# Teste Técnico – Desenvolvedor(a) Full Stack .NET

### Objetivo

Criar uma aplicação **Full Stack** para gestão de telecomunicações, onde seja possível **gerenciar contratos de operadoras, faturas e consumo de telecom**, utilizando **.NET Core, Angular e boas práticas de desenvolvimento**.

### Descrição do Desafio

Você deve desenvolver um **Sistema de Gestão de Telecom**, permitindo que a empresa controle os contratos de telefonia e internet das suas filiais. A aplicação deve conter:

### Funcionalidades

#### Cadastro de Operadoras

* Criar, editar, listar e excluir operadoras de telecomunicações (Vivo, Claro, TIM, etc.).
* Campos: **ID, Nome da Operadora, Tipo de Serviço (Móvel, Fixo, Internet), Contato Suporte**.

#### Gestão de Contratos

* Cadastrar contratos por filial da empresa.
* Campos: **ID, Nome da Filial, Operadora, Plano Contratado, Data de Início, Data de Vencimento, Valor Mensal, Status (Ativo/Inativo)**.
* Associar cada contrato a uma **Operadora** previamente cadastrada.

#### Registro de Faturas

* Cadastrar faturas mensais para cada contrato.
* Campos: **ID, Contrato, Data de Emissão, Data de Vencimento, Valor Cobrado, Status (Paga/Pendente/Atrasada)**.
* Criar uma funcionalidade que calcula automaticamente **o total de gastos da empresa** no mês.

#### Dashboards

Criar um **dashboard interativo** no front-end usando **amCharts ou Chart.js** contendo os seguintes gráficos:

* **Gráfico de Pizza:** Distribuição das faturas por status (**Paga, Pendente, Atrasada**).
* **Gráfico de Barras:** Evolução mensal do total de faturas emitidas e pagas no último ano.
* **Indicadores Numéricos (Cards):**
  + Total de faturas emitidas.
  + Valor total faturado.

#### Notificações de Vencimento (Opcional)

Criar um **serviço background** que verifica se há planos vencendo nos próximos **5 dias** e dispara um e-mail.

**Requisitos Técnicos**

**Back-End (.NET Core 6+)**

* Criar API REST com **.NET Core / C#**.
* Uso de **boas práticas** (SOLID, Clean Code, DDD - opcional).

**Banco de Dados**

* A aplicação deve permitir o uso de um banco  **PostgreSQL**.
* Para facilitar a execução local, o candidato pode usar **EF Core In-Memory**, garantindo que os dados persistam durante a execução da aplicação.

**Front-End (Angular 17+)**

* Criar interface em **Angular** para gerenciar os dados.
* Implementar **CRUD** completo das entidades.
* Utilizar **Bootstrap ou Material UI** para um design simples e responsivo.

**Formato de entrega:**

* Código fonte no **GitHub/GitLab** (repositório público ou privado).
* README com instruções para rodar o projeto.

**Critérios de Avaliação**

* **Qualidade do código** (legibilidade, organização e boas práticas).
* **Estrutura do projeto** (separação de camadas, modularização).
* **Funcionalidade** (se atende aos requisitos).
* **Uso correto de tecnologias** (Angular, .NET, SQL).