



# Diseño y definición de tablas

## Práctica integradora

### Objetivo

Al igual que hicimos con el armado de wireframes y bocetos antes de comenzar a trabajar con HTML y CSS, siempre es **recomendable** que pensemos en las **necesidades** de datos que va a tener **nuestro sistema** y que diseñemos la base de datos **primero con lápiz y papel** (o su equivalente digital 🧐💡) antes de ponernos a crear tablas.

¡Vamos a ello! 🕶️👍✨

## Consignas

A continuación vamos a plantear **diferentes escenarios** que necesitan de una base de datos para poder **almacenar la información** del sistema. Nuestro **rol** será el de **diseñar una base de datos** junto con sus tablas, campos y relaciones para satisfacer esas necesidades. Es importante interpretar lo que leamos y no tomar todo línea por línea.

Para esta tarea recomendamos utilizar la herramienta de [diagrams.net](https://diagrams.net). En cuyo caso, debemos seleccionar: "Save to device" → "New diagram" → "Entity Relationship Diagram".

Al terminar, recomendamos exportar el resultado como HTML ("File" → "Export as" → "HTML"), de esa manera podremos volver a importarlo a la herramienta si queremos seguir trabajándolo.

Algunas sugerencias de cómo encarar el ejercicio:

- Empezar por determinar las entidades principales que se desprendan de los requisitos, pueden ser usuarios, productos, etc.
- Luego, pensar en qué datos necesitaremos almacenar cada una de ellas.
- Determinar qué información necesitamos separar en tablas secundarias, normalmente son cosas como categorías, temáticas, colores, etc.
- Finalmente, establecer las relaciones entre las tablas que creamos.

Terminado todo este ejercicio, imaginemos la base poblada de información y funcionando para ver si detectamos alguna falla en el diseño que pensamos.

Ya lo dijimos antes... las etapas de planificación son de las más importantes de cualquier proyecto. De hecho, muchos proyectos fallan al salir a producción por no haber hecho bien esta tarea. 🤓👉✨

## Escenario 1 — Playground

Vamos con un viejo conocido. Para entrar en calor, debemos modelar la base de datos que necesitaría una versión simplificada de Playground.



### Micro desafío 1 - Los usuarios y los cursos:

Queremos tener usuarios, los **usuarios** tendrán *nombre, apellido, email, contraseña y categoría*. Los usuarios podrán tener categoría de *estudiantes, docentes, editores o administradores*.

Lo siguiente que queremos es poder almacenar los **cursos**, que tendrán un *título, una descripción, una imagen, una fecha de inicio, una fecha de finalización y un cupo máximo*.

Los cursos tendrán **unidades** (para organizar el contenido) que tendrán un *título, una descripción y una fecha de inicio*.

Los usuarios (de cualquier tipo) podrán estar asociados a cursos.



### Micro desafío 2 - El contenido del curso:

Las unidades contendrán **clases** que también tendrán un *título, una descripción, una fecha de inicio, y una marca de visibilidad* (si el bloque está visible o no).

Las clases contendrán **bloques**. Los bloques tendrán un *título y una marca de visibilidad*. Los bloques podrán ser de diferente tipo: texto, video, presentación, PDF o archivo.

Los bloques también tendrán que poder guardar el *contenido*, sea texto o una URL, en caso de que el tipo sea video, presentación, PDF o archivo.

## Escenario 2 — Bazar Digital

Queremos modelar el sistema de un Bazar. Aquí se pone un poco más difícil porque hay parte de los requerimientos que no son para la base de datos y los podremos resolver en el sistema luego. Así que habrá que determinar cuáles se resuelven en la base. 😊🔥✨



### Micro desafío 1 - Los artículos y los empleados:

En charlas con los dueños entendemos que el bazar va a vender **vajilla, juguetes y artículos de jardín**, y van a tener **empleados** comunes, un coordinador, empleados de mostrador y repositores.

Todos los artículos tendrán *nombre, precio, descripción, stock* y aclararán si son para *uso profesional* o no.

Todos los empleados tendrán *nombre, apellido, dni, sueldo* y un *rol*.



### Micro desafío 2 - El stock y las ventas:

Los empleados de mostrador son los únicos que podrán cobrar y tendrán una **caja** donde guardar lo que cobran, y los repositores podrán ir a la **bodega de almacenamiento**. Para eso esperamos poder saber cuántos productos están exhibidos y/o en el depósito.

Los empleados pueden vender un artículo generando una comisión adicional del 10% del valor del artículo. Para las ventas esperamos poder almacenar *fecha, artículos, medio de pago* y *total de la compra*.

## Conclusión

Como toda habilidad, se requiere de práctica para mejorar. No nos preocupemos si al principio es difícil armar una estructura coherente de base de datos. Es cuestión de ir probando y, con el tiempo, vamos a encontrar las estructuras que mejor funcionan para cada caso que tengamos que resolver.

No olvidemos esta práctica, pues pronto tendremos que diseñar, en equipo, la base de datos del proyecto integrador. 🤪🚀

**¡Hasta la próxima!**