

Os casos de teste que se seguem foram escolhidos com base na análise dos riscos associados ao GitHub, estas funcionalidades pareceram-me mais críticas para o bom funcionamento da aplicação. Estas funcionalidades trazem mais valor para o utilizador, se por alguma razão existirem falhas, pode ter um grande impacto nos utilizadores do GitHub.

### **1. Login no GitHub com credenciais erradas**

De modo a testar a segurança do login, não foi criado um teste para validar que o utilizador entrou no GitHub, mas sim, que obteve erro ao tentar entrar na página. Existe uma validação na página de login, de modo a verificar se a página web alertou o utilizador que o user e/ou password estavam errados, e não efetuou o login na página.

Assim é possível verificar que existe um alerta para o utilizador quando as credenciais estão erradas, e a página principal do GitHub não é apresentada.

**Impacto:** Se um utilizador conseguir entrar na página com credenciais erradas, teria um grande impacto para o utilizador e para os gestores do GitHub. Digamos que seria um desastre em termos de segurança, para todos os utilizadores, uma vez que poderia haver pessoas a entrar na conta de outras com credenciais erradas.

### **2. Criar um repositório no GitHub**

Sendo o GitHub uma plataforma para guardar código-fonte, arquivos e realizar o controlo de versões, um teste de criação de repositório era essencial. Este teste, valida se um repositório foi criado com sucesso.

**Impacto:** Se por algum motivo, não fosse possível criar repositórios a qualquer momento no GitHub, levaria a que os utilizadores analisassem outras plataformas para o mesmo efeito. Isto é algo que nenhuma plataforma pretende que aconteça, para isso, é importante testar a criação de repositórios.

### **3. Validar criação de repositório com um nome existente**

Uma vez que não é possível adicionar um repositório com um nome existente, este teste serve de validação de erro, quando existe a tentativa de adicionar um nome que já existe noutros repositórios.

**Impacto:** Se fosse possível criar um repositório com um nome que já existe, iria gerar muita confusão aos utilizadores, olhando para o nome e não saber ao certo que ficheiros estão nesse repositório. Isto é algo que não pode acontecer, sendo que deve ser testado sempre que possível.

### **4. Eliminar repositório GitHub**

Como existe o teste de criar repositório, também é importante validar que os repositórios são eliminados com sucesso.

**Impacto:** Se um utilizador tiver vários repositórios, e não conseguir apagar os que pretende, vai gerar confusão na organização dos repositórios, e dúvidas em relação a repositórios que já não estão em uso. Para evitar estas situações, é importante que seja possível eliminar um repositório a qualquer momento.

## 5. Bifurcar um repositório GitHub

Este teste pesquisa um repositório, e realiza o fork sobre o mesmo. Esta funcionalidade é muito importante, no que diz respeito a partilha de projetos. É possível várias pessoas trabalharem sobre o mesmo projeto, sem comprometer o projeto inicial.

**Impacto:** Caso não fosse possível bifurcar um repositório, levaria a problemas de gestão de código, sendo que todas as pessoas associadas ao projeto, caso não conseguissem fazer fork, consequentemente iriam trabalhar no projeto inicial, e poderia levar a erros em funcionalidades, esmagamento de código entre commits, problemas de versões.

## 6. Criação de Issues

A criação de issues é importante no que diz respeito à organização, pois com os issues criados é possível acompanhar erros que aconteceram ou solicitações.

**Impacto:** Caso não fosse possível criar issues, haveria problemas de organização, pois, ninguém saberia os problemas existentes, necessidades do projeto. Com o recorrer às issues, até pode ser mais rápido a resolução dos temas solicitados, sendo que esta funcionalidade deve estar sempre disponível e sem falhas.

## 7. Eliminar ou fechar uma issue

Se uma issue é criada, a sua eliminação ou fecho é igualmente importante.

**Impacto:** Numa situação em que estão abertos issues, e não é possível fechar ou eliminar pode trazer alguns problemas. Pode existir um issue que está tratado, mas por algum motivo não foi possível fechar, o que leva, a que uma pessoa pegue no tema, pensando que o problema ainda não foi resolvido. Isto é algo que não deve mesmo acontecer, até porque estaria uma pessoa a gastar tempo em algo desnecessário.

## 8. Editar um ficheiro e fazer merge para a branch principal

Este teste realiza o commit para uma nova branch de uma alteração num ficheiro, prosseguir com um pull request e adicionar as alterações ao ramo principal. Este teste é importante para validar que é possível realizar merge entre branches, quando não existem conflitos. Validado também a criação de nova branch.

**Impacto:** Numa situação onde não é possível a realização das alterações para a branch principal, vai gerar problemas. O código fica desatualizado, á medida que os utilizados realizam

novas alterações, podem criar novas branches o que pode cair no esquecimento o merge de uma branch antiga. Devido a este factor, é necessário validar que esta funcionalidade se encontra disponível, e que as alterações foram adicionadas a branch principal.

#### **9. Validar proteção de merge em branch específica**

Este teste valida a existência de restrições nas configurações de uma branch específica, valida se é possível realizar um pull request.

**Impacto:** Se as restrições de branch não estivessem a ter o comportamento esperado, era possível qualquer pessoa fazer o merge de uma branch para a principal, o que levaria a problemas de código, faltando revisão do mesmo e validando que a estrutura do código é a pretendida. Com este teste conseguimos validar que não é possível realizar o merge para a branch principal, antes de existir um review.

#### **10. Criar projeto com template específico**

Este caso de teste destina-se á criação de um projeto, em que o template selecionado é específico, de modo a organizar e priorizar o trabalho com o template pretender.

**Impacto:** Estando esta funcionalidade inacessível, poderia gerar várias versões diferentes de elementos comuns a vários projetos, gerando assim dificuldades de gestão e trabalho.