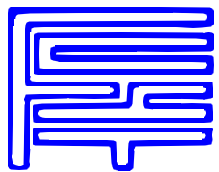
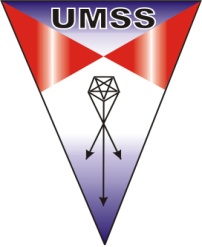
**UNIVERSIDAD MAYOR DE SAN SIMÓN**

**FACULTAD DE CIENCIAS Y TECNOLOGÍA**

**INGENIERÍA INFORMÁTICA**

Reporte Técnico

Proyecto Tetris

**Docente: Lic. Vladimir Costas**

**Estudiante: Maiko Alberto Morales Román**

**Materia: Programación Web**

**Cochabamba, 15 de Octubre de 2016**

# Problema

El problema dado es construir un juego arcade para la plataforma web. El juego elegido fue **Tetris**, las características a considerar son las siguientes:

* El juego debe funcionar correctamente
* Las piezas deben (según acciones del usuario) rotar, así como avanzar a la derecha, a la izquierda y hacia abajo.
* El juego debe terminar cuando no se puede crear una nueva pieza.
* El juego debe tener niveles de dificultad.
* El juego debe mostrar el score y una lista de los puntajes más altos.
* El juego debe almacenar en una base de datos la información necesaria relacionada con los scores.

# Solución

La solución planteada es la siguiente:

Se tienen bloques en un tablero, en el que se representaran las piezas que aparecen de forma aleatoria.

Las piezas no son más que arreglos de números que indican el color de un bloque, si el número es 0 el bloque está vacío, del uno al seis indica el ID de la pieza y también un color identificador.

Cada cierto tiempo se representa el estado del juego, al mismo tiempo se verifica si una pieza terminó de caer para generar una nueva, o si el juego ha terminado para pedir al usuario su nombre y registrar su score.

Cada vez que el juego termina el usuario ingresa su nombre y se almacena éste junto con su score en una base de datos.

Si una fila está completa se procede a eliminar la fila y bajar las que están por encima de la fila removida, en este momento se calcula el nuevo score y si se ha llegado a un score objetivo entonces se aumenta el nivel y por consiguiente la dificultad del juego haciendo que las piezas bajen rápidamente.

Se tiene una lista con las puntuaciones más altas y sus respectivos dueños, mostrando siempre los scores de mayor a menor.

# Tecnologías que se usan

Para lograr el objetivo de desarrollar un juego arcade se utilizó las siguientes tecnologías:

### Meteor

Meteor es una plataforma full-stack de código abierto para desarrollar aplicaciones web, móviles y de escritorio. Nos brinda una serie de tecnologías para construir aplicaciones reactivas y conectadas entre clientes, herramientas, y paquetes desde Node.js y la comunidad de Javascript.

* Meteor nos permite desarrollar en **un solo lenguaje**, Javascript, en **todos los entornos**, servidores de aplicación, buscadores web y dispositivos móviles.
* Meteor usa data **on the wire**, lo que significa que el servidor envía información, no HTML, y el cliente la representa.
* Meteor abraza el **ecosistema**, brindando lo mejor de la comunidad activa de Javascript.
* Meteor provee **reactividad completa**, permitiendo al UI reflejar sin ningún problema el estado real de la aplicación con el mínimo esfuerzo de desarrollo.

### MongoDB

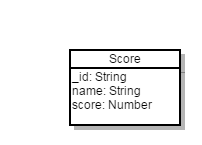
MongoDB es un DBMS **NoSQL** orientado a documentos de código abierto, que forma parte de la nueva familia de sistemas de bases de datos NoSQL. MongoDB guarda estructuras de datos en documentos similares a JSON con un esquema dinámico (MongoDB utiliza una especificación llamada BSON), haciendo que la integración de datos en ciertas aplicaciones sea más fácil y rápida.

### HTML5

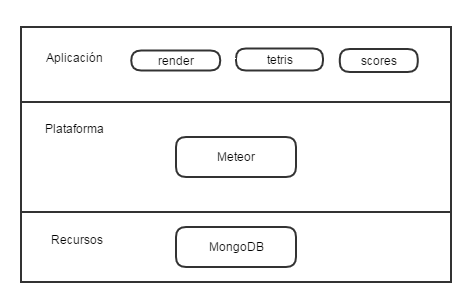
Para renderizar los bloques del juego arcade se utilizó el elemento canvas de HTML5.  
Este elemento se usa para dibujar gráficos, dinámicamente, usando Javascript, el elemento canvas es sólo un contenedor para gráficos, es necesario utilizar Javascript para realmente dibujar los gráficos utilizando varios métodos para dibujar figuras, establecer colores, texto, e imágenes.

# Diagrama de Base de Datos

Para la base de datos sólo se diseñó una entidad Score ya que la aplicación no necesita almacenar información adicional.



# Arquitectura

La arquitectura de la aplicación es muy simple:  
Arquitectura juego Tetris: