



UNIVERSIDADE DE ÉVORA
ESCOLA DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA

Linguagens de Programação

por

Mafalda Rosa 40021,
Miguel Carvalho 4116

A máquina possui três zonas de memória. Assim sendo, foi criado um *ArrayList* para cada parte da memória.

```
public class TISC {  
    private ArrayList<Object> mem;  
    private ArrayList<Object> a;  
    private ArrayList<Object> exe;
```

Verificamos que cada tipo de instrução tem os mesmos argumentos à exceção das instruções do tipo *exit* e *chamada de funções*. Para cada tipo de instrução foram criadas classes em que o construtor recebe os argumentos dessa mesma instrução.

Após a criação das classes, ao ficheiro *regact.cup* foram adicionados os objetos através de um método criado no *TISC.java*, onde se encontra a nossa memória.

```
PUBLIC VOID ADD_INST(OBJECT OBJ)  
{  
    MEM.ADD(OBJ);  
}  
  
PUBLIC VOID ADD_LABEL(LABEL LABEL)  
{  
    MEM.ADD(LABEL);  
}
```

Instruções Aritméticas: A classe *Instrucao_Art*, contém uma string com o nome da operação aritmética.

Instruções de Salto: A classe correspondente é a classe *Jumps*, que tal como a anterior contém uma string com o nome, e contém outra string que corresponde à etiqueta.

Instruções para Manipulação de Inteiros: *Man_int*, a classe contém uma string com o nome e um *int* para o inteiro.

Instruções de Acesso a Variáveis e a argumentos: Ambas as classes têm dois inteiros e uma string, *Var_Acess* e *Arg_Acess*.

Instruções para Chamada de Funções: A classe *Call_Func* contém três construtores. Cada construtor recebe os argumentos de cada instrução.

Instruções de Saída: *Exit*, esta classe, tal como a anterior, possui mais que um construtor.