

Sistema de Gestão Escolar – Arquitetura de Sistemas

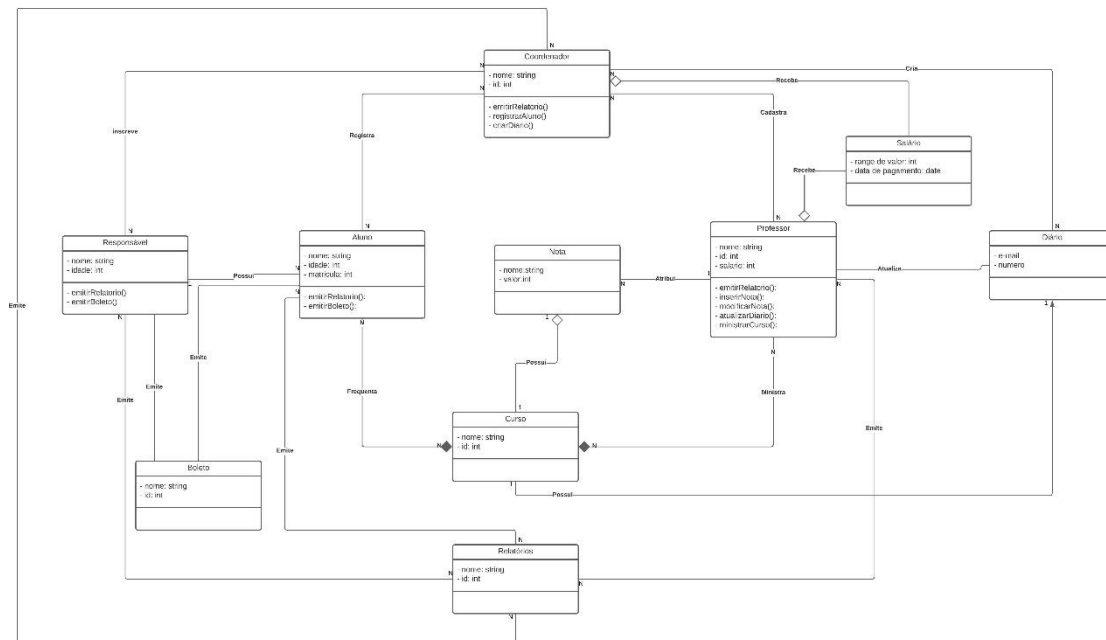
Livia de Oliveira Faria¹, Lucas Geraldo Ribeiro de Faria¹, Wesley Bernardes de Souza Júnior¹

¹Fundação de Ensino e Pesquisa de Itajubá (FEPI)
– Itajubá – MG – Brasil

{00020451, 00019790, 00020321}@fepi.edu.br

1. Questão

O nosso diagrama de classes ficou definido da seguinte forma, perante aos requisitos propostos pelo exercício:



2. Questão

A) Existem diversos padrões arquiteturais que pode ser utilizados para atender as necessidades do projeto, sendo assim, vale salientar:

- **Arquitetura em Camadas (Layered Architecture):** esse padrão envolve a divisão do sistema em camadas distintas, como a camada de apresentação (interface do usuário), a camada de lógica de negócios (ou camada de aplicação) e a camada de acesso a dados.
- **Model-View-Controller (MVC):** esse padrão divide a aplicação em três componentes principais: o Modelo (que representa os dados e a lógica de negócios), a Visão (que cuida da apresentação e interface do usuário) e o Controlador (que atua como intermediário entre o Modelo e a Visão).
- **Arquitetura Hexagonal (Port and Adapter):** esse padrão promove a ideia de que o núcleo da aplicação (lógica de negócios) deve ser independente de qualquer tecnologia ou detalhe de implementação, incluindo a interface do usuário e o acesso ao dados, consistindo assim, em camadas concêntricas em torno do núcleo, onde os adaptadores (portas) são usados para conectar o núcleo a diferentes interfaces e fontes de dados.
- **Model-View-ViewModel (MVVM):** esse padrão é semelhante ao MVC, mas coloca mais ênfase na vinculação de dados e na separação dos componentes

de interface do usuário (Visão) e lógica de negócios (Modelo) usando um intermediário chamado ViewModel.

- **Arquitetura de Microserviços:** esse padrão arquitetural divide um sistema em conjuntos de serviços independentes, cada um com sua própria lógica de negócios e banco de dados, na qual cada microserviço pode usar a arquitetura de camadas internamente para organizar seu próprio código.

B) O padrão de arquitetura proposto pela equipe para resolução do desafio, foi o Model View Controller (MVC), promovendo uma estrutura organizada e modular, facilitando a manutenção e escalabilidade do código, além disso, o MVC é compatível com a implementação de testes, permitindo testar de forma eficaz as partes individuais do sistema, além de separar claramente as responsabilidades e as capacidades de realizar teste de unidade e integração.

C) Há várias plataformas e frameworks que podem ser utilizados para dar suporte ao nosso desenvolvimento, sendo eles:

Plataforma/Framework	Explicação
Django (Python)	Django é um framework web em Python que segue o padrão de arquitetura MVC. Ele fornece um ambiente altamente produtivo para desenvolver aplicativos web, com um ORM (Object-Relational Mapping) eficiente para gerenciar o modelo de dados, um sistema de roteamento e controle de URL para o controlador e suporte a templates para as visualizações.
Ruby on Rails (Ruby)	Ruby on Rails é um framework web em Ruby que também segue o padrão MVC. Ele oferece uma estrutura altamente coesa para desenvolver aplicativos da web de forma rápida e eficiente, com convenções bem definidas que se alinham ao MVC.
Spring Framework (Java)	O Spring Framework é uma plataforma abrangente para desenvolvimento de aplicativos Java e suporta uma variedade de abordagens arquiteturais, incluindo MVC. O Spring MVC é um módulo específico para desenvolvimento de aplicativos web que adota o padrão MVC.

ASP.NET (C#)	ASP.NET é um framework da Microsoft para o desenvolvimento de aplicativos web. Ele inclui o ASP.NET MVC, que permite a criação de aplicativos web baseados em MVC usando C# e .NET. É uma escolha sólida para desenvolvedores que trabalham no ecossistema da Microsoft.
--------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3. Questão

A configuração do cronograma de desenvolvimento do projeto, utilizando a metodologia Scrum, ficou definido como:

- Responsabilidades

O papel de *Scrum Master* será desempenhado pela Lívia de Oliveira Faria, já o papel de *Product Owner* será desempenhado pelo Lucas Geraldo Ribeiro de Faria, por fim, o time de desenvolvimento será desempenhado pelo aluno Wesley Bernardes de Souza Júnior.

- *Sprint planning*

O nosso planejamento das sprint ficou definidos conforme o quadro abaixo:

Identificação	Duração
Sprint 1	02/10 até 16/10
Sprint 2	16/10 até 30/10
Sprint 3	30/10 até 13/11
Sprint 4	13/11 até 27/11
Sprint 5	27/11 até 11/12

- *Daily scrum*

As reuniões de alinhamento diário serão feitas no período da manhã, às **09:00 horas** e , no período da tarde, às **17:00 horas**.

- Reuniões para revisão do andamento da *sprint*

Esse encontro será feito nas **quintas-feiras** às **19:00 horas**.

- Retrospectiva da *sprint*

Essa reunião será nas **segundas-feiras** às **19:00 horas**.

- *Product backlog*

No sistema a ser projetado, será a criação das telas de cadastro de alunos, professores, coordenadores, cursos e responsáveis; criação das telas de emissão de relatórios, diários e boletos; e telas para inserção de notas, além da tela para cálculo de salário.

- ***Sprint backlog***

Essas são as tarefas definidas para cada sprint, sua prioridade e tempo estimado de trabalho:

Tarefa	Prioridade	Tempo de trabalho
Criação das telas de cadastro de usuário, professores, coordenadores, cursos e responsáveis	Essencial	02/10 até 16/10
Criação das telas de emissão de relatórios, diários e boletos	Essencial	16/10 até 30/10
Criação das telas de inserção de notas ao alunos	Essencial	30/10 até 13/11
Criação da tela para o cálculo dos salários	Essencial	13/11 até 27/11

- ***Incremento/Entrega***

As primeiras entregas serão as telas de cadastros, para os usuários começarem com as entradas de dados. As seguintes entregas serão as telas de relatório, criação de diários e boletos, criação das telas de inserção de notas ao alunos e criação das telas para o cálculo dos salários.

Tarefa	Tempo de trabalho
Criação das telas de cadastro de usuário, professores, coordenadores, cursos e responsáveis	Sprint 1 até o começo da Sprint 2
Criação das telas de emissão de relatórios, diários e boletos	Sprint 3
Criação das telas de inserção de notas ao alunos	Sprint 4
Criação da tela para o cálculo dos salários	Sprint 5

