

Hito de programación 1er trimestre

Mario Alajarín Escobar 1ºDAM

Fase 1:

Un algoritmo es un conjunto de instrucciones destinadas a resolver un problema o a desarrollar una actividad.

En mi caso hay que desarrollar dos actividades, el proceso de registro o inicio de sesión de un usuario y la compra de unos productos.

He empezado haciendo pseudocódigo con "PSeInt" y lo he desarrollado a base de condicionales.

Aquí tenemos la primera parte del pseudocódigo:

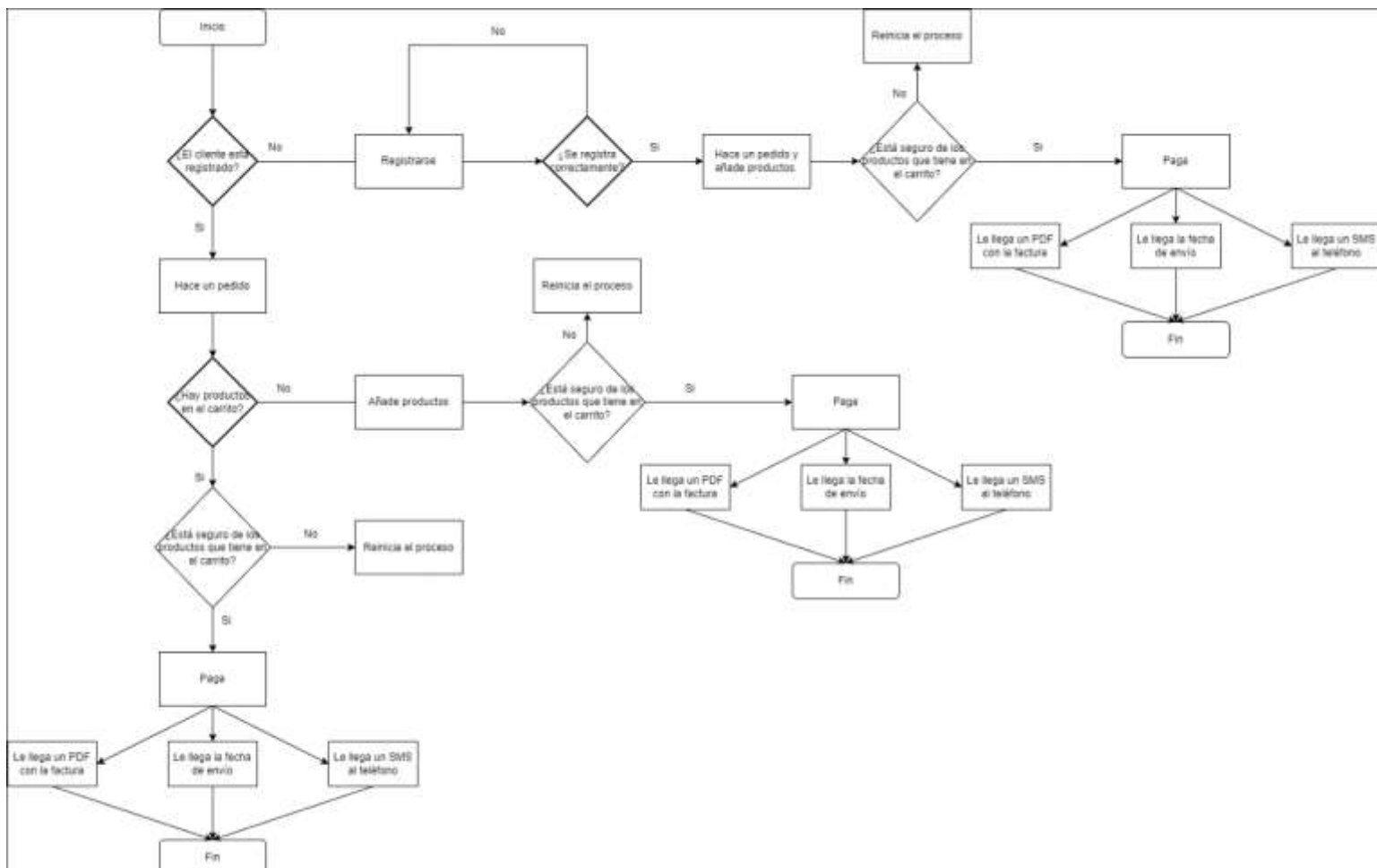
```
1  Algoritmo HitoProgramaciónMarioA
2      Escribir '¿Estás registrado?'
3      Leer registrar
4      Si registrar=='si' Entonces
5          Escribir '¿Hay productos en el carrito?'
6          Leer carrito
7          Si carrito=='si' Entonces
8              Escribir '¿Estás seguro de los productos que hay en el carrito?'
9              Leer seguro
10             Si seguro=='si' Entonces
11                 Escribir 'Paga'
12                 Escribir 'Le llega un PDF'
13                 Escribir 'Le llega la fecha de envío'
14                 Escribir 'Le llega un SMS al teléfono'
15             SiNo
16                 Escribir 'Reinicia el proceso'
17             FinSi
18         SiNo
19             Escribir 'añade productos'
20             Escribir '¿Estás seguro de los productos que hay en el carrito?'
21             Leer seguro
22             Si seguro=='si' Entonces
23                 Escribir 'Paga'
24                 Escribir 'Le llega un PDF'
25                 Escribir 'Le llega la fecha de envío'
26                 Escribir 'Le llega un SMS al teléfono'
27             SiNo
28                 Escribir 'Reinicia el proceso'
29             FinSi
30         FinSi
31     SiNo
```

Creamos el algoritmo y hacemos la primera pregunta (“¿Estás registrado?”) y a partir de ahí empezamos a desarrollar el código a base de condicionales.

Esta es la segunda parte del código.

```
32      Escribir 'Se registra correctamente'
33      Leer registroCorrecto
34      Si registroCorrecto=='si' Entonces
35          Escribir 'Hace un pedido y añade productos'
36          Escribir '¿Estás seguro de los productos que hay en el carrito?'
37          Leer seguro
38          Si seguro=='si' Entonces
39              Escribir 'Paga'
40              Escribir 'Le llega un PDF'
41              Escribir 'Le llega la fecha de envío'
42              Escribir 'Le llega un SMS al teléfono'
43          SiNo
44              Escribir 'Reinicia el proceso'
45          FinSi
46      SiNo
47          Mientras registroCorrecto=='no' Hacer
48              Escribir 'Se registra correctamente'
49              Leer registroCorrecto
50              Si registroCorrecto=="si" Entonces
51                  Escribir 'Hace un pedido y añade productos'
52                  Escribir '¿Estás seguro de los productos que hay en el carrito?'
53                  Leer seguro
54                  Si seguro=='si' Entonces
55                      Escribir 'Paga'
56                      Escribir 'Le llega un PDF'
57                      Escribir 'Le llega la fecha de envío'
58                      Escribir 'Le llega un SMS al teléfono'
59                  SiNo
60                      Escribir 'Reinicia el proceso'
61                  FinSi
62              Fin Si
63          FinMientras
64      FinSi
65  FinSi
66 FinAlgoritmo
67
```

El pseudocódigo es un poco lioso, es mejor verlo con un diagrama de flujo:



Aquí se explican todos los casos que puede haber en esta actividad (voy a subir la imagen junto al código y el PSeInt a GitHub)

En PDF la imagen se ve borrosa, sin embargo, en Word se ve bien, voy a subir la imagen junto al hito por si acaso.

Fase 2:

Explicación del código paso por paso (he hecho dos archivos.py, uno con comentarios y otro sin comentarios, voy a utilizar el que no tiene comentarios para explicarlo aquí, pero están subidos ambos a GitHub):

Empezamos por el principio:

```
1 import random
2 import pandas as pd
3 from datetime import date
```

Importamos estos módulos que serán importantes más adelante.

```
5 Finicio = date.today()
6 Ffinal = date(2022, 12, 30)
7
8 fechaAleatoria = Finicio + (Ffinal - Finicio) * random.random()
9
```

Con “date” del módulo “datetime” he hecho una fecha de inicio que muestre la fecha de hoy y una fecha final que es fija, todo esto lo he creado para que escoja una fecha aleatoria dentro de ese rango (esto va a ser esencial para que a la hora del pago de los productos nos digan una fecha de envío)

```
10 productos=["manzanas","peras","melon","fresas","platanos"]
11 precio=[0.20,0.15,3,0.10,0.25]
12 stock=[540,245,120,765,400]
```

Creamos tres listas con el nombre, el precio y el stock de los productos.

```
15 class Cliente:
16     def preguntaRegistro(self):
17         pregRegistro=input("¿Está usted registrado? (si/no): ")
18         if pregRegistro.lower()=="si":
19             self.nombreCliente=input("Ingrese el nombre de usuario: ")
20             self.contraseñaCliente=input("Ingrese la contraseña: ")
21             print()
22             print("Ha iniciado sesión satisfactoriamente.")
23             print()
24             print("-----")
25             print()
26         elif pregRegistro.lower()=="no":
27             self.nombreCliente=input("Ingrese el nombre de usuario: ")
28             self.correoCliente=input("Ingrese el correo electrónico: ")
29             self.contraseñaCliente=input("Ingrese una contraseña: ")
30             self.telefono=input("Ingrese su número de teléfono: ")
31             self.dni=input("Ingrese el DNI: ")
32             print()
33             print("-----")
34             print()
```

Creamos la clase “Cliente” y hacemos la función o método llamada “preguntaRegistro” que sirve para que nos pregunte si estamos registrados o no, si estamos registrados entonces iniciamos sesión poniendo nuestro nombre de usuario

y contraseña, sin embargo, si la respuesta es que no estamos registrados entonces tendremos que registrarnos poniendo el nombre de usuario, el correo electrónico, una contraseña, el número de teléfono y el DNI (los print() y los print("-----") son decorativos).

```

36         while "@" not in self.correoCliente:
37             print("Hay un error en el registro, pruebe de nuevo.")
38             print()
39             print("-----")
40             print()
41             correoClienteNuevo=input("Ingrese el correo electrónico de nuevo: ")
42             if correoClienteNuevo.__contains__("@"):
43                 datos=[self.nombreCliente,correoClienteNuevo,self.contraseñaCliente,self.telefono,self.dni]
44                 print("El usuario se ha registrado correctamente.")
45                 print()
46                 df1=pd.DataFrame(datos, index=["Usuario","e-mail","contraseña","Tlf","DNI"])
47                 df1.set_axis(["Datos"], axis=1)
48                 print("Datos: \n%s" % df1)
49                 print()
50                 print("-----")
51                 print()
52                 break
53             if self.correoCliente.__contains__("@"):
54                 datos=[self.nombreCliente,self.correoCliente,self.contraseñaCliente,self.telefono,self.dni]
55                 print("El usuario se ha registrado correctamente.")
56                 print()
57                 df1=pd.DataFrame(datos, index=["Usuario","e-mail","contraseña","Tlf","DNI"])
58                 df1.set_axis(["Datos"], axis=1)
59                 print("Datos: \n%s" % df1)
60                 print()
61                 print("-----")
62                 print()

```

Como no existen correos electrónicos sin "@" entonces he hecho un bucle en el que si no hay "@" en el correo, nos lo preguntará indefinidamente hasta que lo ingresemos correctamente, obviamente, si escribimos un correo correcto después de haber ingresado uno erróneo el correo que está bien sobrescribirá al que está mal, y para hacer una pequeña interfaz gráfica he utilizado el módulo "Pandas" ya que así nos mostrará el usuario, el correo electrónico, la contraseña, el teléfono y el DNI en forma de esquema (las líneas 48 y 59 sirven para poner un título al esquema, en este caso, "Datos").

```

66 cliente1=Cliente()
67 cliente1.preguntaRegistro()

```

Instanciamos a la clase Cliente y llamamos al método preguntaRegistro().

```

69 print("Disponemos de los siguientes productos: ")
70 print()
71 dict={"NombreProducto": productos, "Precio": precio, "Stock": stock}
72 df2=pd.DataFrame(dict, index=["Producto 1","Producto 2","Producto 3","Producto 4","Producto 5"])

```

Aquí entran en juego las tres listas creadas al principio, ya que gracias a "Pandas" podremos hacer un esquema con el nombre del producto, el precio y el stock como columnas y Producto 1...Producto 5 como filas.

```

74 print(df2)
75 print()
76 print("-----")
77 print()
78

```

Imprimimos el esquema.

```

79 class Producto(Cliente):
80     def __init__(self) -> None:
81         self.nombreProducto=input("Ingrese el nombre del producto deseado: ")
82         self.cantidadProducto=input("Ingrese el número de productos: ")
83

```

Creamos la clase Producto que hereda de Cliente, también hacemos un constructor, así como dos variables que nos pregunten el nombre del producto deseado y el número de productos.

```

84 def productos(self):
85     j=len(productos)
86     i=0

```

Creamos el método productos para el cálculo, la variable "j" para guardar el número de elementos de la lista productos y la variable "i", un totalizador.

```

87 while i < j:
88     if productos[i]==self.nombreProducto:
89         if int(self.cantidadProducto) > stock[i]:
90             print("No es posible, el stock máximo es "+str(stock[i]), ". vuelva a iniciar el proceso.")
91         else:
92             comprobar=input("En su carrito tiene {self.cantidadProducto} {productos[i]}, ¿está seguro de su compra?: ")
93             print()
94             if comprobar.lower()=="si":
95                 self.paisCliente=input("Ingrese el país desde el que hace el pedido: ")
96                 precioSinIVA=float(self.cantidadProducto)*precio[i]
97                 stockRestante=stock[i]-int(self.cantidadProducto)
98                 print("El precio sin IVA es {precioSinIVA}€, quedan {stockRestante} unidades en stock.")

```

Hacemos un bucle en el que mientras "i" sea menor que "j" ocurrirán una serie de procesos, a partir de ahí hacemos un condicional para que haga una acción si el nombre del producto está en la lista, en el caso de que esté en la lista continuará, sino se parará el proceso.

Si se cumple el condicional anterior haremos otro condicional en el que si la cantidad de producto deseada es mayor que el stock nos diga que no es posible, en el caso de que esa condición no se cumpla haremos una variable llamada "comprobar" en la que nos preguntará si estamos seguros de los productos que tenemos en el carrito.

Si la respuesta es sí entonces nos pedirá el país desde el que hacemos el pedido y calcularemos el precio sin el IVA y el stock restante y nos lo imprimirá.

```
99         if self.paisCliente=="España":
100             precioConIVA=precioSinIVA*1.21
101             print(f"El total con el IVA es {precioConIVA:.2f}€.")
102             print()
103             print(f"Se le ha mandado un PDF al correo electrónico.")
104             print(f"Se le ha mandado un SMS al número de teléfono con el cual puede ver el seguimiento.")
105             print(f"La fecha de envío es {fechaAleatoria} (Año-Mes-Día)")
106         elif self.paisCliente!="España":
107             precioConIVA=precioSinIVA*(random.uniform(1,2))
108             print(f"El total con el IVA añadido de su respectivo país es {precioConIVA:.2f}€.")
109             print()
110             print(f"Se le ha mandado un PDF al correo electrónico.")
111             print(f"Se le ha mandado un SMS al número de teléfono con el cual puede ver el seguimiento.")
112             print(f"La fecha de envío es {fechaAleatoria} (Año-Mes-Día)")
```

Tenemos un último condicional, ya que si el país del cliente es España, entonces nos calculará el precio con IVA multiplicando el precio sin IVA por 1.21, y después de eso nos lo imprimirá, aparte de decirnos que nos han mandado un PDF por el correo electrónico, un SMS al número de teléfono del cliente para ver el seguimiento y la fecha de envío gracias a la variable "fechaAleatoria" del principio, en el caso de que el cliente no sea de España entonces le calculará un precio con IVA multiplicando el precio sin IVA por un número aleatorio entre 1 y 2 con dos decimales.

Gracias al ":.2f" el "precioConIVA" nos saldrá únicamente con dos decimales.

```
113         else:
114             print("Vuelva a iniciar el proceso para seleccionar los productos deseados.")
```

Ese "else" se ejecuta si le decimos que no estamos seguros de la compra.

```
116         i=i+1
```

Creamos una suma usando el totalizador dentro del bucle para que todo se pueda ejecutar.

```
118         producto1=Producto()
119         producto1.productos()
```

Instanciamos la clase Producto y llamamos al método productos.

Fase 3:

En sí he utilizado una mezcla entre programación estructurada, programación orientada a objetos y programación funcional ya que he usado clases, funciones... aunque lo que más abunda son los condicionales y los bucles ya que ha sido la manera en la que he hecho esta aplicación.

En algunos casos he hecho el código a prueba de error, como cuando el cliente se registra o inicia sesión y en otros casos (que me gustaría haberlo hecho) no, ya que no he sabido cómo hacerlo, este es el caso de todas aquellas opciones en las cuales he puesto que si da error que vuelva a iniciar el proceso desde el principio, es probable que existan módulos que puedan hacerlo automáticamente, pero lo más cerca que he estado de conseguirlo es mediante bucles “while” y condicionales.