



## PROYECTO 2ª EVALUACIÓN - GESTIÓN PACIENTES Y VISITAS

La clínica 'Saca-Muelas' nos ha pedido que le realicemos un desarrollo web para la gestión de sus pacientes y visitas.

### PRIMERA PARTE

La primera parte del desarrollo se centrará en la resolución de la gestión de visitas.

De ellas se quiere gestionar:

- El paciente que acude
- La fecha
- El importe
- Si la ha pagado o se queda debida

Se deberá poder:

- Visualizar la información
- Añadir visitas
- Eliminar visitas
- Modificar visitas

Además nuestro cliente quiere que las personas que estén gestionando las visitas puedan ver de una manera visual y rápida las visitas pagadas y las que no. También necesita que se diferencien aquellas visitas con un precio superior a 250€.

### SEGUNDA PARTE

Con la primera parte en funcionamiento, ahora abordamos la gestión de los pacientes.

De los pacientes se pretende almacenar:

- Identificador de paciente
- Nombre
- Dirección

De los pacientes se podrá:

- Visualizar la información
- Añadir pacientes
- Eliminar pacientes
- Modificar pacientes

Además de poder visualizar el listado entero de visitas, deberemos de poder visualizar el listado de visitas de un paciente en particular.

### TERCERA PARTE

Ahora nuestro cliente quiere obtener estadísticas y datos globales para poder tomar decisiones.

Le gustaría contar con una pantalla en la que pueda ver información en tiempo real de:

- Total de € en facturas.
- Total de € de visitas cobradas y total de € de visitas no cobradas.
- Balance total.
- Número de visitas cobradas y número de visitas no cobradas.



-Total de pacientes.

-Total de pacientes con todas las visitas pagadas y total con alguna no pagada.

## NOTAS:

- Puedes utilizar el archivo data.php para generar visitas ficticias y empezar a 'jugar'.
- Tal y como hicimos en la práctica anterior, nos hará de 'pasarela' un archivo csv.
- Separa las clases y las páginas (visualización, edición,...) en ficheros diferentes.
- Obligatorio autoloader.
- css en archivos separados al efecto.
- Notación camelCase.
- Nombre de clases, propiedades, métodos, funciones, variables, etc. en inglés y significativos.
- Comentarios en inglés. Comenta sólo lo necesario. Recuerda que el código debería hablar por sí mismo.
- Repositorio en GitHub privado y yo como colaborador (VicenteMonfort). Commits habituales y trabajo en clase.
- Puedes incluir un readme.md para explicar lo que creas necesario.

## RÚBRICA DE CORRECCIÓN

HASTA PRIMERA PARTE - PUNTUACIÓN MÁXIMA 6

HASTA SEGUNDA PARTE - PUNTUACIÓN MÁXIMA 8

HASTA TERCERA PARTE - PUNTUACIÓN MÁXIMA 10

### GRADO DE CONSECUCIÓN DE CADA PARTE

- Para poder superar una parte, debe de estar superada la anterior.
- Para poder superar una parte, esta deberá de cumplir los requisitos en la especificación de la parte.
- Cualquier requisito no cubierto, podrá decrementar la nota.
- Se considera un error grave, cualquier código implementado y no funcional.
- Se considera un error grave, que sobre cualquier porción de código implementado y ante cualquier pregunta por parte del profesorado, no se sepa responder correctamente.
- Se considera un error grave, que ante una modificación sugerida por el profesorado en un tiempo concreto, no se consiga de manera exitosa.