

Relatório Milestone 1 - P2

WePayU

Aluno: José Matheus Santana Alves (Zé)

Descrição Geral da Arquitetura

O sistema proposto é um sistema de folha de pagamento de uma empresa. O cálculo da folha de pagamento apresenta diferenças de acordo com o tipo de empregado a receber do seu trabalho e algumas características atreladas ao empregado. Um empregado pode ser horista, comissionado ou assalariado, e portanto a arquitetura do sistema é desenhada em torno dessa dinâmica. As principais características (atributos) necessários para os cálculos serão guardados dentro da classe Empregado, que serve como classe mãe para as classes EmpregadoHorista, EmpregadoComissionado e EmpregadoAssalariado. A partir dessa herança é possível designar métodos e atributos diferentes, para diferentes tipos de empregados, possibilitando que o cálculo da folha de pagamento considere as individualidades da situação atual de cada empregado.

Para que seja possível criar, manipular e analisar os empregados, é implementado uma business logic que possibilita ações que permitem as operações do CRUD (Create, Read, Update & Delete), além de operações específicas para cada tipo de empregado para que sejam registrados no sistema as informações de seu trabalho, como horas trabalhadas, que serão levadas em consideração durante o cálculo da folha de pagamento.

A business logic do sistema se sustenta em diversas classes utilitárias criadas para dar suporte às operações necessárias para funcionamento do sistema, como armazenamento e manipulação dos objetos das classes Empregado, persistência dos dados, tratamento de entrada e tratamento de exceções. Esses artefatos controlam o fluxo do sistema, mantêm seu estado e garantem a entrega correta das informações.

Principais componentes e suas interações

Como supracitado, o sistema funciona em torno da classe de Empregado, que guarda as principais informações para entrega da folha de pagamento e garante, com a herança, o tratamento individualizado para tipos de empregados.

Empregado

Guarda os atributos:

- ID;
- Nome;
- Endereço;
- Tipo de Empregado;

- Salário;
- Se é sindicalizado ou não;
- Informações sindicais; e
- Método de pagamento utilizado.

E possui os métodos responsáveis para acesso e registro desses atributos (getters e setters). Essa classe serve como classe mãe para as classes `EmpregadoHorista`, `EmpregadoComissionado` e `EmpregadoAssalariado`.

EmpregadoHorista

Guarda os mesmos atributos de `Empregado` e guarda uma lista de cartões de pontos, representados pela classe `CartaoDePonto`, que mantém as informações da data de lançamento do cartão e a quantidade de horas trabalhadas. Possui getters e setters, além de método para adicionar novo cartão de ponto à lista e métodos para contabilizar a quantidade de horas trabalhadas dentro de determinado intervalo de tempo.

EmpregadoComissionado

Guarda os atributos de `Empregado` e guarda uma taxa de comissão, além de uma lista de vendas realizadas, representadas pela classe `ResultadoDeVenda`, que armazena a data e o valor de uma venda. Além de getters e setters, possui métodos para adicionar uma nova venda e para contabilizar o valor das vendas realizadas dentro de um determinado período.

EmpregadoAssalariado

Não guarda nenhum atributo além dos guardados em `Empregado`. Existe para que a diferenciação entre empregados seja possível de maneira mais dinâmica.

MetodoPagamento

Classe utilizada para guardar as informações relacionadas aos métodos de pagamento utilizados em `Empregado`. É classe mãe de `Banco`, `Correios` e `EmMaos`, que representam os métodos de pagamento diferentes possíveis.

MembroSindicato

Guarda as informações referentes ao sindicato dos empregados. Guarda:

- Identificador (ID) de membro;
- Taxa sindical; e
- Lista de taxas de serviço;
 - Taxa de serviço representada pela classe `TaxaServico` que guarda data e valor do serviço.

Possui getters e setters, métodos para adicionar uma nova taxa de serviço e para contabilizar o valor das taxas dentro de determinado período.

BusinessLogic

Classe que realiza toda a lógica de negócio do sistema. Ela fornece identificadores aos empregados, se utilizada das classes Persistence e Xstream, utilizadas para persistência dos dados.

Possui os métodos para:

- zerar o sistema;
- encerrar o sistema;
- criar novo empregado;
- consultar atributo de empregado;
- alterar atributo de empregado;
- consultar empregado pelo nome;
- remover empregado do sistema;
- lançar novo cartão de pontos;
- lançar nova venda;
- lançar nova taxa de serviço;
- consultar vendas;
- consultar horas dos cartões de pontos;
- consultar taxas de serviço;
- calcular o total da folha de pagamento.

TratamentoEntrada

Classe utilitária que auxilia no tratamento de entradas no sistema e no tratamento de exceções durante o funcionamento do programa.

ListaEmpregados

Classe utilitária que auxilia na manipulação dos empregados armazenados no sistema.