



# PokéBook - Uma rede Social Pokémon

Lucas Amin  
Thiago Bispo  
Willian Lima



# O que é o sistema?

- Rede Social em que a pessoa é um treinador Pokémon;
- Pode adicionar amigos;
- Pode postar mensagens;
- Pode capturar Pokémons completando as missões (*quests*);
- As *quests* são mini-games que envolvem reconhecimento de imagens (reconhecimento dos Pokémons nas imagens inseridas pelo usuário).

# Objetivos do Sistema

- O sistema foi criado para promover diversão, entretenimento e integração das pessoas.
- O formato de completar quests torna o website divertido para usuários.
- O fato de ser uma rede social permite a interação entre as pessoas.

# Sistema no Mundo Real

- Seria utilizado como uma rede social qualquer, entrando na url do website e utilizando-o.

# Tecnologias Empregadas

- O site foi feito utilizando Django;
- O HTML usa Bootstrap;
- O Banco de Dados implementado é o Azure da Microsoft;
- O reconhecimento de imagens é feita pela biblioteca Tensorflow;

## Conceitos:

- Machine Learning → Deep Learning
- Aplicações Web
- Banco de Dados em Nuvem

# Django

- É um framework Python Web de alto nível.
- Permite desenvolvimento rápido e limpo.
- É seguro, grátis e *open source*.
- Possui design pragmático.



# django

# Como Funciona o Django



# Bootstrap

## Examples

Wrap any text and an optional dismiss button in `.alert` and one of the four contextual classes (e.g., `.alert-success`) for basic alert messages.

### No default class

Alerts don't have default classes, only base and modifier classes. A default gray alert doesn't make too much sense, so you're required to specify a type via contextual class. Choose from success, info, warning, or danger.

#### EXAMPLE

**Well done!** You successfully read this important alert message.

**Heads up!** This alert needs your attention, but it's not super important.

**Warning!** Better check yourself, you're not looking too good.

**Oh snap!** Change a few things up and try submitting again.

```
<div class="alert alert-success" role="alert">...</div>
<div class="alert alert-info" role="alert">...</div>
<div class="alert alert-warning" role="alert">...</div>
<div class="alert alert-danger" role="alert">...</div>
```

## Single button dropdowns

Turn a button into a dropdown toggle with some basic markup changes.

#### EXAMPLE



```
<!-- Single button -->
<div class="btn-group">
  <button type="button" class="btn btn-default dropdown-toggle" data-toggle="dropdown" aria-haspopup="true"
    aria-expanded="false">
    Action <span class="caret"></span>
  </button>
  <ul class="dropdown-menu">
    <li><a href="#">Action</a></li>
    <li><a href="#">Another action</a></li>
    <li><a href="#">Something else here</a></li>
    <li role="separator" class="divider"></li>
    <li><a href="#">Separated link</a></li>
  </ul>
</div>
```



# Microsoft Azure

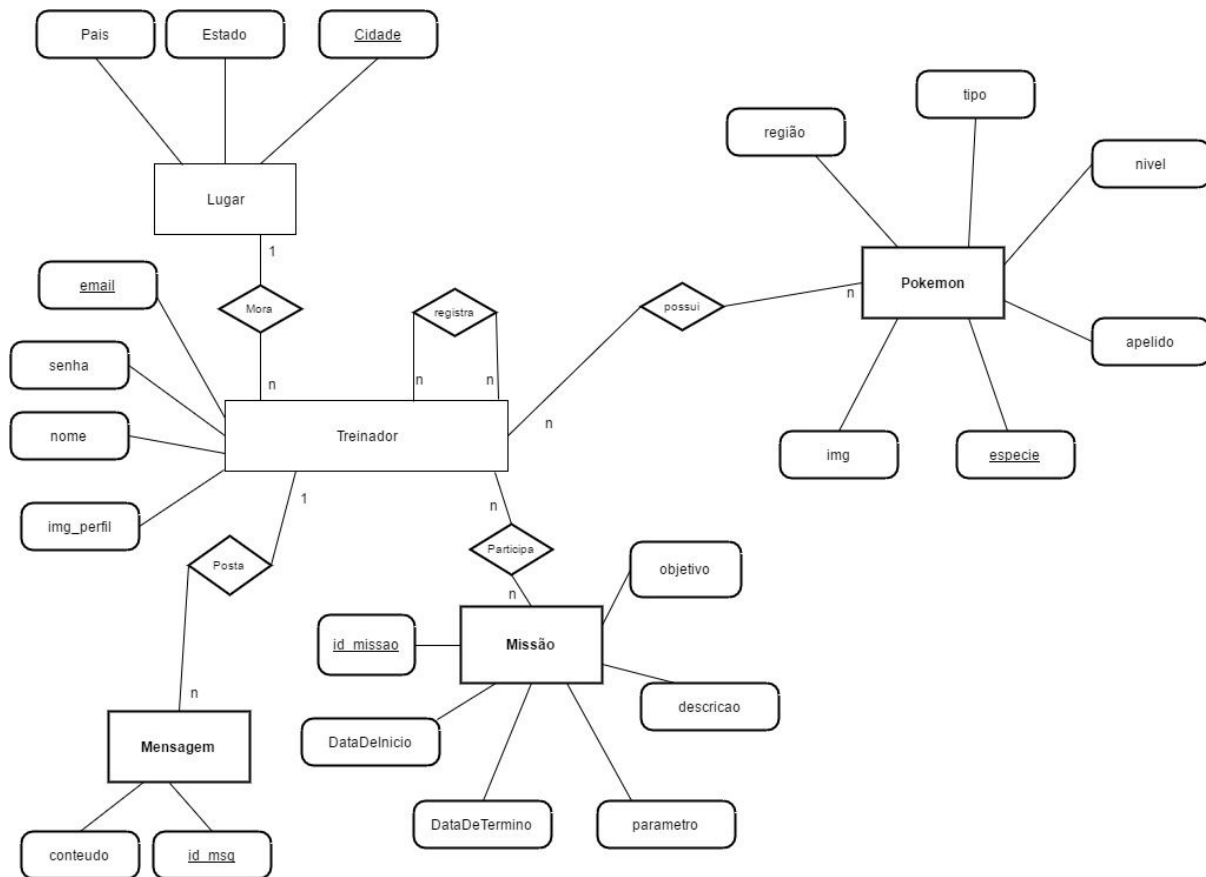


The screenshot displays the Microsoft Azure portal interface for configuring an App Service plan. The left sidebar shows the navigation menu with categories like "New", "Resource groups", "All resources", and "Recent". The main content area is divided into three panes:

- Essentials:** Shows the "Monitoring" tab with a "Requests and errors" section. A message states "No available data." Below this is a button to "Add a group".
- Settings:** Contains a search bar and several sections: "SUPPORT & TROUBLESHOOTING" (Check health, Troubleshoot, New support request), "GENERAL" (Quick start, Properties, Application settings), "APP SERVICE PLAN" (App Service Plan, Scale Up, Scale Out), and "FEATURES" (Backups, Authentication / Authorization, Diagnostics logs).
- Scale setting:** Shows the "Scale by" dropdown set to "CPU Percentage". The "Instances" section shows a current value of 1 and a target range of 1 to 10. A green circle with the number 3 highlights the "Instances" value.

The "Scale Out (App Service Plan)" option in the Settings pane is highlighted with a green circle and the number 2. The "Scale by" dropdown in the Scale setting pane is highlighted with a green circle and the number 1.

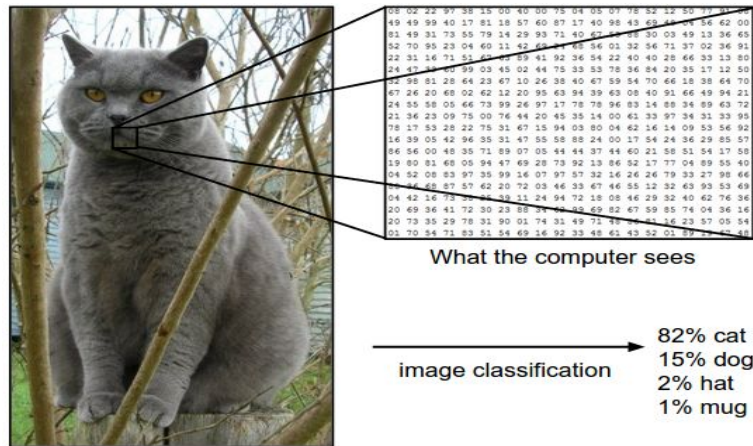
# Modelo Relacional



# Reconhecimento de imagem

Utilização de recursos de deep learning para reconhecimento e classificação de imagens

- Input
- Treinamento
- Grafo



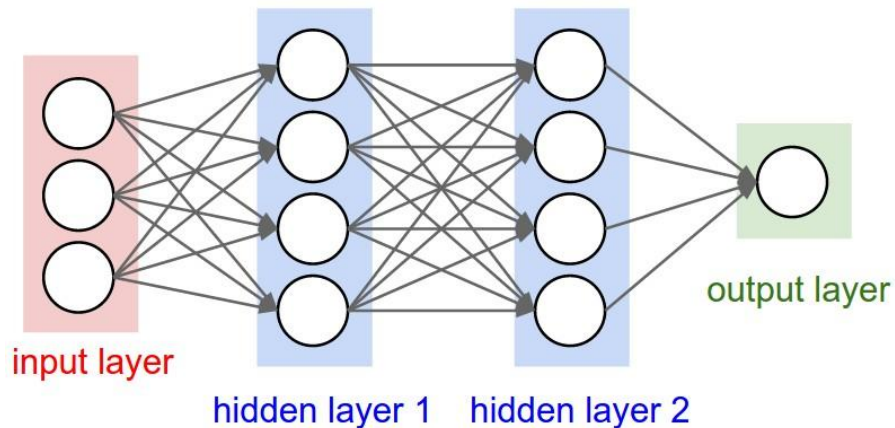
# Deep learning

Utilização de redes neurais artificiais a fim de “ensinar” um algoritmo a identificar padrões.

Utilização de grafos computacionais

Os nós representam operações matemáticas

Os vértices representam o fluxo dos dados



# Tensorflow

O Tensorflow é uma biblioteca open-source desenvolvida pela Google voltada para deep learning.



# Resultados

Em média 100 imagens de input para cada um dos pokémons identificados, são eles:

Kabutops  
Sandshrew  
Chansey  
Bulbasaur  
Charmander

Squirtle  
Pikachu  
Lapras  
Ghastly  
Paras

Ghastly



Ghastly (score = 0.9981)  
Pikachu (score = 0.0012)  
Charmander (score = 0.0006)  
Sandshrew (score = 0.0001)  
Chansey (score = 0.0000)  
Kabutops (score = 0.0000)  
Squirtle (score = 0.0000)  
Lapras (score = 0.0000)  
Bulbasaur (score = 0.0000)  
Paras (score = 0.0000)



Paras



Paras (score = 0.9627)  
Charmander (score = 0.0253)  
Pikachu (score = 0.0095)  
Chansey (score = 0.0011)  
Squirtle (score = 0.0009)  
Bulbasaur (score = 0.0002)  
Sandshrew (score = 0.0002)  
Ghastly (score = 0.0001)  
Kabutops (score = 0.0000)  
Lapras (score = 0.0000)



Pikachu



Pikachu (score = 0.9850)  
Charmander (score = 0.0083)  
Bulbasaur (score = 0.0021)  
Sandshrew (score = 0.0013)  
Kabutops (score = 0.0011)  
Paras (score = 0.0009)  
Chansey (score = 0.0009)  
Squirtle (score = 0.0003)  
Lapras (score = 0.0001)  
Ghastly (score = 0.0000)



# Confiabilidade

O resultado obtido após testar diversos parâmetros, após 2 horas de treinamento, foi obtido uma taxa de acertos de 87%.

Charmander  
(disfarçado)



Charmander (score = 0.6868)  
Sandshrew (score = 0.2207)  
Squirtle (score = 0.0364)  
Chansey (score = 0.0235)  
Lapras (score = 0.0160)  
Pikachu (score = 0.0091)  
Kabutops (score = 0.0035)  
Bulbasaur (score = 0.0026)  
Ghastly (score = 0.0008)  
Paras (score = 0.0006)



Bulbasaur  
(disfarçado)



Charmander (score = 0.7394)  
Squirtle (score = 0.1736)  
Bulbasaur (score = 0.0183)  
Ghastly (score = 0.0170)  
Pikachu (score = 0.0161)  
Sandshrew (score = 0.0158)  
Chansey (score = 0.0102)  
Lapras (score = 0.0054)  
Kabutops (score = 0.0037)  
Paras (score = 0.0005)



Squirtle  
(disfarçado)



Squirtle (score = 0.9463)  
Paras (score = 0.0354)  
Chansey (score = 0.0082)  
Pikachu (score = 0.0060)  
Sandshrew (score = 0.0027)  
Charmander (score = 0.0009)  
Bulbasaur (score = 0.0003)  
Lapras (score = 0.0002)  
Kabutops (score = 0.0000)  
Ghastly (score = 0.0000)



# Desafios e Soluções

## **Desafio:**

Desenvolver um aplicativo web pelo Django.

## **Solução:**

Prática e videoaulas.

## **Desafio:**

Implementar a biblioteca Tensorflow em um website.

## **Solução:**

Utilizar python para produzir o website.

## **Desafio:**

Desenvolver páginas em HTML.

## **Solução:**

Tutoriais e Bootstrap.

## **Desafio:**

Integrar o website com o banco de dados.

## **Solução:**

Pesquisas, leituras e tutoriais do Django e da Microsoft.



# Possíveis Melhorias

- Reconhecer se o email é válido na criação de contas.
- Permitir marcação de amigos nas mensagens.
- Criar outras *quests*.
- Mostrar as *quests* que o usuário já participou.
- Ajustes para um melhor treinamento do modelo de deep learning.
- Implementação de mais pokémons.
- Outros tipos de minigames.
- Layout do site em geral.

# Fim

Obrigado pela atenção!

