



Ingeniería del Software - Prácticas 2019/2020

PRÁCTICA Nº 1

Última modificación: 31 de enero de 2020

Enunciado

1ª Práctica. SCRUM: Realización de un *Sprint* completo

1. Objetivos de la Práctica

Sois un equipo de desarrollo que se acaba de incorporar a una empresa que emplea Scrum como metodología de desarrollo de productos software.

Scrum es una estrategia de gestión del trabajo para el desarrollo incremental de *productos software* mediante equipos multifuncionales que se auto-organizan. Para ello, Scrum define una estructura de roles, reuniones, reglas y artefactos. Los equipos son responsables de auto-organizarse, crear y adaptar sus procesos dentro de este marco de trabajo general.

Scrum es un modelo orientado a la gestión de incertidumbre y entrega rápida de valor al cliente. Para organizar el trabajo a lo largo del tiempo, Scrum utiliza iteraciones de longitud fija denominadas *sprints*. En cada *sprint* el equipo Scrum construye un incremento de producto potencialmente entregable. El objetivo de esta práctica es la realización de un *sprint* de tres semanas a las que se añade una semana adicional para la preparación de los entregables. La duración total de la práctica es por tanto de cuatro semanas.

En esta práctica se aplicará Scrum en las etapas de análisis y diseño del ciclo de vida con el objetivo de finalizar con una primera versión de los documentos de análisis y diseño (cuyas plantillas se aportan). El equipo recibirá las especificaciones de un sistema que se desea desarrollar, en particular el ámbito y la descripción del sistema.

En la Figura 1 se muestra el proceso completo de un *sprint*.

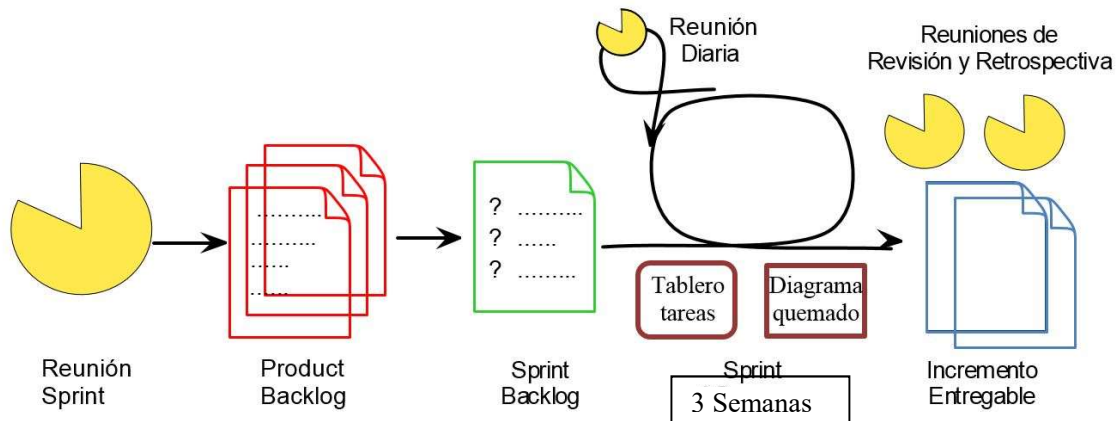


Figura 1: Proceso Scrum de un sprint

2. Desarrollo de la práctica

La práctica durará cuatro semanas y se estructurará de la siguiente forma:

1. Primer día de clase: formación de equipos, explicación de la metodología Scrum, explicación de la práctica y planificación de la misma.
2. Primera semana: fuera de las horas destinadas a la clase de prácticas, los equipos se auto-organizan, llevan a cabo la reunión *sprint planning*, realizan el *product backlog* con las *historias de usuario* y preparan una primera versión del *sprint backlog*.
 - El rol de *Scrum Master* será asumido por el profesor y, con este rol, explicará las dudas y resolverá las dificultades que el equipo encuentre, cambiando de rol si el profesor lo estime necesario. El equipo de estudiantes forma el *Development Team*, de los que uno toma el rol de *Product Owner*.
3. Segundo día de clase: Comienza el *sprint*. Los grupos se reúnen con el profesor para realizar las *daily meetings* (máximo 10 minutos por daily meeting).
4. Tres semanas siguientes:
 - a. Después de refinar el *sprint backlog*, pasar las *historias de usuario* a tarjetas y plantear el *tablero de tareas* inicial del *sprint*, (lo que hará el equipo a continuación de celebrar la primera daily meeting) durante este tiempo los equipos deben definir más en detalle los requisitos seleccionados, reunirse en los *daily meetings* e ir actualizando el *tablero de tareas* y el *diagrama de quemado*. Al final, llevarán a cabo la reunión *sprint review*. (Nota: aunque en un sprint real los *daily meetings* son diarios, para efectos de la práctica se realizarán en las clases por lo que se requerirá un mínimo de tres).
 - b. Se van realizando los documentos de análisis y diseño objetos de este *sprint* y de esta práctica.
 - c. Se recomienda vivamente que los estudiantes lleven al día la planificación. Las *daily meeting* que se celebrarán en las clases serán más productivas cuanto más al día los estudiantes lleven el trabajo. El profesor tomará en consideración que así se haga.



5. Una vez finalizado el *sprint* el equipo reunirá todo el material para llevar a cabo la entrega documental solicitada en la práctica.

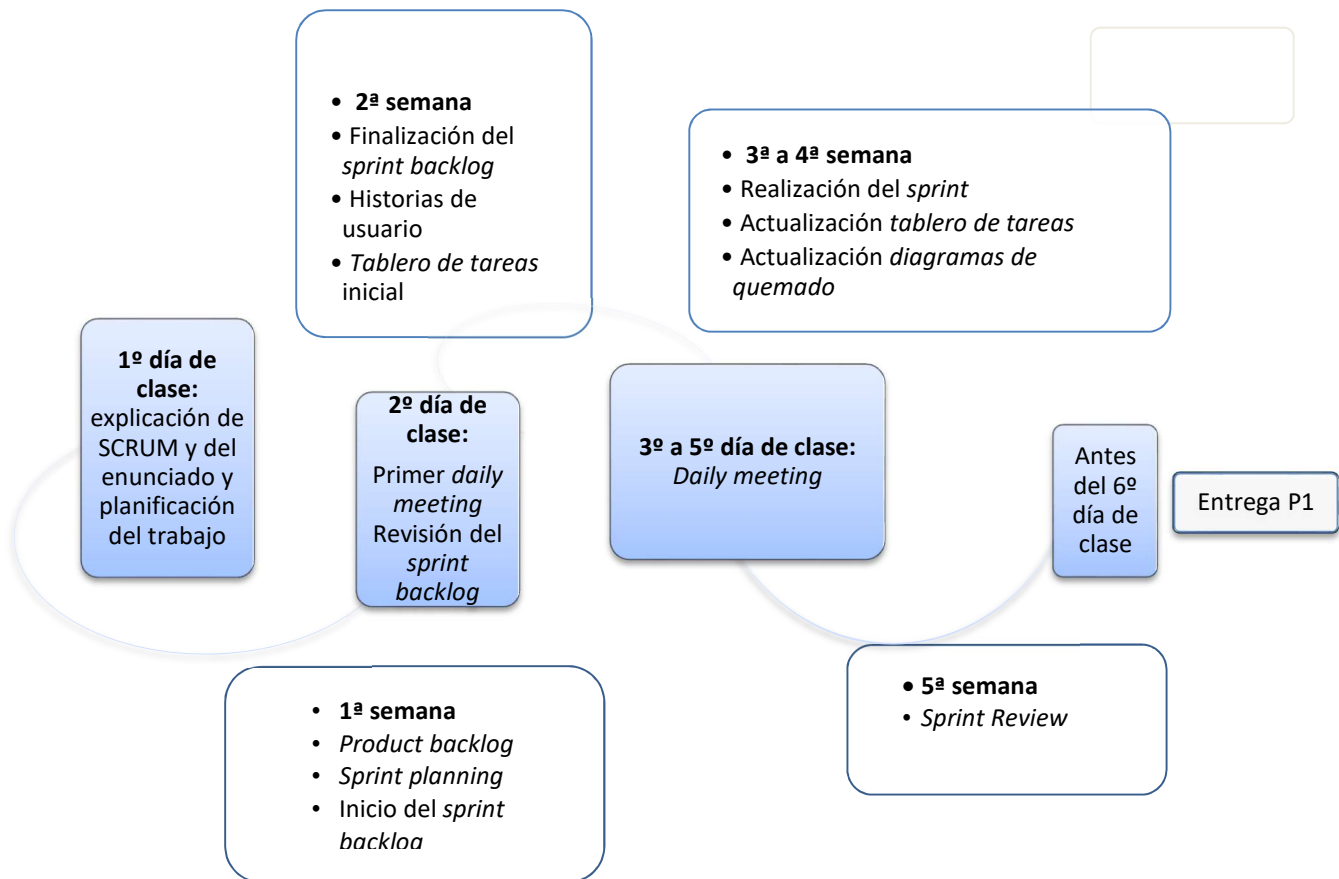


Figura 2: Resumen de ejecución de la práctica

Detalle del trabajo a realizar (Figura 2):

1. El equipo recibe la propuesta del sistema a desarrollar. En particular, la aplicación sobre la que se va a trabajar es un Sistema de Envío de Comida a Domicilio (CometelooToo). En el Anexo A de este enunciado se proporciona una descripción de CometelooToo.
2. Se elabora el *product backlog*, que es el conjunto de todas las *historias de usuario* donde cada historia representa un requisito del cliente.
3. El *product owner* junto con el equipo de desarrollo priorizará y dimensionará (a un nivel alto) el *product backlog* para saber qué historias se van a implementar en este *sprint*.
4. Se lleva a cabo la reunión *sprint planning* que es la que marca el comienzo del *sprint*. Las *historias de usuario* seleccionadas para ser completadas en este *sprint* se pasan al *sprint backlog* donde se descomponen en tareas de corta duración necesarias para ejecutar completamente cada requisito y el esfuerzo estimado por tarea y día.
5. Las *historias de usuario* seleccionadas se pasan a formato tarjeta para poder manejarlas con mayor facilidad. Las tarjetas con las *historias de usuario* y las tareas asociadas se colocan en el *tablero de tareas* para ser manejadas con mayor eficiencia. Los miembros



del equipo van auto-asignándose estas tareas que se revisan en las *daily meetings*. Simultáneamente se va actualizando el diagrama de quemado para mostrar el trabajo pendiente.

6. A lo largo de tres semanas se ejecutan ciclos de Scrum. Se van realizando las *daily meetings* en las clases (máximo 10 minutos por reunión), aunque queda a criterio de cada equipo la posibilidad de realizar más si así se considera necesario. Aunque esta reunión se hace habitualmente utilizando únicamente post-it, en esta práctica, a efectos de corrección, se realizará de cada reunión un acta informal sumarial de media página aproximadamente. Además, se actualiza el *tablero de tareas* moviendo las tareas al estado que corresponda y modificando la asignación si fuera necesario. El estado de cómo queda el tablero después de cada *daily meeting* debe quedar plasmado en la correspondiente acta.
7. Al final del *sprint* se entrega el incremento objetivo (resultado del sprint) en forma de dos documentos, el documento de análisis que incluye, principalmente, el diagrama de casos de uso de la aplicación y la especificación de los casos de uso de al menos tres de los requisitos principales (especificados en el Anexo A) y el documento de diseño que, para el ámbito de esta práctica, contendrá esencialmente el diagrama de clases de la aplicación y los diagramas de secuencia relacionados con los casos de uso anteriores. Se utilizarán como plantilla los documentos que se proporcionan separadamente.
8. El equipo se reúne para valorar cuáles han sido los resultados del sprint en relación a los objetivos planteados. Esta reunión se llama *sprint review*.
9. Se elaboran los entregables de la práctica, es decir, los productos y documentos ya mencionados. Éstos se enumeran en el apartado “Material a entregar” de este enunciado.

Todos los artefactos nombrados anteriormente se describen en el Anexo B, facilitando el formato cuando corresponda.

Todas las reuniones nombradas anteriormente se describen en el Anexo C, facilitando el formato de acta cuando corresponda.

3. Casos de uso mínimos a especificar por los estudiantes

Se especifican en el Anexo A (Apartado A.3).

4. Aspectos prácticos

Se deben tener en cuenta los siguientes criterios básicos a los que debe ajustarse la práctica:

1. Debido a la naturaleza de la práctica, no se entregará una memoria única, sino el conjunto de documentos que se detallan en el apartado Material a entregar de este enunciado.
2. Ejemplos de formato de los distintos artefactos y documentos se proporcionan en los Anexos B y C. No es obligatorio utilizar estos esquemas exactamente igual pero sí tienen que contener, al menos, los campos indicados en las plantillas ejemplo. La evolución del *tablero de tareas* se puede presentar con fotos del tablero.



3. Se deben indicar las referencias bibliográficas utilizadas, incluidas aquellas fuentes de donde se saquen otros datos (revistas especializadas, Internet, etc.).

Material a Entregar

El material a entregar debe constar de 10 ficheros incluidos en un fichero comprimido. Los nombres de los ficheros y su contenido se muestran en la siguiente tabla:

Nombre fichero	Contenido
ProductBacklog.X	<i>Product backlog</i> inicial con los requisitos enunciados en forma de <i>Historias de usuario</i> (Figura 3).
SprintBacklog.X	El <i>sprint backlog</i> y el <i>tablero de tareas</i> iniciales del <i>sprint</i> (Figuras 4 y 6) donde están incluidas las tarjetas de <i>historias de usuario</i> (Figura 5).
DailyMeetings.X	Compendio de todas las actas informales de los <i>daily meetings</i> y los <i>tableros de tareas</i> (Figura 6).
Diagramas de quemado.X	Compendio de los diagramas de quemado.
Tablero de tareas.X	Tablero de tareas remanente al final del <i>sprint</i> .
ProductBacklog Completado.X	<i>Product backlog</i> completado.
Sprint Review.X	Acta de la reunión Sprint Review llevada a cabo al final del <i>sprint</i> .
Documento de análisis.X	El resultado del <i>sprint</i> : Documento de análisis del sistema CometelooToo.
Documento de diseño.X	El resultado del <i>sprint</i> : Documento de diseño del sistema CometelooToo.
Documento de Reflexión.X	Documento de Reflexión correspondiente a la realización de esta Práctica N° 1.

Nota: la extensión .X indica formato Word, pdf o Excel.

Un estudiante de cada equipo debe entregar todo el material a través de la plataforma Moodle.



Criterios de Valoración

La práctica se calificará en base a la siguiente rúbrica:

CRITERIO	PUNTUACIÓN MÁXIMA
<i>Product Backlog</i> inicial	1
<i>Sprint Backlog</i> y <i>Tablero de tareas</i> inicial (tarjetas de <i>Historias de usuario</i> y tareas)	1,5
Actas de las <i>daily meetings</i> incluyendo la evolución del <i>Tablero de tareas</i> . <i>Diagramas de quemado</i> . <i>Tablero de tareas</i> remanente. <i>Product Backlog</i> completado.	2,3
Acta del <i>Sprint Review</i>	0,7
Documento de análisis y Documento de diseño	3
Calidad del material	1,5
Total	10

Fecha de Entrega

Todos los Grupos: 9 de marzo 23:55 hs.

La entrega de las prácticas fuera de plazo tendrá la siguiente penalización: -1 punto para el primer día de retraso y -0,5 puntos por cada día de retraso en la entrega a partir del primero. La contabilización de días de retraso se hará de acuerdo a días naturales, es decir todos los días, haya clase o no, días laborables, sábados y festivos, cuentan a la hora de penalizar. Si los días de retraso superan la semana, la práctica se recogerá, pero la nota de dicha práctica será un cero.

Herramienta para Realizar la Práctica

Cualquier procesador de textos, otras herramientas.

Referencias Bibliográficas

- [1] Schwaber, Ken; Beedle, Mike. *Agile software development with Scrum*. Prentice Hall, 2001.
- [2] James O. Coplien, Neil B. Harrison. *Organizational Patterns of Agile Software Development*. Prentice-Hall, 2004.
- [3] Rising, L. and Janoff, N. S. *The Scrum software development process for small teams*. IEEE Software, pages 26–32, 2000.
- [4] <http://scrumtrainingseries.com/>



Escuela Politécnica Superior

Ingeniería Informática | Ingeniería del Software



Retroalimentación y Calificaciones de la Práctica

Las notas y retroalimentación se publicarán, mediante la plataforma Moodle, como máximo 3 semanas después de la entrega de la práctica.



ANEXO A. – APLICACIÓN COMETELOOTO

El sistema software sobre el que se va a aplicar Scrum y, en particular, se va a realizar el *sprint* objeto de esta práctica es un Sistema de Envío de Comida a Domicilio (CometeloToo). El ámbito y la descripción del sistema se presentan a continuación.

A.1. ÁMBITO DEL SISTEMA

CometeloToo, empresa creada por estudiantes emprendedores de la EPS, quiere implementar una aplicación basada en una idea de negocio en la que han estado trabajando, y que pretende cubrir un nicho de mercado centrado en la gestión y reparto, de manera integral, de comida a domicilio.

En la aplicación convivirán cuatro perfiles distintos de usuario: propietarios de restaurantes, repartidores autónomos, clientes del servicio y un administrador único que se encargará de tareas de validación. La aplicación debe permitir a los distintos usuarios registrarse utilizando, inicialmente, una información básica compuesta por usuario, contraseña y nombre completo. Una vez registrados, los usuarios se autenticarán (y podrán introducir más información) mediante un nombre de usuario y una contraseña, excepto para el caso del administrador único, que estará ya creado por defecto con un nombre de usuario y contraseña únicos. Toda la información se guardará/leerá de una base de datos externa a la aplicación a desarrollar.

Los propietarios de los restaurantes, una vez registrados y autenticados, podrán introducir información sobre sus locales (nombre del restaurante, CIF, número de teléfono, domicilio completo y cuenta bancaria de facturación), así como un listado de ítems que ofrecen, y que puede estar compuesto por productos de elaboración propia (identificando su nombre, precio, listado de ingredientes y listado de alérgenos), productos elaborados por proveedores externos (identificando su nombre, precio, y nombre del proveedor), y menús. Los menús estarán compuestos por distintos productos (de elaboración propia o elaborados por proveedores externos) a decidir por los propietarios, así como un precio, que será la suma de los productos que lo integran. Un menú puede incluir a su vez otros sub-menús, y así sucesivamente, compuestos por una serie de agrupaciones de productos por tipología o para un público específico (por ejemplo, bocadillos, vegetariano, celíacos, infantil con diferentes sub-menús por edades, etc.). Los propietarios podrán modificar la información de sus productos y menús en cualquier momento.

Los repartidores autónomos, una vez registrados y autenticados, podrán introducir información sobre su servicio (teléfono móvil, NIF, medio de transporte, precio por kilómetro, franja horaria de trabajo, una cuenta bancaria, así como un listado de códigos postales de las zonas por las que los repartidores están dispuestos a trabajar para realizar los repartos). Una vez comprobada la información por parte del administrador, el repartidor quedará activo en el sistema, y la empresa le proporcionará un GPS que se utilizará dentro de la aplicación a través de una librería externa, y que permitirá calcular el tiempo estimado de entrega de cada pedido y el kilometraje para su posterior pago.

Los clientes del servicio, una vez registrados y autenticados, podrán introducir información personal (número de teléfono, domicilio completo y un número de tarjeta de crédito). A partir de ese momento, los clientes podrán realizar pedidos de comida a domicilio.



Para realizar un pedido, en una fecha y hora determinadas, el cliente confeccionará un listado de productos que quiere pedir, y que podrán provenir de los restaurantes dados de alta en el sistema. El listado podrá incluir productos individuales y menús (incluyendo sub-menús). Una vez confirmado el pedido por parte del cliente, el sistema buscará aquellos repartidores que estén disponibles y de servicio (es decir, que trabajen en la hora predeterminada para el reparto), y dentro de la zona de reparto, teniendo en cuenta que coincida con los códigos postales de los restaurantes elegidos, así como con el código postal del cliente que ha realizado el pedido. Se asignará aquel que esté más cercano al primer restaurante del pedido. Una vez confirmado que existe repartidor disponible, el sistema presentará al cliente el precio final del pedido, que estará compuesto por la suma de sus productos y menús (incluyendo sub-menús). En caso contrario, el pedido quedará en espera hasta que haya repartidores disponibles. Una vez aceptado el precio, se procederá al cobro del servicio a través de una pasarela externa de pagos/cobros. Una vez realizada la transacción de manera correcta, se informará al usuario y se le dará un tiempo estimado de espera para el reparto. En caso contrario, se informará del problema con el cobro y el pedido se cancelará. Cuando el pedido haya sido entregado (lo cual será marcado por los clientes en su aplicación), el pedido quedará cerrado. Un repartidor sólo podrá atender a un pedido a la vez.

El primer día de cada mes, la empresa llevará a cabo un proceso de liquidación a partir de la totalidad de pedidos realizados desde comienzos hasta finales del mes anterior. Para ello, cobrará a cada restaurante un 30% sobre el total de productos y menús (con sub-menús) incluidos en los pedidos. En el mismo proceso, la empresa realizará el pago a los repartidores involucrados en función del kilometraje total realizado en ese mismo intervalo de tiempo, en función del precio por kilómetro estipulado por cada uno. La liquidación se realizará mediante la pasarela externa de pagos/cobros.

Cualquier incidencia en la transacción de cobro será remitida a los restaurantes para que actualicen la información de su cuenta bancaria.

A.2. CASOS DE USO POSIBLES A ESPECIFICAR POR LOS ESTUDIANTES

Los casos de uso más relevantes para esta aplicación (Nótese que simplemente se da una referencia y que los estudiantes podrán añadir, quitar, fusionar o dividir según su criterio. De igual manera nótese que la relación entre Caso de Uso e Historia de Usuario no es necesariamente uno a uno, de manera que un Caso de Uso podría estar en varias Historias de Usuario o una Historia de Usuario puede contener varios Casos de Uso) son:

Caso de Uso 1: Registrarse en el Sistema

Caso de Uso 2: Entrar en el Sistema

Caso de Uso 3: Crear Usuario

Caso de Uso 4: Registrar Restaurante

Caso de Uso 5: Registrar Plato



Caso de Uso 6: Registrar Menú

Caso de Uso 7: Modificar Datos Restaurante

Caso de Uso 8: Alta Repartidor

Caso de Uso 9: Validar Datos de Repartidor

Caso de Uso 10: Realizar Pedido

Caso de Uso 11: Cobrar Pedidos

Caso de Uso 12: Liquidación Restaurantes

Caso de Uso 13: Liquidación Repartidores

A.3. CASOS DE USO OBLIGATORIOS Y MÍNIMOS A INCLUIR EN LA PRÁCTICA

De estos casos anteriores, aquellos que deberá implementar cada equipo, como mínimo, son los listados a continuación. Asimismo, se deberán crear los correspondientes diagramas exigidos para cada uno de estos tres casos.

- Caso de Uso 5: Registrar Plato - Valor para el cliente: 5 sobre 5
- Caso de Uso 10: Realizar Pedido - Valor para el cliente: 5 sobre 5
- Caso de Uso 12: Liquidación Restaurantes - Valor para el cliente: 5 sobre 5

Si se desean llevar a cabo un mayor número de casos de uso, el equipo podrá elegir aquellos que considere y le proporcionará un valor que estime oportuno.



ANEXO B. - ARTEFACTOS

B.1. PRODUCT BACKLOG

Representa la definición del proyecto a realizar. Se divide en objetivos expresados como *historias de usuario*, cada una aportando valor de negocio incremental e individual. Una historia es un requisito de negocio visto desde el punto de vista de un usuario. Se escriben con el siguiente formato:

"Como [Rol], quiero hacer [descripción de la funcionalidad], con el objetivo de [Razón o Resultado]",

donde, [Rol] es el tipo de Usuario (quien), [descripción de la funcionalidad] es lo que el sistema debe permitir realizar (el qué) y [Razón o Resultado] es el beneficio o valor buscado (el por qué). Así, las *historias de usuario* se enfocan en definir lo que el usuario necesita hacer, sin describir el cómo y cada una debe proporcionar algún valor al cliente.

Además, deben incluir los campos:

- valor para el cliente: es el beneficio o retorno que el cliente recibirá tras la ejecución completa del requisito u objetivo. Para esta práctica se toman los valores de 1 a 5 siendo 1 la calificación mínima y 5 la máxima. Este dato viene dado por el cliente (en el caso de la práctica se indica en el enunciado).
- estimación: es el esfuerzo que tiene que realizar el equipo para completar el requisito u objetivo. Viene dado en persona-día, y la realiza el equipo de desarrollo basándose en la dificultad de las tareas necesarias para llevar a cabo el objetivo.

Teniendo en cuenta el valor y la estimación, el equipo debe priorizar los requisitos que se llevarán a cabo en el sprint por orden de ejecución. Una vez priorizados los requisitos, el equipo deberá seleccionar aquellos que, por tiempo y recursos, se podrán llevar a cabo y completar en el presente *sprint*. Tanto la priorización como la selección deben quedar debidamente justificada en la documentación.

En la Figura 3 se proporciona un ejemplo de formato de *Product Backlog*.



Identificador (ID) de la Historia	Enunciado de la Historia	Valor para el cliente	Esfuerzo en p-d	Iteración (Sprint)	Prioridad	Comentarios
XXX-XXXX	Como [Rol], quiero hacer [descripción de la funcionalidad], con el objetivo de [Razón o Resultado]					
XXX-XXXX	Como [Rol], quiero hacer [descripción de la funcionalidad], con el objetivo de [Razón o Resultado]					

Figura 3: Ejemplo de formato para el *Product backlog*

B.2. SPRINT BACKLOG

Se compone de las *historias de usuario* seleccionadas para ser realizadas en el presente *sprint* más las tareas que debe desarrollar el equipo necesarias para poder completar y cumplir con estas *historias*. Las *historias de usuario* elegidas han sido acordadas con el cliente y pueden ser demostradas al final de la iteración en forma de incremento de producto.

Para cada uno de los requisitos se muestran sus tareas y el esfuerzo (persona-día) necesario para finalizarlas. El equipo se auto-assigna cada tarea. Para efectos de la práctica bastará con incluir un número que indique la cantidad de personas que trabajan ese día en esa tarea.

En la Figura 4 se proporciona un ejemplo de formato de *Sprint Backlog*.

Identificador (ID) de la Historia	Enunciado de la Historia	Tareas	Día 1	Día 2	Día 3	Día 4	Día 5	Día 6	Día 7	..
XXX-XXXX	Como un [Rol], necesito [descripción de la funcionalidad], con la finalidad de [Razón o Resultado]	Tarea 1								
		Tarea 2								
		Tarea 3								
		Tarea n								
XXX-XXXX	Como un [Rol], necesito [descripción de la funcionalidad], con la finalidad de [Razón o Resultado]	Tarea 1								
		Tarea 2								
		Tarea 3								
		Tarea m								

Figura 4: Ejemplo de formato para el *Sprint backlog*



B.3. HISTORIAS DE USUARIO Y TABLERO DE TAREAS

A partir del *sprint backlog* se recomienda implementar un *tablero de tareas* o *scrum taskboard* que permita gestionar la lista de *historias de usuario* a completar en la iteración de manera eficaz. Las *historias de usuario* se pasan a formato tarjeta y después se colocan en una columna a la izquierda del tablón. Al lado de cada *historia*, en fila, se ponen las tareas necesarias para completarlo, en forma de post-its, que se van moviendo hacia la derecha para cambiarlas de estado (pendientes de iniciar, en desarrollo, en revisión, terminadas). Es útil usar post-it de distintos colores o con algún distintivo para cada miembro del equipo, de manera que se pueda ver en qué tareas está trabajando cada cual.

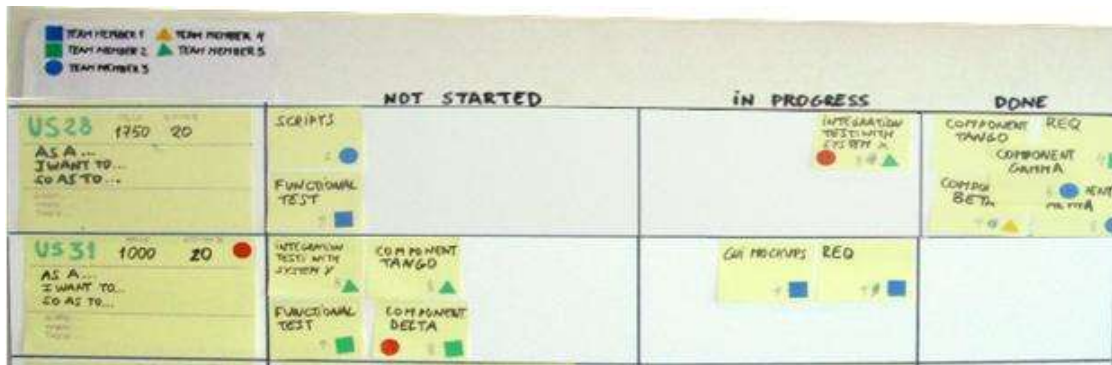
Este tablero permite ver las tareas donde el equipo está teniendo problemas y no avanza, con lo que le permite tomar decisiones al respecto.

En la Figura 5 se proporciona un ejemplo de formato de tarjetas de *historias de usuario*.

Historia de usuario	
Identificador:	Nombre de historia:
Usuario:	
Estimación:	Iteración asignada:
Desarrollador responsable:	
Descripción:	
Observaciones:	

Figura 5: Ejemplo de formato para las tarjetas de las Historias de usuario

En la Figura 6 se proporciona un ejemplo de *tablero de tareas* y los campos mínimos que debe tener la matriz del tablero.



Historia de usuario	Pendiente	En desarrollo	En revisión	Terminadas

Figura 6: Ejemplo de *tablero de tareas* y campos del mismo para la práctica

B.4. DIAGRAMA DE QUEMADO

El diagrama de quemado o *burndown chart* se utiliza para saber de forma visual cuánto le falta al equipo para terminar las historias comprometidas en un sprint en curso, es decir, representa el trabajo pendiente. De esta forma, muestra el avance del equipo respecto a la totalidad del proyecto.

En la práctica es un diagrama de dos ejes. En el eje horizontal se muestra el tiempo en días hábiles de duración del sprint (o iteración). En el eje vertical se muestra el número de tareas que representan el trabajo remanente.

Puedes utilizar <http://www.burndowngenerator.com/>, cualquier otro generador de diagramas de quemado que facilite la tarea o realizarlo sin ninguna herramienta específica.



ANEXO C. - REUNIONES

C.1. SPRINT PLANNING

En esta reunión el equipo planifica la iteración y elabora la táctica que le permitirá conseguir el mejor resultado posible con el mínimo esfuerzo. Esta actividad la realiza el equipo dado que ha adquirido un compromiso y es, por tanto, el responsable de organizar su trabajo y quien mejor conoce cómo realizarlo. En particular, en esta reunión el equipo debe:

- Seleccionar los requisitos, a partir del *Product backlog*, que se llevarán a cabo en este *sprint* en función de sus prioridades. Pasar las *historias de usuario* de dichos requisitos a formato tarjeta (Figura 5).
- Definir las tareas necesarias para poder completar cada requisito o *historia de usuario*, detallando el esfuerzo que conllevará cada una.
- Plantear el *tablero de tareas* (Figura 6) inicial.
- Auto-asignarse las tareas que cada miembro del equipo puede realizar.

La salida de esta reunión es el *sprint backlog* y el *tablero de tareas* (explicados en el Anexo B) donde se moverán las tareas a lo largo del *sprint* cada vez que cambien de estado.

C.2. DAILY MEETING

Ésta es una reunión dinámica y corta (**no más de 10 minutos**) dirigida por el *Scrum Master* cuyo objetivo es facilitar la transferencia de información y la colaboración entre los miembros del equipo para aumentar su productividad, poniendo de manifiesto puntos en que se puedan ayudar unos a otros.

Cada miembro del equipo inspecciona el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para, al finalizar la reunión, poder hacer las adaptaciones necesarias que permitan cumplir con el compromiso conjunto que el equipo adquirió para la iteración en el *sprint planning*.



Cada miembro del equipo debe responder las siguientes preguntas:

- ¿Qué hice ayer que ayudó al equipo de desarrollo a lograr el objetivo del *Sprint*?
- ¿Qué haré hoy para ayudar al equipo de desarrollo a lograr el objetivo del *Sprint*?
- ¿Veo algún impedimento que evite que el equipo de desarrollo o yo logremos el objetivo del *Sprint*?

Como apoyo, el equipo cuenta con el *tablero de tareas* y el *diagrama de quemado*, donde **en cada reunión** se actualiza el estado de cada tarea en el tablero y comprueba si es necesario realizar alguna reasignación. Así mismo, el diagrama de quemado debe ser correspondientemente actualizado.

El acta informal que se pide para la práctica asociada a estas reuniones debe incluir las respuestas de los miembros del equipo a las tres preguntas anteriormente enumeradas y el tablero para que se vean los cambios de estados que se han producido en las tareas. Para facilitar esta tarea, en cada acta se pueden incluir fotos del *tablero de tareas* para presentar la evolución de las tareas. Esto es sólo una sugerencia, cualquier otra forma de incluir el tablero de tareas en las actas, lo que es obligatorio, queda a elección del equipo. En la Figura 7 se ofrece un ejemplo de formato de estas actas.



Daily meeting DD-MM-AAA

Lugar:

Participantes y roles:

Contestación de cada miembro del equipo a las preguntas:

- ¿Qué hice ayer que ayudó al equipo de desarrollo a lograr el objetivo del *Sprint*?
- ¿Qué haré hoy para ayudar al equipo de desarrollo a lograr el objetivo del *Sprint*?
- ¿Veo algún impedimento que evite que el equipo de desarrollo o yo logremos el objetivo del *Sprint*?

Tablero de tareas con el estado actual:

Comentarios:

Figura

7: Formato de actas de los *Daily Meetings*

Nota: Estas reuniones deben ser diarias, según SCRUM, pero para efectos de esta práctica, y atendiendo a motivos de carga de trabajo y de disponibilidad de los estudiantes para reunirse, se llevarán a cabo al menos en los días de clase de prácticas de la segunda a la cuarta semana mientras dure el *sprint*. En cada *Daily Meeting*, excepto el primero, en el tablero de tareas se actualizarán las tareas desde el último *daily meeting* y se asignarán las tareas hasta el siguiente *daily meeting*.



C.3. SPRINT REVIEW

Esta reunión, convocada por el *Scrum Master*, proporciona un punto de inspección para conocer y evaluar el progreso del proyecto al final de cada *sprint*. Basándose en esta inspección, se identifica qué ha ido bien, qué ha ido mal y qué pasos seguir a continuación.

El propósito del *sprint review* es que el equipo presente al *Product Owner* y al resto de *stakeholders* (clientes y usuarios principalmente) las funcionalidades o requisitos que se han completado durante el *sprint*. Aquellas funcionalidades o requisitos que no están completos no pueden ser presentados. En el caso de esta práctica, el equipo de desarrollo y el *Product Owner* se reunirán y decidirán lo mencionado anteriormente.

El *sprint review* comienza con un miembro del equipo presentando las metas del *sprint*, el *product backlog* comprometido y el *product backlog* completado. Todos los miembros del equipo pueden comentar los aspectos que fueron o no satisfactorios durante el *sprint*.

El acta a entregar con los resultados de esta reunión debe incluir, al menos, los requisitos que se han satisfecho, los que no y los que quedan pendientes para el próximo *sprint*.

Nota: La reunión *sprint retrospective*, que daría lugar al inicio del siguiente *sprint*, queda fuera del alcance de esta práctica.