

## **MESTRADO EM ENGENHARIA INFORMÁTICA E TECNOLOGIA WEB**

### **UNIDADE CURRICULAR**

#### **Arquitetura e Padrões de Software 2022-2023**

#### **“Propostas de projeto”**

**Activity Provider: Avaliação de Desenvolvimento de Estudante de Programação com base em um CRUD**

Professores:

Hugo Paredes

Leonel Morgado

# INTRODUÇÃO

## Proposta de Activity Provider: Avaliação de Desenvolvimento de Estudante de Programação com base em um CRUD

### Descrição da Atividade:

A atividade "Avaliação de Desenvolvimento de Estudante de Programação com base em um CRUD" tem como objetivo avaliar as habilidades e conhecimentos de estudantes de programação no desenvolvimento de um sistema CRUD (Create, Read, Update, Delete). Um CRUD é uma aplicação básica que realiza operações de criação, leitura, atualização e exclusão de dados em um banco de dados.

### FUNCIONAMENTO DA ATIVIDADE:

#### 1. Interface de Configuração da Atividade (Professor):

Nesta interface, o professor configura a atividade de avaliação. O professor terá acesso a campos onde pode definir os requisitos do CRUD a ser implementado, tais como:

- Entidades e Atributos: O professor pode especificar as entidades e atributos que devem ser presentes no sistema CRUD. Por exemplo, em um sistema de gerenciamento de produtos, as entidades podem ser "Produto" e "Categoria" com atributos como "Nome", "Preço", etc.
- Funcionalidades do CRUD: O professor pode determinar quais operações de CRUD os estudantes devem implementar para cada entidade. Por exemplo, se a entidade for "Produto", as funcionalidades podem incluir "Criar novo produto", "Listar produtos", "Atualizar produto", "Excluir produto", etc.
- Tecnologias Permitidas: O professor pode definir quais tecnologias e linguagens de programação os estudantes podem usar para implementar o CRUD.

#### 2. Interface de Deploy da Atividade (Aluno):

Nesta interface, os estudantes acessam as especificações da atividade "Avaliação de Desenvolvimento de Estudante de Programação com base em um CRUD". Eles receberão:

- Descrição detalhada do projeto: Incluindo informações sobre as entidades, atributos e funcionalidades que devem ser implementadas no CRUD.
- Modelo ER do Banco de Dados: Os estudantes podem visualizar um modelo Entidade-Relacionamento (ER) que representa a estrutura do banco de dados que deve ser usado no sistema CRUD.
- Download da Aplicação Base: Os estudantes farão o download de uma aplicação base (por exemplo, uma aplicação Node.js + Express.js) que já possui a configuração básica do CRUD com rotas e conexão ao banco de dados.

#### 3. Desenvolvimento do CRUD:

Os estudantes implementam as funcionalidades do CRUD com base nas especificações fornecidas pelo professor. Eles usarão a aplicação base fornecida como ponto de partida e irão adicionar o código necessário para criar, ler, atualizar e excluir dados do banco de dados.

#### 4. Testes e Validação:

Após o desenvolvimento, os estudantes devem realizar testes e validações para garantir o correto funcionamento do CRUD. Eles também podem usar a aplicação base para testar as funcionalidades implementadas.

#### 5. Submissão e Avaliação:

Os estudantes submetem suas soluções para a avaliação do professor por meio da interface da atividade. O professor terá acesso aos códigos enviados pelos estudantes e avaliará a implementação do CRUD em relação às especificações definidas anteriormente.

## 6. Analíticas e Feedback:

O Activity Provider fornecerá analíticas sobre o desempenho de cada estudante na atividade. Isso incluirá:

- Percentagem de conclusão das funcionalidades do CRUD.
- Avaliação de cada operação de CRUD implementada pelo estudante.
- Feedback individualizado para cada estudante sobre pontos fortes e áreas de melhoria.

## Especificações da Atividade

"Avaliação de Desenvolvimento de Estudante de Programação com base em um CRUD" para Registro na Inven!RA:

### Especificação do Ficheiro JSON de Configuração (json-params.json):

O ficheiro JSON de configuração, que será gerado pela plataforma Inven!RA com base nas especificações fornecidas pelo professor, terá a seguinte estrutura:

```
{
  "atividadeCRUD": {
    "data": {
      "valor": "Data",
      "tipo": "Date"
    },
    "titulo": {
      "valor": "Titulo",
      "tipo": "String"
    },
    "descricao": {
      "valor": "Descricao",
      "tipo": "String"
    },
    "funcionalidades": [
      "funcionalidade": {
        "nome": {
          "valor": "nome",
          "tipo": "String"
        },
        "descricao": {
          "valor": "descricao",
          "tipo": "String"
        },
        "entidade": {
          "valor": "nome",
          "tipo": "String"
        },
        "atributos": [
          "atributo": {
            "nome": {
              "valor": "nome",
              "tipo": "String"
            }
          }
        ]
      }
    ]
  }
}
```

```

        }
    },
    ],
    "criar": {
        "valor": "criar",
        "tipo": "boolean"
    },
    "ler": {
        "valor": "ler",
        "tipo": "boolean"
    },
    "actualizar": {
        "valor": "actualizar",
        "tipo": "boolean"
    },
    "remover": {
        "valor": "remover",
        "tipo": "boolean"
    },
    },
    ],
    "tecnologias":[
        "tecnologia":{
            "nome": {
                "valor": "nome",
                "tipo": "String"
            }
        }
    ],
    "modeloEr": {
        "valor": "Modelo ER do Banco de Dados",
        "tipo": "file"
    },
    "aplicacaoBase": {
        "valor": "Aplicação Base",
        "tipo": "file"
    }
}
}

```

### Especificação dos Dados Analíticos (json-analytics.json):

Os dados analíticos fornecidos pela plataforma Inven!RA serão retornados em um array de documentos JSON, cada um contendo informações sobre o desempenho individual de um estudante na atividade. Cada documento JSON terá a seguinte estrutura:

```

{
    "qualitativas": [
        "Percentagem de conclusão das funcionalidades":{
            "tipo": "double"
        }
    ],
    "quantitativa": {

```

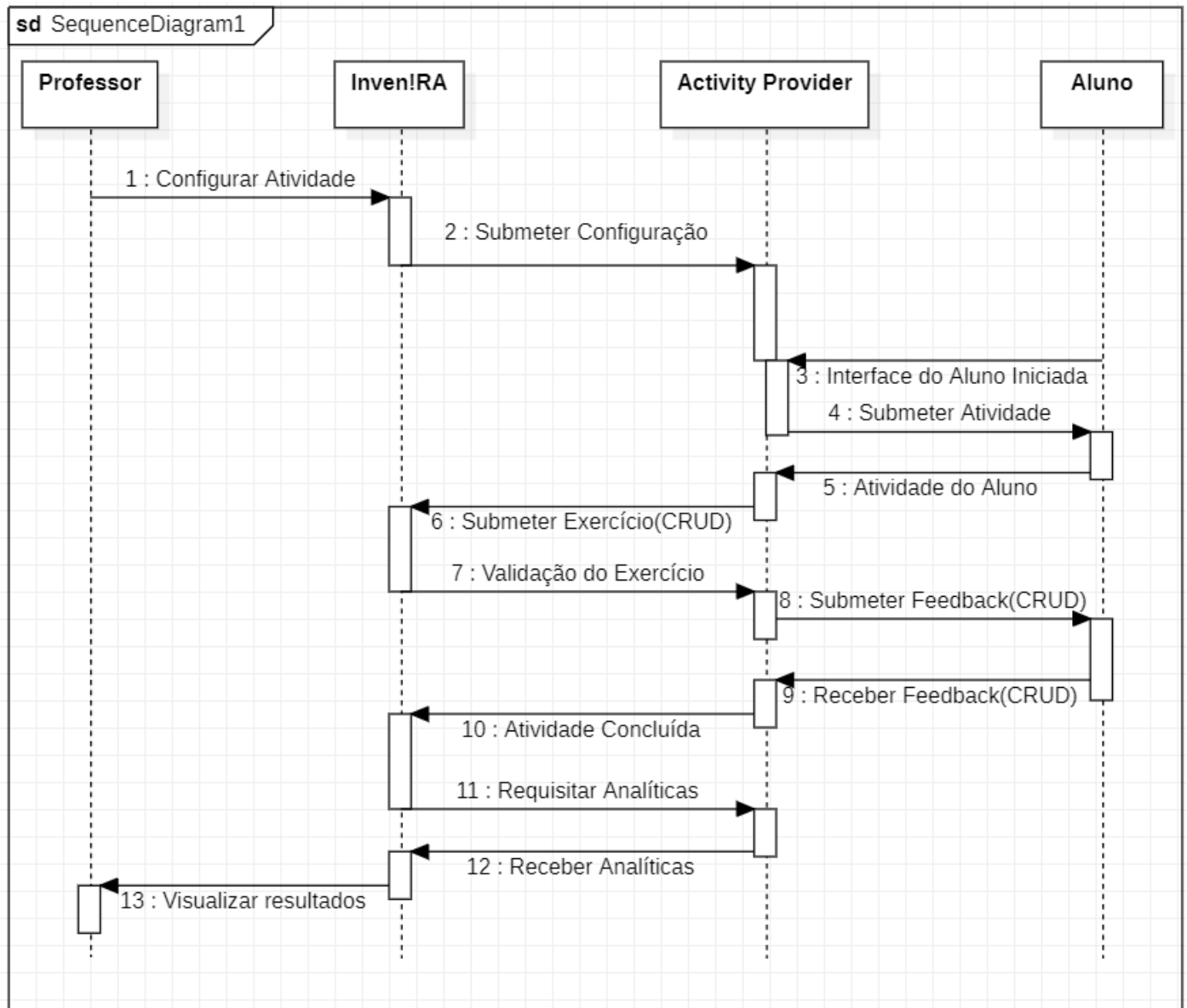
```

"crusds implementados": {
  "crud": {
    "nome":{
      "valor": "Nome",
      "tipo": "String"
    },
    "avaliacao":{
      "valor": "Avaliacao (1-10)",
      "tipo": "integer"
    },
    "comentario":{
      "valor": "Comentario",
      "tipo": "String"
    }
  }
},
"pontosfortes": {
  "valor": "Pontos fortes",
  "tipo": "String"
},
"melhoraia": {
  "valor": "Areas de melhoria",
  "tipo": "String"
}
}

```

Essas especificações permitem que a plataforma Inven!RA colete os dados relevantes da atividade e forneça aos professores informações detalhadas sobre o desempenho de cada aluno, permitindo avaliações mais precisas e feedbacks mais construtivos na implementação de um CRUD.

### Diagramas de sequência



Neste diagrama de sequência, demonstra-se a interação entre a interface do professor e a interface do aluno durante a atividade de avaliação do desenvolvimento com base em um CRUD. O professor configura a atividade, e o aluno inicia a atividade após receber a configuração. O aluno realiza as ações necessárias do CRUD, submete seus exercícios, recebe feedback e, finalmente, conclui a atividade. O professor pode então requisitar as analíticas sobre o desempenho do aluno na atividade e visualizar os resultados.

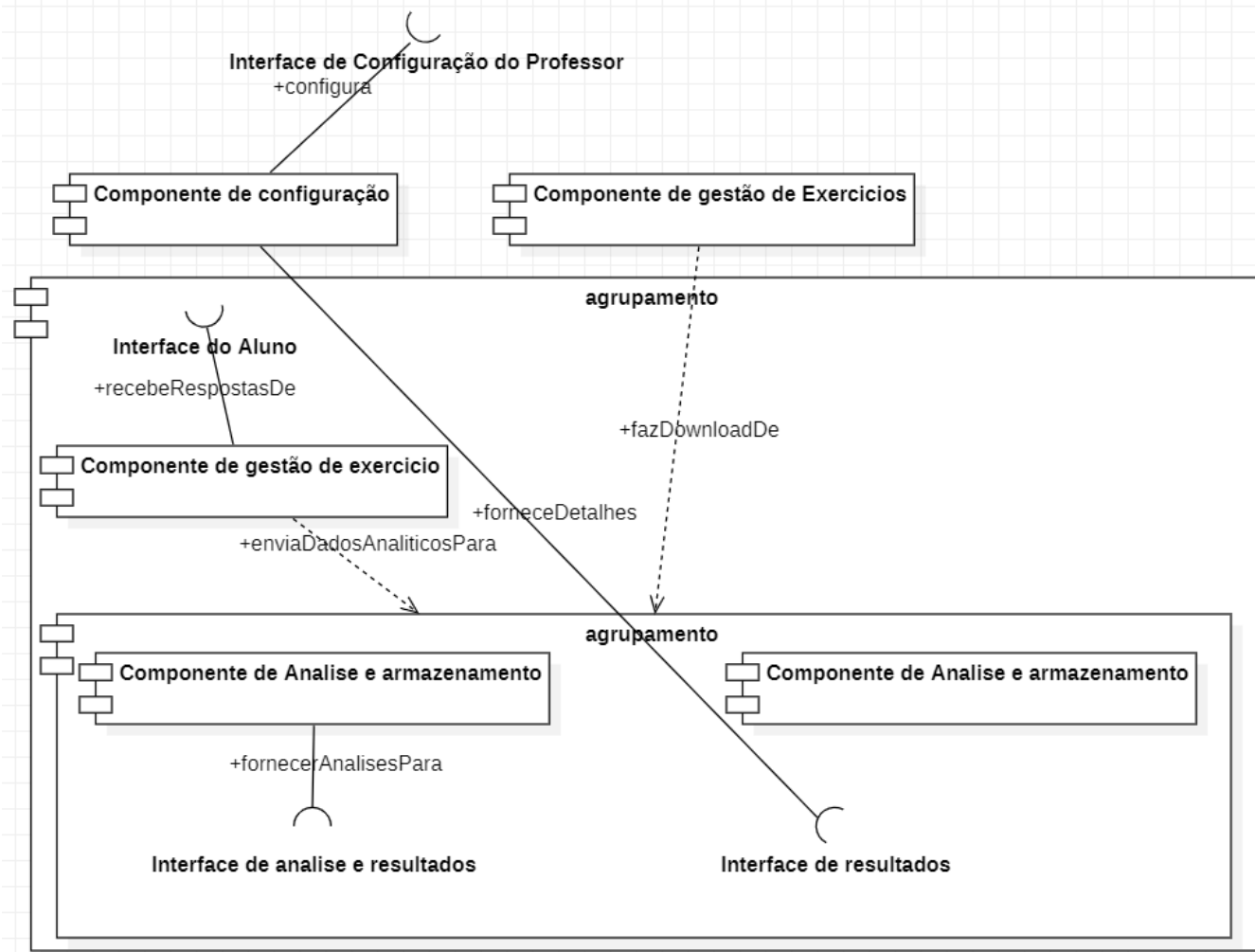
## DIAGRAMA DE COMPONENTES DO PROJETO

O diagrama de componentes mostra as interfaces utilizadas para comunicação externa (com a plataforma Inven!RA e com os usuários) e também as comunicações internas entre os componentes, juntamente com os portos de início e fim dessas comunicações. Vamos representar os seguintes componentes:

1. Interface de Configuração do Professor
2. Componente de Configuração
3. Interface do Aluno
4. Componente de Gestão de Exercícios (CRUD)
5. Componente de Análise e Armazenamento de Dados

## 6. Interface de Análise e Resultados

A seguir, apresentamos o **Diagramas de sequência**:



A Interface de Configuração do Professor é responsável por receber as configurações do professor para a atividade. Essas configurações são processadas pelo Componente de Configuração, que gerencia as informações sobre a atividade e os exercícios do CRUD. O Componente de Gestão de Exercícios (CRUD) é responsável por gerenciar os exercícios enviados pelos alunos, validar as respostas e fornecer feedback.

O Componente de Análise e Armazenamento de Dados é responsável por coletar e armazenar os dados analíticos sobre o desempenho dos alunos na atividade. A Interface de Análise e Resultados permite ao professor visualizar os resultados e análises sobre o progresso e desempenho dos alunos.

Com este diagrama de componentes, é possível seguir o fluxo de comunicação desde a configuração da atividade pelo professor até a análise dos resultados pelo professor, passando pela submissão dos exercícios pelos alunos e armazenamento dos dados analíticos.

## PADRÕES DE SOFTWARE E PADRÕES DE CRIAÇÃO

URL de teste do projeto: <https://github.com/rogeriomandala/APS>

## **PADRÕES DE ESTRUTURA**

Para o desenvolvimento da activity provider escolheu-se o padrão composite, que é um padrão de design estrutural que permite tratar objetos individuais e coleções de objetos de maneira uniforme. Permite criação de uma hierarquia de objetos em forma de árvore, onde os objetos individuais e os grupos de objetos são tratados da mesma maneira.

No activity provider do projeto tem uma classe base abstrata que define a interface comum para todos os objetos na hierarquia, dentro da classe temos métodos para adicionar, remover e acessar objetos filhos, de seguida, existe tempos as classes concretas que implementam as interfaces e representam os objetos individuais na hierarquia. Junto, existe a classe composta que implementa a interface da classe base, mas também contém uma coleção de objetos filhos que trata os objetos individuais e os grupos de objetos da mesma maneira, chamando os métodos apropriados em cada objeto.

A vantagem do padrão Composite é que ele permite que o activity provider trate objetos individuais e grupos de objetos de maneira uniforme, simplificando o código e tornando-o mais flexível. O exemplo usado é a iteração sobre todos os objetos na hierarquia, independentemente de serem objetos individuais ou grupos de objetos.

## **PADRÕES DE COMPORTAMENTO**

Para o desenvolvimento da activity provider escolheu-se o padrão Observer. Onde uma classe mantém uma lista de objetos dependentes e notifica automaticamente esses objetos quando ocorrem mudanças. Dessa forma, cada vez que um método da atividade é criado, lido, atualizado ou deletado, a classe notificará os usuários, que podem ser responsáveis por executar a avaliação necessária. A escolha da implementação desse padrão em Grails é feita criando um serviço do Grails, que é responsável por manter uma lista dos alunos que estão sendo avaliados. A classe fornece métodos para adicionar e remover usuários, além de métodos para notificar os observadores sobre as mudanças ocorridas.

Cada aluno possui um controlador do Grails, que será responsável por receber as notificações da classe e executar várias avaliações do desenvolvimento do aluno com base nas mudanças ocorridas na atividade.

Dessa forma, sempre que um método da classe for criado, lido, atualizado ou deletado, o objeto será notificado e irá notificar os alunos para que suas avaliações sejam atualizadas. permitindo uma fácil extensibilidade, pois novos alunos podem ser adicionados sem modificar a classe existentes.

## **ANTIPADRÕES E REFATORIZAÇÃO**

Feita a análise visual, dentro da revisão do código, não constatei alterações diretas em variáveis globais, não há repetição de código desnecessária e a complexidade do código está dentro de limites aceitáveis. Indicando que não incorreu nenhum antipadrão.



## CONCLUSÃO

A atividade "Avaliação de Desenvolvimento de Estudante de Programação com base em um CRUD" fornecerá uma oportunidade para os estudantes aplicarem seus conhecimentos de programação na criação de um sistema CRUD funcional. O uso de analíticas permitirá ao professor avaliar o desempenho individual de cada estudante e fornecer feedback construtivo para o desenvolvimento contínuo das habilidades de programação. Além disso, a disponibilização de uma aplicação base facilitará o início do desenvolvimento e garantirá a uniformidade nas tecnologias utilizadas pelos estudantes.

Link: do projeto: <https://github.com/rogeriomandala/APS>