

Cuenta

+emitirTarjetaCredito(nif:String,crédito:double):TarjetaCredito

PARÁMETROS DE ENTRADA	CLASES DE EQUIVALENCIA	CONJUNTO DE VALORES DE PRUEBA
<code>optCliente.isPresent()</code>	True,False	True,False
<code>titular.getNif().equals(cliente.nif)</code>	True, False	True,False
<code>encontrado</code>	True, False	True, False

CASOS DE PRUEBA			
<code>optCliente.isPresent()</code>	<code>titular.getNif().equals(cliente.nif)</code>	<code>encontrado</code>	Salida
False	X	X	ClienteNoEncontradoException()
True	True	X	encontrado=true
True	False	False	ClienteNoAutorizadoException()
True	False	True	return tarjeta

+emitirTarjetaDebito(nif:String):TarjetaDebito

PARÁMETROS DE ENTRADA	CLASES DE EQUIVALENCIA	CONJUNTO DE VALORES DE PRUEBA
<code>optCliente.isPresent()</code>	True,False	True,False
<code>titular.getNif().equals(cliente.nif)</code>	True, False	True,False
<code>encontrado</code>	True, False	True, False

CASOS DE PRUEBA			
<code>optCliente.isPresent()</code>	<code>titular.getNif().equals(cliente.nif)</code>	<code>encontrado</code>	Salida
False	X	X	ClienteNoEncontradoException()
True	True	X	encontrado=true
True	False	False	ClienteNoAutorizadoException()
True	False	True	return tarjeta

+transferir(numeroCuentaDestino:Long,importe:double,concepto:String):void

PARÁMETROS DE ENTRADA	CLASES DE EQUIVALENCIA	CONJUNTO DE VALORES DE PRUEBA
numeroCuentaDestino	==this.getId(), !=this.getId()	==this.getId(), !=this.getId()
Importe		
concepto		

CASOS DE PRUEBA	
numeroCuentaDestino	Salida
==this.getId()	CuentaInvalidaException()
!=this.getId()	Transferencia exitosa

-retirar(importe:double,concepto:String):void

PARÁMETROS DE ENTRADA	CLASES DE EQUIVALENCIA	CONJUNTO DE VALORES DE PRUEBA
importe	(-inf,0.00],[0.00,getSaldo], (getSaldo, +inf)	-5.00, 0.00, 0.00<importe<getSaldo, getSaldo<importe<+inf,-0.01, 0.01

CASOS DE PRUEBA	
importe	Salida
importe<=0	ImporteInvalidoException()
getSaldo<importe	SaldoInsuficienteException()
0<importe<=getSaldo	Retirada exitosa

-ingresar(importe:double,concepto:String):void

PARÁMETROS DE ENTRADA	CLASES DE EQUIVALENCIA	CONJUNTO DE VALORES DE PRUEBA
importe	(-inf,0.00],[0.00,+inf)	-5.00, 0.00, 0.00<importe<getSaldo, getSaldo<importe<+inf,-0.01, 0.01

CASOS DE PRUEBA	
importe	Salida
importe<=0	ImporteInvalidoException()
importe>0	Ingreso exitoso

+addTitular(cliente:Cliente):void

PARÁMETROS DE ENTRADA	CLASES DE EQUIVALENCIA	CONJUNTO DE VALORES DE PRUEBA
creada	True,False	True,False

CASOS DE PRUEBA	
creada	Salida
True	CuentaYaCreadaException()
False	this.titulares.add(cliente)

+retiroForzoso(importe:double,concepto:String):void

PARÁMETROS DE ENTRADA	CLASES DE EQUIVALENCIA	CONJUNTO DE VALORES DE PRUEBA
importe	(-inf,+inf)	0.00

CASOS DE PRUEBA	
importe	Salida
(-inf,+inf)	Retiro forzoso realizado

+ingresar(importe:double):void

PARÁMETROS DE ENTRADA	CLASES DE EQUIVALENCIA	CONJUNTO DE VALORES DE PRUEBA
importe	(-inf,+inf)	0.00

CASOS DE PRUEBA	
importe	Salida
(-inf,+inf)	Ingreso realizado

+retirar (importe:double):void

PARÁMETROS DE ENTRADA	CLASES DE EQUIVALENCIA	CONJUNTO DE VALORES DE PRUEBA
importe	(-inf,+inf)	0.00

CASOS DE PRUEBA	
importe	Salida
(-inf,+inf)	Retiro realizado

TarjetaCredito

+comprarPorInternet(pin:int,importe:double):Integer

PARÁMETROS DE ENTRADA	CLASES DE EQUIVALENCIA	CONJUNTO DE VALORES DE PRUEBA
pin	Correcto,Incorrecto	Correcto,Incorrecto
this.isActiva()	True, False	True,False
intentos	(-inf, 0],[0,3],[3,+inf)	-5,2,5,0,3,-1,1,2,4
importe	(-inf,0.00], (0.00,getCreditoDisponible), [getCreditoDisponible,+inf)	0.00, -10.00, 0.00<x<getCreditoDisponible, getCreditoDisponible<x<+inf

CASOS DE PRUEBA				
this.isActiva()	pin	intentos	importe	Salida
False	X	X	X	TarjetaBloqueadaException()
True	Incorrecto	0,1,	X	PinInvalidoException()
True	Incorrecto	2	X	this.activa=false
True	Correcto	X	importe>getCDisp	SaldoInsuficienteException()
True	Correcto	X	importe<=0	ImporteInvalidoException()
True	Correcto	X	0<importe<getCDisp	return token

+sacarDinero(pin:int,importe:double):void

PARÁMETROS DE ENTRADA	CLASES DE EQUIVALENCIA	CONJUNTO DE VALORES DE PRUEBA
pin	Correcto, Incorrecto	Correcto, Incorrecto
this.isActiva()	True,False	True,False
intentos	(-inf, 0],[0,3],[3,+inf)	-5,2,5,0,3,-1,1,2,4
importe	(-inf,0.00], (0.00,getCreditoDisponible), [getCreditoDisponible,+inf)	0.00, -10.00, 0.00<x<getCreditoDisponible, getCreditoDisponible<x<+inf
comision	(-inf,0.00],[0.00,3.00],[3.00,inf)	-2.00,1.50,4.50, 0.00, 3.00, -0.01, 0.01, 2.99,3.01

CASOS DE PRUEBA					
this.isActiva()	pin	intentos	importe	comision	Salida
False	X	X	X	X	TarjetaBloqueadaException()
True	Incorrecto	0,1	X	X	PinInvalidoException()
True	Incorrecto	2	X	X	this.activa=false
True	Correcto	X	importe>getCDisp	X	SaldoInsuficienteException()
True	Correcto	X	importe<=0	X	ImporteInvalidoException()
True	Correcto	X	0<importe<getCDisp	X	Operación exitosa

+liquidar():void

PARÁMETROS DE ENTRADA	CLASES DE EQUIVALENCIA	CONJUNTO DE VALORES DE PRUEBA
gastos	(-inf,0.00],[0.00,+inf)	-2.00,2.00,0.00,-0.01,+0.01
m.isLiquidado()	True, False	True,False
m.getImporte()	(-inf,0.00],[0.00,+inf)	-3.00,3.00,0.00,-0.01,0.01

CASOS DE PRUEBA			
m.isLiquidado()	m.getImporte()	gastos	Salida
False	X	X	this.liquidado=true
True	X	X	No hay que liquidar

+comprar(pin:int,importe:double):void

PARÁMETROS DE ENTRADA	CLASES DE EQUIVALENCIA	CONJUNTO DE VALORES DE PRUEBA
pin	Correcto,Incorrecto	Correcto,Incorrecto
this.isActiva()	True,False	True,False
intentos	(-inf, 0],[0,3],[3,+inf)	-5,2,5,0,3,-1,1,2,4
importe	(-inf,0.00], (0.00,getCreditoDisponible), [getCreditoDisponible,+inf)	0.00, -10.00, 0.00<x<getCreditoDisponible, getCreditoDisponible<x<+inf

CASOS DE PRUEBA				
this.isActiva()	pin	intentos	importe	Salida
False	X	X	X	TarjetaBloqueadaException()
True	Incorrecto	0,1,	X	PinInvalidoException()
True	Incorrecto	2	X	this.activa=false
True	Correcto	X	importe>getCDisp	SaldoInsuficienteException()
True	Correcto	X	importe<=0	ImporteInvalidoException()
True	Correcto	X	0<importe<getCDisp	Compra realizada

+getCreditoDisponible():Double

PARÁMETROS DE ENTRADA	CLASES DE EQUIVALENCIA	CONJUNTO DE VALORES DE PRUEBA
gastos	(-inf,0.00],[0.00,+inf)	-2.00,2.00,0.00,-0.01,+0.01
m.isLiquidado()	True, False	True,False
m.getImporte()	(-inf,0.00],[0.00,+inf)	-3.00,3.00,0.00,-0.01,0.01

CASOS DE PRUEBA			
m.isLiquidado()	m.getImporte()	gastos	Salida
False	X	X	return credito-gastos
True	X	X	return credito-gastos

+cambiarPin(pinViejo:int,pinNuevo:int):void

PARÁMETROS DE ENTRADA	CLASES DE EQUIVALENCIA	CONJUNTO DE VALORES DE PRUEBA
pinViejo	==this.pin, !=this.pin	==this.pin, !=this.pin
pinNuevo	(-inf,0],[0,+inf)	-1000,1000,0,-1,1

CASOS DE PRUEBA		
pinViejo	pinNuevo	Salida
==this.pin	X	this.pin=pinNuevo
!=this.pin	X	PinInvalidoException()

#bloquear():void

PARÁMETROS DE ENTRADA	CLASES DE EQUIVALENCIA	CONJUNTO DE VALORES DE PRUEBA
this.activa	True,False	True,False

CASOS DE PRUEBA	
this.activa	Salida
True	this.activa=false
False	this.activa=false

TarjetaDebito

+comprarPorInternet(pin:int,importe:double):Integer

PARÁMETROS DE ENTRADA	CLASES DE EQUIVALENCIA	CONJUNTO DE VALORES DE PRUEBA
pin	Correcto,Incorrecto	Correcto,Incorrecto
this.isActiva()	True, False	True,False
intentos	(-inf, 0],[0,3],[3,+inf)	-5,2,5,0,3,-1,1,2,4

CASOS DE PRUEBA			
this.isActiva()	pin	intentos	Salida
False	X	X	TarjetaBloqueadaException()
True	Incorrecto	0,1,	PinInvalidoException()
True	Incorrecto	2	this.activa=false
True	Correcto	X	return token

+cambiarPin(pinViejo:int,pinNuevo:int):void

PARÁMETROS DE ENTRADA	CLASES DE EQUIVALENCIA	CONJUNTO DE VALORES DE PRUEBA
pinViejo	==this.pin, j=this.pin	==this.pin, j=this.pin
pinNuevo	(-inf,0],[0,+inf)	-1000,1000,0,-1,1

CASOS DE PRUEBA		
pinViejo	pinNuevo	Salida
==this.pin	X	this.pin=pinNuevo
j=this.pin	X	PinInvalidoException()

+sacarDinero(pin:int,importe:double):void

PARÁMETROS DE ENTRADA	CLASES DE EQUIVALENCIA	CONJUNTO DE VALORES DE PRUEBA
pin	Correcto,Incorrecto	Correcto,Incorrecto
this.isActiva()	True, False	True,False
intentos	(-inf, 0],[0,3],[3,+inf)	-5,2,5,0,3,-1,1,2,4

CASOS DE PRUEBA			
this.isActiva()	pin	intentos	Salida
False	X	X	TarjetaBloqueadaException()
True	Incorrecto	0,1,	PinInvalidoException()
True	Incorrecto	2	this.activa=false
True	Correcto	X	Operación exitosa

+comprar(pin:int,importe:double):void

PARÁMETROS DE ENTRADA	CLASES DE EQUIVALENCIA	CONJUNTO DE VALORES DE PRUEBA
pin	Correcto,Incorrecto	Correcto,Incorrecto
this.isActiva()	True, False	True,False
intentos	(-inf, 0],[0,3],[3,+inf)	-5,2,5,0,3,-1,1,2,4

CASOS DE PRUEBA			
this.isActiva()	pin	intentos	Salida
False	X	X	TarjetaBloqueadaException()
True	Incorrecto	0,1,	PinInvalidoException()
True	Incorrecto	2	this.activa=false
True	Correcto	X	Operación exitosa

#bloquear():void

PARÁMETROS DE ENTRADA	CLASES DE EQUIVALENCIA	CONJUNTO DE VALORES DE PRUEBA
this.activa	True,False	True,False

CASOS DE PRUEBA	
this.activa	Salida
True	this.activa=false
False	this.activa=false

Tarjeta

+comprobar(pin:int):void

PARÁMETROS DE ENTRADA	CLASES DE EQUIVALENCIA	CONJUNTO DE VALORES DE PRUEBA
pin	Correcto,Incorrecto	Correcto,Incorrecto
this.isActiva()	True, False	True,False
intentos	(-inf, 0],[0,3],[3,+inf)	-5,2,5,0,3,-1,1,2,4

CASOS DE PRUEBA			
this.isActiva()	pin	intentos	Salida
False	X	X	TarjetaBloqueadaException()
True	Incorrecto	0,1,	PinInvalidoException()
True	Incorrecto	2	this.activa=false
True	Correcto	X	Comprobación realizada

+confirmarCompraPorInternet(token:int):void

PARÁMETROS DE ENTRADA	CLASES DE EQUIVALENCIA	CONJUNTO DE VALORES DE PRUEBA
token	==this.compra.getToken(), !=this.compra.getToken()	==this.compra.getToken(), !=this.compra.getToken()

CASOS DE PRUEBA	
this.isActiva()	Salida
==this.compra.getToken()	Confirmación realizada
!=this.compra.getToken()	TokenInvalidoException()