

Unidad 1 - Ingeniería. Fundamentos. Análisis y especificación del problema.
Restricciones. Soluciones: alternativas, comparación. Selección. Métodos y técnicas: científicos, empíricos, experimentales. Modelos de representación.

Problema: Surge del deseo de transformar un estado, forma o condición de las cosas a otro.

Pueden ser problemas cerrados o abiertos.

Problemas Cerrados: tienen UNA solución única.

Problemas Abiertos: varias posibles soluciones.

Restricción: Condición que la solución al problema DEBE cumplir. MUY importante cuestionarlas.

Criterio: Suelen ser comunes. Varía su prioridad. Sus valores pueden oscilar en un rango.

Comparación de soluciones:

| | Criterio 1 | Criterio 2 | ... | Criterio n | Total |
|-------------------|-------------------|-------------------|-----|-------------------|--------------|
| | <i>Peso: p1%</i> | <i>Peso: p2%</i> | | <i>Peso: pn%</i> | <i>100%</i> |
| Solución A | $A1 * p1$ | $A2 * p2$ | | $An * pn$ | $T1$ |
| Solución B | $B1 * p1$ | $B2 * p2$ | | $Bn * pn$ | $T2$ |
| Solución C | $C1 * p1$ | $C2 * p2$ | | $Cn * pn$ | $T3$ |

1. Otorgamos una valoración numérica a cada uno
2. Multiplicamos por el peso del criterio
3. Sumamos el total de impacto de criterios para cada solución

Análisis y especificación del problema:

Formulación – Análisis – Búsqueda de soluciones – Elección – Especificación de la solución.

Formulación: ¿Cuál es el problema? Describirlo. Método de la caja negra.

Análisis: Recopilación de información relevante y detallada sobre el problema y las restricciones. ¿Problema abierto o cerrado? Descomposición del problema. Estado de origen y estado deseado. Restricciones a cumplir y criterios (Técnicos, económicos, de tiempo, estéticos, humanos, de seguridad).

Búsqueda de soluciones: Conjunto de requerimientos, variables (tienen diferentes valores), búsqueda de soluciones parciales y de opciones, imaginación y lluvia de ideas. Analizar variables de una por vez. Combinación de soluciones parciales.

Elección: Criterios de selección → atributos (ver también cuanta relevancia tiene cada uno), impacto de la solución en el atributo, impacto de cada atributo, comparación (relación costo/beneficio), elección.

Especificación: Solución elegida, descripción detallada de todas las características de la solución (evaluación: factibilidad, rentabilidad. Construcción. Controles/Monitoreo), Modelos, Informes.

PROCESO EN ING. DE SOFTWARE: el mismo + IMPLEMENTACION + MANTENIMIENTO

Formulación – Análisis – Búsqueda de soluciones – Elección – Especificación de la solución – Implementación - Mantenimiento

Implementación: Desarrollo de nuevo producto, elaboración de documentos de uso y desarrollo, capacitación a usuarios, puesta en funcionamiento y control de soluciones existentes, certificación y auditoría de sistemas en producción.

Mantenimiento: Modificaciones de la implementación, incorporación de nueva funcionalidad, actualización de funcionalidades existentes, corrección de errores, migración a versiones nuevas de hardware o software, adaptación a nuevos entornos, integración con otros productos.

MEDICIONES:

Se tienen que identificar los siguientes:

Entidad: Objeto o evento del mundo real (a que se mide).

Atributo: Característica o propiedad de una entidad (que se le mide).

Medición: el acto de determinar una medida. Proceso en el cual se asignan números o símbolos a los atributos de entidades del mundo real, de manera tal que se los describa de acuerdo a las reglas definidas.

Medida: Elemento que proporciona un indicio cuantitativo de la extensión, cantidad, tamaño de algún atributo de un producto o proceso.

Métrica: método de medición elegido y la escala de medición. EJ: *Método de medición: Metro / Medidor Laser *Escala: Absoluta.

Indicador: Métrica o combinación de Métricas que proporcionan comprensión acerca del proceso de software o el producto en sí. Proporcionan comprensión para realizar ajustes, para incorporar cambios que permitan hacer mejor las cosas.

ESCALAS:

Escala Nominal: Se definen clases entre las que se ubican las entidades de acuerdo al valor del atributo medido. El sistema de relación empírico consiste solo de diferentes clases, no hay noción de orden entre ellas. Cualquier representación es aceptable.

Defecto en automotor. Atributo: ubicación

• $M1(x) =$

100 mecánico

200 eléctrico

600 chasis



Escala Ordinal: Define clases o categorías. Diferentes clases ordenadas con respecto al atributo. Mientras preserve el ordenamiento es aceptable. Los números solo representan un ranking, no tiene sentido sumar o restar.

Defecto en automotor. Atributo: gravedad

• $M1(x) =$

1 trivial

2 moderado

3 complejo

4 irresoluble



• $M2(x) =$

1 trivial

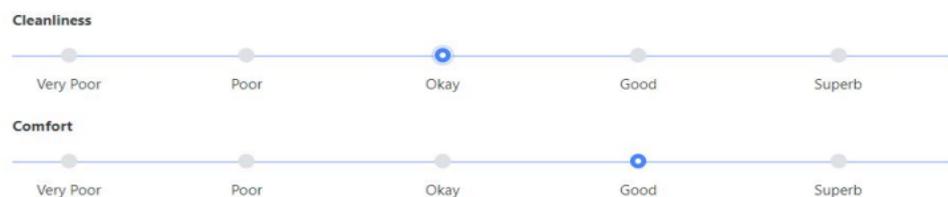
8 moderado

12 complejo

40 irresoluble

Escala de Intervalos: Preserva el orden, preserva diferencias, se puede sumar y restar.

Opinión en
una
encuesta



Escala de proporciones: Da información acerca de diferencia de proporciones existentes entre cada clase. Preserva el orden, existe el elemento cero, operaciones aritmeticas son válidas.

Costo monetario de un producto. Atributo: valor

• $M1(x) =$

Cualquier valor mayor o igual a cero es válido

\$20 es más caro que \$15

\$40 es el doble que \$20



• $M2(x) =$ Valor expresado en otra moneda

$M2(x) = M1(x) * k$

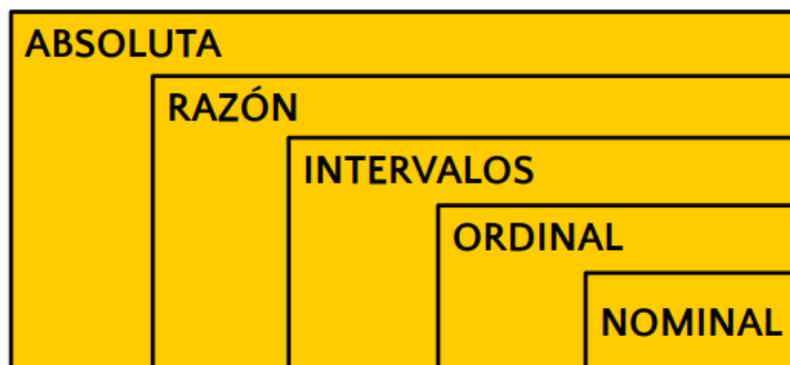
Escala Absoluta: Contar el número de elementos. Operaciones aritméticas son validas.

Monedas en un monedero. Atributo: cantidad

• $M1(x) =$

el número resultante de contar la cantidad de monedas

no importa el valor que simboliza cada moneda



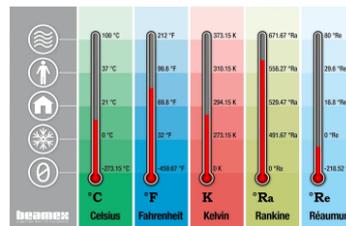
MODELOS DE REPRESENTACIÓN:

Son descripciones de la naturaleza o comportamiento de un objeto real.

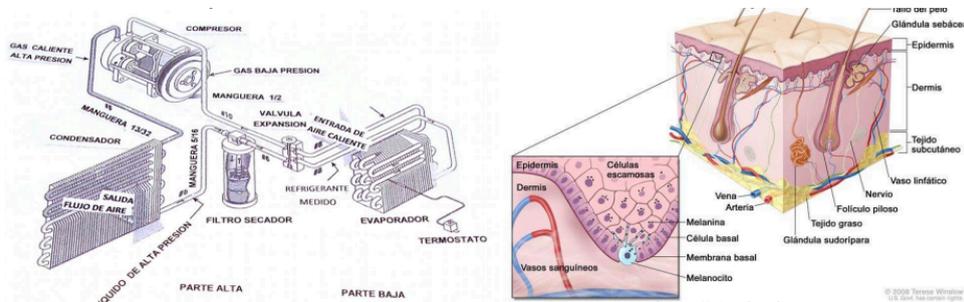
Físicas o icónicas: Semejanza física con objeto real. Representación en dos o tres dimensiones. Conserva proporciones. EJ: Una maqueta de un Cadillac 1930, o de un Avión.



Gráficas: visualizan relaciones y magnitudes relativas. EJ: Gráfico de deuda externa, dolar..



Esquemáticos: Representación simbólica de un objeto real. EJ: Esquema de un velero.



Matemáticos: cálculos de predicción y comunicación. Fórmula para revalorizar pensiones.

$$C_k^n = \frac{n!}{k!(n-k)!}$$

Simulaciones: Experimentación usando una representación de un objeto real.



EJ: Digital: Accidente 3D, video 3D de un tsunami.

EJ: Analógica: experimentación de un accidente, muñecos similares a humano.

Unidad 2 – Ingeniería de software Definiciones. Principios de ingeniería de software. Paradigmas. Participantes. Equipo de desarrollo. Roles. Producto y Proceso. Ciclo de vida. Modelos de proceso. Análisis. Ingeniería de requerimientos. Modelado: de datos, funcional, dinámico. Herramientas y técnicas de modelado. Especificaciones formales. Lenguajes. Diseño: general y detallado. Arquitecturas. Patrones. Pruebas.

Software: Instrucciones que cuando se ejecutan proporcionan las características, función y desempeño buscados, manipulan la información de forma adecuada. Es producto y herramienta para la construcción de un producto al mismo tiempo.

Hablamos de sistemas de software porque está compuesto por elementos como: Hardware sobre el que ejecutan, software que permite operar el hardware, usuarios que configuran y usan el sistema, bases de datos, y otras piezas de hw y sw adicionales.

Los sistemas existen en un contexto que define sus características, con el cual interactúan. Además, se definen en función de los límites que permiten entender que está comprendido en el sistema y que no. Qué funciones provee el sistema y cuáles le son provistas por otros sistemas. Que genera internamente y que obtiene de su contexto.

En cada sistema identificamos, las actividades que ocurren en el sistema, los objetos o entidades involucrados en las actividades, las relaciones entre los objetos y actividades, y el límite o frontera del sistema que lo limita de su contexto, para identificar cuáles entradas va a recibir el sistema y las salidas que va a producir.

Ingeniería de software: aplicación de un método sistemático, disciplinado y cuantificable para el desarrollo, la operación y el mantenimiento de software. Uso de principios de la ingeniería para desarrollar de forma económica software confiable, eficiente y que utilice máquinas reales.

Búsqueda de soluciones efectivas a problemas resolubles con computadoras.

Método: técnica formal para producir un resultado.

Procedimiento: combinación de herramientas y técnicas que producen un resultado. Formas de concretar el método.

Herramienta: instrumento o sistema automatizado para realizar una tarea de mejor manera.

Paradigma: enfoque particular o filosofía.

PARTICIPANTES DEL PROCESO DE DESARROLLO DE SOFTWARE:

Cliente: Necesita un desarrollo de software, plantea problema o situación. El lo paga.

Desarrollador: Equipo o persona que construye el software para el cliente.

Usuario: Utilizara el software creado, tiene necesidades específicas sobre el sistema.

Miembros del equipo de desarrolladores:

Analistas de requerimientos: Determinan que desea el cliente, documentan los requerimientos.

Diseñadores: toma la descripción de los deseos del cliente, describe lo que tiene que hacer el sistema.

Programador: Implementar requerimientos en el código.

Testers: Detectar defectos.

El trabajo se entrega al cliente, quien verifica que el producto obtenido cumpla los requerimientos.

Capacitadores: entrenan al cliente en el uso del sistema.

Equipo de mantenimiento: Trabaja en corregir defectos y/o cambiar aspectos del sistema con el correr del tiempo.

—

Proceso y flujos de proceso:

Proceso: conjunto de actividades, acciones y tareas a ejecutar que suceden en etapas para crear algún producto.

Actividad: Objetivo amplio, se desarrolla en cualquier dominio de aplicación, tamaño del proyecto o grado de rigor del trabajo. EJ: Rendir y cursar las materias de la carrera.

Acción: conjunto de tareas para obtener un producto del trabajo. EJ: Rendir y aprobar parcial.

Tarea: objetivo pequeño, bien definido, resultado tangible. EJ: Realizar ejercicio 1 tp.

El proceso del software consiste de:

Comunicación - Planificación - Modelado - Construcción - Despliegue

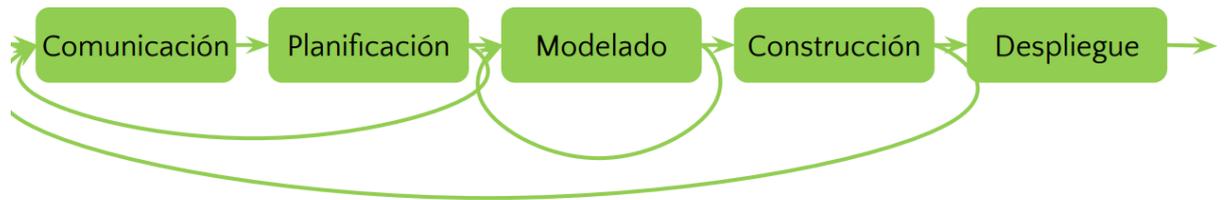
Se aplican a lo largo del proyecto para administrar y controlar el avance, la calidad los cambios y los riesgos. EJ: seguimiento y control, administración del riesgo, aseguramiento de la calidad, revisiones técnicas, medición, administración de la configuración.

La organización de las tareas y actividades se organiza respecto a su secuencia y tiempo.

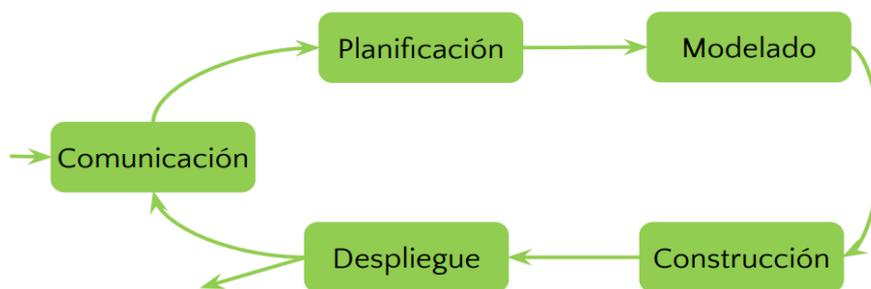
Lineal: secuencia de actividades



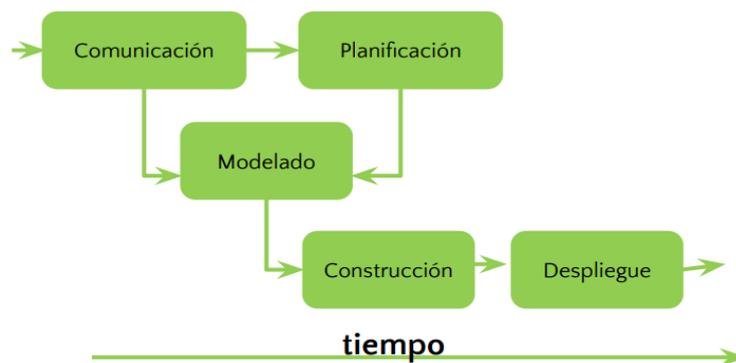
Iterativo: repite una o más de las actividades antes de pasar a la siguiente.



Evolutivo: cada circuito lleva a una versión más completa del producto

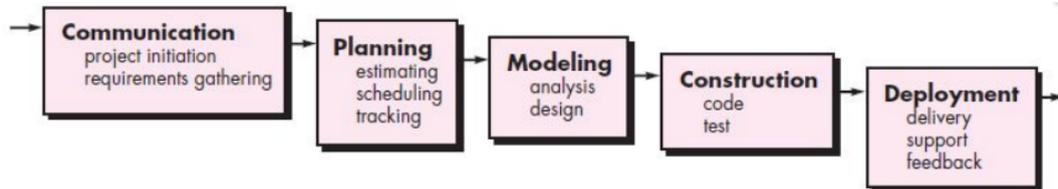


Paralelo: una o más actividades en paralelo con otras.



Ciclo de vida - Modelos de proceso

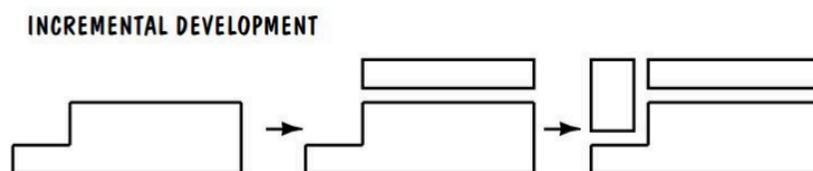
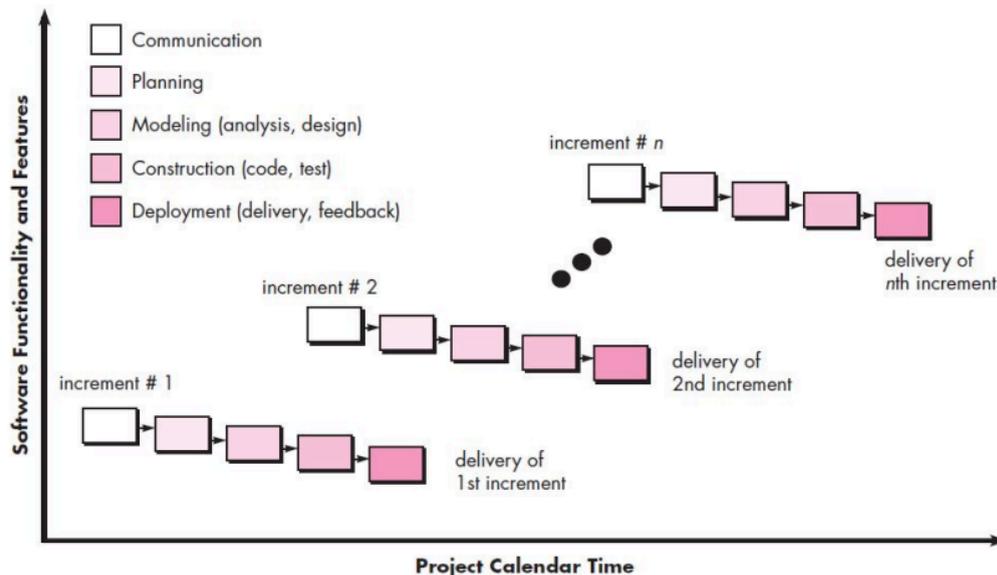
Modelo de cascada: Trabajo fluye de manera lineal, requerimientos bien definidos o para mejora de un sistema existente



Ventajas: Simple de explicar y entender para personas no familiarizadas con el desarrollo de software. Menciona los productos intermedios necesarios para avanzar

Desventajas: No es fiel a la realidad. No es flexible, no sirve cuando hay requerimientos dinámicos. El cliente debe esperar al final para ver una versión funcional.

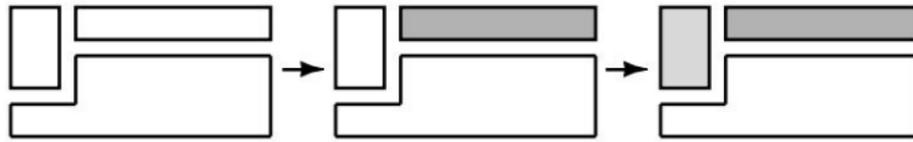
Modelo incremental: Secuencias lineales aplicadas de manera escalonada. Puede haber paralelismo entre distintas secuencias. Cada secuencia produce un incremento en el software, nueva parte de ese producto. Útil cuando no hay personal suficiente, se trabaja con porciones reducidas de trabajo.



S. Pfleeger

Modelo Iterativo o Evolutivo: El sistema se entrega completo al comienzo. Las iteraciones modifican (corrigen, mejoran) la funcionalidad.

ITERATIVE DEVELOPMENT

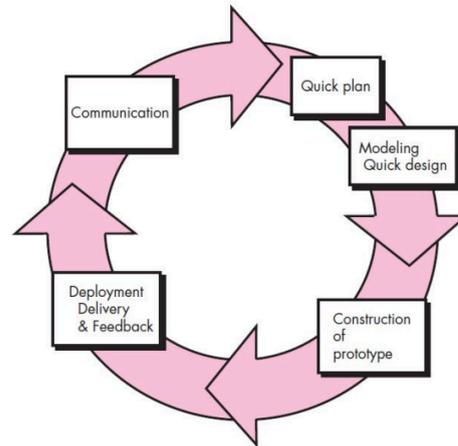


S. Pfleeger

Modelo Evolutivo - Prototipos: Se planea rápidamente una iteración para producir un primer prototipo. Cada iteración refina el conocimiento, el plan y el modelo, y genera un nuevo prototipo. Útil cuando sólo se plantean objetivos generales en vez de requerimientos detallados. Prototipos desechables o evolucionan.

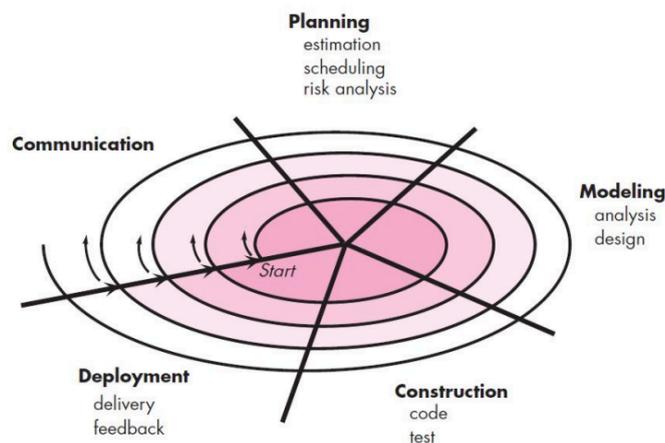
Ventajas: Dar una idea más temprana al usuario, obtener mejor definición de los requerimientos.

Desventajas: Riesgo de perder de vista que lo obtenido es un prototipo y no el producto definitivo.



y

Modelo Evolutivo - Espiral: Se itera de forma espiral. Las primeras iteraciones pueden producir modelos o prototipos. Cada iteración ajusta todos los documentos: definición, modelos, costos, incluso las iteraciones restantes. Se puede usar durante todo el ciclo de vida del sistema.



Modelos de proceso especializados: Características de procesos ya vistos, pero para enfoques específicos: desarrollo basado en componentes, orientado a aspectos y **métodos formales**.

Métodos Formales: Especificar, desarrollar y verificar un sistema de software basándose en notación matemática rigurosa. Evitar ambigüedad, inconsistencia, incompletitud.

Ventajas: Posible verificar y validar de manera rigurosa, permite descubrir ambigüedades e inconsistencias, bajo costo de mantenimiento casi por completo.

Desventajas: Hay pocos desarrolladores con capacitación adecuada para usarlo, difícil utilizar para comunicarse con los clientes.

Metodologías Ágiles:

Manifiesto ágil: Se firma el manifiesto para el desarrollo ágil de software. Se plantea desarrollar software moviendo el foco de los procesos y las herramientas a los individuos y sus interacciones. Se valora más la colaboración con el cliente que la negociación de los cambios. Responder a un cambio antes que seguir un plan.

A medida que el proyecto avanza más, cambiar los requerimientos cuesta más. El proceso debe ser adaptable, debe hacerlo incrementalmente y debe permitir la retroalimentación con el cliente. Hay que intentar que el inevitable cambio surja tan temprano como sea posible. Diferentes modelos enfatizan distintos principios, pero se preserva el “espíritu ágil” (se enfatizan la competencia, el enfoque común, colaboración, capacidad de resolver problemas, confianza y respeto, mas que el conocimiento técnico). Ejemplos de modelos basados en esta metodología:

Programación extrema: Enfatiza la colaboración estrecha pero informal entre clientes y desarrolladores y la retroalimentación constante.

Cuatro actividades estructurales:

Planificación: Recabar requerimientos.

Diseño: Manténlo sencillo. Si es difícil, se realizan prototipos.

Codificación: No codificar, diseñar pruebas unitarias, luego desarrollar el código. Se trabaja de a pares.

Pruebas: Automatizarlas cuando sea posible. Efectuar pruebas de integración a diario.

Scrum: Considera actividades estructurales: Requerimientos, análisis, diseño, evolución, entrega. Cada actividad realiza las tareas bajo un patrón de proceso llamado sprint, que son unidades de trabajo necesarias para alcanzar un requerimiento (deben ser entre 2 semanas y 2 meses de duración). Cada día de trabajo toma lugar una reunión scrum, donde se pregunta: Que hiciste desde la última reunión, qué obstáculos encontraste, y que planea hacer antes de la próxima reunión.

Ingeniería de requerimientos:

Requerimiento: Característica del sistema o descripción de algo que el sistema es capaz de hacer con el objeto de satisfacer su propósito.

Ingeniería de requerimientos: Las tareas alrededor del requerimientos. Lo que hacemos para entender que desea el cliente, evaluar la factibilidad, negociar acuerdos razonables, especificar soluciones sin ambigüedades, validar especificación, administrar requerimientos. En esta se incluyen las **siguientes actividades:**

Concepción: Inicio de un proyecto de software, se establece el entendimiento básico del problema, se identifican participantes, se reconocen diferentes puntos de vista, se identifican áreas de interés común, preguntas entre cliente y participantes.

Indagación: preguntas de alcance (detalles que confunden o innecesarios, frontera mal definida), de entendimiento (dar algo por obvio, no está seguro de lo que necesita, requerimientos que no se pueden probar) y volatilidad (requerimientos cambiantes con el tiempo). Hay que poder distinguir los requerimientos; los que deben ser absolutamente satisfechos, los deseables pero no indispensables, y los posibles pero que pueden eliminarse. Identificar QUE, en esta etapa, no el CÓMO: No confundir requerimientos con soluciones.

Hay requerimientos funcionales (que hará el sistema, interacciones entre el sistema y el entorno) y no funcionales (describen restricciones que limitan elecciones en construcción de la solución). Identificar escenarios y casos de uso (que quieren conseguir los actores en la interacción con el sistema, variaciones en la interacción, que información adquiere, produce o cambia el actor)

Elaboración: Expandir y refinar información obtenida. Se comienza con la elaboración de modelos que permitan representar la función, el comportamiento y la información trabajada por el sistema.

Negociación: Toma lugar cuando el cliente pide más de lo que se puede conseguir, o hay requerimientos conflictivos. Se eliminan y/o modifican requerimientos hasta lograr satisfacción.

Especificación: Puede tener muchas formas: Documentos escritos, modelos gráficos, modelos matemáticos formales, conjuntos de escenarios de uso, prototipos, o combinaciones de los mismos.

Validación: Se analiza la especificación para garantizar que todos los requerimientos han sido enunciados, no hay ambigüedades, se corrigieron inconsistencias y errores, y se siguieron los estándares establecidos. Esto involucra a ingenieros de software, clientes, usuarios, expertos, etc...

Administración: Se debe identificar, controlar y dar seguimiento a los requerimientos y a los cambios que sufran. Determinar cómo se administrarán los cambios.

Modelado de requerimientos:

Modos de representación:

Basados en el escenario: requerimientos desde el punto de vista de “actores” del sistema.

De datos: ilustran el dominio de información del problema.

Orientados a clases: Desde el punto de vista de los objetos que manipula el sistema para cumplir los requerimientos del sistema.

Orientados al flujo: transformación de los datos a medida que se mueven en el sistema.

De comportamiento: modo en el que se comporta el software como consecuencia de eventos.

Expresión de requerimientos: Los requerimientos pueden ser expresados de múltiples formas:

Caso de uso: Se narra desde el punto de vista del actor, es como dar un tutorial o explicación personal (es una narrativa informal).

Secuencia de acciones: Se enumeran las acciones o pasos a seguir en cuenta.

Diagrama Gráfico: Es un gráfico de dibujo a semejanza del sistema.

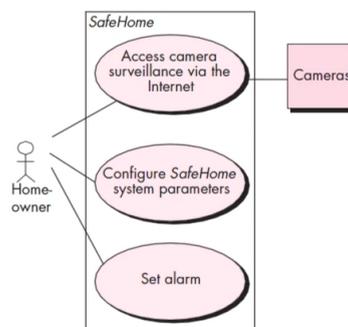
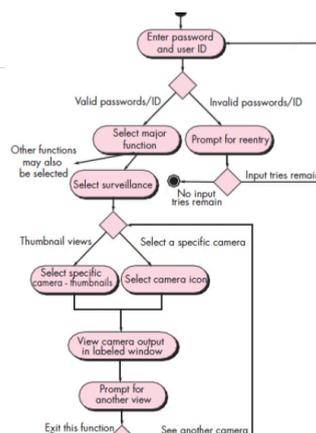


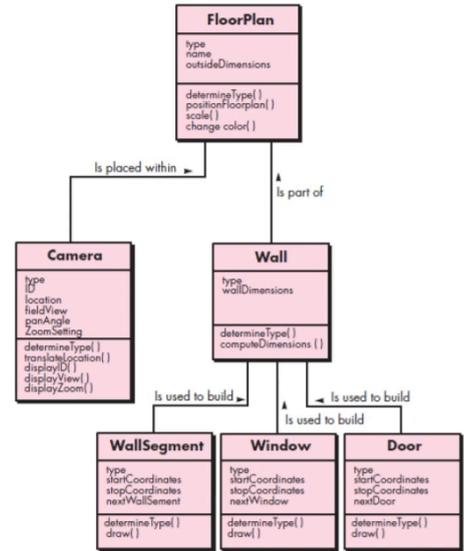
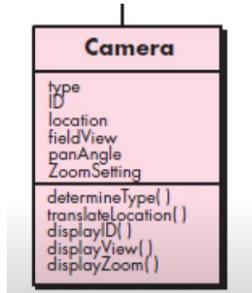
Diagrama de actividad: Representación gráfica del flujo de interacción, especifica diferentes caminos resultados obtenidos tras tomar diferentes decisiones.



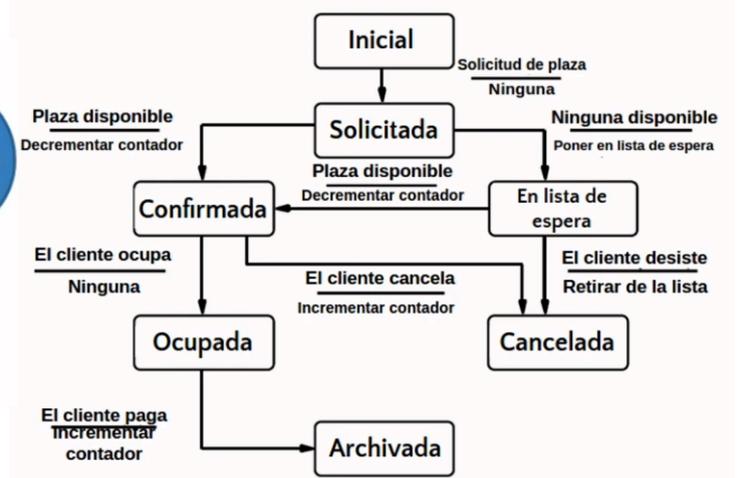
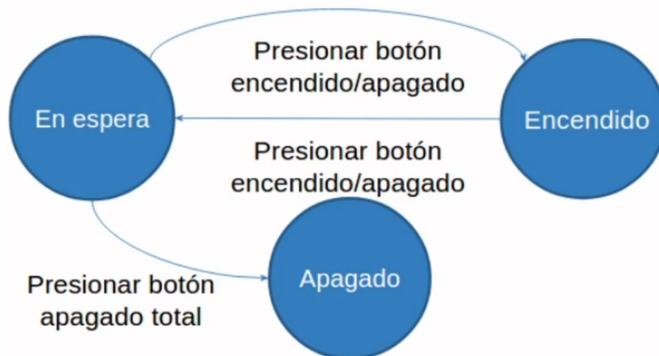
Descripciones estáticas: objetos y clases, operaciones, relaciones entre objetos.

Especificar atributos y definir operaciones.

Hay que identificar clases, objetos y atributos, y ocurrencias que suceden dentro del contexto.



Descripciones dinámicas: Descripciones funcionales y diagramas de transición. Modelan como se expresan, como ocurren los cambios en las entidades de las descripciones estáticas. Diagramas de transición:



Tablas de decisión: De acuerdo al conjunto de condiciones se definen las acciones a realizar del sistema.

| | R1 | R2 | R3 | R4 | R5 |
|--------------------------------|----|----|----|----|----|
| Notas altas de examen | V | F | F | F | F |
| Grados superiores | - | V | F | F | F |
| Actividades externas | - | - | V | F | F |
| Buenas recomendaciones | - | - | - | V | F |
| | | | | | |
| Enviar carta de rechazo | | | X | X | X |
| Enviar formularios de admisión | X | X | | | |

Validación de requerimientos: Se determina si la especificación es consistente con la definición de los requerimientos. Se asegura de que cada especificación pueda ser rastreada hasta su requerimiento en el documento de definición. Luego, se chequea la definición para ver si cada requerimiento es rastreable hasta la especificación.

Técnicas de validación:

| | |
|------------------------|---|
| Técnicas manuales | Lectura. Cruce de referencias manual. Entrevistas. Revisiones. Listas de comprobación. Modelos manuales para chequeo de funciones y relaciones. Escenarios. Pruebas matemáticas. |
| Técnicas automatizadas | Cruce de referencias automatizado. Modelos automatizados para poner en ejecución funciones. Prototipos. |

Medición de los requerimientos: Se requiere detalle del proceso de los requerimientos, y de la calidad de los mismos. Se enfoca en producto, proceso y recursos.

Características:

Correctos: sin errores

Consistentes: sin conflictos ni ambigüedades:

Completos: externa e internamente.

Realistas: ¿puede hacer lo solicitado realmente?

Necesarios: ¿sin restricciones innecesarias.

Verificables: ¿Se puede probar que cumplen?

Rastreables: ¿Se pueden identificar fácilmente?

Diseño de software: Diseñar un sistema es determinar un conjunto de componentes y de interfaces entre componentes que satisfagan un conjunto específico de requerimientos.

Conceptos de diseño:

Abstracción: Se plantean distintos niveles de abstracción. Abstracciones de procedimiento (secuencia de instrucciones), abstracciones de datos (conjunto que describe un objeto). Más abstracción, soluciones en términos amplios. Menos abstracción, descripciones mas detalladas.



Arquitectura: Estructura de organización de los componentes, la forma en la que interactúan, y las estructuras de datos que utilizan. A partir de ella sale el diseño más detallado. Se usan patrones.

Patrones: Describen estructuras de diseño que resuelven un problema particular de diseño dentro de un contexto específico

Separación de Intereses: Problemas complejos se resuelven más fácilmente si se lo subdivide en partes independientes.

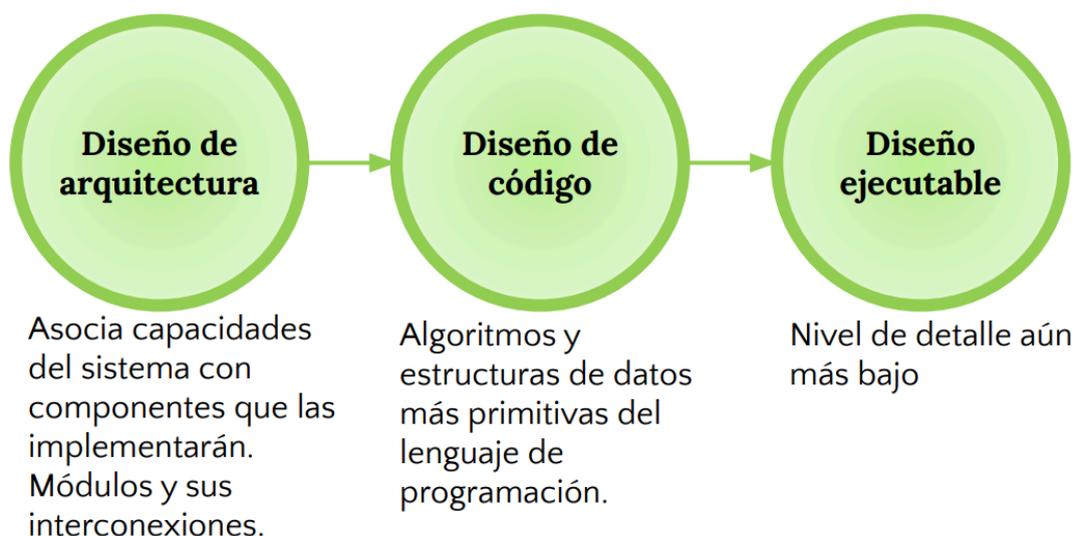
Modularidad: El software se divide en partes separadas (módulos). Un sistema es modular cuando cada actividad la realiza un único componente que tiene bien definidas sus entradas y salidas. El costo de integrar módulos crece a medida que son más módulos.

Ocultamiento de la información: La información de cada módulo debe ser inaccesible a los que no necesiten de ella. Los módulos tienen que conseguir intercambiar solamente la información necesaria para que el software funcione.

Independencia funcional: Cada módulo debe resolver un conjunto específico de requerimientos. Resultado de división de problemas y ocultamiento de información.

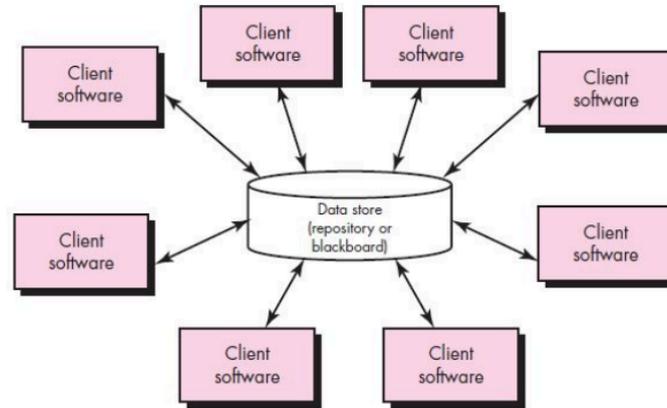
Refinamiento: aka desde arriba hacia abajo, se empieza desde una descripción general, y se va refinando hasta llegar a un código en un lenguaje de programación.

Diseño arquitectónico:



Estilos de arquitectura:

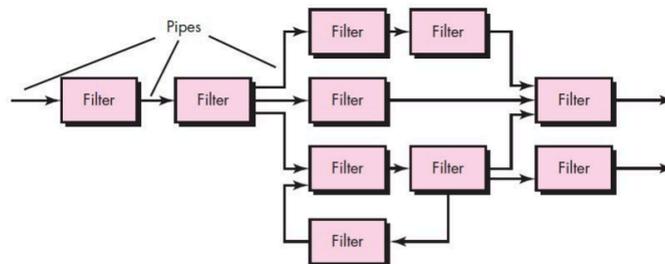
Centrada en los datos (Repositorios): Almacenamiento central de datos. Conjuntos de componentes que operan sobre el almacenamiento: almacenar, recuperar, actualizar.



Flujo de datos: tuberías y filtros:

Tuberías: corrientes de datos para entrada y salida.

Filtros: realizan transformaciones de los datos. Son independientes entre sí y se pueden ejecutar en simultáneo. Se pueden reutilizar en otros sistemas y quitar e incorporar nuevos en cualquier momento. Desventajas: al ser filtros independientes, se puede duplicar la funcionalidad que efectúan otros.



De llamada y retorno: Clásica estructura jerárquica, programa principal y subprogramas componentes.

Diseño orientado a objetos: Los componentes de un sistema encapsulan datos para trabajarlos. La comunicación entre componentes se consigue mediante pasaje de mensajes.

Capas: Se define un número de capas, cada una implementa operaciones. De afuera para adentro es operaciones de interfaz del usuario, utilidades y funciones de aplicación, componentes implementan interfaz con el sistema operativo. Cada capa presta servicios a la capa superior y es cliente de la inferior. Es simple agregar o modificar capas, se afecta solo a capas adyacentes, pero no es sencillo estructurar sistemas de esta forma.

Características de un buen diseño:

Que cada componente del diseño sea lo más independiente del resto posible, ya que modificarlo no afecta al resto de componentes, y es más fácil identificar fallas. Para medir la independencia de los componentes usamos dos conceptos:

Acoplamiento: Dos componentes están acoplados si existe dependencia entre ellos. Altamente acoplados, mucha dependencia entre ellos. Poco acoplados, pocas dependencias y/o interconexiones débiles. Depende de: Referencias hechas de un componente a otro, cantidad de datos pasados entre ellos, grado de control de uno sobre otro, y complejidad de interfaz entre los componentes.

* Acoplamiento de contenido: Cuando un componente A, modifica un componente B. B depende completamente de A.

* Acoplamiento común: Almacenamiento de datos es común a dos o más módulos.

* Acoplamiento de control: Cuando un componente pasa parámetros de control a otro.

* Acoplamiento de molde: Se utiliza una estructura de datos compleja para pasar información de un módulo a otro.

* Acoplamiento de datos: Solo se comparten datos.



Cohesión: Grado de adhesión interna que tiene el componente. A mayor grado de cohesión, más relacionadas están sus partes internas entre sí. Un componente es cohesivo si todos sus elementos están orientados a realizar una única tarea y son esenciales para llevarla a cabo.

* Cohesión coincidental: las partes no tienen relación alguna entre sí.

* Cohesión lógica: los elementos están relacionados lógicamente (todas las funciones para la entrada de datos sin importar el origen, por ejemplo).

* Cohesión temporal: las funciones de un componente están relacionadas por el momento en el que ocurren o son invocadas.

* Cohesión de procedimiento: Cuando las funciones se agrupan únicamente porque están relacionadas por un procedimiento, hay un orden de ejecución.



* Cohesión comunicativa: Cuando un componente asocia funciones que trabajan en el mismo conjunto de datos.

* Cohesión secuencial: Si la relación entre las funciones es que se produce la salida que sirve de entrada a la siguiente.

* Cohesión funcional: Es la ideal. En estos componentes las partes están relacionadas por la función que realizan: es una única función y todas las partes son esenciales para realizarla.

Un buen diseño minimiza el acoplamiento y maximiza la cohesión.

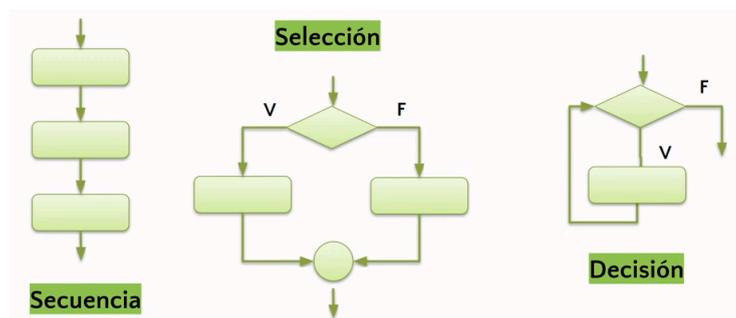
Diseño de situaciones no deseadas: Identificar las excepciones (situaciones que se sabe que no son las esperadas). El diseño puede incluir manejo de excepciones de modo que el sistema pueda reaccionar de manera satisfactoria ante un evento de este tipo.

Excepciones comunes: Fracasar al proporcionar un servicio, proporcionar un servicio erróneo, o corrupción de datos. Acciones a tomar: Intentar (restaurar el sistema y probar otra forma), corregir (restaurar el sistema y probar nuevamente), e informar (restaurar el sistema e informar de lo ocurrido sin proporcionar el servicio).

Implementación de una solución:

Codificación del software: varios lenguajes de programación y personas involucradas, se coopera y se coordina. Se definen estándares de programación: Estilo, formato, código y documentación. A partir del diseño del componente el programador tiene libertad para ser creativo en la implementación. Cada componente implica al menos:

* Estructuras de control: Se elige uno según la estructura del código. Balance entre facilidad de comprensión del código y eficiencia.



* Algoritmos: Se tiene en cuenta el tiempo de ejecución, y el costo (de escritura, de prueba y modificación). No vale la pena sacrificar claridad y corrección por velocidad de ejecución.

* Estructuras de datos: Se busca que el almacenamiento y manipulación de los datos sea directa, y que ayude a simplificar el programa. La estructura de datos puede determinar la estructura del programa

Documentación del código: Explica que hace el programa y como lo hace.

Documentación Interna: material descriptivo escrito dentro del código (quien lo escribió, justifica las decisiones, ayuda a comprensión).

Documentación Externa: la que acompañe al programa (descripción del problema, de los algoritmos elegidos y de los datos).

Prueba de los programas: Se concentra en búsqueda de defectos, encontrarlos. La prueba es para demostrar su existencia. La prueba es destructiva, es exitosa si encuentra defectos o fallas.

Tipos de defectos:

Algorítmicos: error en la lógica de un componente que no produce la salida apropiada.

De sintaxis: en las estructuras del lenguaje.

De computación y precisión: implementación errónea de fórmulas o cálculos sin exactitud necesaria.

De documentación: Cuando no corresponde con lo que hace el programa.

Por estrés o sobrecarga: se sobrepasan las estructuras.

De capacidad o límites: Se deteriora el desempeño del sistema por superar sus límites.

De sincronización: coordinación inadecuada de procesos simultáneos.

De rendimiento: el sistema no opera con el desempeño esperado

De recuperación: respuesta a una falla no es la esperada.

Visión de los objetos de pruebas:

Caja cerrada o negra: Se analiza el objeto sin conocer el contenido. Alimentar la caja negra con entradas y observar las salidas.

Caja abierta o blanca: Considera la estructura, y ajusta los casos de prueba en base a esta.

Organización de pruebas:

De módulo, de componente o de unidad: Pruebas para verificar si funciona correctamente con las entradas correctas y esperadas.

De integración: Si los componentes funcionan correctamente como sistema.

Funcional: Determinar si las funciones que se describieron en la especificación se cumplen en el sistema.

De desempeño: Comparar sistema con requerimientos de hardware y software.

De aceptación: Se prueba el sistema con el cliente para ver si cumple los requerimientos desde su visión.

De instalación: para verificar que funcione correctamente una vez instalado

Casos de prueba: Conjunto de casos de entrada a ser utilizados para probar, cambian según el enfoque (comprobación de sentencias, del camino, pruebas de la ramificación...).

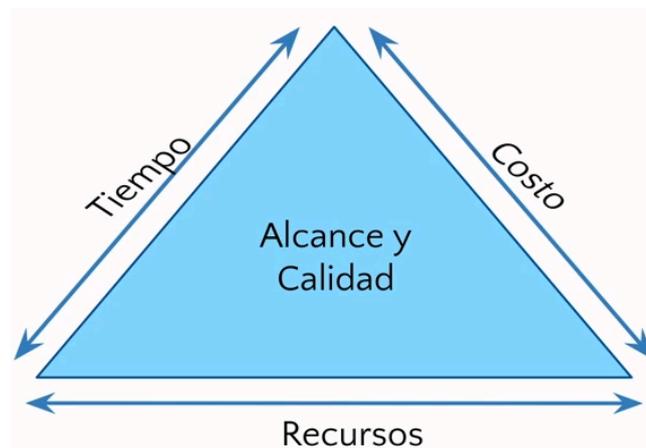
Planificación de la prueba: - pasos: determinar objetivos, diseñar casos de prueba, describir los casos, verificarlos, ejecutarlos, y evaluar los resultados.

Equipo de prueba: Idealmente el equipo de pruebas debe ser independiente al equipo de implementación. Se encargan de organizar, diseñar y ejecutar las pruebas, en simultáneo con el desarrollo del proyecto y tan pronto como el desarrollo lo permita.

Unidad 3 – Administración de proyectos. Proyecto. Actividades. Trabajo. Esfuerzo. Recursos. Tiempo. Estimaciones. Riesgos. Equipos. Organizaciones. Planificación. Seguimiento. Técnicas y herramientas. Teoría de la medición. Mediciones y métricas. Estándares. Calidad: interna y externa. Calidad del producto y del proceso. Seguridad. Estándares. Documentación.

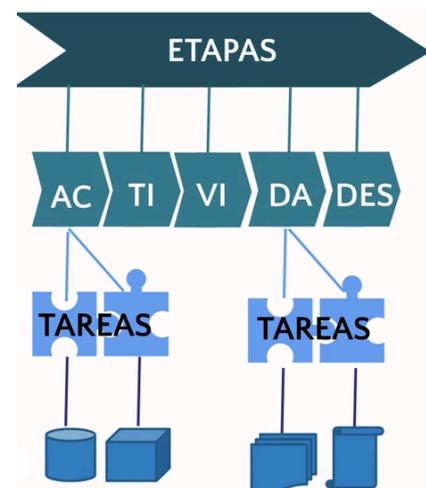
Administración de proyectos: Que actividades involucra el desarrollo, cuanto tiempo llevara, que costos, por dónde empezar. Para responder a esto se determina: El alcance del proyecto, determinar que tareas se realizarán, construir un cronograma, y analizar costos y riesgos.

Proyecto de software: Secuencia de actividades (porción de trabajo definida y ordenada) únicas, complejas y conectadas (actividades complejas, nuevas e interrelacionadas entre sí) que tienen un propósito y que deben ser completadas para un tiempo determinado. Dentro del presupuesto y de acuerdo a la especificación (expectativas del cliente). El proyecto tiene restricciones: alcance, calidad, costo, tiempo, y recursos.



Descomposición del trabajo:

Desarrollo del proyecto: Se puede separar en fases o etapas, subdivididas en actividades, y se continúa subdividiendo hasta llegar a tareas. Cada tarea tiene un entregable



Estimaciones:

Duración: Tiempo transcurrido necesario para terminar una tarea. Va a depender del esfuerzo que requiera la tarea, y los recursos que se le asignen.

Recursos: Muchos tipos de recursos: personal, instalaciones, equipamiento, dinero, materiales. Las personas son el recurso más difícil de administrar. Se especifican las habilidades o perfiles necesarios, y cuando y cuanto se los necesita.

Costo: Se obtiene una estimación, y los costos de todos los elementos variarán.

Riesgo: Evento no deseado que tiene consecuencias negativas. Siempre existen, con diferente probabilidad de ocurrencia y efecto. Administrar el riesgo al realizar las actividades necesarias para minimizar la probabilidad de ocurrencia y los efectos posibles que conlleva.

Ciclo de vida de administración de riesgos: Identificar, evaluar, mitigar, monitorear, controlar, y gestionar el riesgo ocurrido. Una vez identificados, se evalúa la probabilidad de ocurrencia y el impacto de esta. Se debe evaluar el impacto sobre todas las restricciones y para aquellos con mayor probabilidad e impacto se elabora un plan de riesgos.

Estimación de riesgos: Solo los riesgos que generen mayor impacto recibirán mayor atención. Se elabora un plan de reducción, supervisión y gestión del riesgo.

Técnicas de estimación: Similitud con otras tareas, datos históricos, asesoramiento experto, técnica delphi, técnica tres puntos, técnica delphi de banda ancha.

Técnica Delphi: Resume la experiencia de todo el equipo. Cada miembro estima una duración, se tabulan los resultados, se justifican las estimaciones extremas, y se estima de nuevo la duración. Se repite 3 veces y se promedian los resultados de la tercer pasada.

Técnica de 3 puntos: Se basa en la idea de que la duración es variable. La misma tarea puede tener diferentes duraciones. Estimaciones Optimista, Más probable y Pesimista.
 $E = (O + 4M + P) / 6$

Técnica Delphi de banda ancha: Combinación de las anteriores. Las 3 pasadas de Delphi, en cada pasada los miembros dan sus estimaciones para Optimista, Más probable y Pesimista, se recopilan los valores, se descartan los extremos y se calculan los promedios.

Estimación de recursos: Se pueden usar las técnicas anteriores, menos los recursos humanos. Para las personas se especifican las habilidades necesitadas, en qué etapa del proyecto se los necesita, y cuantas personas para cada perfil.

Para recursos humanos podemos usar las matrices de habilidad:



Organización de equipos: Se tienen en cuenta: capacidad, interés, experiencia, habilidades de comunicación, responsabilidad individual y compartida, capacidad de gestión. Se obtiene mayor productividad y calidad si la gente se siente cómoda.

Incrementar la cantidad de personas no necesariamente va a incrementar la productividad (más canales de comunicación, más necesidad de documentar, más responsabilidad). Cada persona debe mantener una comunicación fluida con el resto de integrantes.

Cada uno debe tener un rol distinto, debe haber un líder que conozca a los integrantes y sus capacidades y personalidades. Debe poder motivar a todos los miembros del equipo, habilidades para organizar, buscar las mejores herramientas y métodos.

Hay diferentes estilos de trabajo y personalidades y hay que comprenderlos para saber qué esperar de cada integrante y decidir cómo delegar tareas.

Actividades: Existen dependencias entre las actividades:

Técnicas: Porque una tarea requiere la salida de otra para poder comenzar el trabajo.

De gestión: decisiones del administrador del proyecto para conseguir objetivos.

Entre proyectos: cuando un proyecto grande es dividido en subproyectos.

De fechas: restricciones de fecha de comienzo o de completitud. Evitar.

Cuatro tipos de dependencia:

Fin a comienzo: Tarea predecesora A, debe finalizar para que B comience. La más común.

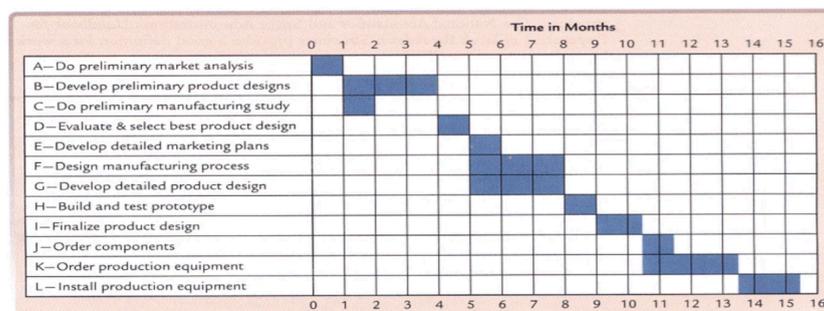
Comienzo a comienzo: La predecesora A debe comenzar para que sucesora B pueda comenzar.

Fin a fin: Predecesora A debe finalizar para que Sucesora B finalice.

Comienzo a fin: La predecesora A debe comenzar para que la Sucesora B finalice.

Planificación: Una vez estimadas la duración de tareas, recursos, y tareas de administración de riesgos a realizar, se comienza a planificar. Se utilizan diagramas de red que muestran las dependencias y el orden y el tiempo en el que se realizan.

Diagrama de Gantt: Un rectángulo por tarea, muestra que tareas se pueden trabajar en simultáneo. La longitud corresponde a la duración de la tarea. Es simple pero deja de lado demasiado detalle. No muestra dependencias.



Diagramas de red: Las tareas son nodos, las dependencias los arcos que unen. Muestra el flujo secuencial del trabajo. Incluye detalles de la planificación, fechas de comienzo, fin de tareas.

Diagramas con actividades en los arcos:

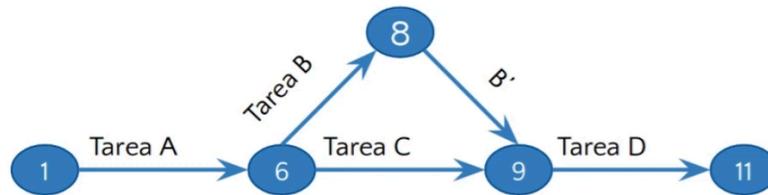


Diagrama con actividades en los nodos:



● Diagrama de precedencias para el proyecto



Diagrama de precedencia nos ayuda a determinar la fecha fin de proyecto, planificación más temprana del proyecto, identificar proyectos temprano, concurrencia de actividades, detectar retrasos.

Camino crítico: Es el camino o secuencia de tareas más larga del diagrama, que determina la terminación del proyecto y si se retrasa, retrasa todo.

Calidad:

Calidad del diseño: características que los diseñadores especifican para un producto. Aumenta la calidad del producto si se fabrica de acuerdo a las especificaciones.

Calidad de la conformidad: grado en el que el resultado cumple con sus metas.
Satisfacción del usuario = producto que funciona + buena calidad + entrega dentro de presupuesto y plazo.

Calidad del software: Proceso eficaz de software que se aplica de manera que permita crear un producto útil que proporciona valor medible a quienes lo producen y quienes lo utilizan.

Costos de calidad: Hay costos por buscar calidad, y costos por falta de calidad. Existen costos de prevención, evaluación y fallas internas y externas. Aumentan a medida que avanza el proyecto

Actividades de prevención:

Administración: planear y coordinar actividades de control y aseguramiento de calidad.

Pruebas: planear las pruebas.

Técnicas: agregadas para desarrollar modelos completos de requerimientos y diseño.

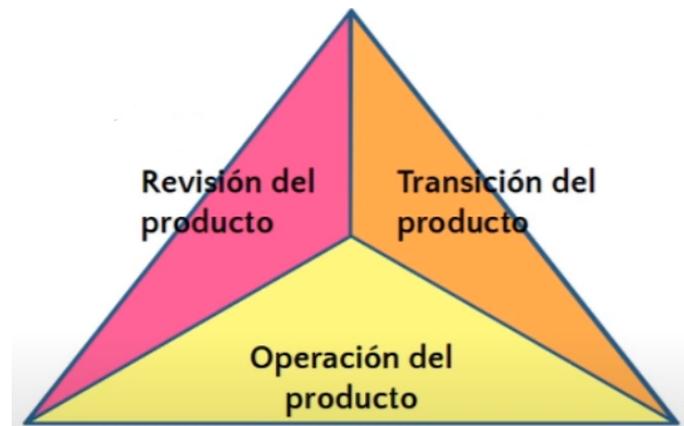
Capacitación: toda la capacitación asociada con actividades de prevención.

—

Evaluación: Investigar condición del producto antes de ingresar a cada proceso. Revisiones técnicas, obtención de datos, medidas, pruebas.

Costos por Falla: Se eliminarían si no hubiese errores antes o después de enviar el producto. Internos (detectados antes del envío del producto) y Externos (encontrados después de enviar el producto al consumidor). Más avanzado en el proyecto se encuentran errores, más esfuerzo van a requerir arreglarlos y más efectos van a tener.

Factores de calidad de McCall: Clasificación de factores que afectan a la calidad del software.



Operación del producto:

Correctitud: grado en el que satisface sus especificaciones y en el que cumple con los objetivos del cliente

Confiabilidad: grado en el que se espera que cumpla con su función y con la precisión requerida.

Usabilidad: esfuerzo que se requiere para aprender, operar, preparar las entradas e interpretar las salidas del programa.

Integridad: grado en el que es posible controlar el acceso de personas no autorizadas al software o a los datos.

Eficiencia: cantidad de recursos de cómputo y de código que requiere para llevar a cabo su función.

Revisión del producto:

Facilidad de recibir mantenimiento: esfuerzo requerido para detectar y corregir un error en el programa (esto sería el mantenimiento correctivo solamente, hay adaptativo, perfectivo y preventivo)

Flexibilidad: esfuerzo necesario para modificar un programa que se encuentra en producción.

Susceptibilidad de someterse a pruebas: esfuerzo requerido para probarlo a fin de garantizar que realiza la función pretendida.

Transición del producto:

Portabilidad: esfuerzo necesario para transferir un programa de un ambiente de hw o sw a otro diferente.

Reusabilidad: grado en el que un programa o sus partes pueden volver a utilizarse en otra aplicaciones.

Interoperabilidad: esfuerzo requerido para acoplar un sistema con otro.

Estos factores se miden de manera indirecta.

Estándares: Documentos, recomendaciones, reglas producidas por organizaciones reconocidas. Su adopción es opcional aunque pueden ser impuestas por reglamentaciones o clientes.

ISO: Organización internacional de estandarización. Norma ISO 9000 (estándares relacionados con la administración de la calidad en los sistemas, provee guías y herramientas para organizaciones que quieran asegurar la calidad de sus productos y servicios) . Norma ISO 9001 (establece sistema de administración de calidad). La norma ISO 9000 define los siguientes principios: foco en el cliente, liderazgo, participación de la gente, método de procesos, decisiones basadas en evidencia, mejora continua, gestión de relaciones.

Norma ISO/IEC 25000 (SQuaRE): evolución de otras normas.

CMM: Modelo de madurez de capacidad.

CMMI: Integración de modelo de madurez de capacidad.

Enfoque de mejoramiento de procesos, define elementos claves de un proceso efectivo y un camino de mejora evolutivo. 5 niveles de madurez (inicial, repetible, definido, dirigido, optimizado).



Certificación: Organizaciones deben certificar cumplimiento de normas o modelos. Hay empresas que evalúan y pueden otorgar la certificación. Es costoso pero otorga beneficios a las organizaciones (se mejoran procesos, eficiencia, competitividad, calidad y se gana imagen frente a clientes.).

Proceso de medición:

Métrica basada en funciones: La intención es predecir el tamaño del producto de software que resultará. El tamaño puede indicar la complejidad del diseño, el esfuerzo que requerirá codificarlo, integrarlo o probarlo. Ej:

Puntos de función (PF): Se usa para predecir la funcionalidad de un sistema. Estima costos o esfuerzo requerido para diseñar, codificar y probar el software. Predice el número de errores durante pruebas y el número de componentes y/o líneas de código del sistema a implementar.

Para calidad de especificación: Se mide la calidad del software. Sugiere representar las características mediante una o más métricas.

Métricas del diseño arquitectónico: Se enfocan en características de la arquitectura del programa. Son de caja negra (no requieren conocimiento sobre el funcionamiento interior de los componentes). Si aumenta la complejidad arquitectónica aumenta el esfuerzo de integración de módulos y de pruebas. Medidas propuestas: Card y Glass

Del diseño orientado a objetos: Características medibles de un sistema orientado a objetos: tamaño, complejidad, acoplamiento, suficiencia, completitud, cohesión, primitivismo, similitud, volatilidad. Hay tres tipos de métricas de este tipo: Orientadas a la clase (CK), orientadas a la clase (MOOD) y de Lorenz y Kidd.

De la interfaz del usuario: La calidad y usabilidad. Colores, tipografías, gráficos.

Para el código: LOC: Ver cuantas líneas de código hay en el programa o módulo, ver si vale la pena comprimir líneas o no, comentarios, líneas en blanco.

Halstead: Expresiones de longitud del programa, volumen de información, nivel del programa, lenguaje, etc..

Para las pruebas: La mayor parte se enfoca en el proceso de las pruebas, no en las características técnicas de las pruebas. Métrica de Halstead (estima el esfuerzo de pruebas mediante métricas derivadas de las de Halstead) y Métricas para pruebas OO (consideran aspectos de encapsulación y herencia).

Para el mantenimiento: Pueden usarse todas las anteriores. Se utiliza el índice de madurez de software, y se basa en los cambios ocurridos en cada versión del software (cantidad de módulos originales, actuales, agregados, borrados).

Métricas del proceso: El objetivo es proporcionar indicadores para mejorar el software a largo plazo. Se recopilan durante mucho tiempo. Permiten valorar el estado del proyecto en marcha, rastrear riesgos potenciales, descubrir áreas del problema con anticipación, ajustar el flujo de trabajo, evaluar la habilidad del equipo para controlar calidad de los productos.

Métricas del proyecto: Útiles durante estimación. Resultados anteriores se usan como base de estimaciones actuales. Minimizar calendario de desarrollo, valorar calidad de producto actual, modificar enfoque técnico.

Proceso de medición: Puede caracterizarse mediante 5 actividades:

Formulación: Derivación de medidas y métricas apropiadas.

Recolección: de datos para obtener las métricas formuladas.

Análisis: cálculo de métricas y aplicación de matemáticas.

Interpretación: conocimiento ganado a partir de la representación usada.

Retroalimentación: Recomendaciones derivadas de la interpretación y análisis de los resultados.

**Unidad 4 – Sistemas de información. Teoría de la información y la comunicación.
Definición de sistemas. Teoría general de sistemas. Sistemas de software.
Clasificación de sistemas. Tecnologías de la información.**

Dato: Hecho aislado. Número de documento, valor de venta, nombre de una materia.

Información: conjunto de datos organizados de tal forma que poseen un valor adicional. Ventas totales de una empresa, ventas totales por productos, productos más rentables. Debe ser accesible, exacta, completa, económica, flexible, relevante, confiable, segura, simple, oportuna, verificable.

Conocimiento: Comprensión de un conjunto de información y de las formas en que esta puede convertirse en algo útil para realizar una tarea específica o tomar una decisión.

Nivel de insulina: dato

Historia médica de la persona: información

Insulina observada en relación a la historia médica de la persona permite que alguien con el conocimiento pueda hacer un diagnóstico.

Sistema: Conjunto de elementos o componentes que interactúan para alcanzar un objetivo. (Entradas, mecanismos de procesamiento, salidas, retroalimentación. EJ: Lavado de autos).

Sistema de información: Conjunto de componentes interrelacionados que recaban, procesan, almacenan y distribuyen datos e información y proporcionan un mecanismo de retroalimentación para cumplir un objetivo. Ayuda a organizaciones a incrementar ganancias, mejorar servicio, reducir costos, aumentar la velocidad.

Entrada: Actividad consistente en recopilación y captura de datos.

Procesamiento: Conversión o transformación de datos en salidas útiles.

Salida: producción de información útil en forma de reportes o documentos.

Retroalimentación: información proveniente del sistema que se utiliza para realizar cambios en las actividades de entrada y procesamiento.

Sistema de información **manual**: calcular tendencias en base a recopilación de documentos, lectura, intuición.

Sistema de información **computarizado**: grandes cantidades de info, cálculos y predicciones.

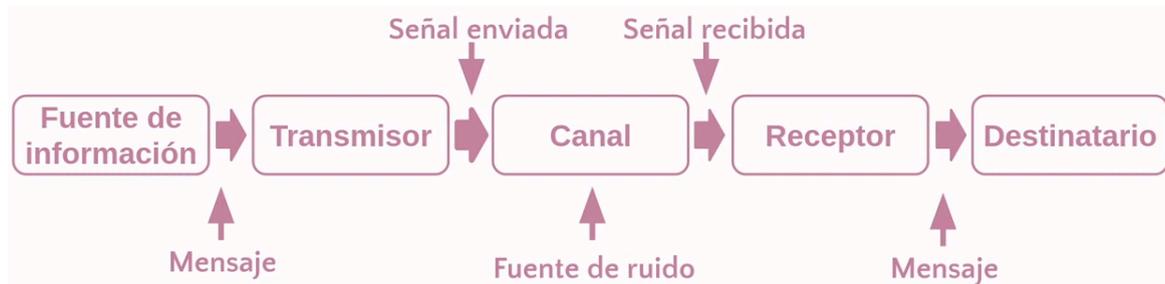


Desempeño del sistema:

Eficiencia: que se produce dividido que se consume.

Eficacia: en qué grado el sistema alcanza sus objetivos

Modelo de comunicación:



Teoría de la información: Se estudian los mecanismos de codificación para enviar información de emisor a receptor.

Fuente de información: Selecciona el mensaje deseado de un conjunto de mensajes.

Transmisor: transforma o codifica esta información de forma apropiada al canal.

Señal: mensaje codificado por el transmisor.

Canal: medio por el cual las señales son transmitidas al punto de recepción.

Fuente de ruido: distorsiones o adiciones no deseadas por la fuente de información que afectan a la señal que afectan a la señal (de sonido, de imagen, transmisión).

Receptor: decodifica o vuelve a transformar la señal transmitida en el mensaje original haciéndolo llegar a destino.

Proceso de comunicación (niveles de análisis): Técnico (Se analizan problemas de fidelidad de la transmisión del emisor al receptor), Semántico (significado del mensaje e interpretación), Pragmático (Efectos conductuales de la comunicación, influencia y efectividad del mensaje).

Telecomunicaciones: Transmisión electrónica de señales para las comunicaciones por teléfono, radio, televisión. El modelo general es: Sistema de cómputo o equipo emisor envía una señal a otro dispositivo por medio de una transmisión, este mismo envía esa señal a otro sistema de cómputo que es el equipo receptor de la señal.

De tipo medio: sustancia material que pueda transportar una señal para permitir comunicación entre emisor y receptor.

De tipo protocolo: reglas que gobiernan el intercambio de información a través de un medio de comunicación.

Redes: dispositivos conectados a una red para compartir datos, información y tareas de procesamiento. Hay hw y sw que también permite establecer conexiones (módem, sistema operativos de red, sw de administración de redes).

Software de sistemas: controla una o más computadoras, posibilita la interacción con el usuario, administración de archivos y memoria, conexión de redes, interfaz del usuario (entrada y salida de dispositivos).

Software de aplicación: Procesamiento de texto, bases de datos, manipulación de imágenes, administración de información personal o financiera:

* Software de aplicación a medida:

Ventajas: Se obtiene lo que se necesita, se puede controlar resultados e incorporar modificaciones.

Desventajas: Toma tiempo y recursos desarrollar las funciones necesarias, estrés para el personal, y riesgos potenciales.

* Software de aplicación comercial:

Ventajas: Menor costo inicial, mayor calidad asegurada, seguro se cubran todas las necesidades básicas del negocio.

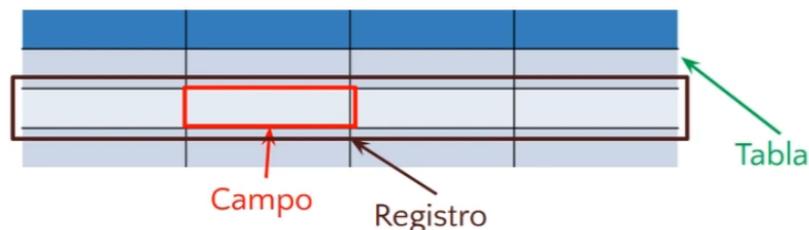
Desventajas: Se pagará por características que no se usen, puede requerir modificaciones costosas.

Sistemas de bases de datos:

Sistemas de administración de bases de datos: grupo de programas que manipulan bases de datos y brindan una interfaz entre esta, los usuarios y otros programas de aplicación. Mantiene la integridad y la seguridad de los datos.

Las **bases de datos** consisten en Campos (bloque básico de información), Registros (conjunto de campos relacionados) y Tablas (conjunto de registros relacionados).

Ventajas: Mejor uso de datos corporativos, protección e integridad de datos, mantenimiento de datos más sencillo, separación de datos y programa.



Desventajas: Mayor complejidad y costo. Dificultad para reparar fallas.

Sistemas de información en organizaciones: con objetivos de lucro (maximizar valor de la inversión) y no lucro (grupos sociales o religiosos, entidades educativas).

Sistema organizacional: Toman como entrada recursos (materiales, personal, dinero), se transforma el producto y se toma como salida bienes y servicios, a lo que se le agrega un valor llamado valor agregado.

Fin de los usos de sistemas de información: Reducir costos y mejorar productividad, ventaja competitiva, y retorno de la inversión. Si se invierte X cantidad, recuperar X y un poco más.

Clasificación de sistemas de información:

Comercio electrónico: Buscar e identificar artículos en venta - seleccionar artículos / negociar precios - pagar - obtener producto - soporte post-venta.

e-Commerce: Transacción de negocios que se lleve a cabo de manera electrónica entre: empresas, empresa y consumidor, consumidores, negocios y sector público.

m-Commerce: Transacciones en cualquier lugar y momento mediante dispositivos móviles, tabletas, etc. Venta de objetos, marketing, finanzas, subastas.

Ventajas: Reducción de costos y tiempos, servicio más ágil, mejora en el servicio al cliente.

Desventajas: Desafíos como definir la estrategia de comercio, enfrentar problemas de privacidad de clientes y crear estrategias de construcción de confianza. Si se busca un comercio global, rediseñar algo apto para todo público y cultura (tiempo, distancia, divisas, leyes). Configurar seguridad en la página para evitar robo de datos, encriptar información y alertar a clientes de phishing. Tener en cuenta el soporte de hw e infraestructura que aguante.

Sistemas Empresariales: Garantiza que la información se pueda compartir a través de todas sus funciones de negocios y todos su niveles de gestión para soportar su operación y administración.

Sistemas de procesamiento de transacciones: para registrar operaciones de negocios (entrada de pedidos, control de inventarios, cuentas por pagar y cobrar).

Las transacciones se procesan: por lotes (se acumulan transacciones por un periodo, y se procesan como unidades o lotes) o por procesamiento en línea (se procesan transacciones de inmediato, no existen demoras).

De planificación de recursos empresariales: conjunto de programas integrados que administra las operaciones de negocios vitales de una compañía. Administración de producción y cadena de suministro, de la relación con el cliente y las ventas. Reemplazan muchos sistemas con un conjunto de programas unificados más eficiente y fácil de usar.

Sistemas de Información y soporte a las decisiones:

Sistema de información administrativa: formado por un conjunto de personas, procedimientos, bases de datos y dispositivos que proporcionan información rutinaria a los administradores y personas que toman decisiones, proporcionan reportes y estándares y están conectados por una misma base de datos.

Sistema de soporte a las decisiones: formado por un conjunto de personas, procedimientos, software, bases de datos y dispositivos que se utilizan para ayudar a tomar decisiones que resuelvan problemas. Le dan soporte a la decisión administrativa, se enfocan en tomar decisiones eficientes. Hay sistemas de soporte a grupos (para toma de decisiones grupales) o sistemas de soporte ejecutivo (para decisiones estratégicas).

Sistemas de información de Negocios especializados:

Administración del conocimiento: conjunto organizado de personas, procedimientos, software, bases de datos y dispositivos utilizados para crear, almacenar, compartir y usar el conocimiento y la experiencia de la organización.

Inteligencia artificial: Duplicar o replicar inteligencia humana. Adquirir características de inteligencia humana (Ej: diagnósticos médicos, asistir en diseño y desarrollo de sistema). Aprenden de la experiencia, manejo de situaciones complejas, resolver problemas con información imprecisa, determinar qué es importante, reaccionar ante situaciones nuevas. Sistemas de aprendizaje automático y redes neuronales (sistemas de cómputo inspirados en el cerebro humano).

Sistemas expertos: permiten que una computadora haga sugerencias y funcione como un experto conocedor de un campo determinado, para ayudar al usuario principiante en su desempeño. Tiene una base de conocimiento que almacena toda la información relevante que utiliza el sistema experto. Justifica como llega a ciertas conclusiones.

Realidad virtual: Simulación de un ambiente que puede experimentarse en tres dimensiones.

Unidad 5 – Computación y sociedad.

Impacto social y personal.

Beneficios: aumento de ganancias, bienes y servicios, mejora en la calidad de vida, facilidad para la comunicación.

Problemas: Desperdicio y errores, crímenes, problemas con la privacidad, conflicto éticos.

Desperdicios: Desechos tecnológicos (hw, sw), gasto de recursos innecesario (construcción, mantenimiento), desperdicio de tiempo, dinero y de energía.

Errores: Por procedimientos inadecuados o por falta de retroalimentación (en entrada o captura de datos, mal manejo de archivos y de salidas, control o capacidad del cómputo inadecuada).

Prevención de errores: Organizaciones deben establecer políticas y procedimientos para identificar los errores y definir cómo controlarlos, deben estar orientados a: adquisición y uso de sistemas (se describen y justifican las adquisiciones). Se enlistan las posibles sistemas y equipos a adquirir y se selecciona entre los ofrecidos.

Políticas: Establecer - Implementar - Monitorear - Revisar

Crímenes: Como herramientas (ciberataques, robo de identidad, juegos de azar) o como objetos (acceso y uso ilegal, malware, robo de información, violaciones de patentes / derechos de autor, estafas).

Prevención de crímenes: Hardware y software especializado para proteger sistema y datos, esquemas de privilegios de acceso, servidores dedicados para distintas aplicaciones, respaldos de datos, auditorías periódicas.

Conflictos de privacidad: Constantemente se generan, recolectan y almacenan datos de las personas. Los sitios web que manejan información delicada deben encontrar la forma de garantizar la seguridad de los clientes e informarlos.

Problemas en el ambiente laboral: Los empleos requieren cada vez más conocimiento en tecnología, las personas temen ser reemplazadas o no aprender a tiempo.

Problemas sanitarios: Estrés ocupacional (necesidad de sentirse productivo), aumenta el sedentarismo, riesgos por usar el equipamiento (sillas, mouse, pantallas).

Prevención: revisar configuración de pantallas, posiciones de escritorio, silla, teclado.

Conflictos éticos: Demanda de centros de datos y procesamiento muy grandes genera consumos de energía muy grandes para operar y refrigerarse. Se contratan empresas que se encargan de la seguridad de datos. Hay gobiernos que controlan la información que aparece en las redes.

Ejemplo: Uso de tecnología para elecciones:

Voto electrónico: incorporación de recursos informáticos en cualquier parte del proceso electoral.

Sistema de recuento automático: Se deben auditar manualmente los resultados, luego del acto eleccionario. Susceptible a ataques y riesgo de perder anonimato.

Sistema de registro electrónico directo: se registra el voto en simultáneo mediante un dispositivo (teclado, botonera o pantalla táctil), registro en memoria del dispositivo.

Recuento inmediato y sin boletas a custodiar, Agilidad VS Seguridad.

Sistemas de votación a través de internet: Mecanismos que permiten emitir sufragio desde computadoras conectadas a internet desde cualquier parte.

Diferentes profesionales definen códigos de ética y conducta profesional. Incluyen reglas morales generales, responsabilidades profesionales específicas, reglas para los líderes, por ejemplo: la ACM.

Aspectos legales:

Propiedad intelectual - Art. 1: La protección del derecho de autor abarcará la expresión de ideas, procedimientos, métodos de operación y conceptos matemáticos pero no esas ideas, procedimientos, métodos y conceptos en sí.

Art 4: Son titulares del derecho de propiedad intelectual el autor de la obra, sus herederos o derechohabientes, los que con permiso del autor traducen, adaptan, modifican, aquellos contratados para elaborar el sistema.

Art 9: Nadie tiene derecho a publicar sin permiso de los autores una producción artística que se haya anotado durante su exposición. Quien haya recibido una licencia para usarlo de los autores, podrá reproducir una única copia de salvaguardia, que deberá estar debidamente identificada y no podrá ser utilizada para otra cosa que para reemplazar el ejemplar original.

Art 55 bis: La explotación de la propiedad intelectual sobre los programas de computación incluirá los contratos de licencia para su uso o reproducción.

Art 57: En el registro nacional de la propiedad intelectual deberá depositar el editor de las obras en el artículo 1, tres ejemplares completos de la obra publicada.

Penas para quien: edite venda o reproduzca una obra inédita o publicada sin autorización de su autor o derechohabientes, quien falsifique obras intelectuales, quien edite, venda o reproduzca con plagio, o el que reproduzca más ejemplares de los autorizados.

¿Por qué registrar software?

Seguridad: probar que existe desde determinada fecha

Prueba de auditoría: presunción de autoría otorgada por el estado en cierta fecha.

Elementos de comparación: en supuestos de plagio y piratería. Se valora en poder judicial.

Protección del usuario de buena fé: El autor de la obra es quien figure en el certificado de registro salvo se pruebe lo contrario.

Para registrarlo depende de si es obra inédita (autores y titulares lo usaron de forma personal o en empresa), obra publicada (puesta en conocimiento público) o contrato de software (licencias de uso y otros).

Protección de datos personales:

Art 1: tiene por objeto la protección integral de datos personales asentados en archivos, registros, bancos de datos, sean públicos o privados destinados a informes, así como también se registra el acceso a las mismas.

Art 2: Datos personales (información referida a personas físicas), datos sensibles (origen racial, étnico, opinión política, religión/filosófica/moral, afiliación sindical, salud y vida sexual).

Datos informatizados: datos personales sometidos al tratamiento o procesamiento electrónico o automatizado.

Están los Responsables de archivo, registro, base o banco de datos (titular de un archivo o registro) y los títulos de los datos (datos personales del titular).

Usuario de datos: quien haga tratamiento de datos, archivos o registros a través de conexión con estos.

Art 5: Tratamiento de datos personales es ilícito cuando el titular no prestó su **consentimiento** expreso e informado (por escrito).

Firma digital: Ley 25.506: Resultado de aplicar a un documento digital un procedimiento matemático que requiere exclusivo conocimiento del firmante, encontrándose ésta bajo su absoluto control, identificando al firmante y detectando cualquier alteración posterior a su firma.

Delitos informáticos:

Art 153: Será reprimido con prisión de 15 días a 6 meses quien acceda indebidamente a algún archivo o comunicación que no le sea dirigida aunque no este cerrado o quien lo desvie de su destino.

Art 153 bis: Reprimido de 15 días a 6 meses el que acceda sin autorización o excediendo la que posea, a un sistema o dato informático de acceso restringido

Art 173: Inciso 16: El que defraudare a otro mediante técnicas de manipulación informática que altere el normal funcionamiento de un sistema informático o transmisión de datos.

Art 183: Reprimido de 15 días a 1 año quien destruye, inutilice o desaparezca una cosa, mueble, inmueble, animal total o parcialmente ajeno.

Se agrega que en la misma pena, incurrirá que alterare, destruyere o inutilizare datos, documentos, programas o sistemas informáticos; vendiere o distribuyere, hiciere circular o introdujere en un sistema informático, cualquier programa destinado a causar daños.

Teletrabajo: Una manera de organizar y realizar el trabajo a distancia con el uso de las tecnologías de la información y comunicación (TICS) en el domicilio del trabajador o en un lugar o establecimiento ajeno al empleador.

Ventajas: Mejora la calidad de vida, ahorra dinero y traslado, más tiempo para actividades extra laborales, facilita inserción de grupos vulnerables, acompaña a mujer durante lactancia, pandemias o catástrofes naturales.

Desventajas: No se garantizan condiciones de trabajo, aísla al empleado, provoca mayor estrés, disminuye productividad, extiende tiempo de trabajo, patrimonio exclusivo del sector privado.

Art 1: Avisar a la ART con información del empleado a implementar teletrabajo.

Auditorías:

Abuso computacional: Cualquier incidente asociado con tecnología de computadoras, en el que una víctima sufre o puede haber sufrido una pérdida y que un perpetrador con intención obtuvo o puede haber obtenido una ganancia.

| TIPOS DE ABUSO | CONSECUENCIAS | |
|----------------------|---------------------------|-----------------------------|
| Hackeo | Destrucción | Daño físico al personal |
| Virus | Hurto | Uso no autorizado |
| Acceso físico ilegal | Modificación | Interrupción de operaciones |
| Abuso de privilegios | Violaciones de privacidad | |



La **auditoría de sistemas de información** es el proceso de recolección y evaluación de evidencia para determinar si un sistema salvaguarda sus activos (hw, sw, instalaciones, documentación, personas), mantiene la integridad de sus datos (completos, sanos, puros, veraces), permite alcanzar sus objetivos de manera efectiva (necesidades del usuario) y usa los recursos de manera eficiente (consumo de recursos).

Control interno: Sistema para conseguir objetivos

Separación de tareas: Iniciar y registrar transacciones, custodiar activos. Determinar correctitud de funcionamiento: separar funciones, ejecutar sw y realizarle cambios.

Delegar: Delegar autoridad y responsabilidades, se asignan dueños a datos.

Reclutamiento y entrenamiento: Poder y responsabilidad depositados en personal responsable de los sistemas incrementa.

Sistema de autorizaciones: Autorizaciones para procedimientos se automatizan (evaluar trabajo de empleados o procesamiento del sistema).

Documentos y registros adecuados: garantizar traza de actividades, bitácora o logs con detalle adecuado.

Control físico sobre los recursos: Centralización de datos lo hace más complejo, consecuencias de pérdida son mayores.

Supervisión gerencial: empleados más cerca de los clientes, más lejos de los supervisores.

Chequeos independientes de performance: detectar olvidos, errores, descuidos, este proceso pierde sentido cuando se automatiza.

Comparar registros con activos: Datos y activos deben ser comparados y chequeados con los registros para detectar inexactitudes.

—

Cambios en la auditoría: Recolección de evidencia (confiabilidad del sistema manual vs computacional, evolución de tecnologías), evaluación de evidencia (es mas difícil hacer trazabilidad de errores, errores de diferentes tipos)