Guía de instalación

Función Puente de doctores IQVIA.

Versión: 1.1.8

[**1. INTRODUCCIÓN 3**](#_ftzeoufl7710)

[DESCRIPCIÓN 3](#_ute7lh2htd2o)

[REQUISITOS 3](#_wr0a0af1k6nc)

[**2. REQUERIMIENTOS DEL SISTEMA Y ENTORNO 3**](#_7i3jc74voxy0)

[DEPENDENCIAS NECESARIAS 3](#_q60tnzbrnu1q)

[PERMISOS Y CONFIGURACIONES 4](#_j32cjaghvort)

[**3. PASOS DE INSTALACIÓN 4**](#_b333ol1p9tqp)

[DESCARGA DEL CÓDIGO 4](#_oad01iilxkvg)

[CONFIGURACIÓN DEL PROYECTO 4](#_fpxfofibx303)

[**4. PRUEBA DE FUNCIONALIDAD 5**](#_b97mufab7bdv)

[**5. ACTUALIZACIÓN DE DEPENDENCIAS 5**](#_pukg9jbswfr2)

# **INTRODUCCIÓN**

Esta guía de instalación explicará el cómo se instalará la función de puente de doctores para la empresa IQVIA **IMPORTANTE esta es una función orientada a un nivel de backend, por lo que la vista de usuario usada a lo largo de este documento fue desarrollada por el grupo de trabajo por lo cual puede diferir del actual sistema web utilizada por IQVIA Inc, quien además de esto tiene todos los derechos reservados de la función realizada.**

## DESCRIPCIÓN

La función “Puente de doctores” corresponde a la generación de un reporte de doctores, el cual se realiza contrastando la información provista por la base de datos de la empresa y un documento procesado por un departamento de la empresa en base a información actualizada de nuevos doctores y parte de los anteriores. Este reporte mostrará un porcentaje de similitud entre los registros de la base de datos y los provistos en el último documento en conjunto con unas alertas y recomendación para la persona a cargo del proceso con el fin de agilizar procesos en la empresa y realizar la trata de casos especiales de forma más rápida.

## REQUISITOS

Esta función hace uso del lenguaje de programación Python en su versión 3.11.3, además de hacer uso del framework Django en su versión 3.1.14 y se utilizó un motor de base de datos Oracle versión 18XE. A la persona a cargo de llevar a cabo la incorporación de la función se le recomienda tener conocimientos en la programación de Front End, Back End, Python y en la integración de software en caso de ser necesario para además incorporarlo a nivel de servidor.

# **REQUERIMIENTOS DEL SISTEMA Y ENTORNO**

## DEPENDENCIAS NECESARIAS

Para el correcto funcionamiento del puente de doctores, serán necesarias las siguientes librerías:

* Fuzzywuzzy.
  + fuzz
  + process
* Python-Levenshtein openpyxl.
  + utils.dataframe: dataframe\_to\_rows.
  + styles: PatternFill, Font, Alignment.
  + Workbook.
  + drawing.image: Image.
* Matplotlib.pyplot.
* io.BystesIO
* Django

## PERMISOS Y CONFIGURACIONES

La persona a cargo de realizar el proceso de puente de doctores debe de tener el rol X para poder llevar a cabo el proceso, esto debido a que solo la gente asignada a su departamento podrá realizar el informe. Si bien es posible que el rol de administrador también podría realizarlo, el departamento correspondiente tiene la obligación de realizar dicho reporte, la revisión y el informe de anormalidades pertinente a los correspondientes departamentos.

# **PASOS DE INSTALACIÓN**

## DESCARGA DEL CÓDIGO

El código correspondiente de la función será provisto por el equipo de trabajo a la empresa, en caso de ser necesario el descargarlo, este puede ser descargado desde el repositorio de GitHub BRAVOBRAYAN\_CORRALESVICENTE\_LAZCANOMARTIN (Enlace directo al repositorio: ), en dicha carpeta se puede encontrar el código de la función para ser utilizado en una carpeta aparte, también se verá un código en el cual se encuentra la función orientada para un ambiente web.

## CONFIGURACIÓN DEL PROYECTO

Para la instalación de las librerías necesarias se debe utilizar el comando pip en una terminal, ya sea para el ambiente web o de servidor. Algunas de las librerías están incluidas en paquetes más grandes.

A continuación el código a utilizar:

*pip install fuzzywuzzy python-levenshtein openpyxl matplotlib django*

**fuzzywuzzy**: Proporciona funciones necesarias para realizar comparaciones de similitud de texto.

**openpyxl**: Permite trabajar con archivos Excel. Contiene submódulos y clases.

**Workbook**: Para crear y manipular libros de Excel.

**matplotlib.pyplot**: Usado para generar gráficos. Importa la librería después de la instalación.

**io.BytesIO**: Es un módulo estándar de Python, por lo que no requiere instalación.

A continuación se muestra el código necesario para configurar las librerías:

*from openpyxl.utils.dataframe import dataframe\_to\_rows*

*from openpyxl.styles import PatternFill, Font, Alignment*

*from openpyxl import Workbook*

*from openpyxl.drawing.image import Image*

*import matplotlib.pyplot as plt*

*from io import BytesIO*

# **PRUEBA DE FUNCIONALIDAD**

La función como tal puede verificar su funcionalidad en un entorno de servidor de forma fácil con Visual Studio, además de su conectividad a la base de datos. Pero por parte de un entorno web la mejor forma de probar si funciona será el realizar el proceso de subir el documento “.csv” que se quiere contrastar y si el entorno comienza la descarga del documento, entonces se ha instalado todo de buena forma. **Importante: Si el entorno no genera la descarga eso significa que la función no se puede ejecutar por alguna razón.**

# **ACTUALIZACIÓN DE DEPENDENCIAS**

En caso de que en algún momento sea necesario el actualizar alguna de las librerías utilizadas se puede realizar de manera individual, o actualizando todas las librerías al mismo tiempo, a continuación se dejan las líneas de código correspondientes:

**Individual:**

*pip install --upgrade <nombre\_libreria>*

**Todas las librerías desactualizadas:**

*pip install --upgrade $(pip list --outdated | awk '{print $1}' | tail -n +3).*