

**Corporación universitaria minuto de Dios**

**Facultad de ingeniería**

**Asignatura:**

Ingeniería de software  
NRC: 1221

**Docente**

**CARLOS ARMANDO LÓPEZ SOLANO**

**Presentado por:**

Daniel Gomez ID: 596017

Sebastián Mora Cedeño ID: 608267

Cristian Avendaño ID:489773

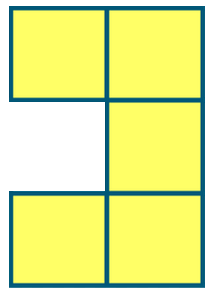
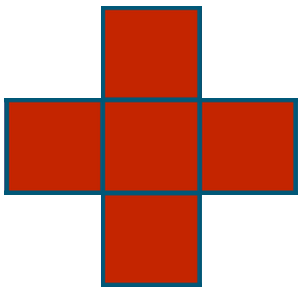
Índice

1. Vista lógica
2. Vista de desarrollo
3. Vista de procesos
4. Vista física

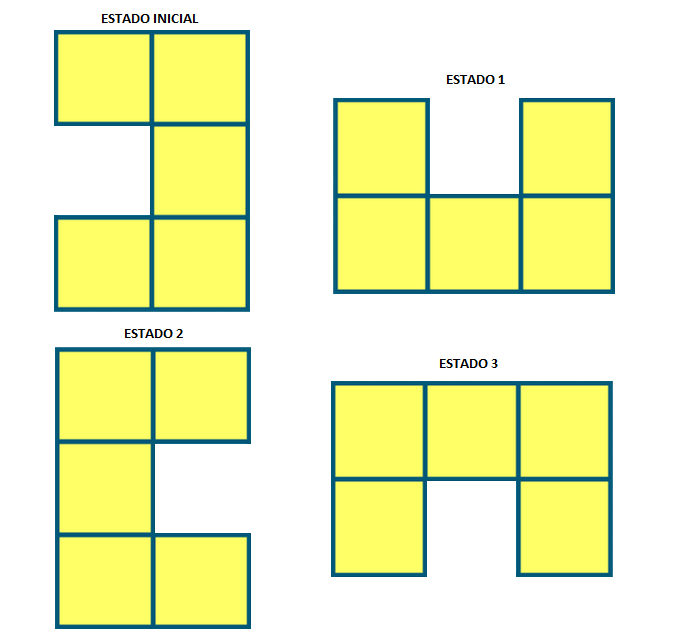
## Vista lógica.

Del proyecto se realizarán las figuras Cruz y C invertida, mediante matrices 3x3 con valores 1 y 0; donde 1 significa la posición de la figura y el 0 un espacio vacío (matriz utilizada para graficar la figura en el tablero) para saber posiciones dentro del mismo, así poder generar puntos y eliminar líneas completas.

Figura C inversa y Cruz.

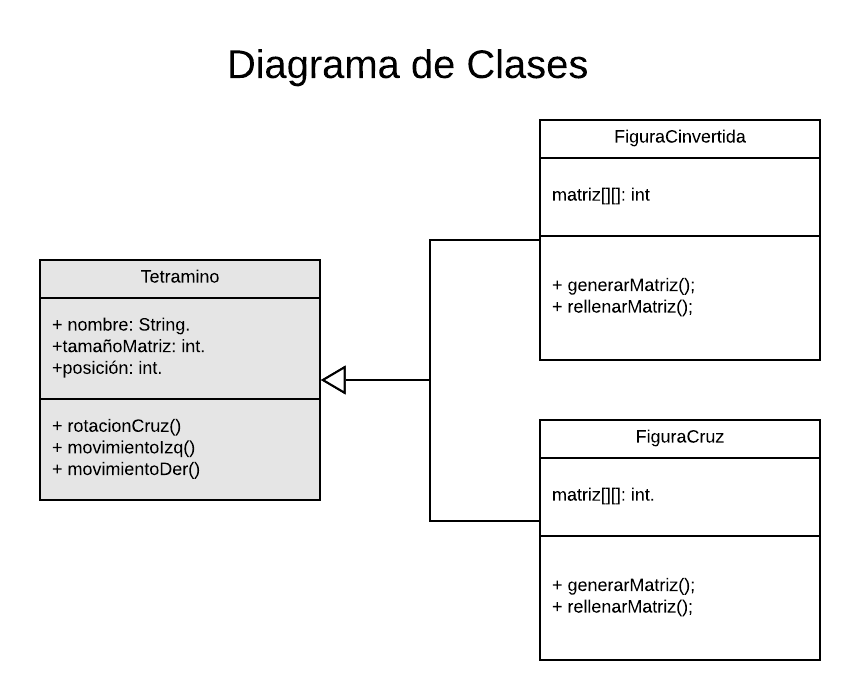
La figura C invertida será la única que tendrá diferentes estados, los cuales serán vistos como los giros posibles que realizará el usuario/jugador.



Para las figuras se utilizarán matrices 3x3, matriz generada en JavaScript y de tipo numérico.



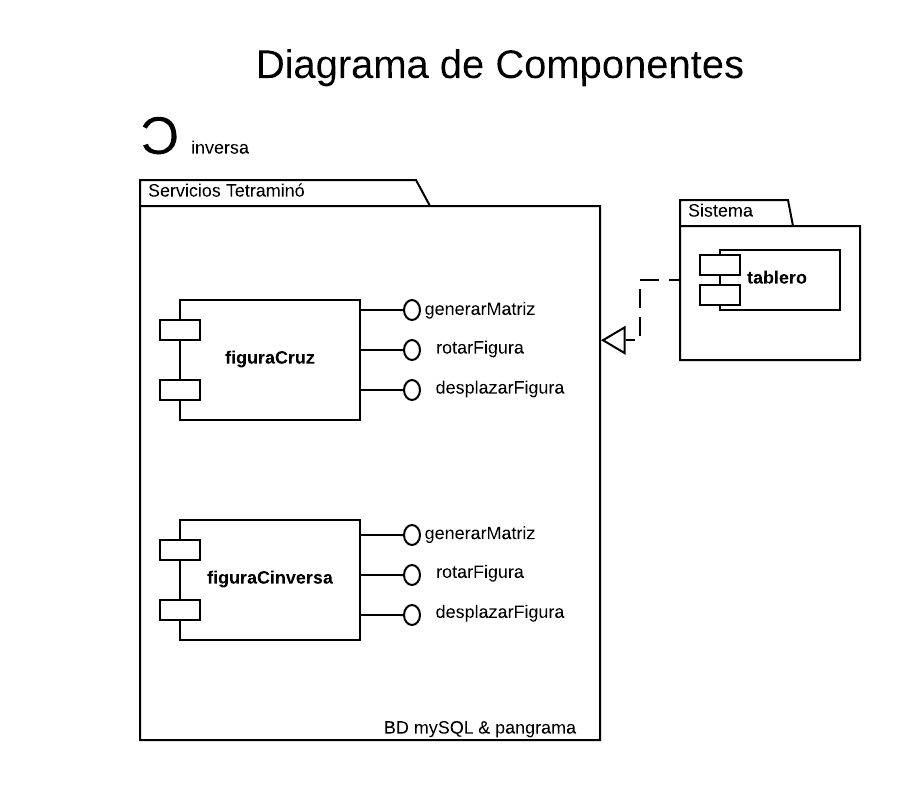
Al final del programa deberá juntar las figuras por medio de un servidor web con el tablero, el cual será una matriz de 22x10 en la que aparecerán aleatoriamente las figuras , además el usuario podrá rotarlas en cualquier dirección además de poder hacer movimientos horizontales “lado este” y “lado oeste “en el eje x , esto con el fin de que cuando una línea horizontal se completa, esa línea desaparece y todas las piezas que están por encima descenderán una posición, liberando espacio de juego y por tanto facilitando la tarea de situar nuevas piezas, y así sucesivamente hasta que  las piezas se amontonan hasta llegar a lo más alto de la matriz interfiriendo la creación de más piezas​ y finalizando el juego.

Diagrama de clases a utilizar para la creación de las figuras tetra minó que después serán requeridas por parte del tablero cada vez que utilice una de las dos figuras.

**2. Vista de desarrollo**

Para la creación del software que permitirá la implementación de las figuras de Tetris (Cruz y C invertida), al igual que sus respectivas rotaciones se ha utilizado el lenguaje de JavaScript. Se utilizan matrices de tipo binario, en las cuales los números uno equivaldrán a la posición de la figura y los ceros a espacios en blanco donde podrá rotar más adelante la figura permitiendo así la impresión por medio del método imprimirFigura, Por medio de key Action se registrará la tecla que utilizará el usuario para hacer la petición de giro ya sea tanto a la derecha como a la izquierda, este giro sólo será implementado en la figura c invertida, dado que la figura Cruz no necesita girar (todos sus lados son iguales). La figura C invertida consta de 4 estados, los cuales serán interpretados como las cuatro posibles posiciones que tendrá al momento de girar, los mismos serán creados a partir de la matriz binaria. Será utilizada la plataforma 000webhost para la implementación del servicio de web generando así la concatenación del servicio de figuras con el servicio de tablero montado en web por nuestros compañeros; el cual generará los request de figuras y dará inicio a la generación de la misma.

Diagrama de componentes a utilizar para los servicios que se ofrecerán al sistema del tablero, dichos servicios permitirán la creación, rotación y movimientos de las figuras en el tablero generado por terceros.



LISTA DE REQUERIMIENTOS DE LAS FIGURAS

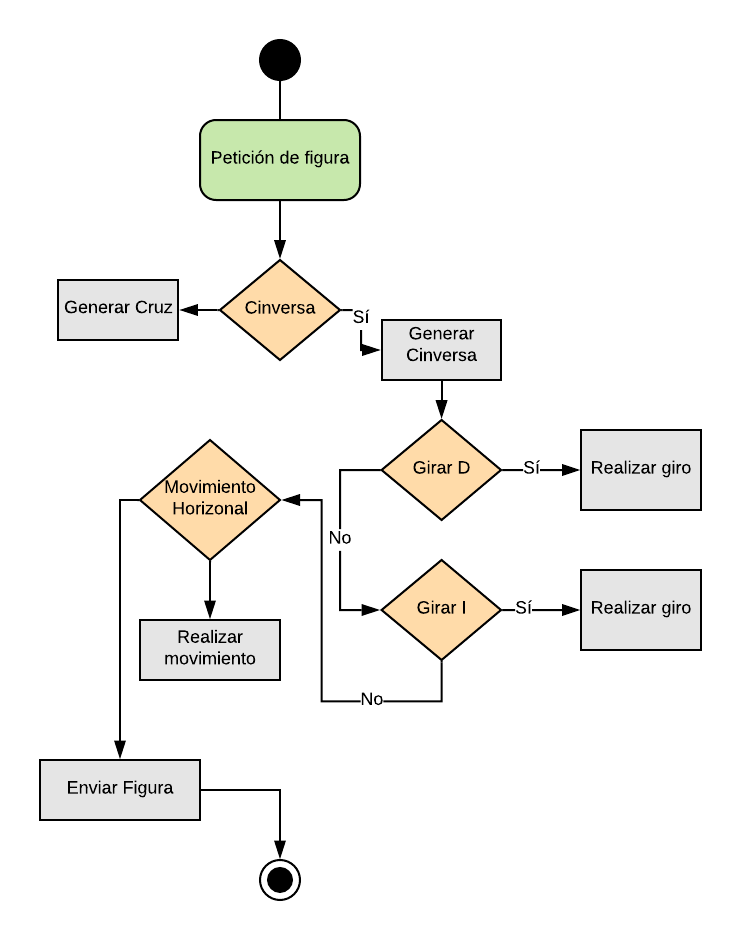
|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | **1** | **Estado** | **Solicitado** |
| Nombre: | Rotar figura | Prioridad | Alta |
| Descripción: | La figura generada por la máquina debe tener la posibilidad de rotación (girar sobre le mismo eje) de la forma que el usuario lo desee. | | |
| Entrada: | Mediante keycode se ubicará de acuerdo a las flechas hacia donde rota la figura. | | |
| Salida: | La rotación de la figura según la indicación del jugador. | | |
| Restricciones: | Luego de tocar el fondo la figura no podrá volver a moverse/girar. | | |
| Observaciones: | La figura no puede atravesar los bordes así sea girando. | | |
| Encargado: | Programador | Solicitante: | Cliente |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | **2** | **Estado:** | **Solicitado** |
| Nombre: | Generación de figura Cruz. | Prioridad | Alta |
| Descripción: | Se genera la figura Cruz, mediante de una matriz, la cual será enviada al tablero de juego. | | |
| Entrada: | Se generará una matriz con valores de 0 y 1, donde el 1 indica la forma que se le va a dar a la figura. | | |
| Salida: | Generación de la figura en la parte superior de tablero(posicion 22,5). | | |
| Restricciones: | Debe ser solicitada para generarse. | | |
| Observaciones: | La figura no necesita girar, solo descenso. | | |
| Encargado: | Programador | Solicitante: | Cliente |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID:** | **3** | **Estado** | **Solicitado** |
| Nombre: | Generación de Figura C invertida | Prioridad | Alta |
| Descripción: | Se genera la figura C invertida, mediante de una matriz 3x3, la cual será enviada al tablero de juego. | | |
| Entrada: | Se generará una matriz con valores de 0 y 1, donde el 1 indica la forma que se le va a dar a la figura. | | |
| Salida: | Generación de la figura C en la parte superior de tablero(posicion 22,5). | | |
| Restricciones: | Se genera a partir de la solicitud del tablero | | |
| Observaciones: | Generación aleatoria. | | |
| Encargado: | Programador | Solicitante: | Docente |

**3. Vista de procesos**

Al realizarse la petición por parte del tablero de una figura se procede a la verificación de cual de las dos posibles figuras se crearán, en el caso de la cruz se generará y permitirá realizar movimientos horizontales, de lo contrario en el caso de la Cinvertida se preguntará sobre giros sobre el mismo eje y movimientos horizontales, luego de esto se generará la figura y enviará al tablero de nuevo. Este proceso se realizará cada vez que se solicite una figura.



**4. Vista física**

Las figuras correspondientes a Cinvertida se realizarán en NetBeans y en programación JavaScript, se generará el API el cual podrá ser visualizado desde GitHub; además el Tetris contará con consumo de datos de un Web service, conectados mediante un servidor MySql (glassFish) y REST para su implementación total con todos los componentes de cada grupo.

