Guia Prático: Como Trabalhar com o Sistema Usando Git.

Para quem é este guia?

Este material é para qualquer colaborador responsável por desenvolver, alterar ou manter sistemas internos da empresa, garantindo total rastreabilidade e padronização.

Estrutura do Projeto

Nome do projeto: ***
Local do repositório: GitHub

O que tem dentro?

- pasta_do_projeto/: onde estão os arquivos do sistema (PHP, HTML, JS, etc.)
- sql/: arquivos relacionados ao banco de dados (ex: scripts de criação de tabelas)
- prints/: imagens para documentação ou prints do sistema
- README.md: arquivo que explica o projeto, objetivos e instruções básicas

---

Processo Completo: Do recebimento da demanda à entrega

1. Recebimento da Requisição

- Toda solicitação deve vir com um formulário de requisição padrão preenchido.
- Deve ser feito um planejamento para entrega das solicitações, buscando dar foco nas que possuem prioridade maior ou prazo menor de acordo com a complexidade.
- Fazer reuniões de alinhamento das demandas.
- Alimentar gestão de projeto com os status das demandas, sempre atualizando o progresso delas.

2. Clonagem do Repositório dentro do Visual Studio Code

`git clone https://github.com/USUARIO/nome_sistema.git`

`cd nome_sistema`

`code .`

3. Criar uma Nova Branch para Trabalhar

`git checkout -b feat/nome-da-demanda`

4. Desenvolver Localmente

- Trabalhe no código dentro da pasta do projeto
- Teste localmente com servidor (XAMPP)
- Verifique se nada quebrou antes de seguir

5. Comitar as Alterações

`git add .`

`git commit -m "feat: adiciona tela de conferência de recebimento"`

6. Enviar para o GitHub

`git push origin feat/nome-da-demanda`

7. Abrir um Pull Request

- Acesse o GitHub
- Clique em “Compare & Pull Request”
- Descreva o que foi feito e relacione com a requisição

8. Revisão

- Outro desenvolvedor revisa antes de aprovar
- Se aprovado, será feito o **merge** para a branch principal (main)

Tabela de Comandos Git Úteis

Comando	O que faz
git status	Verifica arquivos modificados
git add .	Adiciona todos os arquivos modificados
git commit -m "mensagem"	Cria um commit com a descrição
git push origin nome-da-branch	Envia para o GitHub
git pull	Atualiza seu código local com o GitHub
git checkout -b nome	Cria e troca para uma nova branch

Dicas Importantes

- Sempre use nomes de branch com padrão: feat/, fix/, refactor/
- Use commit mensagens claras e descritivas
- Nunca trabalhe direto na branchmain
- Sempre relate a qual requisição cada commit pertence

Principais termos do Git

Branch: Linha paralela de desenvolvimento no código, onde você pode trabalhar isoladamente sem afetar a branch principal.

Main (ou Master): Branch principal do projeto, onde normalmente fica o código estável.

Commit: Registro de uma alteração feita no código. Cada commit salva um “snapshot” do projeto.

Merge: Ação de juntar as mudanças de uma branch em outra (geralmente da branch de desenvolvimento para a principal).

Checkout: Comando para mudar de uma branch para outra.

Pull: Atualizar sua branch local com as mudanças feitas na branch remota (online).

Push: Enviar seus commits locais para o repositório remoto (ex: GitHub).

Controle de Versões

- **Versão Inicial:** 1.0
- **Última Atualização:** 23/07/2025

Termo de Ciência e Concordância

Declaro, para os devidos fins, que li, compreendi e estou ciente dos termos estabelecidos na Política de trabalho com **GitHub** e **Visual Studio Code** da empresa **Comercial Atacado Souza Distribuidor**. Comprometo-me a seguir rigorosamente as diretrizes descritas neste documento, bem como atuar de forma ética, responsável e conforme os procedimentos definidos.

Data: ____ / ____ / _____

Nome Completo: _____

Cargo / Função: _____

Assinatura: _____