UNIVERSIDADE FEDERAL DE ALAGOAS

ALLAN DOUGLAS SILVA DOS SANTOS

RELATÓRIO DE PROJETO

MACEIÓ-ALAGOAS

2024

Relatório de projeto apresentado à disciplina de programação 2, referente ao período de 25/01/24 a 06/02/24.

Maceió - AL

06/02/24

RESUMO

O presente relatório tem como objetivo apresentar o projeto de desenvolvimento de um sistema de folha de pagamento denominado WePayU. Este projeto é parte integrante do currículo da disciplina de programação, com foco na linguagem Java, e visa proporcionar aos estudantes uma experiência prática na construção de sistemas de informação de qualidade.

SUMÁRIO

1- INTRODUÇÃO

O sistema de folha de pagamento WePayU foi concebido para administrar os pagamentos aos empregados de uma empresa de forma eficiente e precisa. Para alcançar esse objetivo, o sistema foi projetado em etapas incrementais, com funcionalidades definidas por meio de User Stories, que descrevem as interações e requisitos funcionais do sistema.

O propósito principal do sistema é garantir que os empregados recebam seus salários corretamente e no momento adequado, levando em consideração suas modalidades de pagamento, como horistas, assalariados e comissionados. Além disso, o sistema deve ser capaz de gerenciar as deduções, impostos e taxas sindicais aplicáveis a cada empregado.

A metodologia de desenvolvimento adotada baseia-se em testes de aceitação automáticos, que validam a lógica de negócios do sistema, garantindo que ele atenda aos requisitos do cliente. Cada milestone representa uma etapa do desenvolvimento, na qual novas funcionalidades são adicionadas ao sistema de acordo com as User Stories priorizadas.

2-CONFIGURAÇÃO

Foram adicionadas as seguintes dependências:

- MySQL, Hibernate, Lombok, Hibernate-Validator

```
<dependencies>
        <dependency>
            <groupId>com.mysql</groupId>
            <artifactId>mysql-connector-j</artifactId>
            <version>8.2.0</version>
        </dependency>
        <dependency>
            <groupId>org.hibernate.orm</groupId>
            <artifactId>hibernate-core</artifactId>
            <version>6.4.2.Final</version>
        </dependency>
        <dependency>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
            <version>1.18.30</version>
            <scope>provided</scope>
        </dependency>
        <dependency>
            <groupId>org.hibernate.validator</groupId>
            <artifactId>hibernate-validator</artifactId>
            <version>8.0.1.Final</version>
        </dependency>
    </dependencies>
```

As demais configurações de usuário e senha estão salvos em “persistence.xml” e as consultas SQL estão em “consulta.xml”

3- DESENVOLVIMENTO

O sistema segue uma arquitetura baseada no padrão Model-View-Controller (MVC), que é uma abordagem comumente utilizada para desenvolvimento de software que separa as responsabilidades de apresentação, lógica de negócio e manipulação de dados. Essa separação permite uma melhor organização do código, facilita a manutenção e promove a reutilização de componentes.

Este projeto segue uma estrutura de camadas comum, dividida em controladores, serviços, domínios (entidades), DTOs (Data Transfer Objects) e DAO (Data Access Object). Vamos analisar mais detalhadamente os principais aspectos do código:

1. Hibernate e Mapeamento Objeto-Relacional:

O Hibernate é um framework de persistência que simplifica a interação com bancos de dados relacionais, permitindo que as entidades Java sejam mapeadas diretamente para tabelas no banco de dados. No código fornecido, as anotações como @Entity, @Table, @ManyToOne e @JoinColumn são utilizadas para definir o mapeamento das entidades para o banco de dados.

2. Lombok:

Lombok é uma biblioteca que automatiza a geração de código boilerplate, como construtores, getters e setters. Isso simplifica o código e reduz a quantidade de código repetitivo. No código fornecido, as anotações @Getter, @Setter, @NoArgsConstructor, e @AllArgsConstructor são usadas para gerar automaticamente esses elementos.

3. Validator Hibernate:

A validação de dados é uma parte crucial de qualquer aplicação. O Hibernate Validator é utilizado neste projeto para validar campos específicos das entidades. Por exemplo, a anotação @NotBlank é usada para garantir que determinados campos não estejam vazios.

4. Enums:
Enums são usados para representar conjuntos fixos de valores. No projeto, enums são utilizados para representar tipos de pagamento, métodos de pagamento, contratos, etc.

5. Estrutura de Entidades:
O projeto está estruturado em torno de entidades como Agenda, Banco, Cartao_ponto, Empregado e Sindicato. Cada entidade representa uma tabela no banco de dados e contém atributos correspondentes aos campos dessa tabela.

REFERÊNCIAS

http://www.dsc.ufcg.edu.br/~jacques/projetos/common/easyaccept/usermanual.html