



# Modelos de Construcción de Software

## *Grupo 7*

- CÓRDOVA SANDOVAL, Rafael Anthony 17200268
- CHÁVEZ SILUPÚ, Erick Alberto 17200267
- CUELLO APAZA, Alexander Gabriel 17200269
- BALTAZAR ALIAGA, Sebastián Pablo 15200198
- ANDÍA CUSI, Juan Vicente 16200055



# ***METODOLOGÍAS ESTRUCTURADAS:***

## ***a) Modelo en espiral***

El desarrollo en espiral es un modelo de procedimiento para el desarrollo de software elaborado por Barry W. Boehm en el año 1986. Parte de la base de que el desarrollo de aplicaciones se debe llevar a cabo en un ciclo iterativo que se debe repetir tantas veces como sea necesario hasta alcanzar el objetivo.

## ***b) Modelo en cascada***

Es un procedimiento lineal que se caracteriza por dividir los procesos de desarrollo en sucesivas fases de proyecto. Al contrario que en los modelos iterativos, cada una de estas fases se ejecuta tan solo una vez. Los resultados de cada una de las fases sirven como hipótesis de partida para la siguiente.

# Modelo en Espiral:

Fase 1:

Definición de objetivos y alternativas y descripción de las condiciones generales

Fase 2:

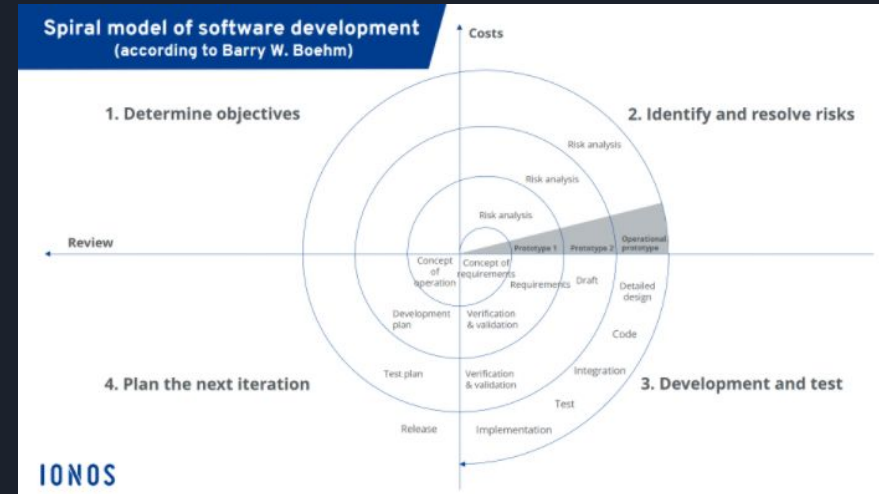
Valoración de las alternativas

Fase 3:

Desarrollo y revisión del resultado intermedio

Fase 4:

Planificación del siguiente ciclo





## Ventajas

- Modelo flexible y genérico
- Posible integración temprana de promotores y usuarios
- Comprobaciones periódicas y enfocadas al riesgo
- Conciliación perfecta entre exigencias técnicas y diseño
- Máximo control sobre los costes, recursos y la calidad del proyecto de software
- Apropiado para entornos técnicos novedosos

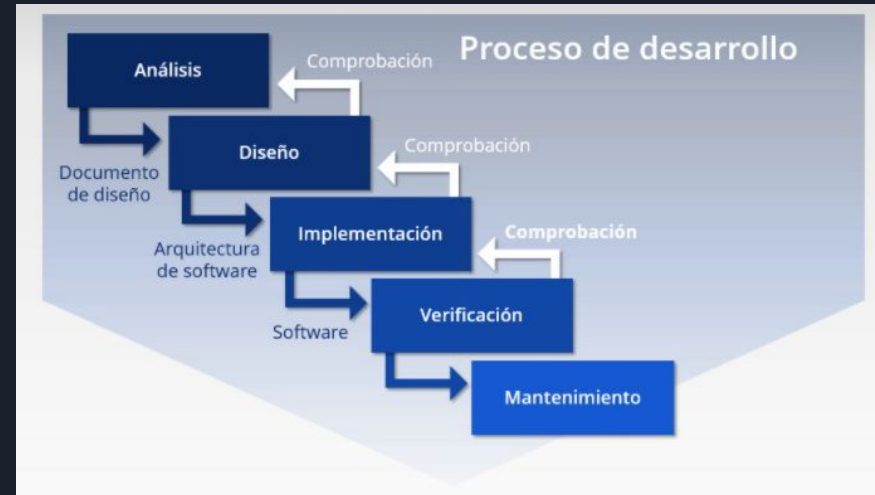
## Inconvenientes

- Gran esfuerzo de gestión
- Las decisiones periódicas pueden dilatar el proceso de desarrollo
- Hay errores e incongruencias conceptuales que se abren paso fácilmente al producto final a través del proceso de desarrollo desglosado
- Know-how en análisis y gestión de riesgo esencial, pero no siempre disponible
- No es apropiado para pequeños proyectos con un riesgo manejable

# Modelo en Cascada:

Fases definidas por Royce aunque en la práctica, se aplican diversas versiones del modelo. Los más habituales son los modelos que dividen los procesos de desarrollo en cinco fases.

- Análisis: planificación, análisis y especificación de los requisitos.
- Diseño: diseño y especificación del sistema.
- Implementación: programación y pruebas unitarias.
- Verificación: integración de sistemas, pruebas de sistema y de integración.
- Mantenimiento: entrega, mantenimiento y mejora.





## Ventajas

- ✓ Una estructura sencilla gracias a unas fases de proyecto claramente diferenciadas.
- ✓ Buena documentación del proceso de desarrollo a través de unos hitos bien definidos.
- ✓ Los costes y la carga de trabajo se pueden estimar al comenzar el proyecto.
- ✓ Aquellos proyectos que se estructuran en base al modelo en cascada se pueden representar cronológicamente de forma sencilla.

## Inconvenientes

- ✗ Por norma general, los proyectos más complejos o de varios niveles no permiten su división en fases de proyecto claramente diferenciadas.
- ✗ Poco margen para realizar ajustes a lo largo del proyecto debido a un cambio en las exigencias.
- ✗ El usuario final no se integra en el proceso de producción hasta que no termina la programación.
- ✗ En ocasiones, los fallos solo se detectan una vez finalizado el proceso de desarrollo.

# METODOLOGÍA ORIENTADA A OBJETOS:

El modelo Orientado a Objetos difiere considerablemente del modelo estructurado ya que aquí no se realiza un problema en términos de tareas (subrutinas) ni en términos de datos, sino que se analiza el problema como un sistema de objetos que interactúan entre sí.

Una vista rapida a la metodología orientada a objetos (OO):



# Modelo Iterativo:

- Esta idea es la base de varios métodos de desarrollo de software como RUP y otros métodos de desarrollo ágiles.
- La idea básica es desarrollar el sistema siguiendo etapas incrementales caracterizadas por generación de sucesivas versiones que van abarcando requerimientos hasta completar el sistema.
- Iterativo: cada vez visitamos las etapas del modelo en cascada, rehacemos, refinamos y extendemos lo hecho.
- Incremental: regularmente integramos los avances para generar una versión con sentido para el cliente.







# *Estilo arquitectónico orientado a objetos:*

Las características clave del estilo arquitectónico orientado a objetos son:

- Abstracción. Esto le permite reducir una operación compleja a una generalización que conserva las características básicas de la operación.
- Composición. Los objetos pueden ensamblarse a partir de otros objetos y pueden elegir ocultar estos objetos internos de otras clases o exponerlos como interfaces simples.
- Herencia. Los objetos pueden heredar de otros objetos y usar la funcionalidad en el objeto base o anularlo para implementar un nuevo comportamiento.
- Encapsulación. Los objetos exponen la funcionalidad sólo a través de métodos, propiedades y eventos, y ocultan los detalles internos, como el estado y las variables de otros objetos.
- Polimorfismo. Esto le permite anular el comportamiento de un tipo base que admite operaciones en su aplicación mediante la implementación de nuevos tipos que son intercambiables con el objeto existente.
- Desacoplamiento. Los objetos se pueden desacoplar del consumidor definiendo una interfaz abstracta que el objeto implementa y que el consumidor puede entender.

# MODELO ORIENTADO A OBJETOS

## Instancia

Es la particularización, realización específica u ocurrencia de una determinada clase, entidad (modelo entidad-relación) o prototipo.

## Atributos

Los atributos están asociados a clases y objetos, y describen la clase o el objeto de alguna manera.

## Clases

Una clase es esencialmente un proyecto, a partir del cual puede crear objetos. Una clase define las características de un objeto, incluyendo las propiedades que definen los tipos de datos que ese objeto puede contener y los métodos que describen el comportamiento del objeto.

## Mensajes

Los mensajes son el medio a través del cual interactúan los objetos. Un mensaje estimula la ocurrencia de cierto comportamiento en el objeto receptor. El comportamiento se realiza cuando se ejecuta un método.

## Herencia

Significa que una clase adquiere la funcionalidad de otra de nivel superior, para luego potenciarla con características particulares. Es decir, la herencia significa que la clase derivada tendrá todos los miembros (propiedades, métodos y eventos) de la clase base, y luego agregará los propios.

## Encapsulamiento

Es el proceso de ocultar todos los secretos de un objeto que no contribuyen a sus características esenciales.

## Métodos

es un modo, manera o forma de realizar algo de forma sistemática, organizada y/o estructurada. Hace referencia a una técnica o conjunto de tareas para desarrollar una tarea.

# METODOLOGÍAS ÁGILES:

## XP:

La metodología XP o Programación Extrema es una metodología ágil y flexible utilizada para la gestión de proyectos.

Extreme Programming se centra en potenciar las relaciones interpersonales del equipo de desarrollo como clave del éxito mediante el trabajo en equipo, el aprendizaje continuo y el buen clima de trabajo.

Esta metodología pone el énfasis en la retroalimentación continua entre cliente y el equipo de desarrollo y es idónea para proyectos con requisitos imprecisos y muy cambiantes.

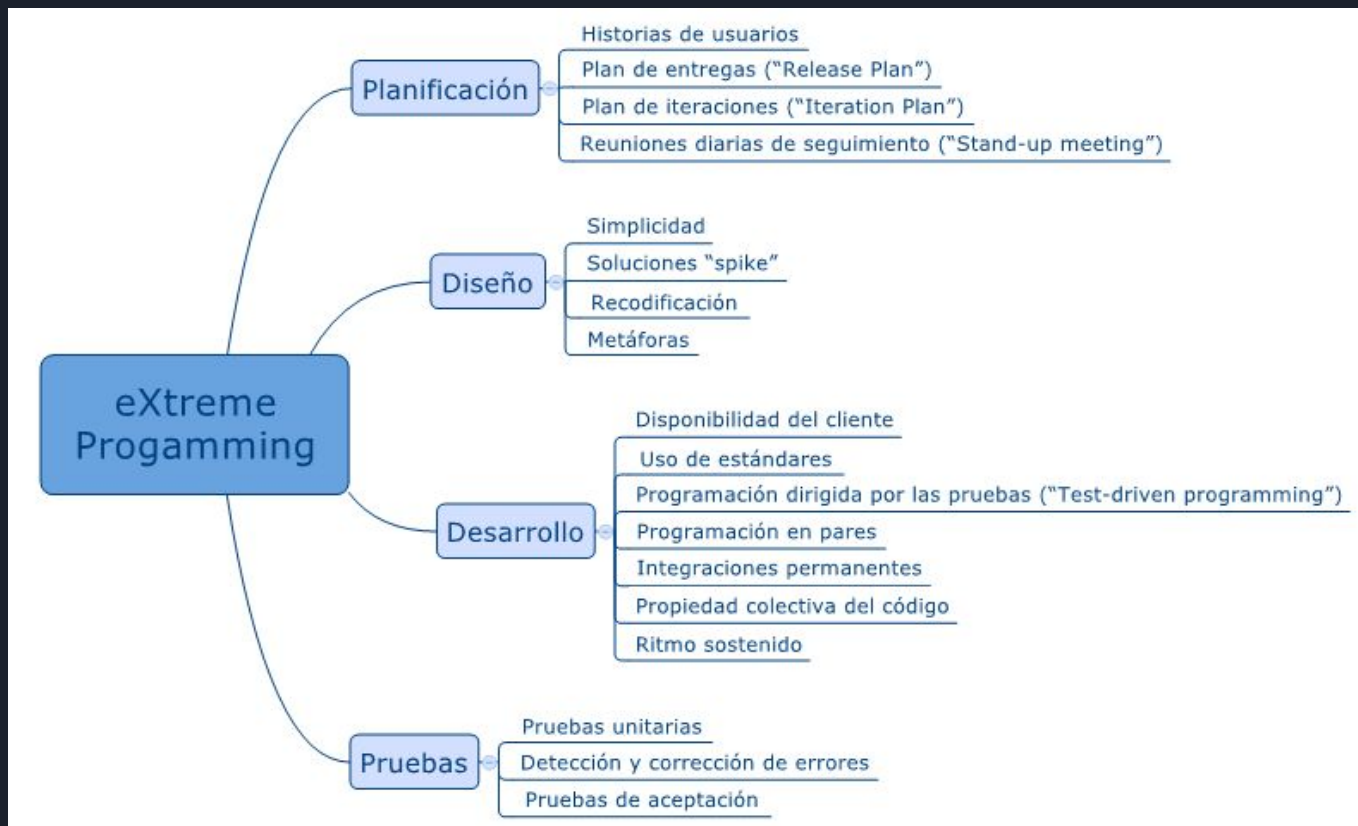


# Características y Valores:

- Se considera al equipo de proyecto como el principal factor de éxito del proyecto
- Software que funciona por encima de una buena documentación.
- Interacción constante entre el cliente y el equipo de desarrollo.
- Planificación flexible y abierta.
- Rápida respuesta a cambios.



# Etapas de la Metodología XP:



# Propuesta Arquitectónica Gráfica / Ejemplos del desarrollo de la metodología:

**Encuestas**

Responsabilidad	Colaboración
Guardar información de las encuestas	
Verificar información	

**Encuestados**

Responsabilidad	Colaboración
Guardar información de los encuestados	
Verificar participantes	participantes
Verificar carnet	Alumnos

## CASO DE PRUEBA

Código: 7

Nº Historia de Usuario: 7

Historia de Usuario: **Encuesta de Alumnos**

Condiciones de Ejecución: **Se requiere que el usuario encargado de esta tarea se encuentre con sesión iniciada en el sistema**

Entrada/Pasos de Ejecución:

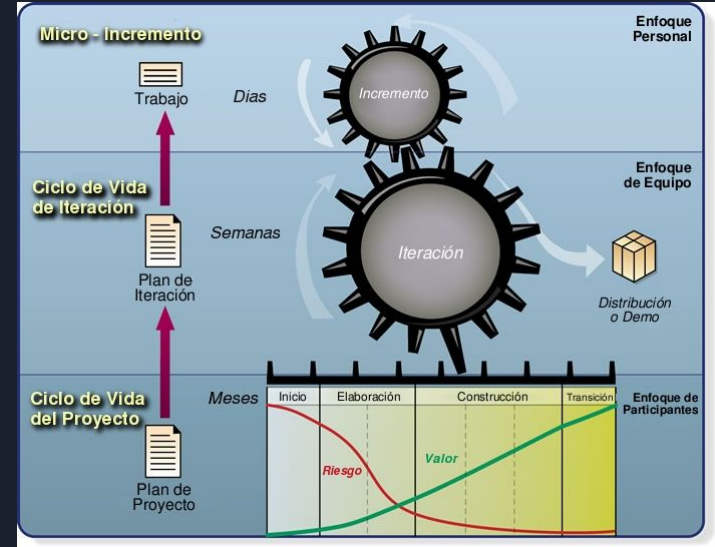
**El usuario administrador del sistema tendrá que seleccionar en el menú encuestas la opción ENCUESTA ESTUDIANTE, Se le mostrará un formulario en donde tendrá que definir la información para las encuestas de alumnos Posteriormente presionar sobre el botón GUARDAR**

Resultado Esperado: **Encuesta de Alumnos guardada.**

Evaluación de la Prueba: **La prueba se concluyó satisfactoriamente.**

# Open-UP:

OpenUP es un Proceso Unificado que aplica enfoques iterativos e incrementales dentro de un ciclo de vida estructurado, utiliza una filosofía ágil que se enfoca en la naturaleza de colaboración en el desarrollo de software. Es una herramienta diagnóstica que puede extenderse para hacer frente a una amplia variedad de proyectos. Está basado en casos de uso, la gestión del riesgo, y una arquitectura centrada en impulsar el desarrollo.





## Principios:



## Elementos:

### Disciplinas

- OpenUP se centra en las siguientes disciplinas: requisitos, arquitectura, desarrollo, pruebas, gestión de proyecto, gestión de la configuración y del cambio.

### Tareas

- Se define tarea como la unidad de trabajo que debe ser realizada por algún rol.

### Artefactos

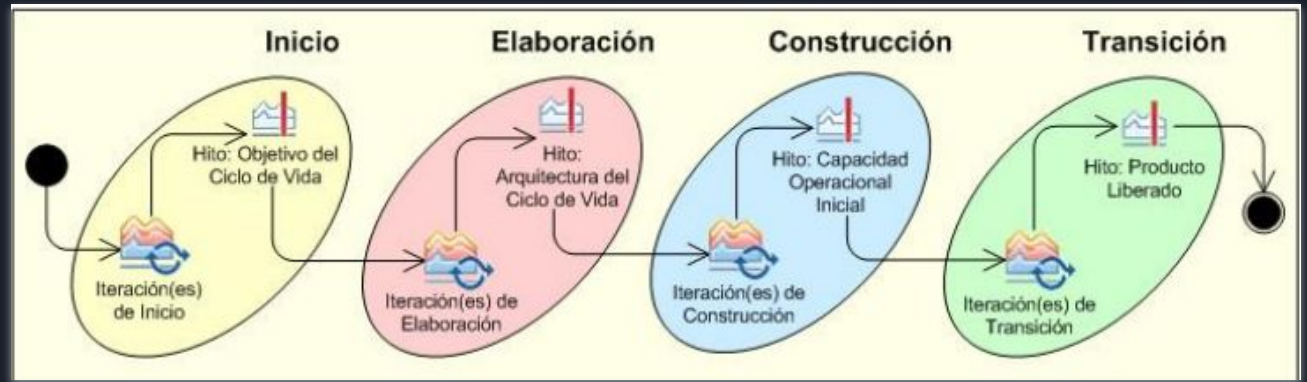
- Un artefacto se considera a todo aquello que una tarea necesita para realizar su función, o bien la produce o modifica.

### Procesos

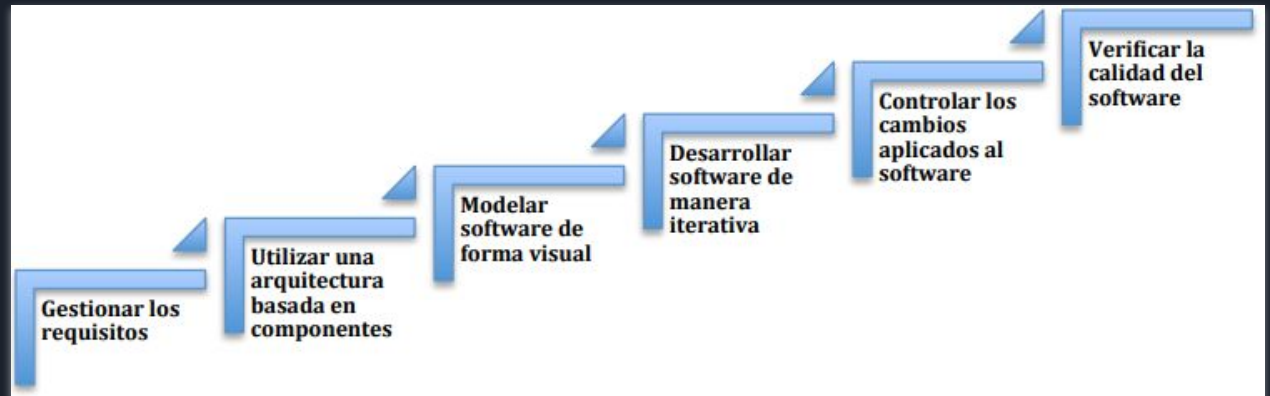
- Los procesos toman los elementos metodológicos y los relacionan entre sí dentro de secuencias temporales que satisfacen las necesidades de distintos tipos de proyecto.



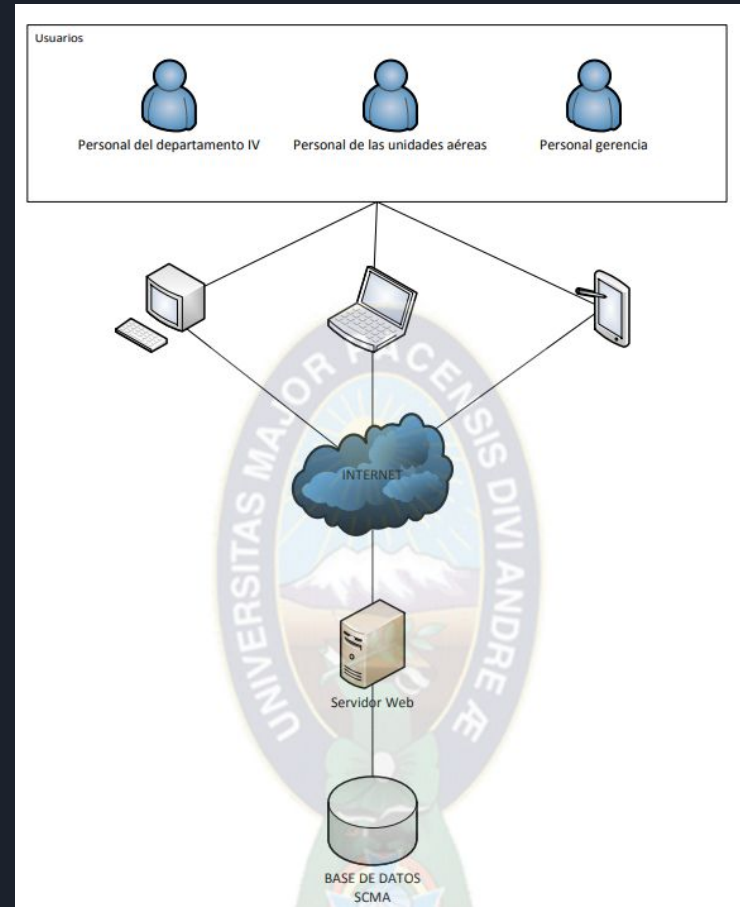
## Ciclo de Vida:



## Prácticas:



# Ejemplo de Modelo Arquitectónico: SISTEMA DE SEGUIMIENTO Y CONTROL PARA COMPONENTES DE AERONAVES.



# Design Thinking:

El Design Thinking es una metodología utilizada por los diseñadores para resolver problemas complejos que, en vez de centrarse en éstos, se orienta a la acción, propiciando el avance hacia la creación del escenario de futuro preferido.



Design thinking utiliza la sensibilidad del diseñador y se basa en la lógica, la imaginación, la intuición y el razonamiento sistémico para explorar las posibilidades de lo que podría ser y para crear los resultados deseados que benefician al usuario final.



## *Características:*

1. **Está centrado en las personas:** sitúa al ser humano (antes que al consumidor) en el centro de todo el proceso.
2. **Consciente del proceso:** se debe tener en claro el proceso de diseño y saber qué métodos se utilizan en cada fase del Design Thinking.
3. **Muéstralo, no lo digas:** se comunica el proceso de ideación de manera visual, llamando la atención de las personas involucradas.
4. **Colaboración radical:** es fundamental que el equipo creativo tenga en sus filas personas de diferentes disciplinas y puntos de vista.
5. **Cultura de la experimentación:** contar con prototipos es fundamental para visualizar la solución al problema.
6. **Incitación a la acción:** de la idea a la acción.



# *Herramientas:*

## *Herramientas para Entender:*

- Creación de “Persona”
- Sistema Gancho
- Crear nuevos usuarios

## *Herramientas para Observar:*

- Mapa de Empatía
- Método AEIOU

## *Herramientas para Definir:*

- Las nueve ventanas
- Mapa de margarita

## *Herramientas para encontrar ideas:*

- Lluvia de Ideas
- SCAMPER

## *Herramientas para desarrollar y probar prototipos:*

- Testeo A/B
- Cuadro de Experimentación

# PROCESO DE DESIGN THINKING

