📝 Case Técnico – Desenvolvedor Pleno Full Stack

Contexto

A empresa precisa de um processo de automação para o **monitoramento da recepção de arquivos** enviados pelas adquirentes.
Esses arquivos contêm informações financeiras que devem ser **registradas e disponibilizadas** para a equipe operacional acompanhar em tempo real.

Atualmente, os arquivos podem estar em duas situações:

- **Recepcionados**
- **Não Recepcionados**

O desafio é desenvolver um **MVP (Minimum Viable Product)** que automatize esse fluxo e permita o acesso via portal web.

Objetivo do Case

Desenvolver uma aplicação que seja capaz de:

1. **Recepcionar arquivos** enviados por adquirentes (UfCard e FagammonCard).
2. **Registrar e armazenar** as informações principais em banco de dados.
3. **Disponibilizar um portal web** que consuma APIs para consulta dos dados.

Layout de Arquivo

Início	Fim	Tam.	Tipo	Descrição	Exemplo
001	001	1	NUM	Tipo de Registro (0)	0
002	011	10	NUM	Estabelecimento	0987564321
012	019	8	NUM	Data Processamento (AAAAMMDD)	20190626
020	027	8	NUM	Período Inicial	20190625
028	035	8	NUM	Período Final	20190625
036	042	7	NUM	Sequência	0000001
043	050	8	ALFA	Empresa (UfCard)	UfCard

Exemplo:
0987564321201906262019062520190625000001UfCard

Início	Fim	Tam.	Tipo	Descrição	Exemplo
001	001	1	NUM	Tipo de Registro (1)	1
002	009	10	NUM	Data Processamento (AAAAMMDD)	20190526
010	017	8	NUM	Estabelecimento	32165487
018	029	12	ALFA	Empresa (FagammonCard)	FagammonCard
030	036	7	NUM	Sequência	0002451

120190526321654 87FagammonCard0002451

Requisitos Obrigatórios (MVP)

Backend / API

- Leitura dos arquivos no formato especificado.
- Validação mínima do layout.
- Registro das informações relevantes no banco de dados.
- Realização de **backup do arquivo** em um diretório seguro.

Portal Web

- Listagem dos arquivos processados.
- Exibição do status (**Recepcionado** ou **Não Recepcionado**).
- Gráfico comparando a quantidade de arquivos recepcionados x não recepcionados.
- Todas as operações devem ser feitas **via API** (sem lógica direta no frontend).

Requisitos Opcionais

- Implementação de **cache**.
- Criação de **testes unitários**.
- Organização da aplicação em camadas (services, controllers, repositories, etc).

- Utilização de **boas práticas de versionamento** (commits claros, branches, etc).

Critérios de Avaliação

A avaliação será feita considerando os seguintes pontos:

1. **Organização do código** (Clean Code, clareza, boas práticas).
2. **Estrutura da API** (endpoints REST claros, separação de responsabilidades).
3. **Modelagem do banco** (simples mas bem estruturada).
4. **Tratamento de erros** (arquivos inválidos, layouts incorretos, etc).
5. **Portal Web** (usabilidade mínima e consumo de API).
6. **Extras / Diferenciais** (cache, testes, documentação clara, dockerização, etc).

Entrega

- Link do projeto no **GitHub pessoal**.
- O repositório deve conter um **README.md** com instruções de execução (como rodar o backend, o frontend e popular o banco).
- Não é necessário realizar deploy em nuvem.