



TADs Básicos

tad Bool where

true : Bool
false : Bool
 \neg : Bool \rightarrow Bool
and : Bool \rightarrow Bool \rightarrow Bool
or : Bool \rightarrow Bool \rightarrow Bool
 (\Rightarrow) : Bool \rightarrow Bool \rightarrow Bool
if _ then _ else _ : Bool \rightarrow E \rightarrow E \rightarrow E

\neg true = false
 \neg false = true
and true x = x
and false x = false
or true x = true
or false x = x
true $\Rightarrow x$ = x
false $\Rightarrow x$ = true
if true then x else y = x
if false then x else y = y

tad Nat where

import Bool

0 : Nat
succ : Nat \rightarrow Nat
(>) : Nat \rightarrow Nat \rightarrow Bool
(\equiv) : Nat \rightarrow Nat \rightarrow Bool
(+) : Nat \rightarrow Nat \rightarrow Nat
(-) : Nat \rightarrow Nat \rightarrow Nat

$x + 0$ = x
 $x + (\text{succ } y)$ = succ ($x + y$)
 $0 - x$ = 0
 $x - 0$ = x
 $(\text{succ } x) - (\text{succ } y)$ = $x - y$
 $0 \equiv 0$ = true
 $(\text{succ } x) \equiv 0$ = false
 $0 \equiv (\text{succ } y)$ = false
 $0 > x$ = false
 $(\text{succ } x) > 0$ = true
 $(\text{succ } x) > (\text{succ } y)$ = $x > y$

```
tad Int where
  import Bool
  0    : Int
  succ : Int → Int
  pred : Int → Int
  neg  : Int → Int
  abs  : Int → Int
  (+)  : Int → Int → Int
  (−)  : Int → Int → Int
  (*)  : Int → Int → Int
  div  : Int → Int → Int
  mod  : Int → Int → Int
  min  : Int → Int → Int
  max  : Int → Int → Int
  (>)  : Int → Int → Bool
  (≡)  : Int → Int → Bool
```

{− se considera la especificación usual para este tipo −}

```
tad Tuplan (A1, ..., An : Set) where
  p1 : Tuplan (A1, ..., An) → A1
  ...
  pn : Tuplan (A1, ..., An) → An
  (−, ..., −) : A1 → ... → An → Tuplan (A1, ..., An)
```

```
p1 (x1, ..., xn) = x1
...
pn (x1, ..., xn) = xn
```

```
tad Maybe (A : Set) where
  import Bool
  Nothing : Maybe A
  just    : A → Maybe A
  isNothing : Maybe A → Bool
  fromJust : Maybe A → A
```

```
isNothing (Nothing) = true
isNothing (Just x)  = false
fromJust (Just x)   = x
```