

**UNIVERSIDAD PERUANAS DE CIENCIAS APLICADAS**   
=====================================================  
**Algoritmos y Estructuras de Datos**  
=====================================================   
  
**Trabajo Final**

WS31

**Integrantes:**

* Andrés Choque Mejicano - u201712902
* Miguel Ángel Oré Quintana - u20171a420
* Sebastián Alejandro Montes Molina - u201712366

**2018**

ÍNDICE

Introducción ………………………………………………………………………………………………………………………………

Marco conceptual ……………………………………………………………………………………………………………………..

Requisitos ………………………………………………………………………………………………………………………………….

Diagrama de clases ……………………………………………………………………………………………………………………

Plan de trabajo ………………………………………………………………………………………………………………………….

Ejecución …………………………………………………………………………………………………………………………………..

Conclusiones………………………………………………………………………………………………………………………………

Referencias ………………………………………………………………………………………………………………………………..

1. Introducción:
2. **Problema:**

En los diferentes temas y códigos desarrollados a lo largo del curso, se hace uso continuamente de algoritmos y tipos de datos abstractos, motivo del curso en el ciclo. Sin embargo, la mejor forma de aceptar y consolidar estos conocimientos adquiridos en el curso es a través de la creación de un juego. Este representa un gran reto para los estudiantes, ya que supone la creación de una aplicación con grandes cantidades de datos en la que muchos aspectos de este se generan continuamente por fracciones a la vez que se guardan.

1. **Objetivos:**

Los principales objetivos de este trabajo son la creación de un videojuego con una temática escogida por nosotros donde el mundo o terreno del videojuego se genera continuamente. Además, en el juego existen beacons o faros que permiten al jugador transportarse a otros beacon. Estos faros guardan información sobre el terreno que les rodea con al menos 6 atributos que responden a la temática del juego.

1. Marco conceptual:

**Algoritmos:**

Es un conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite llevar a cabo una actividad mediante pasos sucesivos que no generen dudas a quien deba hacer dicha actividad. Dados un estado inicial y una entrada, siguiendo los pasos sucesivos se llega a un estado final y se obtiene una solución.

**SFML:**

Simple and Fast Multimedia Library (SFML) es una interfaz de programación de aplicaciones portable, escrita en C++, pero también disponible en C, Python, Ruby, OCaml y D. Su propósito principal es ofrecer una biblioteca alternativa a la biblioteca [SDL](https://es.wikipedia.org/wiki/SDL), usando un enfoque orientado a objetos.

Gracias a sus numerosos módulos, SFML puede ser usada como un sistema mínimo de ventanas para interactuar con OpenGL o como una biblioteca multimedia cuyas funcionalidades permiten al usuario crear videojuegos y programas interactivos.

Temática del juego:

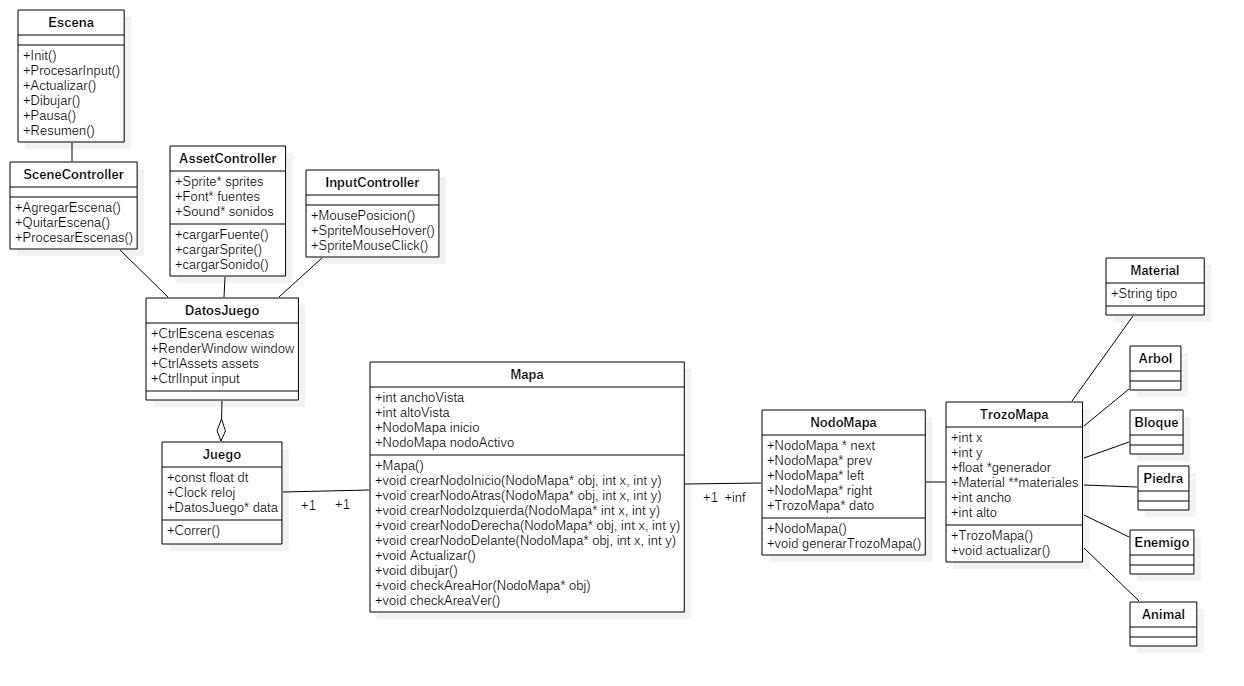
Nuestro juego tendrá una temática de supervivencia en la que el personaje deberá recorrer los distintos escenarios recolectando recursos para la fabricación de elementos que le permitan combatir los distintos personajes y elementos hostiles. Además, los beacon serán portales a diferentes mundos o terrenos que ofrecen nuevos recursos, paisajes y nuevos enemigos.

Nuestro trabajo está basado en diferentes juegos conocidos con similar temática como Minecraft o Moomoo.io, este último del que rescata gran parte de su esencia.

1. Requisitos:

* La temática general y el diseño del juego, será escogida por el equipo de trabajo.
* El juego consiste en un mundo único infinito generado procedural mente por porciones las cuales solo se generan en función a como el jugador va explorando.
* Una vez generada una región, ésta debe seguir existiendo por lo que resta del juego; de modo que, si el jugador regresa a una región previamente visitada, debe encontrarla igual a como la dejó.
* En el mundo existen beacons, los cuales permiten al jugador teletransportarse a cualquier otro beacon en el mundo.
* Cada beacon guarda información de por lo menos 6 atributos de la región donde se encuentra. Ejemplos: cantidad de madera, cantidad de hinchas de un equipo de fútbol.
* Los atributos son usados como criterio de búsqueda para que el usuario seleccione el beacon al que desea viajar. Para esto, el juego debe ofrecer una interfaz de búsqueda por criterios.
* Los atributos serán elegidos según el criterio del Equipo de trabajo y debe guardar relación con el diseño del juego. Es decir, si el atributo de un beacon indica que existen 100 humanoides en la región, se generará dicha región con los 100 humanoides correspondientes.
* Los beacons están uniformemente distribuidos en todo el mundo a intervalos regulares de modo que el jugador tenga que desplazarse una distancia razonable para poder alcanzarlos. Por ejemplo, beacons cada 1000 metros.
* Al empezar el juego, deben existir un millón de beacons sin embargo, la región en el mundo donde se encuentra, no necesariamente debe haber sido generada aún.
* Solo se generará la región del mundo correspondiente a un beacon, cuando el jugador se decide teletransportarse a dicho beacon.
* El jugador debe poder guardar su juego (en archivos) para continuar posteriormente.

1. Diagrama de clases (Entidades):



1. Plan de trabajo

Para el Hito N°1, se definirá la temática del juego, así como las diferentes herramientas que se usarán para la creación de la interfaz y demás elementos del juego. Además, se hará el diagrama de clases de las entidades presentes.

Para el Hito N°2, se añade a la presentación la ejecución del programa en la que se ve el diseño de interfaz de usuario, el diseño de interacción, los tipos de datos abstractos, los componentes y el diagrama de clases de los componentes. Además, se presenta el código con fuente con la implementación clases entidad y el prototipo de la aplicación.

Para el Hito N°3, se presenta el documento con el diseño de archivos, las conclusiones y referencias. Además, el código es entregado funcionando al 100% añadiendo la implementación de componentes y la implementación de persistencia. También se incluye un Power Point para la exposición.

1. Ejecución:
   1. Diseño de la interfaz del usuario

Pantalla de inicio:

La pantalla de inicio estará compuesta por 3 botones: Inicio, Cargar y Salir.

Inicio: Sera el botón de partida, con el cual el juego dará comienzo.

Cargar: Permite acceder a la partida guardada anterior.

Salir: Finalizará el ejecutable y se saldrá del juego.

En el código, esta pantalla inicial está dada como EscenaMenu.hpp.

Pantalla de pausa:

La pantalla de pausa contará con un botón de Continuar, Guardar y Salir.

Continuar: Se proseguirá con el juego.

Guardar: Guardará los datos de la partida actual.

Salir: Se saldrá del ejecutable y finalizará el juego.

En el código, esta pantalla se encuentra como EscenaPausa.hpp.

El juego consiste en un personaje que recorre distintos tipos de terrenos con diferentes recursos (ítems que se usan en el juego para la construcción de elementos) y que hará uso de estos para sobrevivir, enfrentándose a diferentes enemigos.

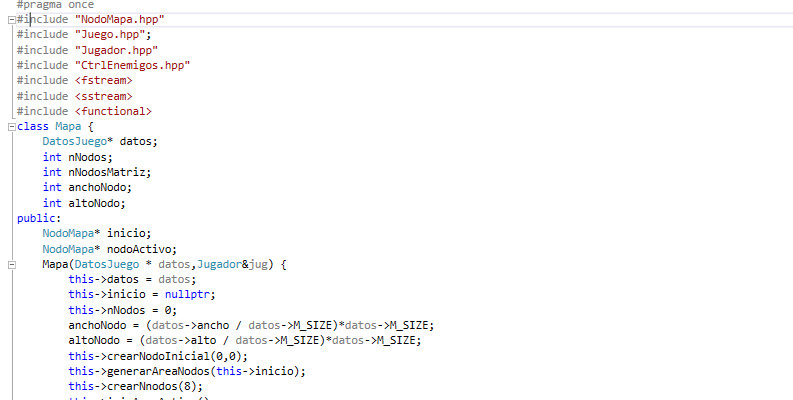
En Jugador.hpp, se describe el movimiento que puede hacer el personaje con las teclas A, S, D y W como direccionales.

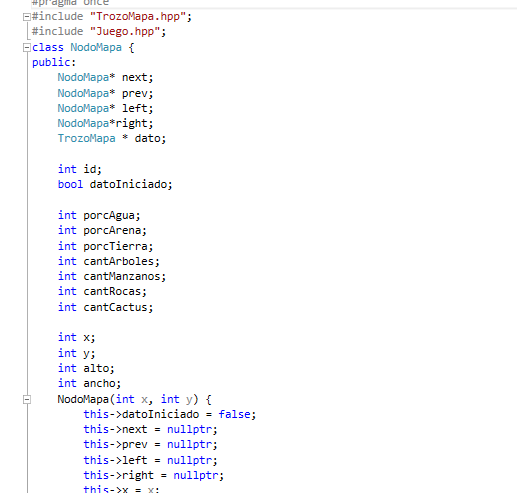
En el juego se incluyen los elementos Manzana, que aumentan la vida del personaje, y la Piedra, que sirve como munición y arma.

El programa utilizará los siguientes Datos Abstractos.

* Nodos

Se utilizaron los nodos para generar el terreno (Mapa.hpp), que está dividido en sectores los cuales se expresan en el código como nodos de tipo Next y Prev.



,

También se utilizaron en la creación de beacons (Portal), cuya función es la de transportar al personaje por atributos, estos beacons, al estar a distancias uniformes se expresan como nodos de tipo Prev y Next.



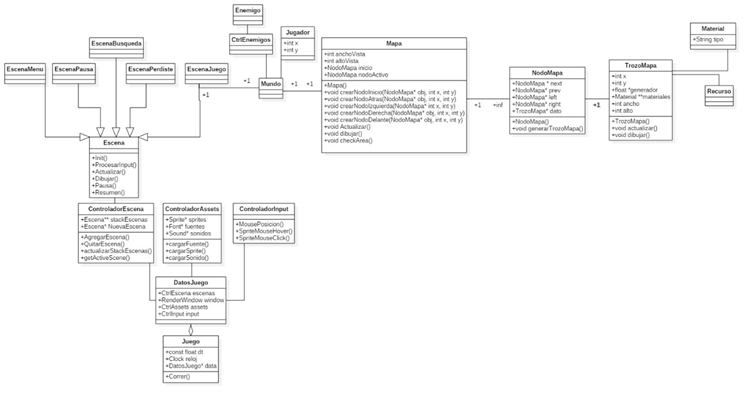
* Listas

Se utilizaron en la creación de los mapas. Se guarda en ellas los distintos atributos y el personaje.

También se utilizó para el guardado y cargado de datos.

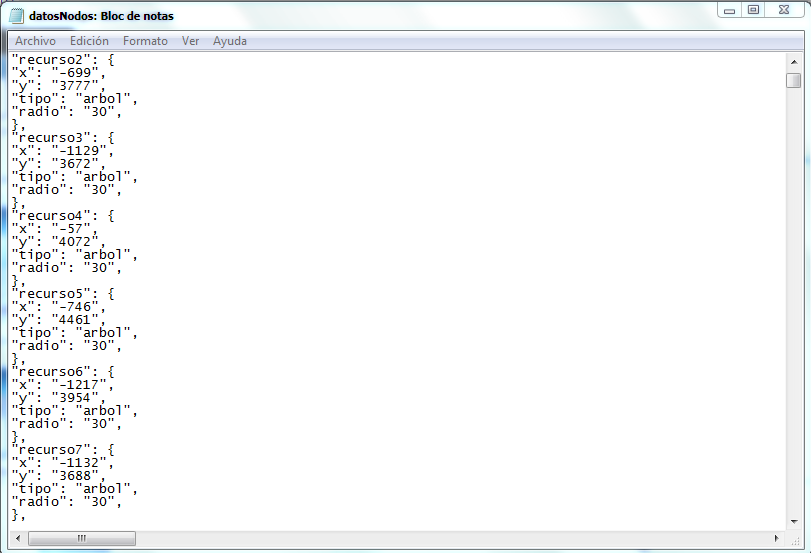


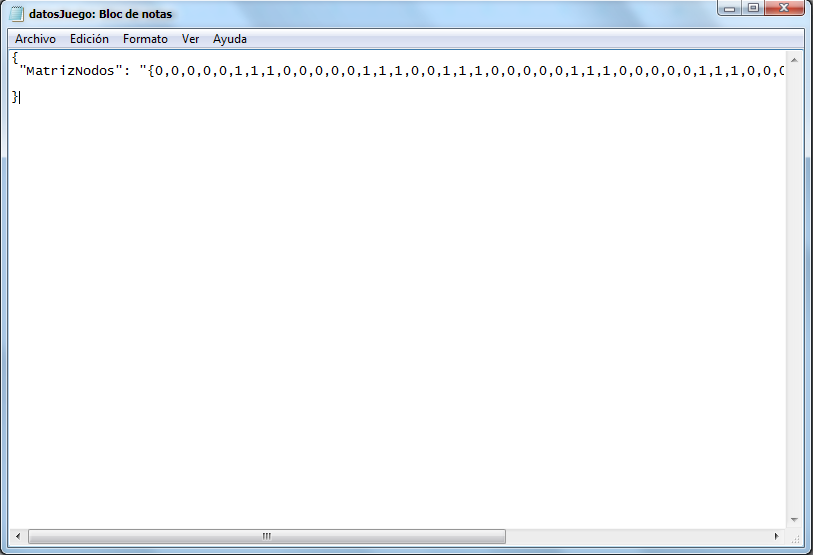
Diagrama de clases con sus respectivos componentes.



Diseño de Archivos.

1.-El guardado de cada objeto y terreno dentro del mundo donde cada uno tiene su posición y tipo de recurso o material.



2.-Guardado del Mundo donde cada nodo es un trozo de mapa determinado.

1. Conclusiones  
   En conclusión el trabajo realizado que es un juego estará construido mediante el uso de la programación orientada a objetos diseñados con diversos tipos de datos abstractos que son independiente de la lógica del juego y con una interfaz del tipo “Simple and Fast Multimedia Library” en sus siglas SFML.
2. Referencias:

<https://www.genbetadev.com/programacion-de-videojuegos/sfml-2-elaborando-un-juego-completo>

https://www.sfml-dev.org/index.php