Conceptos

Software:

- Es toda parte digital detrás de un dispositivo, como los códigos que le permiten su correcto funcionamiento.

- Qué es un software, tipos y para qué sirve. SDI. https://sdindustrial.com.mx/blog/definicion-de-software-y-para-que-sirve/#:~:text=El%20software%20es%20un%20conjunto,operaciones%20de%20un%20Sistema%20Inform%C3%A1tico. (El software es un conjunto de programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un Sistema Informático.)

Ingeniería de Software:

- Es la ingeniería que se encarga del desarrollo, codificación y mantenimiento del software de los dispositivos electrónicos.

- Wikipedia. Ingeniería de software. https://es.wikipedia.org/wiki/Ingenier%C3%ADa\_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, y brindando soporte operacional y de mantenimiento.)

Git

- Es como una plataforma donde puedes subir tu código para producir como versiones, puedes modificarlo y cada vez se creará como una versión y podrás revisarlo.

- Atlassian. Qué es Git. https://www.atlassian.com/es/git/tutorials/what-is-git. (Git es un proyecto de código abierto maduro y con un mantenimiento activo que desarrolló originalmente Linus Torvalds. Un asombroso número de proyectos de software dependen de Git para el control de versiones, incluidos proyectos comerciales y de código abierto.)

Programación

- Se refiere a crear códigos o algoritmos para el funcionamiento del software de un dispositivo electrónico.

- Netec. ¿Qué es programación? https://www.netec.com/que-es-programacion (La programación informática es el arte del proceso por el cual se limpia, codifica, traza y protege el código fuente de programas computacionales, en otras palabras, es indicarle a la computadora lo que tiene que hacer.)

Lenguaje de Programación

- Es el idioma de la computadora por el cual logramos comunicarnos con ella.

- Chackray. Lenguajes de programación: tipos y características. https://www.chakray.com/es/lenguajes-programacion-tipos-caracteristicas/#:~:text=%C2%BFQu%C3%A9%20es%20un%20lenguaje%20de,comporte%20de%20una%20cierta%20manera. (Un lenguaje de programación es un conjunto de reglas gramaticales (tanto sintácticas como semánticas) que instruyen a que un ordenador o dispositivo se comporte de una cierta manera.)

Compilador

- Es el traductor entre el lenguaje de programación y el lenguaje binario de los dispositivos electrónicos.

- Compilador e intérprete: definición y diferencias. IONOS. https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/compilador-e-interprete/#:~:text=llamarse%20lenguajes%20interpretados.-,%C2%BFQu%C3%A9%20es%20un%20compilador%3F,c%C3%B3digo%20m%C3%A1quina%20antes%20de%20comenzar. (Es un programa informático que procesa el código fuente de un proyecto de software durante su tiempo de ejecución, es decir, mientras el software se está ejecutando, y actúa como una interfaz entre ese proyecto y el procesador.)

Ciclo de Desarrollo de Software y sus componentes

- Es como el ciclo de vida de un software y sus componentes son:

* Planificar: Consiste en crear un plan para un excelente desarrollo del proyecto.
* Análisis: Se refiere a analizar todo el proceso, como a quien va dirigido el software, de que manera se podría optimizar, que riesgos hay de que falle, etc.
* Diseño: Consiste en crear un boceto del programa.
* Codificación: Se refiere a crear el código de programación.
* Prueba: Se refiere a hacerle pruebas de diferentes dispositivos para que tenga un correcto funcionamiento.
* Mantenimiento: Se refiere al cuidado del correcto funcionamiento del software.

- Amazon. SDCL. AWS. https://aws.amazon.com/es/what-is/sdlc/#:~:text=El%20ciclo%20de%20vida%20del%20desarrollo%20de%20software%20(SDLC)%20describe,y%20corrigen%20errores%20del%20software. (El ciclo de vida del desarrollo de software (SDLC) es un proceso rentable y eficiente en términos de tiempo empleado por los equipos de desarrollo para diseñar y crear software de alta calidad. El objetivo del SDLC es minimizar los riesgos del proyecto por medio de una planificación anticipada que permita que el software cumpla las expectativas del cliente durante la fase de producción y posteriormente.)

* Planificación: La fase de planificación incluye normalmente tareas como análisis de costos y beneficios, programación, estimación de recursos y asignación. El equipo de desarrollo recopila requisitos de varias partes interesadas, como clientes, expertos internos y externos, así como directivos, para crear un documento de especificaciones con los requisitos del software.
* Diseño: En la fase de diseño, los ingenieros de software analizan los requisitos e identifican las mejores soluciones para crear el software. Por ejemplo, pueden plantearse la integración de módulos ya existentes, elegir la tecnología e identificar herramientas de desarrollo. Decidirán la mejor manera de integrar el nuevo software en cualquier infraestructura de TI existente que la organización pueda tener.
* Implementación: En la fase de implementación, el equipo de desarrollo codifica el producto. Se analizan los requisitos para identificar tareas de codificación más pequeñas que puedan hacerse diariamente para conseguir el resultado final.
* Prueba: El equipo de desarrollo combina las pruebas automáticas y manuales para comprobar si el software tiene errores. Los análisis de calidad incluyen probar el software para detectar errores y comprobar si cumple los requisitos del cliente. Dado que muchos equipos prueban inmediatamente el código que escriben, la fase de pruebas se ejecuta con frecuencia en paralelo a la fase de desarrollo
* Despliegue: Cuando los equipos desarrollan software, lo codifican y prueban en una copia diferente que no es a la que acceden los usuarios. El software que los clientes usan se llama producción, mientras que las otras copias están en el entorno de compilación o entorno de pruebas.
* Mantenimiento: En la fase de mantenimiento, entre otras tareas, el equipo corrige errores, resuelve problemas de los clientes y administra los cambios hechos en el software. Además, el equipo supervisa el rendimiento general del sistema, la seguridad y la experiencia del usuario para identificar nuevas maneras de mejorar el software existente.