**Taller 1 – Jaime Alberto Gaviria**

**¿Qué es ingeniería de software?**

La Ingeniería de Software es una de las ramas de las ciencias de la computación que estudia la creación de software confiable y de calidad, basándose en métodos y técnicas de ingeniería. Brindando soporte operacional y de mantenimiento, el campo de estudio de la ingeniería de software, [[1]](#footnote-1)entendiendo dicho párrafo, es que la ingeniería del software es un eslabón más dentro de todo el campo de la computación, el cual permite a los sujetos inmersos en el área, desarrollar proyectos, en los cuales garantice al usuario final el óptimo funcionamiento de sus productos.

**¿Qué son las metodologías de desarrollo de software? ¿Para qué sirven?**

Cuando se habla de metodologías se debe entender la palabra como formas de hacer, y la palabra desarrollo simplemente para nosotros es la creación de algo, por ende, tiene que ver con el conjunto de procedimientos racionales utilizados para alcanzar un objetivo que requiera habilidades y conocimientos específicos, como por ejemplo una página web. La metodología es una de las etapas específicas de un trabajo o proyecto que parte de una posición teórica y conlleva a una selección de técnicas concretas (en ello podemos tener en cuenta arquitecturas de software existentes) o métodos acerca del procedimiento para el cumplimiento de los objetivos. De igual manera, se puede ver como el conjunto de métodos que se utilizan en una determinada actividad con el fin de formalizarla y optimizarla. en síntesis determina los pasos a seguir y cómo realizarlos para finalizar una tarea; el método es el que nos permite tener la pauta del que hacer y cómo hacer a través del tiempo[[2]](#endnote-1).

**¿Qué son ciclos de vida del software? ¿Y para qué sirven?**

El ciclo de vida del software, básicamente es tener en cuenta cierta obsolencia programada, en donde debemos reconocer que el campo del software constantemente está evolucionando y por ende es necesario a la hora de realizar un proyecto tener presente que dicho debe ser modificado o cambiado al termino de determinado tiempo.

El ciclo de vida contempla las fases necesarias para validar el desarrollo del software y así garantizar que este cumpla los requisitos para la aplicación y verificación de los procedimientos de desarrollo, asegurándose de que los métodos usados son apropiados.

a normativa ISO/IEC/IEEE 12207:2017 establece:

“Un marco común para los procesos del ciclo de vida de los programas informáticos, con una terminología bien definida, a la que pueda remitirse la industria del software. Contiene procesos, actividades y tareas aplicables durante la adquisición, el suministro, el desarrollo, el funcionamiento, el mantenimiento o la eliminación de sistemas, productos y servicios informáticos. Estos procesos del ciclo de vida se llevan a cabo mediante la participación de los interesados, con el objetivo final de lograr la satisfacción del cliente”.[[3]](#endnote-2)

Tenemos ciclos de vida como el espiral, la cascada, por prototipos entre otros.

**¿Qué es ingeniería de requisitos?**

Cuando se habla de requisitos por ejemplo en un proyecto de software, es tener presente que es todo lo que nuestro cliente quiere que ejecute su idea. Es muy importante tener presente que de ello depende el éxito de un proyecto, en especial con aquello que tiene que ver relacionado al tiempo de entregas. Por ello debemos tener en cuenta que para crear tareas relacionadas con la determinación de las necesidades o de las condiciones a satisfacer para un software nuevo o modificado, tomando en cuenta los diversos requisitos de las partes interesadas, que pueden entrar en conflicto entre ellos. Sin embargo, la Ingeniería de requerimientos también es contemplada en otras disciplinas, estando fuertemente vinculada con la administración de proyectos.[[4]](#endnote-3)

Muchas veces se habla de requerimientos en vez de requisitos; esto se debe a una mala traducción del inglés. La palabra requirement debe ser traducida como ***requisito***, mientras que requerimiento se traduce al inglés como request, es importante tener en cuenta que dichos términos por mi parte han sido constantemente confundidos, pero cuando estableces que esto es lo que le da la forma y su correcta recolección permitirá obtener el éxito al final del tiempo pactado.

El propósito de la ingeniería de requisitos es hacer que los mismos alcancen un estado óptimo antes de alcanzar la fase de diseño en el proyecto. Los buenos requisitos deben ser medibles, comprobables, sin ambigüedades o contradicciones.

**¿Cuáles son las técnicas de recolección de información que conoce para qué sirve?**

**Método de observación:** establecer patrones, basado en aspectos o patrones visualizados, tiende a ser algo subjetivo

**Cuestionarios o encuestas**: pueden ser utilizadas para analizar el grado de satisfacción de un cliente

**Focus group**: realizar un estudio estadístico

**Entrevistas:** conocer la percepción de un fenómeno del entrevistado

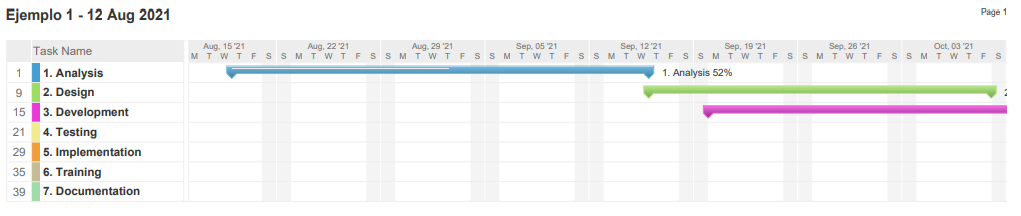
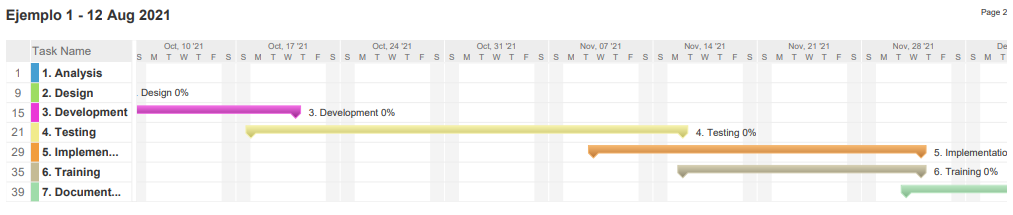
**Describa las dos técnicas que usted considera más relevantes y justifique su respuesta**

Focus group, me parece interesante pues permite tener datos medibles de los fenómenos estudiados

Entrevista, pues permite tener un contacto cercano con el sujeto de estudio

**Que es un diagrama de Gantt**

Un diagrama de Gantt nos permite visualizar en una especie de línea temporal el orden de las actividades y el encargado, teniendo presente el inicio y fin del proyecto y de cada una de las tareas en las cuales dicho proyecto se divide, las tareas siempre deben llevar un responsable y un tiempo estimado de cuando se demorará dicha responsable en realizarla. Además de ello se debe tener en cuenta que algunas tareas dentro de un proyecto dependen de otras, y este diagrama nos permite visualizarlo. Las cualidades principales del diagrama de Gantt es que simplifica a la vista el orden de las tareas, permite gestionar de forma adecuada el tiempo (aunque considero la herramienta trello, como una buena web, análoga a dicho diagrama) y permite realizar modificaciones pues obviamente dicha herramienta, es una ayuda visual para tener presente a la hora de la organización y creación de proyectos

**Ejemplo diagrama de Gantt**[[5]](#endnote-4) 



**¿Qué es un estándar de codificación? Y ¿Cuáles existen?**

Los estándares de código, son parte de las llamadas buenas practicas o mejores prácticas, estas son un conjunto no formal de reglas, que han ido surgiendo en las distintas comunidades de desarrolladores con el paso del tiempo y las cuales, bien aplicadas pueden incrementar la calidad de tu código, notablemente.

Para la organización de los archivos tenemos: comentarios al inicio que incluya el nombre de la clase, versión y fecha; sentencia del paquete (Namespace) al cual pertenece; sentencia de importación (paquetes o librerías); utilización de sangría y que la longitud de la línea de comandos no sea superior a 80 caracteres; realización de las declaraciones en líneas independientes (para permitir comentarios sobre las mismas); utilización de espacios en blancos para permitir la legibilidad; los paquetes, se escribirán siempre en letras minúsculas para evitar que entren en conflicto con los nombres de clases o interfaces; Clases e interfaces deben tener los nombres de clases deben ser sustantivos y deben tener la primera letra en mayúsculas. Si el nombre es compuesto, cada palabra componente deberá comenzar con mayúsculas (también para los métodos y variables); los métodos deben ser verbos escritos en minúsculas; las constantes siempre en mayúsculas[[6]](#endnote-5)

Algunos ejemplos de estándares de codificación (5 establecidos para la seguridad y la protección).

**MISRA C / C ++:** Desarrollado por la Motor Industry Software Reliability Association, describe un subconjunto del lenguaje C o C ++ y las pautas para su uso para mejorar la seguridad de la aplicación. Aunque originalmente estaba destinado a aplicaciones automotrices, su uso creció en otras aplicaciones críticas para la seguridad.

**SEI / SANS CERT:** El Equipo de Respuesta a Emergencias Informáticas (CERT) del Instituto de Ingeniería de Software (SEI) tiene un conjunto de pautas para ayudar a los desarrolladores a crear software más seguro y confiable. Las pautas se dividen en importancia por grupos de "reglas" y "recomendaciones" y son muy completas y amplias e incluyen metadatos de riesgo.

**OWASP Top 10:** El Open Web Application Security Project (OWASP), como su nombre lo indica, es una organización que se compromete a mejorar la seguridad de las aplicaciones web. Como tal, su proyecto OWASP Top 10 proporciona una lista de las vulnerabilidades de seguridad de aplicaciones web más comunes y de mayor impacto. La última versión de OWASP Top 10 está directamente correlacionada con ID de CWE específicos e incluye metadatos de riesgo.

**Estándar de codificación C ++ de vehículos aéreos de combate de ataque conjunto (JSF AV):** Un estándar basado en un subconjunto de MISRA C específicamente para el programa JSF.

**CWE - Top 25 de enumeración de debilidades comunes:** CWE es una lista de debilidades de software descubiertas basada en el análisis de vulnerabilidades reportadas (CVE). El Top 25 enumera las debilidades de seguridad más comunes y peligrosas seleccionadas de la lista más grande de CWE, que son todos exploits que tienen una alta probabilidad de ocurrir y el impacto de explotar la debilidad es grande. CWE también incluye información sobre riesgos en forma de impacto técnico, lo que ayuda a comprender qué problemas son más importantes para su organización.

**Camel Case:** Estableciendo que la primera letra de cada palabra esté en mayúscula a excepción de la primera palabra, estando el resto de letras en minúsculas. Este tipo de notación está muy extendida, siendo su uso muy común tanto en la declaración de variables como en el nombre de funciones y métodos.

**Pascal Case:** Estableciendo que la primera letra de cada palabra esté en mayúscula, estando el resto de letras en minúsculas. Su uso es muy habitual en la definición de los nombres de las clases de múltiples lenguajes, como JavaScript o PHP.

**Snake Case:** Combina las palabras usando un guion bajo \_ como nexo. Existen dos versiones de esta notación, una en la que todas las letras están en mayúscula y otra en la que todas las letras están en minúscula. Esta notación, cuando se usa en mayúscula, es habitual en la declaración de constantes de lenguajes como PHP o JavaScript. En minúscula también es muy utilizada en la declaración de los nombres de los campos de las bases de datos. Además, también se utilizaba en la declaración de variables de PHP y, de hecho, aún es el estándar de muchos desarrolladores cuando programan plugins o temas para WordPress.

**Kebab Case:** Combina las palabras usando un guion - como nexo. Las letras estarán todas en minúscula. Debido a su buen resultado visual y a su simpleza es el estándar que se usa en la creación de URLs.

**¿Qué es un software para manejar versiones? ¿Para qué sirven y cuáles son los más comunes?**

Software implementado para realizar trabajos de manera grupal, el cual nos sirve para establecer de forma ordena y sin pérdida de información sobre uno o varios proyectos en cuestión, dicho software se puede configurar en interfaces como github, gitlab, entre otros.

Algunos de los más comunes son: Git, CVS, Apache subversión (SVN), mercurial, monotone.[[7]](#endnote-6)

**Utilice un software para manejar versiones en la siguiente pregunta**

**Cree un programa que le permita calcular el Fibonacci de un número, además, del numero resultante informar. si es primo o no? Utilice 2 estándares de codificación y uno utilícelo en java y el otro en .NET.**

https://github.com/JaimeGaviria30/Trabajo-Fabio.git

**¿Qué es PSP?**

Es una herramienta para el desarrollador que le permite medir los tiempos implementados en la elaboración de sus proyectos, los cuales a futuro le servirán para cotizar y estimar el tiempo y el costo de sus proyectos.

**¿Cómo estimo el tiempo de desarrollo?**

Se supone que el tiempo de desarrollo se debe estimar con herramientas como el PSP, de alguna manera los desarrolladores también se basan en su experiencia para realizarlo, ya que en ocasiones los softwares de PSP resultan ser demasiado obsesivos con la toma de los datos, además de ello el PSP es implementado en organizaciones donde se emplea el modelo CMMI.[[8]](#endnote-7)

**¿Qué es SCRUM?**

Es una metodología de trabajo grupal basada en iteraciones en donde se establecen metas a corto tiempo, aproximadamente de dos semanas, la idea es realizar constantemente entregas parciales del producto final, generando así resultados completos para facilitar la entrega final al cliente

**¿Cuáles son las fases de SCRUM?**

Requerimientos, backlog (que se hará), sprint planning (selección de requisitos), ejecución del sprint, inspección (revisión y retrospectiva)

**¿Defina que es una base de datos relacional?**

Una base de datos relacional es una recopilación de elementos de datos con relaciones predefinidas entre ellos. Estos elementos se organizan como un conjunto de tablas con columnas y filas. Las tablas se utilizan para guardar información sobre los objetos que se van a representar en la base de datos. Cada columna de una tabla guarda un determinado tipo de datos y un campo almacena el valor real de un atributo. Las filas de la tabla representan una recopilación de valores relacionados de un objeto o una entidad. Cada fila de una tabla podría marcarse con un identificador único denominado clave principal, mientras que filas de varias tablas pueden relacionarse con claves extranjeras. Se puede obtener acceso a estos datos de muchas formas distintas sin reorganizar las propias tablas de la base de datos.[[9]](#endnote-8)

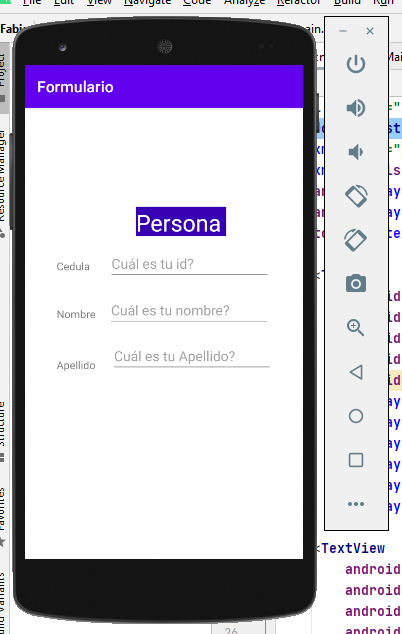
**¿Qué es un diagrama entidad relación?**

Es un diagrama en donde se establecen las relaciones entre objetos y/o personas (entidad) dentro de un sistema, normalmente utiliza rectángulos, rombos u óvalos (atributos) y líneas que representan las relaciones. Se usan para el diseño de bases de datos, solución de problemas, reingeniería.

**¿Que compone un Diagrama Entidad Relación?**

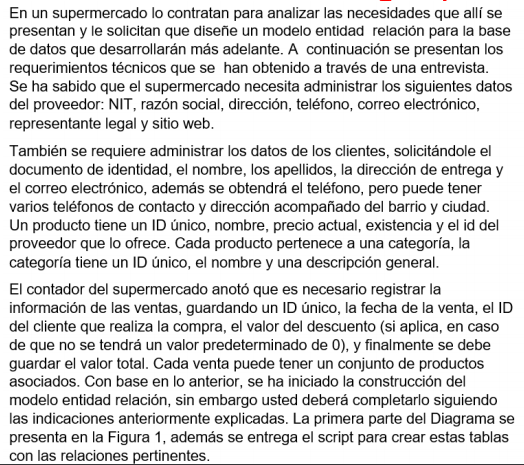
Entidad su tipo (nombre del objeto) de la misma y la categoría (débiles o asociativas); la Relación la cual puede incluso ser recursiva, atributos (simples o que no se pueden dividir, compuestos subatributos y derivados, como por ejemplo la edad que se puede calcular del atributo fecha de nacimiento) y Cardinalidad (uno a uno, uno a muchos y muchos a muchos)[[10]](#endnote-9)

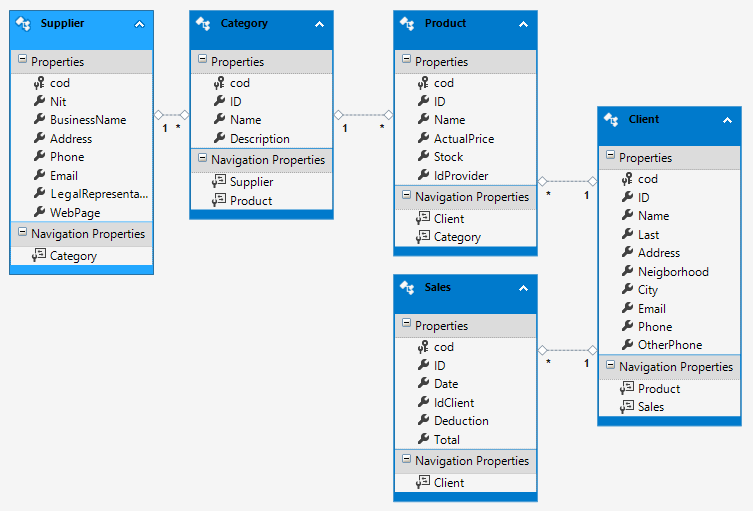
**Desarrolle una aplicación móvil que le permita ingresar la información de cedula, nombre y apellido de una persona.**

****

[**https://github.com/JaimeGaviria30/Trabajo-Fabio/tree/master/Documents/Jaime/Universidad%20-%20Ingenieria%20en%20Informatica/Semestre%205/Fabio/Fibonacci/Movil/Formulario**](https://github.com/JaimeGaviria30/Trabajo-Fabio/tree/master/Documents/Jaime/Universidad%20-%20Ingenieria%20en%20Informatica/Semestre%205/Fabio/Fibonacci/Movil/Formulario)

**Desarrolle el Diagrama Entidad Relación para el siguiente ejercicio. Los atributos deben ser en inglés y descriptivos.**

****

****

**Cree un diccionario de datos para el ejercicio anterior**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Nombre de la tabla:** | | | | | Client | | | |
| **Description de la tabla:** | | | | | Almacenará toda la información correspondiente de los clientes | | | |
| **Columnas de la table** | | | | | | | | |
| N° | | Nombre | Tipo de dato | Nulo | Description | | Valor por defecto | |
| 0 | | cod | | bigint | NN | Para facilitar la gestión sobre el visual studio, será autoincrementable | N/A | |
| 1 | | ID | | Int |  | Representa el código único | N/A | |
| 2 | | Name | | Varchar (50) |  | Nombre del cliente | null | |
| 3 | | Last | | Varchar (50) |  | Apellido del cliente | null | |
| 4  5 | | Address | | Varchar (50) |  | Dirección del cliente | null | |
| 5 | | Neighborhood | | Varchar (50) |  | Barrio donde reside el cliente | null | |
| 6 | | City | | Varchar (50) |  | Ciudad donde reside el cliente | null | |
| 7 | | Email | | Varchar (50) |  | Correo electrónico del cliente | null | |
| 8 | | Phone | | Varchar (50) |  | Teléfono de contacto | null | |
| 9 | | OtherPhone | | Varchar (50) |  | Segunda opción para un teléfono más | null | |
| **Llave primaria y de referencia** | | | | | | | | |
| **Nombre** | | | | | **Columna** | | | |
| PK | | | | | cod | | | |
| FK | | | | | ID (Product) / ID (Sales) | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nombre de la tabla: | | | | Category | | |
| Description de la tabla: | | | | Contendra la clasificación de los productos que se ofrezcan en el supermercado | | |
| Columnas de la tabla | | | | | | |
| N° | Nombre | | Tipo de dato | Nulo | Descripción | Valor por defecto |
| 0 | cod | | bigint | NN | Para facilitar la gestión sobre el visual studio, será autoincrementable | N/A |
| 1 | ID | | Int |  | Representa el codigo unico de la categoria | N/A |
| 2 | Name | | Varchar (50) |  | Nombre de la categoria que puede tener un producto. | null |
| 3 | Description | | Varchar (50) |  | Descripcion general de ia categoria. | null |
| Llave primaria y de referencia | | | | | | |
| Nombre | | | | Columna | | |
| PK | | | | cod | | |
| FK | | | | Nit (Supplier) / ID (Product) | | |
|  | |  | |  | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nombre de la tabla: | | | | Product | | |
| Description de la tabla: | | | | Contiene la información pertinente de los productos | | |
| Columnas de la tabla | | | | | | |
| N° | Nombre | | Tipo de dato | Nulo | Descripción | Valor por defecto |
| 0 | cod | | bigint | NN | Para facilitar la gestión sobre el visual studio, será autoincrementable | N/A |
| 1 | ID | | Int |  | Identificador del producto | N/A |
| 2 | Name | | Varchar (50) |  | Nombre del producto | null |
| 3 | ActualPrice | | Money | NN | Precio actualizado del producto | N/A |
| 4 | Stock | | int |  | Inventario existente del producto | null |
| 5 | IdProvider | | Int |  | Identificación del proveedor del producto | N/A |
| Llave primaria y de referencia | | | | | | |
| Nombre | | | | Columna | | |
| PK | | | | cod | | |
| FK | | | | ID (Client) / ID (Category) | | |
|  | |  | |  | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nombre de la tabla: | | | | Sales | | |
| Description de la tabla: | | | | Contiene la información de las ventas realizadas en el establecimiento | | |
| Columnas de la tabla | | | | | | |
| N° | Nombre | | Tipo de dato | Nulo | Descripción | Valor por defecto |
| 0 | cod | | bigint | NN | Para facilitar la gestión sobre el visual studio, será autoincrementable | N/A |
| 1 | ID | | Int |  | Identificador de las ventas | N/A |
| 2 | Date | | date | NN | Fecha de la venta | null |
| 3 | IdClient | | Varchar (50) |  | Identificación del cliente quien hace la compra | N/A |
| 4 | Deduction | | money |  | Posible descuento que se le puede ofrecer a los clientes | 0 |
| 5 | Total | | money | NN | Valor total de la venta | N/A |
| Llave primaria y de referencia | | | | | | |
| Nombre | | | | Columna | | |
| PK | | | | Cod | | |
| FK | | | | ID (Client) | | |
|  | |  | |  | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nombre de la tabla: | | | | Supplier | | |
| Description de la tabla: | | | | Información de localización y contacto del establecimiento | | |
| Columnas de la table | | | | | | |
| N° | Nombre | | Tipo de dato | Nulo | Descripción | Valor por defecto |
| 0 | cod | | bigint | NN | Para facilitar la gestión sobre el visual studio, será autoincrementable | N/A |
| 1 | Nit | | Int |  | Identificador comercial del supermercado | N/A |
| 2 | BussinessName | | Varchar (50) |  | Nombre del establecimiento | N/A |
| 3 | Address | | Money |  | Dirección del supermercado | null |
| 4 | Phone | | int |  | Teléfono de contacto con el supermercado | null |
| 5 | Email | |  |  | Correo electrónico del supermercado | null |
| 6 | LegalRepresentative | |  |  | Representante legal del negocio | null |
| 7 | WebPag | | Int |  | Página web del supermercado | null |
| Llave primaria y de referencia | | | | | | |
| Nombre | | | | Columna | | |
| PK | | | | cod | | |
| FK | | | | ID (Catergory) | | |
|  | |  | |  | | |

**¿Qué es un mockup?** Es un diseño (imagen) de por ejemplo un formulario de una página web, la que nos permite idealizar la forma final en que queremos ver dicho formulario desplegado

**¿Desarrolle los mockup de los formularios de gestión de la información del ejercicio 21?**

**¿Qué es SQL?**

**¿Qué es DML, DCL Y DDL?**

**¿Qué es integridad referencial?**

**¿Cree el Script de creación de tablas, llaves primarias, llaves foráneas del ejercicio planteado en la pregunta 21? (En Oracle, Postgres YSQL Server)**

**¿Qué es un procedimiento almacenado?**

**¿Cree los procedimientos almacenados de inserción y consulta de cada una de las tablas planteadas en el ejercicio 21? (En Oracle, Postgres Y SQL Server**

1. <https://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software> [↑](#footnote-ref-1)
2. <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf> parafraseo [↑](#endnote-ref-1)
3. <https://intelequia.com/blog/post/2083/ciclo-de-vida-del-software-todo-lo-que-necesitas-saber> [↑](#endnote-ref-2)
4. <https://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_requisitos> tomado de dicha página, complementado a partir de experiencias anteriores dentro del programa. [↑](#endnote-ref-3)
5. <https://software.projectmanager.com/project/plan/E> Creada en dicha página web. [↑](#endnote-ref-4)
6. <http://www.saludcapital.gov.co/Formatos%20Gua%20Desarrollo%20SDS/Estandares%20de%20Desarrollo%20de%20Software_SDS_v1.3.8.pdf> [↑](#endnote-ref-5)
7. <https://www.drauta.com/5-softwares-de-control-de-versiones> [↑](#endnote-ref-6)
8. Visto en el curso de Gestión de proyectos de SW [↑](#endnote-ref-7)
9. <https://aws.amazon.com/es/relational-database/> [↑](#endnote-ref-8)
10. <https://www.lucidchart.com/pages/es/que-es-un-diagrama-entidad-relacion/#section_0> [↑](#endnote-ref-9)