

**Discussão de Problema – Diagrama de Classes e de Pacotes**

INFORMAÇÕES DOCENTE						
CURSO:	DISCIPLINA:	TURNO	MANHÃ	TARDE	NOITE	PERÍODO/SALA:
ENGENHARIA DE SOFTWARE	PROJETO DE SOFTWARE		x		x	4º
PROFESSOR (A): João Paulo Carneiro Aramuni						

**Diagrama de Classes e de Pacotes para um sistema de IA como Serviço (AIaaS)**

**Cenário:** Um Sistema de IA como Serviço (AIaaS) permite que usuários criem projetos de Inteligência Artificial sem precisar gerenciar a infraestrutura subjacente. O sistema oferece ferramentas para upload de datasets, treinamento, avaliação e implantação de modelos de forma escalável, permitindo que os melhores modelos sejam disponibilizados como APIs. Além disso, o AIaaS controla métricas de desempenho, gerencia pagamentos e garante segurança de acesso.

**Requisitos para o sistema AIaaS:**

1. O usuário acessa o sistema para criar um projeto de IA e cadastrar os datasets necessários.
2. O sistema verifica a validade e integridade dos datasets enviados.
3. Se os datasets estiverem válidos, o usuário poderá iniciar o treinamento de um modelo de IA.
4. Após o treinamento, o usuário deve avaliar o modelo utilizando métricas fornecidas pelo sistema (como acurácia, precisão, recall e F1-score).
5. O sistema permite implantar o modelo como uma API, disponibilizando um endpoint para consumo em aplicações externas.
6. O usuário pode monitorar o desempenho do modelo em produção e receber alertas sobre degradação de performance.
7. O usuário pode cancelar ou substituir um modelo implantado, e o sistema atualizará o status do projeto e da API.
8. Para recursos premium (como GPUs avançadas ou deploy escalável), o usuário deve realizar o pagamento, e o sistema confirma a transação.

**Parte 1 – Diagrama de Classes UML - Pacotes e Classes sugeridas**

model (Domínio )

- Usuario
- ProjetoIA
- Dataset
- ModeloIA
- Metrica
- Implantacao
- Pagamento

view (Apresentação )

- DashboardView
- DatasetView
- ModeloView



**PUC Minas**

controller (🟡)

- ProjetoController
- DatasetController
- ModeloController
- PagamentoController

service (🟢)

- TreinamentoService
- AvaliacaoService
- ImplantacaoService
- PagamentoService

repository (🟡)

- ProjetoRepository
- DatasetRepository
- ModeloRepository
- PagamentoRepository

facade (🔴)

- IAFacade

security (⬛)

- AuthService
- Permissao

## Tarefa

1. Construa o Diagrama de Classes UML usando as classes acima.
2. Inclua atributos e métodos em cada classe.  
Ex.:
  - Usuario: id, nome, email | criarProjeto(), realizarPagamento()
  - ProjetoIA: id, nome, descricao | adicionarDataset(), treinarModelo()
3. Represente associações e cardinalidades entre as classes (1..1, 1..\*, 0..1, etc.).
4. Diferencie as camadas por cores no diagrama (Domínio 🟢, View 🟢, Controller 🟡, Service 🟢, Repository 🟡, Facade 🔴, Security ⬛).

## Relações principais entre as classes

- Usuario → ProjetoIA (1..\*): um usuário pode ter vários projetos.
- ProjetoIA → Dataset (0..\*): cada projeto pode ter vários datasets.
- ProjetoIA → ModeloIA (0..\*): cada projeto pode gerar vários modelos.
- ModeloIA → Metrica (1..\*): cada modelo é avaliado por métricas.
- ModeloIA → Implantacao (0..1): um modelo pode gerar uma implantação.
- Usuario → Pagamento (0..\*): cada usuário pode realizar vários pagamentos.
- Controllers usam Services que acessam Repositories.
- IAFacade orquestra operações complexas (treinar, implantar, avaliar).
- AuthService/Permissao controlam acesso aos usuários.

### Especificação das interações (sequência de uso das classes)

1. Usuário cria um ProjetoIA (ProjetoController → ProjetoService → ProjetoRepository).
2. Usuário faz upload de Dataset (DatasetController → DatasetService → DatasetRepository).
3. Usuário solicita treino de ModeloIA (ModeloController → TreinamentoService → ModeloRepository).
4. Sistema calcula Métrica (AvaliacaoService).
5. Usuário escolhe implantar modelo (ModeloController → ImplantacaoService → Implantacao).
6. Usuário efetua Pagamento (PagamentoController → PagamentoService → PagamentoRepository).
7. IAFacade simplifica o fluxo de alto nível.
8. AuthService/Permissao validam acesso a todas as ações.

### Parte 2 – Diagrama de Pacotes UML - Pacotes sugeridos

- **model** → classes de domínio (Usuario, ProjetoIA, Dataset, ModeloIA, Metrica, Implantacao, Pagamento)
- **view** → classes de apresentação (DashboardView, DatasetView, ModeloView)
- **controller** → classes de controle (ProjetoController, DatasetController, ModeloController, PagamentoController)
- **service** → classes de serviço (TreinamentoService, AvaliacaoService, ImplantacaoService, PagamentoService)
- **repository** → classes de persistência (ProjetoRepository, DatasetRepository, ModeloRepository, PagamentoRepository)
- **facade** → classe de fachada (IAFacade)
- **security** → classes de segurança (AuthService, Permissao)

### Tarefa

1. Construa o Diagrama de Pacotes UML usando os pacotes acima.
2. Indique dependências entre pacotes:
  - Controllers → Services
  - Services → Repositories
  - Facade → Services
  - Views → Controllers
  - Security → qualquer pacote que precise de autenticação/autorização
3. Use cores para diferenciar pacotes, se desejar, seguindo o padrão do Diagrama de Classes.