



INSTITUTO FEDERAL DE PERNAMBUCO
CURSO TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Plano de Testes

Running Food

Versão 1.6

PAULISTA - 2023

ARTHUR VINÍCIUS BEZERRA DA SILVA
BEATRIZ GONÇALVES FONTES
ERIDIANA ANGÉLICA DIAS DO NASCIMENTO
MARIA LETÍCIA DA SILVA MANGUINHO
VITÓRIA CARLA COSTA DE LIRA

Histórico de Alterações

Data	Versão	Descrição	Autor
30/05/2023	1.0	Definição e descrição dos casos de testes	<i>Arthur Vinicius da Silva Bezerra; Beatriz Gonçalves Fontes; Maria Letícia da Silva Manguinho; Vitória Carla Costa de Lira.</i>
31/05/2023	1.1	Realização e documentação de testes	<i>Arthur Vinicius da Silva Bezerra; Beatriz Gonçalves Fontes;</i>
01/06/23	1.2	Realização e documentação de testes	<i>Maria Letícia da Silva Manguinho; Vitória Carla Costa de Lira</i>
02/06/23	1.3	Realização e documentação de testes	<i>Arthur Vinicius da Silva Bezerra</i>
03/06/2023	1.4	Realização e documentação de testes	<i>Beatriz Gonçalves Fontes; Maria Letícia da Silva Manguinho; Vitória Carla Costa de Lira</i>
06/06/2023	1.5	Realização e documentação de testes	<i>Arthur Vinicius da Silva Bezerra; Beatriz Gonçalves Fontes; Maria Letícia da Silva Manguinho; Vitória Carla Costa de Lira.</i>
07/06/2023	1.6	Conclusão	<i>Arthur Vinicius da Silva Bezerra; Beatriz Gonçalves Fontes; Maria Letícia da Silva Manguinho; Vitória Carla Costa de Lira.</i>

Conteúdo

1 Introdução.....	4
2 Tipos de Teste.....	5
2.1 Iteração 1.....	5
2.2 Iteração 2.....	5
2.3 Iteração 3.....	6
2.4 Iteração 4.....	6
2.5 Iteração 5.....	6
2.6 Iteração 6.....	7
2.7 Iteração 7.....	7
2.8 Iteração 8.....	7
2.9 Iteração 9.....	8
2.10 Iteração 10.....	8
3 Recursos.....	9
3.1 Ambiente de Teste – Software & Hardware.....	9
3.2 Ferramentas de Teste.....	9
4 Cronograma.....	10

1 Introdução

No processo de desenvolvimento de um jogo utilizando-se da linguagem de programação python, juntamente com sua biblioteca pygame, foi necessário testar algumas funcionalidades, casos de testes, para a ótima execução do jogo, bem como a entrega de uma boa qualidade de código, o qual todas as demonstrações desses mesmos testes estão devidamente presentes em um repositório online no github presente neste link: <https://github.com/MariaManguinho/TESTESDESOFTWARE>.

No decorrer da criação do jogo foram adicionadas algumas features a fim de melhorar o jogo e retirar possíveis bugs, que por conseguinte, iria interferir na jogabilidade e experiência do usuário ao longo de sua trajetória.

Diante do contexto, tornou-se necessário a realização de mais testes para garantir a qualidade do jogo e a satisfação do usuário. Os testes foram realizados de forma manual em um ambiente de teste controlado e supervisionado pelos seus desenvolvedores, utilizando uma abordagem de teste em caixa branca de modo unitário, conforme descrito mais abaixo. Além disso, foi criado um cronograma, o qual foi seguido a fim de executar os planos de testes devidamente descritos e elaborados para cada caso de teste, conforme a data estabelecida na tabela de cronograma. Tendo os testes iterados cinco vezes e tornando a média das assertividades na coluna resultados em forma de porcentagem que se encontra com mais detalhes no decorrer deste documento.

2 Tipos de Teste

2.1 Iteração 1

Objetivo:	<i>Verificar se o jogador terá apenas as 3 vidas para a realização do jogo, juntamente com a mudança de cores dos corações que conforme as perdas devem mudar de vermelho para branco.</i>
Técnica:	<input checked="" type="checkbox"/> Manual <input type="checkbox"/> Automática
Estágio do teste: <input type="checkbox"/> Integração <input type="checkbox"/> Sistema <input checked="" type="checkbox"/> Unidade <input type="checkbox"/> Aceitação	Abordagem do teste <input checked="" type="checkbox"/> Caixa branca <input type="checkbox"/> Caixa preta
Responsável(is):	<i>Arthur Vinícius Bezerra da Silva</i>

2.2 Iteração 2

Objetivo:	<i>O timer no canto superior direito deve decrementar iniciando do 120 até o 0, conforme o tempo passa, possuindo um tempo total de 2 minutos para o cumprimento do jogo.</i>
Técnica:	<input checked="" type="checkbox"/> Manual <input type="checkbox"/> Automática
Estágio do teste: <input type="checkbox"/> Integração <input type="checkbox"/> Sistema <input checked="" type="checkbox"/> Unidade <input type="checkbox"/> Aceitação	Abordagem do teste <input checked="" type="checkbox"/> Caixa branca <input type="checkbox"/> Caixa preta
Responsável(is):	<i>Beatriz Gonçalves Fontes</i>

2.3 Iteração 3

Objetivo:	<i>A música da aplicação deve mudar de acordo com a situação do jogador. Sendo a música de background do jogo em si a “musica_fundo”, a de jogo finalizado com sucesso “win” e em caso de game over todos os sons devem ser encerrados.</i>
Técnica:	<input checked="" type="checkbox"/> Manual <input type="checkbox"/> Automática
Estágio do teste: <input type="checkbox"/> Integração <input type="checkbox"/> Sistema <input checked="" type="checkbox"/> Unidade <input type="checkbox"/> Aceitação	Abordagem do teste <input checked="" type="checkbox"/> Caixa branca <input type="checkbox"/> Caixa preta
Responsável(is):	<i>Maria Letícia da Silva Manguinho</i>

2.4 Iteração 4

Objetivo:	<i>Confirmar se o jogo foi encerrado, uma vez que o jogador perdeu todas as vidas, devendo ser exibida a imagem de game over.</i>
Técnica:	<input checked="" type="checkbox"/> Manual <input type="checkbox"/> Automática
Estágio do teste: <input type="checkbox"/> Integração <input type="checkbox"/> Sistema <input checked="" type="checkbox"/> Unidade <input type="checkbox"/> Aceitação	Abordagem do teste <input checked="" type="checkbox"/> Caixa branca <input type="checkbox"/> Caixa preta
Responsável(is):	<i>Vitória Carla Costa de Lira</i>

2.5 Iteração 5

Objetivo:	<i>Validar se o jogador conseguiu finalizar as entregas nas 3 casas, consequentemente, ganhando e encerrando o jogo, juntamente com a imagem final.</i>
Técnica:	<input checked="" type="checkbox"/> Manual <input type="checkbox"/> Automática
Estágio do teste: <input type="checkbox"/> Integração <input type="checkbox"/> Sistema <input checked="" type="checkbox"/> Unidade <input type="checkbox"/> Aceitação	Abordagem do teste <input checked="" type="checkbox"/> Caixa branca <input type="checkbox"/> Caixa preta
Responsável(is):	<i>Arthur Vinícius Bezerra da Silva</i>

2.6 Iteração 6

Objetivo:	<i>Conferir se o jogador não consegue ultrapassar determinados cenários do jogo, como, por exemplo, a grama, o cercado das casas, as extremidades da tela, etc.</i>
Técnica:	<input checked="" type="checkbox"/> Manual <input type="checkbox"/> Automática
Estágio do teste: <input type="checkbox"/> Integração <input type="checkbox"/> Sistema <input checked="" type="checkbox"/> Unidade <input type="checkbox"/> Aceitação	Abordagem do teste <input checked="" type="checkbox"/> Caixa branca <input type="checkbox"/> Caixa preta
Responsável(is):	<i>Beatriz Gonçalves Fontes</i>

2.7 Iteração 7

Objetivo:	<i>Constatar que os carros existentes como empecilhos no jogo, a cada nova execução aparecem em ordens diferentes e velocidade. Validar também o sumiço dos carros devendo ser de forma gradativa quando encostam no limite da tela.</i>
Técnica:	<input checked="" type="checkbox"/> Manual <input type="checkbox"/> Automática
Estágio do teste: <input type="checkbox"/> Integração <input type="checkbox"/> Sistema <input checked="" type="checkbox"/> Unidade <input type="checkbox"/> Aceitação	Abordagem do teste <input checked="" type="checkbox"/> Caixa branca <input type="checkbox"/> Caixa preta
Responsável(is):	<i>Maria Letícia da Silva Manguinho</i>

2.8 Iteração 8

Objetivo:	<i>Certificar que o jogador retorna ao ponto inicial do jogo a cada vida que perde.</i>
Técnica:	<input checked="" type="checkbox"/> Manual <input type="checkbox"/> Automática
Estágio do teste: <input type="checkbox"/> Integração <input type="checkbox"/> Sistema <input checked="" type="checkbox"/> Unidade <input type="checkbox"/> Aceitação	Abordagem do teste <input checked="" type="checkbox"/> Caixa branca <input type="checkbox"/> Caixa preta
Responsável(is):	<i>Vitória Carla Costa de Lira</i>

2.9 Iteração 9

Objetivo:	<i>Verificar se o jogador muda a sua posição e a direção de sua mobilidade, de acordo com as teclas do teclado responsáveis pelo comando.</i>
Técnica:	<input checked="" type="checkbox"/> Manual <input type="checkbox"/> Automática
Estágio do teste: <input type="checkbox"/> Integração <input type="checkbox"/> Sistema <input checked="" type="checkbox"/> Unidade <input type="checkbox"/> Aceitação	Abordagem do teste <input checked="" type="checkbox"/> Caixa branca <input type="checkbox"/> Caixa preta
Responsável(is):	<i>Arthur Vinicius da Silva Bezerra Beatriz Gonçalves Fontes Maria Leticia da Silva Manguinho Vitória Carla Costa de Lira</i>

2.10 Iteração 10

Objetivo:	<i>Todo o jogo deve ser encerrado ao ser clicado no “X” no canto superior esquerdo da tela, efetuando assim a ação de QUIT.</i>
Técnica:	<input checked="" type="checkbox"/> Manual <input type="checkbox"/> Automática
Estágio do teste: <input type="checkbox"/> Integração <input type="checkbox"/> Sistema <input checked="" type="checkbox"/> Unidade <input type="checkbox"/> Aceitação	Abordagem do teste <input checked="" type="checkbox"/> Caixa branca <input type="checkbox"/> Caixa preta
Responsável(is):	<i>Arthur Vinicius da Silva Bezerra Beatriz Gonçalves Fontes Maria Leticia da Silva Manguinho Vitória Carla Costa de Lira</i>

3 Recursos

3.1 Ambiente de Teste – Software & Hardware

Como ambiente de testes no âmbito do software, tem-se a versão 10 do sistema operacional Windows, bem como, a versão 3.9.7 do python, linguagem de programação utilizada na aplicação, além da versão 22.3.1 do instalador de pacotes do python, pip.

No enfoque do hardware tem-se a seguinte configuração da máquina utilizada para rodar os testes manuais: Processador Intel I5-10210U contendo 4 núcleos e 8 threads e com 6MB de cache, além de operar entre 1.6 GHz até 4.2 GHz. Como a aplicação possui a questão gráfica, fora utilizado como placa de vídeo uma Intel UHD Graphics 620, além do mais o ambiente conta com 8GB de RAM DDR4 operando a 2666MHz em dual channel. Já o armazenamento do ambiente de teste é um SSD do tipo NVME Gen3x2 de 256GB e como input temos apenas o uso do teclado e mouse.

Com relação aos recursos humanos necessários para a realização e documentação dos testes, foram imprescindíveis a cooperação de 2 pessoas para realização destes, a colaboração de 2 pessoas para a documentação, e a assessoria dos 4 membros da equipe para a análise da qualidade e do desempenho.

3.2 Ferramentas de Teste

Os testes ocorreram de modo manual utilizando o editor de código Visual Studio Code para executar a aplicação, a fim de podermos testar a aplicação.

4 Cronograma

Data	Atividade de Teste	Resultados	Membro responsável
31/05/2023	<p>1. Verificar se o jogador tem apenas 3 vidas para a realização do jogo, juntamente com a mudança de cores dos corações que conforme as perdas devem mudar de vermelho para branco.</p> <p>Teste1: início do jogo Entrada1: $cont_vidas = 2$ Saída1: jogador possui 3 corações vermelhos</p> <p>Teste2: perdeu uma vida Entrada2: jogador colide com algum dos obstáculos (carro) Saída2: contador passa a ser $cont_vidas = 1$, e jogador possui 2 corações vermelhos</p> <p>Teste3: perdeu a segunda vida Entrada3: jogador colide com algum dos obstáculos (carro) Saída3: contador passa a ser $cont_vidas = 0$, e jogador possui 1 corações vermelhos</p> <p>Teste4: perdeu a terceira e última vida Entrada4: jogador colide com algum dos obstáculos (carro) Saída4: jogador possui três corações brancos, ou seja, perdeu todas as vidas.</p>	90%	Arthur Vinicius da Silva Bezerra
31/05/2023	<p>2. Analisar se o timer no canto superior direito está decrementado, iniciando do 120 até o 0, conforme o tempo passa, possuindo um tempo total de 2 minutos para o cumprimento do jogo.</p> <p>Teste1: tempo inicial Entrada1: $cont = 120$ Saída1: tempo total de 2 minutos</p> <p>Teste2: tempo após 60 segundos</p>	100%	Beatriz Gonçalves Fontes

	<p>Entrada2: cont = 60</p> <p>Saída2: jogador possui 60 segundos restantes</p> <p>Teste3: tempo após 2 minutos</p> <p>Entrada3: cont = 0</p> <p>Saída3: jogo encerrado</p>		
01/06/2023	<p>3. Certificar se a música da aplicação está mudando de acordo com a situação do jogador. Sendo a música de background do jogo em si a “música_de_fundo”, a de jogo finalizado com sucesso “win” e em caso de game over todos os sons devem ser encerrados.</p> <p>Teste1: música inicial</p> <p>Entrada1: jogo iniciado</p> <p>Saída1: som_fundo = pygame.mixer.music.load(os.path.join(diretorio_sons, "musica_fundo.mp3"))</p> <p>Teste2: música de colisão</p> <p>Entrada2: jogador colide com algum obstáculo</p> <p>Saída2: batida_carro = pygame.mixer.Sound(os.path.join(diretorio_sons, "batida.mp3"))</p> <p>Teste3: música de missão</p> <p>Entrada3: jogador realiza uma missão</p> <p>Saída3: casa_ok = pygame.mixer.Sound(os.path.join(diretorio_sons, "casa_ok.mp3"))</p> <p>Teste4: jogador venceu o jogo</p> <p>Entrada4: jogador realiza todas as entregas, vencendo o jogo.</p> <p>Saída4: win = pygame.mixer.Sound(os.path.join(diretorio_sons, "win.mp3"))</p>	100%	Maria Letícia da Silva Manguinho
01/06/2023	<p>4. Confirmar se o jogo foi encerrado, uma vez que o jogador perdeu todas as vidas, devendo ser exibida a imagem de game over.</p>	93%	Vitória Carla Costa de Lira

	<p>Teste1: Morte por objeto</p> <p>Entrada1: jogador com 1 coração colide com algum dos obstáculos (carro), passando a possuir três corações brancos e <code>cont_vidas = 0</code>.</p> <p>Saída1: mensagem de game over é exibida na tela.</p> <p>Teste1: Morte por tempo</p> <p>Entrada1: jogador utiliza todo o seu tempo para cumprir a atividade, tendo <code>cont = 0</code>.</p> <p>Saída1: mensagem de game over é exibida na tela.</p>		
02/06/2023	<p>5. Validar se o jogador conseguiu finalizar as entregas nas 3 casas, consequentemente, ganhando e encerrando o jogo, juntamente com a imagem final.</p> <p>Teste1: primeira entrega</p> <p>Entrada1: jogador chegou na casa 1</p> <p>Saída1: <code>cont_casa1 += 1</code> <code>cont_total += 1</code></p> <p>Teste2: segunda entrega</p> <p>Entrada2: jogador chegou na casa 2</p> <p>Saída2: <code>cont_casa2 += 1</code> <code>cont_total += 1</code></p> <p>Teste3: terceira entrega</p> <p>Entrada3: jogador chegou na casa 3</p> <p>Saída3: <code>cont_casa3 += 1</code> <code>cont_total += 1</code></p>	100%	Arthur Vinicius da Silva Bezerra
03/06/2023	<p>6. Conferir se o jogador não consegue ultrapassar determinados cenários do jogo, como, por exemplo, a grama, o cercado das casas, as extremidades da tela, etc.</p> <p>Teste1: cenário de baixo</p> <p>Entrada1: jogador tentar ultrapassar o cenário de baixo</p> <p>Saída1: <code>entregador.y -= 1</code> <code>entregador.x += 1</code></p> <p>Teste2: cenário de cima 1</p>	81%	Beatriz Gonçalves Fontes

	<p>Entrada2: jogador tentar ultrapassar o cenário de cima 1</p> <p>Saída2: entregador.y += 1 entregador.x += 1</p> <p>Teste3: cenário de cima 2</p> <p>Entrada3: jogador tentar ultrapassar o cenário de cima 2</p> <p>Saída3: entregador.y += 1 entregador.x -= 1</p> <p>Teste4: cenário de cima 3</p> <p>Entrada4: jogador tentar ultrapassar o cenário de cima 3</p> <p>Saída4: entregador.y += 1 entregador.x -= 1</p> <p>Teste5: cenário de cima 4</p> <p>Entrada5: jogador tentar ultrapassar o cenário de cima 4</p> <p>Saída5: entregador.y += 1 entregador.x -= 1</p>		
03/06/2023	<p>7. Constatar que os carros existentes como empecilhos no jogo, a cada nova execução aparecem em ordens diferentes e velocidade. Validar também o sumiço dos carros devendo ser de forma gradativa quando encostam no limite da tela.</p> <p>Teste1: velocidade dos carros</p> <p>Entrada1: jogo inicia</p> <p>Saída1: #direita pos_x_azul1 += velocidade_carros pos_x_vermelho1 += velocidade_carros pos_x_azul2 += velocidade_carros pos_x_verde1 += velocidade_carros #esquerda pos_x_verde2 -= velocidade_carros pos_x_vermelho2 -= velocidade_carros pos_x_vermelho3 -= velocidade_carros pos_x_azul3 -= velocidade_carros</p>	89%	Maria Letícia da Silva Manguinho

	<p>Teste2: ordem diferentes dos carros e sumiço gradativos dos carros</p> <p>Entrada2: jogo inicia</p> <p>Saída2: #direita</p> <p style="padding-left: 40px;"><i>pos_x_azul1 = randint(-1000, 0)</i></p> <p style="padding-left: 40px;"><i>pos_x_azul2 = randint(-1400, -1100)</i></p> <p style="padding-left: 40px;"><i>pos_x_vermelho1 = randint(-1000, 0)</i></p> <p style="padding-left: 40px;"><i>pos_x_verde1 = randint(-1400, -1100)</i></p> <p>#esquerda</p> <p style="padding-left: 40px;"><i>pos_x_verde2 = randint(800, 1000)</i></p> <p style="padding-left: 40px;"><i>pos_x_vermelho3 = randint(1500, 2000)</i></p> <p style="padding-left: 40px;"><i>pos_x_vermelho2 = randint(800, 1000)</i></p> <p style="padding-left: 40px;"><i>pos_x_azul3 = randint(1500, 2000)</i></p>		
03/06/2023	<p>8. Certificar que o jogador retorna ao ponto inicial do jogo a cada vida que perde.</p> <p>Teste1: início do jogo</p> <p>Entrada1: entregador.x = 595</p> <p style="padding-left: 40px;"><i>entregador.y = 550</i></p> <p>Saída1: jogador inicia no ponto inicial, ou seja, entregador.x = 595 e entregador.y = 550.</p> <p>Teste2: perdeu uma vida</p> <p>Entrada2: jogador colide com algum dos obstáculos (carro)</p> <p>Saída2: jogador volta ao ponto inicial, ou seja, entregador.x = 595 e entregador.y = 550.</p> <p>Teste3: perdeu a segunda vida</p> <p>Entrada3: jogador colide com algum dos obstáculos (carro)</p> <p>Saída3: jogador volta ao ponto inicial, ou seja, entregador.x = 595 e entregador.y = 550.</p>	100%	Vitória Carla Costa de Lira

06/06/2023	<p>9. Verificar se o jogador muda a sua posição e a direção de sua mobilidade, de acordo com as teclas do teclado responsáveis pelo comando.</p> <p>Teste1: botão para cima é pressionado</p> <p>Entrada1: <code>pygame.key.getPressed</code> obtém o valor de <code>pygame.K_UP</code> e o eixo y do entregador é maior que 0</p> <p>Saída1: posição do sprite assume o seguinte vetor: [0, 88, 44, 132] e o entregador terá o valor do eixo y decrementado uma unidade.</p> <p>Teste2: botão para baixo é pressionado</p> <p>Entrada2: <code>pygame.key.getPressed</code> obtém o valor de <code>pygame.K_DOWN</code> e o eixo y do entregador é menor que o resultado da subtração entre a altura da tela e a altura do entregador.</p> <p>Saída2: posição do sprite assume o seguinte vetor: [0, 0, 44, 44] e o entregador terá o valor do eixo y incrementado uma unidade.</p> <p>Teste3: botão para direita é pressionado</p> <p>Entrada3: <code>pygame.key.getPressed</code> obtém o valor de <code>pygame.K_RIGHT</code> e o eixo x do entregador é menor que o resultado da subtração entre a largura da tela e a largura do entregador.</p> <p>Saída3: posição do sprite assume o seguinte vetor: [0, 88, 44, 132] e o entregador terá o valor do eixo x incrementado uma unidade.</p> <p>Teste4: botão para esquerda é pressionado</p> <p>Entrada4: <code>pygame.key.getPressed</code> obtém o valor de <code>pygame.K_LEFT</code> e o eixo x do entregador é maior que 0</p> <p>Saída4: posição do sprite assume o seguinte vetor: [0, 44, 44, 88] e o entregador terá o valor do eixo x decrementado uma unidade.</p>	100%	<p>Arthur Vinicius da Silva Bezerra; Beatriz Gonçalves Fontes; Maria Letícia da Silva Manguinho; Vitória Carla Costa de Lira.</p>
06/06/2023	<p>10. Constatar que o jogo deve ser encerrado ao ser clicado no “X” no canto superior esquerdo da tela, efetuando assim a ação de QUIT.</p> <p>Teste1: o botão quit foi pressionado</p> <p>Entrada1: <code>pygame.event.get().type = pygame.QUIT</code></p>	100%	<p>Arthur Vinicius da Silva Bezerra; Beatriz Gonçalves Fontes;</p>

	<p><i>Saída1: variável de controle do loop while é setada como false [jogo = false], é chamada a função pygame.quit() e o programa é finalizado</i></p>		<p><i>Maria Letícia da Silva Manguinho; Vitória Carla Costa de Lira.</i></p>
--	--	--	--

5 Controle Estatístico de Qualidade

Nesta sessão extraordinária encontra-se o link de visualização da planilha presente no Google Sheets, contendo os dados utilizados para gerar os seguintes gráficos: gráfico de qualidade e gráfico das amplitudes.

5.1 Link Planilha

https://docs.google.com/spreadsheets/d/1Sql5GiPLngnq2TZ5bqxpIUwtPsryGA2tM3MoHx_P1cg/edit?usp=sharing

5.2 Gráficos

Gráfico das Amplitudes (R-Barra)

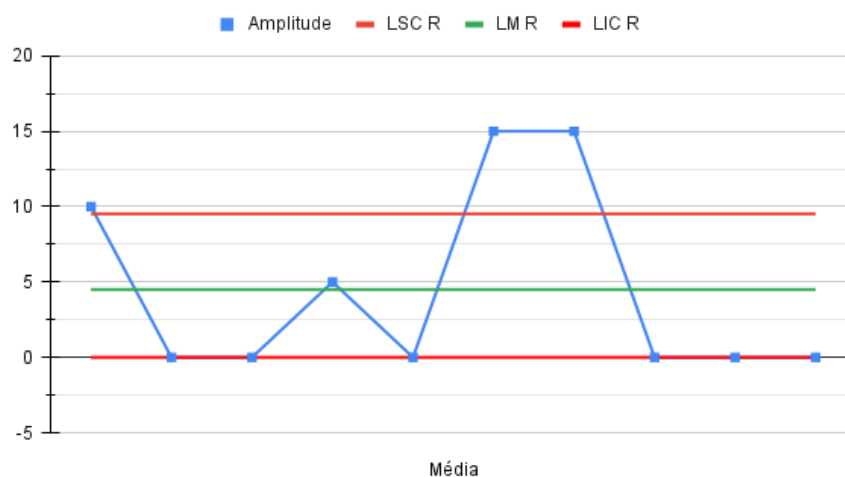


Gráfico de Qualidade (X-Médias)

