2020

Coloquio RPA - Teléfono

PSEUDOCÓDIGO

DE ROSA, MARÍA ALUMINÉ - MONTES, DIEGO NICOLÁS
FAI 2186 FAI 2152

Referencias:

Con este color se indican las funciones de string(java) o texto(pseudocódigo)

Ctrl + click en los enlaces para ir a las funciones

```
ALGORITMO telefono() RETORNA Ø
(* Mostrará el menú reiteradas veces hasta que el usuario ingrese la opción
 finalizar *)
    ENTERO opcion
    TEXTO telefono, error, segundoTelefono, respuesta, cadenaNumeros
    REPETIR
        mostrarMenu()
        ESCRIBIR("Elija una opción")
        LEER(opcion)
        SEGUN opcion HACER

            caso 1:
                ESCRIBIR("Ingrese el número de teléfono a validar")
                LEER(telefono)
                SI (validarTelefono(telefono)) ENTONCES
                    respuesta ← “El número de teléfono es válido”
                SINO
                    respuesta ← “El número de teléfono no es válido”
                FIN SI

            caso 2:
                ESCRIBIR("Ingrese el número de teléfono a validar")
                LEER(telefono)
                SI (validarTelefono(telefono)) ENTONCES
                    respuesta ← “El número de teléfono es válido”
                SINO
                    error ← errorTelefono(telefono)
                    respuesta ←"El número de teléfono no es válido: " , error
                FIN SI

            caso 3:
                ESCRIBIR("Ingrese un número de teléfono")
                LEER(telefono)
                SI (validarTelefono(telefono)) ENTONCES
                    ESCRIBIR("Ingrese otro número de teléfono")
                    LEER(segundoTelefono)
                    SI (validarTelefono(segundoTelefono)) ENTONCES
                        SI (NOT (verificarEsMenor(telefono,segundoTelefono))) ENTONCES
                            SI (NOT (verificarSonIguales(telefono, segundoTelefono)))ENTONCES
                                respuesta ← "El primer teléfono es mayor al segundo"
                            SINO
                                respuesta ← "El primer teléfono es igual al segundo"
```

```
                    FIN SI
                SINO
                    respuesta ←"El primer teléfono es menor al segundo"
                FIN SI
            SINO
                error ← errorTelefono(segundoTelefono)
                respuesta ←"El segundo número de teléfono no es válido:  " , error
            FIN SI
        SINO
            error ← errorTelefono(telefono)
            respuesta ←"El primer número de teléfono no es válido:  ", error
        FIN SI

caso 4:
        ESCRIBIR("Ingrese un número de teléfono")
        LEER(telefono)
        SI (validarTelefono(telefono)) ENTONCES
            ESCRIBIR("Ingrese otro número de teléfono")
            LEER(segundoTelefono)
            SI (validarTelefono(segundoTelefono)) ENTONCES
                SI (verificarSonIguales(telefono,segundoTelefono)) ENTONCES
                    respuesta ← "El primer teléfono es igual al segundo"
                SINO
                    SI (verificarEsMenor(telefono, segundoTelefono))ENTONCES
                        respuesta ← "El primer teléfono es menor al segundo"
                    SINO
                        respuesta ← "El primer teléfono es mayor al segundo"
                    FIN SI
                FIN SI
            SINO
                error ← errorTelefono(segundoTelefono)
                respuesta ←"El segundo número de teléfono no es válido: ", error
            FIN SI
        SINO
            error ← errorTelefono(telefono)
            respuesta ←"El primer número de teléfono no es válido: " , error
        FIN SI

caso 5:
        respuesta ←  devolverMayor()

caso 6:
```

```
            ESCRIBIR("Ingrese un número de teléfono")
            LEER(telefono)
            SI (validarTelefono(telefono)) ENTONCES
            |       respuesta ← obtenerOcurrencias(telefono)
            SINO
            |       respuesta ← "El número ingresado no es válido. Por favor intente nuevamente."
            FIN SI

caso 7:
            ESCRIBIR("Ingrese el número de teléfono a incrementar")
            LEER(telefono)
            SI (validarTelefono(telefono)) ENTONCES
            |       respuesta ← "El número de teléfono incrementado es: ",  incrementarTelefono(telefono)
            SINO
            |       respuesta ← "El número ingresado no es válido. Por favor intente nuevamente."
            FIN SI

caso 8:
            ESCRIBIR("Ingrese una cadena de números")
            LEER(cadenaNumeros)
            SI (validarCadena(cadenaNumeros)) ENTONCES
            |       respuesta ← "El número menor es: " , devolverMenor(cadenaNumeros)
            SINO
            |       respuesta ← "La cadena ingresada no es válida. Por favor intente nuevamente."
            FIN SI

caso 9:
            ESCRIBIR("Ingrese un número de teléfono")
            LEER(telefono)
            SI (validarTelefono(telefono)) ENTONCES
            |       ESCRIBIR("Ingrese una cadena de números")
            |       LEER(cadenaNumeros)
            |       SI (validarCadena(cadenaNumeros)) ENTONCES
            |       |       respuesta ← obtenerOcurrenciasCadena(telefono, cadenaNumeros)
            |       SINO
            |       |       respuesta ← "La cadena ingresada no es válida. Por favor intente nuevamente."
            |       FIN SI
            SINO
            |       respuesta ← "El número ingresado no es válido. Por favor intente nuevamente."
            FIN SI

caso 10:
```

```
                respuesta ← “Se ha finalizado el programa”

            ∀o:
                respuesta ← “La opción ingresada no corresponde a una opción válida, por favor inténtelo nuevamente”
        FIN SEGÚN
        ESCRIBIR(respuesta)
    HASTA (opcion = 10)
FIN ALGORITMO teléfono

MODULO validarTelefono(TEXTO telefono) RETORNA LOGICO
    (*
      Modulo 1.
      Verifica si un telefono tiene una estructura válida.
      Estructura válida: CCCC-NNNNNNNNN.
    *)
    LOGICO valido ← FALSO
    TEXTO caracteristica, numero

    SI (longitud(telefono) = 14) ENTONCES
        SI (posicion(telefono,4) = '-') ENTONCES
            caracteristica ← cortarCaracteristica(telefono)
            numero ← cortarTelefono(telefono)
            SI (isNumeric(caracteristica) AND isNumeric(telefono)) ENTONCES
                valido ← VERDADERO
            FIN SI
        FIN SI
    FIN SI
    RETORNA valido
FIN MODULO validarTelefono
```

```
MODULO errorTelefono(TEXTO telefono) RETORNA TEXTO
    (*
       Modulo 2.
       Determina por qué un teléfono no es válido.
    *)
    TEXTO caracteristica, numero
    TEXTO error ← ""

    SI (longitud(telefono) = 14) ENTONCES
        SI (posicion(telefono,4) = '-') ENTONCES
            caracteristica ← cortarCaracteristica(telefono)
            numero ← cortarTelefono(telefono)
            SI (NOT (isNumeric(caracteristica))) ENTONCES
                error ← error , “La característica ingresada no es un número” (* con salto *)
            FIN SI
            SI (NOT (isNumeric(numero))) ENTONCES
                error ← error , “El teléfono ingresado no es un número” (* con salto *)
            FIN SI
        SINO
            error ← error , “El formato es incorrecto” (* con salto *)
        FIN SI
    SINO
        error ← error , "La cantidad de dígitos es incorrecta” (* con salto *)
    FIN SI
    RETORNA error
FIN MODULO errorTelefono
```

```
MODULO verificarEsMenor(TEXTO telefonoUno, TEXTO telefonoDos) RETORNA LOGICO
    (*
       Modulo 3.
       Verifica si el primer número(sin considerar la característica), es
       menor que el segundo.
    *)
    LOGICO esMenor ← FALSO
    TEXTO telefonoUno ← cortarTelefono(telefonoUno)
    TEXTO telefonoDos ← cortarTelefono(telefonoDos)

    SI (convertirAEntero(telefonoUno) < convertirAEntero(telefonoDos)) ENTONCES
        esMenor ← VERDADERO
    FIN SI
    RETORNA esMenor
FIN MODULO verificarEsMenor

MODULO verificarSonIguales(TEXTO telefonoUno, TEXTO telefonoDos) RETORNA LOGICO
    (*
       Modulo 4.
       Verifica si ambos números son iguales.
    *)
    LOGICO sonIguales ← FALSO

    SI (igual(telefonoUno,telefonoDos)) ENTONCES
        sonIguales ← VERDADERO
    FIN SI
    RETORNA sonIguales
FIN MODULO verificarSonIguales
```

```
MODULO devolverMayor() RETORNA TEXTO
    (*
       MODULO 5
       Solicita al usuario una secuencia de números de teléfono mientras
       este lo desee. Finalmente devuelve el mayor.
    *)
    LOGICO continuar
    TEXTO telefonoMayor ← "No ha ingresado ningún teléfono válido"
    TEXTO telefono
    ENTERO numero
    ENTERO numeroMayor ← 0
    REPETIR
        ESCRIBIR("Ingrese un número de teléfono")
        LEER(telefono)
        SI (validarTelefono(telefono)) ENTONCES
            numero ← convertirAEntero(cortarTelefono(telefono))
            SI (numero>numeroMayor) ENTONCES
                numeroMayor ← numero
                telefonoMayor ← "El número mayor es: " , telefono
            FIN SI
        SINO
            ESCRIBIR("El número de teléfono no es válido")
        FIN SI
        ESCRIBIR("Desea continuar (VERDADERO:si|FALSO:no)?")
        LEER(continuar)
    HASTA (continuar = FALSO)
    RETORNA telefonoMayor
FIN MODULO devolverMayor
```

```
MODULO obtenerOcurrencias(TEXTO telefono) RETORNA TEXTO
    (*
       Modulo 6.
       Recibe por parámetro un teléfono. Luego solicita otros hasta que el
       usuario desee. Finalmente indica si el teléfono recibido por parámetro está
       duplicado. De ser así, también indica cuántas veces.
    *)
    LOGICO continuar
    TEXTO resultadoOcurrencias, telefonoNuevo
    ENTERO cantidadOcurrencias ← 0
    REPETIR
        ESCRIBIR("Ingrese un nuevo número de teléfono")
        LEER(telefonoNuevo)
        SI (validarTelefono(telefonoNuevo)) ENTONCES
            SI (verificarSonIguales(telefono,telefonoNuevo)) ENTONCES
                cantidadOcurrencias ← cantidadOcurrencias + 1
            FIN SI
        SINO
            ESCRIBIR("El número de teléfono no es válido.”)
        FIN SI
        ESCRIBIR("Desea continuar (VERDADERO:si|FALSO:no)?")
        LEER(continuar)
    HASTA (continuar = FALSO)
    resultadoOcurrencias ← obtenerTextoOcurrencias(cantidadOcurrencias)
    RETORNA resultadoOcurrencias
FIN MODULO obtenerOcurrencias
```

```
MODULO incrementarTelefono (TEXTO telefono) RETORNA TEXTO
    (*
      Modulo 7.
      Se incrementa el número telefónico en 1 unidad.
     *)
    TEXTO telefonoIncrementado
    ENTERO numero ← convertirAEntero(cortarTelefono(telefono))
    (*
      Si el número es 999999999, no se puede incrementar uno más, por lo que se transforma en 000000000
      Utilizamos dos métodos diferentes con el propósito de mostrar las diferentes maneras de hacerlo.
      El método reemplazarTelefono es nuestra traducción a pseudocódigo del método de string cadena.replace()
    *)
    SI (numero = 999999999) ENTONCES
        telefonoIncrementado ← reemplazarTelefono(convertirEnteroATexto(numero),"000000000")
    SINO
        numero ← numero + 1
        telefonoIncrementado ← cortarCaracteristica(telefono), "-", numero
    FIN SI
    RETORNA telefonoIncrementado
FIN MODULO incrementarTelefono
```

```
MODULO devolverMenor(TEXTO cadena) RETORNA TEXTO
    (*
       Modulo 8.
       Recibe por parámetro una cadena de telefonos contenidos en un texto.
       Retorna cuál de ellos es el menor.
    *)
    TEXTO telefonoMenor ← "0299-999999999"
    TEXTO telefono
    ENTERO cantTelefonos ← (longitud(cadena)/14)
    ENTERO posicionInicial ← 0
    ENTERO posicionFinal ← 14
    ENTERO i ← 0
    (*
       Obtenemos la cantidad de teléfonos que existen, y luego, a través de subcadena,
       obtenemos cada uno de ellos.
    *)
    MIENTRAS (i<cantTelefonos) HACER
        telefono ← subcadena(cadena,posicionInicial,posicionFinal)
        SI (verificarEsMenor(telefono,telefonoMenor)) ENTONCES
            telefonoMenor ← telefono
        FIN SI
        posicionInicial ← posicionInicial + 14
        posicionFinal ← posicionFinal + 14
        i ← i+1
    FIN MIENTRAS
    RETORNA telefonoMenor
FIN MODULO devolverMenor
```

```
MODULO obtenerOcurrenciasCadena (TEXTO telefono, TEXTO cadenaNumeros) RETORNA TEXTO
    (*
       Modulo 9.
       Dado un número de telefono determinado "A", y una secuencia de teléfonos
       recibidos por parámetro en un solo texto, devueve la
       cantidad de ocurrencias de A en la secuencia.

    *)
    TEXTO resultadoOcurrencias, telefonoNuevo
    ENTERO cantidadOcurrencias ← 0
    ENTERO cantTelefonos ← (longitud(cadenaNumeros)/14)
    ENTERO posicionInicial ← 0
    ENTERO posicionFinal ← 14
    ENTERO i ← 0
    MIENTRAS (i<cantTelefonos) HACER
        telefonoNuevo ← subcadena(cadenaNumeros,posicionInicial,posicionFinal)
        SI (verificarSonIguales(telefono,telefonoNuevo)) ENTONCES
            cantidadOcurrencias ← cantidadOcurrencias + 1
        FIN SI
        posicionInicial ← posicionInicial + 14
        posicionFinal ← posicionFinal + 14
        i ← i+1
    FIN MIENTRAS
    resultadoOcurrencias ← obtenerTextoOcurrencias(cantidadOcurrencias)
    RETORNA resultadoOcurrencias
FIN MODULO obtenerOcurrenciasCadena
```

```
MODULO mostrarMenu() RETORNA Ø
    (*
     Modulo 10.
     Muestra las opciones de un Menú.
    *)
      ESCRIBIR("-----------------------------------------------------------------------------------------------------------------------------------------------------------")(*con salto*)
      ESCRIBIR("MENÚ")(*con salto*)
      ESCRIBIR("")(*con salto*)
      ESCRIBIR("-----------------------------------------------------------------------------------------------------------------------------------------------------------")(*con salto*)
      ESCRIBIR("1-Verificar si un número de teléfono es válido")(*con salto*)
      ESCRIBIR("2-Determinar porque un número de teléfono no es válido")(*con salto*)
      ESCRIBIR("3-Verificar si un número de teléfono es mayor que otro ")(*con salto*)
      ESCRIBIR("4-Verificar si un número de teléfono es igual a otro")(*con salto*)
      ESCRIBIR("5-Ingresar una secuencia de números de teléfonos y obtener aquel número más grande")(*con salto*)
      ESCRIBIR("6-Ingresar una secuencia de números de teléfonos y un número de teléfono determinado A, y hallar cantidad de ocurrencias de A.")(*con
      salto*)
      ESCRIBIR("7-Incrementar un número de teléfono en una unidad")(*con salto*)
      ESCRIBIR("8-Dada una secuencia de números de teléfonos (recibido por parámetro en un texto) obtener aquel número más grande")(*con salto*)
      ESCRIBIR("9-Dada una secuencia de números de teléfonos (recibido por parámetro en un texto) y un número de teléfono determinado A, hallar
      cantidad de ocurrencias de A en la secuencia")(*con salto*)
      ESCRIBIR("10-Terminar ")(*con salto*)
      ESCRIBIR("")(*con salto*)
      ESCRIBIR("-----------------------------------------------------------------------------------------------------------------------------------------------------------")(*con salto*)
      ESCRIBIR("")(*con salto*)
FIN MODULO mostrarMenu


MODULO cortarTelefono (TEXTO telefono) RETORNA TEXTO
    (*
       Módulo adicional.
       Este módulo retorna solo el número del teléfono, sin tener en cuenta su característica.
       Ejemplo: Si recibe "0299-154567890", devuelve "154567890".
    *)
    TEXTO numTelefono
    numTelefono ← subcadena(telefono,5,14)
    RETORNA numTelefono
FIN MODULO cortarTelefono
```

```
MODULO cortarCaracteristica (TEXTO telefono) RETORNA TEXTO
    (*
       Módulo adicional.
       Este módulo retorna solo la característica del teléfono, sin tener en cuenta el resto del número.
       Ejemplo: Si recibe "0299-154567890", devuelve "0299".
    *)
    TEXTO caracteristica
    caracteristica ← subcadena(caracteristica,0,4)
    RETORNA caracteristica
FIN MODULO cortarCaracteristica

MODULO validarCadena (TEXTO cadena) RETORNA LOGICO
    (*
       Módulo adicional.
       Este módulo valida que la cadena de teléfonos ingresados por el usuario sea válida.
    *)
    LOGICO valido ← VERDADERO
    ENTERO cantTelefonos
    ENTERO i ← 0
    ENTERO posicionInicial ← 0
    ENTERO posicionFinal ← 14
    TEXTO telefono

    SI ((longitud(cadena) MOD 14) = 0) ENTONCES
        cantTelefonos ← (longitud(cadena)/14)
        MIENTRAS (i<cantTelefonos AND valido) HACER
            telefono ← subcadena(cadena,posicionInicial,posicionFinal)
            SI (NOT(validarTelefono(telefono))) ENTONCES
                valido ← FALSO
            FIN SI
            posicionInicial ← posicionInicial + 14
            posicionFinal ← posicionFinal + 14
            i ← i+1
        FIN MIENTRAS
    SINO
        valido ← FALSO
    FIN SI
    RETORNA valido
FIN MODULO validarCadena
```

```
MODULO obtenerTextoOcurrencias (ENTERO cantidadOcurrencias) RETORNA TEXTO
    (*
      Módulo adicional.
      Convierte las ocurrencias, según la cantidad que existan, a texto.
    *)
    TEXTO textoOcurrencias
    SEGUN cantidadOcurrencias HACER
        caso 0:
          textoOcurrencias ← "No hay ocurrencias"
        caso 1:
          textoOcurrencias ← "Existe una ocurrencia"
        ∀o:
          textoOcurrencias ← "Existen ", cantidadOcurrencias , " ocurrencias"
    FIN SEGUN
    RETORNA textoOcurrencias
FIN MODULO obtenerTextoOcurrencias

MODULO isNumeric (TEXTO cadena) RETORNA LOGICO
    (*
      Módulo adicional.
      Módulo que valida que el parametro recibido sea de tipo numérico.
      Utilizamos un try/catch, debido a que lo hemos utilizado en otros proyectos.
      Este nos permite intentar convertir el texto a entero y si no lo logra arroja una excepción,
      permitiendonos darnos cuenta de que dicho texto no contiene números.
    *)
    LOGICO isNumeric
    INTENTAR
        convertirAEntero(cadena)
        isNumeric ← VERDADERO
    CAPTURAR (error)
        isNumeric ← FALSO
    FIN INTENTAR
    RETORNA isNumeric
FIN MODULO isNumeric
```

Código .JAVA:

Link a repositorio de github: https://github.com/aluminedr/Telefono | **Link para clonar el repositorio:** https://github.com/aluminedr/Telefono.git