

CONCURSO PÚBLICO – 01/2022

Área de Conhecimento: Arquitetura de Software/Desenvolvimento de Software e Gestão de Configuração de Software

PROVA ESCRITA – PADRÃO DE RESPOSTA

QUESTÃO 1: Disserte sobre a arquitetura de sistemas de software. Discuta modelos monolíticos e arquiteturas cliente-servidor baseadas em (duas e três) camadas. Discuta ainda arquiteturas orientadas a serviços e suas características. Apresente os pontos positivos e negativos de cada abordagem e ilustre sua explanação com exemplos. Comente sobre a aplicação dessas abordagens no desenvolvimento de aplicações para dispositivos móveis.

Candidato deve citar e descrever características (principalmente topologia, pontos positivos e negativos) de cada tipo de arquitetura (monolítica, em camadas e orientada a serviços), conforme descrito no Capítulo 1 de Lazzeri (2009).

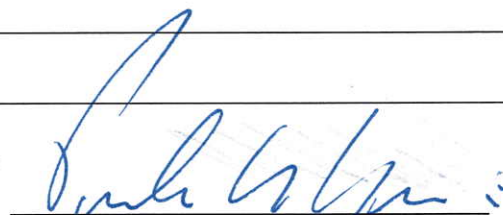
Candidato deve ainda comentar sobre a aplicação das abordagens no desenvolvimento de aplicações para dispositivos móveis, conforme seção 13.6.6 de Pressmann e Maxim (2016), com ênfase para escolha entre cliente magro (think cliente) e cliente rico (rich cliente)



Avaliador 1 (Pablo Schoeffel)

Avaliador 3 (Jordan Paulesky Juliani)

Membros da Banca:



Avaliador 2 (Paulo Cesar R. Gomes)



Presidente da Banca (Pablo Schoeffel)

CONCURSO PÚBLICO – 01/2022

Área de Conhecimento: Arquitetura de Software/Desenvolvimento de Software e Gestão de Configuração de Software

PROVA ESCRITA – PADRÃO DE RESPOSTA

QUESTÃO 2. Além de tipos de testes tradicionais de software, como testes funcionais, de regressão, de interface, de segurança e de performance, Molinari (2017) e Pressmann e Maxim (2016) descrevem outros tipos de testes específicos usados no mundo de dispositivos móveis. Cite e descreva esses outros tipos de testes.

O candidato deve citar e descrever o funcionamento, preferencialmente citando exemplos, de pelo menos 5 outros tipos de testes de software de aplicações móveis.

Molinari (2017), no capítulo 11, classifica diferentes técnicas para testes funcionais móveis: baseado no projeto, baseado no aparelho, na interrupção de uso, na rede e nos parâmetros de segundo. Ainda descreve testes de integração entre app e sistema operacional, teste de integração entre app e hardware do aparelho, testes de integração entre app e backend, testes de compatibilidade, teste operacional, teste de instalação, desinstalação e atualização, teste de utilização de recursos de hardware, teste de interrupção: bateria, ligação e alertas, teste de conectividade, teste de consumo de bateria, teste de memória e espaço em disco, teste de experiência do usuário.

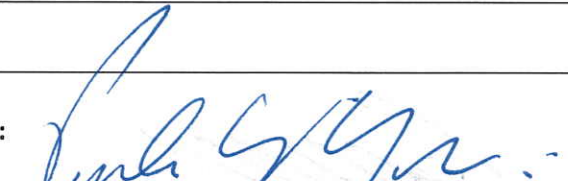
Pressmann e Maxim (2016), no capítulo 26, descrevem testes de gestos, teste de entrada e reconhecimento de voz, teste de entrada por teclado virtual, teste de alertas e condições extraordinárias.



Avaliador 1 (Pablo Schoeffel)

Avaliador 3 (Jordan Paulesky Juliani)

Membros da Banca:



Avaliador 2 (Paulo Cesar R. Gomes)



Presidente da Banca (Pablo Schoeffel)

CONCURSO PÚBLICO – 01/2022

Área de Conhecimento: Arquitetura de Software/Desenvolvimento de Software e Gestão de Configuração de Software

PROVA ESCRITA – PADRÃO DE RESPOSTA

QUESTÃO 3: Disserte sobre Pipeline de integração e entrega contínuas, descrevendo um exemplo da criação de um projeto de integração contínua vinculado ao repositório Git de um projeto simples.

É esperado que o candidato descreva os passos de um pipeline de integração e entregas contínuas, detalhando como isso pode ser implementado utilizando ferramentas automatizadas.

No Capítulo 3, Humble e Farley (2013) descrevem pré-condições (controle de versão, processo automatizado de compilação e aceitação da equipe), pré-requisitos para a integração contínua (checkins regulares, criar conjunto de testes automatizados, manter processo de compilação e testes curto, com usar software de integração contínua e algumas práticas essenciais.

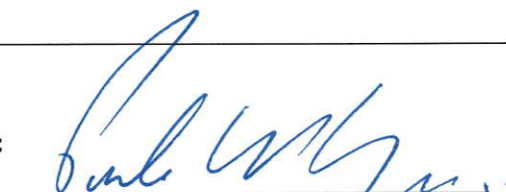
Monteiro et al. (2021), nas páginas 193-202, descreve como funciona o pipeline de integração e entregas contínuas, detalhando 3 passos: criação de um projeto, gráficos e logs, trigger build.



Avaliador 1 (Pablo Schoeffel)

Avaliador 3 (Jordan Paulesky Juliani)

Membros da Banca:



Avaliador 2 (Paulo Cesar R. Gomes)



Presidente da Banca (Pablo Schoeffel)

CONCURSO PÚBLICO – 01/2022

Área de Conhecimento: Arquitetura de Software/Desenvolvimento de Software e Gestão de Configuração de Software

PROVA ESCRITA – PADRÃO DE RESPOSTA

QUESTÃO 4: O aplicativo “Cronômetro” é um aplicativo móvel, desenvolvido para Android em Java, que deve mostrar o tempo corrido de um cronômetro, que pode ser acionado por três botões:

1. Iniciar – quando clicar nesse botão, os segundos começam a ser incrementados
2. Parar – quando clicar nesse botão, os segundos param de ser incrementados
3. Reset – Quando clicar nesse botão, os segundos voltam a zero.

Considerando isso, explique em detalhes o código XML do layout descrito abaixo e crie um protótipo de tela de acordo com código

Conforme projeto demonstrado por GRIFFTIHS, GRIFFTIHS (2016), pág. 119, espera-se que o candidato crie um protótipo de baixa fidelidade, contendo os elementos gráficos condizentes com o layout do código, incluindo alinhamento.

É esperado que o candidato explique o tipo de layout utilizado (linhas 1 a 9)

É esperado que o candidato explique cada um dos elementos gráficos da interface: TextView (linhas 11 a 20), Button Start (linhas 22 a 30), Botão Stop (linhas 32 a 40), Botão Reset (linhas 42 a 50)

É esperado que o candidato explique o uso das Strings extras (string.xml), identificadas nas linhas 30, 40 e 50.



Avaliador 1 (Pablo Schoeffel)

Avaliador 3 (Jordan Paulesky Juliani)

Membros da Banca:



Avaliador 2 (Paulo Cesar R. Gomes)



Presidente da Banca (Pablo Schoeffel)

CONCURSO PÚBLICO – 01/2022

Área de Conhecimento: Arquitetura de Software/Desenvolvimento de Software e Gestão de Configuração de Software

PROVA ESCRITA – PADRÃO DE RESPOSTA

QUESTÃO 5. Considerando o mesmo projeto descrito na Questão 4, explique em detalhes o código

JAVA descrito abaixo.

Conforme projeto demonstrado por GRIFFTIHS e GRIFFTIHS (2016), pág. 157 e 158, espera-se que o candidato explique as variáveis usadas na classe StopwatchActivity (linhas 10 a 12).

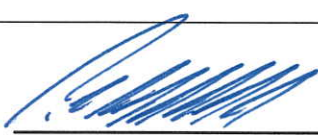
Espera-se que o candidato explique o uso do objetivo savedInstanceState, que obtém o estado anterior do cronômetro, caso a atividade tenha sido destruída e recriada.

Espera-se que o candidato explique o funcionamento dos métodos onPause, onResume, onClickStart, onClickStop e onClickReset

Espera-se que o candidato explique a regra de negócio do método runTimer, que usa um Handler para incrementar os segundos e atualizar o elemento de exibição de textos.

Espera-se que o candidato complemente o código do método “run”, similar ao descrito abaixo em Griffthihs e Griffthihs (2016 pag. 158):

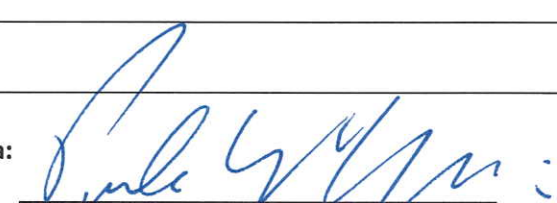
```
int hours = seconds/3600;  
int minutes = (seconds%3600)/60;  
int secs = seconds%60;  
String time = String.format("%d:%02d:%02d", hours, minutes, secs);  
timeView.setText(time);  
if (running) {  
    seconds ++;  
}
```



Avaliador 1 (Pablo Schoeffel)

Avaliador 3 (Jordan Paulesky Juliani)

Membros da Banca:



Avaliador 2 (Paulo Cesar R. Gomes)



Presidente da Banca (Pablo Schoeffel)



Assinaturas do documento



Código para verificação: **0D30ZAH0**

Este documento foi assinado digitalmente pelos seguintes signatários nas datas indicadas:



JORDAN PAULESKY JULIANI (CPF: 005.XXX.389-XX) em 07/11/2022 às 17:34:10

Emitido por: "SGP-e", emitido em 30/03/2018 - 12:41:57 e válido até 30/03/2118 - 12:41:57.

(Assinatura do sistema)

Para verificar a autenticidade desta cópia, acesse o link <https://portal.sgpe.sea.sc.gov.br/portal-externo/conferencia-documento/VURFU0NfMTIwMjJfMDAwNDk5MTdfNDk5ODZfMjAyMI8wRDMwWkFIMA==> ou o site <https://portal.sgpe.sea.sc.gov.br/portal-externo> e informe o processo **UDESC 00049917/2022** e o código **0D30ZAH0** ou aponte a câmera para o QR Code presente nesta página para realizar a conferência.