

# RELATÓRIO EXPLICATIVO – SISTEMA DE COWORKING

Este relatório descreve o funcionamento e a estrutura do sistema de Coworking desenvolvido, com uso de Programação Orientada a Objetos (POO). Ele demonstra como cada classe trabalha em conjunto para formar as funcionalidades de cadastro, autenticação, verificação de capital, reservas, pagamentos e geração de relatórios.

## **1. Arquitetura Geral do Sistema**

O sistema foi estruturado com modularização completa. A classe principal SistemaCoworking atua como coordenadora, instanciando e conectando módulos como SistemaReservas, SistemaPagamentos, Relatorios, MenuPrincipal e VerificadorCidades. Isso segue os princípios de responsabilidade única.

## **2. Classe Pessoa e Herança**

A classe abstrata Pessoa representa qualquer usuário do sistema. Ela define atributos essenciais como nome, email, senha, CEP, cidade, endereço buscado automaticamente via API ViaCEP, além de saldo, histórico de pagamentos e reservas. As subclasses Visitante, ClienteMais e Vip herdam Pessoa e implementam regras próprias de descontos e privilégios.

## **3. Consulta de CEP e Endereço**

Através da API ViaCEP, o sistema converte o CEP informado em um endereço completo automaticamente, incluindo logradouro, bairro, UF e principalmente a cidade — usada nas regras de verificação para reservar espaços apenas em capitais brasileiras.

## **4. Controle de Capitais**

A classe VerificadorCidades funciona como um Singleton, garantindo uma única instância global com a lista de capitais brasileiras. Ela permite verificar se a cidade de um usuário é capital e comparar cidades ignorando diferenças de maiúsculas/minúsculas.

## **5. Classe Espaco e Tipos**

Espaco representa um local físico para aluguel, contendo atributos como nome, tipo de sala, valor por hora, localização, capacidade e se o espaço é exclusivo para usuários VIP. Os tipos de espaço são definidos no enum TipoEspaco.

## **6. Reservas**

SistemaReservas lida com toda a lógica de reserva: filtrar espaços pela cidade do usuário, verificar se o usuário mora em uma capital, impedir conflitos de horários, validar acesso VIP e descontar o valor do saldo do usuário. Caso haja sucesso, uma instância de Reserva é criada.

## **7. Pagamentos**

O módulo SistemaPagamentos permite adicionar saldo ao usuário, registrar transações e armazená-las em arquivos. Cada pagamento é salvo e também incluído no histórico do usuário.

## **8. Relatórios**

A classe Relatorios gera informações importantes como taxa de ocupação, espaços disponíveis ou ocupados, total gasto em reservas e histórico financeiro do usuário. Tudo é exibido no console.

## **9. MenuPrincipal (Interface do Usuário)**

A classe MenuPrincipal controla todas as interações com o usuário via console. Inclui cadastro, login, aluguel, visualização de espaços, pagamento, benefícios, relatórios e logout. É o ponto principal da UI.

## **10. Funcionamento do Sistema**

O sistema garante fluxo completo: o usuário se cadastra, tem a cidade detectada via CEP, adiciona saldo, vê espaços na sua capital, faz reservas, consulta histórico e gera relatórios. Todas as funcionalidades foram implementadas conforme as regras propostas no projeto.