

A thick, dark red vertical bar runs along the left edge of the page, starting from the top and extending almost to the bottom, with a small gap near the bottom.

BITÁCORA CORTE 3

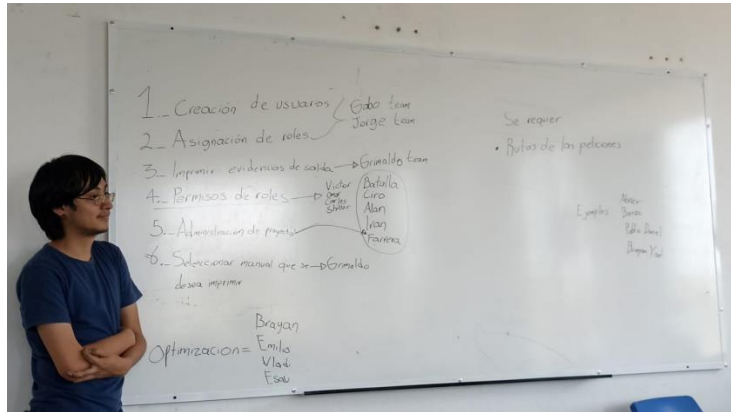
MANTENIMIENTO DE SOFTWARE

MARISOL SOLIS LÓPEZ

203411

INGENIERÍA EN DESARROLLO DE SOFTWARE

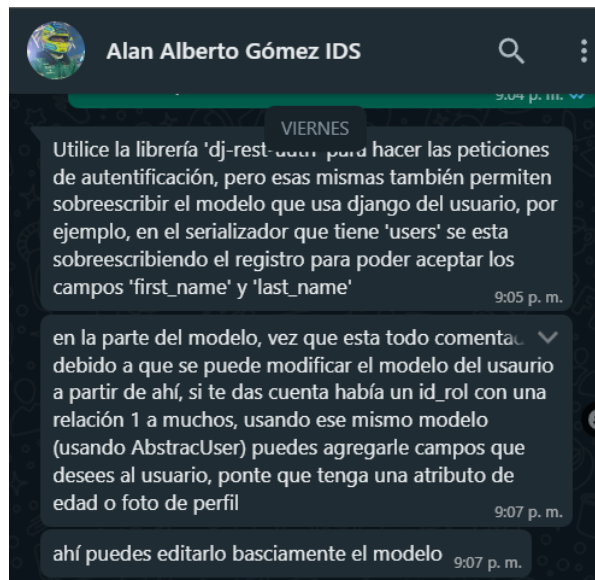
Fecha	Actividad	Descripción	Contratiempos/Comentarios	Evidencia
07/03/2023	Asignación de actividades por equipos	El Scrum Master asignó las actividades correspondientes al sprint 3	Debido a que los equipos no estaban completos, hubo retraso al revisar quienes ya tenían actividades y quienes no	Evidencia 1
08/03/2023	Asignación de actividades individuales	El jefe de equipo asignó actividades de acuerdo a los roles que desempeñamos en el equipo, en mi caso fue desarrollar el backend junto a un compañero		Evidencia 2
10/03/2023	Comienzo del desarrollo backend	Se comenzó por clonar el repositorio con los últimos cambios y creando una nueva rama	Se solicitó ayuda ya que no se comprendía la forma en que se desarrolló el módulo de User	Evidencia 3 Evidencia 4 Evidencia 5 Evidencia 6
11/03/2023	Informes de avance	Se tenían avances en el adapter.py, admin.py y serializers.py	Debido a problemas con la base de datos, hubo retraso en la entrega de avances	Evidencia 7 Evidencia 8 Evidencia 9
12/03/2023	Culminación de backend	Solución a los problemas con la base de datos y creación de métodos put y delete en user		Evidencia 10 Evidencia 11 Evidencia 12 Link1
14/03/2023	EndPoints	Se documentaron en un Word los endpoints desarrollados en el backend		Evidencia 13



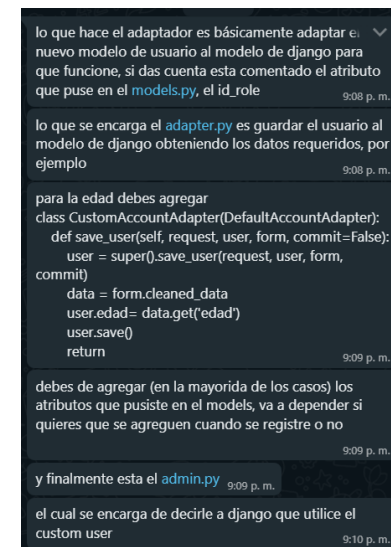
Evidencia 1



Evidencia 2



Evidencia 3



Evidencia 4

el cual se encarga de decirle a django que u
custom user

```
# from django.contrib import admin  
# from .models import CustomUser  
  
# admin.site.register(CustomUser) 9:10 p. m.  
solo debes de descomentarlo 9:10 p. m.
```

Evidencia 5

Cuando hagas todo eso, debes especificarle a la librería
'dj-rest-auth' que ahora debe utilizar tu modelo
personalizado 9:11 p. m.

para ello en el archivo [settings.py](#) debes agregar lo
siguiente 9:11 p. m.

```
# REST_AUTH_REGISTER_SERIALIZERS = {  
#     'REGISTER_SERIALIZER':  
#     'user.serializers.CustomRegisterSerializer'  
# }
```

 9:11 p. m.

eso es si deseas sobre escribir el serializador de registro
9:12 p. m.

y si deseas sobre escribir el modelo de usuario
AUTH_USER_MODEL = 'user.CustomUser' 9:12 p. m.

Evidencia 6

```

solop@solop:~/Documents/avo Cuatrimestre/Mantenimiento de Software/proyecto/qualitor_backend (users)
$ python manage.py makemigrations role
Traceback (most recent call last):
  File "C:\Users\solop\Documents\avo Cuatrimestre\Mantenimiento de Software\proyecto\qualitor_backend\manage.py", line 22, in <module>
    main()
  File "C:\Users\solop\Documents\avo Cuatrimestre\Mantenimiento de Software\proyecto\qualitor_backend\manage.py", line 18, in main
    execute_from_command_line(sys.argv)
  File "C:\Users\solop\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9.0bz52kfra8p0\LocalCache\local-packages\Python39\site-packages\django\core\management\__init__.py", line 446, in execute_from_command_line
    utility.execute()
  File "C:\Users\solop\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9.0bz52kfra8p0\LocalCache\local-packages\Python39\site-packages\django\core\management\__init__.py", line 440, in execute
    self.fetch_command(subcommand).run_from_argv(self.argv)
  File "C:\Users\solop\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9.0bz52kfra8p0\LocalCache\local-packages\Python39\site-packages\django\core\management\base.py", line 480, in run_from_argv
    self.execute(*args, **cmd_options)
  File "C:\Users\solop\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9.0bz52kfra8p0\LocalCache\local-packages\Python39\site-packages\django\core\management\base.py", line 488, in execute
    output = self.handle(*args, **options)
  File "C:\Users\solop\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9.0bz52kfra8p0\LocalCache\local-packages\Python39\site-packages\django\core\management\commands\makemigrations.py", line 141, in handle
    loader.check_consistent_history(connection)
  File "C:\Users\solop\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9.0bz52kfra8p0\LocalCache\local-packages\Python39\site-packages\django\db\migrations\loader.py", line 327, in check_consistent_history
    raise InconsistentMigrationHistory(
django.db.migrations.exceptions.InconsistentMigrationHistory: Migration account.0001_initial is applied before its dependency user.0001_initial on database 'default'.

```

```

$ python manage.py migrate
Traceback (most recent call last):
  File "C:\Users\solop\Documents\avo Cuatrimestre\Mantenimiento de Software\proyecto\qualitor_backend\manage.py", line 22, in <module>
    main()
  File "C:\Users\solop\Documents\avo Cuatrimestre\Mantenimiento de Software\proyecto\qualitor_backend\manage.py", line 18, in main
    execute_from_command_line(sys.argv)
  File "C:\Users\solop\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9.0bz52kfra8p0\LocalCache\local-packages\Python39\site-packages\django\core\management\__init__.py", line 446, in execute_from_command_line
    utility.execute()
  File "C:\Users\solop\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9.0bz52kfra8p0\LocalCache\local-packages\Python39\site-packages\django\db\migrations\executor.py", line 18, in __init__
    self.loader = MigrationLoader(self.connection)
  File "C:\Users\solop\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9.0bz52kfra8p0\LocalCache\local-packages\Python39\site-packages\django\db\migrations\loader.py", line 56, in __init__
    self.build_graph()
  File "C:\Users\solop\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9.0bz52kfra8p0\LocalCache\local-packages\Python39\site-packages\django\db\migrations\loader.py", line 305, in build_graph
    self.graph.ensure_not_cyclic()
  File "C:\Users\solop\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9.0bz52kfra8p0\LocalCache\local-packages\Python39\site-packages\django\db\migrations\graph.py", line 284, in ensure_not_cyclic
    raise CircularDependencyError(
django.db.migrations.exceptions.CircularDependencyError: role.0003_remove_rolemodel_id_user, user.0001_initial, role.0002_rolemodel_id_user

```

Evidencia 7

```

adapter.py X
qualitor_backend > user > adapter.py
You, 3 days ago | 2 authors (Alan Gomez and others)
1 from allauth.account.adapter import DefaultAccountAdapter
2
3 class CustomAccountAdapter(DefaultAccountAdapter):
4     def save_user(self, request, user, form, commit=False):
5         user = super().save_user(request, user, form, commit)
6         data = form.cleaned_data
7         user.id_role = data.get('id_role')
8         user.id_project = data.get('id_project')
9         user.save()
10        return user

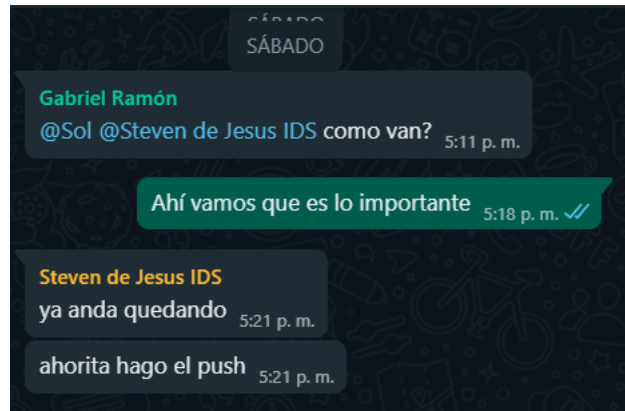
```

```

admin.py X
qualitor_backend > user > admin.py > {} CustomUser
You, 3 days ago | 2 authors (You and others)
1 from django.contrib import admin
2 from .models import CustomUser
3
4 admin.site.register(CustomUser)

```

Evidencia 8



Evidencia 9

```
urlpatterns = [
    path('', include('dj_rest_auth.urls')),
    path('register/', include('dj_rest_auth.registration.urls')),
    path('list/', UserList.as_view()),
    re_path('list/(?P<id_users>\d+)/', UserListDetail.as_view()),
    re_path('delete/(?P<id_users>\d+)/', UserDelete.as_view()),
]
```

Evidencia 10

```
You, 2 days ago [1 author (You)]
class UserListDetail(APIView):
    def custom_response(self, msg, response, status):
        data = {
            "messages": msg,
            "pay_load": response,
            "status": status,
        }
        res = json.dumps(data)
        response = json.loads(res)
        return response

    def get_object(self, id_users):
        try:
            return CustomUser.objects.get(pk = id_users)
        except CustomUser.DoesNotExist:
            return 0

    def get(self, request, id_users, format=None):
        id_response = self.get_object(id_users)
        if id_response != 0:
            id_response = CustomUserDetailsSerializer(id_response)
            return Response(self.custom_response("Success", id_response.data, status=status.HTTP_200_OK))
        return Response(self.custom_response("Error", f"User with id: {id_users} not found", status=status.HTTP_400_BAD_REQUEST))
```

Evidencia 11

```
You, 2 days ago [1 author (You)]
class UserDelete(APIView):
    def custom_response(self, msg, response, status):
        data = {
            "messages": msg,
            "pay_load": response,
            "status": status,
        }
        res = json.dumps(data)
        response = json.loads(res)
        return response

    def get_object(self, id_users):
        try:
            return CustomUser.objects.get(pk = id_users)
        except CustomUser.DoesNotExist:
            return 0

    def delete(self, request, id_users, format=None):
        id_response = self.get_object(id_users)
        if id_response != 0:
            id_response.delete()
            return Response(self.custom_response("Success", f"User with id: {id_users} deleted", status=status.HTTP_200_OK))
        return Response(self.custom_response("Error", f"User with id: {id_users} not found", status=status.HTTP_400_BAD_REQUEST))
```

Evidencia 12



<u>Ruta</u>	<u>Método</u>	<u>Función</u>	<u>Parametros</u>	<u>Retorno</u>
http://127.0.0.1:8000/api/v1/user/	POST	Login	username password	token
http://127.0.0.1:8000/api/v1/user/register/	POST	Register	username password1 password2 first_name last_name email id_role id_project is_superuser	registro de usuario
http://127.0.0.1:8000/api/v1/user/list/	GET	Listar todos los usuarios		Lista de todos los usuarios registrados
http://127.0.0.1:8000/api/v1/user/list/id_user/	GET	Listar a un usuario por id	id en la ruta	Usuario con el id solicitado
http://127.0.0.1:8000/api/v1/user/delete/id_user/	DELETE	Eliminar a un usuario en específico	id en la ruta	Eliminar a un usuario

Evidencia 13

Link 1

https://github.com/203429/qualitor_backend