

Proyecto I0I



Ismael Aliaga Molina

1º Desarrollo de Aplicaciones Web

080 formación

Fase1	2
Creación de la base de datos.....	2
Maquetando la estructura de nuestra aplicación	6
Conectando nuestra aplicación con la base de datos	9

Creación de la base de datos

Lo primero en este proyecto ha sido pensar la estructura, para ello tenemos que tener claro las normas y reglas que tiene que seguir el proyecto:

- Todos los empleados comienzan con 100 puntos
- Cada hora que el empleado se ausente del trabajo sin una debida justificación, descontará 10 puntos
- Cada hora que el empleado llegue tarde, descontará 5 puntos
- Cada hora que el empleado esté en su puesto y realizando sus actividades pertinentes, sumará 1 punto
- No habrá máximo ni mínimo de puntos ni por arriba, ni por abajo.
- Cuando un empleado tenga menos de 50 puntos, no opta al premio en ese mes y será retirado del equipo en el que estuviera trabajando (en caso de estar en alguno)

Por otro lado, tenemos que tener en cuenta los requisitos que quiere el cliente que tenga el proyecto:

- Que su servicio esté en la nube con su dominio
- Tendrá parte pública y parte privada (para poder manipular los datos)
- Podrá gestionar a los empleados (usará alias para que no sean fácilmente identificados)
- Controlar que el total de horas por el día indicado, introducido para un empleado sea correcto.
- Obtener un informe a una fecha en concreto, que muestre el estado de los puntos de todos los empleados

Teniendo claro todos los puntos anteriores, comenzamos a diseñar la base de datos. En un primer momento pensé en mostrar los puntos de los empleados con un mínimo de 0 puntos y un máximo de 100 puntos y que a la hora de visualizar los datos se mostrara con una barra de progreso, viendo esta idea me salieron 3 tablas (Trabajadores, Registros y Barra_Progreso).

En la tabla trabajadores se iba a introducir un id para cada usuario los nombres y apellidos ficticios, otro campo con el nombre real, otro con la puntuación de cada trabajador y un campo con clave foránea que fuera a la tabla Barra_Progreso. En la tabla de Barra_Progreso una id de clave primaria, un campo para poner colores hexadecimales y una descripción para saber el color en texto. Para terminar la tabla de Registros tendría una clave foránea con los id de los trabajadores, una clave primaria para los id de los registros, un campo de

fecha, otro campo de disponibilidad _inicial, otro campo disponibilidad_modificar y el último campo disponibilidad_total.

Una vez con este planteamiento de la base de datos y revisando los requisitos y reglas que tenía que seguir, me di cuenta que no hacía falta restringir los puntos de 0 a 100 y me surgía un problema con el apartado de la barra de progreso a la hora de poder calcularla.

Por ello y rigiéndome por lo que el cliente nos pide volví a rehacer la base de datos y me quedé con dos tablas (Trabajadores y Registros) ya que no vi la necesidad de crear otra tabla con los requisitos actuales y es bastante funcional para poder seguir evolucionándola en un futuro.

En la tabla trabajadores:

- id_trabajador (es un id inequívoco para identificar a cada trabajador)
- Nombre_real (es el nombre real del trabajador)
- Nombre (es un nombre ficticio para preservar sus datos reales)
- Apellidos (son unos apellidos ficticios para preservar sus datos reales)
- Disponibilidad (son los puntos actuales de los que pueden disponer los trabajadores, para cumplir con el proyecto por defecto al crearlo tiene el valor de 100)
- Grupo (indica a que grupo pertenece cada trabajador)
- Salario (indica si el trabajador va a tener el bonus del 66.67% de salario o no)
- Baja (indica si el trabajador está trabajando en la empresa o se encuentra dado de baja de la empresa)

En la tabla registro:

- id_registro (es un id inequívoco para identificar a cada registro)
- id_trabajador (un campo relacionado con el id_trabajador de la tabla de trabajadores para saber a qué trabajador pertenece cada registro)
- fecha (indica la fecha en la que se ha producido la modificación de los puntos de la disponibilidad)
- disponibilidad_inicial (indica los puntos que tiene antes de modificarlos)
- disponibilidad_modificar (indica el total de puntos en el día que se va a modificar ya sea sumar o restar puntos, se realiza de esta manera atendiendo al siguiente requisito (*El cómputo de horas se hará de forma totalizada al día, simplemente necesita saber el número de horas total que ha estado cada empleado, que ha faltado y que ha llegado tarde.*))
- disponibilidad_total (indica los puntos restantes de sumar la disponibilidad_inicial con la disponibilidad_modificar, de esta manera nos queda la disponibilidad con la que se queda el trabajador real en ese día y nos sirve tanto para pasarla a la disponibilidad actual que tiene el trabajador en la tabla trabajador como a la hora de realizar el informe saber el total de puntos que tenía cada trabajador el día indicado siguiendo el siguiente requisito del proyecto (*Obtener un informe a una fecha en concreto, que muestre el estado de los puntos de todos los empleados*))

A continuación, adjunto los diagramas conceptuales, lógicos y físicos de la base de datos final ya diseñada.

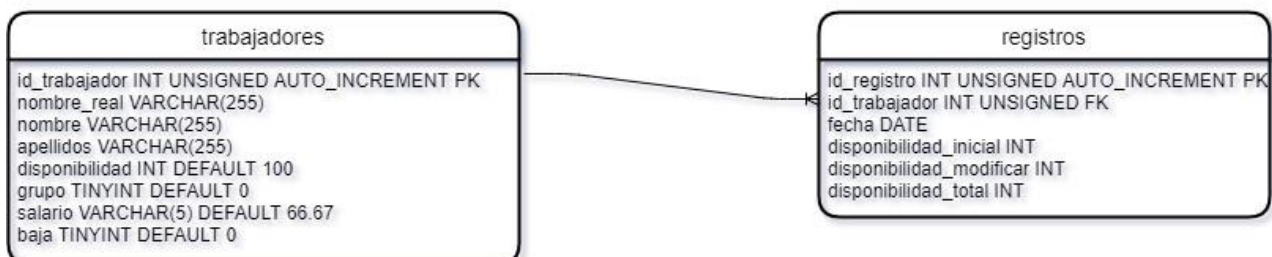
Modelo Conceptual



Modelo Lógico



Modelo Físico



Maquetando la estructura de nuestra aplicación

Una vez diseñada la base de datos me he centrado en visualizar y crear la estructura base para cada apartado. La estructura que he pensado para maquetar la web es la siguiente:

Parte pública

- Tabla donde se muestra los trabajadores dados de alta en la empresa, el id de cada empleado, el nombre y apellidos ficticios (cada empleado sabrá que id le pertenece y sus alias, pero no sabrá el de sus compañeros), los puntos de disponibilidad para cada uno de ellos en el momento actual de la visualización, a que grupo pertenecen y la bonificación salarial en %

Parte privada

- Se mostrará la misma tabla de la parte pública, pero añadiéndole un menú a la web en el que se podrá seleccionar crear trabajadores y generar informes y se le añadirá a la tabla de los trabajadores un apartado nuevo para editar y gestionar a los trabajadores, asignarles la puntuación diaria y poder darlos de alta o de baja dependiendo la situación que se encuentre cada trabajador.
- Tabla donde se muestra toda la información anterior comentada, pero de los empleados que se encuentran dados de baja en la empresa y apareciendo solo la opción para dar de alta o baja al empleado que se encuentra dado de baja para poder reactivarlo si fuera necesario.

Estas tablas en un principio hacerlas con la estructura básica de tablas de HTML, pero investigando vi dos alternativas posibles de hacerlas gracias a CSS, el primer candidato es FlexBox, el cual es muy potente y permite adaptar y personalizar el contenido de una manera sencilla. El siguiente candidato fue Grid, esta opción fue por la que opté porque, aunque FlexBox se podía adaptar a lo que tenía en mente Grid me solucionaba todos los posibles problemas que pudiera tener y además que era una forma más directa de realizar las tablas con menos configuración de CSS y utilizando solamente <div> en el HTML.

Para poder hacernos una mejor idea de como va a funcionar Grid en el proyecto voy a adjuntar la tabla en HTML de la vista pública y como le damos formato en CSS, esto se adapta a todas las demás tablas que vamos a tener que crear.

HTML

```
<div class="contenedor_tabla">
  <div class="contenedor_titulo">Disponibilidad</div>
  <div class="grid_header">ID</div>
  <div class="grid_header">Nombre</div>
  <div class="grid_header">Apellidos</div>
  <div class="grid_header">Disponibilidad</div>
  <div class="grid_header">Grupo</div>
  <div class="grid_header">Salario</div>
  <?php
  while ($trabajadores->fetch()) {
    echo "
    <div class=grid_items>" . $id_trabajador . "</div>
    <div class=grid_items>" . $nombre_trabajador . "</div>
    <div class=grid_items>" . $apellidos_trabajador . "</div>
    <div class=grid_items>" . $disponibilidad_trabajador . "</div>
    <div class=grid_items>" . $grupo_trabajador . "</div>
    <div class=grid_items>" . $salario_trabajador . "</div>
    ";
  }
  ?>
</div>
```


CSS

```
.contenedor_tabla {
  width: 90%;
  background-color: var(--color_fondo_tabla);
  margin: 100px auto;
  box-shadow: 0 0 20px #333;

  display: grid;
  grid-template-columns: repeat(6, 1fr);
  grid-auto-rows: 50px;
}

.contenedor_titulo {
  grid-column-start: 1;
  grid-column-end: 7;
  background: var(--color_cabeceras);
  display: flex;
  align-items: center;
  justify-content: center;
  color: var(--color_letras_botones_titulos);
  font-weight: bold;
  font-size: 1.5em;
}

.grid_header, .grid_items {
  display: flex;
  justify-content: center;
  align-items: center;
}

.grid_header {
  font-weight: bold;
}
```

Conectando nuestra aplicación con la base de datos

Una vez diseñada la estructura de cada apartado que vamos a necesitar en el proyecto llega la hora de añadirle las funcionalidades.

En la parte pública vamos a mostrar como hemos comentado anteriormente el id del trabajador, su nombre y apellidos, lo puntos totales a la fecha de visualización, el grupo al que pertenecen en ese momento y el bonus de salario.

En la parte privada vamos a tener:

La zona de administrar los trabajadores en la que se mostrara todos los datos de la parte pública, pero añadiéndole un menú a la web en el que se podrá seleccionar crear trabajadores y generar informes y se le añadirá a la tabla de los trabajadores un apartado nuevo para editar y gestionar a los trabajadores, asignarles la puntuación diaria y poder darlos de alta o de baja dependiendo la situación que se encuentre cada trabajador.

- Para añadir trabajadores vamos a tener un formulario que va a recoger el nombre real del trabajador y el nombre y apellidos ficticios para el trabajador y lo va a insertar en la tabla de la base de datos de trabajadores.
- Para actualizar la información de los trabajadores dados de alta vamos a realizar otro formulario en el que directamente nos aparecerán los datos del trabajador que podemos cambiar que son el grupo y el nombre y apellidos ficticios y podremos modificarlos directamente.
- Para añadir modificar la disponibilidad vamos a contar con un formulario que va a estar dividido en 1º hora, 2º hora, 3º hora, 4º hora, 5º hora, 6º hora. Estas horas corresponden con un día laboral normal, cuando el día sea martes que solo hay jornada de 5 horas o viernes de 4 horas, el administrador va a poder asignar los siguientes tipos de valores:
 - Sin puntuación ((valor = 0) Se utiliza para asignar que hora del día que está introduciendo no es laboral, por ejemplo, un lunes el responsable de asignar los puntos deberá añadir la puntuación a cada una de las 6 horas, en cambio el viernes solo le asignaría puntuación a las 4 primeras horas y

las 5 y 6 horas seleccionaría la opción sin puntuación para indicarle al programa que esas horas no son laborables y por lo tanto no las tenga en cuenta)

- Asistencia (Le suma 1 punto por cada asistencia al trabajador)
- Retraso (Le resta 5 puntos por cada retraso al trabajador)
- Falta (Le resta 10 puntos por cada falta al trabajador)

Hay que tener en cuenta que automáticamente cuando un trabajador baje de 50 puntos de disponibilidad se le quitara del grupo que estaba asignado y dejará de tener el bonus salarial.

- Para dar de alta o baja a un trabajador vamos a tener un desplegable en el que le vamos a indicar el estado en el que está el trabajador y dependiendo del estado lo mostrará en la tabla de usuario dados de alta o los que están dados de baja.
- Para generar los informes a una fecha concreta vamos a tener un calendario donde seleccionaremos la fecha de la que queremos el informe. Una vez solicitado el informe se nos mostrará la fecha del informe, el nombre y apellidos ficticios de cada trabajador y la disponibilidad total con la que se quedaron el día solicitado en el informe.