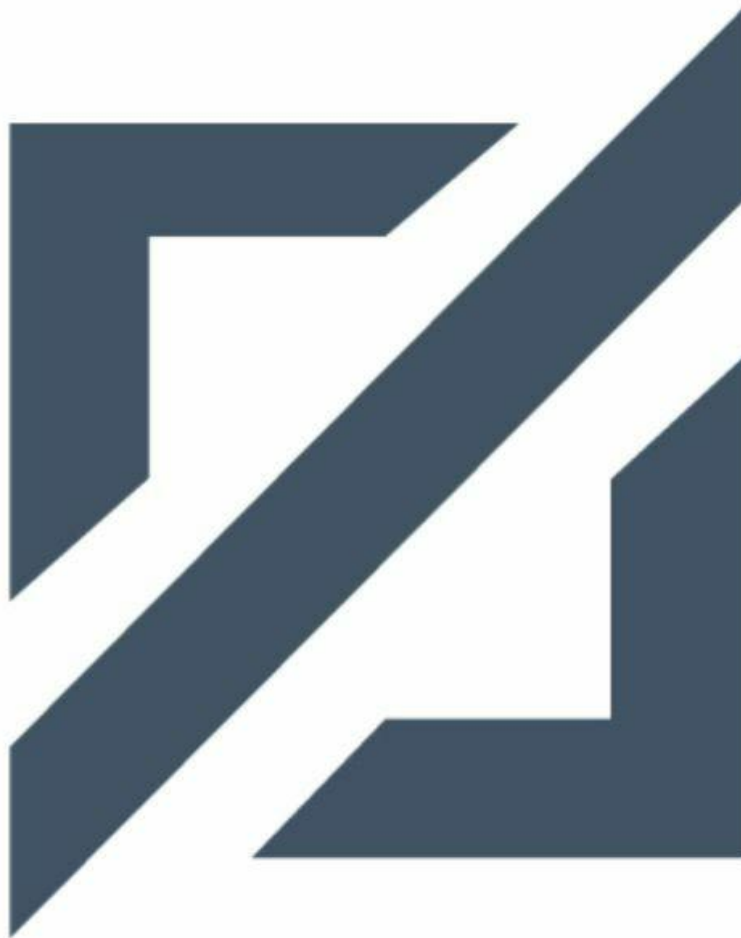


# V-WEB



Trabalho Final

## Objetivo

Avaliar o aluno e suas capacidades desenvolvidas durante o curso no desenvolvimento de aplicações web propondo um desafio.

## Regras

1. O trabalho será avaliado por inteiro valendo de 0 a 5.
2. O trabalho deve ser entregue até o prazo estabelecido.
3. É atribuído nota 0 ao trabalho entregue após o prazo de entrega estabelecido sem uma justificativa.
4. Caso o trabalho não seja enviado até o prazo, o aluno deve enviar um e-mail para [petcomp@ufc.br](mailto:petcomp@ufc.br) justificando o atraso da entrega e enviar o trabalho junto.
5. Caso exista problemas de desenvolvimento e cause atraso na entrega do trabalho, o aluno deve mandar um e-mail para [petcomp@ufc.br](mailto:petcomp@ufc.br) explicando a situação.
6. Será aceito apenas trabalhos feitos com Vue.js. Caso seu trabalho tenha sido feito com outro framework ou biblioteca para criação de interface de usuários, será atribuído automaticamente nota 0.

## Envio

O aluno deve enviar o trabalho para o e-mail [petcomp@ufc.br](mailto:petcomp@ufc.br). No e-mail deve ter seu nome completo, sua matrícula e o link do repositório no github ou um arquivo com a extensão **zip** sem a pasta **node\_modules**.

## Prazo

O trabalho vai ser aceito até o dia 16/06/2019 às 23:59.

## Descrição do Projeto

O aluno deve escolher uma API pública(**Olhar a seção links de ajuda**) e desenvolver uma aplicação utilizando os dados fornecidos por essa API. O site deve estar estilizado e obrigatoriamente utilizar flexbox layout ou grid layout, podendo ser opcionalmente responsivo.

## Apoio

Segue abaixo um algoritmo de início do projeto para auxiliar na criação da ideia

1. Escolha uma API pública que lhe pareça fornecer dados interessante.
2. Desenvolva uma ideia que utilize algum dado da API.
3. Desenvolva uma identidade visual para o projeto. Paleta de cores e Fontes.
4. Desenhe as telas do projeto. Seja em um papel, illustrator ou gravit.io.
5. Inicie o projeto criando a pasta, iniciando o projeto e instalando o vuejs.
6. Baseado nas telas, comece a criar seus componentes e páginas.

## Links de ajuda

Caso você não saiba onde encontrar uma API pública, no github possui um repositório com uma [lista enorme de API's](#). Segue abaixo uma lista com três API's utilizadas em aula:

1. [Link 1](#).
2. [Link 2](#).
3. [Link 3](#).

Caso você tenha dúvida sobre alguma ferramenta da tecnologia, abaixo segue links que explicam melhor cada tecnologia.

1. [Vue.js](#).
2. [Flexbox Layout](#).
3. [Grid Layout](#).
4. [Fetch API](#).

Um bom programa para desenhar as telas, caso você deseje, é o [gravit](#).