



UNIVERSIDADE FEDERAL DE ALAGOAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMAÇÃO 2



WALTER SOARES COSTA NETO

Sistema de Folha de Pagamento - WePayU

Maceió, AL. Outubro de 2025

WALTER SOARES COSTA NETO

Sistema de Folha de Pagamento - WePayU

Relatório requerido como avaliação parcial
da disciplina Programação 2 pelo Prof. Dr.
Mário Hozano Lucas de Souza.

Maceió, AL. Outubro de 2025

SUMÁRIO

1. Introdução	3
2. Estrutura de diretórios	3
3. Padrões de Projeto Utilizados	4
4. Principais Benefícios da Arquitetura	5
5. Conclusão	5

1. Introdução

Este relatório descreve o desenvolvimento de uma aplicação de Folha de Pagamento utilizando a linguagem de programação Java, estruturado de forma modular e seguindo princípios de baixo acoplamento, alta coesão e extensibilidade. O projeto buscou integrar os principais conceitos da Programação Orientada a Objetos (POO), sendo eles: princípio da responsabilidade única, encapsulamento, herança e polimorfismo. Além disso, os seguintes design patterns foram adotados: Command, Factory, Facade, Service Layer, e Domain Model. Isso foi feito com o objetivo de organizar o código em camadas bem definidas e facilitar a manutenção e evolução do sistema.

O objetivo central do sistema é gerenciar empregados e suas remunerações, permitindo o registro de vendas, cartões de ponto, taxas de serviço e a execução de folhas de pagamento.

2. Estrutura de diretórios

src/br/ufal/ic/p2/wepayu/

|

— cmds/	# Implementações de comandos do sistema
— Exception/	# Exceções customizadas
— factories/	# Fábricas de criação de objetos (Empregado, Sindicato)
— models/	# Classes de domínio (Entidades e Objetos de Negócio)
— services/	# Serviços que contêm a lógica de negócio
— utils/	# Utilitários e classes auxiliares
— Facade.java	# Interface principal do sistema (ponto de entrada)

3. Padrões de Projeto Utilizados

1. **Command:** Localizado na pasta ***cmds/***. Cada operação do sistema (ex: *CriarEmpregadoCmd*, *RodaFolhaCmd*) é encapsulada em um objeto comando, que implementa uma interface comum (*Cmd*). Isso permite execução dinâmica e desacoplamento entre a interface e a lógica de negócio.
2. **Factory:** Localizado na pasta ***factories/***. Criação centralizada de objetos complexos, como *EmpregadoFactory* e *MembroSindicatoFactory*, evitando dependência direta de construtores e permitindo instanciar subclasses dinamicamente.
3. **Facade:** Oferece uma interface única e simplificada para o sistema. A *Facade* coordena comandos, serviços e modelos, escondendo a complexidade interna das camadas inferiores.
4. **Service Layer:** Localizado em ***services/***. Contém a lógica de negócio (ex: *FolhaPagamentoService*, *EmpregadoService*). Cada serviço encapsula operações específicas do domínio, garantindo separação entre regra de negócio e controle.
5. **Domain model:** Localizado em ***models/***. Representa o núcleo da aplicação, modelando entidades como *Empregado*, *Comissionado*, *Horista*, *Assalariado*, *CartaoPonto*, *Venda*, *MembroSindicato*, entre outras.

4. Principais Benefícios da Arquitetura

- **Extensibilidade:** Novos comandos e tipos de empregados podem ser adicionados sem modificar a Facade ou outros módulos centrais.
- **Baixo acoplamento:** Camadas comunicam-se por interfaces bem definidas.
- **Alta coesão:** Cada pacote tem responsabilidade clara.
- **Testabilidade:** Serviços e comandos podem ser testados isoladamente.
- **Organização limpa:** Segue boas práticas de DDD (Domain-Driven Design) e GoF patterns.

5. Conclusão

O projeto WePayU é uma implementação sólida e bem estruturada de um sistema de folha de pagamento, que combina padrões de projeto clássicos (Command, Factory, Facade) com uma arquitetura em camadas.

Essa combinação garante clareza, manutenção facilitada e escalabilidade, tornando o sistema adequado tanto para fins didáticos quanto como base para aplicações empresariais mais amplas.