Ciclo de vida (Modelo iterativo)

Sistema de facturación

**Alcance de producto:**

* Se desarrollará un sistema web que permitirá a los usuarios realizar la facturacion para su local, teniendo una lista de clientes que ya estén registrados en el sistema pudiendo encontrarlos y añadiéndoles sus facturas, a su vez también pudiendo modificarlas en caso de ser necesario, creando los detalles de venta de cada una de ellas

**Alcance del proyecto:**

* **Objetivo:** Crear aplicación web funcional para realizar la facturación de los clientes de un local
* **Objetivos Especifico:**
  + El usuario puede añadir facturas a los distintos clientes
  + Se va a poder modificar cualquiera de las facturas de un cliente dado por el usuario
  + Al momento de añadir una facturación se creará con los detalles de la fecha de venta y el aporte monetario de la misma, a su vez teniendo una descripción del producto
  + Se creará la facturación de un producto dado y se le asignará al cliente
* **Supuestos:**
  + Suponemos que todo el equipo de trabajo se encuentra capacitado para el desarrollo del proyecto
  + Se disponen de los recursos necesarios y prototipos a seguir
  + Se implementarán medidas de seguridad para asegurar la privacidad de cada cliente
* **Restricciones:** 
  + Se estima un costo aproximado del proyecto $ 200.000
  + Se cuenta con un tiempo límite de 3 meses
  + El pago del proyecto se realizará en 2 entregas, uno al comenzar y otro al finalizar el proyecto

**Versión 1: (Modulo no funcional)**

* **Captura de requisitos:**
  + Entrevista con el cliente
  + Diagrama de Casos de Uso
  + Cliente: ABMC, encontrarCliente
  + Producto: ABMC, obtenerPrecio
* **Análisis:**
  + Cliente: código, dni, domicilio, nombre
  + Producto: código, descripción, nombre, precio
* **Diseño:**
  + Diagrama de clases detallado:
    - Cliente: encontrarCliente()
    - Producto: getPrecio()
  + Base de datos:
    - Tabla Cliente: {int código, int dni, varchar(50) domicilio, varchar(50) nombre}
    - Tabla Producto: {int codigo, varchar(50) descripcion, varchar(50) nombre, double precio}
  + Vistas:
    - Vista ABMC cliente.
    - Vista ABMC producto.
* **Implementación:**
  + Cliente: boolean:encontrarCliente(double), boolean:Holaaaa().
  + Producto: float:getPrecio(String)
* **Testing:**
  + Caja negra
  + Caja blanca
* **Entregables:**
  + **ABMC Cliente, ABMC producto**

**Versión 2: (Modulo no funcional)**

* **Captura de requisitos:**
  + Factura: addFactura, modificarFactura
* **Análisis:**
  + Factura: Cliente, código, fecha, montoTotal, producto
* **Diseño:**
  + Diagrama de clases detallado:
    - Factura: addFactura(), modificarFactura(), getter, setter.
  + Base de datos:
    - Tabla Factura: {int codigo, Date fecha, doublé montoTotal, varchar(50) destino, FORENGKEY(id\_producto), FORENGKEY(id\_cliente)}
  + Vistas:
    - Vista ABMC Factura
* **Implementación:**
  + Factura: void:addFactura(int, int, float, float), Cliente:modificarFactura(String, int), getters, setters.
* **Testing:**
  + Caja negra + Caja negra versión 1
  + Caja blanca + Caja blanca versión 1
* **Entregables:**
  + **ABMC factura, Vista de facturación**

**Version 3:**

* **Captura de requisitos:**
  + Cliente: ABMC factura
  + producto: ABMC factura
  + detalleFactura: CrearVenta
* **Analisis:**
  + detallefactura: cantidad, codigoDetalle, descripción, precio
* **Diseño:**
  + Diagrama de clases detallado:
    - detalleFacturacion: crearVenta()
  + Base de datos:
    - Tabla detalleFactura: {int cantidad, int codigoDetalle, varchar(50) descripcion, double precio, FORENGKEY(id\_factura)}
  + Vistas:
    - Vista detalleFactura
* **Implementación:**
  + detalleFactura: void:crearVenta(date, float)
* Testing
  + Caja negra versión 3 + Caja negra versión 2
  + Caja blanca versión 3 + Caja blanca versión 2
* **Entregables**
  + vistas detalleFactura, tabla de base de datos, diagrama de clases de factura, pruebas completas de testing

