Clase 05. Testing de Agilidad

# Conceptos básicos de metodologías ágiles

## Metodologías ágiles

Las metodologías ágiles, como su nombre lo indica, son formatos de trabajo (o frameworks) que buscan mayor velocidad en la entrega de producto de calidad.

## Historia de la agilidad

### Registro del ciclo

En 2001 un grupo de directivos, gerentes y personas importantes dentro de la industria de sistemas se juntaron en un retiro grupal para charlar sobre distintas problemáticas que enfrentaban con el ciclo de desarrollo de software. En este viaje que realizaron, debatieron sobre distintas condiciones que buscaban centralizar para agilizar los desarrollos y mejorar la calidad y tiempo de entrega.

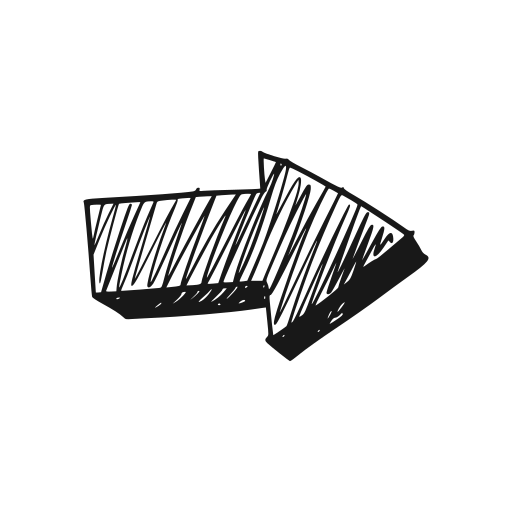
### Nacimiento del manifiesto

Es así como este grupo de profesionales de IT armaron el manifiesto ágil que contiene estos cuatro enunciados:

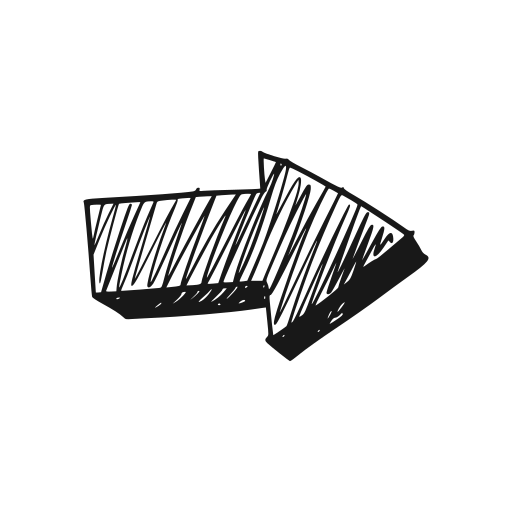
* Individuos e interacciones sobre procesos y herramientas.
* Software funcionando sobre documentación extensiva.
* Colaboración con el cliente sobre negociación contractual.
* Respuesta ante el cambio sobre seguir un plan.

### 

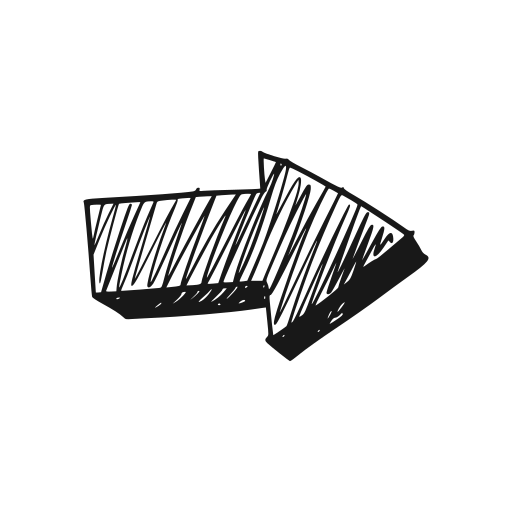
### Valores del manifiesto ágil

**Individuos e interacciones**

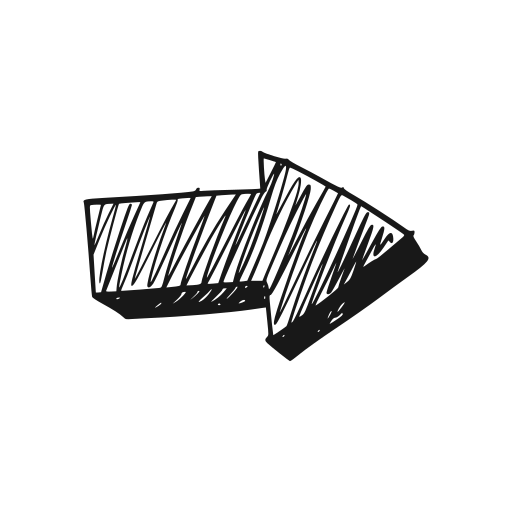
El desarrollo ágil se centra mucho en las personas. Los equipos de personas construyen software, y es a través de la comunicación e interacción continua que se llega a eso, y no en depender de herramientas o procesos, que los equipos pueden trabajar con mayor eficacia.

**Software funcionando**

De cara al cliente, software funcionando es mucho más útil y valioso que documentación y ofrece al equipo de desarrollo la oportunidad de una retroalimentación rápida. Además, cuando un software funcional está disponible mucho antes en el ciclo de vida de desarrollo, puede generar una importante ventaja en el tiempo de comercialización de él.

**Colaboración con el cliente**

Los clientes tienen grandes dificultades para especificar el sistema que necesitan. Colaborar directamente con ellos mejora la posibilidad de comprender exactamente lo que requieren y no generar pérdidas de tiempo (y dinero) en entregas innecesarias o equivocadas.

**Respuesta ante el cambio**

El cambio es inevitable en los proyectos de software. El proceso puede cambiar debido a cientos de circunstancias. El proceso de desarrollo debe estar preparado para cualquier imprevisto que suceda. Por lo tanto, tener flexibilidad o adaptabilidad en el trabajo para aceptar cualquier cambio es más importante que simplemente adherirse estrictamente a un plan.

### Principios del manifiesto ágil

* Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
* Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo.
* Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
* Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
* Los proyectos se construyen en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
* El método más eficiente y efectivo de comunicar información al equipo es la conversación cara a cara.
* El software funcionando es la medida principal de progreso.
* Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
* La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
* La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
* Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
* A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

### Enfoques ágiles

Como anticipamos, existen diversos frameworks ágiles. Es importante conocerlos, para anticipar cómo será la forma de trabajo de la empresa donde estaremos.

Enfoques:

* SCRUM
* PROGRAMACIÓN EXTREMA
* KANBAN
* SCRUMBAN

#### Scrum

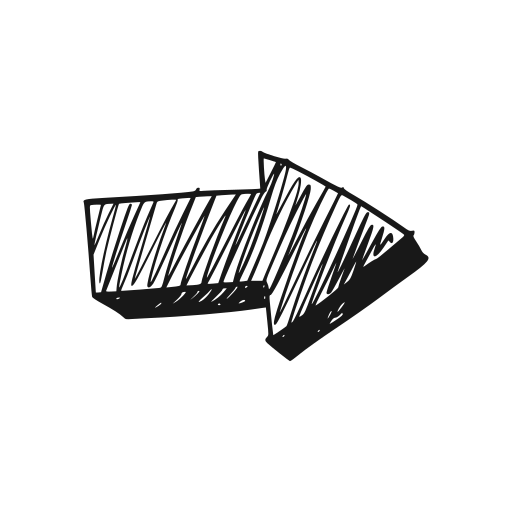
Scrum es un marco de gestión ágil que contiene los siguientes instrumentos y prácticas:

* Sprints
* Incremento de producto
* Trabajo acumulado del producto
* Trabajo acumulado del sprint
* Definición de preparado y hecho
* Acotamiento del tiempo y transparencia

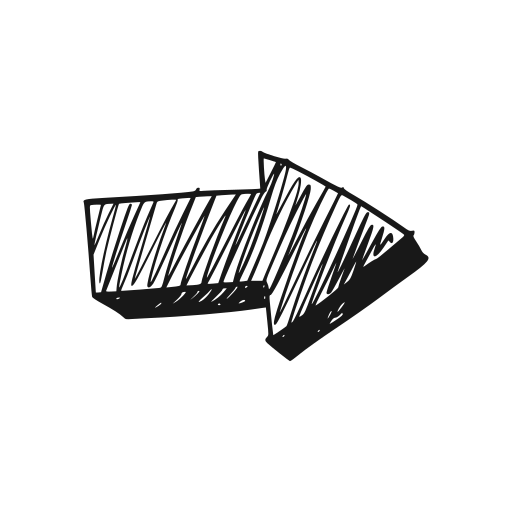
#### Programación extrema

La Programación Extrema (XP) adopta cinco valores para guiar el desarrollo:

* Comunicación
* Simplicidad
* Retroalimentación
* Valor
* Respeto

****XP describe un conjunto de principios como directrices adicionales: humanidad, economía, beneficio mutuo, auto-similitud, mejora, diversidad, reflexión, flujo, oportunidad, redundancia, fallo, calidad, pasos de bebé y responsabilidad aceptada.

Usar para slides de sólo texto. Si no alcanza, no sobrecargar, usar otra con el mismo título para indicar que continúa el mismo módulo.

****XP describe trece prácticas principales:

1. Sentarse juntos
2. Equipo completo
3. Espacio de trabajo informativo
4. Trabajo energizado
5. Programación en pareja
6. Historias
7. Ciclo semanal
8. Ciclo trimestral
9. Holgura
10. Construcción en diez minutos
11. Integración continua
12. Programar probando primero
13. Diseño incremental.

#### Kanban

Es un enfoque de gestión que a veces se utiliza en los proyectos ágiles. El objetivo general es visualizar y optimizar el flujo de trabajo dentro de una cadena de valor añadido. Kanban utiliza tres instrumentos:

* Tablero Kanban
* Limitación de trabajo en curso
* Plazo de ejecución.

#### Scrumban

Scrum y Kanban poseen ciertas similitudes, lo que logra que a veces se toman partes de ambos frameworks en una mezcla de metodologías. De esta forma, poseemos el tablero de Kanban, con los roles y tiempos de Scrum.

### Ceremonias ágiles

Como vimos, las distintas metodologías pueden enlazarse entre sí para generar nuevas formas de trabajo, acorde a las necesidades de los equipos.

#### Tipos de ceremonias ágiles

##### Daily

Son reuniones diarias, de entre 10 y 20 minutos, donde los miembros del equipo comentan sus tareas realizadas y próximas a realizar. También se comenta si hay alguna novedad para el equipo, o se pide ayuda para alguna tarea, si fuese necesario.

##### Planning

Es el espacio de planeamiento del sprint, tomando en cuenta las necesidades próximas del negocio, se define qué tickets se trabajaran en las próximas semanas del proyecto.

##### Review

También conocida como demo, es el momento donde el equipo muestra lo trabajado de forma funcional. Las mismas pueden ser para el propio equipo, o para otros equipos de desarrollo o miembros del negocio.

##### Retro

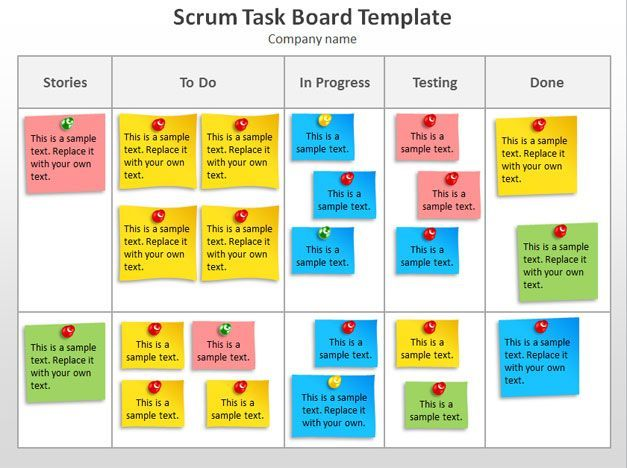
Como la humanidad y la comunicación son parte importante de estos frameworks, se utilizan las retro como retrospectivas del trabajo. Estas suceden al final de cada sprint para revisar qué cosas se trabajaron bien y cuáles mal, que se puede mejorar y plantear acciones de mejora real. También es un lugar para charlar y debatir otras cuestiones que el equipo necesite.

# Tableros ágiles

## Construyendo un tablero

Los tableros sirven para establecer los distintos estados por los que un ticket deberá pasar antes de presentarse como realizado. Cada miembro del equipo deberá establecer qué necesidad de estado necesita para un ticket.

Ejemplo de tablero



En este ejemplo podemos ver cómo se establecieron 5 estados. Stories para que permanezcan esperando.

To do, como tickets listos para ser trabajados.

In Progress para establecer cuáles se están desarrollando.

Testing para identificar cuáles ya se pueden y se están testeando.

Done para marcar cuáles pasaron por todo el circuito.

### Responsabilidad

La construcción y mantenimiento de un tablero ágil, es responsabilidad de todas las partes del equipo. Cada vez que un ticket tiene que cambiar de estado, son los distintos miembros los que deben actualizarlo.

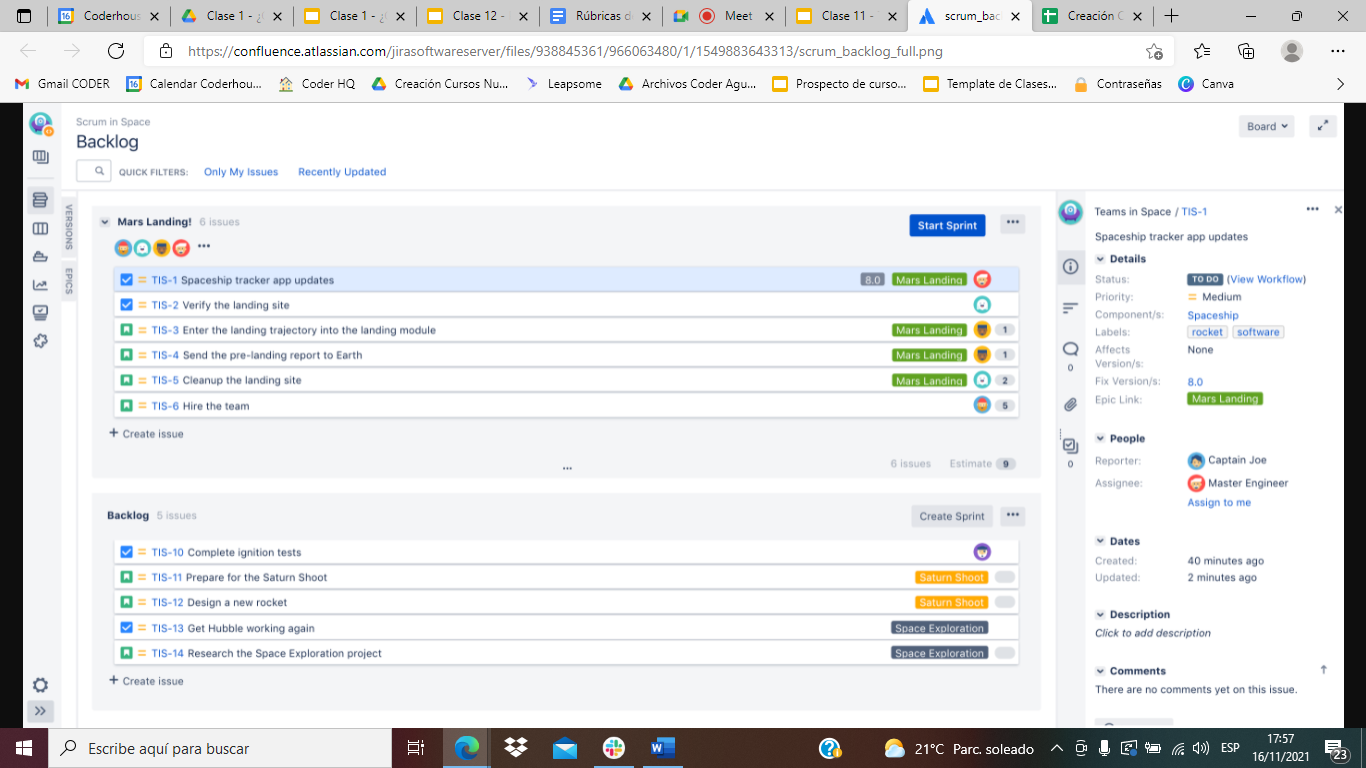
### Tablero en ceremonias

#### Backlog

Otra parte del tablero, quizás un poco más invisible, es el backlog. Es donde se apilan todos los tickets por resolver a futuro. Cada vez que alguien del negocio recibe una nueva necesidad, se encuentran bugs o hay que modificar algo, se crean los tickets en el backlog.

Usar para slides de texto e imagen. Si no alcanza, no sobrecargar, usar otra con el mismo título para indicar que continúa el mismo módulo.

Ejemplo Backlog



#### Tickets en planning

Una vez que llegamos al estado de la planning, tomaremos los tickets que tenemos en el backlog, teniendo en cuenta si quedaron algunos por resolver durante el sprint.

# Testing dentro de agilidad

## Métodos de prueba ágil

Hay algunas prácticas de prueba que pueden realizarse en todo proyecto de desarrollo para producir software de calidad.

Los testers de los proyectos ágiles desempeñan un papel fundamental cuando se trata de guiar el uso de estas buenas prácticas de prueba a lo largo del ciclo de desarrollo.

### MÉTODOS APLICADOS A TESTING

#### Desarrollo Guiado por Pruebas

El desarrollo guiado por pruebas (Test Driven Development o TDD) se utiliza para desarrollar código guiado por casos de prueba automatizados.

Ayuda a los desarrolladores a concentrarse en resultados esperados. Las pruebas se automatizan y se utilizan en la integración continua.

#### Desarrollo Guiado por Pruebas de Aceptación

Se define los criterios de aceptación y las pruebas

durante la creación de las historias de usuario.

Este tipo de desarrollo es un enfoque colaborativo que permite a todos los implicados entender cómo tiene que comportarse el software y qué necesitan los desarrolladores, testers y product owners para garantizar este comportamiento.

#### Desarrollo Guiado por el Comportamiento

Éste permite al desarrollador concentrarse en probar el código basándose en el comportamiento esperado.

Como las pruebas se basan en el comportamiento exhibido del software, las pruebas suelen ser más fáciles de entender para los demás miembros del equipo.

## Cuadrantes de prueba

Los cuadrantes de prueba alinean los niveles de prueba con los

tipos de prueba adecuados en la metodología ágil.

El modelo de cuadrantes de prueba, ayuda a garantizar que todos los tipos y niveles de prueba se incluyan en el ciclo de vida de desarrollo.

### Orientación de los Cuadrantes

Cada cuadrante de prueba, está orientado a distintas cosas:

* El negocio (o usuario)
* La tecnología (o desarrollador)

Otras pruebas del trabajo realizado por el equipo ágil confirman el comportamiento del software. O también algunas pruebas pueden verificar el producto.

#### Cuadrante Q1

Nivel unitario, está orientado a la tecnología y apoya a los desarrolladores. Este cuadrante contiene pruebas unitarias. Estas pruebas deben automatizarse e incluirse en el proceso de integración continua.

#### Cuadrante Q2

Nivel sistema, de cara al negocio, y confirma el comportamiento del producto. Este cuadrante contiene pruebas funcionales, ejemplos, pruebas de historia, prototipos de experiencia de usuario y simulaciones.

Estas pruebas comprueban los criterios de aceptación y pueden ser manuales o automatizadas.

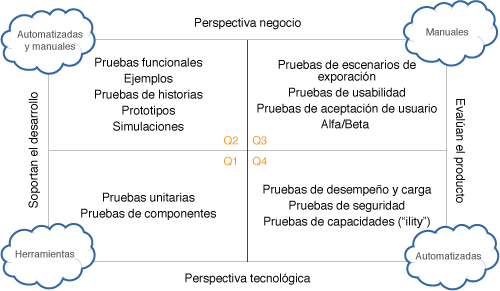
#### Cuadrante Q3

Nivel de aceptación de sistema o usuario, de cara al negocio, y contiene pruebas que critican el producto, utilizando escenarios y datos realistas.

Este cuadrante contiene pruebas exploratorias, escenarios, flujos de procesos, pruebas de usabilidad, pruebas de aceptación de usuario, pruebas alfa y pruebas beta.

#### Cuadrante Q4

Nivel de aceptación de sistema u operativa, orientado a la tecnología, y contiene pruebas que critican el producto. Este cuadrante contiene pruebas de rendimiento, carga, estrés y escalabilidad, pruebas de seguridad, mantenibilidad, gestión de la memoria, compatibilidad e interoperabilidad, migración de datos, infraestructura y pruebas de recuperación.

Fuente: [SG](https://sg.com.mx/revista/45/aseguramiento-agil-calidad)

# Rol del tester en el equipo ágil

## El Rol de un Probador

Se ha hecho una referencia general a los métodos y técnicas ágiles, y al rol de un tester dentro de varios ciclos de vida ágiles. Acá observamos específicamente el rol de un tester en un proyecto que sigue un ciclo de vida Scrum.

### Trabajo en Equipo

Para que un tester ágil tenga éxito, debe poseer buenas prácticas interpersonales, ya que es una de las patas de la agilidad. El mismo necesita buenas prácticas como: interdisciplinario, auto organizado, ubicación común, colaborativo, empoderado, comprometido, transparente, creíble, abierto a la retroalimentación, resiliente.

### Sprint Cero

El sprint cero es la primera iteración del proyecto en la que tienen lugar muchas actividades de preparación. El tester colabora con el equipo en las siguientes actividades: Identificar el alcance del proyecto, planificar, adquirir e instalar las herramientas necesarias, crear una estrategia de prueba, realizar un análisis del riesgo, especificar la definición de hecho y definir cuándo continuar o detener la prueba, entre otras actividades.

### Prácticas de Prueba Ágil

Hay muchas prácticas útiles para los testers en un equipo de ágil:

* Trabajo en pareja: dos miembros del equipo se sientan juntos en un puesto de trabajo para realizar una prueba u otra tarea.
* Diseño de prueba incremental: los casos de prueba se construyen a partir de las historias de usuario y otras pruebas.
* Mapa Mental: los mapas mentales son una herramienta útil a la hora de probar.

### Competencias de un Tester Ágil

Además de estas aptitudes, un tester en un equipo ágil debe ser competente en la automatización de la prueba, el desarrollo guiado por pruebas, la prueba de aceptación, caja blanca, caja negra y la prueba basada en la experiencia.

### Competencias interpersonales

Como las metodologías ágiles dependen, en gran medida, de la colaboración, la comunicación y la interacción entre los miembros del equipo, los testers de un equipo ágil deben tener buenas competencias interpersonales.

#### Tipos de competencias interpersonales

* Ser positivos y estar orientados a las soluciones con los miembros del equipo y los implicados.
* Mostrar un pensamiento crítico, orientado a la calidad y escéptico sobre el producto.
* Recabar información de los implicados de forma activa.
* Evaluar y comunicar con precisión los resultados de las pruebas, el avance de las mismas y la calidad del producto.
* Trabajar eficazmente para definir historias de usuario que puedan ser probadas, especialmente los criterios de aceptación.
* Colaborar dentro del equipo, trabajando en pareja con los programadores y otros miembros del equipo
* Planificar y organizar su propio trabajo
* Responder rápidamente a los cambios, incluyendo la modificación, adición o mejora de los casos de prueba.

## El rol del Tester en un equipo ágil

El papel de un tester dentro de un equipo ágil incluye actividades que generan y proporcionan retroalimentación, no sólo sobre el estado de la prueba y el avance de la misma, sino también sobre la calidad del proceso y del grupo de desarrollo.

### Actividades que forman parte del rol del Tester - Otros roles

* Comprender, implementar y actualizar la estrategia de prueba.
* Medir e informar de la cobertura de la prueba en todas las dimensiones de cobertura aplicables.
* Configurar, utilizar y gestionar los entornos de prueba y los datos de prueba.
* Informar de los defectos y trabajar con el equipo para resolverlos.
* Entrenar a otros miembros del equipo en los aspectos relevantes de la prueba.
* Asegurar el uso adecuado de las herramientas de prueba.
* Asegurar que se programen las tareas de prueba adecuadas durante la planificación de la entrega y la iteración.
* Colaborar activamente con los desarrolladores y los implicados del negocio para aclarar los requisitos.
* Participar de forma proactiva en las retrospectivas del equipo, sugiriendo e implementando mejoras.

## Glosario

**Scrum:** es una metodología de trabajo que permite agilizar la entrega de valor al cliente en iteraciones cortas de tiempo.

**Scrum Master:** Rol de Scrum. Es responsable de que las prácticas, valores y reglas se realicen. Nexo entre la gerencia y el equipo. Se asegura que se resuelven los impedimentos y se toman decisiones rápidas.

**Sprint:** es un periodo fijo de tiempo (entre 2 a 4 semanas) durante el cual el trabajo tiene que ser completado.

**TDD:** Desarrollo guiado por pruebas.

**BDD:** Desarrollo guiado por comportamiento.

**ATDD:** Desarrollo guiado por pruebas de aceptación.

#### ¿QUIERES SABER MÁS? TE DEJAMOS MATERIAL AMPLIADO DE LA CLASE

* [Historia de la Agilidad](https://open.spotify.com/episode/4gFBJkzljoTY84K1yCCcXA?si=7d3f053c20854242) | RedHat
* [Programa Estudio Agil](https://141647fc-7a5b-4407-a84a-78567bb3f5b0.filesusr.com/ugd/acfdb9_7f9fe7d65c394d8e9d888b4527a069d8.pdf) | SSTQB

#DEMOCRATIZANDOLAEDUCACIÓN