GASPAR RODRIGUEZ VALERO

SKYSCRAPERS   Ver.: 1.0

Sistema de Decision



# Introducción

El sistema de decisiones es aquel que se encarga de gestionar y decidir que realizara en la siguiente actualización del bucle general del juego. Para hacemos uso de dos sistemas de decisión, una máquina de estados y behaviour trees. Nuestra IA dispone de varios estados y dependiendo de uno u otro gestiona un árbol diferente. Así creando un comportamiento complejo en situaciones concretas.

# Máquina de Estados



Los NPC a lo largo del juego pueden sufrir diferentes situaciones que deba abordar de formas diferentes y estas decisiones están recogidas en diferentes estados, estos son:

* Estándar: Es su estado inicial, es podría llamar estado de tranquilidad, las decisiones llevadas a cabo son de autoabastecimiento (comer, bebe y curarse), rutinas (Hablar, Vigilar y Patrullar) y acciones por eventos externos (buscar ruido o responder un aviso).
* Alerta: Cuando nuestro NPC percibe algo que le pueda alterarle cambiara a este, Alerta continúa gestionando al igual que estándar, rutinas, pero al encontrarse en un estado alterado las llevara a cabo de forma más minuciosa, es decir, se alteran sus sensores para ser más sensibles.
* Combate: Como es evidente se entra en el cuándo NPC se dispone a luchar contra el jugador, la única forma de alcanzar el estado es cuando la IA ve al jugador con sus sensores y se dispone de combatir, puede ser que sus parámetros alcancen un punto crítico por el cual decida pedir ayuda o en el peor de los casos se asuste.
* Asustado: El “objetivo” de un NPC cuando alcanza este es buscar ayuda y si no encuentra ayuda alguna huira del jugador buscando en zonas a compañeros suyos.

# Behaviour Tree

Son aquellos arboles de decisión que se ejecutan en cada estado y que dependiendo de este puede tomar unas decisiones u otras. A continuación, se mostrarán los arboles con un código de colores que diferencia los tipos de nodos.

* Azul – Nodos de Secuencia: El objetivo de estos nodos es ejecutar sus nodos hijo de forma secuencial y continuar hasta que uno de sus hijos le devuelva un false lo que implicar finalizar su ejecución, en caso contrario devolverá true lo que implicara que todos sus hijos fueron ejecutados y devolvieron estados positivos.
* Violeta – Nodos de Secuencia Positiva: Su funcionamiento es similar que el anterior, pero continuara hasta que uno de sus hijos devuelva un true o caso contrario un false lo que implicar que ningún hijo puedo ejecutarse con éxito.
* Rojo – Nodos de Condición: Estos nodos se encargan de verificar que se dan las condiciones para ser ejecutado lo que ayuda a verificar si debería llevar a cabo una decisión u otra.
* Verde – Nodos de Acción: Llevan a cabo las decisiones tomadas por el NPC, como moverse, curarse o cubrirse. Estos nodos tienen la capacidad de volver un estado diferente “RUNNING”, esta respuesta se lleva a cabo cuando la acción no ha sido terminada en esa iteración lo que marcara todos los nodos por encima suya para que la próxima vez que se ejecute el árbol vaya directamente a este nodo inconcluso.
* Blanco – Nodos rutina: Son nodos de acción cuya finalidad es gestionar las rutinas pre programadas y que se llevan a cabo en última instancia, es decir, cuando no tiene nada más importante que realizar.

## ESTANDAR

## 

## ALERTA



## COMBATE



## ASUSTADO



# Funcionamiento

En la versión actual requiere de pseudoblackboard[[1]](#footnote-1) que gestiona los datos no pertenecientes al NPC que se dispone a gestionar su sistema de decisión y una segunda clase que gestiona los datos propios del NPC, con dichas clases gestiona en esta versión el sistema de decisión. Esta información va recorriendo el árbol hasta alcanzar el nodo de acción o nodo de condición que hare uso de la información necesaria almacenada en las clases “Blackboard” y “datosNPC”.

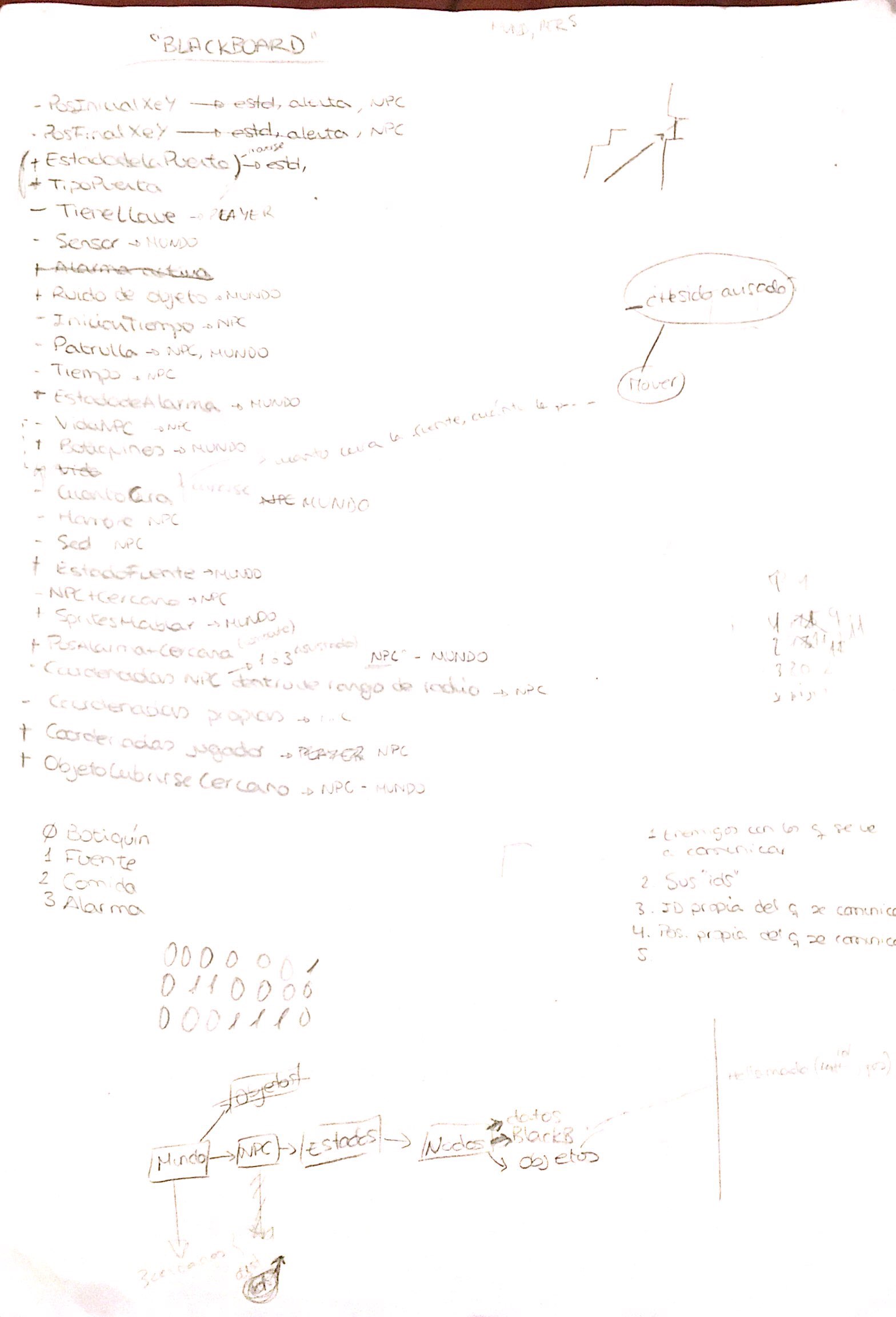
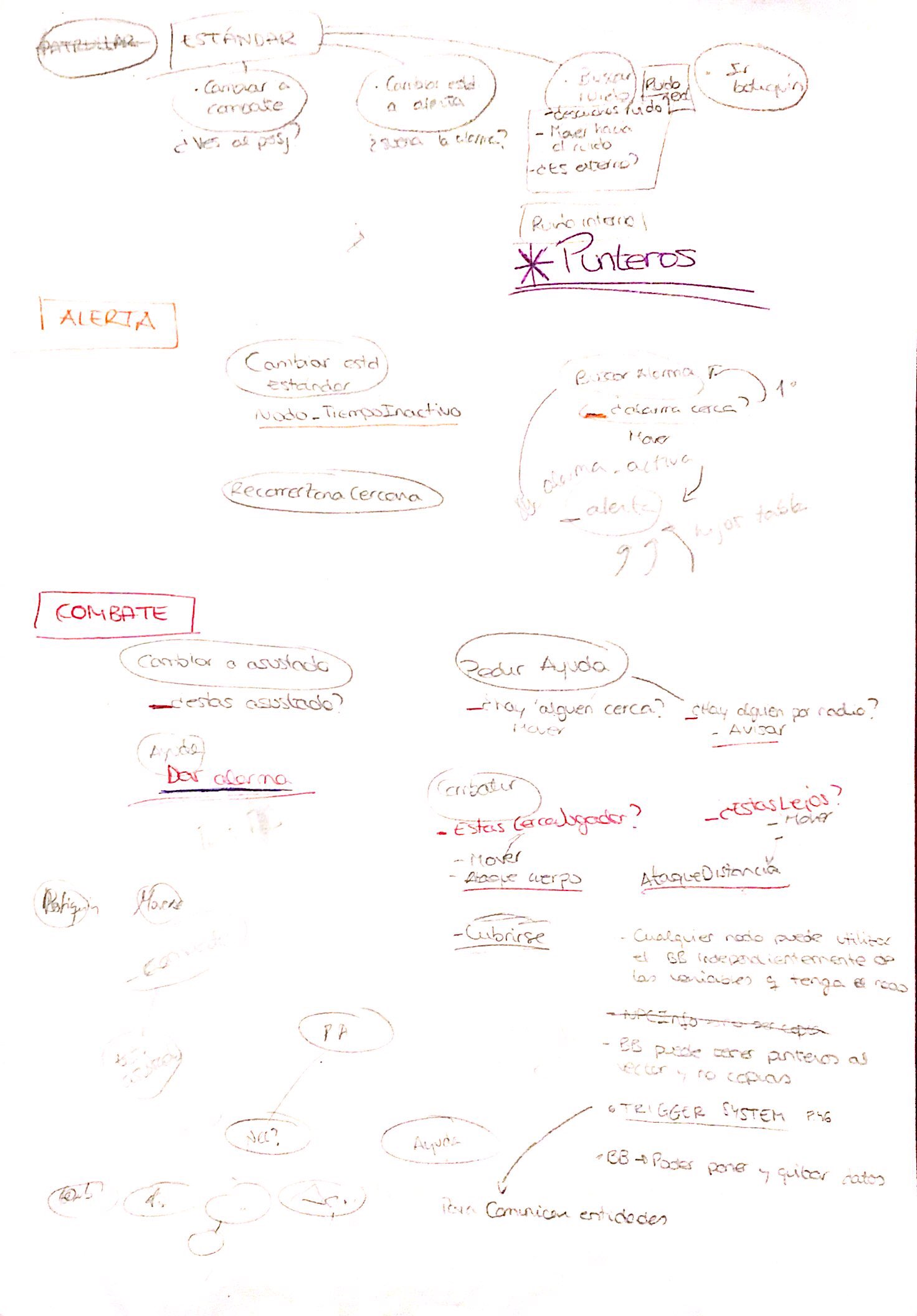
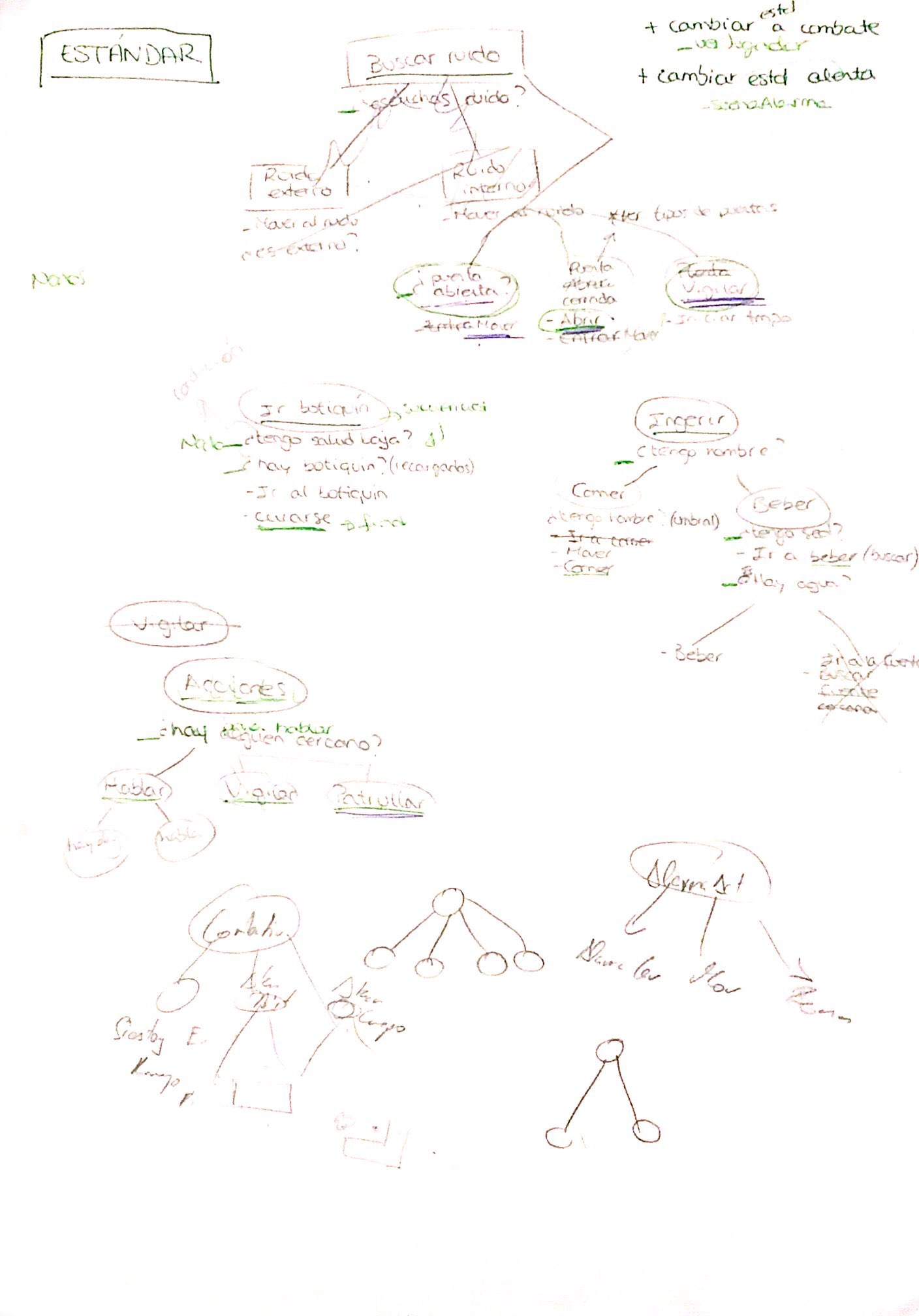
# Notas

## Versión 1.0

Primera versión funcional del sistema de decisión, muchas de las decisiones se hallan deshabilitadas a espera de la creación de un Trigger System y un Blackboard dinámico con su correspondiente arquitectura. En esta versión se han comprobado:

* Arboles de decisión probados (Orden correcto y gestión del estado running optimizado)
* Primera versión del nodo mover (nodo critico que gestiona mucha de las decisiones del sistema)
* Comprobada la máquina de estados para la gestión de árboles.
* Árbol estándar testeado en sistema 3D
  + Se cura, alimenta o bebe cuando lo requiere
  + Realiza su rutina cuando debe.

# Bocetos y Anotaciones



1. Pseudoblackboard: Hace referencia a un primer intento de implementación de la arquitectura blackboard, ya que la clase correspondiente coincide más con almacén de datos provisional. [↑](#footnote-ref-1)