

Actualizar la página de Comité de gobierno digital y publicar archivos



PERÚ

Ministerio de Salud

Viceministerio de Prestaciones y Aseguramiento en Salud

Dirección de Redes Integradas de Salud Lima Sur

[Inicio](#) [Directorio de Funcionarios](#) [Serums](#)

COMITÉ DE GOBIERNO DIGITAL

1 ACTAS DE TRABAJO

2 ACTAS FIRMADAS

3 DOCUMENTOS DE TRABAJO

4 DOCUMENTACIÓN

- [Aviso de Sinceramiento Diris Lima Sur](#)

Disponible en formato pdf

- [RD 960-2024-DIRIS-LS-DG - CONFORMACION DEL COMITÉ DE GOBIERNO DIGITAL](#)

Disponible en formato pdf

Actualizar la página web de Estadística y publicar reportes



PERÚ

Ministerio de Salud

Viceministerio de Prestaciones y Aseguramiento en Salud

Dirección de Redes Integradas de Salud Lima Sur

[Inicio](#) [Directorio de Funcionarios](#) [Serums](#)

1 TABLEROS ESTADÍSTICOS

2 DATOS ESTADÍSTICOS

3 NORMATIVIDAD

4 MANUALES

5 VIDEOS TUTORIALES

Barrido Nacional de Vacunación contra el Sarampión 2024

- [* Vacunómetro del barrido SPR](#)
- [* Formato web del conteo rápido SPR](#)
- [* Videotutorial del sistema de información del barrido SPR](#)

Tablero del de Vacuna Regulares del Periodo 2024

- [* TABLERO: PRODUCCIÓN DE VACUNAS REGULARES](#)

Información de Consulta Externa

- [* TABLERO: SEGUIMIENTO DE ATENCIONES](#)
- [* TABLERO: MORBILIDAD Y PROCEDIMIENTOS](#)
- [* TABLERO: PRODUCCIÓN DE PERSONAL](#)

Indicadores de Gestión

- [* TABLERO: CONTROL Y SEGUIMIENTO DE INDICADORES DE DESEMPEÑO](#)
- [* TABLERO: CONTROL Y SEGUIMIENTO DE INDICADORES DE FED](#)
- [* TABLERO: INDICADORES DE CONVENIO](#)
- [* TABLERO: INDICADORES TRAZADORES](#)

Control de Calidad

- [* TABLERO: CONTROL DE CALIDAD DE REGISTROS DE ESTRATEGIAS Y ETAPA DE VIDA](#)
- [* TABLERO: DATOS OBSERVADOS DE ANEMIA](#)

Avances de Metas Fisicas

- [* TABLERO: AVANCES DE METAS FISICAS DE NUTRICION](#)
- [* TABLERO: PADRÓN NOMINAL](#)

Actualizar la página web de Oficina de Epidemiología, Inteligencia Sanitaria y Docencia e Investigación y publicar archivos



PERÚ

Ministerio de Salud

Viceministerio de Prestaciones y Aseguramiento en Salud

Dirección de Redes Integradas de Salud Lima Sur

[Inicio](#) [Directorio de Funcionarios](#) [Serums](#)

Oficina de Epidemiología, Inteligencia Sanitaria y Docencia e Investigación

1 BOLETINES EPIDEMIOLÓGICOS

2 SALAS SITUACIONALES

3 SALAS TEMÁTICAS

4 ALERTAS EPIDEMIOLÓGICAS

5 FICHAS EPIDEMIOLÓGICAS

6 ASIS

7 MAPAS EPIDEMIOLÓGICOS

2025

• Boletín Epidemiológico N° 02

Boletín Epidemiológico del 05 al 11 de enero

Disponible en formato PDF

• Boletín Epidemiológico N° 01

Boletín Epidemiológico del 29 diciembre de 2024 al 04 de enero de 2025

Disponible en formato PDF

2024

• Boletín Epidemiológico N° 51

Boletín Epidemiológico del 15 al 21 de diciembre

Disponible en formato PDF

• Boletín Epidemiológico N° 50

Boletín Epidemiológico del 08 al 14 de diciembre

Disponible en formato PDF



PERÚ

Ministerio
de Salud

Viceministerio
de Prestaciones y
Aseguramiento en Salud

Dirección de Redes
Integradas de Salud
Lima Sur

[Inicio](#) [Directorio de Funcionarios](#) [Serums](#)

1 BOLETINES EPIDEMIOLÓGICOS

2 SALAS SITUACIONALES

3 SALAS TEMÁTICAS

4 ALERTAS EPIDEMIOLÓGICAS

5 FICHAS EPIDEMIOLÓGICAS

6 ASIS

7 MAPAS EPIDEMIOLÓGICOS

• **Sala Situacional Dengue N° 51**

Sala Situacional de Dengue del 15 al 21 de diciembre
Disponible en formato PDF

• **Sala Situacional Dengue N° 47**

Sala Situacional de Dengue del 17 al 23 de noviembre
Disponible en formato PDF

• **Sala Situacional Dengue N° 46**

Sala Situacional de Dengue del 10 al 16 de noviembre
Disponible en formato PDF

• **Sala Situacional Dengue N° 45**

Sala Situacional de Dengue del 03 al 09 de noviembre
Disponible en formato PDF

• **Sala Situacional Dengue N° 44**

Sala Situacional de Dengue del 27 de octubre al 02 de noviembre
Disponible en formato PDF

• **Sala Situacional Dengue N° 43**

Sala Situacional de Dengue del 20 al 26 de octubre
Disponible en formato PDF



PERÚ

Ministerio
de Salud

de Prestaciones y
Aseguramiento en Salud

Integradas de Salud
Lima Sur

[Inicio](#) [Directorio de Funcionarios](#) [S](#)

1 BOLETINES EPIDEMIOLÓGICOS

2 SALAS SITUACIONALES

3 SALAS TEMÁTICAS

4 ALERTAS EPIDEMIOLÓGICAS

5 FICHAS EPIDEMIOLÓGICAS

6 ASIS

7 MAPAS EPIDEMIOLÓGICOS

• **AE 012-2024**

Disponible en formato PDF

• **AE 011-2024**

Disponible en formato PDF

• **AE 010-2024**


Disponible en formato PDF

• **AE 009-2024**

Disponible en formato PDF

Atender incidencias de boletas-cafae, sistema de haberes.

Se actualizó el DNI de los empleados a la fecha de emisión actual para que puedan ingresar al sistema de boleta cafae.



PERÚ

Ministerio de Salud

Dirección de Redes Integradas de Salud Lima Sur

INCENTIVOS LABORALES

DNI del Trabajador: *

Fecha de emisión DNI: *


CUI (Código de verificación): *

dd/mm/aaaa

Fecha: *

31 de Enero de 2021

☐ No soy un robot


reCAPTCHA
Privacidad - Condiciones

Consultar

Copyright © 2024 , Todos los derechos reservados - DIRIS LIMA SUR.❤

Se creó un script para automatizar la actualización de los DNI

```

1 import csv
2 import requests
3 import pandas as pd
4
5 # URL base de la API
6 base_url = "http://192.168.1.4/api/web-service/person-complete/"
7
8 # Encabezados que necesitas enviar con la solicitud
9 headers = {
10     'Content-Type': 'application/json',
11     'Authorization': 'token eed63ab26117dacf4986f37ca1e61c4ccafc2aea'
12 }
13
14 # Función para hacer la petición GET y obtener la respuesta
15 def get_person_data(dni):
16     try:
17         # Realizamos la petición GET con los encabezados
18         response = requests.get(f"{base_url}{dni}", headers=headers)
19
20         # Verificamos si la respuesta fue exitosa
21         if response.status_code == 200:
22             return response.json() # Retornamos la respuesta en formato JSON
23         else:
24             print(f"Error: {response.status_code} al obtener datos para el DNI {dni}")
25             return None
26     except Exception as e:
27         print(f"Error al hacer la solicitud para el DNI {dni}: {e}")
28         return None
29
30 # Leer el archivo CSV con los DNIs (suponiendo que la primera columna es el DNI)
31 dni_list = []
32 with open('dni_list.csv', newline='', encoding='utf-8') as csvfile:
33     reader = csv.reader(csvfile)
34     next(reader) # Si el archivo tiene una fila de encabezado, descómlala
35     for row in reader:
36         if row: # Aseguramos que no esté vacía la fila
37             dni_list.append(row[0]) # Asumimos que el DNI está en la primera columna
38
39 # Verifica que estamos leyendo todos los DNIs
40 print("DNIs leídos desde el archivo CSV:", dni_list)
41
42 # Lista para almacenar las respuestas
43 responses = []
44
45 # Procesamos cada DNI
46 for dni in dni_list:
47     print(f"Consultando datos para el DNI: {dni}")
48     response_data = get_person_data(dni)
49
50     if response_data:
51         responses.append(response_data)
52
53 # Opcional: Guardar las respuestas en un archivo CSV
54 df = pd.DataFrame(responses)
55 df.to_csv("respuestas_api.csv", index=False)
56
57 # Imprimir el número total de consultas realizadas
58 print(f"Consultas finalizadas. Se realizaron {len(responses)} consultas.")

```

Administrar el hardware de los sistemas comprobando que el componente físico funcione correctamente.

Se administró y se comprobó que funcione correctamente los siguientes servidores:

N°	ESTADO	NOMBRE	IP
1	OPERATIVO	SERV_APLI	192.168.1.8
1.1	OPERATIVO	SERV-API	38.43.129.230/192.168.1.4
1.2	OPERATIVO	SER-APP-SGD	192.168.1.12:8181
1.3	OPERATIVO	SERV-DB-SGD	192.168.1.24:5432
1.4	OPERATIVO	SERV-DB-SGD [Prueba]	192.168.1.22
1.5	OPERATIVO	SERV-LAB	192.168.4.220
1.6	OPERATIVO	SERV-APP-MYSQL8	192.168.1.29
1.7	OPERATIVO	SERVER_UBUNTU_APPS	192.168.1.101
1.6	INOPERATIVO	SERV-GPO-II	192.168.2.11
1.7	INOPERATIVO	SERV-INDICADORES	192.168.1.103
1.8	INOPERATIVO	SERV-JAVA-APPS	192.168.2.7
1.10	INOPERATIVO	SERV-SISMED	192.168.1.102
2	OPERATIVO	SERV-VIRTUAL	192.168.1.28
2.1	OPERATIVO	SERV-SISMED	192.168.1.102
2.2	OPERATIVO	SERV-JAVA-APPS	192.168.2.7
3	OPERATIVO	SERV-APP-DIRISLS	192.168.1.26
4	OPERATIVO	S-TRAMITE	192.168.1.30
4.1	OPERATIVO	SERV-GLPI-HELPDESK	192.168.2.6
4.3	OPERATIVO	SERV-NEXTCLOUD	192.168.1.123
4.4	VERIFICAR	SERV-PROJECT	192.168.1.9
5	OPERATIVO	SERV-DOM	129.21.20.4
6	OPERATIVO	SRVA60B44E04	192.168.1.21
7	OPERATIVO	SERV-FACT-ELECT-FARM	38.43.129.235/192.168.1.10
8	OPERATIVO	SERV-DEV-DOCKERS	192.168.1.124

Monitorear y asegurar el buen funcionamiento de las aplicaciones de la DIRIS Lima Sur.

Se aseguró el funcionamiento de aplicaciones:

- HABERES (funcionando correctamente)
- CAFAE (funcionando correctamente)
- CERTIFICACIONES (funcionando correctamente)
- VANCAN (funcionando correctamente)
- RECLAMACIONES (funcionando correctamente)
- INTRANET DE LA DIRIS LIMA SUR (funcionando correctamente)

Proceso de carga de boletas CAFAE (Mensual)

1. Exportar archivos como “xlsx”

Identificar los siguientes archivos y convertirlos a xlsx utilizando excel.

1. PRCONCEP.BAS
2. PRDSTOTE.BAS
3. PREXCEL3.BAS
4. PRMPAGOS.BAS
5. PRTEMPLATE.BAS

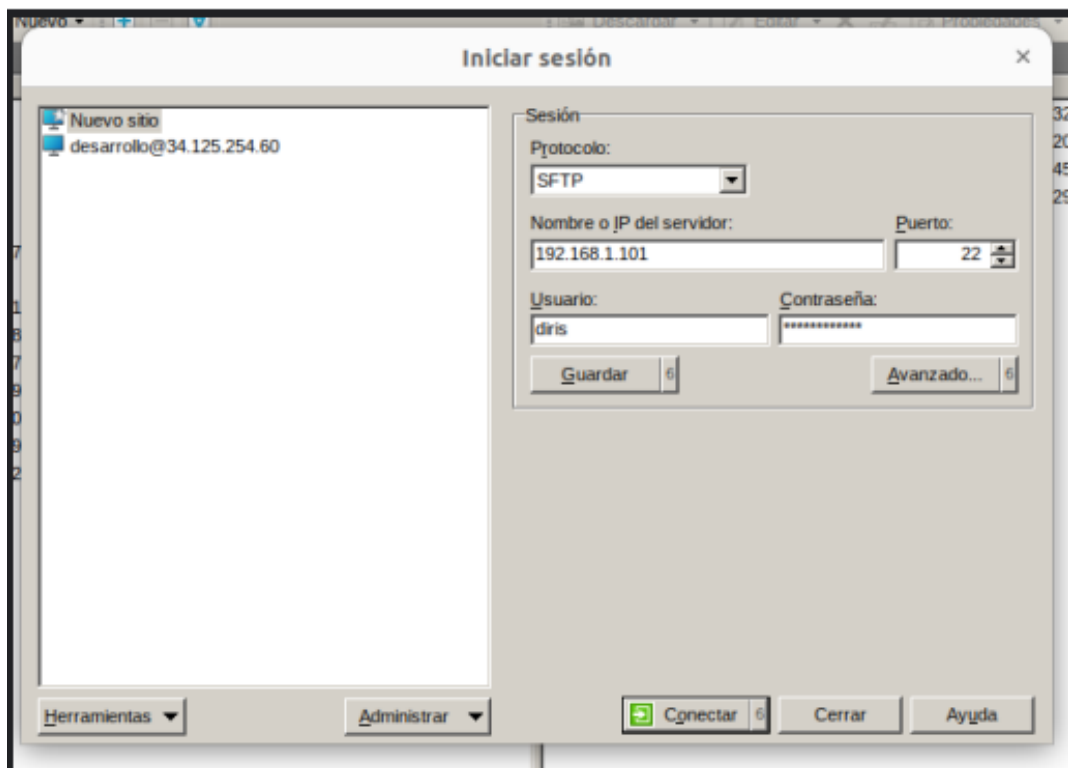
2. Ingresar al servidor

Este paso se repite cada vez que se desea ingresar al servidor

IP: 192.168.1.101

PAS: XXXXX

- CON SSH O WINSOCP (Para copiar archivos)



- CON SSH (Para ejecutar los comandos; utilizar programas como: PowerShell, Git Bash, Putty etc.)

```
ssh diris@192.168.1.101
```

- Activar el entorno virtual

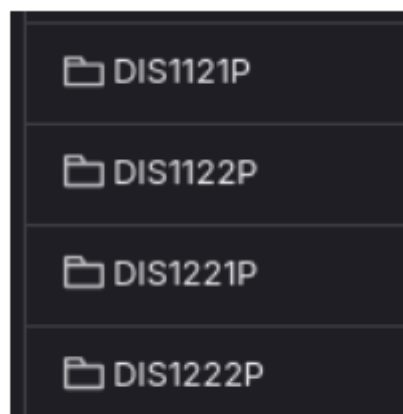
```
source venvs/diris_cafae/bin/activate
```

- Dirigirse al proyecto

```
cd /home/diris/apps/diris_cafae
```

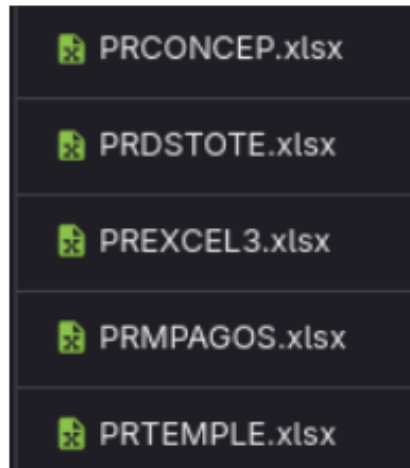
3. Copiar los archivos en el proyecto

En la ruta dentro del proyecto: **diris_cafae/apps/load_data**, crear la carpeta con el periodo respectivo:



DIS0823P -> DIS(MES)(AÑO)P

Dentro de la carpeta copiar los siguientes archivos convertidos anteriormente:



4. Ingresamos a DJANGO SHELL con el entorno virtual activado

En la raíz del proyecto ejecutamos el siguiente comando:

```
python manage.py shell --settings=main.settings.prod
```

5. Importamos las dependencias

Ejecutamos el siguiente comando:

```
exec(open('diris_shell.py').read())
```

6. Cargamos los archivos con la información

Ejecutamos los siguientes comandos con el período como parámetro de entrada:

- #DATOS DEL TRABAJADOR (GENERAL) INICIO

```
run_load_trabajador('0524') -> MES+AÑO  
run_load_fecha('0524') -> MES+AÑO  
run_load_planilla('0524') -> MES+AÑO  
run_load_ingreso('0524') -> MES+AÑO  
run_load_concepto('0524') -> MES+AÑO  
run_load_descuento('0524') -> MES+AÑO  
run_load_judicial('0524') -> MES+AÑO
```
- #HACER CORRER AL FINAL

```
run_load_total('0524') -> MES+AÑO}
```

```
exit()
```

7. Actualizamos la fecha que hemos subido para que se muestre en la aplicación

- Nos dirigimos a la carpeta forms y abrimos consulta.py

```
cd /home/iris/apps/iris_cafae/apps/index/forms
```

- Ejecutamos el comando para editar

```
nano consulta.py
```

8. Editamos consulta.py

```
GNU nano 6.2 consulta.py
import calendar
import datetime

from django import forms
from django.forms import CharField, DateField, IntegerField

PERIODOS = []

# PERIODO: 2021
for i in range(12):
    last_day = calendar.monthrange(2021, i + 1)[1]
    fecha = datetime.date(2021, i + 1, last_day)
    PERIODOS.append(tuple([fecha, fecha]))

# PERIODO: 2022
for i in range(12):
    last_day = calendar.monthrange(2022, i + 1)[1]
    fecha = datetime.date(2022, i + 1, last_day)
    PERIODOS.append(tuple([fecha, fecha]))

# PERIODO: 2023 julio
for i in range(8):
    last_day = calendar.monthrange(2023, i + 1)[1]
    fecha = datetime.date(2023, i + 1, last_day)
    PERIODOS.append(tuple([fecha, fecha]))

class ConsultaForm(forms.Form):
    dni = CharField(label='DNI del Trabajador', required=True, max_length=8, min_length=8)
    fecha_emision = DateField(label='Fecha de emisión DNI', required=True, widget=forms.DateInput(attrs={'type': 'date'}))
    cui = IntegerField(label='CUI (Código de verificación)', required=True)
    fecha = forms.DateField(label='Fecha', widget=forms.Select(choices=PERIODOS))

Ayuda  Guardar  Buscar  Cortar  Ejecutar  Ubicación  M-U  Deshacer  M-A  Poner marca  M-I  A llave
Salir  Leer fich.  Reemplazar  Pegar  Justificar  Ir a línea  M-E  Rehacer  M-C  Copiar  AQ  Buscar atrás
```

- Cambiamos el range según el mes que hemos actualizado y luego Ctrl+O y enter para guardar los cambios en el archivo

9. Ejecutamos el comando para reiniciar el servidor y aplicar los cambios

Ejecutamos el siguiente comando

```
cd /home/iris/apps/iris_cafae
sudo supervisorctl restart iris_cafae
```

Proceso de carga de HABERES (Mensual y Anual)

Ingresar al servidor

Este paso se repite cada vez que se desea ingresar al servidor

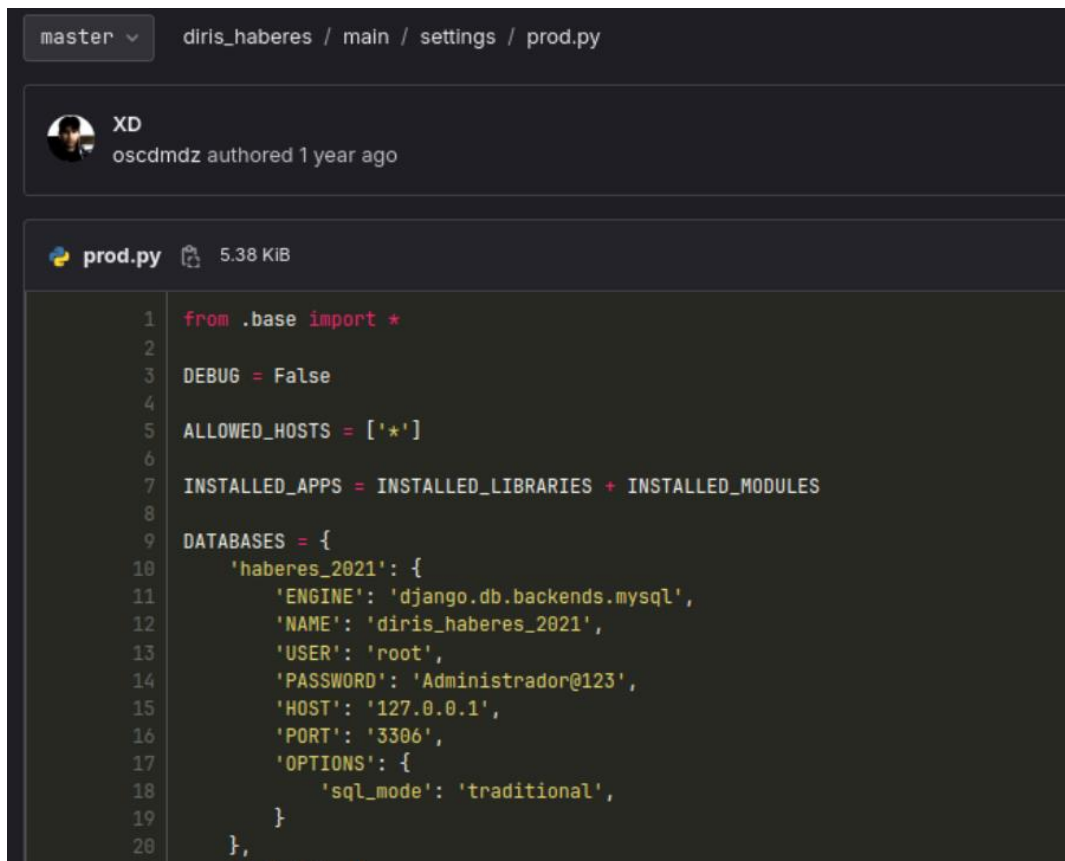
```
ssh diris@192.168.1.101
```

Es necesario crear una Base de Datos y realizar las migraciones por cada año que se requiere migrar. (SOLO UNA VEZ)

- Crear la base de datos






```
CREATE DATABASE IF NOT EXISTS diris_haberes_2024 DEFAULT  
CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
```

- Agregar configuración de la base de datos en el proyecto (Archivo prod.py)



```
master ▾  diris_haberes / main / settings / prod.py  
  
XD  
oscdmdz authored 1 year ago  
  
prod.py 5.38 KiB  
  
1  from .base import *  
2  
3  DEBUG = False  
4  
5  ALLOWED_HOSTS = ['*']  
6  
7  INSTALLED_APPS = INSTALLED_LIBRARIES + INSTALLED_MODULES  
8  
9  DATABASES = {  
10     'haberes_2021': {  
11         'ENGINE': 'django.db.backends.mysql',  
12         'NAME': 'diris_haberes_2021',  
13         'USER': 'root',  
14         'PASSWORD': 'Administrador@123',  
15         'HOST': '127.0.0.1',  
16         'PORT': '3306',  
17         'OPTIONS': {  
18             'sql_mode': 'traditional',  
19         }  
20     },
```

- Copiar los siguientes archivos a la carpeta load, e internamente deberá estar distribuido por año (crear carpeta con el número del año "Ejem. 2024") y mes (crear carpeta con el número de mes "Ejem. 01")

master ▾		diris_haberes / apps / load_data / 2021 / 01 /		+ ▾
		XD		oscdmdz authored 1 year ago
Name		Last commit		
..				
	PLMOVMAE.xlsx	XD		
	PLTABCAR.xlsx	XD		
	PLTABCON.xlsx	XD		
	PLTABEST.xlsx	XD		

Activar el entorno virtual

```
source vens/diris_haberes/bin/activate
```

- Dirigirse al proyecto

```
cd /home/diris/apps/diris_haberes
```

-----ESTO SE HACE SI ES UNA BASE DE DATOS RECIÉN CREADA Y NUNCA SE LE LLENÓ CON DATOS-----

Migración

```
python manage.py migrate --settings=main.settings.dev --database=haberes_2021
```

Creación de super usuario

```
python manage.py createsuperuser --settings=main.settings.prod --database=haberes_2021
```

1. Cargar información de ESTABLECIMIENTOS

Nombre del archivo: PLTABEST.xlsx

Esta información se debe cargar **Anualmente**

- ```
python manage.py shell --settings=main.settings.prod
```
- ```
exec(open('diris_shell.py').read())
```
- ```
load_establecimiento(2021)
```

## 2. Carga información de CARGOS

Nombre del archivo: PLTABCAR.xlsx

Esta información se debe cargar **Anualmente**

- ```
python manage.py shell --settings=main.settings.prod
```
- ```
exec(open('diris_shell.py').read())
```
- ```
load_cargos(2021)
```

3. Cargar información de CONCEPTOS

Nombre de archivo: PLTABCON.xlsx

Esta información se debe cargar **mensualmente**.

- ```
python manage.py shell --settings=main.settings.prod
```
- ```
exec(open('diris_shell.py').read())
```
- ```
load_concepto(5,2024)
```

#### 4. Carga información de CONSTANCIAS

Nombre del archivo: PLMOVMAE.xlsx

Esta información se debe cargar **mensualmente**.

- `python manage.py shell --settings=main.settings.prod`
- `exec(open('diris_shell.py').read())`
- `load_constancia(5,2024)`