



CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CAMPUS V – DIVINÓPOLIS - TÉCNICO EM INFORMÁTICA – 3º B
DISCIPLINA PID

Guia de desenvolvimento do TCC Coruja

Gabriel Coelho Costa

Professor/Orientador: André Maravilha

Divinópolis, 24 de fevereiro, 2021.

SUMÁRIO

1. Backend	03
1.1. Get Started	03
1.2. Lógica do Carógrafo	04
1.2.1. Script de Povoamento Inicial	05
1.2.2. Script de Virada de Ano	06
1.3. Considerações Finais	07

1. BACKEND

1.1. GET STARTED

Caso tenha instalado o sistema pela primeira vez siga os passos abaixo, respeitando a ordem em que estão dispostos, para configurá-lo corretamente e iniciar o desenvolvimento do backend.

- Acesse o diretório “/backend” no terminal e execute o comando “adonis serve --dev”.

Isso inicializará o servidor backend.

- Criar um schema, em uma conexão mysql, chamado “coruja”.

Todos parâmetros referentes a configuração do banco de dados estão no arquivo “backend/.env” e possuem o prefixo “DB_”, exemplo: “DB_PORT”. Porém é aconselhável que se mantenha os valores atuais, tendo em vista a padronização do sistema por toda a equipe de desenvolvimento.

- No terminal, acesse o diretório “/backend” e rode o comando “adonis migration:run”.

Os arquivos de migration permitem um controle das mudanças na estrutura do banco de dados. Ao executar tal comando, automaticamente todas tabelas, e suas respectivas configurações, serão criadas.

- Ainda no diretório “/backend”, rodar o comando “adonis seed --files CarographSeeder.js”.

Os arquivos de seed permitem a inserção de dados no banco de forma automática. Nesse caso, todas as modalidades, seus respectivos cursos e classes, e todos status são cadastrados no sistema. Os valores inseridos são extraídos dos arquivos de configuração presentes no diretório “backend/database/factoryDatas”, os quais já estão configurados para a atual demanda do sistema.

- Todas as requisições que deveriam ser feitas após o login no sistema exigem que

um token seja enviado juntamente. Nesse caso, é necessário que um pedagogo seja cadastrado e suas credencias sejam utilizadas para o login. Para isso, não é ideal que um pedagogo seja inserido manualmente no banco, visto que algumas etapas necessárias seriam burladas, como por exemplo a criptografia de sua senha. O correto é realizar uma requisição post na rota “base_url/pedagogue”, passando um json no corpo da requisição. Essa, e todas as outras requisições do sistema, estão salvas no diretório “Insomnia Settings”, e elas devem ser importadas para a ferramenta Insomnia, para que possam ser utilizadas.

1.2. LÓGICA DO CARÓGRAFO

Concluídas todas etapas da seção “GET STARTED”, o sistema está devidamente configurado. Porém, o objetivo principal do Coruja é gerenciar os alunos e seus respectivos históricos escolares ao longo do período acadêmico. Diante disso, duas funcionalidades do sistema se destacam: povoamento inicial do carógrafo e virada de ano.

O povoamento inicial do carógrafo é responsável por inserir e resgatar dados antigos, que, por exemplo, poderiam estar sendo salvos de maneira manuscrita. Dessa forma, através dessa funcionalidade, a transição do controle dos dados dos alunos da forma manual, por meio de documentos manuscritos, para a forma digital, por meio do sistema Coruja, se torna mais suave. Ou seja, esse recurso deve ser utilizado em um único momento: assim que o sistema for utilizado pela primeira vez.

Já a virada de ano é responsável por atualizar os dados dos alunos para o ano seguinte ao que se encerra. Sendo assim, a cada ano as classes de primeira série de todos os cursos recebem alunos novos e novos registros do histórico escolar dos alunos atuais são criados para o ano seguinte baseado em seus status no ano anterior. Ou seja, esse recurso deve ser utilizado a cada ano que finaliza, preparando o sistema para o próximo ciclo que se inicia.

Portanto, juntas, essas duas funcionalidades mantêm o sistema em funcionamento. Porém, como o desenvolvimento do backend se deu de forma antecipada ao desenvolvimento do front, ele foi projetado para funcionar de maneira independente, se necessário. Esse funcionamento se caracteriza por scripts que simulam as requisições que supostamente seriam realizadas pelo frontend, gerando dados de forma automática, que se adequam à atual situação do sistema e suprem tal carência. Além disso, tal arquitetura é extremamente vantajosa por permitir a realização de testes com uso massivo de dados, o

que reduz a chance de erros passarem despercebidos.

Nesse caso, foram criados um script para o recurso de povoamento inicial e outro para o recurso de virada de ano. Dessa forma, é necessário o domínio de como executar tais códigos para que o sistema seja utilizado de maneira correta.

1.2.1. SCRIPT DE POVOAMENTO INICIAL

O script de povoamento inicial se encontra no diretório “backend/database/seeds/GenerateObjectCarographSeeder.js”. Siga os passos abaixo para executá-lo corretamente.

- No terminal, acesse o diretório “/backend” e execute o comando “adonis seed --files GenerateObjectCarographSeeder.js”.

Nas primeiras linhas desse arquivo, algumas variáveis, como “yearStart” e “yearEnd”, são inicializadas de forma manual, tais valores podem ser alterados a fim de testar as mais diversas situações.

Se tudo ocorrer conforme o esperado, um arquivo chamado “CarographSeederOutput.json” será criado no diretório “backend/database/seeds/JSONSeedersFiles”.

- O conteúdo desse arquivo deve ser formatado, utilizando indentação JSON, copiado e passado como corpo da requisição post para a rota “base_url/carograph/seedCarograph”.

Se nenhuma exceção for gerada e o status code 200 for recebido, o sistema então foi corretamente povoado com valores de acordo aos cursos, status e classes do sistema.

No caso do recurso de povoamento inicial estar funcionando de forma integrada ao frontend, uma requisição get para a rota “base_url/carograph/seedCarographObject/yearStart/:yearStart/yearEnd/:yearEnd” deve ser feita, e a mesma terá um objeto devidamente estruturado como resposta. O frontend deve manipular esse objeto e preenchê-lo de forma adequada. Feito isso, uma nova requisição, dessa vez post, para a rota “base_url/carograph/seedCarograph” deve ser feita, passando o

objeto já preenchido no corpo.

```
objeto de povoamento inicial → organiza as classes pelo ano que entraram

data[ //array de cursos
{
  course_id: 1,
  students: [*todos estudantes do curso, em todos os anos],
  years: [
    *ex: 2018-2020 = [{2018}, {2019}, {2020}]
    {
      year: 2018,
      *apenas 1º, exceção quando firstYear = year (nesse caso 2018), series: [{1, 2, 3...}]
      series: [
        {
          series: 1,
          statusStudents: [
            //statusYear → year - (year + duracao - serie) ou year - yearEnd
            {
              statusYear: 2018,
              status: [
                //array de matrícula aprovados, reprovados, jubilados ...
                {
                  status_id: 1,
                  enrollments: ["201815TII0335"] // array que relaciona a matricula dos estudantes com
                                                    //o id de seus respectivos status no ano
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

Figura 1 - Visão geral da estrutura do objeto de povoamento inicial

1.2.2. SCRIPT DE VIRADA DE ANO

O script de virada de ano se encontra no diretório “backend/database/seeds/GenerateObjectNewYearSeeder.js”. Siga os passos abaixo para executá-lo corretamente.

- Faça uma requisição get para a rota “base_url/carograph/newYearObject”.

Uma resposta contendo um objeto JSON será devolvida. Esse objeto deve ser copiado e colado no arquivo “backend/database/seeds/JSONSeedersFiles/NewYearInput.json”.

- No terminal, acesse o diretório “/backend” e execute o comando “adonis seed --files GenerateObjectNewYearSeeder.js”.

Se tudo ocorrer conforme o esperado, um arquivo chamado “NewYearOutput.json” será criado no diretório “backend/database/seeds/JSONSeedersFiles”.

- O conteúdo desse arquivo deve ser formatado, utilizando indentação JSON, copiado e passado como corpo da requisição post para a rota “base_url/carograph/newYear”.

Se nenhuma exceção for gerada e o status code 200 for recebido, o sistema então foi corretamente atualizado para o ano seguinte.

No caso do recurso de virada de ano estar funcionando de forma integrada ao frontend, uma requisição get para a rota “base_url/carograph/newYearObject” deve ser feita, e a mesma terá um objeto devidamente estruturado como resposta. O frontend deve manipular esse objeto e preenchê-lo de forma adequada. Feito isso, uma nova requisição, dessa vez post, para a rota “base_url/carograph/newYear” deve ser feita, passando o objeto já preenchido no corpo.

```
objeto de virada de ano
{
  year: "2022" //ano que se encerra; sempre considera o último que possui registro, caso não haja nenhum registro
  //considera o ano atual
  data[
    //array de todos cursos do sistema
    {
      course_id: 1,

      currentClassStatus: [{ // array de séries do curso

        series: 1
        students: [], // array de estudantes da série
        statusStudents: [{ // array que relaciona o id dos estudantes com o id de seus respectivos status no ano
          status_id: 1,
          students_id: [211, 223, 224]
        }]
      }],

      nextClassStudents: [] // array vazio, será preenchido pelo frontend; são os alunos que vão
        //entrar na primeira série do curso, com o status de cursando
    }
  ]
}
```

Figura 2 - Visão geral da estrutura do objeto de virada de ano

1.3. CONSIDERAÇÕES FINAIS

Espera-se que o objetivo desta seção do documento, de esclarecer e explicitar as principais funcionalidades presentes no backend do sistema, tenha sido cumprido. No mais, o restante dos recursos é simples e extremamente intuitivo. No caso de dúvidas não contempladas, sugere-se analisar as rotas configuradas no Insomnia, isso pode fornecer uma visão geral do projeto. Lembre-se, portanto, de importar os arquivos do diretório “Insomnia Settings”. Também é recomendado consultar a documentação oficial do Adonis, framework backend que está sendo implementado no Coruja. Além disso, considere o termo “base_url”, presente nas descrições das rotas do sistema, como sendo o trecho da url comum a todas as rotas, como por exemplo “http://127.0.0.1:3333”. Ou seja,

“base_url/carograph” e “http://127.0.0.1:3333/carograph” apontam para a mesma rota.