrego Conde	15 min
Task 043	Jaime Moscoso Bernal	15 min
Task 044	Ana Conde Marrón	2h
Task 045	Jaime Borrego Conde	2h 15min
Task 046	Jaime Moscoso Bernal	20 min
Task 047	Jaime Borrego Conde	10 min
Task 048	Gonzalo Martínez Fernández	30 min
Task 049	Gonzalo Martínez Fernández	15 min
Task 050	Gonzalo Martínez Fernández	20 min
Task 051	Enrique Muñoz Pérez	15 min
Task 052	Gonzalo Martínez Fernández	15 min
Theory Classes	Todos	4h
Coordination Meetings	Todos	2h 30min
Laboratory Lecture	Todos	4h
Follow-up	Todos	4h

Tabla 2: Miembros-Tiempo Empleado

Presupuesto

Se desglosan los gastos de cada rol en el desarrollo del incremento:

Rol	Coste por hora
Manager	27,46€
Operador	19,79 €
Analista	18,01€
Desarrollador	15,63 €
Tester	13,38€

Tabla 3: Rol-Coste por hora

Rol	Horas	Coste total
Manager	5,25	€144,17
Operador	0,33	€6,54
Analista	2	€36,02
Desarrollador	91,75	€1434,05
Tester	0	€0
Total:		€1620,78

Tabla 4: Rol-Horas-Coste Total

Conclusiones

Este documento trata varios puntos entre los que se pueden encontrar las descripciones de las tareas, los roles asignados y los costes estimados.

En primer lugar, se citan las tareas junto con las descripciones, estas poseen la misma estructura que en el tablero Kanban del repositorio en GitHub, pero en Español (España).

El hecho de que estén expresadas de la misma forma tanto en el documento de planificación como en el tablero Kanban, es un indicador del nivel de detalle que tienen las tareas. A continuación, se asignan las tareas a los distintos miembros, así como una estimación del tiempo. Es una primera estimación, por lo que no tiene por qué ajustarse a lo que ocurrirá.

Seguidamente, se asignan los roles a cada uno de los miembros del grupo. Esta asignación de los roles nos permitirá un mayor rendimiento a la hora de la obtención del incremento, siendo además los roles compatibles entre sí. Cabe destacar que el mánager, al tener una mayor carga de trabajo pues realiza más documentación, tiene menos tareas de desarrollador.

Por último, se hace una estimación de costes, aquí se observa que cuando un mánager hace tareas de desarrollador cobra como desarrollador no como mánager, y así para cada rol, de esta manera, se cobra por lo que se hace y no solo por el tiempo invertido.

Bibliografía

Intencionadamente en blanco.