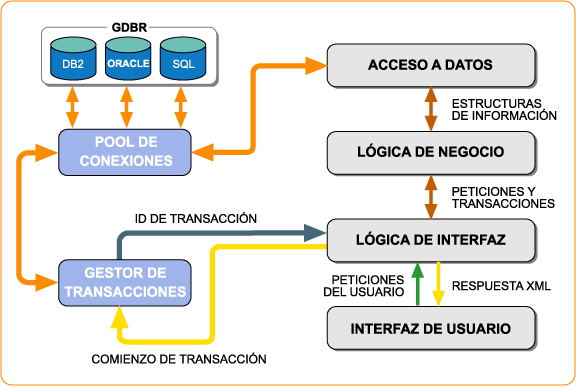
**ESTUDIO DE CASO VIRTUALMIND**

**SOFTWARE DE BIBLIOTECA VIRTUAL**

**KEVIN HERRERA & ANDRES GUTIERREZ**

1. **Que metodología se ha seleccionado para el diseño del software**

Metodología por capas

****

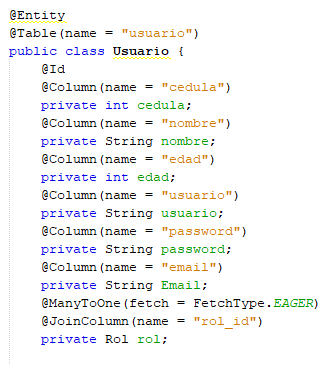
La metodología seleccionada es el modelo por capas, el cual separa por partes los componentes de un sistema de software.

**La forma en la que la implementamos fue la siguiente:**

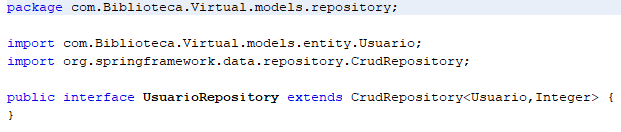
1. Creación de la base de datos (MariaDB)
2. Mapeo de tablas (JAVA)
3. Creación de las QUERYS (PETICIONES/LOGICA DE NEGOCIO)
4. Creación de lógica de interfaz (Visual Studio Code) <----->Creación de la interfaz de usuario.

**Lógica de Negocio**

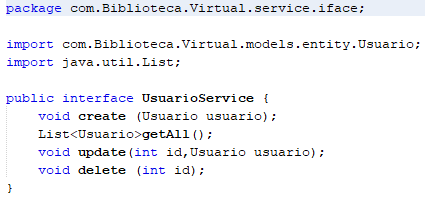
* **Models:**

** Entity:**

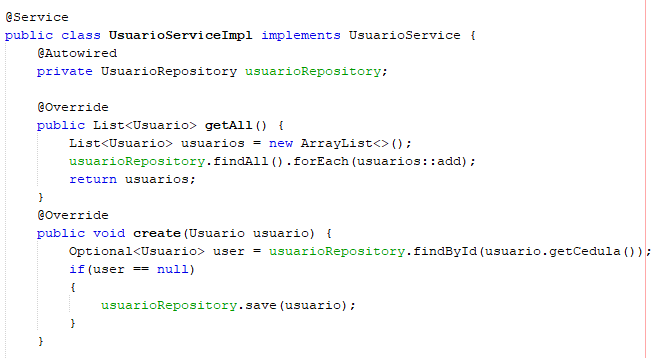
**Repository:**



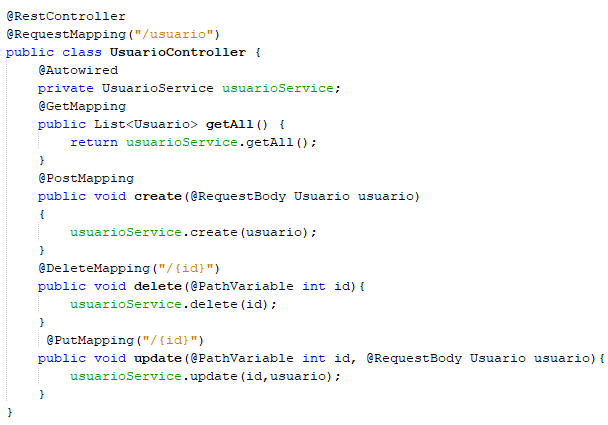
* **Service:**

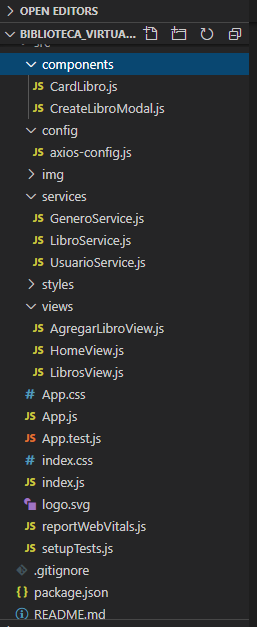
** Iface:**

**Implementación:**

****



* **Controller:**
* **Lógica de Interfaz/Interfaz de Usuario**



1. **Identificación de los stakeholders**

Tipo interno:

Administradores, Programadores,

Tipo externo:

Usuarios (Lectores/Público en general).

1. **Roles del proceso de desarrollo de software**

* Administradores: Kevin Herrera, Andrés Gutiérrez
* Programadores: Kevin Herrera, Andrés Gutiérrez
* Analistas: Kevin Herrera, Andrés Gutiérrez
* Diseñador: Andrés Gutiérrez
* Documentador: Kevin Herrera

1. **Técnicas para la recolección de requisitos**

**Tormenta de ideas(brainstroming)**

Es una técnica de reuniones en grupo cuyo objetivo es la generación de ideas en un ambiente libre de críticas o juicios.

**Introspección**

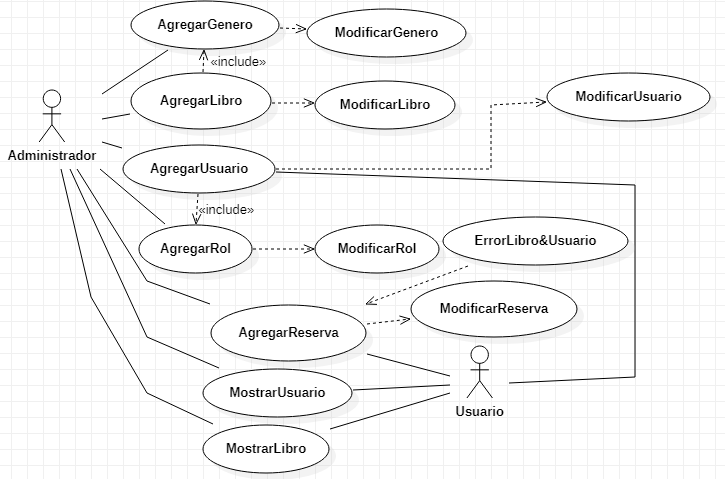
Esta técnica recomienda que el analista se ponga en el lugar del interesado y trate de imaginar cómo desearía éste la aplicación de software.

**Análisis de datos**

Consiste en obtener la información sobre los [requerimientos funcionales](http://www.pmoinformatica.com/2018/05/que-es-requerimiento-funcional.html) y [requerimientos no funcionales de software](http://www.pmoinformatica.com/2015/05/requerimientos-no-funcionales-ejemplos.html).

1. **Casos de Uso**

* Diagrama de Casos de uso



1. **Modelado de sistemas**

* Herramientas Case de Documentación: Microsoft Word
* Herramientas Case de Codificación: Zulu, Apache NetBeans IDE, Visual Studio Code
* Herramientas Case de Modelación de Bases de Datos: DBeaver, MariaDB
* Herramientas Case de Diagramación: StarUML

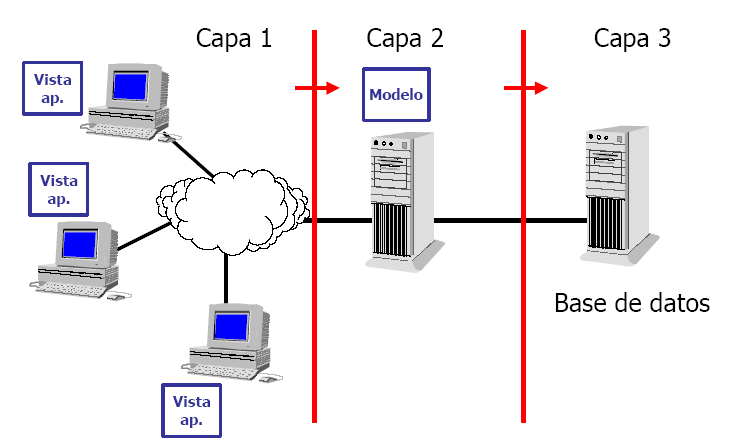
**Modelo de Espiral**

La Espiral es visualizada como un proceso que pasa por un cierto número de iteraciones, con el representante diagrama de cuatro cuadrantes de las siguientes actividades:

1. formular planes para: identificar los objetivos de software seleccionados para aplicar el programa, aclarar las restricciones de desarrollo del proyecto;
2. Risk análisis: un análisis y evaluación de los programas seleccionados, para examinar cómo identificar y eliminar el riesgo;
3. La puesta en práctica del proyecto: la puesta en práctica del desarrollo de software y la verificación;

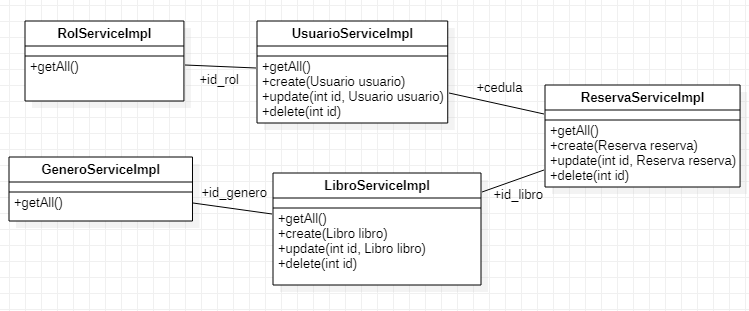
El modelo en espiral dirigida por riesgos, haciendo hincapié en las condiciones de las opciones y limitaciones a fin de apoyar la reutilización del software, la calidad del software puede ayudar como un objetivo especial de integración en el desarrollo de productos.

1. **Arquitectura del Software**

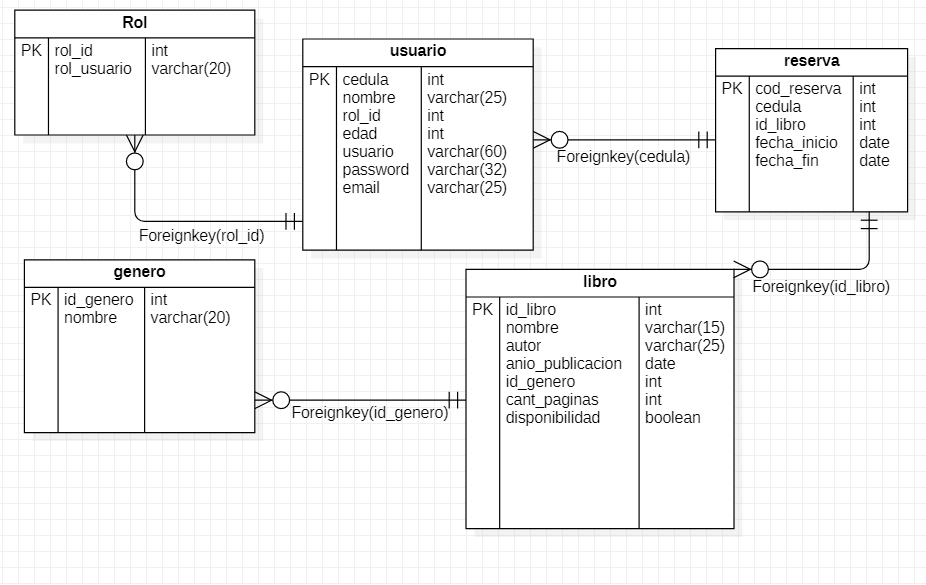


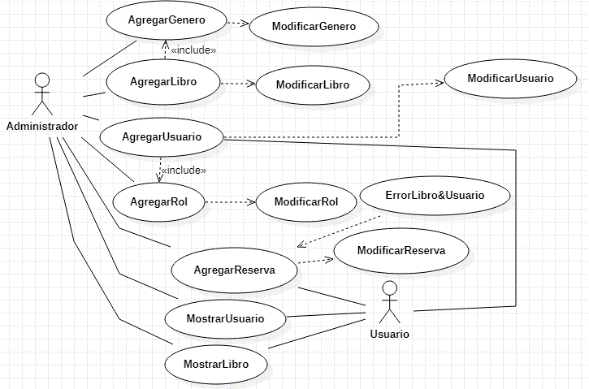
1. **Modelo 4+1**

**Logical View**

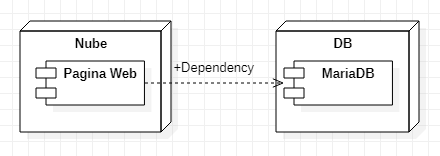


**Development View**



**Scenarios (Casos de Uso)**

**Physical View**



**Process View**

