Definir el ciclo de vida de desarrollo aplicado a un caso práctico. Utilizar el proyecto

planteado en clase

Requerimientos

Diseño

Codificación y Test Unitario

Integración del sistema

Operación y Mantención

Modelo de ciclo de vida Cascada.

**RQ1:** El sistema debe permitir registrar los clientes. Los datos del cliente son: Nombre, dirección de vivienda, cédula de identidad, correo electrónico, y número de teléfono celular.

**RQ2:** El sistema debe permitir almacenar la información del cliente ya ingresado en la computadora

**RQ3:** El sistema debe contar con funcionalidades de control de caja como: añadir y quitar artículos, valor del artículo.

**RQ4:** El sistema debe permitir ingresar la información del registro del trabajador. Los datos a ingresar son: nombre, cédula de identidad, correo electrónico, número celular, dirección de vivienda, id de empleado.

**RQ5:** El sistema debe permitir ingresar la cantidad de billetes y monedas en caja al final de cada día laborable, así como realizar el cálculo del arqueo de caja

**RQ6:** El sistema debe crear una factura a partir de los productos vendidos al cliente. La información colocada en la factura debe incluir nombres completos, cédula de identidad, correo electrónico, dirección de vivienda, y teléfono, así como los datos de la empresa: dirección del establecimiento, numero del teléfono celular, RUC, nombre del trabajador y un desglose de los productos que el cliente compró. Se debe considerar el 12% del IVA en cada factura.

¿Por qué escogimos este modelo de ciclo de vida cascada?

El modelo de ciclo de vida en cascada se eligió, porque los requerimientos del software a desarrollar están claros, al inicio estaban aun por definirse, pero luego de una reunión en grupo se lograron aclarar todos los requerimientos que el usuario necesita para su local y no será necesario estar regresando a la especificación de requerimientos.

Dicho modelo nos ayudará a tener una idea clara de cada etapa que vamos haciendo, ya que, una vez terminada la una fase, pasaremos a la siguiente fase y los resultados de la fase anterior pasa a dicha fase.

Este modelo nos permitirá optimizar el tiempo al momento de desarrollar e implementar el software, porque hasta que una fase no esta bien culminada no podremos pasar a la siguiente así evitaremos tener errores de una fase anterior en el futuro.