

Documentação Testes Utilizando SPEC

Gerado por Doxygen 1.8.6

Sábado, 8 de Julho de 2017 01:04:25

Sumário

1	Índice dos Arquivos	1
1.1	Lista de Arquivos	1
2	Arquivos	3
2.1	Referência do Arquivo teste_spec.c	3
2.1.1	Funções	3
2.1.1.1	describe	3
	Índice	12

Capítulo 1

Índice dos Arquivos

1.1 Lista de Arquivos

Esta é a lista de todos os arquivos e suas respectivas descrições:

teste_spec.c	3
--	---

Capítulo 2

Arquivos

2.1 Referência do Arquivo teste_spec.c

Funções

- `describe` ("Testes Utilizando Specs")

2.1.1 Funções

2.1.1.1 `describe` ("Testes Utilizando Specs")

Diagrama da funcao ProcuraNomeUsuario

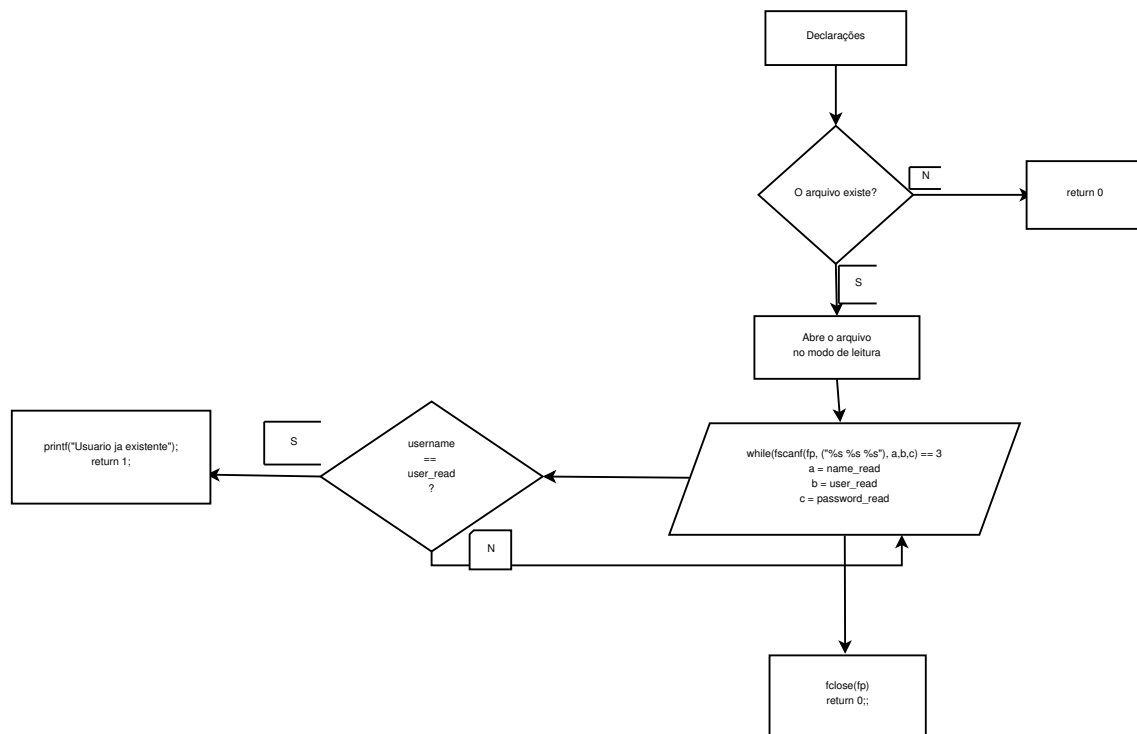


Figura 2.1: Fluxograma

Nessa função optamos por testar seguindo o critério de cobertura por decisões, onde checamos se a funcao

ProcuraNomeUsuario responde como esperado quando a entrada equivale a um usuario já existente. A escolha de critério de cobertura ser por decisão foi basicamente devido ao fato de que os trechos críticos para o bom funcionamento está na forma que a entrada se relaciona com os condicionais, enquanto o restante das instruções presentes basicamente realizam interface com o usuario.

Diagrama da funcao ValidarDisciplina

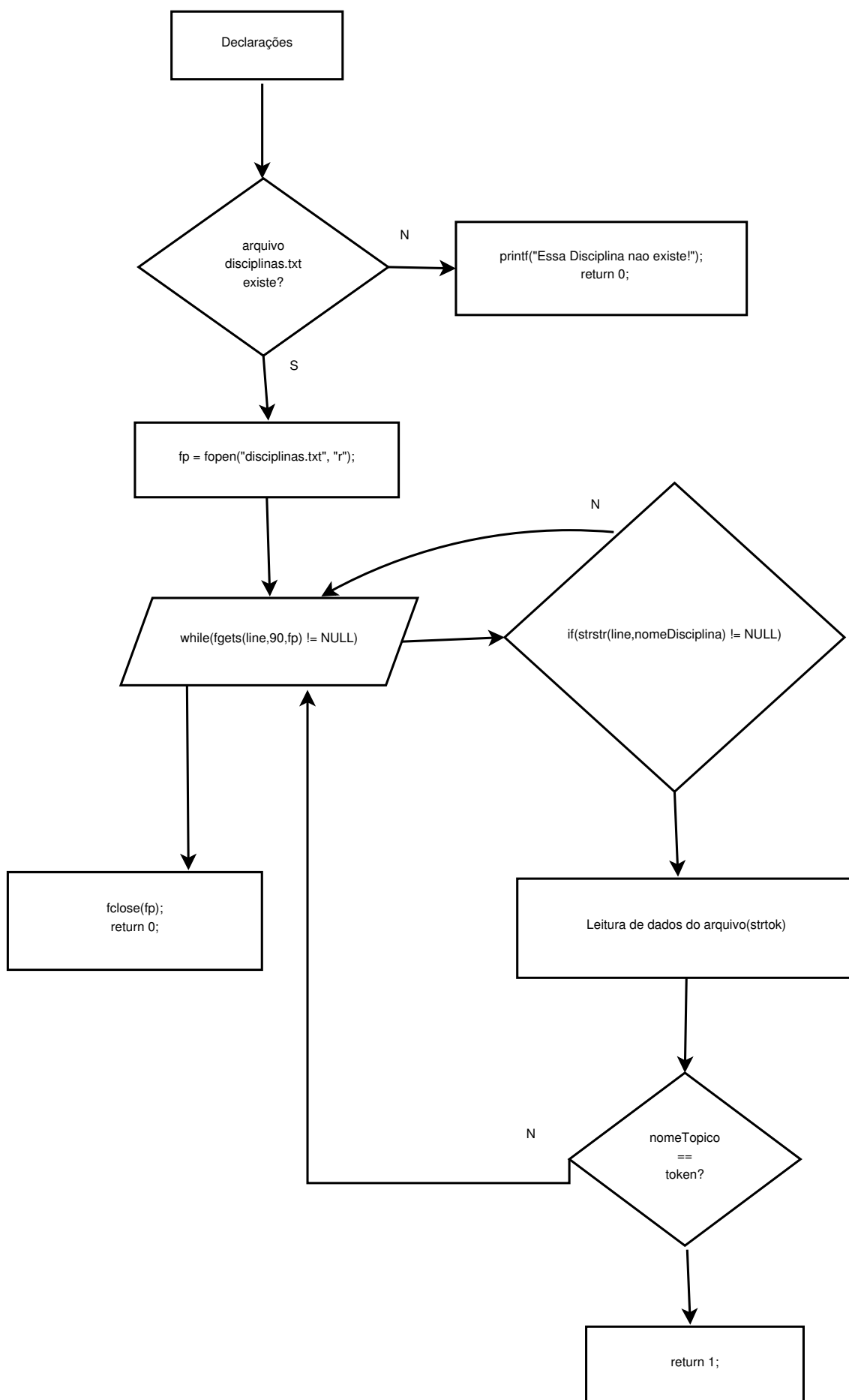


Figura 2.2: Fluxograma

No teste dessa função também utilizamos o critério de cobertura de decisões, onde checamos o comportamento da função ValidarDisciplina quando a entrada é composta de uma disciplina e um tópico ambos válidos, e também quando não são válidos. Assim, Avaliamos se a função realiza corretamente o seu * papel de determinar a validade/invalidade de uma dupla disciplina e tópico. A justificativa por trás da cobertura ser de decisão também é devido a falta de * necessidade de realizar uma cobertura de instruções, uma vez que o bom funcionamento da função está atrelada ao comportamento dos condicionais.

Diagrama da funcao TelaLogin

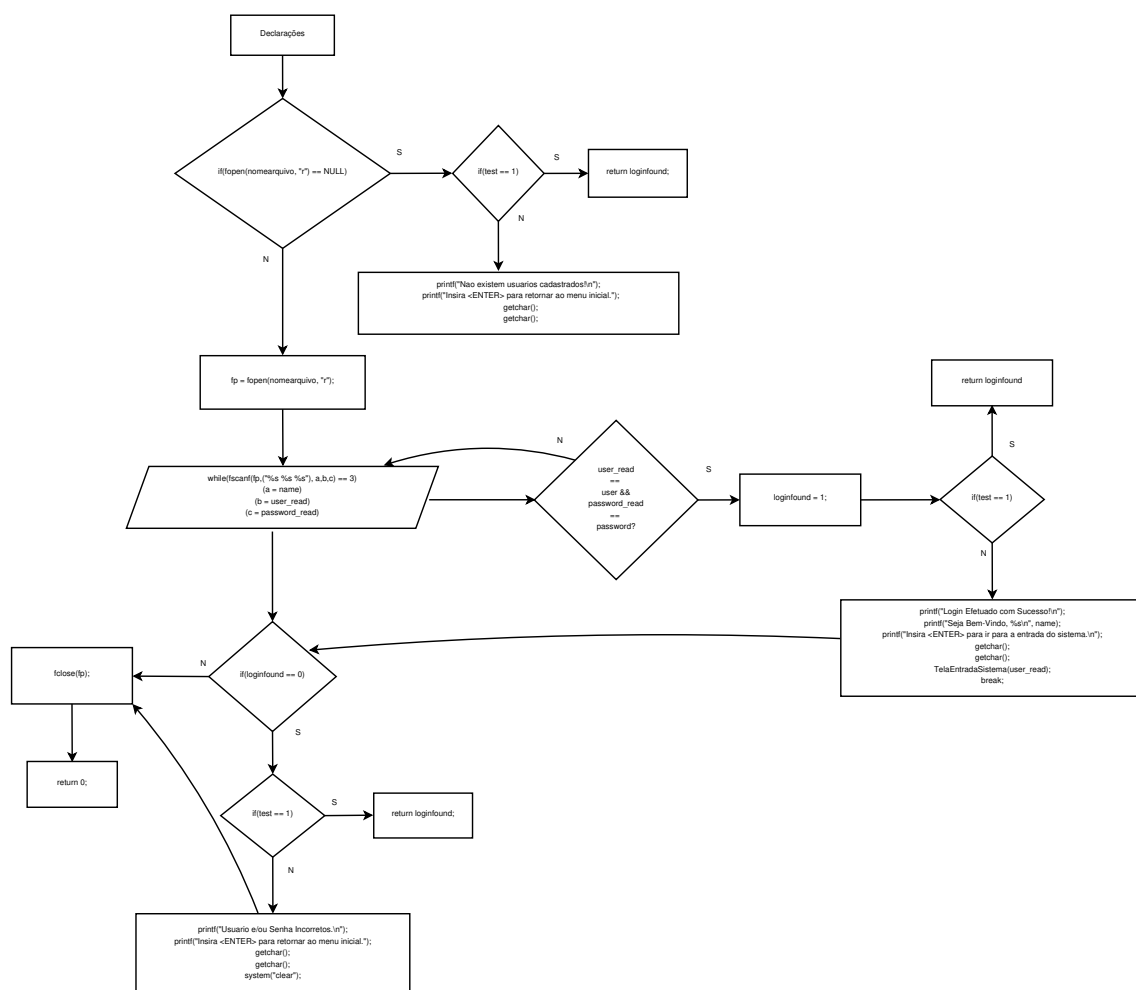


Figura 2.3: Fluxograma

Pelo critério de cobertura de decisões, checamos se a função TelaLogin responde cada classe de entrada como deveria. Assim, inserimos três classes de entrada para checarmos o funcionamento da função: entrada válida, entrada inválida quanto ao usuário, entrada inválida quanto ao arquivo.

Diagrama da funcao TelaCadastro

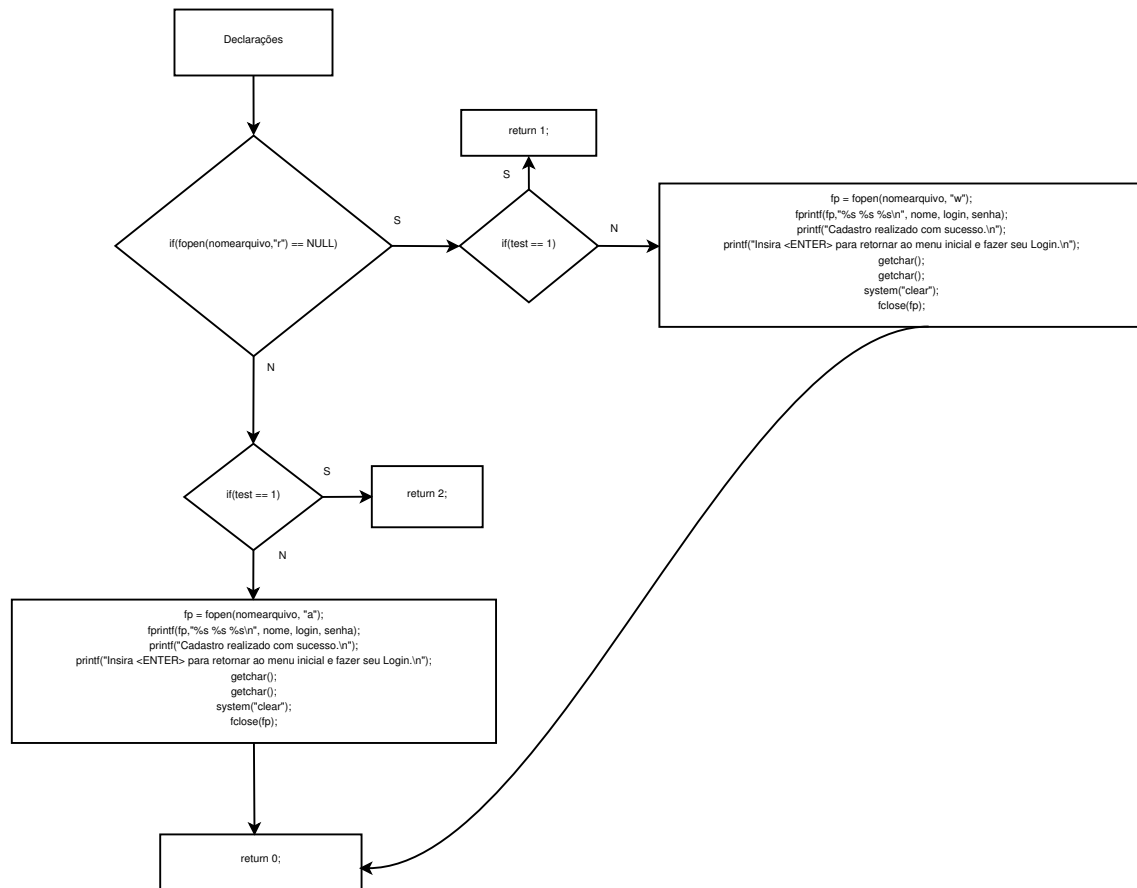


Figura 2.4: Fluxograma

Seguindo o critério de cobertura por decisões, checamos se a função `TelaCadastro` realiza o cadastro de um usuário corretamente quando o arquivo de usuários ainda não existe, e também se ele incrementa o arquivo com o novo usuário quando o arquivo já existe. Novamente, a cobertura por instruções seria desnecessária por demandar uma quantidade maior de recursos sendo que a avaliação do desempenho das funções depende unicamente da correção dos condicionais.

Diagrama da função `TelaRecuperarSenha`

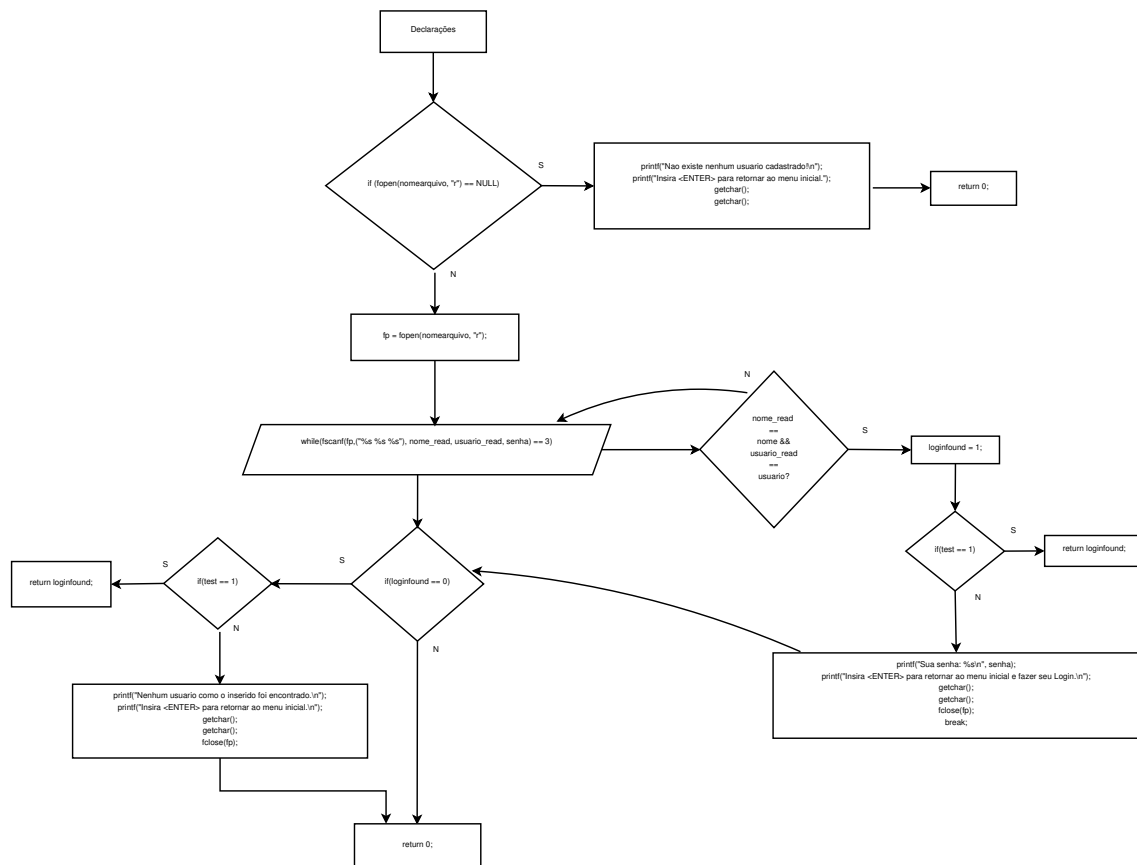


Figura 2.5: Fluxograma

Também seguindo o critério de cobertura por decisões, checamos se a função TelaRecuperarSenha responde como esperado quando a entrada se trata de um usuário existente, bem como quando se trata de um usuário inexistente. Novamente, é interessante trabalhar com classes válidas e inválidas para determinar a relação de correção entre entradas e condicionais.

Diagrama da função InserirLista

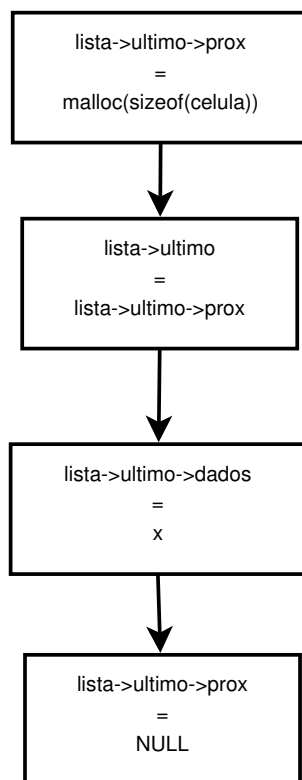


Figura 2.6: Fluxograma

Pelo critério de cobertura de instruções, checamos se todas as instruções da função `InsererLista` são exercitadas.

Diagrama da função `DesalocarLista`

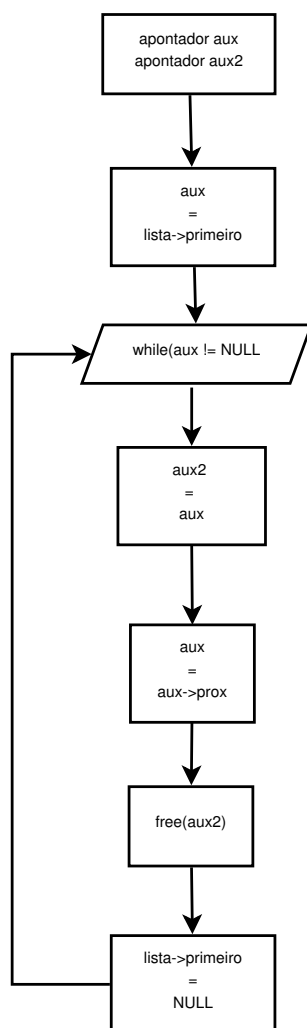


Figura 2.7: Fluxograma

Pelo critério de cobertura de instruções, checamos se todas instruções de DesalocarLista são devidamente exercitadas

Diagrama da funcao CriaLista

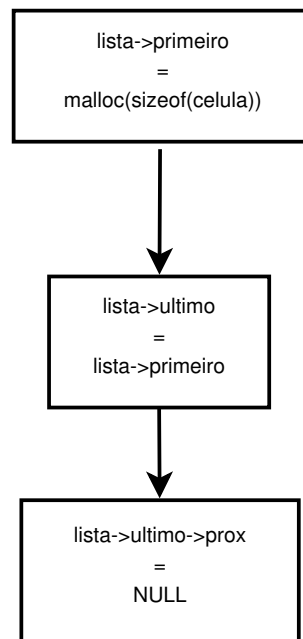


Figura 2.8: Fluxograma

Pelo critério de cobertura de instruções, checamos se todas instruções de CriaLista são devidamente exercitadas

Índice Remissivo

describe

teste_spec.c, [3](#)

teste_spec.c, [3](#)

describe, [3](#)