

### Histórico da Revisão

Data	Versão	Descrição	Autor
4/abril/2021	1.0	Release Inicial - Protótipo do Plano de Teste	Rita Garcia

### Índice

1. [Objetivos](#)
2. [Requisitos para o Teste](#)
3. [Estratégia do Teste](#)
4. [Recursos](#)
5. [Marcos do Projeto](#)
6. [Produtos de Trabalho](#)
7. [Tarefas do Projeto](#)

## Plano de Teste

### Para o

### Protótipo de Arquitetura

## 1. Objetivos

### 1.1 Objetivo

Este documento descreve o plano para testar o protótipo de arquitetura do Aplicativo de Recomendações.

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.

## 1.2 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 [16].

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. Registro em Recomendações
2. Sistema de Serviços
3. Página de Recomendações

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 recomendações.
2. Tempo de resposta para acesso ao Sistema Serviços.
3. Tempo de resposta para acesso ao Sistema da Página de Recomendações.
4. Tempo de resposta do membro quando o sistema está carregado com 200 membros com login efetuado.
5. Tempo de resposta do membro quando existem 50 acessos simultâneos ao banco de dados da Página de Recomendações.

## 1.3 Referências

As referências aplicáveis são:

1. Course Billing Interface Specification, WC93332, 1985, Wylie College Press.
2. Course Catalog Database Specification, WC93422, 1985, Wylie College Press.
3. Course Registration System Vision Document, WylIT387, V1.0, 1998, Wylie College IT.
4. Course Registration System Glossary, WylIT406, V2.0, 1999, Wylie College IT.

5. Course Registration System Use Case Spec - Close Registration, WyIT403, V2.0, 1999, Wylie College IT.
6. Course Registration System Use Case Spec - Login, WyIT401, V2.0, 1999, Wylie College IT.
7. Course Registration System Use Case Spec - Maintain Professor Info, WyIT407, Version 2.0, 1999, Wylie College IT.
8. Course Registration System Use Case Spec - Register for Courses, WyIT402, Version 2.0, 1999, Wylie College IT.
9. Course Registration System Use Case Spec - Select Courses to Teach, WyIT405, Version 2.0, 1999, Wylie College IT.
10. Course Registration System Use Case Spec - Maintain Student Info, WyIT408, Version 2.0, 1999, Wylie College IT.
11. Course Registration System Use Case Spec - Submit Grades, WyIT409, Version 2.0, 1999, Wylie College IT.
12. Course Registration System Use Case Spec - View Report Card, WyIT410, Version 2.0, 1999, Wylie College IT.
13. Course Registration System Software Development Plan, WyIT418, V1.0, 1999, Wylie College IT.
14. Course Registration System Iteration Plan, Elaboration Iteration #E1, WyIT420, V1.0, 1999, Wylie College IT.
15. Course Registration System Software Architecture Document, WyIT431, V1.0, 1999, Wylie College IT.
16. Course Registration System Integration Build Plan for the Architectural Prototype, WyIT430, V1.0, 1999, Wylie College IT.
17. Course Registration System Requirements Attributes Guidelines, WyIT404, V1.0, 1999, Wylie College IT.

## **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.

(Nota: Um release futuro desse Plano de Teste pode utilizar o Rational RequisitePro para vincular diretamente aos requisitos nos Documentos de Caso de Uso e na Especificação Complementar.)

### **2.1 Teste de Integridade dos Dados e do Banco de Dados**

Verificar acesso ao Banco de Dados da Página de Recomendações.

Verificar acessos de leitura simultâneos ao registro.

Verificar interrupção durante atualizações da Página de Recomendações.

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 da Página de Recomendações existente. O Sistema de Recomendações 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 Serviços 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 computador pessoal com um microprocessador 486 ou superior."

Especificação Complementar, seção 9.3: "O componente do cliente do sistema deve operar em qualquer computador pessoal com pelo menos um microprocessador 486."

## **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 12.1: "A interface com o usuário de desktop deve estar em conformidade com o Windows 10."

Especificação Complementar, seção 5.1: "A interface com o usuário de desktop deve estar em conformidade com o Windows 10."

## **2.5 Teste de Desempenho**

Verificar o tempo de resposta para acesso ao sistema Serviços.

Verificar o tempo de resposta para acesso ao subsistema Página de Recomendações.

Verificar o tempo de resposta para login remoto.

Verificar o tempo de resposta para a submissão remota de registro em curso.

## **2.6 Teste de Carga**

Verificar a resposta do sistema quando estiver carregado com 200 membros com login efetuado.

Verificar a resposta do sistema quando existir 50 acessos simultâneos de membros a Página de Recomendações.

## **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: "O componente do cliente do sistema deve ser executado no Windows 10."

Especificação Complementar, Seção 9.4: "A interface baseada na Web do Sistema de Recomendações deve ser executada nos navegadores Chrome 85 e Internet Explorer 11."

## **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:**

- Chamar cada processo e método de acesso a banco de dados, propagando cada um com dados válidos e inválidos (ou pedidos de dados).
- Inspecionar 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)

**Crítérios de Conclusão:** Todos os processos e métodos de acesso ao banco de dados funcionam conforme projetado e sem nenhuma corrupção de dados.

**Considerações Especiais:**

- 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:**

- 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:**

- Todos os testes planejados foram executados.
- Todos os defeitos identificados foram tratados.

**Considerações Especiais:**

- O acesso ao Servidor WINDOWS e ao Sistema da Página de Recomendações e Sistema de Serviços 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:

- 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:**

- 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:**

- 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:** Validar o Tempo de Resposta do Sistema para funções de negócios ou transações designadas sob as duas condições a seguir:

- volume normal previsto
- volume de pior caso previsto

**Técnica:**

- 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, *consulte as considerações especiais a seguir*).

**Critérios de Conclusão:**

- 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 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 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 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)

**Critérios 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*

O teste de configuração verifica a operação do software em diferentes configurações de software e de hardware. Na maior parte dos ambientes de produção, as especificações de hardware específicas para as estações de trabalho cliente, as conexões de rede e os servidores de banco de dados variam. As estações de trabalho cliente podem ter softwares diferentes carregados, como aplicativos, drivers etc. A qualquer momento, muitas combinações diferentes podem estar ativas e utilizando recursos diferentes.

**Objetivo do Teste:** Validar e verificar se os Aplicativos cliente funcionam corretamente nas estações de trabalho cliente prescritas.

**Técnica:**

- Utilizar scripts de Teste de Integração e do Sistema
- Abrir / fechar diversos aplicativos do PC, como parte do teste ou antes do início do teste.
- Executar transações selecionadas para simular atividades do usuário para dentro e fora de diversos aplicativos de PC.
- Repita o processo acima, minimizando a memória convencional disponível no cliente.

**Critérios de Conclusão:** Para cada combinação do Protótipo e aplicativo de PC, as transações são concluídas com êxito e sem falhas.

**Considerações Especiais:**

- Quais Aplicativos de PC estão disponíveis e acessíveis nos clientes?
- Quais os aplicativos normalmente usados?
- Que dados estão em execução nos aplicativos (isto é, planilha grande aberta no Excel, documento de 100 páginas no Word).
- Os sistemas inteiros, servidores de rede, bancos de dados etc. também devem ser documentados como parte deste teste.

### *3.1.12 Teste de Instalação*

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

### 3.2 Ferramentas

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

	<b>Ferramenta</b>	<b>Versão</b>
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 Project	A Definir
	Microsoft Word	
	Microsoft Excel	
Ferramentas de DBMS	A Definir	A Definir

## 4. Recursos

Esta seção apresenta os recursos recomendados para o teste do protótipo de arquitetura do Sistema de Recomendações, suas principais responsabilidades e seu conhecimento ou configuração de habilidades.

### 4.1 Funções

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

### Recursos Humanos

<b>Função</b>	<b>Recursos Mínimos Recomendados</b>	<b>Responsabilidades Específicas/Comentários</b>
Gerente de Testes	1 – Rita Garcia	<p>Fornece supervisão de gerenciamento</p> <p>Responsabilidades:</p> <ul style="list-style-type: none"> <li>• Fornecer direção técnica</li> <li>• Adquirir recursos apropriados</li> <li>• Relatório de gerenciamento</li> </ul>
Designer de Teste	Aline Soares Joni Ramos	<p>Identifica, prioriza e implementa casos de teste</p> <p>Responsabilidades:</p> <ul style="list-style-type: none"> <li>• Gerar plano de teste</li> <li>• Gerar Conjunto de Teste</li> <li>• Avaliar eficácia do esforço de teste</li> </ul>
Testador do Sistema	Joni Ramos	<p>Executa os testes</p> <p>Responsabilidades:</p> <ul style="list-style-type: none"> <li>• Executar testes</li> <li>• Registrar resultados</li> <li>• Recuperar-se de erros</li> <li>• Documentar defeitos</li> </ul>
Administrador do Sistema de Teste	Edmilson Bispo	<p>Assegura que os ativos e o ambiente de teste são gerenciados e mantidos.</p> <p>Responsabilidades:</p> <ul style="list-style-type: none"> <li>• Administrar o sistema de gerenciamento de teste</li> <li>• Instalar / gerenciar o acesso do trabalhador aos sistemas de teste</li> </ul>
Administração de Banco de Dados / Gerenciador de Banco de Dados	Aline Soares	<p>Assegura que os ativos e o ambiente de dados de teste (banco de dados) são gerenciados e mantidos.</p>

		Responsabilidades: <ul style="list-style-type: none"> <li>• Administrar os dados de teste (banco de dados)</li> </ul>
Designer	Aline Soares	Identifica e define as operações, atributos e associações das classes de teste  Responsabilidades: <ul style="list-style-type: none"> <li>• Identifica e define a(s) classes de teste</li> <li>• Identifica e define os pacotes de teste</li> </ul>
Implementador	Aline Soares	Implementa e faz o teste de unidade das classes de teste e pacotes de teste  Responsabilidades: <ul style="list-style-type: none"> <li>• Cria os pacotes e classes de teste implementados no Conjunto de Teste.</li> </ul>

## 4.2 Sistema

A tabela a seguir descreve os recursos do sistema para o teste do protótipo do Sistema de Recomendações.

### Recursos do Sistema

Recurso	Nome / Tipo / Número de Série
Servidor do Sistema de Recomendações	Número de Série: X179113562b
	ID da Versão: CCDB-080885
	ID da Versão: BSSS-88335
PCs de Teste do Cliente	
	Número de Série: A8339223



	Número de Série: B9334022 Número de Série: B9332544
	Número de Série: R3322411 (Secretário) Número de Série: A8832234 (Laboratório TI) Número de Série: W4592233 (Laboratório de TI)
Repositório de Testes	
	Número de Série: X179773562b
PCs de Desenvolvimento de Teste - 6	Número de Série: A8888222 Número de Série: R3322435 Número de Série: I88323423 Número de Série: B0980988 Número de Série: R3333223 Número de Série: Y7289732

## 5. Marcos do Projeto

O teste do Protótipo de Arquitetura do Sistema de Recomendações 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	12 de abril	15 de abril
Design de Teste do Protótipo	3	15 de abril	18 de abril
Desenvolvimento de Teste do Protótipo	4	19 de abril	23 de abril
Execução de Teste do Protótipo	3	24 de abril	26 de abril
Avaliação de Teste do Protótipo	1	29 de abril	29 de abril

## 6. Produtos de Trabalho

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

<b>Produtos de Trabalho</b>	<b>Proprietário</b>	<b>Revisão / Distribuição</b>	<b>Data Comprometida</b>
Plano de Teste	Rita Garcia	Equipe de Gerenciamento de Projeto Sênior	15 de abril
Ambiente de Teste	Edmilson Bispo	-	18 de abril
Conjunto de Teste	Aline Soares e Joni Ramos	Revisão por Profissional Interno	23 de abril
Conjuntos de Dados de Teste	Aline Soares	Revisão por Profissional Interno	23 de abril
Scripts de Teste	Aline Soares	-	23 de abril
Stubs de Teste, Drivers	Aline Soares	-	23 de abril
Relatórios de Defeitos do Teste	Joni Ramos	Equipe de Gerenciamento de Projeto Sênior	26 de abril
Resultados do Teste	Joni Ramos	-	26 de abril
Relatório de Avaliação de Teste	Joni Ramos	Equipe de Gerenciamento de Projeto Sênior	29 de abril

### 6.1 Conjunto de Teste

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

## **6.2 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.

## **6.3 Relatórios de Defeitos**

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

## **7. Tarefas do Projeto**

A seguir são mostradas as tarefas relacionadas ao teste do protótipo de arquitetura do Sistema de Recomendações:

### **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