

CMYK 5

Prácticas Ágiles para el Desarrollo de Software





Ramiro Olivencia

- Analista en Sistemas / UTN
- Lead Dev en EY GDS
- ~~Serum~~ Master certificado
- Mentor en FrontendCafé



FrontendCafé



Service
Design
Club.

Temas de la charla

- Arribar a una definición de agilidad
- Aprender sobre prácticas, metodologías y frameworks ágiles
- Conocer sobre prácticas ágiles para usar en nuestro proceso de desarrollo y construir nuestra metodología



Dinámica

- Encuestas y nubes de palabras
- Menciones a recursos
- Metáforas visuales
- Presentar para, después, profundizar
- ¡Preguntas!



Agilidad

- Enfocar el trabajo en entregar valor
- Feedback rápido, reflexión y análisis de resultados
- Colaboración, tanto entre integrantes como con clientes
- Búsqueda y experimentación para mejora constante





**Para vos, ¿qué es la
agilidad?**



slido



Para vos, ¿qué es la agilidad?

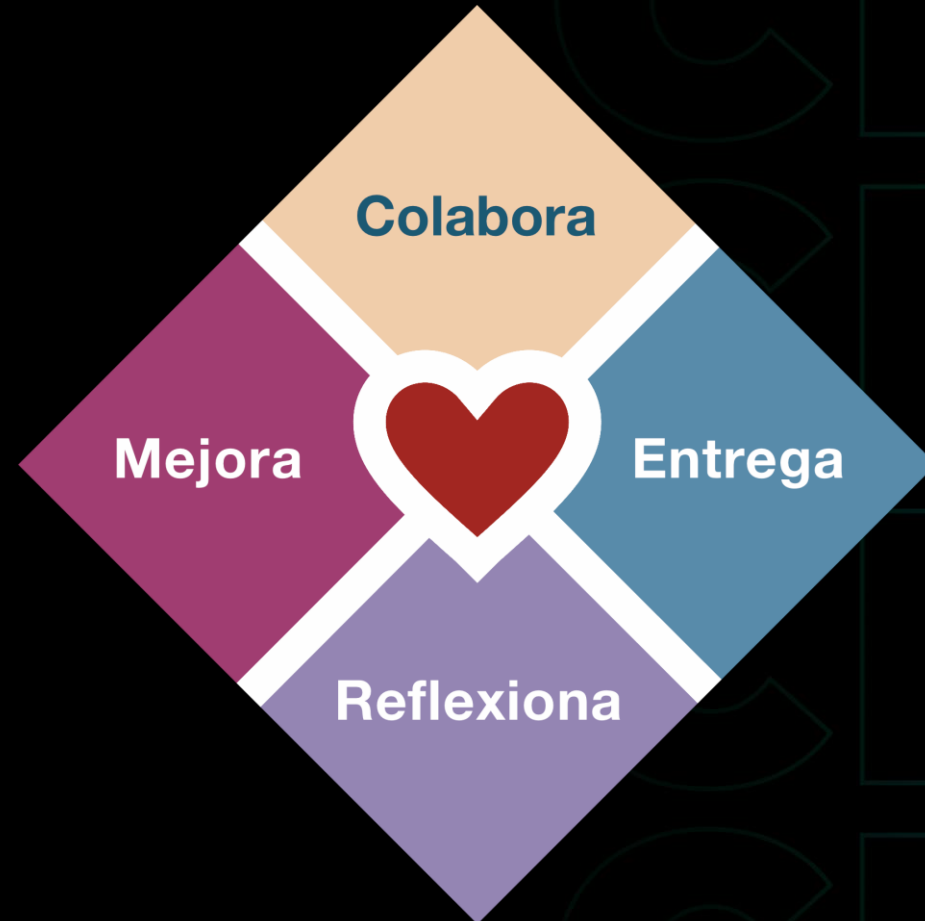
ⓘ Start presenting to display the poll results on this slide.

... según el manifiesto ágil

- **Individuos e interacciones,**
por sobre procesos y herramientas
- **Entrega de valor,**
por sobre documentación extensiva
- **Colaboración con el cliente,**
por sobre negociación contractual
- **Respuesta al cambio,**
por sobre seguir un plan



... según Heart of Agile





¿Y una metodología?



slido



¿Y qué dirías que es una metodología?

ⓘ Start presenting to display the poll results on this slide.

Definiciones

Herramienta

Recursos materiales o digitales que usamos para ejecutar una determinada tarea o proceso.

Proceso

Enfoque con un conjunto definido de reglas, actividades y prácticas para resolver problemas específicos bajo un abordaje repetible.

Framework

Marco de gestión que provee de reglas y líneas guía para lograr un objetivo, usualmente sin detalles o rigidez excesiva.

Metodología

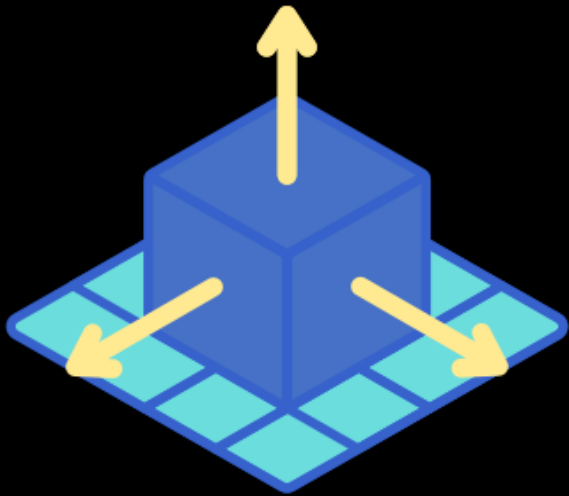
Conjunto definido, articulado y gestionado de prácticas para la ejecución del trabajo dentro un proyecto específico.

Práctica

Actividades atómicas realizadas de forma regular por un equipo para llevar el trabajo a término.



Metodologías



Escala



Precisión



Comunicación



Frameworks

- **XP**
- **Scrum** (bueno, había que nombrarlo)
- **Kanban**
- **SAFe**
- **Disciplined Agile**
- **Heart of Agile**



Prácticas – Parte 1

- Definición de hecho
- Visualización del trabajo
- Limitación el trabajo concurrente (WIP)
- Unit testing
- Pair & mob programming
- Refinamiento del backlog



Definición de Hecho



Mediante una **checklist**, equipo y clientes deciden cuándo un ítem de trabajo puede ser considerado **hecho**.

Permite determinar y reflejar los posibles estados de los ítems de trabajo del proyecto y determinar **criterios de calidad y aceptación** sobre el trabajo realizado.



Visualización del Trabajo

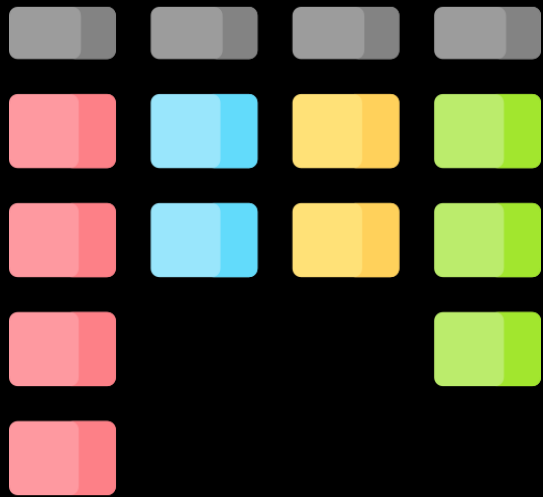


Busca brindar una representación visible a nuestro **modelo de flujo de trabajo**.

Permite que tanto equipo como clientes accedan a información condensada acerca del estado actual de los ítems de trabajo del proyecto, generando un **radiador de información** de fácil consumo.



Limitación del WIP

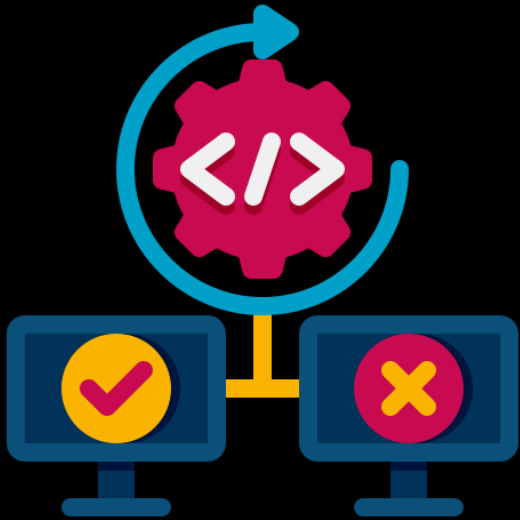


Se basa en que cuantos más elementos de trabajo hay en curso, más se cambia el contexto en una ventana de tiempo.

Permite identificar cuellos de botella en la gestión de un proyecto, sirviendo de **radiador de información** para la introducción de **modificaciones y mejoras**.



Unit Testing

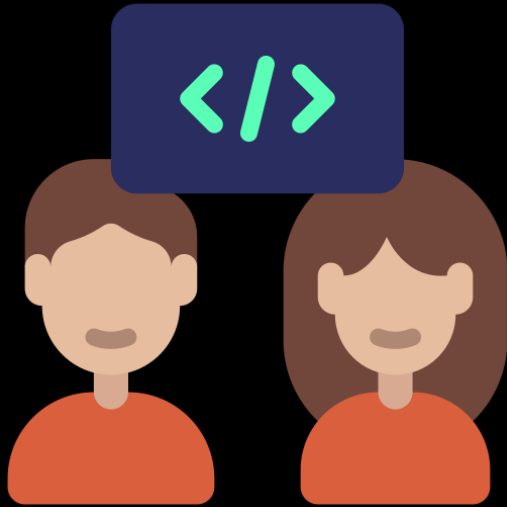


Técnica mediante la cual el equipo de desarrollo puede probar y verificar la correctitud de módulos aislados.

Permite la **reducción de defectos futuros** al modificar funcionalidad ya existente y moldea la forma de programar del equipo de desarrollo, generando código con **menos interdependencias**.



Pair & Mob Programming



Actividad en la que dos (pair) o más (mob) desarrolladores **colaboran** en tiempo real en la programación de un código determinado.

Permite una **colaboración** táctico-estratégica en la escritura del código, generando mejor código y sirviendo como plataforma para el **aprendizaje** e **intercambio** entre pares.



Backlog Refinement



Consiste en dividir el trabajo futuro pendiente en partes más pequeñas, permitiendo definir ítems de trabajo más **precisos** y más **pequeños**.

Da la ventaja de poder realizar una mejor clasificación y priorización del trabajo pendiente, más su posterior planificación y realización.





Abrimos la ronda de preguntas



slido



Audience Q&A Session

① Start presenting to display the audience questions on this slide.

Prácticas – Parte 2

- Integración y Entrega continua (CI/CD)
- Demos y Retrospectives
- Daily standups
- Linting & código opinado
- Refactoring
- Spikes



Demos & Retrospectives



Permiten **inspeccionar** y **generar feedback** del trabajo entregado y el funcionamiento del equipo, en ese orden.

Brindan la oportunidad de generar **ajustes y adaptarse al cambio** en la realización del trabajo futuro y la forma de trabajar del equipo, respectivamente



CI/CD

La CI permite la **automatización** de compilaciones y pruebas. La CD permite la puesta en producción de manera rápida, segura y **sostenible** a lo largo del tiempo.

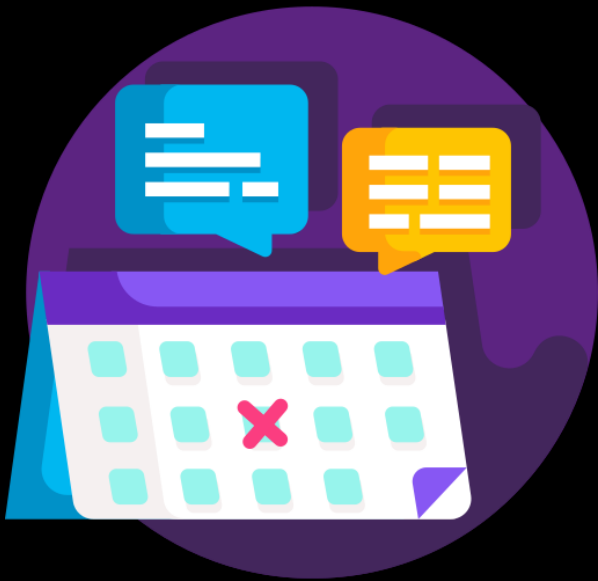


La ventaja de usar CI es **detección temprana** de problemas, más el añadido de calidad al proyecto.

La CD permite que el despliegue de software pase a ser una **tarea de rutina** más y no un **impedimento** o una **tarea bloqueante** en el desarrollo.



Daily Standup



Provee un mecanismo de **inspección** diaria del trabajo realizado, los **impedimentos** y el trabajo a realizar a continuación

Permiten detectar de manera rápida impedimentos, mejorar la **comunicación** y proveer un marco para una rápida **toma de decisiones**.



Refactoring



Consiste en la **mejora** de la estructura interna de una porción de código existente, preservando su comportamiento externo.

Permite mejorar atributos objetivos del código y plantear la escritura del mismo de manera estratégica, facilitando la **comprensión** del propósito del código y las **futuras modificaciones y adaptaciones**.



Linting & Opinionated Code



Permite la aplicación reglas de estilo y forma de escritura del código **consensuadas** por el equipo, más la posibilidad de aplicarlas de manera **automática**.

Acarrea el beneficio de la **unificación de criterios** respecto de cómo escribir y presentar el código de un proyecto.



Spikes



Es un tipo de ítem de trabajo, y una práctica (spiking) que permite realizar una exploración sobre un trabajo complejo sobre cuya **complejidad** el equipo no tiene un **consenso**.

Permite abordar la complejidad del trabajo a realizar y **reducir las incógnitas** sobre el trabajo a planificar y realizar.



Otras Prácticas

- Estimaciones
- Sprints
- Plannings
- Product Discovery
- User Stories
- Value Mapping





Sesión final de Preguntas



slido



Audience Q&A Session

① Start presenting to display the audience questions on this slide.

CMYK 5

¡Gracias!



Service
Design
Club.