

DSL de edicion de imagenes

Idea general

Se otorgará una estructura para representar una secuencia de edición sobre imágenes. Se hará uso de la biblioteca `hip` para el manejo de imágenes en `haskell`.

Alcances

El lenguaje puede ser utilizado por programas o aplicaciones como backend para edición de imágenes. Ejemplos puede ser una aplicación de filtros para fotos (en los cuales cada filtro es una expresión del lenguaje) o un programa que agregue una marca de agua a una imagen.

Gramática

`Imagen ::= String` --La dirección de la imagen

`Exp ::= Imagen`

- | `BinOp Op Exp Exp`
- | `BoolOp BOp Exp Exp`
- | `Fun F Exp Float`
- | `Complement Exp`

`Op ::= Normal`

- | `Add`
- | `Sub`
- | `Diff`
- | `Div`
- | `Mult`
- | `Darken`
- | `Lighten`
- | `Multiply`
- | `Screen`
- | `Overlay`
- | `HardLight`
- | `SoftLight`
- | `Hue`
- | `Luminosity`
- | `Exclusion`

```
BOp ::= And
      | Or
      | Xor
F ::= Temp
   | Sat
   | Mult
   | Power
```

Funciones de observación

Para la función de observación utilizaré una de las funciones provistas por la biblioteca.

```
maybeIndex :: MArray arr cs e => Image arr cs e -> (Int, Int) ->
Maybe (Pixel cs e)
```

Esto permitirá acceder a los píxeles de la imagen.