

# Laboratório de Computação Paralela Projeto AUDI

Isadora Oliveira, Luis Krupp, Samara Vilefort

29 de março de 2019

## 1 Projeto

Você foi contratado pela AUTomóveis DIVinópolis(AUDI) para desenvolver um software para controlar a fabricação de carros.

A fabricação de carros é composta por várias linhas de produção, como por exemplo: produção de pneus, produção de motores, produção de carrocerias, produção de bancos e produção de eletrônica embarcada.



Figure 1: Linha de produção

Cada linha de produção é representada por uma classe específica na sua aplicação, como por exemplo:

- LPMotores é a classe que controla a produção de Motores.
- Ao final da produção de cada carro, eles devem ser colocados na carreta, com capacidade para 10 veículos.

- A aplicação deverá evitar a espera ocupada, uma vez que cada linha de produção tem um tempo diferente de entrega dos componentes e um estoque máximo, segundo a tabela:

Item	Quantidade por Veículo	Tempo de produção por item	Estoque Máximo
Motor	1	12 segundos	10
Carroceria	1	15 segundos	20
Pneus	4	9 segundos	100
Bancos	5	6 segundos	25
Eletrônica	1	7 segundos	8

Figure 2: Tabela itens de fabricação

## 2 Desenvolvimento

### 2.1 Linguagem

Para o desenvolvimento do sistema será utilizado JAVA, é uma linguagem de programação e plataforma de computação lançada pela Sun Microsystems em 1995 [4]. Em 2009 a Oracle comprou a Sun, fortalecendo a marca. A Oracle sempre foi, junto com a IBM, uma das empresas que mais investiram e fizeram negócios através do uso da plataforma Java[1].

Como qualquer linguagem de programação, a linguagem JAVA tem sua própria estrutura, regras de sintaxe e paradigma de programação. O paradigma de programação da linguagem JAVA baseia-se no conceito de orientação a objetos [5].

Grande parte das aplicações são desenvolvidas com linguagens de programação que utilizam o paradigma de Orientação a Objetos. Este paradigma possibilita ao desenvolvedor criar sistemas encapsulando lógicas de negócio e escrevendo códigos menores e mais organizados[6].

### 2.2 IDE - *Integrated Development Environment*

Eclipse é um IDE para desenvolvimento Java, porém suporta várias outras linguagens a partir de plugins como C/C++, PHP, ColdFusion, Python, Scala e plataforma Android. Ele foi feito em Java e segue o modelo *open source* de desenvolvimento de software[2].

O projeto Eclipse foi iniciado na IBM que desenvolveu a primeira versão do produto e doou como software livre para a comunidade. O gasto inicial da IBM no produto foi de mais de 40 milhões de dólares. Hoje, o Eclipse é o IDE Java mais utilizado no mundo[3].

## 2.3 Implementação

Para cada produto que será fabricado foi criado uma classe contendo os atributos, conforme descrito na tabela do projeto e seus respectivos valores.

Exemplo:

```
1
2     private String nome = "Motor";
3     private final int tempo = 12;
4     private final int estoque = 10;
5     private final int quantidade = 1;
6
7     public String getNome() {
8         return nome;
9     }
10
11    public void setNome(String nome) {
12        this.nome = nome;
13    }
14
15    public int getTempo() {
16        return tempo;
17    }
18
19    public int getEstoque() {
20        return estoque;
21    }
22
23    public int getQuantidade() {
24        return quantidade;
25    }
```

Em outra classe usamos o método **initComponents()** para poder inicializar todos os componentes como janelas, campo de texto, imagens, botões, etc... Todos os produtos deverão ser instanciados, pois, assim conseguimos obter os dados sobre seus atributos ao longo do código. O método run

O método **initComponents()** é usada para iniciar os componentes: campos de texto, imagens, botões.. na tela e instanciando todos os produtos que serão utilizados na fabricação do carro, desta forma poderemos obter dados sobre seus atributos ao longo do código.

Toda vez que chamar a classe produtor serão passados informações sobre o estoque, tempo, nome do produto, quantidade e ponte sincronizadora, assim o sistema será atualizado com o que já foi produzido. A interface Runnable é usada para iniciar a Thread e chamar o método run. No método run é contabilizado o tempo para produzir a peça e decrementado para passar para o carro quando for construído. Ponte Sincronizadora é que realizará a montagem de todos os

veículos e irá atualizar na tela o que foi produzido de peça, a quantidade do estoque e a quantidade de carro que foi produzido. O restante do código trata das configurações do swing.

## References

- [1] Caelum. Apostila java e orientação a objetos.
- [2] Fundation Eclipse. Sobre a fundação eclipse. 2019.
- [3] Java EE. Java ee productivity report. 2012.
- [4] Oracle. What is java technology and why do i need it? 2017.
- [5] J Steven Perry. Fundamentos da linguagem java. 2016.
- [6] Erick Scudero. A trajetória de um desenvolvedor mobile: tudo que você precisa saber. 2018.