



TRABALHO FINAL

Este trabalho aborda os conceitos trabalhados em sala de aula com cenários diferentes de aplicação e representa o trabalho final da disciplina. Abaixo seguem as orientações para execução e entrega deste trabalho.

- A turma será dividida em seis grupos de três e dois de quatro acadêmicos e cada grupo terá um tema sorteado para resolução do trabalho;
- **O trabalho deverá ser entregue em arquivo formato TXT ou PY até às 23hs do dia 07 de dezembro. Não serão aceitos trabalhos entregues no formato PDF;**
- O trabalho deverá ser entregue na tarefa indicada para o trabalho final no SIGAA.
- Basta que apenas um dos integrantes faça o envio do arquivo;
- O algoritmo deve ler um arquivo em formato TXT com a lista de objetos conforme cada cenário e carregar o conteúdo do arquivo em formato JSON. Caso o arquivo não exista (primeira execução do algoritmo), ele deve ser criado;
- O algoritmo deve permitir ao usuário cadastrar um novo objeto, remover um objeto existente ou visualizar os objetos já cadastrados conforme cada cenário de aplicação;
- Ao fim da execução do algoritmo, o arquivo deve ser atualizado com os objetos que constam na lista do algoritmo.

A descrição de cada cenário de aplicação segue nas próximas páginas.



CENÁRIO 1: LOCADORA DE VEÍCULOS

Este algoritmo deve simular o cadastro de veículos e aluguéis de uma Locadora de Veículos. O algoritmo deve permitir o cadastro de veículos com os seguintes atributos: modelo (texto), ano (inteiro), cor (texto) e acessórios. Os acessórios devem ser registrados como um objeto para cada veículo com os seguintes atributos do tipo booleano: ar, vidro elétrico e direção.

Os aluguéis devem ser registrados no objeto de cada veículo como uma propriedade em formato de lista. Para cada aluguel o algoritmo deve registrar as seguintes propriedades: nome do cliente (texto), valor (float) e quilometragem (inteiro). O algoritmo deve permitir ao usuário registrar quantos aluguéis por veículo ele desejar.

Abaixo segue um exemplo da estrutura de como deve ficar o objeto de cada veículo.

```
[{
  'modelo': 'modelo',
  'ano': 1233,
  'cor': 'branco',
  'acessorios': {
    'ar': 'true',
    'vidro-eletrico': 'true',
    'direcao': 'true'
  },
  'alugueis': [
    {
      'cliente': 'cliente',
      'valor': 1231.12,
      'quilometragem': 12312312
    },
    {
      'cliente': 'cliente',
      'valor': 1231.12,
      'quilometragem': 12312312
    }
  ]
}]
```



CENÁRIO 2: ALUNOS

Este algoritmo deve simular o cadastro de alunos e resultados de disciplinas. O algoritmo deve permitir o cadastro de alunos com os seguintes atributos: nome (texto), idade (inteiro), endereço (texto) e nome dos pais. Os nomes dos pais devem ser registrados como um objeto para cada aluno com os seguintes atributos do tipo texto: nome do pai e nome da mãe.

Os resultados de disciplinas devem ser registrados no objeto de cada aluno como uma propriedade em formato de lista. Para cada resultado o algoritmo deve registrar as seguintes propriedades: disciplina (texto), nota (float) e situação (aprovado/reprovado). O algoritmo deve permitir ao usuário registrar quantos resultados de disciplina por aluno ele desejar.

Abaixo segue um exemplo da estrutura de como deve ficar o objeto de cada aluno.

```
[{
  'nome': 'nome',
  'idade': 33,
  'endereço': 'endereço',
  'pais': {
    'nome-do-pai': 'nome do pai',
    'nome-da-mae': 'nome da mae'
  },
  'resultado-final': [
    {
      'disciplina': 'nome da disciplina',
      'nota': 5.12,
      'situacao': 'prova final'
    },
    {
      'disciplina': 'nome da disciplina',
      'nota': 8.12,
      'situacao': 'aprovado'
    }
  ]
}]
```



CENÁRIO 3: VENDAS DE AUTO PEÇAS

Este algoritmo deve simular o cadastro de vendas de uma auto peças. O algoritmo deve permitir o cadastro de vendas com os seguintes atributos: cliente (texto), valor total (float), método de pagamento (texto) e entrega. A entrega deve ser registrada como um objeto para cada venda com os seguintes atributos do tipo texto: endereço e descrição.

Os produtos devem ser registrados no objeto de cada venda como uma propriedade em formato de lista. Para cada produto o algoritmo deve registrar as seguintes propriedades: nome (texto), valor unitário (float) e quantidade (inteiro). O algoritmo deve permitir ao usuário registrar quantos produtos por venda ele desejar.

Abaixo segue um exemplo da estrutura de como deve ficar o objeto de cada venda.

```
[{
  'cliente': 'nome',
  'valor-total': 336.5,
  'metodo-pgto': 'cartao',
  'entrega': {
    'endereco': 'endereco',
    'descricao': 'descricao entrega'
  },
  'produtos': [
    {
      'nome-produto': 'nome do pproduto',
      'quantidade': 2,
      'valor-unitario': 10.45
    },
    {
      'nome-produto': 'nome do produto',
      'quantidade': 1,
      'valor-unitario': 10.2
    }
  ]
}]
```



CENÁRIO 4: CADASTRO CLIENTE DE LOJA

Este algoritmo deve simular o cadastro de clientes e compras de uma loja. O algoritmo deve permitir o cadastro de clientes com os seguintes atributos: nome (texto), renda (float), endereço (texto) e documentos. Os documentos devem ser registrados como um objeto para cada cliente com os seguintes atributos do tipo texto: CPF e RG.

As compras devem ser registradas no objeto de cada cliente como uma propriedade em formato de lista. Para cada compra o algoritmo deve registrar as seguintes propriedades: data (texto), valor (float) e quantidade de produtos (inteiro). O algoritmo deve permitir ao usuário registrar quantas comprar por cliente ele desejar.

Abaixo segue um exemplo da estrutura de como deve ficar o objeto de cada cliente.

```
[{
  'nome': 'nome do cliente',
  'endereço': 'endereço',
  'renda': 3000,
  'documentos': {
    'cpf': '123.123.123-12',
    'rg': '12312312-1'
  },
  'compras': [
    {
      'data': '12/12/12',
      'quantidade-produtos': 2,
      'valor': 120.45
    },
    {
      'data': '20/12/12',
      'quantidade-produtos': 3,
      'valor': 128.45
    }
  ]
}]
```



CENÁRIO 5: ESTOQUE COMPUTADORES

Este algoritmo deve simular o cadastro de computadores em um estoque. O algoritmo deve permitir o cadastro de computadores com os seguintes atributos: descrição (texto), valor (float), quantidade (inteiro) e fornecedor. O fornecedor deve ser registrado como um objeto para cada computador com os seguintes atributos do tipo texto: CNPJ e nome.

Os periféricos devem ser registrados no objeto de cada computador como uma propriedade em formato de lista. Para cada periférico o algoritmo deve registrar as seguintes propriedades: nome (texto) descrição (texto). O algoritmo deve permitir ao usuário registrar quantos periféricos por computador ele desejar.

Abaixo segue um exemplo da estrutura de como deve ficar o objeto de cada computador.

```
[{
  'descricao': 'descricao geral',
  'quantidade': 10,
  'valor': 3000,
  'fornecedor': {
    'cnpj': '01.367.770/0001-30',
    'nome': 'nome do fornecedor'
  },
  'periféricos': [
    {
      'nome': 'cpu',
      'descricao': 'descricao'
    },
    {
      'nome': 'ram',
      'descricao': 'descricao'
    }
  ]
}]
```



CENÁRIO 6: ESTOQUE DE BEBIDAS

Este algoritmo deve simular o cadastro de bebidas em um estoque. O algoritmo deve permitir o cadastro de bebidas com os seguintes atributos: nome (texto), valor (float), quantidade (inteiro) e informações nutricionais. As informações nutricionais devem ser registradas como um objeto para cada bebida com os seguintes atributos: calorias (float) e contém gluten (booleano).

Os fornecedores devem ser registrados no objeto de cada bebida como uma propriedade em formato de lista. Para cada fornecedor o algoritmo deve registrar as seguintes propriedades: nome (texto) e cnpj (texto). O algoritmo deve permitir ao usuário registrar quantos fornecedores por bebida ele desejar.

Abaixo segue um exemplo da estrutura de como deve ficar o objeto de cada bebida.

```
[{
  'nome': 'nome da bebida',
  'quantidade': 10,
  'valor': 3000,
  'informacoes-nutricionais': {
    'calorias': 700,
    'contem-gluten': 'true'
  },
  'fornecedores': [
    {
      'cnpj': '01.367.770/0001-30',
      'nome': 'nome do fornecedor'
    },
    {
      'cnpj': '01.367.770/0001-30',
      'nome': 'nome do fornecedor'
    }
  ]
}]
```



CENÁRIO 7: CADASTRO DE FILMES

Este algoritmo deve simular o cadastro de filmes e atores. O algoritmo deve permitir o cadastro de filmes com os seguintes atributos: nome (texto), nota da crítica (float), valor investido (float) e data de lançamento. A data de lançamento deve ser registrada como um objeto para cada filme com os seguintes atributos: ano (inteiro), mês (texto) e dia (inteiro).

Os atores devem ser registrados no objeto de cada filme como uma propriedade em formato de lista. Para cada ator o algoritmo deve registrar as seguintes propriedades: nome (texto) e idade (inteiro). O algoritmo deve permitir ao usuário registrar quantos atores por filme ele desejar.

Abaixo segue um exemplo da estrutura de como deve ficar o objeto de cada filme.

```
[{
  'nome': 'nome do filme',
  'nota-critica': 10,
  'valor-investido': 3000,
  'lançamento': {
    'ano': 2019,
    'mes': 'agosto',
    'dia': 23
  },
  'atores': [
    {
      'nome': 'ator 1',
      'idade': 33
    },
    {
      'nome': 'ator 2',
      'idade': 33
    }
  ]
}]
```




CENÁRIO 8: CADASTRO DE FICHA DE DIETA

Este algoritmo deve simular o cadastro de fichas de dieta de pacientes. O algoritmo deve permitir o cadastro de fichas de dieta com os seguintes atributos: paciente (texto), idade (inteiro), objetivo (texto) e medidas. As medidas devem ser registradas como um objeto para cada paciente com os seguintes atributos do tipo float: peso, torax, abdomen e cintura.

As refeições devem ser registrados no objeto de cada paciente como uma propriedade em formato de lista. Para cada refeição o algoritmo deve registrar as seguintes propriedades: descrição (texto) e hora (texto). O algoritmo deve permitir ao usuário registrar quantas refeições por paciente ele desejar.

Abaixo segue um exemplo da estrutura de como deve ficar o objeto de cada paciente.

```
[{
  'paciente': 'nome do paciente',
  'idade': 10,
  'objetivo': 'objetivo da dieta',
  'medidas': {
    'peso': 100,
    'torax': 99,
    'abdomen': 99,
    'cintura': 99
  },
  'refeicoes': [
    {
      'descricao': 'descricao da refeicao',
      'hora': '11:11'
    },
    {
      'descricao': 'descricao da refeicao',
      'hora': '11:11'
    }
  ]
}]
```