

# **Inteligência Artificial**

# Licenciatura em Engenharia de Sistemas Informáticos

Identificação dos elementos do Grupo

António Gonçalves | Aluno 10851

Nelson Peixoto | Aluno 11271

André Fernandes | Aluno 3893

Resumo

O projeto proposto tem por objetivo a criação de uma rede neural para descobrir o melhor caminho de ponto a ponto, isto é normalmente designado por “Problema do Caixeiro Viajante”. Antes de implementar as soluções propostas foram analisadas diversas e possíveis formas para essa mesma resolução.

A linguagem de programação utilizada foi C#. Como auxilio a algumas das soluções apresentadas foi pelo grupo adotada a framework AForge.NET. AForge.NET é uma framework em linguagem C# desenhada para desenvolvedores no campo da Visão por Computador e Inteligência Artificial.

A linguagem de programação utilizada foi c# na ferramenta *Visual Studio*. De forma a agilizar o processo de desenvolvimento utilizamos a ferramenta Entity Framework. Esta ferramenta permitiu um nível mais alto de abstração ao lidar com dados a base de dados.

Para o desenvolvimento de pesquisa nos dados utilizamos LAMBDA ou sintaxe de método, pertencente à biblioteca LINQ.

No final do projeto foram obtidos resultados satisfatórios, tendo sido implementados serviços CRUD, de forma síncrona e assíncrona e utilizador *WEB Services* externos.

Pesquisa Informada

Gready

A\*

Redes Neurais PERCEPTRON

Noção

A rede Perceptron é das mais antigas de todas as redes neurais e a mais simples do tipo de rede direta. A rede Perceptron é constituída apenas por uma camada de entrada e uma camada de saída. A cada entrada na rede tem relacionado um peso e na saída o valor será a soma dos produtos de cada entrada pelo respetivo peso.

Redes Neurais ALGORITMO ELÁSTICO

Noção

A técnica de algoritmo elástico adota uma noção de anel elástico percorrendo todos os pontos e fechando numa rede. Este algoritmo percorre entre o ponto a trabalhar (neste caso prático, uma cidade) e os pontos vizinhos anterior e posterior mais próximos e fecha sempre a rede (neste caso, uma rota) e faz a cada iteração uma atualização das coordenadas o mais próximo possível do ponto original, ou seja, em função do maior ou menor número de iterações utilizadas ele fica mais próximo ou mais distante do ponto original.

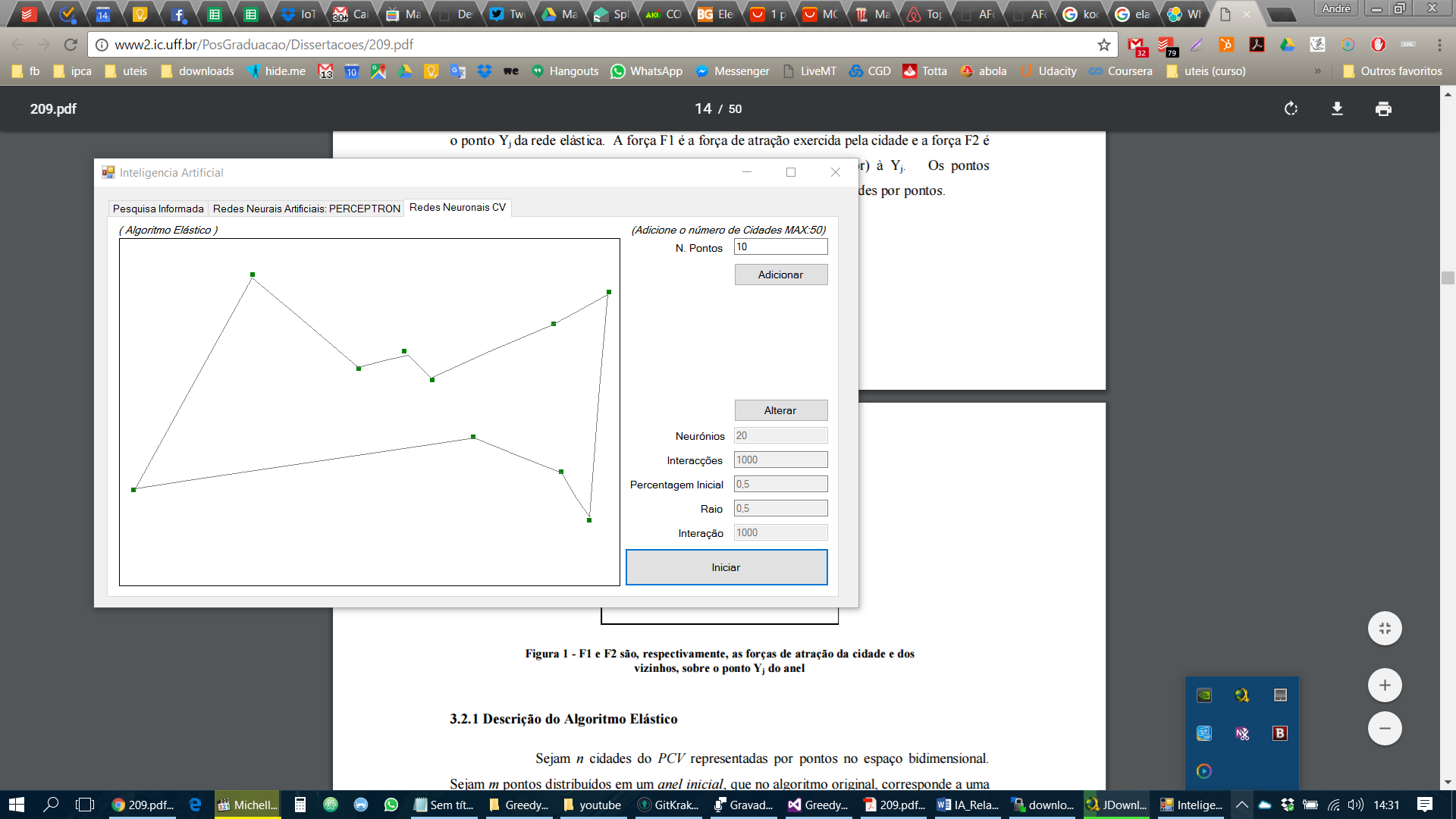


Figura - Exemplo 1 com 1000 iterações

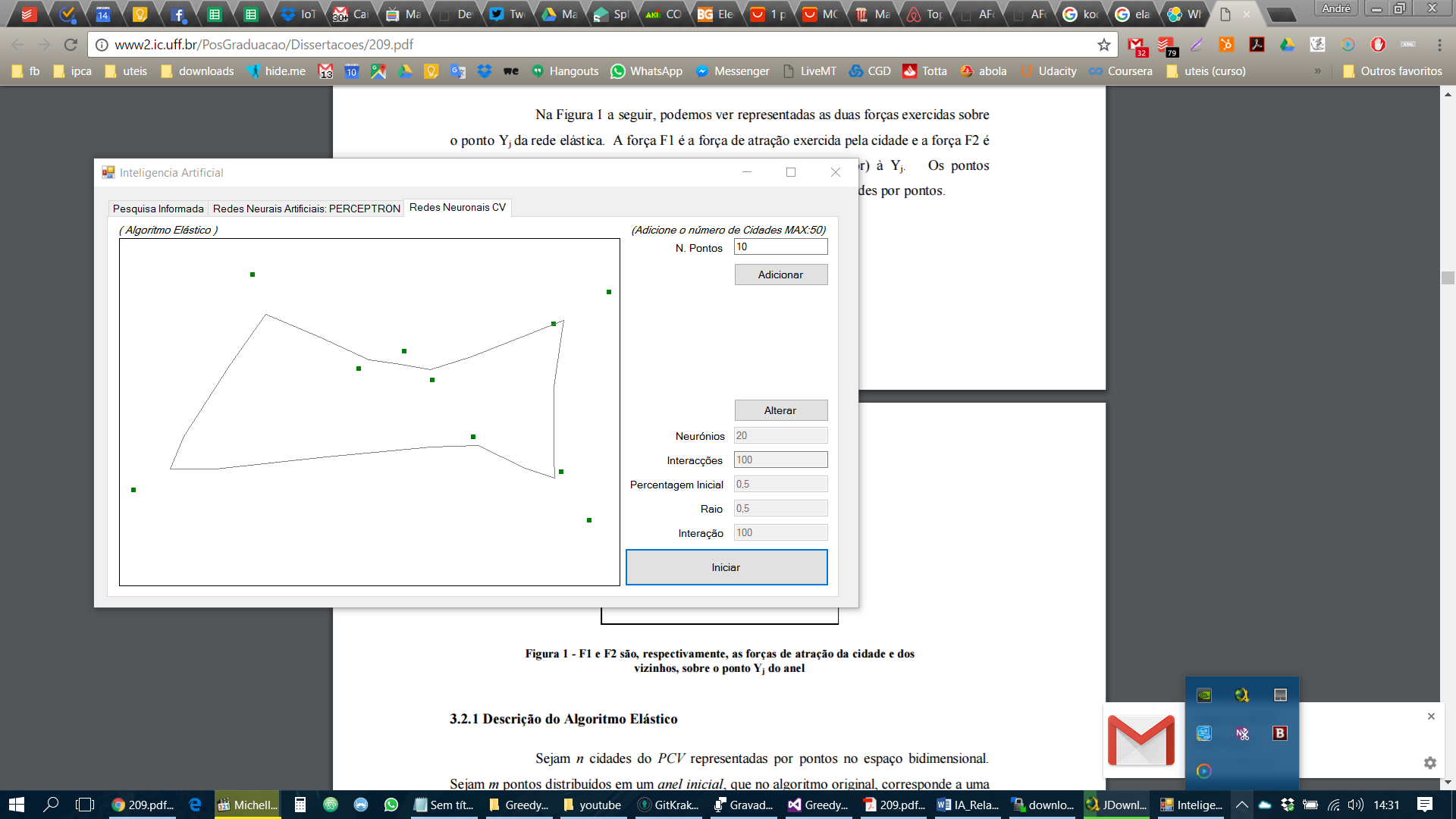


Figura - Exemplo 1 com 100 iterações