



UNIVERSIDAD DEL SURESTE: DE LA FRONTERA COMALAPA.

DOCENTE: Ing. Liliana Lizeth Mejia Salas.

ASIGNATURA: Ingeniería de software.

ALUMNO: Ramiro Gerardo Resendíz Valdéz.

CUATRIMESTRE: Octavo (8^{vo}).

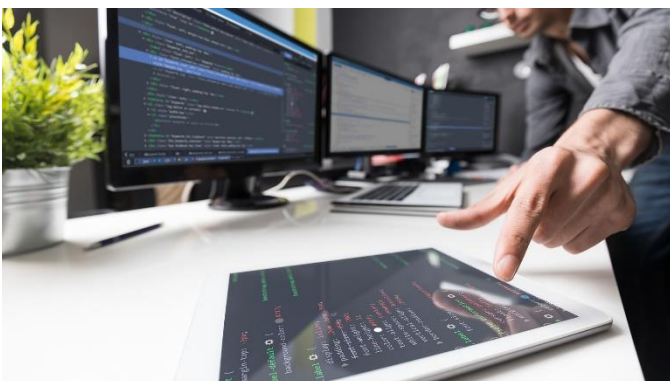
CARRERA: Ingeniería en sistemas computacionales.

GRUPO: ISC13SDC0220-A.

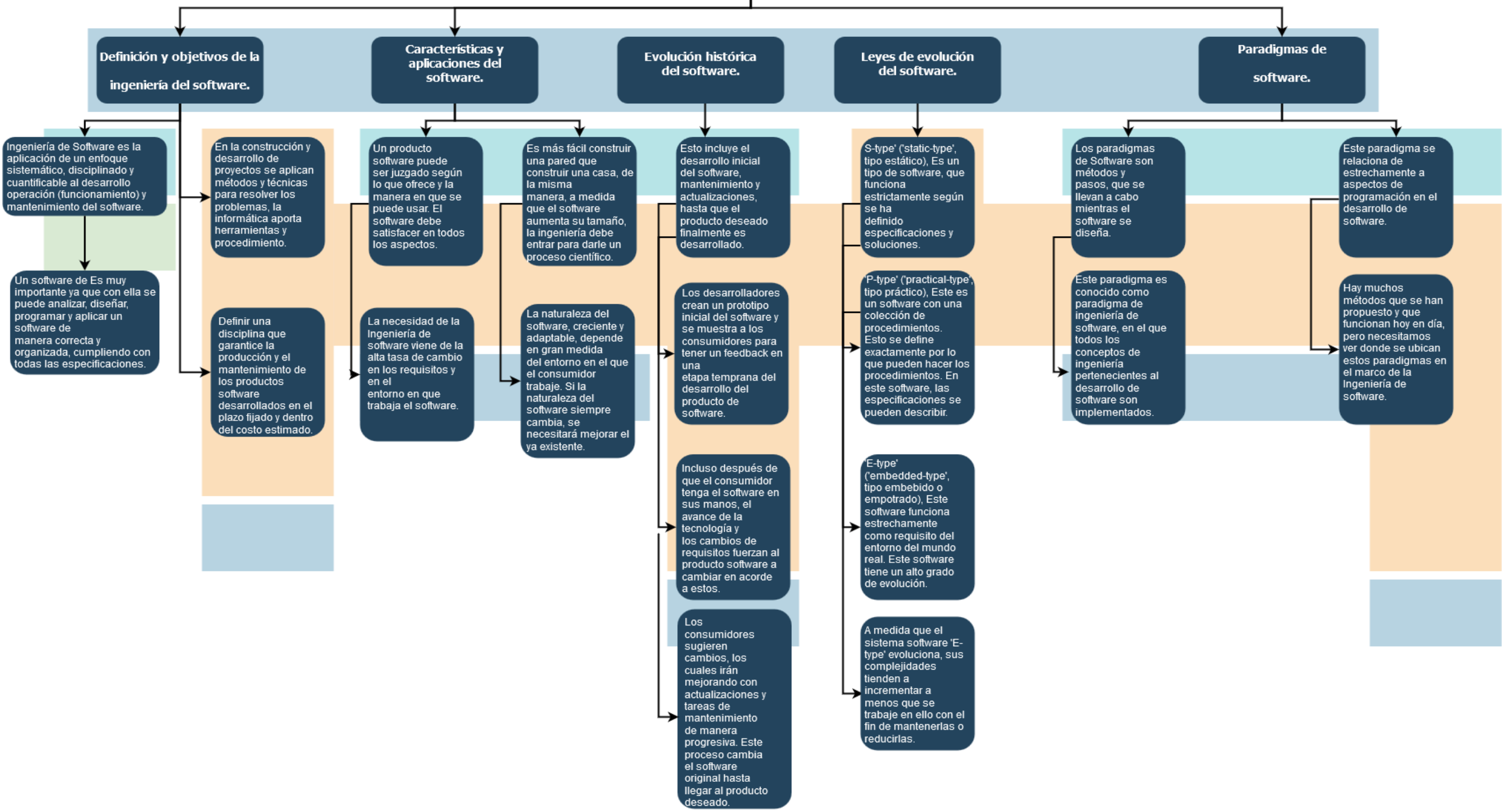
UNIDAD: Primera (1^{ra}).

TRABAJO: Mapa conceptual de la 1.1 a la 1.5 de la antología.

FECHA DE ENTREGA: 11/Enero/2023



Fundamentos de la ingeniería del software.



Definición y objetivos de la ingeniería del software.

Ingeniería de Software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo operación (funcionamiento) y mantenimiento del software.

Un software de Es muy importante ya que con ella se puede analizar, diseñar, programar y aplicar un software de manera correcta y organizada, cumpliendo con todas las especificaciones.

En la construcción y desarrollo de proyectos se aplican métodos y técnicas para resolver los problemas, la informática aporta herramientas y procedimiento.

Definir una disciplina que garantice la producción y el mantenimiento de los productos software desarrollados en el plazo fijado y dentro del costo estimado.

Características y aplicaciones del software.

Un producto software puede ser juzgado según lo que ofrece y la manera en que se puede usar. El software debe satisfacer en todos los aspectos.

La necesidad de la Ingeniería de software viene de la alta tasa de cambio en los requisitos y en el entorno en que trabaja el software.

Es más fácil construir una pared que construir una casa, de la misma manera, a medida que el software aumenta su tamaño, la ingeniería debe entrar para darle un proceso científico.

La naturaleza del software, creciente y adaptable, depende en gran medida del entorno en el que el consumidor trabaja. Si la naturaleza del software siempre cambia, se necesitará mejorar el ya existente.

Evolución histórica del software.

Esto incluye el desarrollo inicial del software, mantenimiento y actualizaciones, hasta que el producto deseado finalmente es desarrollado.

Los desarrolladores crean un prototipo inicial del software y se muestra a los consumidores para tener un feedback en una etapa temprana del desarrollo del producto de software.

Incluso después de que el consumidor tenga el software en sus manos, el avance de la tecnología y los cambios de requisitos fuerzan al producto software a cambiar en acorde a estos.

Los consumidores sugieren cambios, los cuales irán mejorando con actualizaciones y tareas de mantenimiento de manera progresiva. Este proceso cambia el software original hasta llegar al producto deseado.

Leyes de evolución del software.

S-type' ('static-type', tipo estático), Es un tipo de software, que funciona estrictamente según se ha definido especificaciones y soluciones.

P-type' ('practical-type, tipo práctico), Este es un software con una colección de procedimientos. Esto se define exactamente por lo que pueden hacer los procedimientos. En este software, las especificaciones se pueden describir.

E-type' ('embedded-type', tipo embebido o empotrado), Este software funciona estrechamente como requisito del entorno del mundo real. Este software tiene un alto grado de evolución.

A medida que el sistema software 'E-type' evoluciona, sus complejidades tienden a incrementar a menos que se trabaje en ello con el fin de mantenerlas o reducirlas.

Paradigmas de software.

Los paradigmas de Software son métodos y pasos, que se llevan a cabo mientras el software se diseña.

Este paradigma es conocido como paradigma de ingeniería de software, en el que todos los conceptos de ingeniería pertenecientes al desarrollo de software son implementados.

Este paradigma se relaciona de estrechamente a aspectos de programación en el desarrollo de software.

Hay muchos métodos que se han propuesto y que funcionan hoy en día, pero necesitamos ver donde se ubican estos paradigmas en el marco de la Ingeniería de software.