

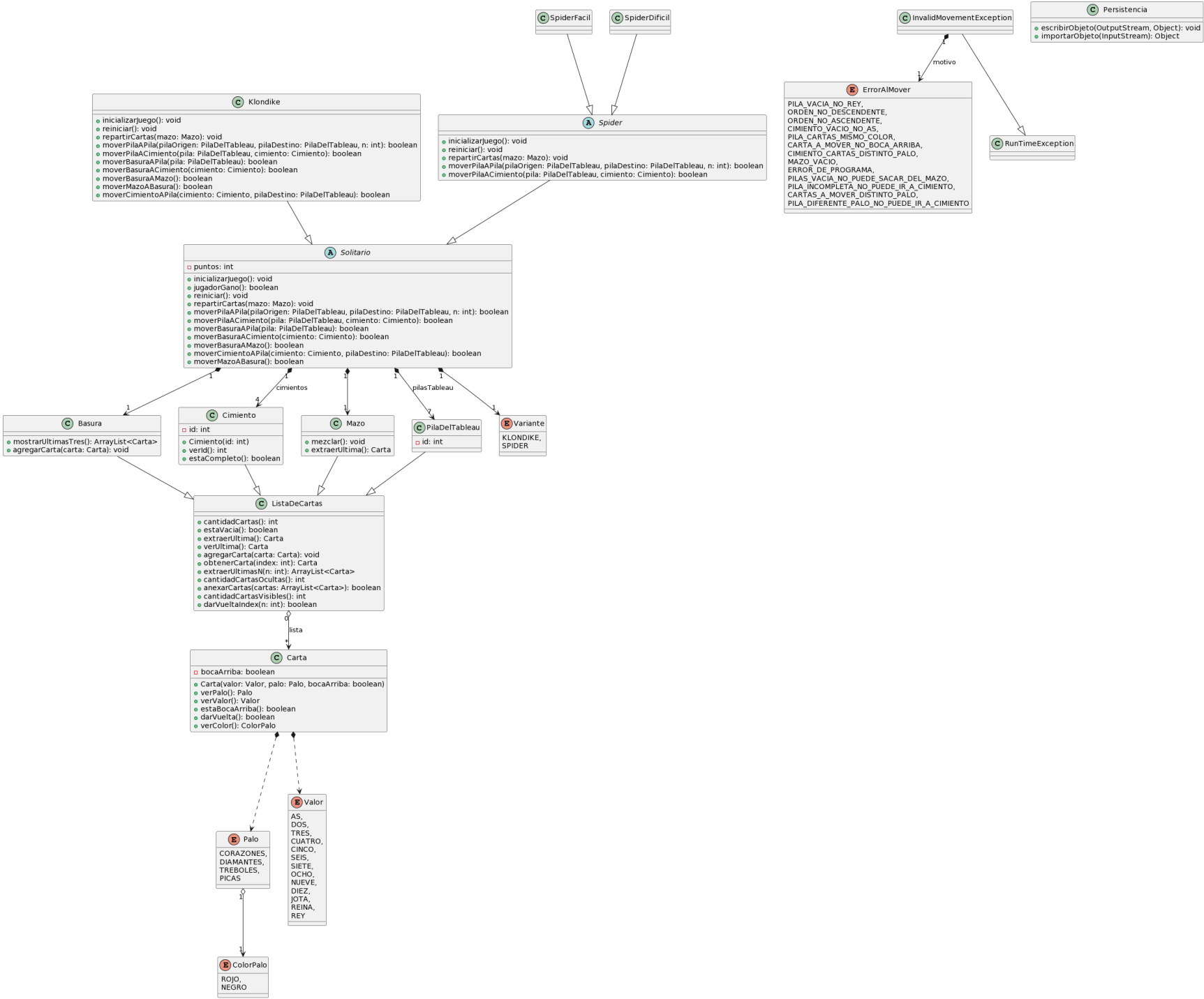
Etapa 2 - Algoritmos y Programación III

Alumnos:

Spaltro, Francisco	102098	fspaltro@fi.uba.ar
Peña, Leandro	102282	lpena@fi.uba.ar

Corrector: Diego Essaya

Diagrama UML



Código UML

```
@startuml

class Basura {
    + mostrarUltimasTres(): ArrayList<Carta>
    + agregarCarta(carta: Carta): void
}

class ListaDeCartas {
    + cantidadCartas(): int
    + estaVacía(): boolean
    + extraerUltima(): Carta
    + verUltima(): Carta
    + agregarCarta(carta: Carta): void
    + obtenerCarta(index: int): Carta
    + extraerUltimasN(n: int): ArrayList<Carta>
    + cantidadCartasOcultas(): int
    + anexarCartas(cartas: ArrayList<Carta>): boolean
    + cantidadCartasVisibles(): int
    + darVueltaIndex(n: int): boolean
}

class Carta {
    - bocaArriba: boolean
    + Carta(valor: Valor, palo: Palo, bocaArriba: boolean)
    + verPalo(): Palo
    + verValor(): Valor
    + estaBocaArriba(): boolean
    + darVuelta(): boolean
    + verColor(): ColorPalo
}

class Cimientos {
    - id: int
    + Cimiento(id: int)
    + verId(): int
    + estaCompleto(): boolean
}

class Valor {
    + AS
    + DOS
    + TRES
    + CUATRO
    + CINCO
    + SEIS
    + SIETE
    + OCHO
    + NUEVE
    + DIEZ
    + JOTA
    + REINA
    + REY
}

class Palo {
    + CORAZONES
    + DIAMANTES
    + TREBOLES
    + PICAS
}

class ColorPalo {
    + ROJO
    + NEGRO
}

class InvalidMovementException {
    + motivo
}

class Persistence {
    + escribirObjeto(OutputStream, Object): void
    + importarObjeto(InputStream): Object
}

class ErrorAlMover {
    + PILA_VACIA_NO_REY
    + ORDEN_NO_ASCENDENTE
    + ORDEN_NO_DESCENDENTE
    + CIMENTO_VACIO_NO_AS
    + PILA_CARTAS_MISMO_COLOR
    + CARTA_A_MOVER_NO_BOCA_ARRIBA
    + CIMIENTO_CARTAS_DISTINTO_PALO
    + MAZO_VACIO
    + ERROR_DE_PROGRAMA
    + PILAS_VACIA_NO_PUEDE_SACAR_DEL_MAZO
    + PILA_INCOMPLETA_NO_PUEDE_IR_A_CIMIENTO
    + CARTAS_A_MOVER_DISTINTO_PALO
    + PILA_DIFERENTE_PALO_NO_PUEDE_IR_A_CIMIENTO
}

class RunTimeException

Klondike --> Spider
SpiderFacil --|> Spider
SpiderDificil --|> Spider
Spider --|> Solitario
Solitario --> Basura : 1
Solitario --> Cimientos : 4
Solitario --> Mazo : 1
Solitario --> PilaDeTableau : 1
Solitario --> Variante : 1
Basura --> ListaDeCartas
Cimientos --> ListaDeCartas
Mazo --> ListaDeCartas
PilaDeTableau --> ListaDeCartas
Variante --> ListaDeCartas
ListaDeCartas --> Carta : lista
Carta --> Valor
Carta --> Palo
Palo --> ColorPalo
InvalidMovementException --> ErrorAlMover : motivo
Persistence --> RunTimeException
```

```

enum ColorPalo {
    ROJO,
    NEGRO
}

class InvalidMovementException{
}

enum ErrorAlMover {
    PILA_VACIA_NO_REY,
    ORDEN_NO_DESCENDENTE,
    ORDEN_NO_ASCENDENTE,
    CIMIENTO_VACIO_NO_AS,
    PILA_CARTAS_MISMO_COLOR,
    CARTA_A_MOVER_NO_BOCA_ARRIBA,
    CIMIENTO_CARTAS_DISTINTO_PALO,
    MAZO_VACIO,
    ERROR_DE_PROGRAMA,
    PILAS_VACIA_NO_PUEDE_SACAR_DEL_MAZO,
    PILA_INCOMPLETA_NO_PUEDE_IR_A_CIMIENTO,
    CARTAS_A_MOVER_DISTINTO_PALO,
    PILA_DIFERENTE_PALO_NO_PUEDE_IR_A_CIMIENTO
}

class Klondike {
    + inicializarJuego(): void
    + reiniciar(): void
    + repartirCartas(mazo: Mazo): void
    + moverPilaAPila(pilaOrigen: PilaDelTableau, pilaDestino: PilaDelTableau, n: int): boolean
    + moverPilaACimiento(pila: PilaDelTableau, cimiento: Cimiento): boolean
    + moverBasuraAPila(pila: PilaDelTableau): boolean
    + moverBasuraACimiento(cimiento: Cimiento): boolean
    + moverBasuraAMazo(): boolean
    + moverMazoABasura(): boolean
    + moverCimientoAPila(cimiento: Cimiento, pilaDestino: PilaDelTableau): boolean
}

abstract class Spider{
    + inicializarJuego(): void
    + reiniciar(): void
    + repartirCartas(mazo: Mazo): void
    + moverPilaAPila(pilaOrigen: PilaDelTableau, pilaDestino: PilaDelTableau, n: int): boolean
    + moverPilaACimiento(pila: PilaDelTableau, cimiento: Cimiento): boolean
}

class SpiderFacil{
}

class SpiderDificil{
}

class Mazo {
    + mezclar(): void
    + extraerUltima(): Carta
}

enum Palo {
    CORAZONES,
    DIAMANTES,
    TREBOLES,
    PICAS
}

class PilaDelTableau {
    - id: int
}

class Persistencia {
    + escribirObjeto(OutputStream, Object): void
    + importarObjeto(InputStream): Object
}

abstract class Solitario {
    - puntos: int
    + inicializarJuego(): void
    + jugadorGano(): boolean
    + reiniciar(): void
    + repartirCartas(mazo: Mazo): void
    + moverPilaAPila(pilaOrigen: PilaDelTableau, pilaDestino: PilaDelTableau, n: int): boolean
    + moverPilaACimiento(pila: PilaDelTableau, cimiento: Cimiento): boolean
    + moverBasuraAPila(pila: PilaDelTableau): boolean
    + moverBasuraACimiento(cimiento: Cimiento): boolean
    + moverBasuraAMazo(): boolean
    + moverCimientoAPila(cimiento: Cimiento, pilaDestino: PilaDelTableau): boolean
    + moverMazoABasura(): boolean
}

enum Valor {
    AS,
    DOS,
    TRES,
    CUATRO,
    CINCO,
    SEIS,
    SIETE,
    OCHO,
    NUEVE,
    DIEZ,
    JOTA,
    REINA,
    REY
}

enum Variante {
    KLONDIKE,
    SPIDER
}

class RunTimeException {}

ListaDeCartas "0" o--> "*" Carta : lista
Basura --|> ListaDeCartas
Carta *-.-> Valor
Carta *-.-> Palo
Palo "1" o--> "1" ColorPalo
InvalidMovementException "1"*-->"1" ErrorAlMover : motivo
InvalidMovementException --|> RunTimeException
Cimiento --|> ListaDeCartas
Klondike --|> Solitario

```

```
Spider --|> Solitario
SpiderFacil --|> Spider
SpiderDificil --|> Spider
Mazo --|> ListaDeCartas
PilaDelTableau --|> ListaDeCartas
Solitario "1"*-->"1" Variante
Solitario "1"*-->"1" Mazo
Solitario "1"*-->"4" Cimientos : cimientos
Solitario "1"*-->"7" PilaDelTableau :pilasTableau
Solitario "1"*-->"1" Basura
@enduml
```

Detalle de algunas pruebas

Se detallan algunos movimientos de las pruebas que se hicieron de los métodos de la clase Spider.

SpiderFacil

```
public void testMoverPilaAPila()
```

Se inicializa el juego

```
// Arrange
Spider spider = new SpiderFacil(Variante.SPIDER, Palo.CORAZONES, true);
spider.inicializarJuego();
```

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6	PILA 7	PILA 8	PILA 9
REY TRÉBOLES	7 TRÉBOLES	AS TRÉBOLES	8 TRÉBOLES	2 TRÉBOLES	10 TRÉBOLES	5 TRÉBOLES	REY TRÉBOLES	8 TRÉBOLES	3 TRÉBOLES
REINA TRÉBOLES	6 TRÉBOLES	REY TRÉBOLES	7 TRÉBOLES	AS TRÉBOLES	9 TRÉBOLES	4 TRÉBOLES	REINA TRÉBOLES	7 TRÉBOLES	2 TRÉBOLES
JOTA TRÉBOLES	5 TRÉBOLES	REINA TRÉBOLES	6 TRÉBOLES	REY TRÉBOLES	8 TRÉBOLES	3 TRÉBOLES	JOTA TRÉBOLES	6 TRÉBOLES	AS TRÉBOLES
10 TRÉBOLES	4 TRÉBOLES	JOTA TRÉBOLES	5 TRÉBOLES	REINA TRÉBOLES	7 TRÉBOLES	2 TRÉBOLES	10 TRÉBOLES	5 TRÉBOLES	REY TRÉBOLES
9 TRÉBOLES	3 TRÉBOLES	10 TRÉBOLES	4 TRÉBOLES	JOTA TRÉBOLES	6 TRÉBOLES	AS TRÉBOLES	9 TRÉBOLES	4 TRÉBOLES	REINA TRÉBOLES
8 TRÉBOLES	2 TRÉBOLES	9 TRÉBOLES	3 TRÉBOLES						

Quito las 2 primeras cartas de la pila 9 y doy vuelta el resto de cartas de la misma pila.

```
// Act
PilaDelTableau pilaNueve = spider.obtenerPilaDelTableau(9);
pilaNueve.extraerUltima();
pilaNueve.extraerUltima();
pilaNueve.darVueltaIndex(1);
pilaNueve.darVueltaIndex(0);
```

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6	PILA 7	PILA 8	PILA 9
REY TRÉBOLES	7 TRÉBOLES	AS TRÉBOLES	8 TRÉBOLES	2 TRÉBOLES	10 TRÉBOLES	5 TRÉBOLES	REY TRÉBOLES	8 TRÉBOLES	3 TRÉBOLES
REINA TRÉBOLES	6 TRÉBOLES	REY TRÉBOLES	7 TRÉBOLES	AS TRÉBOLES	9 TRÉBOLES	4 TRÉBOLES	REINA TRÉBOLES	7 TRÉBOLES	2 TRÉBOLES
JOTA TRÉBOLES	5 TRÉBOLES	REINA TRÉBOLES	6 TRÉBOLES	REY TRÉBOLES	8 TRÉBOLES	3 TRÉBOLES	JOTA TRÉBOLES	6 TRÉBOLES	AS TRÉBOLES
10 TRÉBOLES	4 TRÉBOLES	JOTA TRÉBOLES	5 TRÉBOLES	REINA TRÉBOLES	7 TRÉBOLES	2 TRÉBOLES	10 TRÉBOLES	5 TRÉBOLES	
9 TRÉBOLES	3 TRÉBOLES	10 TRÉBOLES	4 TRÉBOLES	JOTA TRÉBOLES	6 TRÉBOLES	AS TRÉBOLES	9 TRÉBOLES	4 TRÉBOLES	
8 TRÉBOLES	2 TRÉBOLES	9 TRÉBOLES	3 TRÉBOLES						

Muevo las ultimas 3 cartas de la pila 9 a la pila 8

```
PilaDelTableau pilaOcho = spider.obtenerPilaDelTableau(8);
//Muevo creando la pilaOcho del 8 al As
spider.moverPilaAPila(pilaNueve, pilaOcho, 3);
```

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6	PILA 7	PILA 8	PILA 9
REY TRÉBOLES	7 TRÉBOLES	AS TRÉBOLES	8 TRÉBOLES	2 TRÉBOLES	10 TRÉBOLES	5 TRÉBOLES	REY TRÉBOLES	8 TRÉBOLES	
REINA TRÉBOLES	6 TRÉBOLES	REY TRÉBOLES	7 TRÉBOLES	AS TRÉBOLES	9 TRÉBOLES	4 TRÉBOLES	REINA TRÉBOLES	7 TRÉBOLES	
JOTA TRÉBOLES	5 TRÉBOLES	REINA TRÉBOLES	6 TRÉBOLES	REY TRÉBOLES	8 TRÉBOLES	3 TRÉBOLES	JOTA TRÉBOLES	6 TRÉBOLES	
10 TRÉBOLES	4 TRÉBOLES	JOTA TRÉBOLES	5 TRÉBOLES	REINA TRÉBOLES	7 TRÉBOLES	2 TRÉBOLES	10 TRÉBOLES	5 TRÉBOLES	
9 TRÉBOLES	3 TRÉBOLES	10 TRÉBOLES	4 TRÉBOLES	JOTA TRÉBOLES	6 TRÉBOLES	AS TRÉBOLES	9 TRÉBOLES	4 TRÉBOLES	
8 TRÉBOLES	2 TRÉBOLES	9 TRÉBOLES	3 TRÉBOLES					3 TRÉBOLES	
								2 TRÉBOLES	
								AS TRÉBOLES	

```
testMoverPilaACimientos
```

Inicializo el juego, elimino las ultimas 2 cartas de la pila 9 y doy vuelta todas las cartas de la pila 8 y 9.

```
// Arrange
Spider spider = new SpiderFacil(Variante.SPIDER, Palo.CORAZONES, true);
spider.inicializarJuego();

// Act
PilaDelTableau pilaNueve = spider.obtenerPilaDelTableau(9);
pilaNueve.extraerUltima();
pilaNueve.extraerUltima();
pilaNueve.darVueltaIndex(1);
pilaNueve.darVueltaIndex(0);
PilaDelTableau pilaOcho = spider.obtenerPilaDelTableau(8);

//Muevo creando la pilaOcho del 8 al As
spider.moverPilaAPila(pilaNueve, pilaOcho, 3);

//Muevo de la pilaOcho a la pila 7

pilaOcho.darVueltaIndex(0);
pilaOcho.darVueltaIndex(1);
pilaOcho.darVueltaIndex(2);
pilaOcho.darVueltaIndex(3);
```

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6	PILA 7	PILA 8	PILA 9
REY TRÉBOLES	7 TRÉBOLES	AS TRÉBOLES	8 TRÉBOLES	2 TRÉBOLES	10 TRÉBOLES	5 TRÉBOLES	REY TRÉBOLES	8 TRÉBOLES	
REINA TRÉBOLES	6 TRÉBOLES	REY TRÉBOLES	7 TRÉBOLES	AS TRÉBOLES	9 TRÉBOLES	4 TRÉBOLES	REINA TRÉBOLES	7 TRÉBOLES	
JOTA TRÉBOLES	5 TRÉBOLES	REINA TRÉBOLES	6 TRÉBOLES	REY TRÉBOLES	8 TRÉBOLES	3 TRÉBOLES	JOTA TRÉBOLES	6 TRÉBOLES	
10 TRÉBOLES	4 TRÉBOLES	JOTA TRÉBOLES	5 TRÉBOLES	REINA TRÉBOLES	7 TRÉBOLES	2 TRÉBOLES	10 TRÉBOLES	5 TRÉBOLES	
9 TRÉBOLES	3 TRÉBOLES	10 TRÉBOLES	4 TRÉBOLES	JOTA TRÉBOLES	6 TRÉBOLES	AS TRÉBOLES	9 TRÉBOLES	4 TRÉBOLES	
8 TRÉBOLES	2 TRÉBOLES	9 TRÉBOLES	3 TRÉBOLES					3 TRÉBOLES	
								2 TRÉBOLES	
								AS TRÉBOLES	

Doy vuelta todas las cartas de la pila 7 y muevo todas las cartas de la pila 8 a la 7

```
PilaDelTableau pilaSiete = spider.obtenerPilaDelTableau(7);
spider.moverPilaAPila(pilaOcho, pilaSiete, 8);

// Ya tengo la pilaSiete con las cartas del Rey al As, muevo al Cimient
pilaSiete.darVueltaIndex(0);
pilaSiete.darVueltaIndex(1);
pilaSiete.darVueltaIndex(2);
pilaSiete.darVueltaIndex(3);

Cimiento cimientoCero = spider.obtenerCimiento(0);
spider.moverPilaACimiento(pilaSiete, cimientoCero);
```

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6	PILA 7	PILA 8	PILA 9
REY TRÉBOLES	7 TRÉBOLES	AS TRÉBOLES	8 TRÉBOLES	2 TRÉBOLES	10 TRÉBOLES	5 TRÉBOLES	REY TRÉBOLES		
REINA TRÉBOLES	6 TRÉBOLES	REY TRÉBOLES	7 TRÉBOLES	AS TRÉBOLES	9 TRÉBOLES	4 TRÉBOLES	REINA TRÉBOLES		
JOTA TRÉBOLES	5 TRÉBOLES	REINA TRÉBOLES	6 TRÉBOLES	REY TRÉBOLES	8 TRÉBOLES	3 TRÉBOLES	JOTA TRÉBOLES		
10 TRÉBOLES	4 TRÉBOLES	JOTA TRÉBOLES	5 TRÉBOLES	REINA TRÉBOLES	7 TRÉBOLES	2 TRÉBOLES	10 TRÉBOLES		
9 TRÉBOLES	3 TRÉBOLES	10 TRÉBOLES	4 TRÉBOLES	JOTA TRÉBOLES	6 TRÉBOLES	AS TRÉBOLES	9 TRÉBOLES		
8 TRÉBOLES	2 TRÉBOLES	9 TRÉBOLES	3 TRÉBOLES				8 TRÉBOLES		
							7 TRÉBOLES		
							6 TRÉBOLES		
							5 TRÉBOLES		
							4 TRÉBOLES		
							3 TRÉBOLES		
							2 TRÉBOLES		
							AS TRÉBOLES		

Y se mueve toda la pila 7 al Cimiento.

SpiderDifícil

```
public void moverPilaAPilaTest() {
```

Se inicializa el juego con Corazones y Tréboles.

```
// Arrange
ArrayList<Palo> palos = new ArrayList<Palo>();
palos.add(Palo.CORAZONES);
palos.add(Palo.TREBOLES);

var spider = new SpiderDifícil(Variante.SPIDER, palos, true);
spider.inicializarJuego();
```

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6	PILA 7	PILA 8	PILA 9
REY TRÉBOLES	7 TRÉBOLES	AS TRÉBOLES	8 TRÉBOLES	2 TRÉBOLES	10 TRÉBOLES	5 TRÉBOLES	REY TRÉBOLES	8 TRÉBOLES	3 TRÉBOLES

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6	PILA 7	PILA 8	PILA 9
REINA TRÉBOLES	6 TRÉBOLES	REY TRÉBOLES	7 TRÉBOLES	AS TRÉBOLES	9 TRÉBOLES	4 TRÉBOLES	REINA TRÉBOLES	7 TRÉBOLES	2 TRÉBOLES
JOTA TRÉBOLES	5 TRÉBOLES	REINA TRÉBOLES	6 TRÉBOLES	REY TRÉBOLES	8 TRÉBOLES	3 TRÉBOLES	JOTA TRÉBOLES	6 TRÉBOLES	AS TRÉBOLES
10 TRÉBOLES	4 TRÉBOLES	JOTA TRÉBOLES	5 TRÉBOLES	REINA TRÉBOLES	7 TRÉBOLES	2 TRÉBOLES	10 TRÉBOLES	5 TRÉBOLES	REY CORAZONES
9 TRÉBOLES	3 TRÉBOLES	10 TRÉBOLES	4 TRÉBOLES	JOTA TRÉBOLES	6 TRÉBOLES	AS TRÉBOLES	9 TRÉBOLES	4 TRÉBOLES	REINA CORAZONES
8 TRÉBOLES	2 TRÉBOLES	9 TRÉBOLES	3 TRÉBOLES						

Se intenta mover 5 cartas de la pila 5 a la pila 6, pero no están boca arriba. Ocurre una excepción.

```
boolean resultado = true;
try {
    spider.moverPilaAPila(spider.obtenerPilaDelTableau(6), spider.obtenerPilaDelTableau(5), 5);
}
catch (InvalidMovementException e) {
    resultado = false;
    assertEquals(ErrorAlMover.CARTA_A_MOVER_NO_BOCA_ARRIBA, e.obtenerMotivo());
}
```

Luego, se intenta mover 1 carta de la pila 1 a la pila 3. Resultado exitoso.

```
resultado = true;
try {
    spider.moverPilaAPila(spider.obtenerPilaDelTableau(1), spider.obtenerPilaDelTableau(3), 1);
} catch (InvalidMovementException e) {
    resultado = false;
}
assertTrue(resultado);
```

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6	PILA 7	PILA 8	PILA 9
REY TRÉBOLES	7 TRÉBOLES	AS TRÉBOLES	8 TRÉBOLES	2 TRÉBOLES	10 TRÉBOLES	5 TRÉBOLES	REY TRÉBOLES	8 TRÉBOLES	3 TRÉBOLES
REINA TRÉBOLES	6 TRÉBOLES	REY TRÉBOLES	7 TRÉBOLES	AS TRÉBOLES	9 TRÉBOLES	4 TRÉBOLES	REINA TRÉBOLES	7 TRÉBOLES	2 TRÉBOLES
JOTA TRÉBOLES	5 TRÉBOLES	REINA TRÉBOLES	6 TRÉBOLES	REY TRÉBOLES	8 TRÉBOLES	3 TRÉBOLES	JOTA TRÉBOLES	6 TRÉBOLES	AS TRÉBOLES
10 TRÉBOLES	4 TRÉBOLES	JOTA TRÉBOLES	5 TRÉBOLES	REINA TRÉBOLES	7 TRÉBOLES	2 TRÉBOLES	10 TRÉBOLES	5 TRÉBOLES	REY CORAZONES
9 TRÉBOLES	3 TRÉBOLES	10 TRÉBOLES	4 TRÉBOLES	JOTA TRÉBOLES	6 TRÉBOLES	AS TRÉBOLES	9 TRÉBOLES	4 TRÉBOLES	REINA CORAZONES
8 TRÉBOLES		9 TRÉBOLES	3 TRÉBOLES						
			2 TRÉBOLES						

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6	PILA 7	PILA 8	PILA 9
REY TRÉBOLES	7 TRÉBOLES	AS TRÉBOLES	8 TRÉBOLES	2 TRÉBOLES	10 TRÉBOLES	5 TRÉBOLES	REY TRÉBOLES	8 TRÉBOLES	3 TRÉBOLES
REINA TRÉBOLES	6 TRÉBOLES	REY TRÉBOLES	7 TRÉBOLES	AS TRÉBOLES	9 TRÉBOLES	4 TRÉBOLES	REINA TRÉBOLES	7 TRÉBOLES	2 TRÉBOLES
JOTA TRÉBOLES	5 TRÉBOLES	REINA TRÉBOLES	6 TRÉBOLES	REY TRÉBOLES	8 TRÉBOLES	3 TRÉBOLES	JOTA TRÉBOLES	6 TRÉBOLES	AS TRÉBOLES
10 TRÉBOLES	4 TRÉBOLES	JOTA TRÉBOLES	5 TRÉBOLES	REINA TRÉBOLES	7 TRÉBOLES	2 TRÉBOLES	10 TRÉBOLES	5 TRÉBOLES	REY CORAZONES
9 TRÉBOLES	3 TRÉBOLES	10 TRÉBOLES	4 TRÉBOLES	JOTA TRÉBOLES	6 TRÉBOLES	AS TRÉBOLES	9 TRÉBOLES	4 TRÉBOLES	REINA CORAZONES
8 TRÉBOLES		9 TRÉBOLES	3 TRÉBOLES						
			2 TRÉBOLES						

Se prueba el movimiento entre cartas de diferente color pero en orden descendente, moviendo el Jota de Tréboles de la pila 4 a la pila 9.

```
resultado = true;
try {
    spider.moverPilaAPila(spider.obtenerPilaDelTableau(4), spider.obtenerPilaDelTableau(9), 1);
} catch (InvalidMovementException e) {
    resultado = false;
}
assertTrue(resultado);
```

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6	PILA 7	PILA 8	PILA 9
REY TRÉBOLES	7 TRÉBOLES	AS TRÉBOLES	8 TRÉBOLES	2 TRÉBOLES	10 TRÉBOLES	5 TRÉBOLES	REY TRÉBOLES	8 TRÉBOLES	3 TRÉBOLES
REINA TRÉBOLES	6 TRÉBOLES	REY TRÉBOLES	7 TRÉBOLES	AS TRÉBOLES	9 TRÉBOLES	4 TRÉBOLES	REINA TRÉBOLES	7 TRÉBOLES	2 TRÉBOLES
JOTA TRÉBOLES	5 TRÉBOLES	REINA TRÉBOLES	6 TRÉBOLES	REY TRÉBOLES	8 TRÉBOLES	3 TRÉBOLES	JOTA TRÉBOLES	6 TRÉBOLES	AS TRÉBOLES
10 TRÉBOLES	4 TRÉBOLES	JOTA TRÉBOLES	5 TRÉBOLES	REINA TRÉBOLES	7 TRÉBOLES	2 TRÉBOLES	10 TRÉBOLES	5 TRÉBOLES	REY CORAZONES
9 TRÉBOLES	3 TRÉBOLES	10 TRÉBOLES	4 TRÉBOLES		6 TRÉBOLES	AS TRÉBOLES	9 TRÉBOLES	4 TRÉBOLES	REINA CORAZONES
8 TRÉBOLES		9 TRÉBOLES	3 TRÉBOLES						JOTA TRÉBOLES
			2 TRÉBOLES						

public void movimientoACimiento()

Inicializo el juego con Corazones y Tréboles.

```
// Arrange
ArrayList<Palo> palos = new ArrayList<Palo>();
palos.add(Palo.CORAZONES);
palos.add(Palo.TREBOLES);

var spider = new SpiderDifícil(Variante.SPIDER, palos, true);
spider.inicializarJuego();
```

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6	PILA 7	PILA 8	PILA 9
REY TRÉBOLES	7 TRÉBOLES	AS TRÉBOLES	8 TRÉBOLES	2 TRÉBOLES	10 TRÉBOLES	5 TRÉBOLES	REY TRÉBOLES	8 TRÉBOLES	3 TRÉBOLES
REINA TRÉBOLES	6 TRÉBOLES	REY TRÉBOLES	7 TRÉBOLES	AS TRÉBOLES	9 TRÉBOLES	4 TRÉBOLES	REINA TRÉBOLES	7 TRÉBOLES	2 TRÉBOLES
JOTA TRÉBOLES	5 TRÉBOLES	REINA TRÉBOLES	6 TRÉBOLES	REY TRÉBOLES	8 TRÉBOLES	3 TRÉBOLES	JOTA TRÉBOLES	6 TRÉBOLES	AS TRÉBOLES
10 TRÉBOLES	4 TRÉBOLES	JOTA TRÉBOLES	5 TRÉBOLES	REINA TRÉBOLES	7 TRÉBOLES	2 TRÉBOLES	10 TRÉBOLES	5 TRÉBOLES	REY CORAZONES
9 TRÉBOLES	3 TRÉBOLES	10 TRÉBOLES	4 TRÉBOLES	JOTA TRÉBOLES	6 TRÉBOLES	AS TRÉBOLES	9 TRÉBOLES	4 TRÉBOLES	REINA CORAZONES
8 TRÉBOLES	2 TRÉBOLES	9 TRÉBOLES	3 TRÉBOLES						

Se busca mover la pila 0 al cimient

```
boolean resultado = true;
try {
    spider.moverPilaACimient(spider.obtenerPilaDelTableau(0), spider.obtenerCimient(0));
} catch (InvalidMovementException e) {
    resultado = false;
    assertEquals(ErrorAlMover.PILA_INCOMPLETA_NO_PUEDE_IR_A_CIMIENT, e.obtenerMotivo());
}
assertFalse(resultado);
```

Se da vuelta las cartas de la pila 0 y pila 1. Luego, se mueven las cartas de la pila 1 a la pila 0 y se obtiene un resultado exitoso.

```
resultado = true;
for(int i = 0; i < 5; i++)
    spider.obtenerPilaDelTableau(0).darVueltaIndex(i);
for(int i = 0; i < 5; i++)
    spider.obtenerPilaDelTableau(1).darVueltaIndex(i);
try {
    spider.moverPilaAPila(spider.obtenerPilaDelTableau(1), spider.obtenerPilaDelTableau(0), 6);
} catch (InvalidMovementException e) {
    resultado = false;
}
assertTrue(resultado);
```

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6	PILA 7	PILA 8	PILA 9
REY TRÉBOLES		AS TRÉBOLES	8 TRÉBOLES	2 TRÉBOLES	10 TRÉBOLES	5 TRÉBOLES	REY TRÉBOLES	8 TRÉBOLES	3 TRÉBOLES
REINA TRÉBOLES		REY TRÉBOLES	7 TRÉBOLES	AS TRÉBOLES	9 TRÉBOLES	4 TRÉBOLES	REINA TRÉBOLES	7 TRÉBOLES	2 TRÉBOLES
JOTA TRÉBOLES		REINA TRÉBOLES	6 TRÉBOLES	REY TRÉBOLES	8 TRÉBOLES	3 TRÉBOLES	JOTA TRÉBOLES	6 TRÉBOLES	AS TRÉBOLES
10 TRÉBOLES		JOTA TRÉBOLES	5 TRÉBOLES	REINA TRÉBOLES	7 TRÉBOLES	2 TRÉBOLES	10 TRÉBOLES	5 TRÉBOLES	REY CORAZONES
9 TRÉBOLES		10 TRÉBOLES	4 TRÉBOLES	JOTA TRÉBOLES	6 TRÉBOLES	AS TRÉBOLES	9 TRÉBOLES	4 TRÉBOLES	REINA CORAZONES
8 TRÉBOLES		9 TRÉBOLES	3 TRÉBOLES						
7 TRÉBOLES									
6 TRÉBOLES									
5 TRÉBOLES									
4 TRÉBOLES									
3 TRÉBOLES									
2 TRÉBOLES									

Luego, se mueve la ultima carta de la pila 6 a la pila 0.

```
try {
    spider.moverPilaAPila(spider.obtenerPilaDelTableau(6), spider.obtenerPilaDelTableau(0), 1);
} catch (InvalidMovementException e) {
    resultado = false;
}
assertTrue(resultado);
```

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6	PILA 7	PILA 8	PILA 9
REY TRÉBOLES		AS TRÉBOLES	8 TRÉBOLES	2 TRÉBOLES	10 TRÉBOLES	5 TRÉBOLES	REY TRÉBOLES	8 TRÉBOLES	3 TRÉBOLES
REINA TRÉBOLES		REY TRÉBOLES	7 TRÉBOLES	AS TRÉBOLES	9 TRÉBOLES	4 TRÉBOLES	REINA TRÉBOLES	7 TRÉBOLES	2 TRÉBOLES
JOTA TRÉBOLES		REINA TRÉBOLES	6 TRÉBOLES	REY TRÉBOLES	8 TRÉBOLES	3 TRÉBOLES	JOTA TRÉBOLES	6 TRÉBOLES	AS TRÉBOLES
10 TRÉBOLES		JOTA TRÉBOLES	5 TRÉBOLES	REINA TRÉBOLES	7 TRÉBOLES	2 TRÉBOLES	10 TRÉBOLES	5 TRÉBOLES	REY CORAZONES

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6	PILA 7	PILA 8	PILA 9
9 TRÉBOLES		10 TRÉBOLES	4 TRÉBOLES	JOTA TRÉBOLES	6 TRÉBOLES		9 TRÉBOLES	4 TRÉBOLES	REINA CORAZONES
8 TRÉBOLES		9 TRÉBOLES	3 TRÉBOLES						
7 TRÉBOLES									
6 TRÉBOLES									
5 TRÉBOLES									
4 TRÉBOLES									
3 TRÉBOLES									
2 TRÉBOLES									
AS TRÉBOLES									

Finalmente, se mueve la pila 0 al cimiento de forma exitosa.

```
try {
    spider.moverPilaACimiento(spider.obtenerPilaDelTableau(0), spider.obtenerCimiento(0));
} catch (InvalidMovementException e) {
    resultado = false
}
assertTrue(resultado);
```