educación IT
CODE YOUR FUTURE

Programación Web .NET Core

Módulo 4

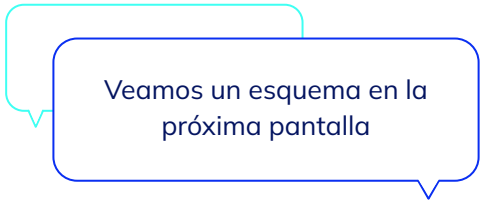
Planificación en la fase de diseño del proyecto

Metodologías de desarrollo de proyectos

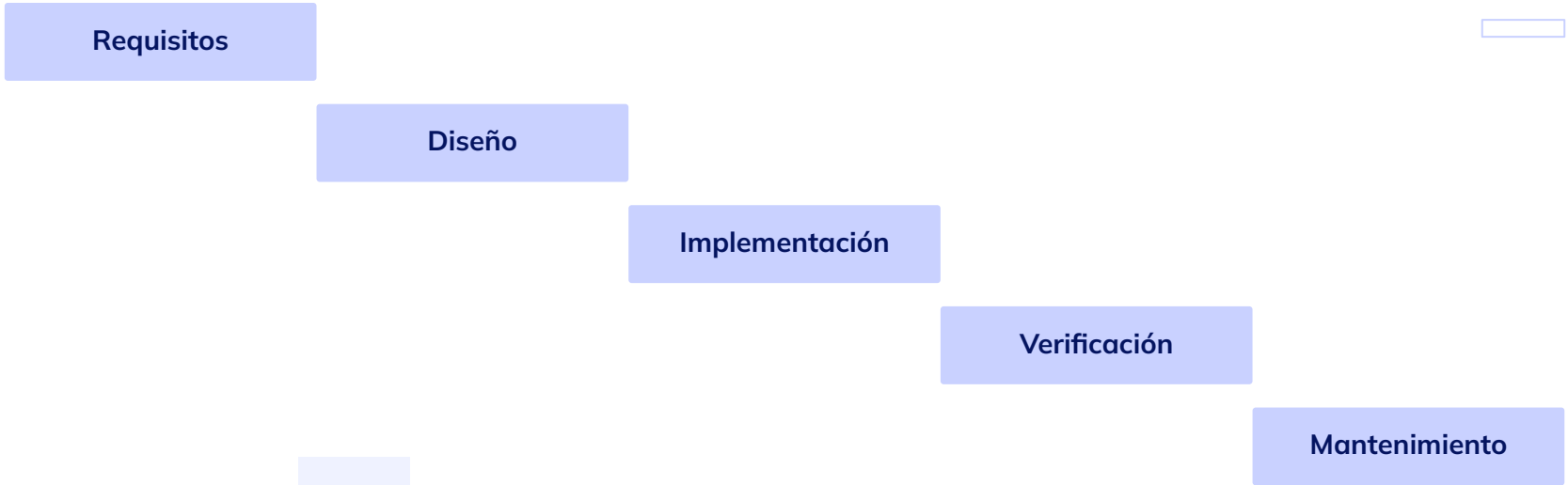
Desarrollar una aplicación web o una aplicación de intranet, muchas veces, es un proceso complejo que involucra varios equipos de desarrolladores que desempeñan diferentes roles.

Existe una amplia gama de metodologías para organizar el proceso y asegurarse de que todos trabajen en conjunto. Las metodologías de desarrollo describen las fases del proyecto, los roles de las personas, los entregables que concluyen cada fase, y otros aspectos.

Muchas organizaciones tienen una metodología estándar. Las metodologías más usadas incluyen el **modelo de desarrollo de software ágil**, **programación extrema** y **pruebas de desarrollo**.



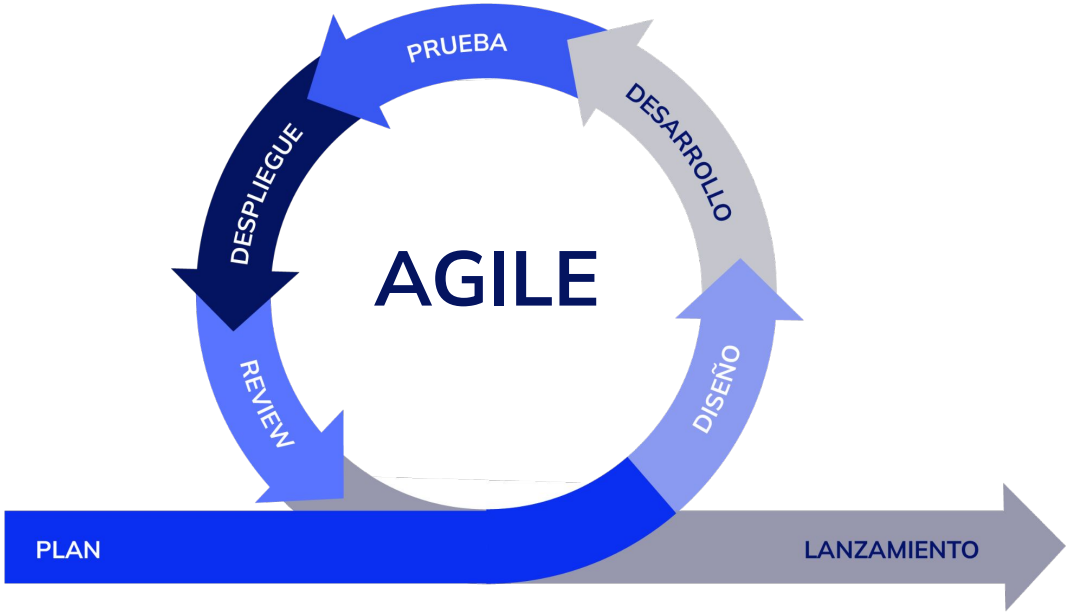
Fases tradicionales en el desarrollo de un proyecto



¿Cuáles son y qué objetivos persigue cada una?

Modelo de desarrollo ágil	Proporcionar en poco tiempo piezas pequeñas del sistema, en funcionamiento. Implica que equipos reducidos de desarrolladores y representantes empresariales se reúnan regularmente en persona durante el ciclo de vida del desarrollo del software.
Programación extrema	Crear una aplicación desde el inicio con la resolución de tareas críticas. Los desarrolladores prueban la solución y obtienen comentarios de las partes interesadas para generar los requisitos detallados, que evolucionan a lo largo del ciclo de vida del proyecto.
Desarrollo basado en pruebas	Crear una aplicación comenzando con un proyecto de prueba. Los cambios en el código se pueden probar individualmente o en grupo, a lo largo del proyecto.
UML **Lenguaje de modelado unificado**	Los diagramas UML se utilizan con fines de planificación y documentación en todos los modelos de desarrollo de proyectos.

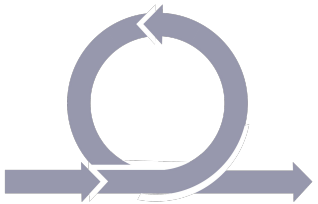
Modelo de desarrollo ágil



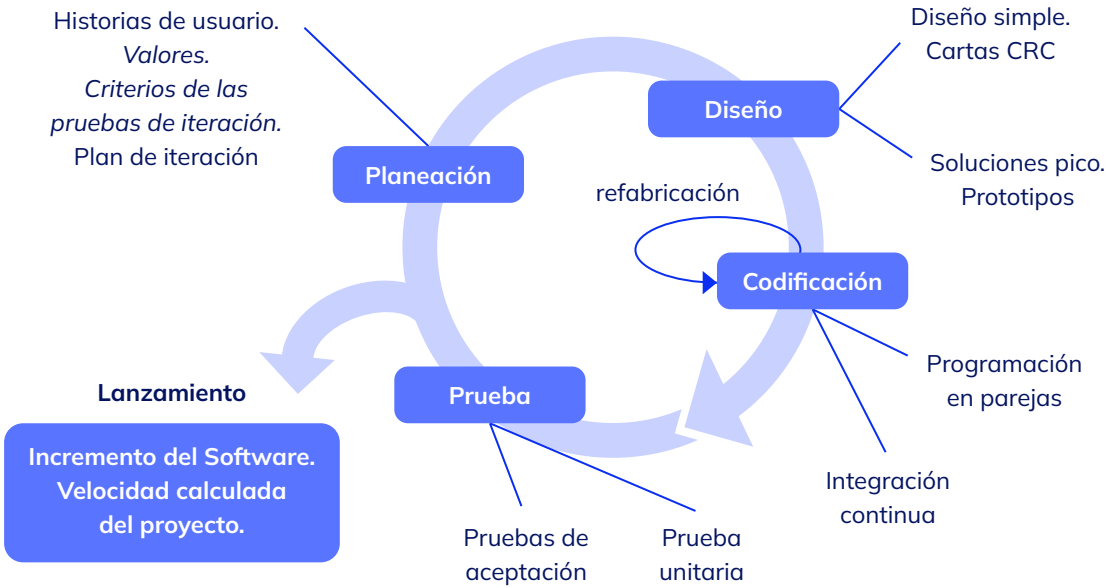
La metodología Agile está diseñada para **integrar circunstancias y requisitos cambiantes** a lo largo del proceso de desarrollo.

Los **proyectos ágiles** se caracterizan por:

- **Desarrollo incremental:** Desarrollo en ciclos rápidos que se basan en ciclos anteriores. Cada iteración se prueba a fondo.
- **Énfasis en las personas y las interacciones:** Los desarrolladores escriben código en función de lo que hacen las personas.
- **Énfasis en software de trabajo:** Los desarrolladores escriben soluciones que las partes interesadas pueden evaluar en cada iteración para validar si resuelve un requisito.
- **Estrecha colaboración con los clientes:** Los desarrolladores debaten con los clientes y las partes interesadas en el día a día para comprobar los requisitos base.



Programación extrema (XP)



La **programación extrema** evolucionó a partir del desarrollo ágil de software, la fase de diseño preliminar se reduce al mínimo y los desarrolladores se centran en resolver algunas tareas críticas. Al finalizar estas tareas críticas, prueban la solución simplificada y obtienen comentarios de partes interesadas. Esta retroalimentación ayuda a los desarrolladores a poder identificar los requisitos detallados, que evolucionan a lo largo de la vida y ciclo del proyecto.

La programación extrema define una *historia de usuario* para cada *rol de usuario*.

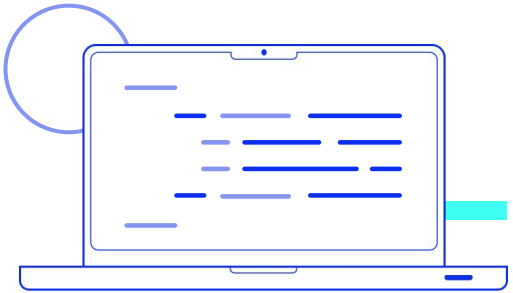
Una historia de usuario describe todas las interacciones que un usuario con un rol específico podría realizar dentro de la aplicación completa. La colección de todas las historias de usuario para todos los roles de usuario, describen la aplicación completa.

En la **programación extrema**, los desarrolladores suelen trabajar en **parejas**. Un desarrollador escribe el código y el otro lo revisa para asegurarse de que utiliza soluciones simples y se adhiere a las mejores prácticas. La prueba del desarrollo es impulsada por TDD Test Driven Development.

TDD: Test Driven Development

TDD significa ***desarrollo basado en pruebas***. Es una técnica que utiliza **pruebas unitarias** para impulsar el desarrollo de la aplicación.

En Microsoft Visual Studio es posible agregar y ejecutar pruebas unitarias, se pueden ejecutar individualmente o en grupos. En proyectos MVC es muy sencillo crear pruebas unitarias.



UML: Lenguaje de modelado unificado

¿Qué es UML? ¿Qué diagramas lo componen?

El **Lenguaje Unificado de Modelado** o **UML** (***Unified Modeling Language***) es un lenguaje estandarizado. Está especialmente desarrollado para ayudar a todos los intervinientes en el desarrollo y modelado de un sistema o un producto de software. **Permite describir, diseñar, especificar, visualizar, construir y documentar todos los artefactos** que lo componen, al servirse de varios tipos de diagramas. Muchos de los diagramas confeccionados en UML son usados en las diferentes metodologías de desarrollo de software.

Tipos de diagramas UML:

- Diagrama de clases.
- Diagrama de componentes.
- Diagrama de despliegue.
- Diagrama de actividades.
- Diagrama de estado.
- Diagrama de secuencia.
- Diagrama de casos de uso.
- Otros.

Recopilación de requisitos

Recopilación de requisitos

Al encargar el desarrollo de un sistema, se investigan los requisitos funcionales y técnicos.

Requerimientos funcionales

Describen cómo la aplicación se comporta y asimismo cómo responde a los usuarios. También se denominan ***requisitos de comportamiento***.

Los requerimientos funcionales incluyen:

- **Requisitos de la interfaz de usuario:** Describen cómo interactúa el usuario con la aplicación.
- **Requisitos de uso:** Describen lo que puede hacer un usuario con la aplicación.
- **Requisitos comerciales:** Describen cómo la aplicación cumplirá las funciones comerciales.
- **Requerimientos técnicos:** Describen las características técnicas de la aplicación y se relacionan con la disponibilidad, seguridad o rendimiento. Estos requisitos, a veces, se denominan *no funcionales* o *no conductuales*.

Los requisitos funcionales se reúnen con **entrevistas a las partes interesadas**: usuarios, administradores, desarrolladores, responsables de presupuestos y gerentes de equipo. Cada uno de estos grupos puede tener diferentes conjuntos de prioridades que la aplicación debe cumplir.

Puede crear requisitos funcionales mediante:

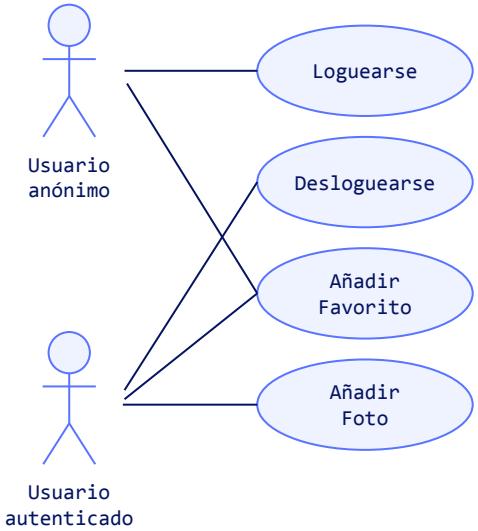
- ***Escenarios de uso***.
- ***Casos de uso***.
- ***Modelado de requisitos*** en la metodología ágil.
- ***Historias de usuario*** en la metodología de programación extrema.

Escenarios y casos de uso

Un **escenario de uso** es un ejemplo específico -del mundo real-, de una interacción entre la aplicación y un usuario, que incluye nombres y valores de entrada sugeridos.

El **caso de uso** es similar, pero más generalizado. No incluye nombres de usuario, ni valores de entrada. Describe múltiples rutas de una interacción, que depende de lo que el usuario proporciona como entrada u otros valores.

El siguiente gráfico de ejemplo, es un diagrama de caso de uso simple de UML.



Ejemplos

Escenario de uso *Agregar Foto:*

1. Juan Pérez hace clic en el enlace ***Agregar foto*** en el menú principal del sitio.
2. Juan proporciona la entrada, ***Juan P***, en el cuadro ***Nombre de usuario*** y la contraseña en el cuadro ***Contraseña*** para poder autenticarse en el sitio.
3. Juan escribe el título, ***Amanecer***, para la foto.
4. Juan busca el archivo JPEG de su nueva foto.
5. Juan hace clic en el botón llamado ***Cargar***.
6. La aplicación web almacena la nueva foto y muestra la galería de fotos a Juan.

Caso de uso *Agregar foto*:

1. El usuario hace clic en el enlace ***Agregar foto*** en el menú principal del sitio.
2. Si el usuario es anónimo, aparece la página de inicio de sesión y el usuario proporciona las credenciales.
3. Si las credenciales son correctas, aparecerá la vista ***Crear foto***.
4. El usuario escribe un título.
5. El usuario especifica el archivo de foto a cargar.
6. El usuario opcionalmente escribe una descripción para la foto.
7. El usuario hace clic en el botón ***Cargar***.
8. La aplicación web almacena la nueva foto y muestra la galería al usuario.

Al analizar escenarios y casos de uso, se identifican **requisitos funcionales** de todo tipo. Por ejemplo, del caso de uso anterior, surgen los requisitos de interfaz de usuario.

La página web que permite a los usuarios crear una nueva foto debe incluir cuadros de texto de ***Título*** y ***Descripción***, un archivo control de entrada para el archivo de la foto y un botón ***Cargar*** para guardar la foto.

Análisis en el modelado de requisitos

En el modelado de requisitos, el análisis se caracteriza de la siguiente manera:

- **Modelado de requisitos iniciales:** En la fase de diseño inicial, los desarrolladores identifican y registran algunos casos de uso de manera informal, sin detalles completos.
- **Modelado justo a tiempo:** Antes de escribir código que implemente un caso de uso, un desarrollador lo discute con los usuarios relevantes. En este punto, el desarrollador agrega **detalles completos** al caso de uso.

En un desarrollo con la implementación de **metodología Ágil, los desarrolladores hablan con los usuarios y otras partes interesadas en todo momento,** y no solo lo hacen en el princi-
pio y en el fin del proyecto.

- **Test de aceptación:** Una prueba de aceptación es una prueba que la aplicación debe pasar para todas las partes interesadas. En este *test*, se debe garantizar que **se cumplan los requisitos del caso de uso**.

Historias de usuarios en programación extrema

En proyectos de **programación extrema**, los desarrolladores hacen análisis de requisitos funcionales al comienzo del desarrollo: crean ***historias de usuarios***, en lugar de casos de uso, o escenarios de usuario. Estas historias son **ejemplos muy amplios de interacción** entre la aplicación y un usuario, y frecuentemente **se indican en una sola oración**, como por ejemplo:

"Los usuarios pueden cargar fotos y proporcionar un título y una descripción a las nuevas fotos".

Las historias contienen solo los **detalles mínimos** para permitir a los desarrolladores estimar el esfuerzo involucrado en el desarrollo de la solución. **Los programadores extremos discuten cada historia de usuario con las partes interesadas, justo antes de escribir código.**

educación IT
CODE YOUR FUTURE

¡Sigamos trabajando!