

# Un lenguaje para la creación de DSLs. A language to making DSLs.

Autor: Eleazar Díaz Delgado  
Director: Casiano Rodríguez León

Escuela Superior de Ingeniería y Tecnología  
Departamento de Ingeniería Informática y de Sistemas  
Universidad de La Laguna

7 de Septiembre de 2018

- 1 Introducción
- 2 Objetivos
- 3 Estado
- 4 Usos
- 5 Conclusiones y futuras mejoras
- 6 Bibliografía

Un lenguaje diseñado para crear DSLs

## Objetivos generales

- REPL
- Scripts
- Archivos de configuracion
- Interfaz con Github

## Limitación del intérprete

- Linux, con posible soporte a MacOS
- Limitada cantidad de operaciones sobre objetos
- Manejo de la memoria

## REPL

- Básica funcionalidad del REPL
- Diversos comandos implementados
- Permite inserción de código multilínea

```
>>> if $cat somefile.txt$.exec().null():  
...     print "It is empty"  
...  
It is empty  
none  
>>> |
```

## Esquema interno

- Interoperabilidad
- AST flexible
- Reuso de funciones de Haskell

```
methodsTh
  [ fn "init" [| T.init :: T.Text -> T.Text |]
  , fn "null" [| T.null  :: T.Text -> Bool  |]
  ]
```

- Pequeños scripts.
- Scripts con configuraciones
- REPL



```
class Utils:  
    fun ...
```

# Conclusión y futuras mejoras I

- Implementación de la API de Github
- Mejorar la implementación de la memoria
- Poner en práctica el posible uso de FUSE en el DSL de Github
- Muchas posibles mejoras y características

```
use Github()

for repo in logged_user.repos | r/tfg-.*/:
  cd repo; use repo
  linter_info = !$ jslinter .

  if linter_info.status == 2:
    templateissue = Issue {
      title -> "linter fails",
      message ->
        "Doesn't pass linter:\n" ++ linter_info.output,
      assignees ->
        [ logged_user ] ++ collaborators,
    }
  new_issue templateissue
```

# Conclusions and future work I

- Implement Github API as DSL
- Improve current implemented memory manager
- Make prototype using libfuse in Github DSL
- A lot more of improvements and features

```
use Github()

for repo in logged_user.repos | r/tfg-.*/:
  cd repo; use repo
  linter_info = !$ jslinter .

  if linter_info.status == 2:
    templateissue = Issue {
      title -> "linter fails",
      message ->
        "Doesn't pass linter:\n" ++ linter_info.output,
      assignees ->
        [ logged_user ] ++ collaborators,
    }
  new_issue templateissue
```



Shayan Najd. Simon Peyton Jones.

Trees that grow.

*Journal of Universal Computer Science*, vol. 23, no. 1, 2017.



GHC User's Guide Documentation, Template Haskell.

[https://downloads.haskell.org/~ghc/latest/docs/html/users\\_guide/glasgow\\_exts.html#template-haskell](https://downloads.haskell.org/~ghc/latest/docs/html/users_guide/glasgow_exts.html#template-haskell).

Accessed: 2018-09-02.



GHC User's Guide Documentation, Overlapping Instances.

[https://downloads.haskell.org/~ghc/latest/docs/html/users\\_guide/glasgow\\_exts.html#overlapping-instances](https://downloads.haskell.org/~ghc/latest/docs/html/users_guide/glasgow_exts.html#overlapping-instances).

Accessed: 2018-09-02.



GHC User's Guide Documentation, QuasiQuoter.

[https://downloads.haskell.org/~ghc/latest/docs/html/users\\_guide/glasgow\\_exts.html#template-haskell-quasi-quotation](https://downloads.haskell.org/~ghc/latest/docs/html/users_guide/glasgow_exts.html#template-haskell-quasi-quotation).

Accessed: 2018-09-02.



Gabriel Gonzalez.

Why free monads matter.

<http://www.haskellforall.com/2012/06/you-could-have-invented-free-monads.html>.

Accessed: 2018-09-02.



Libfuse.

<https://github.com/libfuse/libfuse>.

Accessed: 2018-09-13.

Gracias por su atención