File - C:\aprendizado\java\nélioUdemy\chess-system\src\chess\pieces\Pawn.java

```java
package chess.pieces;

import boardgame.Board;
import boardgame.Position;
import chess.ChessMatch;
import chess.ChessPiece;
import chess.Color;

import java.util.Arrays;

public class Pawn extends ChessPiece {

    public Pawn(Board board, Color color) {
        super(board, color);
    }


    @Override
    public boolean[][] possibleMoves() {
        boolean[][] mat = new boolean[getBoard().getRows()][getBoard().getColumns()];

        int pieceStep = (getColor() == Color.BLACK) ? 1: -1;

        Position p = new Position(position.getRow() + pieceStep, position.getColumn());
        if (getBoard().positionExists(p) && !getBoard().thereIsAPiece(p)) {
            mat[p.getRow()][p.getColumn()] = true;
            if (getMoveCount() == 0) {
                p.setRow(p.getRow() + pieceStep);
                if (getBoard().positionExists(p) && !getBoard().thereIsAPiece(p))
                    mat[p.getRow()][p.getColumn()] = true;
                p.setRow(p.getRow() - pieceStep);
            }
        }
        p.setColumn(p.getColumn()-1);
        if (getBoard().positionExists(p) && isThereOpponentPiece(p))
            mat[p.getRow()][p.getColumn()] = true;
        p.setColumn(p.getColumn()+2);
        if (getBoard().positionExists(p) && isThereOpponentPiece(p))
            mat[p.getRow()][p.getColumn()] = true;

        return mat;
    }

    @Override
    public String toString() {
        return "P";
    }
}

```