

Laboratório de Desenvolvimento Web

Prof. Ricardo Maroquio
Enunciado do Primeiro Trabalho

Prazo de entrega: até 31/10/2022 (segunda-feira)

Forma de entrega: repositório do GitHub

Regime de trabalho: individual ou em dupla

Resumo: este primeiro trabalho consiste na criação de um backend para uma aplicação web para academias de musculação usando Node.js, Express.js e Mongoose (MongoDB). Deve ser criada uma API conforme padrão estabelecido no roteiro atualizado da aula de 18 de outubro de 2022, disponível em nosso grupo do Telegram.

Descrição do problema: uma academia de musculação deseja informatizar as fichas de exercícios dos alunos. Uma ficha é composta por um conjunto de exercícios ordenados, sendo que cada exercício tem um tipo, uma carga (peso), um número de repetições e um número de séries. Por exemplo, um dos exercícios da ficha pode ser do tipo “supino”, com 30kg de peso e 3 séries de 10 repetições. Cada tipo de exercício tem um nome e trabalha um determinado grupo muscular. Cada tipo possui também um peso máximo e mínimo de execução e um degrau de ajuste de peso (de 1 em 1 quilo, ou de 5 em 5 quilos, por exemplo, dependendo do equipamento). O tipo de exercício “supino”, por exemplo, pode ser executado com um peso entre 10kg e 200kg, variando de 5 em 5 quilos, e trabalha o grupo muscular “peitoral”. A ficha de um aluno sempre é criada ou alterada por um instrutor, sendo que ela tem uma data de início da vigência, um conjunto de exercícios a serem executados e um conjunto de registro das execuções. Cada execução tem uma data e hora de início e de fim e, opcionalmente, gera algumas anotações. É comum os alunos seguirem fichas diferentes em dias diferentes, portanto, alunos podem ter mais de uma ficha ativa.

O que deve ser feito: você deve criar, para cada coleção a seguir, uma API para realização de CRUD seguindo os mesmos padrões do roteiro apresentado em aula. Para fins de padronização, as rotas dessa API devem ser:

/api/alunos
/api/instrutores
/api/gruposmusculares
/api/tiposexercicios
/api/fichas

Esquemas das coleções:

1. Arquivo /models/GrupoMuscular.js

```
import mongoose from "mongoose";

const { Schema } = mongoose;

const GrupoMuscularSchema = new Schema({
  nome: {
    type: String,
    required: true,
    trim: true,
  },
});

export default mongoose.model("GrupoMuscular", GrupoMuscularSchema);
```

2. Arquivo /models/TipoExercicio.js

```
import mongoose from "mongoose";
import GrupoMuscular from "../GrupoMuscular.js";

const { Schema } = mongoose;

const TipoExercicioSchema = new Schema({
  nome: {
    type: String,
    required: true,
    trim: true,
  },
  pesoMinimo: {
    type: Number,
    min: 0,
    max: 100,
    required: true,
  },
  pesoMaximo: {
    type: Number,
    min: 0,
    max: 200,
    required: true,
  },
  degrauPeso: {
    type: Number,
    min: 0,
    max: 25,
    required: true,
  },
  grupoMuscular: GrupoMuscular.schema,
});

export default mongoose.model("Exercicio", TipoExercicioSchema);
```

3. Arquivo /models/Exercicio.js (não requer API)

```
import mongoose from "mongoose";

const { Schema } = mongoose;

const ExercicioSchema = new Schema({
  _id: false,
  tipo: TipoExercicio.schema,
  ordem: {
    type: Number,
    required: true,
    min: 1,
    max: 100,
  },
  peso: {
    type: Number,
    min: 0,
    max: 100,
    required: true,
  },
  repeticoes: {
    type: Number,
    min: 1,
    max: 100,
    required: true,
  },
  series: {
    type: Number,
    min: 1,
    max: 100,
    required: true,
  },
});

export default mongoose.model("Exercicio", ExercicioSchema);
```

4. Arquivo /models/Instrutor.js

```
import mongoose from "mongoose";

const { Schema } = mongoose;

const InstrutorSchema = new Schema(
  {
    nome: {
      type: String,
      required: true,
      trim: true,
    },
    dataNascimento: {
      type: Date,
      required: true,
    },
    sexo: {
      type: String,
      required: true,
      trim: true,
      enum: ["M", "F"],
    },
    email: {
      type: String,
      required: true,
      trim: true,
      unique: true,
      index: true,
    },
    senha: {
      type: String,
      required: true,
    },
    ativo: {
      type: Boolean,
      default: false,
    },
  },
  { timestamps: true }
);

export default mongoose.model("Instrutor", InstrutorSchema);
```

5. Arquivo /models/Ficha.js

```
import mongoose from "mongoose";
import Exercicio from "../Exercicio.js";
import TipoExercicio from "../TipoExercicio.js";

const { Schema } = mongoose;

const FichaSchema = new Schema(
  {
    aluno: { type: Schema.Types.ObjectId, ref: "Aluno" },
    instrutor: { type: Schema.Types.ObjectId, ref: "Instrutor" },
    ativa: {
      type: Boolean,
      required: true,
    },
    dataInicio: {
      type: Date,
      required: true,
    },
    exercicios: [
      Exercicio.schema,
    ],
    execucoes: [
      {
        dataHoraInicio: {
          type: Date,
          required: true,
        },
        dataHoraFim: {
          type: Date,
          required: true,
        },
        observacoes: {
          type: String,
          trim: true,
        },
      },
    ],
  },
  { timestamps: true }
);

export default mongoose.model("Ficha", FichaSchema);
```

6. Arquivo /models/Aluno.js

```
import mongoose from "mongoose";

const { Schema } = mongoose;

const AlunoSchema = new Schema(
  {
    nome: {
      type: String,
      required: true,
      trim: true,
    },
    dataNascimento: {
      type: Date,
      required: true,
    },
    sexo: {
      type: String,
      required: true,
      trim: true,
      enum: ["M", "F"],
    },
    email: {
      type: String,
      required: true,
      trim: true,
      unique: true,
      index: true,
    },
    senha: {
      type: String,
      required: true,
    },
    ativo: {
      type: Boolean,
      default: false,
    },
    fichas: [
      {
        type: Schema.Types.ObjectId,
        ref: "Ficha",
      },
    ],
  },
  { timestamps: true }
);

export default mongoose.model("Aluno", AlunoSchema);
```

Bom trabalho!