| **SISTEMA DE SOPORTE Y ASISTENCIA A USUARIOS** |
| --- |

MANUAL DE INSTALACIÓN

**TEAMFIT**

V1.0

| **Créditos** | | |
| --- | --- | --- |
| Profesores a cargo | * Francisco Savignone * Manuel Tello | |
| Integrantes | * Abraham Daque * Thomas Gonzalez * Diego Rivera * Benjamin Ramirez * Jeremías Canto * Javiera Marchant * Felipe Diaz * Marcelo Criado | |

**TABLA DE CONTENIDO.**

[**1. INTRODUCCIÓN 3**](#_x0hj9g5byt99)

[1.1. OBJETIVO 3](#_2h2p0r1w45ux)

[1.2. REQUERIMIENTOS PARA UTILIZAR EL PROGRAMA 3](#_7u06asawe4h2)

[1.3. REQUERIMIENTOS PREVIOS PARA INSTALAR EL PROGRAMA 3](#_6m6o1ht1pstr)

[**2. PASOS PARA LA INSTALACIÓN DEL PROGRAMA 4**](#_5w7de2b1buo8)

[2.1.1. DESCARGAR Y DESCOMPRIMIR EL ARCHIVO 4](#_ovcn9e4eaff6)

[2.1.2. CREAR UN AMBIENTE VIRTUAL 4](#_ti00fbx57gih)

[2.1.3. ABRIR EL PROYECTO 5](#_tksl4wfw2lgh)

[2.1.4. INSTALAR LAS DEPENDENCIAS 5](#_8mhcpb2wej7t)

[**3. CONFIGURACIÓN DE LA BASE DE DATOS 6**](#_18h7m2z238pa)

[3.1.1. CREAR UN USUARIO ADMINISTRADOR EN PGADMIN 4 6](#_khdv6j6usq9x)

[3.1.2. CREAR LA BASE DE DATOS 6](#_sosng3u6ptlk)

[3.1.3. RESTAURAR LA BASE DE DATOS DESDE EL RESPALDO 6](#_d7g8cbrqhmr1)

[**4. CONFIGURACIÓN DE LA APLICACIÓN 7**](#_n7ljms1x35bx)

[4.1.1. CONFIGURAR “SETTINGS.PY” PARA LA CONEXIÓN A LA BASE DE DATOS 7](#_adylnkj5raez)

[4.1.2. VERIFICAR EL ENTORNO DE TRABAJO 7](#_bwqski6o2fpa)

[4.1.3. REALIZAR LAS MIGRACIONES DE LA BASE DE DATOS 7](#_id3dzw35n0mo)

[4.1.4. APLICAR LAS MIGRACIONES 8](#_ch4p8mklmo2k)

[4.1.5. INICIAR EL SERVIDOR DE DESARROLLO 8](#_qk8zersgtjf)

[**5. ACCESO AL SISTEMA Y CREACIÓN DE USUARIO 8**](#_sfu6xujhb00k)

[5.1.1. INICIAR SESIÓN EN LA APLICACIÓN 8](#_1ulvwmo3x1w1)

[5.1.2. VERIFICACIÓN FINAL 9](#_rd2bv0yfcqq6)

# 

# **INTRODUCCIÓN**

## **1.1. OBJETIVO**

El siguiente manual de instalación tiene como objetivo guiar a los usuarios encargados de implementar y levantar el sistema de Teamfit, mediante una guía básica y fácil de comprender, desde la creación del ambiente virtual hasta la puesta en marcha de la aplicación web.

### **1.2. REQUERIMIENTOS PARA UTILIZAR EL PROGRAMA**

| CPU | Intel 3 o superior. |
| --- | --- |
| Memoria RAM | Mínimo 4GB. |
| Sistema Operativo | Windows 8 o superior. |
| Navegador | Google Chrome. |
| Resolución de pantalla mínima | Mínimo 1280 \* 720. |
| Conexión a internet | Obligatoria. |

### **1.3. REQUERIMIENTOS PREVIOS PARA INSTALAR EL PROGRAMA**

| Lenguaje | Python 3.0 Obligatorio |
| --- | --- |
| Motor de base de datos | Postgres Obligatorio  PGAdmin 4 Obligatorio |
| Herramienta de administración de base de datos | DBeaver Opcional |

| \***NOTA:** Este es un proyecto funcionando en Django, por lo que las instrucciones para permitir el funcionamiento del programa también pueden buscarse por otros medios |
| --- |

# **PASOS PARA LA INSTALACIÓN DEL PROGRAMA**

## **2.1.1. DESCARGAR Y DESCOMPRIMIR EL ARCHIVO**

* Descarga los archivos del proyecto desde la rama “Main” en el repositorio de GitHub “Teamfit”.
* Descomprimir los archivos en una carpeta de tu elección.

## **2.1.2. CREAR UN AMBIENTE VIRTUAL**

Para instalar correctamente el programa “Teamfit” es necesario crear un ambiente virtual para alojar el proyecto, esto se puede realizar mediante el uso de comandos en el “**Command Prompt (CMD)**”, los cuales también servirán para crear la carpeta(?.

| \***NOTA:** El proceso puede variar entre Windows y Linux., para este caso sólo se considerará el proceso en Windows debido a que el programa se desarrolló mediante este sistema operativo. |
| --- |

**PASOS PARA CREAR UN AMBIENTE VIRTUAL PARA ALOJAR EL PROGRAMA:**

* Abre el **Command Prompt** (CMD).
* Crea una carpeta para tu ambiente virtual. Por ejemplo en este caso se utilizará “**carpeta\_proyecto**”

| mkdir **carpeta\_proyecto**  cd **carpeta\_proyecto** |
| --- |

* Para crear el ambiente virtual debe ocupar el siguiente comando:

| python -m venv **nombre\_carpeta** |
| --- |

| \***NOTA:** Los nombres utilizados para la carpeta y el entorno virtual son simplemente para uso de ejemplo, el usuario es libre de colocar cualquier nombre de su agrado a estos apartados. |
| --- |

## **2.1.3. ABRIR EL PROYECTO**

Para abrir el proyecto se debe utilizar un editor de código, para este caso se recomienda utilizar Visual Studio Code, debido a que el programa fue desarrollado en este ambiente de trabajo.

**PASOS PARA ABRIR EL PROYECTO:**

* Abrir Visual Studio Code y carga la carpeta donde se descomprimio los archivos del proyecto.
* Antes de continuar se debe verificar que las carpetas principales como core, teamfit, requirements.txt, entre otros, se encuentren en la carpeta.
* Abrir una nueva terminal dentro de Visual Studio Code (**Terminal > New Terminal**).
* Ahora debe activar el ambiente virtual, creado anteriormente, ejecutando el siguiente comando en la terminal creada en el paso anterior:

| **nombre\_carpeta**/Scripts/activate |
| --- |

Esto activará el entorno virtual para el proyecto, Una vez realizado este proceso debe de regresar a la carpeta principal del proyecto.

## **2.1.4. INSTALAR LAS DEPENDENCIAS**

Con el ambiente virtual creado y activado, el siguiente paso es el instala las librerías necesarias para que el programa se ejecute correctamente, en la consola de comandos de Visual Studio Code debemos usar el siguiente comando:

| pip install -r requirements.txt |
| --- |

# **CONFIGURACIÓN DE LA BASE DE DATOS**

## **3.1.1. CREAR UN USUARIO ADMINISTRADOR EN PGADMIN 4**

* Abre PGAdmin 4.
* Crea un usuario con permisos de administrador llamado “**azure\_pg\_admin**” (sin las comillas).

## **3.1.2. CREAR LA BASE DE DATOS**

* En PGAdmin 4, haz clic derecho sobre el icono del servidor servidor y selecciona **Create > Database**.
* Nombra la base de datos como “**teamfit**” y guarda los cambios.

| \***NOTA**: Este paso tiene dos formas de ser creado, la primera es utilizando la interfaz gráfica y la segunda mediante comandos. Se recomienda crearla utilizando la interfaz gráfica. |
| --- |

## **3.1.3. RESTAURAR LA BASE DE DATOS DESDE EL RESPALDO**

* Para realizar el respaldo debemos hacer clic derecho sobre la base de datos “**teamfit**” y seleccionar la opción “**Restore**”.
* Al seleccionar la opción, se abrirá la ventana de restauración, aquí debemos seleccionar el archivo de respaldo o script llamado “**Respaldo DB - PGAdmin.sql**” el cual se encuentra en el repositorio de GitHub.
* Si se han seguido los pasos correctamente, la base de datos debería respaldarse correctamente.

# **CONFIGURACIÓN DE LA APLICACIÓN**

## **4.1.1. CONFIGURAR “SETTINGS.PY” PARA LA CONEXIÓN A LA BASE DE DATOS**

* Abre el archivo “**settings.py**” del proyecto, que se encuentra en la carpeta principal.
* En este archivo debemos configurar las credenciales de PostgreSQL (usuario, contraseña, puerto, nombre de la base de datos) en la sección “**DATABASES”**.

Ejemplo de la configuracion en “setting.py”

| DATABASES = {  'default': {  'ENGINE': 'django.db.backends.postgresql',  'NAME': 'teamfit',  'USER': 'azure\_pg\_admin',  'PASSWORD': 'tu\_contraseña',  'HOST': 'localhost',  'PORT': '5432',  }  } |
| --- |

## **4.1.2. VERIFICAR EL ENTORNO DE TRABAJO**

Antes de continuar debemos asegurarnos de que el “**Command Prompt**” que estamos utilizando para la creación del ambiente esté posicionado en el mismo nivel que el proyecto en Visual Studio.

## **4.1.3. REALIZAR LAS MIGRACIONES DE LA BASE DE DATOS**

Ahora necesitamos realizar la migraciones utilizando el siguiente comando en la consola de windows:

| python manage.py makemigrations |
| --- |

## 

## **4.1.4. APLICAR LAS MIGRACIONES**

Para aplicar las migraciones sin reemplazar las tablas existentes, debemos ejecutar el siguiente comando en la consola de windows:

| python manage.py migrate --fake |
| --- |

Mediante este paso nos aseguraremos que las migraciones se apliquen correctamente sin sobrescribir la base de datos existente. Se recomienda realizar esto para evitar problemas con el Script de base de datos.

## **4.1.5. INICIAR EL SERVIDOR DE DESARROLLO**

Para inicia el servidor de desarrollo de Django debemos ingresar el siguiente comando:

| python manage.py runserver |
| --- |

Esto levantará la aplicación en **http://127.0.0.1:8000/**. Se recomienda utilizar Google Chrome como navegador para acceder a la aplicación, ya que es el navegador en el que se ha probado.

# **ACCESO AL SISTEMA Y CREACIÓN DE USUARIO**

## **5.1.1. INICIAR SESIÓN EN LA APLICACIÓN**

* Si todos los pasos se han seguido correctamente el sistema estará disponible en la URL **http://127.0.0.1:8000/**.
* Para ingresar al sistema se deben usar las siguientes credenciales predeterminadas para iniciar sesión:
  + **Usuario**: Admin
  + **Contraseña**: Admin@123

| \***NOTA**: Si no puedes iniciar sesión o el sistema muestra errores, puedes crear un nuevo usuario administrador con el siguiente comando:   | python manage.py createsuperuser | | --- | |
| --- | --- |

* Si el nuevo usuario necesita un perfil de usuario, se deberá crear un perfil en la base de datos utilizando una **Query SQL** debido a que el usuario fue creado a través de líneas de comandos.

## **5.1.2. VERIFICACIÓN FINAL**

Una vez que se haya iniciado sesión correctamente, el sistema debería funcionar sin problemas. Ahora se puede comenzar a utilizar la aplicación según lo especificado en el **manual de usuario**.

# **Solución de problemas comunes**

* **Error de conexión a la base de datos**: Verifica que las credenciales en **settings.py** sean correctas y que PostgreSQL esté ejecutándose.
* **Problemas al realizar migraciones**: Si las migraciones fallan, asegúrate de que la base de datos esté configurada correctamente y que no existan conflictos con las tablas previas.