

# Desafio de Solução Tech - BHub

## Candidato

Alvaro Zaragoza, +55 11 9730-8569, alvaro@denzar.com.br

## Ponto de Partida

Estou considerando que já temos uma aplicação de gestão de vendas funcionando, e que o objetivo é desenvolver novos recursos para permitir ações diferenciadas após o pagamento dos pedidos de venda.

Estou considerando também que já temos o front-end independente do back-end.

## Solução proposta

Criar novas tabelas no banco de dados (MySQL) para armazenar e gerir as ações que devem ocorrer no momento em que os pedidos de venda são pagos.

Tabelas:

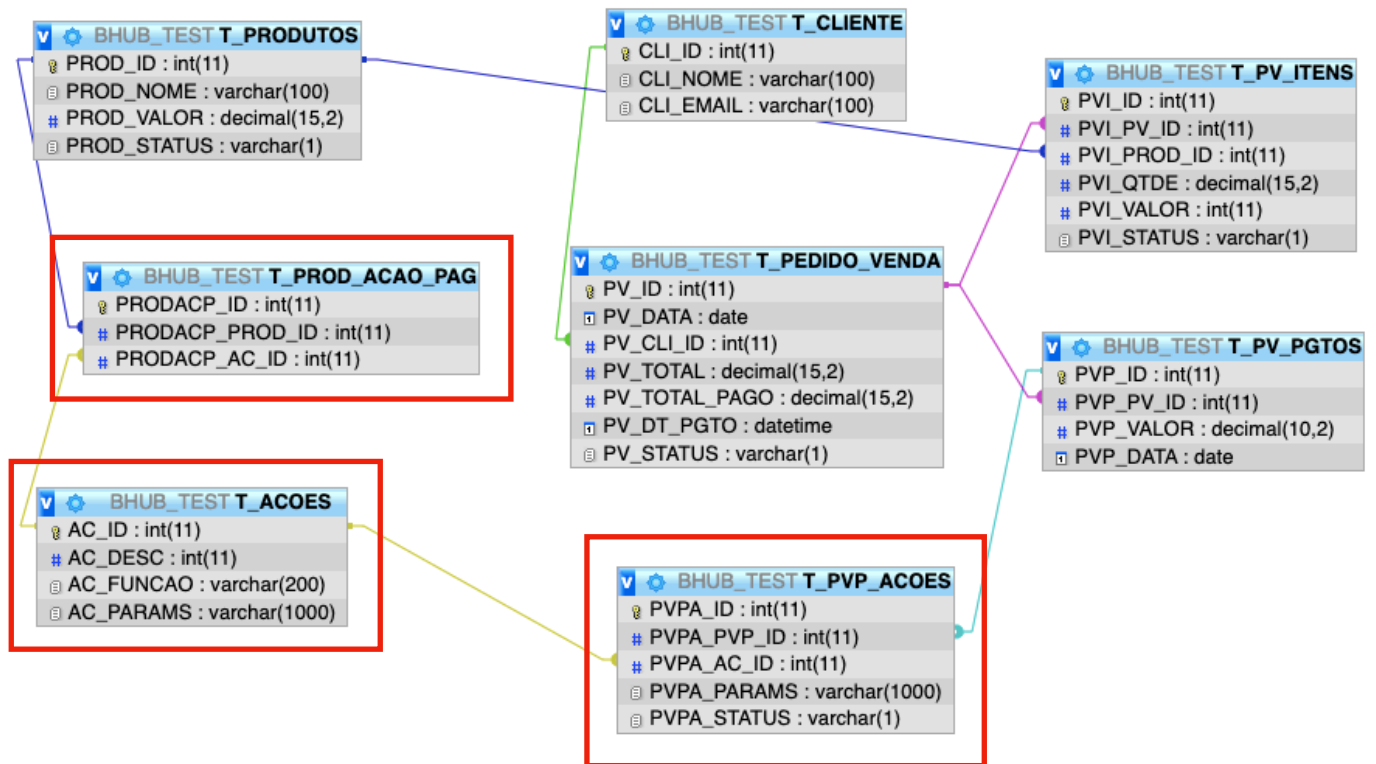
- 1) Tabela de Ações (T\_ACOES): ID, descrição, função e parâmetros da ação que será executada. A função pode ser um código dentro da própria aplicação ou até mesmo uma chamada de API externa. Os parâmetros serão os nomes dos campos (variáveis) necessários para a execução da função, e devem ser especificados em formato STRING separados por vírgula. Ex.: "PV\_ID, PROD\_ID"
- 2) Tabela de Ações por Produto Pago (T\_PROD\_ACAO\_PAG): ID do registro, ID do produto e ID da ação. Tabela para ligar ações em produtos pagos.
- 3) Tabela de Ações por Pagamento de PV (T\_PVP\_ACOES) (opcional): Nesta tabela serão armazenadas todas as ações que **foram** executadas após o pagamento de um pedido de venda. O armazenamento, apesar de opcional, servirá para documentar e permitir o rastreamento para saber se todas as ações foram devidamente programadas e executadas. Nesta tabela, na coluna de parâmetros (PVPA\_PARAMS), o JSON conterá não apenas o campo, mas também seu conteúdo passado. Ex.: {"PV\_ID":1,"PROD\_ID":1}

Criar um módulo a ser incorporado no back-end para codificar as funções que serão executadas. Neste projeto esse arquivo é o ACOES.PY localizado na pasta SERVER da aplicação.

Criar uma nova rota no back-end que deverá ser acionada após a confirmação do pagamento e levará como parâmetros todas as informações necessárias para a execução da função da ação.

## Tabelas no MySQL

Abaixo, apenas a título de ilustração, temos as novas tabelas que serão adicionadas e suas relações com outras tabelas do banco de dados principal.



## Novas tabelas

Além das tabelas, criei também uma VIEW (v\_pv\_itens\_acoes) para facilitar o envio das funções e parâmetros para o processamento das ações necessárias.

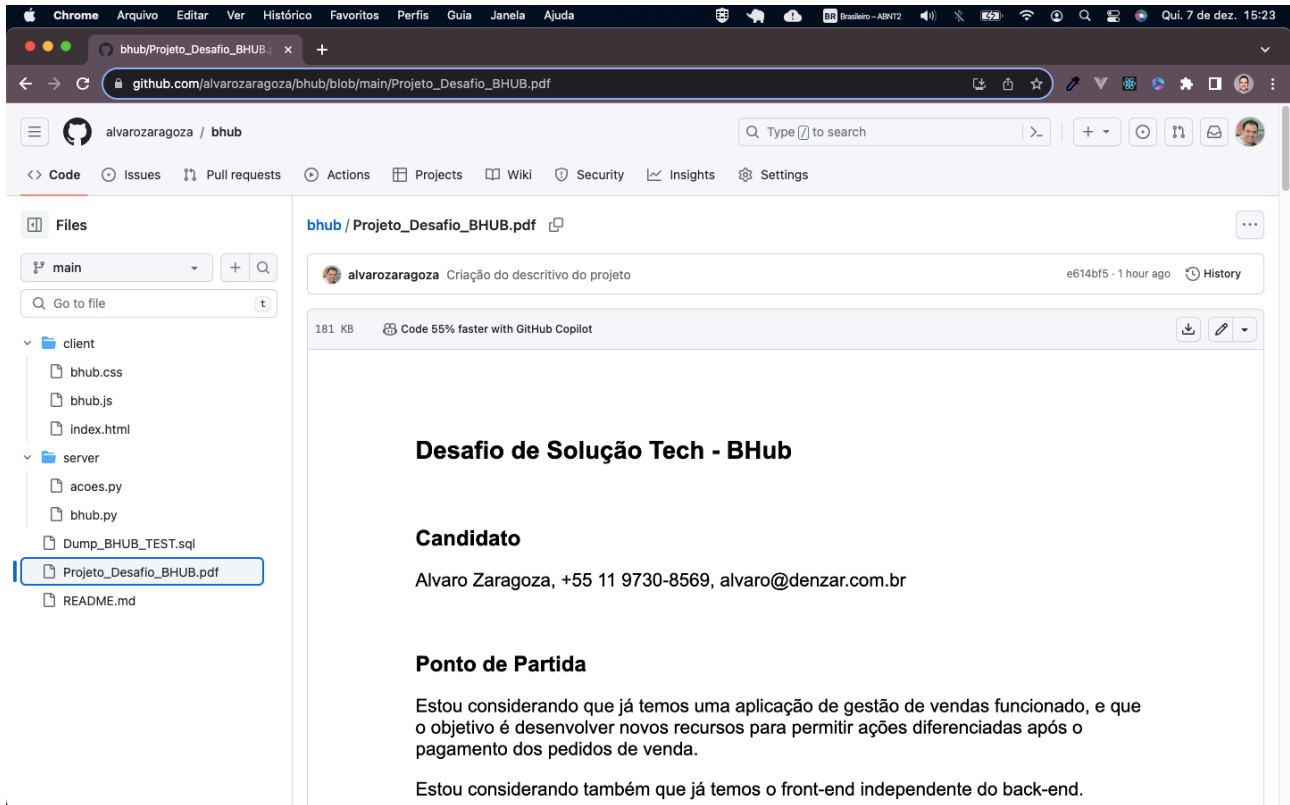
Um dump do banco de dados para testes, inclusive com dados, está anexo ao projeto.

## Estrutura da aplicação

Abaixo temos a estrutura da aplicação criada e disponibilizada no GitHub.

Repositório:

<https://github.com/alvarozaragoza/bhub>



## Para testar

- 1) Crie uma pasta no seu computador, acesse o repositório <https://github.com/alvarozaragoza/bhub> e copie todo o conteúdo para a mesma.
- 2) Acesse seu MySQL e crie o banco de dados BHUB\_TEST. Selecione-o e execute o script 'Dump\_BHUB\_TEST.sql' copiado para a pasta que você criou no passo 1. Para acessar o banco de dados esta aplicação usa: host=localhost, user=root e password=root. Caso seja diferente no seu MySQL, abra o arquivo BHUB.PY da pasta SERVER e coloque as informações corretas nas linhas 12, 13 e 14, e salve as modificações.
- 3) Abra uma janela TERMINAL do seu sistema operacional e acesse a pasta SERVER dentro da pasta que você criou no passo 1.
- 4) Para o funcionamento do back-end (SERVER) é necessário ter instalado a versão 3.0 ou superior do Python. Caso não tenha, acesse <https://www.python.org/downloads> e siga os passos de instalação.
- 5) Após a instalação do Python, é necessário instalar as bibliotecas FLASK e FLASK-CORS, caso ainda não tenha no seu ambiente Python. Para instalar execute os seguintes comandos no prompt do seu terminal:

```
pip install Flask
```

```
pip install -U flask-cors
```

- 6) Para iniciar o servidor back-end, ainda no prompt de comando do terminal aberto na pasta SERVER, execute:

```
python3 bhub.py
```

- 7) Entre as várias mensagens que serão apresentadas deve aparecer uma linha contendo 'Running on http://127.0.0.1:5000'. Esse é o endereço e a porta (5000) que o Front-end (CLIENT) tentará acessar. Caso ocorra algum erro, ou a porta seja diferente de 5000, solicite ajuda através da página do GitHub, informada no passo 1.
- 8) Agora é só abrir o arquivo INDEX.HTML da pasta CLIENT no seu navegador preferido, e testar.
- 9) Em caso de dúvidas envie sua mensagem através do GitHub ou para o email: [alvaro@denzar.com.br](mailto:alvaro@denzar.com.br).