# 2 MARCO TEORICO METODOLOGIA AGIL

## 2.1 Propósito de Scrum

Scrum es un proceso en el que se aplican de manera regular [un conjunto de buenas prácticas](https://proyectosagiles.org/fundamentos-de-scrum) para **trabajar colaborativamente, en equipo**, y obtener [el mejor resultado posible](https://proyectosagiles.org/beneficios-de-scrum) de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un [estudio de la manera de trabajar de equipos altamente productivos](https://proyectosagiles.org/historia-de-scrum).

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en **entornos complejos**, donde se necesita **obtener resultados pronto**, donde los **requisitos son cambiantes o poco definidos**, donde la **innovación**, la **competitividad**, la **flexibilidad** y la **productividad**son fundamentales.

Scrum también se utiliza para resolver situaciones en que **no se está entregando al cliente lo que necesita**, cuando **las entregas se alargan demasiado, los costes se disparan**o**la calidad no es aceptable**, cuando se necesita **capacidad de reacción ante la competencia**, cuando **la moral de los equipos es baja y la rotación alta**, cuando es necesario **identificar y solucionar ineficiencias sistemáticamente** o cuando se quiere trabajar utilizando un **proceso especializado en el desarrollo de producto**.

## 2.2 Visión General de Scrum

Un marco de trabajo por el cual las personas pueden acometer problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible productiva y creativamente. Scrum es:

* Ligero
* Fácil de entender
* Extremadamente difícil de llegar a dominar

Scrum es un marco de trabajo de procesos que ha sido usado para gestionar el desarrollo de productos complejos desde principios de los años 90. Scrum no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varias técnicas y procesos. Scrum muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo, de modo que podamos mejorar.

El marco de trabajo Scrum consiste en los Equipos Scrum, roles, eventos, artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de Scrum y para su uso.

Las reglas de Scrum relacionan los eventos, roles y artefactos, gobernando las relaciones e interacciones entre ellos. Las reglas de Scrum se describen en el presente documento.

Las estrategias específicas para usar el marco de trabajo Scrum son diversas y están descritas en otros lugares.

## 2.3 El Equipo Scrum (Scrum Team)

El Equipo de Scrum está comprendido por:

### **2.3.1 Product Owner (dueño del producto):**

A veces es el mismo cliente. En otros casos, especialmente cuando se trata de proyectos complejos, actúa como su representante directo, el de los usuarios del producto y, en general, el de todas aquellas partes que tengan algún interés en él. Es el único con la potestad para decidir las funcionalidades y características del producto.

Tiene un diálogo directo y permanente con el Scrum Máster, que es su nexo directo con quienes ejecutan las labores. Sólo entra en contacto con el Scrum Team al final de cada una de las iteraciones para evaluar las entregas parciales.

### **2.3.2 Scrum Máster (director o figura visible del proyecto):**

Es el encargado de garantizar que el proceso cumplirá con las directrices del modelo Scrum. Muchos lo denominan líder de proyecto, pero en realidad es mucho más que eso. Es el encargado de mantener una visión global del mismo y de emplearse a fondo ante cualquier circunstancia, sea la que sea. Además, fluctúa entre el plano práctico y el plano directivo; es decir, interactúa de igual manera con el Product Owner y con los integrantes del Scrum Team que están a su cargo.

### **2.3.3 Scrum Team (equipo de trabajo):**

Hace referencia al grupo de personas que ejecuta las tareas propuestas. Aquí entran tanto los arquitectos, ingenieros, programadores, diseñadores y demás profesionales como las personas que realizan labores administrativas.

Es posible que dentro del Scrum Team surja algún líder o primer responsable; cuando no es así, esta labor la asume el [Scrum Máster](https://www.obs-edu.com/blog-project-management/scrum/como-conseguir-la-certificacion-scrum-master/). En cualquier caso, es importante que sus integrantes definan los roles de equipo. Su relación con el Product Owner se reduce a la presentación de los resultados de cada iteración.

## 2.4 El Dueño del Producto (Product Owner)

Las responsabilidades del Cliente / Product Owner (que puede ser interno o externo a la organización que realiza el desarrollo del producto) son:

* Conocer el mercado y los comportamientos de los clientes / usuarios finales, con muy buena visión de Negocio.
* Ser el representante de todas las personas interesadas (stakeholders) para conseguir una buena definición de los objetivos del producto o proyecto y de los resultados esperados.
* Las “personas interesadas” pueden ser internas o externas a la organización: promotores del proyecto y usuarios finales.
* Idealmente el Product Owner también debería ser algún tipo de usuario, un consumidor final del producto, para poder experimentar directamente si se consiguen los beneficios que hipotéticamente le debería aportar.
* Encargarse de que exista una priorización clara del trabajo a hacer. De este modo, dirige los resultados del proyecto para maximizar su ROI (*Return Of Investment*):
* Es el propietario de la planificación del proyecto: crear y mantener la [lista priorizada con los requisitos](https://proyectosagiles.org/cliente-product-owner/lista-requisitos-priorizada-product-backlog) necesarios para cubrir los objetivos del producto o proyecto, conocer el valor que aportará cada requisito y calcula el [ROI](https://proyectosagiles.org/cliente-product-owner/priorizacion-requisitos-valor-coste) a partir del coste de cada requisito (estimación que le proporciona el [equipo](https://proyectosagiles.org/cliente-product-owner/equipo-team)).
* Repartir los objetivos/requisitos en iteraciones, de manera que prioriza tanto el proporcionar beneficios al usuario final como reducir riesgos respecto hipótesis) y establece un calendario de entregas.
* Antes de iniciar cada iteración, [replanificar el proyecto](https://proyectosagiles.org/cliente-product-owner/replanificacion-proyecto) en función de los requisitos que aportan más valor en ese momento, de los requisitos completados en la iteración anterior y del contexto del proyecto en ese momento (demandas del mercado, movimientos de la competencia, etc.).
* Actuar como interlocutor único ante el equipo, con autoridad para tomar decisiones(es decir, con libertad para decidir qué hacer de manera alineada con los objetivos de su área de trabajo, que no se vea invalidada por personas en una posición superior).

### 2.4.1 Colaborar con el equipo para planificar, revisar y dar detalle a los objetivos de cada iteración:

* Participar en la [reunión de planificación de iteración](https://proyectosagiles.org/cliente-product-owner/planificacion-iteracion-sprint-planning), proponiendo los requisitos más prioritarios a desarrollar, respondiendo a las dudas del equipo y detallando los requisitos que el equipo se comprometer a hacer.
* Estar disponible durante el transcurso de la iteración para responder a las preguntas que puedan aparecer.
* [No cambiar los requisitos que se están desarrollando en una iteración](https://proyectosagiles.org/cliente-product-owner/ejecucion-iteracion-sprint), una vez está iniciada.
* Participar en la [reunión de demostración de la iteración](https://proyectosagiles.org/demostracion-requisitos-sprint-review), revisando los requisitos completados.
* Proporcionar suficiente dedicación para hacer este trabajo (usualmente un Product Owner no suele poder estar con más de 1-2 equipos, en función del tipo de producto y equipos).
* Notar que una distancia de 3-5 metros del equipo puede llegar a ser una barrera demasiado alta para interactuar de manera ágil con el equipo (a menos que se encuentre físicamente en otra ciudad). Sería difícil de justificar que el Product Owner esté situado en la misma ciudad y no esté la mayor parte del tiempo sentado con sus equipos.

Por ello**, una buena selección de Product Owners (y una esfuerzo especial en su coaching de Producto/Cliente) es fundamental para un inicio de transformación ágil exitoso**, es decir, aportar resultados a la empresa de manera más ágil.

## 2.5 El Equipo de Desarrollo (Development Team)

El equipo en Agile incluye al [Cliente / Product Owner](https://proyectosagiles.org/cliente-product-owner/) y al [Facilitador / Scrum Master](https://proyectosagiles.org/facilitador-scrum-master/). Cuando se habla específicamente de “equipo de desarrollo” se refiere al conjunto de personas más “técnicas” que de manera conjunta desarrollan el producto del proyecto. Tienen un objetivo común, comparten la responsabilidad del trabajo que realizan (así como de su [calidad](https://proyectosagiles.org/calidad-agilidad-cuarto-encuentro-agil-barcelona)) en cada [iteración](https://proyectosagiles.org/desarrollo-iterativo-incremental) y en el proyecto.

Para ir alcanzando un estado de alto rendimiento, este equipo es:

### 2.5.1 Multidisciplinar.

* + Agile se basa en ir desarrollando un producto “tangible” de forma iterativa para obtener feedback en intervalos cortos de tiempo sobre si el producto genera en el cliente el resultado esperado. Para ello, idealmente el equipo no debe depender de otros pues generaría esperas y retrasos en la entrega de valor (necesidad de coordinación, integración e incluso falta de *ownership*), con lo cual sólo se puede hacer con equipos multidisciplinares y autónomos del resto de la organización. De este modo:
    - Los miembros del equipo disponen de las habilidades necesarias para poder identificar y ejecutar todas las tareas que permiten proporcionar al cliente los requisitos previstos en la iteración.
    - Tienen que depender lo mínimo de personas externas al equipo, de manera que no se ponga en peligro la previsión del trabajo a revisar al final de cada iteración.
    - Se crea una sinergia que permite que el resultado sea más rico al nutrirse de las diferentes experiencias, conocimientos y habilidades de todos, “colaboración creativa”.

El equipo de desarrollo ágil en ocasiones es llamado Squad al englobar, con una visión más amplia, cualquier skill o rol necesario para incluir toda la cadena de valor dentro del equipo, de modo que todos sus miembros están guiados por el mismo propósito, conocen al cliente, el dominio y definen conjuntamente problema a solucionar. En función de las necesidades para desarrollar el producto, el Squad puede incluir a desarrolladores, UX, Ops, Data, … y, por supuesto, al Product Owner y al Scrum Master.

### 2.5.2 Tamaño de equipo de 5 a 9 personas.

* + Para ser ágil, el tamaño del equipo está entre 5 y 9 personas.
    - Por debajo de 5 personas cualquier imprevisto o interrupción sobre un miembro del equipo puede comprometer seriamente la previsión de objetivos a mostrar al [cliente / Product Owner](https://proyectosagiles.org/cliente-product-owner) al finalizar la iteración.
    - Por encima de 9 personas, la comunicación y colaboración real entre todos los miembros se hace más difícil y se acaban formando subgrupos, donde todo el mundo no está interesado por los mismos objetivos del Sprint (ver los [requisitos de Scrum](https://proyectosagiles.org/requisitos-de-scrum)).
  + De cualquier manera, se puede trabajar en Scrum con 3 personas y se ha utilizado en proyectos con centenares de personas, agrupadas en varios equipos. Cuando es necesario que más de un equipo trabaje de manera ágil en un mismo producto/proyecto, existen diferentes técnicas que permiten esta colaboración (sincronización regular entre Product Owners; Scrum de Scrums entre miembros de los equipos; ajustes de las reuniones de Scrum para hacer algunas más generales/comunes y poder gestionar dependencias conjuntas [como si se tratase de un fractal de reuniones, ver [LESS](https://less.works/)], siempre completando incrementos de producto de manera regular. De cualquier modo, hay que recordar que el objetivo final es conseguir que cada equipo sea el máximo de independiente de los otros para realmente poder ágiles.

### 2.5.4 Estable y dedicado.

* + Los equipos ágiles trabajan en iniciativas estratégicas (que se mantienen en el tiempo) o bien van repitiendo tipos de proyectos similares (van ajustando su misión como equipo). Un equipo de alto rendimiento cuesta tiempo conseguirlo (y después hay que aprovechar esa *performance*), es un activo fundamental para la empresa y su *building block* básico, con lo cual prevalecerá el interés por mantenerlo por encima de desmantelarlo.
  + Dado que el equipo debe ser estable, sus miembros deben cambiar lo mínimo posible, para poder aprovechar el esfuerzo que les ha costado construir sus relaciones interpersonales, engranarse y establecer su organización del trabajo.
  + Los miembros del equipo están dedicados a tiempo completo a esa iniciativa o proyecto, para evitar dañar su productividad por cambios de tareas en diferentes proyectos, para evitar interrupciones externas, para trabajar todos en las mismas prioridades y así poder mantener la previsión de lo que se podría entregar en cada iteración.

### 2.5.5 Se auto-organizan y piensan juntos:

* + Para crear sinergias con qué conseguir un resultado mejor en menos tiempo, en términos de valor para cliente.
  + Para mejorar su proceso de trabajo y el sistema que les rodea, reflexionando regularmente (por ejemplo haciendo [Retrospectivas](https://proyectosagiles.org/retrospectiva-sprint-retrospective/)) e introduciendo acciones de mejora en su backlog de tareas a realizar.
  + La manera de conseguir esto se basa en compartir información y que sus miembros confíen entre ellos. Para ello, realiza de manera conjunta las siguientes actividades:
    - Seleccionar los requisitos que prevén completar en una iteración, de forma que estén preparados para ser entregados al cliente.
    - Estimar la complejidad de cada requisito en la [lista de requisitos priorizada](https://proyectosagiles.org/lista-requisitos-priorizada-product-backlog) del producto o proyecto.
    - En la [reunión de planificación de la iteración](https://proyectosagiles.org/planificacion-iteracion-sprint-planning/), decidir cómo va a realizar su trabajo:
      * Seleccionar los requisitos que pueden completar en cada iteración, realizando al cliente las preguntas necesarias.
      * Identificar todas las tareas necesarias para completar cada requisito.
      * Estimar el esfuerzo necesario para realizar cada tarea.
      * Cada miembro del equipo se autoasigna a las tareas.
    - Durante la iteración, trabajar de manera conjunta para conseguir los objetivos de la iteración. Cada especialista lidera el trabajo en su área y el resto colaboran si es necesario para poder completar un requisito.
    - Al finalizar la iteración:
      * [Demostrar](https://proyectosagiles.org/demostracion-requisitos-sprint-review) al cliente los requisitos completados en cada iteración.
      * Hacer una [retrospectiva](https://proyectosagiles.org/retrospectiva-sprint-retrospective) la final de cada iteración para mejorar de forma continua su manera de trabajar.

### 2.5.6 Estar sentados juntos.

* + Todos los miembros del equipo trabajan en la misma localización física, para poder maximizar la comunicación entre ellos mediante conversaciones cara a cara, diagramas en pizarras blancas, etc.
    - De esta manera se minimizan otros canales de comunicación menos eficientes, que hacen que las tareas se transformen en un “pasa pelota” o que hacen perder el tiempo en el establecimiento de la comunicación (como cuando se llama repetidas veces por teléfono cuando la persona no está en su puesto).
    - Notar que una distancia de 3-5 metros entre miembros de un equipo puede llegar a ser una barrera demasiado alta para favorecer una relación realmente ágil entre ellos, encontrar soluciones juntos rápidamente y resolver impedimentos. Es decir, a partir de esa distancia (o si las personas están en otros pisos o edificios de una ciudad) los efectos pueden ser bastante contraproducentes, equivalentes a tener un equipo distribuido.
  + En el caso de equipos remotos, que no trabajen en la misma ciudad, es conveniente garantizar un tiempo de calidad cara a cara, como mínimo:
    - 1-2 semanas para el [inception inicial del proyecto](https://proyectosagiles.org/2009/06/08/introduccion-estimacion-planificacion-agil/) e integración “social” de los miembros del equipo.
    - 2 o más días por iteración, para actividades como [Refinements](https://proyectosagiles.org/replanificacion-proyecto/), [Retrospectivas](https://proyectosagiles.org/retrospectiva-sprint-retrospective) o [Planificaciones de iteración](https://proyectosagiles.org/planificacion-iteracion-sprint-planning).

### 2.5.7 Interaccionar con Stakeholders.

* + Este equipo interacciona frecuentemente con los Stakeholders para compartir conocimiento de negocio, alinear prioridades y medir los resultados del valor entregado al cliente.

## 2.6 El Scrum Master

Su principal misión es conseguir un equipo de alto rendimiento (incluyendo al [Cliente / Product Owner](https://proyectosagiles.org/cliente-product-owner/) y a las relaciones con la organización y stakeholders). Se encarga de conseguir el equipo que conozca y sienta los principios y valores de Agile, así como la teoría y prácticas de Scrum, con el objetivo de que los usen en sus procesos de toma de decisiones. El Scrum Master actúa como facilitador de reuniones donde pensar de manera conjunta y quita los obstáculos más allá del equipo que le impiden ser ágil.

De este modo, es el coach y lider al servicio del [equipo](https://proyectosagiles.org/equipo-team),  llevando a cabo las siguientes responsabilidades:

* Velar por que todos los participantes del proyecto sigan los [valores y principios ágiles](https://proyectosagiles.org/historia-de-scrum" \l "manifiesto-agil), las[reglas y proceso de Scrum](https://proyectosagiles.org/como-funciona-scrum) y guiar la colaboración intraequipo y con el cliente / Product Owner) de manera que las sinergias sean máximas. Esto implica:
  + Asegurar que exista una [lista de requisitos priorizada](https://proyectosagiles.org/lista-requisitos-priorizada-product-backlog) y que esté [preparada](https://proyectosagiles.org/replanificacion-proyecto) antes de la siguiente iteración.
  + Facilitar las reuniones de Scrum ([planificación de la iteración](https://proyectosagiles.org/planificacion-iteracion-sprint-planning), [reuniones diarias de sincronización del equipo](https://proyectosagiles.org/reunion-diaria-de-sincronizacion-scrum-daily-meeting), [demostración](https://proyectosagiles.org/demostracion-requisitos-sprint-review), [retrospectiva](https://proyectosagiles.org/retrospectiva-sprint-retrospective)), de manera que sean productivas y consigan sus objetivos.
  + Enseñar al equipo a autogestionarse. No da respuestas, si no que guía al equipo con preguntas para que descubra por sí mismo una solución.
* Quitar impedimentos que el equipo tiene en su camino para conseguir el objetivo de cada iteración (proporcionar un resultado útil al [cliente](https://proyectosagiles.org/cliente-product-owner) de la manera más efectiva) y poder finalizar el proyecto con éxito. Estos obstáculos se identifican de manera sistemática en las [reuniones diarias de sincronización del equipo](https://proyectosagiles.org/reunion-diaria-de-sincronizacion-scrum-daily-meeting) y en las reuniones de [retrospectiva](https://proyectosagiles.org/retrospectiva-sprint-retrospective).
* Proteger y aislar al equipo de interrupciones externas durante la [ejecución de la iteración](https://proyectosagiles.org/ejecucion-iteracion-sprint)(introducción de nuevos requisitos, “secuestro” no previsto de un miembro del equipo, etc.). De esta manera, el equipo puede mantener su productividady el compromiso que adquirió sobre los requisitos que completaría en la iteración [notar, sin embargo, que el equipo debe reservar tiempo para colaborar con al cliente en la [preparación de la lista de requisitos para la próxima iteración](https://proyectosagiles.org/replanificacion-proyecto)].
* Trabajar con otros Scrum Masters y el equipo de transformación / mejora continua con el objetivo de que la organización sea cada vez más ágil.

Notar que una distancia de 3-5 metros del equipo puede llegar a ser una barrera demasiado alta para poder descubrir los comportamientos y tipos de relaciones que hay entre sus miembros, y los impedimentos que les están dificultando el día a día.

## 2.7 Eventos de Scrum

Los eventos se usan en Scrum para crear un patrón constante y minimizar la necesidad de reuniones no definidas.

Todos los eventos son “time boxeados”, es decir, tienen una duración máxima, pudiendo acabarse, siempre que se logre el objetivo del evento.

Existen 5 eventos diferentes:

1. Sprint
2. Sprint Planning, o Planificación del Sprint en castellano
3. Daily Scrum, normalmente llamada Daily
4. Sprint Review, o Revisión del Sprint
5. Sprint Retrospective, o Retrospectiva del Sprint

### 2.7.1 EL SPRINT

El Sprint es la base del Scrum. Es un periodo de tiempo de 1 mes, 3 o 2 semanas, durante el cual **se crea un incremento de producto, utilizable y potencialmente liberable**. Lo ideal es que siempre tengan la misma duración.

Para entender el párrafo anterior, debemos ver cada Sprint como un proyecto con un horizonte no mayor a un mes. Y**al igual que un proyecto, el Sprint se utiliza para lograr algo**. Cada Sprint tiene una definición de lo que se va a desarrollar, un diseño y un plan flexible que guiará su construcción, el trabajo y el producto resultante.

Un **sprint está formado** por el Sprint Planning, las Daily Scrums, el trabajo de desarrollo, Sprint Review y Sprint Retrospective.

La **limitación máxima de un mes es para minimizar riesgo y desperdicio**. Si es más tiempo, en el mundo tan cambiante en el que estamos, la definición de lo que se está construyendo puede quedarse obsoleta, aumentando además la complejidad de lo creado.

Gracias a su uso, se garantiza que, en **2, 3 o 4 semanas, se va a realizar una inspección y una adaptación del progreso hacia el objetivo**. Es decir, si el equipo está desarrollando algo que luego no funciona o se ha quedado obsoleto, sólo se han empleado en ello los recursos de un mes.

En otra entrada veremos cuando se puede cancelar un sprint.

### 2.7.2 SPRINT PLANNING

El trabajo que se realizará en el Sprint está planificado en Sprint Planning, mediante la **colaboración de todo el equipo de Scrum.**

La duración del Sprint planning debe durar como máximo 8 horas para un Sprint de un mes, y si el Sprint es más corto, el evento suele ser más corto.

El **Scrum Master debe asegurarse de que el evento tenga lugar y que los asistentes entiendan su propósito,** asi como que no excedad su duración.

El Sprint Planning se realiza para responder a las siguientes preguntas de las que hablaré en profundidad en otro post:

1. ¿Cual va a ser el incremento de valor del próximo Sprint (Sprint Goal)?
2. ¿Cómo se logrará el trabajo necesario para lograrlo?

### 2.7.3 DAILY SCRUM

Es un **evento que se celebra cada día, de duración de 15 minutos,** en el que el Development Team sincroniza las actividades y crea un plan para las próximas 24 horas. Lo ideal es que se haga siempre a la misma hora y en el mismo lugar para que todo el mundo lo recuerde de forma fácil

La Daily Scrum sirve para**inspeccionar el progreso hacia el Sprint Goal** y cómo avanza el progreso. Para ello, se inspecciona el trabajo realizado en el último día y se comenta el trabajo que se va a realizar ese día.

### 2.7.4 SPRINT REVIEW

La Sprint Review (Revisión del Sprint) se lleva a cabo al final del Sprint con el objetivo de inspeccionar el incremento y adaptar el Product Backlog si es necesario. El **Sprint Team y los stakeholders revisan lo que se ha hecho en el sprint** y en función de eso y de cualquier cambio producido en el Product Backlog, se decide cuales son las **siguientes acciones para optimizar el valor.**

Es una reunión de 4 horas de duración máxima para Sprint de un mes, asegurándose el Scrum Master de que tiene lugar y que los asistentes entiendan su propósito. Es el Product Owner el que inivita a los stakeholders que cree conveniente a participar.

Lo que se debe obtener al final de un Sprint Review es un **Product Backlog revisado y priorizado con los elementos más importantes a trabajar durante el próximo Sprint.**

### 2.7.5 SPRINT RETROSPECTIVE

Este evento es una oportunidad para que el Scrum Team se inspeccione a sí mismo y cree un plan para mejorar en aquellas áreas que lo necesiten. Por ello **se celebra después del Sprint Review y antes del Sprint Planning.**

El Scrum Master se asegura que el evento de una duración de máximo 3 horas para Sprints de 1 mes de duración, tenga lugar y que los asistentes entiendan su propósito, participando como un miembro más del equipo.

La tarea de Scrum Master es animar al Scrum Team a mejorar su proceso de desarrollo, buscando prácticas para hacerlo más efectivo y agradable para el próximo Sprint, y a aumentar la calidad del producto al adaptar la definición de “Done” según corresponda.

Al final de la Retro, el Scrum Team debe haber identificado las mejoras a implementar en el próximo Sprint, por ello normalmente se suelen poner 3 columnas, las cosas positivas realizadas que se deben mantener haciendo, las negativas, y las acciones a mejorarlas con un responsable que se ocupe de que se cumplan.

Y es que si hay una práctica que el equipo ya está haciendo bien, los beneficios de esta práctica y cómo asegurarse de seguir realizándola o mejorarla se discute también.

Estas mejoras pueden haberse detectado e implementado antes, pero la Retro permire una oportunidad formal para enfocarse en la inspección y la adaptación.

## 2.8 El Sprint

En Scrum un proyecto se ejecuta en bloques temporales cortas y fijas ([iteraciones](https://proyectosagiles.org/desarrollo-iterativo-incremental) de un mes natural y hasta de dos semanas). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto que sea potencialmente entregable, de manera que cuando el [cliente (Product Owner)](https://proyectosagiles.org/cliente-product-owner) lo solicite sólo seanecesario un esfuerzo mínimo para que el producto este disponible para ser utilizado. Para ello, durante la iteración el [equipo](https://proyectosagiles.org/equipo-team) colabora estrechamente y se llevan a cabo las siguientes dinámicas:

* Cada día el [equipo](https://proyectosagiles.org/equipo-team) realiza una [reunión de sincronización](https://proyectosagiles.org/reunion-diaria-de-sincronizacion-scrum-daily-meeting), donde cada miembro inspecciona el trabajo de los otros para poder hacer las adaptaciones necesarias, comunica cuales son los impedimentos con que se encuentra, actualiza el estado de la [lista de tareas de la iteración (Sprint Backlog)](https://proyectosagiles.org/lista-tareas-iteracion-sprint-backlog) y los [gráficos de trabajo pendiente (Burndown charts)](https://proyectosagiles.org/graficos-trabajo-pendiente-burndown-charts).
* El [Facilitador (Scrum Master)](https://proyectosagiles.org/facilitador-scrum-master) se encarga de que el equipo pueda cumplir con su compromiso y de que no se merme su productividad.
  + Elimina los obstáculos que el equipo no puede resolver por sí mismo.
  + Protege al equipo de interrupciones externas que puedan afectar su compromiso o su productividad.

### 2.8.1 Recomendaciones

* Para poder completar el máximo de requisitos en la iteración, se debe minimizar el número de objetivos/requisitos en que el equipo trabaja simultáneamente(*WIP, Work In Progress*), completando primero los que den más valor al cliente. Esta forma de trabajar, que se ve facilitada por la propia estructura de la [lista de tareas de la iteración](https://proyectosagiles.org/lista-tareas-iteracion-sprint-backlog), permite tener más capacidad de reacción frente a cambios o situaciones inesperadas.

### 2.8.2 Restricciones

* No se puede cambiar los objetivos/requisitos de la iteración en curso.
  + En la [reunión de planificación de la iteración](https://proyectosagiles.org/planificacion-iteracion-sprint-planning) el equipo adquirió el compromiso de completar en la iteración unos requisitos concretos, basó su plan y organización en ellos. Cambiar los objetivos/requisitos de la iteración dificulta la concentración del equipo, baja su moral y su compromiso.
  + El hecho de no poder cambiar los objetivos/requisitos de la iteración una vez iniciada facilita que el cliente cumpla con su responsabilidad de conocer qué es lo más prioritario a desarrollar, antes de iniciar la iteración.
  + Notar que Scrum minimiza esta necesidad ya que, por un lado, los objetivos/requisitos que se están desarrollando eran los más prioritarios justo antes de iniciar la iteración y, por otro lado, las iteraciones en Scrum son suficientemente cortas (2 o 4 semanas) como para que la probabilidad de cambios una vez iniciada la iteración sea mínima.

### 2.8.3 Terminación anormal de la iteración

Sólo en situaciones muy excepcionales el cliente o el equipo pueden solicitar una terminación anormal de la iteración. Esto puede suceder si, por ejemplo, el contexto del proyecto ha cambiado enormemente y no es posible esperar al final de la iteración para aplicar cambios, o si el equipo encuentra que es imposible cumplir con el compromiso adquirido. En ese caso, se dará por finalizada la iteración y se dará inicio a otra mediante una [reunión de planificación de la iteración](https://proyectosagiles.org/planificacion-iteracion-sprint-planning).

## 2.9 Reunion de Planificacion de Sprint (Sprint Planning Meeting)

La planificación de las tareas a realizar en la [iteración](https://proyectosagiles.org/desarrollo-iterativo-incremental) se divide en dos partes:

* Primera parte de la reunión. Se realiza en un [timebox](https://proyectosagiles.org/timebox) de cómo máximo 4 horas:
  + El [cliente](https://proyectosagiles.org/cliente-product-owner) presenta al equipo la [lista de requisitos priorizada](https://proyectosagiles.org/lista-requisitos-priorizada-product-backlog) del producto o proyecto, pone nombre a la meta de la iteración (de manera que ayude a tomar decisiones durante su ejecución) y propone los requisitos más prioritarios a desarrollar en ella.
  + El [equipo](https://proyectosagiles.org/equipo-team) examina la lista, pregunta al cliente las dudas que le surgen, añade más [condiciones de satisfacción](https://proyectosagiles.org/introduccion-estimacion-planificacion-agil" \l "historias-usuario) y selecciona los objetivos/requisitos más prioritarios que se compromete a completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.
* Segunda parte de la reunión. Se realiza en un [timebox](https://proyectosagiles.org/timebox) de cómo máximo 4 horas\* . El equipo planifica la iteración, 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, es el responsable de organizar su trabajo y es quien mejor conoce cómo realizarlo.
  + Define las tareas necesarias para poder completar cada objetivo/requisito, creando la [lista de tareas de la iteración (Sprint Backlog)](https://proyectosagiles.org/lista-tareas-iteracion-sprint-backlog) basándose en la [definición de hecho](https://proyectosagiles.org/definicion-de-hecho-definition-of-done/).
  + Realiza una estimación conjunta del esfuerzo necesario para realizar cada tarea.
  + Cada miembro del equipo se autoasignaa las tareas que puede realizar.

Estos son tiempos máximos en el caso de iteraciones mensuales. En iteraciones de tamaño menor el tiempo es proporcionalmente inferior, y se puede ir reduciendo conforme el equipo va ganando experiencia en este tipo de reuniones, aunque también dependerá de la complejidad a desarrollar en la iteración.

### 2.9.1 Beneficios

* Productividad mediante comunicación y creación de sinergias:
  + Todos los miembros del equipo tienen una misma visión del objetivo y se ha utilizado los conocimientos y las experiencias de todos para elaborar la mejor solución entregable en el mínimo tiempo y con el mínimo esfuerzo, eliminando tareas innecesarias, detectando conflictos y dependencias entre tareas, etc.
* Potenciación del compromisodel equipo sobre el objetivo común de la iteración:
  + Es todo el equipo quien asume la responsabilidad de completar en la iteración los requisitos que selecciona. Facilita la ayuda de cualquier miembro si se detecta algún impedimento que bloquea el progreso de la iteración, especialmente si cuando se está llegando al final de la iteración es necesaria la participación de todos para poder completar los objetivos comprometidos.
  + Es cada una de las personas la que se responsabiliza de realizar sus tareas (a las que se autoasignó) en los tiempos que proporcionó. Si existe falta de compromiso con respecto al resto de miembros del equipo se hará muy evidente en las [reuniones diarias de sincronización del equipo (Scrum daily meeting)](https://proyectosagiles.org/reunion-diaria-de-sincronizacion-scrum-daily-meeting).
* Una estimación conjunta es más fiable, dado que tiene en cuenta los diferentes conocimientos, experiencia y habilidades de los integrantes del equipo.

## 2.10 Objetivo del Sprint (Sprint Goal)

El Sprint Goal es el objetivo más importante que tiene que cumplir el equipo Scrum durante el sprint. Debe ser alcanzable mediante la realización de un conjunto “coherente” de ítems del Product Backlog.

Este conjunto puede representar, bien una funcionalidad, bien cualquier otra cosa que sirva para que el equipo trabaje en una causa común, en lugar de en iniciativas separadas.

El Sprint Goal sirve de guía al [Development Team](https://www.paradigmadigital.com/techbiz/el-development-team-mas-que-un-equipo-de-desarrolladores/) para saber por qué están construyendo “este” Incremento. Si durante el sprint se descubre que el trabajo a realizar es diferente, el Development Team y el Product Owner pueden renegociar el alcance del Sprint Backlog durante el sprint.

Por otro lado, el Sprint Goal también sirve para que los stakeholders, el Product Owner y el Development Team estén alineados respecto a qué es lo más importante, lo más prioritario. Es decir, guía el trabajo del equipo y determina, en gran medida, las expectativas de los stakeholders.

Generalmente, los stakeholders y/o el sponsor estarán satisfechos si el equipo alcanza el Sprint Goal. No olvidemos que el Sprint Backlog es una predicción del trabajo que realizará el equipo, mientras que el Sprint Goal es un compromiso.

## 2.11 Scrum Diario (Daily Scrum)

El objetivo de esta reunión es facilitar la transferencia de información y la colaboración entre los miembros del [equipo](https://proyectosagiles.org/equipo-team) para aumentar su productividad, al poner de manifiesto puntos en que se pueden 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](https://proyectosagiles.org/desarrollo-iterativo-incremental), obstáculos que pueden impedir este objetivo) para al finalizar la reunión poder hacer las [adaptaciones](https://proyectosagiles.org/control-predictivo-control-empirico) necesarias que permitan cumplir con el compromiso conjunto que el equipo adquirió para la iteración (en la [reunión de planificación de la iteración](https://proyectosagiles.org/planificacion-iteracion-sprint-planning)).

Cada miembro del equipo debe responder las siguientes preguntas en un [timebox](https://proyectosagiles.org/timebox) de cómo máximo 15minutos:

* ¿Qué he hechodesde la última reunión de sincronización? ¿Pude hacer todo lo que tenía planeado? ¿Cuál fue el problema?
* ¿Qué voy a hacera partir de este momento?
* ¿Qué impedimentostengo o voy a tener para cumplir mis compromisos en esta iteración y en el proyecto?

Como apoyo a la reunión, el equipo cuenta con la [lista de tareas de la iteración](https://proyectosagiles.org/lista-tareas-iteracion-sprint-backlog), donde se actualiza el estado y el esfuerzo pendiente para cada tarea, asi como con el [gráfico de horas pendientes en la iteración](https://proyectosagiles.org/graficos-trabajo-pendiente-burndown-charts).

### 2.11.1 Beneficios

* Aumentar la productividad en el proyecto y potencia el compromiso de equipo, dado que cada miembro pone de manifiesto delante del resto:
  + Las tareas que pueden afectar a otros miembros del equipo, por que impactan en su trabajo o por que hay dependencias (especialmente si existe un retraso).
  + Los impedimentos con que se encuentra. La reunión de sincronización permite identificar más problemas a tiempo. El resto de miembros del equipo pueden ofrecer ayuda a otros en la realización de tareas o para resolver problemas que ya tuvieron anteriormente. El [Facilitador (Scrum Master)](https://proyectosagiles.org/facilitador-scrum-master) se encargará de solucionar los impedimentos que el equipo no puede solucionar por sí solo o que le quitan tiempo para cumplir con su compromiso fundamental de desarrollo de requisitos.
  + Las tareas no planeadas que está realizando que el equipo no conoce y puede que noestén alineadas con el compromiso del equipo, aunque él crea que lo que está haciendo es lo mejor que se puede hacer.
  + Cuales son sus necesidades. Cada miembro entiende las necesidades de los otros miembros del equipo respecto a su trabajo, de manera que pueden colaborar y adaptar sus trabajos para que den el máximo valor y no realizar tareas que no proporcionan ningún beneficio al resto del equipo.
  + Cual es su ritmo de trabajo. Se hace visible si de manera continua un miembro del equipo está realizando tareas por debajo del rendimiento esperado. Se evita que una persona señale con el dedo a otra, dado que la reunión de sincronización pone a todos los miembros del equipo en la misma situación de tener que explicar en qué tareas están trabajando.
  + Cuales son los criterios que está utilizando para realizar sus tareas, de manera que estén alineados con los objetivos comunes del equipo.
* Fomentar el aprendizaje de los miembros del equipo, ya que pueden ver cómo trabajan los otros según sus especialidades y experiencias.
* Conocer el estado de la iteración, ver si es posible completar los requisitos a que se comprometió el equipo, en vista de las desviaciones y de las tareas pendientes.

### 2.11.2 Restricciones

* La reunión diaria de estado y sincronización del equipo no es para resolver problemas, los problemas se resuelven después de la reunión.
  + No a todos los miembros del equipo les interesan todos los detalles de cada tema.
  + En la reunión los miembros del equipo programan reuniones entre ellos donde colaborar sincronizando tareas, ayudando a resolver problemas, etc.
  + No puede haber una persona explicando su estado mientras otras "se han apartado" de la reunión para comentar un tema particular. Si apareciese alguna conversación de interés común (que debe ser rápida), debe poder ser escuchada por todo el equipo sin distraer el principal objetivo de que todos conozcan en qué están trabajando los demás. Si la mini conversación no es del interés de todos, debe hacerse después de la reunión.
* El equipo debe contar con unos criterios consensuados sobre el proceso de ejecución de las de tareas
  + El proceso de ejecución de las tareas debe estar consensuado para evitar que cada reunión sea una exposición de discrepancias entre los miembros del equipo.

### 2.11.3 Recomendaciones

* Realizar la reunión diaria de sincronización de pie, para que los miembros del equipo no se relajen ni se extiendan en más detalles de los necesarios.
* Realizar las reuniones de colaboración entre miembros del equipo justo después de la de sincronización.

## 2.12 Revision de Sprint (Sprint Review)

Reunión informal donde el [equipo](https://proyectosagiles.org/equipo-team) presenta al [cliente](https://proyectosagiles.org/cliente-product-owner) los requisitos completados en la [iteración](https://proyectosagiles.org/desarrollo-iterativo-incremental), en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo, haciendo un recorrido por ellos lo más real y cercano posible al objetivo que se pretende cubrir.

En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el cliente realiza las [adaptaciones](https://proyectosagiles.org/control-predictivo-control-empirico) necesarias de manera objetiva, ya desde la primera iteración, [replanificando el proyecto](https://proyectosagiles.org/replanificacion-proyecto).

Se realiza en un [timebox](https://proyectosagiles.org/timebox) de cómo máximo 4 horas.

### 2.12.1 Beneficios

* El cliente puede ver de manera objetiva cómo han sido desarrollados los requisitos que proporcionó, ver si se cumplen sus expectativas, entender más qué es lo que necesita y tomar mejores decisiones respecto al proyecto.
* El equipo puede ver si realmente entendió cuáles eran los requisitos que solicitó el cliente y ver en qué puntos hay que mejorar la comunicación entre ambos.
* El equipo se siente más satisfecho cuando puede ir mostrando los resultados que va obteniendo. No está meses trabajando sin poder exhibir su obra.

### 2.12.2 Restricciones

* Sólo se pueden mostrar los requisitos completados, para que el cliente no se haga falsas expectativas y pueda tomar decisiones correctas y objetivas en función de la velocidad de desarrollo y el resultado realmente completado. Un requisito no completado quedará como un requisito más a replanificar.

## 2.13 Retrospectiva de Sprint (Sprint Restrospective)

Con el objetivo de [mejorar de manera continua](https://proyectosagiles.org/control-predictivo-control-empirico) su productividad y la calidad del producto que está desarrollando, el [equipo](https://proyectosagiles.org/equipo-team) analiza cómo ha sido su manera de trabajar durante la [iteración](https://proyectosagiles.org/desarrollo-iterativo-incremental), por qué está consiguiendo o no los objetivos a que se compremetió al [inicio de la iteración](https://proyectosagiles.org/planificacion-iteracion-sprint-planning) y por qué el incremento de producto que acaba de [demostrar](https://proyectosagiles.org/demostracion-requisitos-sprint-review) al cliente era lo que él esperaba o no:

* Qué cosas han funcionado bien.
* Cuales hay que mejorar.
* Qué cosas quiere probar hacer en la siguiente iteración.
* Qué ha aprendido.
* Cuales son los problemas que podrían impedirle progresar adecuadamente.

El [Facilitador](https://proyectosagiles.org/facilitador-scrum-master) se encargará de ir eliminando los obstáculos identificados que el propio equipo no pueda resolver por sí mismo.

Notar que esta reunión se realiza después de la [reunión de demostración al cliente de los objetivos conseguidos en la iteración](https://proyectosagiles.org/demostracion-requisitos-sprint-review), para poder incorporar su feedback y cumplimiento de expectativas como parte de los temas a tratar en la reunión de retrospectiva

Se realiza en un [timebox](https://proyectosagiles.org/timebox) de cómo máximo 3 horas (si la iteración es mensual).

### 2.13.1 Beneficios

* Incrementa la productividad en el proyecto, la calidad del producto (cosa que permite hacerlo crecer de manera sostenida) y potencia el aprendizaje del equipo de manera sistemática, iteración a iteración, con resultados a corto plazo.
* Aumenta la motivación del equipo dado que participa en la mejora de proceso, se siente escuchado, toma decisiones consensuadas (y más sostenibles) para ir eliminando lo que molesta e impide que sea más productivo.

### 2.13.2 Restricciones

* En necesario que el Equipo y el Facilitador dispongan de autoridad, mecanismos y recursos para ir mejorando su forma de trabajar y el contexto del proyecto. Es frustrante encontrar sistemáticamente los mismos obstáculos y no poder solucionarlos.

## 2.14 Artefactos de Scrum

En el marco de trabajo Scrum, denominamos Artefacto a aquellos elementos físicos que se producen como resultado de la aplicación de Scrum. Los tres principales artefactos o herramientas Scrum son: el Product Backlog, Sprint Backlog y el Incremento.

### 2.14.1 Product Backlog

El Product Backlog es un inventario que contiene cualquier tipo de trabajo que haya que hacer en el producto: requerimientos, casos de uso, tareas y dependencias. Es la principal fuente de información sobre el producto en Scrum, una lista, en cualquier formato, que contiene todos los requerimientos que necesitamos implementar en el producto. Esta lista es el resultado del trabajo del Product Owner con el cliente, los distintos stakeholders, sponsors, comités, etc, y refleja el estado real del trabajo pendiente de implementar en el producto, así como el ya realizado.

El Product Backlog debe ser gestionado en exclusiva por el Product Owner, siendo su principal función la de priorizar aquellos elementos que tienen más valor en cada etapa y detallarlos para que el equipo de desarrollo sea capaz de valorarlos y ejecutarlos.

Al comenzar a utilizar Scrum, no es necesario una lista completa y exhaustiva de todos los requerimientos. Es recomendable empezar con los dos o tres requerimientos más urgentes arriba e ir añadiendo elementos conforme vamos descubriendo más necesidades de nuestro producto.

Un Product Backlog contiene distintos elementos:

* Funcionalidades
* Bugs
* Historias de usuario: una forma de expresar elementos de un Product Backlog. Para obtener el máximo valor de una historia de usuarios es necesario expresarlas desde el punto de vista del usuario.
* Tareas técnicas
* Trabajo de investigación

### 2.14.2 Sprint Backlog

¿A qué denominamos Sprint Backlog? Se trata de una lista de elementos en los que trabajar durante la etapa de Sprint. Estos elementos normalmente se componen de tareas técnicas más pequeñas que permiten conseguir un incremento de software terminado.

Todo el trabajo que el Development Team haya seleccionado para hacer durante el siguiente Sprint pasa al Sprint Backlog. Este artefacto es un elemento para visualizar el trabajo a realizar durante cada Sprint y está gestionado por el equipo de desarrollo. Su propósito es mantener la transparencia dentro del desarrollo, actualizándolo durante toda la iteración especialmente a través de los daily Scrums.

El Sprint Backlog permite visualizar, durante cada Sprint, aquellos elementos que aún no han empezado a desarrollarse, aquellos que sí y quiénes están trabajando en los mismos, así como aquellos que están esperando a desplegarse o están completamente terminados.

Este artefacto permite entender cuál es la evolución del trabajo durante el Sprint, así como hacer un análisis de riesgos. Dado que cada Sprint tiene una meta específica (p.e. *permitir que los usuarios se registren en la app móvil*) y hay elementos seleccionados del Product Backlog que tienen más o menos valor, el Sprint Backlog permite analizar hasta donde se ha cumplido el objetivo y que se podría eliminar. De esta forma, maximizamos el retorno de la inversión en desarrollo.

## 2.15 Lista de Producto (Product Backlog)

La lista de objetivos/requisitos priorizada representa la visión y expectativas del [cliente](https://proyectosagiles.org/cliente-product-owner) respecto a los objetivos y entregas del producto o proyecto. El cliente es el responsable de crear y gestionar la lista (con la ayuda del [Facilitador](https://proyectosagiles.org/facilitador-scrum-master) y del [equipo](https://proyectosagiles.org/equipo-team), quien proporciona el coste estimado de completar cada requisito). Dado que reflejar las expectativas del cliente, esta lista permite involucrarle en la dirección de los resultados del producto o proyecto.

* Contiene los objetivos/requisitos de alto nivel del producto o proyecto, que se suelen expresar en forma de[historias de usuario](https://proyectosagiles.org/introduccion-estimacion-planificacion-agil" \l "historias-usuario). Para cada objetivo/requisito se indica el valor que aporta al cliente y el costeestimadode completarlo. La lista está priorizada balanceando el [valor que cada requisito aporta al negocio frente al coste estimado que tiene su desarrollo](https://proyectosagiles.org/priorizacion-requisitos-valor-coste), es decir, basándose en el Retorno de la Inversión (ROI).
* En la lista se indican las posibles iteraciones y las entregas (*releases*) esperadas por el cliente (los puntos en los cuales desea que se le entreguen los objetivos/requisitos completados hasta ese momento), en función de la velocidad de desarrollo del (los) equipo(s) que trabajará(n) en el proyecto. Es conveniente que el contenido de cada iteración tenga una coherencia, de manera que se reduzca el esfuerzo de completar todos sus objetivos.
* La lista también tiene que considerar los riesgos del proyecto e incluir los requisitos o tareas necesarios para mitigarlos.

Antes de iniciar la primera iteración, el cliente debe tener definida la meta del producto o proyecto y la lista de requisitos creada. No es necesario que la lista sea completa ni que todos los requisitos estén detallados al mismo nivel. Basta con que estén identificados y con suficiente detalle los requisitos más prioritarios con los que el equipo empezará a trabajar. Los requisitos de iteraciones futuras pueden ser mucho más amplios y generales. La incertidumbre y complejidad propia de un proyecto hacen conveniente no detallar todos los requisitos hasta que su desarrollo esté próximo. De esta manera, [el esfuerzo de recoger, detallar y desarrollar el resto de requisitos (menos prioritarios) está repartido en el período de ejecución del proyecto](https://proyectosagiles.org/replanificacion-proyecto). En el caso del desarrollo de un producto, la lista va evolucionando durante toda la vida del producto (los requisitos “emergen”). En el caso de un proyecto, conforme éste avance irán apareciendo los requisitos menos prioritarios que falten. Esto produce varias ventajas:

* Se evita caer en parálisis de análisis al inicio del proyecto, de manera que se puede iniciar antes el desarrollo y el cliente puede empezar a obtener resultados útiles.
* Se evita analizar en detalle requisitos no prioritarios que podrían cambiar durante el transcurso del proyecto, dado que el cliente conocerá mejor cuál ha de ser el resultado a conseguir, o bien por que podrían ser reemplazados por otros.
* Puede llegar a un punto del proyecto en que no valga la pena analizar ni desarrollar los requisitos restantes, dado el poco [retorno de inversión (ROI)](https://proyectosagiles.org/priorizacion-requisitos-valor-coste) que tienen.

### 2.15.1 Iteración de entrega (release sprint)

Cuando el cliente solicita una entrega de los objetivos/requisitos completados hasta ese momento, el equipo puede necesitar añadir una iteración de entrega, más corta que las iteraciones habituales, donde realizar alguna tarea que no ha sido necesaria o posible hasta el momento de la entrega final y acabar de corregir defectos detectados en la última demostración.

Idealmente esta iteración de entrega no debería existir (o reducirse a tareas mínimas dentro de una iteración) si:

* Se ha trabajado con una buena [Definición de Hecho](https://proyectosagiles.org/definicion-de-hecho-definition-of-done/) durante cada iteración del proyecto.
* Se hacen entregas muy cortas cada poco tiempo, con lo que la cantidad de cosas a entregar, integrar y probar es pequeña.
* Se está realizando un esfuerzo de automatización de estas tareas de pruebas, iteración y entrega durante todo el proyecto (con lo cual se gana en eficiencia y seguridad).

### 2.15.2 Uso de la lista de objetivos priorizada

* Para medir la velocidad de desarrollo del equipo, ver una progresión constante y extrapolar la fecha de las entregas, es conveniente seguir las siguientes recomendaciones:
  + Los requisitos deben tener un esfuerzo semejante para ser completados.
  + La estimación de un requisito no debe ser superior a 10 días (si las iteraciones son de 20 días laborables).
* Cada requisito tiene asociado un factor de complejidad, que permite ajustar su coste estimado en función de la incertidumbre de la complejidad de su desarrollo en el momento de introducirlo en la lista. Este factor de coste se irá ajustando conforme las iteraciones avancen y el equipo conozca mejor el producto o proyecto, su contexto y su velocidad de desarrollo con las herramientas y tecnologías que utiliza.
* Si un requisito depende de otro, se coloca en algún punto por debajo del que depende.
* Si un requisito no se finaliza en una iteración, se le vuelve a poner en alguna de las siguientes iteraciones, indicando el coste pendiente de desarrollo.
* El “origen” permite saber quién podría participar en la definición de un objetivo/requisito y sería conveniente que estuviese presente en el momento de su demostración.

## 2.16 Lista de Pendientes del Sprint (Sprint Backlog)

Lista de tareas que el [equipo](https://proyectosagiles.org/equipo-team) elabora en la [reunión de planificación de la iteración (Sprint planning)](https://proyectosagiles.org/planificacion-iteracion-sprint-planning) como plan para completar los objetivos/requisitos seleccionados para la [iteración](https://proyectosagiles.org/desarrollo-iterativo-incremental) y que se compromete a [demostrar](https://proyectosagiles.org/demostracion-requisitos-sprint-review) al[cliente](https://proyectosagiles.org/cliente-product-owner) al finalizar la iteración, en forma de incremento de producto preparado para ser entregado.

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

Para cada uno de los objetivos/requisitos se muestran sus tareas, el esfuerzo pendiente para finalizarlas y la autoasignación que han hecho los miembros del equipo.

El progreso de la iteración y su velocidad con respecto a tareas u horas pendientes se muestra mediante un [gráfico de trabajo pendiente (Burndown chart)](https://proyectosagiles.org/graficos-trabajo-pendiente-burndown-charts).

### 2.16.1 Uso de la lista

* Los objetivos/requisitos están ordenados por orden de prioridad para el cliente.
  + Por ello, signos de falta de foco, problemas o impedimentos serían que se estén completando objetivos que no son los primeros de la lista, así como tener [demasiados objetivos/requisitos en progreso simultáneamente](https://proyectosagiles.org/ejecucion-iteracion-sprint).
* Si una tarea depende de otra, se coloca en algún punto por debajo de la que depende.
* Las tareas deben estar identificadas de manera que tengan un coste semejante para ser completadas, entre 4 y 16 horas. Este tamaño permitirá:
  + Que las tareas sean suficientemente pequeñas como para poder detectar progreso o estancamiento de manera diaria.
  + Que las tareas no sean muy pequeñas, que sean suficientemente relevantes, no generen ruido ni microgestión.
  + Medir la velocidad de desarrollo del equipo y extrapolar si es posible finalizarlas dentro de la iteración.

### 2.16.2 El tablero de tareas (Scrum Taskboard)

La lista de objetivos a completar en la iteración ([Product Backlog](https://proyectosagiles.org/lista-requisitos-priorizada-product-backlog) Items) se puede gestionar mediante un tablón de tareas (Scrum Taskboard). Al lado de cada objetivo se ponen las tareas necesarias para completarlo, en forma de post-its, y se van moviendo hacia la derecha para cambiarlas de estado (pendientes de iniciar, en progreso, hechas). Para cada miembro del equipo se puede utilizar adhesivos de colores más pequeños sobre cada tarea, de manera que se pueda ver en qué tareas está trabajando cada cual.

El equipo elabora esta lista de tareas en la segunda parte de la [reunión de planificación de la iteración](https://proyectosagiles.org/planificacion-iteracion-sprint-planning). La lista va evolucionando (nuevas tareas, cambios, estado, esfuerzo pendiente, …) a medida que la iteración avanza, especialmente durante la [reunión diaria de sincronización](https://proyectosagiles.org/reunion-diaria-de-sincronizacion-scrum-daily-meeting).

Este tablón o cuadro de mandos actúa como radiador de información tanto para el equipo como para cualquier otra persona relacionada con el proyecto.

## 2.17 Incremento

Si Scrum tuviera que ser reducido a una sola cosa, sería a entregar una pieza de software terminado en cada Sprint. Un Incremento es el resultado del Sprint, es la suma de todas las tareas, casos de uso, historias de usuario y cualquier elemento que se haya desarrollado durante el Sprint y que será puesto a disposición del usuario final en forma de software, aportando un valor de negocio al producto que se está desarrollando.

Construir software de manera ágil se basa en hacerlo de manera iterativa e incremental. Mediante las iteraciones, nos aseguramos que todo el ciclo de vida del software (planificación, diseño, desarrollo, testeo y entrega) ocurre en 4 semanas o menos. Por supuesto, no podemos construir toda la funcionalidad que queremos en solo cuatro semanas y tenemos que buscar la manera de ir entregando los componentes necesarios justo a tiempo.

## 2.18 Transparencia de los Artefactos

La transparencia en Scrum es una cualidad fundamental para que el funcionamiento sea el correcto. Es clave que todos los miembros del equipo Scrum entiendan y conozcan el rumbo del proyecto. Por consiguiente, es una cualidad imprescindible que **los artefactos de Scrum estén definidos de manera absolutamente clara**.

En la medida en que los artefactos no son completamente transparentes, estas decisiones pueden ser erróneas, el valor puede disminuir y el riesgo puede aumentar. El Scrum Master debe trabajar con el Dueño de Producto, el **Equipo de Desarrollo**y otras partes involucradas para entender si los artefactos son completamente transparentes. Hay prácticas para hacer frente a la falta de transparencia; el Scrum Master debe ayudar a todos a aplicar las prácticas más apropiadas si no hay una transparencia completa.

Un **Scrum Master** puede detectar la falta de transparencia inspeccionando artefactos, reconociendo patrones, escuchando atentamente lo que se dice y detectando diferencias entre los resultados esperados y los reales. La labor del Scrum Master es trabajar con el Equipo Scrum y la organización para mejorar la transparencia de los artefactos.

Tanto tú, como el equipo de desarrollo, podréis tomar decisiones basadas en criterios sólidos si los artefactos son transparentes. En consecuencia, y esa debe ser una labor del Scrum Master, debes asegurarte de que todos los miembros del equipo colaboren para que haya un entendimiento completo.

Éste no suele ser un proceso inmediato. Se tarda un cierto tiempo en que todos los miembros del equipo “hablen el mismo idioma”. Sin embargo, desde el primer momento debes hacer lo posible para aclarar todas las dudas y concretar todos los conceptos ambiguos.

## 2.19 Definicion de “Terminado” (Definition of “Done”)

La Definición de Hecho (Definition of Done) permite:

* Establecer un criterio de calidad, define qué entregables y mínimos de calidad tienen se tienen que cumplir en TODOS los objetivos / requisitos que se van a ir aceptando durante cada iteración del proyecto.
* Tener siempre un producto “potencialmente entregable” al [Product Owner (cliente)](https://proyectosagiles.org/cliente-product-owner/) al finalizar cada iteración, no dejar trabajo pendiente para el final, escondido “debajo de la alfombra”, que pueda impedir utilizar los resultados del proyecto lo antes posible.
  + Esto permite saber claramente en qué punto real se está del proyecto y tomar buenas decisiones al respecto sobre lo que se ha conseguido hasta ese momento y lo que todavía no se sabe con certeza cuándo estará acabado. Con una buena Definición de Hecho, el cliente podrá tomar decisiones correctas cuando al final de cada iteración el equipo le haga una [demostración de los requisitos completados](https://proyectosagiles.org/demostracion-requisitos-sprint-review/): cambiar las prioridades en función de la velocidad de desarrollo, solicitar una entrega del producto desarrollado hasta ese momento, etc.
  + Por otro lado, lo peor que podría suceder es que, aunque el equipo haya ido presentando todos los objetivos / requisitos completados durante el proyecto (en cada iteración), se podría dar el caso de que los días previos a una entrega de repente se den cuenta de que hay que hacer mucho trabajo de integración y finalización (que podría haberse ido haciendo antes, durante el proyecto). Esto también les dificultaría estimar cuándo tiempo necesitan para acabar de terminar el producto (lo cual pondría en peligro la fecha de entrega).

La Definición de Hecho se acuerda entre el [Product Owner (cliente)](https://proyectosagiles.org/cliente-product-owner/) y el [Equipo de desarrollo](https://proyectosagiles.org/equipo-team/) al principio del proyecto y se puede ir mejorando durante su transcurso (si es necesario precisar más las expectativas en cuanto a calidad global o del proceso de trabajo o, simplemente, tras una [Retrospectiva](https://proyectosagiles.org/retrospectiva-sprint-retrospective/), para ir consiguiendo una mejor la calidad del producto final).

## 2.20 Ventajas y Desventajas

**Ventajas de Scrum:**

* El cliente puede comenzar a utilizar el producto rápidamente.
* El cliente puede decidir los nuevos objetivos a realizar.
* Se agila el proceso, porque se divide el problema en pequeñas tareas.
* Menos probabilidad de que se den sorpresas o desarrollos inesperados porque el cliente va viendo poco a poco lo que se está desarrollando.

**Desventajas de Scrum:**

* Existe la tendencia que si se deja una tarea sin terminar y que por las exigencias del Dueño del Producto se deban realizar otras nuevas. Estas tareas no terminadas puedan obstaculizar la planeación de nuevas sprints y se deba volver al problema original.
* Alto nivel de stress de los miembros del equipo, el desgaste puede ser excesivo y estresante lo que puede disminuir el rendimiento.
* La necesidad de contar con equipos multidisciplinarios puede ser un problema, porque cada integrante del equipo debe estar en capacidad de resolver cualquier tarea y no siempre se cuenta con este perfil en la empresa.
* El equipo puede estar tentado de tomar el camino más corto para cumplir con un sprint, que no necesariamente puede ser el de mejor calidad en el desarrollo del producto.

## 2.21 Valores del Trabajo

### 2.21.1 Compromiso

**Este valor ha sido malinterpretado durante mucho tiempo**. Muchos lo entendieron como un contrato estricto no escrito por parte del equipo por el que debían completar, sin excepción ni excusa alguna, el Sprint Goal y el alcance acordado.

O peor, el Product Owner tomaba por compromiso la exigencia al Dev Team de hacer todo lo que fuera necesario para completar todo el trabajo planificado, aun a sabiendas del no cumplimiento del DoD ([Definition Of Done](https://www.scrumalliance.org/community/articles/2008/september/what-is-definition-of-done-(dod))) y de la consiguiente generación de deuda técnica.

Sin embargo, compromiso significa que **cada miembro del equipo Scrum** (es decir, todos, incluido Scrum Master y Product Owner) **hará el máximo esfuerzo posible y será completamente transparente sobre el progreso del Sprint**.

En el mundo del desarrollo software, cualquiera que tenga una mínima experiencia y se haya enfrentado a un proyecto real del mundo real, sabrá que un compromiso cerrado e inviolable en alcance sencillamente es imposible y solo conduce a la frustración.

El matiz es muy importante: en el Sprint Planning el equipo hace una **predicción** del trabajo que podrá llevar a cabo, no una firma de un contrato con sello lacrado. **Compromiso significa dedicación y se refiere a las acciones y el esfuerzo, no al resultado final.**

Por ejemplo, podemos jugar a relacionar este valor con uno de los principios universales, **la Justicia**. Si el Product Owner o un miembro del Dev Team no dedica el suficiente tiempo, energía y voluntad a contribuir a la consecución de los objetivos del equipo, como sí hacen el resto de miembros, ¿sería esta circunstancia justa respecto del resto de integrantes del equipo que sí están comprometidos?

**Los miembros de un equipo Scrum se comprometen con los objetivos del equipo**y, como dice [Gunther Verheyen](https://guntherverheyen.com/2013/05/03/theres-value-in-the-scrum-values/), con la calidad, con aprender, con ser mejores profesionales, con la transparencia, con auto-organizarse cada vez mejor, con ser proactivos, con la entrega de incrementos, con la inspección y adaptación, con la mejora continua del DoD, con aportar el máximo valor posible al producto que desarrollan, con el propio Scrum como framework, e incluso con [el manifiesto Agile](https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/) y los principios que de él emanan.

### 2.21.2 Foco

**“Begin with the end in mind”,** proclama Stephen R. Covey como el segundo hábito más importante en su libro “The 7 Habits of Highly Effective People”.

Todos sabemos que el enemigo número uno de la productividad es la multitarea, la dispersión y la falta de concreción en los objetivos que se pretenden alcanzar. **Scrum proclama que todos los miembros del equipo deben enfocarse en el trabajo planificado en cada Sprint que, en última instancia, permite cumplir los Sprint Goals**.

El equipo debe enfocarse en lo que es más importante ahora, sin preocuparse en exceso por el futuro, que puede ser muy incierto y cambiante. El equipo no debe dedicar tiempo a tareas que puede que no sean necesarias en el futuro, eso sería tirar tiempo y dinero a la basura. [YAGNI](https://en.wikipedia.org/wiki/You_aren't_gonna_need_it) (You Ain’t Gonna Need It) o [KISS](https://es.wikipedia.org/wiki/Principio_KISS) (Keep It Simple Stupid) pueden ayudarnos a recordar que debemos mantener el foco.

He hablado con algunos Product Owners preocupados con la dedicación de ciertos miembros de equipos a determinadas tareas que no estaban incluidas en el Sprint (ni habían surgido en las replanificaciones diarias), simplemente porque habían decidido que algo podría ser buena idea o podría necesitarse en el futuro.

Aunque es difícil trazar una línea entre, por ejemplo, lo que evita deuda técnica o aumenta la calidad, y lo que es una pérdida de foco, **casi siempre hay que decir tres veces NO a un trabajo o tarea que no está directamente relacionada con alcanzar el Sprint Goal**.

Como regla general, lo mejor es siempre hacer el mínimo trabajo posible que me permite cumplir el objetivo.

**Foco es centrarse en cumplir el Sprint Goal y, por extensión, en los ítems del Product Backlog que forman parte del Sprint**. Si el Sprint Goal está bien definido y los ítems del Product Backlog fueron bien priorizados, entonces el equipo estará trabajando en entregar el máximo valor posible en cada momento. Eso es foco.

### 2.21.3 Franqueza

En realidad, la palabra inglesa “openness” puede traducirse como franqueza, sinceridad o actitud abierta y receptiva.

Scrum defiende la transparencia como un pilar básico del empirismo sobre el que se sustenta. **Sin transparencia es imposible llevar a cabo la Inspección y la Adaptación**. Por tanto, el Dev Team debe ser transparente con el trabajo que realiza, con el progreso del mismo, y con el conocimiento que adquiere (documentación).

El Product Owner debe facilitar el acceso a la información relevante a los stakeholders (Product Backlog, Total Cost of Ownership, etc.). Todos los miembros del equipo deben facilitar la transparencia en las comunicaciones y la compartición de información que facilite la colaboración dentro y fuera del equipo.

No solo lo anterior**, el equipo debe estar accesible y disponible para interactuar con los stakeholders** (especialmente en este caso, el Product Owner, por razones obvias) **o con otros miembros de la comunidad** (equipos de arquitectura, de seguridad, etc).

**Incluso los miembros del equipo deben estar abiertos a aprender nuevas habilidades o adquirir nuevos conocimientos que les conviertan en multi-funcionales** (cross-functional teams). Los miembros del equipo deben tener una actitud abierta y proactiva para mejorar sus capacidades y competencias profesionales, lo que además de contribuir al beneficio del equipo, se traduce en crecimiento personal.

### 2.21.4 Respeto

He de reconocer que la primera vez que leí “respeto” en la Guía Scrum pensé que consistía en ser educado con los miembros del equipo (¡cómo no!). Pero más adelante descubrí que el respeto, tal y como lo entiende Scrum y el mundo Agile, es un concepto fascinante.

**Los miembros de un equipo Scrum respetan el conocimiento, las habilidades y la experiencia profesional no solo del resto de miembros del equipo, sino también de aquellas personas con las que se relacionan**, sean de su propia organización o de otra.

Los miembros de un equipo Scrum se respetan entre ellos compartiendo información, fomentando un espíritu colaborador, aprendiendo y tomando decisiones juntos. Los miembros de un equipo Scrum se respetan entre ellos comprendiendo sus roles Scrum y la responsabilidad de cada uno dentro del equipo.

**Los miembros de un equipo Scrum respetan a los sponsors no desarrollando “features” que nadie va a usar después**.  Los miembros de un equipo Scrum respetan a los sponsors no gastando dinero en cosas que no aportan valor alguno al producto. Los miembros de un equipo Scrum respetan a los sponsors no dedicando el esfuerzo a tareas que no aportan ningún valor al producto.

**Los miembros de un equipo Scrum respetan a los usuarios**escuchándolos, mostrando interés por sus problemas y **dándoles una solución**.

**Los miembros de un equipo Scrum respetan a la comunidad** no permaneciendo como una isla o una unidad aislada dentro de la organización, sino **participando activamente para transferir conocimiento y experiencia**.

### 2.21.5 Coraje

**El equipo debe tener coraje para hacer lo correcto**. Por ejemplo, considerar el cambio como algo necesario para adaptarse a un mundo cambiante, a pesar de que a veces nos obligue a deshacer caminos andados.

Hay que tener coraje para ser capaz de desarrollar el producto sin mirar al futuro más de lo necesario, centrándote en lo que sabemos que es importante ahora, en lugar de en lo que podría (o no) ser importante en el futuro.

**El equipo debe tener coraje para resolver los impedimentos** que puedan surgir en el camino, anticipando los riesgos, sacando a la luz los problemas y pensando en una solución. Hay que tener coraje para reconocer los errores cometidos, ser transparentes y aprender de los mismos.

## 2.22 Herramientas de Trabajo

**TAIGA**



Taiga es una herramienta de software libre y código abierto, creada para gestionar y colaborar en proyectos ágiles, principalmente aquellos que utilizan metodología Scrum y kanban, además permite gestionar issues.

De igual manera Taiga posee otros módulos como wiki, videoconferencia (gracias a una solución de terceros), locaina, actualización de equipo y como si fuera poco gracias a su potente API permite la integración con servicios de terceros como Slack, GitHub, GitLab, Bitbucket, HipChat, Gogs, Hall entre otros.

Taiga es distribuida bajo la licencia de código abierto Affero GPL, está escrita en Django (backend) + AngularJS (frontend) y su código fuente está alojado en GitHub para que sea utilizado por la comunidad.

Para comenzar a disfrutar de Taiga debes registrarte de manera gratuita, verificar tu cuenta mediante el correo electrónico que te envían e iniciar sesión con los datos que indicaste anteriormente.

La combinación del marco de trabajo SCRUM con la herramienta de gestión de proyectos Taiga, puede ser aplicado para cualquier proyecto que desees realizar, ya sea a nivel de desarrollo de programas o en la elaboración de un artículo en tu blog como mostraremos en el siguiente caso práctico.

El primer paso es crear proyectos (puede ser un proyecto kanban o un proyecto Scrum), ambos son plantillas para comenzar un proyecto, pero que puedes ir adaptando según tus necesidades y gustos. En este caso elegiremos un proyecto Scrum y le daremos siguiente.

Seguidamente Vamos a dar un Nombre a nuestro proyecto y escribimos una descripción para el mismo

Una vez creado nuestro proyecto en Taiga lo primero que observamos es el BackLog, donde podemos añadir las historias de usuario de nuestro proyecto, cada historia de usuario se estima por lo general en puntos y debemos tener claro que no debería indicar el tiempo de la tarea, es importante destacar que la estimación en taiga se puede hacer por roles.

Puedes añadir tantas historias de usuario como necesite tu proyecto, al crear una nueva historia debes colocar un título, la estimación, el estado, etiquetas y la descripción de la tarea. Además, puedes segmentar la tarea si es requerida por el equipo o por el cliente.

Una vez creada todas las tareas necesarias que necesita nuestro proyecto, debemos crear lo que en Scrum se llama Sprint, que es la agrupación de un conjunto de tareas que puede representar un producto funcional y que está planificado se realice en un período de tiempo determinado.

Un proyecto puede tener tantos Sprint como sean necesarios y cada Sprint debe tener como resultado un prototipo.

En nuestro caso hemos creado un sólo sprint que tiene un día de duración, pero normalmente los sprint deben durar de 3 a 4 semanas en el caso de desarrollo de software y se debe tener un día de descanso entre sprint.

Al sprint hemos añadido todas las tareas antes creadas, Taiga permite hacer esto de una manera fácil arrastrando y soltando cada tarea en el sprint que deseas. También hemos priorizado las tareas con lo que determinamos cuál se debe hacer primero.

Taiga nos permite añadir miembros para que colaboren en las tareas, por ejemplo, un diseñador, para invitar a alguien a colaborar en tu proyecto debes ir al menú de admin y enviar una invitación a su correo electrónico.

Una vez tengamos nuestro sprint ya planificado y con los miembros ya listos para comenzar, nos dirijimos a nuestro panel de tareas del sprint que es un Kanban con varias columnas, las cuales representan cada una lo siguiente:

* **Historia de Usuario:** Todas las historias de usuario que conforman el sprint.
* **Nueva:** Cada historia de usuario se puede dividir en tareas.
* **En Curso:** Son aquellas tareas que se están realizando en este momento.
* **Lista para Testear:** Son aquellas tareas que están terminadas pero que no se han probado.
* **Cerrada:** Son aquellas tareas que han sido terminadas
* **Necesita Información:** Son aquellas tareas que se necesita una información extra para poder ser culminada.

Las tareas son tomadas por el colaborador que la va a realizar, el cuál se encargará de cambiarla por el estado que le corresponda. La idea es que todo el equipo conozca el estado del proyecto y trabajen más sincronizados.

El objetivo del sprint es que todas las tareas sean concretadas, taiga nos ofrece un gráfico que nos permite ver en todo momento como vamos.

# 3 PERSONAS Y ROLES DEL PROYECTO

|  |  |
| --- | --- |
| **Rol** | **Integrante** |
| **Product Owner** | David García Romero |
| **Scrum Master** | Pablo Ramiro Bustos Belzu |
| **Development Team** | Ariel Roger Censo Leaños |
| Henry Inocente Huanca |
| Carlos Eduardo Lopez Muñoz |
| Pierluigi Vaca García |

# 4 MODELOS USADOS PARA EL DESARROLLO DE SCRUM

## 4.1 Sprint Planning Meeting (Reunión de Planeamiento del Sprint) Cronograma de Trabajo

### Pila del Producto

|  |  |  |
| --- | --- | --- |
| **ID** | **Tarea** | **Descripción** |
| 1 | Creación de la Base de Datos | Implementación, Población de la base de datos. |
| 2 | Vista principal de la pag. Web | Diseño de la interfaz principal |
| 3 | Primera Vista del sistema(Login) | Verificación de ingreso de usuario |
| 4 | Gestionar Cliente | Es el manejo de los datos personales del usuario usados para contactarlo al momento de hacer la transacción o trueque. |
| 5 | Gestionar Detalle del Anuncio | Se registrará y se visualizará el detalle del anuncio ofrecido por el cliente en el sitio web. |
| 6 | Gestionar Cuenta Bancaria | Se almacenará las cuentas bancarias que el cliente puede llegar a tener, ya que se permitirá al sistema cobrarse por publicidad vip del producto, o ser intermediario de pago entre el vendedor y el comprador. |
| 7 | Gestionar Usuarios | Se gestionará la cuenta de acceso que tiene cada cliente, ya que si no está registrado no podrá disponer de los servicios que ofrece este sistema de información. |
| 8 | Administrar Soporte | Se dará soporte al cliente por algún reclamo o disconformidad o con algún cambio referente a su cuenta de usuario o anuncio. |
| 9 | Gestionar Inmueble | Se gestionará el registro de datos específicos del inmueble con el nombre adecuado e imagen seleccionados para su publicación. |
| 10 | Gestionar Galería | Se registrará las imágenes de los productos o inmuebles, el cual quiere vender el usuario. |
| 11 | Gestionar Ubicación | Se registrará la ubicación en la que se encuentra el producto o inmueble, el cual quiere vender el usuario. |
| 12 | Gestionar Trueque | Se podrá realizar algún tipo de intercambio entre el cliente y el vendedor. |
| 13 | Gestionar Anuncio | Aquí se llevará a cabo la publicación de un producto o bien inmueble que el cliente disponga a vender. |
| 14 | Gestionar Transacción | Se realizará transacciones para el pago de algún producto o inmueble que el cliente desee. |
| 15 | Administrar Bitácora | Registro de acciones por parte del usuario |
| 16 | Administrar BackUp | Ninguno |
| 17 | Gestionar Cliente (Android) | Es el manejo de los datos personales del usuario usados mediante la plataforma de Android. |
| 18 | Primera Vista Login (Android) | Implementación de la primera vista (Login) en Android. |
| 19 | Administrar Soporte (Android) | Soporte al cliente en la plataforma Android. |

## 4.2 Pila del Sprint (Sprint Backlog) Sprint 0

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Tarea** | **Descripción** | **Estado** |
| 1 | Creación de la Base de Datos | Implementación, Población de la base de datos. | Propuesto |
| 2 | Vista principal de la pag. Web | Diseño de la interfaz principal | Propuesto |
| 3 | Primera Vista del sistema(Login) | Verificación de ingreso de usuario | Propuesto |
| 4 | Gestionar Cliente | Es el manejo de los datos personales del usuario usados para contactarlo al momento de hacer la transacción o trueque. | Propuesto |
| 5 | Gestionar Detalle del Anuncio | Se registrará y se visualizará el detalle del anuncio ofrecido por el cliente en el sitio web. | Propuesto |
| 6 | Gestionar Cuenta Bancaria | Se almacenará las cuentas bancarias que el cliente puede llegar a tener, ya que se permitirá al sistema cobrarse por publicidad vip del producto, o ser intermediario de pago entre el vendedor y el comprador. | Propuesto |
| 7 | Gestionar Usuarios | Se gestionará la cuenta de acceso que tiene cada cliente, ya que si no está registrado no podrá disponer de los servicios que ofrece este sistema de información. | Propuesto |
| 8 | Administrar Soporte | Se dará soporte al cliente por algún reclamo o disconformidad o con algún cambio referente a su cuenta de usuario o anuncio. | Propuesto |
| 9 | Gestionar Inmueble | Se gestionará el registro de datos específicos del inmueble con el nombre adecuado e imagen seleccionados para su publicación. | Propuesto |
| 10 | Gestionar Galería | Se registrará las imágenes de los productos o inmuebles, el cual quiere vender el usuario. | Propuesto |
| 11 | Gestionar Ubicación | Se registrará la ubicación en la que se encuentra el producto o inmueble, el cual quiere vender el usuario. | Propuesto |
| 12 | Gestionar Trueque | Se podrá realizar algún tipo de intercambio entre el cliente y el vendedor. | Propuesto |
| 13 | Gestionar Anuncio | Aquí se llevará a cabo la publicación de un producto o bien inmueble que el cliente disponga a vender. | Propuesto |
| 14 | Gestionar Transacción | Se realizará transacciones para el pago de algún producto o inmueble que el cliente desee. | Propuesto |
| 15 | Administrar Bitácora | Registro de acciones por parte del usuario | Propuesto |
| 16 | Administrar BackUp | Ninguno | Propuesto |
| 17 | Gestionar Cliente (Android) | Es el manejo de los datos personales del usuario usados mediante la plataforma de Android. | Propuesto |
| 18 | Primera Vista Login (Android) | Implementación de la primera vista (Login) en Android. | Propuesto |
| 19 | Administrar Soporte (Android) | Soporte al cliente en la plataforma Android. | Propuesto |

## 4.3 Elementos:Pila del Sprint (Equipo técnico en la iteración por sprint)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Tarea** | **Descripción** | **Prioridad** |
| 1 | Creación de la Base de Datos | Implementación, Población de la base de datos. | Alta |
| 2 | Vista principal de la pag. Web | Diseño de la interfaz principal | Media |
| 3 | Primera Vista del sistema(Login) | Verificación de ingreso de usuario | Alta |
| 4 | Gestionar Cliente | Es el manejo de los datos personales del usuario usados para contactarlo al momento de hacer la transacción o trueque. | Alta |
| 5 | Gestionar Detalle del Anuncio | Se registrará y se visualizará el detalle del anuncio ofrecido por el cliente en el sitio web. | Media |
| 6 | Gestionar Cuenta Bancaria | Se almacenará las cuentas bancarias que el cliente puede llegar a tener, ya que se permitirá al sistema cobrarse por publicidad vip del producto, o ser intermediario de pago entre el vendedor y el comprador. | Media |
| 7 | Gestionar Usuarios | Se gestionará la cuenta de acceso que tiene cada cliente, ya que si no está registrado no podrá disponer de los servicios que ofrece este sistema de información. | Media |
| 8 | Administrar Soporte | Se dará soporte al cliente por algún reclamo o disconformidad o con algún cambio referente a su cuenta de usuario o anuncio. | Alta |
| 9 | Gestionar Inmueble | Se gestionará el registro de datos específicos del inmueble con el nombre adecuado e imagen seleccionados para su publicación. | Alta |
| 10 | Gestionar Galería | Se registrará las imágenes de los productos o inmuebles, el cual quiere vender el usuario. | Alta |
| 11 | Gestionar Ubicación | Se registrará la ubicación en la que se encuentra el producto o inmueble, el cual quiere vender el usuario. | Alta |
| 12 | Gestionar Trueque | Se podrá realizar algún tipo de intercambio entre el cliente y el vendedor. | Alta |
| 13 | Gestionar Anuncio | Aquí se llevará a cabo la publicación de un producto o bien inmueble que el cliente disponga a vender. | Alta |
| 14 | Gestionar Transacción | Se realizará transacciones para el pago de algún producto o inmueble que el cliente desee. | Alta |
| 15 | Administrar Bitácora | Registro de acciones por parte del usuario | Alta |
| 16 | Administrar BackUp | Ninguno | Alta |
| 17 | Gestionar Cliente (Android) | Es el manejo de los datos personales del usuario usados mediante la plataforma de Android. | Alta |
| 18 | Primera Vista Login (Android) | Implementación de la primera vista (Login) en Android. | Alta |
| 19 | Administrar Soporte (Android) | Soporte al cliente en la plataforma Android. | Alta |

## 4.4 Sprint Review

Se lleva a cabo al final del Sprint, para inspeccionar el incremento y adaptar, si es necesario, el Product Backlog. El Equipo Scrum y las partes interesadas colaboran durante la revisión de lo que se hizo en el Sprint del sistema web para publicidad de productos e inmuebles:

* Objetivo.- presentar al propietario del producto el trabajo realizado.
* El propietario del producto identifica lo que se ha "hecho" y lo que no se ha "hecho".
* El equipo de desarrollo discute lo que anduvo bien durante el Sprint, resolviendo algunas dudas sobre la base de datos para su mejor implementación.
* El equipo de desarrollo demuestra el trabajo que se ha "hecho" sobre el sistema web para publicidad de productos e inmuebles.
* El Scrum Master anuncia el lugar y la fecha de la próxima revisión.

## 4.5 Sprint Restrospective

Es una oportunidad para el Equipo Scrum de inspeccionarse a sí mismo y crear un plan de mejoras para ejecutar durante el siguiente sprint. El propósito de la retrospectiva de Sprint es:

* Identificar qué cosas podemos cambiar para hacer el trabajo más agradable y productivo acerca del sistema web de publicidad utilizando herramientas para la implementación en las próximas iteraciones.
* Identificar los temas principales que salieron bien.- Creación de la base de datos, identificar los casos de uso.
* Crear un plan para la implementación de mejoras con respecto a cómo el Equipo Scrum hace su trabajo.- Podemos mejorar la implementación de algunas tareas de la lista de trabajo dando una revisión más minuciosa y dar el resultado satisfactorio del sistema web.