

DOCUMENTO SCRUM

Proyecto sobre la gestión de un campamento

Miembros del grupo: Álvaro Cordero, Cristina Beltrán, Adriana Vicente, David Viejo y Cristian Manzananas.

Índice

1.Estructura y funcionamiento del equipo SCRUM.....	2
2.Historias de usuario.....	4
3.Sprint reviews.....	13
4.Sprint retrospectives.....	14
5.Sprint plannings.....	15
6.Product backlog.....	16
7.Sprint backlog.....	17
8.Descripción del trabajo realizado por cada miembro del equipo...18	

1. Estructura y funcionamiento del equipo Scrum

Definición de nuestro objetivo

El objetivo que queremos alcanzar con nuestro trabajo es desarrollar un software que permita facilitar la dirección a un cliente que dirija un campamento. Con este software permitiremos que se lleve a cabo una buena gestión de los empleados, los acampados, y todas las demás gestiones que se tienen en cuenta a la hora de organizar un campamento.

Para ello hemos formado un equipo compuesto por 5 personas, Álvaro Cordero, Adriana Vicente, Cristina Beltrán, David Viejo y Cristian Manzananas. A continuación definiremos los roles que hemos llevado a cabo cada uno de nosotros.

Roles

Product Owner:

En nuestro proyecto, el Product Owner ha sido David Viejo, que ha sido el que ha definido desde un comienzo todas las funcionalidades que deberíamos realizar para acabar desarrollando un software completo de acuerdo con nuestro objetivo.

Equipo de desarrollo:

En el equipo de desarrollo nos encontramos los 5 miembros del equipo, ya que cada uno de nosotros tendrá que trabajar la parte del software que nos ha correspondido en la división que hemos realizado al comienzo del proyecto.

SCRUM master:

En cuanto al SCRUM master, Álvaro Cordero es el encargado de que todos los miembros del equipo realicen correctamente su parte del proyecto, guiando a todo el equipo para que el producto final sea de alto valor.

Formas de organización del trabajo

La forma en la que nos vamos a organizar durante este proyecto, va a ser repartiéndose las partes que consideramos más importantes a la hora de organizar un campamento, y que cada uno desarrolle su parte del código, teniendo reuniones semanales todos los martes, tanto el laboratorio de IS como por Google Meet posteriormente por la tarde, y teniendo en nuestra disposición un grupo de WhatsApp donde cada miembro del equipo puede llegar a aportar cualquier tipo de

información que considere necesaria para realizar un mejor trabajo en cualquier momento que crea oportuno.

Para el desarrollo del código, se utilizará como entorno de desarrollo Eclipse, donde el lenguaje de programación será Java. Para que se comparta bien el código entre los miembros del equipo de desarrollo, utilizaremos el repositorio de GitHub, donde cada uno de los miembros irá subiendo su parte del código a medida que la vaya realizando, para que así todo el equipo de desarrollo tenga a su disposición la última versión del código en todo momento.

Hablando ya de la parte del código, utilizaremos una arquitectura multicapa, ya que dividimos cada parte del código en las tres partes que conforman la arquitectura multicapa, que son la capa de presentación, la capa de negocio y la capa de integración, llevando a cabo una buena relación entre ellas que nos permita modificar cada una de ellas sin que llegue a afectar a las demás y que, utilizando este tipo de arquitectura, tenemos una mejora de rendimiento y mayor encapsulación que nos permite trabajar de forma más sencilla entre nosotros. También utilizaremos el Modelo-vista-controlador, donde cada parte del código dispondrá de las vistas que sean necesarias, un controlador que maneje todo lo necesario de cada parte del proyecto y el modelo que realice las operaciones necesarias para cada parte.

Como patrones de diseño de software, hemos implementado y utilizado en nuestro proyecto los siguientes: Singleton, Factory Method, Abstract Factory, Builder, Facade y Observador.

2. Historias de usuario

A continuación, dejaremos la descripción de todas las historias de usuario que se han planteado en el primer sprint:

Historia de usuario Pedir Cita

Descripción:

Como acampado, **quiero** pedir una cita con un médico **para** poder tener una evaluación de mi estado de salud.

Validación

- Que el médico esté disponible
- Que este registrado como acampado

Valor 5

Estimación 5 horas

Prioridad Alta

Historia de usuario Eliminar Cita

Descripción: Eliminar Cita para curar a los pacientes

Como Médico **quiero** eliminar una Cita **una** vez haya atendido al acampado

Validación

- Aparezca como atendida la cita

Valor 8

Estimación 4 horas

Prioridad Alta

Historia de usuario Consultar Citas

Descripción: Consultar las citas de mis pacientes

Como Médico **quiero** ver mis citas **para** saber qué pacientes me han solicitado cita

Validación

- Que haya alguna cita solicitada por un paciente

Valor 8

Estimación 4 horas

Prioridad Alta

Historia de usuario Crear Receta

Descripción: Crear receta para curar a los pacientes

Como Médico **quiero** formalizar una receta **para** curar a los pacientes

Validación

- Que tengamos el medicamento para curarlo

Valor 8

Estimación 4 horas

Prioridad Alta

Historia de usuario Eliminar Receta

Descripción: Eliminar receta para curar a los pacientes

Como Médico **quiero** eliminar una receta **una** vez haya adquirido el medicamento el acampado

Validación

- Aparezca como adquirido la receta

Valor 8

Estimación 4 horas

Prioridad Alta

Historia de usuario Consultar Receta

Descripción: Ver las recetas que he mandado

Como Médico **quiero** ver las recetas **para** eliminarlas o modificarlas

Validación

- Que haya alguna receta , y quiera modificarla o eliminarla
- Una vez el acampado haya adquirido el medicamento

Valor 8

Estimación 4 horas

Prioridad Alta

Historia de usuario Aplicación

Descripción:

Como gerentes, **queremos** crear una aplicación **para** administrar el campamento.

Validación

- Que la fecha de la cita esté libre
- Que la hora de la cita esté libre

Valor 10

Estimación XL

Prioridad Alta

Historia de usuario: Gestionar gastos

Descripción:

Como contable, **quiero** gestionar los gastos del campamento **para** poder financiar adecuadamente las actividades y los gastos de explotación.

Validación

- Que haya presupuesto suficiente (ingresos).

Valor: 9

Estimación: 4-8 horas

Prioridad: Alta

Historia de usuario: Gestionar ingresos

Descripción:

Como contable, **quiero** gestionar los ingresos del campamento **para** poder distribuirlos según las necesidades actuales y futuras. Buscaremos maximizar los beneficios.

Validación

- Que el ingreso se haya efectuado correctamente.

Valor: 9

Estimación: 4-8 horas

Prioridad: Alta

Historia de usuario: Gestionar los horarios de limpieza

Descripción:

Como gestor, **quiero** gestionar los horarios de limpieza **para** que los acampados puedan consultar cuando van a limpiar su dormitorio.

Validación

- Que existan

Valor: 4

Estimación: 4-8 horas

Prioridad: Baja

Historia de usuario: Horarios de limpieza

Descripción:

Como gestor, **quiero** gestionar los horarios de limpieza **para** tener un control sobre los horarios de limpieza que hay en el campamento.

Validación

- Que los horarios de limpieza estén actualizados

Valor: 4

Estimación: 4-8 horas

Prioridad: Baja

Historia de usuario: Mantenimiento

Descripción:

Como gestor, **quiero** gestionar el mantenimiento del campamento **para** arreglar las posibles averías que puedan ocurrir.

Validación

- Que se produzca una avería en el campamento

Valor: 6

Estimación: 4-8 horas

Prioridad: Media

Historia de usuario: Gestión de las instalaciones

Descripción:

Como gestor, **quiero** gestionar las instalaciones del campamento **para** que el coordinador de actividades pueda crear actividades.

Validación

- Conocer las instalaciones que tiene el campamento

Valor: 6

Estimación: 4-8 horas

Prioridad: Media

Historia de usuario Añadir Nueva Actividad

Como Gestor de Actividades **quiero** añadir una nueva Actividad a la lista de actividades **para poder** ofrecer una nueva actividad a los acampados.

Validación

Que no exista una actividad que tenga el mismo código que la nueva

Valor 10

Estimación 5h

Prioridad Alta

Historia de usuario Eliminar Actividad

Como gestor de Actividades **quiero** eliminar una actividad ya existente de la lista de actividades puesto que esa actividad ya no está disponible al haberse pasado la fecha de dicha actividad **para poder** tener un mejor control de las actividades que siguen estando disponibles.

Validación

- Que exista la actividad que se desea eliminar

Valor 6

Estimación 3h

Prioridad Media

Historia de usuario Modificar Actividad

Como gestor de Actividades **quiero** modificar una actividad ya existente de la lista de actividades porque se haya producido algún cambio relacionado con dicha actividad **para poder** permitir a los acampados que estén informados sobre .

Validación

- Que exista la actividad que se desea eliminar

Valor 6

Estimación 3h

Prioridad Media

Historia de usuario Visualizar la Lista de Actividades

Como Acampado **quiero** visualizar la lista de actividades que hay en el campamento **para poder** tener una información de lo que haré y en qué lugar se llevará a cabo dicha actividad.

Validación

- Que haya actividades disponibles

Valor 8

Estimación 7h

Prioridad Alta

Historia de usuario Crear Menú

Descripción: Elaborar un menú

Como empleado, **quiero** elaborar un menú adecuado **para** tener un control de las comidas

Validación

- Que dispongamos de los alimentos necesarios
- Que cumpla con las 5 comidas diarias

Valor 5

Estimación 5 horas

Prioridad Alta

Historia de usuario Modificar Menú

Descripción: Rectificar el menú a mejor

Como empleado, **quiero** cambiar el menú **para** que sea más apetitoso

Validación

- Que haya alguna receta que se pueda mejorar

Valor 5

Estimación 4 horas

Prioridad Alta

Historia de usuario Eliminar Menú

Descripción: Descartar el menú

Como empleado, **quiero** suprimir el menú **para** tener un control de las comidas.

Validación

- Que la comida no se coma
- Que no se disponga de los alimentos

Valor 3

Estimación 2 horas

Prioridad Media

Historia de usuario Consultar Menú

Descripción: Ver el menú diario

Como empleado, **quiero** examinar el menú **para** eliminarlo o modificarlo

Validación

- Que haya algún plato dentro del menú que se quiera eliminar o modificar

Valor 7

Estimación 4 horas

Prioridad Baja

Historia de usuario: Ver Menú

Descripción: Mirar el menú del día

Como acampado, **quiero** visualizar el menú **para** saber que voy a comer durante el día

Validación

- Que esté registrado como acampado

Valor 6

Estimación 4 horas

Prioridad Baja

3. Sprint reviews:

Sprint 1:

En la primera sprint review, asistimos todos los miembros del equipo Scrum, y dejamos claro cuáles han sido los elementos del Product Backlog que se han terminado, que han sido tanto las historias de usuario, el mapping inicial sobre el que intentaremos trabajar siguiendo su modelo y además, hemos creado el repositorio GitHub con el que trabajaremos conjuntamente, subiendo cada uno de nosotros lo que vaya avanzando en cada momento, para que todos en el desarrollo cuenten con la versión actualizada.

Como tareas que se habían planteado para realizar durante este sprint, pero que no se han llegado a terminar, tenemos la creación del principio del código, donde no hemos llegado a terminar los modelos de arquitectura y diseño que queremos implementar posteriormente. Además de eso, hemos empezado a diseñar un diagrama de clases provisional, con el que trabajaremos las partes específicas del código cada uno de nosotros posteriormente.

Sprint 2:

En esta segunda sprint review, hemos vuelto a asistir todos los miembros del equipo Scrum.

En esta reunión, hemos revisado las tareas realizadas nuevamente, y hemos definido que hemos finalizado la primera parte del código que quedó pendiente por hacer en el sprint anterior, además de finalizar también el diagrama de clases

provisional que estábamos realizando en el anterior sprint, y que nos permitirá ponernos a desarrollarlo en las partes más específicas del código.

Después tenemos las tareas que hemos empezado a realizar como son ya el diseño final del UML, que de momento es un boceto del diagrama de clases provisional que hemos realizado, el diagrama de modelos, y las clases comunes en el código, como son las de los empleados y los acampados.

Sprint 3:

En este penúltimo sprint review, hemos vuelto a asistir todos los miembros del equipo, y ya hemos ido detallando cuáles son las pocas tareas que nos quedan por realizar para terminar el proyecto.

Entre ellas, en este sprint ya hemos finalizado el diagrama de modelos que estábamos realizando en el anterior sprint, y también hemos terminado las partes comunes del código, que ya funcionan correctamente.

También se ha destacado que cada miembro del equipo ya ha empezado a implementar y trabajar sobre su parte específica del código.

Sprint 4:

Esta ha sido la última sprint review realizada en el proyecto, en la que se ha dado por finalizado todo, realizando una revisión del código completo y su funcionamiento y viendo que todo funciona correctamente.

4. Sprint retrospectives:

Para inspeccionar las tareas que hemos ido realizando en cada sprint, dejamos aquí el tablero Kanban que hemos ido realizando durante todo el proyecto en cada uno de los sprints.

Plan de sprint	TAREAS POR HACER	TAREAS EN PROCESO	TAREAS FINALIZADAS
SPRINT 15 MARZO 2022	UML Diagrama de modelos	Diagrama de clases provisional Código inicial del proyecto	Mapping Repositorio Github Historias de usuario
SPRINT 29 MARZO 2022	Clases específicas en el código	UML Diagrama de modelos Clases comunes en el código	Código inicial del proyecto Diagrama de clases provisional
SPRINT 19 ABRIL 2022	Revisar funcionamiento del código	UML Clases específicas en el código	Clases comunes en el código Diagrama de modelos
SPRINT 3 MAYO 2022			UML Código completo Revisar funcionamiento del código completo

5. Sprint plannings:

Sprint 1:

Nuestro objetivo en este primer sprint es dejar bien claro cuales son los subsistemas que queremos realizar a lo largo del proyecto para nuestro software y definir los miembros del equipo. Para ello, con la idea de David sobre el producto que queremos desarrollar, queremos dejar definidos cuáles son los subsistemas que realizará cada uno de nosotros.

Como prioridades de este sprint tenemos el realizar un mapping de lo que queremos que se implemente a nivel de software y que cada uno de los integrantes del equipo de desarrollo realice las historias de usuario que le corresponden según la funcionalidad que les haya tocado desempeñar.

Para llevar a cabo este sprint, definiremos en una primera reunión en el laboratorio de IS los subsistemas a realizar por cada uno de nosotros, y posteriormente, cada uno subirá las historias de usuario correspondientes a esa repartición en un documento word compartido a través de Google Drive.

Sprint 2:

Como en el primer sprint conseguimos que cada uno de los miembros del equipo de desarrollo hiciera las historias de usuario correspondientes a la parte del código que deberemos implementar posteriormente, nuestro objetivo principal en este sprint será realizar ya la parte del código correspondiente tanto a la creación de factorías, builders y demás elementos de diseño, como la consecución de las clases

comunes. También deseamos desarrollar un diagrama de clases provisional, con el que nos basaremos de momento en el desarrollo del código.

También queremos dar forma al diseño UML con cada tarea que vayamos terminando y sea definitiva en el código. Además de esto, queremos realizar el diagrama de modelos.

Sprint 3:

En el segundo sprint conseguimos realizar completamente el diagrama de clases provisional y tener ya la parte del código que corresponde al inicio del proyecto. Sin embargo, no conseguimos terminar ni el diagrama de modelos, ni las clases comunes del código. Por lo que nuestro objetivo principal a realizar en este sprint es terminar estas dos partes principalmente.

Además de eso, queremos realizar buena parte del código específico de cada uno y poder ir probando toda la funcionalidad del software. Mientras tanto, como hemos realizado también en el segundo sprint, queremos seguir dando forma al diagrama UML definitivo, según vayamos realizando implementaciones definitivas en el código.

Sprint 4:

Al tratarse del último sprint, obviamente lo que queremos realizar en él es tener la funcionalidad completa del software y realizar las pruebas necesarias para comprobar que todo funcione correctamente, y con ello terminar también el diseño UML definitivo.

6. Product backlog

Sprint 1:

En este primer sprint, con la idea principal del product owner sobre lo que necesitamos implementar para tener un software que nos permita gestionar correctamente un campamento.

Para ello, se han realizado las historias de usuario correspondientes a lo que se desea conseguir para un funcionamiento correcto. También se ha realizado un mapping o boceto inicial, con el que se intenta dejar claro cuáles son las funcionalidades que se desean conseguir en el proyecto, y las relaciones que hay entre ellas. Todo ello se ha subido en un repositorio GitHub donde se encuentran todos los miembros del equipo SCRUM para contar con la última versión actualizada en todo momento.

Sprint 2:

En este segundo sprint seguimos sin sufrir ninguna modificación respecto a las historias de usuario principales.

Lo que se ha añadido al producto en este sprint es un diagrama de clases provisional, que reemplaza al mapping del sprint anterior, para tener un diseño algo más claro de lo que se quiere obtener.

También se ha empezado a trabajar en el código, realizando la parte inicial de éste.

Sprint 3:

En este tercer sprint, si que se han hecho modificaciones respecto a las historias de usuario iniciales, ya que en el subsistema de actividades, se ha pensado en no dejar a los acampados elegir las actividades a las que pueden inscribirse, sino que tendrán la lista de actividades completa del campamento. El motivo de este cambio ha sido porque en una reunión entre el product owner y el equipo de desarrollo, se pensó que en los campamentos todas las actividades son comunes para todos los acampados, por lo que no era necesario darles la opción de que se inscribieran en actividades, sino que deberán acudir a las actividades planteadas por el campamento, que han sido añadidas por el gestor de actividades.

Además de estas modificaciones, se han añadido al producto las partes del código ya funcionales que corresponden a la parte común, como son las partes del registro y el inicio de sesión tanto para acampados como para los empleados.

Sprint 4:

En este último sprint no se ha vuelto a realizar ningún tipo de modificación respecto al sprint anterior, y todo ha seguido la línea que llevamos siguiendo durante todo el proyecto.

Por lo que, ya el proyecto cuenta, tanto con el código funcional completo y revisado, como con el diagrama UML ya definitivo.

7. Sprint backlog

Sprint 1:

En este primer sprint, deseamos realizar las historias de usuario que cada uno de los integrantes del equipo les ha tocado. Por lo que David realizará las historias de usuario correspondientes a la parte de sanidad, Cristina las correspondientes al

comedor, Adriana las de contabilidad, Álvaro las de gestoría y Cristian las de actividades.

También cada uno de estos miembros añadirá en un mapping (que es un boceto inicial de un diagrama de clases) su parte correspondiente y lo que espera implementar a continuación en los siguientes sprints.

Por otro lado, también se desea empezar a implementar la primera parte del código, correspondiente a la creación de factorías, builders,...

Sprint 2:

En este segundo sprint, los objetivos serán realizar ya un diagrama de clases más concreto, que será provisional hasta la entrega final y susceptible a modificaciones durante todo el proyecto.

Además de eso, queremos ya implementar la parte del código que se corresponde a las clases comunes de empleados y acampados para ya poder la implementación de las partes específicas.

Sprint 3:

En este tercer sprint, nuestro objetivo será que cada miembro del equipo desarrolle su parte específica del código y ya empezar a probar la funcionalidad completa del código. También, una vez que hemos conseguido realizar la parte común del código en el sprint anterior, queremos completar un diagrama de modelos.

Sprint 4:

En este sprint, ya queremos probar la funcionalidad de todas las partes específicas del código, completando aquellas que no hayan sido finalizadas todavía, y revisar finalmente el diagrama UML que será el definitivo.

8. Descripción del trabajo realizado por cada miembro del grupo

En primer lugar, comenzamos describiendo el trabajo realizado por Álvaro Cordero, que ha sido el encargado de realizar la mayor parte del código, trabajando en las partes comunes del código que son las de empleados y acampados (el registro e inicio de sesión de estas clases). Además de esto, ha sido el encargado de realizar todo lo que tiene que ver con la funcionalidad de la gestoría.

En segundo lugar, David Viejo, fue el que al comienzo del proyecto dio una idea más detallada sobre lo que deberíamos conseguir con el software que íbamos a desarrollar, aportando ideas también durante todo el desarrollo del proyecto para mejorar funcionalidades. Su parte específica del código, ha sido la del apartado de sanidad.

En tercer lugar, Adriana Vicente, ha sido una de las encargadas de realizar el diagrama UML y además, en su parte del código, desarrollar todo lo que tiene que ver con la contabilidad del campamento.

En cuarto lugar, Cristina Beltrán, ha sido la otra encargada de realizar el diagrama UML junto a su compañera Adriana, y que en la parte del código, ha desarrollado las funcionalidades sobre el comedor del campamento.

Por último, Cristian Manzanas, ha sido el encargado de ir modificando el documento SCRUM durante todo el proyecto, y ha sido el que ha trabajado en la parte del código que se corresponde a todo lo que tiene que ver con las actividades.