

Manual Tecnico

ETL for employees and companies

VERSION 1.0

FECHA: 18/04/2022

DEVELOPER: SERGIO DANIEL CIFUENTES ARGUETA

ORGANIZACIÓN: CYCS

1 DESCRIPCIÓN

El software desarrollado consiste en un ETL (Extract, transform and Load), el cuál le permite a la organización mover datos de empleados y compañías desde un almacenamiento de datos tipo CSV hacia una base de datos específica para generar una planilla general con los datos que son requeridos.

2 BUILDING

- Python es un lenguaje interpretado, por lo que django a través del archivo **manage.py** se encarga de realizar las tareas administrativas, así mismo realiza la importación de python para su uso, y todos los demás elementos necesarios.

3 DEVELOPING

- Modelo Vista-Controlador
- Lenguaje **Python**
- Framework **Django**
- Editor utilizado **Visual Studio Code**
- Base de datos **postgreSQL**

4 REQUERIMIENTOS GENERALES

- Navegador

5 EJECUCIÓN DEL SERVIDOR

- Se realiza a través del comando
`python manage.py runserver`

6 ESTRUCTURAS GENERALES EMPLEADAS EN BDD

- PERSONA

```
CREATE TABLE REP_PERSONA (ID_PERSONA INT PRIMARY KEY,
DPI VARCHAR(20),
PRIMER_NOMBRE VARCHAR(50),
SEGUNDO_NOMBRE VARCHAR(50),
PRIMER_APELLIDO VARCHAR(50),
SEGUNDO_APELLIDO VARCHAR(50),
APELLIDO_CASADA VARCHAR(50),
ORDEN_CEDULA VARCHAR(50),
REGISTRO_DEDULA VARCHAR(50),
DIRECCION_RESIDENCIA VARCHAR(50),
NIT VARCHAR(15),
GENERO BOOLEAN,
TELEFONO VARCHAR(15),
CORREO_ELECTRONICO VARCHAR(50),
FECHA_NACIMIENTO DATE);
```

- EMPRESA

```
CREATE TABLE REP_EMPRESA (ID_EMPRESA INT PRIMARY KEY,
NOMBRE_EMPRESA VARCHAR(50),
NIT VARCHAR(15),
CODIGO VARCHAR(15),
DIRECCION VARCHAR(50),
TELEFONO VARCHAR(15)
);
```

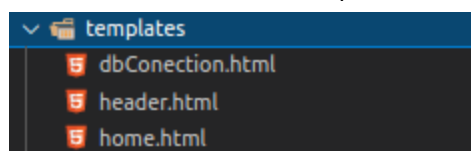
- TRABAJO

```
CREATE TABLE REP_TRABAJO (ID_TRABAJO INT PRIMARY KEY,
ID_PERSONA INT,
FECHA_INICIAL DATE,
FECHA_FINAL DATE,
ID_EMPRESA INT,
NOMBRE_PUESTO VARCHAR(50),
MES_PLANILLA VARCHAR(10),
SALARIO DECIMAL,
FOREIGN KEY (ID_PERSONA) REFERENCES REP_PERSONA(ID_PERSONA),
FOREIGN KEY (ID_EMPRESA) REFERENCES REP_EMPRESA(ID_EMPRESA)
);
```

7 DESCRIPCIÓN GENERAL

- **templates**

Son todas las vistas de las que hace uso la página web y de las que Django se encarga de renderizar en sus respectivas rutas.



- **ddl.sql**

Es el script de definición de la base de datos.

- **etl/settings.py**

Clase que contiene todas las configuraciones generales de uso como por ejemplo middlewares, templates y librerías, lo importante de este archivo es que aquí se encuentran las credenciales para conectar a la base de datos

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'etl',
        'USER': 'postgres',
        'PASSWORD': 'postgres',
        'HOST': 'localhost',
        'PORT': '5432',
    }
}
```

- **etl/urls.py**

Clase que contiene todas las urls a utilizar

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('home/', home, name='home'),
]
```

8 CONEXIÓN A BASE DE DATOS

- **etls/models.py**

Se crea el modelo de la conexión

```
class Connection(models.Model):
    id=models.AutoField(primary_key=True)
    name=models.CharField(max_length=15)
    database=models.IntegerField()
    user=models.CharField(max_length=50)
    password=models.CharField(max_length=100)
    host=models.CharField(max_length=50)
    databaseName=models.CharField(max_length=50)
    port=models.IntegerField()
```

- **etls/views.py**

El modelo anteriormente estipulado se crea en la vista

```
if request.method == 'GET':
    return render(request, 'home.html')
    print("get")
else:
    print("enter")
    # conection=Conection(name="x",database=1,user='postgres',
    #                     password='database123',host='localhost',databaseName='test',
    #                     port=5432)
    conection=Conection.objects.get(name="x")
    conection.save()
    print(conection)
    file=request.FILES['formFile']
    fs = FileSystemStorage()
    filename = fs.save(file.name, file)

    uploaded_file_url = fs.url(filename)
    print(uploaded_file_url)
    start(conection,uploaded_file_url)

    return render(request, 'home.html')
```