

TCM - Projeto SolarDragons

Nome dos Alunos:

David S C Ribeiro - 824218485

Nicolas M Soares - 825116833

João P S Garcia - 825125540

Leonardo Balbino - 825155775

Juan C Fargin - 825151848

Gustavo R Leal - 825149894

1. Introdução
1.1. Tema
1.2. Objetivos a Serem Alcançados
1.3. Escopo Principal
2. Modelo de Processo
3. Requisitos do Sistema
3.1. Requisitos Funcionais
3.2. Requisitos Não-Funcionais
3.3. Especificações de Requisitos
4. Protótipo de Interface
5. Diagramas UML
5.1. Diagrama de Caso de Uso
5.2. Diagrama de Classes
5.3. Diagrama de Sequência
5.4. Diagrama de Atividades
6. Banco de Dados
6.1. Diagrama Conceitual
6.2. Diagrama Lógico
7. Anexo I

1. Introdução

O projeto SolarDragons é um sistema desktop desenvolvido em Java que tem como objetivo simular a economia gerada a partir da instalação de sistemas de energia solar em residências brasileiras.

1.1. Tema

Energia Limpa e Acessível - ODS 7 (ONU)

Simulação de economia e retorno financeiro com a utilização de painéis solares em residências.

1.2. Objetivos a serem alcançados

- Facilitar o entendimento do retorno financeiro com energia solar.
- Promover a conscientização sobre energia limpa.
- Automatizar os cálculos de economia e payback.
- Simular dados reais com base em tarifas por estado.
- Gerar relatórios CSV e apresentar gráficos comparativos.
- Público-alvo: proprietários de residências e interessados em energia solar.
- Plataforma: Java (desktop), com biblioteca Swing e XChart.

1.3. Escopo principal

- Cadastro completo de clientes com dados automatizados via CEP.
- Simulação de economia com cálculo de consumo, investimento e retorno.
- Geração de relatórios em CSV.
- Gráficos interativos comparando investimento inicial e economia em 5 anos

2. Definição do Modelo de Processo

O projeto foi desenvolvido com base na metodologia ágil SCRUM, utilizando sprints curtas, backlog de tarefas e controle com a ferramenta Trello.

3. Requisitos do Sistema de Software

3.1. Requisitos Funcionais

- RF01: O sistema deve permitir o cadastro de clientes com nome, CPF e endereço.
- RF02: O sistema deve consultar o endereço automaticamente via API de CEP.
- RF03: O sistema deve realizar a simulação de economia com base no valor da conta.
- RF04: O sistema deve calcular e exibir o tempo de retorno (payback).
- RF05: O sistema deve permitir a geração de relatórios em CSV.
- RF06: O sistema deve exibir gráficos comparativos de economia vs investimento.

3.2. Requisitos Não-Funcionais

- RNF01: O sistema deve ser desenvolvido em Java 21.
- RNF02: O sistema deve funcionar em ambiente desktop (Swing).
- RNF03: O sistema deve utilizar API externa (ViaCEP) para preenchimento automático.
- RNF04: O sistema deve apresentar resultados em linguagem simples e acessível.
- RNF05: O sistema deve responder com menos de 2 segundos em operações comuns.

3.3. Especificação de Requisitos

RF01 – Cadastro de Cliente

- **Entrada:** Nome, CPF, CEP e número da residência.

-
- **Processamento:** Validação de CPF, consulta à API ViaCEP, preenchimento automático de endereço.
 - **Saída:** Cliente cadastrado em memória, com endereço completo salvo.

RF02 – Consulta de Endereço via CEP

- **Entrada:** CEP digitado pelo usuário.
- **Processamento:** Requisição HTTP GET à API ViaCEP, tratamento do JSON de resposta.
- **Saída:** Logradouro, bairro, cidade e estado preenchidos automaticamente.

RF03 – Simulação de Economia

- **Entrada:** Valor médio da conta de luz.
- **Processamento:** Cálculo do consumo estimado, geração solar, quantidade de módulos, área necessária, custo do sistema e economia anual.
- **Saída:** Exibição dos dados simulados e estimativa de payback (retorno sobre o investimento).

RF04 – Exibição de Gráfico de Economia

- **Entrada:** Nenhuma (interno ao sistema).
- **Processamento:** Análise dos dados de clientes com simulação.
- **Saída:** Gráfico interativo comparando economia acumulada em 5 anos com o investimento inicial.

RF05 – Geração de Relatório CSV

- **Entrada:** Nenhuma (ação via menu).
- **Processamento:** Escrita em arquivo relatorio-clientes.csv com dados simulados de cada cliente.
- **Saída:** Arquivo .csv salvo no diretório do projeto, com dados estruturados por cliente.

RF06 – Apresentação do Payback em Linguagem Acessível

- **Entrada:** Resultado numérico de anos calculado no payback.
- **Processamento:** Conversão do valor em uma frase como:
“Retorno de investimento em 1 ano e 6 meses”.
- **Saída:** Frase descritiva apresentada ao usuário.

4. Protótipo de Interface

Interface Swing com menus de seleção.

Exibição em terminal das simulações (em breve expansão para interface gráfica completa).

Gráfico interativo exibido em janela com XChart.

5. Diagramas da UML

5.1. Diagrama de Caso de Uso

- Cadastrar Cliente
- Realizar Simulação
- Gerar Relatório
- Exibir Gráfico

5.2. Diagrama de Classes

- Cliente,
- SimulacaoEnergia
- Endereco
- Sistema
- ViaCEP
- GraficoEconomia

5.3. Diagrama de Sequência

Sequência de cadastro e simulação
(usuário -> sistema -> API -> retorno -> simulação)

5.4. Diagrama de Atividades

Fluxo de simulação:
Início -> Buscar cliente -> Inserir conta -> Calcular -> Mostrar resultado

6. Banco de Dados

Atualmente o sistema trabalha com dados em memória, mas já estruturado para futura expansão com MySQL.

6.1. Diagrama conceitual(imaginado)

- Tabela Cliente (CPF, nome, endereço, etc.)
- Tabela Simulacao (id, valor conta, consumo, economia, payback, etc.)

6.2. Diagrama Lógico

Relacionamento 1 para 1 entre Cliente e SimulacaoEnergia.

Anexo I

GitHub: [SolarDragons](https://github.com/davidcaetanor/SolarDragons) - <https://github.com/davidcaetanor/SolarDragons>

