

Objetivo: un sistema de gestion de tramites por un gobierno local.

Requerimientos Funcionales:

- Crear cuenta personal (DNI, fecha de nacimiento, nombre, apellidos, email, telefono, direccion, contraseña, 2fa)
- Crear solicitud/permisos (buscar/ver los permisos, pedir permiso, rellenar informaciones)
- Presentar documentos (rellenar documento, subir archivo)
- Realizar pagos (pago por tarjeta, o cuenta bancaria)
- Recibir notificaciones de estado (recibir email cuando el estado de una solicitud cambia)
- Gestionar/programar citas (ver las disponibilidad, elegir, recibir confirmacion, cancelar, cambiar)
- Acceder base de datos preguntas frecuentes
- Recibir asistencia con chatbot (escribir a un chatbot et obtener una respuesta)
- Buscar, revisar y aprobar las solicitudes, anadir un mensaje (elegir el estado, marcar lista de problema, cambiar el estatus)
- Generar de reportes estadisticos de forma periodica y permitir descargarlos en PDF y Excel.

Requerimientos no funcionales:

- Usabilidad: intuitiva, de todos dispositivos (movil, tabletas, computadoras)
- Accesibilidad: en conformidad con las normas, como es una administracion tiene obligaciones
- Seguridad: proteccion de datos personales, Cifrado de datos sensibles, Autenticacion multifactor por todos
- Disponibilidad: 99.8%,
- Rendimiento: tiempos de respuesta rapidos, incluso en los picos
- Escalabilidad: Tiene que tener la capacidad de manejar incrementos en el número de usuarios y productos sin afectar el rendimiento + capacidad pico de demanda

Recomendaciones/Ambigüedades:

- Los empleados pueden crear cuentas por otros empleados.
- Las solicitudes/tasas y otros elementos no pueden ser eliminados una vez creados, pero su estado puede cambiar.
- FAQ: los empleados pueden añadir/eliminar preguntas, y de vez en cuando se puede publicar la FAQ actualizada. Las preguntas no tienen dueños
- El chatbot tiene al max una sola misma conversacion con un usuario.
- Documento se puede crear o subir
- Contratar servicio externo por los servidores, los pagos, y los chatbot.

Clases / Entidades:

Clase: Persona: Atributos: +dni: String +nombre: String +apellidos: String +id: Int +correo: String +telefono: Int -contraseña: String -2fa: String +fechaNacimiento: Date +rol: String +fechaCreacion: Date.

Métodos:

```
-iniciarSesion(String dni, String contraseñas, String 2fa): Boolean  
-cerrarSesion(): void.  
+get y set para todos los atributos,
```

Subclase: Usuario hereda de Persona.:

Atributos:

```
#+solicitudes: List<Solicitud>
+citasProgramadas: List<Cita>
+tasas: List<Tasa>
+notificacionesRecibidas: List<Notififacion>
+conversacionesChatbot: List<Conversaciones>
```

Métodos:

```
crearCuentaUsuario(String dni, String correo,String contrasenas, String 2fa
+pedirCita()
+pagarTasa():Boolean
+hacerSolicitud()
```

Subclase: Empleado hereda de Persona. :

Atributos:

```
#+solicitudesAttribuidas: List<Solicitud>,
```

Métodos:

```
crearCuentaEmpleado(String dni, String correo,String contrasenas, String 2fa
+verSolicitudes:Boolean
+buscarSolicitudes:Boolean
+rechazarSolicitud:Boolean
+aprobarSolicitud:Boolean
```

Clase: Solicitud:

Atributos:

```
-id: Int
-idUsuario: int
-idEmpleado: Empleado
-tipoSolicitud: Int
-documentos: List<Documento>
-estado: Int
-categoria: Int
-tasaAsociadas: list<Tasa>
```

Métodos:

```
-crearSolicitud():Boolean
-modificarSolicitud():Boolean
```

Clase: Seguridad

Atributos:

Métodos:

```
+cifrarDatos(): Boolean;
+verificarDatos(): Boolean;
```

Clase: Autenticación

Atributos:

```
-id: Int
-idUsuario: Int
-contrasenaUsuario: String
-rolUsuario: String
```

Métodos:

```
+validarCredenciales(): Boolean
+cambiarContrasena(String contrasenaNueva): Boolean
+recuperarContrasena(String mailUsuario, int numeroIntentosEntrarContrasena)
```

Clase: Documento

Atributos:

- +id: Int,
- +fechaCreacion: Date,
- +author: Int
- +contenido: File

Métodos:

- +crearDocumento(): Boolean
- +subirDocumento(): Boolean

Clase: Cita:

Atributos:

- id: Int
- fecha: Date
- codigoTipo: Int
- codigoLugar: Int

Métodos:

- crearCita()
- cancelarCita(): Boolean

Clase: Tasa:

Atributos:

- id: Int
- fecha: Date
- estado: Int
- referenciaPago: String
- notification: Notificación

Métodos:

- crearTasa()
- pagarTasa()
- notificarTasa(): Boolean

Clase: Notificación:

Atributos:

- id: Int
- fecha: Date
- texto: String

Métodos:

- +enviarNotificacion(Int idUsuario, Date fecha, Texto): Boolean

A CAMBIAR EN UNA

Clase: FAQ:

Atributos:

- faq: List<PreguntaRespuestaPair>
- PreguntaRespuestaPair: String[]

Métodos:

- createPreguntaRespuesta()
- editPreguntaRespuesta()
- removePreguntaRespuesta()

Clase: Conversacion

Atributos:

id: Int
fecha: Date

Métodos:

crearConversacion(String)
updateConversacion(String)

Clase: Chatbot

Atributos: conversaciones:
List<Conversacion>

Métodos:

+addConversacion(): Boolean
+updateConversacion(): Boolean

Clase: Cita

Atributos:
-idCita: Int
-idUserio: Usuario
-fecha: Date
-tipoCita: Int
-tipoLugar: Int
Tipos:
+crearCita(): Boolean # Con
+verificarDisponibilidad(): Boolean
+cancelarCita(): Boolean
+generarNotificacion(): Boolean

Clase: Planning:

Atributos:

Citas: list<Cita>

Métodos:

+anadirCita(Cita planning): Boolean
+eliminarCita(Cita planning): Boolean
+mostrarPlanning(): Boolean

Clase: Registro:

Atributos:

Solicitudes: list<Solicitud>

Métodos:

+anadirSolicitud()
+buscarSolicitud()
+verSolicitudes()
+publicarReport: Boolean

Clase: Contabilidad

Atributos:

Tasas: list<Tasa>

Métodos:

+anadirTasa()
+publicarReport()

Clase: Personas

Atributos:

-personas:List<Persona>

Métodos:

+anadirPersona(Persona personas)

+elimidarPersona(Persona personas)

+publicarReport()

Relaciones (Cardinalidad):

Usuario - Solicitud: 0.._ Usuario - Cita: 0.._ Usuario - Tasa: 0.._ Usuario - Notificación: 0.._ Empleado - Solicitudes: 0.._ Solicitud - Tasa: 0.._ Solicitud - Documento: 0.._ Registro - Solicitudes 1.._ Tasa - Pago: 0..1 Tasa - Notificación 0.._ Empleado - PreguntaRespuesta 10.._ FAQ - PreguntaRespuesta 0.._ Personas - Persona 1.._ Planning - Cita 1.._ Contabilidad - Tasa 1.._

Tipos de relaciones:

Persona - Usuario: herencia

Persona - Empleado: herencia

Planning - Cita : Composición.

Registro - Solicitud: Composición.

Contabilidad - Tasa: Composición.

FAQ - Pregunta : Composición.

Reporte - Planning,Registro,Contabilidad : Composición

El resto Asociación.