

---

mestrado em engenharia de software (16/17)

---

---

— Programação Web e para a Cloud —

---

# Trabalho 3

---

## — Regras —

---

- Trabalho de Grupo (max 3 elementos)
  - Síntese da pesquisa (max 4 páginas) + Slides Powerpoint
  - Demo(s) Ilustrativo(s)
    - **nível básico** (classificação neste item até 75%): a realização / instalação e teste de um tutorial / demo existente.
    - **nível intermédio** (classificação neste item até 100%) : adaptação / extensão do nível básico a outras situações / funcionalidades.
    - **nível avançado** (bonificação neste item até 25%): a inclusão num contexto de aplicação original / desenvolvimento de aplicação original.
  - Data entrega: 17 Fev (+apresentação a combinar)
- 

## — Temas —

---

### 1. GraphQL

---

## Objetivos

1. Resumo explicando conceito, aplicação, vantagens / desvantagens
2. Comparar abordagem GraphQL com REST
3. Demo.

## Recursos (pontos de partida, exemplos)

- <http://graphql.org/>
- <http://graphql.org/learn/>
- <https://github.com/chentsulin/awesome-graphql>
- <https://github.com/graphql/graphiql>
- <http://graphql.nodaljs.com/>
- <https://developer.github.com/early-access/graphql/>

---

## 2. WebRTC : Peer to Peer : File Transfer

---

### Objetivos

1. Resumo explicando conceito, aplicação
2. Demo

### Recursos (pontos de partida, exemplos)

- <https://www.html5rocks.com/en/tutorials/webrtc/basics/>
- <https://codelabs.developers.google.com/codelabs/webrtc-web/>
- <https://github.com/webrtc>
- <http://peerjs.com/>
- <https://pusher.com/>
- <https://github.com/priologic/easyrtc>
- <http://socket.io/blog/socket-io-p2p/>
- <https://github.com/kgryte/awesome-peer-to-peer>

---

## 3. WebRTC : Peer to Peer : Chat Audio / Video

---

### Objetivos

1. Resumo explicando conceito, aplicação
2. Demo

## Recursos (pontos de partida, exemplos)

- <https://github.com/webrtc>
- <https://www.html5rocks.com/en/tutorials/webrtc/basics/>
- <https://codelabs.developers.google.com/codelabs/webrtc-web/>
- <http://peerjs.com/>
- <https://pusher.com/>
- <https://github.com/priologic/easyrtc>
- <http://socket.io/blog/socket-io-p2p/>
- <https://github.com/kgryte/awesome-peer-to-peer>

---

## 4. Vue.js & vuex

---

### Objetivos

1. Resumo explicando conceito, aplicação, vantagens / desvantagens
2. Comparação React - Flux / Redux
3. Aplicação exemplo.

## Recursos (pontos de partida, exemplos)

- <https://vuejs.org/>
- <https://github.com/vuejs>
- <https://github.com/vuejs/awesome-vue>
- <http://vuex.vuejs.org/en/intro.html>

---

## 5. Reactive Programming

---

### Objetivos

1. Resumo explicando conceito, vantagens / desvantagens
2. Frameworks, Bibliotecas, Implementações, Aplicações
3. Eventual Demo

## Recursos (pontos de partida, exemplos)

- <http://reactivex.io/>
- <https://gist.github.com/staltz/868e7e9bc2a7b8c1f754>
- <https://github.com/Reactive-Extensions/RxJS>
- <https://github.com/ReactiveX/rxjs>
- <https://www.meteor.com/>
- <https://cycle.js.org/>

---

## 6. Mobx

---

## Recursos (pontos de partida, exemplos)

1. Resumo explicando conceito, vantagens / desvantagens
2. Comparação/Relação com Redux, Flux, Reflux, Reactive Programming
3. Demo

## Recursos

- <https://mobx.js.org/>

---

## 7. Google Cloud Vision API

---

## Objectivos (exemplos)

1. Extrair info de imagens usando a Cloud Vision APi
2. Aplicação node.js que consuma a API com imagens pré-existentes.
3. Aplicação node.js que permita fazer upload imagem extrair info
4. Aplicação que permita tirar foto e classificar emoções

## Recursos (pontos de partida, exemplos)

- <https://cloud.google.com/vision/>
- <https://github.com/jhuckaby/webcamjs>
- [https://github.com/amw/jpeg\\_camera](https://github.com/amw/jpeg_camera)
- <https://nodejs.org/api/https.html>

*Bom Trabalho. Pedro Miguel Moreira*