

PROGRAMACIÓN WEB FULL NIVEL 2 STACK

Módulo 3: Metodologías de desarrollo



Metodologías de desarrollo de software

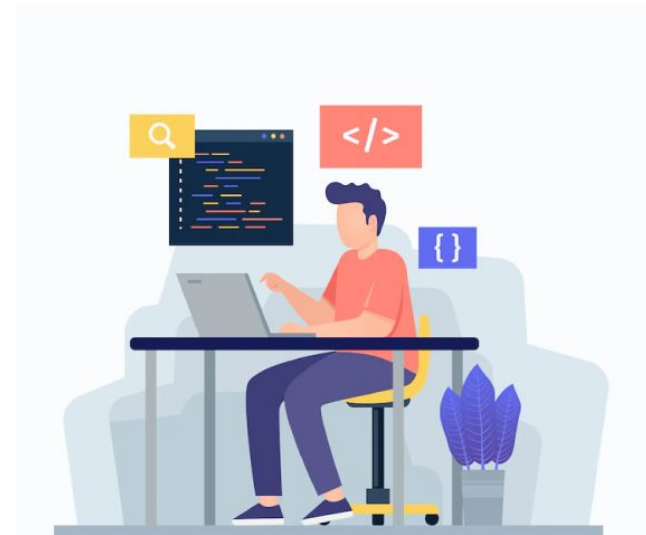
Las metodologías de desarrollo de software engloban una serie de métodos y técnicas organizativas aplicadas a la creación y diseño de software informático. Estos métodos persiguen organizar de manera eficiente a los equipos implicados para que desarrollen con éxito sus funciones teniendo en cuenta aspectos como la dificultad del proyecto, la planificación, e presupuesto, los costes, los profesionales disponibles, e lenguaje que se va a utilizar, etc.



Metodologías tradicionales

Su principal característica es que definen de manera rígida todos los requisitos cuando comienza el proyecto de creación del software, de tal forma que no permiten los cambios y los procesos de desarrollo no son nada flexibles, justo lo contrario que defienden las metodologías ágiles de desarrollo de software.

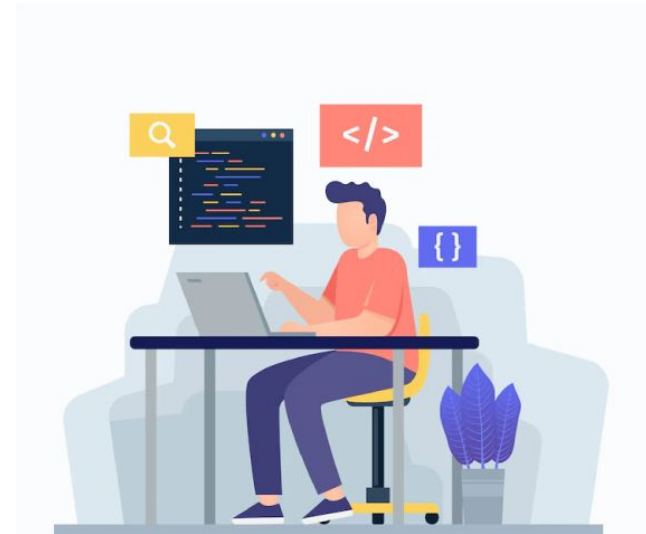
Las metodologías tradicionales apuestan por una organización del trabajo lineal y no se puede comenzar una fase sin terminar la anterior. Tampoco permite una vuelta atrás cuando se avanza en el proceso, por lo que si es necesario implementar algún cambio de última hora no se adaptan bien creando desajustes importantes.



Metodologías ágiles

Se caracterizan por su enorme agilidad y flexibilidad en los procesos de trabajo. Gracias a su implementación, los equipos trabajan de una manera mucho más eficiente y productiva porque saben exactamente lo que tienen que hacer y hay una comunicación fluida entre ellos. Por otro lado, la metodología se adapta perfectamente a los imprevistos y las necesidades que surgen durante el proceso porque los ciclos se reducen y son mucho más rápidos.

Las metodologías ágiles permiten la creación de equipos de trabajo independientes y autosuficientes que están al tanto de todo el proceso de desarrollo. Además, el **cliente** también puede aportar y corregir porque supervisa en tiempo real cómo avanza su proyecto.



Metodologías ágil: Lean

Se caracteriza porque apuesta por las personas y su compromiso con el proyecto, dejando en un segundo plano los costes. Selecciona equipos de trabajo muy cualificados para desarrollar las tareas en un tiempo reducido. Este método potencia la cohesión del equipo, el aprendizaje y reacciones rápidas durante el proceso.

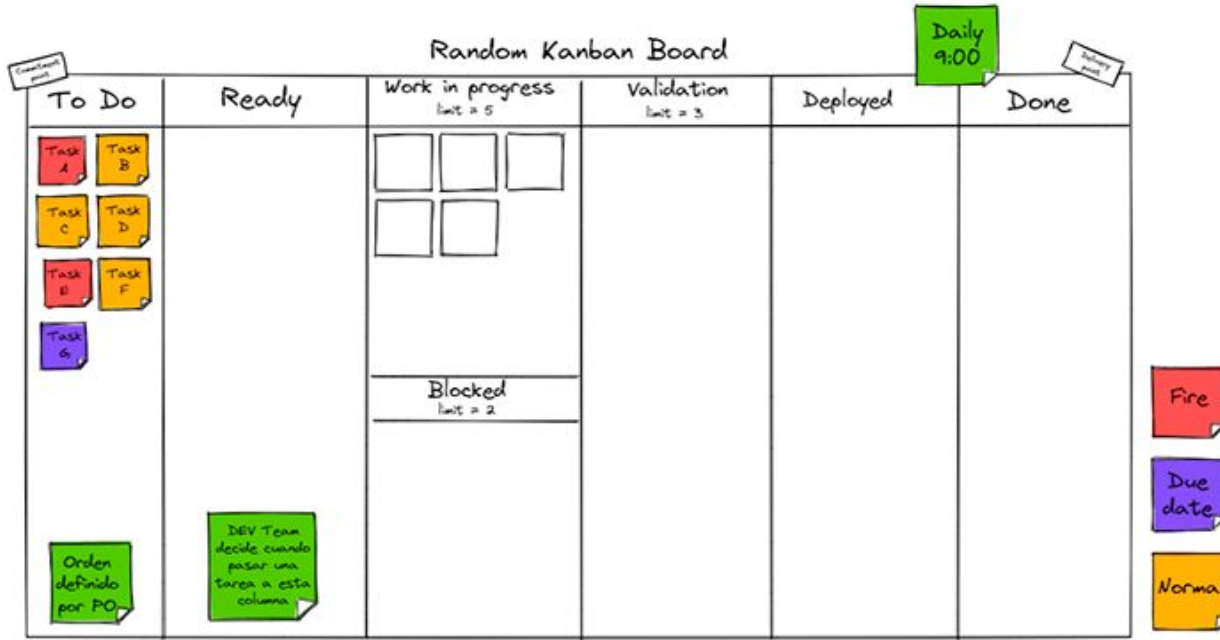


Metodologías ágil: Kanban

Este método fue desarrollado por la marca de coches japonesa Toyota. Su principal característica es que divide el proceso en tareas mínimas y las coloca en un tablero de trabajo donde se pueden ver las tareas que están en curso, pendientes o finalizadas. El trabajo se centra en las tareas prioritarias y con el valor del producto siempre en primer lugar.

Metodologías ágil: Kanban

ejemplo tablero Kanban:



Scrum

La metodología Scrum permite abordar proyectos complejos desarrollados en entornos dinámicos y cambiantes de un modo flexible. Está basada en entregas parciales y regulares del producto final en base al valor que ofrecen a los clientes. Dicho en otras palabras: Scrum sirve para mejorar el trabajo colaborativo entre equipos.

Se trata de una metodología que ayuda a los equipos a aprender y organizarse en base a las experiencias a la vez que aborda problemas e invita a reflexionar sobre los éxitos y fracasos. Todo ello bajo una serie de herramientas y recursos que permite a los equipos organizarse con mayor agilidad.

Es una opción de gestión ideal para acometer proyectos desarrollados en entornos complejos que exigen rapidez en los resultados y en los que la flexibilidad es un requisito imprescindible. Scrum ofrece agilidad y el, resultado, siempre, valor.

Scrum características del equipo

Equipos autónomos: los equipos Scrum están pensados para operar sobre la marcha, con un orden y dinámica únicos que carecen de jerarquía. Estos equipos se consideran autoorganizados, exhiben autonomía, crecimiento continuo y colaboración.

Fases de desarrollo solapadas: las personas de un equipo Scrum deben trabajar para sincronizar sus ritmos para cumplir con los plazos de entrega. En algún momento del desarrollo, el ritmo de cada individuo comienza a solaparse y sincronizarse con el de los demás y, finalmente, se forma un ritmo colectivo dentro del equipo.

Aprendizaje múltiple: Scrum es un marco que se basa en gran medida en prueba y error. Los miembros del equipo Scrum también tienen como objetivo mantenerse al día con las condiciones cambiantes del mercado. Es por eso que el aprendizaje es fluído y rota entre los diferentes miembros de la organización.

Seguimiento sin control: como mencionamos, los equipos Scrum se autoorganizan y operan como una pequeña startup, pero eso no significa que no exista ninguna estructura. Al crear puntos de control a lo largo del proyecto para analizar las interacciones y el progreso del equipo, los equipos Scrum mantienen el control sin obstaculizar la creatividad.

Metodologías de desarrollo **Perfiles** **SCRUM**



El Product Owner

Es responsable de maximizar el valor del producto resultante del trabajo del equipo Scrum. La forma en que se hace esto puede variar ampliamente entre organizaciones, equipos Scrum e individuos.

Los Product Owners maximizan el valor del producto al representar y expresar la voz del cliente durante la duración del proyecto. Ellos son los responsables de entender las necesidades de los clientes, sus motivaciones y qué necesitan. Un producto no es útil para sus clientes si ese producto no cumple con sus expectativas y no satisface sus necesidades.

Las funciones de los product owners son:

- Desarrollar y comunicar explícitamente el objetivo del producto.
- Crear y comunicar claramente los elementos del Backlog del producto (el Backlog del producto contiene todas las características, requisitos y actividades asociadas con los entregables para lograr el objetivo del proyecto).
- Asegurarse de que la cartera de productos sea transparente, visible y entendible para el equipo.

El Scrum Master

Una responsabilidad clave del Scrum Master es ayudar al equipo a comprender y seguir la teoría de Scrum.

El Scrum Master es, por tanto, el responsable de la efectividad del Scrum Team. Lo hacen al permitir que el equipo Scrum mejore sus prácticas, dentro del marco de Scrum.

El Scrum Master se asegura de que se produzcan reuniones importantes, como las Dailys. De la misma manera que un entrenador estaría al tanto del timing en un partido, el Scrum Master tiene la tarea de asegurarse de que la reunión se mantenga dentro del tiempo apropiado.

El Scrum Master actúa como entrenador del Scrum Team: estimulan y motivan al equipo a construir el producto en el marco de tiempo.

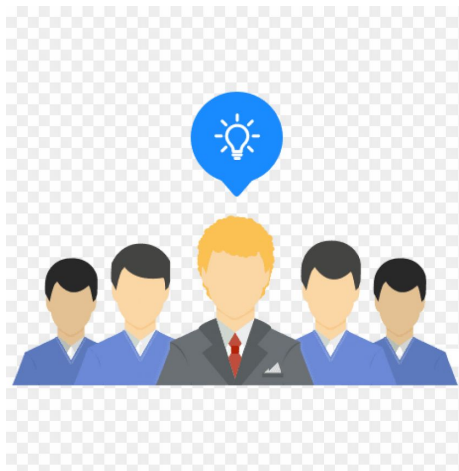


Project Manager

El rol del **Scrum Master** a veces se confunde con el rol del **project manager** o gerente de proyectos. Si bien los dos roles comparten habilidades y cualidades relacionadas, son bastante diferentes.

Un Scrum Master es responsable de ayudar al equipo a comprender la teoría y la práctica de Scrum. Aseguran que los eventos de Scrum se lleven a cabo y ayudan al equipo a concentrarse en entregar valor y a eliminar los impedimentos.

La diferencia con un project manager habitual, es que el Scrum Master no asuma la gestión de cambios en el alcance o scope de un proyecto ni la gestión de las prioridades.



Scrum Team

Es el equipo encargado de desarrollar y entregar el producto. Su trabajo es imprescindible: estamos hablando de una estructura horizontal auto-organizada capaz de auto-gestionarse a sí misma.



Stakeholder

En el mundo de los negocios, **los stakeholders son aquellos individuos o grupos que tienen interés e impacto en una organización y en los resultados de sus acciones**. Algunos de los ejemplos más comunes de stakeholders son los empleados, los accionistas, los clientes, los proveedores, los gobiernos y las comunidades.

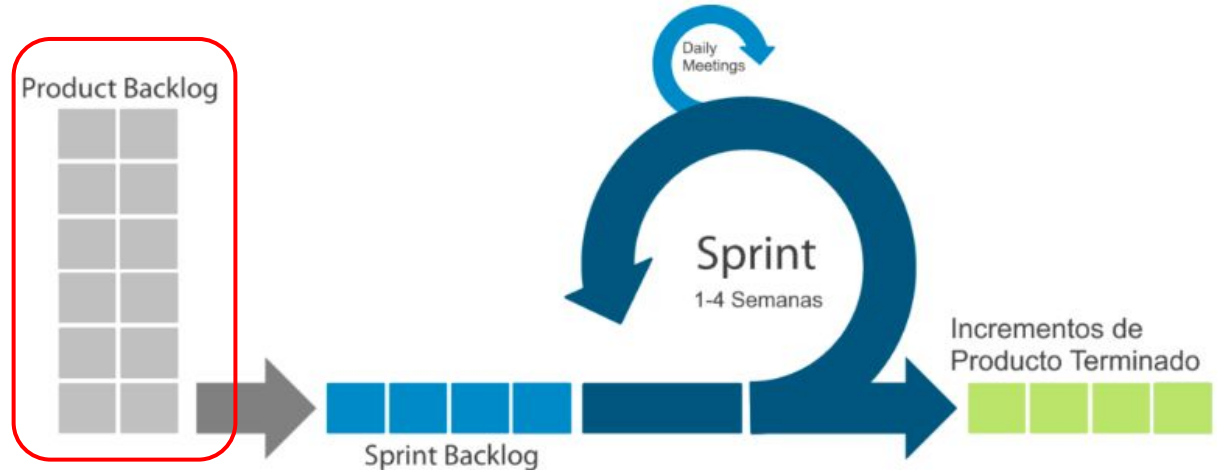


Metodologías de desarrollo **Proceso** **Scrum**



Product Backlog

El proceso comienza con la elaboración del llamado **Product Backlog**. Se trata de un archivo genérico que recoge el conjunto de tareas, los requerimientos y las funcionalidades requeridas por el proyecto. Cualquier miembro del equipo puede modificar este documento pero el único con autoridad para agregar prioridades es el **Product Owner**.

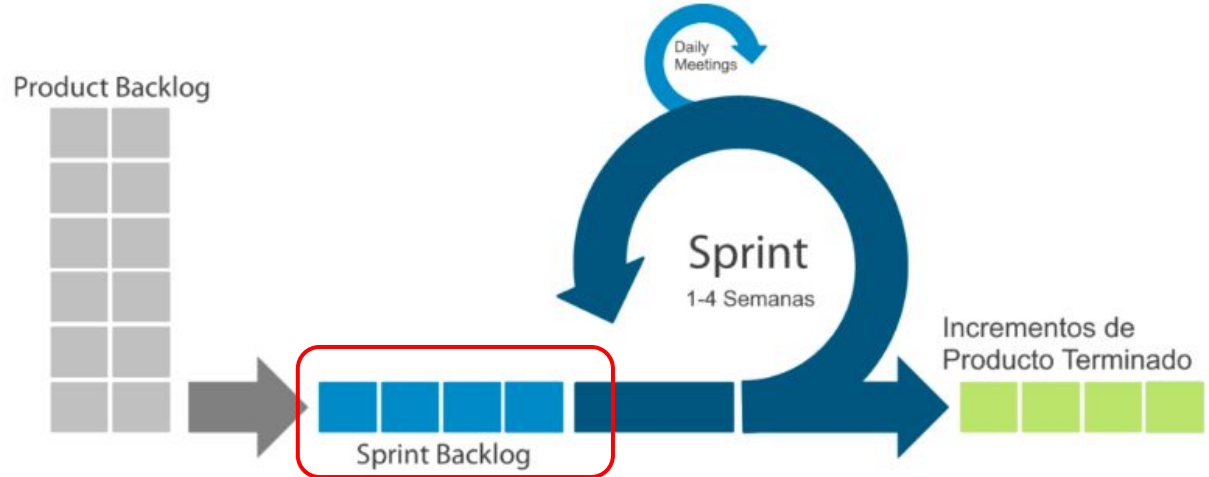


Sprint Backlog

Es un documento que recoge las tareas a realizar y quién las desempeña. Es interesante asignar las horas de trabajo que va a suponer realizar cada una de ellas y asignarlas a un coste. Si su volumen es muy grande, crear metas intermedias será un acierto.

El Sprint es el periodo en el que se realizan todas las acciones pactadas en el Sprint Backlog , que supone entregas parciales para ir testeando el producto final.

El ciclo anterior deberá repetirse hasta que todos los elementos del Blacklog hayan sido entregados. Entre los distintos Sprints no se deben dejar tiempos sin productividad.



Sprint Planning Meeting

¿Quién no ha perdido horas de trabajo inútiles en reuniones poco productivas porque estaban mal preparadas? Esto no tiene cabida en los métodos ágiles. Cada minuto cuesta dinero. Las reuniones han de estar también planificadas, como una parte más de proceso. En este «**Sprint Planning Meeting**» el **Product Owner** prioriza las tareas contenidas en el **Product Backlog**.

Con estas tareas en mente se determina el **objetivo** del **nuevo sprint** priorizando las tareas a realizar por el Scrum Team y asignando tiempo a cada una de ellas. **El objetivo debe ser alcanzable y el equipo sólo abordará un conjunto de tareas asumible.**

Diariamente se hace un seguimiento del proyecto en esta reunión en la que se controla el cumplimiento de las tareas asumidas. Quizás has oído hablar de la **Daily Scrum**, que es el nombre adoptado del inglés. En dicha cita se pactan los objetivos para el día siguiente y se analizan los posibles problemas que hayan limitado o impedido directamente el cumplimiento de los objetivos.

Sprint Daily

El Daily Scrum, conocido comúnmente sólo como “**La Daily**”, es una reunión diaria de aprox 30 minutos en la que participa exclusivamente el **Scrum Team**.

En esta reunión todas y cada una de las personas del **Scrum Team** responden a las siguientes preguntas:

- ¿Qué hice ayer para contribuir al Sprint Goal?
- ¿Qué voy a hacer hoy para contribuir al Sprint Goal?
- ¿Tengo algún impedimento que me impida entregar?

Mucha gente confunde el **Daily Scrum** con el Standup Meeting, sin embargo, este último es una práctica de eXtreme Programming (XP) orientada a controlar y gestionar el trabajo motivada por un manager, mientras que el primer término, Daily Scrum, hace referencia a la práctica que permite la inspección y adaptación a través de la auto-organización del equipo.

Sprint Review

En el **Burn Down** donde marcamos el estado y la evolución del **proyecto** indicando las tareas y requerimientos pendientes de ser retratados.

En esta fase final del **Sprint**, se revisa todo el trabajo, una buena oportunidad para tener **feedback** sobre el desarrollo del producto.

Suele durar una hora por cada semana. La persona encargada de realizar este trabajo es el Product Owner, mientras que el **Scrum Master** se asegura de que esto sea así y de que se cumplan los tiempos establecidos.

Entonces ¿qué ocurre en el Sprint Review? Dentro de esta fase, en la que ya sabemos que el **Product Owner** es el encargado total de la revisión, se encarga de explicar los items del **Product Backlog** y asegurarse de que hayan sido finalizados.

A partir de ahí se hace una review del proyecto y se tratan los siguientes pasos a seguir. Si hay una respuesta por parte del cliente, el **Product Owner reorganizaría el Product Backlog**.

Sprint Retrospective

El objetivo de la retrospectiva es hacer de reflexión sobre el último Sprint e identificar posibles **mejoras** para el próximo. Aunque lo habitual es que el **Scrum Master sea el facilitador**, es normal que distintos miembros del equipo Scrum vayan rotando el rol de facilitador durante la retrospectiva.

Un formato común es analizar qué ha ido bien durante el Sprint, qué ha **fallado** y qué se puede **mejorar**. Este formato se puede facilitar pidiendo a los miembros del equipo Scrum que escriban notas –en post-its- para luego agruparlas y votar aquellos ítems más relevantes, dando la oportunidad a todos de hablar y expresar sus inquietudes.



Metodologías de desarrollo **Software** de Gestión de --- proyectos



Software de gestión de proyectos

Un software de gestión de proyectos es un conjunto de herramientas que mejoran la eficiencia del equipo de la empresa que está llevando a cabo el proyecto y le permite controlar mejor el tema y los recursos asignados.

Dentro del mundo de los proyectos de desarrollo de software se suelen usar distintos softwares, entre ellos se destacan: Jira, redmine, trello.



Jira

Es uno de los software de gestión de pr

<https://www.atlassian.com/es/software/jira>

Free	Standard	Premium	Enterprise
\$0 Siempre gratuito para 10 usuarios Empieza ya	\$7.50 por usuario (media) \$75 al mes Iniciar versión de prueba	\$14.50 por usuario (media) \$145 al mes Iniciar versión de prueba	Facturación anual. Pasa al ciclo de facturación anual para ver precios de Enterprise. Contacto
Para que equipos pequeños planifiquen y hagan un seguimiento más eficiente del trabajo	Para equipos en crecimiento centrados en conseguir más cosas, juntos	Para organizaciones que necesitan escalar la colaboración y llevar un seguimiento del trabajo	Para empresas con necesidades globales de escalado, de seguridad y de gobierno



Scrum

[Más información →](#)



Kanban

[Más información →](#)



Supervisión de errores

[Más información →](#)



DevOps

[Más información →](#)

Jira Software

Añadir detalles del proyecto

Puedes cambiar estos detalles en cualquier momento en la configuración de tu proyecto.

Nombre

Clave 

Plantilla

Cambiar plantilla



Kanban RECOMENDADO

Visualiza tu proyecto y hazlo avanzar utilizando incidencias en un tablero lleno de posibilidades.

Tipo

Tipo de cambio



Gestionados por el equipo RECOMENDADO

Controla tus propios procesos y prácticas de trabajo en un espacio independiente.

Atrás

Crear proyecto

Jira

Selecciona ahora algunas
herramientas y te ayudaremos a
conectarte a ellas más adelante.



Slack



Microsoft Teams



GitHub



GitLab



Zendesk



Google Sheets



CircleCI



Sentry




Jenkins

Omitir

Siguiente

Jira

 Jira Software

Tu trabajo ▾

Proyectos ▾


Filtros ▾

Paneles ▾


Personas ▾


Aplicaciones ▾

Crear


 **CURSO SILICON**
Proyecto de software


PLANIFICACIÓN


 Hoja de ruta


 **Tablero**

DESARROLLO

 Código

 Páginas de proyectos

 Añadir acceso rápido

 Configuración del pr...

¿Necesita más de Jira tu equipo? [Consigue una versión de prueba gratuita de nuestro plan Standard.](#)


Proyectos / CURSO SILICON

En un **tablero** se visualiza el proceso de tu equipo, se gestiona el estado de las tareas y se optimiza el flujo de trabajo.

1/4 **Siguiente**

POR HACER	EN CURSO	LISTO ✓
+ Crear incidencia		

Jira

 Jira Software

Tu trabajo ▾

Proyectos ▾


Filtros ▾

Paneles ▾


Personas ▾


Aplicaciones ▾

Crear


 **CURSO SILICON**
Proyecto de software


PLANIFICACIÓN


 Hoja de ruta


 **Tablero**

DESARROLLO

 Código

 Páginas de proyectos



 Añadir acceso rápido

 Configuración del pr...

¿Necesita más de Jira tu equipo? [Consigue una versión de prueba gratuita de nuestro plan Standard.](#)

Proyectos / CURSO SILICON

Tablero CS

POR HACER

+ Crear incidencia

EN CURSO

LISTO ✓

+

Tu tablero comienza con un simple flujo de trabajo de tres pasos. El trabajo se gestiona de forma abierta, por lo que todo el mundo puede seguir el estado de una tarea a medida que avanza. Cuando lo tengas todo listo para personalizar el flujo de trabajo, añade o edita las columnas.

2/4

Siguiente

Jira

 Jira Software

Tu trabajo ▾

Proyectos ▾

Filtros ▾

Paneles ▾

Personas ▾

Aplicaciones ▾

Crear

 **CURSO SILICON**
Proyecto de software

PLANIFICACIÓN

 Hoja de ruta

 **Tablero**

DESARROLLO

 Código

 Páginas de proyectos

 Añadir acceso rápido

 Configuración del pr...

¿Necesita más de Jira tu equipo? [Consigue una versión de prueba gratuita de nuestro plan Standard.](#)

Proyectos / CURSO SILICON

Tablero CS



POR HACER

+ Crear incidencia

EN CURSO

...

Establecer límite de columna

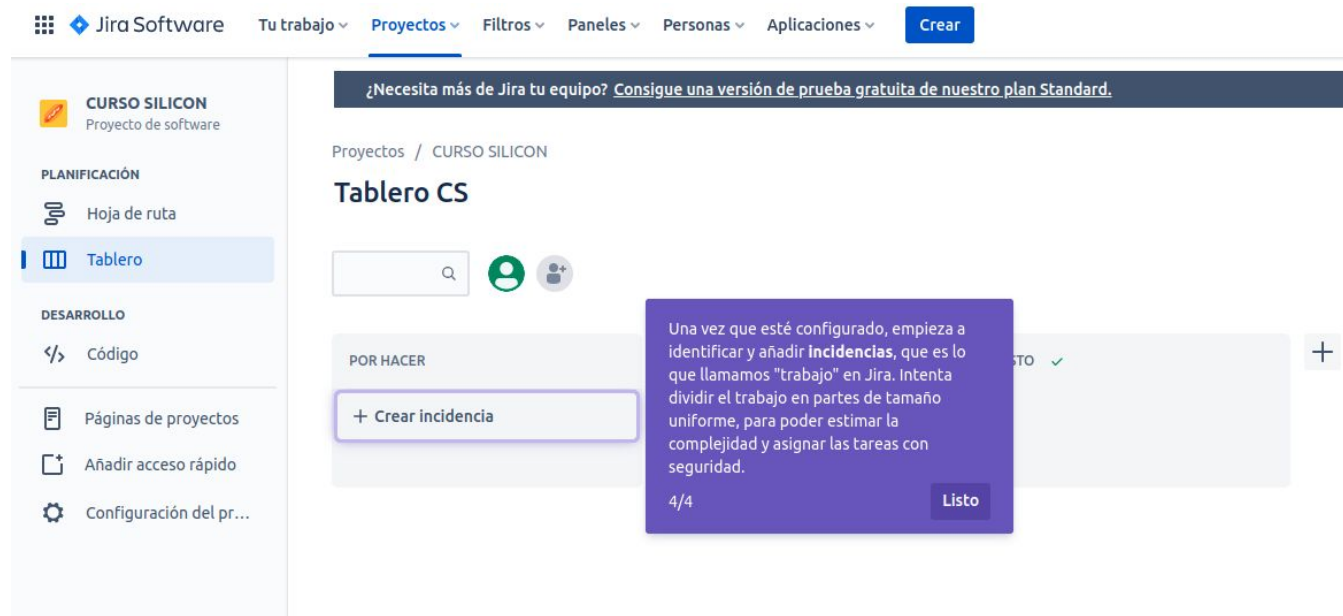
Eliminar

Limitar el trabajo en curso mantiene la atención y evita los cuellos de botella. Cuando tengas una idea de la capacidad, establece un límite de tareas y te avisaremos cuando hayas alcanzado el máximo.

3/4

Siguiente

Jira



Jira Software Tu trabajo ▾ **Proyectos** ▾ Filtros ▾ Paneles ▾ Personas ▾ Aplicaciones ▾ **Crear**

CURSO SILICON
Proyecto de software

PLANIFICACIÓN

- Hoja de ruta
- Tablero**

DESARROLLO

- Código
- Páginas de proyectos
- Añadir acceso rápido
- Configuración del pr...

Proyectos / CURSO SILICON

Tablero CS

¿Necesita más de Jira tu equipo? [Consigue una versión de prueba gratuita de nuestro plan Standard.](#)

+ Buscar [icono de lupa] [icono de usuario] [icono de grupo]

POR HACER

+ Crear incidencia

Una vez que esté configurado, empieza a identificar y añadir **Incidencias**, que es lo que llamamos "trabajo" en Jira. Intenta dividir el trabajo en partes de tamaño uniforme, para poder estimar la complejidad y asignar las tareas con seguridad.

4/4 **Listo**



Slack es una aplicación de mensajería para empresas que conecta a las personas con la información que necesitan. Slack transforma la manera en que se comunican las organizaciones reuniendo a las personas para que trabajen como un equipo unificado.





Slack
Qué es y cómo funciona

Metodologías de desarrollo



Roles en un equipo de desarrollo

PROJECT MANAGER

El project manager es el responsable general de todo el equipo. Sinceramente, los gestores de proyectos no necesitan aprender mucho del sistema de codificación. Solo tienen que estar ahí para gestionar el proyecto, supervisar el presupuesto, el calendario y quizás intentar resolver todos los riesgos posibles. Para cumplir todas las especificaciones del cliente, el gestor de proyectos tiene que supervisar el proyecto cuidadosamente.

DISEÑADOR DE UI/UX

UI significa "interfaz de usuario", mientras que UX significa "experiencia de usuario". Los diseñadores de UI/UX a veces son también desarrolladores, o son diseñadores gráficos con un gran conocimiento del diseño de aplicaciones. Son responsables de crear una experiencia óptima para los usuarios. Tanto si el producto es una aplicación digital como una aplicación basada en la web, los diseñadores de UI/UX investigan cómo ofrecer la mejor experiencia a los usuarios finales.

ARQUITECTOS DE SOFTWARE

Estos miembros del equipo son responsables del diseño de alto nivel de la aplicación, ya sea una aplicación basada en la web o una aplicación móvil. Los arquitectos de software se esfuerzan por seguir las normas técnicas. Además, deben saber que existen normas relacionadas con la codificación, las herramientas y las plataformas. Al igual que los ingenieros de software, los arquitectos de software se aseguran de que sus compañeros desarrolladores sigan ciertos protocolos.

El Tester / quality assurance (QA)

Las pruebas son una parte importante para asegurar que el software funciona de la manera que debería. El papel de 'tester' se realiza a menudo por los desarrolladores para los aspectos técnicos y los usuarios para los aspectos funcionales. Un problema que surge de hacer a los desarrolladores probar su propio código es que, no importa lo bueno que sean, se ven influidos por la forma de su código fue creado. Cuando se prueba, se tendrá en cuenta esas mismas situaciones que ya se tuvieron en cuenta a la hora de escribirlo.

Si se prueba código de otra persona, se puede pensar en escenarios que la otra persona no los pensó. Así que incluso si no se tiene un equipo de Testers dedicado, es una buena idea que cada desarrollador pruebe código de otro desarrollador, en lugar del suyo propio.

El DevOps

El ingeniero de DevOps incorpora procesos, herramientas y metodologías para equilibrar las necesidades durante todo el ciclo de vida del desarrollo de software, desde la programación y la implementación hasta el mantenimiento y las actualizaciones.

El objetivo de DevOps es la unificación y la automatización de procesos, y los ingenieros de DevOps son fundamentales para las tareas relacionadas con la combinación de código y el mantenimiento y la gestión de aplicaciones.

Dentro de un entorno ágil, los desarrolladores, los administradores de sistemas y los programadores pueden trabajar en el mismo producto de forma aislada, sin compartir la información que se necesita para garantizar el valor para el usuario. Los ingenieros de DevOps reducen esa complejidad, ya que acortan la brecha entre las acciones que se necesitan para cambiar en poco tiempo una aplicación y las tareas que se realizan para mantener su confiabilidad.

El Tech Lead

Un technical leader es un desarrollador de software que tiene a su mando un equipo completo de desarrollo. Cuenta con la responsabilidad de que los productos que se lleven a cabo cuenten con la calidad técnica correspondiente.

El Tech Lead debe crear una visión técnica con el equipo de desarrollo y trabajar con todo el grupo para llegar al objetivo. En toda la fase de creación, el technical leader suma las habilidades de otros roles diferentes, eso sí, siempre ligado al código y programando.

Entre las habilidades con las que debe contar un Tech Lead destacan las siguientes:

- Empatía. Debe ponerse en el lugar de los miembros del equipo.
- Motivación. Es importante que el Tech Lead pueda motivar al equipo, además, debe ser capaz de resolver problemas y seguir mejorando sus conocimientos.
- Comunicación. Debe ser capaz de tener un flujo continuo de información con los miembros de su equipo.
- Visión en conjunto.
- Gestión de equipo. Definir tiempo, objetivos y tareas.
- Delegación de tareas. Debe ser capaz de repartir tareas y compartir responsabilidades.