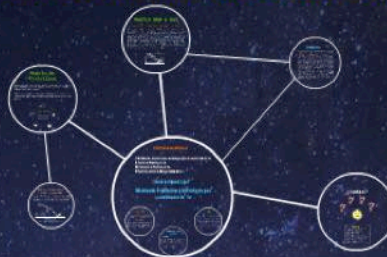


UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO



CENTRO UNIVERSITARIO UAEM TEMASCALTEPEC

ELABORÓ:

LICENCIATURA EN INFORMÁTICA ADMINISTRATIVA

M. EN T. I. GISELA REGINA BAENA CASTRO

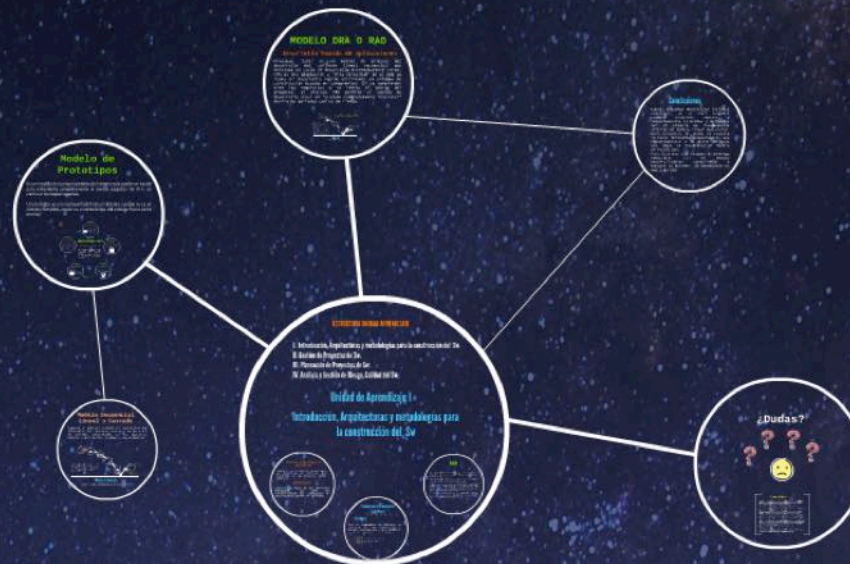
INGENIERÍA DE SOFTWARE

UNIDAD DE COMPETENCIA I:

Introducción, Arquitecturas y Metodologías para la construcción del Sx

FECHA: 14 DE AGOSTO DE 2016

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO



CENTRO UNIVERSITARIO UAEM TEMASCALTEPEC

LICENCIATURA EN INFORMÁTICA ADMINISTRATIVA

INGENIERÍA DE SOFTWARE

UNIDAD DE COMPETENCIA I:

Introducción, Arquitecturas y Metodologías para la construcción del Sw

ELABORO:

M. EN T. I. GISELA REGINA BAENA CASTRO

FECHA: 14 DE AGOSTO DE 2019

ESTRUCTURA UNIDAD APRENDIZAJE

- I. Introducción, Arquitecturas y metodologías para la construcción del Sw.
- II. Gestión de Proyectos de Sw.
- III. Planeación de Proyectos de Sw.
- IV. Análisis y Gestión de Riesgo, Calidad del Sw.

Unidad de Aprendizaje I

Introducción, Arquitecturas y metodologías para la construcción del Sw

Objetivo de la Unidad de Aprendizaje

Conocerá las diferentes metodologías de desarrollo de software para distinguir en que proyectos pueden aplicarse.

Justificación

conocer las fases de los diferentes Paradigmas, así como sus características para identificar en que proyectos pueden ser aplicados

GUIÓN

Los datos presentados solo contienen ideas principales e ideas que reducen la información, estas serán explicadas con el docente al interior de la clase.

Para que el estudiante pueda ampliar sus conocimientos, se incluye la siguiente lista de referencias bibliográficas y lista de recursos.

Así mismo solo contiene información sobre Unidad I, Introducción, Arquitecturas y metodologías para la construcción del software perteneciente al tema Paradigmas de Software, Modelo Línea Secuencial, Modelo de Prototipos y Modelo R4D.

Una vez terminada esta presentación el estudiante tendrá la base para desarrollar sus propias conclusiones de los paradigmas de desarrollo para aplicarlo en su proyecto final.

Paradigmas de la Ingeniería de Software

(Pressman, 2015)

Paradigma:

Para la Ingeniería de Software el paradigma es una agrupación de métodos, herramientas y procedimientos con el fin de describir un modelo.

Contenido

- Modelo secuencial lineal
- Modelo de prototipos
- Modelo R4D

Objetivo de la Unidad de Aprendizaje

Conocerá las diferentes metodologías de desarrollo de software para distinguir en que proyectos pueden aplicarse.

Justificación

conocer las fases de los diferentes Paradigmas, así como sus características para identificar en que proyectos pueden ser aplicados

GUIÓN

Las slides presentadas solo contienen ideas principales e imagenes que refuerzan la información, estas serán explicadas por el docente al interior de la clase.

Para que el estudiante pueda ampliar sus conocimientos, se incluye un apartado de referencia bibliográficas y links de acceso.

Así mismo solo contiene información de la Unidad I, **Introducción, Arquitecturas y metodologías para la construcción del Sw**, específicamente el tema Paradigmas del Software: Modelo Lineal Secuencial, Modelo de Prototipos y Modelo RAD.

Una vez terminado esta presentación el estudiante tendrá las bases para desarrollar sus propias conclusiones de los paradigmas de desarrollo para aplicarlo en su proyecto final

Paradigmas de la Ingeniería de Software

(Pressman, 2015)

Paradigma:

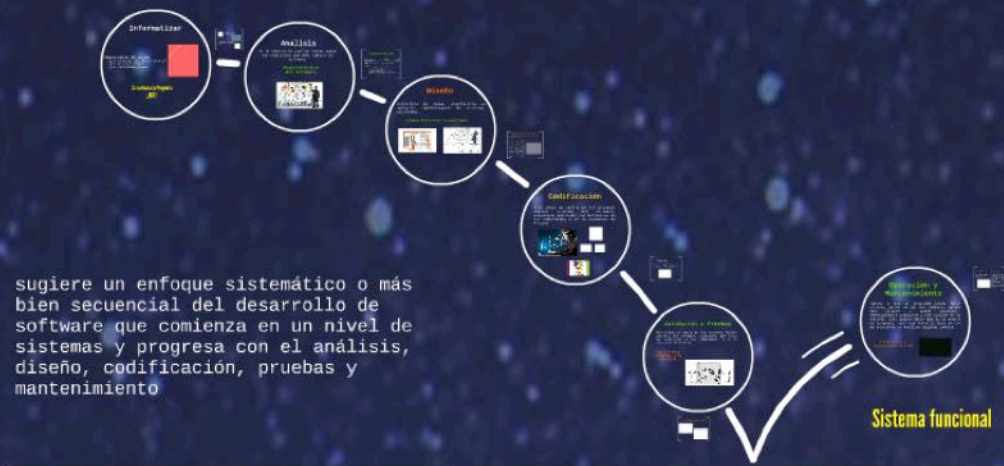
Para la Ingeniería de Software el paradigma es una agrupación de métodos, herramientas y procedimientos con el fin de describir un modelo.

Contenido

- Modelo secuencial lineal
- Modelo de prototipos
- Modelo RAD

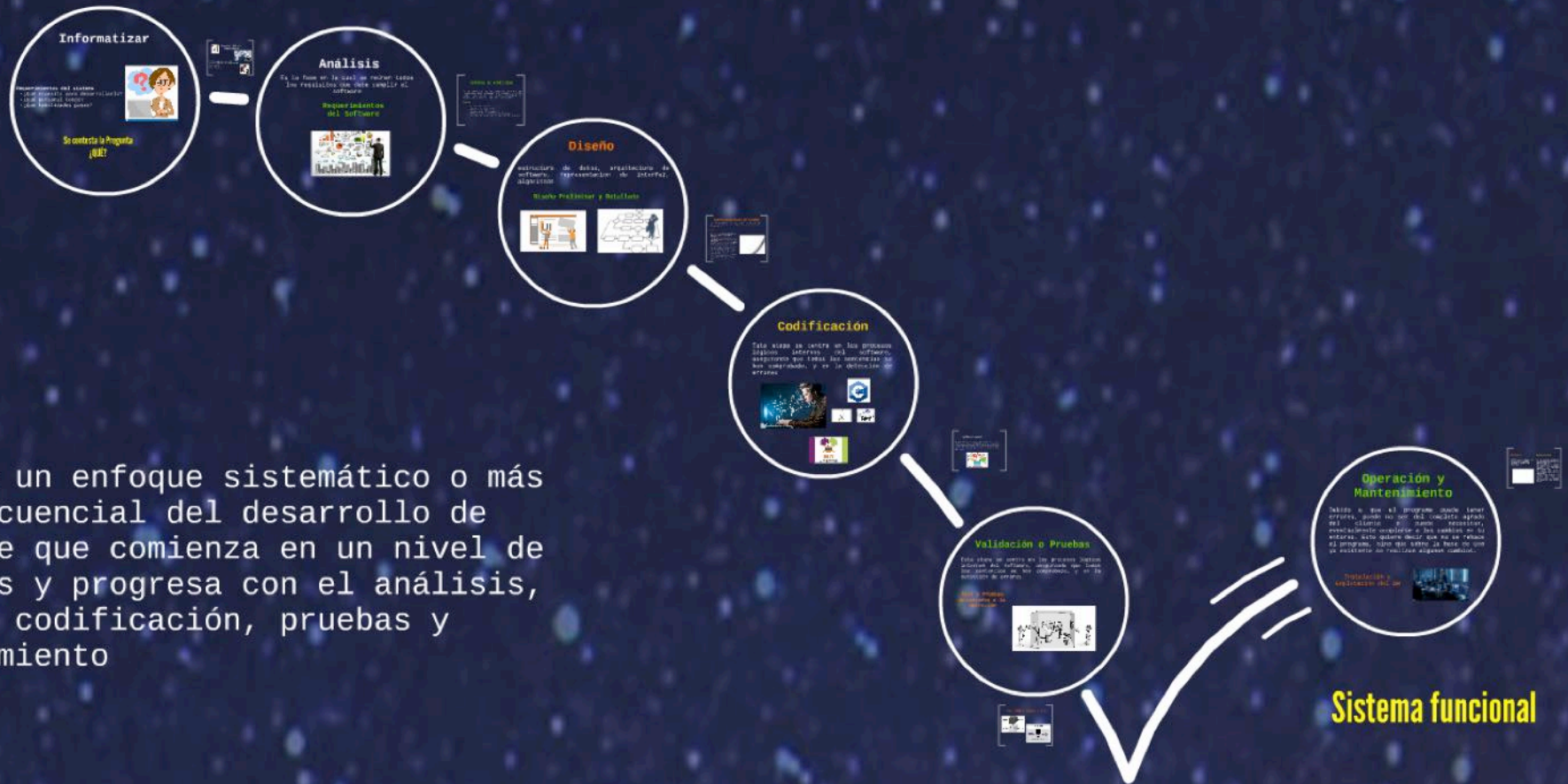
Modelo Secuencial Lineal o Cascada

Sugiere un enfoque sistemático secuencial del desarrollo del software, comienza en un nivel de sistema, progresando con el análisis, diseño, codificación, pruebas y mantenimiento



Modelo de Cascada

Desarrollo atribuido a Royce en 1970



sugiere un enfoque sistemático o más bien secuencial del desarrollo de software que comienza en un nivel de sistemas y progresa con el análisis, diseño, codificación, pruebas y mantenimiento

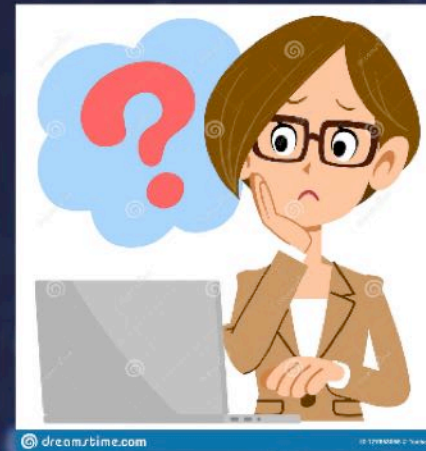
Modelo de Cascada

Desarrollo atribuido a Royce en 1970

Informatizar

Requerimientos del sistema

- ¿Qué necesito para desarrollarlo?
- ¿Qué personal tengo?
- ¿Que habilidades posee?



Se contesta la Pregunta
¿QUÉ?



Recabar Da
Informac

Investigación Inicial
Investigación de Necesidad
Encuestas
Entrevistas



Recabar Datos e Información



Investigación Inicial
Investigación de Necesidades
Encuestas
Entrevistas



Recabar Datos e Información



Definición Inicial de Necesidades



Análisis

Es la fase en la cual se reúnen todos los requisitos que debe cumplir el software

Requerimientos del Software



ESTUDIO

Es el análisis de las necesidades por un periodo de tiempo, que incluye aspectos económicos, técnicos, etc.

Partes:

- Alcance del estudio
- Definición de objetivos
- Determinación de recursos
- Evaluación de riesgos
- Revisión de resultados

ESTUDIO DE VIABILIDAD

Es el análisis de un conjunto concreto de necesidades para proponer una solución a corto plazo, que tenga en cuenta restricciones económicas, técnicas, legales y operativas

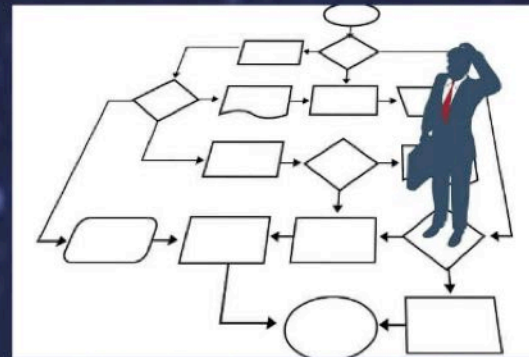
Partes:

- Alcance del Proyecto
- Análisis de la Situación
- Definición de Requisitos
- Determinación del enfoque
- Evaluación de la viabilidad del Proyecto
- Revisión del Estudio de la Viabilidad del Proyecto

Diseño

estructura de datos, arquitectura de software, representación de interfaz, algoritmos

Diseño Preliminar y Detallado



ESPECIFICACIONES

son los documentos que definen las normas, exigencias y aplicados

Items:

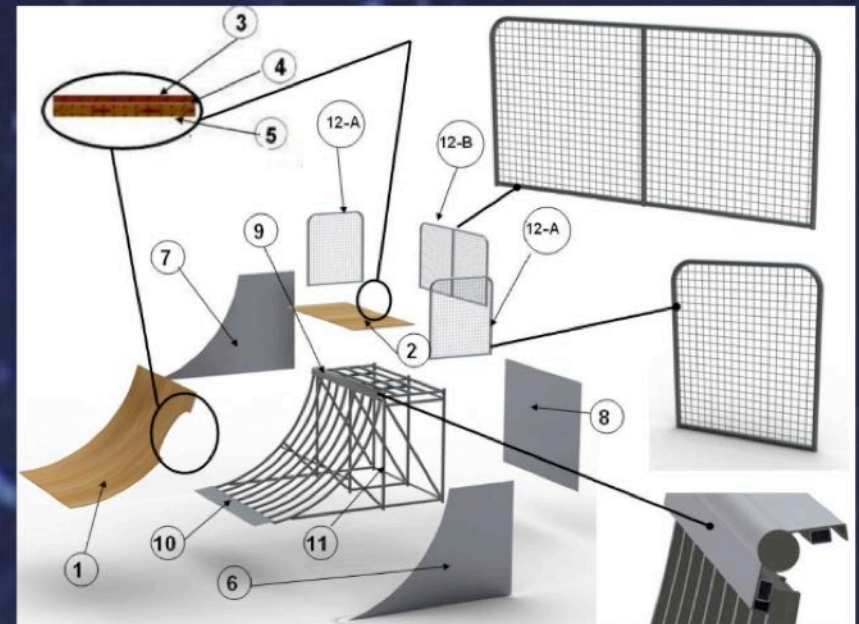
- Definición:** Donde se define la forma concisa a referirse.
- Referencia:** Materiales y herramientas utilizados para especificar.
- Procedimiento:** Donde se describe la forma de ejecutarse este proceso.
- Medición:** Donde se define la precisión como se medirá de este proceso ejecutado para el correspondiente.
- Forma de pago:** Donde se define cómo será pagado el proceso que comprende exacto pago.

ESPECIFICACIONES DE DISEÑO

son los documentos en los cuales se definen las normas, exigencias y procedimientos a ser empleados y aplicados

Items:

- **Definición.** Donde se describe en forma concisa a que ítem se refiere.
- **Materiales y herramientas,** utilizados para ejecutar la tarea específica.
- **Procedimiento de ejecución,** donde se describe la forma en que debe ejecutarse este rubro.
- **Medición,** donde se describe con precisión como se efectuara la medición de este rubro, una vez ejecutado para proceder al pago correspondiente.
- **Forma de pago,** donde se detalla cómo será pagado y que se comprende exactamente en dicho pago.



Codificación

Esta etapa se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en la detección de errores



Aplicaciones

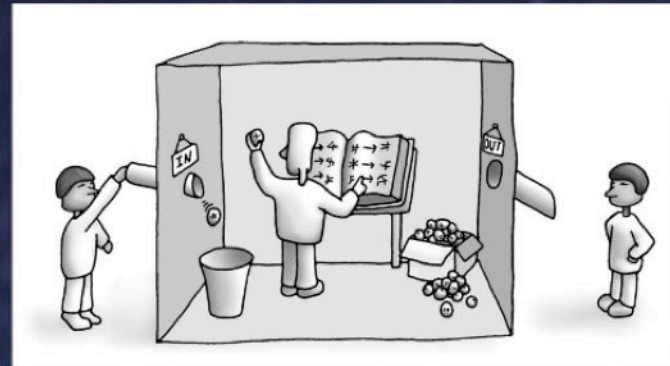
Dependiendo del lenguaje de programación y su versión se crean las bibliotecas y componentes reutilizables dentro del mismo proyecto para hacer que la programación sea un proceso mucho más rápido.



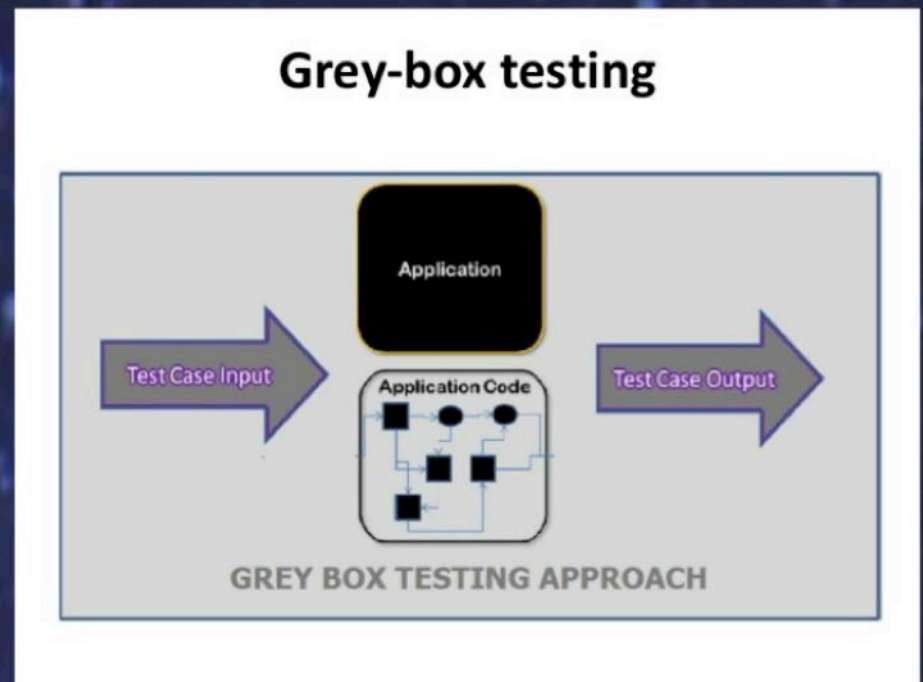
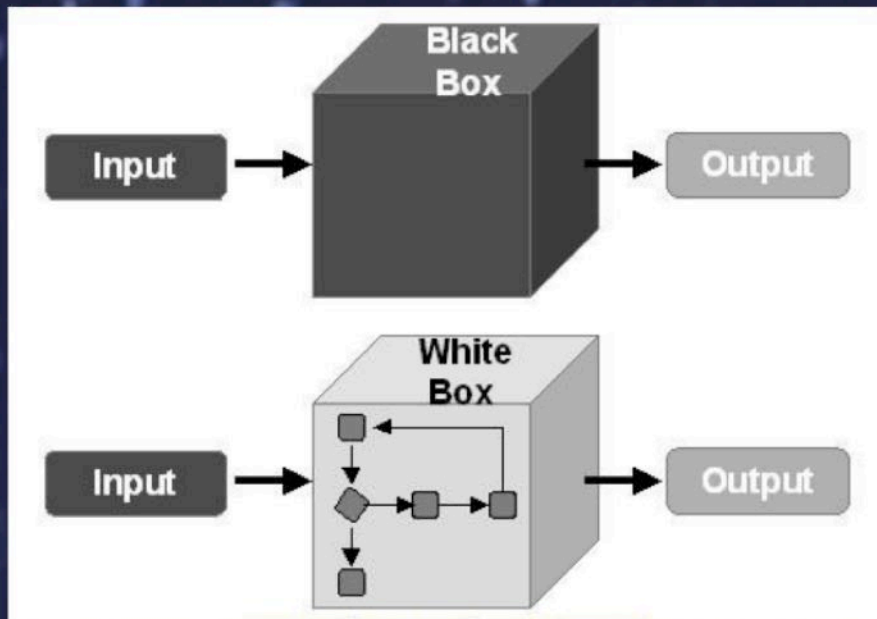
Validación o Pruebas

Esta etapa se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en la detección de errores.

**Test y Pruebas
orientadas a la
Operación**



Caja Negra, Blanca y Gris



Ventajas

- Primer Modelo empleado, por ende ha sido corregido
- Facilita la gestión del desarrollo



Operación y Mantenimiento

Debido a que el programa puede tener errores, puede no ser del completo agrado del cliente o puede necesitar, eventualmente acoplarse a los cambios en su entorno. Esto quiere decir que no se rehace el programa, sino que sobre la base de uno ya existente se realizan algunos cambios.

Instalación y
Explotación del SW



Ventajas

- Primer Modelo empleado, por ende ha sido corregido
- Facilita la gestión del desarrollo



Desventajas

- No es posible establecer los requisitos hasta que el sistema funciona
- Los requisitos no pueden ser fijos, las necesidades cambian, los requisitos deben evolucionar
- El usuario debe esperar mucho tiempo hasta ver los resultados
- Al no poder detectar los errores sino hasta el final, estos se propagan por todas las fases (efecto dominó)

Modelo de Prototipos

Es un modelo del comportamiento del sistema que puede ser usado para entenderlo completamente o ciertos aspectos de él y así clarificar los requerimientos.

Un prototipo es una representación de un sistema, aunque no es un sistema completo, posee las características del sistema final o parte de ellas



Identificar los requerimientos conocidos

Comunicación



Tener una interacción con el cliente para evaluar la petición del software y determinar si el programa a desarrollar es un buen candidato para construir un prototipo.

Debido a que el cliente debe interactuar con el prototipo para determinar el refinamiento del proyecto

Desarrollo, Entrega y Retroalimentación

Programación y prueba

En esta etapa es donde los cambios identificados en el diseño técnico son implementados y probados para asegurar la corrección y completitud de los mismos con respecto a los requerimientos.

Las pruebas serán de realizarse tantas veces sea necesarias para verificar cualquier tipo de anomalía en el sistema.

Modelo de Prototipos o Evolutivo

Propuesto por Gooma en 1984

Este modelo no secuencial, basado en la construcción de simulaciones o modelos ejecutables de aplicaciones más extensos, persigue un objetivo principal: la participación directa del cliente en la construcción del software requerido.

Plan Rápido

Desarrollo de un Modelo que Funcione

Permite a los usuarios conocer lo que se espera y del proceso de desarrollo.

- Lenguaje que se va implementar
- Herramientas y formatos para entrada de datos.
- Módulos esenciales de procesamiento Salida del sistema.



Construcción del Prototipo

En esta etapa el sistema debe ser rediseñado y tener la respectiva documentación guiándose en los estándares que tiene la organización la cual servirá como ayuda en mantenencias futuras del mismo.



Diseño Técnico

En este punto existen dos etapas:
• Producción de una documentación de diseño la cual especifica y describe la estructura del software, interfaces de usuario, funciones y el control de flujo.
• Producción de todo lo requerido para proveer cualquier mantención futura del software.

Modelado del Prototipo

Requerimientos

El objetivo en esta fase es determinar todos los requerimientos y deseos que los usuarios tienen en relación al proyecto que se requiere implementar.



Identificar los requerimientos conocidos

Comunicación



Tener una interacción con el cliente para evaluar la petición del software y determinar si el programa a desarrollar es un buen candidato para construir un prototipo.

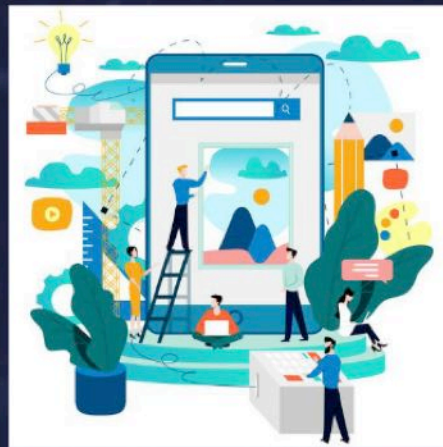
Debido a que el cliente debe interaccionar con el prototipo para determinar el refinamiento del proyecto

Plan Rápido

Desarrollo de un Modelo que Funcione

Permite a los usuarios conocer lo que se espera y del proceso de desarrollo.

- Lenguaje que se va implementar
- Pantallas y formatos para entrada de dato.
- Módulos esenciales de procesamiento Salida del sistema.



Modelado del Prototipo

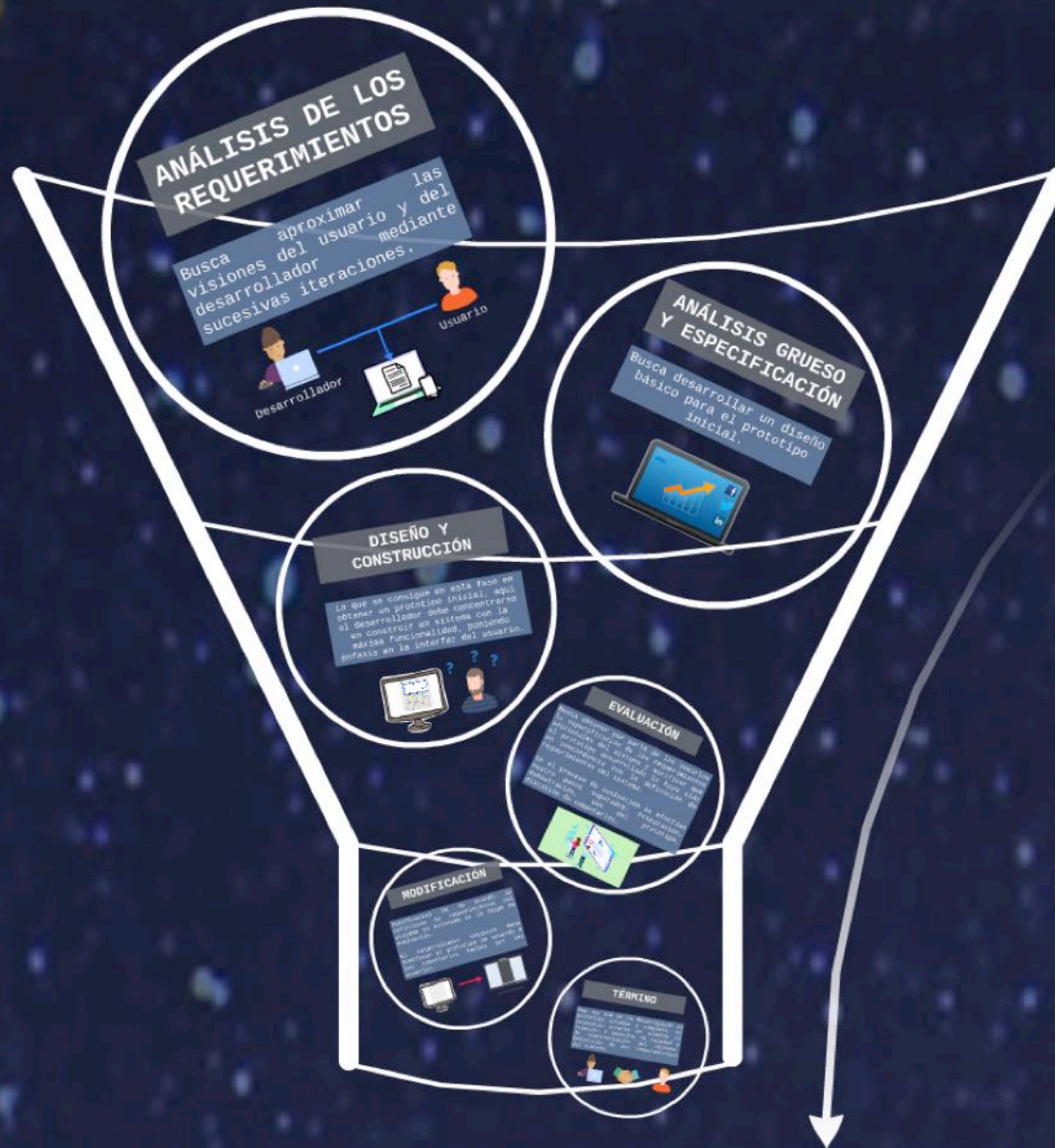
Requerimientos

El objetivo en esta fase es determinar todos los requerimientos y deseos que los usuarios tienen en relación al proyecto que se requiere implementar.

DEFINICIÓN DE REQUERIMIENTOS

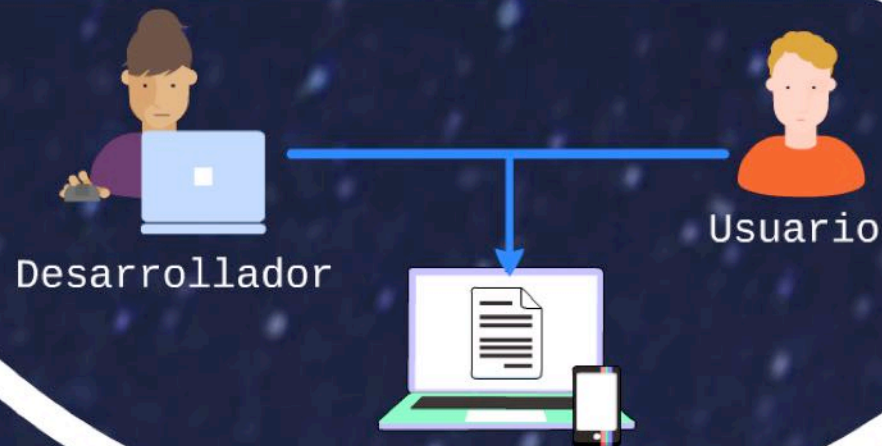


DEFINICIÓN DE REQUERIMIENTOS



ANÁLISIS DE LOS REQUERIMIENTOS

Busca aproximar las visiones del usuario y del desarrollador mediante sucesivas iteraciones.



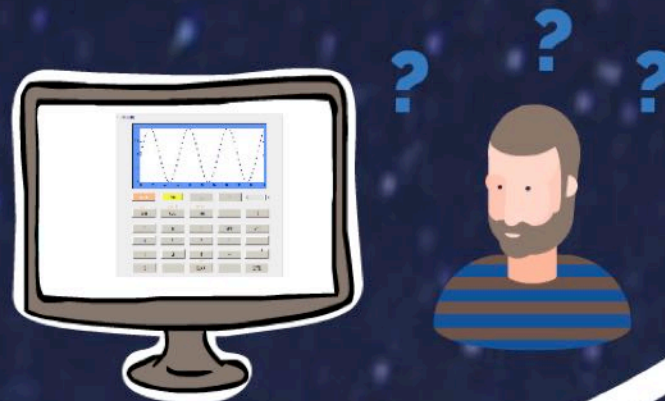
ANÁLISIS GRUESO Y ESPECIFICACIÓN

Busca desarrollar un diseño básico para el prototipo inicial.



DISEÑO Y CONSTRUCCIÓN

Lo que se consigue en esta fase es obtener un prototipo inicial, aquí el desarrollador debe concentrarse en construir un sistema con la máxima funcionalidad, poniendo énfasis en la interfaz del usuario.



EVALUACIÓN

Busca obtener por parte de los usuarios la especificación de los requerimientos adicionales del sistema y verificar que el prototipo desarrollado lo haya sido en concordancia con la definición de requerimientos del sistema.

En el proceso de evaluación se efectúan cuatro pasos separados: Preparación, demostración, uso del prototipo, discusión de comentarios.



MODIFICACIÓN

Modificación Se da cuando la definición de requerimientos del sistema es alterada en la etapa de evaluación.

El desarrollador entonces debe modificar el prototipo de acuerdo a los comentarios hechos por los usuarios.



TÉRMINO

Una vez que se ha desarrollado un prototipo estable y completo, es necesario ponerse de acuerdo en relación a aspectos de calidad y de representación del sistema. Definición de los requerimientos del sistema



Construcción del Prototipo

En esta etapa el sistema debe ser rediseñado y tener la respectiva documentación guiándose en los estándares que tiene la organización la cual servirá como ayuda en mantenciones futuras del mismo.

Diseño Técnico



En este punto existen dos etapas:

- ▶ Producción de una documentación de diseño la cual especifica y describe la estructura del software, interfaces de usuario, funciones y el control de flujo.
- ▶ Producción de todo lo requerido para promover cualquier mantención futura del software.



Desarrollo, Entrega y Retroalimentación

Programación y prueba

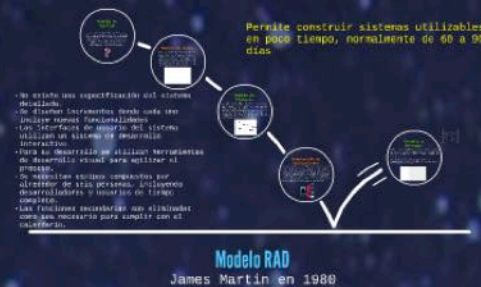
En esta etapa es donde los cambios identificados en el diseño técnico son implementados y probados para asegurar la corrección y completitud de los mismos con respecto a los requerimientos.

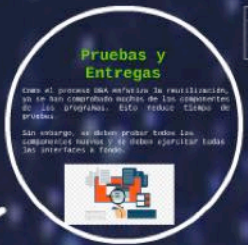
Las pruebas serán de realizarse tantas veces sea necesarias para verificar cualquier tipo de anomalía en el sistema.

MODELO DRA O RAD

Desarrollo Rápido de Aplicaciones

Pressman, 2015: es un modelo de proceso del desarrollo del software lineal secuencial que enfatiza un ciclo de desarrollo extremadamente corto. DRA es una adaptación a "Alta velocidad" en el que se logra el desarrollo rápido utilizando un enfoque de construcción basado en componentes. Si se comprenden bien los requisitos y se limita el ámbito del proyecto, el proceso DRA permite al equipo de desarrollo crear un "sistema completamente funcional" dentro de periodos cortos de tiempo.





Permite construir sistemas utilizables en poco tiempo, normalmente de 60 a 90 días

- No existe una especificación del sistema detallada.
- Se diseñan incrementos donde cada uno incluye nuevas funcionalidades
- Las interfaces de usuario del sistema utilizan un sistema de desarrollo interactivo
- Para su desarrollo se utilizan herramientas de desarrollo visual para agilizar el proceso.
- Se necesitan equipos compuestos por alrededor de seis personas, incluyendo desarrolladores y usuarios de tiempo completo.
- Las funciones secundarias son eliminadas como sea necesario para cumplir con el calendario.

Modelo RAD

James Martin en 1980

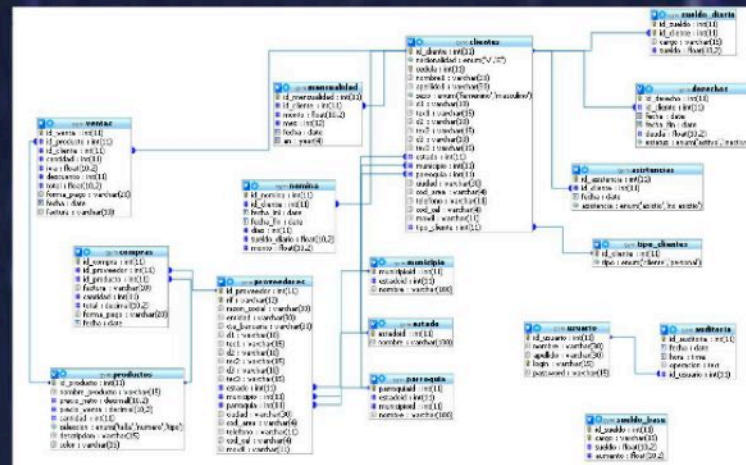
Modelo de Gestión

El flujo de información entre las funciones de gestión se modela de forma que responda a las siguientes preguntas:
¿Qué información conduce el proceso de gestión? ¿Qué información se genera?
¿Quién la genera? ¿A dónde va la información? ¿Quién la procesa?.



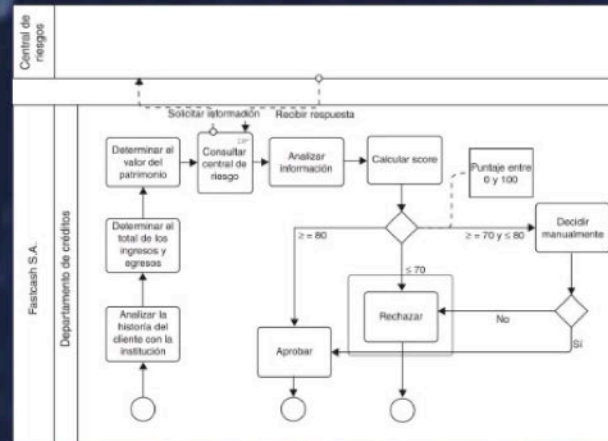
Modelo de Datos

El flujo de información definido como parte de la fase de modelado de gestión se refina como un conjunto de objetos de datos necesarios para apoyar la empresa. Se definen las características (llamadas atributos) de cada uno de los objetos y las relaciones entre estos objetos.



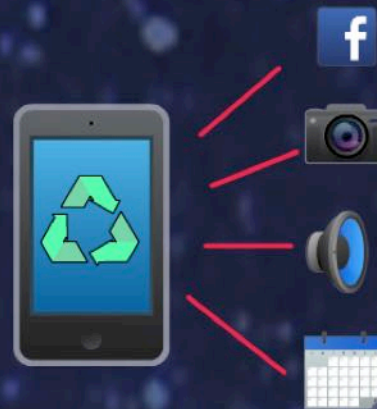
Modelo de Procesos

Los objetos de datos definidos en la fase de modelado de datos quedan transformados para lograr el flujo de información necesario para implementar una función de gestión. Las descripciones del proceso se crean para añadir, modificar, suprimir, o recuperar un objeto de datos. Es la comunicación entre los objetos.



Generación de Aplicaciones

El DRA asume la utilización de técnicas de cuarta generación. En lugar de crear software con lenguajes de programación de tercera generación, el proceso DRA trabaja para volver a utilizar componentes de programas ya existentes (cuando es posible) o a crear componentes reutilizables (cuando sea necesario). En todos los casos se utilizan herramientas automáticas para facilitar la construcción del software.



- Ventajas de RAD**
 - Comprar puede ahorrar dinero en comparación con construir.
 - Los entregables pueden ser fácilmente trasladados a otra plataforma.
 - El desarrollo se realiza a un nivel de abstracción mayor.
 - Visibilidad temprana.
 - Mayor flexibilidad.
 - Menor codificación manual.
 - Mayor involucramiento de los usuarios.
 - Posiblemente menos fallas.
 - Posiblemente menor costo.
 - Ciclos de desarrollo más pequeños.
 - Interfaz gráfica estándar.
- Desventajas**
 - Mayor costo.
 - Menor control de calidad.
 - Menor flexibilidad.
 - Mayor riesgo de errores.
 - Menor involucramiento de los usuarios.
 - Posiblemente más fallas.
 - Posiblemente mayor costo.
 - Ciclos de desarrollo más largos.
 - Interfaz gráfica no estándar.

Pruebas y Entregas

Como el proceso DRA enfatiza la reutilización, ya se han comprobado muchos de los componentes de los programas. Esto reduce tiempo de pruebas.

Sin embargo, se deben probar todos los componentes nuevos y se deben ejercitar todas las interfaces a fondo.



Ventajas vs Desventajas

Ventajas de RAD

- Comprar puede ahorrar dinero en comparación con construir.
- Los entregables pueden ser fácilmente trasladados a otra plataforma.
- El desarrollo se realiza a un nivel de abstracción mayor.
- Visibilidad temprana.
- Mayor flexibilidad.
- Menor codificación manual.
- Mayor involucramiento de los usuarios.
- Posiblemente menos fallas.
- Posiblemente menor costo.
- Ciclos de desarrollo más pequeños.
- Interfaz gráfica estándar.

Desventajas de RAD

- Comprar puede ser más caro que construir.
- Costo de herramientas integradas y equipo necesario.
- Progreso más difícil de medir.
- Menos eficiente.
- Menor precisión científica.
- Riesgo de revertirse a las prácticas sin control de antaño.
- Más fallas (por síndrome de "codificar a lo bestia").
- Prototipos pueden no escalar, un problema mayúsculo.
- Funciones reducidas (por "timeboxing").
- Dependencia en componentes de terceros: funcionalidad de más o de menos, problemas legales.

Conclusiones

- Cuando deseemos desarrollar sistemas complejos, en el cual tengamos tiempos asignados amplios y requerimientos conocidos y definidos por el cliente es recomendable utilizar el Modelo Lineal Secuencial
- Para proyectos en donde el cliente no tiene definidos completamente sus requerimientos o no tiene definida sus ideas, se recomienda el Modelo de Prototipos
- Para Sistemas con tiempos de entrega reducidos, con un equipo desarrolladores consolidado y equipos integrados, se recomienda el Modelado DRA

¿Dudas?



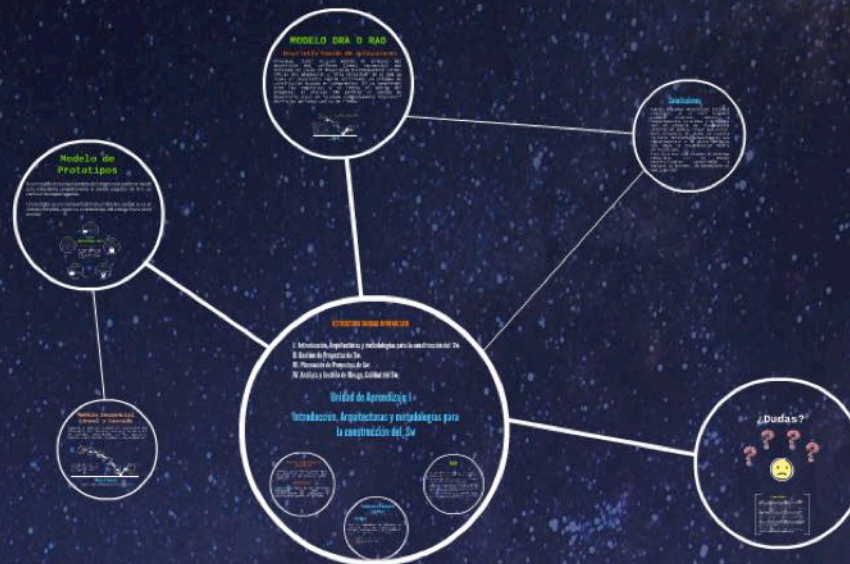
Referencias Bibliográficas

- Consejo Superior de Informática . (2001). Obtenido de https://administracionelectronica.gob.es/pae/Home/pae/Documentacion/pae_Metodolog/pae_Metrica_v3.html#xy0xAna200k
- curiosissimas.wordpress.com. (2015). Obtenido de curiosissimas.wordpress.com: <https://curiosissimas.wordpress.com/linux/modelo-de-desarrollo-rapido-de-aplicaciones/>
- <http://segundomodelo.blogspot.com>. (07 de 05 de 2013). Obtenido de <http://segundomodelo.blogspot.com>: <http://segundomodelo.blogspot.com/2013/05/modelo-de-prototipos-este-modelo-no.html>
- MARIA, N. (25 de 04 de 2015). EducaInternet. Obtenido de EducaInternet: [http://educainternet.es/pictures/25647localizen-welika_1_A_\(2015\)_modelosdesoftware.webnode.es](http://educainternet.es/pictures/25647localizen-welika_1_A_(2015)_modelosdesoftware.webnode.es). Obtenido de <https://modelosdesoftware.webnode.es/>
- Rajja, U. N. (2022). <https://www.studocu.com>. Obtenido de <https://www.studocu.com>: <https://www.studocu.com/en/document/universidad-nacional-de-la-rioja/ingenieria-en-software/summaries/resumen-presman-capitulo-3/2868353/view>
- Roger S. Pressman, B. R. (2015). Software Engineering: A Practitioner's Approach.
- Sánchez, T. C. (junio de 2016). Gestión de Proyectos de Instalaciones de Telecomunicaciones 2º STI. Obtenido de <https://sites.google.com/site/gestiondeproyectos2sti/home>

Referencias Bibliográficas

- Consejo Superior de Informática . (2001). Obtenido de https://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Metrica_v3.html#.XY0xAmaZKUK
- curiosisimos.wordpress.com. (2015). Obtenido de curiosisimos.wordpress.com: <https://curiosisimos.wordpress.com/linux/modelo-de-desarrollo-rapido-de-aplicaciones/>
- <http://segundomodelo.blogspot.com>. (07 de 05 de 2013). Obtenido de <http://segundomodelo.blogspot.com>: <http://segundomodelo.blogspot.com/2013/05/modelo-de-prototipos-este-modelo-no.html>
- MARIA, N. (25 de 04 de 2015). EducaInternet. Obtenido de EducaInternet: <http://educainternet.es/pictures/2564?locale=en>
- Melisa, J. A. (2012). modelosdesoftware.webnode.es. Obtenido de modelosdesoftware.webnode.es: <https://modelosdesoftware.webnode.es/dra/>
- Rioja, U. N. (2018). <https://www.studocu.com>. Obtenido de <https://www.studocu.com>: <https://www.studocu.com/en/document/universidad-nacional-de-la-rioja/ingenieria-en-software/summaries/resumen-pressman-capitulo-3/2868353/view>
- Roger S. Pressman, B. R. (2015). Software Engineering: A Practitioner's Approach.
- Sánchez, T. C. (junio de 2016). Gestión de Proyectos de Instalaciones de Telecomunicaciones 2º STI. Obtenido de <https://sites.google.com/site/gestiondeproyectos2sti/home>

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO



CENTRO UNIVERSITARIO UAEM TEMASCALTEPEC

LICENCIATURA EN INFORMÁTICA ADMINISTRATIVA

INGENIERÍA DE SOFTWARE

UNIDAD DE COMPETENCIA I:

Introducción, Arquitecturas y Metodologías para la construcción del Sw

ELABORO:

M. EN T. I. GISELA REGINA BAENA CASTRO

FECHA: 14 DE AGOSTO DE 2019