



**BENEMERITA UNIVERSIDAD AUTONOMA DE PUEBLA**

**FACULTAD DE CIENCIAS DE LA COMPUTACIÓN**

**INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN**

**Profesor: Luis Yael Méndez Sánchez**

**Materia: Modelos de desarrollo web**

**Período: Otoño 2024**

**Alumnos:**

**Ixchel Naomi Cortes Montaña 202032650**

## **Metodologías ágiles para el desarrollo de aplicaciones web**

### **Introducción:**

Si nos ponemos a hablar entorno a el ámbito del desarrollo web, podemos darnos cuenta que, para las nuevas y las anteriores generaciones la necesidad de entregar aplicaciones o proyectos de alta calidad en plazos de tiempo demasiado cortos a puesto en pie la necesidad de buscar y adoptar el uso de metodologías ágiles, ya que estas metodologías ofrecen un enfoque bastante flexible y colaborativo, permitiendo a los equipos de desarrollo adaptarse a los cambios rápidamente y entregar revisiones del proyecto de manera continua. Algunas de las metodologías más utilizadas o destacadas hoy en día son Scrum, Kanban y XP (Extreme Programming). Cada una de ellas tiene características particulares que las hacen adecuadas para distintos tipos de proyectos y equipos, desde grandes plataformas web hasta proyectos de largo, mediano y corto plazo.

Por lo tanto, se decidió realizar un trabajo que nos ayude a investigar un poco más, además de comprender, comparar y analizar más a detalle estas tres metodologías, destacando en un cuadro comparativo sus ventajas y desventajas dentro del desarrollo web, además de poder analizar un poco más a fondo sobre sus múltiples aplicaciones.

A través de esta comparación, se busca proporcionar una pequeña guía sobre cuál de ellas podría ser más adecuada para cada uno de los diferentes escenarios de desarrollo, tomando en cuenta tanto las necesidades del equipo como las características del proyecto.

Además de que con esta investigación nos podemos ayudar a definir métricas claras para medir la productividad y la calidad dentro de un proyecto y discutir un poco sobre los desafíos más comunes al implementar metodologías ágiles y cómo las empresas los logran superar.

Y finalmente exploraremos cómo las metodologías ágiles se integran de manera exitosa con las prácticas de DevOps en el desarrollo de aplicaciones web para mejorar el ciclo de desarrollo y despliegue continuo.

## Comparación de metodologías ágiles: Scrum, Kanban y XP en el desarrollo web

Las metodologías ágiles son un conjunto de prácticas y principios para el desarrollo de software que se basan en la iteración la colaboración y la flexibilidad son un enfoque dinámico que se adapta a los cambios y las necesidades del proyecto en tiempo real. (Sanchez, L. Y. M, 2024).

La metodología Scrum es un marco de trabajo ágil a través del cual las personas pueden abordar problemas complejos adaptativos a la vez que se entregan productos de forma eficiente y creativa con el máximo valor. Así, Scrum es una metodología que ayuda a los equipos a colaborar y realizar un trabajo de alto impacto. La metodología Scrum proporciona un plan de valores, roles y pautas para ayudar a tu equipo a concentrarse en la iteración y la mejora continua en proyectos complejos. (Martins, 2024)

La metodología XP o la programación extrema es una metodología ágil de gestión de proyectos que se centra en la velocidad y la simplicidad con ciclos de desarrollo cortos y con menos documentación. Al igual que otras metodologías ágiles, la programación extrema es una metodología de desarrollo de software dividido en sprints de trabajo. Los marcos ágiles siguen un proceso iterativo, en el que se completa y revisa el marco al final de cada sprint, refinándolo para adaptarlo a los requisitos cambiantes y alcanzar la eficiencia máxima. (Raeburn, 2024)

### Comparación de roles, eventos y artefactos en Scrum y XP

Metodología	Scrum	XP
<b>Roles</b>	<ul style="list-style-type: none"> <li>○ <b>Product Owner o responsable del producto:</b> Es la persona a cargo de la lista de trabajo pendiente del producto y tiene interacción directa con el cliente, se encargará de hacer saber al equipo de trabajo lo que el cliente desea o necesita. (Martins, 2024)</li> <li>○ <b>Scrum Máster:</b> Es la persona que dirige los distintos eventos de Scrum, debe promover las reuniones diarias de actualización y organizar las reuniones de planificación, revisión y análisis retrospectivo del sprint (Martins, 2024).</li> </ul>	<ul style="list-style-type: none"> <li>○ <b>Tracker:</b> Proporciona retroalimentación al equipo. Debe verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones.</li> <li>○ <b>Customer:</b> Es quien escribe las historias de usuario y las pruebas funcionales para validar su implementación.</li> <li>○ <b>Programmer:</b> Es el programador considerado el más importante miembro del equipo ya que escribe las pruebas unitarias y el código del sistema.</li> </ul>

	<ul style="list-style-type: none"> <li>○ <b>El equipo de Desarrollo:</b> Es un grupo de profesionales que trabajan juntos para entregar el producto. (Sanchez, L. Y. M, 2024).</li> </ul>	<ul style="list-style-type: none"> <li>○ <b>Coach:</b> Es responsable del proceso global y se encarga de guiar a los miembros del equipo para seguir el proceso correctamente.</li> <li>○ <b>Tester:</b> Ayuda al cliente a escribir las pruebas funcionales. Ejecuta pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.</li> <li>○ <b>Big Boss:</b> Es el vínculo entre clientes y programadores. Su labor esencial es la coordinación.</li> <li>○ <b>Consultor:</b> Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto ayuda al equipo a resolver un problema específico y puede que no siempre haya un consultor como parte del equipo.</li> <li>○ <b>Manager:</b> Se encarga de agendar las reuniones, se asegura de que el proceso de juntas sea seguido, registra los resultados de las reuniones para futuros reportes para el Tracker. (Roles, 2015)</li> </ul>
<b>Eventos</b>	<ul style="list-style-type: none"> <li>○ <b>Sprint Planning:</b> Sesión de planificación del sprint.</li> <li>○ <b>Sprints:</b> Semana o semanas donde el equipo trabajará en las tareas pendientes que estableció el Scrum Máster durante la sesión de planificación del sprint.</li> </ul>	<ul style="list-style-type: none"> <li>○ <b>Fase de planeación:</b> Esta fase inicia con las historias de usuario que describen las características y funcionalidades del software.</li> <li>○ <b>Fase de diseño:</b> El proceso de diseño debe procurar diseños simples y sencillos para facilitar el desarrollo</li> </ul>

	<ul style="list-style-type: none"> <li>○ <b>Daily Stand Up:</b> Reunión diaria para el equipo de desarrollo durante 15 minutos, donde se analiza las posibles fallas, atrasos o actualizaciones.</li> <li>○ <b>Sprint Review:</b> Una vez que se haya terminado el sprint de Scrum, el equipo debe reunirse para hacer una revisión de todo lo realizado durante el sprint.</li> <li>○ <b>Sprint Retrospective:</b> Se conversa y reflexiona el análisis retrospectivo del sprint. (Martins, 2024).</li> </ul>	<ul style="list-style-type: none"> <li>○ <b>Fase de codificación:</b> En esta fase los desarrolladores deben diseñar las pruebas de unidad que ejercitarán cada historia de usuario.</li> <li>○ <b>Fase de pruebas:</b> Las pruebas de unidad deben implementarse con un marco de trabajo que permita automatizarlas, con la finalidad de realizar pruebas de integración y validación diarias, esto proporcionará al equipo un indicador del progreso y revelarán a tiempo si existe alguna falla en el sistema. (Pérez, 2011)</li> </ul>
<b>Artefactos</b>	<ul style="list-style-type: none"> <li>○ <b>Product Backlog:</b> Es el artefacto Scrum que recoge la lista del trabajo que debe realizarse (Martins, 2024).</li> <li>○ <b>Sprint Backlog o pila del sprint:</b> Es el trabajo pendiente del sprint, es decir, la serie de trabajos o productos con los que tu equipo se ha comprometido durante el sprint (Martins, 2024).</li> <li>○ <b>Incremento del producto:</b> Se entrega al final de cada sprint, este artefacto consiste en un nuevo producto o función, una mejora o corrección de errores (Martins, 2024).</li> <li>○ <b>Burndown chart:</b> Es una herramienta visual que representa la cantidad de trabajo restante en un Sprint muestra El progreso del equipo y ayuda a identificar posibles obstáculos. (Sanchez, L. Y. M, 2024).</li> </ul>	<ul style="list-style-type: none"> <li>○ <b>Tarjetas de historias de usuario (Story Card):</b> Son descripciones breves y simples de una funcionalidad que se espera implementar, expresadas desde el punto de vista del usuario final.</li> <li>○ <b>Tarjetas de tareas para la descarga de documentos:</b> Son subdivisiones de las historias de usuario.</li> <li>○ <b>El código:</b> Es la implementación técnica que transforma los requerimientos (definidos en las historias de usuario y tareas) en funcionalidad real.</li> <li>○ <b>Pruebas unitarias:</b> Verifican el funcionamiento de componentes o funciones individuales del código</li> <li>○ <b>Pruebas de integración:</b> Verifican la interacción entre múltiples componentes del sistema.</li> <li>○ <b>Pruebas de aceptación:</b> Son realizadas por el cliente o un equipo de calidad para verificar que el sistema cumple con los requerimientos funcionales especificados</li> </ul>

La metodología Kanban se implementa por medio de *tableros* Kanban. Se trata de un método visual de gestión de proyectos que permite a los equipos visualizar sus flujos de trabajo y la carga de trabajo. En un tablero Kanban, el trabajo se muestra en un proyecto en forma de tablero organizado por columnas. Tradicionalmente, cada columna representa una etapa del trabajo. El tablero Kanban más básico puede presentar columnas como Trabajo pendiente, En progreso y Terminado. Las tareas individuales —representadas por tarjetas visuales en el tablero— avanzan a través de las diferentes columnas hasta que estén finalizadas. (Martins, J, 2024)

## **Principios en Kanban.**

Los principios más importantes del sistema Kanban se clasifican en:

### **1. Continuar con el flujo de trabajo existente**

Kanban es un método flexible, lo que significa que puede adaptarse a flujos de trabajo ya existentes o que estén en curso. Gracias a su capacidad de representar visualmente los procesos, ayuda a identificar de una forma más clara los problemas o retrasos de un proyecto.

### **2. Permitir cambios incrementales y evolutivos**

Este principio se basa en que Kanban está diseñado para implementarse gradualmente, es decir, se enfoca en pequeños y continuos cambios de los procesos en curso. Al aplicar modificaciones incrementales, los nuevos ajustes serán más fáciles de entender y adoptar para los miembros del equipo. Lo ideal en este caso es introducir lentamente los cambios y esperar a que todos los integrantes se familiaricen con ellos.

### **3. Respetar el proceso actual**

Consiste en el reconocimiento de los procesos en curso, los roles y las responsabilidades de cada individuo. Mediante el respeto y preservación del proceso actual, se pueden evitar reestructuraciones completas que generen resistencia por parte de los trabajadores. Esto significa que no prohíbe el cambio, pero sí lo prescribe.

### **4. Fomentar todos los niveles de liderazgo**

El sistema o método Kanban anima el liderazgo en todos sus niveles, lo que significa que valora los actos de sus miembros en todas las fases de un proyecto. Para obtener los resultados deseados, es necesario que cada persona asuma la propiedad y responsabilidad de su trabajo para esforzarse en lograr resultados óptimos. En este principio es esencial para que se fomente una mentalidad de mejora continua por parte de todos, ya que ayuda a conseguir un rendimiento óptimo en el ámbito de equipo, departamento y empresa. (Equipo Editorial, 2024)

## Ventajas y desventajas de las metodologías ágiles

Ventajas	Desventajas
<b>Entregas rápidas:</b> Las metodologías ágiles permiten entregar valor al cliente más rápido lo que facilita la adaptación a los cambios y reduce los riesgos	<b>Resistencia al cambio:</b> Las organizaciones con estructuras y procesos bien establecidos pueden resistir a la adopción a metodologías ágiles la resistencia de cambio requiere estrategias de comunicación capacitación y gestión del cambio
<b>Colaboración y comunicación:</b> El trabajo en equipo y la comunicación constante son elementos esenciales en las metodologías ágiles lo que fomenta la sinergia y la eficiencia	<b>Falta de experiencia:</b> La implementación exitosa de metodologías ágiles requiere experiencia y habilidades específicas como la gestión de proyectos ágiles el trabajo en equipo y la comunicación efectiva.
<b>Flexibilidad y adaptación:</b> Las metodologías ágiles se adaptan a los cambios y las necesidades cambiantes permitiendo la flexibilidad y la evolución constante	<b>Falta de cultura ágil:</b> La cultura organizacional juega un papel importante en el éxito de la implementación ágil se requiere una cultura de colaboración transparencia y aprendizaje continuo
<b>Satisfacción del cliente:</b> El enfoque en el cliente y la entrega incremental de valor garantizan una mayor satisfacción y mejores resultados (Sanchez, L. Y. M, 2024).	<b>Complejidad del proyecto:</b> La complejidad del proyecto como la integración de sistemas o la participación de múltiples equipos puede dificultar la implementación ágil se necesitan ajustes al marco ágil para adaptarse a la complejidad

## Ejemplos de empresas que utilizan metodologías ágiles para desarrollo web.

### Scrum

- **Spotify:** Es un caso emblemático de uso de **Scrum**, aunque lo ha adaptado a sus necesidades. Utilizan lo que llaman "Squads" (equipos Scrum pequeños y autónomos) para desarrollar nuevas características en su plataforma de streaming musical. (Ingenio Learning, 2024)
- **BBVA:** Implementó Scrum para agilizar el desarrollo de productos y servicios bancarios, como su plataforma de banca en línea, permitiendo ciclos de entrega más rápidos y adaptaciones a los cambios del mercado financiero.

## Kanban

- **Toyota:** Kanban es originalmente un sistema utilizado por Toyota en la fabricación de automóviles, pero ha sido adaptado para el desarrollo de software. Toyota sigue aplicando Kanban en sus procesos de producción para optimizar el flujo de trabajo, identificando cuellos de botella y mejorando continuamente sus procesos.
- **Zara:** En el sector de la moda, Zara utiliza Kanban para gestionar su cadena de suministro y acelerar el desarrollo y distribución de sus productos. Aunque no es estrictamente desarrollo de software, su implementación de Kanban en procesos logísticos es un ejemplo clave de cómo puede aplicarse para coordinar equipos y producción. (Indeed, 2023)

## XP (Extreme Programming)

- **BMW:** En el desarrollo de su software para vehículos inteligentes, BMW ha implementado XP para fomentar una cultura de desarrollo continuo y pruebas rápidas. El enfoque de XP en la mejora continua y la atención a los detalles técnicos encaja bien con los altos estándares de la industria automotriz.
- **Ford:** Ford ha adoptado XP en ciertos equipos de desarrollo de software para la creación de soluciones tecnológicas automotrices. Al usar XP, Ford ha sido capaz de responder a cambios rápidos en los requerimientos mientras mantiene un alto estándar de calidad de software. (Psico-smart, 2024)



## Conclusión

Una vez que terminamos la investigación y pudimos realizar la comparación entre las metodologías ágiles más comunes como lo son **Scrum**, **Kanban** y **XP** en el contexto del desarrollo de aplicaciones web, nos pudimos dar cuenta de que cada metodología ofrece ventajas específicas que las hacen más adecuadas para distintos tipos de proyectos y organizaciones, Por ejemplo

Nos pudimos dar cuenta de que Scrum es ideal para proyectos complejos y de gran escala, como plataformas web grandes o proyectos con múltiples equipos involucrados, debido a que su estructura que esta basada en roles, eventos y artefactos proporciona una base sólida para la organización y el seguimiento del progreso dentro del proyecto, especialmente si hablamos que se requiere una comunicación constante con el cliente y entregas funcionales en periodos cortos de tiempo, además de que Scrum es especialmente útil en plataformas que evolucionan rápidamente o requieren adaptaciones continuas, como el caso de Spotify o BBVA.

Después si nos enfocamos un poco más a hablar sobre la metodología Kanban nos podemos dar cuenta de que es más adecuada para equipos de trabajo que buscan flexibilidad en la gestión de su flujo de trabajo. Al ser una metodología visual y sin roles estrictamente definidos, Kanban se adapta bien a equipos pequeños que necesitan una solución ágil y menos estructurada, además que es eficiente cuando se necesita un enfoque continuo y fluido, permitiendo adaptarse rápidamente a las prioridades cambiantes sin requerir ciclos iterativos como lo maneja Scrum, algunos de los ejemplos de empresas que utilizan Kanban son Toyota y Zara que lo utilizan ya que les ayuda a optimizar procesos y mantener un flujo constante de trabajo.

Si hablamos de la metodología Extreme Programming o mejor conocida como XP podemos darnos cuenta de que es altamente recomendable si hablamos de proyectos de corto plazo y ciclos rápidos de desarrollo, como las pruebas de software en vehículos inteligentes de BMW o Ford. La metodología se enfoca en la simplicidad del código, la mejora continua y como Scrum la colaboración cercana con el cliente. XP es ideal para proyectos con equipos pequeños y altamente técnicos, donde la calidad del código es crítica y se necesita entregar resultados frecuentes a través de ciclos cortos, muy parecido a lo que maneja Scrum.

Y finalmente nos podemos dar cuenta de que el uso de metodologías ágiles facilita la convivencia y ayuda al desarrollo de proyectos de manera rápida y sencilla siempre y cuando se tenga un grupo de expertos que conozcan bien del tema, ya que como pudimos ver tiene algunas desventajas ya que se requiere experiencia y habilidades específicas como la gestión de proyectos ágiles el trabajo en equipo y la comunicación efectiva, para completar las tareas.

## Impacto de las metodologías ágiles en la productividad y calidad de aplicaciones web

### ¿Qué es la productividad en el desarrollo web?

Según recoge el diccionario de la RAE, la productividad en el desarrollo de software se define como la "relación entre lo producido y los medios empleados para producirlo". En el mundo del desarrollo de software esto puede traducirse en problemas de codificación, bajo interés o motivación en los proyectos, mala distribución de la carga de trabajo o dinámicas de equipo poco colaborativas, tal y como se define en los principios de la Developer Experience.

Medir la productividad de un equipo de desarrollo de software implica evaluar diversos factores cuantitativos y cualitativos para obtener una visión global de la eficiencia del equipo.

### Métricas para calcular la productividad de tu equipo

- **Velocidad:** Ayuda a comprender la capacidad del equipo y a predecir su rendimiento futuro. La velocidad es una métrica utilizada en las metodologías Agile para medir la cantidad de trabajo que un equipo completa durante un sprint. Se calcula en función del número de historias de usuario, tareas o puntos completados comprobando el comportamiento histórico y no centrándose únicamente en un sprint.
- **Lead time y cycle time:** El lead time mide el tiempo que transcurre desde el momento en que se crea una nueva tarea hasta que se completa, mientras que el cycle time comienza cuando se inicia realmente el trabajo en la tarea. Estas métricas ayudan a identificar cuellos de botella en el proceso de desarrollo y oportunidades para agilizar los flujos de trabajo.
- **Bug rate:** La tasa de errores o defectos notificados tras un lanzamiento. Una tasa elevada de errores puede indicar problemas de calidad de código o lanzamientos precipitados. También es útil para detectar oportunidades de mejora de habilidades y aplicación de técnicas como TDD.
- **Code churn:** Se refiere al porcentaje de código de un desarrollador/a que presenta ediciones recientes. Un code churn elevado puede indicar indecisión, falta de claridad o la necesidad de una revisión exhaustiva del código.
- **Encuestas de satisfacción:** Realizar encuestas periódicas entre los miembros del equipo proporciona información sobre cómo se percibe el proceso de desarrollo y las áreas de mejora. Los niveles de satisfacción afectan indirectamente a la productividad, ya que los equipos con mayor satisfacción tienden a ser más comprometidos y eficientes. (Codurance, 2024)

Tener en cuenta la Developer Experience (DevEx) puede contribuir a mejorar la productividad de los equipos de desarrollo de software. La DevEx contempla todas las interacciones del desarrollador/a con sus herramientas, procesos y entorno, con el objetivo de optimizar los procesos de desarrollo de software y hacerlos más eficientes y satisfactorios.

Como pudimos ver anteriormente las metodologías ágiles han ayudado a muchas empresas de alto nivel a mejorar su productividad y a tener un excelente desempeño a la hora de desarrollar sus proyectos, como por ejemplo el caso de BBVA es un ejemplo emblemático de cómo las metodologías ágiles han sido implementadas en el sector bancario para mejorar la eficiencia y la entrega de productos digitales. En 2014, BBVA comenzó una transformación ágil a gran escala con el objetivo de mejorar la experiencia de sus clientes y acelerar el desarrollo de productos. Para ello, adoptaron Scrum y Kanban en varias áreas de desarrollo y gestión de proyectos. (Tena. M, 2024)

Zara y Kanban Zara utiliza Kanban en su cadena de suministro, lo que le permite realizar ajustes incrementales y lanzar nuevos productos en cuestión de semanas, en lugar de meses, como es típico en la industria de la moda. Si bien esto no es estrictamente desarrollo de software, el mismo sistema de Kanban aplicado a sus procesos logísticos ha demostrado ser efectivo para reducir los tiempos de entrega y mantener la alta calidad del producto. Esta metodología también podría aplicarse a equipos de desarrollo web que buscan optimizar procesos y minimizar bloqueos. (Indeed, 2023)

Ford y XP En el desarrollo de software para sus vehículos autónomos, Ford implementó XP (Extreme Programming), lo que le permitió reducir el tiempo de entrega en un 30%. La capacidad de realizar iteraciones rápidas, junto con el enfoque en la calidad técnica y la simplicidad del código, permitió a Ford mejorar su tasa de detección temprana de errores y lanzar productos más fiables. Esto resultó en una reducción significativa de defectos en el software y, por lo tanto, en la mejora de la satisfacción del cliente. (Psico-smart, 2024)

## **Desafíos comunes al implementar metodologías ágiles y cómo las empresas los superan**

**Desafío 1:** Uno de los principales desafíos al adoptar metodologías ágiles es la resistencia al cambio por parte de los empleados y la cultura organizacional. En empresas con procesos rígidos y jerárquicos, los equipos suelen tener dificultades para adaptarse a la estructura colaborativa y dinámica de las metodologías ágiles. Esto puede generar conflictos entre los roles tradicionales y los nuevos requerimientos de agilidad.

**Solución:** Para superar este obstáculo, muchas empresas implementan estrategias de capacitación continua y gestión del cambio, proporcionando a sus empleados las habilidades necesarias para trabajar en un entorno ágil. BBVA, por ejemplo, facilitó talleres intensivos y realizó una reestructuración gradual de sus equipos, asegurándose de que los roles de Scrum Master y Product Owner estuvieran bien definidos, lo que ayudó a reducir la resistencia.

**Desafío 2:** Otro desafío es la falta de experiencia de los equipos con las metodologías ágiles. En proyectos grandes y complejos, los equipos pueden tener dificultades para gestionar las iteraciones cortas y la entrega continua de valor sin comprometer la calidad.

**Solución:** Las empresas como BMW han superado este desafío invirtiendo en la creación de centros de excelencia ágil, donde los equipos de desarrollo reciben apoyo continuo de expertos en Agile y XP. Además, BMW estableció coaches ágiles que acompañan a los equipos en su adopción y en la implementación de buenas prácticas de programación.

**Desafío 3:** En entornos que involucran múltiples equipos, como grandes plataformas de desarrollo web, mantener una sincronización adecuada entre diferentes áreas puede ser un desafío. Esto incluye la integración de los sprints de Scrum con las prácticas de otros equipos que usan Kanban o XP.

**Solución:** Para superar este desafío, empresas como Spotify adoptaron una estrategia de "tribus" y "capítulos", donde se fomenta la colaboración entre equipos que trabajan en diferentes productos, pero comparten el mismo enfoque ágil. De esta manera, se asegura que los equipos no solo sigan trabajando de manera autónoma, sino también que se comuniquen y mantengan una visión compartida del producto final.

## Integración de metodologías ágiles con herramientas DevOps en el desarrollo web

### ¿Qué es DevOps?

DevOps es un conjunto de prácticas, herramientas y filosofía cultural que sirve para automatizar e integrar los procesos que comparten el equipo de desarrollo de software y el de TI. Se centra en el empoderamiento de los equipos, la comunicación y colaboración entre ellos y la automatización de la tecnología.

Un equipo de DevOps está formado por desarrolladores y profesionales de operaciones de TI que colaboran durante todo el ciclo de vida de un producto para aumentar la velocidad y la calidad de la implementación del software. Se trata de una nueva forma de trabajar, un cambio cultural que tiene una profunda repercusión tanto para los equipos como para las organizaciones en las que trabajan.

Los equipos de DevOps utilizan herramientas para automatizar y acelerar los procesos, lo que hace que sean mucho más fiables. Con una cadena de herramientas de DevOps, los equipos pueden ocuparse de aspectos básicos de esta metodología, como la integración continua, la entrega continua, la automatización y la colaboración.

### Principios de DevOps

- **Colaboración:** La premisa clave de DevOps es la colaboración. Los equipos de desarrollo y operaciones se fusionan en un equipo funcional que se comunica, comparte feedback y colabora durante todo el ciclo de desarrollo e implementación. A menudo, esto conlleva que los equipos de desarrollo y operaciones se fusionen en un solo equipo que se encarga de todo el ciclo de vida de la aplicación.
- **Automatización:** Una práctica fundamental de DevOps es automatizar todo el ciclo de vida de desarrollo de software que sea posible. De este modo, los desarrolladores tienen más tiempo para programar y desarrollar nuevas funciones. La automatización es un elemento clave para conseguir una canalización de CI/CD que ayude a reducir los errores humanos y aumentar la productividad del equipo. Con la automatización de los procesos, los equipos consiguen una mejora continua con tiempos de iteración breves que les permiten responder rápidamente al feedback de los clientes.
- **Mejora continua:** La mejora continua se estableció como un elemento básico de las prácticas de metodología ágil, así como de la producción lean y el kata de mejora. Es una práctica que consiste en centrarse en la experimentación, minimizar lo superfluo y optimizar la velocidad, los costes y la facilidad de entrega. La mejora continua también está vinculada a la entrega continua, lo que permite a los equipos de DevOps enviar actualizaciones de forma continuada para mejorar la eficiencia de los sistemas de software. (Atlassian, 2024)

- **Medición:** DevOps abraza el enfoque basado en datos para medir el rendimiento y la eficacia de los procesos. La medición constante proporciona información valiosa sobre la velocidad de entrega, la estabilidad y la calidad del software. Métricas como el tiempo de ciclo, la frecuencia de despliegue, la tasa de errores y la disponibilidad del servicio son cruciales para evaluar el éxito de las prácticas DevOps. Estas mediciones permiten a los equipos identificar áreas de mejora, tomar decisiones informadas y adaptar estrategias para lograr mejores resultados. (Linkedin, 2023)

### ¿Cómo funcionan las prácticas ágiles y DevOps en conjunto?

El enfoque colaborativo de la metodología ágil se extiende aún más cuando se combina con el uso de los equipos de operaciones por parte de DevOps. Cuando los equipos trabajan juntos, los equipos ágiles desarrollan software que DevOps luego entrega al usuario final. Los desarrolladores ágiles proporcionan incrementos de software que DevOps integra e implementa mediante la entrega y la automatización continuas. Los desarrolladores asumen la responsabilidad de escribir las pruebas de software y mantener la canalización de creación e implementación.

### Herramientas populares que integran Agile y DevOps

- **Asana** es, sin duda, uno de los mejores softwares para la gestión ágil de proyectos hoy en día. Permite planificar, organizar y gestionar fácilmente el trabajo de tu equipo y hacer un seguimiento de su progreso. Es una herramienta visual y flexible que facilita enormemente la colaboración y comunicación dentro del equipo, así como el cumplimiento de los plazos de entrega.
- **Monday.com** es una de las herramientas de gestión de proyectos *Agile* más populares en la actualidad. Permite planificar, ejecutar y realizar el seguimiento de proyectos de diferente tamaño y favorece la colaboración entre equipos al proporcionar una visión completa y compartida del trabajo.
- **Jira** es la herramienta de gestión ágil para proyectos de desarrollo de software por excelencia. Permite a los equipos de desarrollo planificar, seguir y entregar productos valiosos a los usuarios. Posibilita organizar las distintas fases del proyecto, asignar tareas a los distintos miembros y realizar el seguimiento del avance del equipo. (Sentrio, 2022)
- **Docker** es una plataforma para el desarrollo de software, automatización de pruebas y entrega de software utilizando imágenes en contenedores. Un contenedor es una versión aislada del software que se ejecuta igual independientemente del sistema operativo en el que se esté ejecutando o del lenguaje de programación que se haya utilizado, evitando la necesidad de hacer girar máquinas virtuales.
- La plataforma Docker incluye interfaces de usuario, de línea de comandos y de programación de aplicaciones (UI, CLI y API), así como compatibilidad y seguridad para cada paso del ciclo de vida de desarrollo de software.

- **Jenkins** es un servidor de automatización de código abierto y gratuito que ayuda a automatizar los procesos de desarrollo de software, como la construcción, facilitando el CI/CD, el lanzamiento y los testeos. Esta herramienta DevOps facilita a los equipos la supervisión de las tareas repetidas, la integración de los cambios con facilidad y la identificación de los problemas con rapidez. (Pathak. A, 2021)

## **Conclusión:**

En el transcurso de la investigación nos pudimos dar cuenta que el uso de metodologías ágiles y su integración con herramientas DevOps en el desarrollo web nos muestra cómo estas prácticas han transformado la forma en que las empresas desarrollan, implementan y mantienen aplicaciones web. Además, hemos visto que metodologías ágiles como por ejemplo Scrum, Kanban y XP han mejorado a gran escala la colaboración, la adaptabilidad y la capacidad de respuesta a los cambios, en proyectos de diferentes escalas tanto grandes como pequeños. Estos enfoques no solo aceleran el desarrollo de software, sino que también aumentan la calidad del producto entregado, mejorando la satisfacción del cliente.

También vimos que empresas como BBVA y otros ejemplos dentro de la industria web muestran que la implementación de estas metodologías, combinadas con herramientas DevOps, tiene un impacto directo en la productividad. DevOps, al automatizar el ciclo de vida del desarrollo, la integración continua (CI) y la entrega continua (CD), ayuda a reducir errores, acortar los tiempos de implementación y mejora la estabilidad del sistema, lo que resulta en una ventaja competitiva para dichas empresas.

Sin embargo, estos beneficios no están exentos de desafíos. La resistencia al cambio, la necesidad de capacitación y la integración de equipos multidisciplinarios son barreras comunes que las empresas enfrentan. Para superarlas, la clave ha sido la adopción de una cultura organizacional ágil y colaborativa, la implementación gradual de nuevas herramientas y la medición constante de resultados para mejorar continuamente los procesos.

Finalmente podemos ver que, la combinación de metodologías ágiles como lo son Scrum, XP, Kanban y DevOps en el desarrollo web ha revolucionado la industria en el ámbito web, mejorando tanto la eficiencia operativa como la calidad del software, y permitiendo a las empresas adaptarse rápidamente a las demandas del mercado, además cumpliendo con que los usuarios tengan una mejor y grata experiencia a la hora de hacer uso de estos.

## Bibliografía:

1. Atlassian. (2024). *Los principios de DevOps*. Atlassian. <https://www.atlassian.com/es/devops/what-is-devops>
2. Equipo Editorial. (2024, febrero 8). *Sistema Kanban: qué es, principios y prácticas*. Indeed.com. <https://www.indeed.com/orientacion-profesional/desarrollo-profesional/kanban-que-es#:~:text=Los%20cuatro%20principios%20clave%20del,liderazgo%20en%20todos%20sus%20niveles>.
3. Indeed, E. (2023, marzo 16). *Kanban: ejemplos de empresas que lo han implementado*. Indeed.com. <https://mx.indeed.com/orientacion-profesional/como-encontrar-empleo/kanban-ejemplos-empresas#:~:text=As%C3%AD%20como%20Toyota%2C%20otras%20compa%C3%B1%C3%ADas,cr%C3%ADticos%20con%20un%20tablero%20Kanban>.
4. Ingenio Learning. (2024). *Cómo las empresas usan SCRUM en sus procesos*. Ingenio Learning. Recuperado el 13 de octubre de 2024, de <https://ingenio.edu.pe/blog/grandes-empresas-que-aplican-scrum/>
5. *Los principios claves de DevOps: colaboración, automatización, medición y mejora continua*. (2023, diciembre 28). LinkedIn.com. <https://www.linkedin.com/pulse/los-principios-claves-de-devops-colaboraci%C3%B3n-automatizaci%C3%B3n-medici%C3%B3n-zadqe/>
6. Martins, J. (2024a, enero 19). *¿Qué es la metodología Kanban y cómo funciona?* Asana. <https://asana.com/es/resources/what-is-kanban>
7. Martins, J. (2024, febrero 15). *Scrum: conceptos clave y cómo se aplica en la gestión de proyectos*. Asana. <https://asana.com/es/resources/what-is-scrum>
8. Pathak, A. (2021, mayo 13). *Explora las 30 Mejores Herramientas de DevOps a Tener en Cuenta en 2024*. Kinsta®; Kinsta. <https://kinsta.com/es/blog/herramientas-devops/>
9. Pérez A., O. A. (2011). Cuatro enfoques metodológicos para el desarrollo de Software RUP – MSF – XP - SCRUM. *Revista inventum*, 6(10), 64. <https://doi.org/10.26620/uniminuto.inventum.6.10.2011.64-78>
10. *Productividad en equipos de desarrollo: Una mirada en profundidad*. (s/f). Codurance.com. Recuperado el 14 de octubre de 2024, de <https://www.codurance.com/es/productividad-en-el-desarrollo-de-software>
11. Psico-smart. (2024, agosto 28). *Métodos ágiles de gestión: su impacto en la eficiencia y colaboración del equipo*. Psico-smart.com. <https://psico-smart.com/articulos/articulo-metodos-agiles-de-gestion-su-impacto-en-la-eficiencia-y-colaboracion-del-equipo-169696>
12. Raeburn, A. (2024, febrero 13). *¿Qué es la programación extrema (XP)? [2024]* •. Asana. <https://asana.com/es/resources/extreme-programming-xp>



13. *Roles*. (2015, noviembre 21). Extreme Programming.  
<https://iswugxp.wordpress.com/roles/>
14. Sanchez, L. Y. M. (2024). *Metodologías ágiles para el desarrollo web*.  
[https://correobuap.sharepoint.com/:b:/s/Section\\_202435-ITIS257-16183-001/ERWk3SBaTFJE174F-JrBjsgBAQdxgKGRZBYS2Wtd8nFWwQ?e=1W3MNX](https://correobuap.sharepoint.com/:b:/s/Section_202435-ITIS257-16183-001/ERWk3SBaTFJE174F-JrBjsgBAQdxgKGRZBYS2Wtd8nFWwQ?e=1W3MNX)
15. Sentries. (2022, mayo 17). *5 herramientas para la gestión ágil de proyectos de software en 2022*. Sentries. <https://sentries.io/blog/mejores-herramientas-gestion-agil/>
16. Tena, M. (2024). *¿Qué es la metodología “ágil”?* BBVA. Recuperado el 14 de octubre de 2024, de <https://www.bbva.com/es/innovacion/metodologia-agile-la-revolucion-las-formas-trabajo/>