**Universidade Federal de Goiás**

**Goiânia, 09 de julho de 2023**

**Curso:** Sistemas de Informação

**Aluno:** Victor Hugo Benatti Franco **Matrícula:** 202203532

**Aluno:** Murilo Henrique Barbosa Silva **Matrícula:** 202204399

**Aluno:** Felipe Augusto Lopes Vera **Matrícula:** 202203507

**Primeira Entrega Projeto – Programação Orientada a Objetos**

**Tema escolhido:** Academia

**Escopo e problema:**

A academia tem enfrentado desafios na gestão de suas informações, o que acarreta em processos manuais e dificuldades operacionais. A falta de um sistema integrado dificulta o controle eficiente de clientes, contratos, contas a receber e a pagar, estoque, fornecedores, produtos, máquinas e manutenções. Essa falta de integração resulta em erros, retrabalho e dificuldades na tomada de decisões financeiras.

Para solucionar esses problemas, o projeto de Programação Orientada a Objetos foi proposto. Através desse projeto, será desenvolvido um sistema que permitirá uma gestão mais eficiente e automatizada de todas as áreas da academia. Com a implementação desse sistema, será possível centralizar e organizar as informações, simplificar os processos operacionais, melhorar o controle financeiro, agilizar a tomada de decisões e oferecer uma experiência aprimorada tanto para os clientes como para os funcionários. Esse projeto tem como objetivo proporcionar uma gestão mais eficaz e otimizada, permitindo que a academia foque em seu principal objetivo e não percam tanto recursos em processos que podem ser automatizados.

**Classes:**

Pessoa: Esta é uma classe abstrata que serve como base para outras classes que representam diferentes tipos de pessoas, como Clientes e Funcionários. Inclui atributos comuns, como email, telefone, ID da pessoa e endereço.

PessoaFisica e PessoaJuridica: Estas são subclasses de Pessoa que adicionam atributos específicos para pessoas físicas (como nome, CPF e data de nascimento) e jurídicas (como nome fantasia e CNPJ).

Cliente: Esta classe estende PessoaFisica e representa um cliente da empresa. Cada cliente tem um ID único, um array de contratos associados a ele e um contador para gerar IDs únicos para novos clientes.

Funcionario: Esta classe estende PessoaFisica e representa um funcionário da empresa. Cada funcionário tem um ID único e um contador para gerar IDs únicos para novos funcionários.

Fornecedor: Esta classe estende PessoaJuridica e representa um fornecedor para a empresa. Cada fornecedor tem um ID único, um array de contas a pagar associadas a ele e um contador para gerar IDs únicos para novos fornecedores.

Endereco: Esta classe representa um endereço, incluindo ID, rua, número, bairro, cidade, estado e CEP.

Produto: Esta classe representa um produto, incluindo atributos como nome, descrição e ID único.

Manutencao: Representa uma manutenção em uma máquina, incluindo atributos como descrição, data, e máquina associada.

Máquina: Representa uma máquina, incluindo atributos como nome, valor, descrição e um array de manutenções associadas.

ContaPagar: Representa uma conta a pagar da empresa, com um ID único, valor da conta, observações e uma lista de produtos associados à conta. Também possui um contador para gerar IDs únicos para novas contas a pagar.

ContaReceber: Similar à classe ContaPagar, esta classe representa uma conta a receber pela empresa. Ela possui um ID único, valor da conta, observações e um contador para gerar IDs únicos para novas contas a receber.

Contrato: Representa um contrato feito pela empresa. Cada contrato tem um ID único, data de assinatura, data de vencimento e um array de contas a receber associadas a ele. Também possui um contador para gerar IDs únicos para novos contratos.

Estoque : Esta classe representa um estoque, incluindo atributos como quantidade, produto ao qual ele se refere, quantidade em estoque e ID único.

MostraEndereço: possui um método abstrato para imprimir o endereço da Pessoa.