

# Clase integradora

Laboratorio de procesamiento de información meteorológica - 2C 2023

Julia Mindlin

7 de septiembre de 2023

## Resumen

Ejercicios para trabajar en clase integrando lo aprendido en las practicas 1 y 2.

### 1. Ejercicio 1

Escribir un programa que almacene la cadena de caracteres contraseña en una variable, pregunte al usuario por la contraseña hasta que introduzca la contraseña correcta.

### 2. Ejercicio 2

Escribir un programa que muestre el eco de todo lo que el usuario introduzca hasta que el usuario escriba “salir” que terminará.

### 3. Ejercicio 3

Ante el siguiente problema:

La pizzería Bacci ofrece pizzas vegetarianas y no vegetarianas a sus clientes. Los ingredientes para cada tipo de pizza aparecen a continuación.

- Ingredientes vegetarianos: Pimiento y tofu.
- Ingredientes no vegetarianos: Peperoni, Jamón y Salmón.

Escribir un programa que pregunte al usuario si quiere una pizza vegetariana o no, y en función de su respuesta le muestre un menú con los ingredientes disponibles para que elija. Solo se puede elegir un ingrediente además de la mozzarella y el tomate que están en todas la pizzas. Al final se debe mostrar por pantalla si la pizza elegida es vegetariana o no y todos los ingredientes que lleva.

encontrar los errores en la siguiente solución:

```

1 # Preguntar al usuario si quiere una pizza vegetariana o no
2 opcion <- readline("Deseas una pizza vegetariana (V) o no vegetariana (NV)? ")
3
4 # Definir los ingredientes disponibles
5 ingredientes_veg <- c("Pimiento", "Tofu")
6 ingredientes_no_veg <- c("Salame", "Jamon", "Anchoas")
7
8 # Inicializar lista de ingredientes seleccionados
9 ingredientes_elegidos <- c("Mozzarella", "Tomate")
10
11 # Mostrar menu de ingredientes segun la eleccion del usuario
12 if (toupper(opcion) == "V") {
13   cat("Ingredientes disponibles para pizza vegetariana:\n")
14   for (i in 1:length(ingredientes_veg)) {
15     cat(i, "-", ingredientes_veg[i], "\n")
16   }
17 } else if (toupper(opcion) == "NV") {
18   cat("Ingredientes disponibles para pizza no veg:\n")
19   for (i in 1:length(ingredientes_no_veg)) {
20     cat(i, "-", ingredientes_no_veg[i], "\n")
21   }
22 } else {
23   cat("Opcion no valida. Debes seleccionar 'V' o 'NV'.")
24   quit(save = "no")
25 }
26
27 # Solicitar al usuario que elija un ingrediente adicional
28 elegir <- readline("Elige un ingrediente adicional (numero): ")
29
30 # Validar la eleccion del usuario
31 if (toupper(opcion) == "V" && elegir >= 1 && elegir <= length(ingredientes_veg)) {
32   ingrediente_elegido <- ingredientes_veg[elegir]
33   es_vegetariana <- TRUE
34 } else if (toupper(opcion) == "NV" && elegir >= 1 && elegir <= length(ingredientes_no_veg)) {
35   ingrediente_elegido <- ingredientes_no_veg[elegir]
36   es_vegetariana <- FALSE
37 } else {
38   cat("Opcion no valida. Debes seleccionar un namero valido de la lista.")
39   quit(save = "no")
40 }
41
42 # Agregar el ingrediente elegido a la lista de ingredientes seleccionados
43 ingredientes_elegidos <- c(ingredientes_elegidos, ingrediente_elegido)
44
45 # Mostrar los ingredientes de la pizza elegida
46 cat("\nHas elegido una pizza", ifelse(es_vegetariana, "vegetariana", "no vegetariana"), "
    con los siguientes ingredientes:\n")
47 cat(paste(ingredientes_elegidos, collapse = ", "), "\n")

```

Listing 1: Solucion de Ejercicio 3 con errores

#### 4. Ejercicio 4

Escribir un programa en el que se pregunte al usuario por una frase y una letra, y muestre por pantalla el número de veces que aparece la letra en la frase.

#### 5. Ejercicio 5

Escribir un programa que pida al usuario un número entero y muestre por pantalla si es un número primo o no.

#### 6. Ejercicio 6

Escribir un programa que pida al usuario un número entero y muestre por pantalla un triángulo rectángulo como el de más abajo.

```
1
3 1
5 3 1
7 5 3 1
9 7 5 3 1
```

```

1
2 # Definir la contraseña correcta
3 contraseña_correcta <- "micontrasena"
4
5 # Inicializar una variable para almacenar la contraseña ingresada por el usuario
6 contraseña_ingresada <- ""
7
8 # Configurar un bucle para solicitar la contraseña hasta que sea correcta
9 while (contraseña_ingresada != contraseña_correcta) {
10 # Solicitar al usuario que introduzca la contraseña
11   contraseña_ingresada <- readline("Introduce la contraseña: ")
12
13 # Verificar si la contraseña es correcta
14   if (contraseña_ingresada == contraseña_correcta) {
15     cat(" Contraseña correcta! Acceso concedido.\n")
16   } else {
17     cat("Contraseña incorrecta. Intentalo de nuevo.\n")
18   }
19 }

```

Listing 2: Solucion de Ejercicio 1

```

1
2 # Inicializar una variable para almacenar la entrada del usuario
3 entrada <- ""
4
5 # Configurar un bucle infinito
6 while (TRUE) {
7   # Solicitar al usuario que introduzca un texto
8   entrada <- readline("Introduce algo (escribe 'salir' para finalizar): ")
9
10  # Verificar si el usuario escribio "salir" y salir del bucle si es asi
11  if (tolower(entrada) == "salir") {
12    cat("Saliendo del programa.\n")
13    break
14  } else {
15    # Mostrar el eco de la entrada del usuario
16    cat("Eco:", entrada, "\n")
17  }
18 }

```

Listing 3: Solucion de Ejercicio 2

```
1
2 falta as.numeric al solicitar numero
```

Listing 4: Solucion de Ejercicio 3

```

1
2 # Solicitar al usuario que introduzca una frase
3 frase <- readline("Introduce una frase: ")
4
5 # Solicitar al usuario que introduzca una letra
6 letra <- readline("Introduce una letra: ")
7
8 # Inicializar el contador
9 contador <- 0
10
11 # Iterar a traves de la frase para contar las ocurrencias de la letra
12 for (i in strsplit(frase, NULL)[[1]]) {
13   if (i == letra) {
14     contador <- contador + 1
15   }
16 }
17
18 # Mostrar el resultado
19 resultado <- sprintf("La letra '%s' aparece %2i veces en la frase '%s'.\n", letra,
20   contador, frase)
21 print(resultado)

```

Listing 5: Solucion de Ejercicio 4

```

1
2 # Solicitar al usuario que introduzca un numero entero positivo mayor que 2
3 n <- as.integer(readline("Introduce un numero entero positivo mayor que 2: "))
4
5 # Inicializar una variable para verificar si n es primo
6 es_primo <- TRUE
7
8 # Comprobar si n es divisible por algun numero entre 2 y n-1
9 for (i in 2:(n-1)) {
10   if (n %% i == 0) {
11     es_primo <- FALSE
12     break
13   }
14 }
15
16 # Mostrar el resultado
17 if (es_primo && n > 1) {
18   cat(n, "es primo\n")
19 } else {
20   cat(n, "no es primo\n")
21 }

```

Listing 6: Solucion de Ejercicio 5



```

1
2
3 # Solicitar al usuario un n mero entero
4 n <- as.integer(readline("Introduce un n mero entero: "))
5
6 # Verificar si el n mero es positivo
7 if (n <= 0) {
8   cat("Por favor, introduce un n mero entero positivo.")
9 } else {
10  # Generar el tri ngulo rect ngulo
11  for (i in seq(1,n,2)) {
12    for (j in seq(i,1,-2)) {
13      cat(j," ")
14    }
15    cat("\n")
16  }
17 }

```

Listing 7: Solucion de Ejercicio 5