



Prueba técnica desarrollo

Índice

Índice	1
Sugerencias y requisitos	2
Ejercicio 1	2
Decodificación	2
Ejercicio 2	4
Sistema de citas online	4



Sugerencias y requisitos

- **No se puede hacer uso de frameworks de PHP y CSS** (ejercicio 2).
- Utiliza contenedores de docker LAMP (Linux Apache Mysql, PHP), para iniciar la aplicación más fácilmente (ejercicio 2).
- Se valorará utilizar el idioma inglés, para programar (variables, comentarios, etc).
- Se debe entregar el código totalmente operativo para su evaluación en **un repositorio público de github**.
- Será necesario un archivo **README.md**: con la información relevante de la aplicación. (cómo iniciarla, menús, configuraciones, etc).
- Se valorará la calidad y claridad del código.

Ejercicio 1

Decodificación

Tiempo estimado: 1 hora.

¿Nos ayudas? Algo ha ocurrido en el sistema de puntuaciones de nuestro juego “Pandora Invaders”. Un hacker enfadado por sus malas puntuaciones nos ha secuestrado el fichero donde se guardan los resultados. Pero afortunadamente no ha desaparecido, sólo lo ha codificado para que sólo él pueda recuperarlo. Desde entonces estamos investigando y hemos descubierto algunos datos sobre la codificación que ha utilizado y que serán útiles para resolver el problema:

- El fichero manipulado parece un fichero CSV. Cada línea contiene la puntuación de un usuario.
- La primera columna es el nombre de usuario.
- La segunda columna son los dígitos del sistema numérico utilizado para codificar la puntuación.
- La tercera columna es la puntuación codificada.



Aquí tenemos el contenido del fichero:

```
Patata,oi8,oo
ElMejor,oF8,Fo
BoLiTa,0123456789,23
Azul,01,01100
OtRo,54?t,?4?
Manolita,kju2aq,u2ka
PiMiEnTo,_-/.!#,#_
```

Pista 1: Los dígitos de los sistemas de numeración siempre aparecen de menor a mayor (ejemplo: 0123456789).

Pista 2: La puntuación de Manolita es 544. Usando el lenguaje que quieras, implementa un código que muestre los resultados con las puntuaciones decodificadas de los usuarios.

Solución:

```
Manolita,544
PiMiEnTo,30
OtRo,38 (ejemplo: ?4? sería la posición 212 donde 2*4 2+1*4 1+2*4 0 = 38)
BoLiTa,23
Azul,12
ElMejor,3
Patata,0
```

Teniendo en cuenta toda esta información necesitamos que uses el sistema que quieras para implementar un código que muestre los resultados de las puntuaciones decodificadas de los usuarios.



Ejercicio 2

Sistema de citas online

Tiempo estimado: Entre 2 y 6 horas.

No se puede hacer uso de frameworks de PHP y CSS.

Se pide desarrollar una mini aplicación en PHP.

Necesitamos crear un sistema de reservas online para una clínica especializada en psicología. Por un lado tendremos un formulario en internet que solicita los siguientes datos:

- Nombre
- DNI
- Teléfono
- Email
- Tipo de cita: Primera consulta, revisión.

Via AJAX, después de que el usuario introduce el DNI, si este existe en la BBDD debemos actualizar el control de tipo de cita, para permitir REVISION. Si no existe, solo se podrá elegir PRIMERA CONSULTA.

Antes de enviar el formulario, validar los campos necesarios.

Las peticiones se irán clasificando por orden de llegada y se asignará la primera hora disponible, teniendo en cuenta que: todos los días de la semana son aceptables para una cita (sin festivos), que las consultas duran una hora y que solo se pueden asignar de 10 de la mañana a 10 de la noche. A medida que se van asignando horas, hay que tenerlas en cuenta para asignar nuevas citas.

No puede darse cita a dos pacientes en la misma hora el mismo día ni tampoco dejar horas vacías.

Se debe asignar día y hora a esa cita, guardarla en BBDD.



Mostrar mensajes de validación. ejemplo: “Se ha generado la cita”.

Implementar modelo de datos en MySQL. Hay libertad para modelar los datos como se quiera en función de las necesidades planteadas.

Implementar formulario de recepción de datos (incluyendo AJAX para saber si es primera consulta o no).

Implementar sistema de asignación de citas/horas, teniendo en cuenta que habrá concurrencia (es decir, pueden llegar N peticiones a la vez por internet).