

Processo Seletivo de Engenheiro de Testes de Software

As primeiras duas (2) semanas serão voltadas exclusivamente para capacitação dos inscritos, através da realização de cursos e consumo de materiais textuais.

As três (3) semanas seguintes são focadas em pôr em prática os conhecimentos adquiridos anteriormente através do desenvolvimento de documentação e testes, mas também contam com conteúdos teóricos a serem aprendidos.

O laboratório exigirá 20 horas de dedicação semanal para realização de suas atividades, significando um total de 100 (20x5) horas para o processo seletivo.

Cursos:

Para sua formação em Qualidade de Software, existe toda uma linha de cursos na plataforma Alura - [Formação QA: processos e automação de testes](#).

Contamos com um total estimado de 52 horas de cursos. O resto das horas de serviço serão dedicadas aos materiais complementares e ao desenvolvimento dos desafios.

Para este processo seletivo, é necessário que vocês entreguem, ao final de cada sprint, os certificados dos cursos de cada sprint, os desafios descritos nas sprints finais e possa discutir fluentemente sobre os assuntos tratados no material indicado.

Ao final de cada sprint é esperada a entrega dos seus respectivos certificados de curso e/ou desafios.

*Para os primeiros cursos, focar em familiarizar-se com as linguagens de programação e pular partes demasiadamente básicas, caso sintá-se confortável.

**Se já tiver familiaridade com alguma dessas linguagens ou conceitos, pode pular o curso referente, mas deve discutir sobre isso com o gerente da área.

Sprint 1 - Conceitos básicos: Javascript, Python e Git

- JavaScript - Curso COMPLETO com 6 Projetos REAL
- seção 1 e 2
- Curso: [Curso em Vídeo python](#) (4h +- de vídeos (20 primeiros vídeos))
- Curso: [Git e Github na plataforma Udemy](#) (4h)

- entregar um relatório de no máximo 2 páginas sobre o que foi aprendido durante a sprint.

Sprint 2 - Conceitos básicos: Qualidade de Software + Testes Unitários e Funcionais

- Introdução ao desenvolvimento de cenários de teste e histórias.
- Curso: Automação de Testes com Selenium WebDriver em Java
- Curso: [Testes Automáticos + Curso COMPLETO de Teste de Software](#) (5hr)
- Artigo: [Tipos de testes: quais os principais e por que utilizá-los?](#)
- Artigo: [Por que e o que é possível testar?](#)
- Artigo: [Introdução aos Testes de Software](#)
- entregar um relatório de no máximo 2 páginas sobre o que foi aprendido durante a sprint.

Sprint 3 - BDD - Histórias de Usuário & Cenários de Teste (framework)

- Artigo: [Metodologia Ágil - O que é?](#)
- Artigo: [Agile testing - o que é e qual o papel do QA num time ágil?](#)
- Teoria: Testagem manual, relatório de teste e descrever cenários de teste com base em histórias de usuário.
- Desafios:
 - Realizar prática em ambiente controlado da seguinte forma: Testar manualmente a Funcionalidade de *pesquisa de ponto no mapa por endereço* do Software *Serviço de Inteligência Geográfica (SIG)*, em desenvolvimento pelo TerraLab, e então *escrever um relatório de teste sobre os testes realizados, sua análise e conclusão* (de acordo com os exemplos encontrados [aqui](#)). A descrição do software (documento de *Briefing*) e da funcionalidade (História de Usuário referente no Gitlab) a serem testados podem ser encontrados [aqui](#) (assim como o *template* para documentos TerraLab). O site da aplicação a ser testada pode ser acessado pelo [link](#).
 - Também devem ser criados uma série de *cenários de teste*, baseados nas *histórias de usuário* fornecidas (que podem ser encontradas [nesta pasta](#)).
- entregar um relatório de no máximo 2 páginas sobre o que foi aprendido durante a sprint.

Sprint 4 - Qualidade de Software como Serviço (SaaS)

- [Curso: Automação de Testes com Selenium WebDriver em Java](#): (5hr)
- Documentação: [The Selenium Browser Automation Project](#)
- Tutorial: [Como realizar um teste automatizado em aplicativos Web utilizando a IDE do CukeTest? - TerraLAB](#)
- Artigo: [Teste unitário com Jest](#) Automação de Testes com Selenium WebDriver em Java
- Documentação: [Iniciando com o Jest · Documentação Jest](#)
- Desafio: Com base em um software bem conhecido, o [SIG](#), criar um teste para uma funcionalidade específica das áreas de back-end e front-end.
 - Front-End: Teste de Aceitação para o Serviço *Pesquisa de Ponto no mapa a partir de endereço* (Utilize a história de usuário fornecida e o cenário de teste desenvolvido anteriormente).

- entregar um relatório de no máximo 2 páginas sobre o que foi aprendido durante a sprint.

Sprint 5 - Qualidade de Aplicações Móveis Automação de Testes com Selenium

WebDriver em Java

- [Curso docker](#)
 - [Documentação](#)
 - Artigo: [Tutorial de pytest para iniciantes](#)
 - Artigo: [Montando cenários de testes com o Pytest](#)
 - Tutorial: [Automação de testes para backend NodeJS utilizando o framework Jest - TerraLAB](#)
 - Documentação: [Full pytest documentation](#)
 - Fazer o teste funcional da funcionalidade (**Remove Project**)
 - Desafio: Com base em um software da área beack-end do TerraLab, entregar um relatório de no máximo 2 páginas sobre o que foi aprendido durante a sprint.
-

Artigos e textos complementares:

Além dos cursos, para uma formação robusta como Engenheiro de Testes/Quality Assurance é necessária a leitura e compreensão dos seguintes materiais complementares:

Sprint 2

1. Artigo: [Tipos de testes: quais os principais e por que utilizá-los?](#)
2. Artigo: [Por que e o que é possível testar?](#)
3. Artigo: [Introdução aos Testes de Software](#)

Sprint 3

4. Artigo: [Metodologia Ágil - O que é?](#)
5. Artigo: [Agile testing - o que é e qual o papel do QA num time ágil?](#)

Sprint 4

6. Documentação: [The Selenium Browser Automation Project](#)
7. Tutorial: [Como realizar um teste automatizado em aplicativos Web utilizando a IDE do CukeTest? - TerraLAB](#)
8. Tutorial: [Automação de testes para backend NodeJS utilizando o framework Jest - TerraLAB](#)
9. Artigo: [Teste unitário com Jest](#)
10. Documentação: [Iniciando com o Jest · Documentação Jest](#)
11. Artigo: [Tutorial de pytest para iniciantes](#)
12. Artigo: [Montando cenários de testes com o Pytest](#)
13. Documentação: [Full pytest documentation](#)

Sprint 5

14. Tutorial: [Automação de testes para iniciantes: Configurando os ambientes de desenvolvimento e teste em dispositivos Android - TerraLab](#)

Cursos complementares (não obrigatórios):

- [Formação QA: processos e automação de testes](#) (outros cursos além do previamente citado se tratam de tecnologias não utilizadas no Lab, como Cypress, ou coisas presumidamente já conhecidas ou facilmente aprendidas através dos conteúdos do nosso Blog, como o uso do GitHub/GitLab e lógica básica de programação, mas podem lhe ser úteis como material de referência)
- [PHP e Selenium: testes de ponta a ponta](#) (contém conceitos interessantes porém PHP não faz parte do nosso escopo de desenvolvimento de testes)
- [Selenium WebDriver e C# parte 1: primeiros passos](#) (contém alguns conhecimentos aplicáveis, mas não utilizamos C# no Lab atualmente)
- [Testes no Android: Mocks e integrações](#) (parte da Formação em testes automatizados Android)
- [Python para Iniciantes - Aula Udemy](#)
- [JavaScript - Curso COMPLETO com 6 Projetos REAIS](#)
- [Curso de JavaScript e TypeScript do básico ao avançado 2022](#)
- [Testes funcionais de aplicações Android com Appium](#)
- [All about NodeJS | Udemy](#)
- [Programação em JavaScript - Cursos disponíveis na plataforma Alura](#)