



# ANBIMA

## DESAFIO TÉCNICO

# Estágio Anbima

## Desafio Técnico — String Posicional de Pedidos de Lanche 🍔

### Objetivo

Não esperamos que o desafio seja 100% concluído nem que o código seja de alta qualidade. O mais importante para nós é entender:

- O caminho que você escolheu para resolver o problema ou parte dele
- Sua capacidade de raciocínio e resolução
- Seu empenho em estudar e pesquisar para chegar em uma solução ou parte da solução

Prazo de entrega: **2 semanas** a partir do Recebimento do teste.

**Dúvidas:** podem se enviadas para [diego.silva@selic.anbima.com.br](mailto:diego.silva@selic.anbima.com.br) e [fellipe.juncal@selic.anbima.com.br](mailto:fellipe.juncal@selic.anbima.com.br) (encaminhar para os 2)

Cada candidato deve:

1. Criar repositórios públicos no GitHub contendo o código fonte

2. Subir os commits de forma incremental (não apenas um commit final). Vamos analisar históricos de commits e evolução da solução
3. Apenas commits feitos até a data limite definida serão considerados na avaliação
4. Se não der tempo de concluir todas as etapas, entregue o que conseguiu, vamos avaliar também organização, clareza e evolução mesmo em soluções parciais
5. Ao **concluir o desafio** ou ao **chegar o prazo final**, o candidato deve enviar o link do repositório com tudo que tenha feito até o momento para os e-mails:  
**diego.silva@selic.anbima.com.br** e **fellipe.juncal@selic.anbima.com.br**

Implementar um mini-sistema com **aplicações em Spring Boot**. O sistema será dividido em dois módulos e um frontend em Angular

- **Módulo A (Entrada/Gateway)**

- Endpoint:

- **POST /pedidos/posicional**
    - Recebe uma linha posicional fixa de 40 caracteres(Content-Type: text/plain).
  - Validações da string posicional:
    - Deve ter exatamente 40 caracteres
    - Campos alfanuméricos (A) → preencher com espaços à direita quando menor que o tamanho
    - Campos numéricos (N) → preencher com zeros à esquerda quando menor que o tamanho
    - Quantidade deve ser numérica entre 01 e 99
  - Conversão:
    - Transformar a string em um **objeto Pedido** com os campos: **id, tipoLanche, proteina, acompanhamento, quantidade, bebida, valor, status, criadoEm**
  - Regras de negócio:
    - Preço base:
      - HAMBURGUER → R\$ 20,00
      - PASTEL → R\$ 15,00
      - Outros → R\$ 12,00
    - Se for **HAMBURGUER + CARNE + SALADA** → **Aplicar 10% de desconto**
    - **valorTotal = precoBase \* quantidade - desconto**
  - Testes Unitários
  - Ações no módulo:
    - Salvar no banco (status = RECEBIDO)
    - Publicar mensagem na fila pedidos.recebidos com esse JSON: {"pedidoId": <ID>}
    - Endpoint para receber string posicional do frontend
    - **Testes unitários**
- **Módulo B (Entrega/Processor)**

**Consome** a mensagem enviada pelo Módulo A, **atualiza** o pedido correspondente para status=ENTREGUE e persiste

  - Listener:

- Consumir mensagens da fila pedidos.recebidos.
- Para cada mensagem: buscar o pedido no banco pelo pedidoId e atualizar status = ENTREGUE
- API de consulta:
  - GET /pedidos → Listar todos os pedidos
  - GET /pedidos/{id} → Consultar um pedido específico
- **Frontend Angular (simples)**
  - Tela “Novo Pedido”:
    - Textarea para colar uma string posicional (40 caracteres)
    - Botão Enviar → chamar POST /pedidos/posicional
    - Exibir o JSON retornado ou mensagem de erro
  - Tela “Pedidos”:
    - Tabela com colunas: id, tipoLanche, proteina, acompanhamento,o quantidade, bebida, valor, status.
    - Botão Atualizar para recarregar tabela

## Layout da String Posicional

Cada linha deve ter **exatamente 40** caracteres.

Se vier com menos, **complete** conforme as regras de preenchimento; se vier com mais, **rejeite**.

**Tipo A = Alfanumérico** (letras, números e espaços). **Preencher com espaços** à direita quando faltar.

**Tipo N = Numérico** (apenas dígitos). **Preencher com zeros à esquerda** quando faltar.

Campo	Posição (1-based)	Tamanho	Tipo	Regras/Exemplo
tipoLanche	1-10	10	A	Ex.: HAMBURGUER / PASTEL (complete com espaços)
proteina	11-20	10	A	Ex.: CARNE / FRANGO (complete com espaços)
acompanha mento	21-30	10	A	Ex.: SALADA / BACON (complete com espaços)
quantidade	31-32	2	N	01 a 99 (preencher com zeros à esquerda)
bebida	33-40	8	A	Ex.: COCA / SUCO (complete com espaços)

Regras de validação da linha

- Comprimento **exato** = **40** (se maior, rejeitar; se menor, rejeitar).
- quantidade ∈ 01..99 (numérico).

Modelo de Domínio (Banco de Dados)

Tabela **pedidos**:

Coluna	Tipo	Observação
id	BIGSERIAL (PK)	
tipo_lanche	VARCHAR(20)	
proteina	VARCHAR(20)	
acompanhamento	VARCHAR(20)	
quantidade	INT	
bebida	VARCHAR(20)	
valor	NUMERIC(10,2)	valor total calculado
status	VARCHAR(20)	RECEBIDO ou ENTREGUE
criado_em	TIMESTAMP	default now()

Endpoints

- **Módulo A (Entrada/Gateway)**
- **POST /pedidos/posicional**
  - Content-Type: text/plain
  - Body: uma linha posicional (40 chars)
  - Ações: validar → converter → calcular valor → salvar (status=RECEBIDO) → publicar na fila {pedidoId}
  - 201 Created com JSON do pedido salvo
- **Módulo B (Entrega/Processor)**
- **Listener da fila pedidos.recebidos**
  - Ao consumir {pedidoId}, atualizar o registro para ENTREGUE

- **GET /pedidos**
  - Lista com todos os pedidos
- **GET /pedidos/{id}**
  - Detalhe de um pedido.

## Frontend Angular

---

### Tela “Novo Pedido”

- Textarea para colar a **linha posicional** (40 chars) e botão **Enviar** (POST do Módulo A).
- Exibe sucesso/erro.

### Tela “Pedidos”

- Tabela com: id, tipoLanche, proteina, acompanhamento, quantidade, bebida, valor, status.
- Filtro por status.
- Botão **Atualizar**.

## Exemplos de Linhas para Teste Manual

---

### 1. Com desconto (HAMBURGUER + CARNE + SALADA, 1 unid)

HAMBURGUER·CARNE····SALADA····01COCA···

- valor esperado: 20,00 com 10% → **18,00**

### 2. Sem desconto (PASTEL + FRANGO + BACON, 2 unid)

PASTEL····FRANGO···BACON····02SUCO···

- valor esperado: 15,00 \* 2 → **30,00**

Observação: nos exemplos, “.” representa espaço visualmente. No arquivo/textarea real são **espaços**.

## Observações Finais

---

- Os **testes unitários** ficam em aberto (variam conforme a implementação).
- Não esperamos que o desafio seja 100% concluído
- Prazo de entrega: **2 semanas** a partir do recebimento do teste.
- Dúvidas: podem ser enviadas para **diego.silva@selic.anbima.com.br** e **fellipe.juncal@selic.anbima.com.br** (encaminhar para os 2)
- Ao **concluir o desafio** ou ao **chegar o prazo final**, o candidato deve enviar o link do repositório para os e-mails: **diego.silva@selic.anbima.com.br** e **fellipe.juncal@selic.anbima.com.br**