Atividade (versão revisada e didática)

Cada aluno deverá, individualmente, acessar o repositório **https://github.com/wiltonwilldepaulo/Jaiminho**, criar um **fork** para a sua conta do GitHub, clonar o projeto localmente e implementar, nos arquivos **fornecedor.html**, **cliente.html** e **empresa.html**, o **preenchimento automático de dados** utilizando **Fetch API** do JavaScript (sem bibliotecas externas).

Nos três formulários, quando o campo **CNPJ perder o foco** (evento blur), o sistema deve consultar a BrasilAPI em **https://brasilapi.com.br/api/cnpj/v1/{CNPJ_SOMENTE_DIGITOS}** e, havendo retorno válido, **preencher automaticamente** os campos pertinentes do cadastro — por exemplo: **Razão Social**, **Nome Fantasia**, **CNPJ**, **CEP**, **Logradouro**, **Bairro**, **Município** e **UF** (e outros que façam sentido de acordo com o layout do formulário). De forma análoga, quando o campo **CEP perder o foco**, o sistema deve consultar a BrasilAPI em **https://brasilapi.com.br/api/cep/v1/{CEP_SOMENTE_DIGITOS}** e **completar Logradouro, Bairro, Cidade e UF**. Em ambos os casos, **sanitizar as entradas** para aceitar **apenas dígitos** (remover máscaras) antes de enviar a requisição.

A implementação deve adotar **boas práticas de UX e acessibilidade**: indicar **carregamento** durante a consulta (ex.: desabilitar botões ou mostrar spinner), exibir **mensagens de erro claras** para valores inválidos ou indisponibilidade do serviço (incluindo tratamento de 404 e falhas de rede), manter rótulos e associações corretas com <label>, e utilizar uma região de feedback com aria-live para informar resultados/erros sem prejudicar a navegação por teclado.

Utilize **somente HTML**,e **JavaScript** (Fetch), mantendo **comportamento idêntico** nos três arquivos solicitados. A entrega consiste em publicar o trabalho em um **repositório público** na sua conta do GitHub contendo a cópia do projeto com as alterações realizadas; recomenda-se disponibilizar uma visualização no **GitHub Pages** e incluir um **README** sucinto com instruções de teste (como abrir os arquivos .html e exemplos de CNPJ/CEP para validação). O prazo final para envio e apresentação é **16/09/2025, às 21:00** . Na avaliação serão considerados: **funcionamento correto** do auto-preenchimento por CNPJ e CEP nos três formulários, **organização e clareza do código**, **qualidade da experiência do usuário** (feedback, acessibilidade, tratamento de erros) e **documentação/ histórico de commits** no repositório.

Check-list de entrega (marque cada item ao concluir)

- Fazer **fork** do repositório Jaiminho para a sua conta do GitHub.
- **Clonar** o fork localmente e abrir o projeto no editor de sua preferência.
- Em **fornecedor.html**, implementar:
    - Evento blur no campo **CNPJ** → chamada à BrasilAPI /cnpj/v1/{cnpj} (somente dígitos).
    - **Preenchimento automático** dos campos (Razão Social, Nome Fantasia, CNPJ, CEP, Logradouro, Bairro, Município, UF…).
    - Evento blur no campo **CEP** → chamada à BrasilAPI /cep/v1/{cep} (somente dígitos).
    - **Preenchimento automático** de Logradouro, Bairro, Cidade e UF.
- Repetir o mesmo comportamento em **cliente.html**.
- Repetir o mesmo comportamento em **empresa.html**.
- **Sanitizar entradas** de CNPJ e CEP (aceitar apenas dígitos).
- Implementar **indicadores de carregamento** durante as requisições (desabilitar ações/spinner).
- Exibir **mensagens de erro** claras para CNPJ/CEP inválidos ou serviço indisponível; tratar 404 e falhas de rede.
- Garantir **acessibilidade básica**: <label> associado, foco de retorno em caso de erro, aria-live para feedback.
- **Reutilizar funções** (ex.: utilitários onlyDigits, fetchJSON, preencherPorCNPJ, preencherPorCEP) para evitar duplicação.
- **Testar** com CNPJs/CEPs válidos e inválidos (simular indisponibilidade de rede).
- Publicar o projeto atualizado em um **repositório público** no seu GitHub.
- Escrever um **README** objetivo: como testar, o que foi implementado e exemplos de uso.