Sistema de Agendamento Transferência Financeira

Autor: Jimmy Zolezzi Slapelis

sumário

Objetivo 3

Resumo 3

Requisitos Funcionais 4

Requisitos não funcionais 5

Linguagem de desenvolvimento utilizada 5

Frameworks utilizados 5

servidor de aplicação 5

bando de dados utilizados 5

Frameworks utilizados 5

configurações do sistema 6

Criação do banco de dados 6

configuração banco de dados no sistema 6

Geração do war para deploy 6

**1. Objetivo**

Desenvolver um sistema de agendamento de transferência financeira entre Contas.

**2.Resumo**

O Sistema terá uma interface visual para o cadastramento e a visualização de contas. Além disso o sistema também possuirá uma tela com os agendamentos de transferência realizados com suas respectivas datas e status.

**3.Requisitos Funcionais**

1.Desenvolver um sistema de agendamento de transferências financeira que receba as seguintes informações:

-Conta de origem (padrão XXXXX-X)

-Conta de destino (padrão XXXXX-X)

-Valor da transferência

-Taxa (a ser calculada)

-Data do agendamento

-Tipo (A, B, C, D)

2. Cada tipo de transação segue uma regra diferente para cálculo da taxa

A: Operações do tipo A tem uma taxa de $2 mais 3% do valor da transferência.

B: Operações do tipo B tem uma taxa de: $10 para pedidos com agendamento até 30 dias da data de cadastro $8 para os demais

C: Operações do tipo C tem uma taxa regressiva conforme a data de agendamento:

* Maior que 30 dias da data de cadastro ­
* 1.2% até 30 dias da data de cadastro ­
* 2.1% até 25 dias da data de cadastro
* ­ 4.3% até 20 dias da data de cadastro
* ­ 5.4% até 15 dias da data de cadastro
* ­ 6.7% até 10 dias da data de cadastro
* ­7.4% até 5 dias da data de cadastro ­ 8.3%

D: Operações do tipo D tem a taxa igual a A, B ou C dependendo do valor da transferência.

* Valores até $25.000 seguem a taxação tipo A
* Valores de $25.001 até $120.000 seguem a taxação tipo B
* Valores maiores que $120.000 seguem a taxação tipo C

3. O usuário deve poder ver todos agendamentos cadastrados.

**4.Requisitos Não Funcionais**

**4.1 Linguagem de Desenvolvimento Utilizada**

Java 8: Versão mais atual da linguagem

**4.2 Frameworks Utilizados**

1) Spring MVC: Framework bastante utilizado no mercado e de fácil implementação.

2) Spring JDBC Template: Possibilita deixar a camada de acesso a base de dados mais limpa e melhor entendimento para os desenvolvedores. Além a sua utilização permite o controle de transações pelo framework do Spring.

3) JUnit: Fácil implementação e possibilita fazer teste unitários do negócio da aplicação melhorando a qualidade do sistema desenvolvido.

**4.2 Servidor da Aplicação**

Tomcat 8: Servidor leve e bastante utilizado no mercado

**4.3 Banco de Dados Utilizado**

MySQL: Banco de dados gratuito com muita adesão no mercado.

**5 Configurações do Sistema**

**5.1 Criação do Banco de Dados**

Utilizando uma ferramenta de gerenciamento de banco de dados para Mysql

Execute os comandos que estão no arquivo bancoDadosCriacao.txt

**5.2 Configuração Banco de Dados**

Realizar a configuração de conexão com banco de dados da aplicação no arquivo spring-context.xml que está na pasta resources do Projeto:

<bean id=*"dataSource"* class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*>

<property name=*"driverClassName"* value=*"com.mysql.cj.jdbc.Driver"* />

<property name=*"url"* value=*"jdbc:mysql://localhost/agendamentotransferenciabd?serverTimezone=UTC"*/>

<property name=*"username"* value=*"root"* />

<property name=*"password"* value=*"123456"* />

</bean>

1) atualizar o caminho da url;

2) atualizar o username e o password;

**5.3 Geração do war da aplicação para Deploy**

A Geração do war da aplicação pode através de linha comando do maven. Basta abrir a pasta do projeto onde está o arquivo pom.xml e executar o seguinte comando:

mvn package

Após gerado war, pode ser feita o deploy da aplicação no Tomcat.