UNIVERSIDADE ESTADUAL DE CAMPINAS

SI304/ST366 – Engenharia de Software II 1 º Sem/2016

Caroline Resende Silveira – ra: 165921

Karina Sayuri Hagiwara – ra: 171419

**DOCUMENTO GERAL**

Histórico de Revisão

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versão | Descrição | Autor |
| 01/04/2016 | 1 | Formação das duplas e início do repositório no GitHub | Karina Sayuri Hagiwara |
| 15/04/2016 | 1.1 | Criação do documento | Karina Sayuri Hagiwara |
| 27/04/2016 | 2 | Elicitação dos requisitos | Karina Sayuri Hagiwara |
| 09/05/2016 | 2.1 | Requisitos conscientes | Karina Sayuri Hagiwara |
| 15/05/2016 | 2.2 | Requisitos de qualidade | Karina Sayuri Hagiwara |
| 16/05/2016 | 3 | Estruturação do documento | Karina Sayuri Hagiwara |
| 16/05/2016 | 3.2 | Atualização da estrutura do documento de especificação | Karina Sayuri Hagiwara |
| 17/05/2016 | 4 | Atualização dos requisitos funcionais | Karina Sayuri Hagiwara |
| 17/05/2016 | 4.1 | Atualização dos requisitos de qualidade | Karina Sayuri Hagiwara |
| 17/05/2016 | 5 | Organização do documento | Karina Sayuri Hagiwara |
| 20/05/2016 | 6 | Caso de uso textual (funcionalidade A) | Karina Sayuri Hagiwara |
| 22/05/2016 | 7 | Caso de uso textual (funcionalidade B) | Karina Sayuri Hagiwara |
| 24/05/2016 | 8 | Caso de uso textual | Karina Sayuri Hagiwara |
| 24/05/2016 | 9 | Caso de uso | Karina Sayuri Hagiwara |
| 25/05/2016 | 9.1 | Atualização do caso de uso e do caso de uso textual | Karina Sayuri Hagiwara |
| 25/05/2016 | 10 | Diagrama de fluxo de dados | Karina Sayuri Hagiwara |
| 26/05/2016 | 11 | Histórico de revisão | Karina Sayuri Hagiwara |
| 26/05/2016 | 12 | Matriz de rastreabilidade | Karina Sayuri Hagiwara |
| 27/05/2016 | 13 | Ponto de função | Karina Sayuri Hagiwara |
| 28/05/2016 | 14 | Atualização do ponto de função | Karina Sayuri Hagiwara |
| 28/05/2016 | 15 | Fator de ajuste | Karina Sayuri Hagiwara |
| 29/05/2016 | 16 | Priorização de kano | Karina Sayuri Hagiwara |
| 29/05/2016 | 17 | Atualização do caso de uso | Karina Sayuri Hagiwara |
| 30/05/2016 | 18 | Tabela (Cartões de atributo) | Karina Sayuri Hagiwara |
| 02/06/2016 | 19 | Manual do usuário | Karina Sayuri Hagiwara |

Sumário

1. Introdução
   1. Escopo

Este projeto tem o intuito de construir um sistema que realize algumas verificações numéricas. Para melhor construção do mesmo, devem-se ser utilizadas das práticas de engenharia de software. Através deste sistema, será possível identificar, dependendo da necessidade do usuário, se um número é par ou ímpar ou se um número é negativo ou positivo.

O sistema será um software que terá duas funções: verificar se para tal número informado pelo usuário é negativo ou positivo e o notificar ou verificar se para tal número informado pelo usuário é ímpar ou par e o notificar. Sendo que o usuário poderá esolher qual função deseja realizar, dessa forma, o controle deste será feito através de um menu de opções, o qual também possibilitará o usuário de finalizar o sistema. Este sistema será denominado sistema PINP.

* 1. Descrição dos stakeholders

Cliente (Usuária):

A aluna Caroline Resende Silveira, 18 anos, brasileira, estudante de Sistemas de Informação na Faculdade de Tecnologia da Unicamp de Limeira, está interessada um sistema que possibilite a mesma de realizar algumas verificações numéricas, como verificar se o número é par ou ímpar ou verificar se o número é negativo ou positivo.

Desenvolvedora:

A aluna Karina Sayuri Hagiwara , 21 anos, brasileira, estudante de Sistemas de Informação na Faculdade de Tecnologia da Unicamp de Limeira, com base nos conceitos de Engenharia de Software II, vistos em aula, será responsável por desenvolver o Sistema PINP e seus respectivos documentos e projetos.

Orientador (Professor):

O professor Pedro Ivo Garcia Nunes, brasileiro, o qual ministra a disciplina Engenharia de Software II, será o orientador do trabalho. Este será responsável pela proposição do desenvolvimento do projeto com base nos conceitos de Engenharia de Software II, vistos em aula.

1. Descrição geral
   1. Descrição do público-alvo

O Sistema PINP tem como público-alvo usuários que precisam ou tem interesse em realizar algumas verificações numéricas, tais como verificar se o número é par ou ímpar ou verificar se o número é negativo ou positivo.

* 1. Restrições

Entrada Padrão

No Sistema PINP a entrada deverá ser conforme a especificada durante a execução do programa. No menu, haverá 3 opções, sendo que é necessário que o usuário digite uma dessa 3 opções, 1, 2 ou 3. Nas demais funcionalidades, deverá ser digitado um número.

Sistema Operacional

O Sistema PINP deverá rodar no Sistema Operacional Linux (Ubuntu).

1. Requisitos
   1. Requisitos funcionais
      1. Assim que iniciado, o sistema deverá mostrar ao usuário um menu com as seguintes opções:
         1. verificar se um numero é positivo ou negativo;
         2. verificar se um número é par ou ímpar;
         3. verificar se ele deseja sair do programa.
      2. Assim que o usuário digitar uma opção, o sistema deverá verificar qual a opção que o usuário deseja realizar.
      3. Assim que o sistema verificar qual opção o usuário deseja realizar, o sistema deverá informar ao usuário se o número é negativo ou positivo quando um número é fornecido pelo usuário, caso ele escolha a opção 1 do menu.
      4. Assim que o sistema verificar qual opção o usuário deseja realizar, o sistema deverá verificar se um número é ímpar ou par quando um número inteiro é fornecido, caso ele escolha a opção 2 do menu.
      5. Assim que o sistema verificar o resultado da operação, caso ele escolha a opção 1 ou 2 do menu, deverá ser retornado o resultado.
      6. Assim que o sistema verificar qual opção o usuário deseja realizar, o sistema verificar qual opção o usuário deseja realizardeverá exibir uma mensagem para notificar o usuário que o sistema foi finalizado, caso ele escolha a opção 3 do menu e desejar sair do programa.
   2. Requisitos de qualidade:
      1. O sistema deverá permanecer aberto enquanto o usuário não escolher a opção 3 do menu e confirmar o desejo de sair do programa.
      2. O sistema deverá ser intuitivo e simples.
      3. O menu deverá ser autoexplicativo.
      4. O sistema deve conter informações que ajudem ao usuário inserir os dados de entrada.

Apêndices

Modelos

Caso de Uso



Caso de Uso Textual

1. Selecionar opção do menu
   1. Identificador: A.
   2. Nome: Exibir menu de opções.
   3. Autores: Karina Sayuri Hagiwara.
   4. Prioridade: Alta.
   5. Criticalidade: É crítica, pois é necessária para as funcionalidades B e C.
   6. Fonte: Caroline Resende.
   7. Responsável: Karina Sayuri Hagiwara.
   8. Descrição: O sistema deverá exibir um menu de opções e o usuário deverá escolher. A entrada será o número da opção desejada, que está relacionada a funcionalidade escolhida. O menu deverá conter as seguintes opções:
      1. verificar se um numero é positivo ou negativo;
      2. verificar se um número é par ou ímpar;
      3. verificar se ele deseja sair do programa.

Após escolhida a funcionalidade, o sistema deverá executar a funcionalidade desejada.

* 1. Trigger: -
  2. Atores: Karina Sayuri Hagiwara.
  3. Pré-condições: Deve ser digitado um número dentre as possíveis opções (1, 2 ou 3)
  4. Pós-condições:
  5. Resultado: Direcionamento para a funcionalidade desejada.
  6. Cenário principal:
  7. Cenário alternativo:

1. Verificar se um número é negativo ou positivo.
   1. Identificador: B.
   2. Nome: Verificar se um negativo ou positivo.
   3. Autores: Karina Sayuri Hagiwara.
   4. Prioridade: Alta.
   5. Criticalidade: -
   6. Fonte: Caroline Resende.
   7. Responsável: Karina Sayuri Hagiwara.
   8. Descrição: O sistema deverá verificar se o número digitado é negativo ou positivo.
   9. Trigger: -
   10. Atores: Karina Sayuri Hagiwara.
   11. Pré-condições: Deve ser digitado um número.
   12. Pós-condições:
   13. Resultado: Saber se um número é positivo ou negativo.
   14. Cenário principal:
   15. Cenário alternativo:
2. Verificar se um número é par ou ímpar.
   1. Identificador: C.
   2. Nome: Verificar se um número é par ou ímpar.
   3. Autores: Karina Sayuri Hagiwara.
   4. Prioridade: Alta.
   5. Criticalidade: -
   6. Fonte: Caroline Resende.
   7. Responsável: Karina Sayuri Hagiwara.
   8. Descrição: O sistema deverá verificar se o número digitado é par ou ímpar.
   9. Trigger: -
   10. Atores: Karina Sayuri Hagiwara.
   11. Pré-condições: Deve ser digitado um número inteiro.
   12. Pós-condições:
   13. Resultado: Saber se um número é par ou ímpar.
   14. Cenário principal:
   15. Cenário alternativo:
3. Imprimir resultado
   1. Identificador: D.
   2. Nome: Imprimir resultado.
   3. Autores: Karina Sayuri Hagiwara.
   4. Prioridade: Alta.
   5. Criticalidade: -
   6. Fonte: Caroline Resende.
   7. Responsável: Karina Sayuri Hagiwara.
   8. Descrição: O sistema deverá imprimir o resultado da verificação que ocorreu nas funcionalidades B (verificar se um número é positivo ou negativo) e C (verificar se um número é par ou ímpar).
   9. Trigger: Imprimir na tela o resultado das funcionalidades B (verificar se um número é positivo ou negativo) ou C (verificar se um número é par ou ímpar) para que seja possível a vizualização deste pelo usuário.
   10. Atores: Karina Sayuri Hagiwara.
   11. Pré-condições: A funcionalidades B (verificar se um número é positivo ou negativo) ou a C (verificar se um número é par ou ímpar) deve ter sido executada.
   12. Pós-condições:
   13. Resultado: Impressão do resultado na tela.
   14. Cenário principal:
   15. Cenário alternativo:
4. Verificar se um número é maior ou menor que 0.
   1. Identificador: E.
   2. Nome: Verificar se um número é maior ou menor que 0.
   3. Autores: Karina Sayuri Hagiwara.
   4. Prioridade: Alta.
   5. Criticalidade: -
   6. Fonte: Caroline Resende.
   7. Responsável: Karina Sayuri Hagiwara.
   8. Descrição: O sistema deverá verificar se um número é maior ou menor que 0, pois caso seja maior que 0, este é positivo, no entanto, caso seja menor que 0, este é negativo.
   9. Trigger: A funcionalidade B (verificar se um número é positivo ou negativo) ao ser executada executa a funcionalidade verificar se um número é maior ou menor que 0, pois este possibitará a funcionalidade de verificar se um número é positivo ou negativo.
   10. Atores: Karina Sayuri Hagiwara.
   11. Pré-condições: A funcionalidades B (verificar se um número é positivo ou negativo) deve ter sido executada.
   12. Pós-condições:
   13. Resultado: Saber se um número é maior ou menor que 0.
   14. Cenário principal:
   15. Cenário alternativo:
5. Verificar se um número é divisível por 2.
   1. Identificador: F.
   2. Nome: Verificar se um número é divisível por 2.
   3. Autores: Karina Sayuri Hagiwara.
   4. Prioridade: Alta.
   5. Criticalidade: -
   6. Fonte: Caroline Resende.
   7. Responsável: Karina Sayuri Hagiwara.
   8. Descrição: O sistema deverá verificar se um número é divisível por 2, pois caso seja divisível por 2, este é par, no entanto, caso não seja, este é ímpar.
   9. Trigger: A funcionalidade C (verificar se um número é ímpar ou par) ao ser executada executa a funcionalidade verificar se um número é divisível por 2, pois este possibitará a funcionalidade de verificar se um número é ímpar ou par.
   10. Atores: Karina Sayuri Hagiwara.
   11. Pré-condições: A funcionalidades C (verificar se um número é ímpar ou par) deve ter sido executada.
   12. Pós-condições:
   13. Resultado: Saber se um número é divisível por 2.
   14. Cenário principal:
   15. Cenário alternativo:

Diagrama de fluxo de dados



Glossários

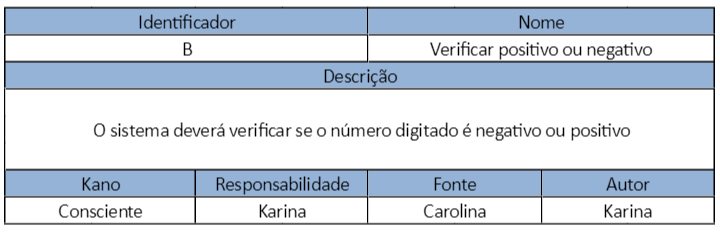
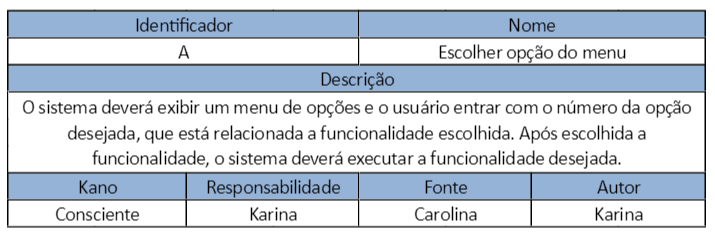
Índice

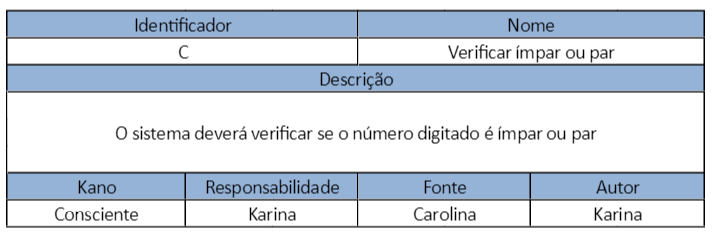
**DOCUMENTO DE ESPECIFICAÇÃO SUPLEMENTAR**

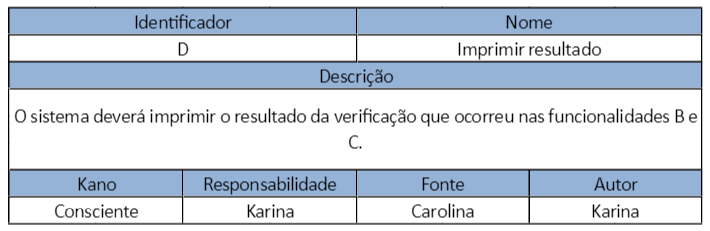
**Priorização de kano**

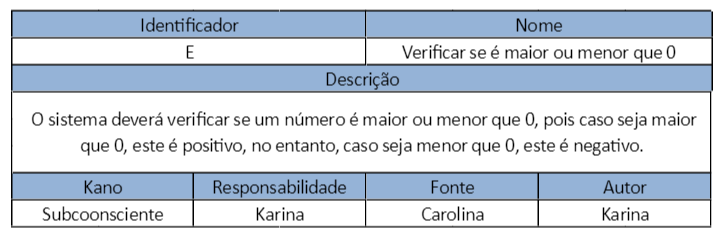
1. Fatores esperados (Requsitios conscientes)
   1. Funcionalidade “Verificar se é negativo ou positivo”
   2. Funcionalidade “Verificar se é par ou ímpar"
   3. Funcionalidade “Menu”
   4. Funcionalidade "Imprimir resultado"
2. Fatores inesperados (Requisitos inconscientes)
   1. **Tratamento/mensagem de erro**
3. Fatores básicos (Requisitos subconscientes)
   1. Enquanto ele não desejar sair do programa, este deve continuar aberto.
   2. Verificar se o número é divisível por 2
   3. Verificar se o número é menor ou maior que 0
   4. Verificar se o usuário realmente deseja sair

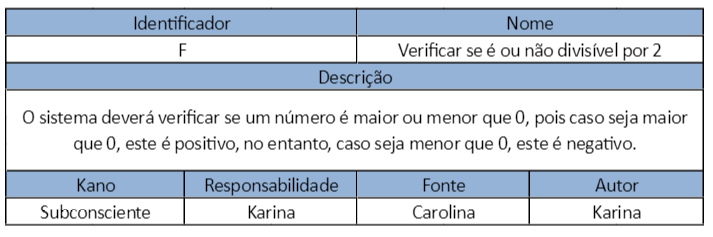
Tabelas (cartões de atributo)











Matriz de rastreabilidade

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F |
| A |  | R | R | R | R | R |
| B | D |  |  | R |  |  |
| C | D | R |  | R |  |  |
| D | R | D | D |  |  |  |
| E | R | D |  | R |  |  |
| F | R |  | D | R |  |  |

**DOCUMENTO DE PLANEJAMENTO E ACOMPANHAMENTO**

Cálculo do ponto de função

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Contagem |  | Simples | Médio | Complexo |  |  |
| Entrada externa | **4** | x | 3 | **4** | 6 | = | **16** |
| Saída externa | **2** | x | 4 | **5** | 7 | = | **10** |
| Consulta externa | 0 | x | 3 | 4 | 6 | = | 0 |
| Arquivo lógico interno | 0 | x | 7 | 10 | 15 | = | 0 |
| Arquivo de interface externa | 0 | x | 5 | 7 | 10 | = | 0 |

Contagem total = 26

Fatores de ajuste

1. O sistema requer salvamento e recuperação confiáveis? 0
2. São necessárias comunicações de dados especializadas? 0
3. Há funções de processamento distribuído? 0
4. O sistema rodará em ambiente operacional existente e intensamente utilizado? 2
5. O desempenho é crítico ? 3
6. O sistema requer entrada de dados online ? 5
7. A entrada de dados online requer múltiplas telas ou operações ? 2
8. Os Arquivos Lógicos Internos são atualizados online ? 0
9. As entradas, saídas e consultas são complexas ? 3
10. O processamento interno é complexo ? 3
11. O código é projetado para ser reutilizável ? 5
12. A instalação está incluída no projeto ? 3
13. O sistema é projetado para múltiplas instalações em diferentes organizações ? 0
14. A aplicação é projetada para facilitar a troca e o uso pelo usuário ? 5

Contagem total do fator de ajuste = 31

FP = 26\* [0,65 + 0,01 \* ∑ Fatores de ajuste]

FP = 26 \* [0,65 + 0,01 \* 31]

FP = 26\* [0,65 + 0,31]

FP = 24,96

Estimativa (Esforço, prazo e custo)

As medidas de produtividade em PF/mês e R$/mês variam conforme o analista/desenvolvedor, dessa forma, para esta estimativa será utilizado 20 PF/mês para um analista programador c júnior.

Cargo: Júnior

Produtividade: 20 FP/mês

O salário mensal de um analista programador c júnior é R$3000,00 [fonte: http://www.trainning.com.br/pagina/salarios?gclid=CK7p1v38qM0CFVQIkQodjlsIaA, Trainning Education]

Portanto:

Esforço: 20 FP / mês

Prazo: 24,96 FP /(20 FP / mês) = 1,248 meses

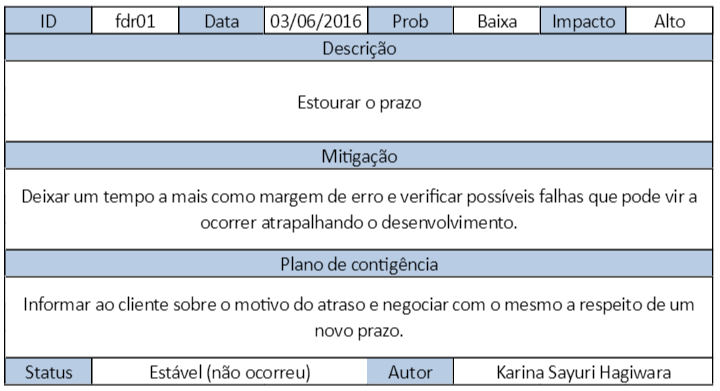
Custo: R$3000,00 \* 1,248 meses = R$3744,00

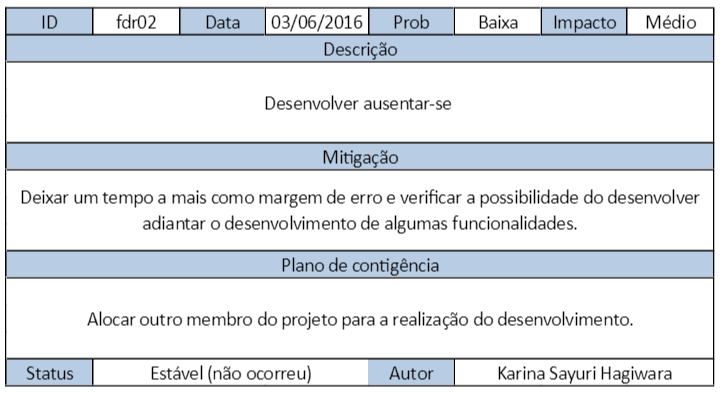
Diagrama Gantt de controle

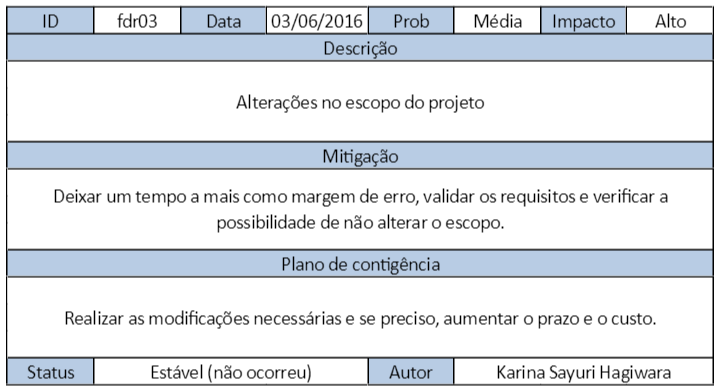
Lista de riscos:

1. Estourar o prazo
2. Desenvolver ausentar-se
3. Alterações no escopo do projeto

Formulário de riscos







Manual do usuário

Sumário

Objetivo

Menu

Ao iniciar o sistema, aparecerá um menu com as opções que o sistema possui. Haverá três possibilidades:

1. Verificar se o número digitado é negativo ou positivo;
2. Verificar se o número digitado é par ou ímpar;
3. Sair do programa.

Você deverá escolher entre as opções e digitar o número referente a opção que deseja realizar e apertar a tecla enter, fazendo com que a opção desejada seja executada e você seja redirecionado para a mesma. A opção número 1 fará com que o sistema verifique se o número digitado é negativo ou postivo. Já a opção número 2 fará com que o sistema verifique se o número digitado é ímpar ou par. Já a opção número 3 fará com que saia do programa.

(Foto do menu e seta apontando para a entrada do usuário)

Observa-se que, é necessário que o número digitado para a escolha da opção esteja entre os números 1, 2 e 3 que são refentes as opções presentes.

(Foto do erro da func)

Verificar se o número digitado é negativo ou positivo

Para verificar se o número é positivo ou negativo, basta digitar o número que deseja verificar e apertar a tecla enter. Após o sistema verificar se o número é positivo ou negativo, o sistema exibirá o resultado da verificação informando se o número é positivo ou negativo.

(Foto da func e seta para a entrada)

Deve-se notar que, é necessário que está verificação é válida apenas para números.

(Foto do erro da func)

Verificar se o número digitado é ímpar ou par

Para verificar se o número é ímpar ou par, basta digitar o número que deseja verificar e apertar a tecla enter. Após o sistema verificar se o número é ímpar ou par, o sistema exibirá o resultado da verificação informando se o número é ímpar ou par.

(Foto da func e seta para a entrada)

Deve-se notar que, é necessário que está verificação é válida apenas para números inteiros.

(Foto do erro da func)

Sair do programa

Ao escolher a opção 3 do sistema, aparecerá a mensagem "Você realmente deseja sair do programa? Digite 1 caso sim ou digite 0 caso contrário.", portanto, para sair do programa, basta digitar 1 caso você realmente queira sair do programa e o sistema será finalizado ou, basta digitar 0 caso você queira continuar no programa e o sistema irá redirecionar para o menu.

(foto sair)

Observa-se que, é necessário que o número digitado esteja entre os números 1 e 0 que são refentes as opções presentes.

(Foto do erro da func)

Dúvidas

Plano de teste

Introdução

Este documento trata dos testes do Sistema PINP para determinar se este atingiu suas especificações e funcionou corretamente no ambiente para o qual foi projetado, levando em consideração os pré-requisitos do Documento de Especificação. Para realizar os testes serão utilizados métodos e critérios de cobertura, os quais serão especificados neste documento.

Roteiro de teste

1. ID: tst01

Caso de teste: Na funcionalidade verificar se o número é maior ou menor que 0, utilizar um número maior que 0.

Roteiro de teste: 8

Resultado esperado: É maior que 0

1. ID: tst02

Caso de teste: Na funcionalidade verificar se o número é maior ou menor que 0, utilizar um número menor que 0.

Roteiro de teste: -100

Resultado esperado: É menor que 0

1. ID: tst03

Caso de teste: Na funcionalidade verificar se o número é maior ou menor que 0, utilizar um número com decimais.

Roteiro de teste: -50.50

Resultado esperado: É menor que 0

1. ID: tst04

Caso de teste: Na funcionalidade verificar se o número é maior ou menor que 0, utilizar o número 0.

Roteiro de teste:0

Resultado esperado: É igual a 0

1. ID: tst05

Caso de teste: Na funcionalidade verificar se o número é negativo ou positivo, utilizar um número positivo.

Roteiro de teste: 100

Resultado esperado: É positivo

1. ID: tst06

Caso de teste: Na funcionalidade verificar se o número é negativo ou positivo, utilizar um número negativo.

Roteiro de teste: -8

Resultado esperado: É negativo

1. ID: tst07

Caso de teste: Na funcionalidade verificar se o número é negativo ou positivo, utilizar um número com decimais.

Roteiro de teste: 50.50

Resultado esperado: É positivo

1. ID: tst08

Caso de teste: Na funcionalidade verificar se o número é negativo ou positivo, utilizar o número 0.

Roteiro de teste: 0

Resultado esperado: É neutro

1. ID: tst09

Caso de teste: Na funcionalidade verificar se o número é ou não divisível por 2, utilizar um número divisível por 2.

Roteiro de teste: 12

Resultado esperado: É divisível por 2

1. ID: tst10

Caso de teste: Na funcionalidade verificar se o número é ou não divisível por 2, utilizar um número não divisível por 2.

Roteiro de teste: -11

Resultado esperado: Não é divisível por 2

1. ID: tst11

Caso de teste: Na funcionalidade verificar se o número é ou não divisível por 2, utilizar um número com decimais.

Roteiro de teste: 25.25

Resultado esperado: Digite um número inteiro

1. ID: tst12

Caso de teste: Na funcionalidade verificar se o número é ou não divisível por 2, utilizar o número 0.

Roteiro de teste: 0

Resultado esperado: É divisível por 2

1. ID: tst13

Caso de teste: Na funcionalidade verificar se é par ou ímpar, utilizar um número par.

Roteiro de teste: -902

Resultado esperado: É par

1. ID: tst14

Caso de teste: Na funcionalidade verificar se é par ou ímpar, utilizar um número ímpar.

Roteiro de teste: 45

Resultado esperado: É ímpar

1. ID: tst15

Caso de teste: Na funcionalidade verificar se é par ou ímpar, utilizar um número com casas decimais

Roteiro de teste: -25.25

Resultado esperado: Digite um número inteiro

1. ID: tst16

Caso de teste: Na funcionalidade verificar se é par ou ímpar, utilizar o número 0.

Roteiro de teste: 0

Resultado esperado: É par

1. ID: tst17

Caso de teste: Na funcionalidade escolher opção do menu, utilizar os números 1, 2 ou 3.

Roteiro de teste: 1

Resultado esperado: Redirecionamento para a funcionalidade.

1. ID: tst18

Caso de teste: Na funcionalidade escolher opção do menu, utilizar os números 1, 2 ou 3.

Roteiro de teste: 2

Resultado esperado: Redirecionamento para a funcionalidade.

1. ID: tst19

Caso de teste: Na funcionalidade escolher opção do menu, utilizar os números 1, 2 ou 3.

Roteiro de teste: 3

Resultado esperado: Sair do programa.

1. ID: tst20

Caso de teste: Na funcionalidade escolher opção do menu, utilizar qualquer número exceto 1, 2 ou 3.

Roteiro de teste: 10

Resultado esperado: Não existe esta opção, digite um número entre as opções 1, 2 ou 3.

Estratégias

Critério de cobertura

Para a realização destes testes foi utilizado a abordagem bottom-up, error guessing, técnica de particionamento por equivalência e análise do valor limite.

Teste de Componente

Para o teste de componente (unitário) foram realizados testes para cada funcionalidade: verificar se um numero é positivo ou negativo, verificar se um número é par ou ímpar, verificar se o número é maior ou menor que 0, verificar se o número é ou não divisível por 2, imprimir resultado e escolher opção do menu.

3.2. Teste de Integração

Para o teste de integração foram realizados testes que verificam a integração do sistema. Para a realização destes testes foi utilizado a técnica de análise do valor limite.

3.3. Teste de Sistema

Para o teste de sistema, o sistema foi testado como um todo.

3.4. Teste de Aceite  
Para o teste de aceite foram realizados os testes:

Teste Alfa (α):

O teste realizado no ambiente de desenvolvimento.

Teste Beta (β):

O teste realizado na máquina do cliente.

Relatório de defeitos

1. ID: rdd01

Caso de teste: tst04

Procedimento: Tratamento do erro no código.

Evidência: Resposta inesperada "É maior que 0", ao invés do resultado esperado "É igual a 0".

1. ID: rdd02

Caso de teste: tst08

Procedimento: Tratamento do erro no código.

Evidência: Resposta inesperada "É positivo", ao invés do resultado esperado "É neutro".

Termo de homologação

Nome do sistema: Sistema PINP

Descrição do sistema: O sistema é responsável por verificar se um número é negativo ou positivo, este também é responsável por verificar se um número é ímpar ou par.

Descrição de ajustes ou alterações:

Atesto que o sistema foi avaliado e aprovado por estar em conformidade com a especificação.

Assinatura: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_