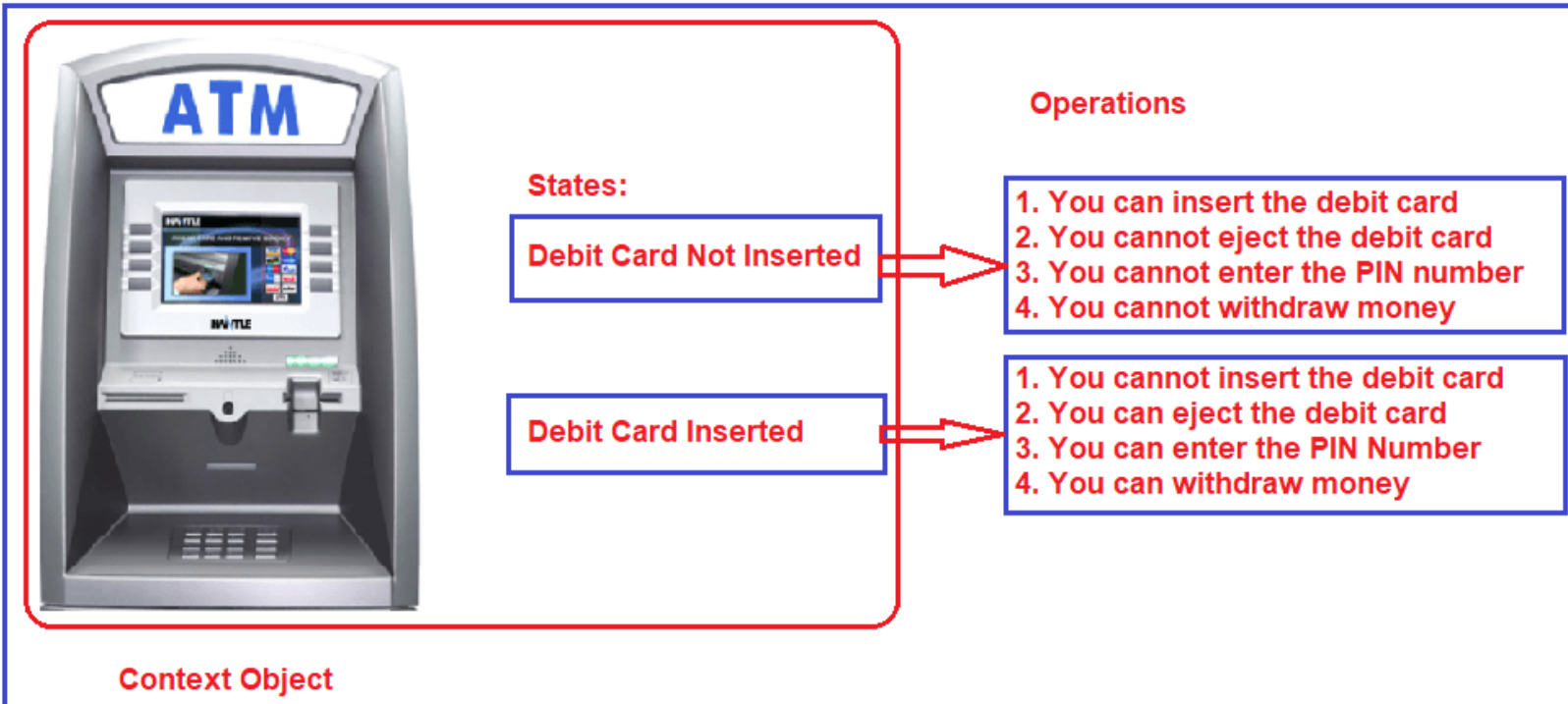


PATRÓN DE DISEÑO STATE

¿EN QUÉ CONSISTE?



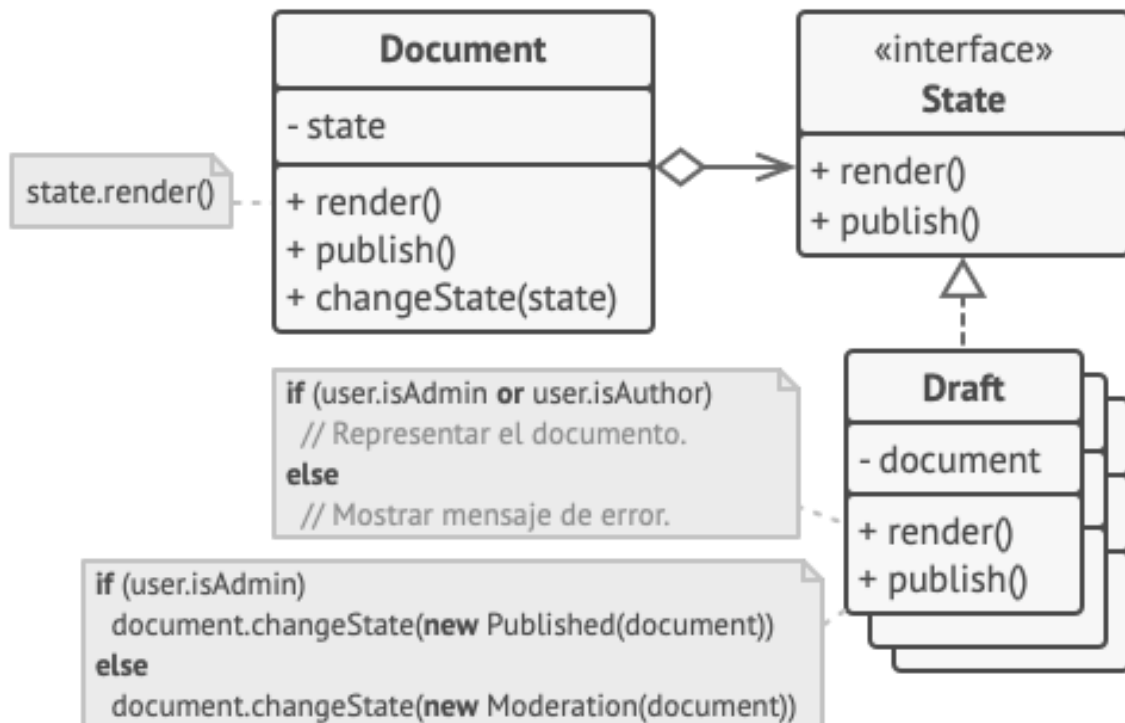
CONTEXTO

ESTADO

ESTADOS CONCRETOS

¿POR QUÉ USARLO?

```
class Document is
  field state: string
  // ...
  method publish() is
    switch (state)
      "draft":
        state = "moderation"
        break
      "moderation":
        if (currentUser.role == "admin")
          state = "published"
        break
      "published":
        // No hacer nada.
        break
  // ...
```



IMPLEMENTACION

```
class Banco:
    def __init__(self):
        #Estado inicial del programa
        self.controlador=ControladorDatos()
        self.estado = EstadoEsperar(self,self.controlador)

    ...

    #Interpretar y procesar los eventos que ocurren en la GUI
    def peticionImprimir(self):
        self.cambiarEstado(EstadoImprimir(self,self.controlador))
        self.estado.realizarAccion()

    def peticionEliminar(self):
        self.cambiarEstado(EstadoEliminar(self,self.controlador))
        self.estado.realizarAccion()

    def peticionBorrarCuenta(self,Cuenta):
        self.estado.realizarAccion(Cuenta)

    def peticionVerCliente(self,Cliente):
        self.cambiarEstado(EstadoVerCliente(self,self.controlador))
        self.estado.realizarAccion(Cliente)

    def peticionInsertar(self,nuevoRegistro):
        self.cambiarEstado(EstadoInsertar(self,self.controlador))
        self.estado.realizarAccion(nuevoRegistro)
    def peticion(self):
        self.estado.realizarAccion()

    ...
```

```

class Estado:
    def __init__(self, contexto, controlDatos):
        self.controlDatos = controlDatos
        self.contexto = contexto

    def realizarAccion(self, textoAIngresar="", ClienteElegido= None, Cuenta=None):
        pass

# Estado esperar
class EstadoEsperar(Estado):
    def realizarAccion(self):
        print("Espera")

# Estado insertar clientes
class EstadoInsertar(Estado):
    def realizarAccion(self, textoAIngresar):
        self.controlDatos.leerRegistro()
        self.controlDatos.agregarRegistro(textoAIngresar)
        self.contexto.cambiarEstado(EstadoEsperar(self.contexto, self.controlDatos))

# Estado imprimir clientes
class EstadoImprimir(Estado):
    def realizarAccion(self):
        self.controlDatos.leerRegistro()
        self.texto = self.controlDatos.getRegistro()
        self.contexto.setInfoTexto(self.texto)
        self.contexto.cambiarEstado(EstadoEsperar(self.contexto, self.controlDatos))

# Estado ver clientes
class EstadoVerCliente(Estado):
    def realizarAccion(self, ClienteElegido):
        self.texto = self.controlDatos.getDatosCliente(ClienteElegido)
        self.contexto.setInfoTexto(self.texto)
        self.contexto.cambiarEstado(EstadoEsperar(self.contexto, self.controlDatos))

...
class EstadoEliminar(Estado):...
class EstadoBorrarCuenta(Estado):...
class EstadoModificarCuenta(Estado):...

...
# Crear una instancia del banco y la interfaz GUI
banco = Banco()
gui = GUI(banco)

```

```
class GUI:
>     def __init__(self, contexto): ...
        #ventanas y funciones del programa
>     def esperar(self): ...
>
>     def imprimirClientes(self): ...
>
>     def insertarCliente(self): ...
>
>     def botonAccion(self): ...
>
>     def eliminarCliente(self): ...
>
>     def elegirCliente(self, obj): ...
>
>     def volverEsperar(self): ...
```

Programa de Banco



Lista de clientes y cuentas

1, Juan Pérez, 1587569425361548, 459.0	3, Juan Gomez, 1256485666423564, 582.0	3, Juan Gomez, 1256485798423564, 582.0
2, Rodrigo Mendoza, 1598264853214574, 90.0	4, Juan Benitez, 1234567891127756, 3525.0	5, Alfredo Irrutia, 1234567899188456, 560.0
5, Alfredo Irrutia, 1234567899188156, 560.0	5, Alfredo Irrutia, 1598764853214574, 0.0	6, Luis Mena, 1598764853214574, 890.0
15, Javie Milei, 1548765487546987, 582.0	15, Javie Milei, 1548865487548287, 582.0	123, John Doe, 1234567890123456, 500.0
12, Cristina Fernandez, 8888865427548287, 582.0		
VOLVER		

Eliminar Cliente

Lista de clientes y cuentas

1, Juan Pérez	3, Juan Gomez	2, Rodrigo Mendoza
4, Juan Benitez	5, Alfredo Irrutia	6, Luis Mena
15, Javie Milei	123, John Doe	12, Cristina Fernandez

VOLVER

Insertar Clientes

Nueva entrada

00

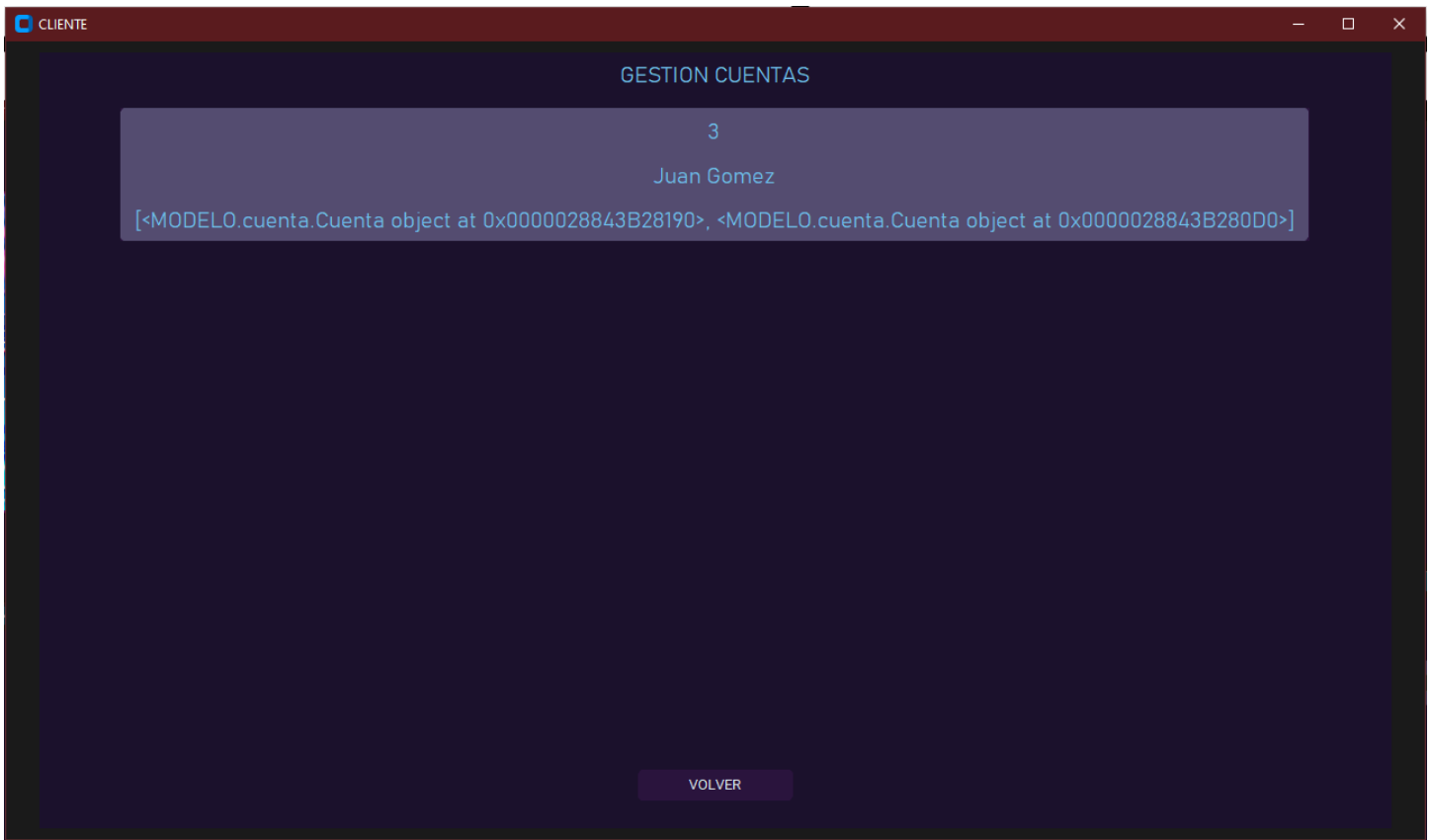
Pedro Espinacas

0011223344556677

100

Ingresar

VOLVER



DESVENTAJAS

COMPLEJIDAD ADICIONAL

USO INAPROPIADO

COORDINACIÓN ENTRE ESTADOS