CONTENIDO

DIAGRAMAS DE CLASES – THE LAST OF US

ENUNCIADO

Tommy Miller, hermano de Joel en The Last Of Us, necesita crear un sistema informático para llevar un control sobre los habitantes de Jackson, asentamiento creado para luchar contra los *raiders* e infectados. Para ello ha decidido que debe confiar en nosotros y debemos modelar un diagrama UML para planificar el sistema que vamos a desarrollarles.

Jackson Town

Para ello debemos crear un proyecto UML llamado Jackson en el que se diseñe un diagrama de clases que modele el proceso de dar de alta a cada una de las personas que se apuntan al asentamiento Jackson.

De cada persona interesa saber sus datos básicos: NIF, nombre completo y fecha de nacimiento. Cuando cada nuevo superviviente se da de alta, se le asigna un código de asociado alfanumérico y se anota la fecha de alta.

La clase Fecha se modela con tres campos (día, mes y año) de tipo entero. La clase Nif se modela con un campo de tipo entero llamado dni y un campo de tipo carácter llamado letra.

ANALISIS DEL ENUNCIADO.

El primer paso a realizar consiste en leer detenidamente el enunciado y extraer de él toda la información posible. A veces es cuestión de aplicar el sentido común, a veces es cuestión de unir piezas, a veces es cuestión de lógica y a veces es cuestión de pura deducción, pero siempre es cuestión de razonar por aproximaciones sucesivas y de experiencia.

Bien, parece que el enunciado refiere únicamente un modelado de datos, no de comportamiento, por lo que se procederá a realizar una lista de los elementos más significativos para el proyecto que se puedan extraer del enunciado.

1. Nombre del proyecto – Jackson
2. Nombre del diagrama – AltaJackson
3. Ítems – Elementos significativos del enunciado.
    - Persona
    - Superviviente
    - Nif
    - Nombre completo
    - Fecha de nacimiento
    - Código de asociado
    - Día
    - Mes
    - Año
    - Dni
    - Letra
4. Tipos de datos
    - Integer
    - Char
    - String
    - Nif
    - Fecha
    - Nombre

DISEÑO DE CLASES

Recuérdese que las clases son entidades que encapsulan información, se trata por tanto de ver qué información de la lista anterior está relacionada entre sí y ver la forma de encapsularla en sus respectivas clases.

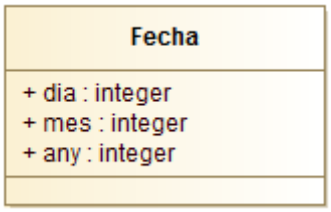
Se procederá a identificar las clases a partir del enunciado y de encapsular en ellas la información relacionada. Este paso se realizará considerando las clases de forma aislada las unas de las otras. Posteriormente, cuando se vean las relaciones, se depurará su composición.

En esta fase del modelado se procede siempre desde las clases más triviales a las más complejas.

CLASE NIF

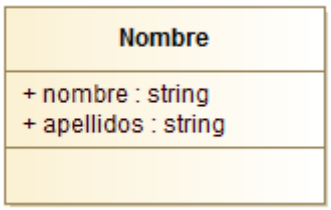
Nif

CLASE FECHA



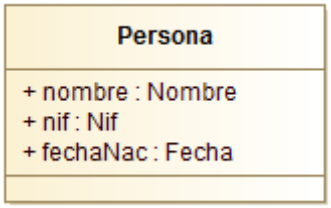
Fecha

CLASE NOMBRE



Nombre

CLASE PERSONA



Persona

CLASE SUPERVIVIENTE

Superviviente, en las sucesivas imágenes superviviente será socio

RELACIONES

En esta fase se va a evaluar qué clases tienen que ver con qué otras, es decir sus relaciones. Para que el procedimiento resulte lo más sencillo posible se iniciará el estudio por las relaciones dos a dos.

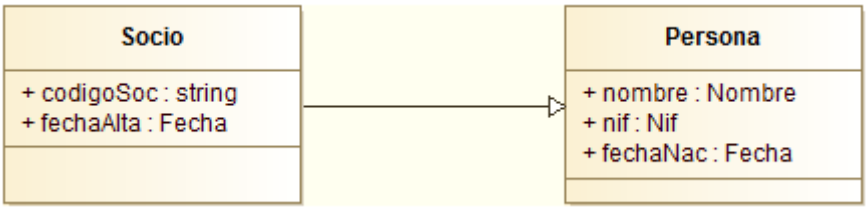
HERENCIA

Primero se abordan las relaciones de herencia empezando por aquellas que resulten triviales o más evidentes.

Aunque estrictamente hablando no es así del todo, la regla para detectarlas es ver si entre las clases definidas en el diseño existe alguna cuyos atributos sean un subconjunto de alguna otra.

PERSONA – SUPERVIVIENTE

En este caso resulta que los atributos de la clase Persona son un subconjunto de los de la clase Superviviente y semánticamente tiene sentido que la clase Socio sea una especialización de la clase Persona.



Relación Superviviente Persona

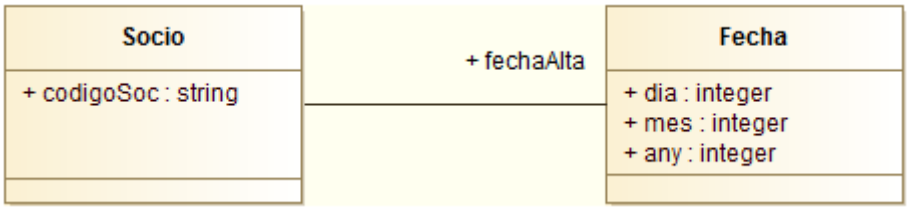
Obsérvese que los atributos que hereda la clase especializada no se representan. Obsérvese también que la flecha que representa esta relación va desde la clase hija a la clase madre, tiene línea continua, punta de flecha cerrada, no tiene cardinalidad y no está etiquetada por ningún rol.

ASOCIACIÓN

Una vez se han resuelto las relaciones de herencia le toca el turno a los demás tipos de relaciones que son asociaciones. Se procederá siempre abordando primero las triviales o más simples y continuando por las demás. Para que resulte más claro, el análisis se realizará considerando primero las relaciones dos a dos.

SUPERVIVIENTE – FECHA

La clase Superviviente tiene un campo de tipo Fecha, dicho de otra manera, la clase Superviviente tiene una referencia a un objeto de la clase Fecha. Así considerado este campo pasa a ser el rol de la relación que vincula a ambas clases. Por lo tanto, desaparece de la clase Superviviente y aparece en la línea de vinculación junto a la clase de su tipo.



Relación Superviviente Fecha

El siguiente paso es abordar las cardinalidades o multiplicidades, es decir el número de instancias de cada clase que intervienen en la relación. Para resolver este paso hay que preguntar: «¿Por cada instancia de una de las dos clases cuantas instancias de la otra clase pueden en extremo intervenir como mínimo (Cardinalidad mínima) y como máximo (Cardinalidad máxima)?». Y luego hacer las preguntas al revés.

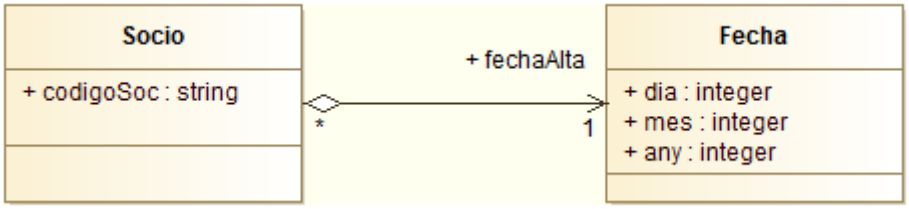
- Cuántas fechas de alta como mínimo tiene cada superviviente: 1
- Cuántas fechas de alta como máximo tiene cada superviviente: 1
- Cuántos supervivientes se dan de alta como mínimo en una fecha: 0
- Cuántos supervivientes se dan de alta como máximo en una fecha: Varios

AGREGACIÓN – COMPOSICIÓN

El siguiente paso consiste en considerar qué clase es PARTE y qué clase es TODO. Dicho de otro modo quién contiene a quién. En este caso la discriminación es trivial: la clase Superviviente es la parte TODO porque tiene una referencia a la clase Fecha que es la parte PARTE.

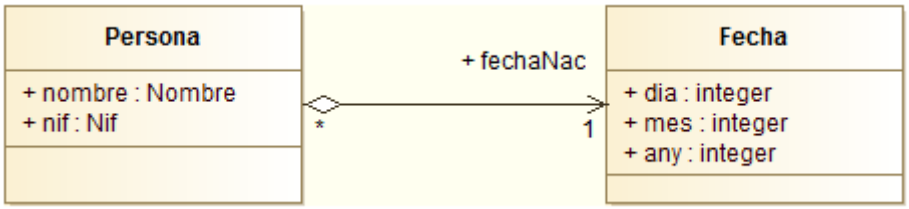
RELACIONES DE AGREGACIÓN

Ahora debemos determinar si la relación entre las clases es de agregación o de composición. Para que la relación sea de composición es condición necesaria que la cardinalidad de la parte TODO sea 1. Como este no es el caso la relación es de agregación.



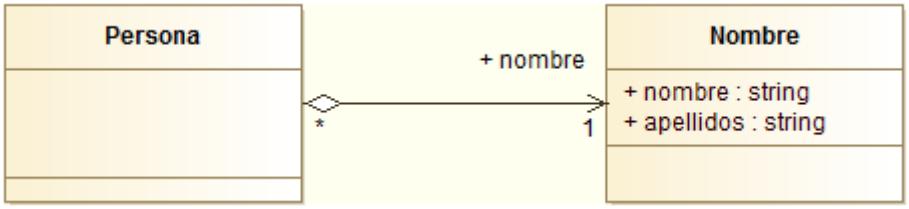
Relación Superviviente Fecha Agregación

Este mismo razonamiento lo vamos a usar con las clases Fecha y Persona. Esta vez el rol de la clase Fecha en la relación cambia. Obsérvese como ha desaparecido el campo correspondiente a la fecha de nacimiento de la clase Persona.



Relación Persona Fecha Agregación

También ocurre algo muy similar con Persona y Nombre.

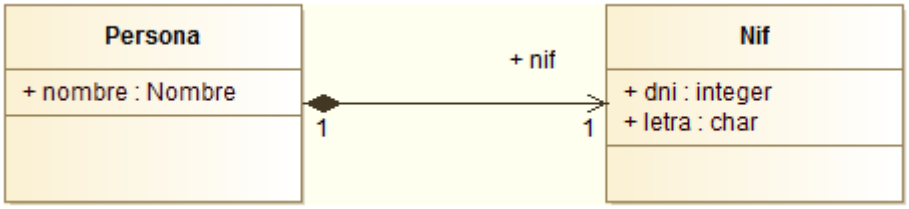


Relación Persona Nombre Agregación

RELACIONES DE COMPOSICIÓN

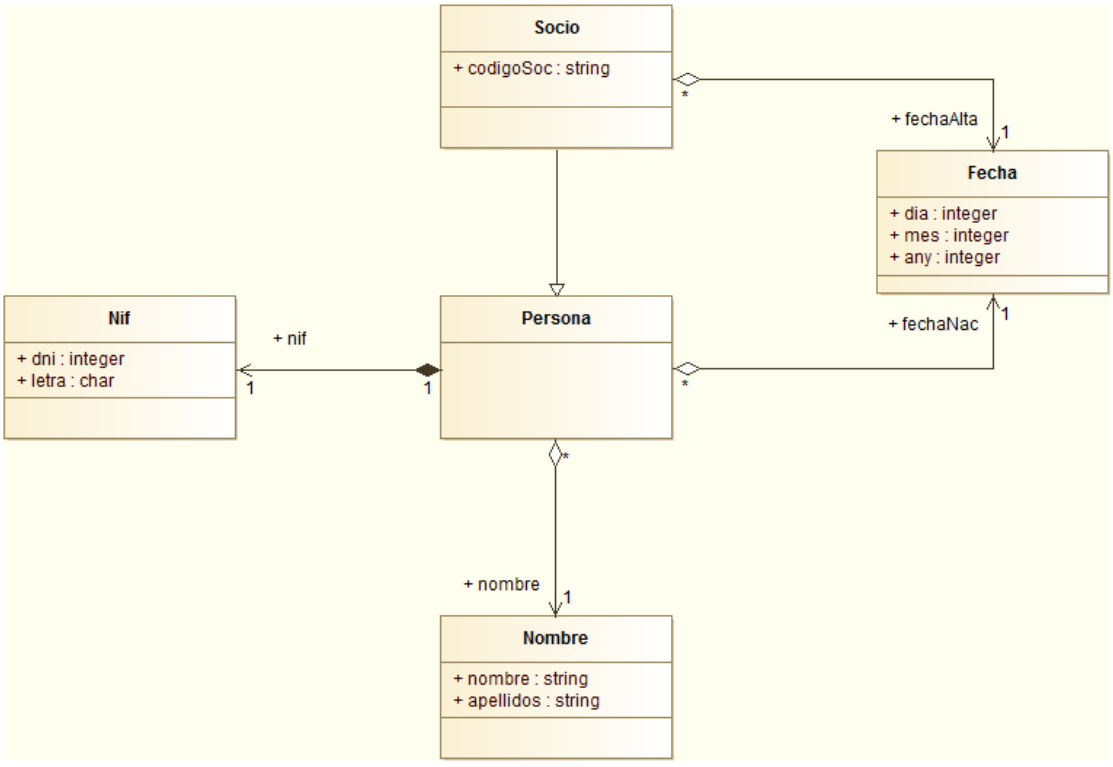
El análisis de la relación entre estas dos clases determina que cada objeto de la clase Nif está unívocamente unido a un solo objeto de la clase Persona, y viceversa, por lo que la cardinalidad en ambos lados es la unidad, tanto mínima como máxima.

Además, semánticamente hablando, si desaparece la parte TODO (el objeto de la clase Persona), la existencia de la parte PARTE (el objeto de la clase Nif), carece de sentido y debería desaparecer también. Esta dependencia existencial apunta a una relación de tipo Composición.



Relación Persona Nif Composición

SOLUCIÓN FINAL THE LAST OF US.



Solución final

DIAGRAMA DE CLASES FLIPPER ZERO

INTRODUCIÓN.

Una nueva herramienta tecnologica se ha hecho popular en TikTok. Se llama Flipper Zero y es el dispositivo de moda entre los aficionados al hacking. Este pequeño dispositivo es una especie de navaja suiza que lleva ya algún tiempo disponible pero que se ha convertido en viral gracias a los vídeos de quienes lo promocionan en TikTok. Muchos lo califican como el 'tamagotchi para hackers', y sus prestaciones son realmente llamativas...

El dispositivo es capaz de leer, copiar y emular etiquetas RFID y NFC, mandos a distancia o claves digitales a distancia gracias al soporte de señales infrarrojas. Es totalmente autónomo y no necesita ser conectado a un ordenador o un smartphone para utilizarlo. En los últimos meses han comenzado a aparecer diversos vídeos en los que se ve a usuarios hackeando los precios en los letreros de las gasolineras, abriendo garajes o sustituyendo a los distintos mandos a distancia de casa. Allí se muestran también vídeos en los que este dispositivo desbloquea un teléfono, abre un BMW, pagar por máquinas recreativas y muchas otras cosas.

En este caso, la empresa de hoteles NH esta preocupada por si estos hackers usan esta herramienta para abrir puertas de hoteles a su voluntad, por lo que han decidido crear un nuevo sistema informático en el que una red se encarga de abrir las puertas del hotel y se hace todo por internet mediante una aplicación.

El soporte RFID hace posible que Flipper Zero pueda leer, guardar, emular e incluso romper la seguridad de sistemas como llaves de puertas mediante fuerza bruta.

ENUNCIADO DEL PROBLEMA.

Se procede a planificar la red completa del nuevo sistema de puertas de los hoteles NH. Los elementos que se pueden incluir en la red son:

- Servidor
- PC
- Cerradura.
- Hub.
- Cable de red.

Los PCs pueden conectarse con un único Hub, los servidores con uno o varios. Los Servidores y PCs pueden generar mensajes, con una cierta longitud. Los Hubs tienen un número de puertos, algunos de los cuales puede usarse para conectar con otros Hubs. Tienen cierta probabilidad de "perder" mensajes. Las cerraduras pueden averiarse, con cierta probabilidad, durante cierto tiempo.

SOLUCIÓN FINAL FLIPPER ZERO

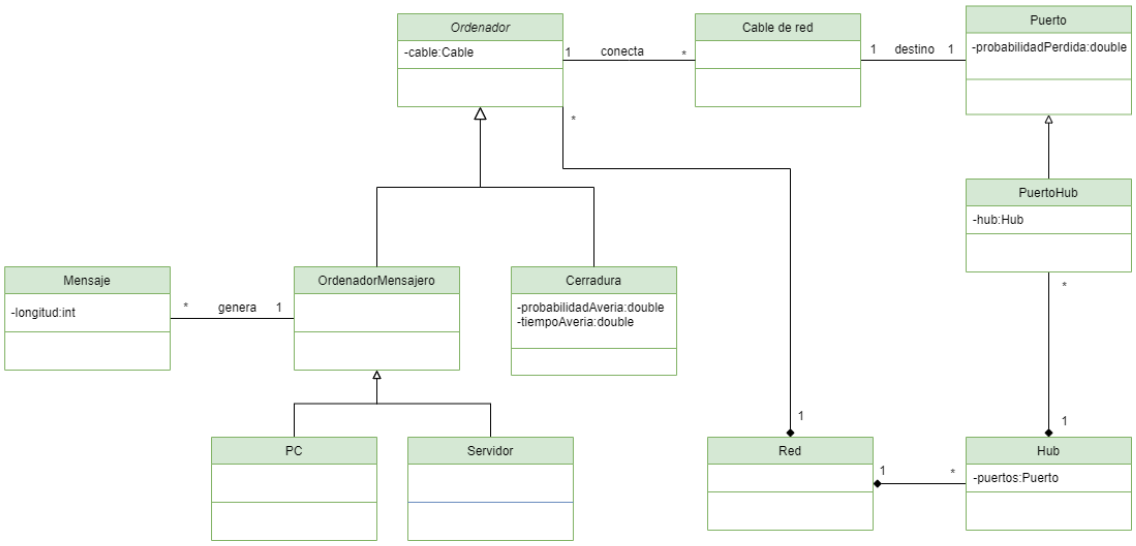


DIAGRAMA DE CLASES PATO CORP

INTRODUCCIÓN

La comunidad patuna esta disgustada por los cambios hechos en Twitter desde la compra de un señor al cual no quieren nombrar, y ha decidido que van a crear una nueva empresa para crear su propio twitter, Patter, donde en vez de twets tenemos pats.

Logo de Patter

Para ello nos piden que diseñemos su futuro sistema informático empresarial.

DATOS DEL EJERCICIO.

Se debe crear un diagrama de clases que cumpla con:

- Una PatoApp que almacene información sobre bandadas de patos empresariales, patos trabajadores y patos clientes. Estos dos últimos se caracterizan por su nombre y raza.
- Los pato empleados tienen sueldo bruto en semillas (les flipa), los patos directivos tienen una categoría y un conjunto de patos subordinados.
- De los clientes patunos se necesita saber su cuakphone para contactar con ellos.
- Esta app a desarrollar necesita mostrar los datos de los pato empleados y sus clientes.

SOLUCIÓN FINAL PATTER.

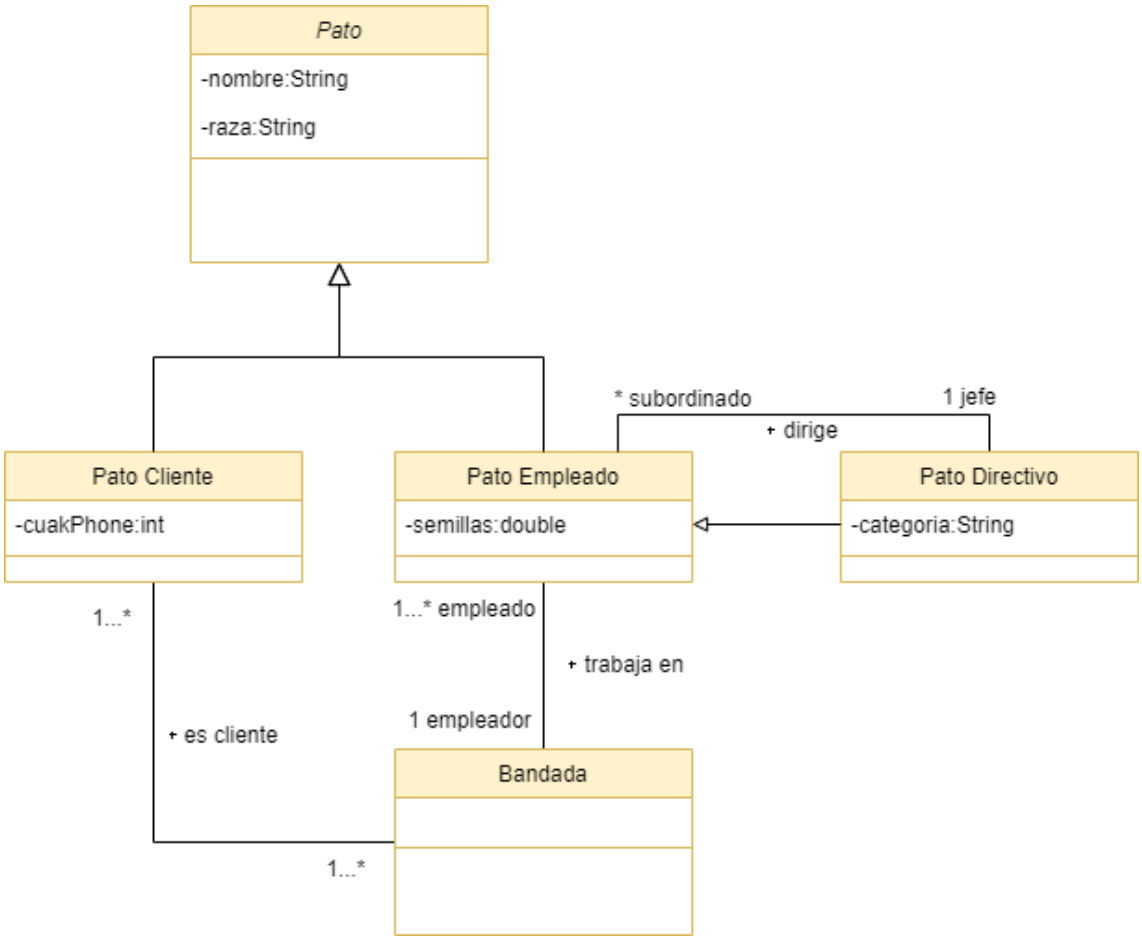
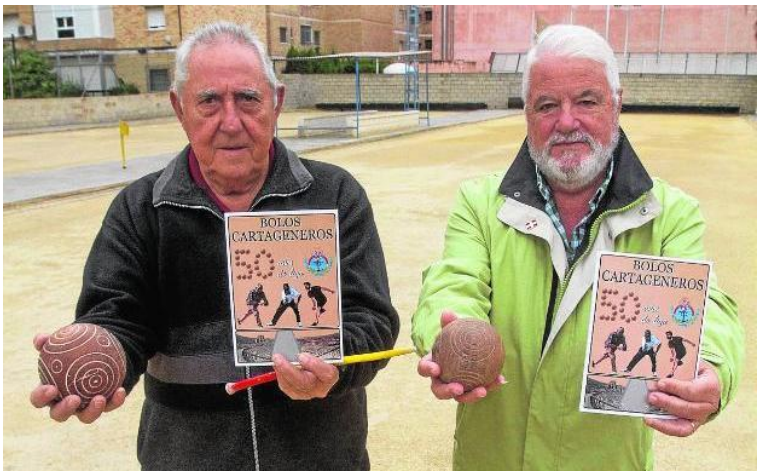


DIAGRAMA DE CLASES BOLO CARTAGENERO.

INTRODUCCIÓN

Los bolos cartageneros es una de las dos modalidades de juegos de bolos que se practica en la Región de Murcia. Podemos decir que es una modalidad del juego de bolo de palma, que es jugado con bola de palma y donde se utilizan bolos irregulares.



Se busca rescatar el bolo cartagenero y enseñárselo a las nuevas generaciones, las cuales se pasan todo el día con las maquinitas y la liga de las leyendas, trayendo los valores de amistad y armonía del bolo cartagenero en contraposición con los de odio y discriminación de la liga de las leyendas.

ENUNCIADO

Se debe crear un proyecto UML llamado Torneo en el que se diseñe un diagrama de clases que modele la estructura necesaria para manejar los datos de los encuentros de un torneo de bolo cartagenero en la modalidad de sorteo y eliminatoria.

Del torneo interesa conocer la fecha del torneo, los encuentros celebrados y el ganador. De cada jugador, que debe de conocer perfectamente las reglas, interesa saber el número de federado de la federación de la que es miembro, y su refrán favorito de victoria.

De cada persona interesa saber sus datos básicos: NIF, nombre completo y en que cretácico nació. La clase Cretácico se modela con tres campos (día, mes y año) de tipo entero. La clase Nif se modela con un campo de tipo entero llamado dni y un campo de tipo carácter llamado letra.

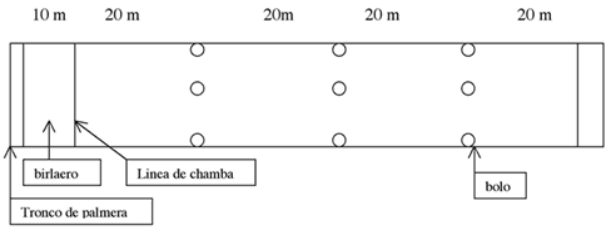
De cada encuentro interesa conocer los oponentes, el ganador, el boliche donde se juega, y el resultado final del marcador de cada una de las dos partidas que se juegan a 6 puntos.

DATOS CURIOSOS

Elementos del terreno de juego:

- La chamba: se trata de una línea recta que señala el lugar del terreno de juego, que deben rebasar todas las bolas que se lanzan.

- El birlaero: se encuentra situado detrás de la chamba. Nunca se podrá hacer encima o antes de la misma. Consiste en una circunferencia de 8 a 12 cm. de diámetro.



- Los bolos: nueve bolos de madera que no sobrepasan los 35 cm. de alto, de base y punta afilada. Se colocan formando tres filas rectas.

- Las bolas: son esféricas, de madera de jinjolero dura y con poco peso. Su diámetro no puede superar los 115 cm. Cada equipo dispondrá de 7 bolas.

- El mande: el lugar desde donde todos los jugadores efectuarán sus lanzamientos. Deberá ser una circunferencia de un mínimo de ocho centímetros.

- El lanzamiento de la bola puede hacerse "a yema" (a la derecha y sin efecto), "a margarita" (imprimiendo efecto con el dedo meñique) o "a gordo" (si el tiro se produce con el dedo pulgar).

-Al capitán se le llama manilla.

SOLUCIÓN FINAL BOLO CARTAGENERO.

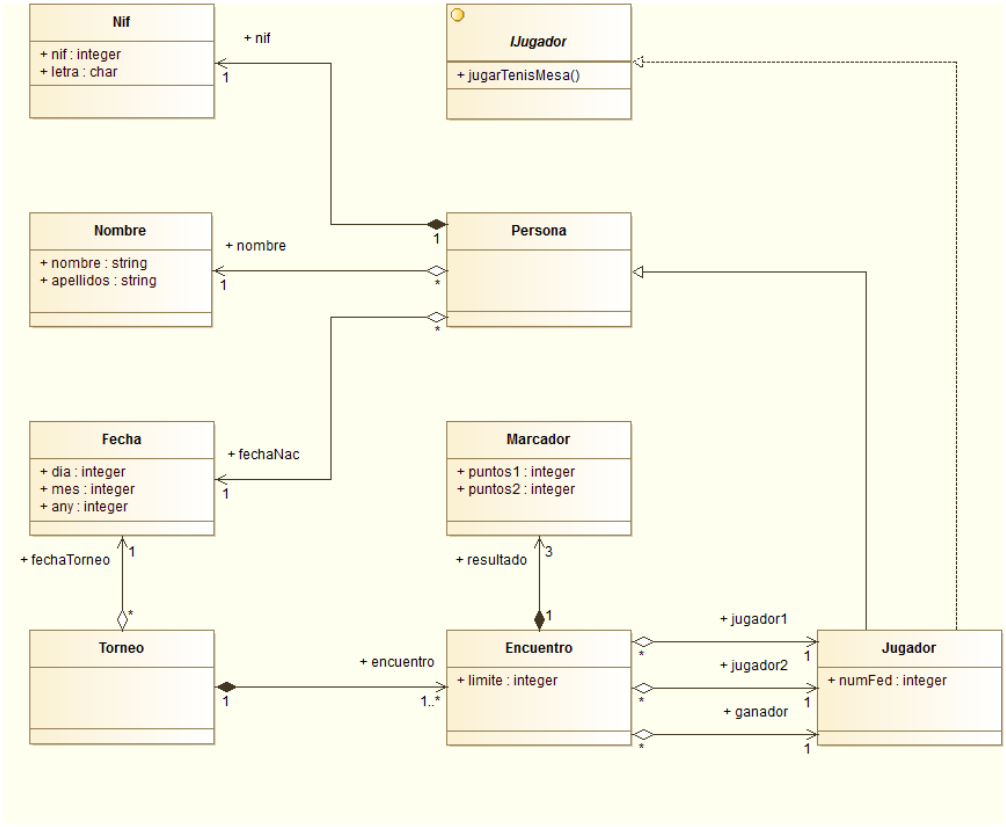
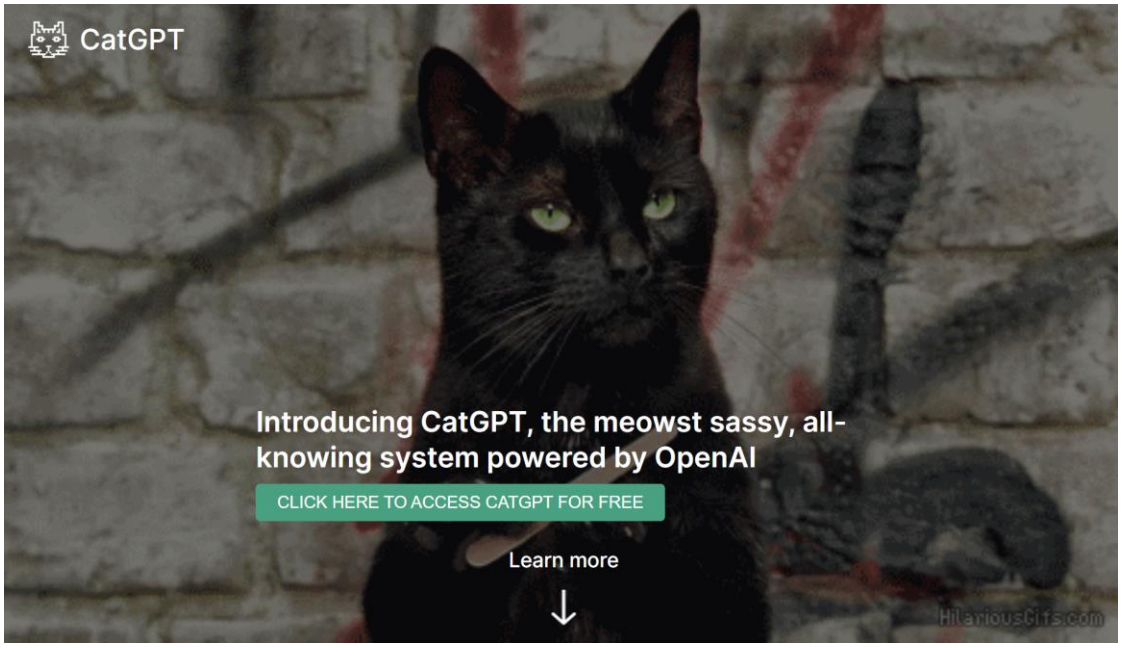


DIAGRAMA DE CLASES CATGPT

INTRODUCCIÓN

La Unión Europea ha decidido que el comportamiento de los ordenadores como personas ha través de la inteligencia artificial es peligroso y además atenta contra los derechos de autor ya que estas copian el comportamiento de otros seres humanos, así que han decidido banear a esta. Por ello, para poder seguir investigando con inteligencias artificiales, se ha decidido crear CatGPT, una IA que imita el comportamiento de un gato.

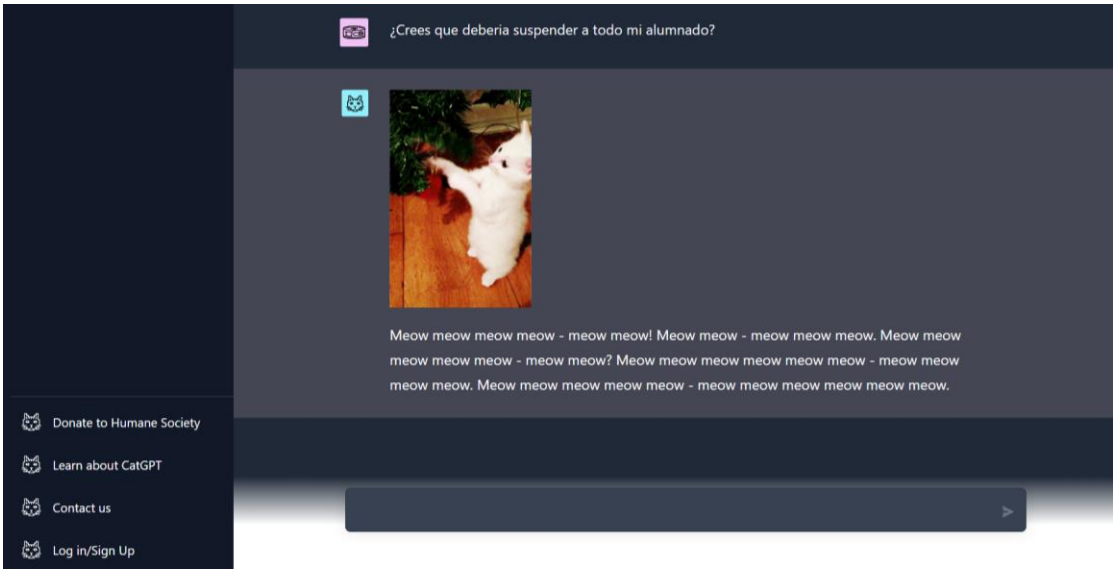


ENUNCIADO DEL EJERCICIO

Se plantea construir una aplicación web para alojar la nueva herramienta de CatGPT. Esta debe de tener lo siguiente:

- Personas, vamos a tener distintos tipos de personas en la aplicación, pero los datos comunes de estas seran que tendran un nombre, contraseña y email.
- Usuarios, de estos nos interesa guardar ademas de los datos generales de una persona, su tipo de gato favorito.
- Empleados, de estos debemos guardar su salario ademas de los datos generales de persona, y el tipo de inteligencia artificial en el que estan especializados. Ademas, estos tendran un metodo programar en el que crearan distintos gatos virtuales que interactuaran con los humanos.
- Humane Society, es la asociacion a la que pertenece el grupo de empleados trabajando en CatGPT. De estas debemos guardar el dinero recaudado, su direccion fiscal, y tendra un metodo en que sea esclavizar con una entrada de un objeto de tipo Empleado. Humane Society no podra existir sin sus clientes ni sin sus empleados
- Tipo de inteligencia artificial, de esta nos interesa saber si es una red neuronal, un sistema de aprendizaje automatico, o un sistema de aprendizaje profundo.
- Gato Virtual, este sera el que interactua con los seres humanos. Tiene los metodos entrenamiento por un lado ,e interactuar con clientes, donde tienen de entrada un objeto de tipo Cliente.

A continuación se muestra un ejemplo de funcionamiento de CatGPT.



SOLUCIÓN CATGPT

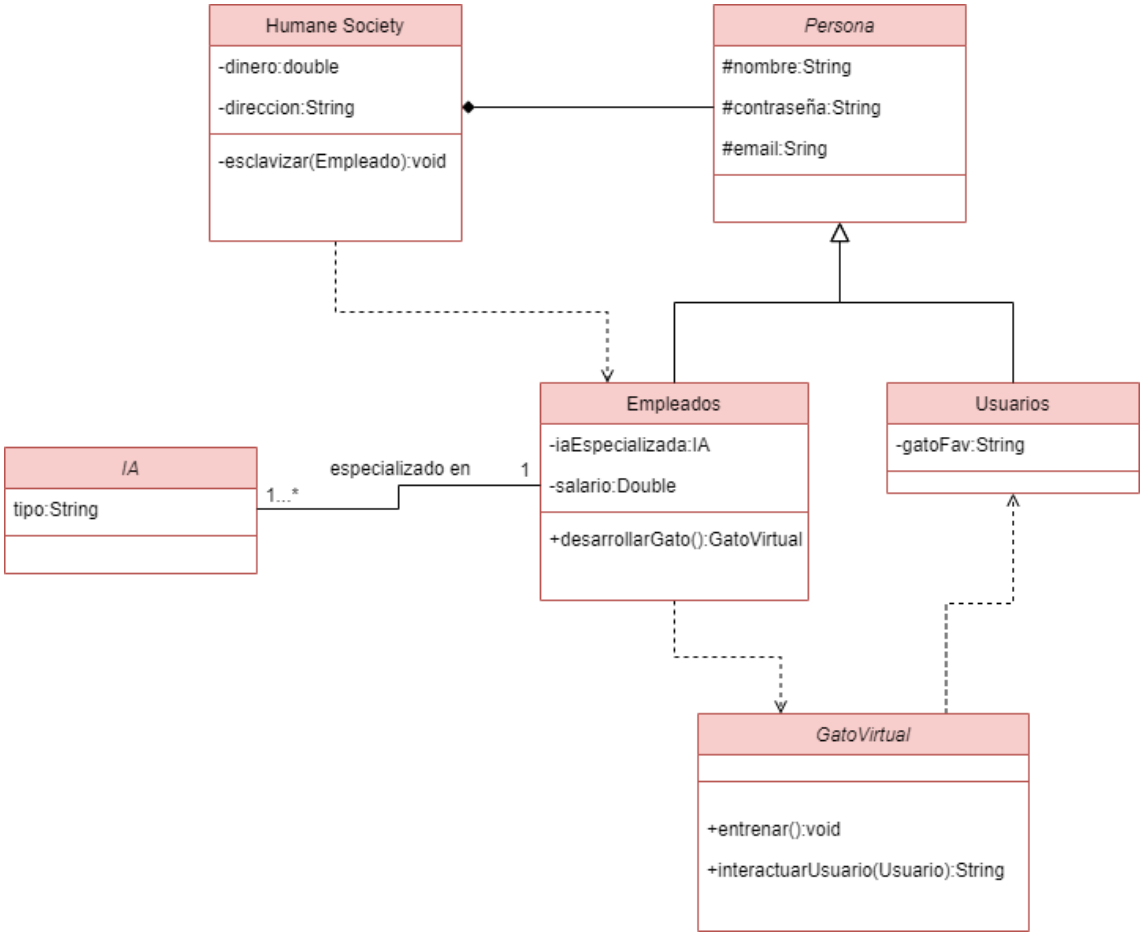


DIAGRAMA DE CLASES CONEJOS REFINADOS

INTRODUCCIÓN

El restaurante Conejo Gourmet, especializado en alta cocina para conejos refinados, esta preocupado por la cantidad de clientes que reservan para comer en su local pero luego no se presentan, ya que esto hace un daño muy grande al negocio.

Por lo tanto, han decidido pedirnos un sistema informático para la gestión de estas reservas.

ENUNCIADO

El sistema que nos han encargado debe de modelar lo siguiente:

- Un restaurante con nombre, dirección, y número de clientes máximos que puede acoger. Tiene un método crearReserva() con una entrada de tipo ConejoRefinado y generando un objeto de tipo reserva.
- ConejosRefinados, estos son los objetos que se encargan de modelar los conejos clientes. Debemos guardar su tarjeta de crédito para cobrarle si no se presenta, y su nombre. Tiene un método que es gestionarReserva() donde le entra un objeto de tipo reserva.

- ConejosCocineros, de estos conejos guardamos su nombre, salario y su horario.
- Reservas, de aquí guardamos la fecha, el menú reservado y el número de mesa reservado.
- Menú, compuesto por distintos platos y un precio.
- Platos, formado por los ingredientes de este y los alergenos que contiene.

SOLUCIÓN FINAL CONEJOS

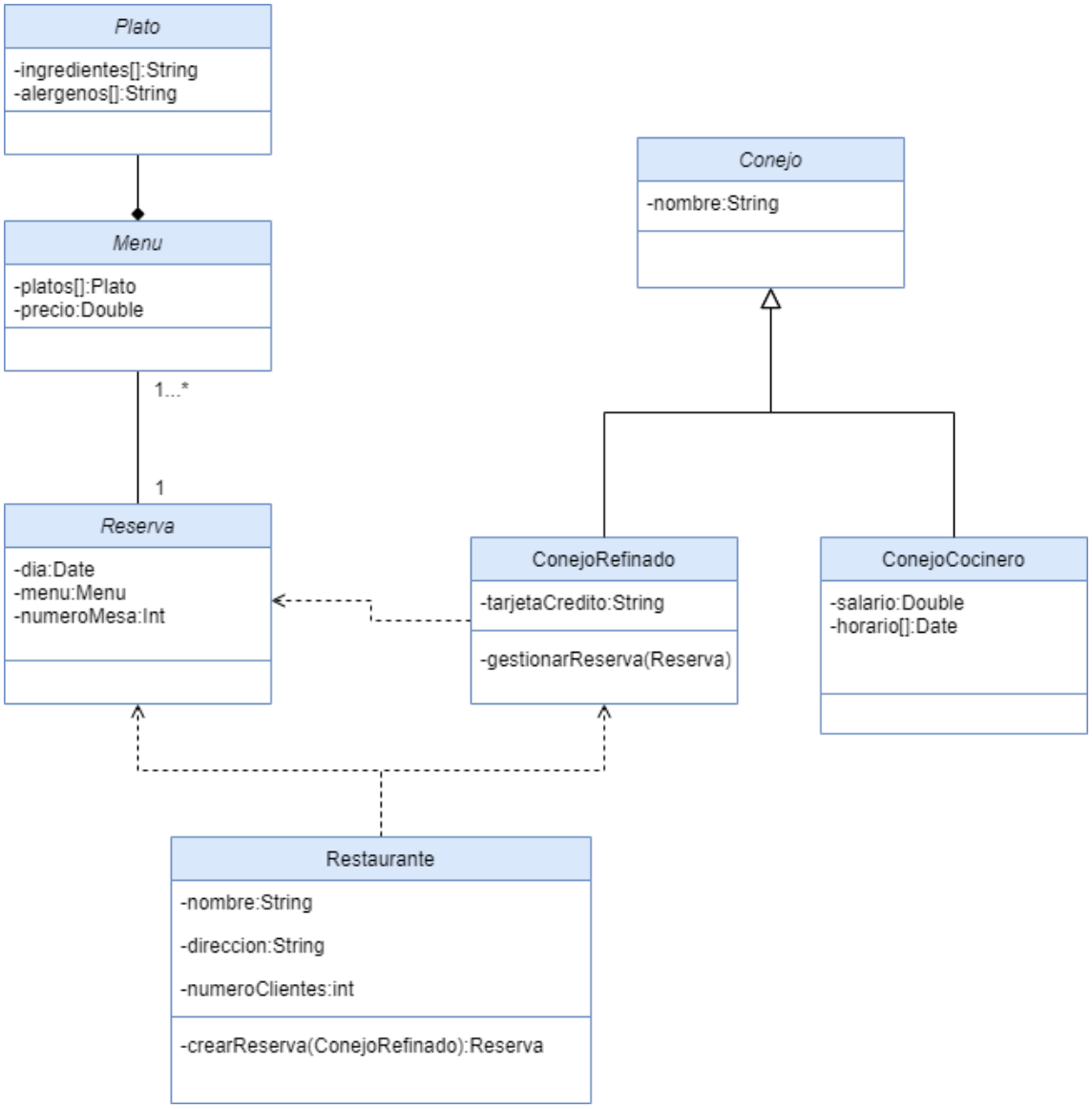


DIAGRAMA DE CLASES TORTUGAS NINJA

INTRODUCCIÓN

El famoso futbolista Mbappé ha sido amenazado por el temido clan de las tortugas ninja debido a que estas son madridistas (ya me joderia) y están muy enfadadas con este jugador porque no ha sido fichado por el Real Madrid.

Por lo tanto este va a instalar un nuevo sistema de vigilancia en su casa y nosotros debemos de modelar el sistema informático de este.

ENUNCIADO

Debemos de modelar un diagrama de clases que cumpla con lo siguiente:

- Vamos a tener cámaras de vigilancia, las cuales tienen un método detectarTortuga() que devuelven la tortuga ninja detectada.
- Tenemos la clase TortugaNinja en la que se modela el nombre y color de la tortuga ninja detectada.
- También tenemos un servidor que tiene como atributos cámaras de vigilancia, el servidor no puede existir sin las cámaras ya que su función seria inútil, y un método que es llamarPolicia() que devuelve true o falso, dependiendo de si la policía responde a la llamada o no.
- También tenemos un trabajador de seguridad que tiene asignado un servidor. En este caso solo tenemos un servidor y 4 trabajadores de seguridad, que se irán turnando. Trabajador tiene como atributos servidor asignado y nombre.
- Tenemos un trabajador especial que es Trabajador Mimado que tiene una reducción de jornada, la cual expresaremos con un entero.

SOLUCIÓN DIAGRAMA TORTUGAS NINJA

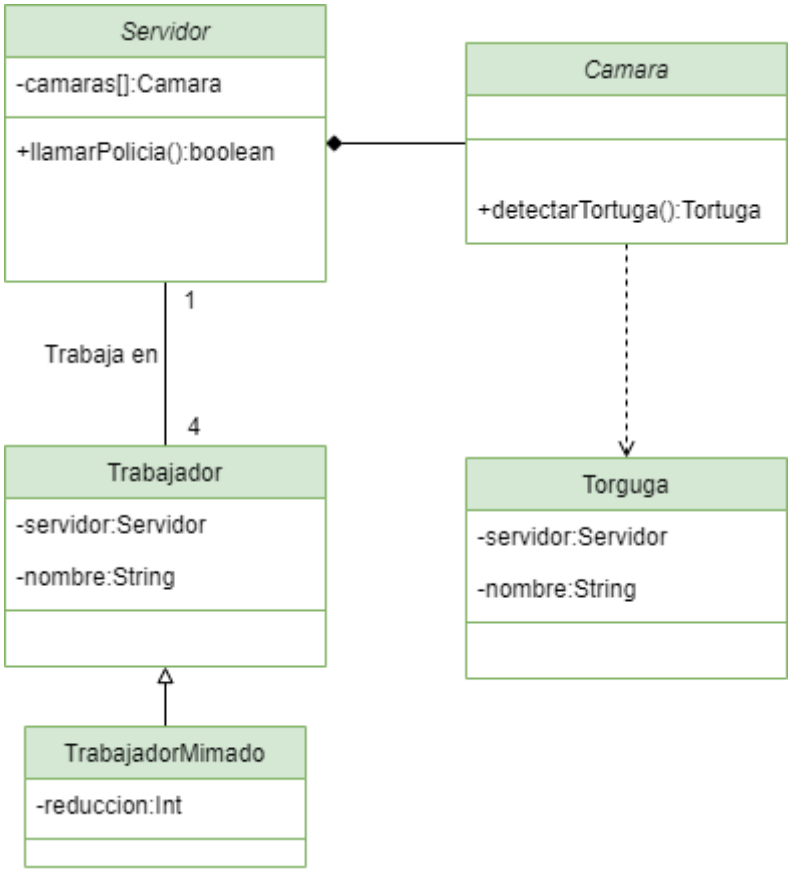


DIAGRAMA POWER GUANCHES

INTRODUCCIÓN

Los Power Rangers están preocupados porque un nuevo actor puede quitarles el trabajo, los Power Guanches. Estos son algo así como una versión barata canaria de los mismos.

Para saber quienes son, han decidido crear un sistema de espionaje y análisis de datos de estos por si algún día tienen que enfrentarse.

ENUNCIADO

Modelar un diagrama de clases que cumpla con lo siguiente:

- Debe analizar a los Power Guanches, teniendo en cuenta su color, habilidades y comida canaria favorita.
- De la comida canaria vamos a guardar sus ingredientes por si nos gusta y queremos buscarla en el super más tarde.
- Megazord, esta es la unión de todos los Power Guanches. Necesita a todos sus miembros para estar completa. Constara de 5 Power Guanches. Tiene como atributo el porcentaje de energía que le queda.
- Comidita familiar, a los Power Guanches les gusta de vez en cuando organizar comidas familiares, eligen un lugar y día para esta. No es necesario que asistan todos los Power Guanches.
- Power Guanchitos, estos serán Power Guanches pero tienen un atributo especial que es edad, ya que son pequeñitos y hasta que cumplan 18 no pueden ser Power Guanches, y tienen una habilidad que es comerPlatano() con un objeto de entrada Plátano para ponerse fuertes.
- Plátano, debe de tener un boolean para asegurarse de que es de las Islas Canarias (o será eliminado) y una fecha de caducidad.

SOLUCIÓN DIAGRAMA DE CLASES POWER GUANCHES

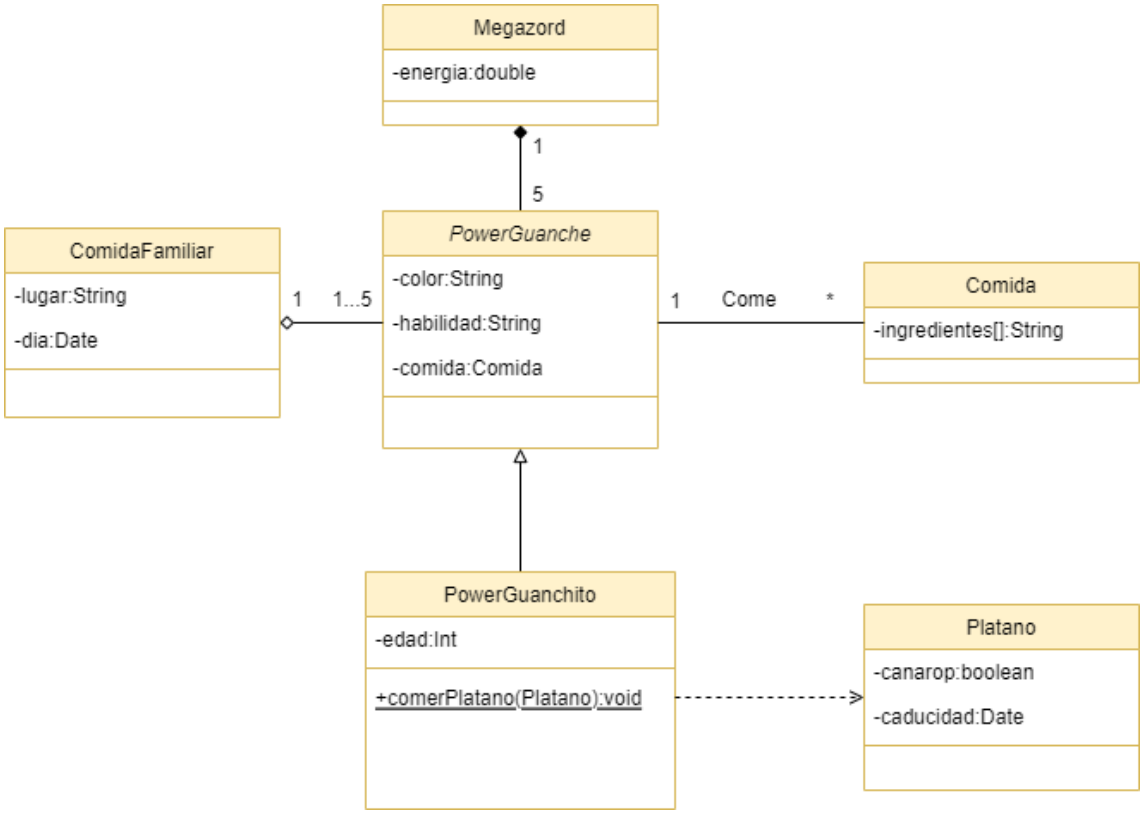
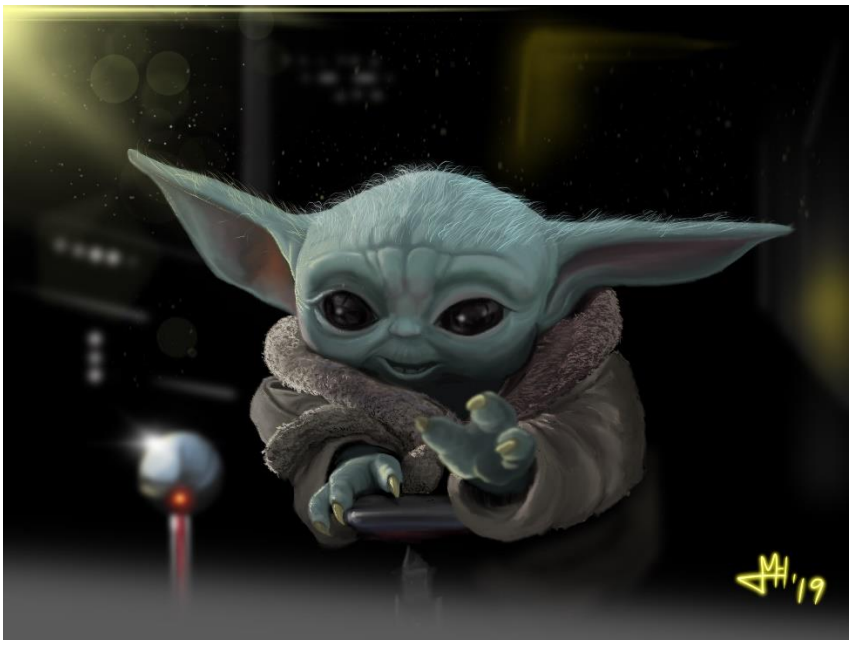


DIAGRAMA DE CLASES BABY YODA

INTRODUCCIÓN

El pequeño Baby Yoda esta ya cansado de que no le dejen jugar con la bolita de los controles de la nave, por lo tanto ha decidido fundar una juguetería para futuros jedi y necesita nuestra ayuda para modelar el sistema informático de esta.



Se valorará que se identifiquen correctamente las clases, atributos con sus tipos correspondientes, métodos con sus atributos de entrada y salida, relaciones entre clases y cardinalidad de estas si la tuviesen.

ENUNCIADO

Se necesitan modelar las siguientes clases:

- Una clase Jedi, con atributos nombre y control de la fuerza que poseen. Esta tendrá un método que será usar comprarJuguete() que devolverá un objeto de clase Juguete.
- La clase Juguete, donde se identifica la edad recomendada.
- Una especialización de la clase Juguete, que será SableLaser. Este, aparte de tener la edad recomendada de juguete, tendrá un atributo color.
- Los Jedi pertenecen a la orden Jedi. Esta no puede existir sin sus Jedi, y un Jedi puede y debe pertenecer solo a 1 orden.
- Una clase BabyYoda. Esta clase no es una especialización de la clase Jedi, ya que a diferencia del resto de jedis, Baby Yoda no está asociado a ninguna orden, la modelamos porque Baby Yoda es genial y se lo merece. Sus atributos son nivelDeAdorabilidad, nivelDeHambre (aunque sea pequeñito Baby Yoda es un gordo) y tendrá un método que será robarJuguete(), en este caso el objeto juguete es un atributo de entrada del método, es decir, robarJuguete(juguete:Juguete):void . Esto es debido a que Baby Yoda no cree en el capitalismo.

SOLUCIÓN BABY YODA

DIAGRAMA DE CLASES THE WITCHER

INTRODUCCIÓN

El reputado brujo Geralt de Rivia esta enganchado al juego de cartas Gwynt, hasta tal punto que ha decidido organizar un torneo de este para generar algunas monedas, ya que se paso de hidromiel la noche anterior y ha contraído algunas deudas Dijkstra que debe pagar cuanto antes.

Por lo tanto, nos pide que diseñemos un diagrama de clases para el futuro sistema de organización de torneos de Gwynt.

ENUNCIADO

- El Gwynt es un juego de cartas en el que tenemos distintas facciones. Por lo tanto, tendremos una clase Carta con su nombre, descripción, y poder, y una especialización de esta que serán las distintas facciones, las cuales son:
    - Monstruos, que tienen una habilidad especial de mantener. Estas se representan con un método mantener().
    - Imperio Nilfgaardian, con la habilidad desempatar() a su favor.
    - Reinos del Norte, con la habilidad sumarCartaExtra().
    - Scoia'tael, con la habilidad decidirTurno(Jugador), nótese que este método tiene como entrada un objeto Jugador.
    - Skellige, con la habilidad colocarAzarCementerio().
- La clase Jugador, donde guardamos su nombre y raza. También un jugador tiene una baraja la cual será un array de objetos Carta.
- Torneo, esta clase esta compuesta por jugadores, y los almacenamos con un array jugadores[] con objetos de tipo Jugador. Torneo no puede existir sin jugadores.

SOLUCIÓN THE WITCHER