**[UNIVERSIDAD NACIONAL JOSÉ MARÍA ARGUEDAS](http://unajma.edu.pe/)**





PROYECTO:

**Juego de tetris**

**MANUAL DEL PROGRAMADOR**

PROYECTO REALIZADO POR:

LIMA ROMÁN VÍCTOR ALBERTO

MERINO HURTADO JUAN CARLOS

**Trabajo presentado como presentación final del curso de**

**INGENIERÍA DE SOFTWARE II**

[**UNIVERSIDAD NACIONAL JOSÉ MARÍA ARGUEDAS**](http://unajma.edu.pe/)

**ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS**

**INGENIERIA DE SOFTWAREII**

**ANDAHUAYLAS - APURIMAC**

**2015**

**ÍNDICE (MANUAL DEL PROGRAMADOR)**

1. INTRODUCCIÓN AL MANUAL DEL PROGRAMADOR ---------------------1
2. Introducción -----------------------------------------------1

2. PROGRAMACIÓN -----------------------------------------------------------------------2

1. fundamentos de programación-------------------------1

2. programación del programa principal--------------- -2

**1.- INTRODUCCIÓN AL MANUAL DEL PROGRAMADOR.**

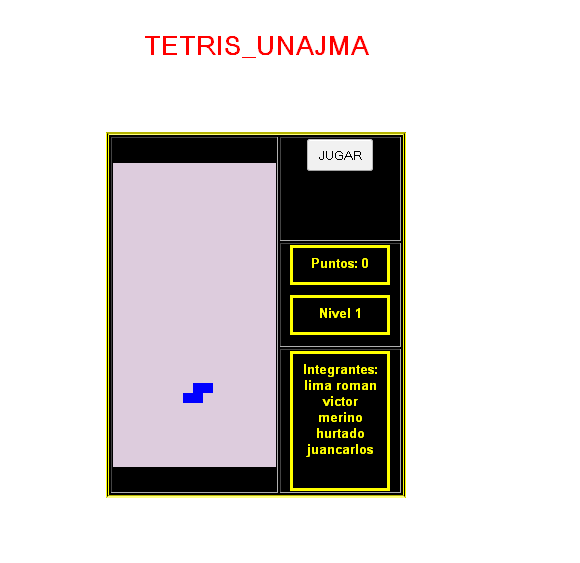
**1.- Introducción.**

El propósito de este manual del programador, es dar al conocer al lector los códigos fuentes del programa realizado .Para ello tratamos de forma amena y concisa cada uno de los códigos junto con la programación utilizada en el desarrollo del software, esto con el fin de que el usuario pueda modificar al gusto alguno de los valores y parámetros funcionales que se encuentran expuestas en la programación del juego tetris

**2.-PROGRAMACION**

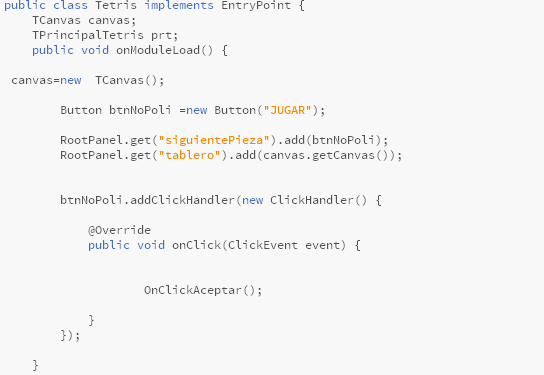
**1.- Fundamentos de programación.**

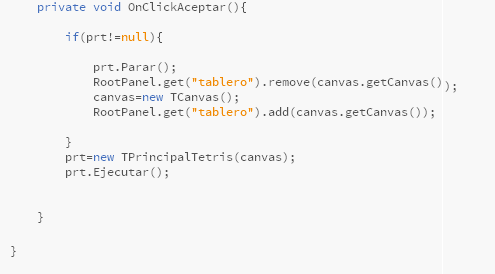
Para iniciar el programa (juego) es necesario después de compilar abrir el archivo war y dentro de ello el archivo teris.thml



**Crear la clase:**

En esta parte lo que vamos hacer es crear el proyecto en el eclipse que esta con el nombre de “tetris”.





**Creamos el formulario:**



**Creamos la clase TCanvas:**

La clase TCanvas es donde vamos a dar dimensiones como altura anchura y el contexto

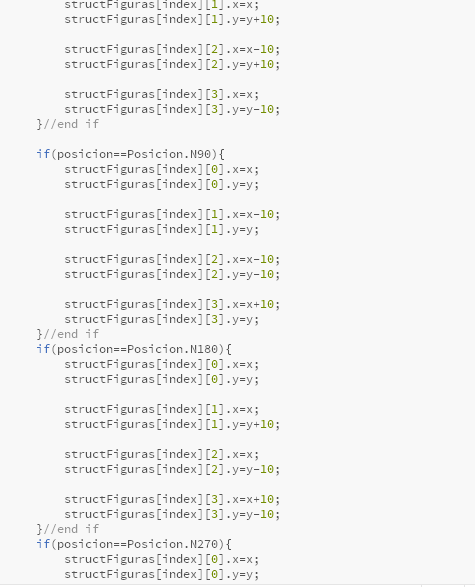
Y lo vamos a crear el canvas.

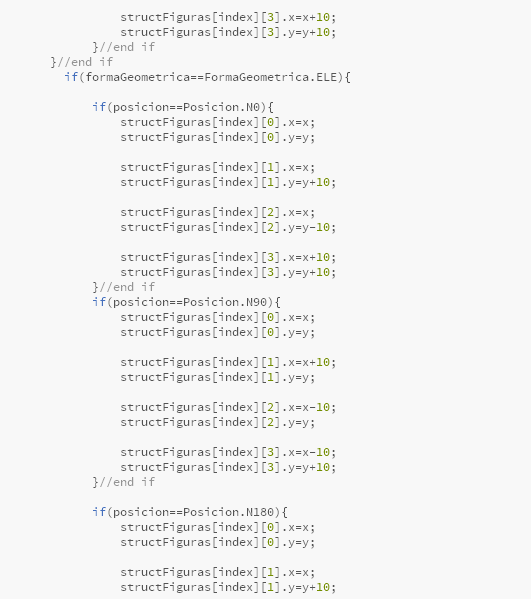


**Creamos la clase Forma:**

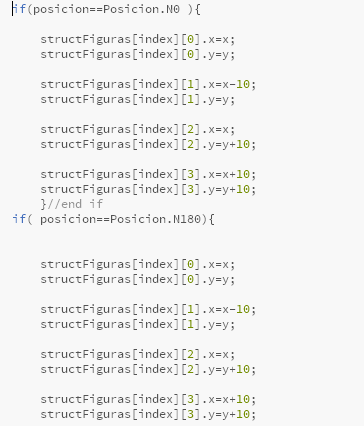
****

****

****

****

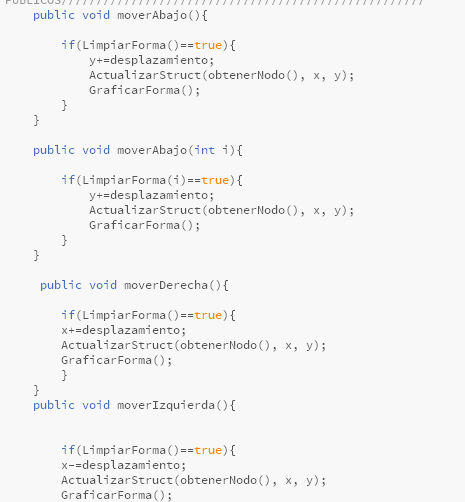
****

****

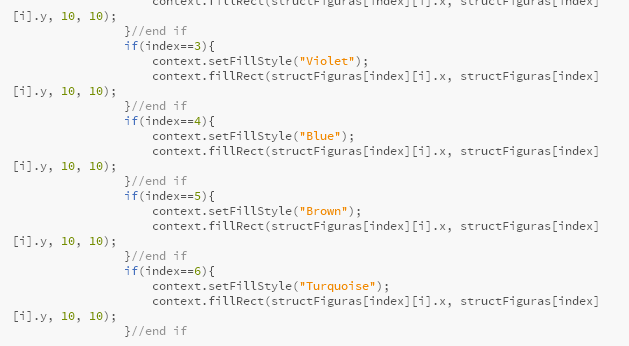
****

****

****

****

****

****

****

**TPrincipalTetris.java**

**public TPrincipalTetris(TCanvas \_canvas){**

**canvas=\_canvas;//constructor asignas un valor y te devuelve canvas**

**listFormas=new ArrayList<TForma>();**

**abajo=false;**

**}//end function**

**public void puntuacion(){**

**}**

**public void Music(){**

**audiofondo=Audio.createIfSupported();**

**audiofondo.setSrc("Trance.mp3");**

**}**

**public void musicrotacion(){**

**audiofondo1=Audio.createIfSupported();**

**audiofondo1.setSrc("turn.wav");**

**}**

**public void musiavajo(){**

**audiofondo3=Audio.createIfSupported();**

**audiofondo3.setSrc("drop.wav");**

**}**

**public void musitralado(){**

**audiofondo2=Audio.createIfSupported();**

**audiofondo2.setSrc("move.wav");**

**}**

**public void Parar(){ //crea una clase parar para que cuando la figura llegue a la base de pare y el tienpo se cansela**

**//Cancelar el Timer GC**

**timer.cancel();**

**timer.run();**

**}**

**public void Ejecutar(){**

**Music();**

**audiofondo.play();**

**CondicionesAleatoriasForma();**

**timer=new Timer() { //crear**

**@Override**

**public void run() {**

**//forma.girarDerecha();**

**if(restricciones(forma)==1){**

**//if (JuegoTerminado(forma)) Window.alert("rerer1111");**

**}**

**else{**

**//Window.alert("rerer2222");**

**//if (restricciones(forma))**

**forma.moverAbajo();**

**}**

**}};**

**timer.scheduleRepeating(400);**

**eventosDelTeclado();**

**//forma.moverAbajo();**

**}//end function**

**private int restriccionesAbajo(TForma \_forma){**

**int estado=0;**

**Nodo[] \_nodo=\_forma.getStructura();**

**for(int i=0;i<4; i++){**

**if(\_nodo[i].y+10>=canvas.getContext().getCanvas().getHeight()){**

**estado=1;**

**break;**

**}**

**}**

**return estado;**

**}//end function**

**//--------------------------PARA LA DERECHAA**

**private int restriccionesDerecha(TForma \_forma){**

**int estado=0;**

**Nodo[] \_nodo=\_forma.getStructura();**

**for(int i=0;i<4; i++){**

**if(\_nodo[i].x+10>=160){**

**estado=1;**

**break;**

**}**

**}**

**return estado;**

**}**

**//------------------------ PARA LA ESQUIERDA.................**

**private int restriccionesEsquierda(TForma \_forma){**

**int estado=0;**

**Nodo[] \_nodo=\_forma.getStructura();**

**for(int i=0;i<4; i++){**

**if(\_nodo[i].x-10<=-10){**

**estado=1;**

**break;**

**}**

**}**

**return estado;**

**}**

**private void CrearFigura(){**

**int seleccionado = (int) Math.round((Math.random()\*6));**

**//0,1,2La clase Math contiene m�todos para realizar operaciones num�ricas b�sicas**

**FormaGeometrica frm=FormaGeometrica.ELE\_INV;**

**if(seleccionado==0){**

**frm=FormaGeometrica.ELE\_INV;**

**}**

**else{**

**if(seleccionado==1)**

**{**

**frm=FormaGeometrica.ELE;**

**}**

**else{**

**if(seleccionado==2)**

**{**

**frm=FormaGeometrica.ZETA;**

**}**

**else{**

**if(seleccionado==3){**

**frm=FormaGeometrica.TE;**

**}**

**else{**

**if(seleccionado==4){**

**frm=FormaGeometrica.ESE;**

**}**

**else {**

**if(seleccionado==5){**

**frm=FormaGeometrica.CUADRO;**

**}**

**else{**

**if(seleccionado==6){**

**frm=FormaGeometrica.LINIA;**

**}**

**}**

**}**

**}**

**}**

**}**

**}**

**forma=new TForma(canvas,frm,Posicion.N0,80,0);**

**listFormas.add(forma);**

**}//end function**

**//Todas las restricciones**

**private int restricciones(TForma \_forma){**

**int estado=0;**

**if(restriccionesAbajo(\_forma)==1){**

**estado=1;**

**CrearFigura();**

**}//end if**

**if(restriccionesFiguras()==1){**

**estado=1;**

**CrearFigura();**

**}//end if**

**return estado;**

**}//end class**

**private int InterseccionX(double \_x1,double \_w1,double \_x2,double \_w2){**

**int estado=0;**

**if(\_x1<\_x2+\_w2 && \_x2+\_w2<\_x1+\_w1){**

**//Window.alert("1");**

**estado=1;**

**}//end if**

**else if(\_x1<\_x2 && \_x2<\_x1+\_w1){**

**//Window.alert("2");**

**estado=1;**

**}//end if**

**else if(\_x2<\_x1+\_w1 && \_x1+\_w1<\_x2+\_w2){**

**//Window.alert("3");**

**estado=1;**

**}//end if**

**else if(\_x2<=\_x1 && \_x1<\_x2+\_w2){**

**//Window.alert("4");**

**estado=1;**

**}//end if**

**return estado;**

**}//end function**

**private int InterseccionY(double \_y1,double \_h1,double \_y2,double \_h2){**

**int estado=0;**

**if(\_y1<=\_y2+\_h2 && \_y2+\_h2<=\_y1+\_h1){**

**estado=1;**

**}//end if**

**else if(\_y1<=\_y2 && \_y2<=\_y1+\_h1){**

**estado=1;**

**}//end if**

**else if(\_y2<=\_y1+\_h1 && \_y1+\_h1<=\_y2+\_h2){**

**estado=1;**

**}//end if**

**else if(\_y2<=\_y1 && \_y1<=\_y2+\_h2){**

**estado=1;**

**}//end if**

**return estado;**

**}**

**private int restriccionesFiguras(){**

**int estado=0;**

**if(listFormas.size()<2)return estado;**

**TForma \_actual=listFormas.get(listFormas.size()-1);**

**salida:**

**for(TForma fgr:listFormas){**

**if(fgr==\_actual)**

**continue;**

**for(int k=0;k<4;k++){**

**Nodo[] \_nodoFormaTemp=fgr.getStructura();**

**for(int i=0;i<4;i++){**

**Nodo[] \_nodoActual=\_actual.getStructura();**

**double \_x1=\_nodoActual[i].x;**

**double \_w1=10;**

**double \_y1=\_nodoActual[i].y;**

**double \_h1=10;**

**//double \_x2=pacman.getX();**

**double \_x2=\_nodoFormaTemp[k].x;**

**double \_w2=10;**

**double \_y2=\_nodoFormaTemp[k].y;**

**double \_h2=10;**

**//ALGORITMO DE INTERSECCION DE AREAS**

**if(InterseccionX(\_x1,\_w1,\_x2,\_w2)==1 && InterseccionY(\_y1,\_h1,\_y2,\_h2)==1){**

**estado=1;**

**break salida; //salimos del bucle inmediatamente**

**}//end if**

**}//end if**

**}//end for**

**}//end for**

**return estado;**

**} //end function**

**private int restriccionesR(TForma \_forma){**

**int estado=0;**

**if(restriccionesDerecha(\_forma)==1){**

**estado=1;**

**}//end if**

**return estado;**

**}//end class**

**private int restriccionesL(TForma \_forma){**

**int estado=0;**

**if(restriccionesEsquierda(\_forma)==1){**

**estado=1;**

**}//end if**

**return estado;**

**} //end class**

**private void CondicionesAleatoriasForma(){**

**CrearFigura();**

**}//end class**

**public void limpiarFila(){**

**for(int i=0;i<=4;i++){**

**for(int j=0;j<=4;j++){**

**}**

**}**

**}//end class**

**public void eventosDelTeclado(){**

**Canvas nativeCanvas=canvas.getCanvas();**

**nativeCanvas.addKeyDownHandler(new KeyDownHandler() {**

**@Override**

**public void onKeyDown(KeyDownEvent event) {**

**// TODO Auto-generated method stub**

**if(event.getNativeKeyCode()==KeyCodes.KEY\_RIGHT){**

**//forma.girarDerecha();**

**if(restriccionesR(forma)==1){**

**//Window.alet("");**

**}**

**else{**

**forma.moverDerecha();**

**musitralado();**

**audiofondo2.play();**

**}**

**}**

**if(event.getNativeKeyCode()==KeyCodes.KEY\_DOWN){**

**//forma.girarDerecha();**

**if(restricciones(forma)==1){**

**//Window.alert("rerer1111");**

**}**

**else{**

**//Window.alert("rerer2222");**

**ComprabarEliminarBase(300);**

**//ComprabarEliminarBase29(290);**

**//ComprabarEliminarBase(200);**

**forma.moverAbajo();**

**musitralado();**

**audiofondo2.play();**

**}**

**}**

**if(event.getNativeKeyCode()==KeyCodes.KEY\_LEFT){**

**//forma.girarDerecha();**

**if(restriccionesL(forma)==1){**

**//Window.alert("");**

**}**

**else{**

**forma.moverIzquierda();**

**musitralado();**

**audiofondo2.play();**

**}**

**}//end if**

**if(event.getNativeKeyCode()==65){//Tecla A**

**if(restriccionesR(forma)==1 ){**

**//Window.alert("");**

**}**

**else{**

**if(restriccionesL(forma)==1){**

**}**

**else**

**{**

**forma.girarDerecha();**

**musicrotacion();**

**audiofondo1.play();**

**}**

**}**

**}//end if**

**if(event.getNativeKeyCode()==83){//Tecla S**

**if(restriccionesL(forma)==1 ){**

**//Window.alert("");**

**}**

**else{**

**if(restriccionesR(forma)==1){**

**} else**

**{**

**forma.girarIzquierda();**

**musicrotacion();**

**audiofondo1.play();**

**}**

**}**

**}//end if**

**}**

**});**

**}//end class**

**//end clas**

**public void pausa(){**

**}**

**private void ComprabarEliminarBase(int rowFin){**

**//int h=canvas.getContext().getCanvas().getHeight();**

**int ctd=0;**

**for(int i=0; i<listFormas.size(); i++){**

**Nodo[] \_nodo= listFormas.get(i).getStructura();**

**for(int k=0;k<\_nodo.length;k++){**

**int yh=(int) (\_nodo[k].y+10);**

**/\***

**if(yh>=h){**

**ctd++;**

**//Window.alert(Integer.toString(ctd));**

**}**

**\*/**

**if(rowFin-10<yh && yh<=rowFin){**

**ctd++;**

**//Window.alert(Integer.toString(ctd)+"dad"+Integer.toString(yh));**

**}**

**}**

**//Window.alert(Integer.toString(yh)+"opo"+Integer.toString(h));**

**}**

**if(ctd==16){**

**musica();**

**EliminarBase(rowFin);**

**}**

**} //function**

**private void ComprabarEliminarBase29(int rowFin){**

**//int h=canvas.getContext().getCanvas().getHeight();**

**int ctd=0;**

**for(int i=0; i<listFormas.size(); i++){**

**Nodo[] \_nodo= listFormas.get(i).getStructura();**

**for(int k=0;k<\_nodo.length;k++){**

**int yh=(int) (\_nodo[k].y+10);**

**/\***

**if(yh>=h){**

**ctd++;**

**//Window.alert(Integer.toString(ctd));**

**}**

**\*/**

**if(rowFin-10<yh && yh<=rowFin){**

**ctd++;**

**//Window.alert(Integer.toString(ctd)+"dad"+Integer.toString(yh));**

**}**

**}**

**//Window.alert(Integer.toString(yh)+"opo"+Integer.toString(h));**

**}**

**if(ctd==5){**

**EliminarBase29(rowFin);**

**}**

**} //function**

**private void EliminarBase(int rowFin){**

**if(rowFin==300){ //30**

**for(int i=0; i<listFormas.size(); i++){**

**//if(listFormas.get(i)==forma)continue;**

**listFormas.get(i).moverAbajo();**

**}**

**}**

**}//end function**

**private void EliminarBase29(int rowFin){**

**if(rowFin==290){ //30**

**for(int i=0; i<listFormas.size(); i++){**

**//if(listFormas.get(i)==forma)continue;**

**listFormas.get(i).moverAbajo();**

**}**

**}**

**}**

**public void musica(){**

**musiavajo();**

**audiofondo3.play();**

**}**

**//end function**

**}//end class**