

# Feature Driven Development

## (Desarrollo Basado en Funciones)

---

Eduardo Flores Gallegos  
Carlos Romo Padilla



# ¿Qué es Feature Driven Development?

- Es una metodología ágil diseñada para el desarrollo de software, basada en la calidad y el monitoreo constante del proyecto.
- Fue desarrollada por Jeff De Luca y Peter Coad a mediados de los años 90. Esta metodología se enfoca en iteraciones cortas, que permiten entregas tangibles del producto en un periodo corto de tiempo, de como máximo dos semanas.[1]

# Características

- Se preocupa por la calidad, por lo que incluye un monitoreo constante del proyecto.
- Ayuda a contrarrestar situaciones como el exceso en el presupuesto, fallas en el programa o el hecho de entregar menos de lo deseado.
- Propone tener etapas de cierre cada dos semanas. Se obtienen resultados periódicos y tangibles.
- Se basa en un proceso iterativo con iteraciones cortas que producen un software funcional que el cliente y la dirección de la empresa pueden ver y monitorear.
- Define claramente entregas tangibles y formas de evaluación del progreso del proyecto.
- No hace énfasis en la obtención de los requerimientos sino en como se realizan las fases de diseño y construcción. [1]





# Ventajas

- El equipo de desarrollo no malgasta el tiempo y dinero del cliente desarrollando soluciones innecesariamente generales y complejas que en realidad no son un requisito 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. Entrega continua y en plazos cortos de software funcional.
- Trabajo conjunto entre el cliente y el equipo de desarrollo.
- Minimiza los costos frente a cambios.[1]



# Desventajas

- Falta de documentación del diseño. El código no puede tomarse como una documentación. En sistemas de tamaño grande se necesitan leer los cientos o miles de páginas del listado de código fuente.
- Problemas derivados de la comunicación oral. Este tipo de comunicación resulta difícil de preservar cuando pasa el tiempo y está sujeta a muchas ambigüedades.
- Fuerte dependencia de las personas. Como se evita en lo posible la documentación y los diseños convencionales, los proyectos ágiles dependen críticamente de las personas.
- Falta de reusabilidad. La falta de documentación hacen difícil que pueda reutilizarse el código ágil.[1]

# Proceso

- Desarrollar un modelo global: Al inicio del desarrollo se construye un modelo teniendo en cuenta la visión, el contexto y los requisitos que debe tener el sistema a construir. Este modelo se divide en áreas que se analizan detalladamente. Se construye un diagrama de clases por cada área.
- Construir una lista: Se elabora una lista que resuma las funcionalidades que debe tener el sistema, cuya lista es evaluada por el cliente. Cada funcionalidad de la lista se divide en funcionalidades más pequeñas para un mejor entendimiento del sistema.
- Planear: Se procede a ordenar los conjuntos de funcionalidades conforme a su prioridad y dependencia, y se asigna a los programadores jefes.
- Diseñar: Se selecciona un conjunto de funcionalidades de la lista. Se procede a diseñar y construir la funcionalidad mediante un proceso iterativo, decidiendo que funcionalidad se van a realizar en cada iteración. Este proceso iterativo incluye inspección de diseño, codificación, pruebas unitarias, integración e inspección de código.
- Construir : se procede a la construcción total del proyecto.[1]



# Equipo de Trabajo

- Arquitecto jefe: Realiza el diseño global del sistema. Ejecución de todas las etapas.
- Director de desarrollo: Lleva diariamente las actividades de desarrollo. Resuelve conflictos en el equipo. Resuelve problemas referentes a recursos.
- Programador Jefe: Analiza los requerimientos. Diseña el proyecto. Selecciona las funcionalidades a desarrollar de la última fase del FDD.
- Propietario de clases: Responsable del desarrollo de las clases que se le asignaron como propias. Participa en la decisión de que clase será incluida en la lista de funcionalidades de la próxima iteración.
- Expertos de dominio: Puede ser un usuario, un cliente, analista o una mezcla de estos. Poseen el conocimiento de los requerimientos del sistema. Pasa el conocimiento a los desarrolladores para que se asegure la entrega de un sistema completo. [1]

# Gracias Por su Atención!!!!

## Referencia

- [1] Donde encontré la información <http://metodologiafdd.blogspot.com/>