***Instituto Tecnológico de Costa Rica***

***Sede Interuniversitaria de Alajuela***

***Lenguajes de Programación***

***Profesora: Samanta Ramijan Carmiol***

***I Semestre 2021***

***Integrantes:***

***-Joan Sánchez Chinchilla -> 2015123867***

***-Tomás Acuna->2018112856***

***-Marco Madrigal->2019053902***

***Tercer Proyecto Programado: Programación Lógica***

**b. Diseño de la solución:**

**b.1 Hechos**

Nuestro diseño de solución cuenta con 4 tipos de hechos diferentes, los cuales se explican a continuación.

-Personajes: Para los personajes se utilizará un identificador ,que nos brindará la posibilidad de escoger alguno de manera aleatoria por medio de una relación, el nombre correspondiente y tres accesorios y/o prendas de vestir que los caracteriza. Se vería de la siguiente manera:

personaje(Id,Nombre,Acc1,Acc2,Acc3). -> personaje(1,marco,pulsera,reloj,gorra).

-Pistas: Las pistas también utilizarán un identificador que cumple la misma función que en los hechos de personaje y el nombre correspondiente de la pista. Se vería de la siguiente manera:

pista(Id,NombrePista). -> pista(1,gorra).

-Objeto Robado: El hecho del objeto robado, al ser único en el universo del juego, simplemente contiene su nombre. Se vería de la siguiente manera:

objetoRobado(NombreObjeto). -> objetoRobado(corona).

-Dados: Los dados también contarán con un identificador, pero a diferencia de personaje y pista, este representa la cara del dado. Cada uno se asocia a lo que contiene esa cara, como por ejemplo una huella o un ojo (donde las huellas funcionan para obtener pistas y los ojos para revelar personajes). Se vería de la siguiente manera:

dado(Id,Cara). -> dado(1,huella). -> dado(2,pista).

-Puntos: Estos hechos que representarán cuantos puntos ha logrado reunir el usuario lanzando los dados. Son acumulativos y se dividen en puntos por huellas y ojos obtenidos. Cada cierto numero de puntos se le desbloqueará al usuario su respectiva recompensa como lo es obtener pistas o revelar personajes. También existen puntos para el culpable en caso de que los dados no sean del mismo tipo. Se vería de la siguiente manera:

puntosHuellas(n). -> puntosHuellas(5).

puntosOjos(n). -> puntosOjos(3).

puntosCulpable(n). ->puntosCulpable(2).

**b.2 Inferencias Lógicas**

Decidimos basar las inferencias lógicas en las pistas ya que la mayor parte del juego reside en ellas. Nuestro diseño de la solución cuenta con las siguientes:

-culpableRandom(): Esta inferencia se encargará de elegir un personaje de manera aleatoria para que sea el culpable de robar el objeto.

-quitarCulpable(): Para verificar que la inferencia anterior esté funcionado, quitarCulpable() funcionará como debug. Además, permitirá llevar un control para tener solamente un culpable por partida.

-decifradorPistas(Pista): Con esta inferencia podremos saber si una pista especifica (accesorio o prenda de vestir), incrimina al culpable.

-pistaRandom(): Esta inferencia nos brindará una pista aleatoria entre las 12 existentes.

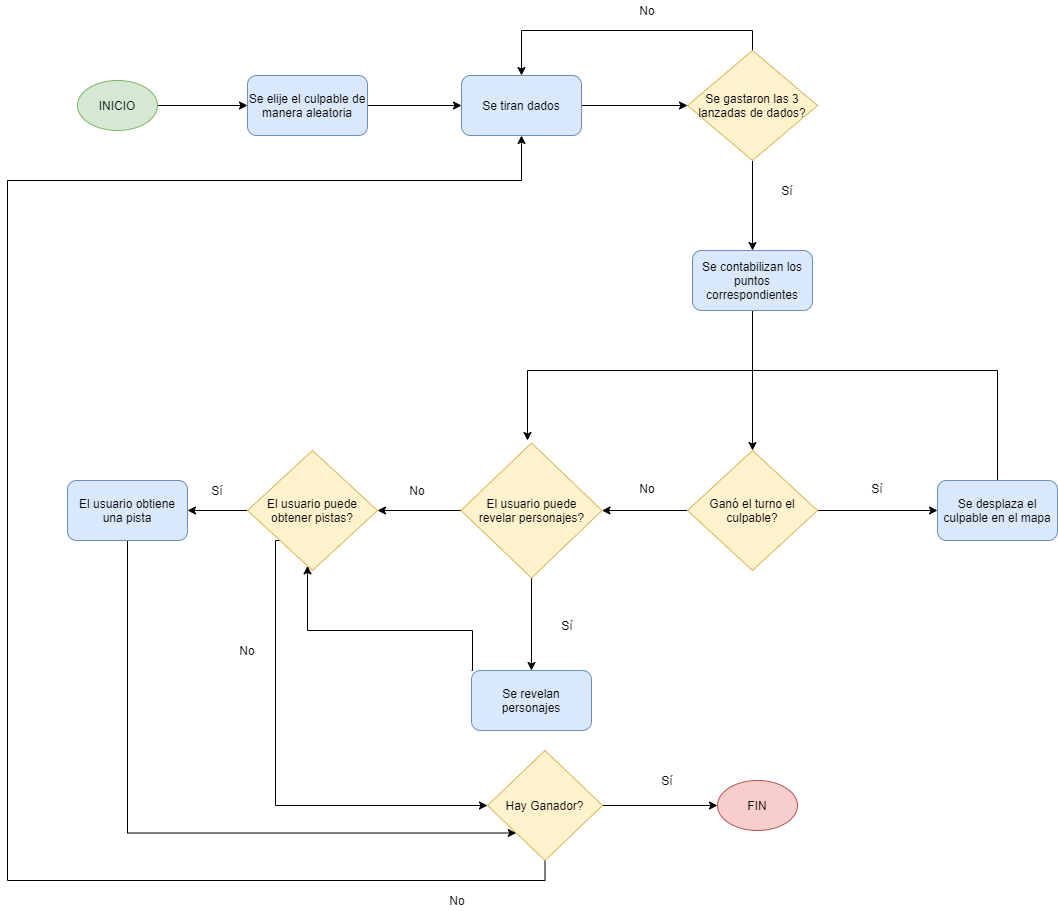
-tirarDado(): La idea de esta inferencia es poder obtener la cara de un dado luego de simular su lanzamiento, el cual puede ser producto de obtener un numero aleatorio.

-descartarPersonaje(Nombre): Con esta inferencia podremos ir descartando personajes. Además, de descartarse el culpable podremos saber si el usuario pierde la partida.

-existeGanador(): Será muy útil para saber si el culpable logró escapar o si el usuario logró atraparlo.

-sumarPuntosHuellas(),sumarPuntosOjos y sumarPuntosCulpable(): Con estas inferencias podremos llevar un conteo de los puntos actuales en cada ámbito.

**b.3 Flujo del programa**



**c. Diseño de la interfaz gráfica:**

-

**d. Lecciones aprendidas:**

-