

Until

Existe una instrucción que es la inversa de `while`

En decir, `while i < 0` es lo mismo que `until i >= 0`

`while` es mucho más utilizado que `until`, pero es bueno conocerlo por si lo encontramos en algún ejemplo.

Aviso sobre la instrucción gets

Durante el desarrollo de programas en general **no** utilizaremos `gets`. Al crear programas utilizaremos argumentos con `ARGV` o valores al azar con `rand`, o traeremos los datos desde archivos.

El uso de esta instrucción `gets` bloquea el programa hasta que el usuario ingresa un valor, esto complica el uso y evaluación de scripts. De todas formas podemos agradecer que `gets` nos ayudó a realizar ejercicios simples con ciclos.

Ejercicio: Validación de password

Objetivos

- Crear un diagrama de flujo con ciclos
- Crear un programa llamado `password_validation.rb` que valide la contraseña de un usuario

Podemos utilizar la misma idea de validación aprendida para impedir al usuario entrar hasta que ingrese la contraseña "password". Recuerda hacer el diagrama de flujo antes del código. Puedes utilizar el del capítulo anterior como base.

Solución

```
puts 'Ingresa su contraseña:'
password = gets.chomp

while password != 'password'
  puts 'La contraseña es incorrecta'
  puts 'Ingresa su contraseña:'
  password = gets.chomp
end

puts "La contraseña ingresado es correcto!"
```

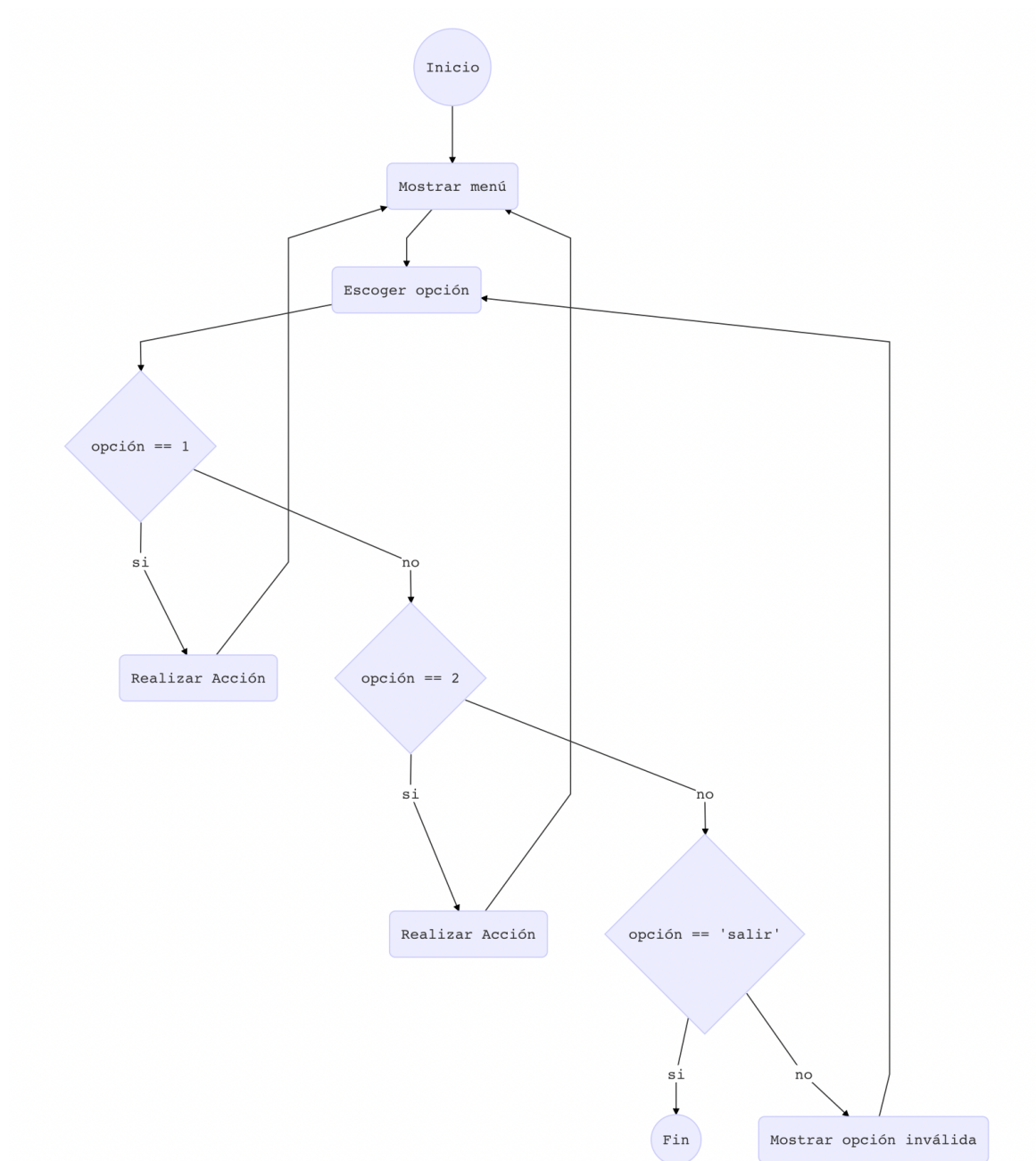
Ejercicio menú de opciones

Podemos implementar de forma sencilla un menú de opciones para el usuario. La lógica es similar a la de la validación de entrada.

- Se muestra un texto con opciones.
- El usuario tiene que ingresar una opción válida -> validación de entrada.
- Si el usuario ingresa la opción 1 mostramos un texto.
- Si el usuario ingresa la opción 2 mostramos otro texto.
- Si el usuario ingresa la opción "salir" terminamos el programa.

Desarrolla el diagrama de flujo de la solución y luego implementa el algoritmo. En el próximo capítulo revisaremos la solución

Diagrama de flujo del Menú



El código para un menú

```
opcion_menu = 'cualquier valor'
while opcion_menu != 'salir' && opcion_menu != 'Salir'
  # Mostrar menú
  puts 'Escoge una opción:'
  puts '-----'
  puts '1 - Acción 1'
  puts '2 - Acción 2'
  puts 'Escribe "salir" para terminar el programa'

  puts 'Ingresa una opción:'
  opcion_menu = gets.chomp

  if opcion_menu == '1'
    puts 'Realizando acción 1...'
  elsif opcion_menu == '2'
    puts 'Realizando acción 2...'
  elsif opcion_menu == 'salir' || opcion_menu == 'Salir'
    puts 'Saliendo...'
  else
    puts 'Opción inválida'
  end
end
```