# Unidad 2

Ciclo de vida de desarrollo de software:

* Ciclo de codificar y corregir
  + Codificar-corregir presenta una serie de efectos negativos:
    - La calidad del producto final es baja
    - Los sistemas a menudo terminan como un descoordinado enredo de código que carece de adaptabilidad
    - Los sistemas desarrollados “ad hoc” son difíciles de extender y mantener
    - Los sistemas terminados son complicados y tienden a tener una especialdiad pobre
  + Definición de ciclo de vida de desarrollo de sofware:

Conjunto de actividades que se llevan a cabo para desarrollar e implantar un software

* + Metodología:   
    Cada metodología especifica de que manera se llevan a cabo las actividades para el desarrollo del sistema
* Ciclo de vida clásico (modelo en cascada):
  + Características:
    - Implantación ascendente
    - Nada esta hecho hasta que todo esta terminado
    - Las fallas mas triviales se encuentran al comienzo del period o de prueba y las mas graves al final
* Ciclo de vida semiestructurado:
  + Características:
    - Implementación descendente:  
      Primero se codifican módulos de alto nivel y luego los de bajo nivel
    - Se utiliza diseño estructurado y programación estructurada
    - El análisis produce una especificación de requerimientos textual narrativa, la cual debe ser convertida por el diseñador
* Ciclo de vida estructurado:
  + Caracteristicas:
    - Las actividades se realizan en forma paralela en progresivos niveles de refinamiento
    - Existe retroalimnetacion entre actividades
    - Las fallas y desvíos se detectan y corrigen oportunamente
    - Utilización de módulos gráficos con apoyo textual

Evolución de las metodologías de desarrollo de sofware:

* Modelo de prototipos: acercamiento al cliente con una plantilla sin programar
* Desarrollo rápido de aplicaciones (DRA): software sin requerimientos cambiarios
* Modelo incremental: se toma las mejores partes funcionales de cada entrega
* Modelo en espiral: similar al incremental pero iterando en el tiempo
* Las metodologías agiles: conjunto de metodso que permiten adaptar el modo de trabajo a las condiciones del proyecto, aportando flexibilidad, eficiencia y, por lo tanto, logrando un mejor producto a menor coste.

Manifiesto ágil:

* Valorar a los individuos e interacciones sobre los procesos y herramientas
* Valorar el software funcionando sobre la documentación extensiva
* Valorar la colaboración con el cliente sobre la negociación contractual
* Valorar la respuesta nate el cambio en lugar de seguir un plan

Meotdologias agiles mas utilizadas:

* Programación extrema
* Scrum
* Kanban
* Lean

Diseño de intefaces:

* Eficacia: significa que las pantallas satisfagan propósitos específicos del sistema de informacion
* Preicsion: de refiere a que el diseño asegura el llenado adecuado
* Facilidad de uso: implica que las pantallas o formas serán explicitas y no requerirán de tiempo adicional para descifrarse
* Consistencia: significa que las pantallas o formas ordeenn los datos de manera similar de una aplicación
* Sencillez: se refiere a mantener en un mínimo los elementos indispensables que centren la atención del usuario
* Atracción: implica que el usuario disfrutara del transito a través de las pantallas o formas cuyo diseños les sean atractivos

Diseño de interfaces 2:

* Diseño de document Fuente:  
  un documento fuente es un formulario utilizado para registrar los datos que luego se introducirán en un ordenador
* Diseño de pantallas:
  + Mantener la pantalla simple
  + Facilitar el movimiento entre pantallas
  + Conservar la consistencia entre pantallas
* Consideraciones al elegir la tecnología de salida
  + Quien la usara
  + Cuantas personas necesitan de la salida
  + Donde se necesita
  + Cual es su propósito
  + Con que frecuencia se la necesita
  + Bajo que regulaciones particulares se produce la salida
  + Cuales son los costos iniciales y mantenimiento
  + Cuales son los requisitos ambuientales para las tecnologías de salida