

Hito individual programación

Fase 1:

En esta fase se explica el qué vas a realizar.

Siguiendo la presentación de la UD1. Algoritmia.

Se nos ha pedido que realicemos una aplicación para el diseño y desarrollo de una aplicación de gestión de pedidos.

Las principales operaciones que necesitaremos gestionar son :

1. Registro de cliente. Se pedirán sus datos personales y de facturación.
2. Selección de productos. Se permitirá añadir productos a la lista de deseos.
3. Compra
4. Pago de los productos solicitados.
5. Se enviará al cliente la factura en PDF a su correo electrónico.
6. Seguimiento. Se enviará por SMS al teléfono móvil y al correo del cliente el código de seguimiento de su pedido.

Para resolver los problemas que se nos presentan a la hora de realizar la aplicación vamos a emplear algoritmos. Un algoritmo es una secuencia precisa de operaciones que resuelven un problema en un tiempo finito.

a)Registro de clientes

Para resolver el problema de registro de clientes vamos a emplear un algoritmo,el algoritmo para el registro de clientes tendrá como datos de entrada el nombre del cliente , la ciudad donde reside , su email, su nacionalidad (iva), tipo de cliente , el procedimiento será un formulario de registro de cliente pidiendo estos campos y los datos de salida serán un mensaje indicando que el cliente se ha registrado correctamente y se redirigirá a una ventana de inicio de sesión.En pseudocódigo quedaria asi :

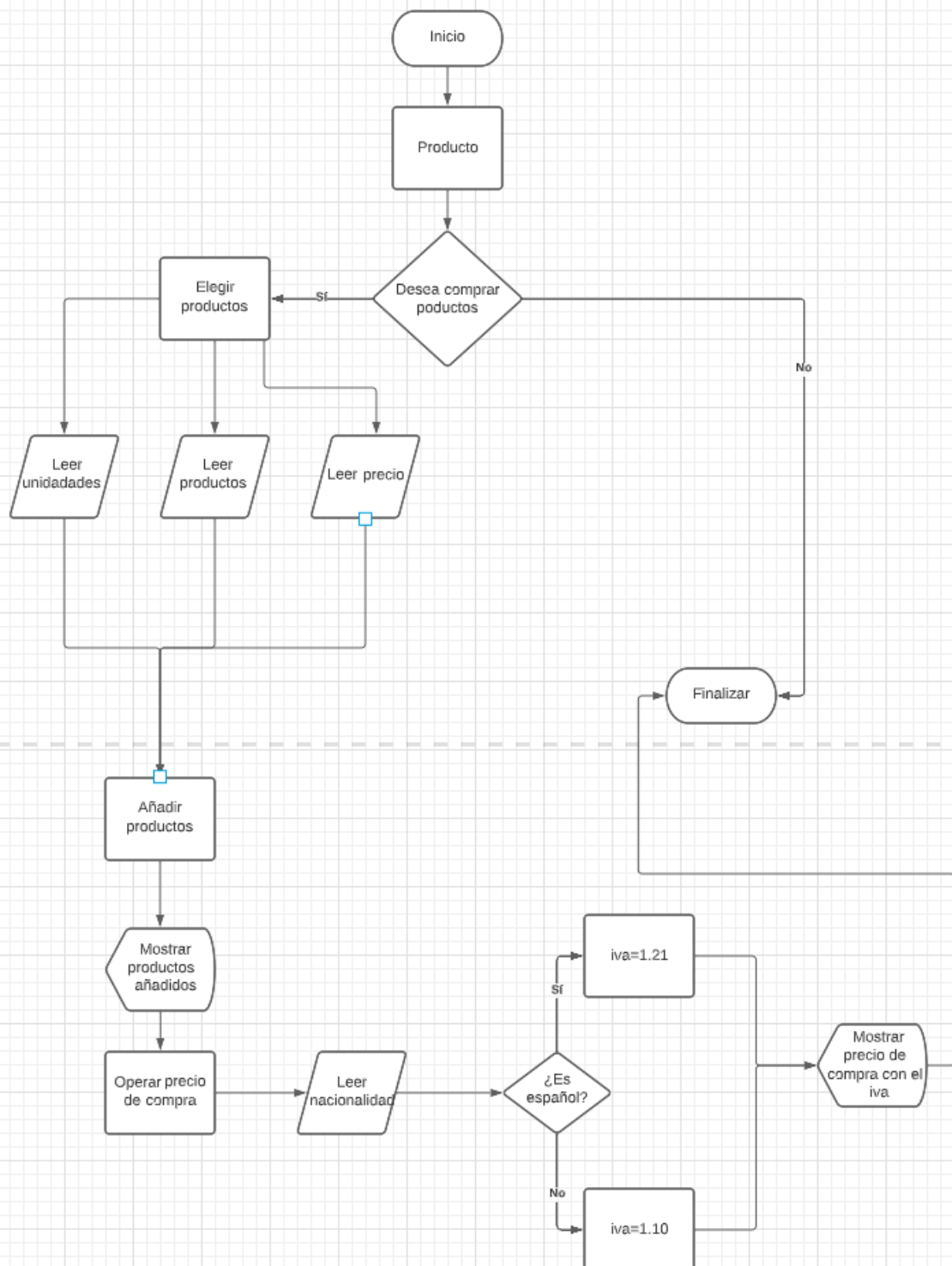
```
1  Algoritmo Registro_De_Clientes
2      // declarar variables
3      // declarar datos de entrada
4      Definir nombre,ciudad,email,nacionalidad,tipoCliente Como Caracter
5      // procedimiento (formulario de registro de cliente pidiendo los datos de entrada)
6      Escribir "Por favor ingrese su nombre:"
7      Leer nombre
8      Escribir "Por favor ingrese la ciudad donde reside:"
9      Leer ciudad
10     Escribir "Por favor ingrese su dirección de correo electrónico:"
11     Leer email
12     Escribir "Por favor ingrese su nacionalidad:"
13     Leer nacionalidad
14     Escribir "Por favor ingrese su tipo de cliente (particular o empresa):"
15     Leer tipoCliente
16     //datos de salida
17     Escribir "Cliente registrado correctamente. Por favor inicie sesión para continuar."
18
19 FinAlgoritmo
```

b)Selección de productos y compra.

Para resolver el problema de compra de un producto vamos a utilizar un diagrama de flujo , que es una representación de la esquematización gráfica de un algoritmo, el cual muestra gráficamente los pasos o procesos a seguir para alcanzar la solución de un problema.

En el diagrama de flujo iniciamos un inicio para la fase de comprar productos, pregunto si quiere realizar una compra o no , si no quiere realizarla se finaliza la fase, en caso de comprar productos se eligen los productos y se leen los datos (como unidades ,productos y precio),posteriormente se añaden los productos a la cesta y te muestra los productos que has añadido.Se calcula el precio y para ello se acceden a los datos puestos en el registro , para ver si su nacionalidad es española o no es española, si es española se aplica un iva de 1.21 y

si no es española se aplica un iva de 1.10, posteriormente se muestra el precio de la compra y se finaliza la fase.

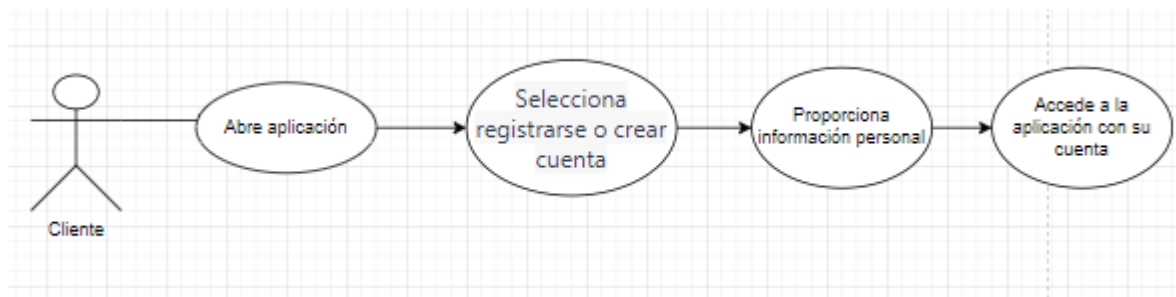


c) Caso de uso

Un caso de uso es una herramienta de análisis y diseño utilizada en el desarrollo de software. Se utiliza para describir los diferentes modos en que un sistema puede ser utilizado por un usuario o sistema externo para alcanzar un objetivo específico.

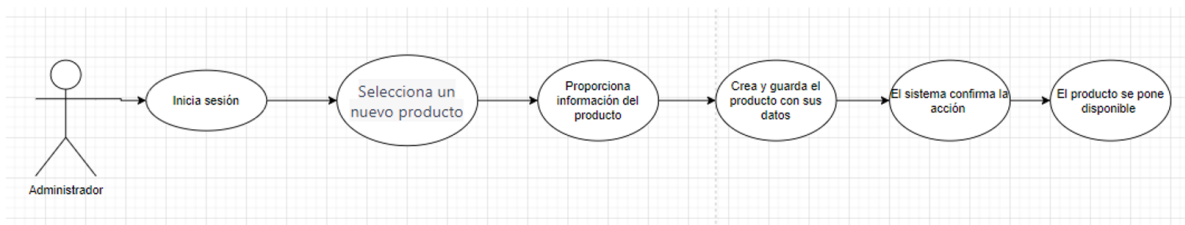
Un caso de uso para un cliente que se da de alta un productos sigue lo siguientes pasos:

- 1.El cliente abre la aplicación.
- 2.El cliente selecciona “Registrarse” o “Crear una cuenta”.
- 3.El cliente rellena el formulario con información personal.
- 4.El sistema permite al cliente acceder a su cuenta.



Un caso de uso para un administrador de nuestra empresa que da de alta un productos sigue lo siguientes pasos:

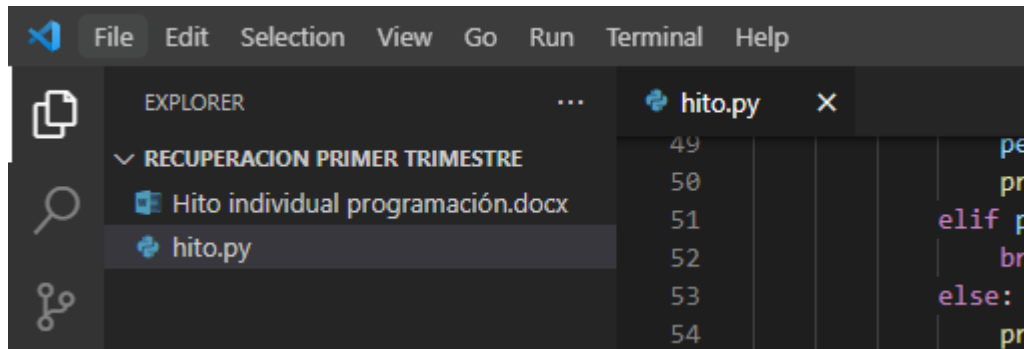
- 1.El administrador inicia sesión.
- 2.El administrador selecciona un nuevo producto.
- 3.El administrador proporciona información del producto y selecciona imágenes o videos.
- 4.El administrador crea y guarda el producto con sus datos.
- 5.El sistema muestra un mensaje de confirmación.
- 6.El producto está disponible para los clientes.



Fase 2:

Esta es la fase de implementación.

Para la fase de implementación voy a utilizar el lenguaje de Python, ya que es el lenguaje que más domino, para el desarrollo del diseño de las clases, atributos, métodos y constructores necesarios para la aplicación. Voy a utilizar programación imperativa y concretamente secuencial y POO. Y para el código lo voy a realizar en Visual Studio ya que es el editor de código que más he usado.



En el código he utilizado varias clases, la clase Cliente, la clase Producto y la clase Pedido.

-La clase Cliente tiene un método constructor (`__init__`) que se llama cuando se crea un nuevo objeto de esta clase. Este método toma varios argumentos (`id`, `id_pedido`, `nombre`, `email`, `país`, `ciudad`, `tipo_de_cliente`) y los asigna a los atributos del objeto. También tiene un método llamado `info_Cliente` que imprime los valores de los atributos.

```
class Cliente:
    def __init__(self, id, id_pedido, nombre, email, pais, ciudad, tipo_de_cliente):
        self.id=id
        self.id_pedido=id_pedido
        self.nombre=nombre
        self.email=email
        self.pais=pais
        self.ciudad=ciudad
        self.tipo=tipo_de_cliente

    def info_Cliente(self):
        print(f'El cliente se llama {self.nombre} y su id es {self.id}')
        print(f'Su email es {self.email}')
        print(f'Es de {self.pais}')
        print(f'Vive en {self.ciudad}')
        print(f'El cliente es {self.tipo}')
        print("El cliente se ha registrado correctamente ")
```

-La clase Producto es una subclase de la clase Cliente y tiene su propio método constructor que llama al constructor de la clase padre usando el

super().__init__(id,id_pedido, ciudad, nombre, email, país, tipo_de_cliente)método. También tiene el método comprar Productos que implementa un bucle para que el cliente elija productos e imprima la factura final, y lee la nacionalidad del Cliente y le aplica el iva correspondiente .

```
class Producto(Cliente):
    def __init__(self, id,id_pedido, ciudad, nombre, email, país, tipo_de_cliente):
        super().__init__(id,id_pedido, ciudad, nombre, email, país, tipo_de_cliente)
    def comprar_Productos(self):
        print('')
        print('Pasamos a comprar productos:')
        print('')
        print('Tenemos 3 productos a la venta' )
        print('producto1=20€(sin contar el iva)')
        print('producto2=25€(sin contar el iva)')
        print('producto3=30€(sin contar el iva)')
        print('')
        print('Para comprar elija el producto que quiera entre producto1, producto2 y producto3')
        print('Cuando ya quiera terminar la compra ponga terminar ')
        pedido=0
        facturación=0
        while True:
            pedidos=input('Diga los productos que quiere añadir a su lista de deseos:')
            if pedidos=='producto1':
                facturación=facturación+20
                pedido=pedido+1
                print('ha añadido el producto 1')
            elif pedidos=='producto2':
                facturación=facturación+25
                pedido=pedido+1
                print('ha añadido el producto 2')
            elif pedidos=='producto3':
                facturación=facturación+30
                pedido=pedido+1
                print('ha añadido el producto 3')
            elif pedidos=='terminar':
                break
            else:
                print('No ha pedido ningún producto')
        print(f'Ha añadido un total de {pedido} productos a su lista de deseos')
        if self.país == 'España':
            factura=facturación*1.21
            print(f'Su factura total es de {factura} $')
        else:
            factura=facturación*1.10
            print(f'Su factura total es de {factura} $')
```

-La clase Pedido también es una subclase de la clase Cliente y tiene su propio constructor que también llama al constructor de la clase principal. Tiene cuatro métodos info_Pedido, finalizarCompra, facturaPDF y Seguimiento, el primero imprime detalles del pedido, el segundo imprime que la compra ha finalizado, el tercero envía al email la factura y el último envía un sms al cliente para que pueda seguir el pedido.

```
class Pedido(Cliente):
    def __init__(self, id, id_pedido, ciudad, nombre, email, pais, tipo_de_cliente):
        super().__init__(id, id_pedido, ciudad, nombre, email, pais, tipo_de_cliente)
    def info_Pedido(self):
        print('')
        print(f'Detalles de pedido: id_pedido - {self.id_pedido}, fecha de compra - {datetime.now()} , nombre del cliente - {self.nombre}')
    def finalizarCompra(self):
        print('Compra finalizada')
    def facturaPDF(self):
        print(f'Su factura sera enviada en pdf a {self.email}')
    def seguimiento(self):
        print(f'Se le ha enviado un sms {self.nombre} para que pueda seguir su pedido.')
```

Y para comprobar que el código funciona correctamente he creado 3 funciones llamadas Cliente1, Cliente2 y Prueba_Validación, que crean instancias de las 3 clases. La primera función tiene un cliente que es español por lo que el iva que se le aplica es de 1.21 y la segunda función tiene a un cliente que no es español por lo que se le aplica un iva de 1.10, cada cliente tiene sus respectivos datos, y la función llamada Prueba_Validación es una simple prueba para ver qué pasaría si el cliente no introduce los datos correctos.

```
78
79 def Cliente1():
80     cliente1=Cliente(1212,12,'Adrian','adrian@gmail.com','España','Madrid','Particular')
81     cliente1.info_Cliente()
82     producto1=Producto(1212,12,'Adrian','adrian@gmail.com','España','Madrid','Particular')
83     producto1.comprar_Productos()
84     pedido1=Pedido(1212,12,'Adrian','adrian@gmail.com','España','Madrid','Particular')
85     pedido1.info_Pedido()
86     pedido1.finalizarCompra()
87     pedido1.facturaPDF()
88     pedido1.seguimiento()
89     Cliente1()
90
91
92 def Cliente2():
93     cliente2=Cliente(1213,21,'Jesús','jesus@gmail.com','Francia','Madrid','Empresa')
94     cliente2.info_Cliente()
95     producto2=Producto(1213,21,'Jesús','jesus@gmail.com','Francia','Madrid','Empresa')
96     producto2.comprar_Productos()
97     pedido2=Pedido(1213,21,'Jesús','jesus@gmail.com','Francia','Madrid','Empresa')
98     pedido2.info_Pedido()
99     pedido2.finalizarCompra()
100    pedido2.facturaPDF()
101    pedido2.seguimiento()
102
103    Cliente2()
104
105
106 def Prueba_Validación():
107     prueba=Cliente(13,21,'Pablo',11,'Portugal','Lisboa','Empresa')
108     prueba.info_Cliente()
109     prueba1=Producto(13,21,'Pablo',11,'Portugal','Lisboa','Empresa')
110     prueba1.comprar_Productos()
111     prueba2=Pedido(13,21,'Pablo',11,'Portugal','Lisboa','Empresa')
112     prueba2.info_Pedido()
113     prueba2.finalizarCompra()
114     prueba2.facturaPDF()
115     prueba2.seguimiento()
116
117
118 #Prueba_Validación()
119
```

Y este sería el resultado del código.

```
El cliente se llama Adrian y su id es 1212
Su email es  adrian@gmail.com
Es de  España
Vive en Madrid
El cliente es Particular
El cliente se ha registrado correctamente

Pasamos a comprar productos:

Tenemos 3 productos a la venta
producto1=20€(sin contar el iva)
producto2=25€(sin contar el iva)
producto3=30€(sin contar el iva)

Para comprar eliga el producto que quiera entre producto1, producto2 y producto3
Cuando ya quiera terminar la compra ponga terminar
Diga los productos que quiere añadir a su lista de deseos:producto1
ha añadido el producto 1
Diga los productos que quiere añadir a su lista de deseos:terminar
Ha añadido un total de 1 productos a su lista de deseos
Su factura total es de 24.2 $

Detalles de pedido: id_pedido - 12, fecha de compra - 2023-01-11 19:10:08.682365 , nombre del cliente - Adrian
Compra finalizada
Su factura sera enviada en pdf a adrian@gmail.com
Se le ha enviado un sms Adrian para que pueda seguir su pedido.
```

```
El cliente se llama Jesús y su id es 1213
Su email es  jesus@gmail.com
Es de  Francia
Vive en Madrid
El cliente es Empresa
El cliente se ha registrado correctamente

Pasamos a comprar productos:

Tenemos 3 productos a la venta
producto1=20€(sin contar el iva)
producto2=25€(sin contar el iva)
producto3=30€(sin contar el iva)

Para comprar eliga el producto que quiera entre producto1, producto2 y producto3
Cuando ya quiera terminar la compra ponga terminar
Diga los productos que quiere añadir a su lista de deseos:producto1
ha añadido el producto 1
Diga los productos que quiere añadir a su lista de deseos:terminar
Ha añadido un total de 1 productos a su lista de deseos
Su factura total es de 22.0 $

Detalles de pedido: id_pedido - 21, fecha de compra - 2023-01-11 19:11:22.479516 , nombre del cliente - Jesús
Compra finalizada
Su factura sera enviada en pdf a jesus@gmail.com
Se le ha enviado un sms Jesús para que pueda seguir su pedido.
```


Fase 3:

En esta fase se define el por qué

He usado el paradigma de programación orientada a objetos (POO) debido a que considero que este paradigma es el que mejor manejo y se ajusta muy bien para realizar la aplicación, ya que ofrece varias ventajas como poder usar encapsulamiento, reutilización, polimorfismo y herencia. POO es un paradigma de programación que se basa en el concepto de "objetos", que pueden tener propiedades y métodos, y pueden interactuar entre sí. Por ello en el código he introducido, las clases Cliente, Producto y Pedidos que se utilizan para definir objetos que representan clientes, productos y pedidos, respectivamente. Cada clase tiene sus propios atributos y métodos, y las clases están relacionadas entre sí a través de la herencia, donde una clase puede heredar las propiedades y métodos de otra clase.

Aunque no he utilizado ninguna validación en este código. Por lo que si el usuario no introduce los datos correctos, el programa no podrá procesarlos y puede provocar que el código se bloquee.