### Nombre: Gallegos Trujillo Christian Javier.

**Carrera:** Sistemas de información

**CASO PRACTICO**

## Product Backlog Priorizado

Negocio va orientado a la gestión de casos de los abogados en cuanto el prototipo web es una herramienta que apoya como valor agregado puede ser utilizado en otras unidades de abogados.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Inicio** | **Dia** | **Fin** | **Tiempo Estimado** | **Tiempo Real** | **Fecha realizar** | **Actividad** |
| 20/03/2024 | Jueves | 20/03/2024 | 2 | 3 | 20/03/2024 | Gestión de caso |
| 20/03/2024 | Miércoles | 20/03/2024 | 2 | 3 | 20/03/2024 | Gestión Seguimiento Plazo |
| 20/03/2024 | Miércoles | 20/03/2024 | 2 | 3 | 20/03/2024 | Gestión de tareas |

|  |  |  |  |
| --- | --- | --- | --- |
| **EPICA** | **Historia de usuario** | **Spring** | **Estado** |
| Gestión de caso | En el prototipo se puede generar los casos de los abogados | 1 | completado |
| Asignación de tareas | El sistema se puede generar la asignación de uso caso y las tareas | 2 | Completado |
| Seguimiento a plazos | El sistema puede revisar la fechas de audiencia | 3 | Curso(Mitad) |

**Tabla 1.**

Historias de Registro de caso

|  |  |
| --- | --- |
| HISTORIA DE USUARIO | |
| **Número:** 1 | **Nombre:** Registro de caso |
| **Usuario:** abogado | **Riesgo en Desarrollo:** Alta |
| **Prioridad en negocio:** Alta | **Iteración asignada:** 1 |
| **Descripción:**   * Se deben ingresar la información esencial de los casos llevados por los abogados. * Se detalles relevantes es importante para la revisión de los casos. | |
| **Observación:** Asignar un caso único para la identificación | |

Nota: La tabla realizada por elaboración propia (2024).

**Tabla 2.**

Historias de Usuario Seguimiento de plazo y calendario.

|  |  |
| --- | --- |
| HISTORIA DE USUARIO | |
| **Número:** 2 | **Nombre:** seguimiento de plazo |
| **Usuario:** abogado. | **Riesgo en Desarrollo:** media |
| **Prioridad en negocio:** media | **Iteración asignada:** 3 |
| **Descripción:**   * Se debe establecer las fechas de audiencia para no pase por alto plazo crítico. * Permite un control de las actividades dentro del calendario * Evitar que los casos no pasen a plazos criticos | |
| **Observación:** recordatorio y notificar | |

Nota: La tabla realizada por elaboración propia (2024).

**Tabla 3.**

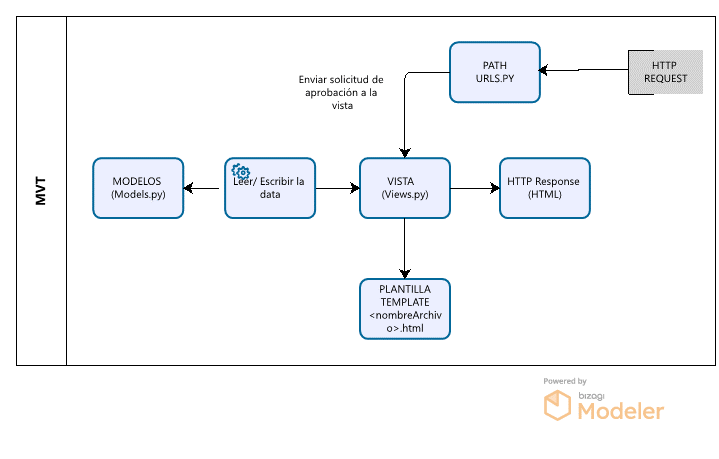
Historias de Usuario Gestión de Vinculación

|  |  |
| --- | --- |
| HISTORIA DE USUARIO | |
| **Número:** 3 | **Nombre:** Asignación de tareas |
| **Usuario:** abogado | **Riesgo en Desarrollo:** Media |
| **Prioridad en negocio:** media | **Iteración asignada:** 2 |
| **Descripción:**   * Los miembros del equipo pueden asignar las tareas * Estas tareas asignadas deben ser específicas para los miembros del equipo * Se registrar las evidencias de las tareas registradas. | |
| **Observación:** | |

Nota: La tabla realizada por elaboración propia (2024).

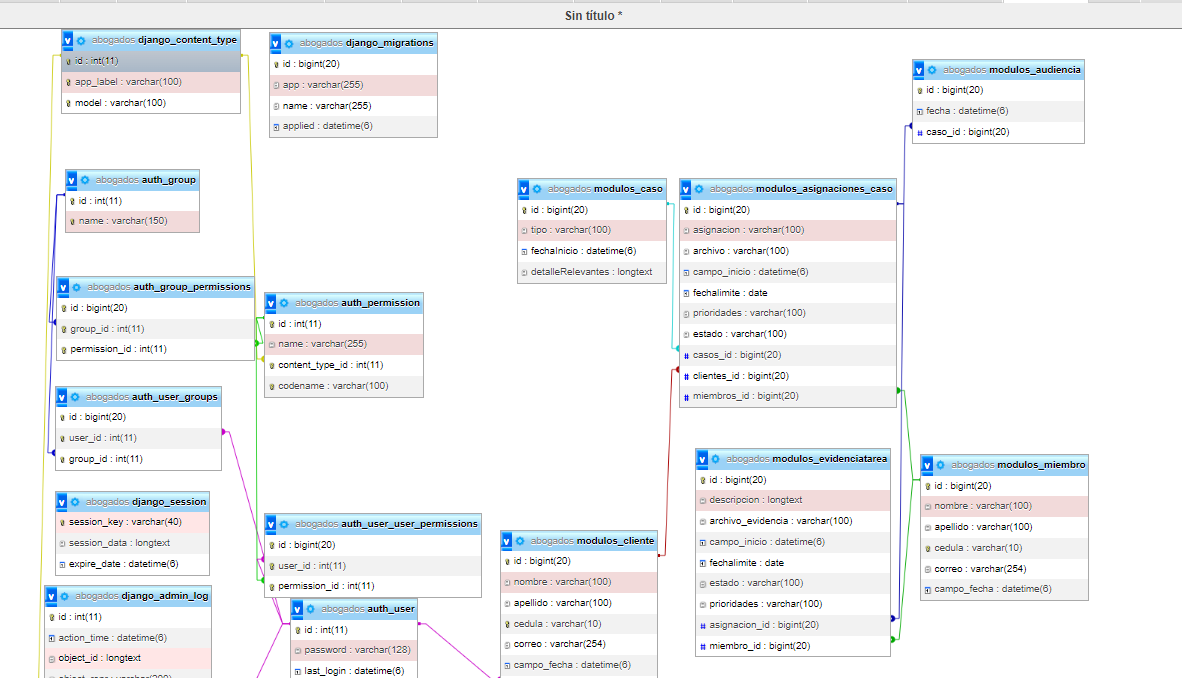
**Arquitectura MVT:**

**Model Vista Template (MVT)**



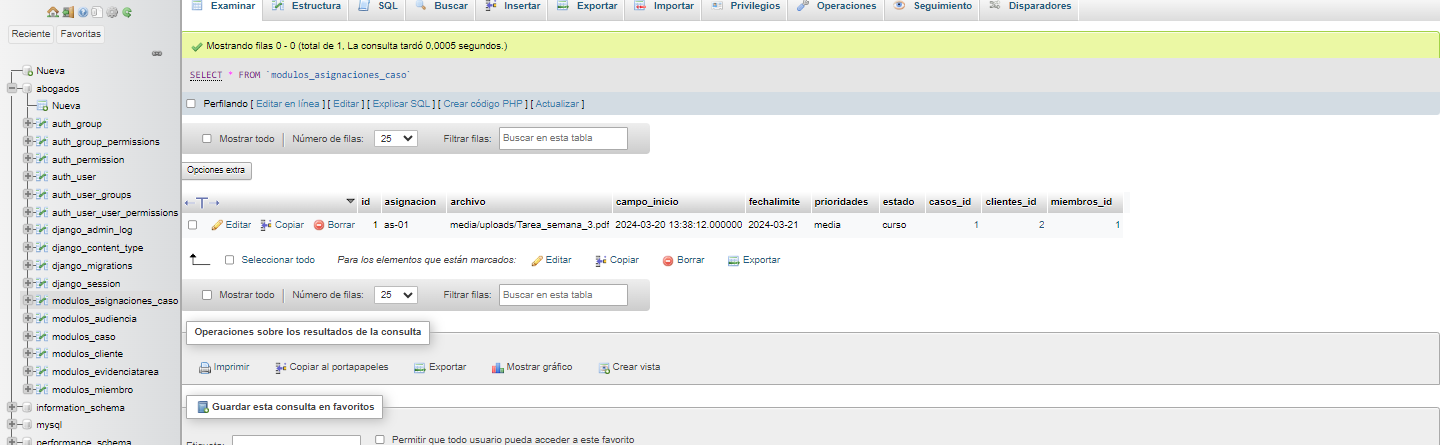
Se lección la arquitectura modelo, vista, template se en diagramas de modelos relacionales y nos permite conectar cada uno de los componentes siguiendo el estadar de los patrones de diseño.

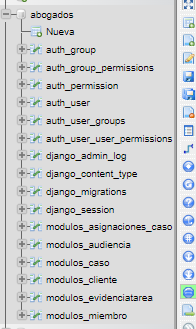
Framework a usar para esta arquitectura Django 4.2.2 haciendo uso del lenguaje de programación Python 3.11 ya que nos permite hacer uso de la arquitectura y ocupar el orm para crear el modelo hacia la base de datos

**Modelo Físico de la base de datos:**Modelos: Caso,Cliente,Miembro y asignación de tarea

La asignación de tarea va ser la tabla intermedia que va a conectar con los demás modelos.

Las tablas cliente y miembro va a interactuar con la asignación de tareas.

La base de datos se llama: sistema de gestión



En Python haciendo uso de la dependencia mysqlclient

Pip install makemigrations

Pip install migrate

**Lenguaje de Programación**

**Asignacion de caso**

from django.db import models

from django.utils import timezone

from .cliente import Cliente

from .miembro import Miembro

from .caso import Caso

import datetime

from django.core.exceptions import ValidationError

class Asignaciones\_caso(models.Model):

    asignacion= models.CharField(max\_length=100 , blank=True, null= True)

    clientes = models.ForeignKey(Cliente, on\_delete=models.CASCADE, related\_name='tabla\_intermedia\_cliente',blank=True, null= True)

    miembros = models.ForeignKey(Miembro, on\_delete=models.CASCADE, related\_name='tabla\_intermedia\_miebro',blank=True, null= True)

    casos = models.ForeignKey(Caso, on\_delete=models.CASCADE, related\_name='tabla\_intermedia\_caso',blank=True, null= True)

    archivo = models.FileField(upload\_to="media/uploads", blank=True, null=True)

    campo\_inicio= models.DateTimeField(default=timezone.now)

    fechalimite = models.DateField(blank=True,null=True)

    prioridades = models.CharField(max\_length=100 ,blank=True, null= True)

    estado = models.CharField(max\_length=100, blank=True, null= True)

    def \_\_str\_\_(self):

        return f"{self.casos} - {self.miembros} - {self.clientes}"

    def clean(self):

        # Convertir la fecha de inicio a un objeto datetime.date

        fecha\_inicio = self.campo\_inicio.date()  # Convertir a datetime.date

        # Validar que la fecha de inicio no sea mayor que la fecha límite

        if self.fechalimite and fecha\_inicio > self.fechalimite:

            raise ValidationError("La fecha límite no puede ser menor que la fecha de inicio.")

**Caso:**

from django.db import models

from django.utils import timezone

class Caso(models.Model):

    tipo = models.CharField(max\_length=100)

    fechaInicio = models.DateTimeField(default=timezone.now)

    detalleRelevantes=models.TextField(blank=True, null= True)

    def \_\_str\_\_(self):

        return self.tipo

**Cliente:**

from django.db import models

from django.utils import timezone

from .caso import Caso

from django.contrib.auth.models import User, Group

from django.db.models.signals import post\_save

from django.dispatch import receiver

class Cliente(models.Model):

    usuario = models.OneToOneField(User, on\_delete=models.CASCADE, null=True, blank=True)

    nombre = models.CharField(max\_length=100,blank=True, null= True)

    apellido = models.CharField(max\_length=100,blank=True, null= True)

    cedula = models.CharField(max\_length=10, unique=True)

    correo = models.EmailField(blank=True, null= True)

    campo\_fecha = models.DateTimeField(default=timezone.now)

    casos = models.ManyToManyField('Caso', through='Asignaciones\_caso', blank=True)

    def \_\_str\_\_(self):

        return f"{self.nombre} {self.apellido}"

@receiver(post\_save, sender=Cliente)

def crear\_usuario\_cliente\_uno(sender, instance, created, \*\*kwargs):

    if created:

        username = instance.cedula  # Utiliza la cédula como nombre de usuario

        password = instance.cedula  # Utiliza la cédula como contraseña inicial

        user = User.objects.create\_user(

            username=username,

            email=instance.correo,

            password=password,

            first\_name=instance.nombre,

            last\_name=instance.apellido,

        )

        # Asigna el usuario al grupo de PERSONA UNO

        group, created = Group.objects.get\_or\_create(name='Cliente')

        user.groups.add(group)

        # Asigna el usuario creado al PERSONA UNO

        instance.usuario = user

        instance.save()

**Evidencia:**

from django.db import models

from .miembro import Miembro

from django.utils import timezone

from .asignaciones\_caso import Asignaciones\_caso

import datetime

from django.core.exceptions import ValidationError

class EvidenciaTarea(models.Model):

    asignacion = models.ForeignKey(Asignaciones\_caso, on\_delete=models.CASCADE, related\_name='evidencias\_tarea',blank=True, null= True)

    descripcion = models.TextField(max\_length=100,blank=True, null= True)

    miembro = models.ForeignKey(Miembro, on\_delete=models.CASCADE)

    archivo\_evidencia = models.FileField(upload\_to="media/evidencias", blank=True, null=True)

    campo\_inicio = models.DateTimeField(default=timezone.now)

    fechalimite = models.DateField(blank=True, null=True)

    estado = models.CharField(max\_length=100,blank=True, null= True)

    prioridades = models.CharField(max\_length=100,blank=True, null= True)

    def \_\_str\_\_(self):

        return f"Evidencia proporcionada por {self.miembro} para la asignación de caso: {self.asignacion}"

    def clean(self):

        # Convertir la fecha de inicio a un objeto datetime.date

        fecha\_inicio = self.campo\_inicio.date()  # Convertir a datetime.date

        # Validar que la fecha de inicio no sea mayor que la fecha límite

        if self.fechalimite and fecha\_inicio > self.fechalimite:

            raise ValidationError("La fecha límite no puede ser menor que la fecha de inicio.")

**Miembro:**

from django.db import models

from django.utils import timezone

from .caso import Caso

from django.contrib.auth.models import User, Group

from django.db.models.signals import post\_save

from django.dispatch import receiver

class Miembro(models.Model):

    nombre = models.CharField(max\_length=100,blank=True, null= True)

    apellido = models.CharField(max\_length=100,blank=True, null= True)

    cedula = models.CharField(max\_length=10, unique=True)

    correo = models.EmailField(blank=True)

    campo\_fecha = models.DateTimeField(default=timezone.now)

    casos = models.ManyToManyField('Caso', through='Asignaciones\_caso', blank=True)

    def \_\_str\_\_(self):

        return f"{self.nombre} {self.apellido}"

@receiver(post\_save, sender=Miembro)

def crear\_usuario\_cliente\_dos(sender, instance, created, \*\*kwargs):

    if created:

        username = instance.cedula  # Utiliza la cédula como nombre de usuario

        password = instance.cedula  # Utiliza la cédula como contraseña inicial

        user = User.objects.create\_user(

            username=username,

            email=instance.correo,

            password=password,

            first\_name=instance.nombre,

            last\_name=instance.apellido,

        )

        # Asigna el usuario al grupo de PERSONA DOS

        group, created = Group.objects.get\_or\_create(name='Miembro')

        user.groups.add(group)

        # Asigna el usuario creado al PERSONA DOS

        instance.usuario = user

        instance.save()

**Seguimiento:**

from .asignaciones\_caso import Asignaciones\_caso

from django.db import models

from django.utils import timezone

from django.contrib import messages

class Audiencia(models.Model):

    caso = models.ForeignKey(Asignaciones\_caso, on\_delete=models.CASCADE)

    fecha = models.DateTimeField()

    def save(self, \*args, \*\*kwargs):

        # Verificar si la audiencia está dentro de un plazo crítico

        if self.plazo\_critico():

            messages.warning("La fecha de la audiencia está dentro de un plazo crítico. Por favor, revísela.")

        super().save(\*args, \*\*kwargs)

    def plazo\_critico(self):

        # Calcular la diferencia de días entre la fecha actual y la fecha de la audiencia

        diferencia\_dias = (self.fecha - timezone.now()).days

        return diferencia\_dias <= 7  # Definimos como plazo crítico las audiencias que tienen 7 días o menos

En cuanto el lenguaje de programación se va a desarrollar es Python 3.12

**Sus dependencias**:

asgiref 3.7.2

chardet 5.2.0

defusedxml 0.7.1

diff-match-patch 20230430

Django 4.2.2

django-import-export 3.2.0

django-smtp-ssl 1.0

et-xmlfile 1.1.0

MarkupPy 1.14

mysqlclient 2.1.1

odfpy 1.4.1

openpyxl 3.1.2

pillow 10.2.0

pip 24.0

PyYAML 6.0

reportlab 4.1.0

setuptools 69.1.0

sqlparse 0.4.4

tablib 3.4.0

tzdata 2023.3

wheel 0.42.0

xlrd 2.0.1

xlwt 1.3.0

**Código fuente:**

Se encuentra dentro de la carpeta

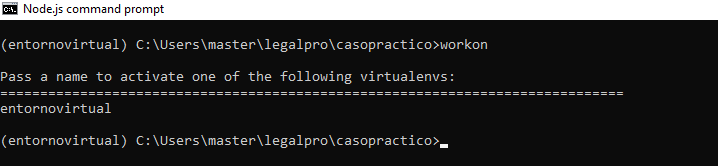
**PASOS PARA INGRESAR A LA APLICACIÓN**

Abrir el node.

****

Luego abrir el entorno virtual

workon entornovirtual

****

Si se va usar desde otra computadora se debe instalar las dependencias

Pip install -r dependencias

La carpeta de las depencias de la carpeta tiene que estar guardado desde la raíz para que funcione.

cd /sglpweb/

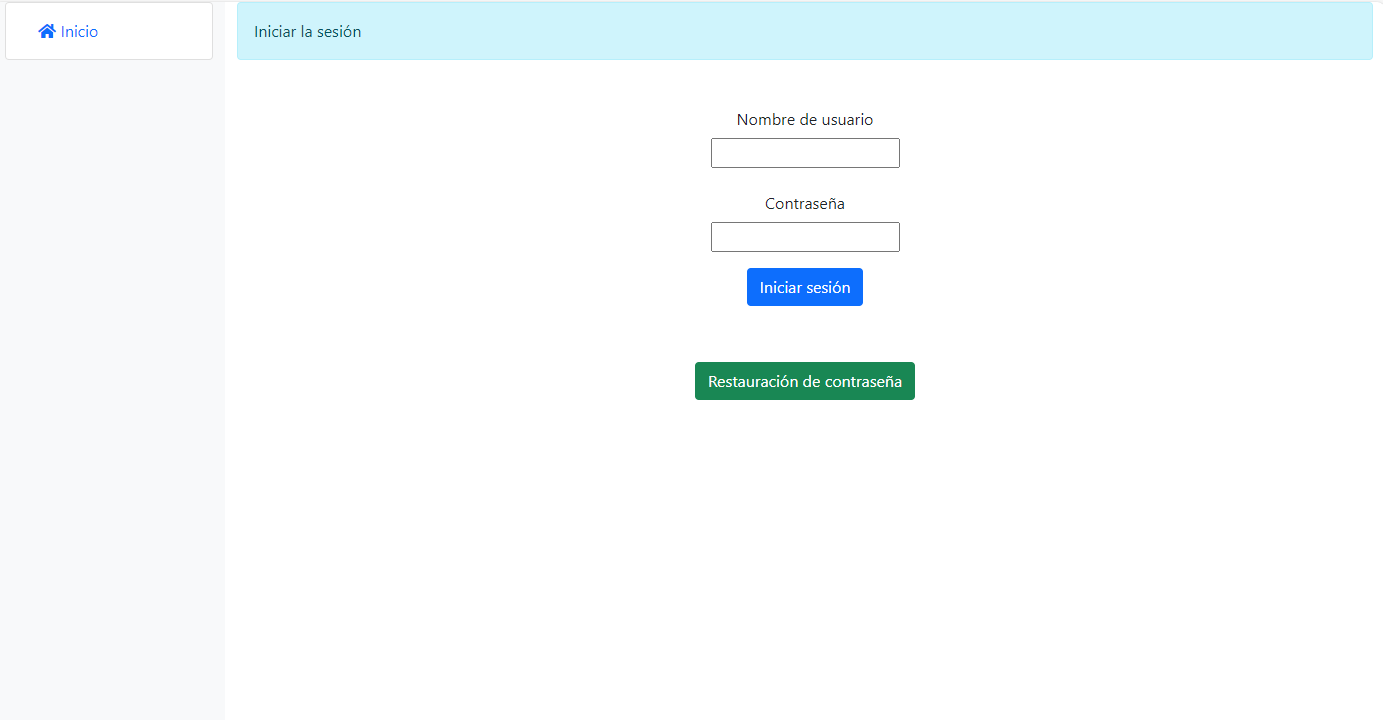
cd/casopractico

python manage.py runserver

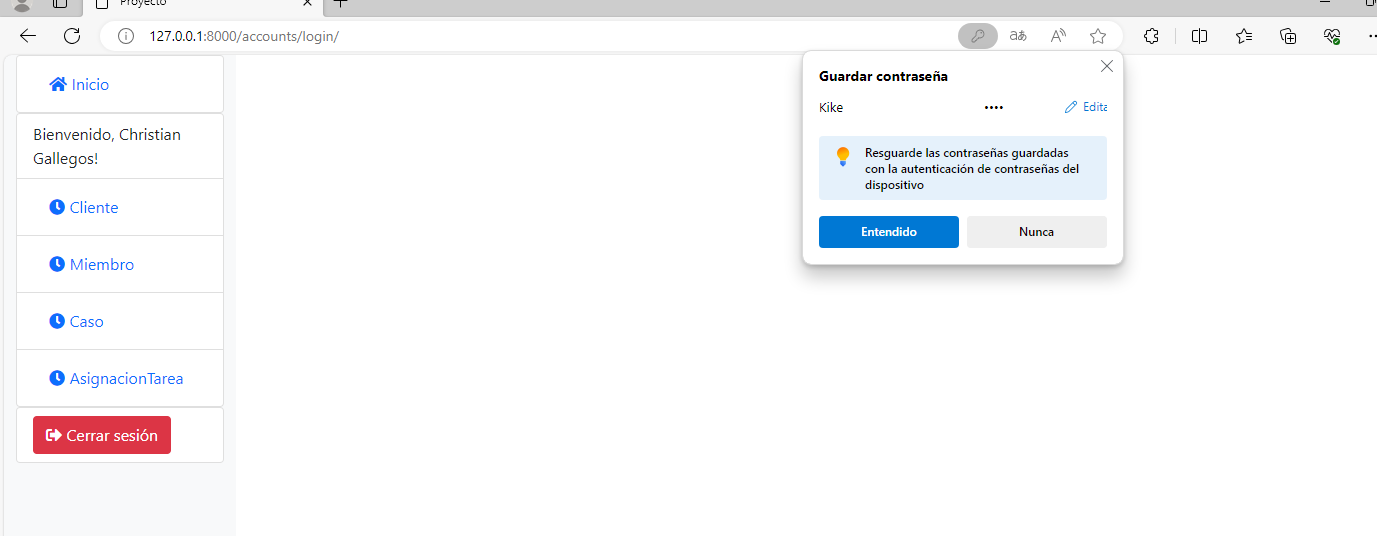
**Login Usuario:**

Usuario: Kike

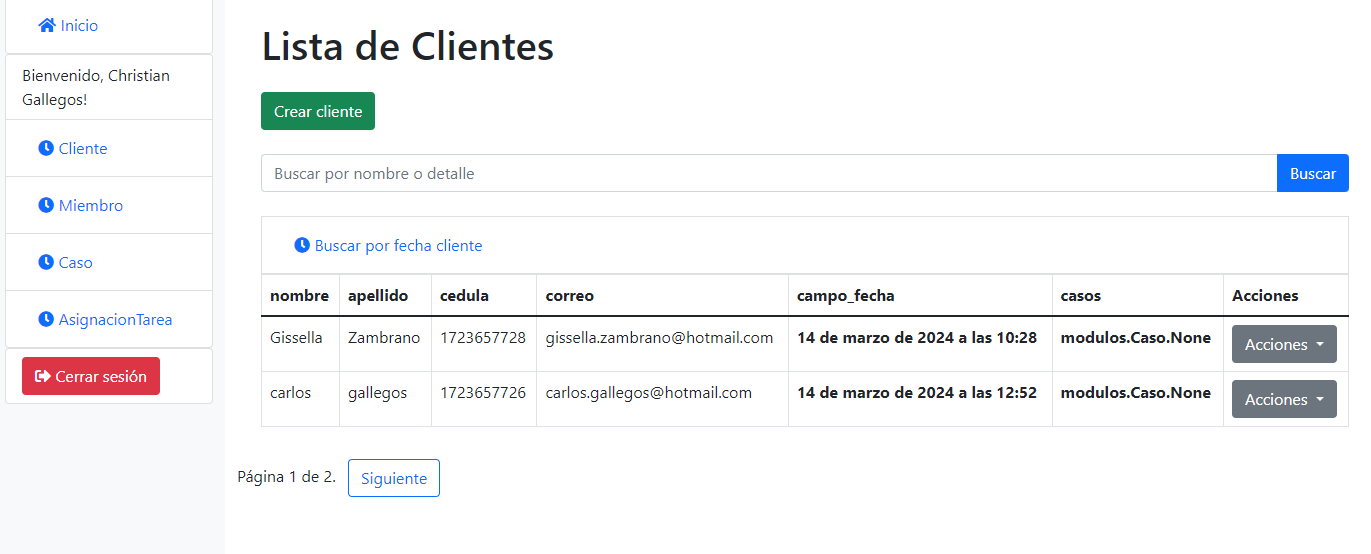
Contraseña:5642

****

**MIEMBRO**

****

**Lista de cliente:**

****

**Lista de casos**Te permite crear un caso asociado al nombre del cliente, tipo de caso, y fecha de inicio. También tiene el id para el buscador

**En acciones puedes ver los casos asignados al cliente.**

****

**Las mismas acciones para los miembros.**

****

**Lista de asignaciones de las tareas**

Los miembros pueden asignar tareas específicas, con fecha limite prioridad y estado.

****

**Agregar una asignación de tarea**

****

**Perfil asignación cliente:**

**Usuario:** 1723657726

**Contraseña:** 1723657726

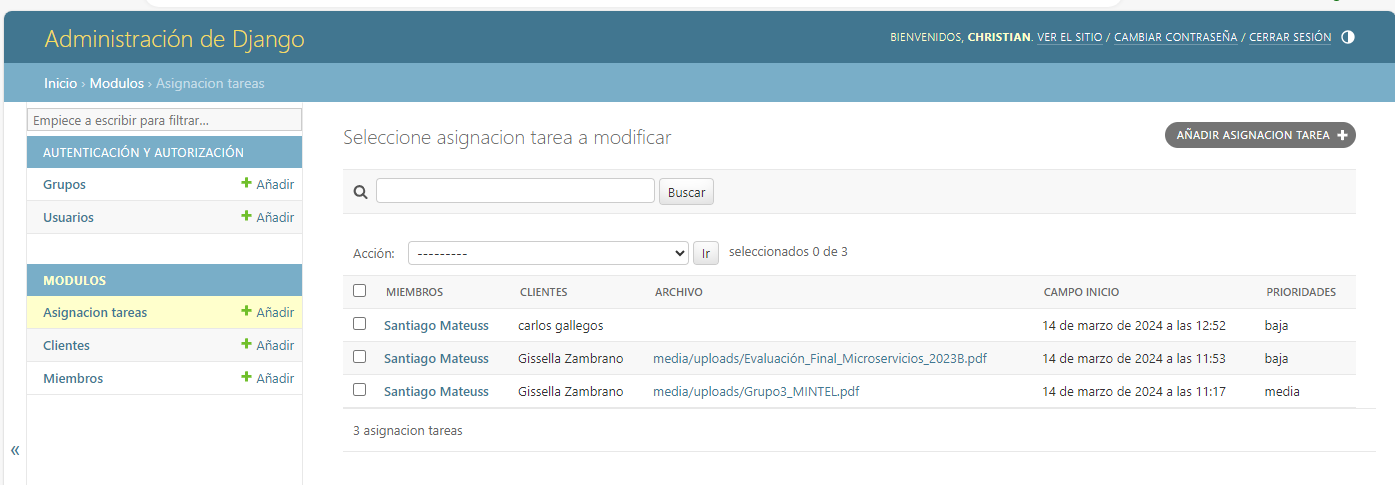
**Aquí se tiene con autentificaciones para que no puedan acceder. A otras pestañas.**

**Solo tiene permisos para visualizaciones.**

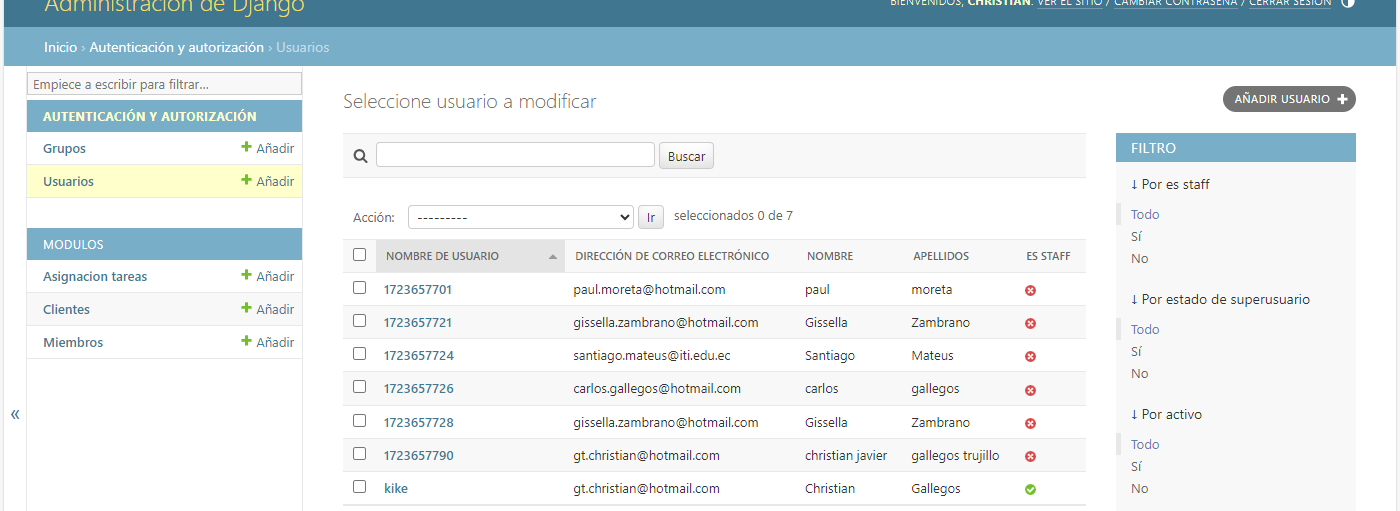
****

**Crear un superuser para usar el admin.**

**Python manage.py createsuperuser**

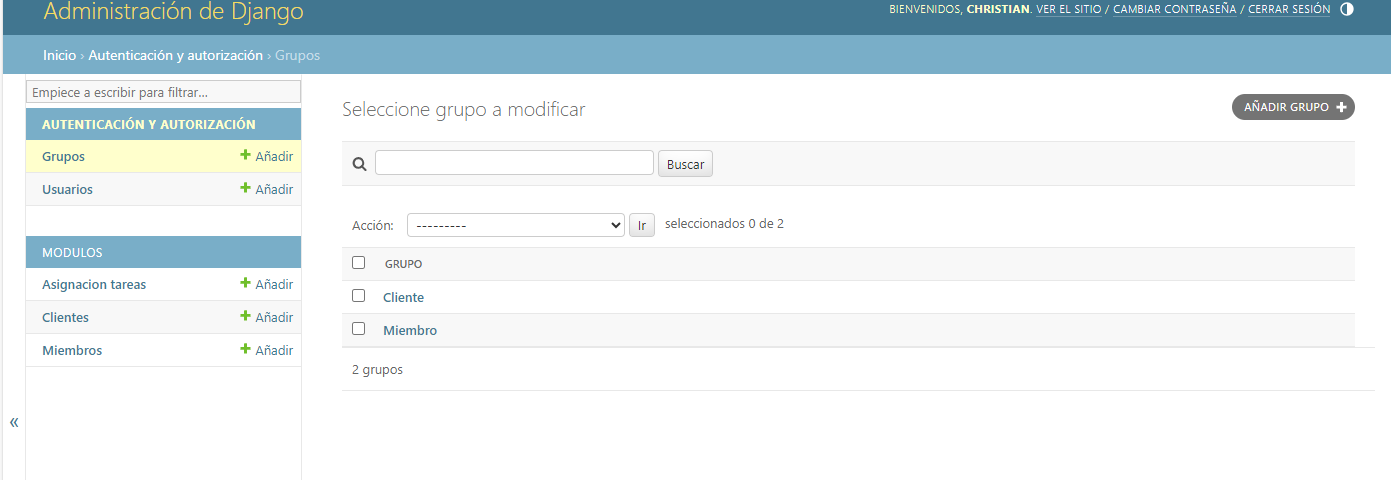
****

**Manejo de usuarios:**

****

**Grupo:**

En grupo cliente definimos que solo tenga permisos de lectura.

****