

**UNifeob**

**PROJETO  
INTEGRADO**

**ESCOLA DE  
NEGÓCIOS**

**2020**



UNIFEOB

Centro Universitário da Fundação de Ensino Octávio Bastos

ESCOLA DE NEGÓCIOS

**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**PROJETO INTERDISCIPLINAR**

PET SÃO JOÃO

SÃO JOÃO DA BOA VISTA, SP  
NOVEMBRO 2020

UNIFEOB

Centro Universitário da Fundação de Ensino Octávio Bastos

ESCOLA DE NEGÓCIOS

**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**PROJETO INTERDISCIPLINAR**

**PET SÃO JOÃO**

MÓDULO 4

Linguagens e Técnicas de Programação IV - Prof. Anderson Luis Ribeiro

Tecnologia da Informação - Prof. Rodrigo Marudi de Oliveira

Interface Usuário Máquina - Prof. Mauro Glória Júnior

Gestão de Recursos Humanos - Prof. Elaine Cristina Paina

Alunos:

Lucas Rabelo Cordeiro, RA 19000553

João Pedro Melchiori Papaleo, RA 19000563

Natan Flamini Barioni, RA 19001080

Paulo Henrique Nogueira Xavier, RA 19000834

Vinicius Urias de Abreu, RA 19001554

SÃO JOÃO DA BOA VISTA, SP  
NOVEMBRO 2020

# SUMÁRIO

1 INTRODUÇÃO	5
2 PROJETO INTERDISCIPLINAR	6
2.1 Linguagens e Técnicas de Programação IV	6
2.2 Tecnologia da Informação	6
2.3 Interface Usuário Máquina	6
2.4 Gestão Recursos Humanos	6
3 CONCLUSÃO	7
4 REFERÊNCIAS	8
5 ANEXOS	9

# 1 INTRODUÇÃO

O projeto teve início com a última turma de ADS (V módulo, formados no primeiro semestre de 2020), porém nós, do IV módulo, estamos dando continuidade ao projeto. A finalidade da apresentação era tornar a ideia em que turmas anteriores fizeram, como o atual Projeto Integrado.

O objetivo é desenvolver um sistema que consiga ajudar os moradores de nossa cidade que possuem pets a encontrar os animais de estimação que estejam perdidos, ou em situações semelhantes, o que ocorre com bastante frequência em qualquer cidade. Como São João é uma cidade bem grande em comparação às outras cidades da região, torna-se necessário uma aplicação como esta, facilitando as buscas e poupando tempo e preocupação dos donos de pets.

A aplicação funciona de um jeito simples e intuitivo para os usuários, sendo composta pelas informações necessárias para a localização dos pets, constando um cadastro de animais de estimação(dados como: cor, raça, nome, localização, donos etc) com o objetivo de deixar a busca por algum animal perdido mais fácil e simples utilizando de tecnologias que permitirão o usuário tirar uma foto de algum animal perdido que viu pela rua, mostrando a localização do mesmo, ou do dono do animal realizar uma publicação na aplicação dizendo que seu animal está desaparecido.

## **2 PROJETO INTERDISCIPLINAR**

### **2.1 Linguagens e Técnicas de Programação IV**

No início, o projeto era compilado em JavaScript, a partir daí, foi decidido junto às equipes, que no desenvolver, o projeto seria migrado para TypeScript, junto com TypeORM e Clean Architecture. Durante o progresso das aulas, foram estudados os conceitos básicos de armazenagem em nuvem da AWS, as funcionalidades das linguagens de programação citadas acima e também realizados reuniões semanais para que pudéssemos apresentar e alinhar os nossos progressos.

Munidos de várias tarefas, no início o objetivo era criar uma documentação da API dentro da plataforma Swagger, sendo assim, foram criados e catalogados vários endpoints para que pudéssemos fazer com que a documentação evoluísse junto a implementação do projeto. Em seguida, o grupo aprofundou seu conhecimento nos funcionamentos das linguagens que seriam utilizadas, possibilitando assim, um avanço dentro do código fonte, que consiste em criar a persistência do Tutor e armazenamento de informações dentro do banco de dados, junto a isto foi utilizado as plataformas do Github e Zenhub para alinharmos nossos progressos e efetuar os pull requests necessários.

### **2.2 Tecnologia da Informação**

Esta matéria foi dedicada ao aprendizado de como funcionam as inteligências artificiais, aprendemos como se comportam diante de diversas situações, tais como identificação de objetos “estranhos” em determinados ambientes que não deveriam estar. Outro grande ponto foi como são criados os neurônios das redes neurais, os tipos de redes neurais existentes e como as informações conseguem ser passadas entre eles, além de como eles podem absorver informações que foram passadas, gerando um poderoso aprendizado de máquina, que permite a elas ter cada vez mais conhecimentos sobre determinadas funções a que serão designadas. Esses aprendizados são bastante utilizados no reconhecimento de imagens que por sua vez são muito utilizados em

reconhecimentos faciais, reconhecimento de terreno, objetos estranhos que acabam aparecendo na agricultura, entre outros. Podem se beneficiar dos aprendizados também qualquer loja, estabelecimento ou o que seja que seja possível falar com seu público e resolver problemas sem haver a necessidade de contato humano, como pode ser visto no caso de chatbots, que são pré-configurados com as informações mais procuradas pelo público, atendendo às necessidades e resolvendo os problemas mais rapidamente. Esses foram os principais aprendizados da matéria no módulo.

### **2.3 Interface Usuário Máquina**

Nesta matéria, aprendemos como funcionam as interfaces e como elas devem se comportar diante dos usuários, além da importância entre o que o usuário consegue ver e ter como experiência em uma aplicação. Aprendemos que para uma boa comunicação e entendimento da aplicação pela parte do usuário, nossa aplicação deve respeitar as Heurísticas de Nielsen, obtendo padrões, prevenção de erros, eficiência, estética e design minimalista, entre outros. Essas técnicas fazem toda diferença tanto na experiência do usuário ao utilizar as aplicações quanto o destaque de uma aplicação bem elaborada, bonita e de fácil usabilidade. Outro assunto bem comentado e aplicado nas nossas aulas foi a criação de protótipos utilizando algumas técnicas para ajudar a aplicação a sair do papel com um conceito praticamente pronto, uma de grande importância é o Crazy Eights, que nos permite, literalmente, desenhar a aplicação em uma folha de papel para que seja possível obter mais facilidade, agilidade e acerto nos protótipos que serão criados.

### **2.4 Gestão Recursos Humanos**

Os conteúdos abordados na matéria de Gestão de Recursos Humanos, contribuíram com o entendimento sobre como são utilizados os principais métodos de avaliação de desempenho dos funcionários, os grandes desafios presentes no setor de RH, onde em muitos dos casos são necessárias várias etapas de planejamento e estratégia, também foi visto a forma de como se conduz o treinamento e

desenvolvimento dentro da empresa, o desenvolvimento interpessoal dentro do local de trabalho e por fim, os principais conceitos de desenvolvimento organizacional, onde o foco é a busca de melhorias, se manter atualizado e conquistar inovações.

### 3 CONCLUSÃO

Concluimos que o objetivo do projeto foi desenvolver um sistema que ajudasse nossa cidade no quesito de animais de estimação perdidos, ou em situações semelhantes, facilitando a busca por algum animal com foto, localização de onde o mesmo se encontra e até mesmo citando o dono.

Para contribuir ao andamento do projeto, os estudos efetuados no início, foram o desenvolvimento de uma documentação da API dentro da plataforma Swagger, permitindo que a documentação possa evoluir no mesmo ritmo da implementação, após o desenvolvimento da documentação, foram efetuados estudos intensivos sobre TypeScript, TypeORM e Clean Architecture, juntamente com os estudos, avançamos dentro do código fonte presente no backend, assim, criamos a função de salvar o usuário “tutor” dentro do aplicativo PetSãoJoão.

Aprendemos junto a matéria Tecnologia da Informação, como funcionam as inteligências artificiais e como se comportam, vimos como são criados os neurônios das redes neurais e seus tipos. Também vimos como aprendem as inteligências artificiais e seus reconhecimentos e classificações por imagem.

Na matéria Interface Usuário concluimos como é importante a primeira impressão e experiência do usuário em uma aplicação, passando uma boa comunicação e com boas explicações para o usuário, obtendo o padrão Heurísticas de Nielsen, como prevenção de erros, eficiência, estética e design minimalista, entre outros.

Como vimos na matéria de Gestão Recursos Humanos, é sempre bom conduzir o bom treinamento e desenvolvimento dentro da empresa, tanto de funcionário para clientes e também no desenvolvimento interpessoal dentro do local de trabalho, assim tornando cada dia mais simples os grandes desafios presentes no setor e buscando sempre melhorias e atualizações para conquistar novas inovações.



## REFERÊNCIAS

Blog da Rocketseat, TypeScript: Vantagens, mitos, dicas e conceitos fundamentais,

Disponível em:

<https://blog.rocketseat.com.br/typescript-vantagens-mitos-conceitos/>,

Acesso em: 2020.

Canal no Youtube da Rocketseat, TypeScript, o início, de forma prática | Masterclass

#07, Disponível em: <https://youtu.be/0mYq5LrQN1s>, Acesso em 2020.

Canal no Youtube da Rocketseat, Node.JS: Iniciando da teoria à prática | Masterclass

#11, Disponível em: <https://youtu.be/DiXbJL3iWVs>, Acesso em 2020.

Canal no Youtube da Rocketseat, O que é API? REST e RESTful? | Mayk Brito,

Disponível em: [https://youtu.be/ghTrp1x\\_1As](https://youtu.be/ghTrp1x_1As), Acesso em 2020.

Canal no Youtube da Rocketseat, SQL no Node.js com Sequelize | Masterclass #01,

Disponível em: <https://youtu.be/Fbu7z5dXcRs>, Acesso em 2020.

Canal no Youtube do DevDojo, Playlist sobre TypeScript, Disponível em:

[https://www.youtube.com/playlist?list=PL62G310vn6nGg5OzjxE8FbYDzCs\\_UqrUs](https://www.youtube.com/playlist?list=PL62G310vn6nGg5OzjxE8FbYDzCs_UqrUs),

Acesso em: 2020.

Canal no Youtube do Fabio Akita, Entendendo GIT | (não é um tutorial!), Disponível

em: <https://youtu.be/6Czd1Yetaac>, Acesso em 2020.

Canal no Youtube do Mateus Silva, Criando uma API de autenticação JWT com TypeScript e TypeORM, Disponível em: <https://youtu.be/TjAXBLszCb0>, Acesso em: 2020.

Canal no Youtube do Otavio Lemos, Playlist sobre Clean Architecture, Disponível em:

[https://www.youtube.com/playlist?list=PLpJijBkNnEt\\_IyTngYRkXXwxqz-Ae74ub](https://www.youtube.com/playlist?list=PLpJijBkNnEt_IyTngYRkXXwxqz-Ae74ub),

Acesso em 2020.

Canal no Youtube do Rafael Leme, Gerando a documentação da API Rest com Postman, Disponível em: <https://youtu.be/Y4TcDe2NgfQ>, Acesso em 2020.

DevMedia, Introdução ao TypeScript,

Disponível em: <https://www.devmedia.com.br/introducao-ao-typescript/36729>,

Acesso em: 2020.

## ANEXOS

The image shows a screenshot of the Visual Studio Code editor with three files open in the editor pane. The leftmost file is `tutorRepository.ts`, the middle is `tutor.ts`, and the rightmost is `tutors-model.ts`. The Explorer pane on the left shows a project structure with folders like `infra`, `database`, `entities`, `repositories`, `models`, and `validators`. The code in the editor is as follows:

```

tutorRepository.ts
1 import { getRepository, Repository } from 'typeorm';
2 import { Tutor } from './tutor';
3
4 export let run = async () => {
5   const userRepository = getRepository(Tutor);
6   const tutors1 = await userRepository.find();
7   console.log(tutors1);
8   const email = "anderson@anderson.com";
9   const rg = "111111111111";
10  const cpf = "123456789";
11  const phone = "123456789";
12  const cep = "12345678";
13  const whatsapp = "";
14  const street = "rua do anderson";
15  const number = "123";
16  const area = "bairro do anderson";
17  const complement = "";
18
19  await userRepository.insert({
20    email,
21    rg,
22    cpf,
23    phone,
24    cep,
25    whatsapp,
26    street,
27    number,
28    area,
29    complement
30  });
31
32  await userRepository.bind(tutors1);
33
34 }

tutor.ts
1 import { ID } from './../core/dto';
2
3 import { Entity } from './entity';
4
5 export class Tutor implements Entity {
6   id?: ID;
7
8   constructor(
9     public name?: string,
10    public email?: string,
11    public rg?: number,
12    public cpf?: string,
13    public phone?: string,
14    public cep?: string,
15    public whatsapp?: string,
16    public street?: string,
17    public number?: string,
18    public area?: string,
19    public complement?: string,
20  ) {}
21 }

tutors-model.ts
1 import { Entity, PrimaryGeneratedColumn,
2   Column, IsString, Length, IsNumber, Length, IsString, Is
3   } from 'typeorm';
4
5 @Entity('tutors')
6 export class tutors implements Tutor {
7   @PrimaryGeneratedColumn('uuid')
8   id: number;
9
10  @Column()
11  @IsString()
12  @Length(10, 30)
13  name: string;
14
15  @Column()
16  @IsString()
17  @Length(10, 30)
18  email: string;
19  @Unique()
20  unique: true;
21
22  @Column()
23  @Min(11)
24  @Max(11)
25  @IsNumber()
26  rg: number;
27
28  @Column()
29  @IsString()
30  @Min(11)
31  @Max(11)
32  cpf: string;
33
34  @Column()
35  @IsString()
36  @Min(11)
37  @Max(11)
38  phone: string;
39
40  @Column()
41  @Min(8)
42  @Max(8)
43  cep: string;
44 }

```

Imagem 1. código do TypeScript com TypeORM e Clean Architecture feito pelo grupo.

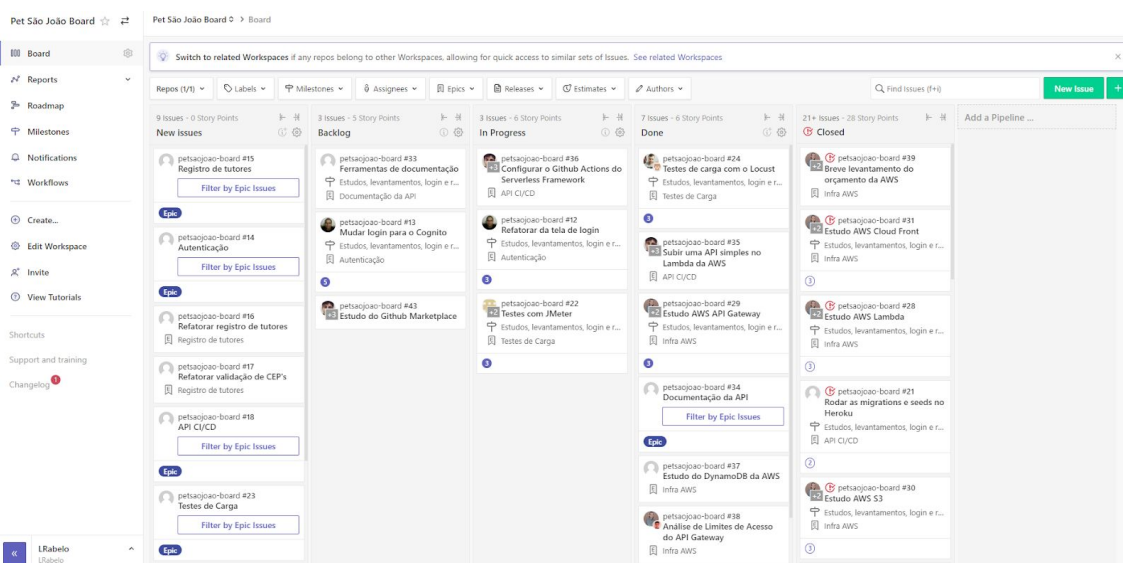


Imagem 2. repositório de ZENHub utilizado pelas equipes do projeto.

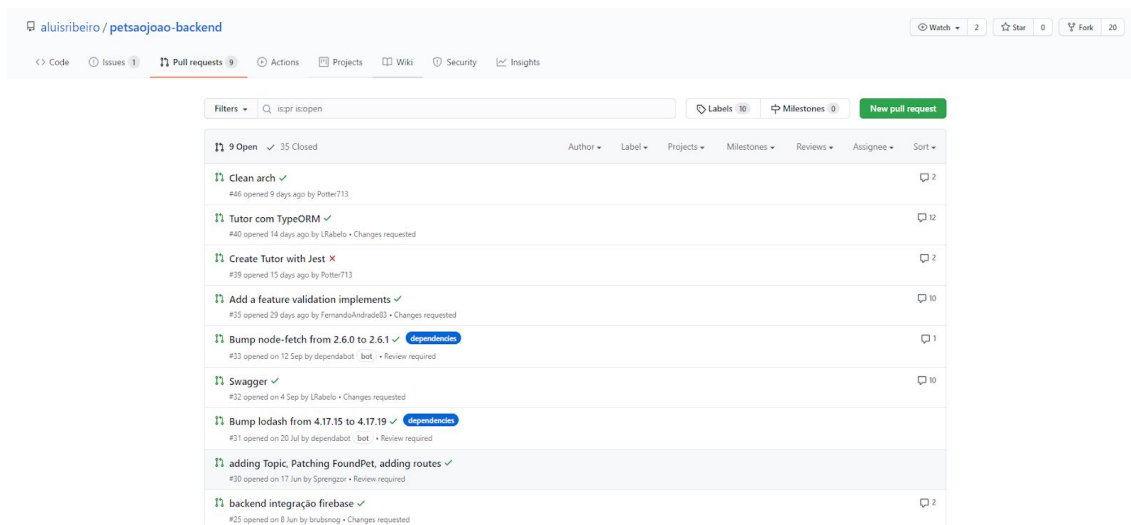


Imagem 3. repositório de GITHub utilizado pelas equipes do projeto.

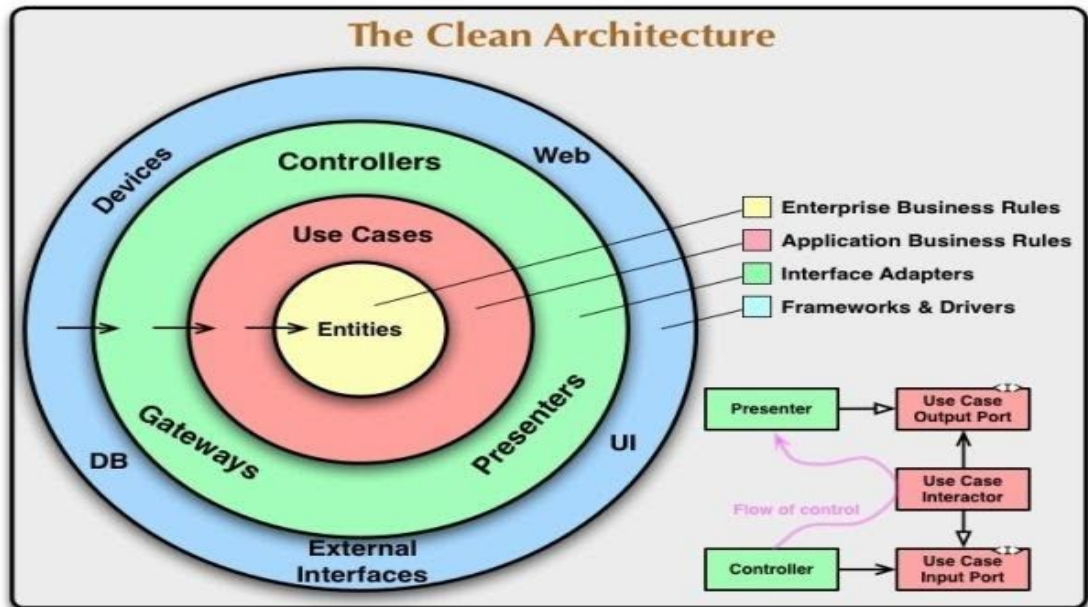


Imagem 4. exemplificação de funcionamento do Clean Architecture.