

**Escola de Artes, Ciências e Humanidades - EACH-USP - USP Leste**

**ACH2147 - Desenvolvimento de Sistemas de Informação Distribuídos**

**Relatório - Exercício Programa I**

Ian Lourenço - 10430937

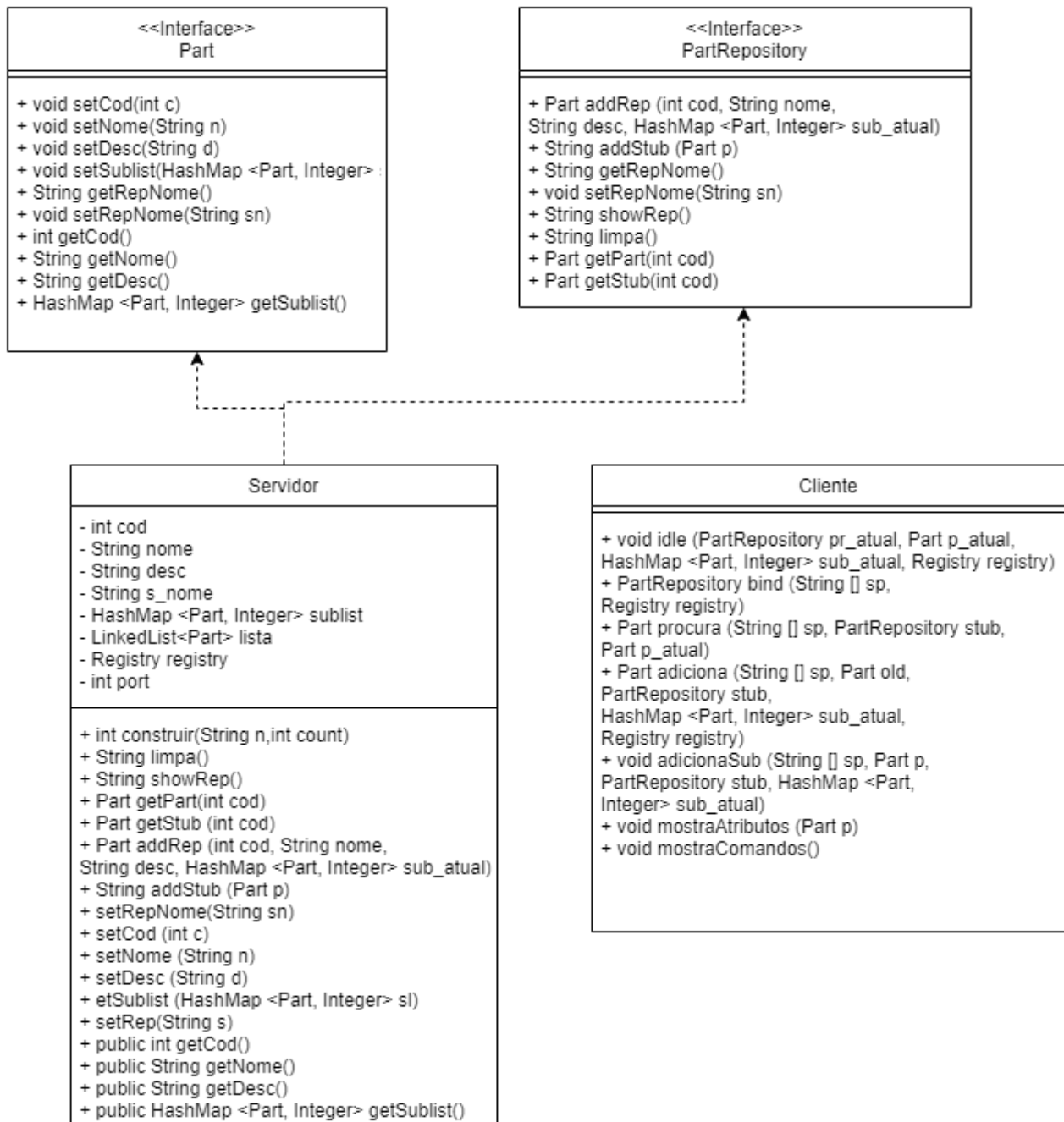
Mark Poll Herrmann - 11208291

Caio Fernandes Barbosa - 7557331

Guilherme Rodrigues Pisni - 11270851

Guilherme Kenzo Silva Oshiro - 11314988

## Diagrama de classes simplificado do programa



## Instruções gerais para o funcionamento

TESTADO APENAS EM WINDOWS

Versão do java:

java 14.0.1 2020-04-14

Java(TM) SE Runtime Environment (build 14.0.1+7)

Java HotSpot(TM) 64-Bit Server VM (build 14.0.1+7, mixed mode, sharing)

### 1ºPasso: Compile todos os arquivos

```
D:\Faculdade\Sistemas distribuídos\EP1>javac *.java  
D:\Faculdade\Sistemas distribuídos\EP1>
```

### 2ºPasso: Inicie o rmiregistry

```
D:\Faculdade\Sistemas distribuídos\EP1>javac *.java  
D:\Faculdade\Sistemas distribuídos\EP1>start rmiregistry  
D:\Faculdade\Sistemas distribuídos\EP1>_
```

### 3ºPasso: Inicie o Servidor

```
D:\Faculdade\Sistemas distribuídos\EP1>javac *.java  
D:\Faculdade\Sistemas distribuídos\EP1>start rmiregistry  
D:\Faculdade\Sistemas distribuídos\EP1>java Servidor
```

Em seguida, o prompt irá pedir que digite a quantidade de servidores que deseja inicializar. Digite o número e aperte enter.

```
D:\Faculdade\Sistemas distribuídos\EP1>javac *.java  
D:\Faculdade\Sistemas distribuídos\EP1>start rmiregistry  
D:\Faculdade\Sistemas distribuídos\EP1>java Servidor  
Digite o numero de servidores que deseja abrir: 3
```

Então, o prompt irá pedir o nome para cada um dos servidores, digite o nome, aperte enter e ele seguirá para o próximo servidor.

```
D:\Faculdade\Sistemas distribuídos\EP1>javac *.java
D:\Faculdade\Sistemas distribuídos\EP1>start rmiregistry
D:\Faculdade\Sistemas distribuídos\EP1>java Servidor
Digite o numero de servidores que deseja abrir: 3
Digite o nome do servidor 1: Servidor 1
Digite o nome do servidor 2: Servidor 2
Digite o nome do servidor 3: _
```

Ao final, será exibida uma lista com o nome dos servidores criados

```
D:\Faculdade\Sistemas distribuídos\EP1>javac *.java
D:\Faculdade\Sistemas distribuídos\EP1>start rmiregistry
D:\Faculdade\Sistemas distribuídos\EP1>java Servidor
Digite o numero de servidores que deseja abrir: 3
Digite o nome do servidor 1: Servidor 1
Digite o nome do servidor 2: Servidor 2
Digite o nome do servidor 3: Servidor 3
-----Nome dos servidores-----
Nome do servidor 1: Servidor 1
Nome do servidor 2: Servidor 2
Nome do servidor 3: Servidor 3
-----
```

#### 4ºPasso: inicialize o cliente

Abra um outro prompt e execute a classe cliente através do comando “java”

```
D:\Faculdade\Sistemas distribuídos\EP1>java Cliente
```

Então o prompt pedirá o nome do Servidor que você deseja conectar, escreva-o e aperte enter, será exibida uma mensagem de sucesso

```
D:\Faculdade\Sistemas distribuídos\EP1>java Cliente
Digite o nome do servidor a qual deseja se conectar: Servidor 2
Conectado ao servidor Servidor 2 com sucesso!
```

**5ºPasso: Utilize os comandos abaixo para manipular o servidor/Part/PartRepository que desejar**

## Implementações

### Comandos Cliente (Separe os parâmetros por espaço):

#### **bind (*nome do servidor*)**

Faz o cliente se conectar a outro servidor e muda o repositório corrente. Este comando recebe o nome de um repositório e obtém do serviço de nomes uma referência para esse repositório, que passa a ser o repositório corrente.

#### **listp ()**

Lista as peças do repositório corrente.

#### **getp (*código da peça*)**

Busca uma peça por código. A busca é efetuada no repositório corrente. Se encontrada, a peça passa a ser a nova peça corrente.

#### **showp ()**

Mostra atributos da peça corrente.

#### **clearlist ()**

Esvazia a lista de sub-pecas corrente.

#### **addsubpart (*quantidade*)**

Adiciona à lista de sub-pecas corrente *n* unidades da peça corrente.

#### **addp (*Nome da peça*) (*Descrição da peça*)**

Adiciona uma peça ao repositório corrente. A lista de sub-pecas corrente é usada como lista de subcomponentes diretos da nova peça. (É só para isto que existe a lista de sub-pecas corrente).

#### **quit ()**

Encerra a execução do cliente.

#### **swaplist ()**

A lista de subcomponentes corrente é adicionada no lugar da lista de subcomponentes da peça atual.

#### **help ()**

Exibe a lista de comandos disponíveis.

#### **addpc ()**

Adiciona a peça atual ao servidor corrente.

## Servidor

Ao iniciar, a classe servidor solicita a quantidade de servidores que deverão ser abertos e o nome de cada um deles.

Após isso, eles serão acessíveis através do comando **bind** da classe Cliente.

```
C:\WINDOWS\system32\cmd.exe
C:\java\DSID>javac *.java
C:\java\DSID>start rmiregistry
C:\java\DSID>java Servidor
Digite o numero de servidores que deseja abrir: 3
Digite o nome do servidor 1: Produtor
Digite o nome do servidor 2: Manufatura
Digite o nome do servidor 3: Loja
-----Nome dos servidores-----
Nome do servidor 1: Produtor
Nome do servidor 2: Manufatura
Nome do servidor 3: Loja
-----
```

## Cliente

Ao ser iniciado, a classe cliente solicita o nome do servidor a ser conectado. E, então, o Cliente ficará em um ciclo, onde espera comandos do usuário e os executa (tais comandos foram explicados previamente).

```
C:\java\DSID>java Cliente
Digite o nome do servidor a qual deseja se conectar: Produtor
Conectado ao servidor Produtor com sucesso!
```

## Exemplo

Conectando com o servidor:

```
C:\java\DSID>java Cliente
Digite o nome do servidor a qual deseja se conectar: Produtor
Conectado ao servidor Produtor com sucesso!
```

Agora vamos demonstrar o uso dos comandos:

Primeiro usamos o comando 'addp' para adicionar peças no servidor:

```
addp Tabua feita de madeira
Peca Tabua incluida com sucesso!
addp Prego feito de metal
Peca Prego incluida com sucesso!
addp Pe_De_Madeira feito de madeira
Peca Pe_De_Madeira incluida com sucesso!
```

Então podemos usar 'addsubpart' para adicionarmos a lista de subpeças carregada:

```
Peca Pe_De_Madeira incluida com sucesso!
addsubpart 4
Peca adicionada a lista de subcomponentes corrente!
```

Usando o comando 'get' podemos carregar uma peça do servidor e, assim podemos adicionar mais peças à lista de subpeças carregadas:

```
getp 25080
Peca selecionada: Tabua Codigo: 25080
addsubpart 1
Peca adicionada a lista de subcomponentes corrente!
getp 85057
Peca selecionada: Prego Codigo: 85057
addsubpart 4
Peca adicionada a lista de subcomponentes corrente!
```

Podemos usar o comando 'bind' para mudarmos de servidor. Quando adicionamos uma peça irá ser carregada, agora, a lista de subpeças:

```
bind Manufatura
Conectado agora ao servidor: Manufatura
addp Mesa Mesa de madeira
Peca Mesa incluida com sucesso!
```

Como pode ser demonstrada pelo comando 'listp':

```
listp
NOME DO SERVIDOR: Manufatura Quantidade de pecas: 1
Peca Numero 0
Codigo: 72868 Nome: Mesa
Descricao: Mesa de madeira
-----SUBCOMPONENTES-----
Codigo: 85057 Nome: Pregos Quantidade: 4
Descricao: feito de metal
Codigo: 25080 Nome: Tabua Quantidade: 1
Descricao: feita de madeira
Codigo: 46553 Nome: Pe_De_Madeira Quantidade: 4
Descricao: feito de madeira
```

Podemos passar a peça criada nesse servidor a outro usando os comandos 'bind' e 'addpc':

```
bind Loja
Conectado agora ao servidor: Loja
```

```
addpc
Peca: Mesa (CODIGO: 72868) Adicionada com sucesso ao servidor: Loja
```