

**Gerenciador de salários**

**Plano de Teste**

**Versão 01.01**

Link Projeto GitHub: [https://github.com/jhunior244/trabalho\\_testes](https://github.com/jhunior244/trabalho_testes)

### **Histórico de revisão**

Data	Versão	Descrição	Autor
03/11/2019	01.00	Teste de unidade e integração	Junior
10/11/2019	01.00	Teste de unidade e integração	Israel
12/11/2019	01.00	Teste de unidade e integração	Samara
18/11/2019	01.00	Teste de unidade e integração	Lucas
27/11/2019	01.01	Testes de sistema	Junior

## Sumário

<b>1. Introdução .....</b>	<b>4</b>
<b>2. Escopo .....</b>	<b>4</b>
<b>3. Referencias .....</b>	<b>4</b>
<b>4. Requisitos do teste .....</b>	<b>4</b>
<b>4.1 Testes de unidade .....</b>	<b>4</b>
<b>4.2 Testes de integração .....</b>	<b>4</b>
<b>5. Resultados dos testes .....</b>	<b>5</b>

## 1. Introdução

Este documento de Plano de Teste tem o objetivo de documentar as informações necessárias para planejar e controlar os testes de validação do projeto Gerenciador de salários v.1.0. O documento descreve o plano geral de testes referente aos cadastros básicos, inserção de dados e tratamento das informações inseridas de forma a direcionar os esforços de teste e os Casos de Teste a serem executados para validar o produto.

## 2. Escopo

Este documento descreve o Plano de Testes a ser usado pelo projeto Gerenciador de Salários para avaliar a qualidade funcional, confiabilidade e performance. Os testes que serão cobertos por este documento serão: cadastro de jogadores, cadastro de salários, e cálculo de salários.

## 3. Referencias

- <https://angular.io/docs>
- <https://spring.io/docs>
- <https://selenium.dev/documentation/en/>
- <http://rest-assured.io/>

## 4. Requisitos do teste

A lista a seguir identifica os itens (casos de uso, requisitos funcionais) que foram identificados como alvos do teste. Esta lista apresenta o que será testado e os resultados dos testes. Detalhes sobre alguns testes serão apresentados em outro documento mostrando os objetos alvo e seus respectivos resultados.

### 4.1 Testes de unidade

- Verificar o tratamento de ids nulas ao tentar obter uma entidade do banco.
- Verificar o tratamento de datas de pagamento invalidas.
- Verificar método de cálculo de salário.

### 4.2 Testes de integração

- Verificar o funcionamento dos métodos de criação, atualização, obtenção, e listagem de jogador.
- Verificar a listagem de ação por jogador, mês e ano.
- Verificar o método de criação de novo salário.
- Verificar a criação e obtenção de ação.

### 4.3 Testes de sistema

- Testar a criação de um jogador.
- Testar a busca do jogador criado.

## 5. Resultados dos testes

### Classe jogador

Método	Caso de teste funcional	Resultado	Tester Responsável
<b>Teste de unidade</b>			
obtem	(TS 1, Y)	ok	Junior
<b>Teste de integração</b>			
obtem	(TS 2, Y)	ok	Junior
cria	(TS 3, Y)	ok	Junior
lista	(TS 4, Y)	ok	Junior
listaTodos	(TS 5, Y)	ok	Junior

### Classe Pagamento

Método	Caso de teste funcional	Resultado	Tester Responsável
<b>Teste de unidade</b>			
calculaPagamento	(TS 1, Y)	ok	Israel
<b>Teste de integração</b>			
obtem	(TS 2, Y)	ok	Israel
cria	(TS 3, Y)	ok	Samara
lista	(TS 4, Y)	ok	Samara

### Classe Acao

Método	Caso de teste funcional	Resultado	Tester Responsável
<b>Teste de integração</b>			
obtem	(TS 1, Y)	ok	Lucas
cria	(TS 2, Y)	ok	Lucas

### Teste de integração

Caso de uso	Resultado	Tester responsável
Criar um jogador	ok	Junior
Buscar um jogador	ok	Junior

**Grupo**

Junior Moreira

Israel Dias

Samara Guimarães

Lucas Duarte