

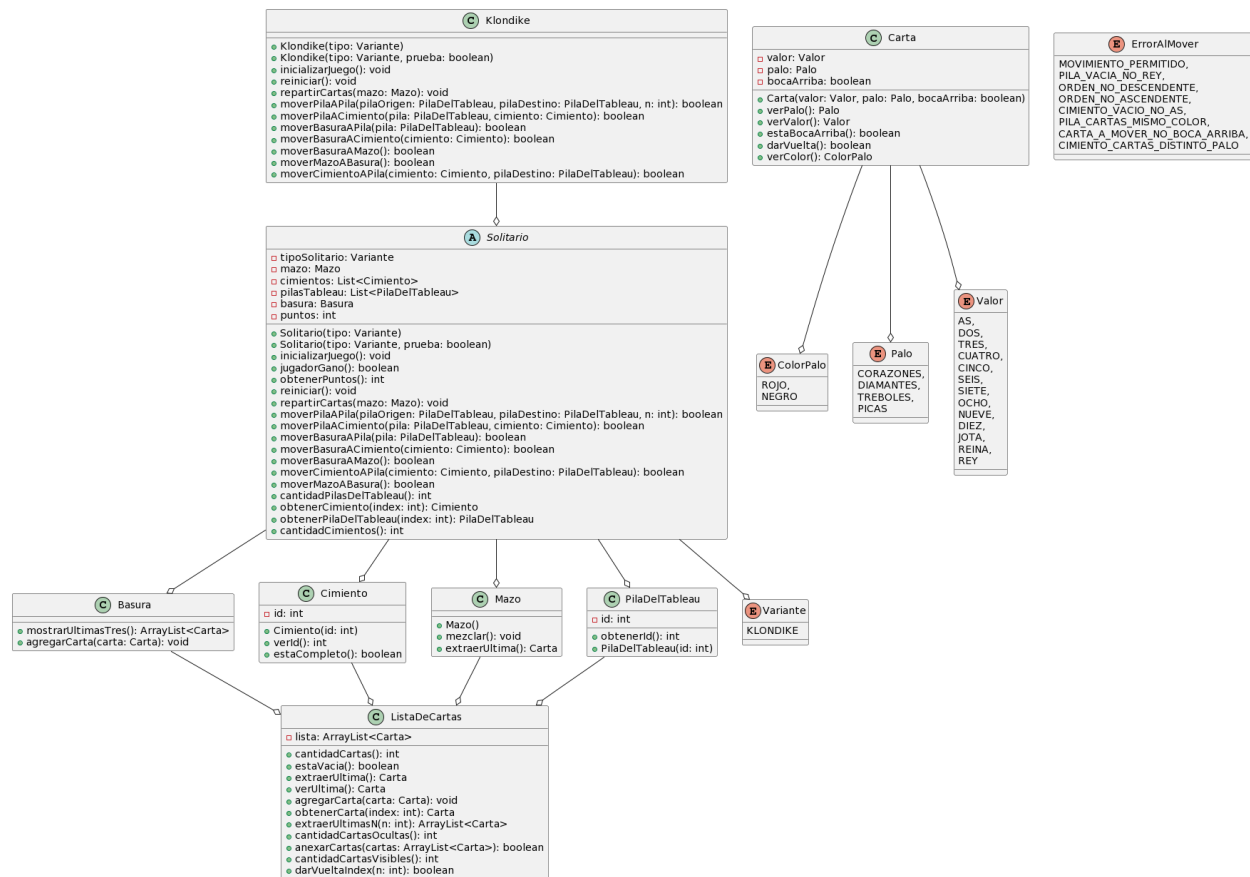
## ETAPA 1

### ALUMNOS:

Peña, Leandro	102298	lpena@fi.uba.ar
Spaltro, Francisco	102098	fspaltro@fi.uba.ar

CORRECTOR: DIEGO ESSAYA

## Diagrama UML



@startuml

```

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

```

```

class ListaDeCartas {

```

```

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

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

enum ColorPalo {
    ROJO,
    NEGRO
}

enum ErrorAlMover {
    MOVIMIENTO_PERMITIDO,
    PILA_VACIA_NO_REY,
    ORDEN_NO_DESCENDENTE,
    ORDEN_NO_ASCENDENTE,
    CIMIENTITO_VACIO_NO_AS,
    PILA_CARTAS_MISMO_COLOR,
    CARTA_A_MOVER_NO_BOCA_ARRIBA,
    CIMIENTITO_CARTAS_DISTINTO_PALO
}

class Klondike {
+ Klondike(tipo: Variante)
+ Klondike(tipo: Variante, prueba: boolean)
+ inicializarJuego(): void
+ reiniciar(): void
+ repartirCartas(mazo: Mazo): void
+ moverPilaAPila(pilaOrigen: PilaDelTableau, pilaDestino: PilaDelTableau, n: int): boolean
+ moverPilaACimientito(pila: PilaDelTableau, cimientito: Cimientito): boolean
+ moverBasuraAPila(pila: PilaDelTableau): boolean
+ moverBasuraACimientito(cimientito: Cimientito): boolean
+ moverBasuraAMazo(): boolean
+ moverMazoABasura(): boolean
+ moverCimientitoAPila(cimientito: Cimientito, pilaDestino: PilaDelTableau): boolean
}

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

enum Palo {
    CORAZONES,
    DIAMANTES,
    TREBOLES,
    PICAS
}

class PilaDelTableau {

```

```

- id: int
+ obtenerId(): int
+ PilaDelTableau(id: int)
}

abstract class Solitario {
- tipoSolitario: Variante
- mazo: Mazo
- cimientos: List<Cimiento>
- pilasTableau: List<PilaDelTableau>
- basura: Basura
- puntos: int
+ Solitario(tipo: Variante)
+ Solitario(tipo: Variante, prueba: boolean)
+ inicializarJuego(): void
+ jugadorGano(): boolean
+ obtenerPuntos(): int
+ 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
+ cantidadPilasDelTableau(): int
+ obtenerCimiento(index: int): Cimiento
+ obtenerPilaDelTableau(index: int): PilaDelTableau
+ cantidadCimientos(): int
}

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

enum Variante {
KLONDIKE
}

Basura --o ListaDeCartas
Carta --o Valor
Carta --o Palo
Carta --o ColorPalo
Cimiento --o ListaDeCartas
Klondike --o Solitario
Mazo --o ListaDeCartas
PilaDelTableau --o ListaDeCartas
Solitario --o Variante
Solitario --o Mazo
Solitario --o Cimiento
Solitario --o PilaDelTableau
Solitario --o Basura

@enduml

```

## Detalle de algunas pruebas

Dado el entorno especializado que se necesitaba para probar algunos métodos, se incluyeron más de una prueba en una misma función. Es por esto que se deja el presente apartado para clarificar algunos movimientos

## 1. `public void testMoverCartasEntreCimientoYPila()`

```
// Arrange
Klondike klondike = new Klondike(Variante.KLONDIKE, true);
klondike.inicializarJuego();
```

- El mazo se crea en orden as - rey, corazones, diamantes, tréboles, picas

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6
K picas	Q picas	10 picas	7 picas	3 picas	J tréboles	5 tréboles
	J picas	9 picas	6 picas	2 picas	10 tréboles	4 tréboles
		8 picas	5 picas	A picas	9 tréboles	3 tréboles
			4 picas	K tréboles	8 tréboles	2 tréboles
				Q tréboles	7 tréboles	1 tréboles
					6 tréboles	K diamantes
						Q diamantes

```
// Act
PilaDelTableau pilaAMover = klondike.obtenerPilaDelTableau(4);
Cimiento cimientoDestino = klondike.obtenerCimiento(0);

pilaAMover.extraerUltima();
pilaAMover.extraerUltima();
```

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6
K picas	Q picas	10 picas	7 picas	3 picas	J tréboles	5 tréboles
	J picas	9 picas	6 picas	2 picas	10 tréboles	4 tréboles
		8 picas	5 picas	A picas	9 tréboles	3 tréboles
			4 picas		8 tréboles	2 tréboles
					7 tréboles	1 tréboles
					6 tréboles	K diamantes
						Q diamantes

- Se completa el cimiento cero con las cartas de picas hasta la J

```
for (int i = 0; i < 3; i++)
    klondike.moverPilaACimiento(pilaAMover, cimientoDestino);
pilaAMover = klondike.obtenerPilaDelTableau(3);
boolean juegoTerminado = klondike.jugadorGano();
for (int i = 0; i < 4; i++)
    klondike.moverPilaACimiento(pilaAMover, cimientoDestino);
pilaAMover = klondike.obtenerPilaDelTableau(2);
for (int i = 0; i < 3; i++)
    klondike.moverPilaACimiento(pilaAMover, cimientoDestino);
pilaAMover = klondike.obtenerPilaDelTableau(1);
klondike.moverPilaACimiento(pilaAMover, cimientoDestino);
```

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6
K picas	Q picas	10 picas	7 picas	3 picas	J tréboles	5 tréboles
	J picas	9 picas	6 picas	2 picas	10 tréboles	4 tréboles
		8 picas	5 picas	A picas	9 tréboles	3 tréboles
			4 picas		8 tréboles	2 tréboles
					7 tréboles	1 tréboles

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6
					6 tréboles	K diamantes
						Q diamantes

CIMIENTO 0	CIMIENTO 1	CIMIENTO 2	CIMIENTO 3
A picas			
2 picas			
3 picas			
4 picas			
5 picas			
6 picas			
7 picas			
8 picas			
9 picas			
10 picas			
J picas			



juegoTerminado == false

- Se mueve la J de picas del cimientto a la pila 6 que tenía como última carta la Q de diamantes (distinto color)

```
PilaDelTableau pilaDestino = klondike.obtenerPilaDelTableau(6);
Cimiento cimientoOrigen = cimientoDestino;
boolean movioJ = klondike.moverCimientoAPila(cimientoOrigen, pilaDestino);
```

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6
K picas	Q picas	10 picas	7 picas	3 picas	J tréboles	5 tréboles
	J picas	9 picas	6 picas	2 picas	10 tréboles	4 tréboles
		8 picas	5 picas	A picas	9 tréboles	3 tréboles
			4 picas		8 tréboles	2 tréboles
					7 tréboles	1 tréboles
					6 tréboles	K diamantes
						Q diamantes
						J picas

CIMIENTO 0	CIMIENTO 1	CIMIENTO 2	CIMIENTO 3
A picas			
2 picas			
3 picas			
4 picas			
5 picas			
6 picas			
7 picas			
8 picas			
9 picas			
10 picas			



movioJ == true

- Se vuelve a mover la J de picas al cimientto y se intenta mover la Q de diamantes al mismo cimientto (distinto palo)

```
klondike.moverPilaACimientto(pilaDestino, cimienttoDestino);
PilaDelTableau pilaConQDeDiamantes = klondike.obtenerPilaDelTableau(6);
boolean movioQ = klondike.moverPilaACimientto(pilaConQDeDiamantes, cimienttoDestino);
```

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6
K picas	Q picas	10 picas	7 picas	3 picas	J tréboles	5 tréboles
	J picas	9 picas	6 picas	2 picas	10 tréboles	4 tréboles
		8 picas	5 picas	A picas	9 tréboles	3 tréboles
			4 picas		8 tréboles	2 tréboles
					7 tréboles	1 tréboles
					6 tréboles	K diamantes
						Q diamantes

CIMIENTO 0	CIMIENTO 1	CIMIENTO 2	CIMIENTO 3
A picas			
2 picas			
3 picas			
4 picas			
5 picas			
6 picas			
7 picas			
8 picas			
9 picas			
10 picas			
J picas			
Q diamantes			



movioQ == false

- Se intenta colocar una carta distinta de A (Q de picas) en un cimientto vacío

```
Cimiento cimienttoNuevo = klondike.obtenerCimiento(1);
boolean moverReinaCimientoVacío = klondike.moverPilaACimientto(klondike.obtenerPilaDelTableau(1), cimienttoNuevo);
```

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6
K picas	Q picas	10 picas	7 picas	3 picas	J tréboles	5 tréboles
	J picas	9 picas	6 picas	2 picas	10 tréboles	4 tréboles
		8 picas	5 picas	A picas	9 tréboles	3 tréboles
			4 picas		8 tréboles	2 tréboles
					7 tréboles	1 tréboles
					6 tréboles	K diamantes

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6
						Q diamantes

CIMIENTO 0	CIMIENTO 1	CIMIENTO 2	CIMIENTO 3
A picas	3 picas		
2 picas			
3 picas			
4 picas			
5 picas			
6 picas			
7 picas			
8 picas			
9 picas			
10 picas			
J picas			



moverReinaCimientoVacio == false

## 2. `public void testMoverBasuraAPila()`

- El mazo se crea en orden as - rey, corazones, diamantes, tréboles, picas

```
// Arrange
Klondike klondike = new Klondike(Variante.KLONDIKE, true);
klondike.inicializarJuego();
```

BASURA	PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6
	K picas	Q picas	10 picas	7 picas	3 picas	J tréboles	5 tréboles
		J picas	9 picas	6 picas	2 picas	10 tréboles	4 tréboles
			8 picas	5 picas	A picas	9 tréboles	3 tréboles
				4 picas	K tréboles	8 tréboles	2 tréboles
					Q tréboles	7 tréboles	1 tréboles
						6 tréboles	K diamantes
							Q diamantes

```
// Act
klondike.moverMazoABasura();
```

BASURA	PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6
J diamantes	K picas	Q picas	10 picas	7 picas	3 picas	J tréboles	5 tréboles
		J picas	9 picas	6 picas	2 picas	10 tréboles	4 tréboles
			8 picas	5 picas	A picas	9 tréboles	3 tréboles
				4 picas	K tréboles	8 tréboles	2 tréboles
					Q tréboles	7 tréboles	1 tréboles
						6 tréboles	K diamantes
							Q diamantes

BASURA	PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6
PilaDelTableau pilaDestino = klondike.obtenerPilaDelTableau(4); boolean seMovio = klondike.moverBasuraAPila(pilaDestino); Carta cartaMovida = pilaDestino.verUltima();							
BASURA	PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6
	K picas	Q picas	10 picas	7 picas	3 picas	J tréboles	5 tréboles
		J picas	9 picas	6 picas	2 picas	10 tréboles	4 tréboles
			8 picas	5 picas	A picas	9 tréboles	3 tréboles
				4 picas	K tréboles	8 tréboles	2 tréboles
					Q tréboles	7 tréboles	1 tréboles
					J diamantes	6 tréboles	K diamantes
							Q diamantes

### 3. public void testMoverBasuraACimienta()

- El mazo se crea en orden as - rey, corazones, diamantes, tréboles, picas

```
// Arrange
Klondike klondike = new Klondike(Variante.KLONDIKE, true);
klondike.inicializarJuego();
```

BASURA	PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6
	K picas	Q picas	10 picas	7 picas	3 picas	J tréboles	5 tréboles
		J picas	9 picas	6 picas	2 picas	10 tréboles	4 tréboles
			8 picas	5 picas	A picas	9 tréboles	3 tréboles
				4 picas	K tréboles	8 tréboles	2 tréboles
					Q tréboles	7 tréboles	1 tréboles
						6 tréboles	K diamantes
							Q diamantes

```
// Act
for (int i = 0; i < 11; i++)
    klondike.moverMazoABasura();
```

BASURA	PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6
J diamantes	K picas	Q picas	10 picas	7 picas	3 picas	J tréboles	5 tréboles
10 diamantes		J picas	9 picas	6 picas	2 picas	10 tréboles	4 tréboles
9 diamantes			8 picas	5 picas	A picas	9 tréboles	3 tréboles
8 diamantes				4 picas	K tréboles	8 tréboles	2 tréboles
7 diamantes					Q tréboles	7 tréboles	1 tréboles
6 diamantes						6 tréboles	K diamantes
5 diamantes							Q diamantes
4 diamantes							
3 diamantes							
2 diamantes							
A diamantes							



```

Cimiento cimientoDestino = klondike.obtenerCimiento(0);
boolean seMovio = klondike.moverBasuraACimiento(cimientoDestino);
Carta cartaMovida = cimientoDestino.verUltima();

```

BASURA	PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6
J diamantes	K picas	Q picas	10 picas	7 picas	3 picas	J tréboles	5 tréboles
10 diamantes		J picas	9 picas	6 picas	2 picas	10 tréboles	4 tréboles
9 diamantes			8 picas	5 picas	A picas	9 tréboles	3 tréboles
8 diamantes				4 picas	K tréboles	8 tréboles	2 tréboles
7 diamantes					Q tréboles	7 tréboles	1 tréboles
6 diamantes						6 tréboles	K diamantes
5 diamantes							Q diamantes
4 diamantes							
3 diamantes							
2 diamantes							

CIMIENTO 0	CIMIENTO 1	CIMIENTO 2	CIMIENTO 3
A diamantes			



seMovio == true

#### 4. `public void testMoverPilaAPilaVacía()`

- El mazo se crea en orden as - rey, corazones, diamantes, tréboles, picas

```

// Arrange
Klondike klondike = new Klondike(Variante.KLONDIKE, true);
klondike.inicializarJuego();

```

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6
K picas	Q picas	10 picas	7 picas	3 picas	J tréboles	5 tréboles
	J picas	9 picas	6 picas	2 picas	10 tréboles	4 tréboles
		8 picas	5 picas	A picas	9 tréboles	3 tréboles
			4 picas	K tréboles	8 tréboles	2 tréboles
				Q tréboles	7 tréboles	1 tréboles
					6 tréboles	K diamantes

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6
						Q diamantes

```
klondike.obtenerPilaDelTableau(0).extraerUltima();
```

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6
	Q picas	10 picas	7 picas	3 picas	J tréboles	5 tréboles
	J picas	9 picas	6 picas	2 picas	10 tréboles	4 tréboles
		8 picas	5 picas	A picas	9 tréboles	3 tréboles
			4 picas	K tréboles	8 tréboles	2 tréboles
				Q tréboles	7 tréboles	1 tréboles
					6 tréboles	K diamantes
						Q diamantes

```
boolean seMovio = klondike.moverPilaAPila(klondike.obtenerPilaDelTableau(1), klondike.obtenerPilaDelTableau(0), 1);
```

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6
<del>J picas</del>	Q picas	10 picas	7 picas	3 picas	J tréboles	5 tréboles
	J picas	9 picas	6 picas	2 picas	10 tréboles	4 tréboles
		8 picas	5 picas	A picas	9 tréboles	3 tréboles
			4 picas	K tréboles	8 tréboles	2 tréboles
				Q tréboles	7 tréboles	1 tréboles
					6 tréboles	K diamantes
						Q diamantes



seMovio == false

## 5. `public void testMoverPilaAPilaPorColor()`

- El mazo se crea en orden as - rey, corazones, diamantes, tréboles, picas

```
// Arrange
Klondike klondike = new Klondike(Variante.KLONDIKE, true);
klondike.inicializarJuego();
```

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6
K picas	Q picas	10 picas	7 picas	3 picas	J tréboles	5 tréboles
	J picas	9 picas	6 picas	2 picas	10 tréboles	4 tréboles
		8 picas	5 picas	A picas	9 tréboles	3 tréboles
			4 picas	K tréboles	8 tréboles	2 tréboles
				Q tréboles	7 tréboles	1 tréboles
					6 tréboles	K diamantes
						Q diamantes

```
// Act
```

```
boolean seMovio = klondike.moverPilaAPila(klondike.obtenerPilaDelTableau(6), klondike.obtenerPilaDelTableau(0), 1);
```

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6
K picas	Q picas	10 picas	7 picas	3 picas	J tréboles	5 tréboles
Q diamantes	J picas	9 picas	6 picas	2 picas	10 tréboles	4 tréboles
		8 picas	5 picas	A picas	9 tréboles	3 tréboles
			4 picas	K tréboles	8 tréboles	2 tréboles
				Q tréboles	7 tréboles	1 tréboles
					6 tréboles	K diamantes



seMovio == true

```
seMovio = klondike.moverPilaAPila(klondike.obtenerPilaDelTableau(0), klondike.obtenerPilaDelTableau(6), 1);
```

PILA 0	PILA 1	PILA 2	PILA 3	PILA 4	PILA 5	PILA 6
K picas	Q picas	10 picas	7 picas	3 picas	J tréboles	5 tréboles
Q diamantes	J picas	9 picas	6 picas	2 picas	10 tréboles	4 tréboles
		8 picas	5 picas	A picas	9 tréboles	3 tréboles
			4 picas	K tréboles	8 tréboles	2 tréboles
				Q tréboles	7 tréboles	1 tréboles
					6 tréboles	K diamantes
						Q diamantes



seMovio == false