

# 1er Parcial

14/Feb/2024

## ¿Qué es ingeniería de software?

Son los estándares, metodologías, procesos usados para resolver problemas ó necesidades.

## ¿Qué es un buen software?

Es accesible, intuitivo, qué sirva, se le de mantenimiento, escalable y sostenible.

## Metodologías

- Cascada
  - Agil
  - Espiral
  - SCRUM
  - Kanban
- 

15/Feb/2024

## Evaluación

### 1er periodo

- 60% Evaluación
- 40% Proyecto

### 2do periodo

- 50% Evaluación
- 50% Proyecto

## 3er periodo

- 40% Evaluación
- 60% Proyecto

Solicita un correo con nombre y apellidos

10 minutos tarde es falta

---

## 19/Feb/2024 - C1: Fundamentos

### Ingeniería

Estudio y aplicación de conocimientos científicos y tecnológicos para poder diseñar y crear algo que atienda a la solución de un problema social o una necesidad humana.

### Software

Conjunto de programas, instrucciones y reglas informáticas que nos van a permitir ejecutar distintas tareas en una computadora.

### Ingeniería de software

Es una disciplina formada por un conjunto de métodos, procesos y técnicas para desarrollar una solución, que atienda una necesidad humana o un problema social *en un tiempo estimado y con alta calidad*.

### Disciplina

Conjunto de reglas o normas, cuyo complemento de manera constante nos va a conducir a lograr nuestros objetivos

### Conceptos básicos

#### Método

Conjunto de estrategias y herramientas para lograr el desarrollo de un producto de software.

## Modelo

Es la representación abstracta de los procesos que tenemos que hacer para desarrollar nuestro producto de software, esta representación nos va a ayudar a entender, a explicar y describir nuestro proyecto, de hecho es el mejor medio de comunicación entre el humano y el sistema.

## Metodología (Modelo de procesos)

Es un conjunto de métodos y técnicas que nos van a ayudar a optimizar las actividades en un proyecto de software, esta metodología nos va a ayudar a gestionar sistemáticamente el desarrollo de un producto de software, ayudar a dividirlo, ayudarnos a asignar tareas y actividades, decir los pasos que tenemos que seguir y cuando y como hay que hacerlos.

## Etapas - Fase

Conjunto de actividades y procesos para lograr un objetivo, en este caso para desarrollar un producto de software.

## Case (computer Aided Software Engineering)

Ingeniería de software asistida por computadora, la cual nos va a ayudar en algunas etapas del desarrollo de software, incluso puede hacer herramientas que nos ayuden en todo el desarrollo

## Estandarización

Es un proceso que nos va a ayudar a validar los métodos en el objetivo de la productividad y de atender a las características de calidad necesarias, también nos va a ayudar a adaptar y ajustar nuestro producto a un modelo o norma.

## Mejora continua

El uso de resultados previamente obtenidos para buscar una mejora

## Planificación

Atiende a definir metas y objetivos y a la definición de estrategias y procedimientos para lograrlos.

## **KPA's (Key Process Area)**

Areas de proceso clave, estas van a ser una guía para los desarrolladores para que sepan que criterios tienen que cumplir y obtener un nivel de madurez.

## **Nivel de madurez**

Es el cumplimiento de criterios y procesos en los niveles de madurez, de acuerdo a las normas, sus niveles van de 0 a 5, procedimientos, formas y técnicas a la hora de desarrollar nuestro producto de software.

## **Atributos de software**

### **Rendimientos**

Numero de transacciones en almacenamiento, servidor y con no malgastar recursos.

### **Eficiencia**

Que cumpla con las especificaciones, velocidad de respuesta y tiempo de ejecución

### **Mantenibilidad**

Cambios que pueda tener el sistema, innovaciones y actualizaciones, pero sobre todo a los cambios y exigencias del cliente

### **Confiabilidad**

Un sistema que pueda confiar en el que siempre este disponible

### **Seguridad**

El manejo de la información que se le dará al usuario, y la protección de esos datos

## Usabilidad

Generar interfaces de usuario fáciles, intuitivo.

## Portabilidad

Que pueda usarse en cualquier dispositivo

---

**22/Feb/2024**

## Roles en el desarrollo de software

Es importante definir los roles para tener en cuenta de quienes realizan acciones, como estructurar el proyecto y lo más importante, evitar conflictos en el quipo de trabajo

### El cliente

- Dice cual es el problema
- Cual es el resultado que espera
- Cuales son los requerimientos
- Participa en las pruebas
- Paga las cuentas

### Diseñador grafico

- Encargado de diseñar las interfaces de usuario (intuitivas, amigables)
- Diseña prototipos rápidos para estar informando al cliente
- Puede participar en el diseño de la arquitectura del sistema
- Va a velar porque el producto final sea de acuerdo al diseño establecido

### Documentador

- A cargo de documentar todo lo que tiene que ver con el desarrollo de producto de software

- Lo va a mantener actualizado, va a documentar los cambios, y en dado caso de eliminar algo tiene que sustentarlo en la documentación

## Analista

- Responsable de entender las peticiones del cliente
- Va a analizar el problema
- Va a hacer una propuesta de solución
- Va a redactar las especificaciones
- Va a hacer la elicitación (proceso mediante el cual se van a descubrir las características y propiedades de un sistema a partir de la comunicación con el cliente y las personas interesadas) ó levantamiento de requerimientos

## Programador

- Va a convertir todas las especificaciones y modelos en código fuente en uno o varios lenguajes, apoyándose de herramientas case

## Tester

- Planificar y diseñar de manera sistemática las pruebas con el objetivo de buscar y eliminar errores, además de verificar la calidad del producto.

## Líder

- Conoce, coordina y motiva a su equipo.
- Es el que tiene la máxima experiencia, por lo tanto, ofrece soluciones rápidas y efectivas.
- Es creativo, está pendiente de las actividades de su equipo
- Está pendiente de los avances del proyecto
- Genera informes
- Mantiene el proyecto en el tiempo y presupuesto estimado
- Es puntual, *pone el ejemplo*.
- Si algo falla, el pone la cara o renegocia con el cliente (Tiempos, presupuesto, cambio de sistema)

## Capacitador

En varias ocasiones cuando se entrega el producto, es necesario capacitar al personal, para que sepan como el sistema soluciona sus problemas, en ocasiones, el administrador del sistema, necesita saber sobre algunas partes del back-end para poderle dar mantenimiento o realizar ajustes.

## **Dominios de aplicación de software**

### **Software de sistemas**

Conjunto de programas desarrollados para darle servicio a otros programas, su principal característica es que tiene gran interacción con el hardware y con interfaz de usuario Drivers, librerías, sistemas operativos, gestores de bases de datos, software de comunicaciones, software para redes, etc.

### **Software de aplicación**

Es un conjunto de programas informáticos que son diseñados para ayudar al usuario a realizar determinadas tareas (office 360, photoshop, exploradores, motores de búsqueda, antivirus, erramientas case, entre otros)

### **Software de ingeniería y ciencias**

Es un conjunto de programas que van a realizar uno o varios algoritmos:

- Simuladores en tiempo real
- Nanotecnología
- Estudio y aplicación de energía nuclear
- Fabricación y ensamble automatizado
- Proyectos e investigaciones

### **Software incrustado**

Conjunto de programas que van a realizar funciones limitadas, su principal característica es que se van a encontrar dentro de un tablero, carcasa, consola o armazón

### **Software de línea de productos**

Estos van a atender a un grupo específico de usuarios, por funciones particulares específicas.

## Aplicaciones Web

Aplicaciones como cualquier otra sin embargo tienen la ventaja de estar almacenadas y ejecutadas en la nube

## Software de inteligencia artificial

Conjunto de programas informáticos, que van a realizar uno o varios algoritmos y/o operaciones matemáticas de alto nivel, además de utilizar la inteligencia artificial.

Es un conjunto de métodos matemáticos, programación y tecnología que tienen como objetivo emular el comportamiento y pensamiento humano.

## Computación en un mundo abierto

### Dispositivos móviles

Son dispositivos portátiles que van a tener una comunicación bidireccional a través de una red WiFi (Tablets, laptops, smartphones, relojes, anillos)

## Construcción de redes

Código abierto en el que podemos contribuir a algo más grande

---

**23/Feb/2024**

Apuntes en cuaderno

---

**26/Feb/2024**

## Modelo lineal secuencial

### Sencillo



## Definición de requerimientos

Que sepamos que quiere el cliente

## Fácil gestión

Sabemos que pasara de una etapa a otra y porque ya tenemos requerimientos desde el inicio

## La secuencia no es común

No serán eficientes porque puede haber cambios, no serán lineales

## Entrega en las etapas finales

Equipos de desarrollo, no haran nada hasta esperar a los equipos que desarrollan modelos

## Estados de bloqueo

Equipos de desarrollo, no haran nada hasta esperar a los equipos que desarrollan modelos

## Costoso

## Modelo Cascada Lineal/ secuencial

Solicita requerimientos desde un inicio, es para sistemas pequeños

A diferencia del lineal secuencial, este va a pedir

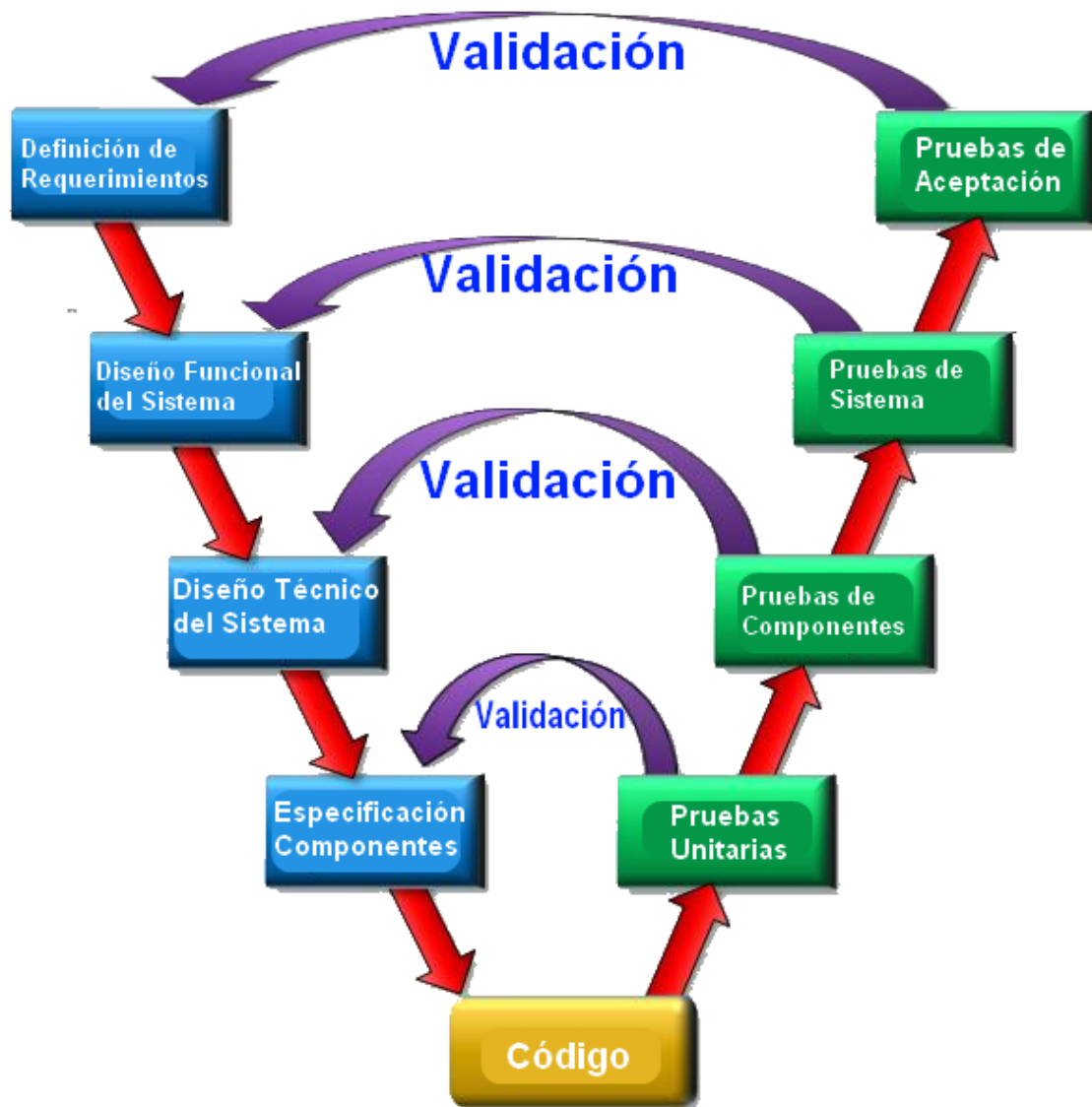
-Permite regresar una y otravez para realic+zar la

## Modelo V

Este modelo a pesar de su forma sigue siendo lineal secuencial, sin embargo, cuando llegue a la etapa de codigo, va a ejecutanuna serie de prueba para modificar los sistemas generados, este sistema trdtr ptoblrms ----- que requie

- Sistemas medianos
- Pruebas
- Mayor corrección

- Confiabilidad
- cantidad



## Modelo incremental

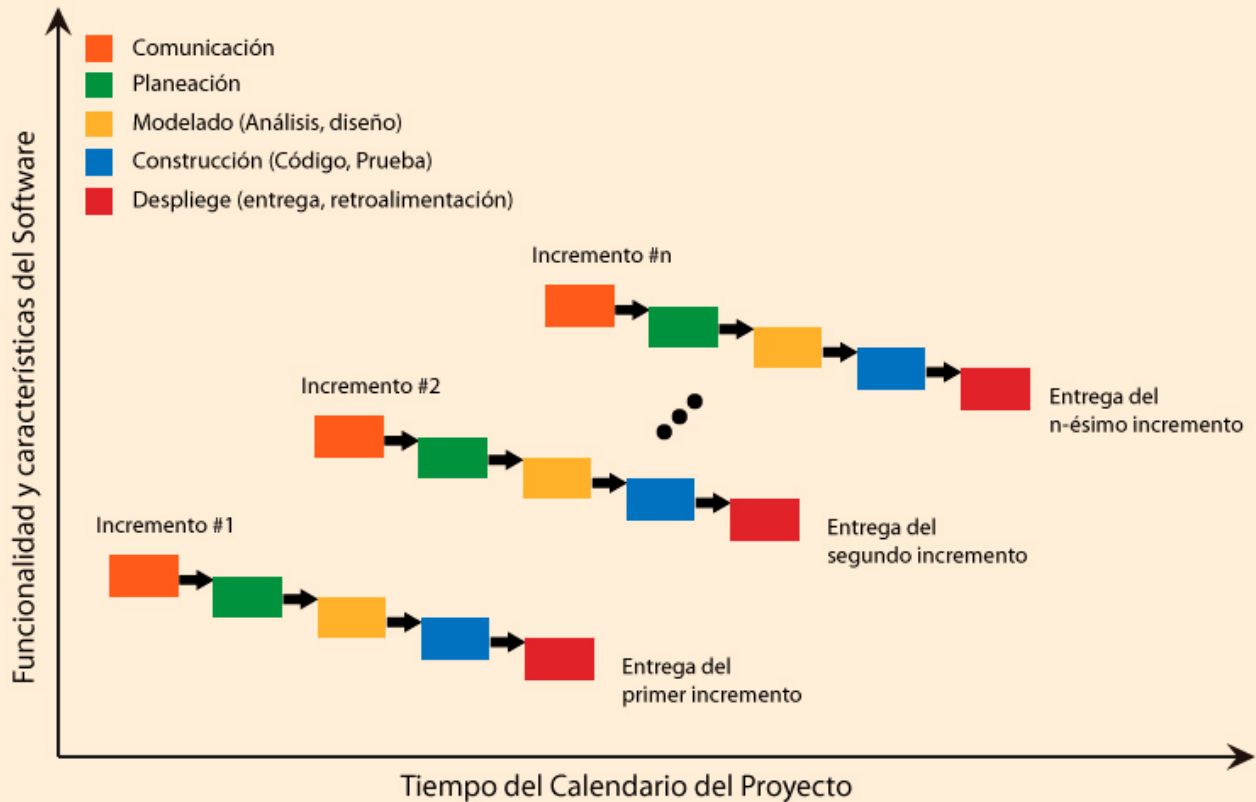
Este modelo se va a desarrollar por medio de secuencias lineales de forma escalonada, cada secuencia lineal nos va a generar un incremento que puede o no entregarse, el primer incremento va a llevar como nombre producto fundamental, a partir de este, el cliente lo usa, lo evalúa, y genera una retroalimentación, con ello se clasifica la siguiente secuencia lineal para generar otra parte del sistema.

En caso de que una secuencia lineal tenga un grave error se puede desechar.

Este modelo nos permite dividir el trabajo en módulos y mantener al cliente informado, en caso de que un componente funcional sea complejo, o requiera de hardware que no se tiene a la

mano, se recurre a generar simuladores

# Modelo Incremental



## Modelo de desarrollo rapido de aplicaciones

Este modelo va a permitir desarrollode software por medio de ciclos y se va a basar en el desarrollo de refinamiento del modelo de datos y afinacion del prototipo, una de sus características principales, es que va a complir

nos va a permitir dividir el proyecto en componentes funcioneles y cada un va a ser dividido por jnksncksdj y se van a apoyar por herramientas case

**Grabación 14min 3:59pm**

## Modelo por prototipos

foto grupo

## Modelo espiral bola de nieve

Va a permitir el desarrollo de software en una serie de entregas evolutivas, este se presenta en una especie de espiral y va a iniciar a funcionar, del centro hacia afuera en el sentido de las manecillas del reloj, la primer vuelta será para definir los requerimientos, para posteriormente hacer los modelos, los prototipos y los componentes funcionales, hasta lograr el sistema completado. cada paso por la etapa de despliegue va a generar un impacto en la etapa de planeación modificando el cronograma de actividades y la estimación de costo

- Proyecto de desarrollo de concepto
- cuando empiece el desarrollo será: proyecto de desarrollo de producto nuevo
- una vez entregado al cliente si ve un falló la metodología lo detectará y será en: proyecto mejora del producto

---

## 29/Feb/2024

### Ámbito

- Contexto: En donde se está desarrollando esto
- Objetivos de información: la información de entrada y de salida
- Función y desempeño: Qué es lo que va a hacer nuestra aplicación

### Recolección de información

- Preguntas de contexto libre: Preguntas para obtener datos de la empresa  
Ejercicio lumen con 5 empresas  
en 10 meses abrir 10 más, se tienen en local y se busca
- Meta-cuestiones:

### Stakeholder

Estas personas son las interesadas en el sistema, usuarios finales y son los entrevistados para las necesidades del sistema

### TFEA (Técnicas para facilitar las especificaciones de aplicación)

- punto de Reunión: Co-working

- Normas
- Agenda flexible: definir los objetivos de la reunión
- Moderador
- Material para trabajar

## ¿Qué es la factibilidad

Es un instrumento que nos va a orientar en la toma de decisiones en la evaluación de un proyecto, para medir sus posibilidades de éxito o fracaso

- Evaluación técnica: Que se necesita?
- Evaluación operativo: Los recursos humanos y la estructura de la empresa, un área de sistemas necesario
- Evaluación ambiental: si hay un ecosistema de por medio
- Evaluación económica: hay un presupuesto? cual es el presupuesto de desarrollo del proyecto? Cual es la rentabilidad del proyecto?
  - Otros gastos: luz internet, viáticos
  - Preparación de datos: gastos de instalación de equipos a un personal
- Evaluación social: Haces un sistema que va a quitar empleados?

Derivado de la técnica, equipo de computo y periféricos, adecuación de instalaciones, instalación de equipos, compra de licencias, derivado de la operativa, cursos de capacitación, modificación de la estructura interna de la empresa, derivado de la etapa de desarrollo, mano de obra, papelería, capacitación, diseño de pruebas, soporte inicial y final, etc.

---

**01/03/2024**

## ¿Qué es un riesgo?

### Riesgo de software

- Tecnología: productos defectuosos
- Requerimientos: Normalmente vienen del origen (De parte del cliente)
- Personal: Conflictos que no puedan desarrollarnos correctamente
- Técnicos:
- Empresariales:

# Componentes y promotores de riesgo

- Rendimiento: estoy cumpliendo con las especificaciones
- Costo: estoy manteniendo el presupuesto indicado
- Apoyo (mantenibilidad): Podrá ser escalable?
- Calendario: Está esto en tiempo y forma?

## Gestión de riesgos

- Identificación de riesgos
- Análisis de riesgos
  - Definir probabilidad: Qué tan probable es que ocurra
  - Definir consecuencia o impacto: Que tanto nos puede impactar
  - Priorizar: De lo más catastrófico a lo más insignificante

## Manejo de riesgos

- Estrategias de prevención (Posibilidad): Prepararme para ello
- Estrategias de minimización (Impacto): Como hago que no me impacte tanto
- Planes de contingencia: Casi rogar por terminarlo, actuar, poner acción

Matriz de probabilidad

Matriz de

## Hoja de formato de riesgo

- Fecha
- Probabilidad
- Impacto
- Descripción
- Tipo de riesgo
- Refinamiento y contexto
  - Subcomisión
- Mitigación y monitoreo
  - Evitar 1...N
- Manejo / Plan de contingencia

- Fecha y estado actual
  - Originador asignado
- 

**07/03/2024**

## **Diapositiva pendiente**

## **Cronograma de actividades**

### **¿Qué es?**

keep

### **Pasos para su elaboración**

- Lista de actividades
- Definir los tiempos
- Evaluar y asignar relaciones de dependencia
- Transmitir lo más importante
- Monitoreo

## **Imagenes Plantilla de escala de tiempo para excel**

Día 0 = 24 de febrero

milestone solo se le dedica un día, se representa con un rombo

![[[ESCOM/24-2/Software Engineer/#^Table]]

**14/03/2024**

## **Métricas**

Estas métricas nos van a ayudar a entender los datos cualitativos del diseño, de lo procesos técnicos y de construcción de un producto de software

### **Características de la métrica**

- Simple y calculable
- Congruente y objetiva
- Efectivo en la retroalimentación a la calidad

1979 Allan J Al-brecht "Creo que necesitamos cobrar por lo que hacemos, en base a eso cobrar"

1986 IFPU Apoya la idea de Allan, lanzan el primer modelo de estimación (Análisis de puntos de función)

## Análisis de puntos de función

Especificaciones funcionales (Interacción y almacenamiento)

Valores de dominio de información

- Num de entradas externas (Interfaces de capturas de datos, QR, Código de barras, VOz)
- Num de salidas externas (Ofrecen información al usuario, mensajes de ayuda, error)
- Num de consultas externas (Flujo de datos bidireccional)
- NUm de archivos logicos internos (Tablas de datos)
- Num de archivos de interfaz internos (Se encuentran en otra entidad, tabla de datos)

Factor ponderado: Simple/Promedio/Complejo

### *Agregar tablas*

Conceptos a recordar de BD

**Fichero:** Un conjunto de datos relacionados que se almacenan juntos. En el contexto de bases de datos, un fichero se suele llamar **tabla**.

**Campo:** La unidad más pequeña de información en una base de datos. Un campo es similar a una columna en una tabla. Cada campo tiene un nombre y un tipo de dato, como texto, número o fecha.

**Entidad:** Una persona, lugar, cosa o evento sobre el que se almacena información en una base de datos. En una tabla, cada fila representa una entidad.

**Atributo:** Una propiedad o característica de una entidad. Un atributo es similar a un campo, pero se usa más específicamente para describir las características de una entidad.

**Registro lógico:** Una colección de campos que representan una sola entidad. Un registro lógico es similar a una fila en una tabla.

Tabla: Puntos de función sin ajustar



**15/03/2024**

Tabla Modos de desarrollo

si tenemos más de un ME (multiplicador de esfuerzo se multiplicaran entre sí)

---

**22/03/2024**

## **COCOMO II**

Factores de escala

precedencia (PREC)

Flexibilidad de desarrollo (FLEX)

Resolución de arquitectura/Riesgo (RESL)

Cohesión de equipo (TEAM)

Madurez del proceso (PMAT)

Tabla: Valores de factor de error

Formulas: Factores de escala

En caso de no detectar un factor de escala, usar Nominal

LDC 30,000

Salario \$16000

PREC nominal= 3.72

FLEX nominal=3.04

Team very low=5.48

PMAT nominal=4.68

RESL nominal =4.24

$\Sigma sf = 21.16$

$E = B + 0.01 \times \Sigma sf$

$E = 0.91 + 0.01 \times 21.16$

$E = 1.12$

$$PM = A \times \text{size}^e$$

$$PM = 2.94 \times 30^{1.12} = 132.65$$

$$F = D + 0.2 \times 0.1 \times \Sigma sf$$

$$F = .28 + 0.2 \times 0.1 \times \Sigma sf = 0.32$$

$$TDEV = C \times (PM)^F$$

$$TDEV = 3.67 \times (132.65)^{.32} = 16.50$$

$$\text{Costo} = PM \times \text{Salario} = 2,122,400$$


---

**25/03/2024**

LDC 100,000

Flex Very Low 5.07

PMAT CMM muy bueno 1.56

PREX alta experiencia 0.74

RUSE very high 1.15

$$\Sigma sf = \text{PREC} + \text{FLEX} + \text{RESL} + \text{TEAM} + \text{PMAT}$$

$$\Sigma sf = (3.72) + 5.07 + (4.24) + (3.29) + 1.56$$

$$\Sigma sf = 17.88$$

$$EM = \text{PREX } RUSE$$

$$EM = 0.74 \times 1.15$$

$$EM = 0.85$$

$$E = B + 0.01 \Sigma sf$$

$$E = 0.91 + 0.01 \times 17.88$$

$$E = 1.08$$

$$PMns = A \times \text{SIZE}^E \times EM$$

$$PMns = 2.94 \times 100^{17.88} \times 0.85$$

$$PMns = 361.21$$

$$F = D + 0.2 \times 0.01 \Sigma sf$$

$$F = 0.28 + 0.2 \times 0.01 \times 17.88$$

$$F = 0.31$$

$$TDEV = C \times (PMns)^F$$

$$TDEV = 3.67 \times (361.21)^{0.31}$$

$$TDEV = 22.78$$

COSTO= PM(salario)

COSTO= 361.21 \* 16,000

COSTO= 5,779,360

---

---

**22/04/2024**

¿Cuáles son los problemas con los procesos actuales y como ayudaría su problema a resolverlos?

¿Cual es la contribución directa que hará el sistema a los objetivos del negocio?

¿La información se puede obtener y transferir a otros sistemas de la organización?

¿El sistema requiere de tecnología que no se ha utilizado previamente en la organización?

¿A que debe ayudar, a quien debe ayudar y quien necesita el sistema?

## **Analisis de requerimientos**

Tecnicas y herramientas

- Modelos de casos de usos
- Diagrama de flujo de datos
- Prototipado
- Entrevistas y cuestionarios
- Requerimientos funcionales y no funcionales
- Validación (inconsistencias)

## **Técnicas de validación**

- Reviews o walk-Thorughs
- Auditorias
- Matrices de trazabilidad
- Prototipos
- Diseño de prueba

# Gestión de requisitos

- Identificación
- Analisis
- Documentación
- Valoración
- Seguimiento y control

## Actividad

Mida temperatura

Mida humedad

Mida luz

Muestre temperatura

Muestre humedad

Muestre luz

Informe errores

---

**25/04/2024**

## Componentes principales del sistema lógico

- Interfaz de usuario
- Lógica del negocio
- Acceso de datos
- Servicios de integración
- Seguridad
- Gestión de errores
- Monitoreo y gestión de rendimiento

## Diseño detallado

- Estructura interna del sistema
- Herramientas
- Modularidad, reusabilidad, escalabilidad

## Diseño de módulos

- Cohesión y acoplamiento

Modulo: un componente y cohesivo del sistema el cual puede ser desarrollado y probado de manera individual y este nos va a ayudar a simplificar y gestionar la ejecución del sistema

Cohesión: grado en que las funciones de un modulo se relacionan y trabajan entre sí para lograr un objetivo

\_: Alta cohesión y bajo acoplamiento \_\_\_\_

## Diseño de estructura de datos

- Mejora la eficiencia del sistema y el mantenimiento
- Se debe adaptar a requerimientos funcionales y no funcionales

## Técnicas de diseño

- Modelado de datos
- Especificación de tipos abstractos de datos

## Base de datos

- Almacenar, recuperar, actualizar o eliminar información

## Modelo de datos


- Conceptual(de dominio):Atributos, clases, características, relaciones, sublimitaciones, seguridad e integridad de los datos
- Lógico de datos: Dará más detalles sobre conceptos y relaciones del dominio, tipos de datos, longitud y relación entre entidades
- Datos Físicos: Representación de como se almacenan los datos físicamente en la base de datos (tablas asociativas, relaciones)

## Modelado de datos

- **Jerárquicos:**

se caracteriza por presentar los datos en una estructura de árbol invertido, donde cada registro tiene un único nodo raíz, del que surgen otros nodos (registros); los nodos en un mismo nivel son nodos padre, cada nodo padre tiene el mismo nodo raíz, y puede tener nodos hijos, pero los nodos hijos solo pueden tener un nodo padre. Este modelo se emplea poco actualmente. En este modelo, los registros de un mismo nivel se clasifican en un orden específico.

- **Relacionales:**

El **modelo de base de datos relacional** es uno de los más comunes. Este modelo es el que emplean las [bases de datos relacionales](#)  y ordena los datos en tablas (relaciones) compuestas por columnas y filas. Cada columna alberga un atributo de la entidad (nombre, dirección, fecha de nacimiento...); a los atributos de una relación se los llama dominio. Escogiendo un atributo en concreto o una combinación de varios tenemos una clave primaria, a la que se puede hacer referencia en otras tablas, en las que será un clave externa. En cada fila (tupla) se incluyen datos sobre una instancia específica de la entidad (por ejemplo, un cliente específico).

- **No relacionales:**

Una base de **datos no relacional**, también conocida como base de **datos** NoSQL (Not Only SQL), es un tipo de sistema de gestión de bases de **datos** diseñado para manejar y almacenar **datos** de manera flexible y escalable, sin seguir el **modelo** de tablas y relaciones de las bases de **datos relacionales** tradicionales.

- **Orientado a objetos:**

define la base de datos como una colección de objetos utilizados en la programación orientada a objetos (es decir, que emplear lenguajes como C++ o Java, por ejemplo). Este modelo de base de datos utiliza tablas también, pero no solo se limita a ellas y permite almacenar información muy detallada sobre cada objeto.

## Sistemas gestores de base de datos (SGBD)

Paga | Libre |  
Oracle | BASE (OpenOffice)  
SQL Server | PostgreSQL

## Selección de herramientas de desarrollo

- Mejora la eficiencia y calidad

## Evaluar y comparar

- Experiencia
  - Funcionalidad
  - Capacidad
  - Escalabilidad
  - Integración
- 

**29/04/2024**

## UI

- ¿Quién se encarga?: Diseñador grafico
- Comunicación
- Diseño amable y eficaz
- Percepción del usuario

## Diseño de interfaz

- Requerimientos
- Tareas
- Ambiente
- Escenarios
- Herramientas: Figma/Sketch/AdobdeXD
- Calidad
- Validación

## Validación

- Capacidad
- Grado de usabilidad: que tan facil es acceder a ella y navegar en ella
- Herramienta útil: Expresso., katalon, selenio IDE, Uiautomator, Parasoft Selenic, Mabi, Testim
- Entender el problema
- Usuarios Finales
- Tareas de los usuarios finales
- Contenido que presenta

- Ambiente

¿Qué querría el usuario que haga el sistema?

¿Donde se encontrara físicamente el sistema?

El usuario estara sentado, de pies, o haciendo otras tareas no relacionadas con la interfaz

Interrupciones, de espacio, iluminación o ruido

Hay consideraciones especiales de factores humanos generadas por los factores ambientales?

¿Cómo encajaría el sistema en el flujo del trabajo o en las actividades diarias?

¿Cuan competente es el usuario técnicamente y que sistemas ya utiliza?

¿Qué estilos de aspecto y comportamientos son los requeridos del usuario?

## Centrarse en el usuario: Tipos de interfaz

- Lenguaje natural: Lenguaje transacción, lenguaje de acción
- Preguntas y respuestas: El sistema hace preguntas por medio del monitor o sonido al usuario, y el usuario responde por teclado o voz
- Grafica de usuario (GUI): Está formada por elementos gráficos para poder interactuar con el usuario
- Basada en texto: Linea de comandos usada en diferentes So y las aplicaciones de terminal
- Realidad virtual: Nos ofrecen una experiencia envolvente e inmersiva, usada en el arte, videojuegos
- Realidad aumentada: sobreponer elementos en el entorno para poder interactuar
- Tangible: Pantallas táctiles, teclado, mouse, controles de juego
- Por voz: dar instrucciones al sistema por medio de voz

## Características

- Usabilidad
- Claridad y consistencia
- Retroalimentación: diga de una manera visual o auditiva lo que esta pasando
- Flexibilidad y eficiencia:
- Diseño visual atractivo
- Accesibilidad
- Orientación y ayuda

## Limitaciones de una interfaz



- Curva de aprendizaje
- Accesibilidad
- Dependencia de habilidades técnicas
- Hardware o dispositivos
- Complejidad de sistema o aplicación
- Cambio de diseño o actualizaciones

## Importante

- No preocupan
- Primera impresión
- El usuario percibe opciones y funciones rápidamente
- Son atractivas

## Principios generales

- Previsión: Intuye tu siguiente movimiento
- Comunicación: Recordar qué esta haciendo
- Consistencia: Comportamiento y funciones
- Eficiencia: Qué solucione el problema
- Reducción de latencia: Reducir el tiempo de espera y hacer diferentes actividades al mismo tiempo
- Aprendizaje: Tiene que ver con la usabilidad
- Integridad: al tener apagones guardar los datos continuamente
- Legibilidad: Legible para jóvenes y adultos mayores
- Seguimiento de estado: "indica donde te quedaste en el archivo"

## Recomendaciones

- Cantidad de texto
- En construcción
- Evitar el desplazamiento
- Menús de navegación y encabezados
- Que la estética no estorbe
- Navegación Obvia
- No necesidad de ser espectacular

## Diseño de la Interfaz

- Inspección de la usabilidad
  - Pruebas de usabilidad
  - Mantenimiento de software
  - Diseño debe responder
    - ¿Dónde estoy?
    - ¿Qué puedo hacer?
    - ¿Dónde he estado?
    - Hacia donde voy?
- Si responde eso la navegación, la navegación es buena

## Diseño responsivo

- Flexibilidad
- Escalabilidad
- Optimización para móviles
- Diseño centrado en el usuario

## Accesibilidad

- Alternativas: elementos multimedia, descripción de imágenes, un narrador, utilizar subtítulos
- Controles
- Contraste: que se pueda regular
- Navegación: Atender la parte cognitiva

## Diseño de interacción

- Flujo de trabajo
- Feedback
- Usabilidad

## Diseño de información

- Jerarquía visual

- Agrupación

## Diseño visual

- Paleta de colores
  - Tipografía
  - Layout
  - Iconos e imágenes
  - Animaciones y transiciones para mejorar la experiencia
- 

**02/05/2024**

## Experiencia de usuario (UX)

- Comprender las necesidades, expectativas y deseos
- Impacto en el usuario
- Objetivo
  - Útiles
  - Agradable
  - Satisfacción
  - Lealtad
- Considera aspectos emocionales
- Difusión
- Generar usuarios asiduos
- Diferencia de la competencia
- Reducir costos - valor agregado

## Proceso del diseño del UX

Investigación -> Análisis de datos -> User persona -> Mapear los flujos de usuario -> Crear wireframes y prototipos -> Conducir las pruebas de los usuarios -> Pasar del diseño al desarrollo

User persona: Personaje ficticio de lo que puede ser un usuario potencial

Mapear: Los caminos que tiene que seguir para alcanzar su meta

Wireframes: Bocetos de como se van a generar

Pruebas: Validar los resultados

## Segun Jesse James Garret

Se puede ver como un iceberg con la profundidad, no detectable por el usuario

### Funcionalidad

- Especificaciones funcionales
- Diseño de la interacción
- Diseño de la interface

### Información

- Requerimientos del contenido
- Arquitectura de la información: Depende el usuario, es lo que podrá ver y gestionar
- Diseño de navegación

### Etapas

- Estrategia: Centrarnos en quienes son los stakeholders, usuarios potenciales, expectativas y necesidades
- Enfoque: Establecer los req funcionales, req de contenido, dar al cliente una posible solución
- Estructura: Checar componentes módulos e interfaces, es aquí donde se define la arquitectura de la información
- Esqueleto: Diseña las interfaces graficas de usuario, la navegación, de que elementos se va a valer para poder valerse
- Superficie: se establece las wireframes de alta calidad listas para el usuario

## Compra y adquisición

Compra memorable  
sentimientos postventa  
Experiencia

- Crea una experiencia intuitiva
- Facilita las compras
- Recompensa la lealtad de tus compradores
- Actualiza tu app

## Sitio Web

- Experiencia
  - Velocidad
  - Capacidad de respuesta
  - Seguridad
  - Contenidos relevantes bien diseñados
  - Customer Journey (Consciencia, Consideración, Compra, Retención, Recomendaciones)
  - Audiencia
- 

## 06/05/2024 Herramientas CASE

### Computer - Aided Software Engineering

#### Actividades:

- Desarrollo de modelos graficos
- Diccionario de datos
- Generación de interfaces
- Depuración de programas
- Conversión de programas

#### Algunas limitaciones:

- No permite el analisis
- No ayuda a la interacción
- Genera demasiado codigo

## Objetivos

- Aplicación practica
- Mejor productividad del equió
- Reducir costos
- Calidad
- Prototipos
- Simplifica y estandariza procedimientos
- Portabilidad y reutilización

## Categorías

- Herramientas de gestion
- Herramientas tecnicas
  - Upper CASE o CASE frontal: Hace alusión al front end, ayuda en etapas iniciales del diseño del sistema, pantallas, BD, modelado
  - Lower CASE o CASE Dorsal: Backend, diseño detallado del sistema, analisis y creación de base de datos, desarrollo de producto, pruebas y mantenimiento
- Herramientas de soporte: Apoya en diseño y gestion del sistema, algunas pruebas y seguridad

## Tipo de herramientas

- ICASE (Integrated CASE)
- IPSE (Integrated Programming Support Enviroment)
- Toolkits: Usadas en BD de Oracle y en la parte de la integración de los clientes
- CAST (Computer-Aided Software Testing)
- MetaCASE

*Va a Mandar fotos por classroom, agregar tablas*

---

## 09/05/2024 Ingeniería Web

### ¿Qué es?

CERN: Organización europea para la investigación nuclear

1988 - Timothy Berners Lee, sugirió al Cern liberar internet para compartir conocimientos y recursos.

1990: Primer servidor y navegador

1991: Primer servidor stanford, California

30 de Abril 1993, se libero WWW, A finales de ese año: 500 servidores.

1994: 10,000 servidores y 10 millones de usuarios

## 3 Conceptos

- HTML
- HTTP
- URL

*Imagen de metodología en espiral*

## Calidad de una aplicación Web

-

Servicios de ayuda y retroalimentación en línea

Capacidades estéticas de interfaz

Servicios especiales

Capacidad de recuperación y de búsqueda

Servicios de búsqueda y navegación

Servicios relacionados con el dominio de aplicación

proceso correcto de enlace

Recuperación de errores

Validación y recuperación de la entrada del usuario

Rendimiento del tiempo de respuesta

Velocidad de generación de páginas

velocidad de generación de gráficos

Facilidad de corrección

Adaptabilidad

Extensibilidad

## Equipo de ingeniería web

- Administradores

- Ingeniero Web
- Especialista de soporte
- Web designer
- Arquitecto cloud
- DBA
- Marketing digital
- Creador de contenido
- Copywriter
- Ilustradores
- Fotografos
- Project Manager

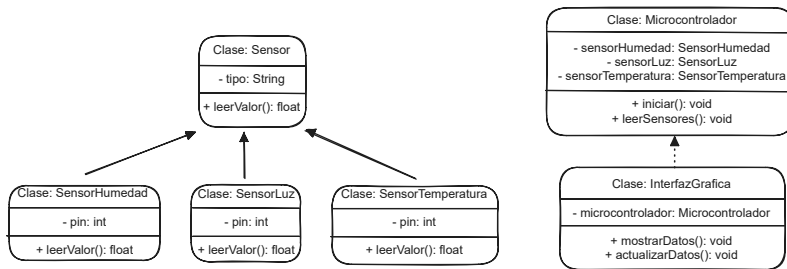
## Proceso de pruebas

1. **Prueba de contenido:** Verifica que el contenido (textos, imágenes, videos, etc.) presente en la aplicación web sea preciso, relevante y esté correctamente formateado.
2. **Prueba de interfaz:** Evalúa la usabilidad y la experiencia del usuario al interactuar con la interfaz gráfica de la aplicación. Se busca asegurar que los elementos visuales estén bien organizados, sean intuitivos y respondan adecuadamente a las acciones del usuario.
3. **Prueba de Navegación:** Comprueba que la navegación dentro de la aplicación web sea fluida y coherente. Se verifica que los enlaces, botones y menús conduzcan correctamente a las diferentes secciones y funcionalidades.
4. **Prueba al nivel de componentes:** Se enfoca en evaluar cada uno de los componentes individuales que conforman la aplicación web, como formularios, widgets, módulos, etc. Se busca identificar posibles errores o comportamientos inesperados.
5. **Prueba de configuración:** Asegura que la aplicación web funcione correctamente con diferentes configuraciones, como diferentes navegadores, sistemas operativos, resoluciones de pantalla, etc.
6. **Prueba de seguridad:** Evalúa la robustez de la aplicación frente a posibles vulnerabilidades y ataques. Se verifica que los mecanismos de autenticación, autorización y cifrado estén implementados correctamente.
7. **Prueba de desempeño:** Mide el rendimiento y la capacidad de respuesta de la aplicación web bajo diferentes cargas de trabajo. Se busca identificar cuellos de botella, tiempos de respuesta lentos o problemas de escalabilidad.

---

## Diagrama de clases para segundo Milestone






---

**16/05/2024**

Meta de una empresa: Reducir costos, tiempos, fortalecer al personal

Ordenar por importancia los procesos que nos ayudan a lograr nuestros objetivos

Especificación y diseño de procesos: Modificar, eliminar procesos

Prototipo: pruebas para ver si me esta llevando a un nivel de calidad

Refinamiento y ejemplificación: Lo mismo que explica y ponerlo en marcha, hacerlo n veces

## Modelo de procesos de Reingenieria de software

Reestructuración: Actualización o Re modernización de las tareas, dentro del proceso ed una reingenieria

Ingeniería inversa: proceso mediante el cual se \_\_\_\_ de un sist, disp o herramienta, con el fin de crear otro similar, de lo contrario seria plagio

Y el plagio es un delito

---

**23/05/2024**

## Modelos de madurez

MPS: Mejoramiento del proceso de software

**¿Qué es?**

**¿Quien lo impulsa?**

Gerentes técnicos, project manager,

## Pasos

1. Valoración: el MPS valora los procesos actuales de desarrollo, ¿Cómo se hacen actualmente los productos de software?
2. Capacitación: Conocimiento de tecnologías a las que se someterá todo el equipo, incluyendo desarrollo personal y trabajo en equipo
3. Selección y justificación: de Herramientas CASE,
4. Implementación del plan
5. Evaluación y afinación: que tanto me ayudó esta herramienta

## Rendimiento sobre la inversión

Que es lo minimo que espero al certificarme

¿Qué es?

Es un mecanismo para evaluar los procesos de una organización, este nos dará un indicio o una idea de la calidad de nuestros procesos, ademas de darnos el nivel de madurez que manejamos, y para ello establece cuatro indicadores

1. conjunto de características: mínimas e indispensables que debe tener una organización para que sus procesos sean de calidad y efectivos
2. Que características cumple nuestra empresa, que características cumple pero son debiles y que características están ausentes
3. Estrategia para auxiliar a la empresa a fortalecer las características que están debiles e implementar las que están ausentes

Imagen: Marco Conceptual MPS

## Grupo impulsor o de apoyo

- Certificadores de calidad: Tester que se encargue que los diseñadores de software sean de calidad
- Formalistas: Se encargan de optimizar los flujos de trabajo, se van a basar en técnicas de modelado, para modificar, optimizar los procesos
- Defensores de las herramientas: Que todo lo que se desarrolle siempre pase por un análisis
- Profesionales: de acuerdo a la planificación se cumpla el producto de software
- Reformadores: Cambian todo lo que tiene que ver con la empresa, desde estructura hasta cambios físicos, pasando por recursos humanos

- Ideólogos: Son las personas que supervisaran que todo el producto que se realice siempre sea llevado por una tecnología (reusabilidad)

## Se establecen mecanismos para

1. Apoyar:
  2. Determinar el grado de absorción
  3. Medir el grado en que ya se adaptaron ciertos cambios
- 

**30/05/2024**

## TSP (Proceso del equipo de software)

### Características de un equipo

Equipo: grupo de trabajo de dos o más personas que buscan un mismo objetivo

- Cohesión: Unión fuerte entre los miembros del equipo
- Metas: Claras y alcanzables
- Retroalimentación: Al pendiente de los avances del proyecto
- Ambiente de trabajo: Valores respeto, cordialidad, empatía

### Problemas comunes

- Falta de cooperación y compromiso
- Falta de liderazgo
- Falta de confianza
- Revisiones entre colegas inefectivas
- Diferencia de distribución de carga de trabajo

### Estrategias de solución

- Withdrawal
- Adaptarse
- Comprometerse

- Confrontar

## Proceso de software de equipo (TSP)

- ¿Que es? Ayudara a definir los procesos de mediana escala
- ¿Para quien es? PM, lideres de proyecto
- ¿Para que es? Capitalizar el esfuerzo, balancear, proceso, producto y equipo de trabajo, ayudar a negociar los compromisos, mejorar el rendimiento y ayudar a conformar los equipos, generar un buen ambiente de trabajo, decir que hacer en cada etapa y como se conecta cada una de ellas

## Objetivos

- Equipos autodirigidos: se evalúan entre ellos y así mismos, y gestionan entre ellos la calidad y la productividad
- Capacitar a los gerentes:
- Acelerar las mejoras
- Brinda guía
- Maximizar calidad
- Predicción: Evitar defectos, mejorar tiempos de producción y hacer estimaciones de tiempo costo y esfuerzo

## Componentes

Foto

## Fases del ciclo de vida

1. Lanzamiento
2. Estrategia: Estimación y análisis de riesgo
3. Planeación: asignación de roles, cronograma de actividades
4. Requerimientos:
5. Diseño
6. Implementación: , codificación, estandarización, pruebas unitarias
7. Pruebas: Verificar la calidad y rendimiento de nuestro producto
8. Postmortem: Analizar todo lo que hicimos y ver como es posible mejorarlo

# Características

- Equipos autodirigidos
  - Planes detallados
  - Lanzamiento de proyectos de 3.5 días
  - Roles, metas, riesgos
  - Calendarios de equipo y personales
- 

**03/06/2024**

No sé que

## Nivel 2: Irrepetible

KPA's

- Gestión de configuración de software
- Aseguramiento de calidad de software
- Gestión software: depende el fin de la empresa se aceptaran los productos a desarrollar
- Planificación: Estimaciones
- Seguimiento y control
- Gestión de requisitos:

## Nivel 3: Definido

KPA's

- Revisiones
- Coordinación de grupos: Autoevalúan, están al pendiente de los avances
- Gestión integrada de software: En base a los conocimientos e infraestructura, ven si se puede aceptar X proyecto
- Programa de entrenamiento: certificaciones y actualizaciones
- Definir el proceso de la organización
- Enfoque de proceso de la organización

## Nivel 4: Gestionado

La retroalimentación sobre los tiempos y manejos del producto  
KPAS

## Nivel 5: Optimizado / Mejora continua

Aquí es donde se aplican los cambios

KPA's

- Gestion de calidad

Los quito muy rapido

## NIVELES

- 5 Optimizado/ Mejora continua
- 4 Gestionado
- 3 Definido
- 2 Repetible
- 1 Inicial / Caos
- 0 Negligente (Tom Schorsch 96)
- 1 Obstrutivo
- 2 Despreciador: Se contratan personas ajenas a la carrera, con conocimientos no completos
- 3 Socavación: Pésimo ambiente laboral, premian al compadrazgo

## Modelo de Madurez de Capacidad

Protege a sus miembros en apoyo a la mejora continua

- Foto sombrilla\*

---

**06/06/2024**

**Modelo de madurez de capacidad integrado (CMMI)**

- ¿Qué es?
- Objetivos
- ¿Para quién es?  
Metamodelo
- Continuo
- Etapas

## Modelo continuo

ME: Metas específicas (ME): Define las características que deben tener las actividades que contemplan las áreas de proceso para ser efectivas (Como los objetivos generales)

(PE) Practicas específicas: Desglosan una meta en un conjunto de actividades relacionadas con el proceso del desarrollo (Como los objetivos específicos)

(MG) Metas genericas: Cada una de las metas genericas, coinciden con un nivel de capacidad, por lo tanto nos otorgara un nivel de capacidad

Practicas Genericas: en un conjunto de actividades para poder alcanzar una meta

## KPAS: Planificación de proyecto

- (ME) Establecimiento de estimaciones

FOTOS

## Niveles de capacidad

- 5: Optimizado
- 4: Administrado cuantitativamente

---

07/06/2024

**MoProsoft: Modelo de procesos para la industria de software**

¿Qué es?

¿Para que es? Fomenta la estandarización

¿Para quien es? Equipos que lo necesiten

*Ayuda a empresas con modelos no establecidos y las que si los tengan*

Ver que características están débiles

Características:

- Es fácil de entender y aplicar
- No es costoso
- Base para otras evaluaciones

Se rige por la norma NYCE-2016

ISO/IEC 2018

Ingeniería de sistemas y software

*Foto Mapa mental*

planificar de manera estrategica la definición de las propuestas de mejora

Recursos humanos: Gestión de recursos: se seleccionara el personal y se contratara, se dará una inducción,

Bienes, servicios e infraestructura: contacta a proveedores, licencias, papeleria insumos

Conocimiento de la organización: repositorio de conocimientos, dependiendo los conocimientos podemos reutilizarlos

Administración de proyectos especificos: desarrollar proyectos de manera sistematica para su entrega en tiempo en forma con su presupuesto establecido en tiempo y forma

Desarrollo y mantenimiento

Diagrama niveles

0 incompleto a 5 Optimizado