

**Dpto. INFORMÁTICA - I.E.S. DOMINGO PÉREZ MINIK**

**MÓDULO PROYECTO**

**C.F.G.S. Administración de. Sistemas Informáticos y en Red**

**SERVIDOR Y SERVICIOS DE APLICACIÓN WEB**



**Autores :** Alexander Garcia Perez  
Sergio Diaz Toledo

**Fecha :** 25/05/2023

**Tutor :** Angel Perez Cabeza

<b>1. INTRODUCCIÓN.....</b>	<b>2</b>
1.1 EXPLICACIÓN Y FUNCIONALIDAD.....	2
1.2 OBJETIVO.....	2
<b>2. SERVIDOR.....</b>	<b>3</b>
2.1 CREACIÓN Y ELECCIÓN DEL SERVIDOR.....	3
2.2 RED.....	3
2.2.1 SWITCH.....	4
<b>3. SERVICIOS.....</b>	<b>5</b>
3.1 DNS.....	5
3.2 SNMP.....	6
3.3 CORREO.....	6
3.4 BACKUP.....	7
3.5 MONITORIZACIÓN.....	8
3.6 FIREWALL.....	9
<b>4. BASE DE DATOS.....</b>	<b>11</b>
4.1 SISTEMA GESTOR Y VERSIÓN.....	11
4.1.1 MySQL - MariaDB*.....	11
4.1.2 MySQL Workbench.....	11
4.1.2 Gestión de tiempo.....	11
4.2 ESTRUCTURA.....	12
4.2.1 Tablas.....	12
4.2.2 Desglose de Tiempo.....	15
4.3 DISPARADORES (TRIGGERS).....	16
4.3.1 Actualizar Precio.....	16
4.3.2 Guardar las Órdenes Canceladas.....	17
4.3.3 Actualizar Ventas.....	17
4.3.4 Desglose de Tiempo.....	18
<b>5. APLICACIÓN WEB.....</b>	<b>19</b>
5.1 INTERFAZ Y FUNCIONALIDAD.....	19
5.1.2 Crear Órdenes.....	20
5.1.3 Visionado de detalles.....	21
5.1 Desglose de tiempo.....	23
<b>6. RECURSOS HUMANOS.....</b>	<b>24</b>
<b>7. RECURSOS MATERIALES.....</b>	<b>25</b>
<b>8. CRONOGRAMA.....</b>	<b>26</b>
<b>9. PRESUPUESTO.....</b>	<b>27</b>
<b>10. POLÍTICA DE SEGUIMIENTO Y EVALUACIÓN.....</b>	<b>28</b>
<b>11. ANEXO.....</b>	<b>29</b>
11.1 GESTOR DE DATOS.....	29
11.2 EXTENSIÓN PDO.....	29
<b>12. BIBLIOGRAFÍA.....</b>	<b>30</b>

# 1. INTRODUCCIÓN

---

## 1.1 EXPLICACIÓN Y FUNCIONALIDAD

Se trata de la instalación y configuración de un servidor, el cual, alojara diferentes aplicaciones o funciones que realizarán las siguientes acciones:

- Servicio de correo
- DNS
- Backups
- Monitorización
- Servicio Web para la aplicación

Implantación de una aplicación web con conexión a base de datos usando tanto HTML como PHP que haga sencillo para el usuario la inserción, visualización y modificación de los datos. Configuración para un switch.

## 1.2 OBJETIVO

El objetivo del servidor será implementar de forma exitosa una base de datos, un crud y permita a los usuarios hacer inserciones en la base de datos, teniendo en cuenta que además se instalarán los servicios necesarios para que los usuarios puedan utilizar este de forma cómoda y sencilla permitiéndoles trabajar solo con el mismo sin necesidad de utilizar otros equipos.

La aplicación web se conectará a una base de datos con diferentes tablas de utilidad para una empresa y funciones tales como crear órdenes para diversos productos, inserción y modificación de empleados, clientes, y productos. Estas tablas para el usuario se verán de manera sencilla ya que las consultas llevadas a cabo en la aplicación unen las tablas de forma que el trabajador final ve los datos necesarios.

## 2. SERVIDOR

---

### 2.1 CREACIÓN Y ELECCIÓN DEL SERVIDOR

Se decide usar un servidor Linux debido a la seguridad del mismo, comparado con los servidores de Windows, además de que al ser de código abierto, tiene feedback de los usuarios además de los programadores de Linux.

Instalación	Duración	Fecha inicio	Fecha Final	Responsable
Configuración base del servidor	2 horas	18/04/2023	18/04/2023	Sergio

### 2.2 RED

Se utilizará una red interna 192.168.0.0/24. Se podrá controlar mediante anydesk, aunque debido a que últimamente han habido bastantes fallos de luz en la zona norte, hemos decidido no utilizar esto en el proyecto final para enseñarlo, en cambio utilizaremos el equipo de Alexander para enseñar el funcionamiento.

Instalación	Duración	Fecha inicio	Fecha Final	Responsable
Configuración de redes	15 minutos	26/04/2023	26/04/2023	Sergio

### 2.2.1 SWITCH

Se simulará la creación e implementación de un switch, el cual estará totalmente configurado y funciona de forma real, teniendo en cuenta de que no se llega a implementar una máquina de tipo switch debido a que el modelo utilizado en el proyecto es un modelo real y el precio oscila entre 250€ y 2000€.

La configuración del switch se podrá encontrar en la documentación que se entrega en el proyecto.

Instalación	Duración	Fecha inicio	Fecha Final	Responsable
Documentación sobre el switch	2 horas	22/04/2023	24/04/2023	Sergio
Configuración del switch	3 horas 30 minutos	24/04/2023	26/04/2023	Sergio
Repaso del switch	30 minutos	10/05/2023	10/05/2023	Sergio

### 3. SERVICIOS

#### 3.1 DNS

Se configurará el servicio DNS para la posterior conexión al servidor, servicio de correo y monitorización. La dirección de DNS será fastconf.net. Se creará la configuración para la zona de búsqueda directa e inversa en el fichero /etc/bind/ con el nombre db.fastconf.net y db.0.168.192.

```
GNU nano 4.8 db.fastconf.net
$TTL 1D
@ IN SOA ns.fastconf.net. root.fastconf.net. (
    1 ; serial
    604800 ; refresh
    86400 ; retry
    2419200 ; expiration
    604800 ; TTL negative cache
)
; Registros NS (Servidores de nombres)
@ IN NS ns.fastconf.net.

; Registros A
ns.fastconf.net. IN A 192.168.0.140
fastconf.net. IN A 192.168.0.140
ns1.fastconf.net. IN A 192.168.0.140
mail.fastconf.net. IN A 192.168.0.140
```

```
GNU nano 4.8 db.0.168.192
$TTL 1D
@ IN SOA ns.fastconf.net. root.fastconf.net. (
    0 ; serial
    604800 ; refresh
    86400 ; retry
    2419200 ; expires
    604800 ; TTL negative cache
)
@ IN NS ns.fastconf.net.
; Registros PTR
140 IN PTR ns.fastconf.net.
1 IN PTR router.fastconf.net.
108 IN PTR ubuntu2.localnet.net.
112 IN PTR nox.localnet.net.
```

Instalación	Duración	Fecha inicio	Fecha Final	Responsable
Configuración de DNS	3 horas	02/05/2023	03/05/2023	Sergio

### 3.2 SNMP

Se configurará un servicio SNMP para el monitorizado del servidor. Este permitirá conectarse desde una aplicación. Este tendrá una comunidad llamada fastconf\_Monitor y será de solo lectura. Cualquier persona que conozca la comunidad será capaz de monitorizar los datos de esta.

```
# arguments: community [default|hostname|network/bits]

# Read-only access to everyone to the systemonly view
rocommunity public default -V systemonly
rocommunity6 public default -V systemonly
rocommunity fastconf_Monitor 192.168.0.2/32
rocommunity fastconf_Monitor 192.168.0.40/32

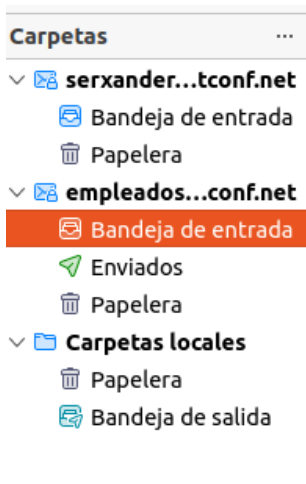
# SNMPv3 doesn't use communities, but users with (optional
```

Instalación	Duración	Fecha inicio	Fecha Final	Responsable
Configuración de SNMP	30 minutos	02/05/2023	02/05/2023	Sergio

### 3.3 CORREO

Se configurará un servicio de correo POP3 para que los usuarios puedan comunicarse en la red interna de forma que no haga falta el uso de apps externas como Whatsapp o Teams. Esto también permitirá ahorrar ancho de banda.

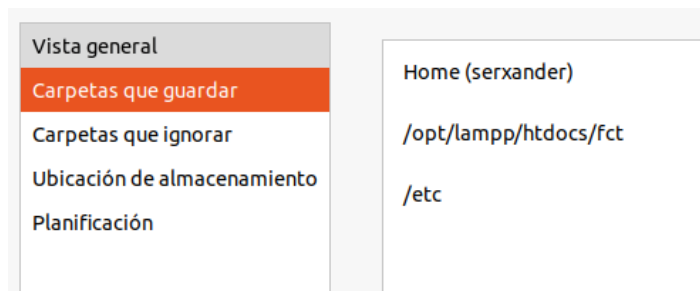
Como software para el uso del cliente usaremos Thunderbird con una cuenta de admin y una de empleado configurada como ejemplo.



Instalación	Duración	Fecha inicio	Fecha Final	Responsable
Configuración de mail	1 hora	13/05/2023	14/05/2023	Sergio

### 3.4 BACKUP

Se configurará un servicio de backup simulando una copia de seguridad a una RAID en un sistema externo, utilizando Deja-Cup. Guardaremos la carpeta HOME del administrador, la carpeta en la cual se guardará la página web y la carpeta etc en la que se guarda toda la información importante de los servicios del servidor, por lo que en caso de que caiga por cualquier ataque, pueda ser levantado otra vez en cuestión de minutos.



Todas las copias se irán copiando de forma que será una copia completa que durará 6 meses antes de ser eliminada.

Instalación	Duración	Fecha inicio	Fecha Final	Responsable
Configuración del backup	30 minutos	20/05/2023	20/05/2023	Sergio



### 3.5 MONITORIZACIÓN

Se configurará en el script de arranque de xampp una línea de código que enseñe los recursos del servidor, además de que estos recursos podrán ser medidos mediante SNMP y serán enviados a la herramienta de monitorización.

Para la monitorización utilizaremos Monit, una herramienta de monitorización que nos permite monitorizar todo tipo de protocolos, procesos, carpetas u otros equipos.

```
root@fastconf: /home/serxander/Escritorio
GNU nano 4.8 /etc/monit/monitrc
#   if status != 0 then alert
#
#
#####
## Includes
#####
##
## It is possible to include additional configuration parts from other files or
## directories.
#
include /etc/monit/conf.d/*
include /etc/monit/conf-enabled/*

# ping
check program ping_testeo with path "/home/serxander/scripts_monit/check_ping.sh"
if status !=0 then alert
```

Instalación	Duración	Fecha inicio	Fecha Final	Responsable
Configuración de script de monitorización	20 minutos	13/05/2023	13/05/2023	Sergio
Configuración de monit	20 minutos	20/05/2023	20/05/2023	Sergio
Configuración de reglas	10 minutos	20/05/2023	20/05/2023	Sergio

### 3.6 FIREWALL

Se configurará el firewall por defecto ufw, el cual es un servicio que viene preinstalado en Ubuntu. Configuraremos que las reglas vayan por IPV6 también y crearemos una serie de reglas determinadas para permitir ssh, entradas por el puerto 22, 2222, permitir http y https, denegar la entrada desde 6000 a 6007 y permitir conexiones desde direcciones específicas.

```
serxander@fastconf:~/Escritorio$ sudo ufw allow ssh
Reglas actualizadas
Reglas actualizadas (v6)
serxander@fastconf:~/Escritorio$ sudo ufw allow 22
Reglas actualizadas
Reglas actualizadas (v6)
serxander@fastconf:~/Escritorio$ sudo ufw allow 2222
Reglas actualizadas
Reglas actualizadas (v6)
serxander@fastconf:~/Escritorio$ sudo ufw enable
El cortafuegos está activo y habilitado en el arranque del sistema
serxander@fastconf:~/Escritorio$ sudo ufw allow http
Regla añadida
Regla añadida (v6)
serxander@fastconf:~/Escritorio$ sudo ufw allow https
Regla añadida
Regla añadida (v6)
serxander@fastconf:~/Escritorio$ sudo ufw allow 80
Regla añadida
Regla añadida (v6)
serxander@fastconf:~/Escritorio$ sudo ufw allow 443
Regla añadida
Regla añadida (v6)
```

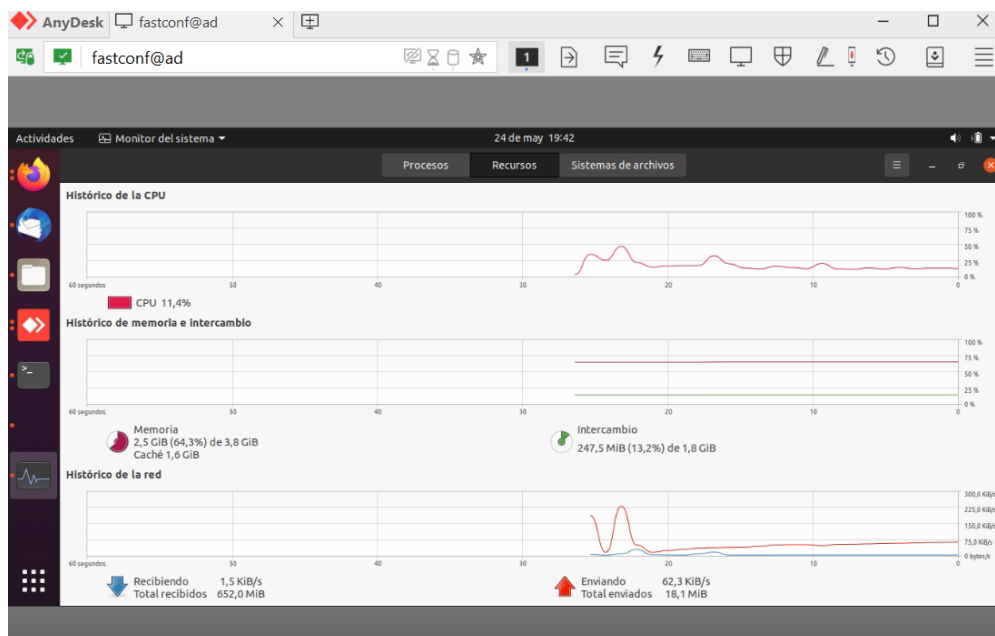
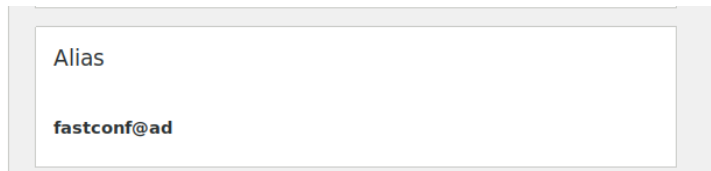
```
serxander@fastconf:~/Escritorio$ sudo ufw deny 6000:6007/tcp
Regla añadida
Regla añadida (v6)
serxander@fastconf:~/Escritorio$ sudo ufw deny 6000:6007/udp
Regla añadida
Regla añadida (v6)
serxander@fastconf:~/Escritorio$ sudo ufw allow from 192.168.0.2
Regla añadida
```

Instalación	Duración	Fecha inicio	Fecha Final	Responsable
Configuración de firewall	1 hora	15/05/2023	15/05/2023	Sergio

### 3.7 CONEXIÓN REMOTA

Se instalará y se configurará un servicio de conexión remota para que los técnicos puedan conectarse desde fuera del servidor. Se usará AnyDesk para la conexión con el servidor debido a que es gratuito y es un software bastante potente y sencillo de usar para los usuarios.

Conseguiremos poner un alias a AnyDesk para poder conectarnos al equipo mediante el alias.



Instalación	Duración	Fecha inicio	Fecha Final	Responsable
AnyDesk	25 min	24/05/2023	24/05/2023	Sergio

## 4. BASE DE DATOS

---

### 4.1 SISTEMA GESTOR Y VERSIÓN

Se instalará un sistema gestor de base de datos y una herramienta gráfica para su manipulación. ([versiones](#))

#### 4.1.1 MySQL - [MariaDB\\*](#)

La razón de esta elección se debe a la compatibilidad con la mayoría de aplicaciones desarrolladas para MySQL gracias a esto la base de datos se podría migrar sin realizar grandes cambios.

Posee una comunidad activa de desarrolladores que comparten conocimiento y solucionan diversos problemas que comienzan a aparecer.

#### 4.1.2 MySQL Workbench

Es una herramienta gráfica que nos permite administrar bases de datos tanto en nuestro equipo como en equipos externos, ya que nos permite establecer conexiones con otros sistemas. En mi caso, elegí esta herramienta por su capacidad de generar modelos físicos y lógicos a partir de un diseño conceptual, lo que facilita la implementación real de la base de datos.

Además, ofrece una interfaz visualmente agradable y sencilla para expresar consultas.

También nos permite monitorear el rendimiento de la base de datos y analizar las consultas y tiempos de respuesta.

#### 4.1.2 Gestión de tiempo

Instalación	Duración	Fecha inicio	Fecha Final	Responsable
MySQL	30 min	25/04/2023	25/04/2023	Alexander
Workbench	30 min	25/04/2023	25/04/2023	Alexander

## 4.2 ESTRUCTURA

La creación de esta base de datos ha sido hecha desde cero buscando las necesidades específicas para nuestro proyecto, se han realizado nueve tablas las cuales la mayoría están conectadas entre sí y poseen diversos disparadores que nos permiten automatizar distintos procesos, como por ejemplo la copia de datos después de una eliminación o la actualización de otros después de un 'update'.

A continuación se explicará la función de cada tabla así como la consulta necesaria para la inserción de cada una.

### 4.2.1 Tablas

#### Tabla Empleados

La tabla "empleados" almacena información relevante sobre los empleados de la organización, como su código, nombre, correo electrónico, bono, fecha de contratación, número de teléfono y el código del departamento al que pertenecen.

```
CREATE TABLE IF NOT EXISTS empleados (  
  Cod_Emp INT AUTO_INCREMENT PRIMARY KEY,  
  Nombre VARCHAR(100),  
  Correo VARCHAR(100),  
  Bonus DECIMAL(10,2),  
  fecha_contrato DATE,  
  telefono VARCHAR(10),  
  Cod_Dep INT,  
  FOREIGN KEY (Cod_Dep) REFERENCES departamentos (Cod_Dep)) ;
```

#### Tabla Clientes

La tabla "clientes" almacena información básica de los clientes, como su nombre, número de identificación, dirección de correo electrónico y número de teléfono. La clave primaria en la columna "DNI" garantiza la unicidad de cada cliente en la tabla.

```
CREATE TABLE IF NOT EXISTS clientes (  
  Nombre VARCHAR(100),  
  DNI VARCHAR(20) PRIMARY KEY,  
  Correo VARCHAR(100),  
  Tel VARCHAR(20)  
);
```

### **Tabla Productos**

La tabla productos almacena información relevante sobre los productos de la organización. Cada producto tiene un código único asignado automáticamente, un nombre descriptivo y un precio asociado. La clave primaria en la columna Cod\_Pro garantiza la unicidad de cada producto en la tabla.

```
CREATE TABLE IF NOT EXISTS productos (  
  Cod_Pro INT AUTO_INCREMENT PRIMARY KEY,  
  Nombre VARCHAR(100),  
  Precio DECIMAL(10,2),  
);
```

### **Tabla Departamentos**

La tabla departamentos almacena información relevante sobre los diferentes departamentos de la organización. Cada departamento tiene un código único asignado automáticamente, un nombre descriptivo y un salario asociado. La clave primaria en la columna Cod\_Dep garantiza la unicidad de cada departamento en la tabla

```
CREATE TABLE IF NOT EXISTS departamentos (  
  Cod_Dep INT AUTO_INCREMENTAL PRIMARY KEY,  
  Nombre VARCHAR(100),  
  Salario DECIMAL(10,2)  
);
```

### **Tabla Identificador**

La tabla identificador permite almacenar los números de serie de los productos, relacionándolos con sus respectivos códigos de producto y garantizando la unicidad de los registros.

```
CREATE TABLE IF NOT EXISTS identificador (  
  Cod_Pro INT,  
  num_serie VARCHAR(50) PRIMARY KEY,  
  FOREIGN KEY (Cod_Pro) REFERENCES productos (Cod_Pro),  
  CONSTRAINT unique_licencia_cod_pro UNIQUE (num_serie, Cod_Pro);  
);
```

### Tabla Ventas

La tabla ventas registra las ventas de productos, almacenando información como el código del producto, el número de serie, el ID de la orden y la fecha de la venta.

```
CREATE TABLE ventas (  
  Cod_Pro INT,  
  num_serie VARCHAR(50),  
  id_orden VARCHAR(20),  
  fecha DATE,  
  FOREIGN KEY (Cod_Pro) REFERENCES productos (Cod_Pro),  
  FOREIGN KEY (num_serie) REFERENCES identificador (num_serie),  
  FOREIGN KEY (id_orden) REFERENCES ordenes (id_orden)  
);
```

### Tabla Órdenes Canceladas

La tabla ordenes\_canceladas registra las órdenes que han sido canceladas, almacenando información como el ID de la orden, el motivo de la cancelación, la fecha de cancelación y el precio asociado. Esta tabla permite tener un registro de las órdenes canceladas y los detalles relacionados con ellas.

```
CREATE TABLE ordenes_canceladas (  
  id_orden VARCHAR(30),  
  motivo VARCHAR(255),  
  fecha_cancelacion DATE,  
  precio DECIMAL(10,2)  
  PRIMARY KEY (id_orden)  
);
```

### Tabla Órdenes

La tabla ordenes almacena información sobre las órdenes realizadas, incluyendo su ID, el empleado y cliente asociados, la fecha y el precio total. Las claves externas aseguran la consistencia de los datos y permiten establecer relaciones entre las órdenes, empleados y clientes.

```
CREATE TABLE IF NOT EXISTS ordenes (  
  id_orden VARCHAR(20) PRIMARY KEY,  
  Cod_Emp INT,  
  Cod_Cli VARCHAR(20),  
  fecha DATE,  
  precio DECIMAL(10,2),  
  FOREIGN KEY (Cod_Cli) REFERENCES clientes (DNI),  
  FOREIGN KEY (Cod_Emp) REFERENCES empleados (Cod_Emp)  
);
```

### Tabla Detalles\_Ordenes

La tabla detalles\_ordenes almacena información detallada sobre los productos incluidos en cada orden, incluyendo el ID de la orden, el código del producto, la cantidad de unidades y el precio unitario. La clave primaria compuesta asegura la unicidad de cada detalle de orden, y la clave externa garantiza la integridad referencial con la tabla productos.

```
CREATE TABLE IF NOT EXISTS detalles_ordenes (  
  id_orden VARCHAR(20),  
  Cod_Pro INT,  
  cantidad INT,  
  precio_unitario DECIMAL(10,2),  
  PRIMARY KEY (id_orden, Cod_Pro),  
  FOREIGN KEY (Cod_Pro) REFERENCES productos (Cod_Pro)  
);
```

#### 4.2.2 Desglose de Tiempo

Tablas	Duración	Fecha inicio	Fecha final	Responsable
Normalización de Base de Datos	7 horas	01/05/2023	03/05/2023	Alexander
Creación e Inserción de Datos	5 horas	04/05/2023	07/05/2023	Alexander



### 4.3 DISPARADORES (TRIGGERS)

Se han hecho cinco triggers con distintas automatizaciones para que se actualicen o se inserten datos en las tablas de forma correcta

#### 4.3.1 Actualizar Precio

Actualiza el precio de la orden cuando se añaden productos nuevos

```
DELIMITER &&
CREATE TRIGGER actualizar_precio_insert AFTER INSERT ON detalles_ordenes
FOR EACH ROW
BEGIN
    UPDATE ordenes SET precio = (
        SELECT SUM(precio_unitario * cantidad)
        FROM detalles_ordenes
        WHERE id_orden = NEW.id_orden
    )
    WHERE id_orden = NEW.id_orden;
END&&
```

Este disparador inserta el precio de una orden multiplicando la cantidad por el precio unitario del producto en cada artículo de una misma orden, luego suma los resultados y nos da el precio total de la orden.

Actualiza el precio de la orden cuando se modifica la tabla detalles\_ordenes

```
DELIMITER &&
CREATE TRIGGER actualizar_precio_update AFTER UPDATE ON detalles_ordenes
FOR EACH ROW
BEGIN
    UPDATE ordenes SET precio = (
        SELECT SUM(precio_unitario * cantidad)
        FROM detalles_ordenes
        WHERE id_orden = NEW.id_orden
    )
    WHERE id_orden = NEW.id_orden;
END &&
DELIMITER ;
```

Este trigger realiza lo mismo que el anterior pero después de un 'update' así los precios de las órdenes están siempre actualizados, aunque los cambios en otra tabla.

### 4.3.2 Guardar las Órdenes Canceladas

Elimina e inserta en otra tabla las órdenes canceladas

```
DELIMITER //

CREATE TRIGGER eliminar_orden
AFTER DELETE ON ordenes
FOR EACH ROW
BEGIN
    -- Elimina en la tabla detalles_ordenes
    DELETE FROM detalles_ordenes
    WHERE id_orden = OLD.id_orden;

    -- Insertar en la tabla ordenes_canceladas
    INSERT INTO ordenes_canceladas (id_orden, motivo, fecha_cancelacion, precio)
    VALUES (OLD.id_orden, 'Orden cancelada', CURDATE(), OLD.precio);
END //

DELIMITER ;
```

Cuando borramos una orden este trigger se encarga de guardar sus datos en la tabla ordenes\_canceladas los datos relacionados en cuestión.

### 4.3.3 Actualizar Ventas

Inserta en la tabla ventas los productos vendidos al hacer una orden

```
DELIMITER //

CREATE TRIGGER insertar_ventas AFTER INSERT ON detalles_ordenes
FOR EACH ROW
BEGIN
    DECLARE cantidad INT;
    SET cantidad = NEW.cantidad;

    WHILE cantidad > 0 DO
        INSERT INTO ventas (Cod_Pro, num_serie, id_orden, fecha)
        SELECT d.Cod_Pro, i.num_serie, NEW.id_orden, CURRENT_DATE()
        FROM detalles_ordenes d
        JOIN identificador i ON d.Cod_Pro = i.Cod_Pro
        WHERE d.id_orden = NEW.id_orden
        LIMIT 1;

        SET cantidad = cantidad - 1;
    END WHILE;
END //

DELIMITER ;
```

Al completar una orden, se extraen productos con su número de serie de la tabla identificador y se añaden a la tabla ventas con el id de orden asignado

#### 4.3.4 Desglose de Tiempo

	Duración	Fecha inicio	Fecha final	Responsable
Disparadores	2 horas	8/05/2023	8/05/2023	Alexander

## 5. APLICACIÓN WEB

La aplicación se basa en la inserción, vista, modificación y eliminación de datos (CRUD), con esto buscamos crear una interfaz sencilla para el usuario que haga todas las acciones de forma automática.

Para el desarrollo de esta tarea se ha usado PHP junto con HTML para la conexión de la aplicación con la base de datos, además de esto, se ha aplicado [la extensión PDO\\*](#) la cual proporciona una interfaz orientada a objetos.

Se centra en poder registrar los productos que se venden, así como crear ordenes con los datos necesarios como el empleado que la realiza, el cliente que la compra, los productos que se venden y el precio conjunto de ellos.

A continuación se mostrará el funcionamiento de la aplicación.

### 5.1 INTERFAZ Y FUNCIONALIDAD

Al acceder se mostrará un menú el cual nos permite seleccionar la acción que queramos llevar a cabo.



El estilo para todas las tablas es el mismo, para facilitar el visionado de los datos se han unido tablas gracias a las claves foráneas.

### 5.1.2 Crear Órdenes

Se ha creado un formulario para la inserción de órdenes. Este formulario genera automáticamente un ID de orden mediante JavaScript, utilizando 10 números aleatorios. Además, se ha añadido un botón que permite agregar más productos a una misma orden, ya que se ha considerado la posibilidad de que el cliente pueda comprar múltiples productos en cantidades variables. Al realizar la inserción, los disparadores previamente configurados se encargan de sumar y calcular los precios correspondientes según la cantidad de productos incluidos en la orden.

Los datos necesarios para la inserción se obtienen de la base de datos, lo que nos permite seleccionar el producto, el empleado (solo del departamento de ventas) y el cliente asociado a la orden.

El formulario se ve de la siguiente manera y añadiremos una orden para ver el funcionamiento de las automatizaciones:

**Nueva Orden**

Agregar Producto

Id de Orden: ORD8961922649

Empleado: Juan Pérez

Cliente: Manuel Evaris Gomez

Producto: Sketch

Cantidad: 2

Producto: Microsoft Office

Cantidad: 2

Producto: Adobe Photoshop

Cantidad: 1

GUARDAR

Ordenes

La orden se guarda correctamente añadiendo al apartado de precio la suma de la multiplicación entre la cantidad y el precio de la unidad del producto, dando así el precio total de la orden.

Id de Orden	Empleado	Cliente	Fecha	Precio	Opciones	
ORD004	Laura Hernández	98765432E	2023-05-04	198.00	<a href="#">Ver Detalles</a>	<a href="#">Cancelar Orden</a>
ORD3522211056	María González	87654321B	2023-05-16	1092.00	<a href="#">Ver Detalles</a>	<a href="#">Cancelar Orden</a>
ORD8961922649	Juan Pérez	56673455R	2023-05-25	695.00	<a href="#">Ver Detalles</a>	<a href="#">Cancelar Orden</a>
ORD9227542136	Laura Hernández	98765432E	2023-05-16	1453.00	<a href="#">Ver Detalles</a>	<a href="#">Cancelar Orden</a>

En este apartado de la aplicación tenemos la posibilidad de cancelar la orden si así lo deseamos, aun así, se guardarán los datos automáticamente en la tabla de órdenes canceladas. Al cancelar alguna orden se abrirá un campo de texto pidiendo que introducimos el motivo por el cual se ha cancelado, este motivo se añade a la tabla de órdenes canceladas. Cuando la guardamos se coge automáticamente tantos productos como cantidad se haya seleccionado y se pasan de la tabla identificador a la tabla ventas así podemos ver que producto en particular ha comprado cada cliente en cada orden to identificado por el número de orden y el número de serie del producto vendido

### 5.1.3 Visionado de detalles

Se ha buscado la facilidad para ver qué datos contiene cada orden así como la posibilidad de descargar un pdf con su desglose de la orden seleccionada, se ha cogido la cantidad y el precio unitario de la cada producto y se han multiplicado para añadir una columna más con el precio completo, además de un pequeño contenedor con el precio total de la orden completa. La búsqueda se realiza por producto ya que el id de la orden es el mismo. Podemos ver un ejemplo del visionado con la orden anterior :

## Detalles de Ordenes

BUSCAR

Nueva Orden

Inicio

Id de Orden	Empleado	Cliente	Fecha	Producto	Cantidad	Precio Unitario	Precio	Opciones
ORD8961922649	Juan Pérez	56673455R	2023-05-25	Microsoft Office	2	99.00	198.00	<a href="#">Modificar</a> <a href="#">Borrar</a>
ORD8961922649	Juan Pérez	56673455R	2023-05-25	Adobe Photoshop	1	299.00	299.00	<a href="#">Modificar</a> <a href="#">Borrar</a>
ORD8961922649	Juan Pérez	56673455R	2023-05-25	Sketch	2	99.00	198.00	<a href="#">Modificar</a> <a href="#">Borrar</a>

PRECIO TOTAL DE LA ORDEN  
695.00

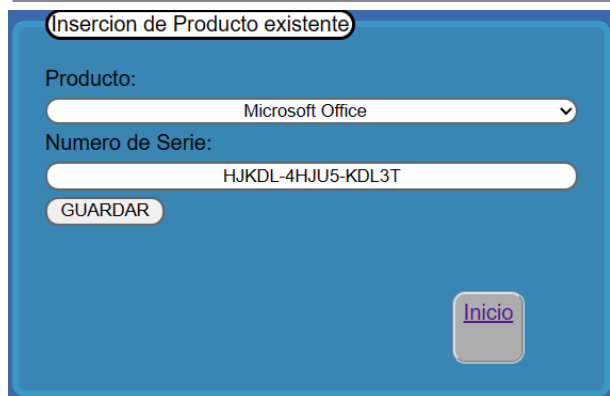
PDF

Al hacer click en pdf lo que se nos mostrará es lo siguiente:

ID	Empleado	Cliente	Fecha	Producto	Cantidad
ORD8961922649	Juan Pérez	56673455R	2023-05-25	Microsoft Office	2
ORD8961922649	Juan Pérez	56673455R	2023-05-25	Adobe Photoshop	1
ORD8961922649	Juan Pérez	56673455R	2023-05-25	Sketch	2

#### 5.1.4 Adición de Productos

Hemos considerado la opción de agregar tanto productos existentes como nuevos productos. De esta manera, establecemos una conexión con la tabla de identificadores donde se almacenan los números de serie de cada producto que añadimos. Estos números se guardan de forma individual, lo que permite una diferenciación única para cada producto. Para añadir productos existentes contamos con un ‘select’ que nos muestra todos los productos de la base de datos y además le podemos añadir el número de serie o identificador.



### 5.1 Desglose de tiempo

Interfaz y funcionalidad	Duración	Fecha inicio	Fecha final	Responsable
Crear Órdenes	5 horas	9/05/2023	10/05/2023	Alexander
Visionar Detalles	4 horas	11/05/2023	11/05/2023	Alexander
Adición de Productos	2 horas	12/05/2023	12/05/2023	Alexander
Visionar demás tablas	6 horas	10/05/2023	15/05/2023	Sergio



## 6. RECURSOS HUMANOS

---

Necesitaremos cuatro especializaciones para llevar a cabo el proyecto :

**Desarrollador web** : Se requiere de alguien capacitado para el desarrollo de la aplicación que sea conocedor tanto de PHP como de HTML y CSS, su objetivo es crear una aplicación con interfaz amigable.

Nombre: Alexander Garcia Perez

**Administrador del Servidor** : Encargado de la configuración y elección del servidor y mantenimiento de red, su objetivo es garantizar el correcto funcionamiento del mismo

Nombre: Sergio Diaz Toledo

**Gestor de Sistemas de Base de Datos** : Encargado de gestionar y administrar la base de datos del proyecto, su objetivo es mantener una base de datos eficiente y realizar consultas o automatizaciones.

Nombre : Alexander Garcia Perez

**Técnico de Redes** : Lleva a cabo la configuración del switch y el router, su objetivo es establecer una red segura y funcional así como asegurar su funcionamiento.

Nombre: Sergio Diaz Toledo

## **7. RECURSOS MATERIALES**

---

Se necesitarán varios materiales divididos en categorías :

### **Equipos Informáticos**

- Ordenadores
- Servidor
- Discos duros

### **Infraestructura de Red**

- Switches
- Router
- Cables Ethernet

### **Software y herramientas**

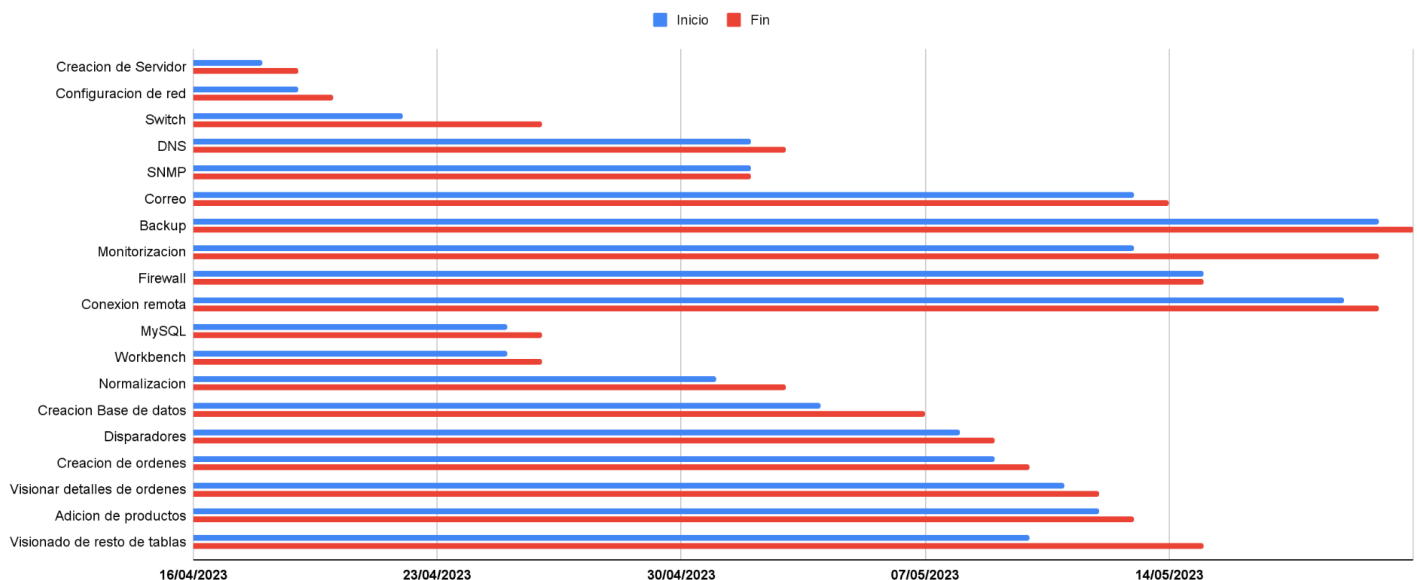
- Sistema Linux
- Editores de texto para el desarrollo de código
- Thunderbird
- Bind9
- Servicio SNMP
- Sistema Gestor de Base de Datos
- Servicio de conexión remota

### **Espacio físico**

- Oficinas o espacios de trabajo adecuados

## 8. CRONOGRAMA

TAREA	ASIGNADO A	PROGRESO	INICIO	FIN
<b>PROYECTO</b>				
Creacion de Servidor	Sergio	100%	18-4-23	19-4-23
Configuracion de red	Sergio	100%	19-4-23	20-4-23
Switch	Sergio	100%	22-4-23	26-4-23
DNS	Sergio	100%	2-5-23	3-5-23
SNMP	Sergio	100%	2-5-23	2-5-23
Correo	Sergio	100%	13-5-23	14-5-23
Backup	Sergio	100%	20-5-23	21-5-23
Monitorizacion	Sergio	100%	13-5-23	20-5-23
Firewall	Sergio	100%	15-5-23	15-5-23
Conexion remota	Sergio	100%	19-5-23	20-5-23
MySQL	Alexander	100%	25-4-23	26-4-23
Workbench	Alexander	100%	25-4-23	26-4-23
Normalizacion	Alexander	100%	30-4-23	5-5-23
Creacion Base de datos	Alexander	100%	4-5-23	7-5-23
Disparadores	Alexander	100%	8-5-23	9-5-23
Creacion de ordenes	Alexander	100%	9-5-23	10-5-23
Visionar detalles de ordenes	Alexander	100%	11-5-23	12-5-23
Adicion de productos	Alexander	100%	12-5-23	13-5-23
Visionado de resto de tablas	Sergio	100%	10-5-23	15-5-23



En el siguiente enlace se encuentra el diagrama completo sin simplificar debido al tamaño del documento.

[https://docs.google.com/spreadsheets/d/1KaX0qvh6GwFvPVIPR7rQGEMEHZDg\\_sLWyHjeFEuxDWE/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1KaX0qvh6GwFvPVIPR7rQGEMEHZDg_sLWyHjeFEuxDWE/edit?usp=sharing)

## 9. PRESUPUESTO

---

Tendremos en cuenta el presupuesto total del software y hardware:

Material	Precio
Switch t2600g-28ts	1.248 €
Licencia AnyDesk(anual)	120 €
Rack	1.200 €
Servidor	600 €
Panel de parcheo	60 €
Cableado	130,50 €
Sistema de discos	1.322,30 €
Pantalla	100 €
SAI	870 €
Router	140 €
<b>Total:</b>	<b>5.791 €</b>

Y además también los gastos en cuanto a recursos humanos

Tarea	Recursos humanos	Precio
Mantenimiento del servidor(anual)	Técnico de sistemas	600 €
Mantenimiento de la red(anual)	Técnico de redes	700 €
Mantenimiento de la app(anual)	Técnico de sistemas	200 €
Mantenimiento de la base de datos(anual)	Gestor de sistemas de base de datos	200 €
Formación en sistemas linux	Técnico de sistemas	200 €
Montaje	Técnico de sistemas/Técnico de redes	200 €

<b>Total:</b>		2100€
---------------	--	-------

## 10. POLÍTICA DE SEGUIMIENTO Y EVALUACIÓN

---

Para el seguimiento correcto de este proyecto ha de revisarse semanalmente el correcto funcionamiento del servidor , así como analizar la configuración del switch para comprobar que todo va como se necesita.

Por otra parte habría que revisar al menos mensualmente la eficacia de la base de datos , debido a que cuantos más datos se encuentren en ella quizás deberíamos añadir índices para agilizar las consultas de datos.

La monitorización siempre estará activa para los errores de red del equipo que aloje el servidor así que por ese lado se podría observar diariamente el funcionamiento del servidor.

## 11. ANEXO

---

### 11.1 GESTOR DE DATOS

La herramienta que gestiona la base de datos será 'MySQL-MariaDB' en su versión 10.4.24 y el software para manejar la base directamente es 'MySQL Workbench' en versión 8.0 .

MariaDB es un fork de MySQL que apareció como alternativa de código abierto creado por los propios desarrolladores de MySQL, todo esto, debido a la adquisición del mismo por Oracle. Aunque se han desarrollado de forma independiente, comparten una estrecha relación y una gran compatibilidad entre sí.

### 11.2 EXTENSIÓN PDO

En PHP, PDO se refiere a una extensión que proporciona una interfaz orientada a objetos para acceder a bases de datos desde aplicaciones PHP. PDO ofrece una capa de abstracción sobre los diferentes sistemas de gestión de bases de datos lo que permite escribir código más portable y fácil de mantener.

Básicamente, PDO actúa como un intermediario entre la aplicación PHP y la base de datos, proporcionando métodos y funciones para conectar, ejecutar consultas, recuperar datos y gestionar transacciones.

## 12. BIBLIOGRAFÍA

---

### Programas y fuentes utilizada

XAMPP : <https://www.apachefriends.org/es/index.html>

Stack Overflow: <https://stackoverflow.com/>

FS : <https://www.fs.com/>

Any Desk : <https://anydesk.com/>

Ultima version PHP : <https://www.php.net/downloads>

DigitalOcean : [www.digitalocean.com](http://www.digitalocean.com)

Todo a desk <https://todoadesk.com>

Qualoom : [www.qualoom.es](http://www.qualoom.es)

RDR : <https://rdr-it.com/>

Tec mint : [www.tecmint.com](http://www.tecmint.com)

Hetman Software : <https://hetmanrecovery.com>