# Diagrama de clases:

**Columna:**

* ArrayList<Celda> celdas
* int qtyCeldas
* int columnaID
* int marcador
* int ANCHO = 60
* int MARGEN\_SUP = 30
* int MARGEN\_IZQ = 10
* Columna(int, int)
* booln recibirFicha(Color)
* int getRowsNumber()
* int getQtyTokens()
* int getColumnWidth()
* int getUpperMargin()
* int getLeftMargin()
* boolean isFull()
* void unloadColumn()
* void draw()
* String toString()

**Tablero:**

* Columna[] columnas
* boolean estaLleno = f
* Tablero(int, int)
* bln recibirFicha(Color,int)
* void unloadBoard()
* int getQtyTokens()
* int getColumnsNumber()
* int getRowsNumber()
* int getSlotsNumber()
* int getBoardColumnWidth()
* int getBoardUpperMargin()
* int getBoardLeftMargin()
* String toString()

**Partida:**

* Tablero tablero
* Ellipse marcador
* boolean esTurnoAzul
* String nombreJugador1
* String nombreJugador2
* Canvas gameCanvas
* int linea
* int columnaActiva = 0
* Color COLOR\_JUGADOR1
* Color COLOR\_JUGADOR2
* Color COLOR\_PANELES
* int ANCHO
* int MARGEN\_IZQ
* int MARGEN\_SUP
* enum Dificultad
* Partida(Dificultad)
* JPanel crearPanelSuperior()
* JPanel crearPanelInferior()
* int ejecutarTurno()
* void reiniciarPartida()
* int[] parametrosJuego(Dific.)
* void leftButtonPressed()
* void rightButtonPressed()
* void playButtonPressed()
* IniciarPartida()
* CerrarPartida()
* String toString()
* void draw()

**Jugador:**

* Nombre
* ColorFichas
* JugarTurno()
* Desplegar (toString())
* Desplegar (draw())

**Interface: Drawable**

* draw()

**Ficha:**

* int anchoLabio
* Ellipse cuerpo
* Ellipse borde
* Ellipse centro
* Ficha (int, int, int, Color)
* void setColor(Color)
* void highlightToken()
* void drawHole()
* String toString()
* void draw()

**Celda:**

* Rectangle celda
* Ficha ficha
* boolean estáLlena = f
* int BORDE = 5
* int celdaID
* Celda(dbl, dbl, int, int)
* bool recibirFicha(Color)
* void unloadSlot()
* bool estaLlena()
* void draw()
* String toString()

## Cont…

**Color (library):**

* Red
* green
* blue
* Color(int, int, int)
* getRed()
* getBlue()
* getGreen()

**Line (library) < Shape**

* X1
* Y1
* X2
* Y2
* Line(double, double double, double)
* getX()
* getY()
* getHeight()
* getWidth()
* setColor()
* translate()
* grow()
* toStirng()
* draw()
* paintShape()

**Rectangle (library) < Shape**

* Color
* filled
* x
* y
* width
* height
* Rectangle()
* Rectangle(dbl,dbl,dbl,dbl)
* getX()
* getY()
* getHeight()
* getWidth()
* translate()
* grow()
* setColor()
* draw()
* fill()
* toStirng()
* paintShape()

**Interface: Shape (library)**

* getX()
* getY()
* getHeight()
* getWidth()
* paintShape()

**Picture (library) < Shape**

* image (BufferedImage)
* label (JLabel)
* source (String)
* x (double)
* y (double)
* xGrow (double)
* yGrow (double)
* Picture()
* Picture(double, double)
* Picture(String)
* load(String)
* getX()
* getY()
* getMaxX()
* getMaxY()
* getHeight()
* getWidth()
* pixels()
* getGrayLevels()
* Picture(int[][])
* toString()
* getColorAt(int)
* getColorAt(int, int)
* setColorAt(int, Color)
* setColorAt(int, int, Color)
* translate(double, double)
* grow(double, double)
* draw()
* paintShape()

**Canvas (library)**

* Canvas (static)
* shapes (ArrayList)
* background (buffImage)
* frame (JFrame)
* component (CanvasCmp)
* MIN\_SIZE (int)
* MARGIN (int)
* LOCATION\_OFFSET (int)
* canvasCmp (class)
* Canvas()
* getInstance()
* show()
* repaint()
* pause()
* snapshot()
* saveToDisk()

**CanvasCmp (library) << JComponent**

* paintComponent()
* getPreferredSize()

**Text (library) < Shape**

* color (Color)
* label (JLabel)
* x (double)
* y (double)
* xGrow (double)
* yGrow (double)
* Text (dbl, dbl, String)
* getX()
* getY()
* getHeight()
* getWidth()
* translate()
* grow()
* setColor()
* draw()
* fill()
* toStirng()
* paintShape()

**Ellipse (library) < Shape**

* Color
* filled
* x
* y
* width
* height
* Ellipse()
* Ellipse(dbl,dbl,dbl,dbl)
* getX()
* getY()
* getHeight()
* getWidth()
* translate()
* grow()
* setColor()
* draw()
* fill()
* toString()
* paintShape()