

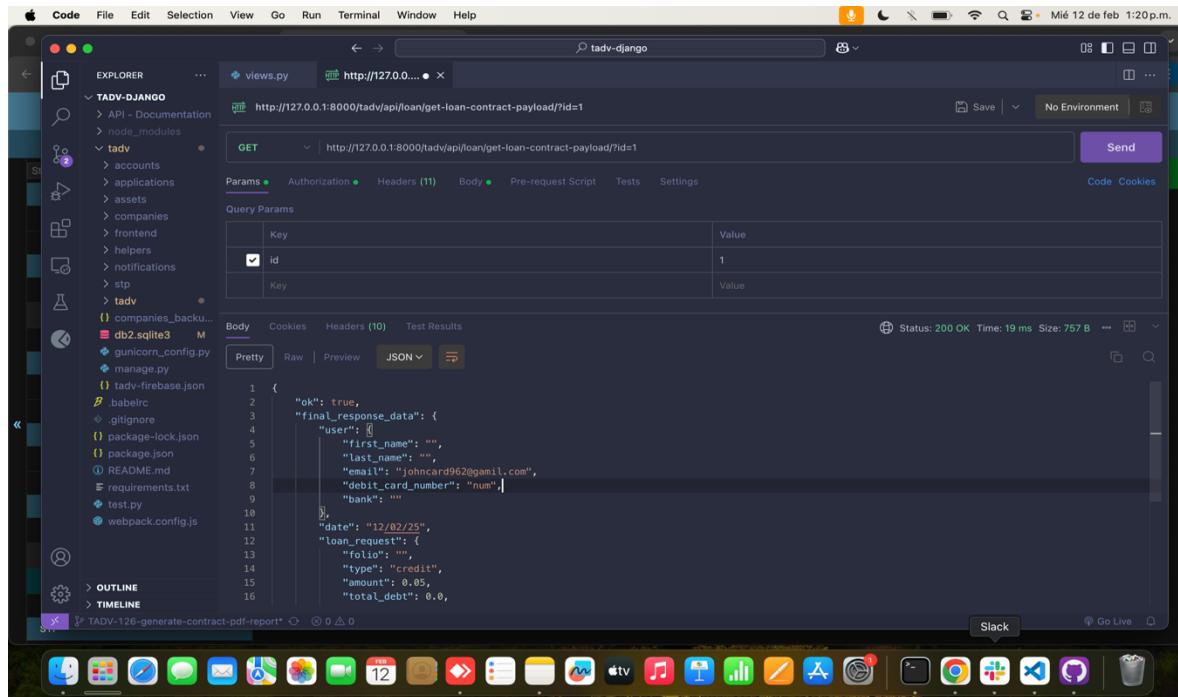
Curso para Desarrolladores Django de EBAC:

The screenshot shows a web browser window for the EBAC LMS. The title bar indicates it's a Chrome browser on a Mac OS X system. The main content is an activity titled "Actividad" for user "Juan Carlos". The activity details a GitHub link: https://github.com/JohnCard/backend_project. A message from "Carlos Ávila" dated "oct 03 13:43" says: "Buen trabajo Juan Carlos! Sigue así!". Below this, a note says: "Súper bien que agregaste tu código en el archivo para que se más fácil identificarlo". A green checkmark icon and the text "¡Actividades completadas con éxito!" are displayed. A progress bar shows "100 / 100". The top navigation bar includes links for "Mis cursos", "En Vivo", and "Centro de ayuda". The bottom status bar shows the date as "Sáb 10 de may 3:58 p.m.".

The screenshot shows a web browser window for the EBAC LMS. The title bar indicates it's a Chrome browser on a Mac OS X system. The main content is a course titled "Backend Python". It lists five lessons with completion status: "Introducción a REST" (1/3), "Configurando Django Rest Framework" (1/4), "Paginación en Django Rest Framework" (1/4), "Autenticación con Django" (1/8), and "Web Scraping" (1/4). A purple question mark icon is visible in the bottom right corner. The top navigation bar includes links for "Mis cursos", "En Vivo", and "Centro de ayuda". The bottom status bar shows the date as "Sáb 10 de may 4:05 p.m.".

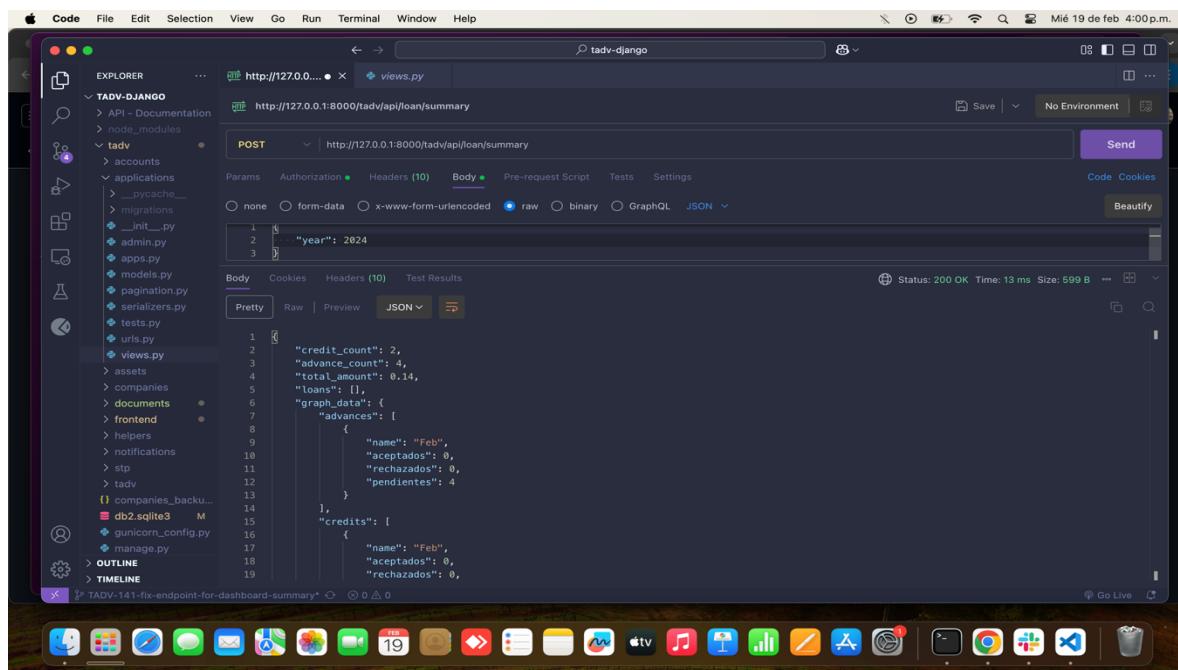
Evidencias de todas mis API view creadas con Django rest framework, documentadas con archivos .md y con librerías para cualquier modelo, diseñadas y personalizables para vistas con enfoques mas específicos, vistas que trabajan con librerías para optimizar procesos similares:

Vistas personalizables:



The screenshot shows a Postman interface with a GET request to `http://127.0.0.1:8000/tadv/api/loan/get-loan-contract-payload/?id=1`. The response body is a JSON object representing a loan contract payload. The response status is 200 OK.

```
1 {
  "ok": true,
  "final_response_data": {
    "user": [
      {
        "first_name": "",
        "last_name": "",
        "email": "johncard962@gmail.com",
        "debit_card_number": "num"
      }
    ],
    "date": "12/02/25",
    "loan_request": {
      "folio": "",
      "type": "credit",
      "amount": 0.05,
      "total_debt": 0.0
    }
  }
}
```



The screenshot shows a Postman interface with a POST request to `http://127.0.0.1:8000/tadv/api/loan/summary`. The request body contains a single key-value pair: `year: 2024`. The response body is a JSON object containing loan summary data for the year 2024. The response status is 200 OK.

```
1 {
  "credit_count": 2,
  "advance_count": 4,
  "total_amount": 0.14,
  "loans": [],
  "graph_data": {
    "advances": [
      {
        "name": "Feb",
        "aceptados": 0,
        "rechazados": 0,
        "pendientes": 4
      },
      {
        "name": "Feb",
        "aceptados": 0,
        "rechazados": 0
      }
    ]
  }
}
```

Vistas que trabajan con archivos

The screenshot shows the Postman application interface. A POST request is being made to `http://127.0.0.1:8000/tadv/api/auth/documents/`. The request body contains three parameters: `file`, `filename`, and `user`. The `file` parameter has a value pointing to a local file path. The `filename` parameter has a value of `filename.pdf`. The `user` parameter has a value of `1`. The response status is `201 Created` with a time of `29 ms` and a size of `536 B`. The response body is a JSON object containing an ID, creation date, update date, filename, file URL, and user information.

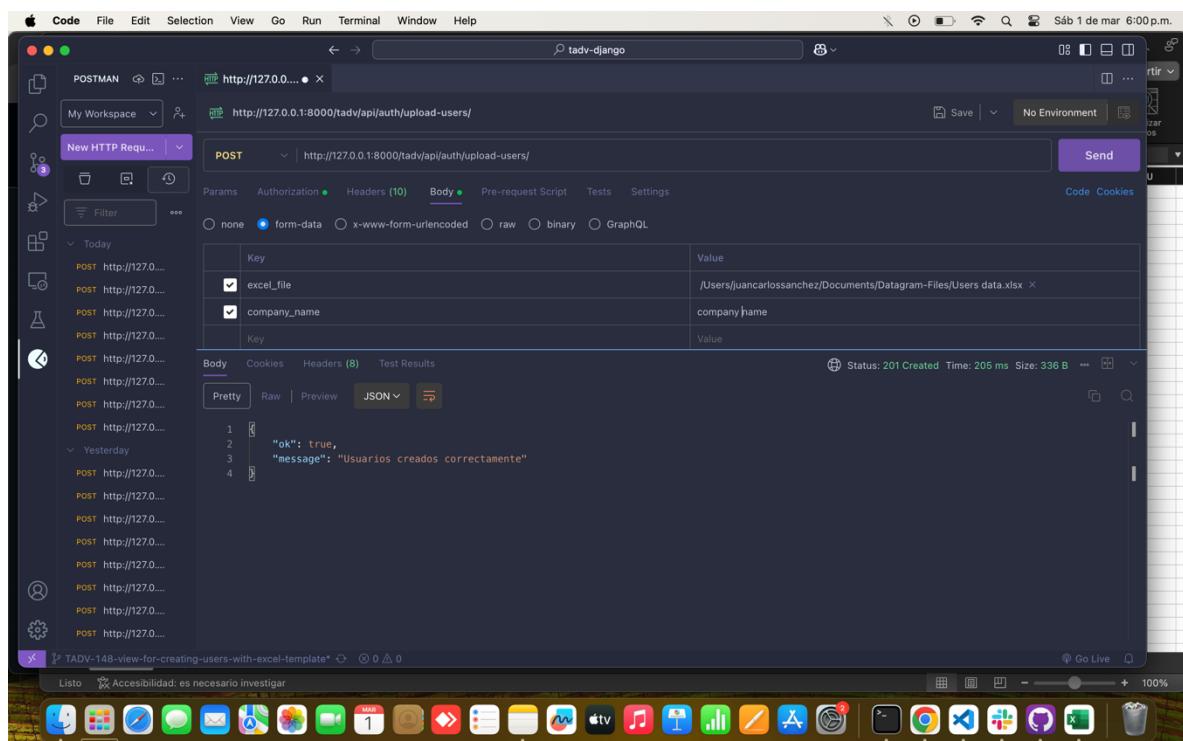
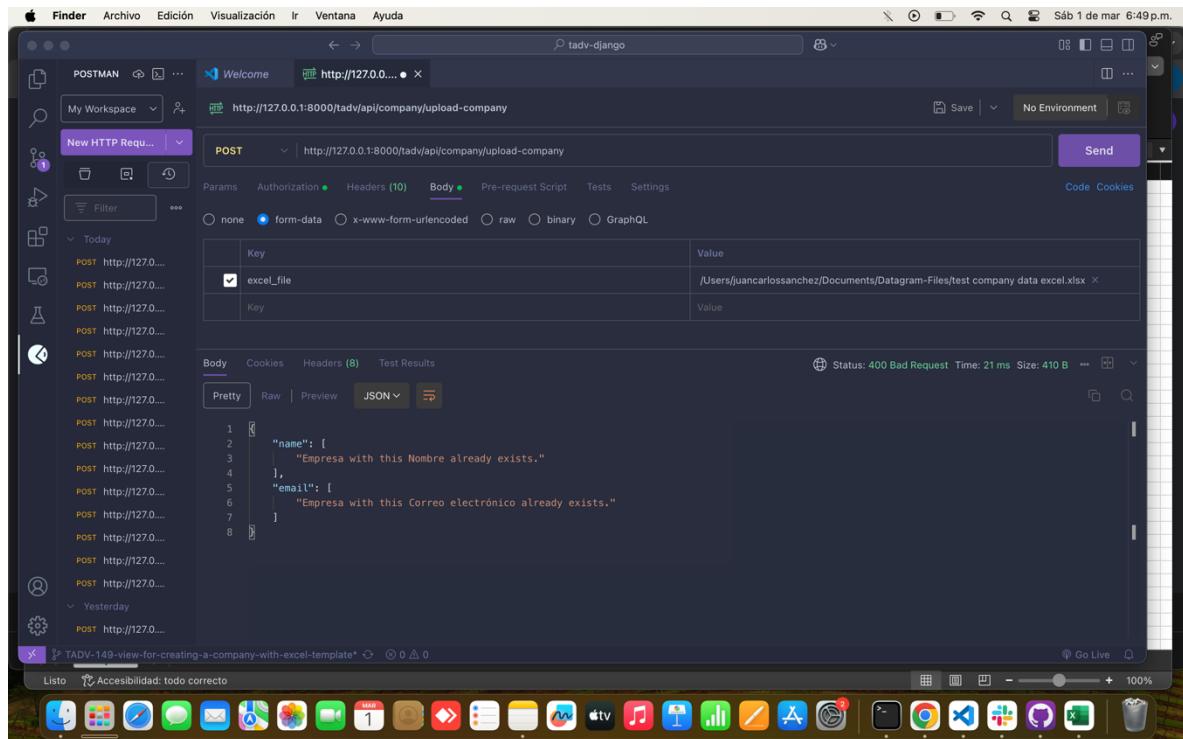
```
1 "id": 15,
2 "created_at": "2025-02-19T06:50:56.839931-06:00",
3 "updated_at": "2025-02-19T06:50:56.839980-06:00",
4 "filename": "filename.pdf",
5 "file": "http://127.0.0.1:8000/tadv/media/ABCD901010XXX/filename.pdf",
6 "user": 1
```

Vistas que se comunican con servicios exteriores

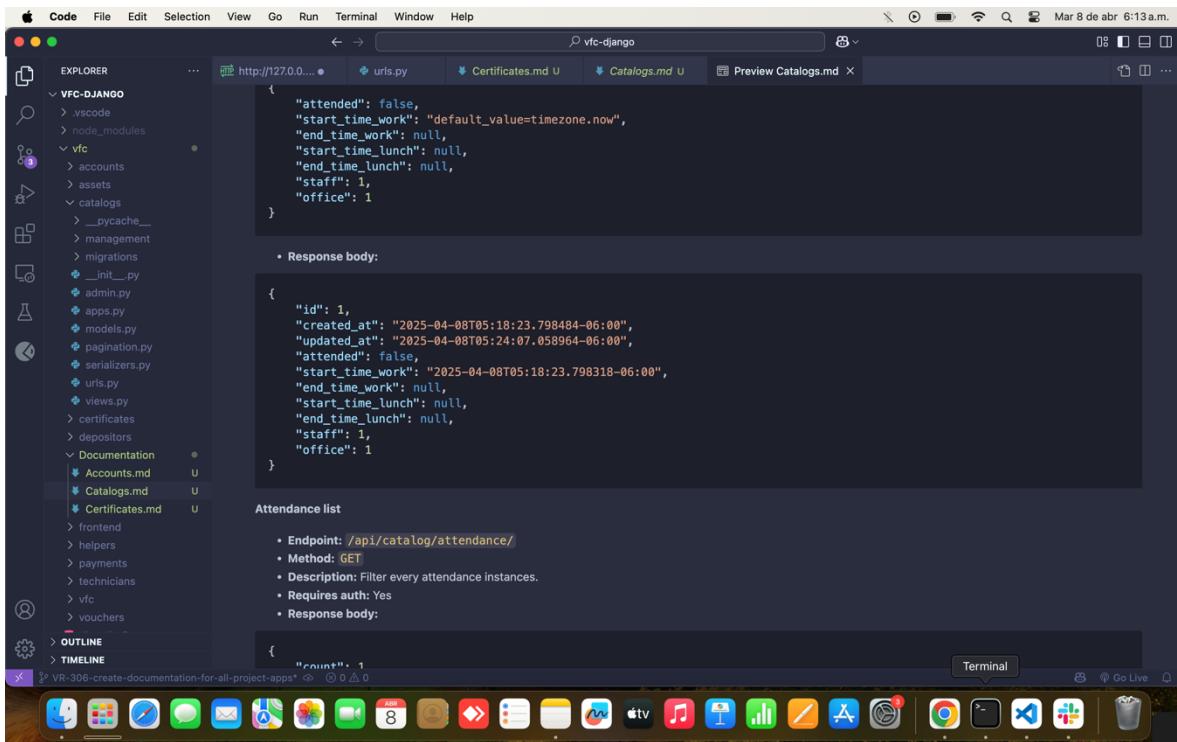
The screenshot shows the Postman application interface. A POST request is being made to `http://127.0.0.1:8000/tadv/api/stp/`. The request body is a raw JSON object with several fields: `data`, `empresa`, `claveRastreo`, `tipoOrden`, and `fechaOperacion`. The response status is `200 OK` with a time of `1042 ms` and a size of `1.13 KB`. The response body is a JSON object containing an `ok` field, a `data` field, and a `mensaje` field indicating successful retrieval of data.

```
1 {
2   "data": [
3     {"empresa": "HDA SOLUCIONES",
4      "claveRastreo": "123456790",
5      "tipoOrden": "E",
6      "fechaOperacion": "20250284"
7   ],
8   "ok": true,
9   "mensaje": "Datos consultados correctamente"
10 }
```

Cargas masivas de datos desde archivos Excel & uso de serializadores para manejar errores



Documentación de APIs



Tasks API

create

POST /tasks/api/v1/tasks/

Request Body

The request body should be a "application/json" encoded object, containing the following items.

| Parameter | Description |
|-----------------------|-------------|
| title required | |
| description | |
| done | |

read

GET /tasks/api/v1/tasks/{id}/

Path Parameters

The following parameters should be included in the URL path.

| Parameter | Description |
|---------------------------|---|
| id required | A unique integer value identifying this task. |

update

PUT /tasks/api/v1/tasks/{id}/

Path Parameters

The following parameters should be included in the URL path.

| Parameter | Description |
|-----------|-------------|
|-----------|-------------|

CRUDS completos (vistas basadas en clases como APIView, UpdateView, DeleteView, CreateView, RetrieveUpdateDestroyAPIView, ListCreateAPIView, etc.)

The image displays two screenshots of the Postman application interface, showing API requests for log entries.

Screenshot 1: A GET request to `http://127.0.0.1:8000/tadv/api/auth/logs`. The response body shows a list of logs with an error message for missing user information.

```
1 {
  "count": 14,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 15,
      "user": null,
      "action": "CREATE",
      "message": "required field"
    },
    {
      "id": 14,
      "user": null,
      "action": "CREATE",
      "message": "required field"
    },
    {
      "id": 13,
      "user": null,
    }
  ]
}
```

Screenshot 2: A GET request to `http://127.0.0.1:8000/tadv/api/auth/log/12`. The response body shows a single log entry with an error message for missing user information.

```
1 [
  {
    "id": 12,
    "user": null,
    "action": "CREATE",
    "message": "required field"
  }
]
```

