

Empresa= (**nit**, nombre, dirección)

Telefono_Empresa = (**nit**, **teléfono**)

Correo_Empresa = (**nit**, **Correo**)

Personas = (**Id**, nombre, sexo, fecha_nacimiento, dirección)

Telefono_Persona = (**id**, **teléfono**)

Correo_Persona = (**id**, **correo**)

Producto = (**id**, nombre, precio, garantía, ofertas)

Ciudad_Producto = (**id**, calle, carrera, estrato)

Vende = (**id persona**, **id producto**)

Tiene = (**nit**, **id producto**)

Consulta = (**id persona**, **id producto**)

GUIA DE APRENDIZAJE No 6

CRISTIAN CAMILO GALLEGO GARCIA

SOLUCION DE GUIA ANALISIS Y DESARROLLO DE SISTEMAS DE
INFORMACION

FICHA 2395871

INSTRUCTOR DANIEL DAVID LEAL LARA

3.2 Actividades de contextualización e identificación de conocimientos necesarios para el aprendizaje.

¿Considera que el informe presentado de acuerdo al análisis del problema es fundamental para el desarrollo e implementación de su proyecto, explique por qué?

Si es importante ya que con la información que obtuvimos podemos conocer cuáles son las necesidades del cliente y que es lo que desea obtener, también el tiempo y presupuesto que vamos a utilizar así como el modelo de sistema, todo esto a través de la información levantada de las encuestas y entrevistas realizadas.

¿Cree que en el resultado del análisis mediante un informe se deben abarcar todos los procesos, además de la normatividad y necesidades del cliente, explique porque?

Si por que con anterioridad debemos conocer que podemos y que no hacer, a través de la normatividad podemos tener una guía y los lineamientos claros para llegar a nuestro fin y también debemos tener un panorama claro de todos los procesos para poderlos involucrar oportunamente de ser necesario pero esto solo podemos saberlo si los conocemos de antemano.

¿Investigue y mencione mínimo tres tipos de herramientas y metodologías que permitan realizar modelos que garanticen la construcción de un software de calidad?

Waterfall (Cascada): nace después de encontrar una falencia en los procedimientos lineales por lo que se propone organizar las etapas de arriba abajo en donde hay una revisión muy estricta en cada etapa antes de pasar a la siguiente, las fases principales son: Análisis, diseño, Implementación, Verificación y Mantenimiento, es importante seguir el orden o la calidad del proyecto final se verá afectada, es el modelo más utilizado, una de sus ventajas es que podemos estimar el costo y el trabajo al comenzar el proyecto, y la desventaja es que el cliente no puede integrarse hasta que el proceso se termine.

Prototipado: En esta metodología podemos realizar prototipos de Software que nos va a permitir probarlo en tiempo real y así ir solucionando los problemas que vayan saliendo e incluir nuevos requerimientos como se vayan necesitando, hay varios tipos pueden ser según su tipo de requisito de interfaz o Funcional y según su extensión, horizontal o vertical, está basado en el método prueba y error para poder entender las especificaciones del producto, una de sus ventajas es que estamos más cerca con el cliente donde él puede ver el avance que llevamos y a la vez podemos obtener información que podamos necesitar, en cuanto a la desventaja está el tiempo de desarrollo que puede ser más largo.

Espiral: Es un modelo que se puede combinar con otros modelos, como su nombre lo indica se desarrolla en forma de espiral de tal modo que cada vuelta nos acerque más a nuestro resultado, cada ciclo debe ser muy analizado y se debe realizar una proyección para pasar al siguiente o para ajustar la vuelta anterior, entre sus fases se encuentra comunicación, planeación, análisis de riesgo, ingeniería de construcción y adaptación y evaluación del cliente, entre sus ventajas esta que es un modelo adaptable, además como se trabaja de la mano del cliente se puede manejar cualquier tipo de riesgo, la desventaja es que puede ser muy costoso y demorado por lo que es recomendado en sistemas grandes.

¿Los diagramas UML permiten el análisis de los requerimientos de un sistema de información, explique por qué?

Los diagramas UML describen lo que nuestro sistema va a realizar pero lo hace en un lenguaje universal para que no solo nosotros podamos entenderlo también lo podría hacer una persona ajena por lo que si permitiría el análisis de requerimientos del sistema.

Consultar y contextualizar las definiciones en diferentes fuentes bibliográficas de:

Modelo: Son métodos o técnicas, que podemos utilizar para lograr sistemas más eficientes y tener opciones dependiendo de las necesidades del cliente, con el podemos estructurar, planificar y controlar el proceso de desarrollo ya que nos da un orden y los requisitos de entrada y salida.

Software: Es lo intangible de nuestro ordenador, se puede definir en un grupo de instrucciones que indican lo que se debe hacer y cómo se debe hacer un ejemplo claro de software son los programas y sistemas operativos. El software no es exclusivo de computadores pues se trata también del tipo de herramienta en cualquier tipo de maquinaria que da las instrucciones comparado con el cuerpo humano el software sería el sistema nervioso quien da las ordenes al cuerpo como tal, su traducción literal sería partes blandas todo lo contrario al Hardware que sería partes duras que sin el software no tendrían utilidad.

Sistema: es un objeto completo formado por partes las cuales se relaciona entre sí, los sistemas tienen composición, estructura y entorno estos a su vez tienen 3 partes importantes una entrada (datos, información que ingresa al sistema), un proceso (Son los cambios que se realizan para generar la salida) y la salida que es el resultado del sistema, los sistemas se pueden clasificar según su composición o naturaleza y pueden ser abiertos y cerrados.

Prototipo: Es un primer modelo el cual sirve de muestra para los siguientes productos nos da una primera idea permitiéndonos corregir errores y ajustando procesos con el fin de que nuestro producto final sea el adecuado y esperado confirmándonos que cuenta con las características y el diseño planteado, ya que lo que necesitamos es que el prototipo simule el flujo deseado de nuestro producto no es necesario que sea de calidad y el diseño puede quedar en segundo plano.

Clase: Es una herramienta que nos sirve para modelar objetos en donde por medio de agrupación de variables, métodos y eventos podemos personalizar el proceso que realizara este objeto además realizar tantas copias como necesitemos de este.

Codificación: es un método en donde los datos se muestran a través de símbolos o números donde deben seguir para que pueda ser procesada por un sistema y pueda reflejarse en información, algunos tipos de información con su tipo de codificación son: Numérica, Enteros (binario natural, complemento a dos, BCD), Alfanumérica: ASCII, Unicode (UTF-8, UTF-16) Otros: Cifrado de clave única, cifrado de clave pública, hash o resúmenes entre otros.

Caso de Uso: es una herramienta dentro de UML que nos permite conocer el sistema desde el punto de vista del usuario donde se puede conocer la funcionalidad del sistema pero sin revelar la estructura interna de este modo alguna persona puede interactuar con el sistema.

Diagrama: es un resumen en forma gráfica de ideas, procesos, sistemas etc para poder dar una comprensión más simple y visual resaltando las ideas principales utilizando elementos gráficos para realizarlo, entre los diagramas tenemos de flujo, conceptuales, florales, sinópticos entre otros, entre las figuras más utilizadas están los globos rectángulos, cuadrados y rombos que se unen por líneas o flechas para formar un todo.

Objeto: es un ente dentro del sistema que se crea de la interacción de varios elementos estos objetos interactúan con otros objetos entonces formarían cada parte del sistema y se componen por 3 partes fundamentales: método, evento y atributo.

Diagramas UML: es un lenguaje estandarizado que permite el entendimiento del modelo de sistemas y permite describir, diseñar, especificar, visualizar, construir y documentar todo lo que lo compone. En pocas palabras nos permite describir todo lo que hará el sistema sin embargo no nos dice como lo hará.

Atributo: son los que forman el objeto y determinan su apariencia o cualidades, cada una de las características individuales mediante el uso de atributos se pueden asignar valores específicos a ciertos elementos

Análisis: Literalmente la palabra análisis hace alusión en observar algo en sus más mínimos componentes en el ámbito de sistemas el análisis nos ayuda para estudiar o evaluar un problema y como se maneja en la compañía para poder corregirlo.

Diseño: es el proceso en el que se define la arquitectura del sistema para satisfacer unos requerimientos cabe aclarar que dependiendo lo requerido existen varios estilos, una forma de caracterizar una aplicación es por la importancia relativa de sus modelos de objetos, dinámico y funcional.

Versiones de UML: UML 1.X (comprende UML 1.1, 1.2, 1.3, 1.4, 1.5): desde finales de los 90 se empezó a trabajar con el estándar UML. En los años sucesivos fueron apareciendo nuevas versiones que introducían mejoras o ampliaban a las anteriores.

UML 2.X (comprende UML 2.1 hasta UML 2.5, 2.6, etc.): en torno a 2005 se difundió una nueva versión de UML a la que podemos denominar UML 2.X. Comprenden varias revisiones.

Herramientas CASE: sus siglas significan Computer Aided Software Engineering (Ingeniería de Software Asistida por Computadora) Nos ayudan al desarrollo y mantenimiento de proyectos a través de estas herramientas, sistematizando el proceso a través de la computadora

HERRAMIENTAS DE CONTROL DE CALIDAD: La mayor parte de las herramientas CASE que afirman que tiene como principal interés el control de calidad son en realidad herramientas métricas que hacen una auditoría del código fuente para determinar si es justa o no a ciertos estándares del lenguaje. Otras herramientas extraen métricas técnicas como base para medir la calidad del software que se está construyendo.

HERRAMIENTAS DE GESTIÓN COMO BASE DE DATOS: El software de gestión de bases de datos sirve como fundamentos para establecer una base de datos CASE. Dado el énfasis acerca de los objetos de configuración, las herramientas de gestión de bases de datos para CASE pueden evolucionar a partir de los sistemas de gestión de bases de datos relacionales (SGBDR) para transformarse en sistemas de gestión de bases de datos orientadas a objetos (SGBDOO).

HERRAMIENTAS DE SEGUIMIENTO DE REQUISITOS: Cuando se desarrollan grandes sistemas, el sistema proporcionado suele no satisfacer los requisitos especificados por el cliente. El objetivo de las herramientas de seguimiento de requisitos es proporcionar un enfoque sistemático para el aislamiento de requisitos, comenzando por las especificaciones del cliente. Las herramientas de trazado de requisitos típicos combinan una evaluación de textos por interacción humana, con un sistema de gestión de bases de datos que almacena y categoriza todos y cada uno de los requisitos del sistema que se "analizan" a partir de las especificaciones originales.

