

Controle de Pontos da Jornada de Trabalho

Elton Francisco de Oliveira
Gabriel Simon
Lorran P. dos Santos
Marcelo Schaefer
Willian Kenji Nishizawa

Repositório:

<https://github.com/KenjiNishi/senior-proway-controle-ponto>

Trello:

<https://trello.com/b/eOqMwjQf/seniorgodev-controle-de-ponto>





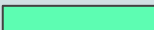


Controle de Pontos - Responsabilidades

- Permitir que funcionários registrem os pontos de entrada e saída da sua jornada de trabalho;
- Armazenar todos os pontos em uma base de dados;
- Realizar os cálculos de horas trabalhadas;
- Realizar os cálculos de horas no banco de horas;
- Realizar os cálculos de horas extras;
- Permitir que o funcionário justifique pontos registrados incorretamente, faltantes ou em outras situações imprevistas;
- Permitir que funcionários acessem a situação dos seus horários individuais;
- Permitir que gerentes tenham acesso a uma visão geral da sua equipe;
- Notificar funcionários quando existir alguma irregularidade no registro de pontos.


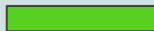




Controle de Pontos - Componentes/Módulos

Módulos:

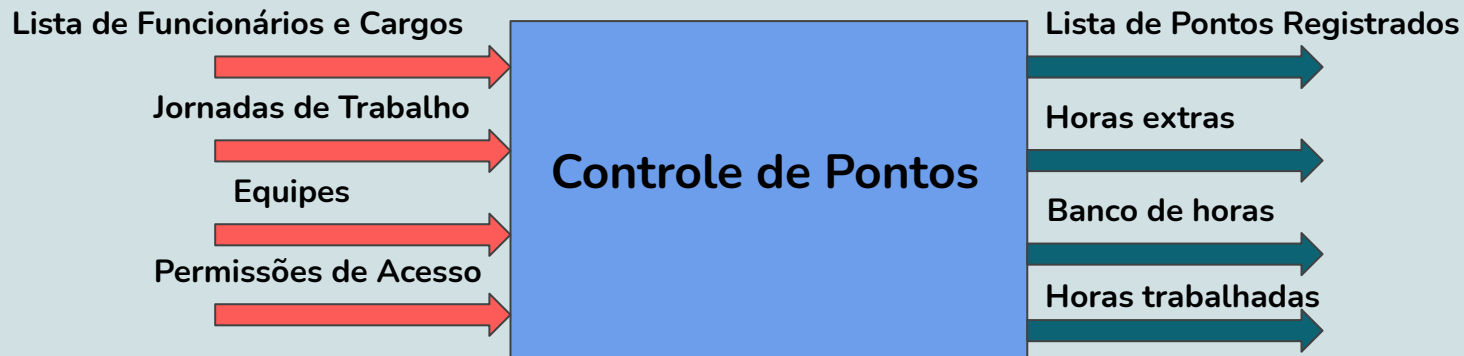
- Registro de Pontos 
- Justificativas 
- Gerência 
- Notificações 
- Ferramentas 

Etiquetas:

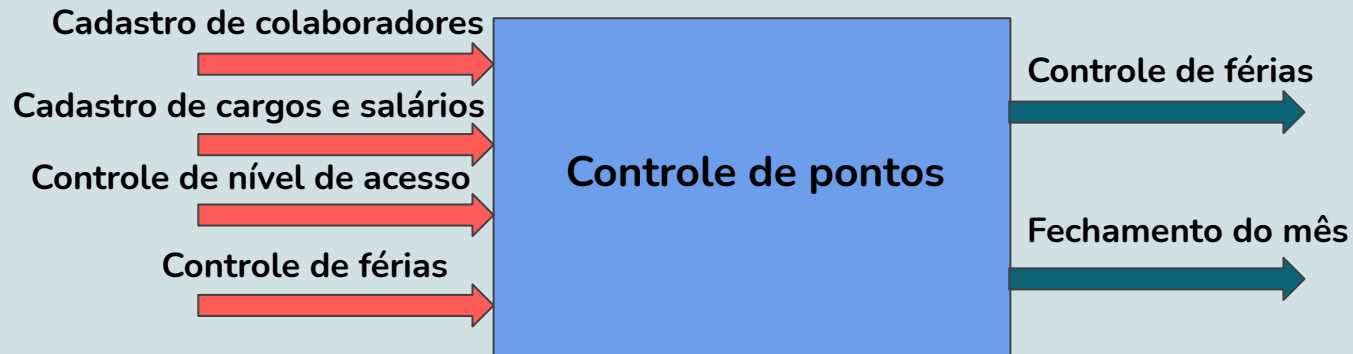
- Regras de negócio 
- Funcionalidades 
- Bug/Problema 
- Componente 



Interface com outros sistemas - Fluxo de Dados



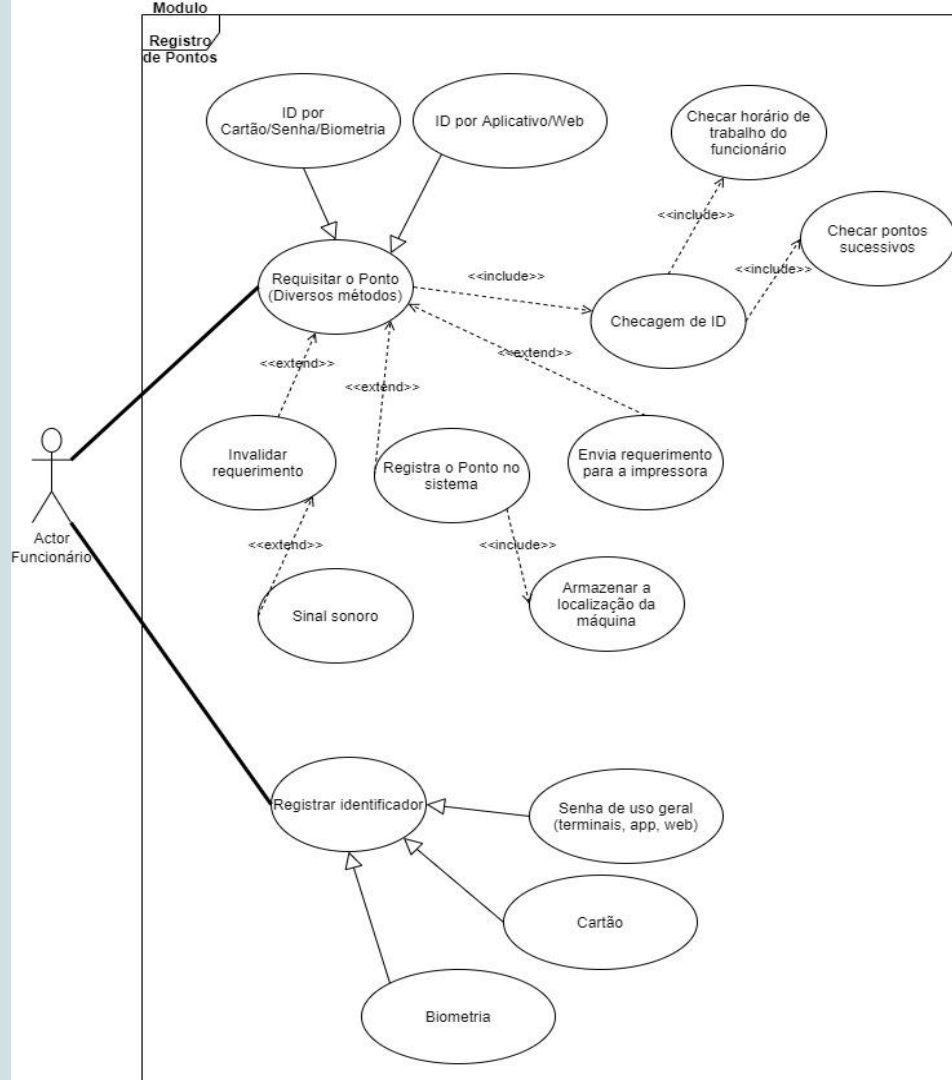
Interface com outros setores

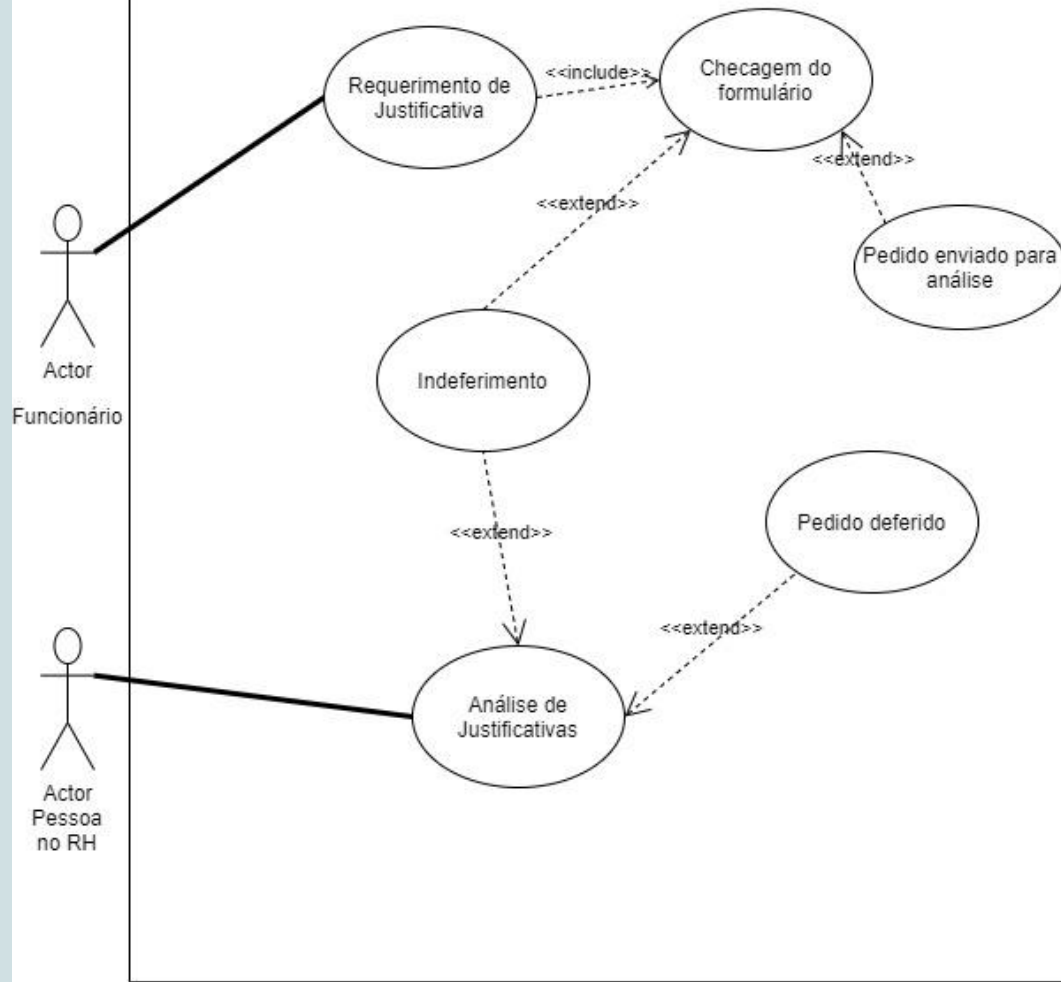




Diagramas UML

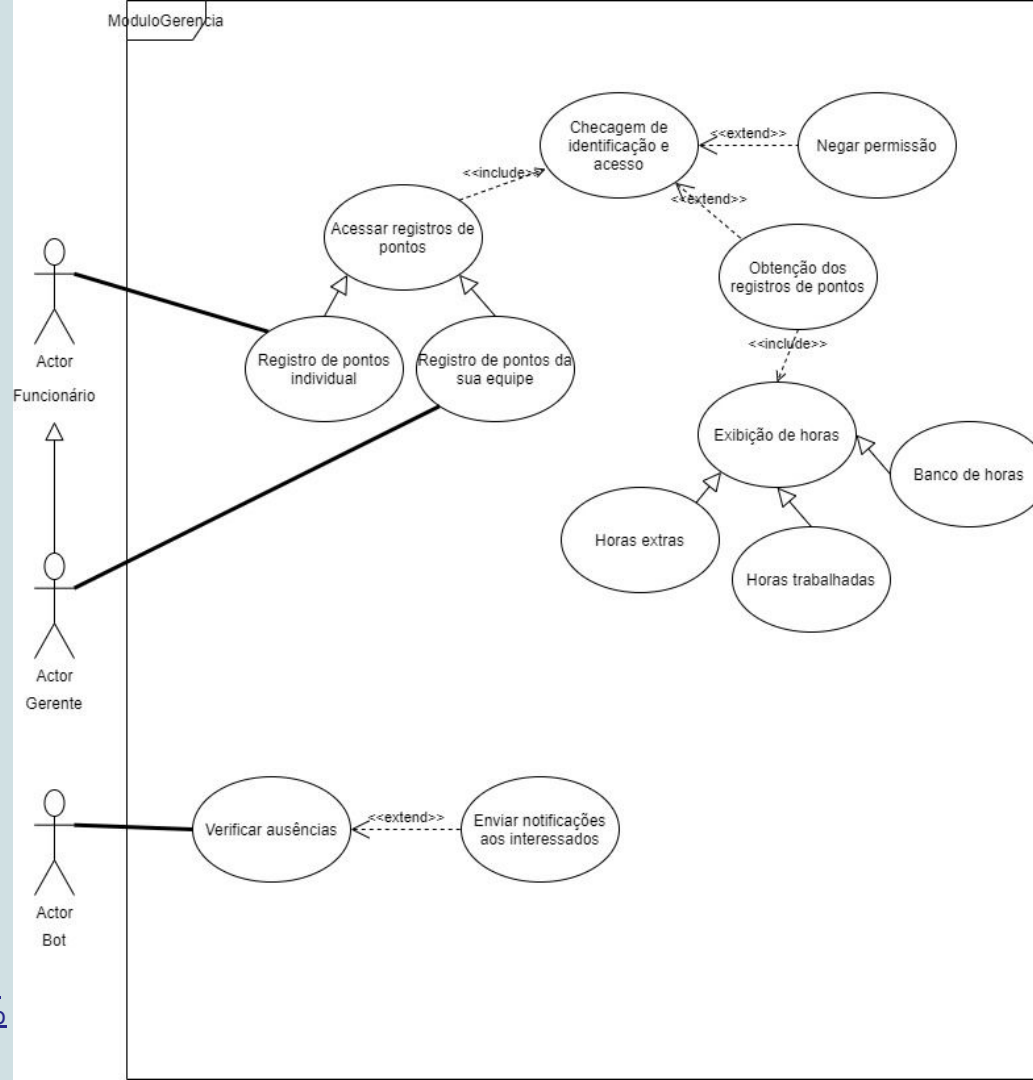
Registro de Pontos



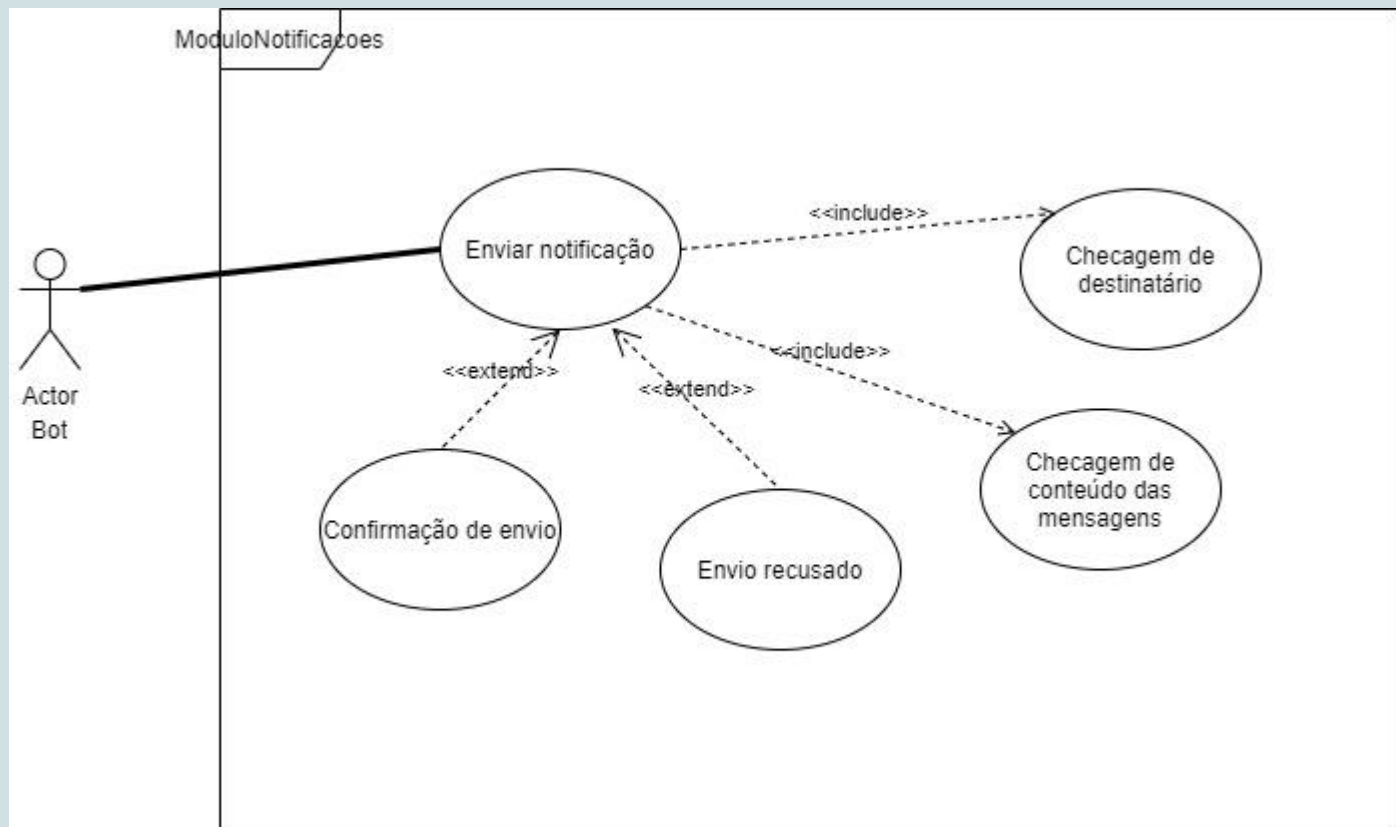


Justificativas de Pontos

Gerência de Pontos



Notificador do sistema





Sprint Review - O que foi feito

- Cálculo de horas trabalhadas no dia
- Retorno da impressora
- Cálculo de minutos trabalhados no dia
- Localização do ponto
- Verificação se a senha é existente
- Envio de notificação
- Checar destinatário
- Verificar a Validade da mensagem
- Validação dia, mês e ano (ainda na main)
- Inclusão de justificativa (ainda na main)
- Entrada na empresa
- Saída para almoço
- Volta do almoço
- Saída da empresa



Sprint Review - O que não foi feito

- Padronização dos formatos de Data para os diferentes métodos;
- Discernimento entre funções do Módulo de Registro de Pontos e Módulo de Gerenciamento (Separação das classes);
- Alguns testes;



Sprint Review - O que foi adicionado

Dividir métodos entre:

- Métodos exclusivos das classes;
- Métodos ferramentas



Sprint Review - Bloqueios e Remoções

- Excesso de métodos semelhantes;
- Falta de familiaridade com as ferramentas;



Sprint Review - Demonstrações

```
15      @Test
16      public void testeHoraEntrada() {
17          String horaEntrada = RegistroPonto.horaEntrada();
18          assertEquals("2021-04-13T08:00", horaEntrada);
19      }
```



Sprint Review - Demonstrações

```
43  /**
44   * Registra hora de entrada
45   *
46   * Este método insere uma hora de entrada logo que o usuário insere o seu cartão
47   * de identificação no aparelho identificador
48   *
49   * @return valor da hora em que se bate o ponto
50   */
51  static String horaEntrada() {
52      entrada = LocalDateTime.of(2021, 4, 13, 8, 00, 00); // Data/hora estáticos entrada
53      System.out.println("Registro Entrada: " + entrada);
54      String horarioEntFormatado = entrada.toString();
55      return horarioEntFormatado;
56  }
```




Sprint Review - Demonstrações

```
39  @Test
40  public void testeNaoBateuHoraSaida() {
41
42      String entrada = RegistroPonto.horaEntrada();
43      String saidaAlmoco = null;
44      String retornoAlmoco = null;
45      String saida = null;
46
47      String saidaNaoRegistrada = RegistroPonto.validacaoHorasNotificacao(entrada, saidaAlmoco, retornoAlmoco, saida);
48      assertEquals("Você não bateu o seu registro de saída", saidaNaoRegistrada);
49  }
```



Sprint Review - Demonstrações

```
134  /**
135  *
136  * Método responsável por validar se o horário de saída foi registrado, se o ponto foi batido consecutivamente (margem de 5 min),
137  * se bateu o ponto final antes das 8h trabalhadas ou se ele cumpriu todo o horário de trabalho (tendo ou não horas extras)
138  *
139  * @param horaEntrada String informa o horário de entrada do usuário
140  * @param saidaAlmoco String informa o horário de saída do usuário para almoço
141  * @param retornoAlmoco String informa o horário de retorno ao local de trabalho do usuário após o almoço
142  * @param horaSaida String informa o horário de saída do usuário da empresa após o fim da jornada de trabalho
143  * @return
144  */
145  static String validacaoHorasNotificacao(String horaEntrada, String saidaAlmoco, String retornoAlmoco,
146      String horaSaida) {
147
148      if (saidaAlmoco == null || horaSaida == null) {
149          String faltandoSaida = "Você não bateu o seu registro de saída";
150          return faltandoSaida;
151      }
```



Sprint Retrospective - Lições

Continuar:

- Comunicação com a equipe;
- Trabalhar em conjunto (pair programming);
- Disponibilidade em ajudar;



Sprint Retrospective - Lições

Começar a fazer:

- Separar 1 hora semanal de trabalho da equipe para focar na familiarização de ferramentas e tecnologias;
- Refinar as Stories/Requerimentos para que as tarefas sejam mais objetivas/fáceis de entender;

Fazer mais:

- Aumentar o nível de interação em plataformas como o Trello e Git para que seja mais fácil de acompanhar o desenvolvimento das outras partes do sistema com os objetivos de evitar: retrabalho, incompatibilidades na hora de integrar classes, etc...



Obrigado!