

| |
|--|
| CURSO: Tecnologia em Análise e Desenvolvimento de Sistemas |
| POLO DE APOIO PRESENCIAL: CONSOLAÇÃO HIGIENÓPOLIS |
| SEMESTRE : 2021/1 |
| COMPONENTE CURRICULAR / TEMA: Prática Profissional em ADS |
| NOME COMPLETO DO ALUNO: Grupo ADS 2021 |
| TIA: 18504345- 19008376 - 20003978 |
| NOME DO PROFESSOR: Fabio Lopes |

Atividade em Grupo – Executando a modelagem.

Nesta síntese, o grupo deverá entregar os modelos UML em conformidade com a especificação entregue na Unidade 1 – Diagramas de sequência e o Diagrama de classes –, desenhado em ferramenta de modelagem como Draw.io ou similar. O Plano de testes deve ser inserido no GitHub do projeto para avaliação.

AppVision

Plano de Teste

Versão <1.0>

Histórico da Revisão

| Data | Versão | Descrição | Autor |
|--------------|--------|---|-----------|
| <16/03/2021> | <1.0> | <Release Inicial - Protótipo do Plano de Teste> | <Viviane> |
| | | | |
| | | | |
| | | | |

Índice

Objetivos

Requisitos para o Teste

Estratégia do Teste

Recursos

Marcos do Projeto

Produtos de Trabalho

Tarefas do Projeto

Plano de Teste

1. Introdução

Finalidade

Este documento descreve o plano para testar o protótipo de arquitetura do AppVision. Este documento de Plano de Teste suporta os seguintes objetivos:

- Identificar informações existentes do projeto e o software que deve ser testado.
- Listar os requisitos de teste recomendados (nível alto).
- Recomendar e descrever as estratégias de teste a serem empregadas.
- Identificar os recursos requeridos e fornecer uma estimativa dos esforços de teste.
- Listar os elementos de produto de trabalho das tarefas de teste.

2. Escopo

Este Plano de Teste descreve os testes de integração e do sistema que serão conduzidos no protótipo de arquitetura após a integração dos subsistemas e componentes identificados no Plano de Integração da Construção para o Protótipo .

Assume-se que o teste de unidade já forneceu por meio de teste de caixa preta, uma cobertura extensiva do código fonte e o teste de todas as interfaces do módulo.

O objetivo da montagem do protótipo de arquitetura era testar a possibilidade e o desempenho da arquitetura selecionada. É crítico que todas as interfaces do sistema e do subsistema sejam testadas, bem como o desempenho do sistema nesse estágio antecipado. O teste dos recursos e da funcionalidade do sistema não será conduzido no protótipo.

As interfaces entre os seguintes subsistemas serão testadas:

1. Cadastro de membro no aplicativo.
2. Cadastro de novos itens (filmes, séries, livros).
3. Catálogo de Itens.

As interfaces externas para os seguintes dispositivos serão testadas:

1. PCs locais
2. PCs remotos.

As medidas de desempenho mais críticas a testar são:

1. Tempo de resposta para login remoto no sistema de cadastro do aplicativo.
2. Tempo de resposta para acesso ao cadastrar itens.
3. Tempo de resposta para acesso ao Sistema de Catálogo de Itens do Aplicativo.
4. Tempo de resposta do membro quando o sistema está carregado com 10 membros com login efetuado.
5. Tempo de resposta do membro quando existem 20 acessos simultâneos ao banco de dados do Catálogo da Home.

2. Requisitos para o Teste

A lista a seguir identifica os itens (casos de uso, requisitos funcionais, requisitos não funcionais) que foram identificados como alvos do teste. Essa lista representa *o que* será testado.

2.1 Teste de Integridade dos Dados e do Banco de Dados

Verificar acesso ao Banco de Dados do Catálogo de Itens.

Verificar acessos de leitura simultâneos ao registro.

Verificar interrupção durante atualizações do Catálogo de Itens.

Verificar a recuperação correta de atualizações dos dados do banco de dados.

2.2. Teste de Função

Documento de Visão, seção 12.2: "O sistema deve fazer interface com o Sistema de Banco de Dados de Catálogo de Itens existente. O AppVision deve suportar o formato de dados conforme definido em [2]."

Documento de Visão, seção 12.2: "O sistema deve fazer interface com o Sistema de Recomendação existente e deve suportar o formato de dados conforme definido em [1]."

Documento de Visão, seção 12.2: "O componente do cliente do sistema deve operar em qualquer navegador disponível no dispositivo."

Especificação Complementar, seção 9.3: "Os dados devem ser persistidos em uma base de dados (Relacional ou NoSQL)."

2.3 Teste de Ciclo de Negócio

Nenhum.

2.4 Teste de Interface com o Usuário

Verificar a facilidade de navegação utilizando um conjunto de amostras de telas.

Verificar se as telas de amostra estão em conformidade com os padrões da GUI.

Documento de Visão, seção 10: "O Sistema deve ter fácil utilização e deve ser apropriado para colaboradores da empresa sem treinamento adicional no Sistema."

Documento de Visão, seção 12.1: "Acessar as funções da rede social através da web ou por aplicativos móveis - somente por meio do navegador disponível no dispositivo."

Especificação Complementar, seção 5.1: "A aplicação deve ser responsiva e leve, evitando demoras no carregamento das funcionalidades."

2.5 Teste de Desempenho

Verificar o tempo de resposta para acesso ao sistema de recomendação externo.

Verificar o tempo de resposta para acesso ao subsistema Catálogo de Itens no Banco de dados externo.

Verificar o tempo de resposta para login remoto.

Verificar o tempo de resposta para a submissão remota de registro de membros.

Documento de Visão, seção 12.3: "O sistema deverá fornecer acesso ao Banco de Dados de Catálogo de Itens em nuvem com um tempo de espera não superior 5 segundos."

Especificação Complementar, seção 7.2: "O sistema deverá fornecer acesso ao Banco de Dados de Catálogo de Itens em nuvem com um tempo de espera não superior 5 segundos."

2.6 Teste de Carga

Verificar a resposta do sistema quando estiver carregado com 10 membros com logon efetuado.

Verificar a resposta do sistema quando existir 20 acessos simultâneos de membros ao Catálogo de Cursos.

2.7 Teste de Estresse

Nenhum.

2.8 Teste de Volume

Nenhum.

2.9 Teste de Segurança e Controle de Acesso

Verificar o Logon a partir de um PC local.

Verificar o Logon a partir de um PC remoto.

Verificar a segurança de Logon por meio de mecanismos de nome de usuário e senha.

2.10 Teste de Failover / Recuperação

Nenhum.

2.11 Teste de Configuração

Documento de Visão, seção 12.2: "Acessar as funções da rede social através da web ou por aplicativos móveis - somente por meio do navegador disponível no dispositivo."

Especificação Complementar, Seção 9.5: "A interface baseada na Web deve ser compatível com o ambiente de tempo de execução Java 1.1 VM."

2.12 Teste de Instalação

Nenhum.

3. Estratégia do Teste

A Estratégia de Teste apresenta a abordagem recomendada para o teste dos aplicativos de software. A seção anterior dos Requisitos de Teste descrevia *o que* será testado; esta descreve *como* será testado.

As principais considerações para a estratégia de teste são as técnicas a serem utilizadas e o critério para saber quando o teste está concluído.

Além das considerações fornecidas para cada teste a seguir, o teste deve ser executado apenas utilizando bancos de dados conhecidos e controlados, em ambientes protegidos.

A estratégia de teste a seguir é genérica por natureza e foi desenvolvida para ser aplicada aos requisitos listados na seção 4 deste documento.

3.1 Tipos de Teste

3.1.1 Teste de Integridade dos Dados e do Banco de Dados

Os bancos de dados e os processos de banco de dados devem ser testados como sistemas separados. Esses sistemas devem ser testados sem os aplicativos (como a interface para os dados). É necessário executar pesquisas adicionais referentes ao DBMS a fim de identificar as ferramentas / técnicas que poderão existir para suportar os testes identificados a seguir.

| | |
|--------------------------|---|
| Objetivo do Teste: | Assegurar que os processos e métodos de acesso ao Banco de Dados funcionem corretamente e sem corrupção de dados. |
| Técnica: | <ul style="list-style-type: none">• Chamar cada processo e método de acesso a banco de dados, propagando cada um com dados válidos e inválidos (ou pedidos de dados).• Inspeccionar o banco de dados para assegurar que os dados foram preenchidos conforme planejado e que todos os eventos do banco de dados ocorreram adequadamente ou revisar os dados retornados para assegurar que os dados corretos foram recuperados (pelas razões corretas) |
| Critérios de Conclusão: | Todos os processos e métodos de acesso ao banco de dados funcionam conforme projetado e sem nenhuma corrupção de dados. |
| Considerações Especiais: | <ul style="list-style-type: none">• Os testes podem exigir drivers ou um ambiente de desenvolvimento DBMS para digitar ou modificar dados diretamente nos bancos de dados.• Os processos devem ser chamados manualmente.• Bancos de dados pequenos ou de tamanho mínimo (número limitado de registros) devem ser utilizados para aumentar a visibilidade de quaisquer eventos não aceitáveis. |

3.1.2 Teste de Função

Os testes do aplicativo devem ter foco em quaisquer requisitos de destino que possam ser rastreados diretamente para casos de uso (ou funções de negócios) e regras de negócios. A meta desse teste é verificar a adequada aceitação, o processamento e a recuperação dos dados, e a implementação apropriada das regras de negócios. Esse tipo de teste baseia-se em técnicas de caixa preta, ou seja, verificar o aplicativo (e seus processos internos) interagindo com o aplicativo por meio da GUI e analisar a saída (resultados). A seguir é identificado um esboço do teste recomendado para cada aplicativo:

| | |
|--------------------------|---|
| Objetivo do Teste: | Assegurar a navegação correta do aplicativo, além da entrada, processamento e recuperação de dados. |
| Técnica: | <ul style="list-style-type: none"> • Executar cada caso de uso, fluxo de caso de uso ou função, utilizando dados válidos e inválidos, para verificar o seguinte: • Os resultados esperados ocorrerão quando forem usados dados válidos. • As mensagens de erro / aviso apropriadas sejam exibidas quando dados inválidos forem utilizados. • Cada regra de negócio será adequadamente aplicada. |
| CrITÉRIOS de Conclusão: | <ul style="list-style-type: none"> • Todos os testes planejados foram executados. • Todos os defeitos identificados foram tratados. |
| Considerações Especiais: | <ul style="list-style-type: none"> • O acesso ao Servidor do Banco de dados e ao Sistema de recomendação, existentes é requerido para executar alguns dos Testes de Sistema identificados no Protótipo. |

3.1.3 Teste de Ciclo de Negócio

Esta seção não é aplicável ao teste do protótipo de arquitetura.

3.1.4 Teste da Interface com o Usuário

O teste da Interface com o Usuário verifica a interação de um usuário com o software. A meta do Teste de UI é assegurar que a Interface com o Usuário forneça ao usuário o acesso e a navegação adequados por meio das funções dos aplicativos. Além disso, o Teste de UI assegura que os objetos contidos na UI funcionem conforme esperado e estejam em conformidade com padrões corporativos ou do segmento de mercado.

| | |
|--------------------|--|
| Objetivo do Teste: | Verifique o seguinte: |
| | <ul style="list-style-type: none"> • A navegação pelo aplicativo reflete os requisitos e funções de negócios, incluindo a navegação janela a janela, campo a campo e o uso de métodos de acesso (teclas de tabulação, movimentos do mouse e teclas aceleradoras) • Objetos e características da janela, tais como menus, tamanho, posição, estado e foco estão em conformidade com os padrões. |

| | |
|--------------------------|---|
| Técnica: | <ul style="list-style-type: none"> • Criar / modificar testes para cada janela a fim de verificar a navegação adequada e os estados de objeto para cada janela e objeto do aplicativo. |
| CrITÉRIOS de Conclusão: | Verificação com êxito de cada janela permanecer consistente com a versão de benchmark ou dentro do padrão aceitável |
| Considerações Especiais: | <ul style="list-style-type: none"> • Nem todas as propriedades de objetos personalizados e de terceiros podem ser acessadas. |

3.1.5 Traçado de Perfil de Desempenho

O teste de desempenho mede tempos de resposta, taxas de transação e outros requisitos sensíveis ao tempo. A meta do teste de Desempenho é verificar e validar se os requisitos de desempenho foram alcançados. O teste de desempenho normalmente é executado várias vezes, cada uma utilizando uma "carga de segundo plano" diferente no sistema. O teste inicial deve ser executado com uma carga "nominal", semelhante à carga normal observada (ou prevista) no sistema de destino. Um segundo teste de desempenho é executado utilizando uma carga de pico.

Além disso, os testes de desempenho podem ser utilizados para traçar o perfil e ajustar o desempenho de um sistema como uma função de condições, como a carga de trabalho ou configurações de hardware.

NOTA: As transações a seguir se referem a "transações comerciais lógicas." Essas são transações são definidas como funções específicas que se espera que um usuário do sistema execute utilizando o aplicativo, como incluir ou modificar um determinado contrato.

| | |
|-------------------------|---|
| Objetivo do Teste: | <p>Validar o Tempo de Resposta do Sistema para funções de negócios ou transações designadas sob as duas condições a seguir:</p> <ul style="list-style-type: none"> - volume normal previsto - volume de pior caso previsto |
| Técnica: | <ul style="list-style-type: none"> • Utilizar Scripts de Teste desenvolvidos para Teste de Modelo de Negócio (Teste do Sistema). • Modificar arquivos de dados (a fim de aumentar o número de transações) ou modificar scripts a fim de aumentar o número de iterações ocorrido em cada transação. • Os scripts devem ser executados em uma máquina (o melhor é avaliar o desempenho de um único usuário, uma única transação) e repetidos com vários clientes (virtuais ou reais, <i>consulte as considerações especiais a seguir</i>). |
| CrITÉRIOS de Conclusão: | <ul style="list-style-type: none"> • Transação Única / usuário único: Conclusão com êxito dos scripts de teste sem nenhum defeito e na alocação de tempo esperada / requerida (por transação) |

- Várias Transações / vários usuários: Conclusão com êxito dos scripts de teste sem nenhum defeito e dentro de alocação de tempo aceitável.

Considerações Especiais:

- O teste abrangente do desempenho inclui ter uma carga "em segundo plano" no servidor. Há vários métodos que podem ser usados para executar esse teste, incluindo:
 - "Encaminhar as transações" diretamente para o servidor, geralmente na forma de chamadas SQL.
 - Criar carga "virtual" de usuários para simular muitos (geralmente várias centenas) de clientes. Para se obter essa carga, geralmente são usadas ferramentas de Emulação de Terminal Remoto. Essa técnica também pode ser utilizada para carregar a rede com "tráfego".
 - Utilizar vários clientes físicos, cada qual executando scripts de teste para inserir carga no sistema.
- O teste de desempenho deverá ser executado em uma máquina dedicada ou em um período de tempo dedicado. Isso permitirá o controle total e a medição exata.
- Os bancos de dados utilizados para teste de Desempenho deverão ter tamanhos reais ou ser igualmente escalados.

3.1.6 Teste de Carga

As medidas do teste de carga sujeitam o sistema em teste a cargas de trabalho variáveis para avaliar a capacidade do sistema em continuar a funcionar corretamente sob essas diferentes cargas de trabalho. A meta desse teste de carga é determinar e assegurar que o sistema funcione adequadamente com uma carga de trabalho superior à carga máxima esperada. Além disso, o teste de carga avalia as características de desempenho (tempos de resposta, taxas de transação e outros aspectos sensíveis ao tempo).

NOTA: As transações a seguir se referem a "transações comerciais lógicas." Essas são transações são definidas como funções específicas que se espera que um usuário do sistema execute utilizando o aplicativo, como incluir ou modificar um determinado contrato.

Objetivo do Teste:

Verificar o Tempo de Resposta do Sistema para casos de negócios ou transações designadas sob condições de carga de trabalho variáveis.

Técnica:

- Utilizar os testes desenvolvidos para o Teste do Ciclo de Negócio.
- Modificar os arquivos de dados (a fim de aumentar o número de transações) ou os testes a fim de aumentar o número de vezes que cada transação ocorre.

Critérios de Conclusão:

- Várias Transações / vários usuários: Conclusão com êxito dos testes sem nenhum defeito e dentro de alocação de tempo aceitável.

Considerações Especiais:

- Os testes de carga devem ser executados em uma máquina dedicada e em um período de tempo dedicado. Isso permitirá o controle total e a medição exata.
- Os bancos de dados utilizados para teste de carga deverão ter tamanhos reais ou ser igualmente escalados.

3.1.7 Teste de Estresse

Esta seção não é aplicável ao teste do protótipo de arquitetura.

3.1.8 Teste de Volume

Esta seção não é aplicável ao teste do protótipo de arquitetura.

3.1.9 Teste de Segurança e Controle de Acesso

O Teste de Segurança e de Controle de Acesso tem como foco duas áreas principais de segurança:

Segurança do aplicativo, incluindo o acesso aos Dados ou às Funções de Negócios.
Segurança do sistema, incluindo acesso remoto ao sistema.

A segurança do aplicativo assegura que, com base na segurança desejada, os usuários têm restrição a funções específicas ou estão limitados aos dados que estão disponíveis a eles. Por exemplo, todos têm permissão para inserir dados e criar novas contas, mas apenas os gerentes poderão excluí-los. Se houver segurança no nível dos dados, o teste assegura que o usuário "tipo" um pode consultar todas as informações do cliente, incluindo dados financeiros; no entanto, o usuário dois consulta apenas os dados demográficos para o mesmo cliente.

A segurança do sistema assegura que apenas os usuários, para os quais o acesso ao sistema foi concedido, sejam capazes de acessar os aplicativos e apenas por meio dos gateways apropriados.

Objetivo do Teste:

Segurança de Função / Dados: Verificar se o usuário pode acessar apenas as funções / dados para os quais seu tipo de usuário tenha recebido permissão.

Segurança do Sistema: Verificar se apenas os usuários com acesso ao sistema e aplicativo(s) têm permissão para acessá-los.

Técnica:

- Segurança de Função / Dados: Identificar e listar cada tipo de usuário e as funções / dados para os quais cada tipo tem permissão.
- Criar testes para cada tipo de usuário e verificar a permissão criando transações específicas para cada tipo de usuário.
- Modificar o tipo de usuário e executar novamente os testes para os mesmos usuários. Em cada caso, verificar

se as funções / dados adicionais estão corretamente disponíveis ou se têm seu acesso negado.

- Acesso ao Sistema (consulte considerações especiais a seguir)

Crítérios de Conclusão:

Para cada tipo de usuário conhecido, a função / dados apropriados estão disponíveis e todas as transações funcionem como esperado e sejam executadas nos testes de Função de Aplicativo anteriores

Considerações Especiais:

- O acesso ao sistema deve ser revisado / discutido com o administrador da rede ou do sistema apropriado. Talvez esse teste não seja necessário, pois pode ser uma função de administração da rede ou do sistema.

3.1.10 Teste de Failover e Recuperação

Esta seção não é aplicável ao teste do protótipo de arquitetura.

3.1.11 Teste de Configuração

Esta seção não é aplicável ao teste do protótipo de arquitetura do AppVision

3.1.12 Teste de Instalação

Esta seção não é aplicável ao teste do protótipo de arquitetura do AppVision

3.2 Ferramentas

As seguintes ferramentas serão empregadas para o teste do protótipo de arquitetura:

| | Ferramenta | Versão |
|--|------------------------------|-----------|
| Gerenciamento de Teste | Rational RequisitePro | A Definir |
| | Rational Unified Process | |
| Design de Teste | Rational Rose | A Definir |
| Controle de Defeitos | Rational ClearQuest | A Definir |
| Teste Funcional | Rational Robot | A Definir |
| Teste de Desempenho | Rational Visual Quantify | A Definir |
| Gerador de Perfil ou Monitor de Cobertura de Teste | Rational Visual PureCoverage | A Definir |
| Outras Ferramentas de Teste | Rational Purify | A Definir |
| | Rational TestFactory | |

| | | |
|--------------------------|-----------------------------------|-----------|
| Gerenciamento de Projeto | Microsoft Word Microsoft Excel | A Definir |
| Ferramentas de DBMS | A Definir | A Definir |

4.1 Funções

Esta tabela mostra as premissas de equipe para o teste do Protótipo.

Recursos Humanos

| Função | Recursos Mínimos Recomendados (número de trabalhadores alocados em período integral) | Responsabilidades Específicas/Comentários |
|-----------------------------------|---|--|
| Gerente de Testes | 1 – Antonio Thiago Wagner Viviane | Fornece supervisão de gerenciamento Responsabilidades: <ul style="list-style-type: none"> • Fornecer direção técnica • Adquirir recursos apropriados • Relatório de gerenciamento |
| Designer de Teste | Viviane Ribeiro | Identifica, prioriza e implementa casos de teste Responsabilidades: <ul style="list-style-type: none"> • Gerar plano de teste • Gerar Conjunto de Teste • Avaliar eficácia do esforço de teste |
| Testador do Sistema | Wagner | Executa os testes Responsabilidades: <ul style="list-style-type: none"> • Executar testes • Registrar resultados • Recuperar-se de erros • Documentar defeitos |
| Administrador do Sistema de Teste | Antonio Lucas | Assegura que os ativos e o ambiente de teste são gerenciados e mantidos. |

| | | |
|---|-----------------|---|
| | | Responsabilidades: <ul style="list-style-type: none"> • Administrar o sistema de gerenciamento de teste • Instalar / gerenciar o acesso do trabalhador aos sistemas de teste |
| Administração de Banco de Dados / Gerenciador de Banco de Dados | Antonio Lucas | Assegura que os ativos e o ambiente de dados de teste (banco de dados) são gerenciados e mantidos. Responsabilidades: <ul style="list-style-type: none"> • Administrar os dados de teste (banco de dados) |
| Designer | Viviane Ribeiro | Identifica e define as operações, atributos e associações das classes de teste Responsabilidades: <ul style="list-style-type: none"> • Identifica e define a(s) classes de teste • Identifica e define os pacotes de teste |
| Implementador | Thiago | Implementa e faz o teste de unidade das classes de teste e pacotes de teste Responsabilidades: <ul style="list-style-type: none"> • Cria os pacotes e classes de teste implementados no Conjunto de Teste. |

4.2 Sistema

A tabela a seguir descreve os recursos do sistema para o teste do protótipo do App Vision.

Recursos do Sistema

| Recurso | Nome / Tipo / Número de Série |
|---------------------------------------|-------------------------------|
| Servidor do Sistema de Banco de Dados | A definir |

| | |
|----------------------------------|-----------|
| Celulares de Teste do Cliente | A definir |
| Repositório de Testes | A definir |
| Desktop Desenvolvimento de Teste | A definir |

5. Marcos do Projeto

O teste do Protótipo de Arquitetura do AppVision incorpora tarefas de teste para cada um dos esforços de teste identificados nas seções anteriores. Marcos do projeto separados são identificados para comunicar o status e as realizações do projeto.

Consulte o Plano de Desenvolvimento de Software [13] e o Plano de Iteração E1 [14] para obter o planejamento do projeto principal ou de fases gerais.

| Tarefa de Marco | Esforço (pd) | Data de Início | Data de Encerramento |
|---------------------------------------|--------------|----------------|----------------------|
| Planejamento de Teste do Protótipo | 2 | A definir | A definir |
| Design de Teste do Protótipo | 3 | A definir | A definir |
| Desenvolvimento de Teste do Protótipo | 4 | A definir | A definir |
| Execução de Teste do Protótipo | 3 | A definir | A definir |
| Avaliação de Teste do Protótipo | 1 | A definir | A definir |

6. Segurança do Aplicativo

A norma que especifica as diretrizes para criação de aplicações seguras é definida como Common Criteria, ou Critério Comum, que deu origem à norma ISO/IEC 15408, e tem como objetivo:

Fornecer um conjunto de critérios fixos que permitam especificar a segurança de uma aplicação de forma não ambígua a partir de características do ambiente da aplicação, e definir formas de garantir a segurança da aplicação para o cliente final. Ou seja, Common Criteria pode ser utilizado para desenvolver um sistema seguro ou avaliar a segurança de um já existente. (LYRA, 2008, p. 173)

A Norma ISO/IEC 15408 apresenta quatro níveis de garantia de segurança (EAL – Evaluation Assurance Level), tais como:

O nível 1 tem por função testar a funcionalidade do sistema, pois visa atender os requisitos solicitados e identificar as proteções necessárias contra ameaças, além disso, deve haver avaliação na versão final, comparando com os requisitos iniciais sem o auxílio do desenvolvedor.

O nível 2 tem por função garantir que as aplicações desenvolvidas estejam atendendo aos padrões preestabelecidos, além disso, que seja realizada uma análise para verificar as vulnerabilidades do sistema e aplicar testes para observar a segurança e validação dos dados.

O nível 3 visa à aplicação de procedimentos de segurança sem que o sistema sofra com mudanças, além de propiciar testes completos do sistema sem precisar da reengenharia.

O nível 4 objetiva garantir maior segurança durante o desenvolvimento do sistema, aplicando alguns procedimentos de teste e verificação dos resultados.

Para se ter segurança em todo o ciclo de vida do desenvolvimento da aplicação, é importante a adoção de critérios de segurança do mais alto nível, com o uso de ferramentas e de processos também robustos.

AppVision trabalha na plataforma Heroku. A plataforma Heroku oferece um serviço do modelo de Computação em Nuvem. Ela está classificada como PaaS (Plataforma como Serviço), que gerencia todos os itens de hardware, dando facilidade para a manipulação, implantação, escalabilidade, controle de tráfego de sistemas desenvolvidos.

“A Infra-estrutura física do Heroku é hospedada e gerenciada em datacenters Amazon e utilizam a tecnologia Amazon Web Service (AWS). Amazon continuamente verifica os riscos e sofre avaliações periódicas para garantir a conformidade com os padrões da indústria. Operações do centro de dados da Amazon foram credenciados em: ISO 27001 - SOC 1/SSAE 16/ISAE 3402 (Anteriormente SAS 70 Tipo II) - PCI

Nível 1 - FISMA Moderado - Sarbanes-Oxley (SOX) – PCI. (Heroku, 2012) A empresa Salesforce.com é auditada anualmente mantendo seus recursos em conformidade com a lei dos EUA Sabanes-Oxley (SOX) de 2002.

“Testes de segurança aplicados por terceiros nas aplicações Heroku são realizadas por empresas de consultoria independentes e respeitáveis em segurança de T.I. Os resultados de cada avaliação são revisados com os avaliadores, risco classificado, e atribuído à resolução a equipe responsável”. (Heroku, 2012)

7. Produtos de Trabalho

Os produtos de trabalho das tarefas de teste definidas neste Plano de Teste estão descritos na tabela a seguir.

| Produtos de Trabalho | Proprietário | Revisão / Distribuição | Data Comprometida |
|---------------------------------|--------------|---|-------------------|
| Plano de Teste | Viviane | Equipe de Gerenciamento de Projeto Sênior | A definir |
| Ambiente de Teste | A definir | - | A definir |
| Conjunto de Teste | A definir | Revisão por Profissional Interno | A definir |
| Conjuntos de Dados de Teste | A definir | Revisão por Profissional Interno | A definir |
| Scripts de Teste | A definir | - | A definir |
| Stubs de Teste, Drivers | A definir | - | A definir |
| Relatórios de Defeitos do Teste | A definir | Equipe de Gerenciamento de Projeto Sênior | A definir |

| | | | |
|---------------------------------|-----------|---|-----------|
| Resultados do Teste | A definir | - | A definir |
| Relatório de Avaliação de Teste | A definir | Equipe de Gerenciamento de Projeto Sênior | A definir |

7.1 Conjunto de Teste

O Conjunto de Teste definirá todos os casos de teste e os scripts de teste associados a cada caso de teste.

7.2 Logs do Teste

Está planejada a utilização do RequisitePro para identificar os casos de teste e monitorar o status de cada um. Os resultados de teste serão resumidos no RequisitePro como não-testados, aprovados, aprovação condicional ou com falha. Resumindo, o RequisitePro será configurado para suportar os seguintes para cada caso de teste, conforme definido nas Diretrizes de Atributos de Requisitos [17]:

- Status do Teste
- Número do Build
- Testado por
- Data do Teste
- Notas do Teste

Será de responsabilidade do Testador do Sistema atualizar o status do teste no RequisitePro.

Os resultados de teste serão retidos no Controle de Configuração.

7.3 Relatórios de Defeitos

O Rational ClearQuest será utilizado para registrar e monitorar defeitos individuais.

8. Tarefas do Projeto

A seguir são mostradas as tarefas relacionadas ao teste do protótipo App Vision:

Planejar Teste

Identificar Requisitos para o Teste

Avaliar Risco

Desenvolver Estratégia de Teste

Identificar Recursos de Teste

Criar Planejamento

Gerar Plano de Teste

Projetar Teste

Análise de Carga de Trabalho (não aplicável para Protótipo)

Desenvolver Conjunto de Teste

Identificar e Descrever Casos de Teste

Identificar e Estruturar Scripts de Teste

Revisar e Acessar Cobertura de Teste

Implementar Teste

Configurar Ambiente de Teste

Registrar ou Programar Scripts de Teste

Desenvolver Drivers e Stubs de Teste

Identificar funcionalidade específica do Teste no design e modelo de implementação

Estabelecer Conjuntos de Dados Externos

Executar Teste

Executar Script de Teste

Avaliar Execução do Teste

Recuperar-se de Teste Interrompido

Verificar os Resultados

Investigar Resultados Inesperados

Registrar Defeitos

Avaliar Teste

Avaliar a Cobertura dos Casos de Teste

Avaliar Cobertura do Código

Analisar Defeitos

Determinar se os Critérios de Conclusão do Teste e os Critérios de Êxito foram alcançados

Criar Relatório de Avaliação do Teste

Referências Bibliográfica.: Direitos Autorais (c) IBM Corporation 1987, 2004. Todos os Direitos Reservados.
Exemplo da Web do Projeto de Registro em Curso - Versão 2001.03 https://www.cin.ufpe.br/~gta/rup-vc/extend.formal_resources/guidances/examples/resources/test_plan_v1.htm

<https://aberto.univem.edu.br/bitstream/handle/11077/875/TESTE%20DE%20PENETRA%C3%87%C3%83O%20PLATAFORMA%20PaaS%20-%20Revis%C3%A3o%20Final.pdf?sequence=1&isAllowed=y>

Plano de Teste - <Nome do Projeto> - Raquel Silveira <https://raquelsilveira.files.wordpress.com> ›

