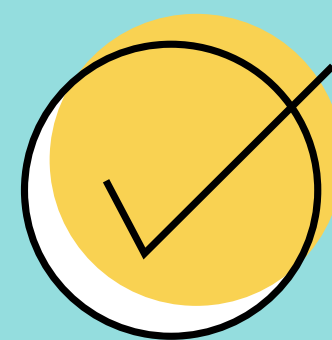


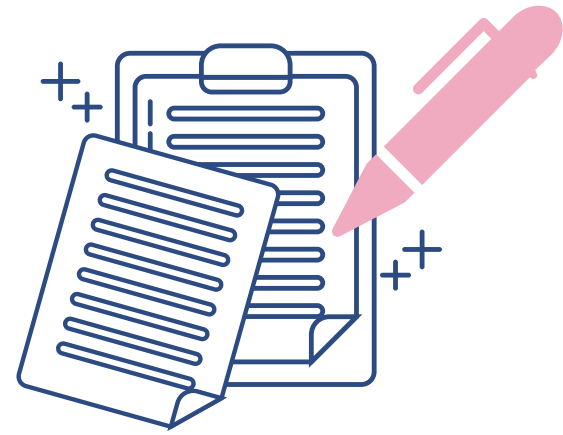
PRECISAMOS ENTENDER



VÁRIOS TIPOS



DE TESTE



IMPORTANTE

- **Testes Funcionais do Sistema** PLANEJAMENTO, CENÁRIOS E CASOS DE TESTE
TÉCNICAS DE TESTE E USO DE VÁRIOS TIPOS DE TESTE
- **Layout do Sistema** INSPECIONA OS ELEMENTOS IMPLEMENTADOS, ALTERADOS,
CONFERE/INSPECIONA (DEVTOLLS)
- **USABILIDADE / UX | IU** COMPORTAMENTO DO SISTEMA - É MAIS ESPECÍFICO,
EXISTEM HEURÍSTICAS (EX: NIELSEN) E ROTEIROS PARA ESSES TESTES



Tipos de testes no geral:

Para na hora da entrevista

VOCÊ FARÁ TESTES

- Testes Funcionais
- Testes Layout
- Testes de Usabilidade - UX/IU
- Testes Exploratórios
- Testes de Regressão
- Testes de Confirmação
- Teste baseado em risco

É BOM CONHECER CADA UM!

Você faz:

- Gestão de Bugs

Você tem o seu:

- Fluxo de teste

Ah é bom você:

- Fazer o mapeamento do sistema, e entender cada fluxo de teste.

ROTEIRO DE TESTES DE USABILIDADE - IU



verificar a facilidade que o software ou site possui de ser claramente compreendido e manipulado pelo usuário.

Teste de usabilidade é uma técnica de caixa-preta. O objetivo é observar usuários reais usando o produto para descobrir problemas e pontos de melhorias. O produto, que pode ser um site, uma aplicação web, um produto físico, não precisa estar completamente desenvolvido.

Teste de usabilidade serve para observar o uso de um produto e investigar questões que envolvem navegação e entendimento da interface.

#1 - Ativação - Validar a ativação após interação com cursor ou evidenciar a ação ao selecionar um item ou botão.	Referência - Mouseover, checkbox marcado e desmarcado, coloração de ícones de status, mudança de cursor com links, ícone favoritos e semelhantes, Menu hambúrguer
#2 - Único/Sobreposição de elementos - Validar que o elemento em evidência ocupa apenas a camada que foi designado e o conteúdo da página é exibido atrás fora do foco	Referência - Modais newsletter, login [..], menu <header, footer>, campo de busca
#3 - Seleção - Validar que a seleção fica visível no momento exato conforme tempo estipulado e desaparece no tempo/modo necessário	Referência - Mensagem informativa configurada para ser exibida por determinado tempo e sumir automaticamente.
#4 - Removível - Validar que é possível remover um item ou ter a liberdade de ir e vir . Abortar uma tarefa, ou desfazer uma operação/ação	Referência - Botão de fechar, voltar, mouseover deve terminar a animação quando não tiver interação com cursor
#5 - Flutuante - Validar que ao selecionar item flutuante e mover a barra de rolagem o elemento segue a rolagem, ou segue flutuando na página	Referência - Menu do header, call to action de chat, filtros laterais no desktop, botão de comprar no mobile
#6 - Colisão - Validar que ao clicar em qualquer função a experiência é suave e o objeto não 'Colide' com outro no seu surgimento ou desaparecimento	Referência - Ao ser exibido, o elemento não deve empurrar outros elementos e nem cobrir. Deve respeitar camadas. Não pode parecer que tá coledindo.

#7 - Visibilidade do status - Validar a ativação após interação com cursor ou evidenciar a ação ao selecionar um item ou botão.	Referência - loading, salvando, imprimindo, em progresso, carregando, aguarde...
#8 - Responsividade - Validar que ao aumentar e diminuir a página, mudar as dimensões - há o mesmo comportamento, ou adequação dos elementos	Referência - Não deve quebrar layout em dimensões diferentes, ao aumentar ou reduzir a tela.
#9 - Browser/Compatibilidade - Validar COMPATIBILIDADE: navegadores diferentes:	Referência - Chrome, Safari, FireFox, Android, iOS...
#10 - Consistência e padrões - Cores, Fontes, Layout... tem um padrão. Todo o site deve manter o padrão de layout	Referência - A página deve ter todos os elementos do protótipo. Um padrão
#11 - CRUD - Criar, Visualizar, Editar, Excluir	Referência - Deve ser possível essas ações sempre que couber
#12 - Espaço Vazio - Validar que os campos a serem preenchidos tem um apontamento do que é para ser feito, não estão em branco.	Referência - Mensagens de instrução de senha, placeholder, títulos dos campos, indicação de obrigatoriedade, campo do e-mail, data...

#13 - Expansão - Validar que ao expandir objetos não há deformação de elementos	Referência - Categorias /muitas subcategorias...
#14 - Repetição - Ações repetidas parecem simples?	Referência - Nome, senha, login, logout
#15 - Recuperação de Erro - Validar que posso me recuperar de qualquer ação	Referência - Desistir, voltar...

ROTEIRO DE TESTES DE USABILIDADE-IU



Teste de usabilidade serve para observar o uso de um produto e investigar questões que envolvem navegação e entendimento da interface.

TESTES FUNCIONAIS

CAIXA PRETA

- Imagine testar as FUNCIONALIDADES desta tela de "Cadastro de Usuários"



Cadastro de Usuários

Dados do Usuário

Nome

CPF

Data de Nascimento

E-mail

Ativo?
☒ Sim ☐ Não

Perfil
- Selecione -
Selecione
AUXILIAR
PREPOSTO

R.G.
 -

Código SSP

Confirmar E-mail

 CANCELAR  SALVAR

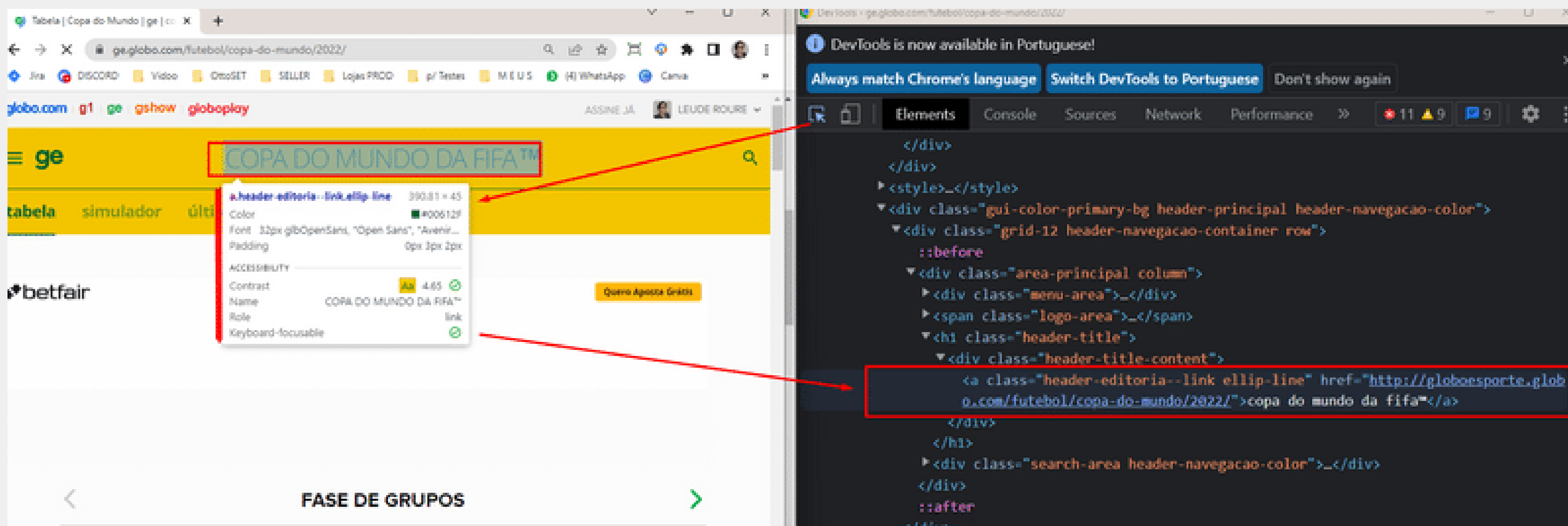
São os testes definidos de acordo com os requisitos funcionais do software. Como não há conhecimento sobre a operação interna do programa, o analista concentra-se nas funções que o software contemplará. Baseado na especificação determina-se as saídas que são esperadas para um determinado conjunto de testes.

VALIDAÇÃO LAYOUT

É para validar que o layout construído está de acordo com o protótipo

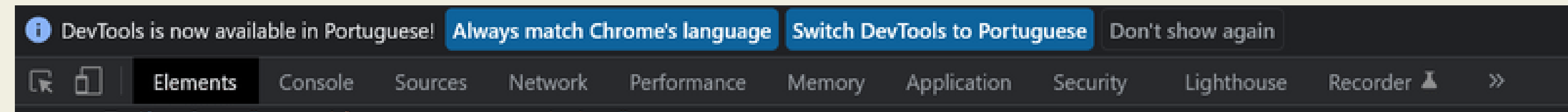
- Aqui você usa normalmente o FIGMA - para visualizar o protótipo
- e outra ferramenta de INSPEÇÃO DE ELEMENTOS, eu usava o DEVTOOLS (podemos falar sobre isso depois)

dessa forma consigo "inspecionar" cada elemento da tela

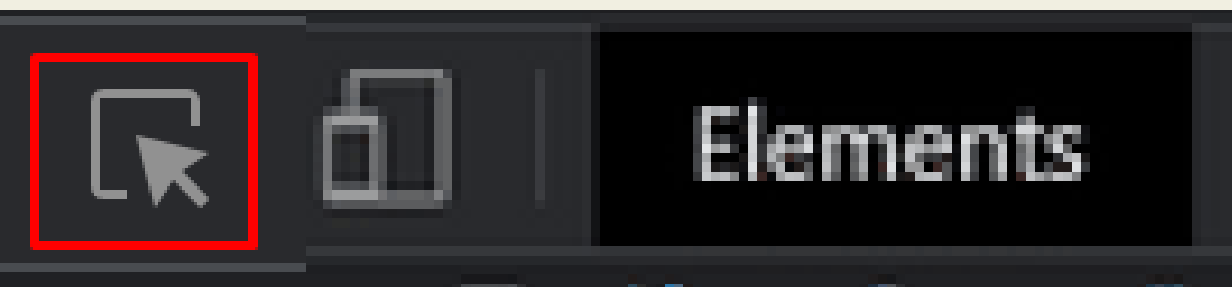


Você precisa conhecer o ferramenta de desenvolvimento do navegador

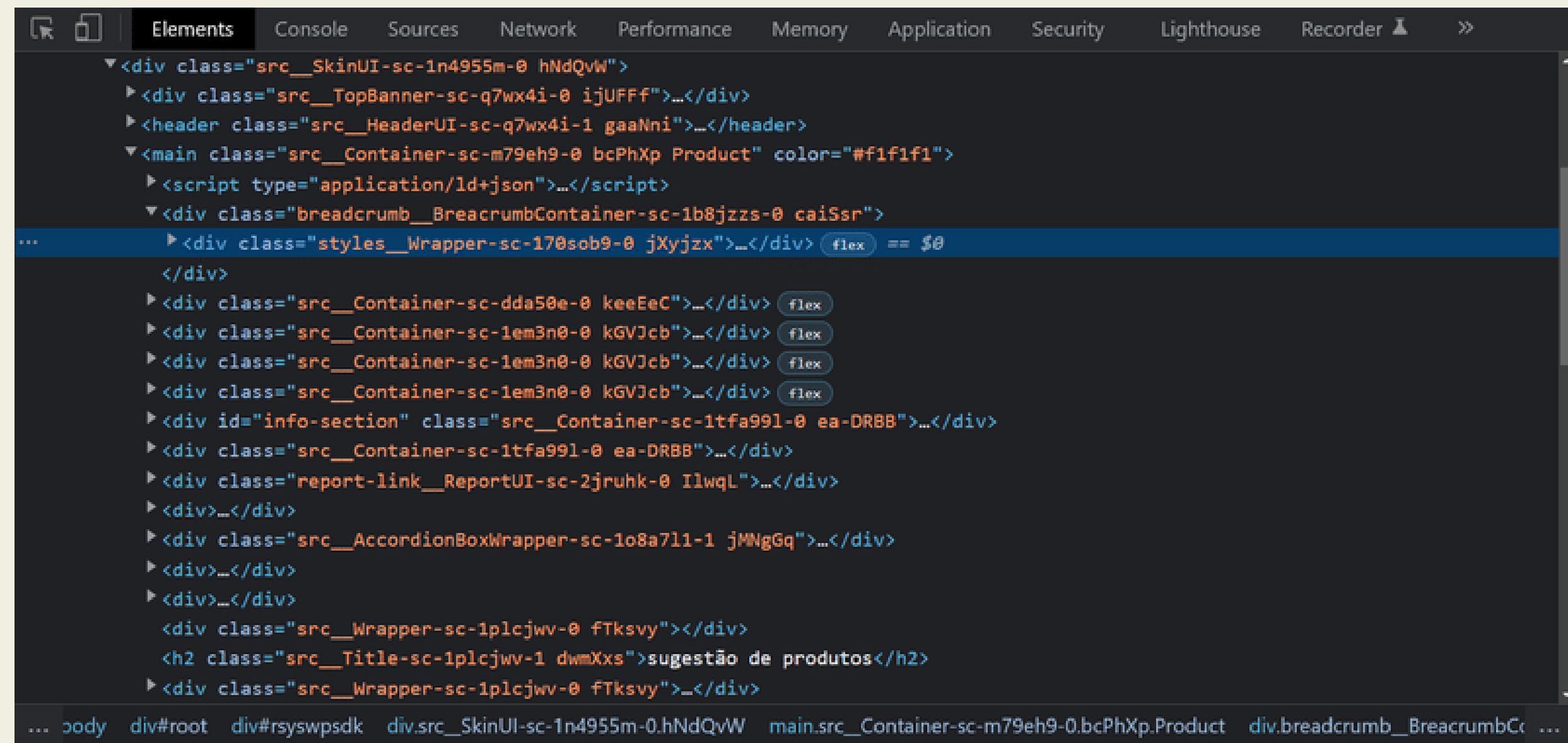
- No site aberto você aperta <F12>



As que mais usamos é



Pra Inspeccionar os elementos



Outros tipos de testes:

CTFL EXPLICA DETALHADAMENTE

Confirmação / Reteste

- Depois da correção testar todos os casos de teste que falharam durante a última execução
- Verifica o sucesso das ações corretivas
- Eu preciso realizar testes para identificar que aquilo que foi mexido, continua funcionando.

Regressão

No desenvolvimento partes que não foram mexidas podem 'quebrar'

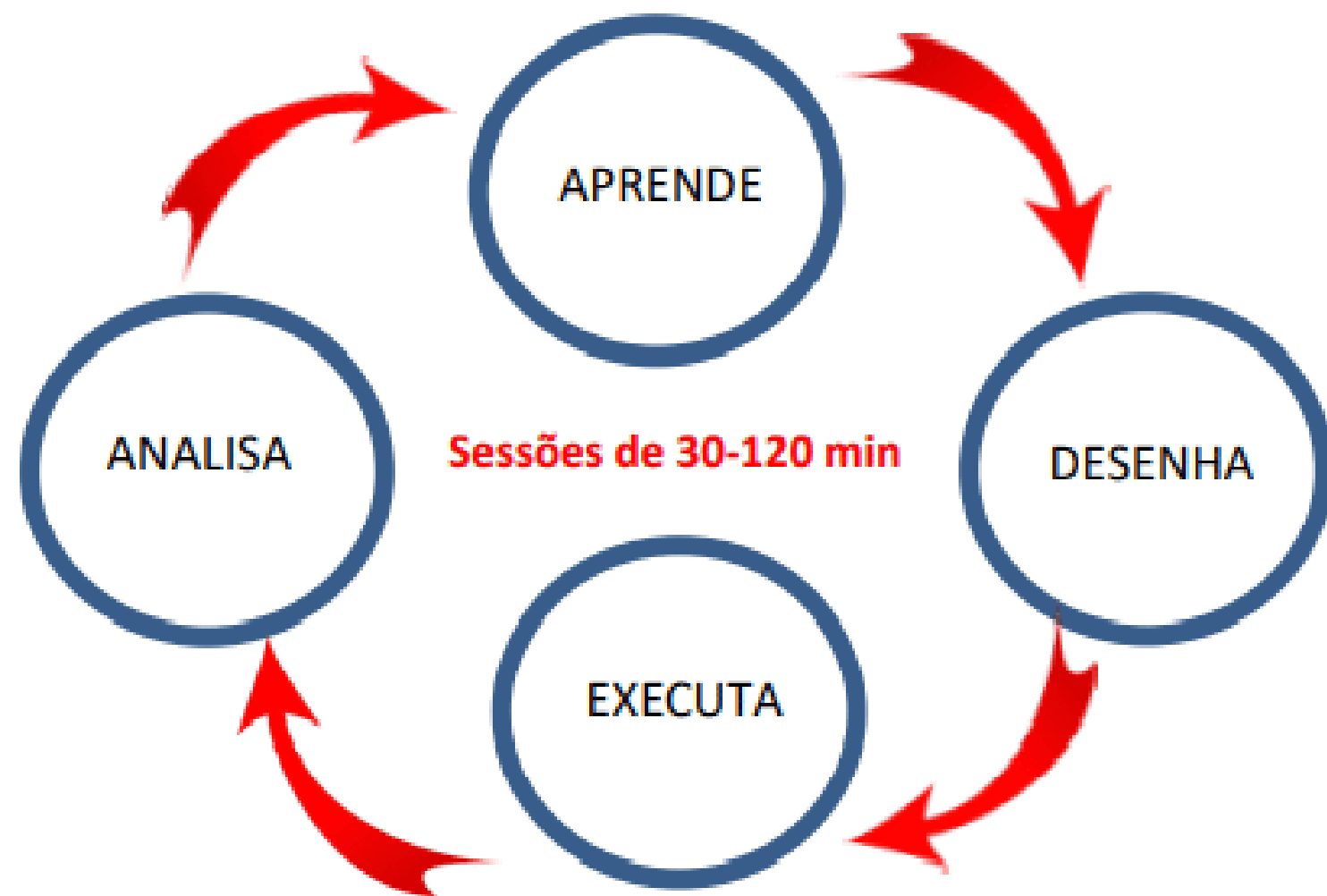
- Por isso preciso fazer testes pra detectar possíveis efeitos colaterais
- Quando testo outras regiões diferentes das modificadas



EXPLORATÓRIO

Testes exploratórios são extremamente úteis para encontrar bugs rapidamente.

- Aqui não há Regra de Negócio, por isso é muito usado quando há pouca documentação. Aqui você explora antes e depois analisa. Quando você está olhando a aplicação, você não faz aleatoriamente (clica aqui, clica ali) deve haver um passo a passo.



EXERCÍCIO:

1. HIPÓTESE
2. PLANEJO/DESENHO
3. EXECUTO
4. ANALISO
5. APRENDE
6. REPITO

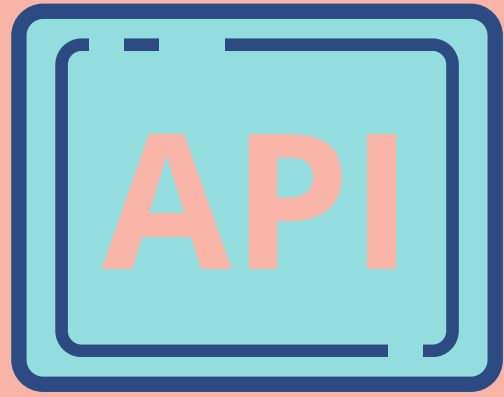
- Campos obrigatórios;
- Habilitar / Desabilitar formulários;
- Interrupção da ação;
- Quebra de fluxos;
- Usabilidade dos menus;
- Estouro de campos.

Teste de Mutação

Vai no código introduz propositalmente um Bug e testa de novo se passar é porque seu teste está falho

Smoke Testing

Teste rápido (de fumaça), são testes básicos que verificam as funcionalidade básicas da aplicação, para determinar se o build ou a versão da aplicação realizada está estável ou não. É uma confirmação para a equipe decidir se avança ou não para novos testes. Consiste em um conjunto mínimo de testes para validar as principais funcionalidades do passado estão Ok.



Teste de Serviço

API - TESTE DE CONTRATO

(nesta camada são executados teste como API, Integração e Contrato)

API Teste unitário e de (UI) User Interface

Validam múltiplos cenários e garantem que o conteúdo do JSON de retorno = CORRETO

TESTE DE CONTRATO

Na fase de serviços (services e microservices),

Garantir a integridade dos dados na comunicação: client/server

Verificação da validade dos MOCKS (simulados - serviço x consumidor)



Porque você conhece, você começa a PENSAR no que pode dar errado

TESTE BASEADO EM RISCO

- Pensar nos riscos, é uma forma de "pensar" os testes:

RISCOS DE PRODUTO: o que será desenvolvido.

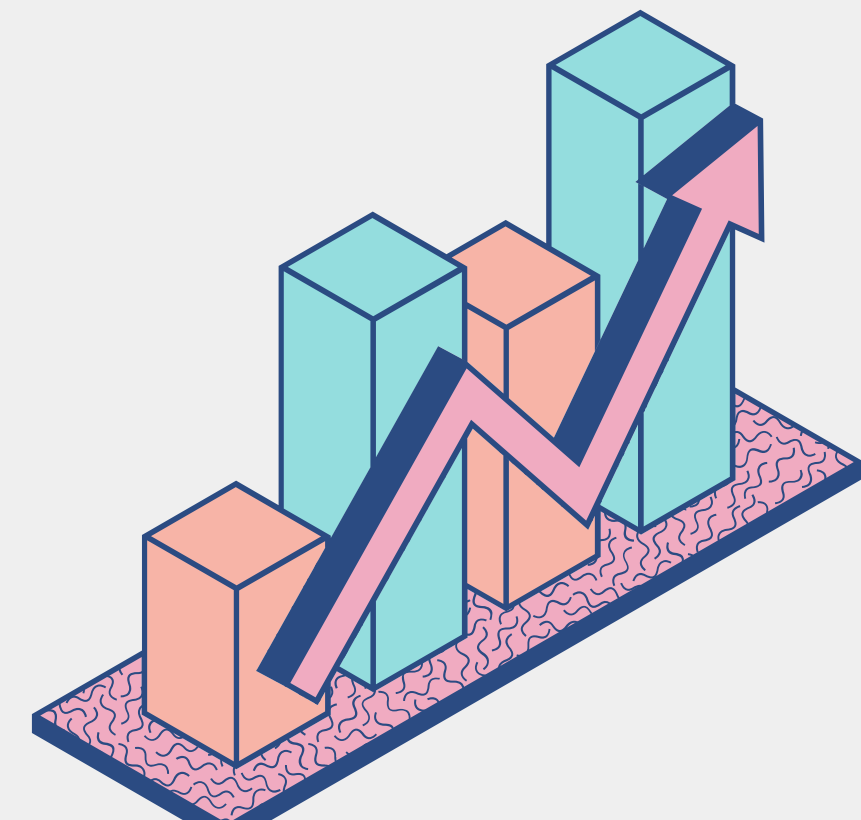
Relacionados diretamente ao sistema: O QUE É TESTADO

RISCOS DE TRABALHO: Gerenciais e Controle.

Relacionados diretamente ao time, aos prazos....

CTFL

O que nós QAs mais fazemos é LEVANTAR RISCOS. Porque a mentalidade está voltada para trazer qualidade. É muito importante saber, e pensar neles - antecipar, e comunicar esses riscos de modo assertivo. Você, e não outro - mas você se posiciona quanto à qualidade.



A person wearing a grey sweater is sitting and writing in a spiral-bound notebook. The notebook is open, and the person's hands are visible, holding a pen. The background is a light blue wall.

IMPORTANTE

REGRAS DE NEGÓCIO e REQUISITOS FUNCIONAIS

PRECISA CONHECER A FUNDO

Requisito funcional refere-se AO QUE o sistema deverá fazer, enquanto a Regra de Negócio refere-se a COMO o sistema deverá fazer.

CTFL



CTFL

LEVANTAR RISCOS SOB A PERSPECTIVA DO USUÁRIO FINAL

PROBABILIDADE X IMPACTO

A **mitigação** é o termo técnico para a tentativa de impedir que o risco se torne um problema em potencial. A **Contingência** é sobre o que eu faço depois que o risco acontece.

CLASSIFIQUE O RISCO A PARTIR DE UMA TABELA:

		ANÁLISE QUALITATIVA				PLANO DE RESPOSTA		
ID do Risco	Descrição do Risco	Probabilidade	Impacto	Grau	Quem	Mitigação	Contigência	Status

Converse com a equipe, ninguém melhor do que eles para saber a severidade dos riscos

FEITO ISSO PRIORIZE OS RISCOS - TESTES



CRUD

teste geral

4 Operações básicas Testes

Ordem ideal para seguir nos fluxos dos testes

1 CRIAR (CREAT)

- Salva corretamente
- Fácil acesso/uso
- Permite cancelar a ação (fácilmente)

2 VISUALIZAR/CONSULTAR (READ)

- Título da página
- O vazio nunca deve estar vazio
- Dados importantes visíveis
- Síntese
- Ordenação dos dados
- Ajuda e pesquisas
- Recuperação de erro
- Mensagens batem com função

3 EDITAR/ATUALIZAR (UPDATE)

- Atualizações são salvas corretamente?

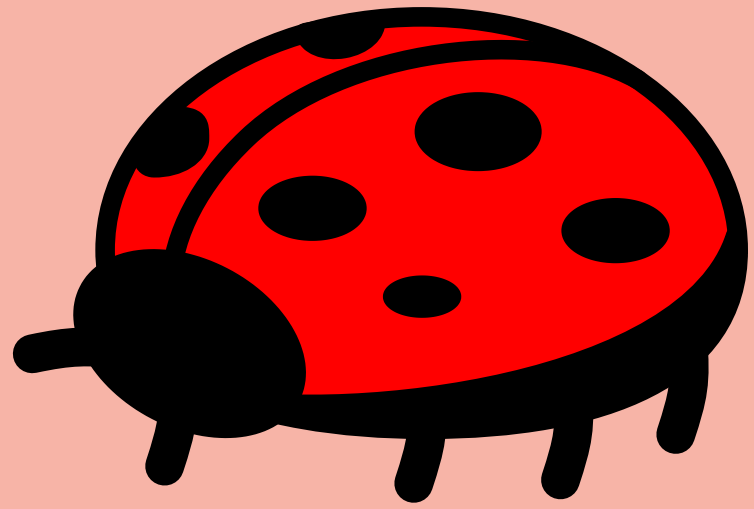
4 EXCLUIR (DELETE)

- Mensagem de alerta e ação de confirmação
- Ação cancelamento

5 ACRESCENTO:

SEMPRE DEIXAR O CAMPO EM BRANCO

FUNCIONALIDADES - COMPORTAMENTO



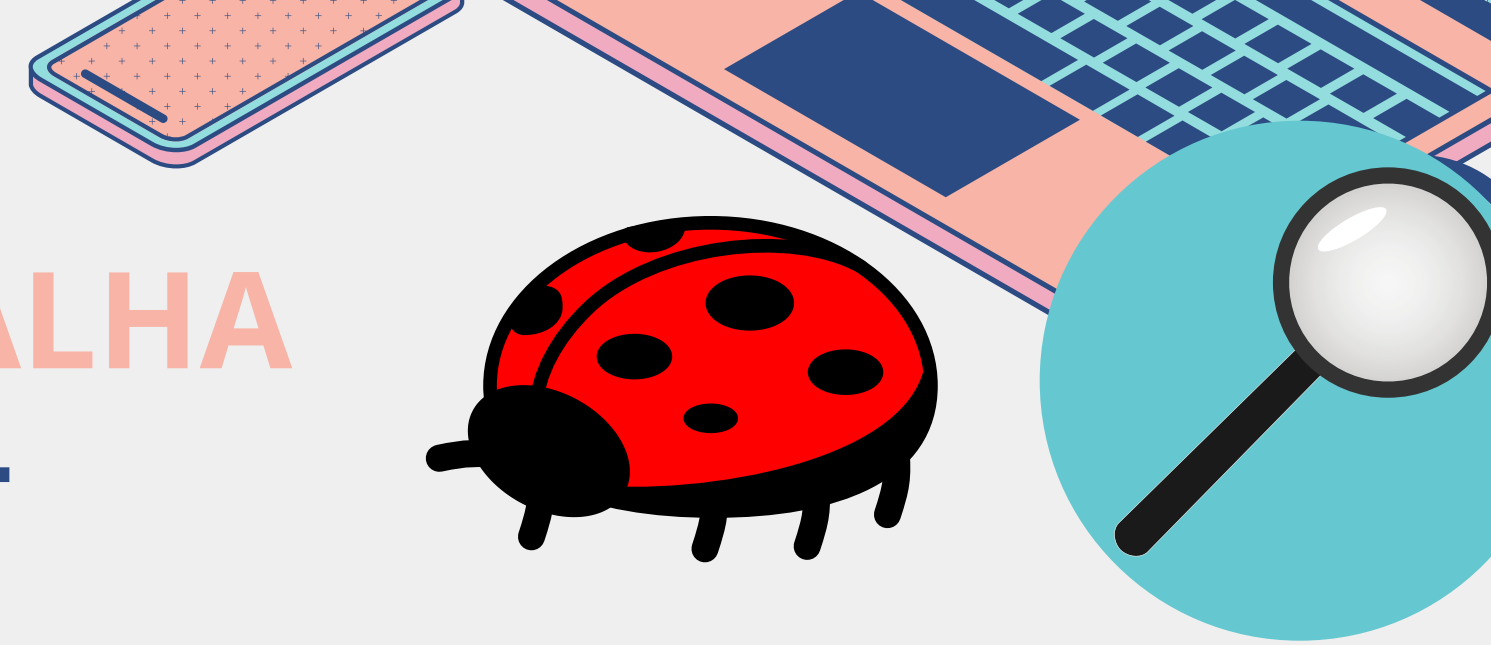
**O que acontece se você rodar o seu teste
e encontrar uma inconsistência?**

Gestão de Bugs





BUG ERRO - DEFEITO - FALHA
Veja esse conceito na CTFL



ATENÇÃO

- ✓ **Você não só abre o Bug - Você acompanha para ver a correção - e Retestar**
- ✓ **Só registre o Bug depois de conseguir reproduzir o comportamento**
- ✓ **Veja como o Sistema gera o Bug - Identifique a causa**



- SAIBA EVIDENCIAR O ERRO E EXPOR ISSO
- SAIBA CLASSIFICAR A COMPLEXIDADE DA ATIVIDADE EXECUTADA
- TUDO DEVE LEVAR A SOLUÇÃO DO PROBLEMA

**Saiba falar sobre
isso na entrevista**



**O DETALHE
TRAZ A
EXCELÊNCIA**

EVIDENCIE TUDO

→ o registro de suas atividades é a
segurança futura que você necessita

OS TESTES EXECUTADOS

PARA MONITORAMENTO, CONTROLE, RETESTE - RASTREIO

INCONSISTÊNCIAS / BUGS ENCONTRADOS

CLASSIFIQUE A GRAVIDADE E IMPACTO, SEJA CLARO E SUSCINTO

EVIDÊNCIAS

PRINTS (PEGUE A URL, O HORÁRIO, A DATA - ISSO É SEGURANÇA PARA VOCÊ)

VÍDEOS

LOGS



**NO FINAL TRAREI AS FERRAMENTAS
QUE USO PARA ISSO**

Níveis de Teste

CTFL



Teste de Unidade

- Códigos
- Arquitetura
- Fluxos de trabalho
- Fluxo de dados do Sistema

Encontrar falhas de funcionamento dentro de uma pequena parte do sistema, isolando parte desse sistema para teste. Feitos, no geral, pelos Dev. Mas o QA traz a cultura desses testes serem implementados, traz a mentalidade de se testar o quanto antes.

Teste de Integração

- Um exemplo são os Mocks

INTERFACES - testa as interações dos componentes unitários - entre si. Módulos que são integrados e testados em grupo. acessando um banco de dados ou fazendo uma chamada externa a outros sistemas:

Uso de mocks - Quando você depende de uma resposta de um site de terceiro, mas você não tem como garantia a resposta que vai acontecer. Então você cria um objeto, simula e faz o seu teste ali de forma controlada.

Teste de Sistema

O cliente avalia o sistema e após a experiência de utilização tem a responsabilidade de aceitar ou não o sistema que foi entregue e pelo qual foi pago.

Teste de Aceitação

O cliente avalia o sistema e após a experiência de utilização tem a responsabilidade de aceitar ou não o sistema que foi entregue e pelo qual foi pago.

PIRÂMIDE DE TESTES

SOBRE TESTES AUTOMATIZADOS

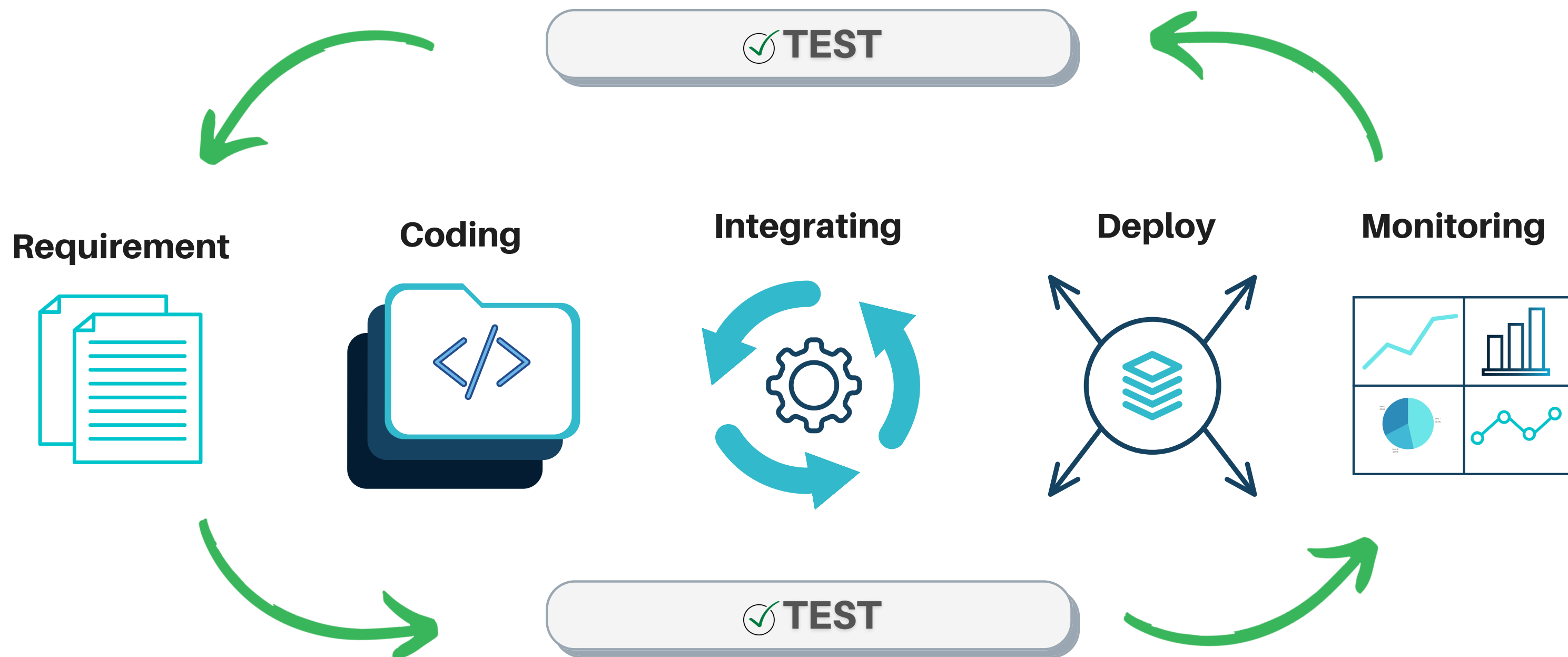


Mostra como deveria ser a organização dos nossos testes

Continuous Testing

ESSE É OUTRO CONCEITO, MAIS COMPLEXO, DEPOIS/MAIS NA FRENTE BUSQUE ENTENDER

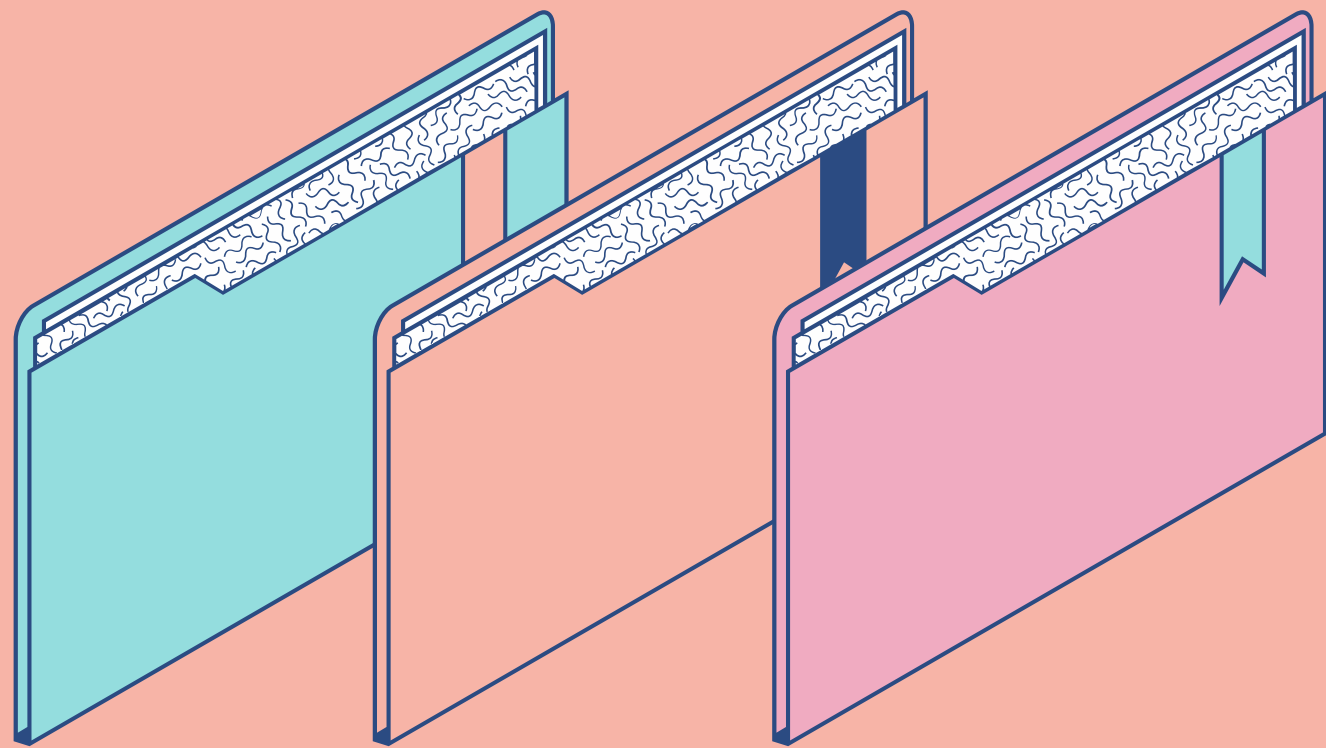
LEUDE ROURE | QA | SM



Shift Left ← Teste em todas as etapas, sendo trazidos cada vez mais para o início

→ **Shift Right**

É ISSO, VOCÊ ESTÁ CAMINHANDO



PARA CUMPRIR

" AS NECESSIDADES DAS ENTREGAS "

E EVIDENCIAR

" AS NECESSIDADES DO USUÁRIO FINAL "



BUSQUE MELHORIA CONTÍNUA

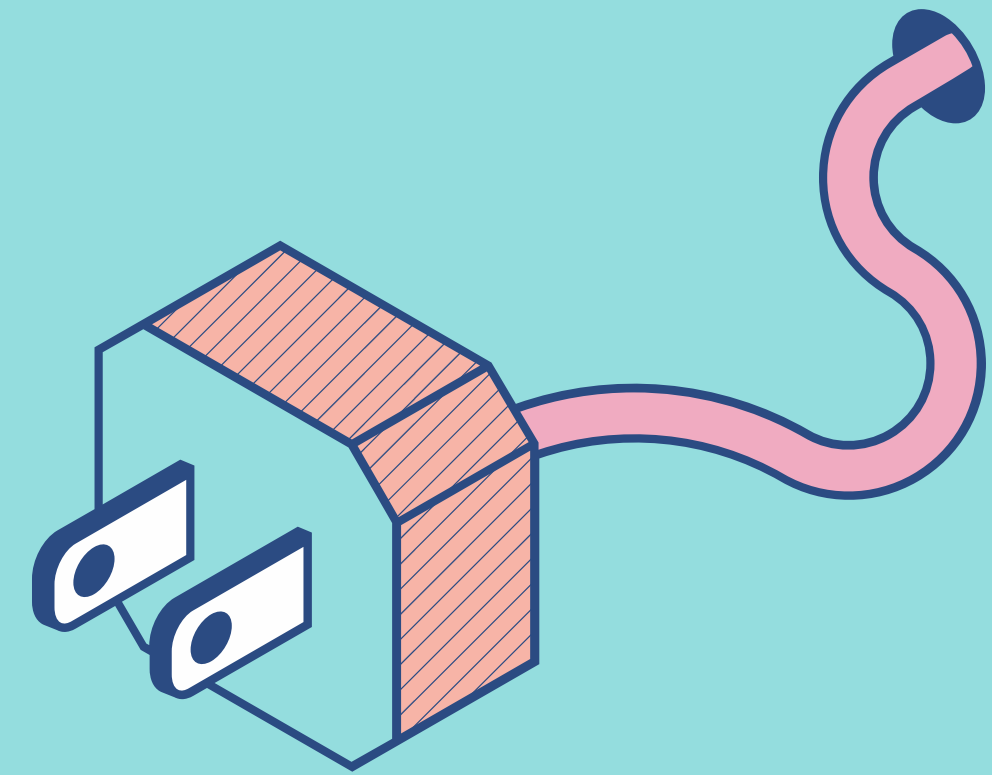
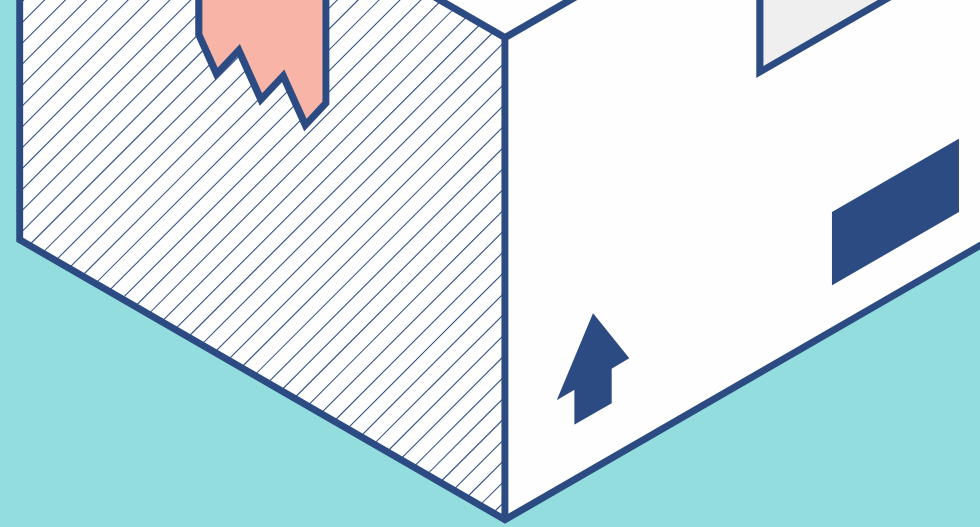
MAPEIE ESTRATÉGIAS - CRIE METAS

ESCUITA ATIVA - COMUNICAÇÃO OBJETIVA

**CELEBRE SEU CRESCIMENTO PESSOAL
PROFISSIONAL**

Escolha um site

- e treine o que estudou aqui
- se veja fazendo essas coisas
- gere isso na sua mente por meio de treinos



ATIVE O QUE AINDA É TEORIA

Pensa em 2 semanas de Sprint

E um cliente que quer parar de andar a pé
Como estaria a Task, como seriam as reuniões, e os
seus testes...

Planeje, Faça Cenários e se possível os Casos de Teste

