

# Um *framework* para manipulação de Grafos

Pedro Coelho  
Rubem Kalebe

Universidade Federal do Rio Grande do Norte  
Departamento de Informática e Matemática Aplicada

2 de junho de 2015

## Visão geral - 1/2

---

- O objetivo é desenvolver um *framework* que facilite a manipulação de grafos, diminuindo os custos e acelerando o processo de desenvolvimento de aplicações que façam uso dessas estruturas.
- O usuário deste *framework* deverá poder criar um grafo vazio e construí-lo a partir de primitivas de edição.
- Devem ser fornecidos mecanismos para criar vértices e para conectar dois vértices através de uma aresta.
- Além disso, deverá ser possível salvar e carregar grafos a partir de arquivos.

## Visão geral - 2/2

---

- O sistema deverá atender aos seguintes requisitos:
  - disponibilizar funções para criação e edição de grafos, como: adicionar e remover vértices e arestas, e atribuir pesos;
  - disponibilizar funções para gerar grafos;
  - permitir extrair propriedades tais como o grafo ser completo, conexo ou desconexo, dentre outras;
  - disponibilizar um subconjunto de algoritmos clássicos, como: *Dijkstra*, *Floyd-Warshall*, *Kruskal*, ordenação topológica, busca em largura e busca em profundidade;
  - permitir persistir e carregar grafos.

# Casos de Uso

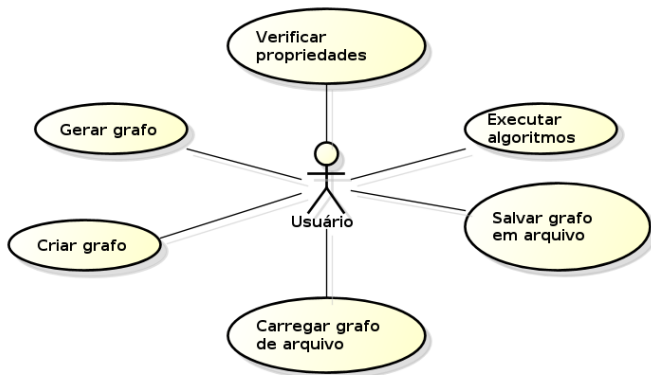


Figura: Diagrama de casos de uso do *framework*.

## Pontos fixos

---

- Criação e edição de um grafo.
- Geração de um grafo.
- Algoritmos clássicos usados no estudo da teoria dos grafos.
- Teste de propriedades.

## Pontos variáveis

---

- Formas de análise de um grafo já gerado.
- Formas de salvar e carregar o grafo.
- Execução de outros algoritmos.
- Verificação de novas propriedades.

## Exemplo 1/3

---

- Considere uma aplicação para dispositivos móveis que serve como guia turístico em uma cidade.
- Dentre as várias funcionalidades dessa aplicação temos a de indicar qual/ quais ônibus o usuário poderá pegar para se locomover de um local para outro.
- Para isso é necessário representar a cidade como um grafo.

## Exemplo 2/3

---

- Considere uma aplicação que precisa modelar o cenário de autoria conjunta de artigos acadêmicos (grafo de colaboração).
- Nesse grafo, os vértices representam pessoas e as arestas ligam duas pessoas se elas tiverem escrito um artigo em conjunto.
- Parecido com outro problema: representar atores que já trabalharam juntos em filmes (grafo de *Hollywood*).



## Exemplo 3/3

---

- Considere uma aplicação que deve analisar um código em *Assembly* e detectar quais as instruções sem dependência de dados e devolver, como saída, quais instruções podem ser executadas em paralelo.
- A dependência de instruções das instruções prévias pode ser representada por um grafo orientado.
- Cada instrução é representada por um vértice e existe uma aresta de um vértice para um segundo vértice se a instrução representada pelo segundo vértice não puder ser executada antes de a instrução representada pelo primeiro vértice ter sido executada.

# Análise de pontos fixos e variáveis dos exemplos

---

## Pontos fixos:

- Criação e manutenção do grafo;
- Execução de algoritmos clássicos.

## Pontos variáveis:

- Variação na representação de vértices e arestas;
- Formas de salvar e carregar o grafo.