



El futuro digital  
es de todos

MinTIC

```
te( "name" );  
"type" );
```

```
if ( type == "sprite" )  
  
std::string item_name = item->Attribute( "name" );  
std::string spritename = item->Attribute( "spritename" );  
float x = boost::lexical_cast<float>( item->Attribute( "x" ) );  
float y = boost::lexical_cast<float>( item->Attribute( "y" ) );  
float offset = boost::lexical_cast<float>( item->Attribute( "offset" ) );  
  
SpriteDescList::iterator sp = sprite_descs.begin();  
for( ; sp != sprite_descs.end(); ++sp )  
    if ( sp->name_ == spritename )  
        break;
```

## Ciclo 3:

Desarrollo de Software



**Misión  
TIC2022**

VERSIÓN 1.0

Unidad de educación  
continua y permanente  
Facultad de Ingeniería



Unidad Camilo Torres  
Calle 44 # 45-67  
Bloque B5 piso 1



(57) + 316 5000  
uec\_ibog@unaleduco

# Componente de Base de Datos

## Actividad Práctica

El objetivo del Ciclo 3 es la construcción de un sistema de software que permita responder a los requerimientos básicos de registro y autenticación en un banco, para esto el sistema estará compuesto por 3 capas (ver imagen 1), una capa de datos llamada *bank\_db*, que se encargará del almacenamiento de los datos, una capa de back-end llamada *bank\_be*, que se encargará del procesamiento y gestión de la información y una capa de front-end llamada *bank\_fe*, la cual será la interfaz con la que los usuarios interactuarán con el sistema.

A lo largo del ciclo se contextualizará, diseñará, desarrollará, probará y desplegará cada una de las capas, siguiendo la estrategia *bottom-up*, la cual indica que primero se deben desarrollar los componentes de la capa inferior (en este caso *bank\_db*) y luego avanzar con los componentes de la siguiente capa (en este caso *bank\_be*), hasta llegar a los componentes de la capa superior (en este caso *bank\_fe*)

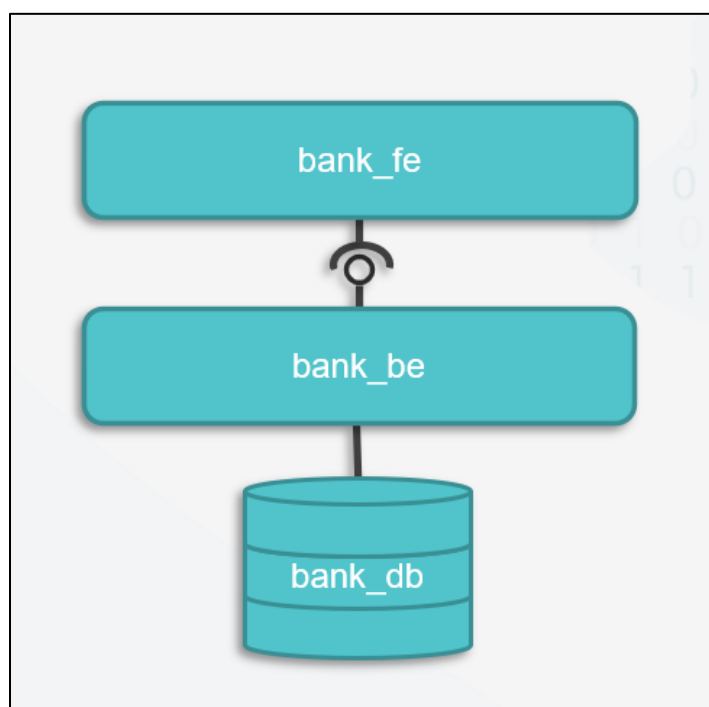


Imagen 1: Capas del sistema.

Teniendo en cuenta la estrategia bottom-up explicada anteriormente, el primer componente a construir es el correspondiente a la capa de datos, llamado *bank\_db*. Sin embargo, antes de especificar las características técnicas de esta capa se debe entender qué es y cuál es la función de una base de datos, así como los conceptos y herramientas importante alrededor de esta.

Una base de datos es una colección de información organizada, usualmente almacenada electrónicamente en un sistema computacional, su objetivo principal es brindar persistencia a información relevante en un sistema. La base de datos es controlada por un sistema gestor de base de datos (Database Management System o *DBMS*, por sus siglas en inglés), el cual funciona como un cliente entre el usuario final y la base de

datos, y se encarga de administrar, entre otras cosas, los datos, el motor de la base de datos y el esquema de la base de datos, para facilitar la organización y la manipulación de los datos. Gracias a los **DBMS** es posible acceder, administrar, modificar, actualizar, controlar y organizar fácilmente los datos. Sin embargo, existen muchos tipos distintos de **DBMS** que se encargan de estas operaciones a su manera, siguiendo distintos paradigmas y esquemas. Así, el **DBMS** que utiliza una base de datos define su tipo y, por ende, el motor que utiliza, el esquema que sigue para almacenar los datos, el lenguaje de dominio específico que se debe utilizar para dar instrucciones a la base de datos, entre otras cosas.

Existen dos tipos de bases de datos principales, relacionales y no relacionales, las bases de datos relacionales son aquellas cuyo **DBMS** sigue un paradigma relacional. Es decir, que organizan los datos en una serie de tablas con filas y columnas, donde cada columna es un atributo de la tabla, y cada fila es un registro. El nombre relacional se debe a que pueden existir relaciones entre atributos de distintas tablas. Una base de datos relacional se puede comparar con una tabla de Excel, sin embargo, esta es mucho más compleja.

Tabla Usuarios					
Atributos	last_login	is_superuser	id	username	password
Registros	NULL	FALSE	1	Admin	pbkdf2_sha256\$2...
	NULL	FALSE	2	Usuario	pbkdf2_sha256\$2...

Imagen 2: Tabla Relacional

En una base de datos relacional, las tablas se relacionan entre sí por medio de llaves primarias y foráneas. Las llaves primarias son aquel atributo único que toda tabla tiene, es decir, son el atributo de una tabla en el cual 2 filas no pueden tener el mismo valor. Generalmente la llave primaria de una tabla es el identificador (id) de cada registro, y se utiliza para encontrar un registro específico en la tabla. Por otro lado, las llaves foráneas son las llaves primarias de una tabla almacenadas como atributos de otras tablas, esto tiene como propósito formar relaciones entre las tablas. Por ejemplo, para indicar que una factura tiene un cliente asociado, en la tabla Facturas se almacena la llave primaria (id) de dicho cliente.

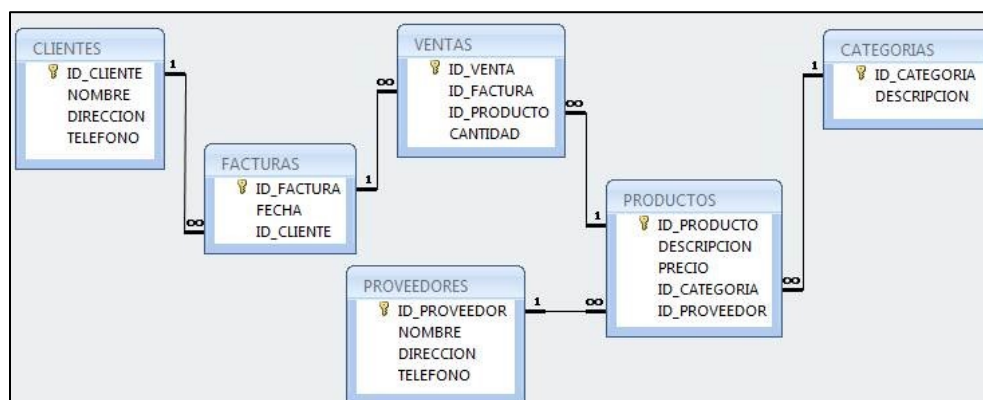


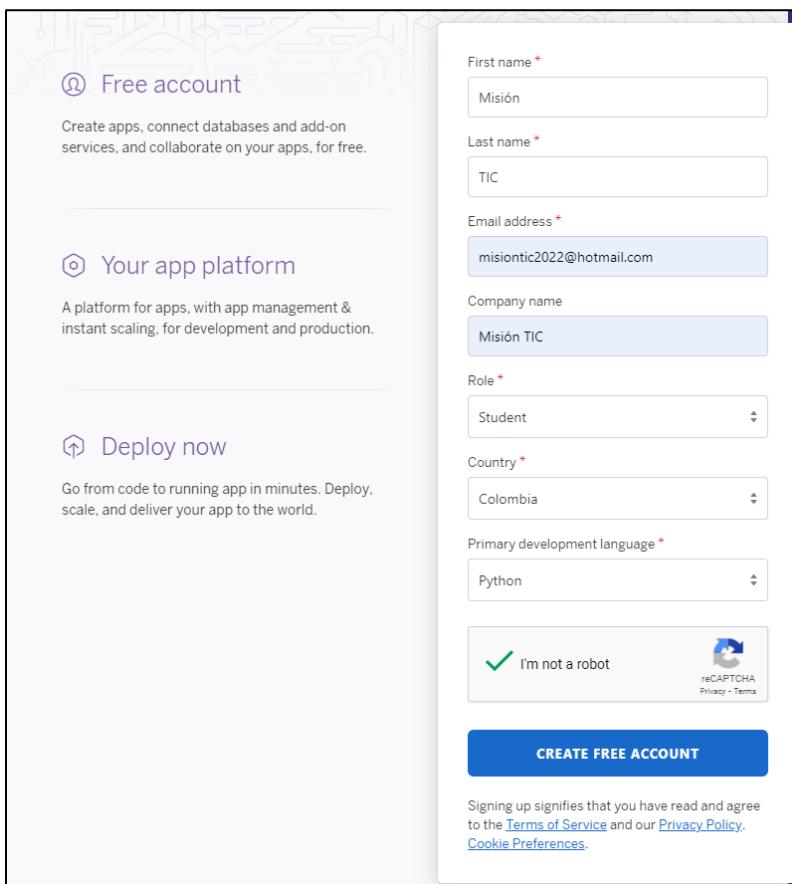
Imagen 3: Esquema Relacional

Generalmente, las bases de datos relacionales utilizan el lenguaje de consulta estructurado SQL para registrar y consultar datos. Algunos de los **DBMS** más famosos que siguen este paradigma son **PostgreSQL**, **MySQL**, **MariaDB** y **Microsoft SQL Server**. Para el componente de base de datos **bank\_db**, se utilizará una base de datos del **DBMS PostgreSQL**, este sistema brindará una base de datos relacional. En esta guía únicamente se creará la base de datos, la generación del esquema se realizará en guías posteriores.

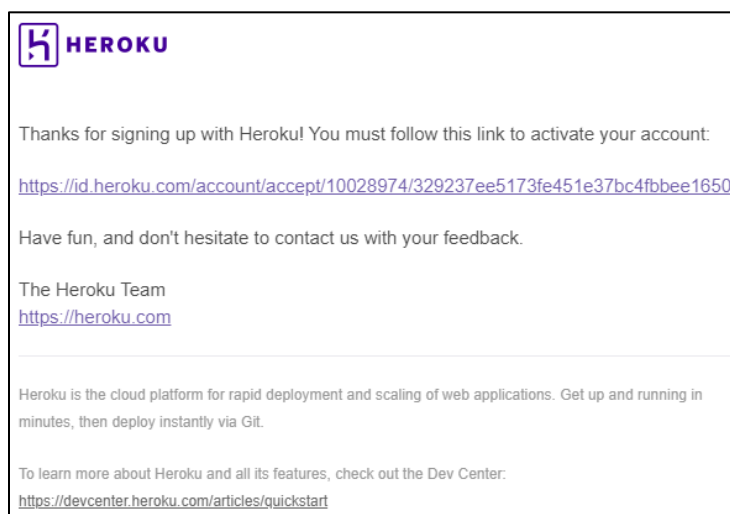
Ahora, si bien se podría desplegar la base de datos de manera local, teniendo en cuenta el propósito del sistema, se realizará el despliegue en la nube, para ello, se utilizará **Heroku**, una Plataforma como Servicio o **PaaS** (por sus siglas en inglés Platform as a Service) el cual brinda servicios de cómputo basados en la nube, sin necesidad de administrar el hardware.

### Crear una cuenta Heroku

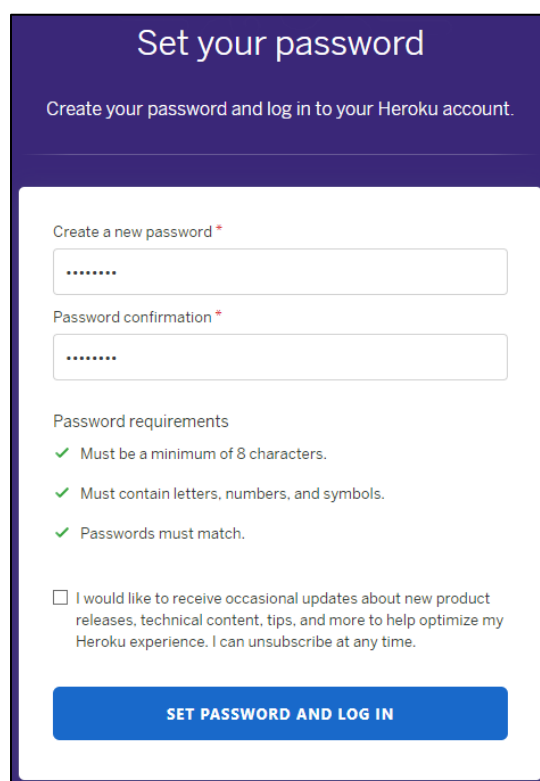
Para utilizar Heroku es necesario crear una cuenta, para ello se ingresa a la página oficial de Heroku <https://signup.heroku.com/> y se completa el formulario con la información solicitada:



Una vez pulsado el botón **CREATE FREE ACCOUNT** se debe acceder al correo electrónico ingresado anteriormente, donde se encontrará un correo de activación que contiene una URL a seguir:



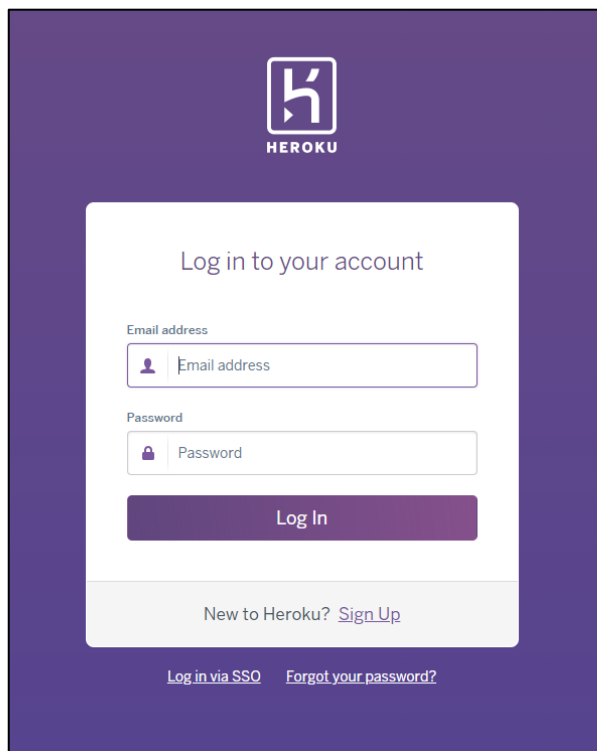
Allí se ingresa la contraseña de la cuenta y se indica si se desea recibir promociones de heroku por correo:

A screenshot of the "Set your password" form on the Heroku website. The form has a purple header with the text "Set your password" and "Create your password and log in to your Heroku account." Below the header, there are two input fields: "Create a new password \*" and "Password confirmation \*". Below these fields, there are three green checkmarks indicating password requirements: "Must be a minimum of 8 characters.", "Must contain letters, numbers, and symbols.", and "Passwords must match." At the bottom, there is a checkbox labeled "I would like to receive occasional updates about new product releases, technical content, tips, and more to help optimize my Heroku experience. I can unsubscribe at any time." and a blue button labeled "SET PASSWORD AND LOG IN".

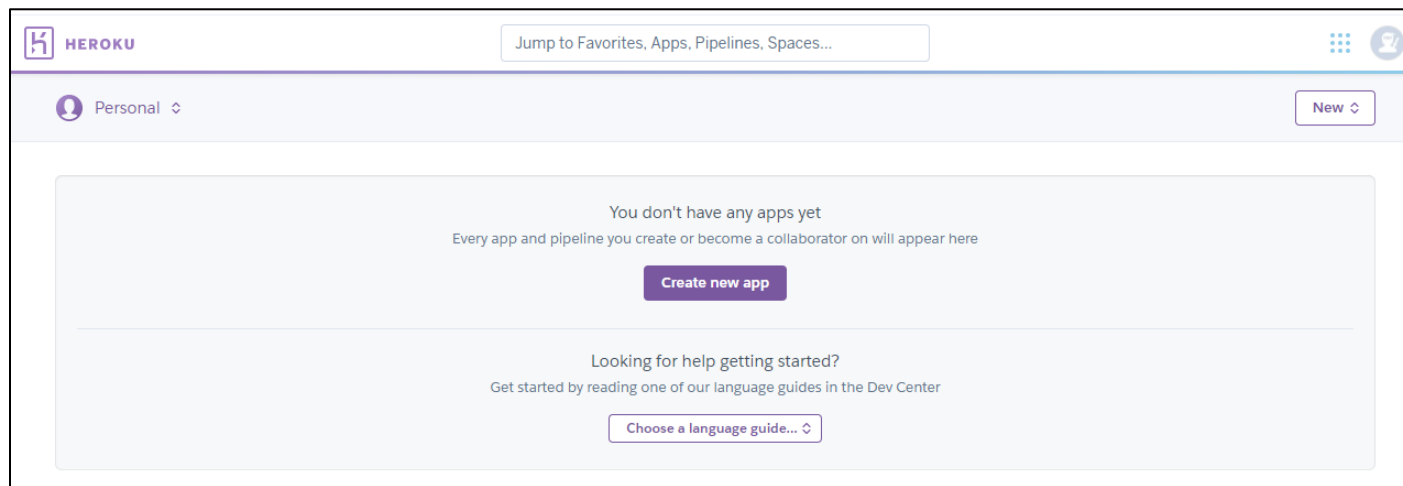
Una vez pulsado el botón **SET PASSWORD AND LOGIN**, se re-direcciona hacia otras dos páginas, una de ellas para confirmar la creación de la cuenta, y la otra para aceptar los Términos de Servicio. Una vez se ha hecho esto, se da por finalizada la creación de la cuenta Heroku, la cual se utilizará para realizar el despliegue de las tres capas de la aplicación del caso de estudio del Ciclo 3.

## Despliegue de la base de datos PostgreSQL en Heroku

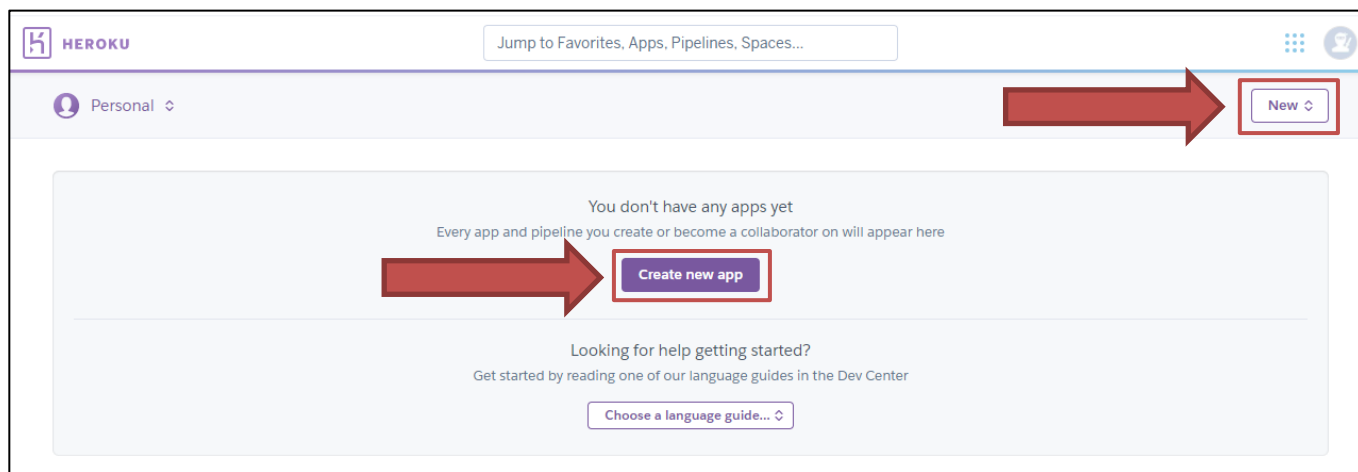
Para realizar el despliegue, se debe dirigir a la página de Heroku <https://www.heroku.com/> e ingresar a la cuenta creada anteriormente:



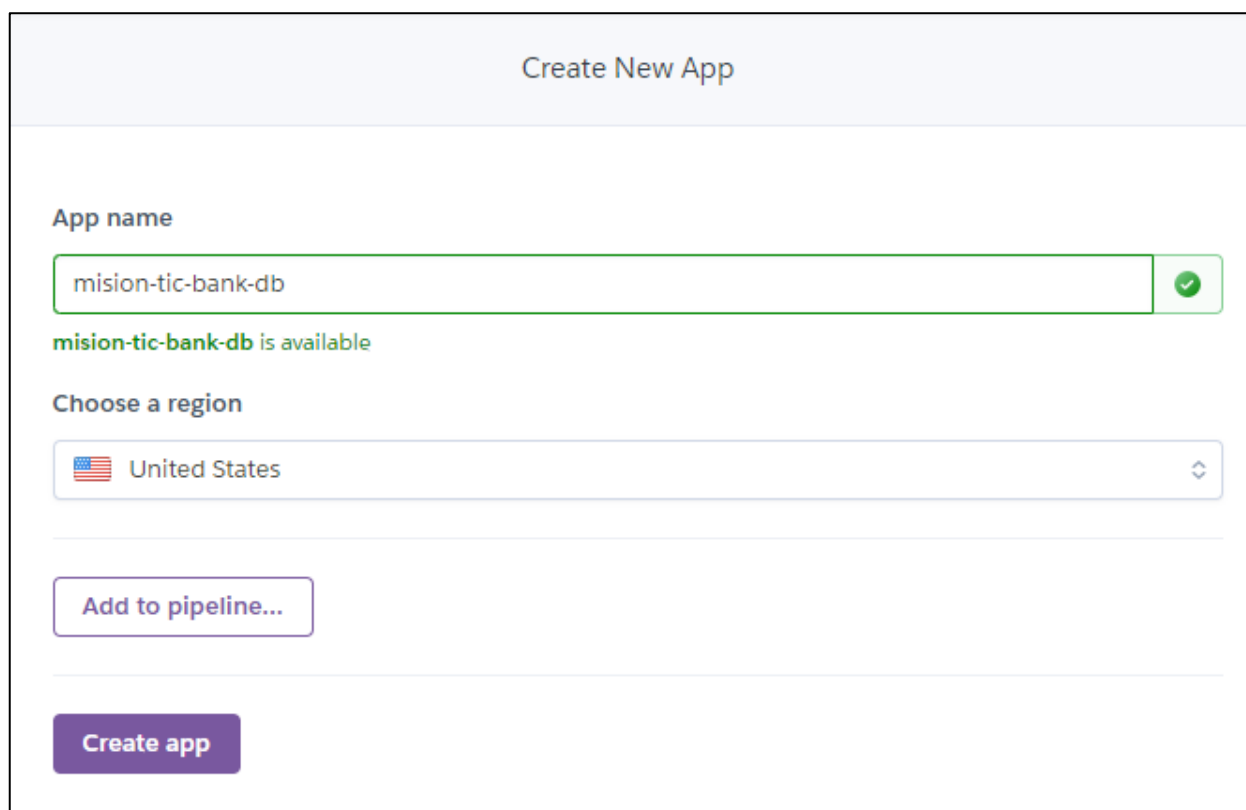
Allí se encuentra el *Dashboard*, es decir, el tablero donde se accede a las distintas aplicaciones asociadas a la cuenta Heroku:



Para desplegar una base de datos en Heroku, primero es necesario crear una aplicación. Esto se puede realizar de dos formas, utilizando el botón [New](#) que se encuentra en la sección superior derecha del [Dashboard](#), o utilizando el botón [Create new app](#) si el [Dashboard](#) está vacío.



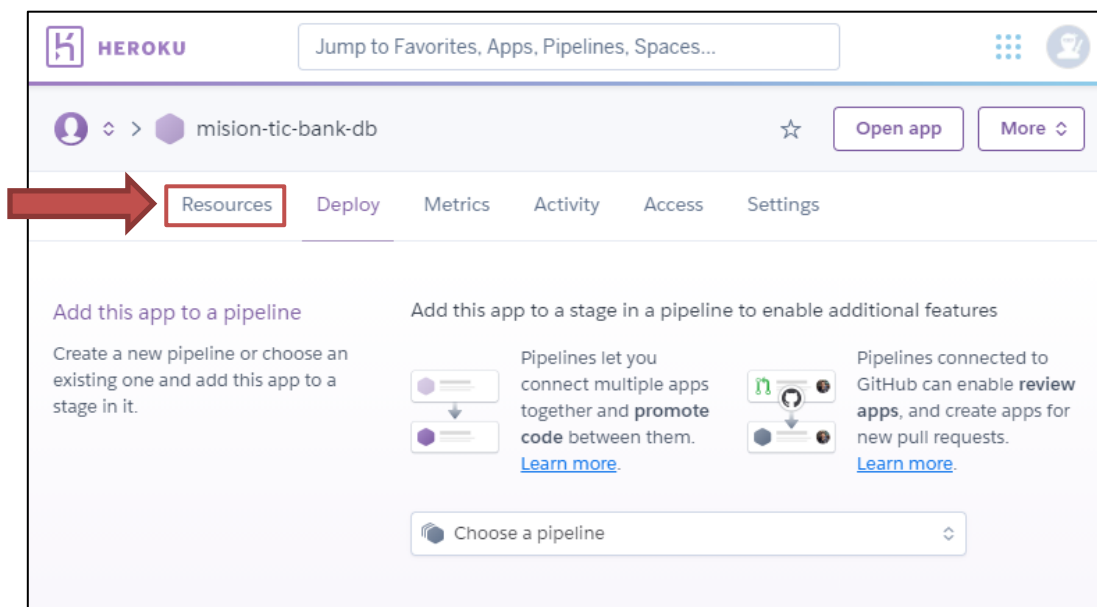
Para crear una aplicación únicamente se solicita el nombre que se le desea poner a esta. Dicho nombre es único dentro de todo Heroku, razón por la cual se sugiere que sea altamente descriptivo.



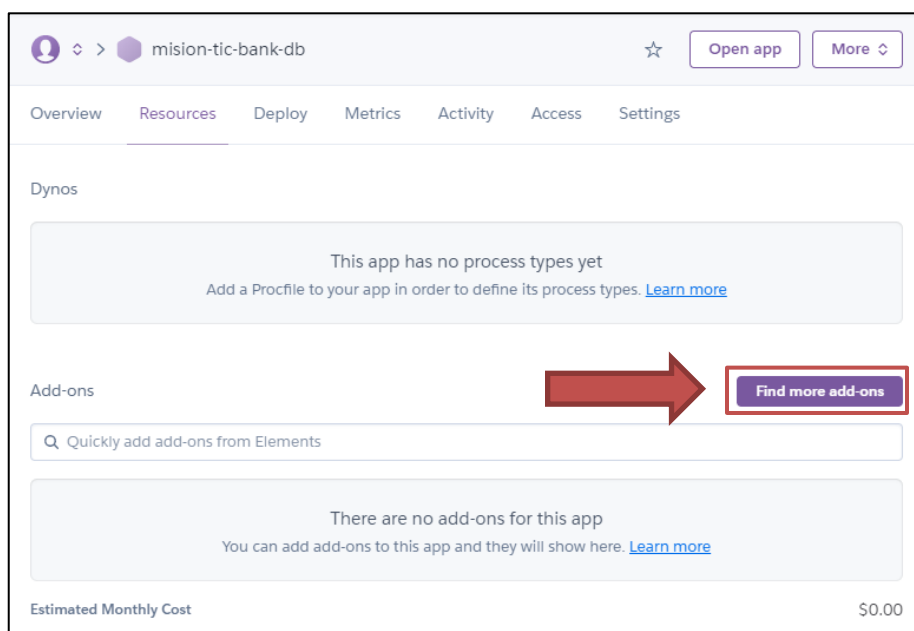
The screenshot shows the 'Create New App' form. The 'App name' field is filled with 'mision-tic-bank-db' and has a green checkmark icon to its right. Below the field, it says 'mision-tic-bank-db is available'. The 'Choose a region' dropdown menu is set to 'United States'. At the bottom, there is an 'Add to pipeline...' button and a 'Create app' button.



Luego de pulsar el botón [Create app](#) se mostrará la pestaña [Deploy](#), donde se pueden observar las funcionalidades que ofrece Heroku para el despliegue de la aplicación. Sin embargo, para desplegar una base de datos en Heroku, se debe asignar un recurso de base de datos a la aplicación. Para ello, se debe dirigir a la pestaña [Resources](#).

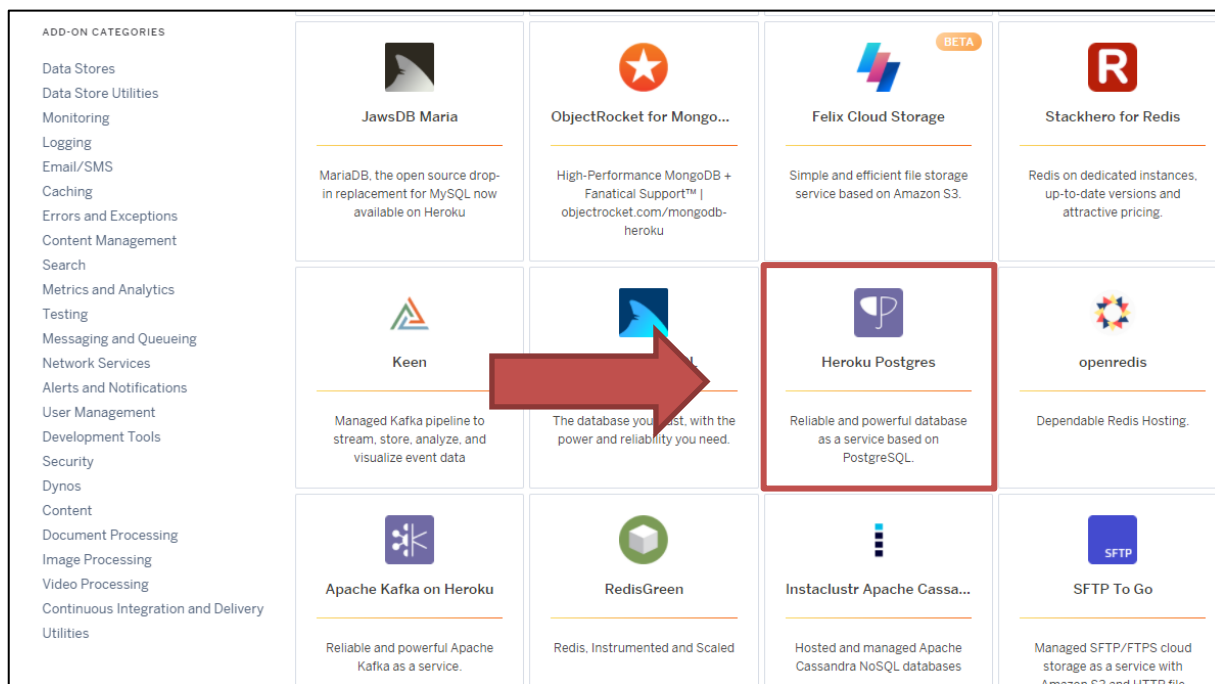


Como su nombre lo indica, en esta pestaña se pueden añadir múltiples recursos a la aplicación, pero en este caso solo se añadirá un recurso de base de datos. Para ello, se pulsa el botón [Find more add-ons](#):

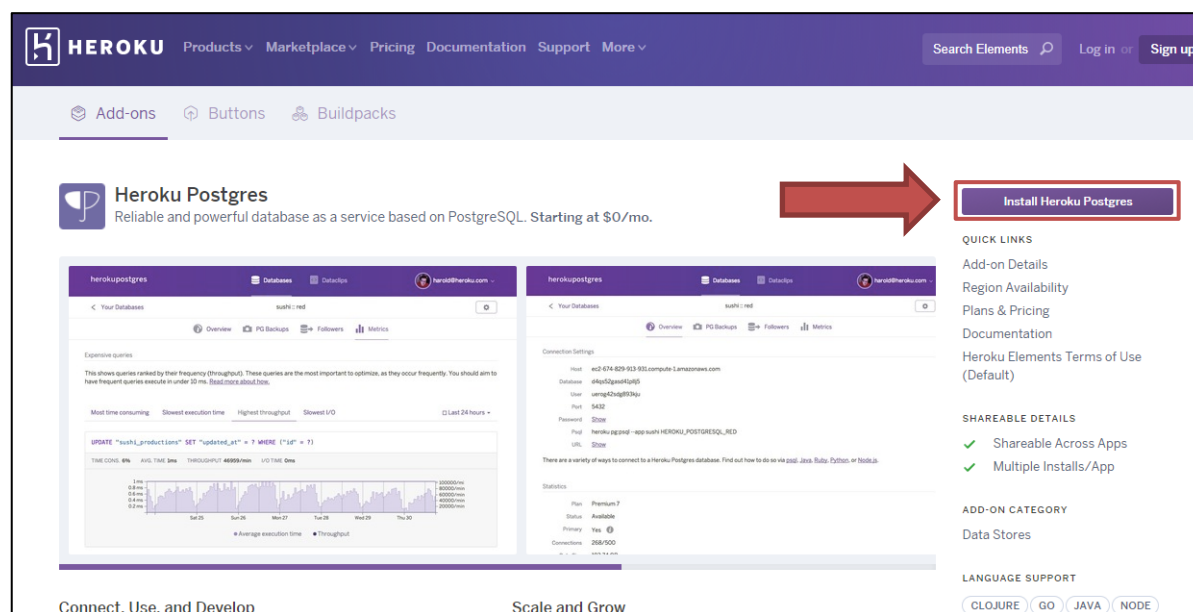





Esto redireccionará a la página donde se encuentran todos los recursos disponibles para la aplicación. Allí se debe buscar y pulsar el recurso [Heroku Postgres](#). Este se encuentra dentro de las primeras opciones en la página:



Para instalar el recurso en la aplicación, se debe pulsar el botón [Install Heroku Postgres](#):




Esto redireccionará a otra página donde se debe seleccionar la aplicación en la cual se desea instalar el recurso. Para ello se debe seleccionar el nombre de la aplicación y el plan **Hobby Dev – Free**, y posteriormente pulsar **Submit Order Form**:




Provision this add-on to an app  
Heroku Postgres  
[View on the Elements Marketplace](#)

Add-on plan  
Hobby Dev - Free



Heroku Postgres  
Hobby Dev - Free

→




mision-tic-bank-db  
Personal apps

In summary this will provision **Heroku Postgres** on your personal **mision-tic-bank-db** application, using the **Hobby Dev** plan (Free).

[Change destination app](#)

By submitting this order form, you agree that the Add-on is governed by the applicable provider's terms of use, and the Heroku Services are governed by the [Salesforce Master Subscription Agreement](#), unless (except for free customers) you have entered into a written Master Subscription Agreement executed by SFDC for the Heroku Services as referenced in the Documentation.

Submit Order Form



En este punto ya se ha agregado el recurso a la aplicación, de nuevo en la sección de **Resources** se puede observar que la base de datos ya existe. Para acceder al panel de control de la base de datos, únicamente se debe dar click sobre el nombre del recurso **Heroku Postgres**.


Overview
Resources
Deploy
Metrics
Activity
Access
Settings

Dynos  
This app has no process types yet  
Add a Profile to your app in order to define its process types. [Learn more](#)

Add-ons  
Find more add-ons

The add-on heroku-postgres1 has been installed. Check out the documentation in its [Dev Center](#) article to get started.

Quickly add add-ons from Elements

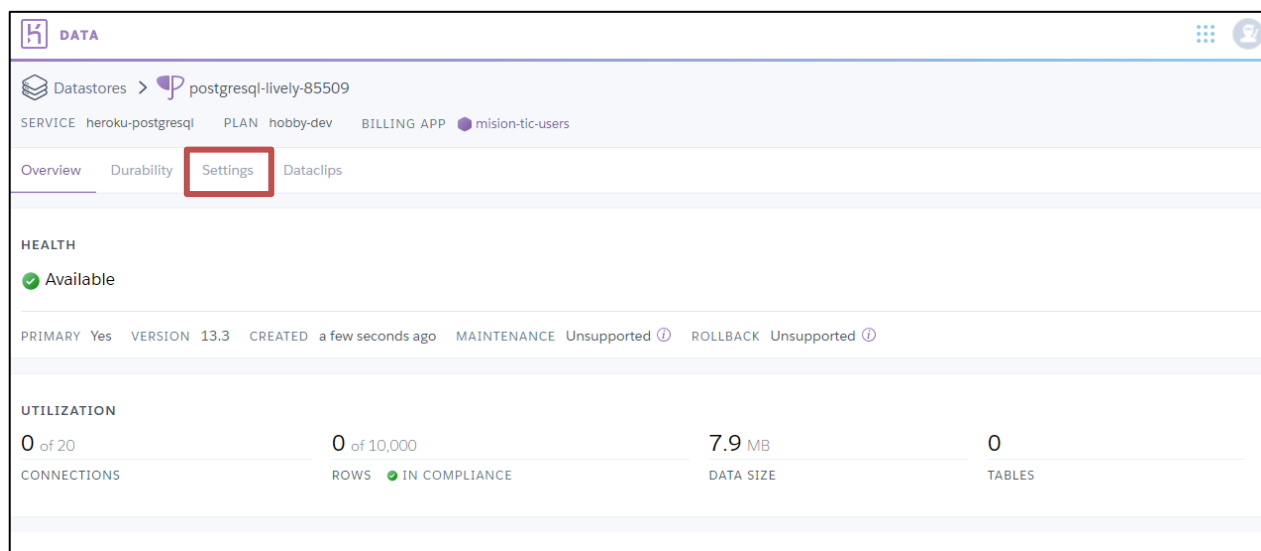

Heroku Postgres

Attached as DATABASE

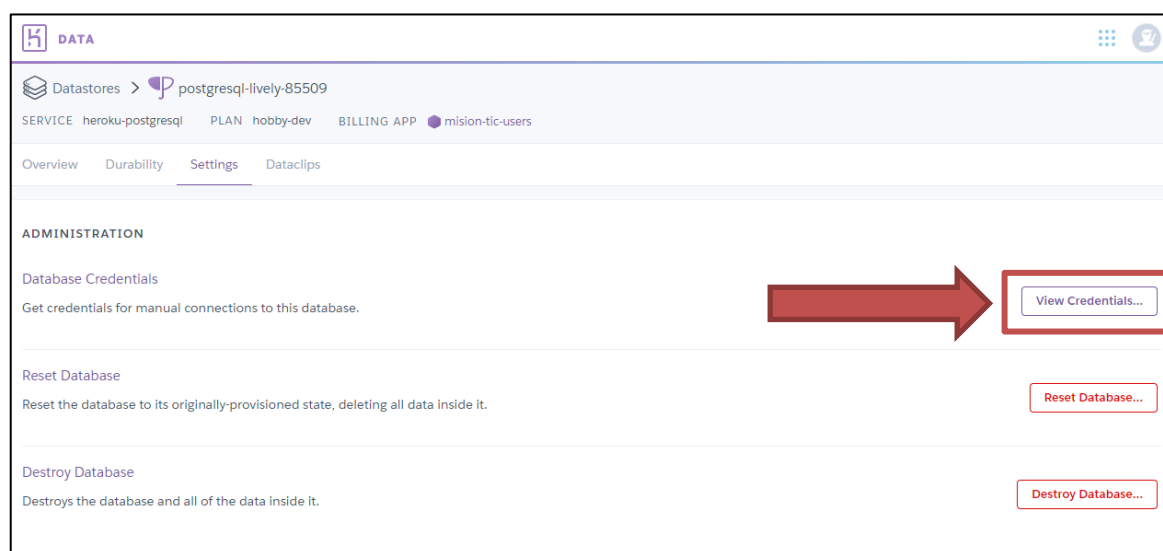
Hobby Dev
Free

Estimated Monthly Cost \$0.00

El panel de control de la base de datos presenta bastante información, como las conexiones a esta, la cantidad de información que almacena, su estado (*Available*), entre otros. La base de datos representa la capa de datos del caso de estudio de este ciclo, lo que significa que la capa lógica hará uso de ella para proveer sus funcionalidades. Por esta razón, es necesario obtener las credenciales de la base de datos que permitirán a la capa lógica acceder a la base de datos. Para obtener estas credenciales se debe dirigir a la pestaña *Settings*.



En la pestaña *Settings* aparecen algunas funcionalidades para eliminar la base de datos, se debe tener cuidado con estas opciones. Para acceder a las credenciales se debe pulsar el botón *View Credentials*:



Allí se mostrarán las credenciales de acceso a la base de datos, lo que incluye el *Host*, la *Database*, el *User*, el *Port* y la *Password*. Todos estos valores son necesarios para establecer una conexión con la base de datos.

ADMINISTRATION

Database Credentials
Get credentials for manual connections to this database.
Cancel

Please note that these credentials are not permanent.  
Heroku rotates credentials periodically and updates applications where this database is attached.

Host	ec2-35-170-85-206.compute-1.amazonaws.com
Database	d2rk7gefr0g3n0
User	pffhdzomulsbrh
Port	5432
Password	106b7a7a04[REDACTED]466dfaa55ba6b9
URI	postgres://pffhdzomulsbrh:106b7a7a[REDACTED]206.compute-1.amazonaws.com:5432/d2rk7gefr0g3n0
Heroku CLI	heroku pg:psql postgresql-lively-85509-[REDACTED]

Se debe recordar que estos valores son las credenciales de acceso a la base de datos, por cual se debe tener precaución con compartirlos.

## Cliente de la Base de Datos

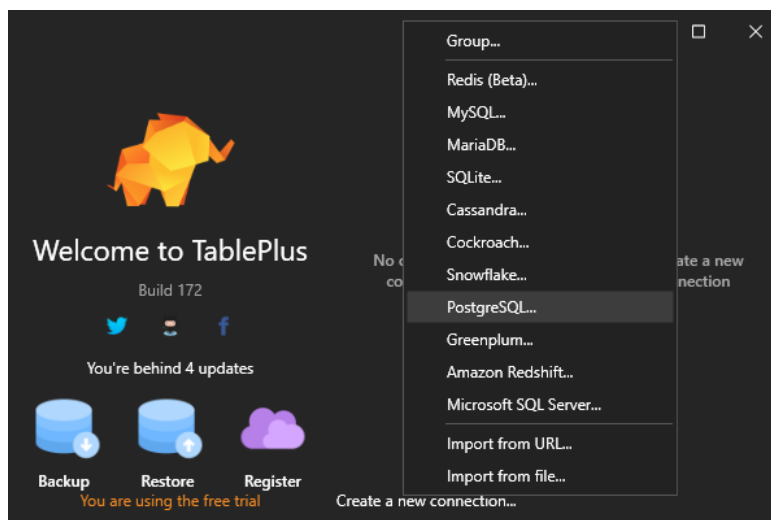
PostgreSQL al igual que la mayoría de los motores de bases de datos funcionan bajo el modelo *Cliente – Servidor*, este modelo se basa en la existencia de un servidor que recibe, procesa y responde peticiones a los clientes. Al realizar el despliegue de la base de datos en Heroku se creó dicho servidor. Para enviar peticiones al servidor PostgreSQL y observar su estado, existen múltiples clientes con una interfaz gráfica que lo permite. Para mostrar el funcionamiento de un cliente PostgreSQL, en esta guía se utilizará *TablePlus*, un cliente sumamente ligero y de fácil acceso. A continuación, se describen los pasos para la instalación del cliente de acuerdo con el sistema operativo:

### Instalación en Windows y Mac:

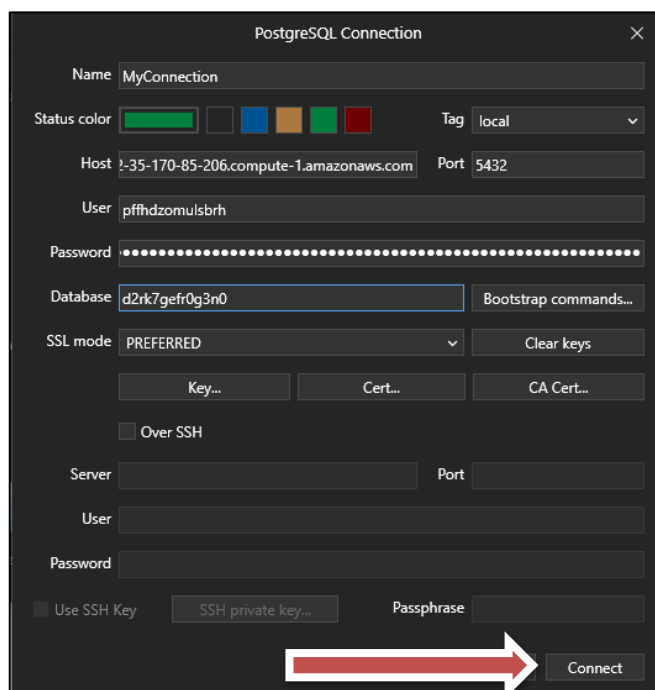
El proceso de instalación en Windows y Mac es bastante fácil, solamente se debe acceder a la página oficial de TablePlus <https://tableplus.com/> descargar el instalador y ejecutarlo, el proceso de instalación simplemente es un wizard en el cual se debe dar *next* en todas las opciones.

pág 12 de 14

Allí se solicitará indicar el motor de la base de datos al que se desea conectar. En este caso **PostgreSQL**:



Posteriormente, se solicitará un nombre para la conexión y las credenciales de la base de datos. Para ello, se rellenan estos campos con los valores generados anteriormente por Heroku. La interfaz solicita el ingreso de múltiples campos, pero solo es necesario ingresar los valores *Host*, *Port*, *User*, *Password* y *DataBase*. Una vez se han ingresado los valores correctamente se inicia la conexión pulsando el botón *Connect*:



Una vez establecida la conexión con el servidor de la base de datos, se podrá ver los datos que este contiene, sin embargo, actualmente no tiene información por lo cual solo se ve la pantalla vacía. A medida se desarrolle el caso de estudio se podrá observar en el cliente como se agregan tablas y datos.

