



INICIO
GRABACIÓN





Arquitectura De TI

**Wilson Cárdenas Cr.
Fundación de educación superior San Jose
Semana 4**



INDICE

- 1 METODOLOGIAS TRADICIONALES Y EMERGENTES DEL DESARROLLO DEL SOFTWARE
- 2 CONCEPTOS (CARACTERISTICAS , FACES , VENTAJAS)
- 3 APLICADOS
- 4 CONCLUSIONES



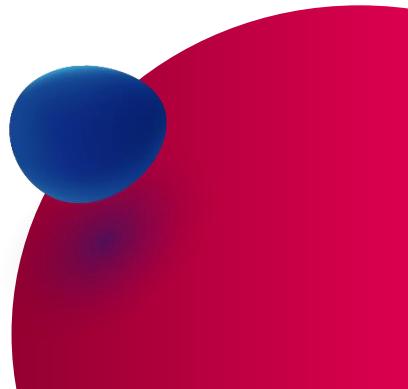
METODOLOGÍAS TRADICIONALES Y EMERGENTES DEL DESARROLLO DE SOFTWARE





METODOLOGÍA

Hace referencia al plan de investigación que permite cumplir ciertos objetivos en el marco de una ciencia.



METODOLOGÍAS TRADICIONALES



Sin embargo, para hacerlo sencillo, se puede definir la metodología de software como un enfoque, una manera de interpretar la realidad o la disciplina en cuestión, que en este caso particular correspondería a la Ingeniería de Software. De hecho, la metodología destinada al desarrollo de software se considera como una estructura utilizada para planificar y controlar el procedimiento de creación de un sistema de información especializada.



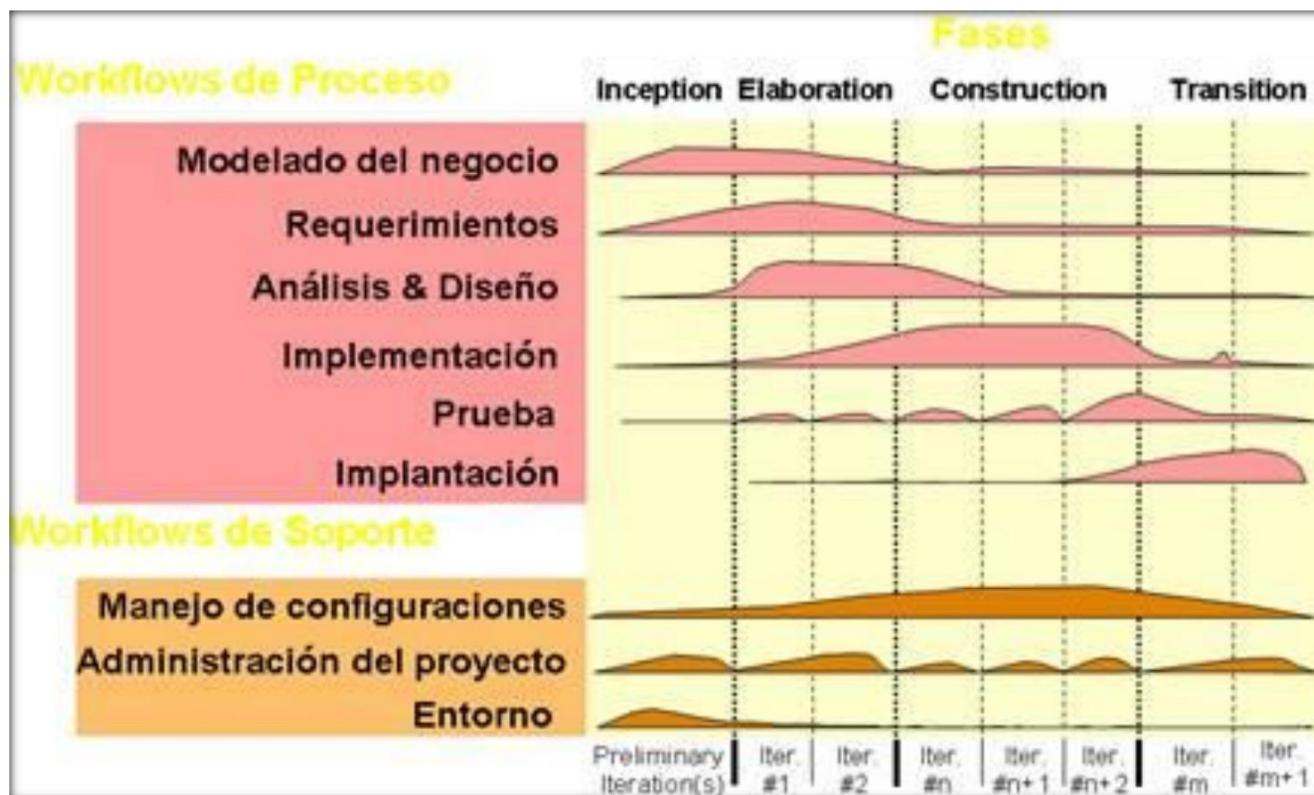
RUP

- (PROCESO DE DESARROLLO UNIFICADO)

RUP (PROCESO DE DESARROLLO UNIFICADO)



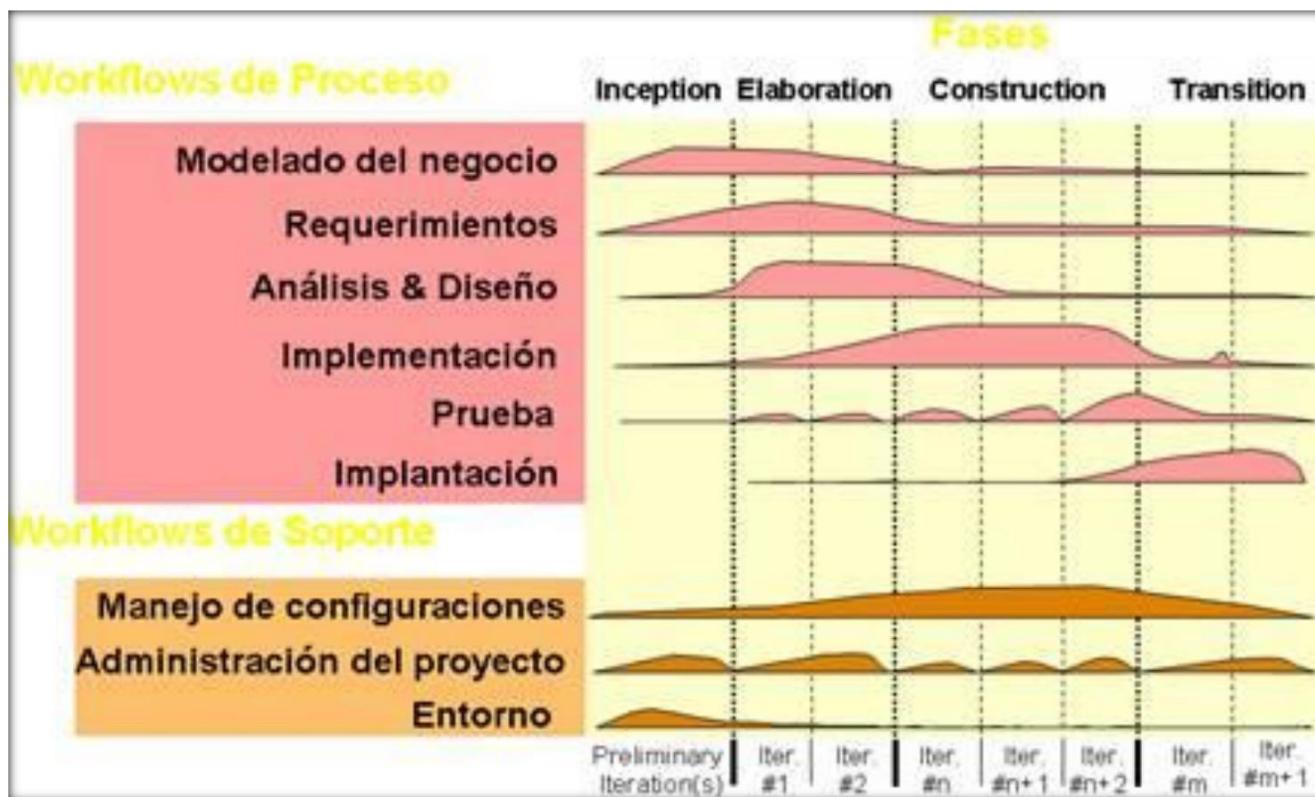
Metodología estándar más utilizada para la creación de sistemas orientados a objetos.





¿Cómo trabaja?

Divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto, con base a las actividades a realizar





CARACTERISTICAS

- Asigna tareas y responsabilidades (quién hace qué, cuándo y cómo).
- Desarrollo iterativo.
- Administración de requisitos.
- Modelado visual del software.
- Verificación de la calidad del software (UML).

FASES

- **FASE DE INICIO:** Definir y acordar el alcance del proyecto.
- **FASE DE ELABORACIÓN:** Se define la arquitectura base del sistema.
- **FASE DE DESARROLLO:** Clarificar los requisitos pendientes, administrar los cambios y las mejoras.
- **FASE DE TRANSICIÓN:** Ajustar los errores y defectos en las pruebas. Verificar que el producto cumpla con las especificaciones.





VENTAJAS

- Evaluación en cada fase que permite cambios de objetivos.
- Funciona bien en proyectos de innovación.
- Es sencillo, ya que sigue los pasos intuitivos necesarios a la hora de desarrollar el software.
- Seguimiento detallado en cada una de las fases.



DESVENTAJAS

- La evaluación de riesgos es compleja.
- El cliente deberá ser capaz de describir y entender a un gran nivel de detalle el alcance del proyecto.



A photograph showing two people in an office setting. A man in a grey t-shirt and a woman in a striped shirt are leaning over a desk, looking at something together. The background is blurred, showing other office workers and a red circular graphic.

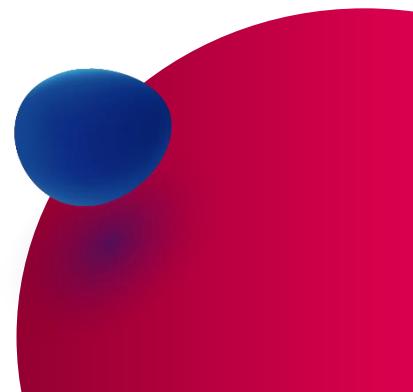
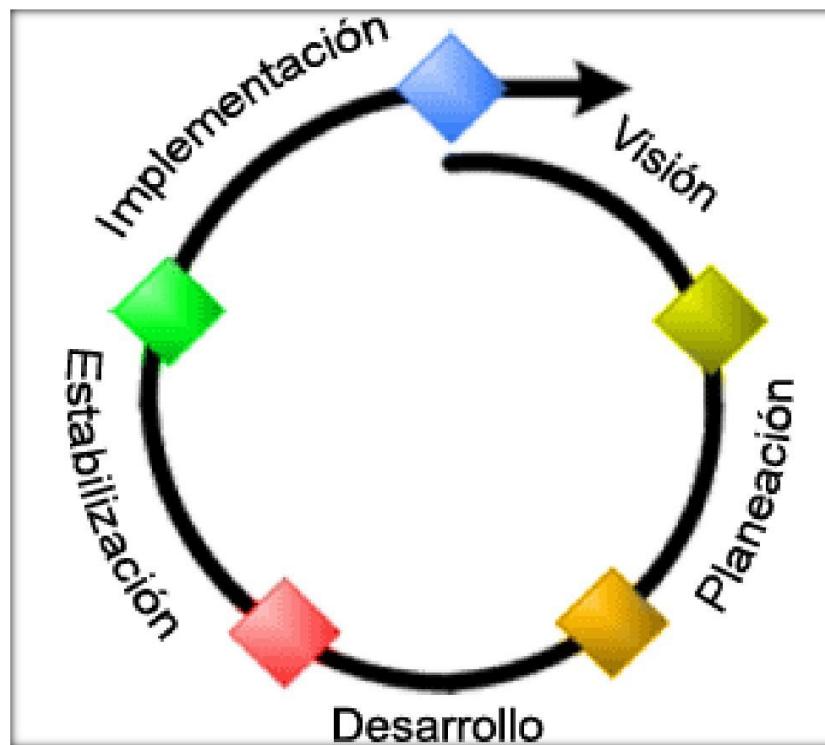
MSF

(MICROSOFT SOLUTION
FRAMEWORK)

MSF (MICROSOFT SOLUTION FRAMEWORK)



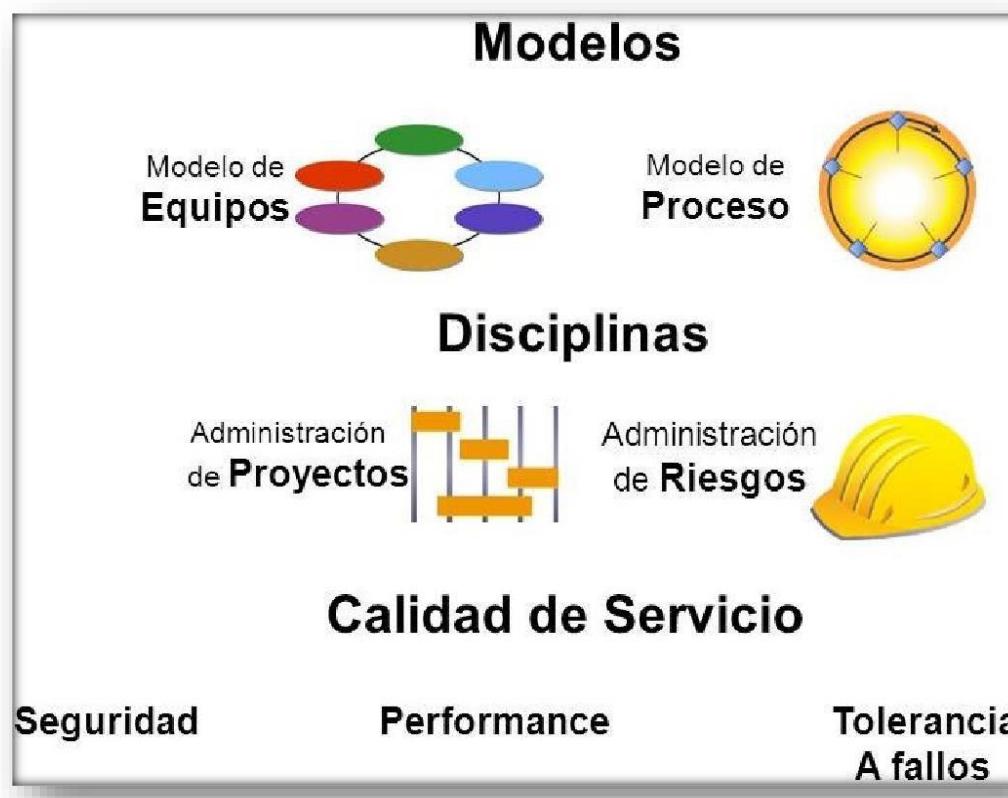
Es un enfoque personalizable para entregar correcta y más rápidamente soluciones tecnológicas.





¿Cómo trabaja?

Permite ajustar la escala de los proyectos, garantiza que los equipos cumplan una variedad de necesidades de las partes interesadas y define roles y responsabilidades en función de los objetivos.





Características

- Establece objetivos, roles y responsabilidades claros para el proyecto.
- Implementa un proceso iterativo, basado en hitos/puntos de control.
- Administra riesgos de forma proactiva.



FASES





VENTAJAS

- Soporte y mantenimiento.
- Aplica mucho el trabajo en equipo y la colaboración.
- Permite la reutilización de componentes ya desarrollados.
- Es un modelo enfocado a los requerimientos del usuario.



DESVENTAJAS

- Al estar basado en tecnología Microsoft, trata de obligar a usar sus propias herramientas.
- Solicita demasiada documentación en sus fases.
- Si el análisis de riesgos se hace muy exhaustivo puede retardar el proyecto.





WIN WIN





Es una adaptación del modelo espiral que se enfatiza en la **participación del cliente** en el proceso de desarrollo de un producto de software.

Esta constituido por 4 fases, planificación, análisis de riesgos, desarrollo y evaluación.

A lo largo de la aplicación de modelo de desarrollo en espiral, estas cuatro fases se van a repetir, con la diferencia de que el proyecto irá aumentando su complejidad, lapsos de tiempo de ejecución, volumen de tareas, etc.





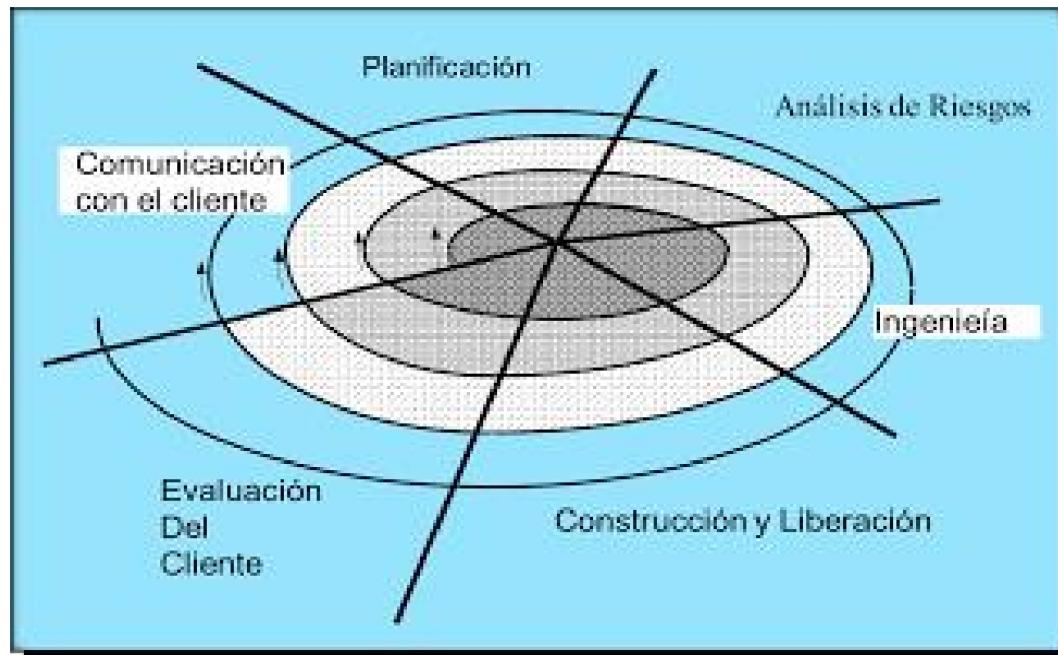
¿Cómo trabaja?

Un ejemplo práctico de modelo en espiral puede ser una aplicación de gestión de flota de transporte. La empresa de logística se acerca a nuestro equipo de desarrollo de software, y nos comunica el **objetivo** de querer tener un software que permita:

- Rastrear a los camiones en ruta.
- Conocer su carga y datos personales del conductor.
- Ver los kilómetros recorridos.

Se trata de un proyecto a largo plazo que representa un gran gasto para la empresa.

Al identificar el largo alcance del objetivo, el equipo de desarrollo coincide en que el modelo de desarrollo en espiral es la metodología adecuada.





¿Cómo trabaja?

Planificación

Se definen los requisitos para este primer vehículo con sus detalles específicos, previa comunicación con el cliente. En este primer ciclo, no se tendrá en cuenta la información personal del conductor, y se hará una estimación del coste, el calendario y los recursos para la iteración.

Análisis de riesgos

Se validará el prototipo pretendido, acorde a los plazos de tiempo y presupuesto entregado al cliente.

Desarrollo

Se desarrolla y valida el software del prototipo, sin la función del conductor y según el alcance acordado.

Evaluación

El prototipo se instala en el vehículo de prueba, que viaja en condiciones controladas por un breve lapso de tiempo. Se evalúa el seguimiento y la asignación de la carga.

VENTAJAS

- Incorpora objetivos de calidad.
- Integra el desarrollo con el mantenimiento.
- Mejoras y nuevos requerimientos sin romper con la metodología, ya que este ciclo de vida no es rígido



DESVENTAJAS

- Genera mucho tiempo en el desarrollo del sistema.
- Modelo costoso.
- Requiere experiencia en la identificación de riesgos





ICONIX

Metodología

- Consiste en un lenguaje de modelamiento y un proceso.
- El lenguaje de modelamiento es la notación gráfica (incluye diferentes tipos de diagramas)
- El proceso define quien debe hacer qué, cuando y como alcanzar un objetivo.



Metodología ICONIX

- Es un proceso simplificado en comparación con otros procesos más tradicionales, que unifica un conjunto de métodos de orientación a objetos con el objetivo de abarcar todo el ciclo de vida de un proyecto.
- Presenta claramente las actividades de cada etapa y exhibe una secuencia de pasos que deben ser seguidos.
- Está entre la complejidad del RUP (Rational Unified Processes) y la simplicidad de XP (Extreme Programming).

Tareas de ICONIX

- Análisis de Requisitos.
 - ◆ Modelo de Dominio.
 - ◆ Prototipación Rápida.
 - ◆ Modelo de Casos de Uso.
- Análisis y Diseño Preliminar.
 - ◆ Descripción de Casos de Uso.
 - ◆ Diagrama de Robustez.
- Diseño.
 - ◆ Diagrama de Secuencia.
- Implementación.
 - ◆ Escribir /Generar el Código.

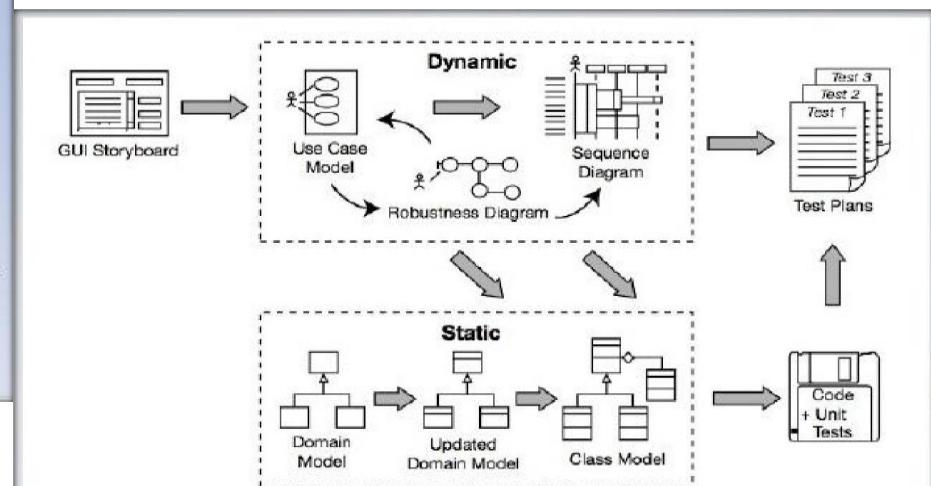
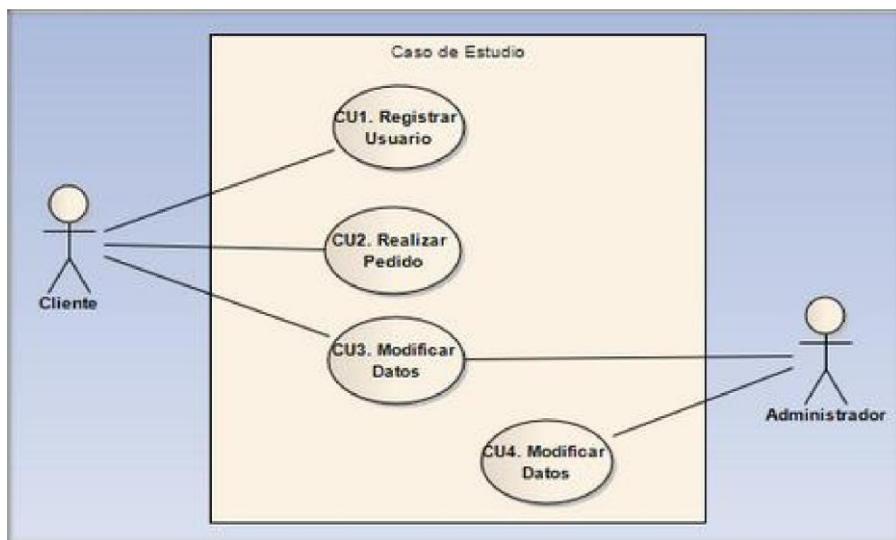


Presenta de forma clara las actividades de cada fase y la secuencia que debe ser seguida, está adaptado a los patrones y ofrece el soporte de UML.

ITERATIVO E INCREMENTAL: El desarrollo del proyecto se organiza en series de mini-proyectos cortos, llamados iteraciones. Cada parte debe de funcionar integralmente.

TRAZABILIDAD: Cada paso está referenciado por algún requisito en específico.

DINÁMICA DEL UML: Diagramas de caso de uso, de secuencia y de colaboración.

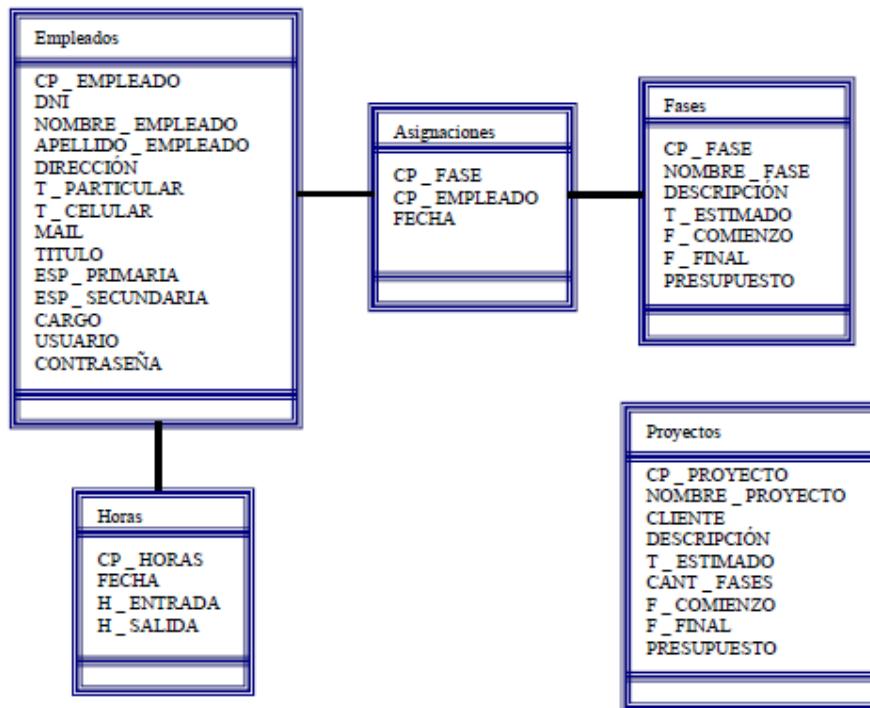


PROCESO ICONIX



Análisis de Requisitos

- Se realiza un relevamiento de todos los requisitos que en principio deberían ser parte del sistema.
- Se debe capturar información sobre lo que les gusta y lo que les desagrada a los usuarios.
- **Modelo de Dominio:**
 - ◆ Con los requisitos se construye el diagrama de clases, que representa el modelo estático del sistema.

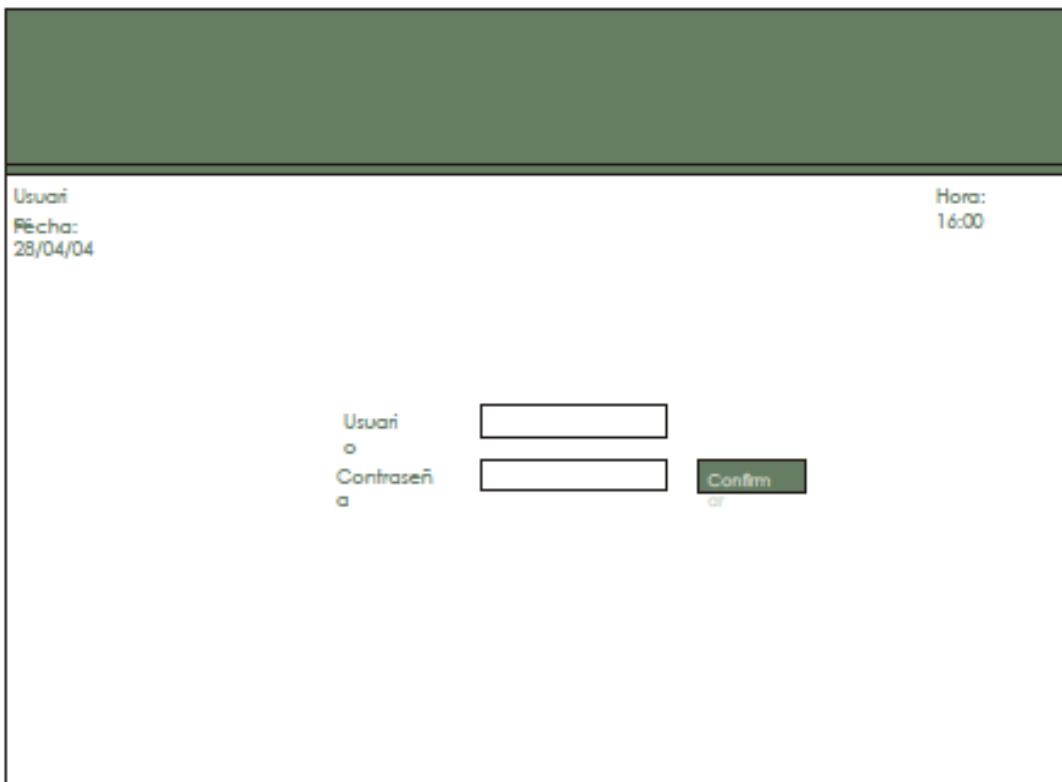




➤ Prototipación Rápida:

- ◆ Se usa para simular el diseño del sistema.
 - ◆ Se espera que los usuarios lo evalúen como si fuera el sistema final.
 - ◆ Los cambios al prototipo son planificados con los usuarios antes de llevarlos a cabo.
- El proceso se repite y finaliza cuando los usuarios y analistas están de acuerdo en que el sistema ha evolucionado lo suficiente como para incluir todas las características necesarias o cuando es evidente que no se obtendrá mayor beneficio con una iteración adicional.

Iniciar Sesión

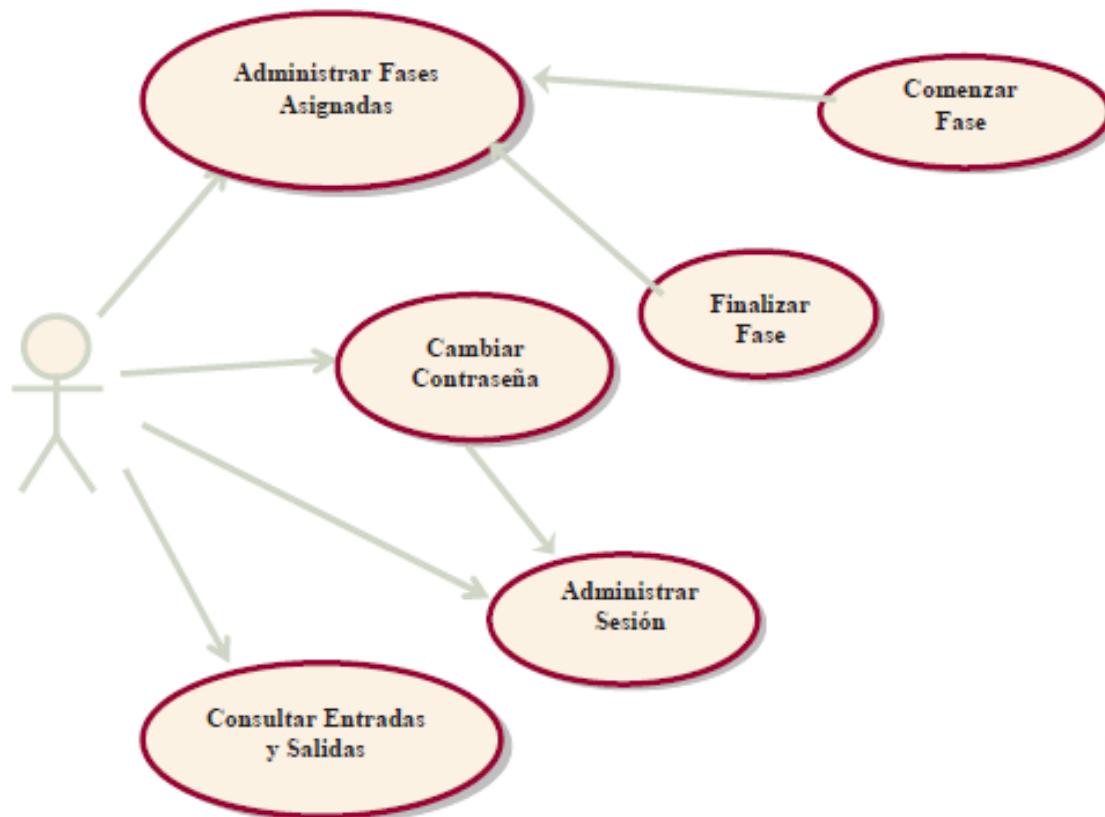


A prototypic user interface for a login session. The interface is divided into several sections:

- Top Section:** A large green rectangular area.
- User Information:** A white rectangular area containing:
 - Label: "Usuario"
 - Text: "Pérez"
 - Text: "28/04/04"
- Date and Time:** A white rectangular area containing:
 - Label: "Hora:"
 - Text: "16:00"
- Login Fields:** A white rectangular area containing:
 - Text: "Contraseña"
 - Text: "1234567890"
 - Text: "Confirmar"

Modelo de Casos de Uso:

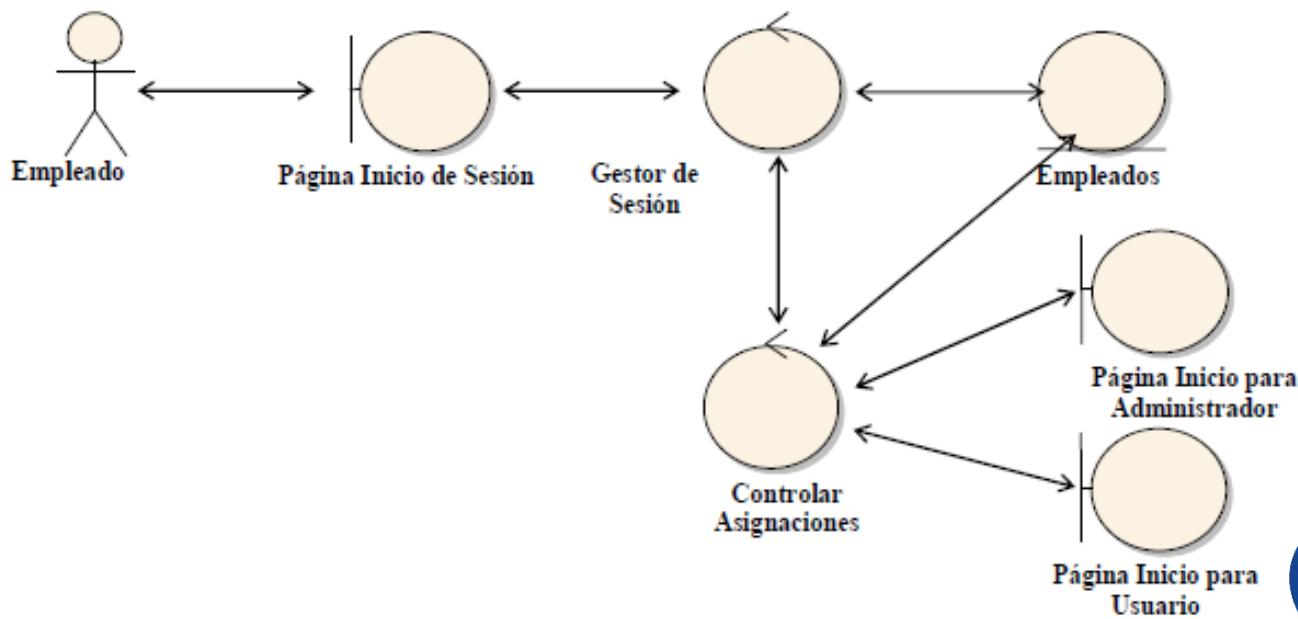
- ♦ El modelo de los casos de uso comprende los actores, el sistema y los propios casos de uso.
- ♦ Los casos de uso permiten a los usuarios estructurar y articular sus deseos; les obligan a definir la manera como querían interactuar con el sistema, a precisar qué informaciones quieren intercambiar y a describir lo que debe hacerse para obtener el resultado esperado.



➤ Diagrama de Robustez:

- ◆ Ilustra gráficamente las interacciones entre los objetos participantes de un caso de uso. Los que pueden ser:

- * Objetos de interfaz. (Pantallas)
- * Objetos entidad. (Almacenamientos)
- * Objetos de control. (Gestores)



VENTAJAS

- Proceso ágil para obtener un sistema informático.
- Dedicada a la construcción de sistemas de gestión de pequeña y mediana
- complejidad, con la participación de los usuarios finales.



DESVENTAJAS

- Necesita información rápida y puntual de los requisitos, del diseño y de las estimaciones.
- Es una metodología que no debe ser usada en proyectos de larga duración.





FDD (DESARROLLO BASADO EN FUNCIONES)



El desarrollo basado en funciones (FDD por sus siglas en inglés) es una metodología de desarrollo de software centrada en el cliente conocido por iteraciones cortas y lanzamientos frecuentes.

Es una metodología ágil diseñada para el desarrollo de software, basada en la calidad y el monitoreo constante del proyecto.

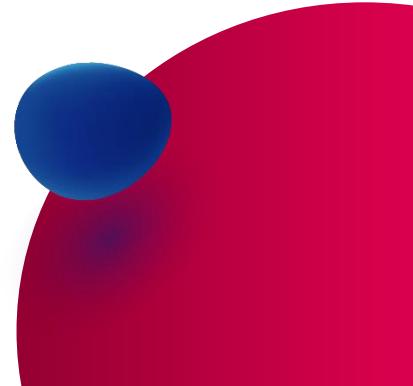




Como funciona

Se enfoca en iteraciones cortas, que permiten entregas tangibles del producto en un periodo corto de tiempo, de como máximo dos semanas.

Al lanzar nuevas funciones de forma incremental, los desarrolladores pueden priorizar las solicitudes de los clientes, responder a las solicitudes de manera oportuna y mantener a los clientes satisfechos. Para lograr esto, los desarrolladores planifican qué características son capaces de crear, dividen las solicitudes complejas en una serie de conjuntos de características más pequeños y luego crean un plan sobre cómo completar cada objetivo con el tiempo.



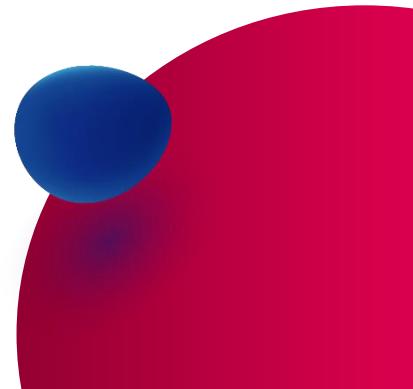
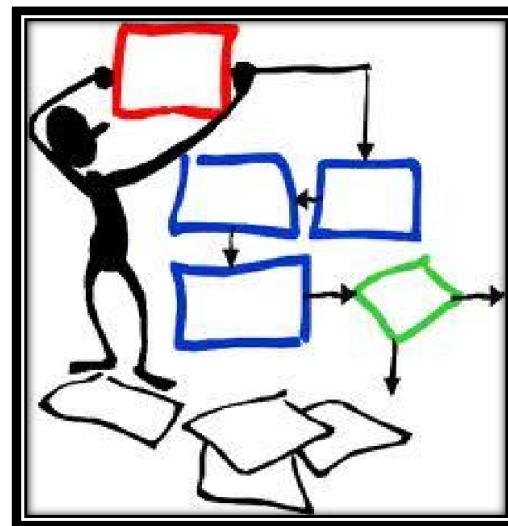


Características

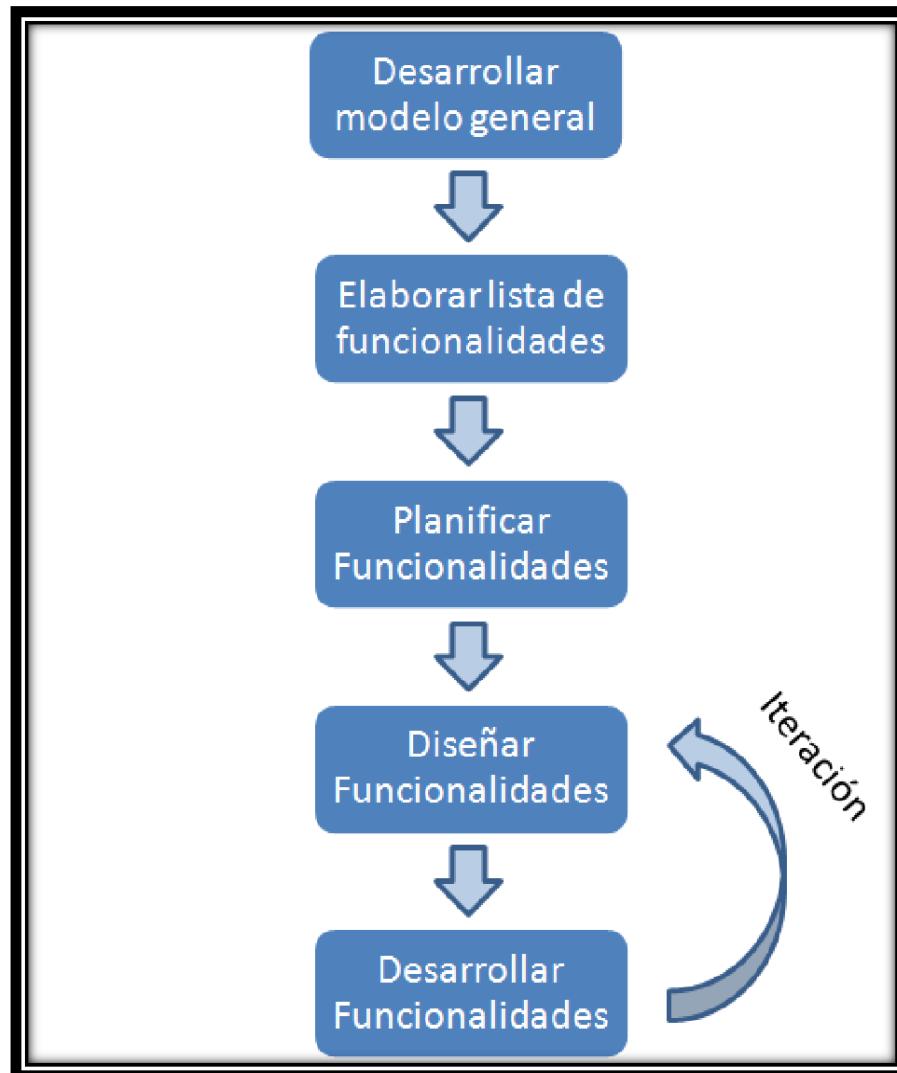
Se preocupa por la calidad y realiza un monitoreo constante del proyecto.

Se basa en un proceso iterativo, con iteraciones cortas que producen un software funcional.

No hace énfasis en la obtención de los requerimientos sino en como se realizan las fases de diseño y construcción.



Fases



VENTAJAS

- El equipo de desarrollo no malgasta el tiempo y dinero del cliente.
- Cada componente del producto final ha sido probado y satisface los requerimientos.
- Rápida respuesta a cambios de requisitos a lo largo del desarrollo.



DESVENTAJAS

- Problemas derivados de la comunicación oral.
- La documentación y los diseños convencionales, los proyectos ágiles
- Dependen críticamente de las personas.
- La falta de documentación hacen difícil que pueda reutilizarse el código ágil.



SCRUM

Scrum es una metodología ágil y flexible para gestionar el desarrollo de software, cuyo principal objetivo es maximizar el retorno de la inversión para su empresa (ROI). Se basa en construir primero la funcionalidad de mayor valor para el cliente y en los principios de inspección continua, adaptación, auto-gestión e innovación.





Como trabaja?

Permite en cualquier momento realinear el software con los objetivos del cliente ya que puede introducir cambios funcionales o de prioridad en el inicio de cada nueva iteración sin ningún problema.

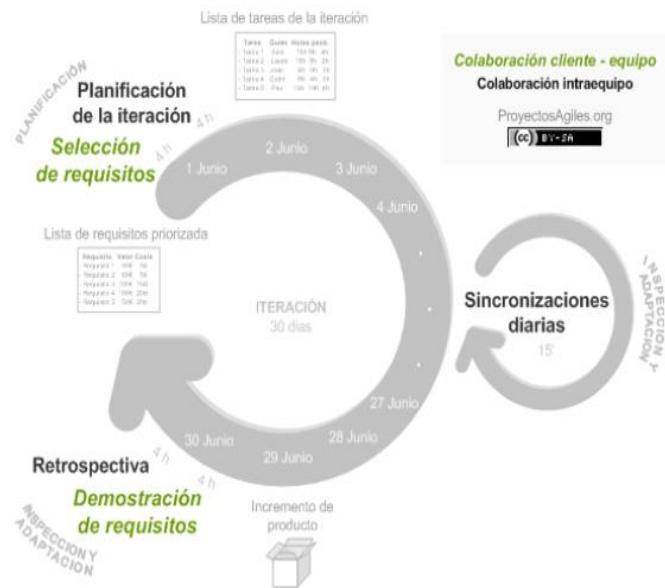
Con la metodología Scrum el cliente se entusiasma y se compromete con el proyecto dado que lo ve crecer iteración a iteración. Asimismo le permite en cualquier momento realinear el software con los objetivos de negocio de su empresa, ya que puede introducir cambios funcionales o de prioridad en el inicio de cada nueva iteración sin ningún problema.





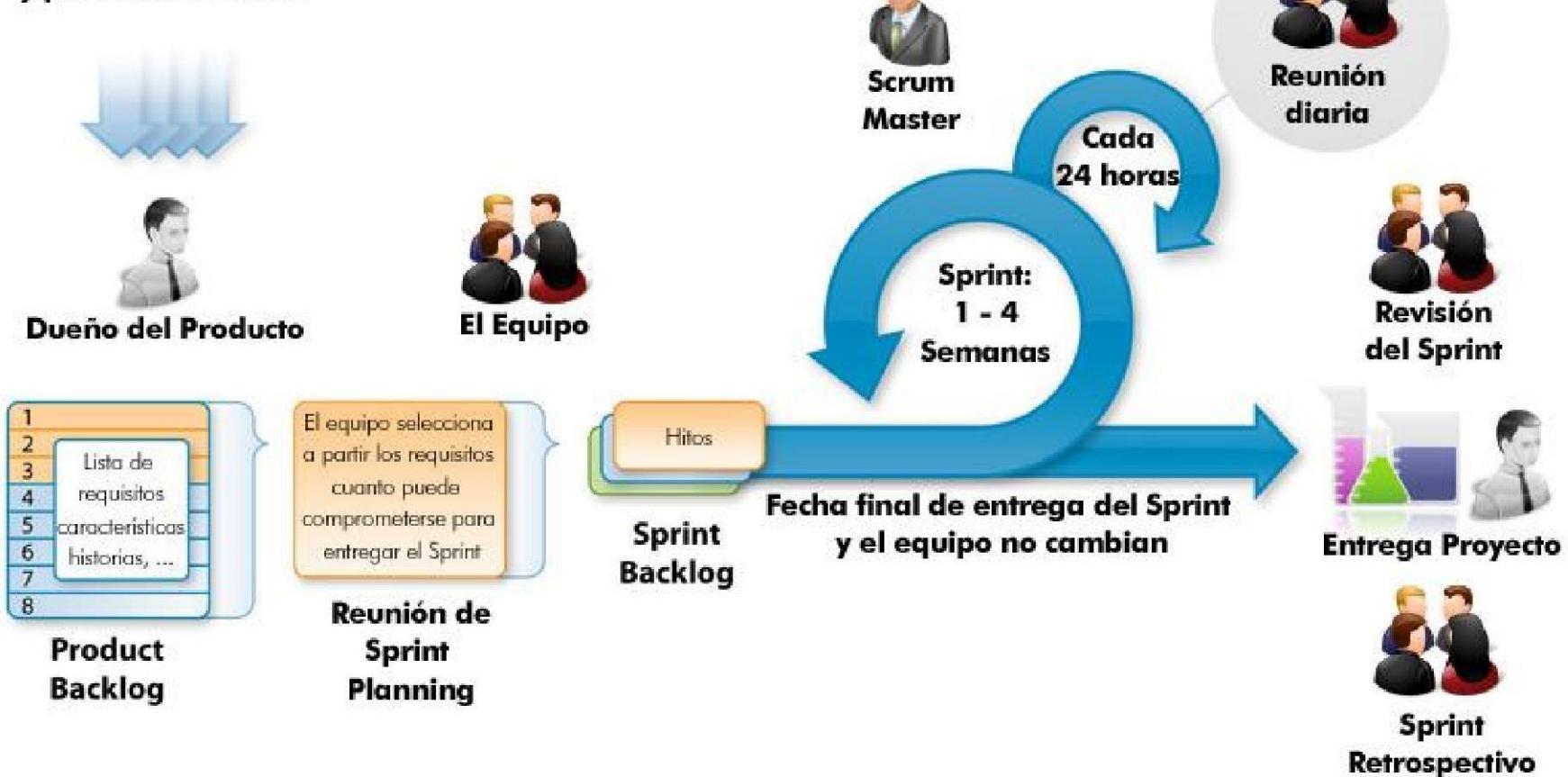
Características

- Es un método ágil.
 - Liviano, iterativo e incremental.
 - Adaptativo y flexible.
 - Predisposición y respuesta al cambio.
 - Comunicación directa cliente-servidor
-
- En Scrum un proyecto se ejecuta en ciclos temporales cortos y de duración fija (iteraciones que normalmente son de 2 semanas, aunque en algunos equipos son de 3 y hasta 4 semanas, límite máximo de feedback de producto real y reflexión). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.



FASES

Aportaciones de los ejecutivos,
equipo, clientes, usuarios
y partes interesadas





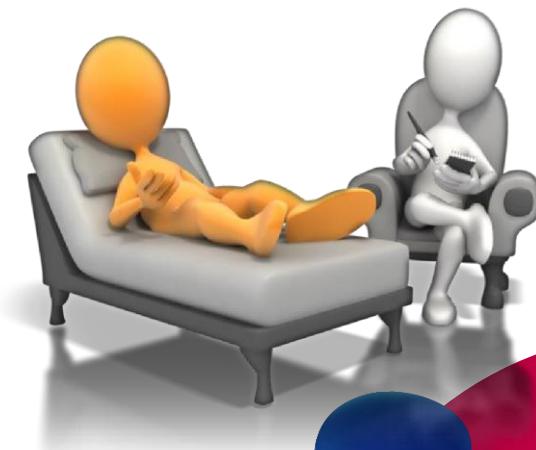
VENTAJAS

- Flexibilidad a cambios.
- Mayor productividad y calidad del software.
- Predicciones de tiempo.
- Reducción de riesgos.



DESVENTAJAS

- Dificultad de aplicación en grandes proyectos.
- Solo funciona bien en equipos pequeños y agiles.
- Si una tarea no esta bien definida puede incrementar costo y tiempo





**FIN DE
GRABACIÓN**