En la antigua Grecia existían muchos dioses, pero tres eran los principales: Zeus, Poseidón y Hades.

**Zeus** era el dios más importante de la mitología griega. Era el señor de todos los dioses y de los humanos. Regía sobre la luz, el cielo despejado y calmo, pero también era el dios del rayo y quien provocaba las tormentas y la lluvia que favorecía los cultivos. Se lo consideraba garante de la justicia y el orden y protector de las personas justas.

**Poseidón** era el dios que reinaba sobre el mar y las corrientes de agua. Era justo, pero con un punto iracundo. Era el que hacía las ahogadillas a los otros dioses.

**Hades** es el antiguo dios griego del inframundo, el lugar al que van las almas humanas después de la muerte. Con el tiempo, su nombre se convirtió en sinónimo de su reino.

Pero estos dioses no se han ido, siguen existiendo y se han puesto en contacto con nosotros para actualizarse un poco. Lo de lanzar rayos y crear tormentas está muy bien pero se han quedado un poco obsoletos y quieren reciclarse. Así que nos han encargado una aplicación web que gestione sus cosejas de dioses prometiendo que, si quedan satisfechos, harán FP Dual con nosotros… pero eso es otra historia.

Nuestra aplicación, muy resumida, servirá para registrar en ella a los humanos que vayan naciendo, con sus características y someterlos a pruebas que regirán su destino final: campos Elíseos o Tártaro.

Humanos

Los humanos se registrarán en el sistema indicando su **nombre**, **correo** y **contraseña**. En el nuevo sistema que pretenden implantar los dioses permitirán que los humanos consulten como va su destino (una ventaja significativa respecto a otros dioses). Tras ser verificado ese usuario se le asignarán valores aleatorios (entre 1 y 5) a los campos: ***sabiduría, nobleza, virtud, maldad*** y ***audacia.***

Los dioses también tienen esos campos (ya que a diferencia de otras religiones estos dioses tienen sus debilidades lo que los hacen más cercanos y campechanos… bueno, mejor más cercanos sólo). De hecho los valores que tienen los dioses para estos campos son:

Zeus [5, 3, 1, 2, 5]

Poseidón [5, 4, 2, 2, 5]

Hades [5, 4, 4, 4, 2]

Lógicamente son los dioses y podrán modificar, si lo desean, los valores de estos campos que tienen asignados. Pero ya volveremos con los dioses luego.

Cuando una **persona nace** (se registra) también será asignado a la “**protección**” de uno de los tres dioses principales. Esta **protección**, como ya veremos, será que ese dios les enviará pruebas que el humano tratará de superar. Dicha protección no se realiza de forma puramente aleatoria, se tendrán en cuenta las características del humano, las características de los dioses y el que tenga una afinidad mayor se queda con el humano. Le hemos preguntado a los dioses por el algoritmo que quieren que usemos para **calcular la afinidad,** pero nos han indicado amablemente: tronando y desatando alguna que otra tormenta, que no tienen ni pajolerakis idea de lo que es un algoritmo y que para eso han contratado nuestros servicios…

Los humanos tendrán también un **destino**, que parte de 0 y que irá aumentando / disminuyendo según esos humanos vayan pasando pruebas en su vida. Al morir, según ese campo, esos humanos irán al Tártaro o a los campos Elíseos. Eso se determinará más adelante.

Al entrar al sistema, el humano, podrá ver las pruebas asignadas y resolverlas; será informado del resultado de las mismas. También podrán consultar sus características y su destino (aunque no podrán modificar nada de esto). Si un humano muere podrá entrar al sistema, pero solo verá una página en la que se resume su vida y dónde ha acabado finalmente y ya no podrá hacer nada en el mismo.

Dioses

Los dioses podrán **crear pruebas y podrán asignarlas** (a aquellos hombres que están bajo su protección) de la siguiente manera: puntualmente, a un/varios hombre/s, a todos o al que mejor se ajuste a esa prueba (al más afín). En esto los dioses están confusos y abiertos a que le propongamos formas útiles de hacerlo (si se nos ocurre alguna) **además de las propuestas**.

Las personas también pueden ser **creadas por los dioses** en número indefinido. Estas personas creadas tendrán las mismas características que las registradas, claro, y ya estarán activadas. Entre los nombres posibles, además de los fake que podamos poner con nuestras herramientas habituales, los dioses nos proponen estos: Jaimeniadis, Javierakis, Manolakos, Laurakas, Raulopoulou, Helena, Mariniakas, Gonzalopulous, Juannakis...

Pruebas

**Oráculo**. En estas pruebas el dios registrará tres tipos de pregunta:

- **Respuesta libre.** Serán del tipo: el dios te hace una pregunta confusa, incoherente, contradictoria, irresoluble... como nosotros en clase, vamos. El caso es que dicho dios registrará la pregunta y las palabras clave asociadas como respuestas correctas. Es decir que, si de entre toda la respuesta, existen esas palabras, se sumará / restará al destino de esa persona lo ganado / perdido al superar o no esa prueba.

Ejemplo:

1. ¿Cuál es el secreto para ser feliz

2. ¿Por qué hay algo en vez de nada?

3. ¿Es el altruismo real o un mito?

4. ¿Tenemos libre albedrío?

Esta **prueba se considera pasada** si el humano, en su respuesta, utiliza más de, por ejemplo, el 70% de las palabras clave asociadas. O lo que los dioses estimen oportuno, estos parámetros podrán ser modificados por los dioses en cualquier momento.

- **Valoración** (de 1-5).

Los dioses propondrán preguntas también, pero estás serán valoradas por el humano del 1 al 5.

Ejemplo:

1. ¿Podemos experimentar el mundo de manera objetiva?

2. ¿Existe una ética objetiva?

3. ¿Le gustaría ser inmortal si pudiera?

Estas preguntas están **asociadas a una característica** del humano y también tendrán una **cantidad de destino** en juego. Si la respuesta es afín (otra vez los dioses nos dejan a nosotros determinar con nuestro algoritmo como calcular esa afinidad) a la característica del humano se pasa la prueba y se suma/resta la cantidad de destino.

- **Elección**, una pregunta en la que se contesta al 50%.

Ejemplo:

1. Estás en una casa y se declara un incendio. Sólo puedes llevar una cosa y hay un gato y un cuadro muy valioso: ¿qué salvas?

2. ¿Eres libre en tus elecciones? (Cuidado de no caer en un bucle recursivo al contestar)

**La respuesta está asociada a una característica**, por ejemplo, esta pregunta implica un 4 de sabiduría y tendrá una cantidad de destino asociada. Será el programadoropoulus o programadoropoulas el que determine si se ha superado la prueba o no (esperemos no fallar a los dioses).

Fin de partidakas

Todos los dioses podrán crear y editar humanos (que quedarán bajo su protección) pero **solo Hades podrá eliminarlos**. El dios Hades tendrá una opción especial dentro de sus habilidades: cepillarse a los humanos. Podrá elegirlos manualmente y que casquen o lanzar un proceso aleatorio que elija a un número determinado de humanos y se los cargue. El dios podrá comprobar qué humanos han atravesado la laguna Estigia (han palmao) en todo momento, con la posibilidad de ordenarlos por la fecha de la muerte.

Los humanos al morir irán a los campos Elíseos o al Tártaro. Esto vendrá determinado por su cantidad neta de destino. **Si esa cantidad supera lo que los dioses exigen para ir al Elíseo,** vivirán eternamente rodeados de ninfas y efebos pero si no... mejor no os lo decimos. **Esta cantidad la puede cambiar cualquier dios en cualquier momento.**

Comunes

Todos los usuarios podrán entrar al sistema, cambiar su foto de perfil, consultar sus datos y cambiar la contraseña. El sistemas deberá proporcionar un mecanismo de recuperación de contraseña basado en el correo electrónico.

Valoraciones

**Se tendrá en cuenta de modo general:**

* Diseño de la BD que recoja toda la información de forma correcta.
* Que lo programado se ajuste a lo pedido.
* Una planificación de tareas coherente.
* Que los sprints de desarrollo cumplan lo planificado.
* El proyecto deberá estar sincronizado con un repositorio Git y Board de planificación. Todo esto será consultable por los profesores en cualquier momento del desarrollo del proyecto.

**Desarrollo web entorno servidor:**

* Elección y agrupación de rutas.
* Verbos y respuestas estándar.
* *Eloquent* y/o *Query Builder*.
* Protección de rutas con *token* de seguridad. *Middlewares* y *abilities*.
* *Migrations*, *Factories* y *Seeders*.
* Validación de datos. *Validator*.
* Algoritmos elegidos para las tareas propuestas.
  + Asignación humano – dios por afinidad.
  + Pasar pruebas puntuales.
  + Oráculo – respuesta libre.
  + Oráculo – valoración.
  + Oráculo – elección.
  + Podremos reutilizar algoritmos para calcular afinidades, con pequeños retoques eso sí, porque estamos tratando con dioses… mejor no provocarles
* Parametrización del sistema.
* Realización de *tests*.

**Desarrollo web en entorno cliente:**

* Generación de código estructurado y modular. Utilización de POO. Utilización correcta de funciones.
* Utilización correcta de estructuras de almacenamiento con JS.
* Utilización correcta del proyecto con webpack e importación y exportación de módulos.
* Gestión correcta de eventos.
* Validación de formularios.
* Utilización correcta de promesas en la comunicación asíncrona con el servidor utilizando fetch.
* Planificación correcta y aplicación eficiente de la metodología SCRUM.

***Si quieres hacer reír a los dioses… cuéntales tus planes.***

**ANEXO: METODOLOGÍA SCRUM**

SCRUM es una metodología ágil en la que se aplican regularmente un conjunto de buenas prácticas para trabajar en equipo y obtener el mejor resultado en el proyecto.

En SCRUM se realizan entregas parciales y regulares del producto final, priorizándolas según el beneficio que aportan al receptor del proyecto (cliente).

Los puntos destacables de esta metodología de trabajo son:

**ROLES:**

* **Product Owner:** se encarga de analizar qué es lo que quiere el cliente final en su proyecto y lo detalla en una lista priorizada (**product backlog).**
* **Scrum Master:** organiza y facilita el trabajo. Defiende la metodología a llevar a cabo ante el product owner.
* **Team:** equipo de desarrolladores. Analizan las tareas del **product backlog**, detallan los ítems que van a contener, su prioridad y el tiempo que les va a llevar realizarlas.

**Una persona puede tener más de un rol. Por ejemplo, el scrum master puede formar parte del equipo de desarrollo.**

**FUNCIONAMIENTO:**

El trabajo se organiza en **sprints**. Un sprint tiene un tiempo acotado (por ejemplo 2 semanas) y en él se incluyen las tareas (historias) establecidas en el **product backlog.**

El sprint comienza con una planificación **(planning)**: se reúne el **team, product owner y scrum master.** Se analiza cada tarea establecida por el product owner en el backlog. Se prioriza, determina la dificultad técnica por el equipo de desarrollo y se establece el tiempo que tardará en completarse. Una vez analizada, se decide si esa tarea **“cabe”** en dicho sprint, en cuyo caso, el team se **COMPROMETE** a realizarla en dicho sprint.

Cuando finaliza el sprint, se realiza una **retrospectiva** analizando lo que se ha hecho bien y lo que se ha hecho mal, para proponer mejoras al respecto. Después se realiza una **demo** de las historias del sprint que se han **completado,** ante el **product owner (**y a veces ante el cliente final).

Cada día del sprint se hará una pequeña reunión denominada **daily** (tiene que ser corta (10’) y se obliga a estar de pie para que no se alargue), en la que cada miembro del equipo de desarrollo cuenta en lo que ha trabajado, las dificultades y si necesita ayuda por parte del resto del equipo.

Cuando comienza un sprint, existe un tablero donde se encuentran todas las tareas sin asignar pendientes del sprint (**backlog**), y que cada desarrollador irá escogiendo (teniendo en cuenta al equipo) y asignando según sus preferencias, ritmo de trabajo, etc.