# **Formulario: Vista Lógica del Proyecto de Software**

**Proyecto:** Madu – Sistema de Gestión de Personas para Pymes  
**Grupo:** 009  
**Fecha:** 23/10/2024

**1. Descripción General del Sistema**

El sistema consiste en un sitio web que se podrá acceder tanto como usuario normal como para los empleadores y sus empleados. La lógica principal detrás de Madu, además de estar enfocado en la gestión de personas para pymes, también es ayudar a las empresas a publicar ofertas laborales y agilizar el proceso de reclutamiento a través de cuestionarios especializados según el cargo.

**Descripción del sistema:**

En el sistema, el usuario podrá acceder mediante el sitio web registrándose o iniciando sesión con su cuenta, como también registrándose en el momento de él encontrar un código QR de la empresa mostrando los cargos y ofertas laborales a disposición para postular. Después de esto, el usuario tendrá acceso a un dashboard/panel de control dónde podrá gestionar sus datos y postulaciones, como también, el empleador podrá gestionar sus empleados y ofertas laborales publicadas.   
Cabe destacar que Madu no pretende ser una competencia directamente de LinkedIn ni ser similar. Más bien, el valor agregado de nuestro sistema es el módulo de reclutamiento y el Onboarding del postulante enfocado principalmente para las pequeñas y medianas empresas.

**2. Principales Clases y Entidades del Sistema**

Describe las clases o entidades principales del sistema, sus atributos y métodos principales, y su propósito dentro del sistema.

**Clase 1: Usuario**

Atributos: **email, nombreUsuario, rol, contraseña, fechaRegistro**   
Métodos: **+login(), +editarPerfil(), +logout()**   
Propósito: **Rol de usuario básico para acceder al sistema de Madu.**

**Clase 2: Empleado**   
Atributos: **idEmpleado, nombre, apellido, direccion, telefono, fechaIngreso, puesto**   
Métodos: **+login(), +editarPerfil(), +logout()**   
Propósito: **Representación en base de datos del empleado de una Pyme.**

**Clase 3: Super Admin**   
Atributos: **idSuperAdmin, nivelAcceso**   
Métodos: **+gestionarComponentes()**    
Propósito: **Administrador con privilegios máximos para gestionar todos los componentes del sistema.**

**Clase 4: Perfil**   
Atributos: **idPerfil, foto, informacionAdicional**   
Métodos: **+editarPerfil()**   
Propósito: **Gestionar la información del perfil de los usuarios en el sistema.**

**Clase 5: Postulaciones**   
Atributos: **idNomina, fechaGeneracion, monto**   
Métodos: **+ver(), +editarPostulacion(), +eliminar()**   
Propósito: **El usuario gestiona sus postulaciones a ofertas laborales.**

**Clase 6: Rol**   
Atributos: **idRol, nombreRol**   
Métodos: **+agregarRol()**   
Propósito: **Definir los diferentes roles y permisos en el sistema.**

**Clase 7: Asistencia**   
Atributos: **idAsistencia, fecha, horaEntrada, horaSalida, estado**   
Métodos: **No Aplica**   
Propósito: **Registrar y controlar la asistencia de los empleados.**

**Clase 8: Empresa**   
Atributos: **idEmpresa, nombre, rut, direccion**   
Métodos: **No Aplica**   
Propósito: **Almacenar la información de las empresas (Pymes) registradas en el sistema.**

**Clase 9: Contacto**   
Atributos: **nombre, apellido, email, teléfono, región, empresa, empleados, área, fechaderegistro**   
Métodos: **+ver(), +editarContacto(), +eliminar()**   
Propósito: **Gestionar la información de contacto de las empresas que quieren incorporar Madu.**

**Clase 10: Empleador**   
Atributos: **idEmpleador, nombre, apellido, run, cargo, fechaContratacion**   
Métodos: **+login(), +editarPerfil(), +logout() , +administrarEmpleados()**  
Propósito: **Gestionar la información de su PYME y todo lo relacionado con esta.**

**3. Relaciones entre clases**

1. **Usuario (Clase Base) y sus especializaciones:**

* Un Usuario "es" un Empleado (Herencia): Representa los empleados de las PYMES que usarán el sistema para marcar asistencia, solicitar permisos, ver nóminas, etc.
* Un Usuario "es" un Empleador (Herencia): Representa a los dueños o administradores de PYMES que gestionarán el sistema.
* Un Usuario "es" un Super Admin (Herencia): Representa al administrador del sistema MADU que puede gestionar todos los componentes.

1. **Usuario con otras clases:**

* Un Usuario "tiene" un Perfil (Agregación): Almacena información adicional como foto e información complementaria del usuario.
* Un Usuario "tiene" un Rol (Agregación): Define los permisos y accesos dentro del sistema.
* Un Usuario "tiene" muchas Postulaciones (Agregación): Gestiona el proceso de reclutamiento mencionado en las características principales.

1. **Empleador con Empresa:**

* Un Empleador "Administra" una Empresa (Asociación): Refleja la capacidad de gestión de los administradores sobre su PYME.

1. **Empleado con Asistencia:**

* Un Empleado "registra" Asistencia (Asociación): Permite el control de asistencia mediante la app móvil, como se menciona en las características principales.

1. **Super Admin con Contacto:**

* Un Super Admin "Gestiona" los Contacto (s) (Asociación): Permite la administración y comunicación con las PYMES interesadas en contratar el servicio.

1. **Relaciones de la clase Empresa:**

* Una Empresa es administrada por Empleador: Refleja la estructura organizacional de la PYME.
* Una Empresa tiene relación con Contacto: Permite a la administración del sistema la comunicación con las PYMES interesadas en contratar el servicio.

Este diagrama de clases soporta las funcionalidades descritas en el documento, como:

* Gestión de empleados
* Control de asistencia
* Sistema de autenticación y roles
* Gestión de perfiles
* Administración empresarial
* Reclutamiento y onboarding

**4. Interfaces y dependencias**

Indica si alguna de las clases o entidades tiene interfaces y cómo interactúan con otros módulos del sistema a través de ellas.

**Descripción de interfaces y dependencias:**

**1.- Interfaz del Usuario:**

**export** **interface** **User** {

uid?: string;

nombres: string;

apellidos: string;

password?: string;

email: string;

telefono: string;

region: string;

ciudad: string;

rut: string;

rol: **UserRole**;

genero: **Gender**;

estadoCuenta: **AccountStatus**;

fechaCreacion: **Date**;

ultimoAcceso: **Date**;

}

**1.1 Dependencias de la Interfaz Usuario:**

**export** **enum** **UserRole** {

EMPLEADOR = 'Empleador',

EMPLEADO = 'Empleado',

USUARIO = 'Usuario',

SUPER\_ADMIN = 'Super Admin'

}

**export** **enum** **Gender** {

MASCULINO = 'Masculino',

FEMENINO = 'Femenino',

OTRO = 'Otro',

NO\_ESPECIFICA = 'Prefiero no especificar'

}

**export** **enum** **AccountStatus** {

ACTIVA = 'Activa',

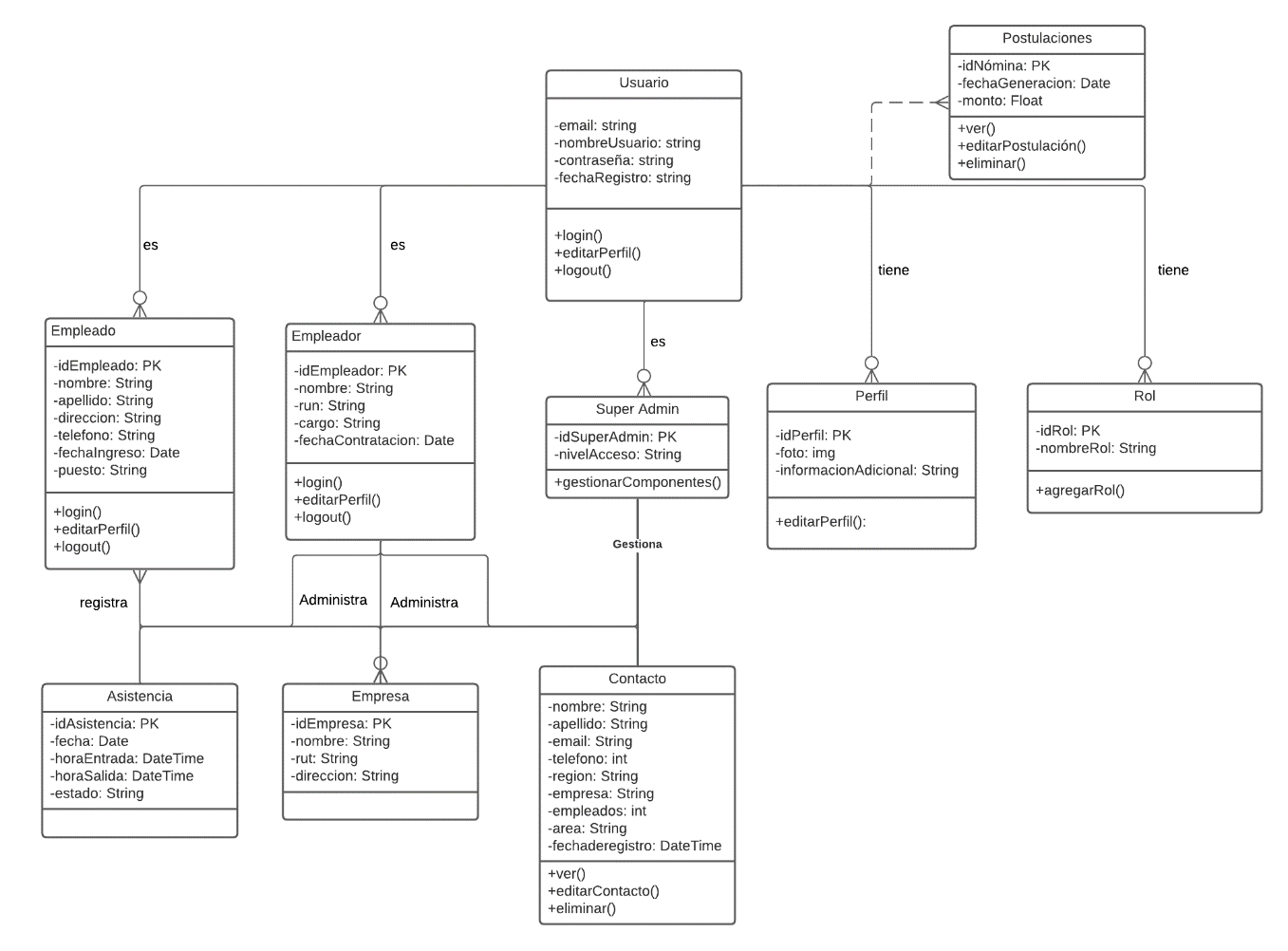
BLOQUEADA = 'Bloqueada',

ELIMINADA = 'Eliminada'

}

**5. Diagrama de Clases del Sistema**

Adjunta un diagrama de clases que representa la estructura lógica del sistema, las clases principales y las relaciones entre ellas.

**Diagrama de clases:**

**6. Diagrama de componentes (opcional)**

Si su sistema lo requiere, adjunte un diagrama de componentes que represente cómo los componentes principales del sistema están organizados y cómo interactúan.

**Diagrama de componentes:**Diagrama

Descripción generada automáticamente

**7. Comentarios adicionales**

Incluye cualquier comentario adicional o aclaración relevante sobre la vista lógica de tu sistema.

**Comentarios:**

Este Informe se basa en la información de la Vista Lógica realizada para el sistema de gestión de Personas Madu.

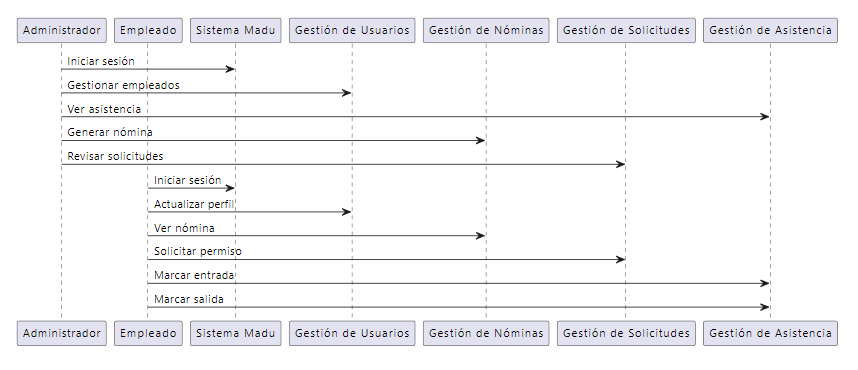
Se adjuntan otros diagramas de la vista lógica:

**Diagrama de Secuencia**

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

**Diagrama de Comunicación**

****