OrganizApp: Sistema de Gestión de Agenda de Actividades

Autor/a : Fernando Torrijos Silva
Tutor/a : Dr. Jesús María González Barahona

ETSIT - Universidad Rey Juan Carlos
Fuenlabrada

Curso Académico 2022/2023

Universidad Rey Juan Carlos

OrganizApp

Universidad Rey Juan Carlos

OrganizApp

Universidad Rey Juan Carlos

Introducción

- Herramienta web de agenda de actividades.
- Visualización, control y gestión de datos.

Universidad Rey Juan Carlos

Objetivos: general y específicos

- Objetivo general
  - Crear una herramienta web, capaz de organizar diariamente las actividades de una persona.
- Objetivos específicos
  - Visualización de datos.
  - Frontend: HTML5, Plotly, Bootstrap, CSS y JavaScript.
  - Backend: Python, Django y MySQL.
  - Despliegue.
  - Multiusuario.
  - Diseño atractivo.
  - Diferentes funcionalidades.
  - Usabilidad en dispositivos móviles.
  - Manejo de errores.
  - Intuitivo y fácil.
  - Mínimo riesgo de uso.

Universidad Rey Juan Carlos

OrganizApp

Universidad Rey Juan Carlos

Tecnologías empleadas

- *Backend*:

- *Frontend*:

-

Universidad Rey Juan Carlos

OrganizApp

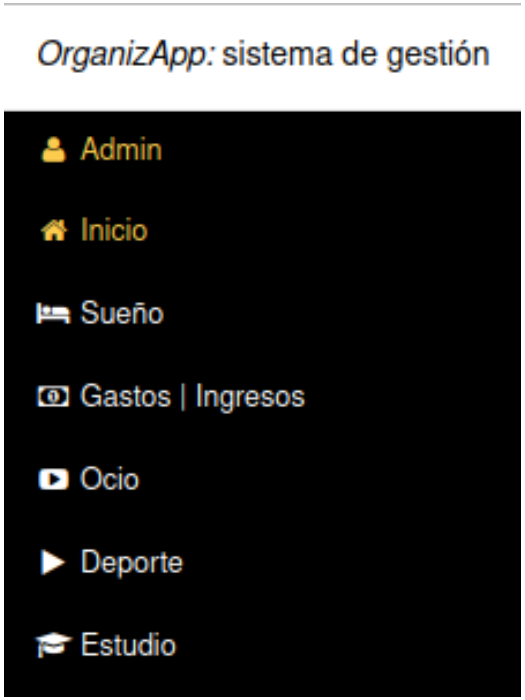
Universidad Rey Juan Carlos

Etapa 1: Configuración y puesta en marcha

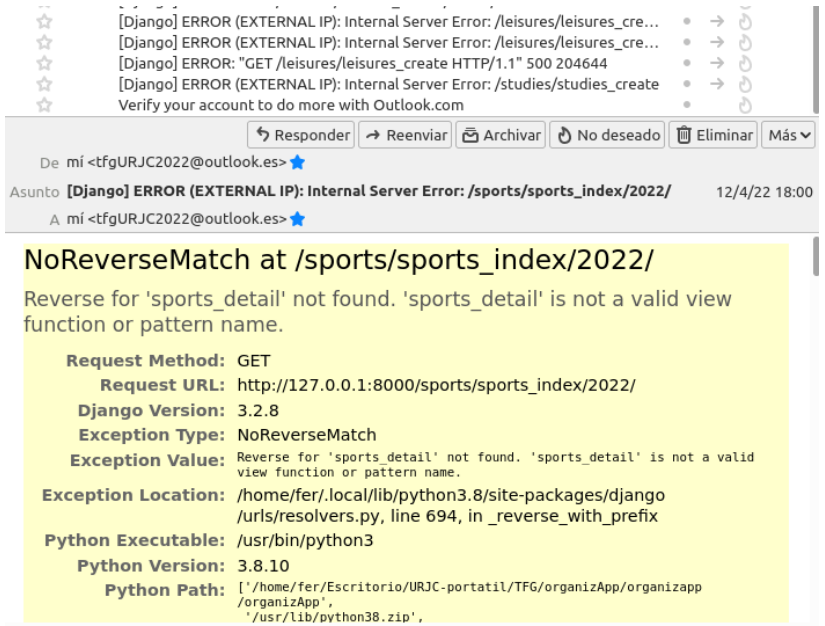
- Configuración de Vim.
- Creación de una cuenta de Gitlab.
- Configuración de Git.
- Preparación del entorno: Django.
- Creación de la Base de Datos.

Universidad Rey Juan Carlos

Etapa 2: Funcionalidades a desarrollar



Funcionalidades del proyecto.



Traceback.

Etapa 3: Diseño e imagen

- Menú y perfil de usuario.
- Intuitivo sin necesidad de manual.
- Edición en ventana emergente o *pop-up*.
- Paleta de colores agradable.

Universidad Rey Juan Carlos

Etapa 4: Visualización gráfica

Muchos más tipos de gráficas con **Plotly**.

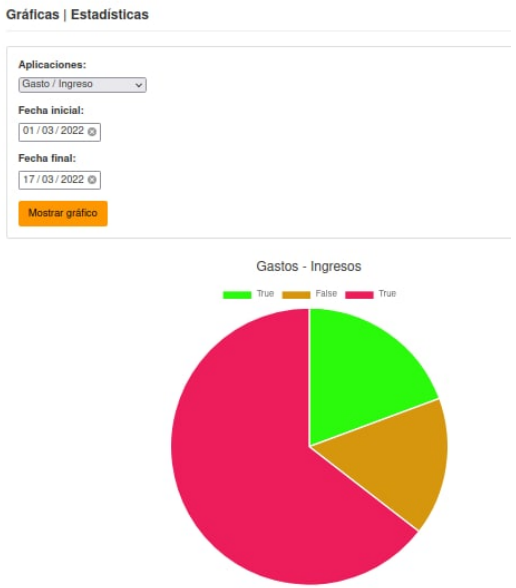
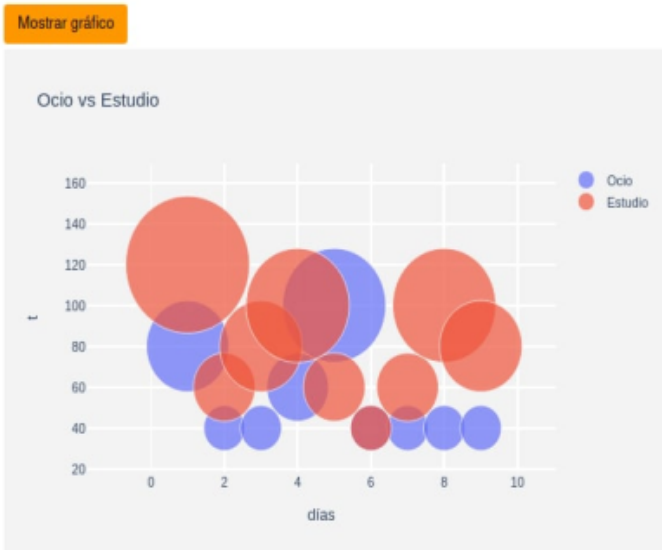


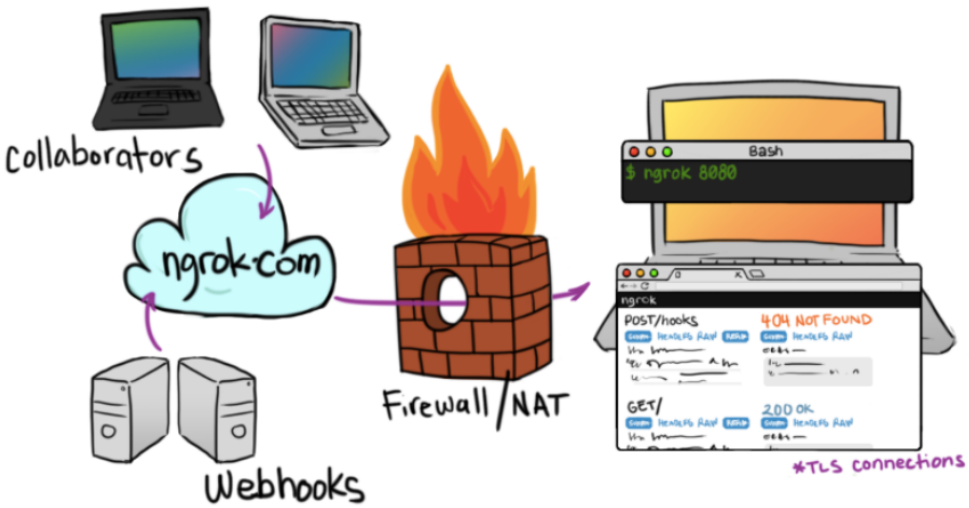
Diagrama circular con *Chart.js*.



Primer boceto 2D con Plotly.

Etapa 5: Despliegue y prueba de usuarios reales

- Uso de ngrok: creación de un túnel específico hacia un servidor local.

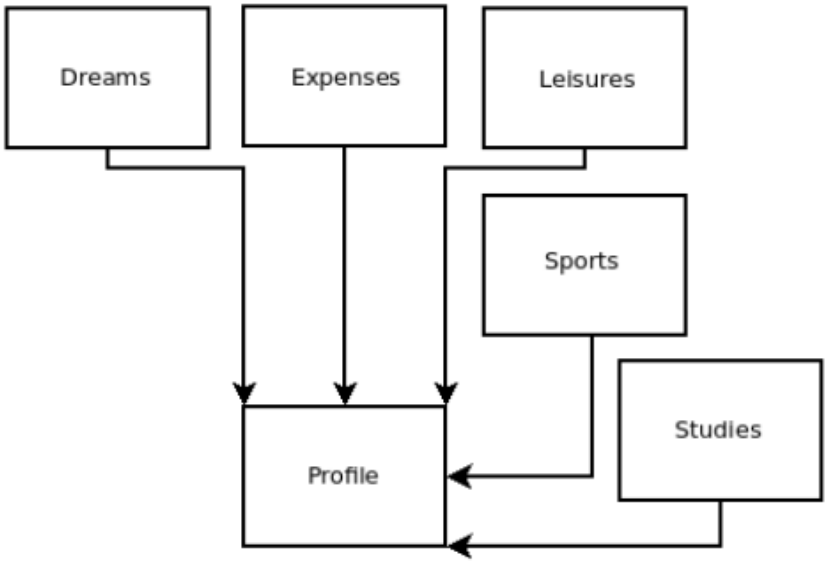


OrganizApp

Universidad Rey Juan Carlos

Visión general

La aplicación tiene su base en seis tablas. Todas tienen relación con *Profile*, para que cada usuario tenga un perfil personalizado e independiente del resto.



Universidad Rey Juan Carlos

Actividades

Cinco actividades desempeñadas por el usuario. En todas ellas se puede: crear, visualizar índice, visualizar detalles, eliminar y editar.

- **Sueño**: gestión de horas dormidas. Muestra: fecha, nombre usuario, horas dormidas y si padece alguna enfermedad.
- **Gastos — Ingresos**: controlar gastos e ingresos por mes del usuario. Uso de un *comando* en Django.
- **Ocio**: registrar lo que se hace en el tiempo libre.
- **Deporte**: actividad deportiva del usuario. Aparecen detalles como: día, usuario, nombre del deporte, zona del cuerpo, tiempo empleado, ...
- **Estudio**: fecha, asignatura y horas estudiadas.

Universidad Rey Juan Carlos

Características personales, menú y gráficas y estadísticas

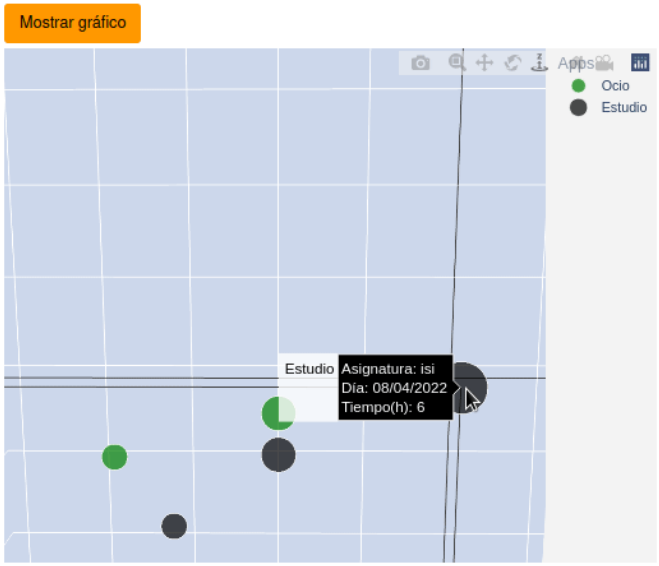
En la *home* o página principal, se puede visualizar el perfil del usuario con las características principales. Está dividida en tres partes:

- Características personales.
- Menú de aplicaciones.
- Gráficas y estadísticas.

Universidad Rey Juan Carlos

Características personales, menú y gráficas y estadísticas

Gráfica 3D.



Visualización de datos.

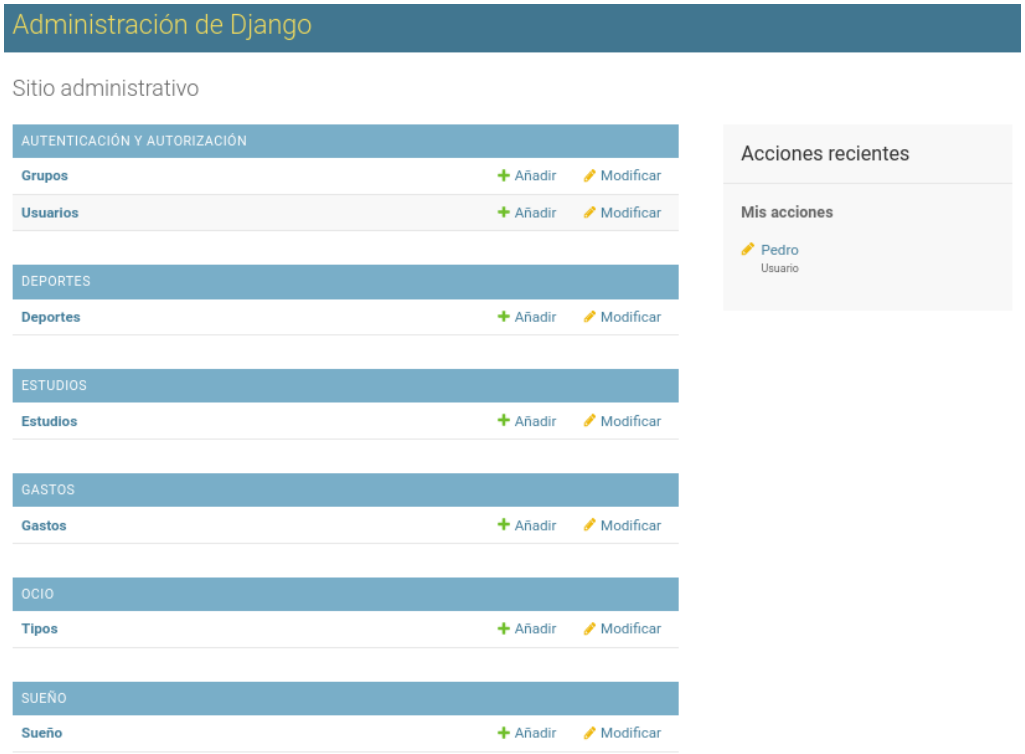
Registro, perfil y salir

- **Registro**. Para poder interactuar con las funcionalidades de la aplicación, hay que registrarse.
- **Perfil**. Detalles personales del usuario. Se puede editar tantas veces como se quiera.
- **Salir**. Al salir, se permite la entrada de otro usuario dando a *Acceder*.

Universidad Rey Juan Carlos

Administrador

Solo el Administrador podrá ver la opción de *Admin*, que permite manejar gráficamente la Base de Datos.

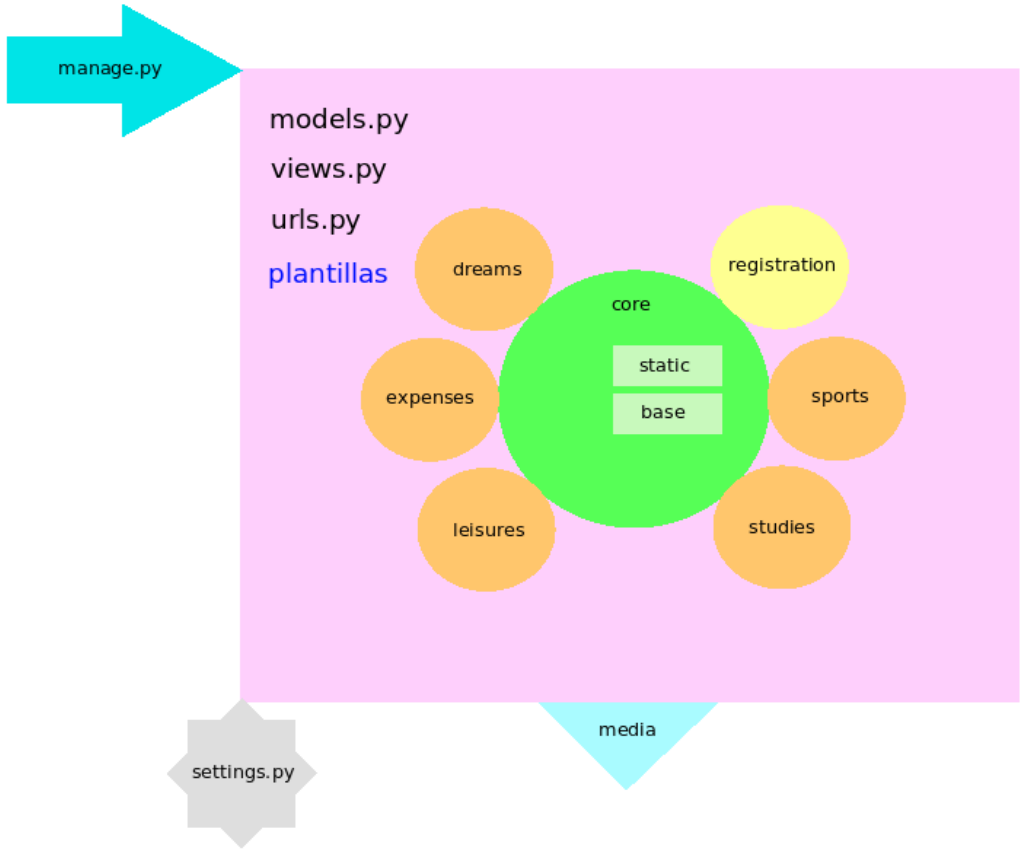


Universidad Rey Juan Carlos

OrganizApp

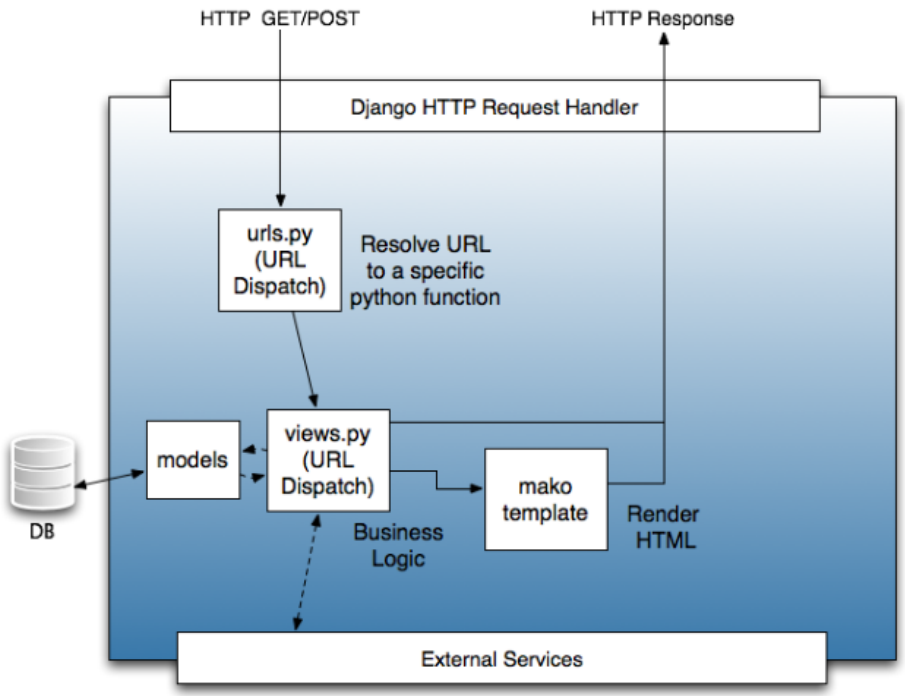
Universidad Rey Juan Carlos

Estructura del código: MVC y plantillas



Esquema general interno aplicación.

Estructura del código: MVC y plantillas



Esquema general interno Django: MVC.

Universidad Rey Juan Carlos

Actividades: software

Si se quiere ver el código fuente del proyecto:
https://gitlab.com/ftorrij/organizapp.

Universidad Rey Juan Carlos

Registro

Aplicación basada en CBV (Class Based Views): modelar las vistas (*views.py*) como clases en vez de funciones (FBV: Function Based Views): nivel más alto de reutilización de código.

- **class SignUpView(CreateView)**.
- **class ProfileUpdate(UpdateView)**.

Universidad Rey Juan Carlos

Núcleo o *core*

La parte donde comienza a crecer la aplicación web.

- Carpeta **static**: CSS, JavaScript, Bootstrap, imágenes, fuentes, etc.
- Plantilla base.

Universidad Rey Juan Carlos

Seguimiento

Creación de fichero de *log* nombrado como *tfg.log*.

```
  File "/home/fer/.local/lib/python3.8/site-packages/django/contrib/admin/templatetags/base.py", line 33, in render
    return super().render(context)
  File "/home/fer/.local/lib/python3.8/site-packages/django/template/library.py", line 214, in render
    _dict = self.func(*resolved_args, **resolved_kwargs)
  File "/home/fer/.local/lib/python3.8/site-packages/django/contrib/admin/templatetags/admin_list.py", line 308, in result_list
    'results': list(results(cl)),
  File "/home/fer/.local/lib/python3.8/site-packages/django/contrib/admin/templatetags/admin_list.py", line 284, in results
    yield ResultList(None, items_for_result(cl, res, None))
  File "/home/fer/.local/lib/python3.8/site-packages/django/contrib/admin/templatetags/admin_list.py", line 275, in __init__
    super().__init__(*items)
  File "/home/fer/.local/lib/python3.8/site-packages/django/contrib/admin/templatetags/admin_list.py", line 200, in items_for_result
    f, attr, value = lookup_field(field_name, result, cl.model_admin)
  File "/home/fer/.local/lib/python3.8/site-packages/django/contrib/admin/utils.py", line 278, in lookup_field
    value = attr()
  File "/home/fer/Escritorio/URJC-portatil/TFG/organizApp/organizapp/organizApp/leisures/models.py", line 44, in __str__
    return self.name
AttributeError: 'Leisures' object has no attribute 'name'
[21/Apr/2022 15:48:42] "GET /admin/leisures/leisures/ HTTP/1.1" 500 351384
[03/May/2022 11:34:16] "GET /favicon.ico HTTP/1.1" 404 3111
[03/May/2022 11:34:42] "GET /static/core/css/bootstrap.min.css.map HTTP/1.1" 404 1849
[03/May/2022 11:35:08] "GET /static/core/css/bootstrap.min.css.map HTTP/1.1" 404 1849
[03/May/2022 11:35:11] "GET /static/core/css/bootstrap.min.css.map HTTP/1.1" 404 1849
[03/May/2022 11:35:13] "GET /static/core/css/bootstrap.min.css.map HTTP/1.1" 404 1849
[03/May/2022 11:35:21] "GET /static/core/css/bootstrap.min.css.map HTTP/1.1" 404 1849
[03/May/2022 11:35:25] "GET /static/core/css/bootstrap.min.css.map HTTP/1.1" 404 1849
[03/May/2022 11:36:40] "GET /static/core/css/bootstrap.min.css.map HTTP/1.1" 404 1849
[03/May/2022 11:36:55] "GET /static/core/css/bootstrap.min.css.map HTTP/1.1" 404 1849
[20/Jun/2022 07:06:26] "GET /favicon.ico HTTP/1.1" 404 3111
[20/Jun/2022 07:07:32] [DREAM] user [espe] | curr_day [2022-06-20] | sleep_hours [5] | illness []
[20/Jun/2022 07:08:07] [EXPENSE] day_expense [2022-06-20] | quant [200] | category [1] | user [espe] | plus [False]
[20/Jun/2022 07:08:07] -- Changing money for espe, date: 2022/06/20, user: espe, quant: 200, plus: False
[20/Jun/2022 07:08:07] -- Old user found: espe
[20/Jun/2022 07:08:07] -- Expenses.update called with: quant=200, user=espe, plus > False
[11/Jul/2022 06:14:59] "GET /favicon.ico HTTP/1.1" 404 3111
[11/Jul/2022 06:15:10] [DREAM] user [espe] | curr_day [2022-07-11] | sleep_hours [5] | illness []
[11/Jul/2022 06:15:39] [EXPENSE] day_expense [2022-07-11] | quant [150] | category [1] | user [espe] | plus [False]
[11/Jul/2022 06:15:39] -- Changing money for espe, date: 2022/07/11, user: espe, quant: 150, plus: False
[11/Jul/2022 06:15:39] -- Old user found: espe
[11/Jul/2022 06:15:39] -- Expenses.update called with: quant=150, user=espe, plus > False
[11/Jul/2022 06:19:48] -- Changing money for espe, date: 2022/07/11, user: espe, quant: 150, plus: True
[11/Jul/2022 06:19:48] -- Old user found: espe
[11/Jul/2022 06:19:48] -- Expenses.update called with: quant=150, user=espe, plus > True
[11/Jul/2022 06:19:48] [EXPENSE] - '2022-07-11' has been _updated_ | quant = 150 | plus = True
[11/Jul/2022 06:32:52] Internal Server Error: /dreams/dreams_create/
```

Trozo de ejemplo de *tfg.log*.

Universidad Rey Juan Carlos

Errores

Los errores se pueden ver de dos maneras en este proyecto:

- Log.
- Correo.

Universidad Rey Juan Carlos

Interacción con la Base de Datos

```
mysql> SELECT * FROM expenses_expenses ORDER BY id DESC LIMIT 1;
+-----+-------------+-------+-----------------------------------+----------+--------------+------+---------+
| id  | day_expense | quant | concept                           | category | note         | plus | user_id |
+-----+-------------+-------+-----------------------------------+----------+--------------+------+---------+
| 134 | 2022-07-26  |   150 | gasolina fin de semana a salamanca | 1        | prueba crear |    0 |       1 |
+-----+-------------+-------+-----------------------------------+----------+--------------+------+---------+
```

Gasto guardado.

```
mysql> SELECT * FROM expenses_expenses ORDER BY id DESC LIMIT 1;
+-----+-------------+-------+---------+----------+---------------+------+---------+
| id  | day_expense | quant | concept | category | note          | plus | user_id |
+-----+-------------+-------+---------+----------+---------------+------+---------+
| 134 | 2022-07-26  |   150 |         | 1        | editar prueba |    0 |       1 |
+-----+-------------+-------+---------+----------+---------------+------+---------+
```

Gasto editado.

```
mysql> SELECT * FROM expenses_expenses ORDER BY id DESC LIMIT 1;
+-----+-------------+-------+---------+----------+------+------+---------+
| id  | day_expense | quant | concept | category | note | plus | user_id |
+-----+-------------+-------+---------+----------+------+------+---------+
| 133 | 2022-07-11  |   150 |         | 1        |      |    1 |       1 |
+-----+-------------+-------+---------+----------+------+------+---------+
```

Gasto eliminado.

Universidad Rey Juan Carlos

Validadores

Sin tener en cuenta los tests y ya teniendo todas las vistas creadas, hay que probar que los campos de los formularios son todos coherentes: fechas, cantidades, horas, etc. Para ello se usan los *validadores*.

```python
# ~~ validators

def clean_date_sport(self):
    date_sport = self.cleaned_data.get('date_sport')
    userS = self.cleaned_data.get('user')

    sports = Sports.objects.filter(
            date_sport=date_sport,
            user=userS)

    # use id
    if self.id:
        sports = sports.exclude(pk=self.id)

    # -- don't repeat days of the month
    if sports.count() > 0:
        raise forms.ValidationError(
            f'{date_sport} ya está creado. Modifique o borre el mismo.')

    if date_sport > datetime.date.today():
        raise forms.ValidationError(
                f'Fecha mayor que la actual: {datetime.date.today()}.')

    return date_sport
```

Validador para la app *sports*.

Universidad Rey Juan Carlos

Testing

Verificación en etapas tempranas del desarrollo de la arquitectura y el diseño de las aplicaciones, controlar, mejorar y avanzar sobre un proceso de construcción de un código más estable [1].

[1] https://craft-code.com/que-es-el-testing-de-software/

Universidad Rey Juan Carlos

Logging

Django permite tener registros o *logs* de todo lo que ocurre en cada acción que se realiza en la aplicación web.

- **Personalizable**: se tendrán registros de aquello que el programador quiera tener.
- **Gran ayuda en procesos concurrentes**.

Universidad Rey Juan Carlos

Despliegue con *ngrok*

- **Despliegue**: desplegar la aplicación de una manera sencilla con *ngrok*. Permite crear túneles que son seguros hacia un servidor local, es decir, a tu ordenador o donde tengas desplegada la aplicación.
- **VPN**: creación de VPN, para poder hacer cambios si fueran necesarios, desde cualquier parte con conexión a Internet.

Para ver los pasos a seguir en la instalación y ejecución de ngrok:*https://www.sdos.es/blog/ngrok-una-herramienta-con-la-que-hacer-publico-tu-localhost-de-forma-facil-y-rapida*.

Universidad Rey Juan Carlos

OrganizApp

Universidad Rey Juan Carlos

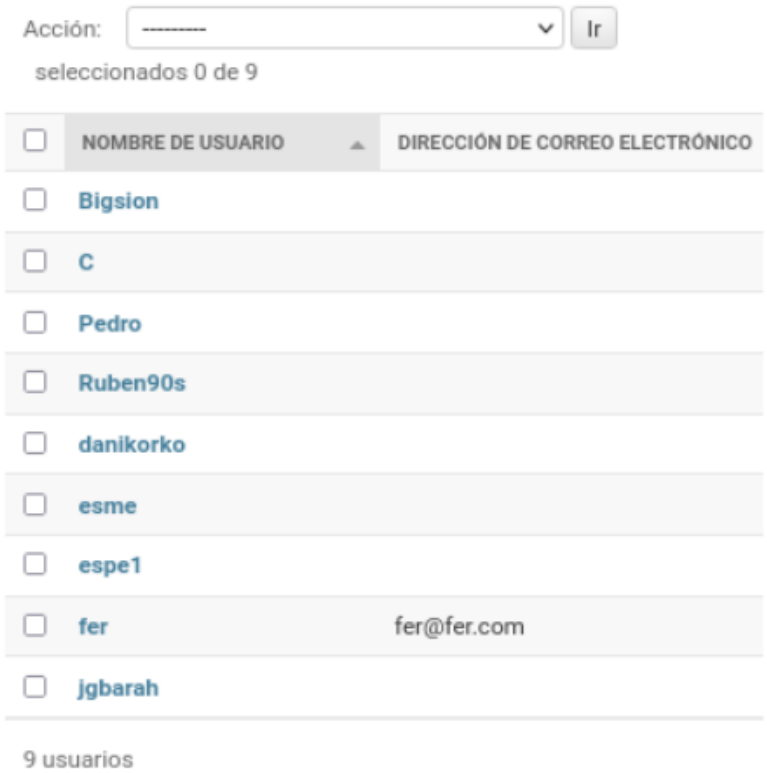
Conocimientos adquiridos

- Mejor manejo de JavaScript.
- ngrok.
- Perfeccionar uso de comandos en Django.
- Mejor uso de Bootstrap.
- Creación de VPN en Raspberry Pi.
- Uso de Plotly.

Universidad Rey Juan Carlos

Prueba de usuarios externos

En total, han colaborado nueve usuarios. Puntos de vista y valoraciones. Gran ayuda a la hora de mejorar la aplicación.



Universidad Rey Juan Carlos

Trabajos futuros

- Lista de tareas para cumplir como objetivo.
- Premiar si se cumplen las tareas.
- Alertas si no se cumplen los objetivos.
- Incidencias dentro de la aplicación.
- Descarga de datos en CSV.
- Indicativo de porcentaje superado.
- Poner muchas más actividades.

Universidad Rey Juan Carlos

Gracias

" Gracias a Espe, Rocis, Alfon, Mili, Adriano, mis padres y mis hermanas. Y a todos mis profesores, por querer sacar mi mejor versión. Gracias a todos, de verdad, sin vosotros no hubiera sido posible."

Universidad Rey Juan Carlos