

Dificultades en la producción de st.

- * **Eficiencia** = se refiere a los delays de crear un software, entender bien q' es lo q' los usuarios necesitan, diseñar el st de manera efectiva p' cumplir con los req. (principalmente el el delayo fundamental p' el programador q' va a crear un producto).
- * **Complejidad** = la complejidad q' puede llegar a tener construir un producto st. ya que tiene muchas partes que interactúan entre si de manera diversa, a medida que el software crece en tamaño y funciones, la comp. aumenta, lo q' hace dif. de entender, probar y mantener.
- * **Conformidad** = la necesidad del st de ajustarse a un entorno externo (hardware), S.O. Las demandas de conformidad pueden limitar las opciones de diseño y desarrollo imponiendo restricciones q' complican el proceso de creación de st.
- * **Necesidad de cambios** = el st debe ser capaz de adaptarse a los cambios en los req. de los usuarios.
- * **Invisibilidad** = es intangible lo q' dificulta la comprensión del estado actual del st y la comunicación entre los miembros del eq. de desarrollo. La inv. complica la gestión del proyecto y la detección temp. de problemas.
- * **Compatibilidad**:

Proceso st = es un conj de actividades y tareas que implica el desarrollo y evolución de un Sist. sof.

Modelos Tradicionales (MTs):

- * Cascada,
- * Prototipo,
- * Desarrollo en fases.
- * Modelo espiral
- * Pup.

Métodos ágiles (MA):

- * desarrollo de st adaptativo
- * Scrum
- * XP

Modelos Tradicionales

Modelo cascada

- es el más antiguo.
- se debe completar un estado antes de empezar el siguiente estado.
- es útil p' q' el desarrollador visualice q' va hacer
- no refleja la realidad → su principal problema

Modelos de Prototipación.

- Permite la construcción rápida del sistema
- Usuario y desarrollador tienen una visión común
- Se reduce riesgo e incertidumbre del desarrollo

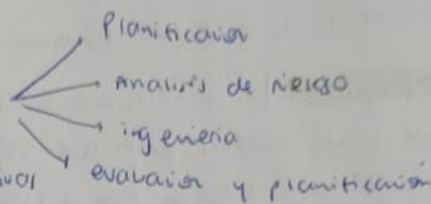
Modelo de desarrollo en 2 fases.

Divide el proceso de desarrollo en 2 etapas cuyo fin es obtener el producto el fin de gestionar y controlar pero puede resultar inflexible ante cambios de req. durante el desarrollo.

Modelo en espiral

= es un modelo iterativo.

en 1ª iteración se evalúa las alt. alternativas y se elige una, los gestores intentan eliminar o minimizar el riesgo.



• RUP

= tiene 4 fases.

→ Planificación, elaboración, construcción y transición.

Se usa preferiblemente el proyecto complejo con equipos grandes.

Promueve una solución disciplinada, que consiste en org. tareas y responsabilidades dentro de una organización. (Pertenece a IBM).

- Basado en el modelo UML.

→ es iterativo. y 1ª iteración resulta en una versión parcial del sistema q' puede ser evaluada y mejorada en ciclos posteriores.

los prob. de los modelos tradicionales

* Incumplimiento de los plazos de entrega.

* Dif. adaptación a los cambios.

* excesiva documentación.

Creado en 2001 tras una reunión celebrada en Utah - IBM

Metodología Ágiles

→ es una metodología que se centra en la entrega

rápida y continua a través de ciclos iterativos e incrementales, es flexible y adaptativa a los requisitos cambiantes por parte de los clientes.

El trabajo se divide en ciclos cortos llamados "Sprints" q' generalmente duran entre 1 y 4 semanas, al final de 1ª iteración se entrega una versión funcional del sf.

Los clientes están involucrados de manera continua en el pro de desarrollo

En la metodología Scrum

es la más utilizada por la metodología ágil, se establecen periodos de tiempo de 15 a 30 minutos de reuniones diarias q' se llevan a cabo todos los días laborales. para tener información de los avances del proyecto, también permite la detección temprana de problemas, facilitando la colaboración y organización entre los miembros del equipo.

3.0. -> Adaptive Software Development -> Diseñado en 2000.

2

La fundamental idea es: "Colocar a un equipo en medio del caos".

Buscamos un equilibrio entre ^{crear} buscar un ambiente que fomente la creatividad e innovación y administrar lo que está pasando.

Lo que no es utilizada pq no se especifican en detalle las prácticas del desarrollo.

(XP) Programación Extrema -> Desarrollado en 1999.

Es un conj de prácticas ya conocidas pero combinadas de manera innovadora al igual que nueva metodología.

Se conoce como "extrema" pq lleva a niveles extremos de aplicación.

-> Se conoce por sus prácticas innovadoras y su énfasis en la colaboración y satisfacción al cliente. Se enfoca en hacer el trabajo de la manera más simple posible, programación en pares. (los desarrolladores trabajan en pares delegando el trabajo del otro, para garantizar q el trabajo siempre sea cubierto).

El 38% del mercado ágil utiliza "XP".

SCRUM = es el más utilizado de los enfoques ágiles.

Los Roles: ✓ Product Owner

✓ Scrum Master

✓ Team (eq. de desarrolladores)

es el encargado de definir y priorizar las características y requisitos del producto.

Representa los requisitos del producto. tmb los intereses del cliente.

Es el guía del equipo scrum, asegurando que sigan las prácticas, ayuda a ^{eliminar} impedimentos y promueve la colaboración efectiva dentro del equipo.

Ventajas de MAS:

✓ Trabajo en equipo.

✓ Se adapta a los requisitos cambiantes.

✓ Límite de tiempo para el ciclo (sprint) de 2 a 3 semanas.

✓ Iteraciones en ciclos cortos q permiten correcciones y verif. más rápidamente.

Desventajas

✓ La flexibilidad de poder adaptarse al cambio en vez de seguir estrictamente un plan puede llevar a caos en el proceso de desarrollo.

✓ Det. las prioridades es difícil cuando hay múltiples personas o empresas involucradas en el proy.

✓ El hecho de mantener la simplicidad puede ser un trabajo extra.

Conceptos de calidad

cap 7

Hay 2 tipos de calidad \rightarrow Calidad de diseño y calidad de concordancia.

* **Calidad de diseño**: son las características que los ingenieros especifican que va tener el producto sf; la tolerancia a fallos y eficiencia, velocidad contribuyen a la calidad del diseño.

* **La calidad de concordancia**: es el cumplimiento de las especificaciones durante el diseño del sf.

Que es la calidad de software = es la concordancia con los requisitos funcionales especificados / establecidos con la documentación y características que se espera de todo sf desarrollado profesionalmente.

Actividades del manejo de la calidad =

- ✓ Aseguramiento de la calidad = establecer procedimientos pl la calidad.
- ✓ Planificación de la calidad = procedimientos aplicables y estándares pl un proyecto en particular y modif. ellos como van requeridos
- ✓ Control de calidad = garantizar q' los procedimientos son seguidos por el eq. de desarrollo.
- ✓ El manejo de calidad debe ser separado del manejo del proy. pl aseg. independiente

Garantía de sf = conjunto de acciones pl garantizar la calidad del sf.
con las tres actividades:

- Aplicación de normas técnicas, realizar revisiones, pruebas de sf, control de cambios, medición, registro y realización de informes.

Atributos de la calidad del sf = seguridad, Fiabilidad, Flexibilidad, Robustez, adaptabilidad, complejidad, eficiencia, facilidad, portabilidad, viabilidad.

mejora de procesos

cap 8.

ISO/IEC 9126.

Los aspectos de la calidad: interna, externa, en uso.

Interna = código fuente.

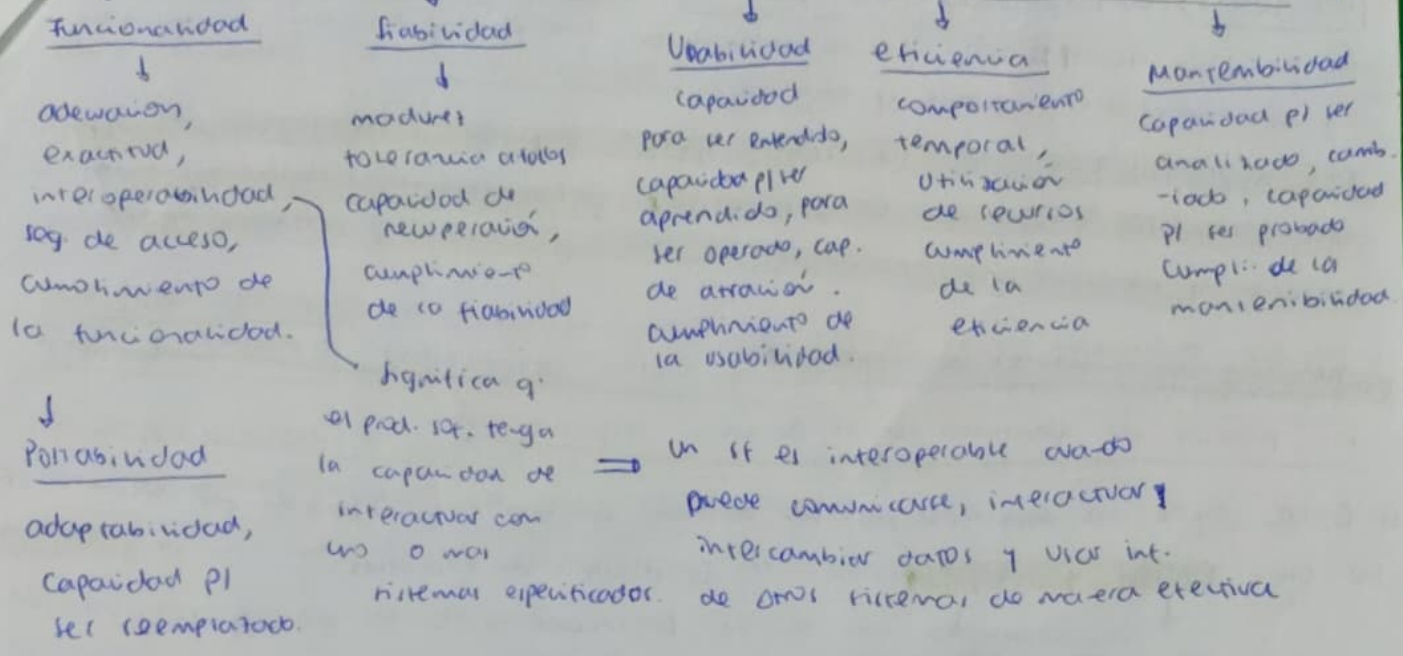
Externa = aquel q' mide el comportamiento del producto, como en una prueba

En uso = la utilización ^{efectiva} por parte del usuario.

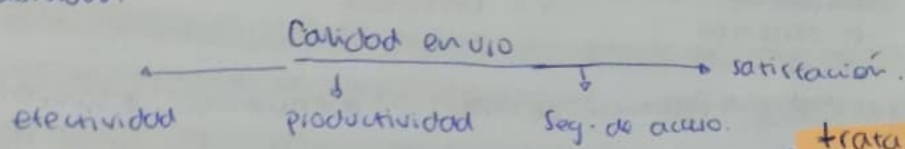
El m. estándar q' se enfoca en ciertos problemas comunes q' pueden influir negativamente en el proyecto y desarrollo sof.

Este estándar ayuda a asegurar la funcionalidad, fiabilidad y facilidad de uso.

Calidad externa e interna



⇒ Un sf es interoperable cuando puede comunicarse, interactuar y intercambiar datos y usar int. de otros sistemas de manera efectiva



ISO/IEC 14598

⇒ Proceso de evaluación.

- * establecer requisitos de evaluación.
- * especificar evaluación.
- * Diseñar evaluación.
- * ejecutar evaluación.

trata sobre la evaluación de la calidad de sf.
es una expansión de la norma ISO/IEC 9126, proporciona marco y guía pl medir la calidad de los productos sf a lo largo de su ciclo de vida.

MARCO DE TRABAJO = la finalidad es mejorar los procesos de sf., mejorar evaluaciones pl determinar la potencialidad y el rendimiento de los procesos.
Los MT se dirigen bien la misma "dirección"
Buscan: mejorar los procesos de sf., aumentar la productividad y la calidad
Disminuir el costo

ISO/IEC 12207

⇒ es la 1ª norma q' describe en detalle los procesos, actividades y tareas q' envuelven el desarrollo, operación y mantenimiento de Prod. sf.
Detalla los diversos procesos envueltos en el ciclo de vida del sf.
estos procesos estan divididos en 3 clases = Procesos Fundamentales, Procesos de apoyo y Procesos Organizacionales.

ISO 9003

Proporciona una guía a las organizaciones para la adquisición, suministro, desarrollo, instalación y mantenimiento del IT y serv. de soporte.

La implementación de ISO 9003 puede ayudar a las organizaciones a mejorar la eficiencia de los procesos, aumentar la satisfacción del cliente y asegurar la calidad de los productos y servicios de IT.

Modelos de madurez de capacidad = Ruta desarrollada al evaluar y mejorar el proceso de Desarrollo de IT en una organización.

A CMM mide la madurez de los procesos en una organización de desarrollo de IT.

Obj: Evaluar y mejorar

Se divide en 5 niveles:

↓
Evaluar los procesos de desarrollo

↓
mejora la eficiencia y calidad en el desarrollo.

1 - inicial

5 - optimizado

2 - Repeatible

3 - Definido

4 - Gestionado

Beneficio = mejora la calidad * Reducción de riesgo * Eficiencia * Competitividad

CBA - IDI (Evaluación basada en CMM para la mejora de procesos internos)

Es mediante la identificación de sus fortalezas y debilidades, con el fin de mejorar el IT.

Consiste en evaluar los procesos de IT mediante un grupo de profesionales externos que trabajan como equipo.

PSF (Proceso de IT personal) = es un método desarrollado para ayudar a los ing. de IT a mejorar sus habilidades y la calidad de su trabajo mediante el enfoque redible y disciplinado.

Objetivo = Ayuda a los ing. a la mejora continua de la calidad del IT y la productividad del ingeniero.

Ayuda a planificar de manera precisa y eficiente sus proyectos.

Analiza si los cambios afectan el rendimiento y buscan formas de optimizarlos.

TSP (El proceso de IT en equipo) = Metodología diseñada para mejorar el trabajo en equipo en PROJ de IT, se enfoca en organizar equipos para trabajar de manera efectiva y producir IT de alta calidad.

Obj = mejora la calidad del IT ya que los equipos producen IT con menor error, incrementa la productividad, fomenta el trabajo, mejora la colaboración y comunicación dentro del equipo, tiempos de entrega más precisos.

Básicamente ayuda a planificar, gestionar y mejorar sus proyectos de manera efectiva enfocándose en la calidad, productividad y trabajo en eq.

4
TQM = es un marco de trabajo p/ mejorar procesos de una organización y garantizar que sigan de manera consistente y eficiente.

Obj. = productos de Alta calidad, eficiente y efectivo, Minimizar los riesgos y operaciones

Instrumenta p/ mejorar y optimizar los procesos de una organización, ayudándola a ser más eficiente, reducir riesgos y mantener una alta calidad en sus productos y servicios.

ISO 9001:2015 proporciona un marco de trabajo p/ la evaluación del proceso y establece los requisitos mínimos p/ realizar una evaluación que asegure la repetibilidad y consistencia de las valoraciones obtenidas

su objetivo es mejorar la capacidad de los procesos de una organización

ISO 9001 es un modelo q' proporciona directrices y buenas prácticas p/ mejorar los procesos de desarrollo de SI de una organización.

Objetivo:

✓ Mejora la calidad del software ✓ Incrementa la eficiencia ✓ Facilita la gestión

Basicamente es una guía p/ mejorar los procesos de desarrollo de SI, ayudando a las organizaciones a ser más eficientes, a producir SI de mejor calidad y a gestionar sus proy. de manera efectiva.

Cambios en el SI

Unidad 9

Estrategia de cambio de SI

- Mantenimiento de SI = es el proceso de actualizar y mejorar un SI a respuesta a nuevas necesidades o requerimientos, pero sin alterar la estructura básica del SI.
- Transformación de Arquitectura = es modificada de una arquitectura centralizada a una distribuida.
- Reingeniería de SI = es mejorarlo pero sin añadirle nuevas funcionalidades de reestructura y reorganiza el código p/ q' sea más fácil de entender y modificar en el futuro.

Dinámica de evolución de programas = es el estudio de los procesos de los cambios de sistema.

Lehman y Belady propusieron "leyes" q' aplicaban a todos los sistemas conforme estos evolucionaban

las leyes de Lehman son:

- * Cambio continuo.
- * Complejidad creciente
- * Crecimiento continuo.

* Cambio continuo: es que debe ser actualizada en si esto no ocurre eventualmente se volvera nepot util en el futuro.

* Complejidad creciente: A medida q' el problema crece, su estructura tiende a ser mas compleja

* Crecimiento continuo: las funcionalidades ofrecidas deben crecer continuamente el mantener a los usuarios satisfechos

Aplicabilidad de las leyes de Lehman: son granmente aplicables a sistemas grandes y levas a medida, desarrollados por grandes organizaciones

Mantenimiento de SI: no involucra grandes cambios, simplemente se modifican componentes ya existentes y se agg nuevos comp. al sistema

El mantenimiento es INEVITABLE. => significa que todo sistema SI necesita ser actualizado y ajustado logicamente, esto se debe a los requerimientos de los usuarios, la aparicion de nuevos errores...

Tipos de mantenimiento:

- ✓ Mantenimiento al reparar fallos: para corregir deficiencia de manera a cumplir ^{los requerimientos}
- ✓ Mantenimiento al adaptar el software a un S.O distinto. = cambiar un sistema al que opere en un S.O distinto al de su implementación.
- ✓ Mantenimiento al agg o modificar funcionalidad del sistema.

Soft. evolutivo: es aquel soft. que va evolucionar continuamente durante toda su existencia

La complejidad depende del = la estructura de datos, estructura de datos, tamaño de procedimientos y modulos.

Unidad 10.

Que es la reingenieria. = es el proceso de rediseñar, reestructurar, reestructurar completamente un sistema al mejorar su rendimiento, eficiencia y adaptabilidad implica revisar el código y la documentación, sin añadir nuevas funcionalidades con el objetivo de hacerlo más comprensible, mantenible y adaptable a futuros cambios.

Ingeniería inversa => basicamente es tomar un producto software terminado descomponerlo al entender como funciona, esto significa analizar el código al entender su estructura, funcionamiento y logica con el obj de mejorar dicho software.

Reestructuración = la reestructuración de lugar a prog. de mayor calidad, mejor documentación, reduce la frustración de los ing. de sot., mejora la productividad, reduce el estrés y el mantenimiento. Hace q' el st sea más sencillo de comprender y depurar.

Reestructuración del código = se lleva a cabo p' construir un diseño q' produce la misma función pero con una mayor calidad q' el prog. original.

Ingeniería Progresiva = es mejorar el sistema de manera progresiva, esto quiere decir agregar nuevas funcionalidades, mejorar las existentes, realizar ajustes p' q' se adapten a las nuevas tecnologías, asegurando q' el sistema evolucione y crezca con el tiempo.

Unidad 12

Práctica = son los conceptos, principios, métodos y herramientas p' elaborar un st de alta calidad.

Unidad 13

Tecnología disruptiva = son innovaciones q' cambian drásticamente la forma en que hacemos algo, ofrecen soluciones más asequibles, eficientes o económicas. Ejemplos de T.D = internet, smartphones, IA.

Por ej cuando los smartphones aparecieron, cambiaron la forma en q' utilizamos los teléfonos, accedemos a internet y hacemos otras actividades, esta tecnología puede hacer que los viejos métodos o productos se vuelvan menos útiles.

TAA

Describe las fases de desarrollo del modelo XP, y al menos 5 de sus prácticas.

Las fases de desarrollo

Prácticas

- Planificación
- Análisis
- Diseño
- Codificación
- Testeo.

- Programación en pareja = los desarrolladores trabajan en Pareja, chequeando el trabajo del otro para garantizar q' el trabajo siempre sea de buena calidad.
- Planificación incremental = el desarrollo se realiza en pequeñas etapas o incrementos (el trabajo se divide en iteraciones).
- Entregas pequeñas = consiste en pequeñas entregas en ciclos cortos y frecuentes, en lugar de esperar a completar todo el proyecto y hacer una entrega, liberándose así de pequeñas partes del proyecto, esto es útil p' la circulación temprana de errores.
- Diseño simple = solo se lleva a cabo los detalles necesarios p' cumplir con los requerimientos.
- Refactorización = proceso de mejorar y mejorar el código p' q' se mantenga simple y mantenible.

- a) Menciona las diferencias y similitudes entre las metodologías Scrum y XP como Scrum manejan bloques de tiempo, pero en XP esto es más reducido que en Scrum.
- ✓ En la metodología XP se promueve el trabajo en pares, mientras que en Scrum se fomenta el trabajo autónomo, y quien que tiene su cuenta propia.
- ✓ En la metodología Scrum se organizan los tareas de acuerdo a la prioridad en el producto backlog, mientras que en XP se organiza de acuerdo a lo que dice el cliente.

Que son las ventajas y desventajas que surge cuando se aplican las met. ág.

Ventajas

- Iteraciones cortas que permite corrección y verificaciones más rápidamente.
- Límite de tiempo de 2 a 3 semanas.
- Se adapta a nuevos riesgos.
- Estilo de trabajo en equipo.

Desventajas

- Su flexibilidad de adaptarse a los cambios puede llevar a caos en el proceso de desarrollo.
- Difícil de mantener el interés de los clientes que participan en el proceso de desarrollo.
- Def. prioridades es difícil cuando existen múltiples empresas o personas involucradas en el proyecto.
- Mantener la simplicidad es un trabajo extra.
- Definir contratos puede ser un problema.

De qué manera inciden los cambios tecnológicos en los procesos de negocios de las empresas?

Los cambios tecnológicos, generados por la tecnología disruptiva, afectan a los procesos de un negocio en muchas ocasiones obligando a la empresa a adaptarse a nuevas tecnologías, con el fin de mantenerse en competencia y no perder el valor del negocio.

Que se entiende por marco de trabajo? es un conjunto de metas para facilitar a una organización la aplicación de las mismas, mediante los marcos de trabajo se permite que todo el personal de una organización se dirija en la misma "dirección".

Una finalidad es mejorar los procesos de st, determinar la potencialidad y rend. de sus procesos y la madurez de la organización.

¿Escala de nivel de madurez de procesos se utiliza en la norma ISO-15504?

6

Proporciona un marco de trabajo y la evaluación del proceso que establece los requisitos mínimos para realizar una evaluación que asegure la repetibilidad y consistencia de los valores obtenidos.

Contra de los 5 niveles.

Nivel 5: Proceso optimizable.

Nivel 4: Proceso predecible.

Nivel 3: Proceso establecido.

Nivel 2: Proceso gestionado.

Nivel 1: Proceso Realizado.

Nivel 0: Incompleto.

NoProSoft (Modelo de Procesos para la industria de ST. es un modelo para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas y prod. de ST.

¿Cuál es su visión de la ing. de ST en la prox. década?

En la prox. década probablemente la ing. de ST sea más automatizada, segura, colaborativa y con fuerte énfasis en la calidad y ética.

Explique: a) causa de los retrasos de los proyectos de desarrollo de ST.

✓ Tema poco realista

✓ Req. cambiantes por el cliente

✓ Dif. técnicas o humanas.

b) causa de la planificación optimista:

será más integrado con herramientas de soporte de tareas de programación.

Considero que la complejidad de los sistemas continuará en aumento,

ya que al dejar algunas tareas el cargo de nuevas herramientas de soporte los desarrolladores se enfocarán en tareas más complejas.

da el del grato. ejercicio 3 a osmani le salio 7
 go tp Ruben Lucas Yo Fabi

```

do while
  sele
  seek
  if found ()
    rule
      seek
      if found () &
      Camtem
      sele j
      seek h
      if found ()
        sele i
        seek
        if found ()
          replace
          replace
          seek
          seek
          if found ()
            rep
            rep
          end if ()
        end if ()
      end if
    end if
  end if
  seek
end do
skip
end do.

```

Nodos: 26

Arroyos: 31

P = 6.

Li = 6

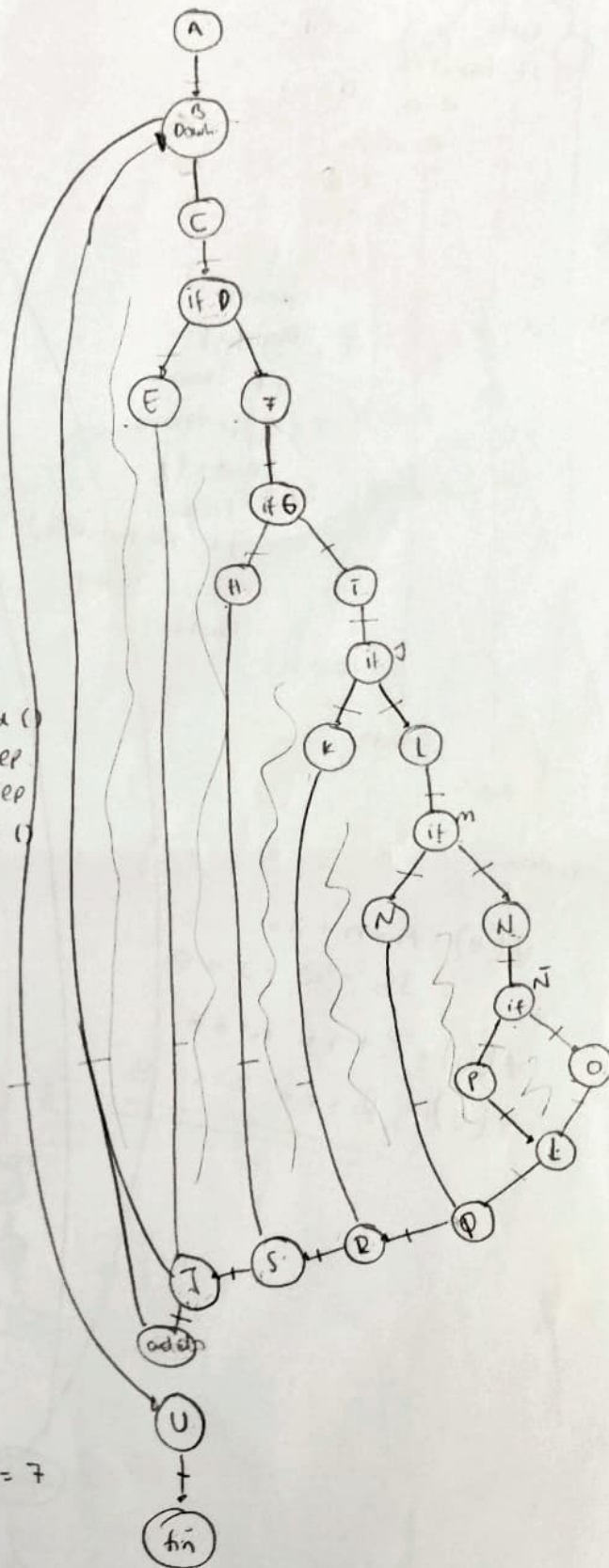
P = nodos predicados =
 (by: fiden 2 omis
 Arroyos: saberes)

Pi = Reg interior $\hat{=}$

$$V(b) = A - N + 2 = 31 - 26 + 2 = 7$$

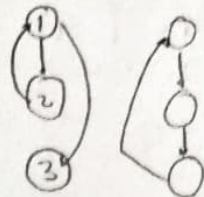
$$V(b) = P + 1 = 6 + 1 = 7$$

$$V(b) = P_i + 1 = 6 + 1 = 7$$



Ques 8

A mini table 7, a tab = 8



Do while
Salab
for k

```

i = a;
if (i < 0) {
  if () {
    ...
  }
  else {
    ...
  }
}
endif

```

```

endif
i = 1
for ()
  rep...
endfor
if ()
  store
  for ()
    ...
  endfor
endif

```

endif

endfor

enddo

$$n = 26$$

$$a = 32$$

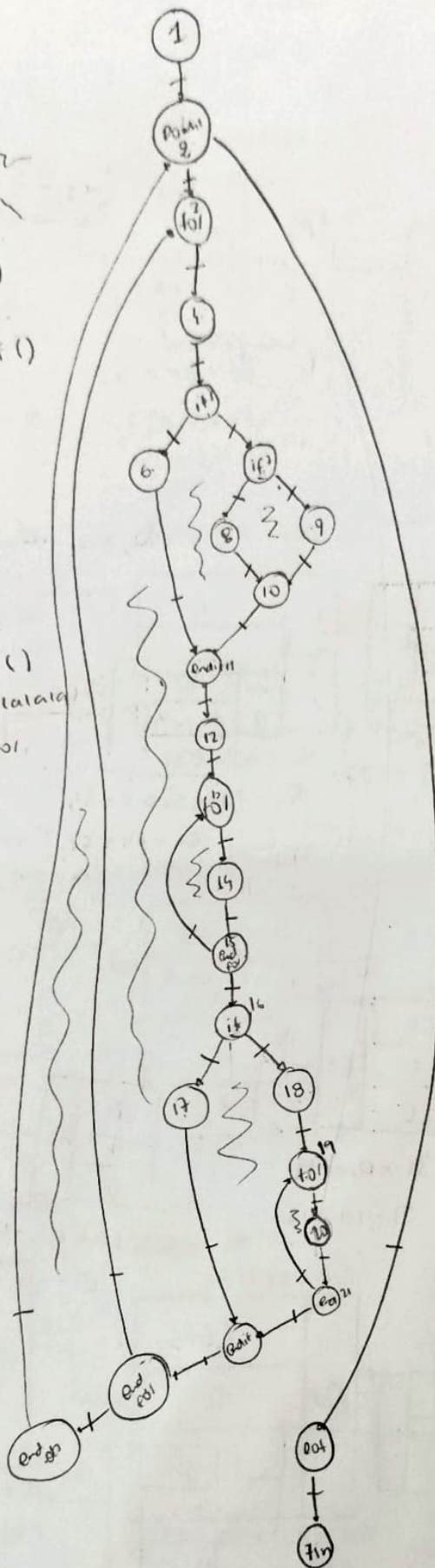
$$P = 7$$

$$B_i = 7$$

$$V(b) = A - n + 2 = 32 - 26 + 2$$

$$V(b) = P + 1 = 8$$

$$V(b) = P + 2 = 9$$



$$IMS = \frac{MT - (Ta + Tm + Te)}{MT}$$

Para A = Mi = 40.

$Ta = 10$

$MT = Mi + Ta - Te$

$Tm = 5$

$MT = 40 + 10 - 7$

$Te = 7$

$MT = 43$

$$IMS = \frac{MT - (Ta + Tm + Te)}{M}$$

M

$$IMS = \frac{43 - (10 + 5 + 7)}{43}$$

43

$0,48$

$$MT = Mi + Ta - Te$$

Modulo
inicial

1^{er} ejercicio.

0	20
0	
40	

$40 \times 0,2 = 8$

$40 - 8 = 32$

10	10
$2 \times 10 = 48$	
0	

$64 \times 0,1 = 7$

$64 - 7 = 57$

$57 \times 0,7 = 40$

40	50
$17 \times 2 = 34$	
50	

$124 \times 0,5 = 62$

$124 - 62 = 62$

62	50
0	
0	

$62 \times 0,5 = 31$

$62 - 31 = 31$

31	60
0	
0	

$31 \times 0,6 = 19$

$31 - 19 = 12$

12	30
0	
0	

$12 \times 0,3 = 4$

$12 - 4 = 8$

En revision

	0
40	

20	0
$8 \times 20 = 60$	
0	

90

50	50
24	
$24 \times 2 = 48$	

154

154	

154	

154	

$$\frac{154 \times 100}{8} = 1925$$

sería mayor en un 1925%

Origi: Caso la fuente
o actividades?

1) Diseño Preliminar

0	20%
0	
40	

$$40 \times 0,2 = 8$$

$$40 - 8 = 32$$

Diseño detallado

16	40%
$3 \times 16 = 48$	
0	

$$16 + 48 = 64$$

$$64 \times 0,1 = 7$$

$$64 - 7 = 57$$

codificación

40	50%
$17 \times 2 = 34$	
50	

$$124 \times 0,5 = 62$$

$$124 - 62 = 62$$

Prueba de validación

62	50%
0	
0	

$$62 \times 0,5 = 31$$

$$62 - 31 = 31$$

Prueba de integración

31	60%
0	
0	

$$31 \times 0,6 = 19$$

$$31 - 19 = 12$$

Prueba del sistema

12	30%
0	
0	

$$12 \times 0,3 = 4$$

$$12 - 4 = 8$$

2) Diseño Preliminar

0	0%
0	
120	

Diseño detallado

39	30%
$2,3 \times 81 = 187$	
0	

$$226 \times 0,3 = 68$$

$$226 - 68 = 158$$

codificación

64	72%
$94 \times 2 = 188$	
150	

$$402 \times 0,42 = 169$$

$$402 - 169 = 233$$

Prueba de validación

233	40%
0	
30	

$$263 \times 0,4 = 106$$

$$263 - 106 = 157$$

Prueba de integración

157	60%
0	
25	

$$182 \times 0,6 = 110$$

$$182 - 110 = 72$$

Prueba del sistema

72	80%
0	
60	

$$132 \times 0,8 = 106$$

Sin Revisión

Diseño Preliminar

0	01
0	
120	

120

Diseño detallado

39	01
187	
0	

226

226

codificación

91	01
$135 \times 2 = 270$	
150	

511

511

Prueba de validación

511	01
0	
30	

541

Prueba de integración

541	01
0	
25	

566

Prueba del sistema

566	01
0	
60	

626

626

$$x = \frac{626 \times 100}{26} = 2407,69$$

Respuesta = El porcentaje sería 2407,69% mayor la cantidad de errores

5) Diseño Preliminar

0	101
0	
16	

14

Diseño detallado

5	30%
$9 \times 2,5 = 23$	
0	

19

codificación

16	301
$3 \times 3 = 9$	
30	

38

$$16 \times 0,1 = 2$$

$$16 - 2 = 14$$

$$28 \times 0,3 = 9$$

$$28 - 9 = 19$$

$$55 \times 0,3 = 17$$

$$55 - 17 = 38$$

Prueba de validación

38	50%
0	
0	

19

Prueba de integración

19	60%
0	
0	

7

Prueba del sistema

7	70%
0	
0	

2

$$38 \times 0,5 = 19$$

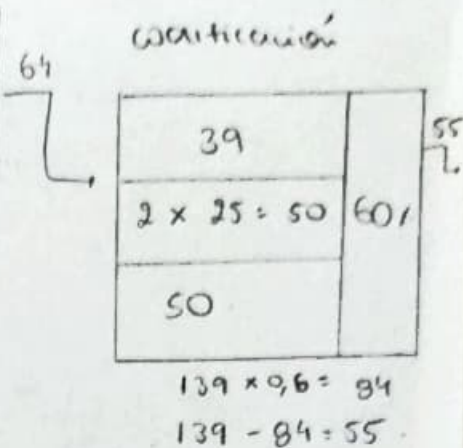
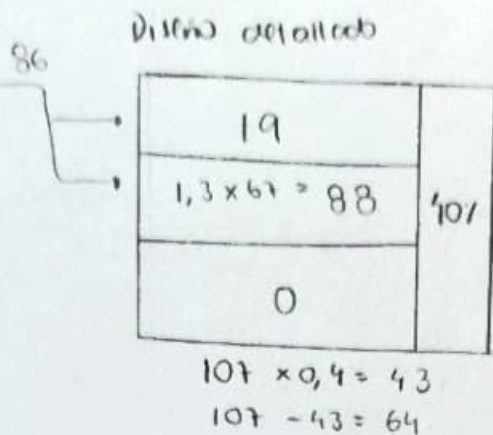
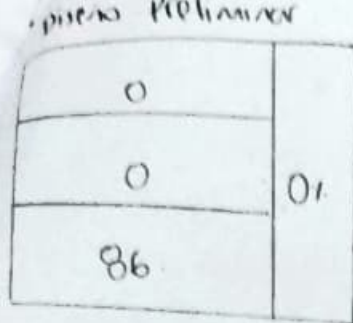
$$38 - 19 = 19$$

$$19 \times 0,6 = 12$$

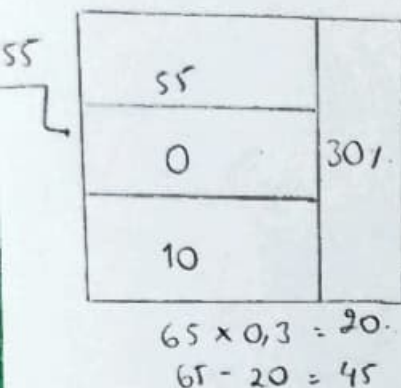
$$19 - 12 = 7$$

$$7 \times 0,7 = 5$$

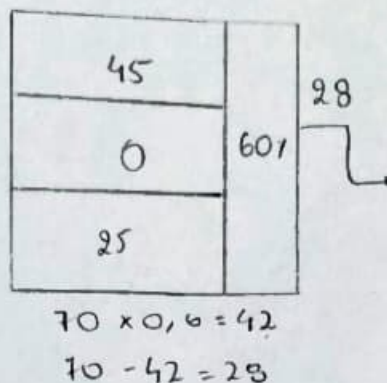
$$7 - 5 = 2$$



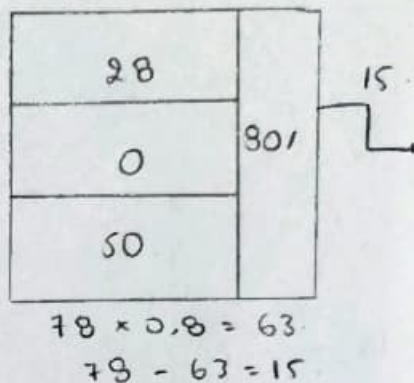
Prueba de validación



Prueba de integración

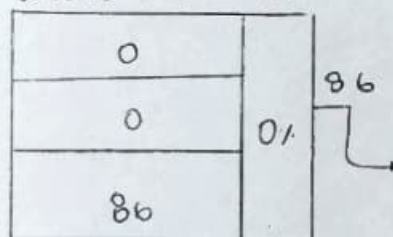


Prueba del sistema

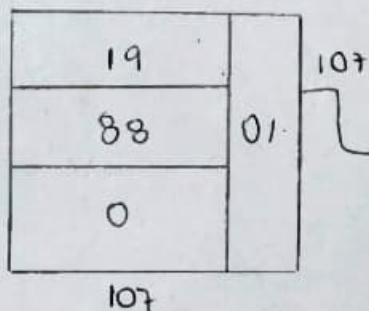


Sin revisión

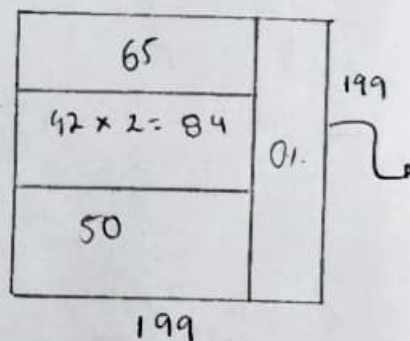
Diseño Preliminar



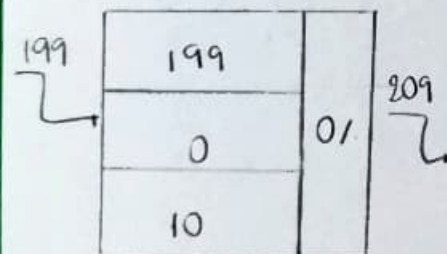
Diseño detallado



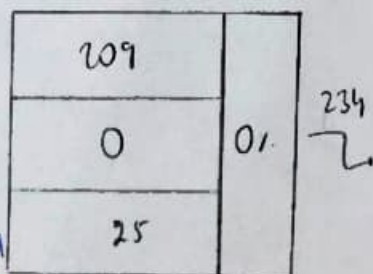
Codificación



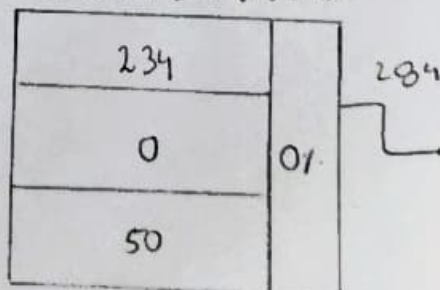
Prueba de validación



Prueba de integración



Prueba del sistema



Respuesta: de no utilizarse el modelo de multiplicación de defectos, el proyecto final tendría 284 errores

Calcular el IMS del sistema A.

$$\text{Minical} = 40 \quad \text{IMS} = \frac{MT - (T_a + T_m + T_e)}{MT}$$

$$T_a = 10$$

$$T_m = 5$$

$$T_e = 7$$

Calcular el MT

$$\text{IMS} = \frac{43 - (10 + 5 + 7)}{43} =$$

$$MT = \text{Minical} + T_a + T_e$$

$$\text{IMS} = 0,4883 \rightarrow \text{El usuario debe hacer una revisión}$$

$$MT = 40 + 10 + 7 = 43$$

Calcular el IMS del sistema B.

$$\text{Minicales} = 131$$

$$\text{IMS} = \frac{MT - (T_a + T_m + T_e)}{MT}$$

$$T_m = 25$$

$$T_e = 7$$

$$\text{IMS} = \frac{124 - (25 + 7)}{124} = 0,7419$$

$$MT = \text{Me} + T_a + T_e$$

$$MT = 131 - 7 = 124$$

No hace falta hacer una revisión

Datos

Errores latentes	40
Errores omplificados	
Nuevos errores	

Deposición de errores

$$V(b) = A - n + 2 =$$

$$V(b) = P + 1 =$$

$$V(b) = R_i + 1 =$$

P = nodos predicados

R_i = Reg. interna.

Para calcular el porcentaje de errores

$$X = \frac{\log' \text{ solo sin rev.} \times 100}{\log' \text{ solo con revisión}}$$

