

# TÉCNICAS DE PROGRAMAÇÃO 1

## Contexto do Projeto

## Introdução

O sistema desenvolvido tem como objetivo facilitar o planejamento de viagens por meio de uma aplicação que possibilita a realização de operações CRUD (Create, Read, Update, Delete). A aplicação oferece uma interface intuitiva para gerenciamento de viagens, destinos, atividades e hospedagens, além de garantir a consistência dos dados por meio da aplicação de regras de negócio específicas.

## Objetivo do Sistema

O sistema desenvolvido permite as seguintes operações e funcionalidades:

### Operações CRUD

- **Viajante:**
  - Criar conta.
  - Autenticar-se no sistema.
- **Viagem, Destino, Atividade, Hospedagem:**
  - Criar.
  - Consultar (ler).
  - Atualizar (com restrições na chave primária).
  - Excluir.

### Funcionalidades Adicionais

- Consultar o custo total de uma viagem.
- Listar:
  - Viagens associadas a um viajante.
  - Destinos associados a uma viagem.
  - Atividades associadas a um destino.
  - Hospedagens associadas a um destino.
- Apresentar listagens com os códigos e nomes dos registros.

## Regras de Negócio

- Não é permitido excluir um registro com dependências associadas (ex: excluir uma viagem que possui destinos).
- A data de uma atividade deve estar dentro do intervalo definido pelas datas de início e término do destino associado.

## Estrutura do Projeto

lua

CopiarEditar

project/

```
-- include/                                # Arquivos de cabeçalho (.hpp)
|  |-- dominios/                            # Domínios
|  |  |-- Nome.hpp
|  |  |-- Codigo.hpp
|  |  |-- Senha.hpp
|  |  |-- Avaliacao.hpp
|  |  |-- Data.hpp
|  |  |-- Dinheiro.hpp
|  |  |-- Duracao.hpp
|  |  |-- Horario.hpp
|  |-- entidades/                          # Entidades
|  |  |-- Conta.hpp
|  |  |-- Destino.hpp
|  |  |-- Hospedagem.hpp
|  |  |-- Viagem.hpp
|  |  |-- Atividade.hpp
|  |  |-- Viajante.hpp
|-- testes/                                # Testes
|  |-- TestBase.hpp
|  |-- TesteNome.hpp
|  |-- TesteCodigo.hpp
|  |-- TesteSenha.hpp
|  |-- TesteAvaliacao.hpp
|  |-- TesteData.hpp
|  |-- TesteDinheiro.hpp
|  |-- TesteDuracao.hpp
|  |-- TesteHorario.hpp
|-- src/                                    # Implementações (.cpp)
|  |-- dominios/
```

```
|      |-- entidades/
|      |-- testes/
|-- test_data/          # Dados de teste
|-- main.cpp            # Ponto de entrada principal
|-- README.md           # Documentação geral
|-- project.cbp         # Arquivo do projeto Code::Blocks
```

## Funcionalidades Concluídas

- Implementação completa das classes de **domínios** e **entidades**.
- Testes unitários desenvolvidos para cada classe de domínio e entidade.
- Estrutura modular do código organizada em pastas.
- Estrutura para dados de teste com arquivos externos.

## Boas Práticas Adotadas

- Segregação de código por funcionalidade (domínios, entidades, testes).
- Organização de arquivos com nomenclatura clara e padrões consistentes.
- Uso de testes unitários para validação de cada classe e funcionalidade.
- Estrutura preparada para extensão (facilitando futuras implementações de camadas adicionais).

## Conclusão

O sistema de planejamento de viagens foi concluído com sucesso, atendendo a todos os requisitos funcionais e não funcionais especificados. As camadas de apresentação e serviço estão implementadas, com suas interfaces devidamente documentadas e codificadas. O sistema foi testado e está pronto para ser integrado em um ambiente de produção.