Haskell - Práctica 2
Uso de módulos y directorios en GHCi

Presentación de imágenes en un navegador web

Para realizar esta tarea, usaremos el módulo `ImagenesSVG.hs` que contiene la implementación de una serie de funciones para manejar imágenes en formato SVG (Scalable Vector Graphics), que es una especificación para describir gráficos vectoriales bidimensionales en formato XML. En este caso las imágenes se definen como un nuevo tipo de dato, denominado `Imagen`. Algunas de las funciones son:

- `imprime :: Imagen -> IO ()`

  Función que muestra una imagen en el navegador.

- `encima :: Imagen -> Imagen -> Imagen`

  Función binaria que toma como entrada dos imágenes y devuelve una nueva imagen compuesta por las imágenes de entrada, una encima de la otra.

- `junto_a :: Imagen -> Imagen -> Imagen`

  Función binaria que toma como entrada dos imágenes y devuelve una nueva imagen compuesta por las imágenes de entrada, una al lado de la otra.

- `sobre :: Imagen -> Imagen -> Imagen`

  Función binaria que toma como entrada dos imágenes y devuelve una nueva imagen compuesta por las imágenes de entrada, una sobre otra.

- `giraH :: Imagen -> Imagen`

  Función que da la vuelta a la imagen sobre el eje horizontal.

- `giraV :: Imagen -> Imagen`

  Función que da la vuelta a la imagen sobre el eje vertical.

- `invierte_color :: Imagen -> Imagen`

  Función que invierte cada pixel de la imagen.

1. **Uso del módulo**

   Desde el terminal podemos ejecutar `GHCi` directamente con:

   ```
   > GHCi ImagenesSVG.hs
   ```

   o bien, dentro del intérprete hacer un `load` del módulo:

   ```
   Prelude> :l ImagenesSVG
   ```

En caso de que el módulo no se encuentre en el directorio correspondiente, podemos ver el directorio por defecto con la opción:

```
Prelude> :show paths
```

Si nos interesa modificar esta ruta, lo hacemos con:

```
Prelude> :cd otroDirectorio
```

En el mismo directorio en el que estemos trabajando, incluimos las imágenes que queremos visualizar, en formato `jpg` y los ficheros `showPic.html` o `refresh.html`.

Abrimos con el navegador `Firefox` o `Google Chrome` el fichero `showPic.html` o bien `refresh.html` si queremos que la imagen en el navegador se actualice automáticamente.

2. **Uso de las funciones** Podemos usar cualquiera de las funciones sobre la imagen `caballo` ya predefinida, siempre que dispongamos en el directorio del fichero `blk_horse_head.jpg`, usado para definir esa imagen. Por ejemplo, si ponemos en el intérprete:

   ```
   *ImagenesSVG> imprime (caballo `junto_a` (giraV caballo))
   ```

   en el navegador veremos:

3. **Definición de nuevas funciones**

   Ej.1 Crear un nuevo fichero `.hs` y definir en él una función

   ```
   cuatroImg ::  Imagen -> Imagen
   ```

   que tome como entrada una imagen y devuelva otra con la imagen cuadruplicada con el siguiente formato:

IMAGEN	IMAGEN EN ESPEJO Y CON COLORES INVERTIDOS
IMAGEN CON COLORES INVERTIDOS	IMAGEN EN ESPEJO

   Para poder usar las funciones definidas en `ImagenesSVG`, importar la librería añadiendo a la cabecera de nuestro fichero:

   ```
   import ImagenesSVG
   ```

   Puede probarse con la imagen predefinida `caballo` cargando nuestro fichero (con `:load`) y a continuación:

   ```
   >imprime (cuatroImg caballo)
   ```

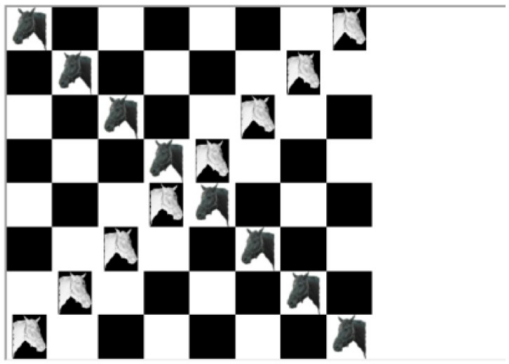
En el navegador debe aparecer algo parecido a esta imagen:

Ej.2 Usando las funciones de la librería y las imágenes `negro` y `blanco`[1], definir una función recursiva que, tomando como entrada dos números `n` y `m`, devuelva una imagen con un tablero tipo ajedrez que contenga las filas y columnas indicadas, respectivamente, por `n` y `m`. De esta forma, con la llamada:

```
>imprime (ajedrez 2 8)
```

obtendríamos en el navegador la imagen:

Ej.3 Finalmente, usar las imágenes `negro`, `blanco` y `caballoPequeño` para conseguir un tablero de ajedrez de 8×8, de manera que aparezca, en cada casilla de la diagonal blanca, un caballo negro mirando a la izquierda y, en cada casilla de la diagonal negra, un caballo blanco mirando hacia la derecha. Es decir:



[1]Necesitamos en el directorio los ficheros `black.jpg` y `white.jpg`.