

**Jhonatan Iván Dávila Muro**

Ingeniero en computacion



**Requisitos**



Jazmines sur #213, Col. Jardines del sol, Zacatecas, Zac.

Edad: 27

Teléfono / celular

:

4921137018

E

-

mail: jhivscode@gmail.com

LinkedIn:

Twitter:



**Administración de usuarios**

Desarrollar en cualquier lenguaje y framework (Preferentemente web).

Integrar pruebas unitarias

de ser posible

.

No importa el orden en que se desarrollen las partes.

**Parte**

**1**

Modelo de

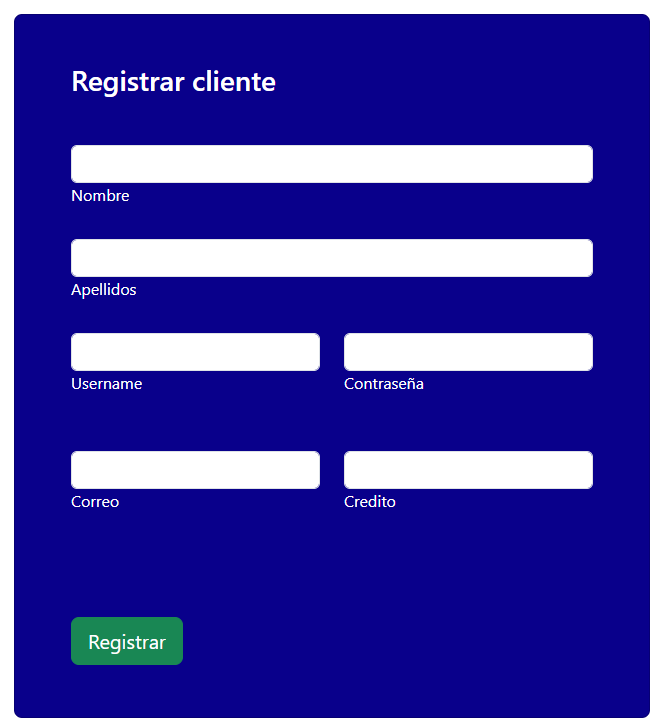
usuario con los siguientes campos: tipo de usuario (Administrador, Distribuidor,

Cliente), créditos totales (int positivo)

, nombre, email, nombre de usuario

.

Se creo la tabla PerfilUsuario que extiende de la tabla usuarios para agregar los campos: tipo usuario, créditos totales y registra. Se agrego el campo registra para saber que usuario registro a los demás usuarios.



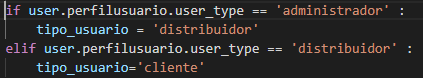


Si es administrador permitir agregar

únicamente distribuidores.

Si es distribuidor agregar únicamente clientes.

Para estos puntos identifique el tipo de usuario que esta logueado desde el lado del servidor para saber qué tipo de usuario se está registrando.



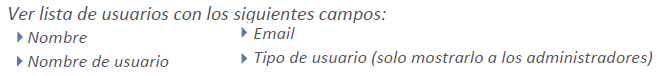


Para este punto se utilizó la librería Re para relizar las operaciones con expresiones regulares necesarias para cumplir con la libreríal.





De la librería “validate\_email\_addres” se uso la función “validate\_email” para verificar que la estructura del correo ingresado sea correcta, se usó esta función para poder usar correos de prueba ya que hay otra función que permite verificar que el correo exista.







Si el usuario que inicio sesión es de tipo administrador se consultan todos los usuarios para mostrar en la tabla.



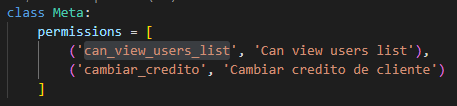


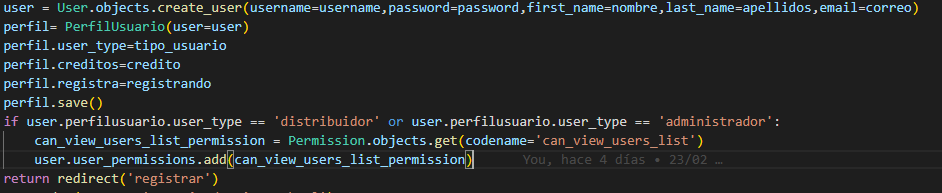
Para esto utilizamos el campo “registra” del perfil usuario para que cuando un administrador vea la tabla de clientes solo vea los clientes que haya registrado distribuidor que este haciendo la solicitud.





Se creo el permiso can\_view\_users\_list para que solo quien tenga asignado el permiso pueda acceder a la lista, por defecto los distribuidores al registrarlos se les agrega. Se agrego un apartado para que cuando un super usuario registre un usuario sea de tipo administrador y también tenga el permiso. Esto se realizó con ayuda de las funciones decoradoras, si no se tiene el permiso se redirige a la página de inicio o home.

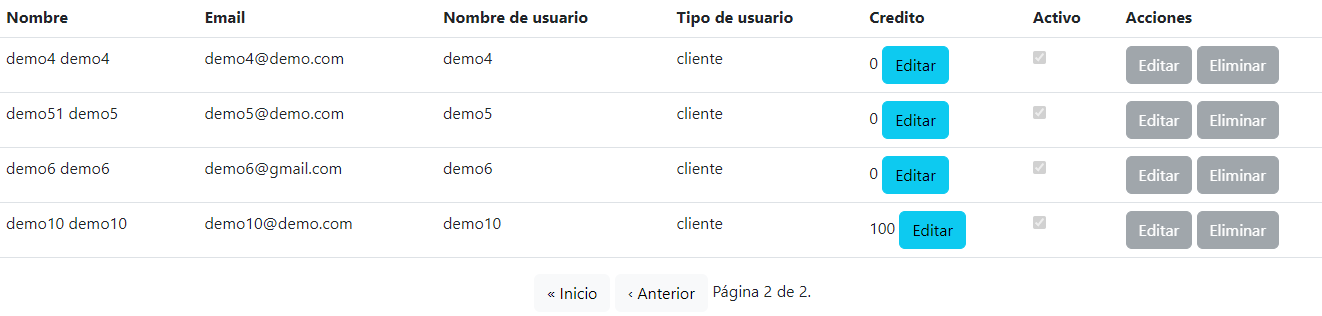






Para implementar el paginado se utilizó la función paginator que incluye la librería paginator del core de django. Se configuro para que mostrara 5 resultados por pagina.

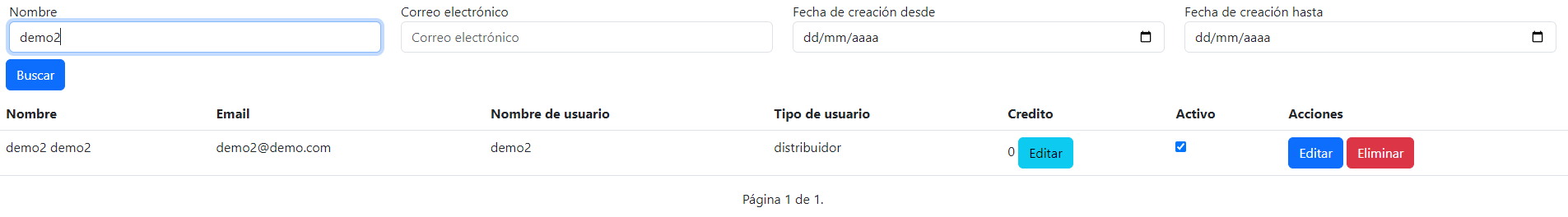


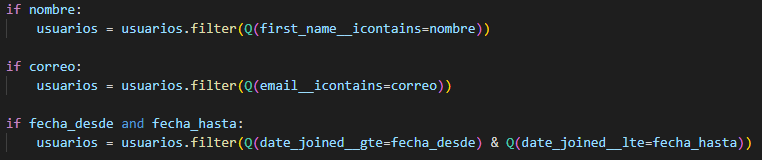






Se uso la función filter que incluye django en sus modelos además de Q para hacer consultas más complejas como agregar “gte” o “lte” en las fechas para buscar a partir de cierta fecha y hasta una fecha específica.









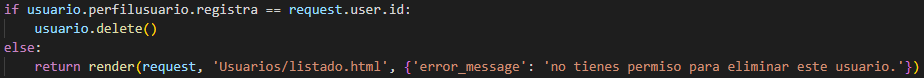
Al administrador solo le aparecen activos los botones de editar y eliminar en los usuarios que el mismo registro. Igualmente, la opción de suspenden un usuario.

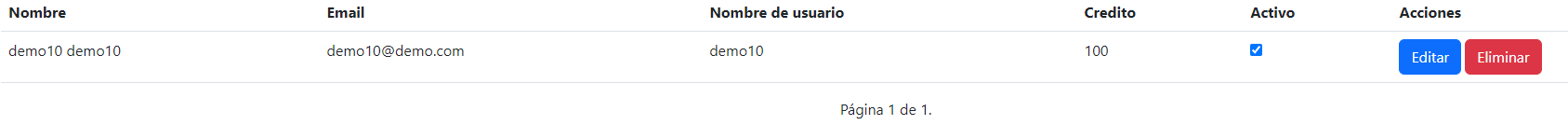
Esto también se comprueba desde el lado del servidor.





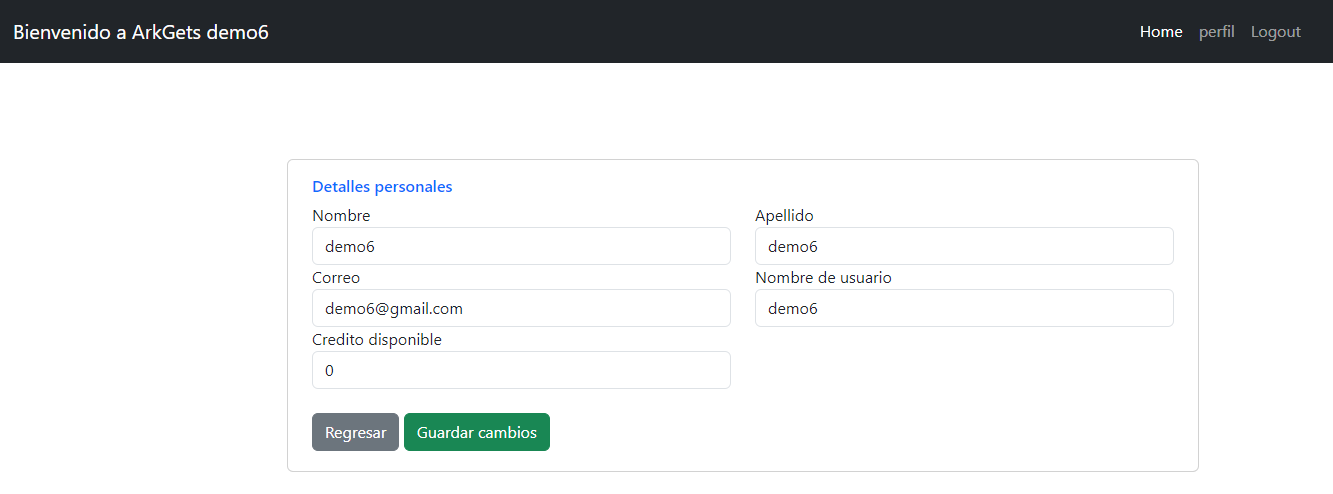
Se realiza la misma funcionalidad de verificar desde el lado del servidor si tiene permiso para editar los clientes solo si el los registro.







En esta vista se muestran los datos del usuario y el campo nombre, apellido y correo son editables.

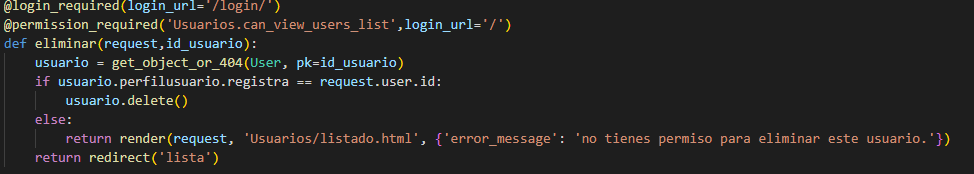






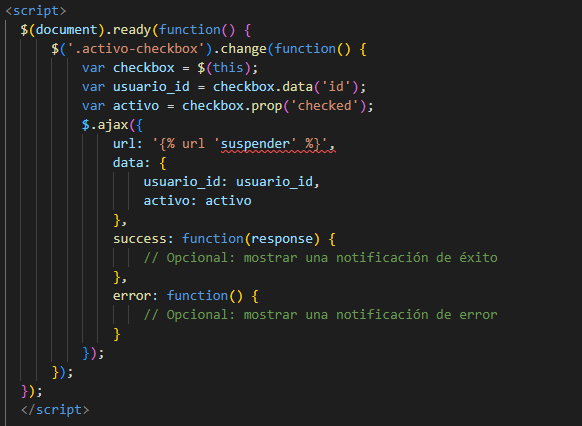
Se agrego el botón eliminar con las mismas condiciones de Editar. También se hace comprobación desde el lado del servidor para ver que tenga permiso de eliminar al usuario seleccionado.





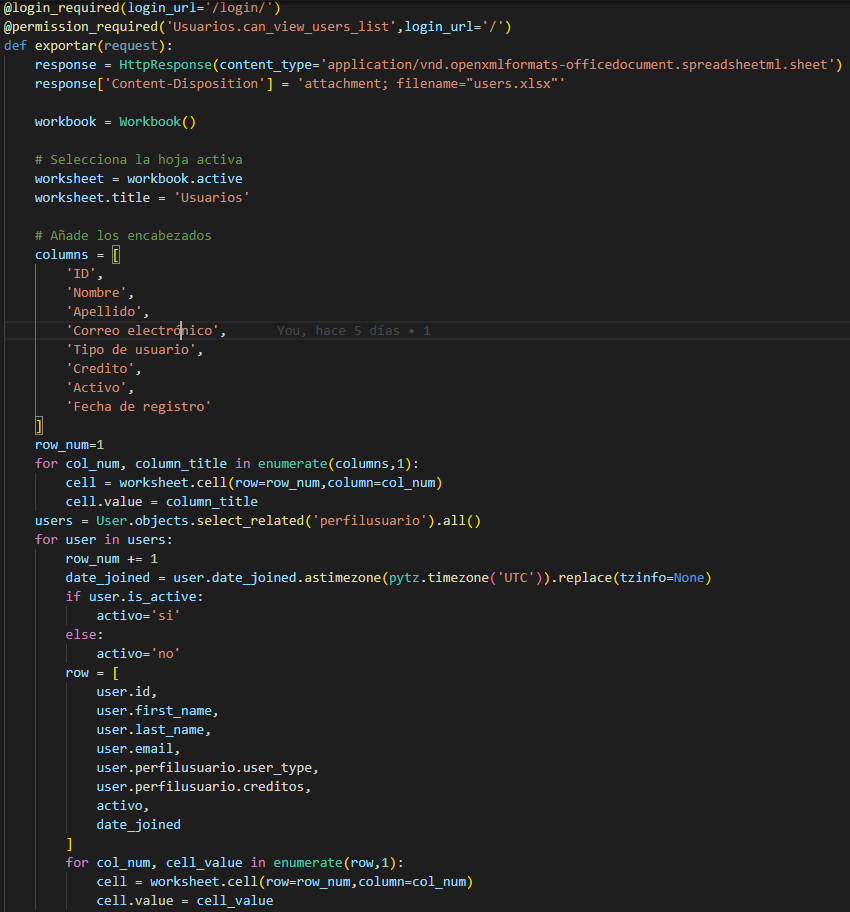


Se agrego la columna “activo” para que quien registre a un usuario pueda ponerlo inactivo, con ayuda de JavaScript la base de datos se actualiza cada que marcan o desmarcan un checkbox

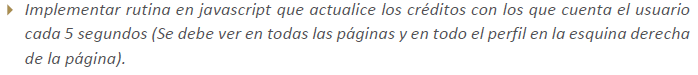




Se agrego un botón para acceder a la función y se genere un archivo en Excel con los usuarios de la base de datos con ayuda de la función workbook.

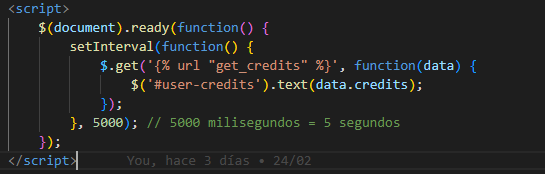


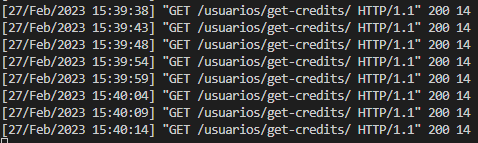




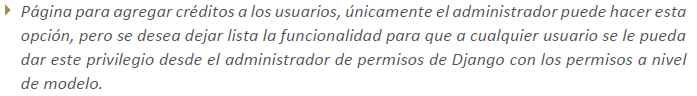
Se creo el script para llamar a una función que obtiene los créditos desde la base de datos cada 5 segundos y pasarle el valor al div donde se muestran los créditos.

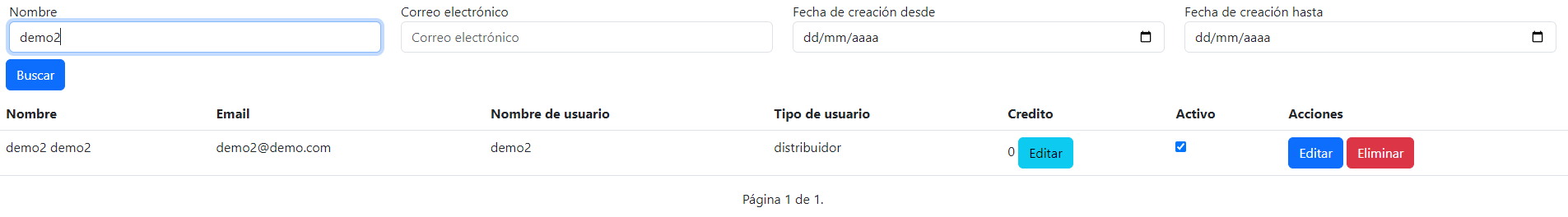






Solicitud al servidor cada 5 segundos.

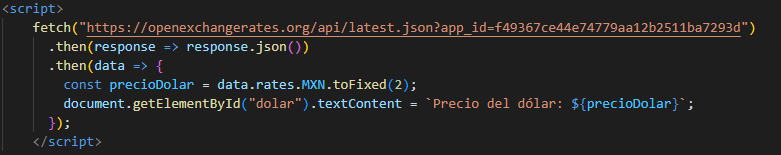


Se agrego una columna que muestra los créditos en la lista de usuarios, y solo quien tiene el permiso “cambiar\_credito” le aparecerá la opción de editar el crédito de el usuario.





Se utilizo un api para obtener el precio del dólar al momento de cargar la pagina con ayuda de javascript.





Se creo una aplicación dentro del projecto a la cual se le puede solicitar la información de los usuarios, como nomsbre y crédito, si se le manda una solicitud GET con cliente\_id responde con los datos específicos de un cliente. También se le puede solicitar el precio del dólar actual.

