pgembed

While extensions like pgvector prepare PostgreSQL to store and query vector data, the process of generating these vectors—the *embedding*—still needs to be performed separatel

Typically, this process requires an application or pipeline to communicate with an external Machine Learning model, such as those offered by OpenAI, or locally with models available ir Ollama. This may require broader technical knowledge from the data team or the involveme of various professionals with different areas of expertise.

In many cases, projects focused on validating the *Minimum Viable Product (MVP)* choose to develop functions within PostgreSQL itself that consume vectorization services and facilitate this activity.

However, it is common for best practices, such as using batched data to reduce network traffic, managing memory consumption, security settings, and resilience mechanisms, to be overlooked in this approach.

Aiming to provide a set of functions for data vectorization in PostgreSQL, focusing on performance, security, and resilience, Tecnisys developed the pgembed extension.

Vectorization Functions

After installing and creating the pgembed extension, various functions for data vectorization will be available in the `pgembed` schema of the database, organized by the embedding API provider and the input data type:

Provider	Single Text Line	Set of Lines	Table
Ollama (local)	`embed_ollama`	`embed_batch_ollama`	`embed_table_ollama`
OpenAI	`embed_openai`	`embed_batch_openai`	`embed_table_openai`
Custom APIs	`embed_custom`	`embed_batch_custom`	`embed_table_custom`

Function Types:

- **Single Text Line** (`embed_*`): Generates the embedding for a single line of text.

  For example:

  ```
  SELECT pgembed.embed_openai('Hello, world!', 'text-embedding-3-small');
  ```

- **Set of Lines** (`embed_batch_*`): Processes multiple lines of text and returns a list of vectors.   For example:

  ```
      SELECT * FROM pgembed.embed_batch_openai(ARRAY['Hello, world!', 'Hello,
  PostgreSQL!'], 'text-embedding-3-small');
  ```

- **Table** (`embed_table_*`): Reads a text column from a table and automatically updates the corresponding embedding column, eliminating the need for manual update operations. For example:

  ```
    SELECT * FROM pgembed.embed_table_openai(
        'public',        -- schema
        'tb_documents',  -- table name
        'content',       -- content column
        'embedding',     -- embedding column
        FALSE,           -- regenerate: only update NULL embeddings (default:
  FALSE)
        1000,            -- batch_size
        'text-embedding-3-small'       -- model
    );
  ```

Advanced Parameters

In addition to the mandatory parameters, the pgembed extension functions have several extremely useful advanced parameters, such as:

- Request timeout (`timeout`, default 60s)
- Whether the SSL certificate should be verified (`verify_ssl`, default TRUE)
- Whether the text should be truncated if it exceeds the context size (`trucante`, default TRUE, for Ollama models)
- JSON for advanced options for each model (`options`, for Ollama models)
- Number of dimensions of the generated embeddings (`dimensions`, for OpenAI models)
- Encoding format of the returned embeddings (`encode_format`, for OpenAI models)

Security and Resilience

In terms of security and resilience, the pgembed extension provides parameters that can be configured at the user, session, and database levels, such as:

- `pgembed.openai_api_key` or `pgembed.custom_api_key` to define the API key used in the service request
- `pgembed.url_allowlist` to define the list of authorized URLs. By default: localhost, 127.0.0.1, 0.0.0.0 (any port), *.openai.com, and api.openai.com
- `pgembed.max_retries` , `pgembed.initial_backoff_ms` , `pgembed.max_backoff_ms` for retry control
- `pgembed.circuit_breaker_threshold` and `pgembed.circuit_breaker_reset_timeout_s` to prevent cascading failures from temporarily blocking requests to unavailable services

Source: https://github.com/tecnisys/pgembed

Installing pgembed via PgSmart CCLI

Command

```
bash title="Terminal input"          pgsmart install
```

1. Select the Database Option.

2. Select the `Extensions` Option.

3. Select the `Integration with Text Embedding APIs (pgembed)` Extension.

4. Inform the PostgreSQL Major version.

4.1 Inform the Release of the PostgreSQL Major version.

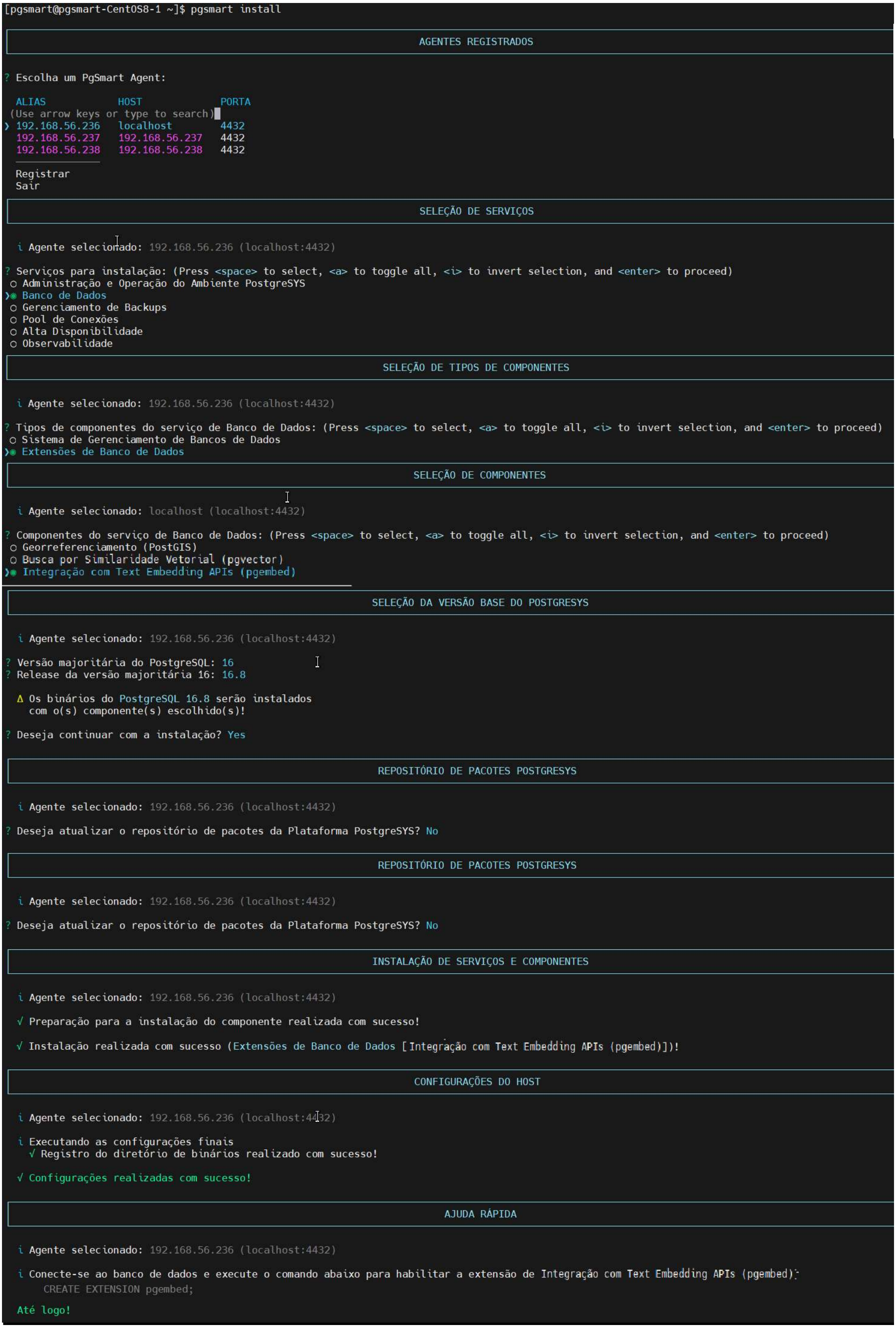
5. Confirm the extension installation:

5.1. Confirm if you wish to install/update the PostgreSYS package repository (this installation is necessary to proceed if the repository is not yet configured).

5.2. If you chose to install/update the Package Repository, inform the Package Repository URL.

> ⓘ NOTE

> The Package Repository installation option is not performed if the pgsys repositories are already present or mapped on the machine.



```
[pgsmart@pgsmart-CentOS8-1 ~]$ pgsmart install

                              AGENTES REGISTRADOS

? Escolha um PgSmart Agent:

  ALIAS            HOST             PORTA
 (Use arrow keys or type to search)
❯ 192.168.56.236   localhost        4432
  192.168.56.237   192.168.56.237   4432
  192.168.56.238   192.168.56.238   4432
  ──────────────
  Registrar
  Sair

                              SELEÇÃO DE SERVIÇOS

  i Agente selecionado: 192.168.56.236 (localhost:4432)

? Serviços para instalação: (Press <space> to select, <a> to toggle all, <i> to invert selection, and <enter> to proceed)
 ○ Administração e Operação do Ambiente PostgreSYS
❯◉ Banco de Dados
 ○ Gerenciamento de Backups
 ○ Pool de Conexões
 ○ Alta Disponibilidade
 ○ Observabilidade

                        SELEÇÃO DE TIPOS DE COMPONENTES

  i Agente selecionado: 192.168.56.236 (localhost:4432)

? Tipos de componentes do serviço de Banco de Dados: (Press <space> to select, <a> to toggle all, <i> to invert selection, and <enter> to proceed)
 ○ Sistema de Gerenciamento de Bancos de Dados
❯◉ Extensões de Banco de Dados

                            SELEÇÃO DE COMPONENTES

  i Agente selecionado: localhost (localhost:4432)

? Componentes do serviço de Banco de Dados: (Press <space> to select, <a> to toggle all, <i> to invert selection, and <enter> to proceed)
 ○ Georreferenciamento (PostGIS)
 ○ Busca por Similaridade Vetorial (pgvector)
❯◉ Integração com Text Embedding APIs (pgembed)

                     SELEÇÃO DA VERSÃO BASE DO POSTGRESYS

  i Agente selecionado: 192.168.56.236 (localhost:4432)

? Versão majoritária do PostgreSQL: 16
? Release da versão majoritária 16: 16.8

  Δ Os binários do PostgreSQL 16.8 serão instalados
    com o(s) componente(s) escolhido(s)!

? Deseja continuar com a instalação? Yes

                        REPOSITÓRIO DE PACOTES POSTGRESYS

  i Agente selecionado: 192.168.56.236 (localhost:4432)

? Deseja atualizar o repositório de pacotes da Plataforma PostgreSYS? No

                        REPOSITÓRIO DE PACOTES POSTGRESYS

  i Agente selecionado: 192.168.56.236 (localhost:4432)

? Deseja atualizar o repositório de pacotes da Plataforma PostgreSYS? No

                     INSTALAÇÃO DE SERVIÇOS E COMPONENTES

  i Agente selecionado: 192.168.56.236 (localhost:4432)

  √ Preparação para a instalação do componente realizada com sucesso!

  √ Instalação realizada com sucesso (Extensões de Banco de Dados [Integração com Text Embedding APIs (pgembed)])!

                             CONFIGURAÇÕES DO HOST

  i Agente selecionado: 192.168.56.236 (localhost:4432)

  i Executando as configurações finais
    √ Registro do diretório de binários realizado com sucesso!

  √ Configurações realizadas com sucesso!

                                 AJUDA RÁPIDA

  i Agente selecionado: 192.168.56.236 (localhost:4432)

  i Conecte-se ao banco de dados e execute o comando abaixo para habilitar a extensão de Integração com Text Embedding APIs (pgembed):
      CREATE EXTENSION pgembed;

  Até logo!
```

Figure 6 - Interactive pgembed extension installation

6. Connect to the database and run the command below:

Terminal input

```
CREATE EXTENSION pgembed;
```

Installing pgembed via PgSmart Web

1. Select the `Home/Manage Environments/Services` tab.
   All services installed in all environments will be displayed.

2. Click `Install`.

3. Select the `Environment` where the Service will be installed.

4. Select the `Database` service.

5. Select the `Base Version` of PostgreSQL [Ex: 16].

6. Select the `PostgreSQL Version` [Ex: 16-2].

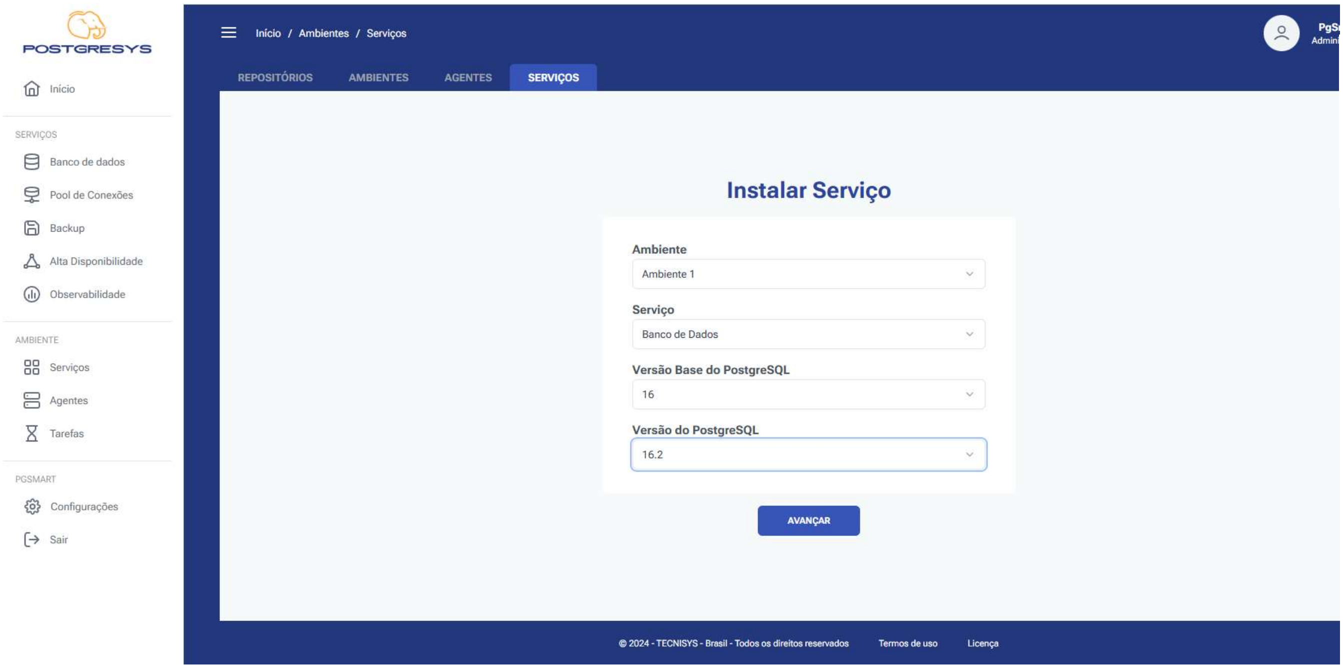
7. Click `Next`.

8. Select the `Integration with Text Embedding APIs (pgembed)` Extension.

9. Inform if you wish to initialize the database instance.

10. Select the `IP address` of one or more PgSmart Agents.

11. Click `Install`.



Instalar Serviço

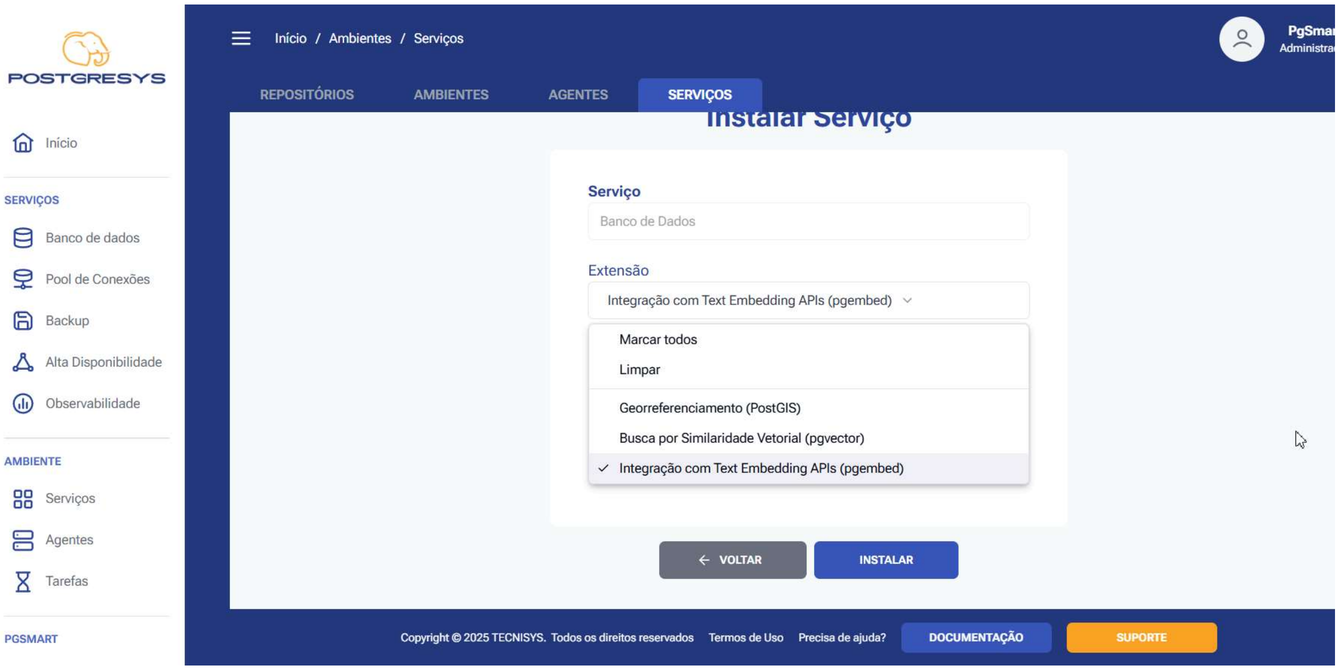
Ambiente
Ambiente 1

Serviço
Banco de Dados

Versão Base do PostgreSQL
16

Versão do PostgreSQL
16.2

Figure 1 - Database Service Installation

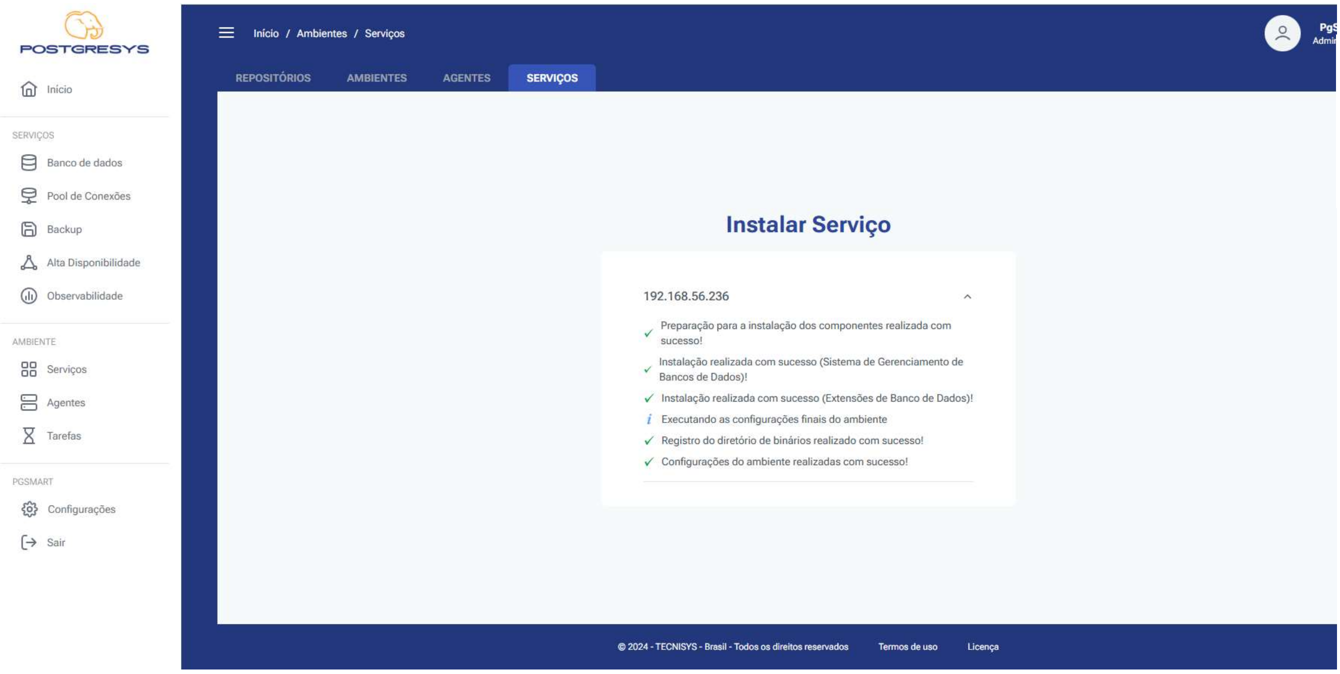


Instalar Serviço

Serviço
Banco de Dados

Extensão
Integração com Text Embedding APIs (pgembed)

Figure 2 - Database Service Installation



Instalar Serviço

192.168.56.236

✓ Preparação para a instalação dos componentes realizada com sucesso!

✓ Instalação realizada com sucesso (Sistema de Gerenciamento de Bancos de Dados)!

✓ Instalação realizada com sucesso (Extensões de Banco de Dados)!

i Executando as configurações finais do ambiente

✓ Registro do diretório de binários realizado com sucesso!

✓ Configurações do ambiente realizadas com sucesso!

Figure 3 - Database Service Installation