

Métodos Formais

Sistema de ponto eletrônico

Docente: Alysson Filgueira Milanez

Discente: Silvio Martins Santos

1. Introdução

O projeto visa desenvolver uma aplicação para registrar o ponto de entrada e saída dos funcionários nos locais de trabalho. O objetivo principal é proporcionar uma solução de baixo custo adequada para pequenas empresas com até 20 funcionários. A aplicação tem a função de substituir o livro de ponto tradicional, ainda utilizado por algumas empresas. O único requisito essencial para a empresa que deseja adotar o sistema é possuir um computador disponível para que os funcionários registrem seus pontos diariamente.

Além de registrar a entrada e saída dos funcionários, o sistema também deve gerar relatórios para a contabilidade, facilitando o cálculo do pagamento através da contagem de horas trabalhadas. Adicionalmente, a empresa pode optar por um relatório que oferece uma melhor visibilidade de todos os registros, auxiliando nas análises que anteriormente eram realizadas manualmente no livro de pontos.

Para o desenvolvimento deste projeto, será utilizada a linguagem Java em conjunto com a JML (*Java Modeling Language*), que é uma linguagem de especificação para programas. Ela incorpora a lógica de Hoare, que inclui pré e pós-condições, além de invariantes, seguindo o paradigma da programação por contrato para garantir uma estrutura sólida e confiável (Wikipedia, 2024). Ou seja, o *OpenJML* realiza a verificação dedutiva de programas para assegurar que a implementação de um programa em Java atenda à sua especificação em JML (OpenJML, 2024). Para este projeto, foi utilizada a versão do [OpenJML para Ubuntu 22.04](#).

Além disso, o PostgreSQL será utilizado como sistema de gerenciamento de banco de dados (SGBD), assegurando a persistência e organização dos dados. O planejamento também inclui a adoção do framework Spring Boot, que oferece uma estrutura simplificada e robusta para a construção de aplicativos Java, facilitando a configuração e integração de diversos componentes.

Padrões de projeto como o DAO (Data Access Object) e o Singleton são utilizados, fornecendo soluções reutilizáveis para problemas comuns no desenvolvimento de software. Esses padrões ajudam a abstrair problemas complexos e promovem a reutilização de código, facilitando a manutenção do software, aumentando a compreensão da estrutura e a qualidade do código, e padronizando soluções.

2. Cenário de uso

2.1 Registro de data e horário de um funcionário.

Descrição: Este caso de uso descreve como o usuário realiza seu ponto de entrada no expediente.
Ator Primário: Funcionário da empresa
Condições prévias: O usuário deve ter seu ID único salvo previamente no banco de dados. O usuário deve ter sua senha salva previamente no banco de dados.
Fluxo principal: <ol style="list-style-type: none">1. O funcionário visualiza o programa no terminal.2. O sistema exibe uma mensagem solicitando o ID do usuário.3. O funcionário digita seu ID corretamente.4. O sistema exibe uma mensagem solicitando a senha do usuário.5. O funcionário preenche sua senha corretamente.6. O sistema exibe uma pergunta se o funcionário está entrando ou saindo do ambiente.7. O funcionário digita a opção referente a opção solicitada.8. O sistema pergunta se o usuário deseja adicionar uma justificativa.9. O funcionário responde a opção referente a “não”.10. O sistema guarda o registro no banco de dados referente às opções escolhidas pelo usuário.
Fluxo excepcional: <ol style="list-style-type: none">1. O funcionário visualiza o programa no terminal.2. O sistema exibe uma mensagem solicitando o ID do usuário.3. O funcionário digita seu ID corretamente.4. O sistema exibe uma mensagem solicitando a senha do usuário.5. O funcionário preenche sua senha de forma errada.6. O sistema pergunta se o funcionário deseja sair ou continuar com uma nova inserção dos seus dados.
pós condição: A data e horário de entrada do funcionário que realizou a ação é registrada

2.2 Gerar relatórios de registros

Descrição: Este caso de uso descreve como um usuário com cargo de Gerente pode gerar um relatório com os registros de movimentações de entrada e saída de funcionários.

Ator Primário: Gerente da empresa

Condições prévias:

1. O usuário deve ter seu ID único salvo previamente no banco de dados.
2. O usuário deve ter sua senha salva previamente no banco de dados.
3. O usuário deve ter seu cargo como gerente.
4. No próprio código deve haver um *path* onde o arquivo será gerado.

Fluxo principal:

1. O funcionário visualiza o programa no terminal.
2. O sistema exibe uma mensagem solicitando o ID do usuário.
3. O funcionário digita seu ID corretamente.
4. O sistema exibe uma mensagem solicitando a senha do usuário.
5. O funcionário preenche sua senha corretamente.
6. O sistema reconhece o usuário com a função de Gerente.
7. O sistema pergunta se ele deseja imprimir um relatório.
8. O funcionário responde a opção correspondente a “sim”.
9. O sistema gera um arquivo e armazena em um *path* previamente designado.

Fluxo excepcional:

1. O cliente visualiza o programa no terminal.
2. O sistema exibe uma tela onde existem 4 botões: entrada, saída, relatórios e alterar registro.
3. O cliente seleciona a opção “entrada” .
4. O sistema solicita as informações de ID e senha.
5. O usuário preenche os campos de forma incorreta.
6. O sistema informa que o usuário errou ao digitar o ID e ao digitar a senha.

pós condição: O horário de entrada do funcionário que realizou a ação é registrada

3. Esboço de especificação formal dos Requisitos

[RF001] Registrar a Entrada do Usuário

- Pré-condição: O usuário está pré-registrado.
- Entrada: ID do usuário (ID_usuario), senha (senha_usuario).
- Pós-condição: Entrada registrada com data e hora atual.

[RF002] Registrar a Saída do Usuário

- Pré-condição: O usuário está pré-registrado.
- Entrada: ID do usuário (ID_usuario), senha (senha_usuario).
- Pós-condição: Saída registrada com data e hora atual.

[RF003] Registro de Horário

- Pré-condição: Uma entrada ou saída foi realizada.
- Pós-condição: Horários e datas de registros são armazenados automaticamente.

[RF004] Justificativa do Funcionário

- Pré-condição: O usuário está fazendo um registro (entrada ou saída).
- Entrada: Justificativa opcional (justificativa).
- Pós-condição: Justificativa registrada com o respectivo registro de entrada ou saída.

[RF005] Relatório de Pontos

- Pré-condição: O usuário que solicita é um gerente com credenciais válidas.
- Entrada: Credenciais do usuário (ID_usuario, senha_usuario).
- Pós-condição: Relatório gerado contendo nome, tipo de registro e presença de justificativa.
- Invariantes: Apenas usuários gerentes podem gerar os relatórios.

4. Referências

WIKIPEDIA. Java Modeling Language. Disponível em: https://en.wikipedia.org/wiki/Java_Modeling_Language. Acesso em: 24 abr. 2024.

OPENJML. OpenJML: Java Modeling Language Tools. Disponível em: <https://www.openjml.org/>. Acesso em: 24 abr. 2024.