

## **RECORD RECODATORUM**

Record Recodatorium es una app web hecha en python y Django, usando SQLite3 para la base de datos y almacenamiento de los mismos, es una app donde autentica el usuario y registra los datos almacenados en el servidor, brindando una experiencia de usuario más segura.

Es una lista de cosas de hacer para fomentar el hábito en las personas, puede llegar a ser una extensión de google chrome pero debido a que en este ciclo está prohibido usar Javascript, queda solo como una webapp.

## Índice:

1. Enlaces correspondientes
2. Pantallazos del front con respecto al código
3. conexión con api rest y postman
4. Historias de usuario sprint 4
5. Informe retrospectiva

## 1. Enlaces correspondientes

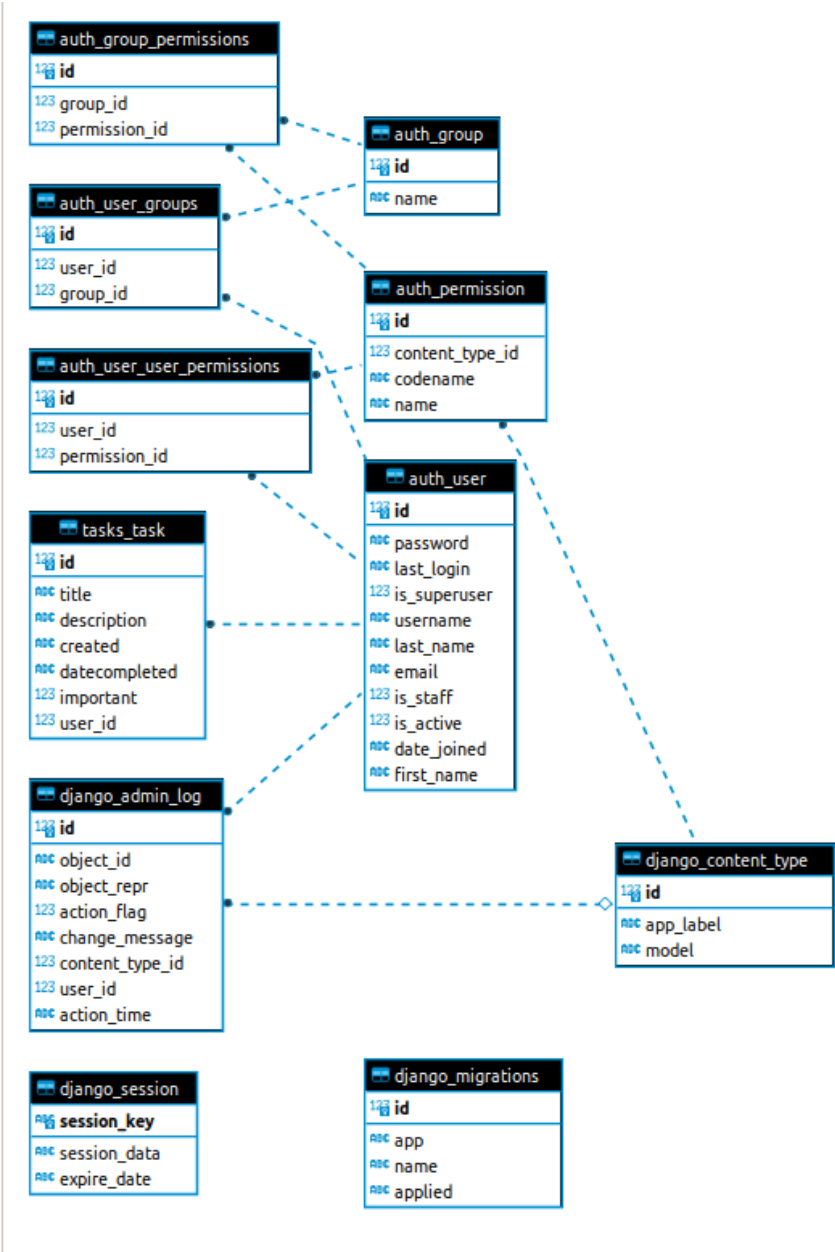
Github:

<https://github.com/herculeskan/Ciclo3>

Integrantes:

CARLOS ESTEBAN LARA GIL	<a href="mailto:Carloسلara034@gmail.com"><u>Carloسلara034@gmail.com</u></a>
-------------------------	---

Diagrama entidad-relación:



## Script de la base de datos

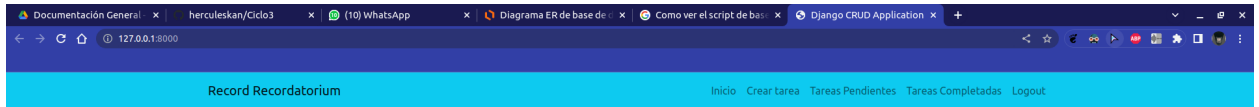
```
1
2 from django.conf import settings
3 from django.db import migrations, models
4 import django.db.models.deletion
5
6
7 class Migration(migrations.Migration):
8
9     initial = True
10
11     dependencies = [
12         migrations.swappable_dependency(settings.AUTH_USER_MODEL),
13     ]
14
15     operations = [
16         migrations.CreateModel(
17             name='Task',
18             fields=[
19                 ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
20                 ('title', models.CharField(max_length=200)),
21                 ('description', models.TextField(blank=True)),
22                 ('created', models.DateTimeField(auto_now_add=True)),
23                 ('datecompleted', models.DateTimeField(null=True)),
24                 ('important', models.BooleanField(default=False)),
25                 ('user', models.ForeignKey(on_delete=django.db.models.deletion.CASCADE, to=settings.AUTH_USER_MODEL)),
26             ],
27         ),
28     ]
```

## VISTAS HTML Y CSS

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7     <title>Django CRUD Application</title>
8     <link
9       rel="stylesheet"
10      href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
11    />
12  </head>
13  <body>
14    <nav class="navbar navbar-expand-lg bg-info">
15      <div class="container">
16        <a class="navbar-brand" href="/">Record Recordatorium</a>
17        <button
18          class="navbar-toggler"
19          type="button"
20          data-bs-toggle="collapse"
21          data-bs-target="#navbarNav"
22          aria-controls="navbarNav"
23          aria-expanded="false"
24          aria-label="Toggle navigation"
25        >
26          <span class="navbar-toggler-icon"></span>
27        </button>
28        <div class="collapse navbar-collapse" id="navbarNav">
29          <ul class="navbar-nav ms-auto">
30            <li class="nav-item">
31              <a class="nav-link" href="/">Inicio</a>
32            </li>
33
34            {% if user.is_authenticated %}
35            <li class="nav-item">
36              <a class="nav-link" href="/create_task">Crear tarea</a>
37            </li>
38            <li class="nav-item">
39              <a class="nav-link" href="{% url 'tasks' %}">Tareas Pendientes</a>
40            </li>
41            <li class="nav-item">
42              <a class="nav-link" href="{% url 'tasks_completed' %}">
43                Tareas Completadas</a>
44            </li>
45
46            <li class="nav-item">
47              <a class="nav-link" href="/logout">Logout</a>
48            </li>
49            {% else %}
50            <li class="nav-item">
51              <a class="nav-link" href="/signup">Sign up</a>
52            </li>
53            <li class="nav-item">
54              <a class="nav-link" href="/signin">Signin</a>
55            </li>
56            {% endif %}
57          </ul>
58        </div>
59      </div>
60    </nav>
61
62    {% block content %} {% endblock %}
63  </body>
64 </html>
65
```

base.html

home



# Record Recordatorium

Record Recordatorium es un app para recordar las cosas que tienes que hacer, registrate gratis e ingresa

[signin](#) [signup](#)

```
1 {% extends 'base.html' %} {% block content %}
2 <main class="container py-5">
3   <section class="card card-body">
4     <h1 class="display-1 text-center">Record Recordatorium</h1>
5     <p>
6       Record Recordatorium es un app para recordar las cosas que tienes que hacer.
7       registrate gratis e ingresa
8     </p>
9     <div class="text-center">
10      <a class="btn btn-primary" href="/signin/">signin</a><a class="btn btn-danger" href="/signup/">signup</a>
11    </div>
12  </section>
13 </main>
14 {% endblock %}
15
```

## Sign in/signin.html

Record Recordatorium

Inicio · Crear tareas · Tareas Pendientes · Tareas Completadas · Logout

### Ingresar

Nombre de usuario:

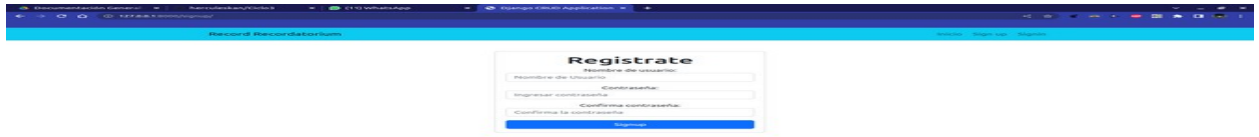
Contraseña:

signin

```
1 {% extends 'base.html' %} {% block content %}
2
3
4 <main class="container">
5     <div class="row">
6         <section class="col-md-4 offset-md-4 mt-5">
7             <form action="/signin/" method="POST" class="card card-body">
8                 {% csrf_token %}
9                 <h1 class="text-center">Ingresar</h1>
10                {{error}}
11                {% csrf_token %}
12                <div class="mb-3">
13                    <label for="username">Nombre de usuario:</label>
14                    <input type="text" name="username" id="username" class="form-control"
15                        placeholder="Nombre de Usuario" />
16                </div>
17                <div class="mb-3">
18                    <label for="password">Contraseña:</label>
19                    <input type="password" name="password" id="password" class="form-control"
20                        placeholder="Ingresar contraseña" />
21                </div>
22                <button class="btn btn-primary">signin</button>
23            </form>
24        </div>
25    </div>
26
27 </form>
28 </section>
29 </div>
30 </main>
31 {% endblock %}
```

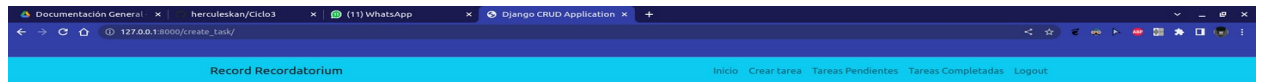


## Sign Up/signup.html



```
1  {% extends 'base.html' %} {% block content %}
2
3  <main class="container">
4    <div class="row">
5      <div class="col-md-4 offset-md-4 mt-5">
6        <form action="/signup/" method="POST" class="card card-body text-center">
7          <h1 class="text-center">Registrate</h1>
8          {{error}}
9          {% csrf_token %}
10         <div class="mb-3">
11           <label for="username">Nombre de usuario:</label>
12           <input
13             type="text"
14             name="username"
15             id="username"
16             class="form-control"
17             placeholder="Nombre de Usuario"
18           />
19         </div>
20         <div class="mb-3">
21           <label for="password1">Contraseña:</label>
22           <input
23             type="password"
24             name="password1"
25             id="password1"
26             class="form-control"
27             placeholder="Ingresa contraseña"
28           />
29         </div>
30         <div class="mb-3">
31           <label for="password2">Confirma contraseña:</label>
32           <input
33             type="password"
34             name="password2"
35             id="password2"
36             class="form-control"
37             placeholder="Confirma la contraseña"
38           />
39         </div>
40         <button class="btn btn-primary">Signup</button>
41       </form>
42     </div>
43   </div>
44 </main>
45 {% endblock %}
46
```

## Crear tarea/create\_task.html



### create task

Title:

Description:

Important: ☐

```
1  {% extends 'base.html' %} {% block content %}
2  <main class="container">
3    <div class="row">
4      <div class="col-md-4 offset-md-4 mt-5">
5        <h1>create task</h1>
6        {{error}}
7        <form action="/create_task/" method="POST" class="card card-body">
8          {% csrf_token %} {{form}}
9
10         <button class="btn btn-primary">Guardar</button>
11       </form>
12     </div>
13   </div>
14 </main>
15
16 {% endblock %}
17
```

## detalle de tarea/task\_detail.html

Record Recordatorium Inicio Crear tarea Tareas Pendientes Tareas Completadas Logout

### aaa

Title:  
aaa

Description:  
assasd

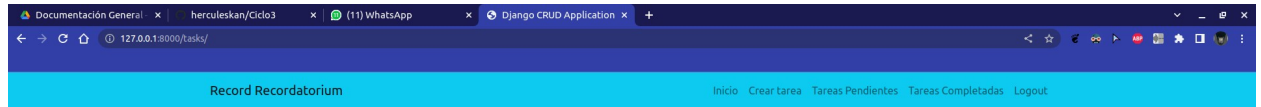
Important: ☐

Update

complete delete

```
1  {% extends 'base.html' %} {% block content %}
2  <main class="container">
3    <div class="row">
4      <div class="col-md-4 offset-md-4">
5        {{error}}
6        <h1 class="text-secondary">{{task.title}}</h1>
7        <form method="POST">
8          {% csrf_token %} {{form.as_p}}
9          <button class="btn btn-primary">Update</button>
10       </form>
11       <div class="mt-5 d-flex justify-content-between">
12
13
14         <form action="{% url 'complete_task' task.id %}" method="POST">
15           {% csrf_token %}
16           <button class="btn btn-secondary">complete</button>
17         </form>
18
19         <form action="{% url 'delete_task' task.id %}" method="POST">
20           {% csrf_token %}
21           <button class="btn btn-danger">delete</button>
22         </form>
23       </div>
24     </div>
25   </div>
26 </main>
27 {% endblock %}
28
```

tareas/tasks.html



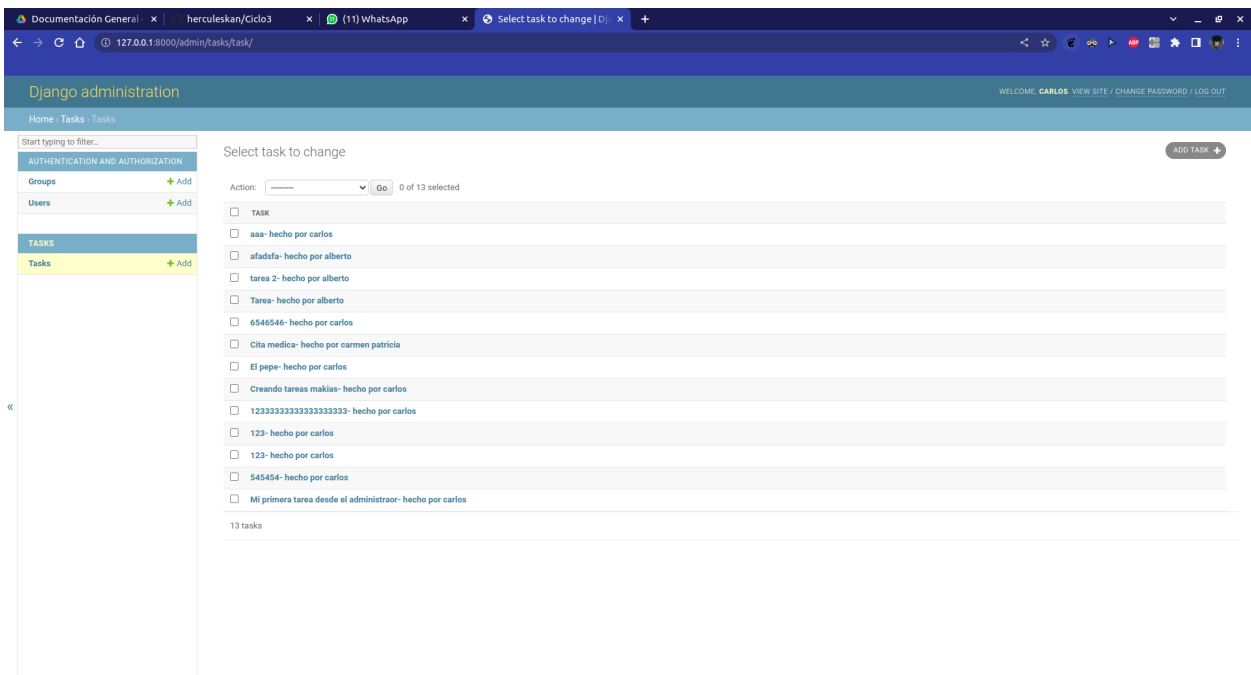
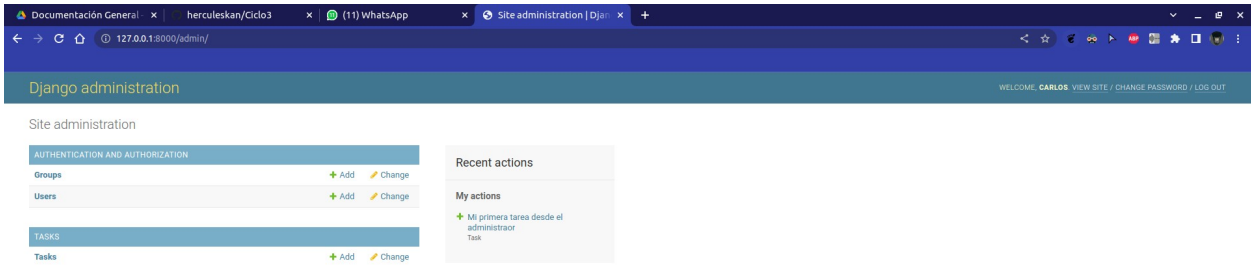
## Tareas Pendientes

**aaa**

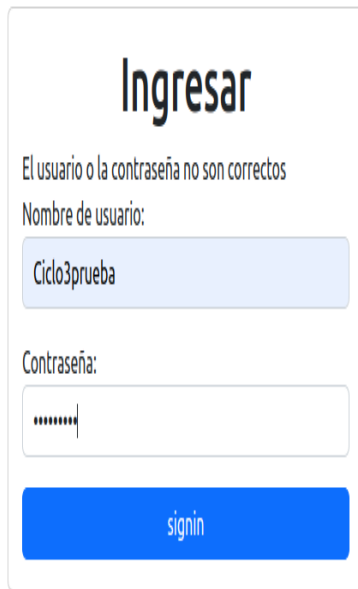
assasd

```
1 {% extends 'base.html' %} {% block content %}
2
3 <main class="container">
4   <div class="row">
5     <div class="col-md-6 offset-md-3">
6       <h1 class="text-center display-3 py-5">Tareas Pendientes</h1>
7
8       <ul class="list-group">
9         {% for task in tasks %}
10          <a class="list-group-item" href="{% url 'task_detail' task.id %}">
11            {% if task.important %}
12              <header class="d-flex justify-content-between">
13                <h1 class="fw-bold text-success">{{task.title}}</h1>
14                <p>{{task.user.username}}</p>
15              {% else %}
16                <h1 class="fw-bold">{{task.title}}</h1>
17              {% endif %}
18            </header>
19            <p>{{task.description}}</p>
20
21            {{task.datecompleted|date:'M j Y:i'}}
22          </a>
23        {% endfor %}
24      </ul>
25    </div>
26  </div>
27 </main>
28
29 {% endblock %}
30
```

### VISTA EN SERVIDOR:



Pruebas unitarias:



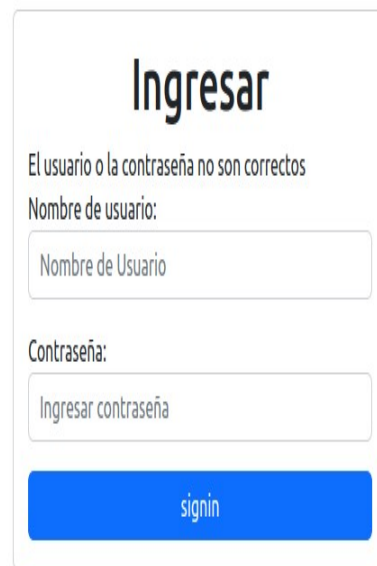
**Ingresar**

El usuario o la contraseña no son correctos

Nombre de usuario:

Contraseña:

signin



**Ingresar**

El usuario o la contraseña no son correctos

Nombre de usuario:

Contraseña:


signin

El usuario no puede entrar si no se ha registrado primero en el sign up.

La protección es tal que si se copia la dirección que se quiere ir, por ejemplo “tasks” te redireccionará al Signin porque necesitas primero ingresar con clave y contraseña para entrar:



Procedemos a registrarnos con el nick “ciclo 3 prueba”:



**Registrate**

Nombre de usuario:

Contraseña:

Confirma contraseña:

Signup

Prestar atención a las barras de navegación.

## Tareas Pendientes

Una vez que el usuario ingresó, la navegación cambia a y se le agrega “crear tareas, tareas pendientes, tareas completadas y el logout

Conexión con postman y API Rest:

Lo primero que se hizo fue instalar el rest\_framework en la carpeta de settings

```
# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'rest_framework',
    'tasks'
```

Se creo un archivo router.py con el siguiente contenido

```
django> router.py > ...
1 from tasks.viewsets import TaskViewSet
2 from rest_framework import routers
3
4 router = routers.DefaultRouter()
5 router.register('tasks', TaskViewSet)
6
7 #url/task
```

se agregó una url para la api

```
path('api/', include(router.urls)),
```

otro archivo serializers.py para que pudiera mostrar lo que hay dentro

```
tasks> serializers.py > TaskSerializer > Meta
1 from rest_framework import serializers
2 from .models import Task
3
4 class TaskSerializer(serializers.ModelSerializer):
5     class Meta:
6         model = Task
7         fields = '__all__'
```



Y para el futuro deploy se hizo el setup con heroku

Overview

Resources

Deploy

Metrics

Activity



Access

Settings

Installed add-ons

\$0.00/month

[Configure Add-ons](#)

 **Heroku Postgres**  Hobby Dev

postgresql-fitted-94284

Se prueba el link en postman

GET

http://127.0.0.1:8000/api/tasks/

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Query Params

KEY	VALUE	DESCRIPTION	***	Bulk Edit
Key	Value	Description		

Body

Cookies

Headers (10)

Test Results

Status: 200 OK

Time: 1724 ms

Size: 330 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

Cookies

Capture requests

Bootcamp

Runner

Trash

Api Root / Task List

## Task List

OPTIONS GET ▾

GET /api/tasks/

```

HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept
[]
  
```

Raw data HTML form

Title

Description

Datecompleted dd/mm/aaaa, --:--

Important ☐

User

POST

En API REST se ve así

## HISTORIAS DE USUARIO

ID: 1 USUARIO: CLIENTE	
NOMBRE DE HISTORIA: INTERFAZ INTUITIVA	
PRIORIDAD: ALTA	Tiempo Estimado: 4 semanas
Responsable: Carlos	
DESCRIPCIÓN: Como cliente, me gustaría una aplicación cuya función sirva para almacenar tareas, que se puedan actualizar y tener una barra de navegación con la cual pueda interactuar con las diferentes partes de la aplicación	
CRITERIO DE VALIDACIÓN: Que el usuario se pueda registrar, y que al no registrarse: no pueda ingresar a la app completa, también que todos los elementos estén bien alineados y responsivos	

ID: 2	USUARIO: ADMINISTRADOR
NOMBRE DE HISTORIA: Backend	
PRIORIDAD: ALTA	Tiempo estimado: 4 semanas
RESPONSABLE: CARLOS El adminstrador debe tener un panel backend en el cuál pueda ver las tareas de todos los usuarios	
CRITERIO DE VALIDACIÓN: Que salga en el http :8000/admin	

ID: 3	USUARIO: TESTER
NOMBRE DE LA HISTORIA: TESTEAR	
PRIORIDAD: MEDIA	TIEMPO ESTIMADO: 4 SEMANAS
Responsable: CARLOS	
El tester tiene que percatarse de que esté funcionando bien la entrada y que sea responsivo	
CRITERIO DE VALIDACIÓN: Que no haya errores en el ingreso	

ID: 4	USUARIO: CLIENTE
NOMBRE DE LA HISTORIA: NAVEGACIÓN	
PRIORIDAD: MEDIA	TIEMPO ESTIMADO: 4 SEMANS
RESPONSABLE: CARLOS	
El cliente debe ser capaz de navegar con fluidez por toda la página usando la barra de navegación	
CRITERIO DE VALIDACIÓN: Que no haya errores a la hora de que el usuario navegue	

## INFORME DE RETROSPECTIVA

### 1. Cosas que han funcionado bien.

La conexión con la base de datos, se pudo migrar los settings de la app para nuevos cambios

### 2. Cuales hay que mejorar.

Hay que ser más rápido y tener paciencia

### 3. ¿Qué quiere probar y hacer en la siguiente iteración

Ver si puedo meterle al vue o algun otro framework para estilizar

### 4. ¿Qué he aprendido?

Heroku y postman

### 5. ¿Cuáles son los problemas que podrían impedirle progresar adecuadamente?

Pues, la verdad siento que es muy poco lo que sé. Me costó mucho debugear y no sé como va a ser el despliegue, cosa que me tiene un poco inseguro