

## **Avaliações - Blue**

# Módulo 4 - Nest + ReactJS

## Codelabs (por semana)

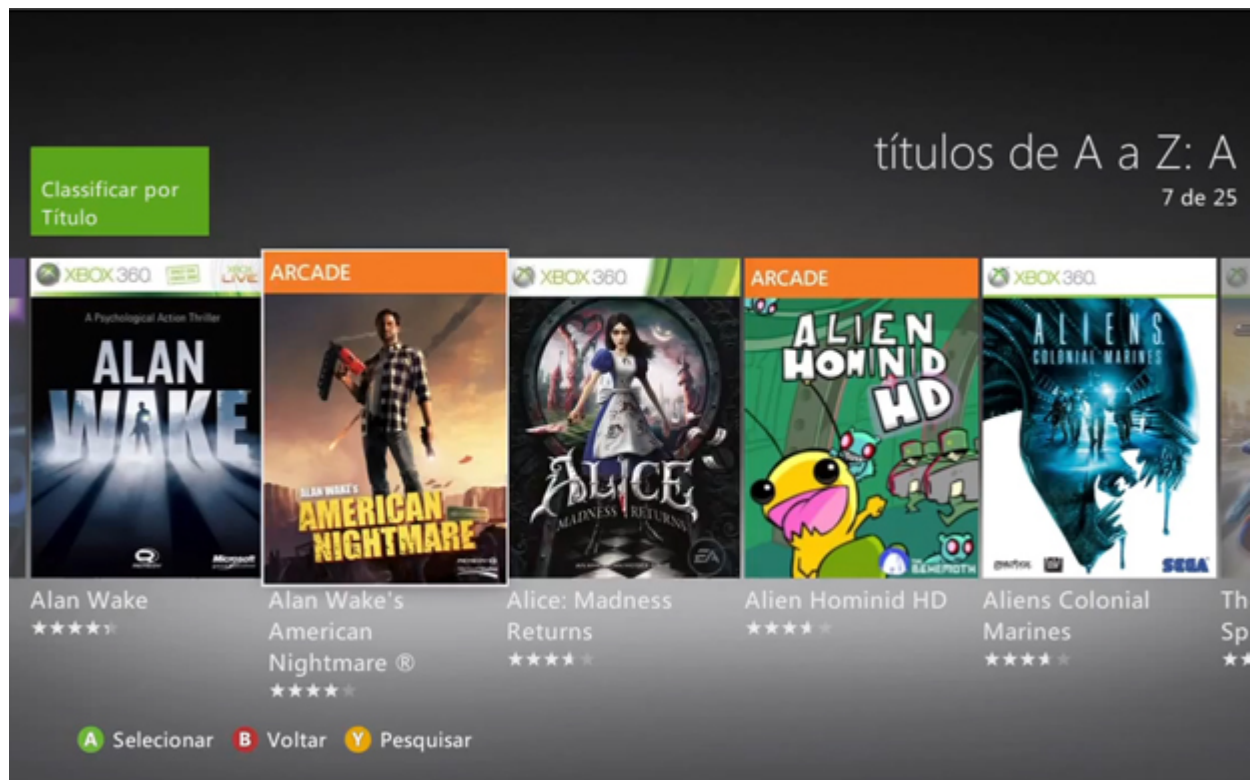
1. **Backend - NestJS** - TypeScript - CRUD + SQL com Prisma
  2. **Frontend - ReactJS** - JavaScript (TypeScript é opcional) - CRUD integrado com Backend
  3. **Backend + Frontend** - NestJS + ReactJS - TypeScript + JavaScript - Relacionando entidades
  4. **Backend + Frontend: Autenticação**
    - Framework Frontend: ReactJS
    - Framework Backend: NestJS
    - Linguagem de Programação: TypeScript
- 

## Projeto

O projeto será uma cópia da Xbox Live, plataforma da Microsoft para o console de videogame Xbox.

O objetivo é que usuários possam se cadastrar, selecionar diferentes perfis e comprar/executar jogos.

Cada jogo exibirá, além das informações básicas, informações de gêneros e prêmios.



Referência de Xbox live, no console

## Projeto 1 - Backend

Criar uma API backend com usuários, perfis, jogos, gêneros e prêmios.

**Nota máxima do projeto:** 10

### Estrutura da Entidade: Usuários

- Nome
- Sobrenome
- E-mail
- Senha
- CPF

### Estrutura da Entidade: Perfis

- Título
- Imagem

## Estrutura da Entidade: Jogos

- Título
- Capa (URL da Imagem)
- Descrição
- Ano
- Nota do IMDB (0 a 10)
- Link do Trailer do YouTube
- Link de uma Gameplay (no commentary)

## Estrutura da Entidade: Gêneros

- Nome

## Relações

- Um usuário x muitos perfis
- Um jogo x muitos gêneros
- Muitos usuários x muitos jogos (jogos favoritos)

## Endpoints (5,5 pontos)

- [CRUD] Usuários (1,5 ponto)
- [CRUD] Perfis (1 ponto)
- [CRUD] Jogos (1 ponto)
- [CRUD] Gêneros (1 ponto)
- [GET] Homepage: lista de jogos, organizados por gênero (1 ponto)

## Requisitos (4,5 pontos)

- Validação de dados em todos os endpoints (0,5 pontos)
- Status Code corretos em todos os endpoints (0,5 pontos)
  - 200, 201, 400, 404, etc
- Persistência de Dados no SQL com Prisma (1,5 pontos)
- Formatação do código utilizando o Prettier/ESLint (0,5 pontos)
- Exportar os arquivos de requisição (0,5 pontos)
- Deploy do projeto no Heroku (0,5 pontos)

- Deploy do banco de dados na ElephantSQL (0,5 pontos)
- 

## Projeto 2 - Frontend (integrado com Backend)

Criar uma aplicação frontend web integrada com a API backend criada no projeto 1.

**Nota máxima do projeto:** 10

### Overview

- Home page contendo uma lista de todos os jogos, organizados por gênero, com jogos favoritos primeiro (em uma sessão separada)
  - Cada jogo deverá estar dentro de um card, contendo as informações básicas:
    - nome
    - ano
    - imagem
    - nota: exibindo como estrelinhas, de 0 a 5
- Página individual do jogo (ao clicar em cima de cada jogo dentro da home page)
  - Exibir todas as informações o jogo e um `<iframe>` com o trailer do YouTube.
- Página de cadastro do usuário (opção para excluir/editar)
- Página de cadastro do perfil (opção para excluir/editar)
- Capriche na UX e na UI (não precisa ser uma cópia do que é atualmente, use a criatividade e adapte ao formato Web)

### Páginas (10 pontos)

- Home page:
  - Lista de jogos (1,5 pontos)
  - Card de cada jogo (1 ponto)
  - Conexão com as páginas de criação, visualização, edição e exclusão de jogos (0,4 pontos)
- Página de visualização do jogo, exibindo todas as informações do filme (1,5 pontos)
- Página de criação do usuário (1,6 pontos)
- Página de cadastro do perfil (1,5 pontos)
- Tratamento dos erros de acordo com os status enviados pelo backend (1,5 pontos)

- Sistema de Login com usuário e senha (1 ponto)
- 

## **Composição da nota final do módulo**

- [P1] = 50%
- [P2] = 50%