

# Construção de um compilador de MiniJava para Dalvik usando Objective Caml

Fulano

`fulano@gmail.com`

Ciclano

`ciclano@gmail.com`

Beltrano

`beltrano@gmail.com`

Faculdade de Computação  
Universidade Federal de Uberlândia

22 de abril de 2014

# Lista de Figuras

2.1	Página de entrada do site <a href="http://studiosum.com.br/CC1/index.php">http://studiosum.com.br/CC1/index.php</a> . . . . .	9
-----	---	---

# Lista de Tabelas

# Lista de Listagens

1.1	exemplo1.ml	8
A.1	prog1.mj	11
A.2	prog1.mj	12

# Sumário

<b>Lista de Figuras</b>	<b>2</b>
<b>Lista de Tabelas</b>	<b>3</b>
<b>1 Introdução</b>	<b>6</b>
1.1 Linha de comando . . . . .	6
1.2 Códigos de programas . . . . .	7
<b>2 Imagens</b>	<b>9</b>
 <b>Apêndice</b>	 <b>11</b>
<b>A Exemplos de programas escritos em MiniJava</b>	<b>11</b>
A.1 prog1.mj . . . . .	11
A.2 fact.mj . . . . .	12

# Capítulo 1

## Introdução

Este documento foi escrito para auxiliar na confecção do relatório da disciplina. É necessário olhar os fontes deste documento em L<sup>A</sup>T<sub>E</sub>X para compreender algumas coisas.

### 1.1 Linha de comando

Para dar instruções sobre linha de comando use um ambiente que preparei (veja o preâmbulo desse documento para aprender como criar seu próprio ambiente):

```
\begin{terminal}  
> sudo apt-get install tree  
\end{terminal}
```

Um programa bastante útil é o `tree` que lista o conteúdo de um diretório e de seus subdiretórios em forma de árvore. Para instalá-lo no Ubuntu, use:

```
> sudo apt-get install tree
```

A seguir é mostrado o uso:

```
> tree --charset=ASCII  
.  
|-- codigos  
|   |-- exemplo1.ml  
|   `-- MiniJava  
|       |-- fact.mj  
|       `-- prog1.mj  
|-- compiladores.aux  
|-- compiladores.idx  
|-- compiladores.lof  
|-- compiladores.log  
|-- compiladores.lol  
|-- compiladores.lot  
|-- compiladores.out  
|-- compiladores.pdf  
|-- compiladores.tex
```

```
|-- compiladores.toc
|-- relatorio.tex

2 directories, 14 files
```

## 1.2 Códigos de programas

Para introduzir a listagem do código no documento existem pelo menos duas formas básicas, ambas usando o pacote `listings`:

1. Diretamente do documento  $\text{\LaTeX}$  usando por exemplo

```
\begin{ocaml}
type binop =
  Plus
| Minus
| Times
| Div
| Mod
| Le
| Leq
| Ge
| Geq
| Eq
| Neq
| And
| Or
\end{ocaml}
```

cujo resultado é

```
type binop =
  Plus
| Minus
| Times
| Div
| Mod
| Le
| Leq
| Ge
| Geq
| Eq
| Neq
| And
| Or
```

2. Lendo o código diretamente do arquivo:

```
\lstinputlisting[caption={exemplo1.ml}]{codigos/exemplo1.ml}
```

cujo resultado é mostrado na listagem [1.1](#).

Listagem 1.1: exemplo1.ml

```
1 type binop =  
2   Plus  
3   | Minus  
4   | Times  
5   | Div  
6   | Mod  
7   | Le  
8   | Leq  
9   | Ge  
10  | Geq  
11  | Eq  
12  | Neq  
13  | And  
14  | Or  
15  
16 type uop = Uminus | Neg  
17  
18 type mptype = Integer | Boolean | Void | Record of string | Pointer
```

---



# Capítulo 2

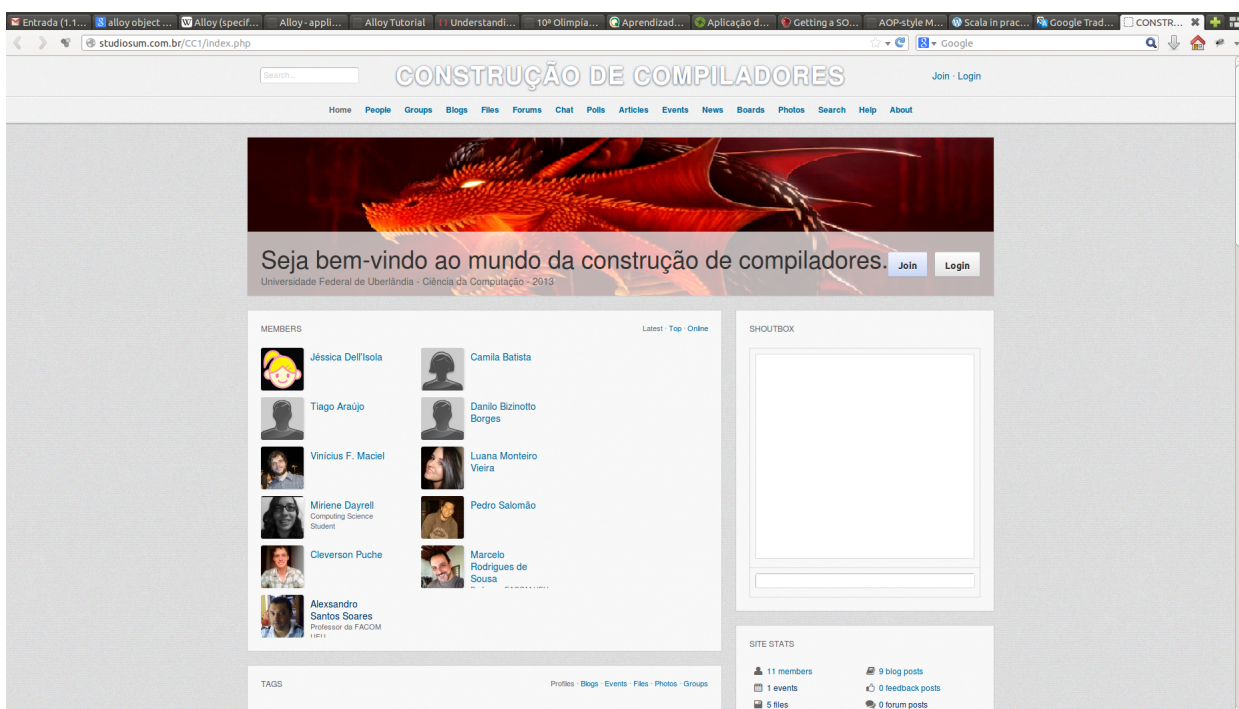
## Imagens

Se desejar incluir imagens você poderá usar o comando a seguir:

```
\begin{figure}[!ht]
\centering
\caption{Página de entrada do site
        \url{http://studiosum.com.br/CC1/index.php} \label{fig:1}}
\includegraphics[scale=0.27]{imagens/fig1.png}
\end{figure}
```

O comando anterior gerará a imagem a seguir, dado que o arquivo `fig1.png` esteja no subdiretório `imagens`.

**Figura 2.1:** *Página de entrada do site <http://studiosum.com.br/CC1/index.php>*



Note que a Figura [2.1](#) pode ser referenciada em qualquer parte do documento. Você também pode incluir diretamente outros formatos de imagens tais como o jpg.

# Apêndice A

## Exemplos de programas escritos em MiniJava

### A.1 prog1.mj

Listagem A.1: prog1.mj

```
1 class A
2 {
3     public static void main(String[] a)
4     {
5         System.out.println(new C().m(1));
6     }
7 }
8
9
10 class C {
11     int a; int b; int c;
12
13     public int m(int d) {
14         int j;
15
16         j = a+b+d;
17
18         System.out.println(a+c);
19
20         System.out.println(a);
21         System.out.println(j);
22         System.out.println(b);
23
24         return j;
25     }
26 }
27
28 class E {
29     int a;
30
31     public int E() {
32         a = 5;
33
34         return a;
```

```

35         }
36     }
37
38     class N {
39         int b;
40         int a;
41
42         public int N() {
43             b = 10;
44             a = E.E() + b;
45
46             return a;
47         }
48
49
50     }
51
52     class D {
53         int d;
54
55         public int D() {
56             d = E.E() + N.N();
57
58             return d;
59         }
60     }

```

---

## A.2 fact.mj

### Listagem A.2: prog1.mj

```

1  class factorials
2  {
3      public static void main(String[] args)
4      {
5          int a;
6          fact A = new fact();
7          System.out.println("\nthe factorial of 6 is ");
8          a = A.f1(6);
9          System.out.println( a );
10         System.out.println("with tail-recursion it's also ");
11         a = A.f2(6,1);
12         System.out.println( a );
13         System.out.println("without recursion it's still ");
14         a = A.f3(6);
15         System.out.println( a );
16     }
17 }
18
19
20 class fact
21 {
22     public int f1(int n)
23     {
24         int a;
25         if (n<2) a=1;
26         else a = n * (this.f1(n-1));

```

## A.2

```
27         return a;
28     }
29
30     public int f2(int n, int accum)    // accum must be initially 1
31     {
32         int a;
33         if (n<2) a = accum;
34         else a = this.f2(n-1, accum*n);
35         return a;
36     }
37
38     public int f3(int n)
39     {
40         int accum = 1;
41         while (1<n)
42         {
43             accum = accum*n;
44             n = n-1;
45         }
46         return accum;
47     }
48 } // fact
```

---