Resolución: ejercicio 1

object ValidarNumero {  
 def validarNumero(numero: Int): Unit = {  
 if (numero < 0) {  
 throw new Exception("El número debe ser positivo.")  
 }  
 *println*("Número válido")  
 }  
  
 def main(args: Array[String]): Unit = {  
 try {  
 *validarNumero*(-5)  
 } catch {  
 case ex: Throwable => *println*("Se ha producido la excepción: " + ex.getMessage)  
 }  
 finally {  
 *println*("Fin de la validación.")  
 }  
 }  
}

#Ejercicio2

object ValidarCorreo {  
 def validarCorreo(correo: String): Unit = {  
 if (!correo.contains("@")) {  
 throw new Exception("El correo electrónico no es válido.")  
 }  
 *println*("Correo válido")  
 }  
  
 def main(args: Array[String]): Unit = {  
 try {  
 *validarCorreo*("usuario@ejemplo.com")  
 *validarCorreo*("usuarioejemplo.com") // Este lanzará una excepción  
 } catch {  
 case ex: Throwable => *println*("Se ha producido la excepción: " + ex.getMessage)  
 } finally {  
 *println*("Fin de la validación del correo.")  
 }  
 }  
}

Ejercicio 3:

object ValidarLongitud {  
 def validarLongitud(cadena: String): Unit = {  
 if (cadena.length < 5) {  
 throw new Exception("La cadena debe tener al menos 5 caracteres.")  
 }  
 *println*("Cadena válida")  
 }  
  
 def main(args: Array[String]): Unit = {  
 try {  
 // validarLongitud("Hola") // Este lanzará una excepción  
 *validarLongitud*("Hola Mundo") // Este imprimirá "Cadena válida"  
 } catch {  
 case ex: Throwable => *println*("Se ha producido la excepción: " + ex.getMessage)  
 } finally {  
 *println*("Fin de la validación de longitud.")  
 }  
 }  
}