

Preparación proyecto final

WEB_SCRAPING - ETL-MONGODB- ELASTIC-GITHUB- render.com

Render.com / GitHub / python
(flask)

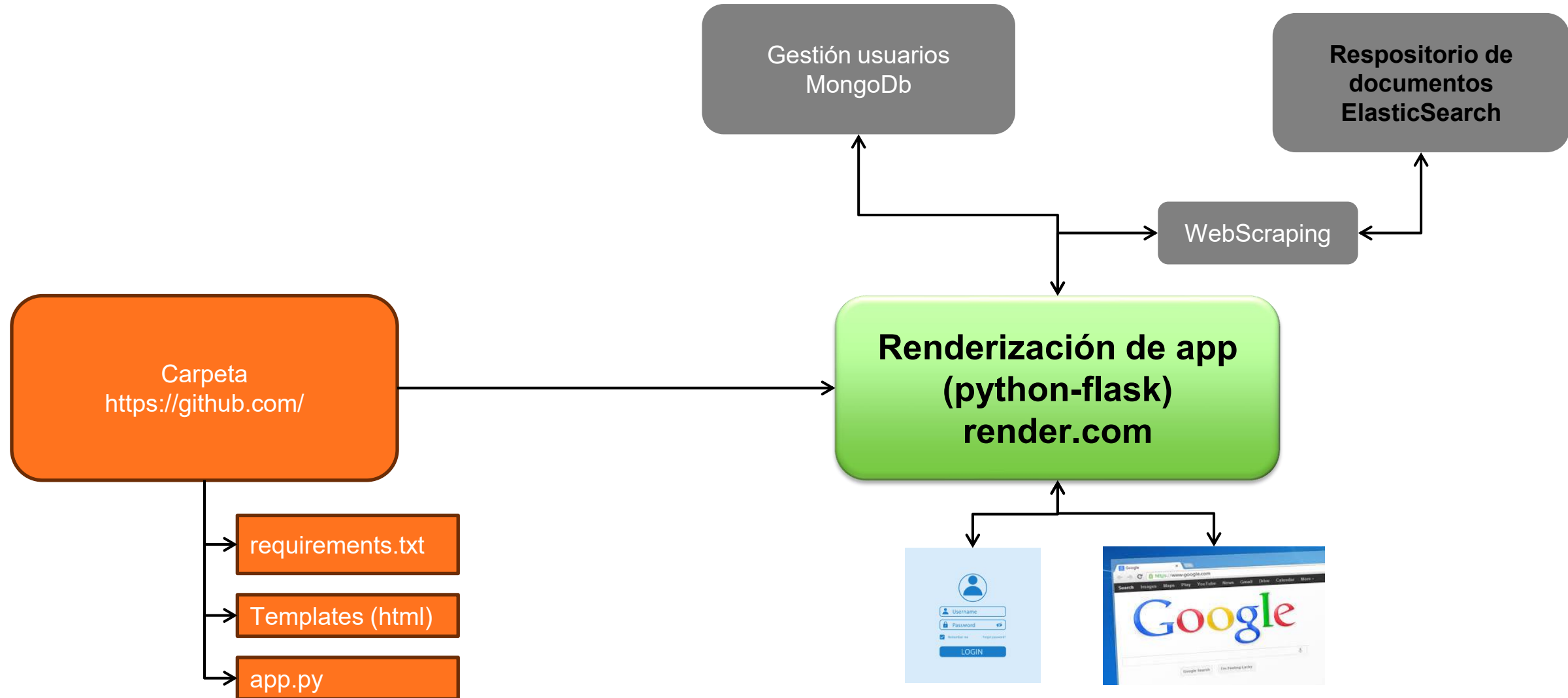
Luis Fernando Castellanos Guarin
2025



UNIVERSIDAD
CENTRAL



1. Proyecto final (renderización web)



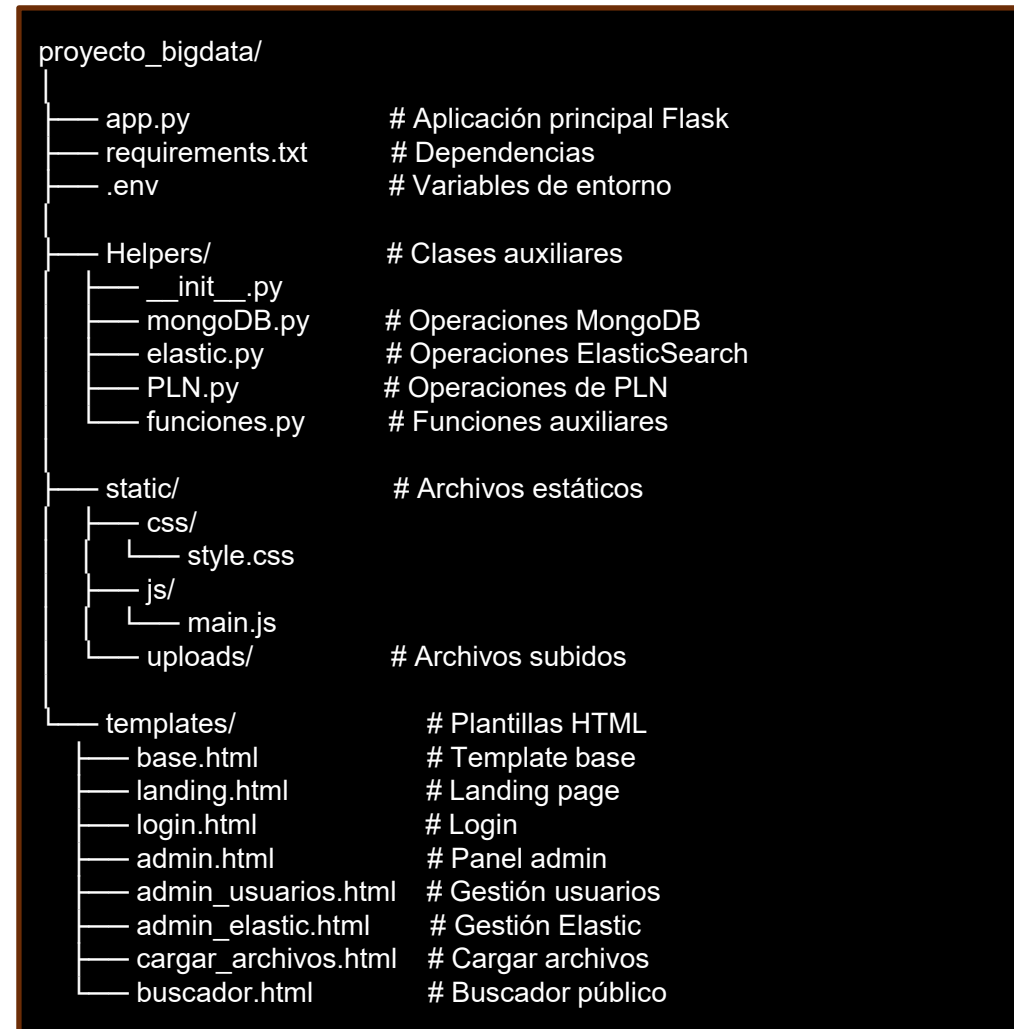
2. Requerimientos

1. Tener cuenta en <https://github.com/>
 - Crear una carpeta a la que llamaremos “**Proyecto_BigData**”
 - en esta carpeta crearemos todo el proyecto en python (flask)
2. Tener cuenta en render.com
 - **Crear proyecto**
 - **Indexar carpeta de GitHub (usuario)**

3. Que vamos a crear?

Un proyecto web para BIGDATA que contenga:

1. Una landingPage
2. Una pagina de login con validación en usuario en mongodb (mongoAtlas)
3. Una pagina admin (solo accesible después de hacer login) con acceso a 3 funciones: "Administrar usuarios", "Administrar elastic", "cargar archivos a elastic"
 - 3.1 Una sub-pagina para administrar usuarios
 - 3.2 Una pagina para administrar Elastic
 - 3.3 Una pagina para cargar archivos a Elastic (PLN)
4. una pagina con un buscador



Versión 1

1. crear una landingPage con bootstrap 5 que tenga un fondo y estilo profesional pero sencillo header + un body donde este el nombre del proyecto + footer con nombre del creador, fecha y versión
2. Crear una pagina de login que tenga un formulario que permita validar un usuario/contraseña en mongodb (mongoAtlas)
3. Crear una pagina de administración (solo accesible después de hacer login) con acceso a 3 funciones: "Administrar usuarios", "Administrar elastic", "cargar archivos a elastic" y que tenga un estilo (colores diferentes) a la landingPage

```
proyecto_bigdata/  
├── app.py                # Aplicación principal Flask  
├── requirements.txt      # Dependencias  
├── .env                 # Variables de entorno  
├── Helpers/             # Clases auxiliares  
│   ├── __init__.py  
│   ├── mongoDB.py      # Operaciones MongoDB  
│   └── funciones.py     # Funciones auxiliares  
├── static/              # Archivos estáticos  
│   ├── css/  
│   │   └── style.css  
│   ├── js/  
│   │   └── main.js  
│   └── uploads/         # Archivos subidos  
└── templates/           # Plantillas HTML  
    ├── base.html        # Template base  
    ├── landing.html     # Landing page  
    ├── login.html       # Login  
    └── admin.html       # Panel admin
```

- Crearemos una pagina para administrar usuarios, siempre que el usuario tenga el permiso "admin_usuarios":true
- Crearemos una pagina para administrar elastic (crear/modificar/eliminar index o columnas de un index) usando un campo "comando_json" y un botón para ejecutar el comando, siempre que el usuario tenga el permiso "admin_elastic":true

```
proyecto_bigdata/  
├── app.py           # Aplicación principal Flask  
├── requirements.txt # Dependencias  
├── .env            # Variables de entorno  
├── Helpers/  
│   ├── __init__.py # Clases auxiliares  
│   ├── mongoDB.py  # Operaciones MongoDB  
│   ├── PLN.py      # Operaciones de PLN  
│   └── funciones.py # Funciones auxiliares  
├── static/         # Archivos estáticos  
│   ├── css/  
│   │   └── style.css  
│   ├── js/  
│   │   └── main.js  
│   └── uploads/    # Archivos subidos  
└── templates/      # Plantillas HTML  
    ├── base.html   # Template base  
    ├── landing.html # Landing page  
    ├── login.html  # Login  
    ├── admin.html  # Panel admin  
    ├── admin_usuarios.html # Gestión usuarios  
    └── admin_elastic.html # Gestión Elastico
```

Haremos todo del numeral 3.3 y 4

FIN BIGDATA 2025 - S2



UNIVERSIDAD
CENTRAL



Gracias