

**FIAP** GRADUAÇÃO

# **DIGITAL BUSINESS ENABLEMENT**

*Prof. THIAGO T. I. YAMAMOTO*

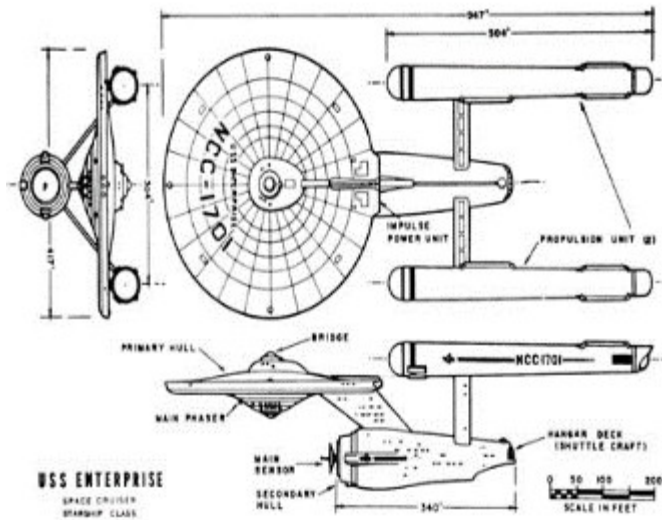
#02 - JAVA APPLICATION



- ♦ Revisão – Metodologia de Desenvolvimento de Sistemas
- ♦ Revisão - Java
- ♦ Desenvolvimento de Java Application
- ♦ Build & Deployment de Java Application

# COMO CONSTRUIMOS APLICAÇÕES?

# REVISÃO: COMO CONSTRUIMOS APLICAÇÕES FIAP



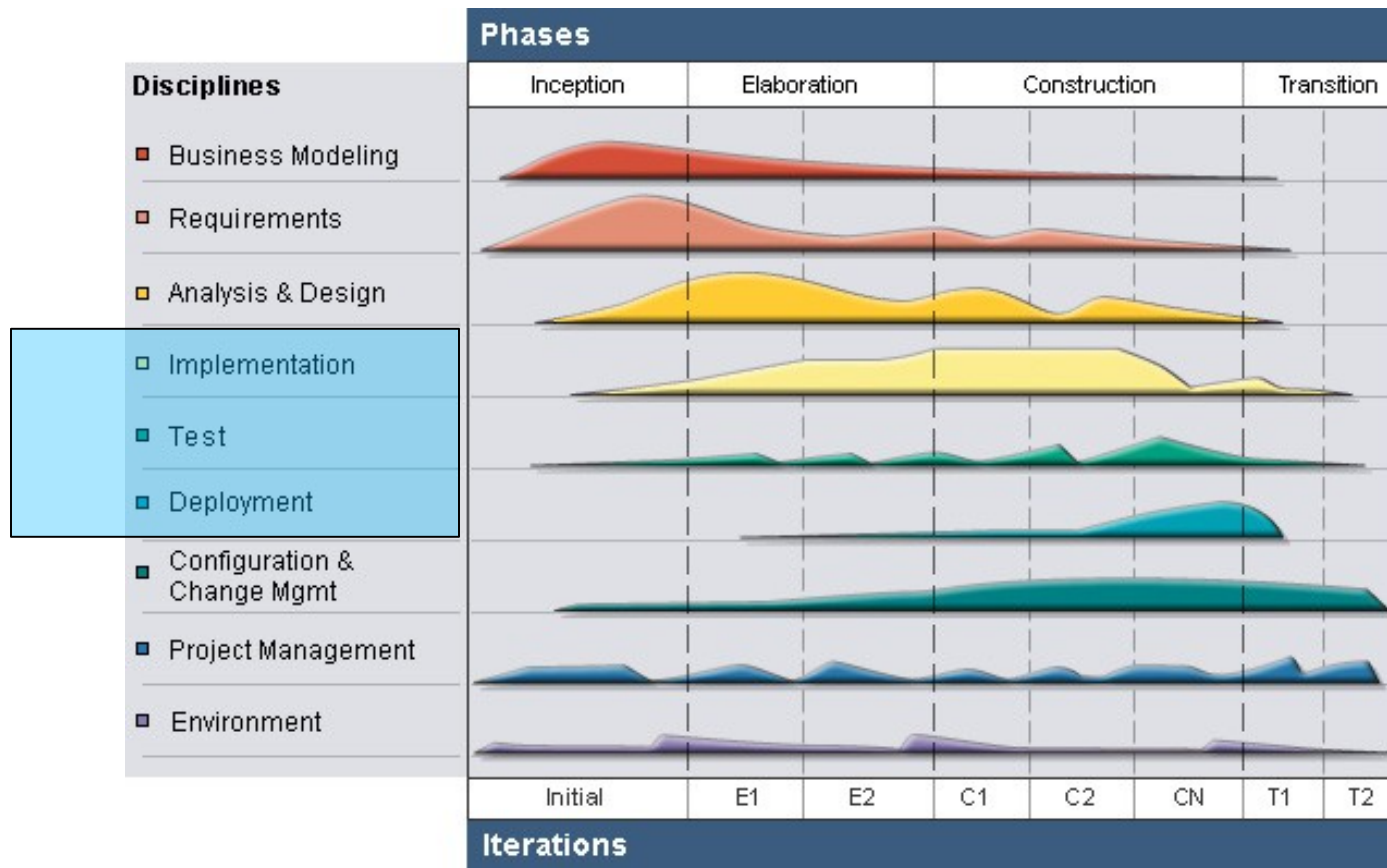
+



=



# REVISÃO: METODOLOGIA RUP

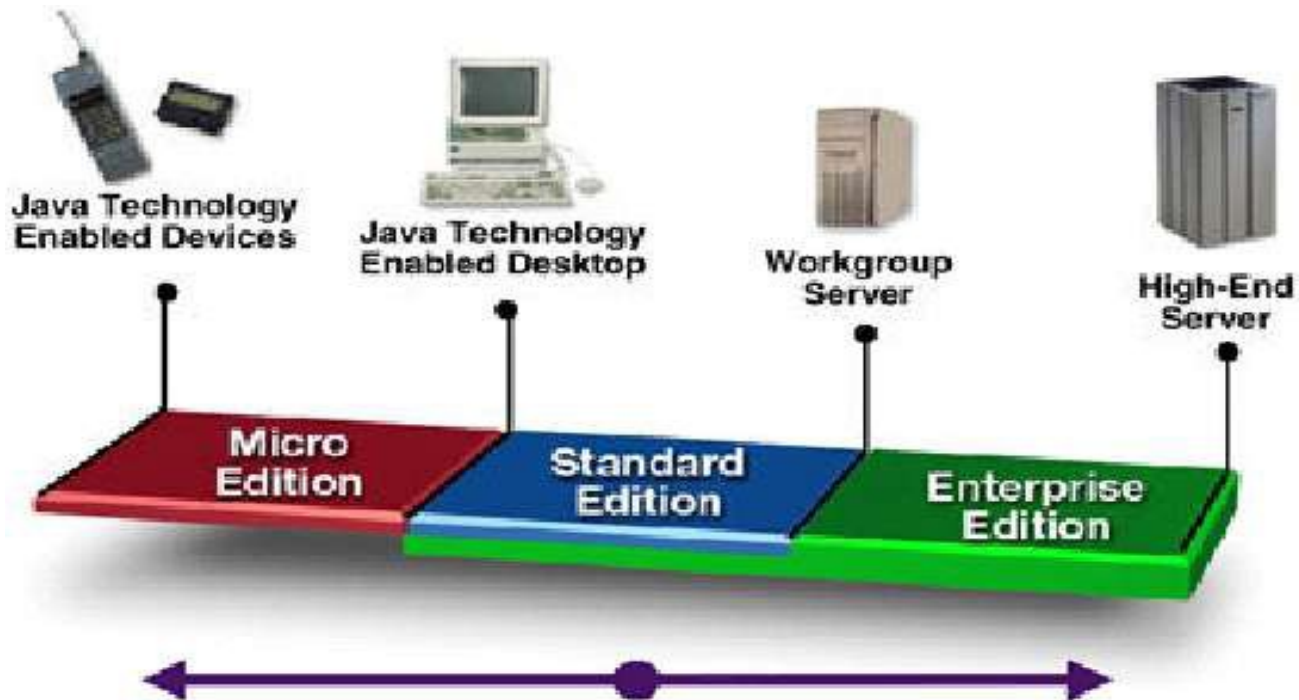


Processo de Desenvolvimento: RUP (Rational Unified Process)

Boas Práticas de Desenvolvimento Java:

<http://www.oracle.com/technetwork/java/codeconventions-150003.pdf>

# REVISÃO: JAVA





- As coisas que um objeto conhece sobre si mesmo se chamam atributos ou propriedades que **são as características de um objeto**;
- As coisas que um objeto pode fazer se chamam métodos que **são as ações do objeto**.

Aluno		
Atributos	nome : String matricula : int	Conhece
Método	estudar()	Faz

Aluno
nome : String
matricula : int
estudar()

```
Aluno maria = new Aluno();
```

```
maria.nome = "Maria";  
maria.matricula = 921;  
maria.estudar();
```

```
Aluno joao = new Aluno();
```

```
joao.nome = "Raimundo";  
joao.matricula = 330;  
joao.estudar();
```

Um método pode receber parametros (atributos) que influenciam no resultado final do método.

```
void pular(double altura){  
    if (altura<1){  
        System.out.println("Sucesso! ");  
    } else {  
        System.out.println("Não foi possível");  
    }  
}
```

# I REVISÃO: TIPOS DE DADOS

- Texto: Utiliza o objeto **String**

```
String nome = new String();
```

- Numérico: primitivo **int** e **double** não aceitam null e os objetos **Integer** e **Double** aceitam null e sua interação é por métodos.

```
int valor = 7;
```

```
double valorMoeda = 12.43
```

```
Integer valorNovo = new Integer(7);
```

```
Double valorMoedaNovo = new Double(12.43);
```

- Booleano: primitivo **boolean** não aceita null e o objeto **Boolean** aceita null e sua interação é por métodos.

```
boolean status = false;
```

```
Boolean statusNovo = new Boolean(true);
```

- Datas: A classe **java.util.Calendar** armazena uma data e a classe **java.text.DateFormat**, formata a sua apresentação

```
Calendar hoje = Calendar.getInstance();
```

```
Calendar inicioAula = new GregorianCalendar(2012, 1, 1);
```

```
DateFormat fmt = DateFormat.getDateInstance(java.text.DateFormat.MEDIUM);
```

```
System.out.println("\tData de Hoje: " + fmt.format(hoje.getTime()));
```

- Um objeto pode encapsular seus dados de forma que o usuário não terá acesso direto ao dado, dando a oportunidade de realizar validações e evitando inconsistências;
- Uma técnica para encapsulamento é usar get e set.

```
Aluno maria = new Aluno();  
maria.setRG("10.232.134");  
maria.setDataNascimento("22/10/1980");  
maria.setNome("Maria Eduarda da Silva");  
  
System.out.println(maria.getNome());
```

- Interface declara um comportamento abstrato que pode ser usado futuramente por uma classe
- Uma aplicação imediata de interface é para a declaração de constantes
- Sempre evite espalhar literais pelo código
- É recomendado que a constante seja utilizada com caixa alta (Oracle Code Conventions)

```
public interface IndiceFinanceiroInterface {  
  
    public static final String TAXA_JUROS = "7,14 %";  
    public static final double PI = 3.1416;  
  
}
```

# I REVISÃO: TRATAMENTO DE ERROS FIAP

Um objeto pode lançar exceções, caso haja um comportamento não esperado. Em programação podemos manipular estas exceções;

```
int a=1, b=0;
try {
    if (b == 0) throw new Exception("Elemento B não pode ser Zero");
    int c = a/b;
    System.out.println(c);
} catch (Exception e) {
    System.out.println(e.getMessage());
} finally {
    System.out.println("Fim");
}
```

# I REVISÃO: COLEÇÕES

Podemos armazenar e manipular coleções de objetos com arrays.

```
String[] a = new String[3];
a[0]= "thiago";
a[1]= "fiap";
//*****

Aluno[] c = new Aluno[3];
Aluno b = new Aluno();
b.setNome("Thiago");
c[0] = b;
//*****

ArrayList<Aluno> d = new ArrayList<Aluno>();
d.add(b);
//*****

System.out.println(a[1]);
System.out.println(c[0].getNome());
System.out.println(d.get(0).getNome());
```



Java Application ou Java Console são utilizadas como aplicações com interface texto ou aplicações com processamento em lote (batch);

```
public static void main(String[] args) {  
    Scanner teclado = new Scanner(System.in);  
  
    System.out.println("*** Digite o seu Nome:");  
    String txtNome = teclado.next();  
  
    System.out.println("*** Nome:" + txtNome);  
}
```

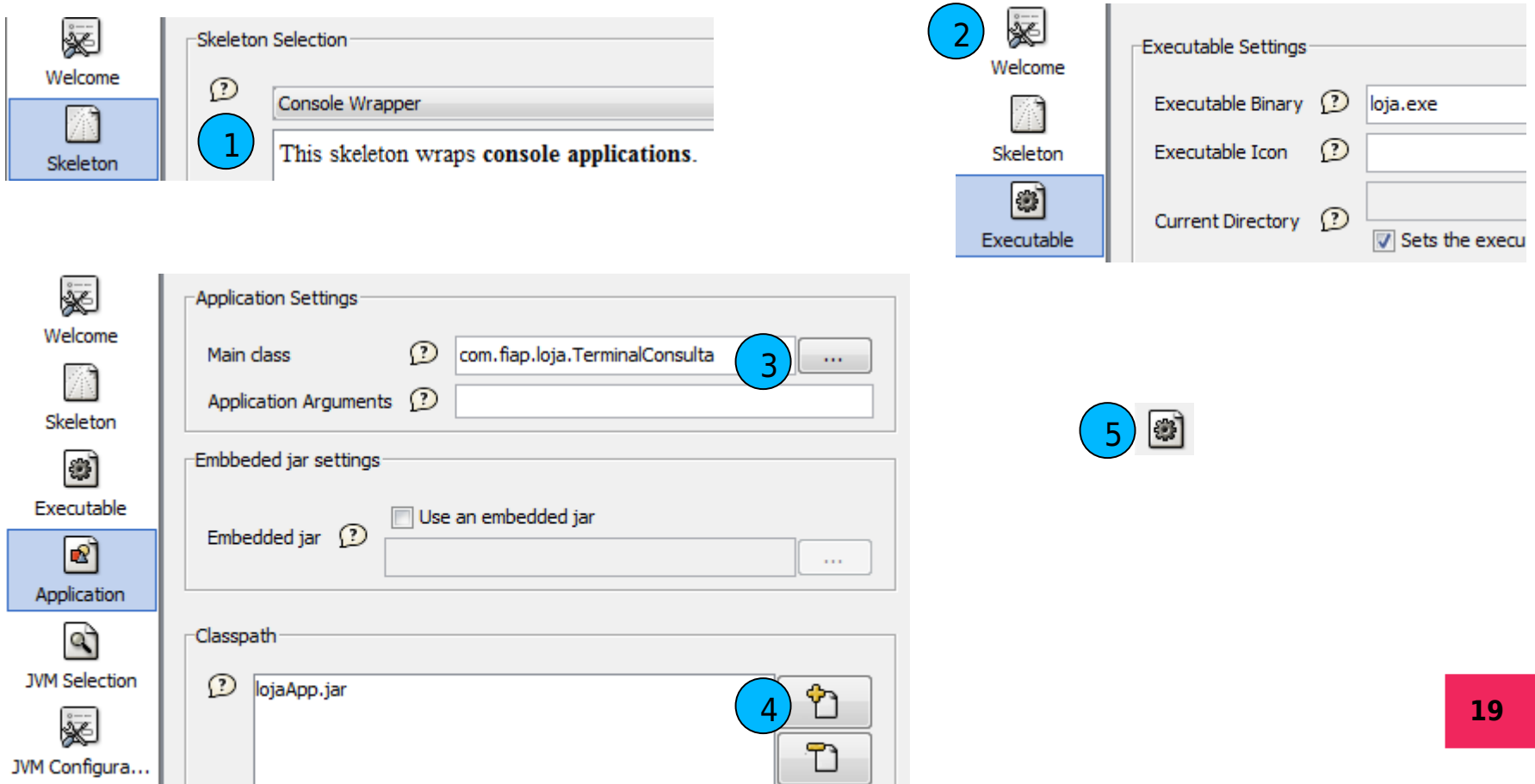
# EXERCÍCIO REVISÃO

FIAP



# BUILD & DEPLOYMENT

- O deployment de Java é realizado pelo empacotamento de classes (.class) agrupadas em um arquivo de deployment (.jar) pela opção **Export > Java > Runnable Jar File**
- Na console do Win/Mac/Linux digite: **java -cp minhaApp.jar com.fiap.MinhaAplicacao**
- Para estações Windows há a opção de gerar um Exe que age como wrapper (casca) para aplicações Windows. Necessário preparar a aplicação com algum utilitário externo. Um exemplo é o **Jsmooth** (<http://jsmooth.sourceforge.net/>)



## Desenvolvimento

- Criar um projeto Java Application com o nome “loja-app”
- Desenvolver uma classe “com.fiap.loja.TerminalConsulta” em que o usuário informe o código do produto e a aplicação retorne a descrição do produto e a data do dia. As regras de negócio para cadastro do produto são:
  - Caso o código do produto seja igual a 401, retornar a descrição: Camiseta Masculina Manga Curta Branca
  - Caso o código do produto seja igual a 402, retornar a descrição: Camiseta Feminina Manga Longa Rosa
  - Caso seja informado outro código, retornar a descrição: Produto Não Cadastrado!
  - Usar teclas de atalho <ctrl+shift+o>, <ctrl+i>, <ctrl+shift+f>, <ctrl+1>, <ctrl+3> quando necessárias
  - Usar Eclipse para fazer debug da aplicação (breakpoint, step into (F5) e step over (F6)) quando necessário
- Fazer o javadoc da aplicação

# EXERCÍCIO - DEV APP CONSOLE

## Teste

- Testar a aplicação conforme especificação acima
- Usar tecla de atalho <ctrl+F11> (Execução)

```
Java - loja-app/src/com/fiap/loja/TerminalConsulta.java - Eclipse
File Edit Source Refactor Navigate Search Search Project Run Run Window Help

Package Explorer
loja-app
├── src
│   └── com.fiap.loja
│       └── TerminalConsulta.java
└── JRE System Library [JavaSE-1.7]

TerminalConsulta.java
package com.fiap.loja;

import java.text.DateFormat;
import java.util.Calendar;
import java.util.Scanner;

/**
 * @author thiagoyamamoto
 */
public class TerminalConsulta {

    /**
     * @param args
     */
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int codigoProduto;

        String descricaoProduto = "Produto não cadastrado";

        DateFormat dfmt = DateFormat.getDateInstance(DateFormat.MEDIUM);

        System.out.println("FIAP Data: "+ dfmt.format(Calendar.getInstance().getTime()));
        System.out.println("*****");

        <terminated> TerminalConsulta [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (03/02/2013 21:53:51)
        FIAP Data: 03/02/2013
        *****
        Código do produto:401
        Descrição: Camiseta Masculina Manga Curta Branca
```

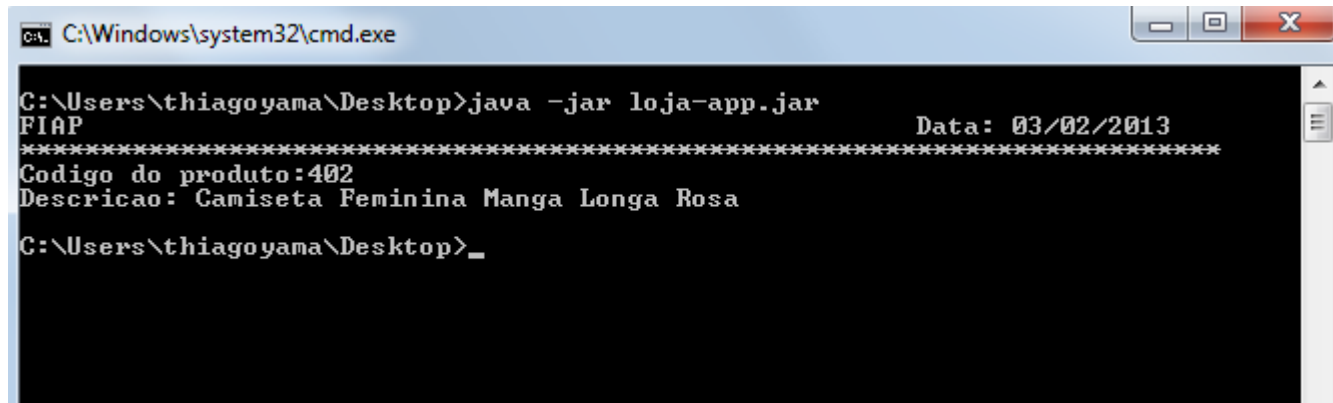
# EXERCÍCIO - DEV APP CONSOLE

## Build & Deployment

- Fazer o deployment em formato .jar
- Executar a aplicação em linha de comando, fora do eclipse
- Montar deployment para Windows (Usar Jsmooth)

## Executando

### JAR:

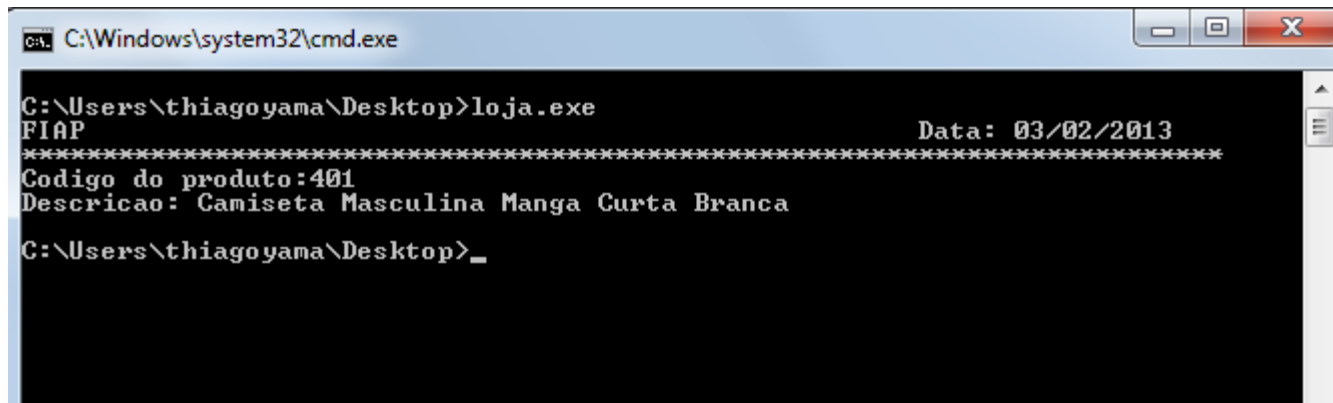


```
C:\Windows\system32\cmd.exe

C:\Users\thiagoyama\Desktop>java -jar loja-app.jar
FIAP                                     Data: 03/02/2013
*****
Codigo do produto:402
Descricao: Camiseta Feminina Manga Longa Rosa
C:\Users\thiagoyama\Desktop>_
```

## Executando

### EXE:



```
C:\Windows\system32\cmd.exe

C:\Users\thiagoyama\Desktop>loja.exe
FIAP                                     Data: 03/02/2013
*****
Codigo do produto:401
Descricao: Camiseta Masculina Manga Curta Branca
C:\Users\thiagoyama\Desktop>_
```

Copyright © 2013 - 2017 Prof. Me. Thiago T. I. Yamamoto

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).

*“Para de perseguir o dinheiro e comece a perseguir o sucesso”  
– Tony Hsieh*