

# Equipo 3: Punto de venta para equipo luminoso



18 NOVIEMBRE

**lumin**

**Creado por:**

Ángel Isidro López Cadena	201655344
Carlos Renato Hernandez Bravo	201641765
Edgar Castro Martínez	201748214
Hugo Jesús Álvarez Fuentes	201639984
Jorge García Adauta	201641468



---

# INDICE

¿Qué es lumin?	3
Descripción	3
Buscando entidades y atributos para el diseño de la base de datos	3
Entidad <i>users</i>	4
Entidad <i>inventory</i>	4
Entidad <i>sales</i>	4
Entidad <i>sales_inventory</i>	5
Estableciendo tecnologías de implementación de la base de datos.	5
Relación de entidades	5
Diagrama de Entidad Relación	6

---

# ¿Qué es lumin?

## Descripción

lumin, es una plataforma web diseñada para facilitar el control sobre el registro y venta de equipo luminario en cualquier negocio o empresa afín a la venta de este tipo de equipo.

lumin es un sitio web diseñado para ejercer funciones de un punto de venta, como es el control sobre el registro de inventario de equipo luminario mediante una base de dato; así como, ofrecer reportes de ventas realizadas sobre inventario, Registro de productos por categoría, visualización del catálogo de productos, alta y baja de productos y vendedores. Por último, el sistema es capaz de ser visualizado en distintos dispositivos de manera eficiente.

## Buscando entidades y atributos para el diseño de la base de datos

El propósito de lumin es gestionar y administrar de forma eficiente la venta de equipo luminario de cualquier negocio que se dedique a la venta de dichos enseres. Una de las ventajas de Lumin es que no cualquier miembro del personal de la tienda es capaz de gestionar el inventario existente, es decir, que existirá un miembro administrador capaz de modificar cada producto del inventario, añadir stock, o incluso la eliminación de aquello que ya no sea indispensable dentro de los productos existentes en la tienda. Por otro lado, un miembro vendedor únicamente podrá realizar una venta, además de iniciar sesión.

El miembro administrador también puede crear cuantos usuarios vendedores sea necesario, así como gestionarlos.

Por otro lado, el inventario se conforma de productos, los cuales poseen atributos como precio, nombre, clave de producto, entre otros.

Las ventas impactan en el inventario, y además, se debe reflejar cuál usuario concretó dicha venta, qué productos fueron los que se vendieron y en qué día y hora se reflejó la venta.

Tomando en cuenta lo anterior, se encontraron tres entidades que representan una solución al problema antes planteado.

---

## Entidad *users*

Dicha entidad, como su nombre lo dice, representa los usuarios. Entre los atributos encontrados se encuentran:

- ID\_USER: Un atributo entero y único que representa el número de usuario. Funge como Primary Key.
- USERNAME: Una cadena única que simboliza el nombre del usuario. Funge como Primary Key.
- NAME\_: Una cadena que simboliza al nombre del vendedor o del administrador.
- PASSWRD: La contraseña del usuario con la que este podrá acceder al sistema.
- ISADMIN: Un entero de un dígito con valor de 0 a 1 que dará a entender al sistema si este se trata del usuario administrador o un vendedor.

## Entidad *inventory*

Representa al inventario de la tienda y los productos que este contiene.

- ID\_PRODUCT: Valor entero único que representa a la clave de un producto único. Se trata de una Primary Key,
- NAME\_PRODUCT: Atributo de tipo cadena en donde se establecerá el nombre del producto.
- PRICE: Atributo de punto decimal para el precio del producto.
- CANT\_STOCK: Entero que representa la cantidad existente de piezas de un producto en específico dentro del inventario.
- CATEGORY: Cadena que representa la categoría a la que pertenece un producto.
- DESCRIPTION: Se trata de un atributo que describe las características de un producto en especial.
- ISAVAILABLE: Entero de un dígito con valor de 0 a 1 que representa si un producto se encuentra disponible o no en el catálogo.

## Entidad *sales*

Como se mencionó anteriormente, es necesario saber qué usuario generó una venta en específico. Además, una venta puede contener uno o varios productos diferentes dentro de ella y varios productos pueden pertenecer a una venta. Al no poder incluir de forma estática cuantos productos se venderán, se generó una entidad extra en la que se pueda romper la relación de muchos a muchos. En primera instancia, los atributos contemplados dentro de esta entidad fueron los siguientes:

- ID\_SALE: Entero que representa a la clave de la venta. Se trata de una Primary Key.
- ID\_USER: Atributo relacional (Foreign Key) que apunta quién realizó la venta.
- DATE\_SOLD: Fecha y hora en la que se generó la venta.

#### **Entidad *sales\_inventory***

Como se mencionó anteriormente, una venta puede contener varios productos y varios productos pueden pertenecer a una venta. Esta entidad ayuda a poder mapear qué productos se vendieron dentro de una sola venta generada por un usuario (la clave de la venta y del inventario forman una llave primaria para evitar la repetición), además, aquí se contempla cuantas unidades de un producto se vendieron en específico.

- ID\_SALE: Llave primaria y foránea que ayuda a mapear a qué venta pertenece un producto vendido.
- ID\_PRODUCT: Llave primaria y secundaria que ayuda a mapear el producto vendido en el inventario.
- CANT\_ITEMS: Entero que representa cuántas unidades de un producto se vendieron.

## **Estableciendo tecnologías de implementación de la base de datos.**

Para la creación de la base de datos se decidió trabajar con MySQL Workbench, pensando en que la interfaz en dicho software es más amigable e intuitivo que la existente en el entorno AMP de XAMPP. Posteriormente, la base de datos se exporta a XAMPP para su uso.

## **Relación de entidades**

La relación de entidades se ve de dicha forma:

- Un usuario puede generar una o varias ventas; varias ventas pertenecen a un usuario.
- Varias ventas pueden contener varios productos. De forma inversa, muchos productos pueden pertenecer a varias ventas, por lo que fue necesario crear la entidad intercesora *sales\_inventory* para poder romper esta relación de muchos a muchos y así tener más orden en la base de datos.

# Diagrama de Entidad Relación

Después de normalizar las entidades, el presente diagrama representa a la base de datos de lumin.

