

Avaliação Parcial - Testing I [Turma 2]

Esta avaliação é o nosso primeiro checkpoint da disciplina e é relacionada aos conteúdos abordados durante a primeira parte da matéria.

- A duração é de 1h e 40 minutos.
- Os alunos deverão responder, de forma INDIVIDUAL, este formulário que contém: 12 perguntas, sendo 4 múltipla escolha e 8 abertas.
- Após enviar as respostas não será possível editá-las.

Boa prova!

gustavobarretto@gmail.com [Alternar conta](#)

 Rascunho salvo.

A foto e o nome associados à sua Conta do Google serão registrados quando você fizer upload de arquivos e enviar este formulário.. Só o e-mail informado por você faz parte da sua resposta.

***Obrigatório**

E-mail *

gustavobarretto@gmail.com

Qual é a proposta do Teste de Regressão? (0,5 pts)

- ☐ Confirmar que um componente está funcionando de acordo com o especificado.
- ☒ Garantir que as modificações feitas no software não introduziram nenhum defeito.
- ☐ Evitar que uma tarefa seja completada quando ainda não foi finalizada.
- ☐ Verificar se houve sucesso nas ações de correção.
- ☐ Garantir que as funcionalidades principais do software estejam funcionando.

Limpar seleção

 [Pedir acesso para editar](#)

Imagine que você seja o testador designado para testar a funcionalidade de login do Facebook. Escreva um caso de teste positivo e um caso de teste negativo. Não é necessário escrevê-los em um template, mas caso queira pode utilizar o template criado em sala. Tire print da tela e faça o upload aqui. (2,5 pts)

 Adicionar arquivo

 Pedir acesso para editar



Para cada um dos testes abaixo selecione o ambiente em que eles podem ser executados: (1,0 pt)

	Desenvolvedor (DEV)	Testes (QA)	User Acceptance (UAT)	Staging/Preparação (STG)
Teste de fumaça (Smoke Test)	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Testes de carga, estresse e escalabilidade	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Testes de compatibilidade	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Testes de Integração	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Testes de casos de uso	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Testes de Sistema	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Testes de segurança	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Testes de acessibilidade	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Testes exploratórios	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Testes de migração de dados	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

Limpar seleção



Pedir acesso para editar



Você está criando um modelo de Relatório de Defeitos para sua empresa. Quais são os campos que você deve incluir? Lembre-se que nem todos são obrigatórios, porém alguns são imprescindíveis. Descreva a finalidade de cada um dos campos. (1,5 pts)

São imprescindíveis em um Relatório de Defeitos: i) a identificação exclusiva dos bugs, com ID único de rastreamento e identificação dentro do sistema a ser testado, facilitando o tratamento do erro e evitando a redundância de testes; ii) a descrição específica do que está ocorrendo, estabelecendo minuciosamente todas as circunstâncias que envolvem o defeito, levando, assim, aos dois últimos essenciais requisitos adiantes; iii) a descrição onde a falha está sendo reproduzida, uma vez que o defeito só é considerado defeito se for reproduzido pelo software; iv) a narrativa de todas as etapas a serem realizados para a reprodução do defeito, descrevendo cada passo que deve ser realizado para que ele apareça no sistema.

Observe as declarações abaixo e marque a opção CORRETA: (0,5 pts)

- I. Os Testes Funcionais permitem validar o funcionamento de algo.
- II. Os Testes Não-Funcionais são considerados de Caixa Branca.
- III. Há uma regra que diz que o Teste de Regressão deve ser executado apenas uma vez durante a Sprint.
- IV. O Teste de Componente não precisa, necessariamente, envolver código.

- ☐ Todas as afirmações estão incorretas.
- ☐ As afirmações II. e IV. estão corretas.
- ☒ Todas as afirmações estão corretas.
- ☐ Apenas as afirmações I. e III. estão corretas.
- ☐ Apenas a afirmação I. está correta.

Limpar seleção



Pedir acesso para editar



Descreva a diferença entre testes de caixa-preta e testes de caixa-branca. Dê um exemplo de cada. (1,5 pts)

O teste de Caixa Preta é baseado, principalmente, pela equipe de QA, cuja análise independe de qualquer conhecimento da estrutura interna do software, mas tão somente conhecimento acerca dos seus requisitos. Nesse teste, o testador realiza testes relativos aos próprios requisitos do software, especificações de uso e casos de uso, verificando possíveis falhas (retorno não esperado do software) ou defeitos (retornos inesperados no estado intermediário do software). Neste diapasão, pode-se incluir, a título ilustrativo, a maior parte dos testes pertencentes ao QA, como teste de aceitabilidade, cenários e casos de uso de usuários, análise de limites, particionamento de equivalência e todo e qualquer teste que envolva a interação um possível usuário e o software.

No que diz respeito à Caixa Branca, este realizado principalmente pela equipe de desenvolvimento, exige, necessariamente, conhecimento da estrutura interna do software. O conhecimento da estrutura é necessário para a realização de testes relativos ao código, arquitetura ou estrutura do software. Testes de integração, unitários ou componentes e sistemas podem se realizados na Caixa Branca, quando a junção das partes da arquitetura precisam ser avaliadas quanto à sua execução harmônica.

Nome completo: *

GUSTAVO TELES BARRETTO

Selecione a afirmação INCORRETA a respeito de Testes Estáticos x Testes Dinâmicos: (0,5 pts)

- ☐ Teste Dinâmico é focado nos comportamentos.
- ☒ Com Testes Dinâmicos, o custo de selecionar um defeito é menor.
- ☐ Teste dinâmico detecta os defeitos e falhas ao executar o Software.
- ☐ Testes Estáticos detectam erros em produtos de trabalho.
- ☐ Teste Estático é focado em melhorar a consistência e a qualidade dos produtos de trabalho.

Limpar seleção



Pedir acesso para editar



Descreva a finalidade de um Teste de Fumaça. (1,5 pts)

O teste fumaça (Smoke Test) objetiva realizar testes que atestem a qualidade das funcionalidades essenciais do software. Se refere à toda funcionalidade essencial ou que represente o Most Value Product (MVP) do que deve ser realizado pelo software. As funcionalidades principais são testadas pelo smoke test, considerando um dos testes mais importantes a serem realizados nos softwares.

Quantos e quais são os Níveis de Teste? Descreva cada um deles. (1,5 pts)

São quatro níveis de teste: teste de componente, teste de integração, teste de sistema e teste de aceitação.

No teste de componente, também chamado de teste unitário, realiza-se teste sobre os comportamentos funcionais e não funcionais de partes do software, também denominado de componentes. Esse teste previne a propagação de defeitos ou falhas nos outros componentes e é realizado em códigos, modelo de dados ou na especificação dos componentes.

O teste de integração é a verificação da funcionalidade harmônica entre as unidades ou componentes do sistema. É a etapa pratica logo após o teste de componente. É a verificação da interação entre os componentes previamente construídos pelo desenvolvedor.

O teste de sistema, por sua vez, representa na realização de testes do sistema como um todo. Nesta etapa, já se fez a análise e os testes dos seus componentes, da integração paulatina entre eles e, agora, faz-se o teste do software como um todo. A intenção é de verificar se o software funcionará de forma harmônica quando da união de todos os módulos/componentes do sistema.

Por fim, temos o teste de aceitação, que representa a verificação do sistema como um todo, incluindo o seu comportamento durante a interação com o usuário. Seu conceito é abrangente, incluindo análises de processos de negócio, requisitos de usuário ou de negócio até análise legal e regulamentar do sistema, procedimentos de instalação e relatórios de risco. É a verificação prática da manipulação do sistema e seus desdobramentos quando utilizado pela sociedade.



Pedir acesso para editar



O campo de entrada referente a idade, da interface abaixo, deve aceitar apenas valores de 20 até 64. Utilizando as técnicas de particionamento de equivalência e de análise do valor limite, quais valores você utilizaria para realizar o teste deste campo? (1,0 pt)

Calculadora de IMC Adulto

Preencha os itens abaixo com os seus dados e em seguida clique em calcular.

Nome:

Idade:

Peso (kg):

Altura (m):

Valores números abaixo de 20, igual à 20 e 64 e acima de 64. Além disso, valores alfabéticos, strings ou tipos de dados não adequados ao campo indicado, como símbolos e acentos.

Descreva o que acontece em cada uma das etapas do Ciclo de vida do Teste de Software: Análise de Requisitos, Fase de Planejamento, Integração do Caso de Teste, Configuração do Ambiente de Teste, Fase de Implementação, Encerramento. (1,5 pts)

A Análise de Requisitos é o momento em que se desenha a solução do problema, isto é, do defeito ou falha identificado no software. O Planejamento é o momento em que se estuda a solução mais adequada para o defeito/falha e o processo a ser aplicado para o alcance dessa solução, incluindo a escolha dos equipamentos necessários, assim como pessoal apto à implementação do teste. Posteriormente, passa-se para a etapa de Integração do Caso de Teste, onde se verificam se as hipóteses que dão solução e que serão aplicadas, sob minuciosa análise, apresentam viabilidade de serem aplicadas pela equipe testadora. Já a Configuração do Ambiente verifica toda a indumentária necessária para a realização do teste, bem como estrutura o ambiente necessário à realização do teste. Por fim, é feita a Implementação, momento em que se valida todas as hipóteses dadas à solução do defeito/falha. É nesta última etapa que são produzidos os relatórios e documentos necessários para revisão, registro e armazenamento das falhas e defeitos ocorridas, das soluções utilizadas e de tudo aquilo que fora necessário providenciar para a resolução do problema.

 [Pedir acesso para editar](#)

Descreva a Técnica de Teste: Análise de valor limite. (1,5 pts)

Trata-se de técnica de teste específica para verificar os limites de entrada dos dados a serem recebidos pelo software. O teste verifica a inserção esperada e inesperada dos mais diferentes tipos de dados que possam ser recebidos pelo software, sejam eles acima daquele valor esperado ou diferente daquele valor esperado. A título ilustrativo, podemos indicar os valores números acima daqueles sugeridos pelo sistema. O sistema deve evitar defeitos ou falhas mediante impedimento da inserção do dado no sistema. A mesma resolução se aplica à inserção de valores alfabéticos em campos de valores numéricos e vice-versa. O software deve impedir a inserção de tais dados ou corresponder de modo que não gere nenhum defeito ou falha no software.

Enviar

Limpar formulário

Nunca envie senhas pelo Formulários Google.

Este formulário foi criado em Digital House. [Denunciar abuso](#)

Google Formulários



Pedir acesso para editar

