Ahorcado

Programación funcional imperativa

Juan Pedro Villa Isaza

Stack Builders

1 de junio de 2015

Haskell es:

- Funcional
- *Puro*
  - Inmutabilidad
  - Sin efectos secundarios
  - Transparencia referencial
- ...

```haskell
reverse :: [a] -> [a]
reverse []     = []
reverse (x:xs) = reverse xs ++ [x]
```

```
> reverse "haskell"
"lleksah"
```

Haskell es:

- *Interesante*

Stack Builders

Haskell es:

- *Interesante*
- ¿Inútil?

Stack Builders

```
main :: IO ()
main = putStrLn (reverse "haskell")
```

```
> main
lleksah
```

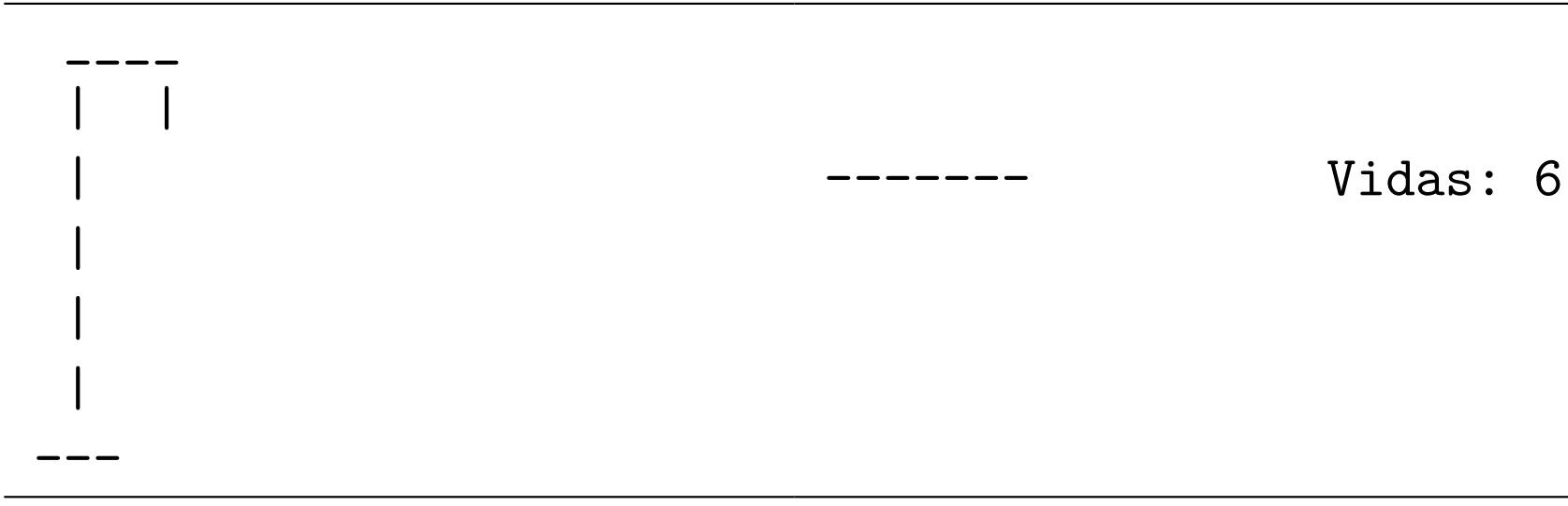
```haskell
main :: IO ()
main = do
  line <- getLine
  putStrLn (reverse line)
```

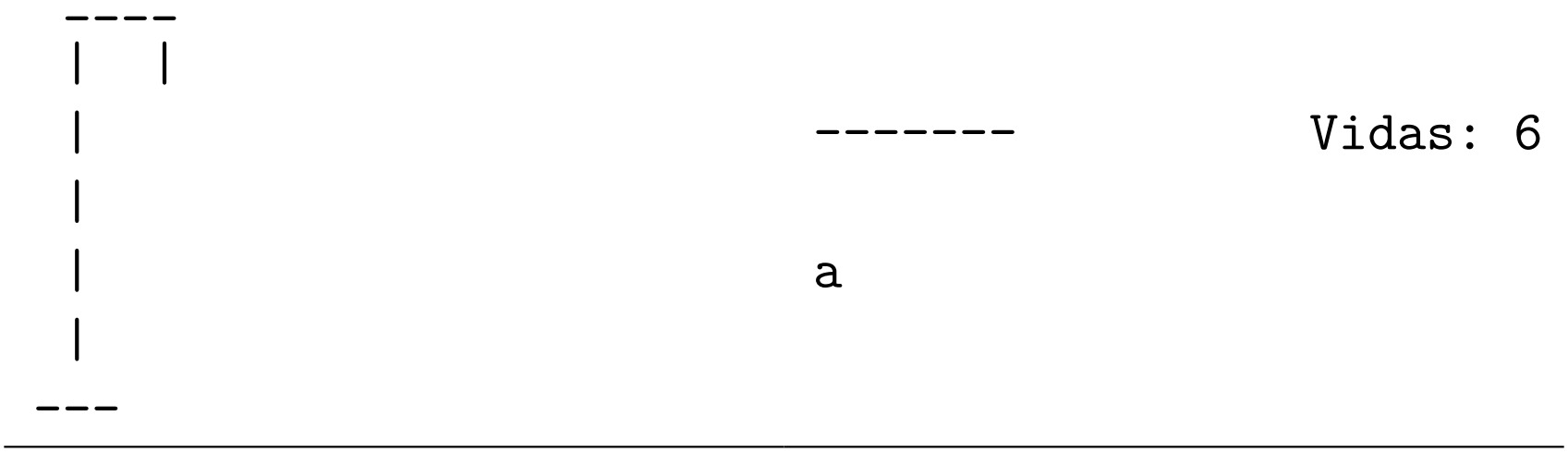
```
> main
haskell
lleksah
```

```
type IO a = Mundo -> (a,Mundo)
```

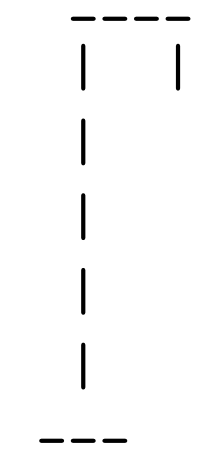
Haskell es el mejor lenguaje de programación imperativa del mundo.

—Simon Peyton Jones



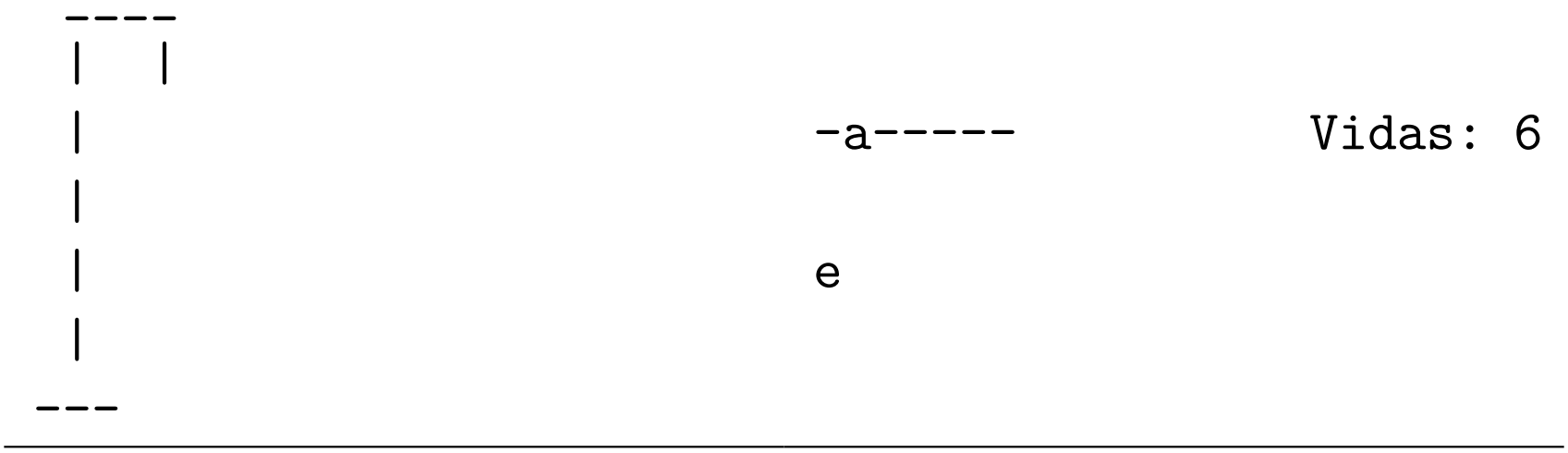


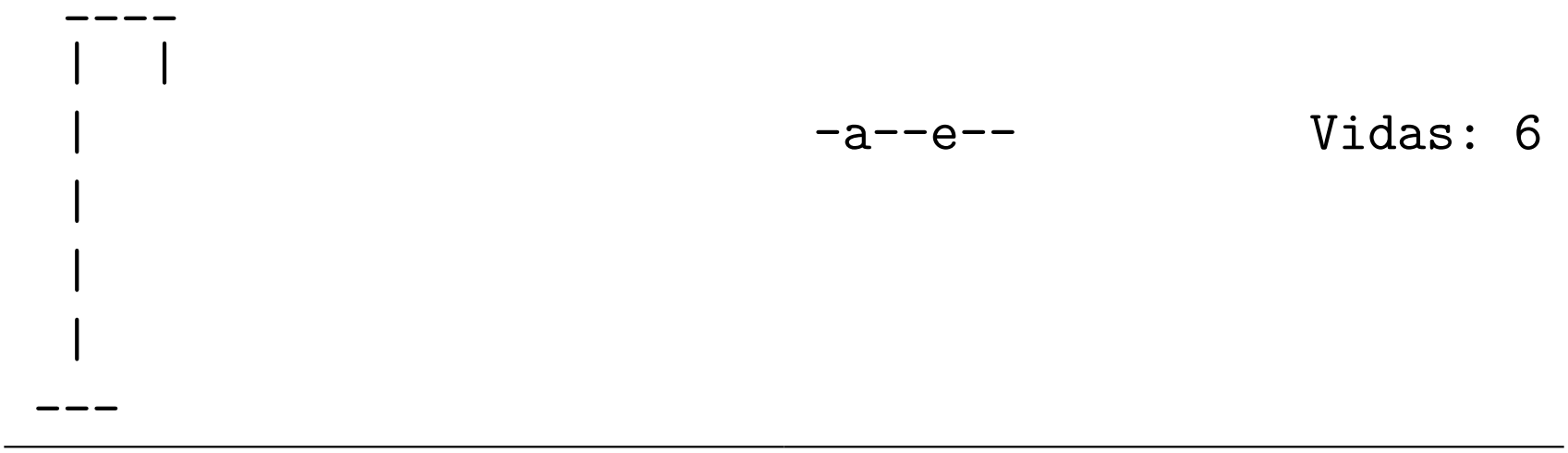
Stack Builders

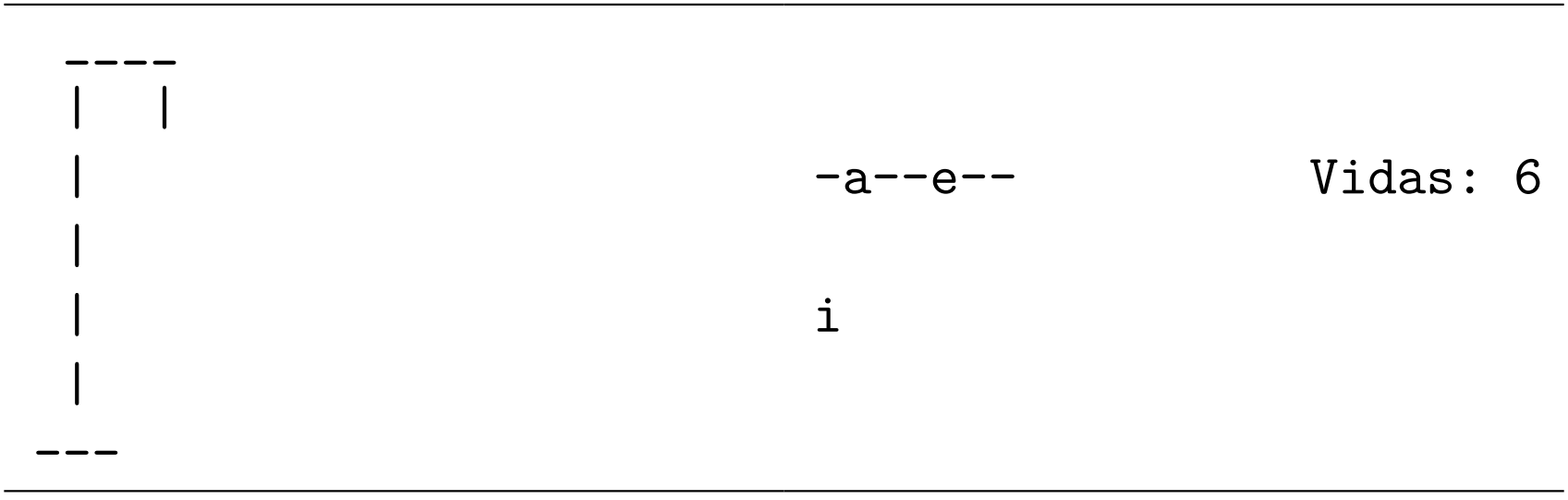


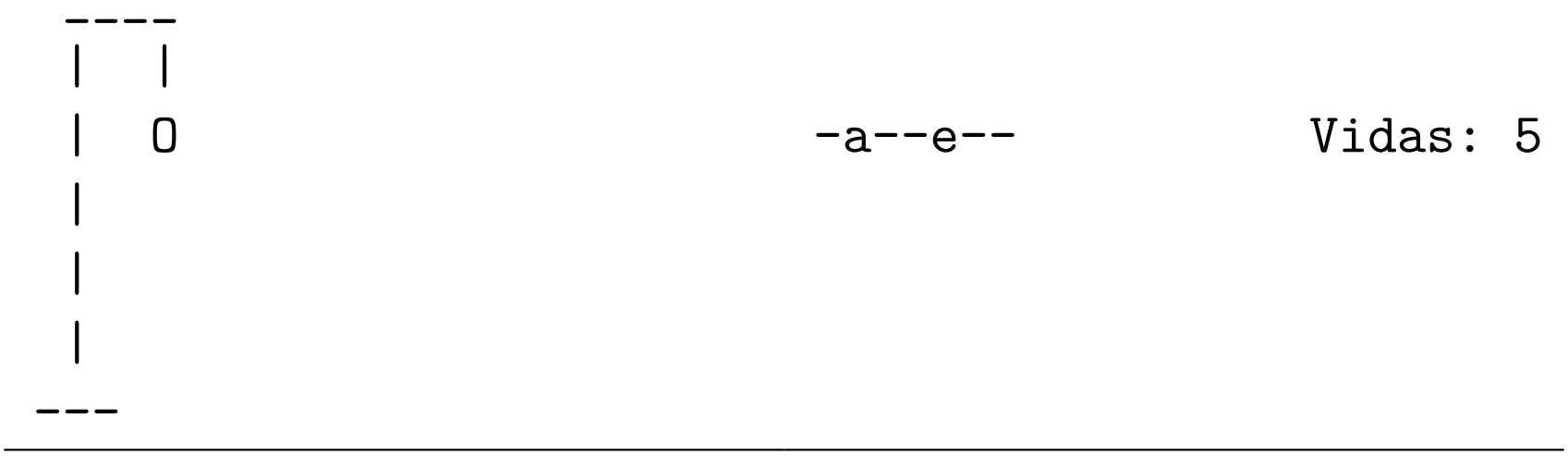
-a-----

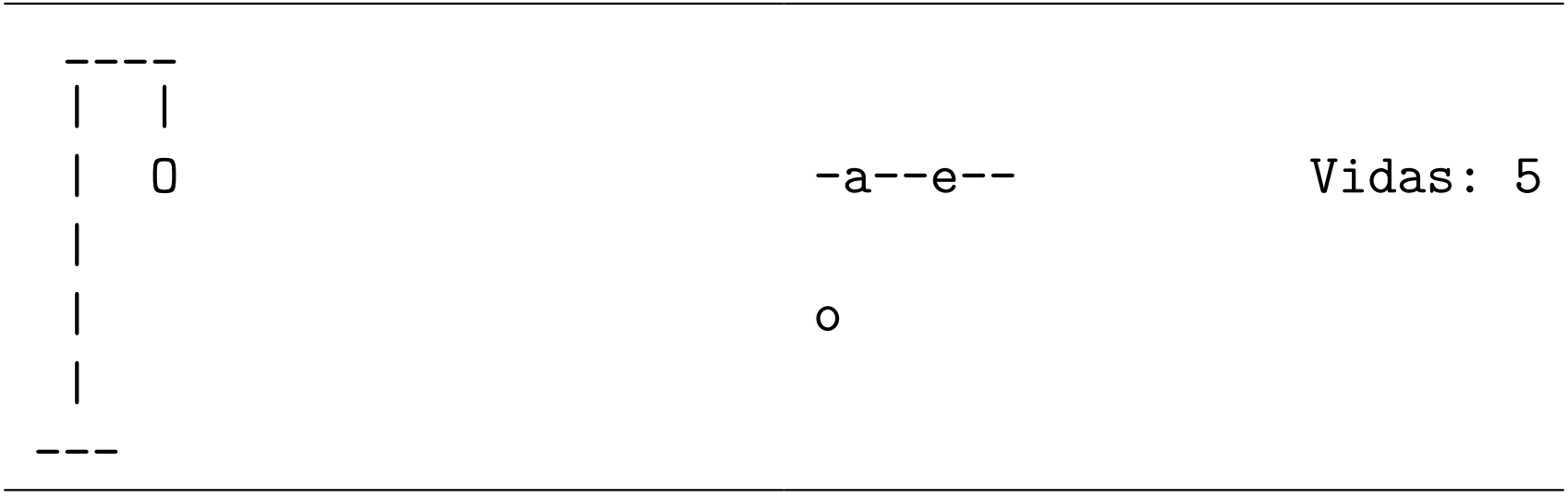
Vidas: 6

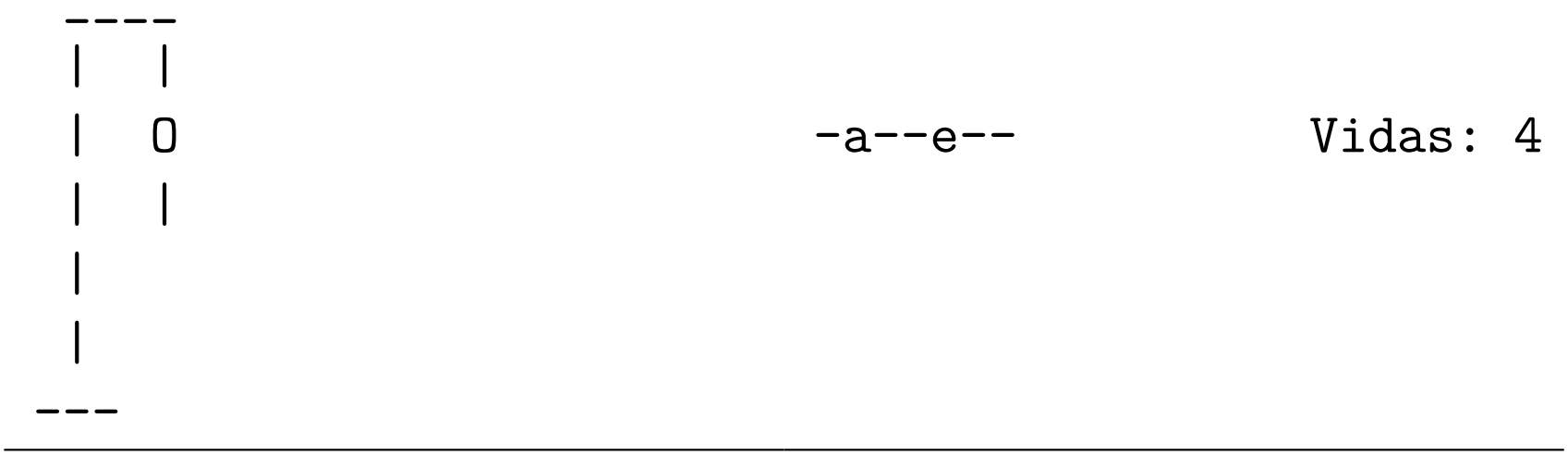


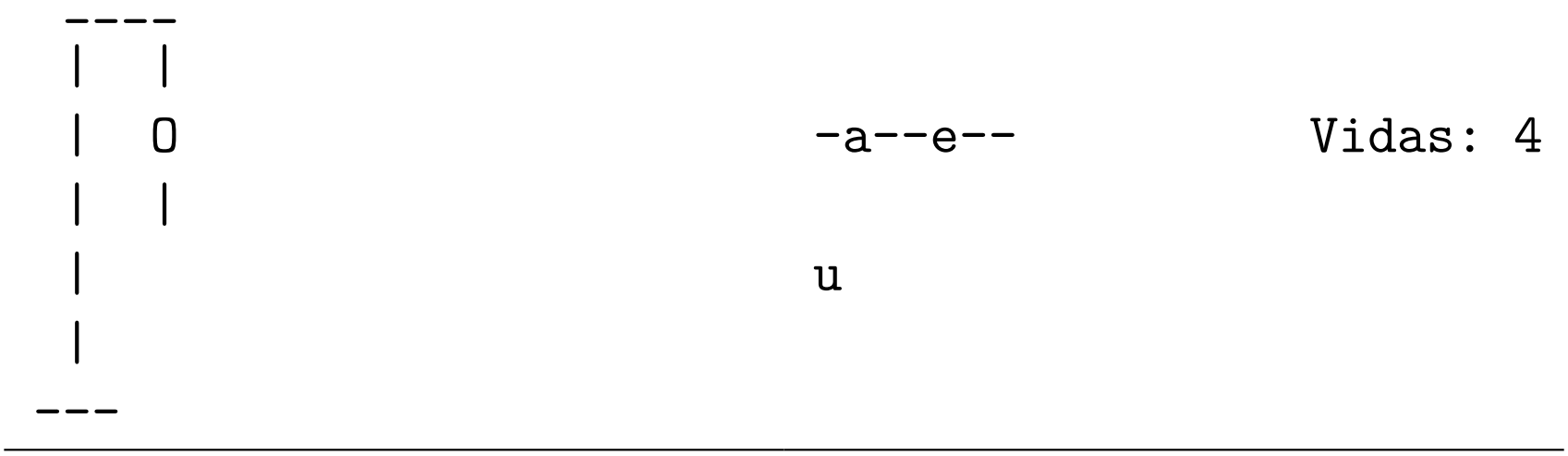


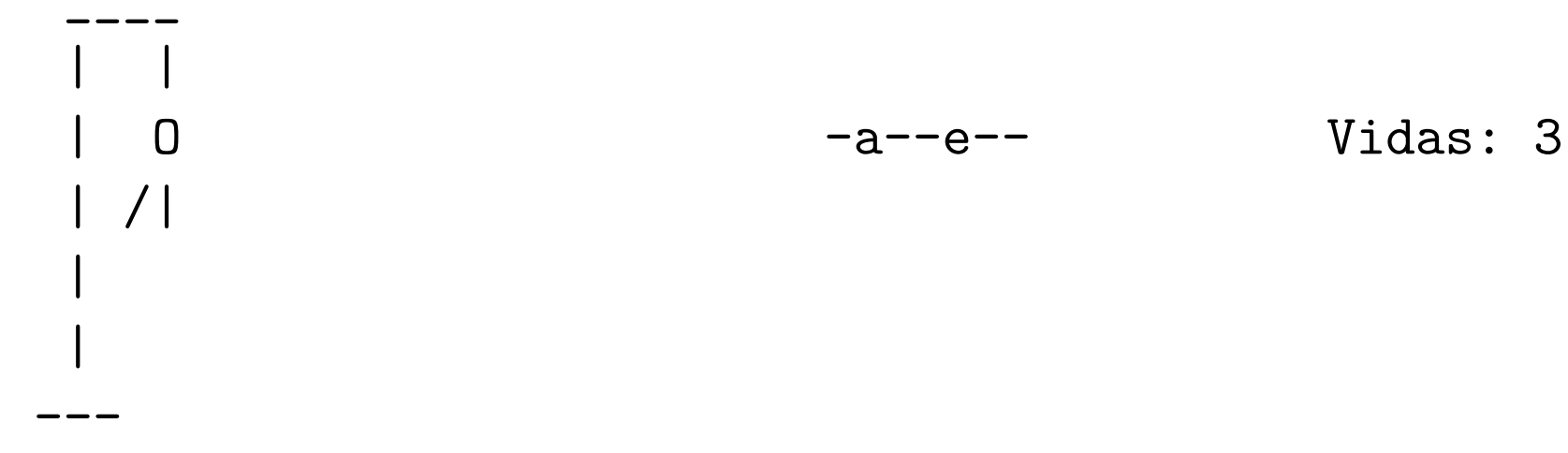


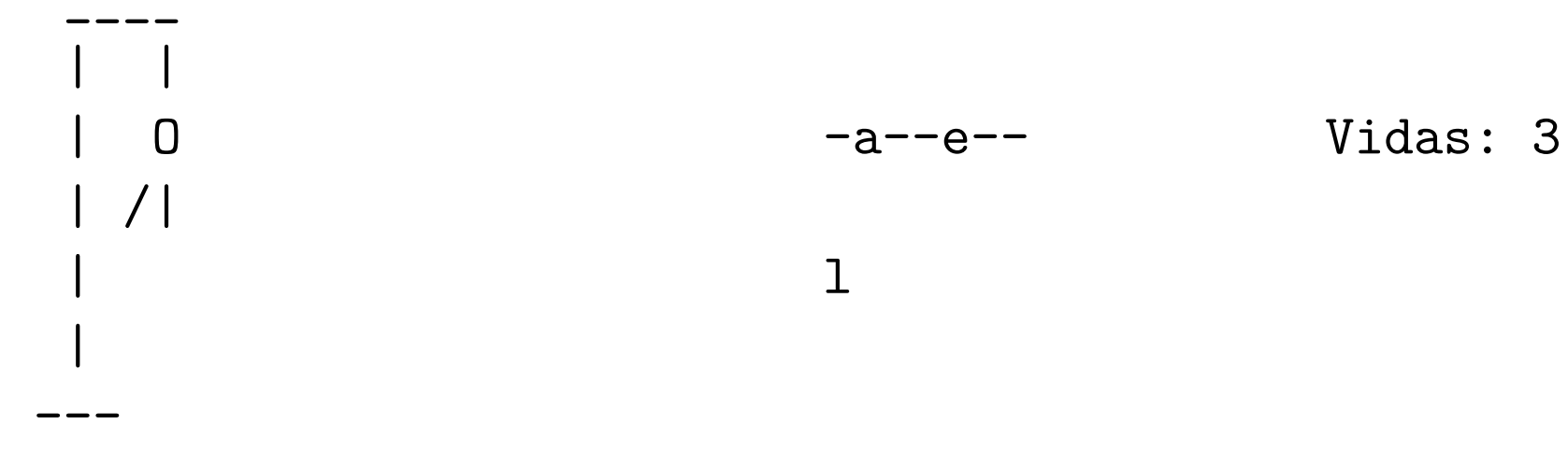


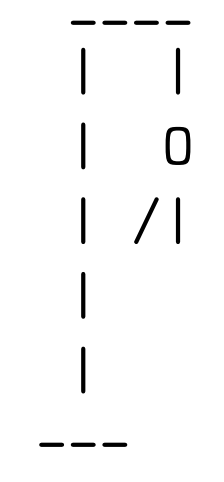






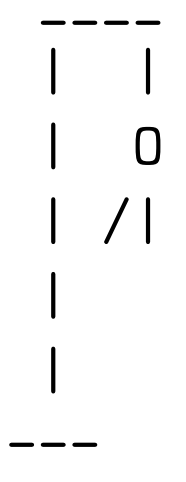






-a--ell

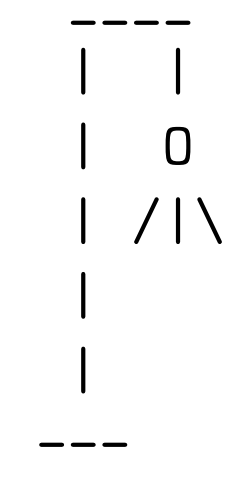
Vidas: 3



-a--ell

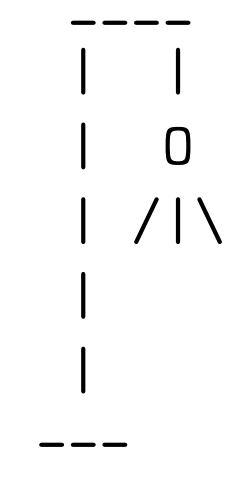
Vidas: 3

y



-a--ell

Vidas: 2

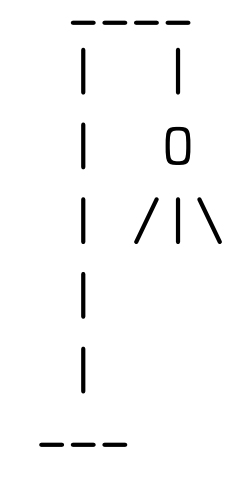


-a--ell

Vidas: 2

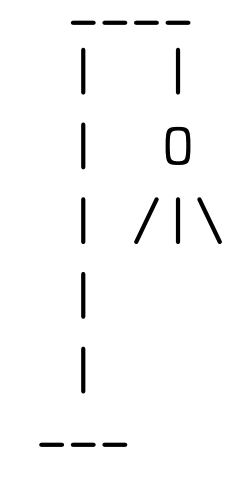
s

Stack Builders



-as-ell

Vidas: 2



-as-ell

Vidas: 2

haskell

¡haskell!

```
data HangmanGame = HangmanGame
  { hangmanGameStatus :: HangmanGameStatus
  , hangmanGameWord   :: HangmanWord
  }
```

```
data HangmanGameStatus
  = Lost
  | Playing Int
  | Won

type HangmanWord = [(Char,Bool)]
```

- ```
  nextHangmanGame :: HangmanGame
                  -> [Char]
                  -> HangmanGame
  ```

- ```
  nextHangmanWord :: HangmanWord
                  -> Char
                  -> Maybe HangmanWord
  ```

```
playHangmanGame :: HangmanGame -> IO ()
```

```
main :: IO ()
main = do
  ...
  playHangmanGame (HangmanGame ...)
  ...
```

https://github.com/stackbuilders/hangman-off

Gracias al sistema de tipos, distinción clara entre:

- `IO` y no `IO`
- Acciones y funciones
- Código impuro y código puro
- Programación imperativa y programación no imperativa

- Programación funcional:
  - Razonamiento ecuacional
  - ...
- Programación funcional imperativa:
  - Menos dolores de cabeza
  - ...

Haskell es el mejor lenguaje de programación imperativa del mundo.

—Simon Peyton Jones

Según las respuestas de más de 370000 personas:

- "El aprendizaje de este lenguaje mejoró mi habilidad como programador":

1. Haskell
2. Standard ML
3. Scheme
4. Coq
5. Common Lisp

45. Shell
46. MATLAB
47. PHP
48. Visual Basic
49. COBOL

`http://hmrp.pl/zdDNYP`

Según las respuestas de más de 370000 personas:

- "Recomendaría a la mayoría de programadores aprender este lenguaje, sin importar si tienen o no una necesidad específica de hacerlo":

1. Haskell
2. Scheme
3. Coq
4. Clojure
5. Erlang

44. ActionScript
45. Pascal
46. Fortran
47. Visual Basic
48. COBOL

`http://hmrp.pl/zuMkvG`

Según las respuestas de más de 370000 personas:

- "El aprendizaje de este lenguaje cambió significativamente la manera en que uso otros lenguajes":

1. Haskell
2. Scheme
3. Coq
4. Common Lisp
5. Erlang

43. Fortran
44. R
45. AWK
46. Visual Basic
47. COBOL

`http://hmrp.pl/yOtN9k`