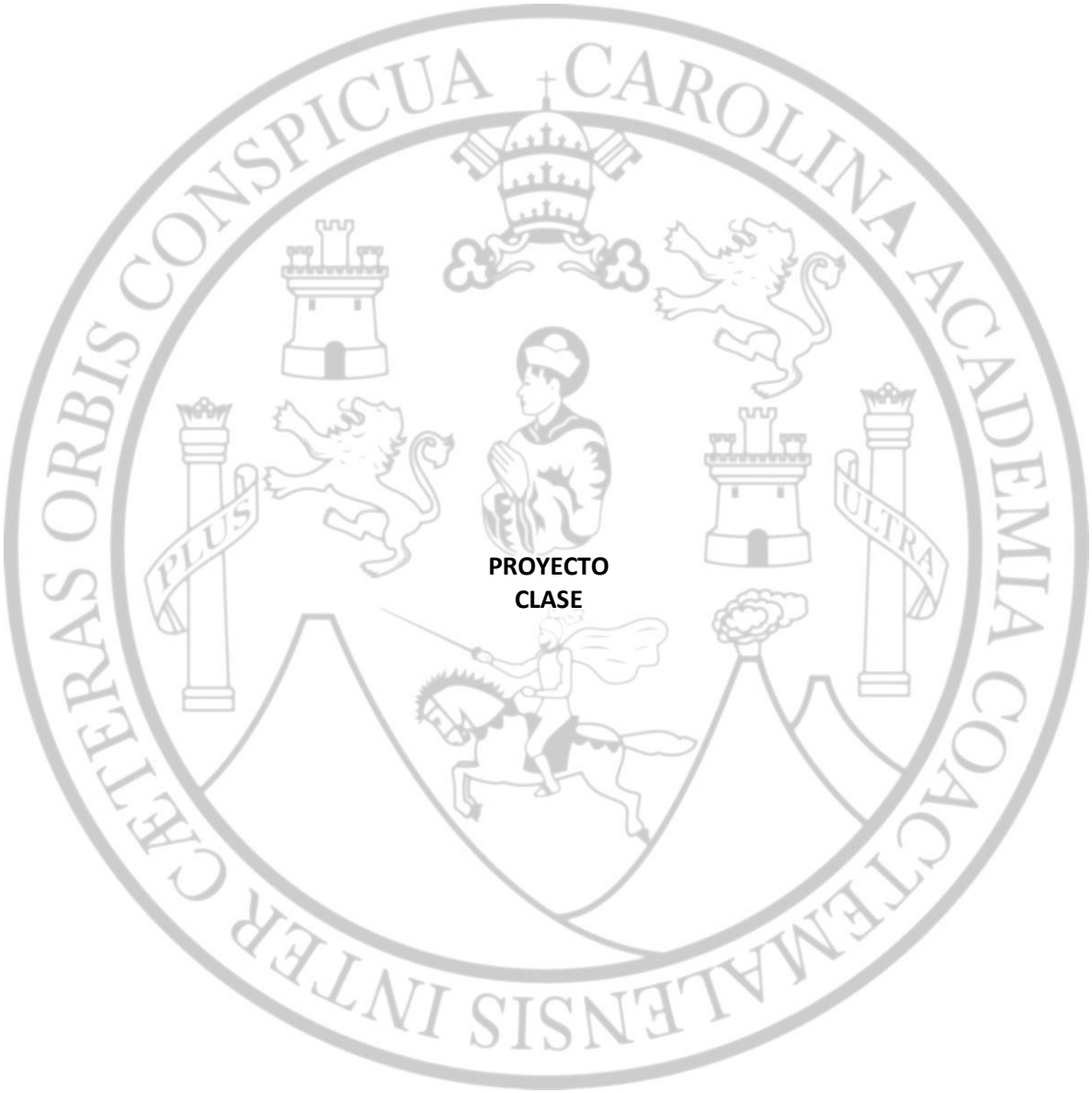
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERIA
ESCUELA DE CIENCIAS Y SISTEMAS
EDD

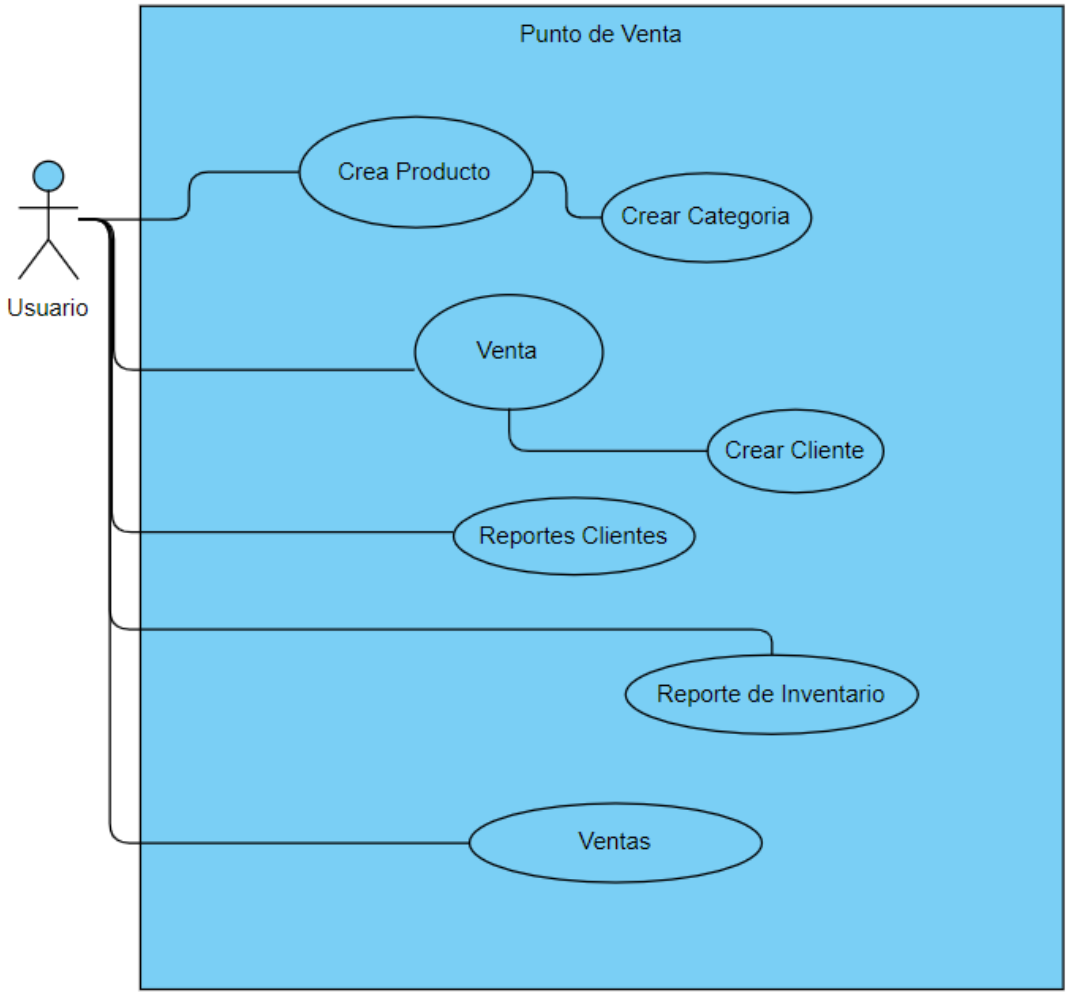


PROYECTO
CLASE

RAFAEL ANGEL CHOCOJ XINICO
201314826

Análisis

Casos de uso



Implementación

Requerimientos De Entorno de Programación

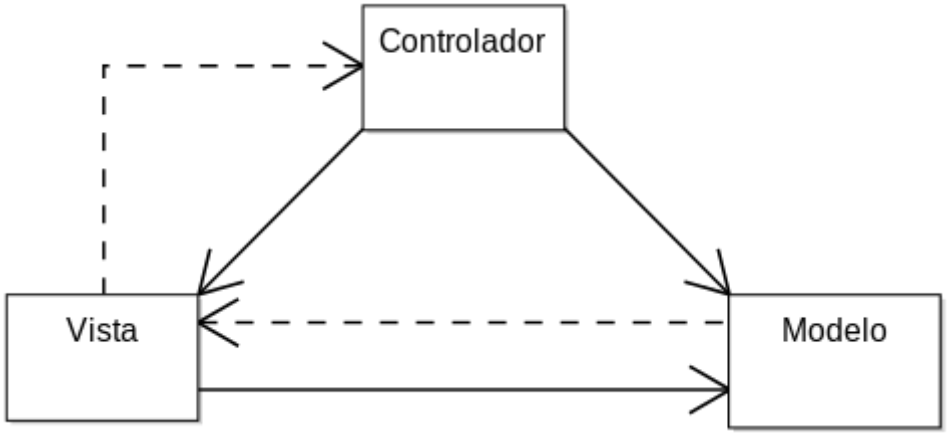
- Windows 8.1, Windows 7, Win10
- NetBeans IDE
- Java

Requerimientos De Instalación

- Windows 8.1, Windows 7, Win10
- Java, Maquina Vital
- jdk-8u73-windows-x64
- librería itextpdf
- PDF

Arquitectura Propuesta

Modelo vista controlador



Modelo-vista-controlador (MVC) es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado, define componentes para la representación de la información, y por otro lado para la interacción del usuario.

Modelo: Es la representación de la información con la cual el sistema opera, por lo tanto, gestiona todos los accesos a dicha información, tantas consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación. Las peticiones de acceso o manipulación de información llegan al modelo a través del controlador.

Controlador: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al modelo cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro). También puede enviar comandos a su vista asociada si se solicita un cambio en la forma en que se presenta el modelo.

Vista: Presenta el modelo (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario), por tanto, requiere de dicho modelo la información que debe representar como salida.

Mantenimiento

Adaptable – modificar el sistema para hacer frente a cambios en el ambiente del software.
Perfectivo – implementar nuevos, o cambiar requerimientos de usuario referentes a mejoras funcionales para el software
Correctivo – diagnosticar y corregir errores, posiblemente los encontrados por los usuarios.