**Ejercicios Análisis Requerimientos Funcionales y No Funcionales**

**Ejercicio 6**

**Objetivos:** portal en línea para mejorar la experiencia de sus clientes y optimizar sus operaciones internas.

**Requerimientos Funcionales:**

* Los clientes podrán:
  + consultar pólizas de seguro
  + realizar reclamaciones
  + actualizar sus datos personales
  + pagar primas
  + acceso a información detallada sobre coberturas y exclusiones para diferentes tipos de seguros (incluyendo vehículos, hogar, y vida).
* Agentes de seguros podrán:
  + gestionar las cuentas de los clientes
  + procesar reclamaciones
  + generar informes personalizados sobre las actividades del cliente y las tendencias de seguros
* El sistema debe poder:
  + ofrecer funcionalidades automatizadas para:
    - la renovación de pólizas
    - emisión de recordatorios antes de la expiración de las pólizas de seguro

**Requerimiento NO Funcionales:**

* Seguridad: autenticación robusta y medidas de protección de datos para garantizar la privacidad y seguridad de la información del cliente. Deberá cumplir con las regulaciones locales e internacionales sobre la protección de datos.
* Disponibilidad: portal debe ser accesible 24/7, con un tiempo de actividad garantizado del 99.9%.
* Rendimiento: capaz de soportar altos volúmenes de usuarios simultáneos, especialmente durante los periodos de renovación de pólizas.
* Usabilidad: interfaz de usuario debe ser clara y fácil de usar en una variedad de dispositivos

**Ambigüedades:**

Funcionalidad automatizada para la renovación de pólizas de cara a clientes o agentes?- De cara a clientes, suponemos que es en la cuenta del cliente, cuya póliza se renueva automáticamente.

Recordatorios es para cliente o para agentes?- es para clientes y se envían por el mail del cliente

**Para la creación del Diagrama**

**Clases/Entidades**

**Clase Persona:**

**Atributos:** +nombre: String, +apellidos: String, +id: Int, +status: String, +correo: String, +telefono: Int, +direccion: String, -contrasenya: String, +fechaNacimiento: Date, +rol: String

**Métodos:** +crearUsuario(String nombre, String apellidos, Int id, String status, String correo, Int telefono, String direccion, String contrasenya, Date fechaNacimiento, String rol): Boolean, +eliminarUsuario(Int ID) : void,-iniciarSesion(String correo, String contrasenya): Boolean, +get y +set para todos los atributos, -cerrarSesion(): void, + modificarDatos():Boolean,

**Subclase Cliente:** Hereda de Clase Persona

**Atributos:** +dni: String, +edad: int, + idPolizaContratada:List, +status: String, nombrePolizaContratada:List<Poliza>, +prima: float, +fechaCaducidad: Date <Poliza>, +reclamaciones:List<Reclamacion>

**Métodos:** +busquedaPolizaNombre(String nombre):List<Poliza>, +busquedaPolizaTipo(String tipo):List<Poliza>, +busquedaPolizaCobertura (List coberturas):List<Poliza>, +calculoPrimaPorCliente(Float precio): float, +realizarPago( ): boolean

**Subclase AgenteSeguro:** Hereda de Clase Persona

**Atributos:** +ClientesActivos:List<Cliente>, +reclamaciones:List<Reclamacion> , +polizasContratadas:List<Cliente>

**Métodos:** +crearCliente(String nombre, String apellidos, Int id, String status, String correo, Int telefono, String direccion, String contrasenya, Date fechaNacimiento, String rol): Boolean, +gestionarReclamacion(Int id, String status): Boolean, +generarInforme(Cliente cliente, Poliza poliza, Reclamacion reclamacion): File

**Clase Reclamacion:**

**Atributos:** +nombre: String, +id:int, +tipo: String, +status: String,

**Métodos:** +crearReclamacion(): Boolean, +filtroReclamacionPorTipo(): Boolean

**Clase Poliza**

**Atributos:** +nombre: String, +id: int, +tipo: String +precio: float, +coberturas:List, +exclusiones:List, +requisitos:List, +fechaCaducidad: Date, +get y set para todos los atributos

**Métodos:** +contratarPoliza(String tipo, Float precio, Date fechaCaducidad, Cliente): boolean, +renovarPoliza(String tipo, Float precio, Date fechaCaducidad, Cliente): boolean

**Clase Notificacion:**

**Atributos:** +fecha:Date, +text: String, +id: Int

**Métodos:** +generarNotificacion(String nombre, String tipo, Date fechaCaducidad): String (mensaje), +enviarNotificacion(String correo): void (enviaria las notificaciones por correo)

**Relaciones:**

**Tipos de relaciones:**

Persona –Cliente: Herencia

Persona – AgenteSeguro: Herencia

AgenteSeguro-Cliente: Asociación 🡪 Cardinalidad: 0..\*

Cliente-AgenteSeguro: Asociación 🡪 Cardinalidad: 1..\*

AgenteSeguro – Poliza: Asociación 🡪 Cardinalidad: 1..\*

Poliza – AgenteSeguro: Asociación 🡪 Cardinalidad: 1..\*

AgenteSeguro-Reclamacion: Asociación 🡪 Cardinalidad: 0..\*

Reclamacion- AgenteSeguro: Asociación 🡪 Cardinalidad: 1

Cliente-Poliza: Asociación 🡪 Cardinalidad: 1..\*

Cliente- Reclamacion: Composición 🡪 Cardinalidad: 0..\*

Cliente-Notificacion: Asociación 🡪 Cardinalidad: 0..\*