

Importância do portfólio

Um portfólio é uma coleção de materiais que mostram sua experiência de trabalho e habilidades para clientes e empregadores em potencial. Essas amostras de trabalhos produzidos fornecem uma prova do que o profissional é capaz de entregar.

O portfólio para desenvolvedor é o passo mais importante para conquistar o primeiro emprego ou a recolocação profissional na área de tecnologia. Basicamente, o portfólio complementa as informações do currículo e demonstra como você trabalha na prática. Portanto, é uma vitrine indispensável.

Além disso, um portfólio pode ser importante para mostrar resultados. Assim, é possível que o profissional crie uma apresentação em que mostre ao empregador que pode alcançar objetivos em uma determinada área ou função. Portanto, portfólios podem evidenciar habilidades técnicas e até comportamentais dos profissionais.

A atividade de desenvolvimento é bastante prática. Por isso, apenas um currículo não é capaz de expressar todas as habilidades do developer. Neste caso, um portfólio para desenvolvedor é um ponto-chave para passar nas entrevistas.

Portfólio para desenvolvedor Júnior

Nesse sentido, o portfólio tem ainda mais valor quando é feito por um desenvolvedor iniciante, no nível Júnior. Isso porque, na maioria das vezes, ele não tem experiência. Mas possui conhecimento das stacks solicitadas na vaga.

Um bom portfólio abrirá muitas portas. De acordo com um levantamento, 71% dos empregadores concordam parcialmente ou totalmente que a qualidade de um portfólio influencia suas decisões de contratação.

Embora um portfólio deva mostrar sua criatividade e a forma como você trabalha, ele também deve ter uma estrutura clara para a pessoa sobre quem você é e seus valores. Saber como organizá-lo e formatá-lo é fundamental.

É importante que seja seletivo e mostre seu melhor trabalho, ao invés de tudo que você já criou;

Mantenha seu portfólio atualizado; ao concluir novos projetos, remova exemplos desatualizados ou irrelevantes.

Onde criar e publicar portfólio para desenvolvedor?

GitHub

O GitHub é uma plataforma de compartilhamento de código-fonte muito usada por v comunidade.

A plataforma é sondada pelos tech recruiters, que podem avaliar todas as etapas do seu projeto e utilizá-lo como embasamento para as perguntas da entrevista. Por isso, é muito importante deixar seu GitHub atualizado e com os projetos mais recentes e alinhados com a sua área de atuação.

LinkedIn

O seu perfil profissional no LinkedIn também pode ser de extrema importância para divulgar os seus atributos, criar novas conexões, colaborar com a sua comunidade e manter seu currículo atualizado.

Como o LinkedIn é considerada a principal rede social profissional, você pode aproveitar as funcionalidades desse canal para conseguir um emprego. Enfim, é possível postar vídeos de você codando, integrar seus principais certificados, receber recomendações da sua rede de contatos e ainda interagir nas comunidades voltadas ao seu perfil profissional.

Git e GitHub

À primeira vista, pode parecer que "Git" e "Github" são a mesma coisa, mas a resposta é não.

São ferramentas distintas, mas colaboram de maneira integrada para tornar o desenvolvimento de software mais eficiente.

Aqui está um resumo rápido sobre cada uma delas — que também nos ajuda a diferenciá-las:

O Github é uma “rede social dev” em que é possível armazenar e compartilhar projetos de desenvolvimento de software.

O Git é um sistema de controle de versão de arquivos; em outras palavras, é responsável por guardar o histórico de alterações sempre que alguém modificar algum arquivo que está sendo monitorado por ele.

Desta maneira, o Git e o GitHub são pilares fundamentais que auxiliam as equipes de desenvolvimento a controlar o versionamento de código, rastrear mudanças, colaborar de forma eficiente e garantir que o trabalho em equipe flua sem problemas.

O que é Git e para que serve?

Em uma equipe, apenas poder acessar o código de outras pessoas colaboradoras não é suficiente.

Mais do que isso: precisamos manter o histórico dos nossos arquivos e das nossas modificações.

Afinal de contas, muitas vezes mudamos arquivos em grupo, num movimento único — que, no contexto do Git, é um commit. O que, em tradução literal para português, significa “compromisso” ou “comprometer-se” às alterações em um repositório.

Dessa forma, podemos voltar atrás e recuperar o estado do sistema: como ele era ontem, ou no ano passado, comparar as mudanças para encontrar bugs e estudar otimizações.

Como funciona o Git?

Todos os nossos arquivos, assim como seus históricos, ficam em um repositório e existem vários sistemas que gerenciam repositórios assim, como CVS (Sistema de Versões Concorrentes) e SVN (Subversion do Apache). O Git é uma alternativa com um funcionamento mais interessante ainda: ele é distribuído e todo mundo tem uma cópia inteira do repositório, não apenas o "servidor principal".

E uma das vantagens disso é que cada pessoa pode desenvolver offline, realizando seus commits e outras operações sem depender de uma conexão constante com o servidor principal.

Modelos de Colaboração com Ramificações

Todas essas funcionalidades que vimos sobre o Git se justificam porque diversas organizações utilizam essa ferramenta como sistema de controle de versões.

Por isso, conhecimentos sobre o Git são comumente encontrados em requisitos de vagas de emprego no ramo de desenvolvimento de software.

É comum que haja projetos que utilizam ramificações diferentes para criar novas features, fazer testes, correções de bugs e outras atividades.

O que é controle de versão no Git

O controle de versão, também conhecido como controle de fonte, é a prática de rastrear e gerenciar as alterações em um código de software. Os sistemas de controle de versão são ferramentas de software que ajudam as equipes de software a gerenciar as alterações ao código-fonte ao longo do tempo.

Controle de versão distribuído

Existem dois tipos de sistemas de controle de versão. Em um deles, pode haver um único servidor central que armazena o projeto com seu histórico, com o qual as pessoas desenvolvedoras precisam interagir. Isso é característico de um sistema de Controle de Versão Centralizado.

No outro tipo, cada pessoa desenvolvedora pode manter uma cópia do projeto em sua máquina local, o que é conhecido como Controle de Versão Distribuído, que é o caso do Git.

Com o Git, cada pessoa desenvolvedora tem uma versão completa do histórico do projeto. Isso proporciona independência e permite o desenvolvimento em paralelo.

Quais são os principais comandos do Git?

O Git é uma ferramenta bastante robusta e oferece diversos utilitários para gerenciar as versões de um projeto em linha de comando. Confira os principais comandos da ferramenta:

- Git init;
- Git clone;
- Git status;
- Git add;
- Git commit;
- Git log;
- Git branch;
- Git checkout;
- Git diff.

O que é GitHub?

O GitHub tem sim muita relação com o Git.

GitHub é uma plataforma para gerenciar seu código e criar um ambiente de colaboração entre devs, utilizando o Git como sistema de controle.

Ela vai facilitar o uso do Git, escondendo alguns detalhes mais complicados de setup. É lá que você provavelmente vai ter seu repositório e usar no dia a dia.

O sistema web que o GitHub possui permite que você altere arquivos lá mesmo, apesar de não ser muito aconselhado, pois você não terá um editor, um ambiente de desenvolvimento e de testes.

Para se comunicar com o GitHub e mexer nos arquivos do seu repositório, você pode usar o comando do git e suas diretivas de commit, pull e push.

Como criar um repositório no GitHub

Criar um repositório no GitHub é um processo essencial para compartilhar seu código com outras pessoas desenvolvedoras. Aqui estão os passos básicos para criar um repositório:

1. **Acesse sua Conta:** Certifique-se de estar logado na sua conta do GitHub. Se você não tiver uma conta, siga as etapas para criar uma, conforme explicado anteriormente.

2. **Página Inicial:** Na página inicial do GitHub, clique no botão "New" (Novo) localizado no canto superior direito.
3. **Nome e Descrição:** Preencha o nome do seu repositório e uma breve descrição. Escolha se deseja que o repositório seja público (visível para todos) ou privado (acessível apenas por convite).
4. **Opções de Inicialização:** Você pode optar por inicializar o repositório com um arquivo README, que é uma boa prática para fornecer informações sobre o projeto. Além disso, você pode escolher uma licença para o seu código, se desejar.
5. **.gitignore:** Você pode especificar tipos de arquivos que o Git deve ignorar ao rastrear alterações. Por exemplo, você pode selecionar uma linguagem de programação específica para gerar um arquivo .gitignore correspondente.
6. **Escolha um Template (Opcional):** Se o seu projeto se encaixa em um dos modelos de projeto disponíveis, você pode escolher um para iniciar com estrutura pré-definida.
7. **Create Repository:** Após preencher todas as informações necessárias, clique no botão "Create repository" (Criar repositório) para criar o seu repositório.

Seu repositório estará pronto e você poderá começar a adicionar arquivos, fazer commits e colaborar com outras pessoas.

Repositórios remotos e locais

No contexto do controle de versão com o Git, é importante entender a diferença entre repositórios locais e repositórios remotos.

- **Repositórios Locais:** Um repositório local é a cópia do seu projeto que reside no seu computador. É onde você faz as alterações, cria commits e mantém o histórico do projeto. Você pode trabalhar offline em um repositório local sem necessidade de conexão com a internet.
- **Repositórios Remotos:** Um repositório remoto é uma versão do seu projeto hospedada em um servidor na web, como o GitHub. Eles são usados para compartilhar seu código com outros desenvolvedores e colaborar em projetos. Você pode enviar (push) as alterações do seu repositório local para o repositório remoto e também obter (pull) as alterações feitas por outras pessoas colaboradoras.

Fontes: <https://coodesh.com/blog/candidates/dicas/portfolio-para-desenvolvedor-dicas-incriveis-de-como-fazer-o-seu/#:~:text=O%20portf%C3%B3lio%20para%20desenvolvedor%20%C3%A9,Portanto%2C%20%C3%A9%20uma%20vitrine%20indispens%C3%A1vel.>

<https://profissoes.vagas.com.br/o-que-e-um-portfolio/>



<https://www.alura.com.br/artigos/o-que-e-git-github#:~:text=O%20Github%20%C3%A9%20uma%20%E2%80%9Ccrede,est%C3%A1%20sendo%20monitorado%20por%20ele.>