

# **Informe de Seguridad:** **Posible Exposición de Información** **Sensible en un Sitio Web** **Institucional**

Miguel Ángel Roldán de Haro

Granada, a 3 de Marzo de 2025

---

## Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Metodología</b>	<b>3</b>
Google Dorks Utilizados:	4
<b>3. Hallazgos</b>	<b>9</b>
1. Archivos de Bases de Datos SQL	9
2. Archivos de Configuración (PHP, XML, INI)	9
3. Archivos de Texto con Contraseñas y Usuarios	9
<b>4. Archivos de Autenticación de Usuarios</b>	<b>10</b>
Ejemplos específicos:	10
1. Consulta SQL con Posible Exposición de Datos	10
2. Exposición de Dirección IP Predeterminada (en el mismo documento)	11
3. Archivo bbddJARA_InnoDB.sql con Datos Sensibles	12
4. Exposición de Credenciales en Archivos de Configuración	29
5. Archivos de Configuraciones Internas:	33
<b>4. Análisis de Riesgos</b>	<b>40</b>
1. Exposición de Archivos SQL	40
2. Falta de Validación de Entradas y Riesgo de Inyección SQL	41
3. Manejo Inseguro de Cookies	42
4. Creación de Usuarios Administradores sin Verificaciones Adecuadas	43
5. Almacenamiento de Contraseñas en Texto Plano	44
<b>5. Recomendaciones. Formas de Mitigación o Corrección</b>	<b>45</b>
1. Revisión y Eliminación de Archivos Sensibles	46
2. Fortalecer la Autenticación y Autorización	46
3. Uso de Cifrado para Proteger Datos Sensibles	47
4. Mejoras en la Configuración del Servidor Web	47
5. Implementar Monitorización y Auditoría de Seguridad	48
6. Políticas de Seguridad y Capacitación del Personal	48
7. Realizar Pruebas de Penetración Regulares	49
<b>7. Conclusión</b>	<b>49</b>
<b>8. Contacto</b>	<b>50</b>

---

---

## 1. Introducción

El presente informe de seguridad tiene como objetivo documentar de manera detallada los hallazgos relacionados con la posible exposición de información sensible en un sitio web institucional. Cabe destacar que estos descubrimientos fueron fruto de la casualidad y que, en ningún momento, existió la intención deliberada de realizar un análisis de seguridad o de identificar vulnerabilidades en la plataforma.

Durante una búsqueda más exhaustiva y sin un propósito específico de auditoría, se encontraron indicios de posibles riesgos de seguridad, lo que motivó la elaboración de este informe. Para profundizar en la evaluación, se utilizaron herramientas de búsqueda avanzada, conocidas como Google Dorks , las cuales facilitaron la identificación de archivos y configuraciones potencialmente expuestos al público.

A partir de estos hallazgos accidentales, se llevó a cabo un análisis detallado de riesgos y se proponen recomendaciones concretas para mitigar las vulnerabilidades detectadas, con el único objetivo de contribuir a la mejora de la seguridad del sitio web.

## 2. Metodología

El presente informe se basa en hallazgos fortuitos identificados durante una búsqueda rutinaria en Internet, sin que existiera en ningún momento la intención expresa de auditar o evaluar la seguridad del sitio web institucional. Tras detectar indicios de una posible exposición de información sensible, se llevó a cabo un análisis más detallado utilizando técnicas avanzadas de búsqueda con el fin de evaluar el alcance de la situación.

---

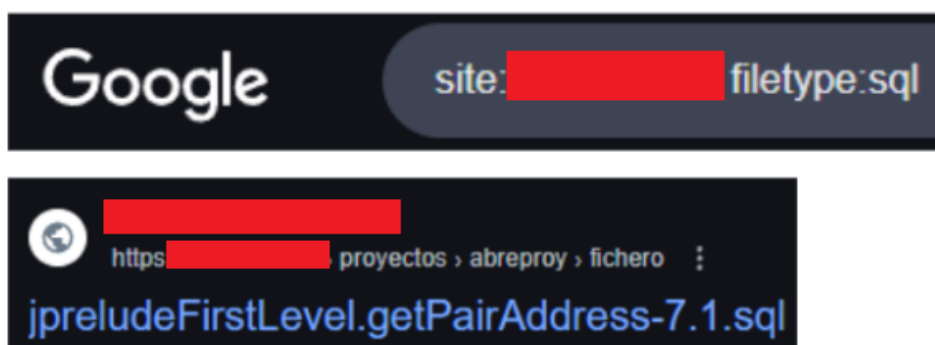
---

Para esta evaluación, se utilizaron consultas avanzadas en motores de búsqueda, comúnmente conocidas como Google Dorks . Esta técnica permite realizar búsquedas más específicas y profundas en los sitios web, facilitando la identificación de archivos potencialmente sensibles o expuestos sin las debidas restricciones de acceso.

Las consultas realizadas fueron las siguientes:

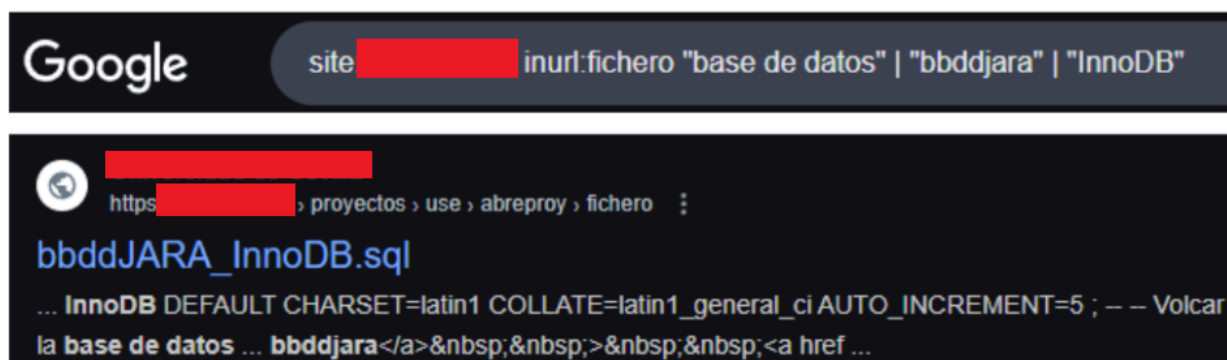
## Google Dorks Utilizados:

**site:portal-institucional.edu filetype:sql**



Objetivo: Identificar archivos de tipo SQL que pudieran contener registros de bases de datos.

**site:portal-institucional.edu inurl:fichero "base de datos" | "bbddjara" | "InnoDB"**



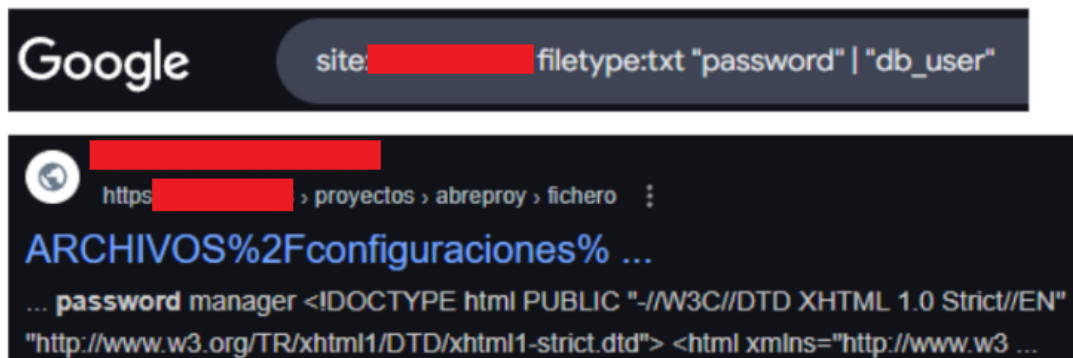
Objetivo: Localizar archivos con referencias a bases de datos, incluyendo el término "InnoDB", asociado comúnmente con sistemas de gestión de bases de datos.

**site:portal-institucional.edu filetype:php | filetype:xml | filetype:ini "db\_password" | "database"**



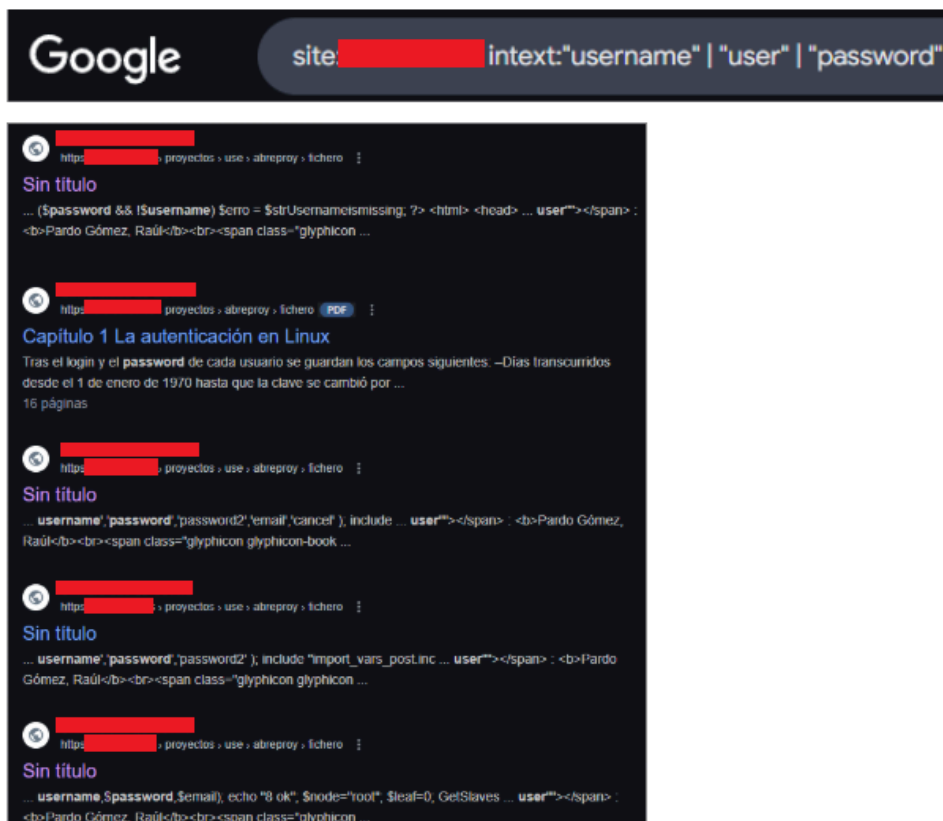
Objetivo: Detectar archivos de configuración (PHP, XML, INI) que pudieran incluir credenciales de acceso a bases de datos.

`site:portal-institucional.edu filetype:txt "password" | "db_user"`



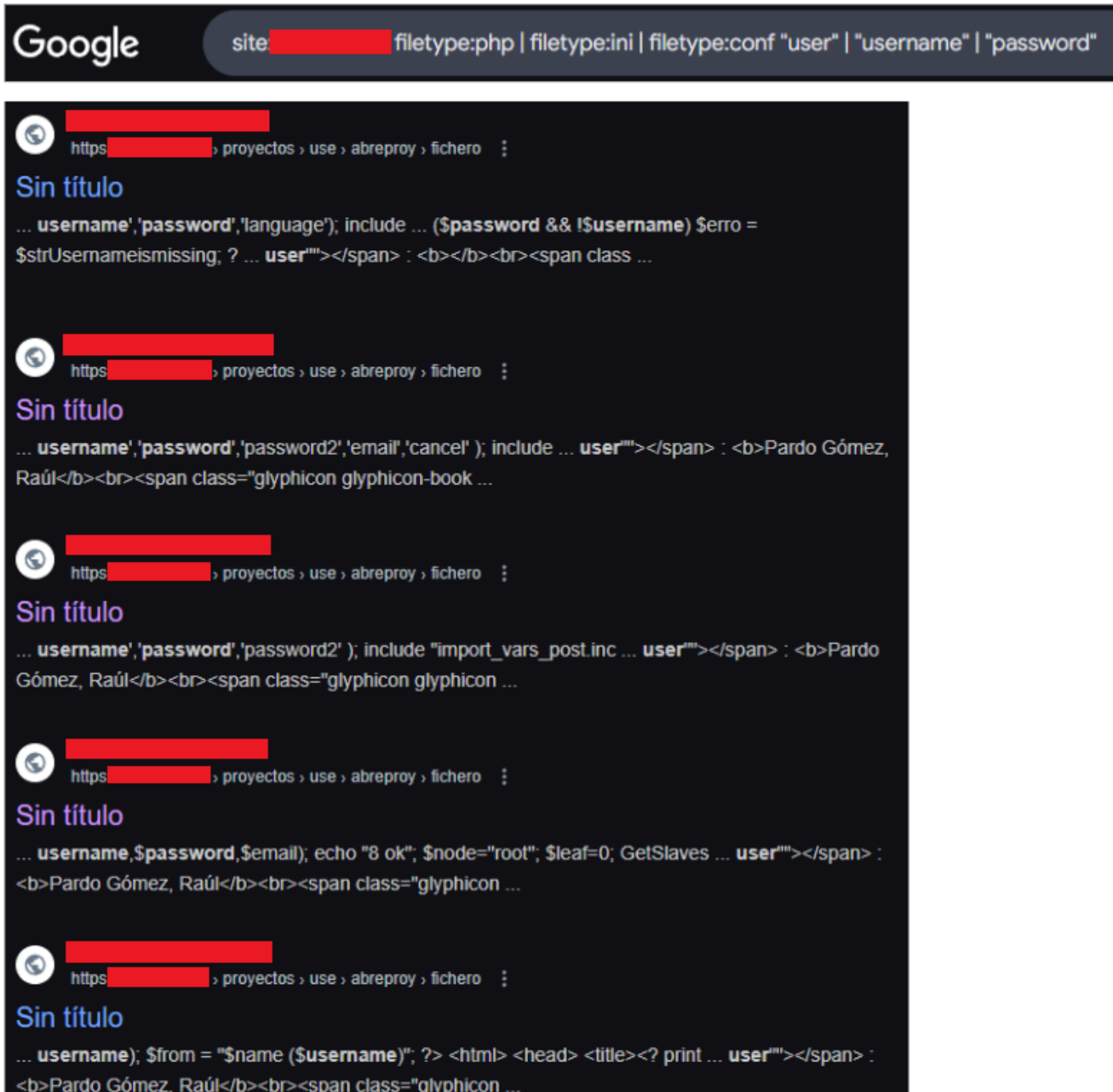
Objetivo: Encontrar archivos de texto que contuvieran posibles contraseñas o nombres de usuario de bases de datos.

`site:portal-institucional.edu intext:"username" | "user" | "password"`



Objetivo: Identificar archivos con términos relacionados con autenticación de usuarios.

`site:portal-institucional.edu filetype:php | filetype:ini | filetype:conf "user" | "username" | "password"`




Objetivo: Buscar archivos de configuración que contengan información relacionada con usuarios y contraseñas.

**site:portal-institucional.edu filetype:sql | filetype:txt | filetype:php "user" | "login" | "username"**

+

Google site: [redacted] filetype:sql | filetype:txt | filetype:php "user" | "login" | "username"


---

 [redacted]  
https://[redacted] › proyectos › use › abreproy › fichero

**Sin título**

... **username** = addslashes(strip\_slashes(\$username)); **username** = addslashes ... **user"**></span> :  
<b></b><br><span class="glyphicon glyphicon-book"></span> ...


---

 [redacted]  
https://[redacted] › proyectos › use › abreproy › fichero

**Sin título**

... **username**, \$password, \$email); echo "8 ok"; \$node="root"; \$leaf=0; GetSlaves ... **user"**></span> :  
<b>Pardo Gómez, Raúl</b><br><span class="glyphicon ...


---

 [redacted]  
https://[redacted] › proyectos › use › abreproy › fichero

**Sin título**

... **username**, 'password', 'password2', 'email', 'cancel' ); include ... **login** Header ("Location: index.php"); }  
// Verifica si las claves coinciden if ...


---

 [redacted]  
https://[redacted] › proyectos › use › abreproy › fichero

**Sin título**

... **username**, 'password', 'password2' ); include "import\_vars\_post.inc ... **login** # De otra forma solicita de  
nuevo otro usuario root if (\$name && \$username ...

---

 [redacted]  
https://[redacted] › proyectos › use › abreproy › fichero

**Sin título**

... **username**); \$from = "\$name (\$username)"; ?> <html> <head> <title><? print ... **user"**></span> :  
<b>Pardo Gómez, Raúl</b><br><span class="glyphicon ...



---

Objetivo: Buscar archivos relacionados con la autenticación de usuarios.

## 3. Hallazgos

Durante la revisión accidental del sitio web institucional, se identificaron indicios de una posible exposición de información sensible. A continuación, se detallan los principales hallazgos encontrados de forma fortuita:

### 1. Archivos de Bases de Datos SQL

Se detectaron archivos potencialmente expuestos que podrían contener registros sensibles, como credenciales de usuarios o información relacionada con bases de datos. Estos archivos estaban accesibles mediante consultas específicas, lo que podría representar un riesgo significativo si no se restringe adecuadamente el acceso a ellos.

### 2. Archivos de Configuración (PHP, XML, INI)

Se encontraron archivos de configuración que incluían posibles credenciales de acceso a bases de datos. Estos archivos suelen contener información crítica, como nombres de usuario y contraseñas, que, en caso de ser expuestos, podrían comprometer la seguridad del sistema.

### 3. Archivos de Texto con Contraseñas y Usuarios

Se identificaron archivos de texto que contenían términos como "password" y "db\_user", lo que podría indicar una exposición inadvertida de información sensible. Estos archivos podrían ser utilizados para acceder a sistemas internos sin la debida autorización.

---

## 4. Archivos de Autenticación de Usuarios

Mediante las búsquedas realizadas, se detectaron archivos con referencias a nombres de usuario y términos como "login". Esta información podría ser utilizada de manera indebida si cae en manos de actores malintencionados.

### Ejemplos específicos:

#### 1. Consulta SQL con Posible Exposición de Datos

site:portal-institucional.edu filetype:sql

```
SELECT CASE WHEN t2.address IS NULL OR t2.address = ' ' THEN '127.0.0.1' ELSE t2.address END,t0.address,MAX(t0.time)
FROM
(
  SELECT t0._message_ident,t3.address,t0.time AT TIME ZONE 'GMT' AS time
  FROM Prelude_CreateTime AS t0,Prelude_Address AS t3
  WHERE (t0.time >= '2006-03-21 12:50:06' AND t0.time <= '2006-03-21 13:30:52') AND t0._parent_type = 'A'
  AND t3._index = 0 AND t3._parent_type = 'T' AND t3._message_ident = t0._message_ident
  GROUP BY 1, 2, 3
) AS t0
LEFT JOIN
(SELECT _message_ident, address FROM
(SELECT t2._message_ident,t2.address,t0.time FROM Prelude_Address AS t2,Prelude_CreateTime AS t0
WHERE t2._parent_type = 'S' and t2._message_ident = t0._message_ident AND t2._index = 0
AND t0.time >= '2006-03-21 12:50:06' AND t0.time <= '2006-03-21 13:30:52' AND t0._parent_type = 'A' GROUP BY 1,2,3) AS t2) as t2
ON (t2._message_ident = t0._message_ident)
```

#### Descripción del Hallazgo:

Se identificó una consulta SQL compleja que podría exponer información sensible, incluyendo direcciones IP y marcas de tiempo a través de las tablas Prelude\_CreateTime y Prelude\_Address. Esta exposición podría ser problemática si la consulta es accesible públicamente o si no existen controles adecuados de seguridad.

---

**Impacto Potencial:**

Divulgación de Datos Sensibles: Las direcciones IP y las marcas de tiempo pueden utilizarse para identificar patrones de acceso y potencialmente facilitar ataques dirigidos.

Riesgo de Explotación: Si un atacante tuviera acceso a esta consulta, podría aprovecharla para obtener información sensible mediante técnicas de explotación como la inyección SQL.

**Recomendación:**

- Restringir el acceso a las consultas SQL únicamente al personal autorizado.
- Implementar procedimientos almacenados en lugar de permitir la ejecución de consultas directas desde interfaces públicas.

## ***2. Exposición de Dirección IP Predeterminada (en el mismo documento)***

```
SELECT CASE WHEN t2.address IS NULL OR t2.address = ' ' THEN '127.0.0.1' ELSE t2.address END,t0.address,MAX(t0.time)
```

**Descripción del Riesgo:**

El uso de la IP predeterminada 127.0.0.1 en caso de valores nulos podría enmascarar registros importantes de auditoría. Esto dificulta la detección de accesos no autorizados, ya que las entradas sin una dirección válida se registran como provenientes de la IP local.

**Impacto Potencial:**

Ocultación de Actividad Sospechosa: Los intentos de acceso externos podrían no ser detectados correctamente.

Complicaciones en la Auditoría: La uniformidad de las IP registradas podría impedir identificar accesos maliciosos.

---

---

**Recomendación:**

Utilizar un valor nulo o una dirección específica que permita identificar correctamente accesos anómalos.

**3. Archivo *bbddJARA\_InnoDB.sql* con Datos Sensibles**

**site:portal-institucional.edu inurl:fichero "base de datos" | "bbddjara" | "InnoDB"**

**Descripción del Hallazgo:**

Se encontró un volcado completo de la base de datos jara, el cual contenía tanto la estructura como los datos de diversas tablas, incluyendo:

- **clientes:** Posible exposición de datos personales de usuarios.
- **facturas\_recibidas y pedidos:** Información financiera y de transacciones.
- **usuarios:** Incluye potencialmente credenciales de acceso sin cifrar.

**Impacto Potencial:**

**Exposición de Datos Personales:** Podría violar regulaciones de privacidad, como el RGPD.

**Riesgo de Ataques de Ingeniería Social:** La información expuesta podría ser utilizada para suplantación de identidad o fraudes.

**Recomendación:**

- Eliminar inmediatamente el archivo expuesto.
  - Realizar una auditoría de seguridad para identificar posibles accesos no autorizados previos.
-

---

## Estructura de la base de datos:

Base de datos: jara

Tablas incluidas:

### acciones

```
--  
-- Estructura de tabla para la tabla `acciones`  
--  
CREATE TABLE IF NOT EXISTS `acciones` (  
  `id_accion` int(10) NOT NULL auto_increment,  
  `nombre` varchar(20) collate latin1_general_ci NOT NULL default '',  
  `descripcion` varchar(50) collate latin1_general_ci default NULL,  
  PRIMARY KEY (`id_accion`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=5 ;  
  
--  
-- Volcar la base de datos para la tabla `acciones`  
--  
  
INSERT INTO `acciones` VALUES (1, 'Tarea tecnica', '');  
INSERT INTO `acciones` VALUES (2, 'Tarea administrativa', NULL);  
INSERT INTO `acciones` VALUES (3, 'Otros', NULL);  
INSERT INTO `acciones` VALUES (4, 'Cerrar', '');
```

### albaran\_entrada

---

```
--
-- Estructura de tabla para la tabla `albaran_entrada`
--

CREATE TABLE IF NOT EXISTS `albaran_entrada` (
  `id` int(5) NOT NULL auto_increment,
  `fecha_entrada` datetime default NULL,
  `id_tercero` int(11) default NULL,
  `personal_recepcion` int(11) default NULL,
  `num_alb` varchar(100) collate latin1_general_ci default NULL,
  PRIMARY KEY (`id`),
  KEY `FK_TERCERO` (`id_tercero`),
  KEY `FK_PERSONAL_RX` (`personal_recepcion`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=1 ;

--
-- Volcar la base de datos para la tabla `albaran_entrada`
--
```

#### albaran\_salida

```
--
-- Estructura de tabla para la tabla `albaran_salida`
--

CREATE TABLE IF NOT EXISTS `albaran_salida` (
  `id` int(5) NOT NULL auto_increment,
  `id_tercero` int(11) default NULL,
  `fecha_expedido` datetime default NULL,
  `nota_expedido` varchar(50) collate latin1_general_ci default NULL,
  `personal_expedido` int(11) default NULL,
  `pedido` varchar(50) collate latin1_general_ci default NULL,
  `factura` varchar(50) collate latin1_general_ci default NULL,
  `fecha_entregado` datetime default NULL,
  `nota_entregado` varchar(50) collate latin1_general_ci default NULL,
  `personal_entregado` int(11) default NULL,
  PRIMARY KEY (`id`),
  KEY `FK_PEDIDO` (`pedido`),
  KEY `FK_TERCERO_RX` (`id_tercero`),
  KEY `FK_PERSONAL_EXP` (`personal_expedido`),
  KEY `FK_PERSONAL_TX` (`personal_entregado`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=1 ;

--
-- Volcar la base de datos para la tabla `albaran_salida`
--
```

#### clientes

```
--
-- Estructura de tabla para la tabla `clientes`
--

CREATE TABLE IF NOT EXISTS `clientes` (
  `id_cliente` int(10) NOT NULL auto_increment,
  `nombre` varchar(50) collate latin1_general_ci NOT NULL default '',
  `localizacion` varchar(80) collate latin1_general_ci default NULL,
  `telefono` int(15) default NULL,
  `correo_electronico` varchar(50) collate latin1_general_ci default NULL,
  PRIMARY KEY (`id_cliente`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=1 ;

--
-- Volcar la base de datos para la tabla `clientes`
--
```

### facturas\_recibidas

```
--
-- Estructura de tabla para la tabla `facturas_recibidas`
--

CREATE TABLE IF NOT EXISTS `facturas_recibidas` (
  `id_fact_rx` int(5) NOT NULL auto_increment,
  `id_tercero` int(5) default NULL,
  `num_factura` varchar(100) collate latin1_general_ci default NULL,
  `fecha_factura` date default NULL,
  `base` double default NULL,
  `cuota_iva` double default NULL,
  `fecha_vencimiento` date default NULL,
  `descripcion` varchar(50) collate latin1_general_ci default NULL,
  `num_reg` varchar(50) collate latin1_general_ci default NULL,
  `tecnico` int(11) default '0',
  `fecha_insercion` date default '0000-00-00',
  `recargo` double default '0',
  `retencion` double default '0',
  `verificado` int(11) default NULL,
  `personal_verificado` int(11) default NULL,
  `fecha_verificado` date default NULL,
  `correcto` int(11) default '0',
  `a_credito` int(11) default '0',
  `con_anticipo` int(11) default '0',
  `pago_previo` int(11) default '0',
  `pago_contado` int(11) default '0',
  `obs_pago` varchar(100) collate latin1_general_ci default '',
  PRIMARY KEY (`id_fact_rx`),
  UNIQUE KEY `num_reg_rx` (`num_reg`),
  KEY `FK_TERCERO_FACT` (`id_tercero`),
  KEY `FK_PERSONAL_FACT_RX` (`tecnico`),
  KEY `FK_PERSONAL_VERIF` (`personal_verificado`),
  KEY `FK_NUM_REG_FACT` (`num_reg`),
  KEY `FK_NUM_REG_INDEX` (`num_reg`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=1 ;

--
-- Volcar la base de datos para la tabla `facturas_recibidas`
--
```

---

## lineas\_otros

```
--  
-- Estructura de tabla para la tabla `lineas_otros`  
--  
CREATE TABLE IF NOT EXISTS `lineas_otros` (  
  `id_lineas_otros` int(5) NOT NULL auto_increment,  
  `descripcion` varchar(100) collate latin1_general_ci default NULL,  
  `destino_analitico` varchar(100) collate latin1_general_ci default NULL,  
  `base` double default NULL,  
  `num_reg_fact` varchar(50) collate latin1_general_ci default NULL,  
  PRIMARY KEY (`id_lineas_otros`),  
  KEY `FK_NUM_REGOTROS` (`num_reg_fact`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=1 ;  
  
--  
-- Volcar la base de datos para la tabla `lineas_otros`  
--
```

## material

```
--  
-- Estructura de tabla para la tabla `material`  
--  
CREATE TABLE IF NOT EXISTS `material` (  
  `id_material` int(5) NOT NULL auto_increment,  
  `id_alb_in` int(11) default NULL,  
  `id_alb_out` int(11) default NULL,  
  `uds` double default NULL,  
  `descripcion` varchar(50) collate latin1_general_ci default NULL,  
  `num_serie` varchar(50) collate latin1_general_ci default NULL,  
  `destino` varchar(50) collate latin1_general_ci default NULL,  
  `base` double default NULL,  
  `num_reg_fact` varchar(50) collate latin1_general_ci default NULL,  
  PRIMARY KEY (`id_material`),  
  KEY `PK_ALB_IN` (`id_alb_in`),  
  KEY `FK_ID_ALBOUT` (`id_alb_out`),  
  KEY `FK_ALB_NUM_REG_FACT` (`num_reg_fact`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=1 ;  
  
--  
-- Volcar la base de datos para la tabla `material`  
--
```

---



## material\_compu

```
--
-- Estructura de tabla para la tabla `material_compu`
--

CREATE TABLE IF NOT EXISTS `material_compu` (
  `ID` int(11) default NULL,
  `codigo` text collate latin1_general_ci,
  `descripcion` text collate latin1_general_ci,
  `precio` double default '0'
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci;

--
-- Volcar la base de datos para la tabla `material_compu`
--
```

## material\_compu\_p

```
--
-- Estructura de tabla para la tabla `material_compu_p`
--

CREATE TABLE IF NOT EXISTS `material_compu_p` (
  `ID` int(11) default NULL,
  `codigo` text collate latin1_general_ci,
  `descripcion` text collate latin1_general_ci,
  `pvp_e` text collate latin1_general_ci
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci;

--
-- Volcar la base de datos para la tabla `material_compu_p`
--
```

## menu\_principal

```
--
-- Estructura de tabla para la tabla `menu_principal`
--

CREATE TABLE IF NOT EXISTS `menu_principal` (
  `id` int(5) NOT NULL auto_increment,
  `nombre` varchar(100) collate latin1_general_ci default NULL,
  `url` varchar(100) collate latin1_general_ci default NULL,
  `descripcion` varchar(100) collate latin1_general_ci default NULL,
  `grupo` int(11) default NULL,
  PRIMARY KEY (`id`),
  KEY `PK_GRUPO_PADRE` (`grupo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=8 ;

--
-- Volcar la base de datos para la tabla `menu_principal`
--

INSERT INTO `menu_principal` VALUES (1, 'Notas', NULL, 'Notas', 2);
INSERT INTO `menu_principal` VALUES (2, 'Pedidos', NULL, 'pedidos', 1);
INSERT INTO `menu_principal` VALUES (4, 'Logistica', NULL, 'logistico', 4);
INSERT INTO `menu_principal` VALUES (5, 'Administracion', NULL, 'administracion', 5);
INSERT INTO `menu_principal` VALUES (6, 'Herramientas', NULL, 'herramientas', 6);
INSERT INTO `menu_principal` VALUES (7, 'Informacion', NULL, 'informacion de procesos', 7);

-----
```

## menús

```
--
-- Estructura de tabla para la tabla `menus`
--

CREATE TABLE IF NOT EXISTS `menus` (
  `id_menu` int(10) NOT NULL auto_increment,
  `nombre` varchar(20) collate latin1_general_ci NOT NULL default '',
  `url` varchar(50) collate latin1_general_ci NOT NULL default '',
  `descripcion` varchar(50) collate latin1_general_ci default NULL,
  `orden` int(10) default '0',
  `grupo` int(11) default NULL,
  `tipo_accion` varchar(100) collate latin1_general_ci default '',
  PRIMARY KEY (`id_menu`),
  KEY `PK_GRUPO` (`grupo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=52 ;

--
-- Volcar la base de datos para la tabla `menus`
--

INSERT INTO `menus` VALUES (1, 'Mis notas', 'ListadoNotas.php?mis_notas=1', 'Listado de notas asignadas al usuario', 1, 2, '');
INSERT INTO `menus` VALUES (2, 'Nueva nota', 'NuevaNota2.php', 'Crear una nueva nota', 1, 2, '');
INSERT INTO `menus` VALUES (4, 'Notas por cerrar', 'NotasPorCerrar.php', '', 1, 2, '');
INSERT INTO `menus` VALUES (5, 'Notas Pospuestas', 'ListadoNotas.php?nevera=1', 'Notas pospuestas hasta una fecha indicada', 1, 2, 'Mis objetos notas pospuestas');
INSERT INTO `menus` VALUES (7, 'Filtrado de notas', 'FiltroNotas.php', '', 1, 2, '');
INSERT INTO `menus` VALUES (11, 'Nueva Fra Recibida', 'FacturaNueva.php', '', 2, 5, '');
INSERT INTO `menus` VALUES (12, 'Consulta fra', 'FacturaSelec.php', '', 2, 5, '');
INSERT INTO `menus` VALUES (13, 'Lista Fras', 'FacturaSelec.php?env=1&actualizar=1', 'Modifica una factura', 2, 6, '');
INSERT INTO `menus` VALUES (14, 'Fact pend verificar', 'FacturaSelec.php?env=1&verificar=1', 'Facturas recibidas pendientes de verificar', 2, 5, '');
INSERT INTO `menus` VALUES (21, 'Abrir Pedido', 'Pedidos.php', 'inserta nuevo pedido o lo modifica', 2, 1, '');
INSERT INTO `menus` VALUES (22, 'Lista Pedido', 'ListaPedidos.php', 'lista pedidos', 2, 6, '');
INSERT INTO `menus` VALUES (23, 'Mis pedidos', 'ListaPedidos.php?mis_pedidos=1', 'Lista pedidos abiertos y asignados a mi', 2, 1, '');
INSERT INTO `menus` VALUES (24, 'Mis facturaciones', 'ListaPedidos.php?pend_frar=1', 'pedidos cerrado pend de facturar', 2, 1, '');
INSERT INTO `menus` VALUES (31, 'Recepcionar', 'AlbaranEntrada.php', 'administracion de albaranes entrada', 2, 4, '');
INSERT INTO `menus` VALUES (32, 'Alb pend fact', 'ListadoAlbOut.php?facturar=1', 'Albaranes de salida entregados pendientes de factu', 2, NULL, '');
INSERT INTO `menus` VALUES (33, 'Expedir', 'AlbaranSalida.php', 'administracion de albaranes salida', 2, 4, '');
INSERT INTO `menus` VALUES (34, 'Mis entregas', 'ListadoAlbOut.php?entregar=1', 'Albaranes de salida pendientes de entregar', 2, 4, '');
INSERT INTO `menus` VALUES (35, 'Lista Alb entrada', 'ListaAlbIn.php', 'lista y modificacion de albaranes entrada', 2, 6, '');
INSERT INTO `menus` VALUES (36, 'Lista Alb Salida', 'ListadoAlbOut.php?actualizar=1', 'Lista y modif de albaranes salida', 2, 6, '');
INSERT INTO `menus` VALUES (37, 'Lista Material', 'ListadoMaterial.php', 'Lista material', 2, 4, '');
INSERT INTO `menus` VALUES (42, 'Datos Terceros', 'DatosTercero.php', 'Informacion de los terceros', 2, 6, 'Filtrado de objeto');
INSERT INTO `menus` VALUES (50, 'Notas Retrasadas', 'NotasRetrasadas.php', 'Notas con f.compromiso vencidas y retrasadas', 1, 7, '');
INSERT INTO `menus` VALUES (51, 'Pedidos ICT', 'PedidosICT.php', 'Lista los pedidos ICT abiertos', 2, 7, 'Filtrado de objeto');
```

## nota

```
--
-- Estructura de tabla para la tabla `nota`
--

CREATE TABLE IF NOT EXISTS `nota` (
  `id_nota` int(10) NOT NULL auto_increment,
  `usuario_cliente` int(10) NOT NULL default '0',
  `fecha_solicitud` datetime NOT NULL default '0000-00-00 00:00:00',
  `resumen` varchar(40) collate latin1_general_ci default NULL,
  `descripcion` text collate latin1_general_ci,
  `fecha_solicitada` date default '0000-00-00',
  `prioridad_cliente` int(10) default NULL,
  `personal_recepcion` int(10) NOT NULL default '0',
  `fecha_compromiso` date default '0000-00-00',
  `prioridad_asignada` int(10) NOT NULL default '0',
  `tipo_nota_inicial` int(10) NOT NULL default '0',
  `tipo_averia_inicial` int(10) NOT NULL default '0',
  `tipo_nota_final` int(10) default NULL,
  `tipo_averia_final` int(10) default NULL,
  `fecha_aviso_usuario` date default NULL,
  `tipo_cierre` int(10) NOT NULL default '0',
  `facturacion` tinyint(1) NOT NULL default '0',
  `fec_pospuesta` date default NULL,
  PRIMARY KEY (`id_nota`),
  KEY `PK_USUARIO_CLIENTE` (`usuario_cliente`),
  KEY `PK_PRIORIDAD_cli` (`prioridad_cliente`),
  KEY `PK_TIPO_NOTA` (`tipo_nota_inicial`),
  KEY `PK_PERSONAL_RX` (`personal_recepcion`),
  KEY `PK_TIPO_CIERRE` (`tipo_cierre`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=1 ;

--
-- Volcar la base de datos para la tabla `nota`
--
```

## pedidos

```
--
-- Estructura de tabla para la tabla `pedidos`
--

CREATE TABLE IF NOT EXISTS `pedidos` (
  `id` int(11) NOT NULL auto_increment,
  `cod_pedido` varchar(100) collate latin1_general_ci default NULL,
  `ofertas_ID_inicio` int(11) default NULL,
  `denominacion` text collate latin1_general_ci,
  `descripcion` text collate latin1_general_ci,
  `cliente` int(11) default NULL,
  `responsable` int(11) default NULL,
  `fecha_compromiso` datetime default NULL,
  `clase_id` int(11) default NULL,
  `tipo_id` int(11) default NULL,
  `observaciones` text collate latin1_general_ci,
  `importe_aprox` double default NULL,
  `margen_estandar` double default NULL,
  `financiacion` text collate latin1_general_ci,
  `fecha_inicio_estim` datetime default NULL,
  `horas_estimada` int(11) default NULL,
  `fecha_encargo` datetime default NULL,
  `tipo_pedido_id` int(11) default NULL,
  `estado` int(11) default NULL,
  `meses_estim` int(11) default '0',
  `factura` varchar(100) collate latin1_general_ci default '[NULL]',
  `destino_a` varchar(100) collate latin1_general_ci default '',
  `destino_b` varchar(100) collate latin1_general_ci default '',
  `destino_c` varchar(100) collate latin1_general_ci default '',
  PRIMARY KEY (`id`),
  KEY `PK_PEDIDOS_CLASE` (`clase_id`),
  KEY `PK_RESPONSABLE` (`responsable`),
  KEY `PK_TIPO_ID_PED` (`tipo_id`),
  KEY `PK_ESTADO` (`estado`),
  KEY `PK_SOPORTE_TIPO_PED` (`tipo_pedido_id`),
  KEY `PK_CLIENTE` (`cliente`),
  KEY `PK_PEDIDO_CODIGO` (`cod_pedido`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=1 ;

--
-- Volcar la base de datos para la tabla `pedidos`
--
```

## pedidos\_clase

```
--
-- Estructura de tabla para la tabla `pedidos_clase`
--

CREATE TABLE IF NOT EXISTS `pedidos_clase` (
  `id` int(5) NOT NULL auto_increment,
  `denominacion` varchar(50) collate latin1_general_ci default NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=5 ;

--
-- Volcar la base de datos para la tabla `pedidos_clase`
--

INSERT INTO `pedidos_clase` VALUES (1, 'Proyecto');
INSERT INTO `pedidos_clase` VALUES (2, 'Iguala');
INSERT INTO `pedidos_clase` VALUES (3, 'Accion puntual');
INSERT INTO `pedidos_clase` VALUES (4, 'Compra de material');

--
-----
```

---

## pedidos\_estado

```
--
-- Estructura de tabla para la tabla `pedidos_estado`
--

CREATE TABLE IF NOT EXISTS `pedidos_estado` (
  `id` int(5) NOT NULL auto_increment,
  `nombre` varchar(50) collate latin1_general_ci default NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=8 ;

--
-- Volcar la base de datos para la tabla `pedidos_estado`
--

INSERT INTO `pedidos_estado` VALUES (1, 'Pendiente de abrir');
INSERT INTO `pedidos_estado` VALUES (2, 'Abierto');
INSERT INTO `pedidos_estado` VALUES (3, 'Terminado, pendiente de facturar');
INSERT INTO `pedidos_estado` VALUES (4, 'Facturarado sin cerrar');
INSERT INTO `pedidos_estado` VALUES (5, 'Cerrado y facturado');
INSERT INTO `pedidos_estado` VALUES (6, 'Cancelado');
INSERT INTO `pedidos_estado` VALUES (7, 'Especiales');

-- -----
```

## pedidos\_soporte

```
--
-- Estructura de tabla para la tabla `pedidos_soporte`
--

CREATE TABLE IF NOT EXISTS `pedidos_soporte` (
  `id` int(5) NOT NULL auto_increment,
  `denominacion` varchar(50) collate latin1_general_ci default NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=5 ;

--
-- Volcar la base de datos para la tabla `pedidos_soporte`
--

INSERT INTO `pedidos_soporte` VALUES (1, 'Verbal');
INSERT INTO `pedidos_soporte` VALUES (2, 'Presupuesto sellado');
INSERT INTO `pedidos_soporte` VALUES (3, 'Hoja de encargo');
INSERT INTO `pedidos_soporte` VALUES (4, 'Hoja pedido cliente');

-- -----
```

---

## pedidos\_tipo

```
--
-- Estructura de tabla para la tabla `pedidos_tipo`
--

CREATE TABLE IF NOT EXISTS `pedidos_tipo` (
  `id` int(5) NOT NULL auto_increment,
  `denominacion` varchar(50) collate latin1_general_ci default NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=23 ;

--
-- Volcar la base de datos para la tabla `pedidos_tipo`
--

INSERT INTO `pedidos_tipo` VALUES (1, 'Accion puntual');
INSERT INTO `pedidos_tipo` VALUES (5, 'Proyecto ICT');
INSERT INTO `pedidos_tipo` VALUES (6, 'Accion puntual');
INSERT INTO `pedidos_tipo` VALUES (7, 'Venta equipos informatico');
INSERT INTO `pedidos_tipo` VALUES (8, 'Pagina Web');
INSERT INTO `pedidos_tipo` VALUES (9, 'Hosting');
INSERT INTO `pedidos_tipo` VALUES (10, 'Infraestructura');
INSERT INTO `pedidos_tipo` VALUES (11, 'Portal de internet');
INSERT INTO `pedidos_tipo` VALUES (12, 'Reparaciones');
INSERT INTO `pedidos_tipo` VALUES (13, 'Consumibles');
INSERT INTO `pedidos_tipo` VALUES (14, 'Aplicaciones');
INSERT INTO `pedidos_tipo` VALUES (15, 'Redes');
INSERT INTO `pedidos_tipo` VALUES (16, 'comunicaciones');
INSERT INTO `pedidos_tipo` VALUES (17, 'Igualas');
INSERT INTO `pedidos_tipo` VALUES (18, 'formacion');
INSERT INTO `pedidos_tipo` VALUES (19, 'otros');
INSERT INTO `pedidos_tipo` VALUES (20, 'Proyecto ing. industrial');
INSERT INTO `pedidos_tipo` VALUES (21, 'Alquiler');
INSERT INTO `pedidos_tipo` VALUES (22, 'Solucion informatica comp');
--
-----
```

## perfil\_accion

```
--
-- Estructura de tabla para la tabla `perfil_accion`
--

CREATE TABLE IF NOT EXISTS `perfil_accion` (
  `perfil` int(10) NOT NULL default '0',
  `accion` int(10) NOT NULL default '0',
  PRIMARY KEY (`perfil`,`accion`),
  KEY `PK_PERFIL_ACCION` (`perfil`),
  KEY `PK_ACCION_PERFIL` (`accion`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci;

--
-- Volcar la base de datos para la tabla `perfil_accion`
--
```

---

### perfil\_menu

```
--  
-- Estructura de tabla para la tabla `perfil_menu`  
--  
  
CREATE TABLE IF NOT EXISTS `perfil_menu` (  
  `accion_menu` int(10) NOT NULL default '0',  
  `perfil` int(10) NOT NULL default '0',  
  PRIMARY KEY (`accion_menu`,`perfil`),  
  KEY `PK_PERFIL_MENU` (`perfil`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci;  
  
--  
-- Volcar la base de datos para la tabla `perfil_menu`  
--
```

### perfil\_personal

```
--  
-- Estructura de tabla para la tabla `perfil_personal`  
--  
  
CREATE TABLE IF NOT EXISTS `perfil_personal` (  
  `id_perf_personal` int(11) NOT NULL auto_increment,  
  `personal_id` int(11) default NULL,  
  `perf_id` int(11) default NULL,  
  PRIMARY KEY (`id_perf_personal`),  
  KEY `PK_PERSONAL_PERFIL` (`personal_id`),  
  KEY `PK_PERFIL_PERSONAL` (`perf_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=1 ;  
  
--  
-- Volcar la base de datos para la tabla `perfil_personal`  
--
```

### perfiles

```
--  
-- Estructura de tabla para la tabla `perfiles`  
--  
  
CREATE TABLE IF NOT EXISTS `perfiles` (  
  `id_perfil` int(10) NOT NULL auto_increment,  
  `nombre` varchar(20) collate latin1_general_ci NOT NULL default '0',  
  `descripcion` varchar(50) collate latin1_general_ci default NULL,  
  PRIMARY KEY (`id_perfil`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=1 ;  
  
--  
-- Volcar la base de datos para la tabla `perfiles`  
--
```

---

---

## personal

```
--
-- Estructura de tabla para la tabla `personal`
--

CREATE TABLE IF NOT EXISTS `personal` (
  `id_personal` int(11) NOT NULL auto_increment,
  `login` varchar(10) collate latin1_general_ci NOT NULL default '',
  `password` varchar(40) collate latin1_general_ci NOT NULL default '',
  `nombre` varchar(30) collate latin1_general_ci default NULL,
  `perfil` varchar(20) collate latin1_general_ci NOT NULL default '',
  `telefono` int(11) default NULL,
  `correo_electronico` varchar(50) collate latin1_general_ci default NULL,
  `iniciales_nom` varchar(100) collate latin1_general_ci default '',
  PRIMARY KEY (`id_personal`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=1 ;

--
-- Volcar la base de datos para la tabla `personal`
--
```

## prioridad

```
--
-- Estructura de tabla para la tabla `prioridad`
--

CREATE TABLE IF NOT EXISTS `prioridad` (
  `id_prioridad` int(10) NOT NULL auto_increment,
  `nombre` varchar(20) collate latin1_general_ci NOT NULL default '0',
  `descripcion` varchar(50) collate latin1_general_ci default NULL,
  PRIMARY KEY (`id_prioridad`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=4 ;

--
-- Volcar la base de datos para la tabla `prioridad`
--

INSERT INTO `prioridad` VALUES (1, 'Maxima', NULL);
INSERT INTO `prioridad` VALUES (2, 'Media', NULL);
INSERT INTO `prioridad` VALUES (3, 'Minima', NULL);
```

## resolución

```
--
-- Estructura de tabla para la tabla `resolucion`
--

CREATE TABLE IF NOT EXISTS `resolucion` (
  `id_resolucion` int(10) NOT NULL auto_increment,
  `nota` int(10) NOT NULL default '0',
  `fecha_asignada` datetime NOT NULL default '0000-00-00 00:00:00',
  `persona_asignada` int(10) NOT NULL default '0',
  `proxima_accion` int(10) NOT NULL default '0',
  `observaciones` text collate latin1_general_ci,
  `tarefas_realizadas` text collate latin1_general_ci,
  `instrucciones` text collate latin1_general_ci NOT NULL,
  `minutos` int(10) default NULL,
  `asignado` int(10) NOT NULL default '0',
  PRIMARY KEY (`id_resolucion`),
  KEY `nota` (`nota`),
  KEY `FK_PERSONA_ASIGNADA` (`persona_asignada`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=1 ;

--
-- Volcar la base de datos para la tabla `resolucion`
--
```

## terceros

```
--
-- Estructura de tabla para la tabla `terceros`
--

CREATE TABLE IF NOT EXISTS `terceros` (
  `ID_tercero` int(11) NOT NULL auto_increment,
  `numero_tercero` int(11) default NULL,
  `denominacion_tercero` text collate latin1_general_ci,
  `nombre_razon` text collate latin1_general_ci,
  `nif_cif` text collate latin1_general_ci,
  `dir_facturacion` text collate latin1_general_ci,
  `poblacion_fact` text collate latin1_general_ci,
  `cp_fact` text collate latin1_general_ci,
  `provincia_fact` text collate latin1_general_ci,
  `dir_envio` text collate latin1_general_ci,
  `poblacion_envio` text collate latin1_general_ci,
  `cp_envio` text collate latin1_general_ci,
  `provincia_envio` text collate latin1_general_ci,
  `telefono` text collate latin1_general_ci,
  `fax` text collate latin1_general_ci,
  `correo_electronico` text collate latin1_general_ci,
  `contacto` text collate latin1_general_ci,
  `cargo_contacto` text collate latin1_general_ci,
  `telefono_contacto` text collate latin1_general_ci,
  `condiciones_pago` text collate latin1_general_ci,
  `condiciones_cobro` text collate latin1_general_ci,
  `notas` mediumtext collate latin1_general_ci,
  `web` mediumtext collate latin1_general_ci,
  `revisado` tinyint(1) default NULL,
  `tipo_ppal_3` text collate latin1_general_ci,
  `sector` text collate latin1_general_ci,
  `comercial` text collate latin1_general_ci,
  `cliente` tinyint(1) default NULL,
  `proveedor` tinyint(1) default NULL,
  `empleado` tinyint(1) default NULL,
  PRIMARY KEY (`ID_tercero`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=1 ;

--
-- Volcar la base de datos para la tabla `terceros`
--
```



---

## tipo\_averia

```
--  
-- Estructura de tabla para la tabla `tipo_averia`  
--  
  
CREATE TABLE IF NOT EXISTS `tipo_averia` (  
  `id_tipo_averia` int(10) NOT NULL auto_increment,  
  `nombre` varchar(20) collate latin1_general_ci NOT NULL default '',  
  `descripcion` varchar(50) collate latin1_general_ci default NULL,  
  PRIMARY KEY (`id_tipo_averia`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=6 ;  
  
--  
-- Volcar la base de datos para la tabla `tipo_averia`  
--  
  
INSERT INTO `tipo_averia` VALUES (1, 'Aplicaciones', NULL);  
INSERT INTO `tipo_averia` VALUES (2, 'Equipo consumidor', NULL);  
INSERT INTO `tipo_averia` VALUES (3, 'Comunicaciones', NULL);  
INSERT INTO `tipo_averia` VALUES (4, 'Explot. Admon', NULL);  
INSERT INTO `tipo_averia` VALUES (5, 'Iguales', 'Iguales');  
-- -----
```

## tipo\_cierre

```
--  
-- Estructura de tabla para la tabla `tipo_cierre`  
--  
  
CREATE TABLE IF NOT EXISTS `tipo_cierre` (  
  `id_tipo_cierre` int(10) NOT NULL auto_increment,  
  `nombre` varchar(20) collate latin1_general_ci NOT NULL default '',  
  `descripcion` varchar(50) collate latin1_general_ci default NULL,  
  PRIMARY KEY (`id_tipo_cierre`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=1 ;  
  
--  
-- Volcar la base de datos para la tabla `tipo_cierre`  
--
```

---

---

## tipo\_nota

```
--  
-- Estructura de tabla para la tabla `tipo_nota`  
--  
  
CREATE TABLE IF NOT EXISTS `tipo_nota` (  
  `id_tipo_nota` int(10) NOT NULL auto_increment,  
  `nombre` varchar(20) collate latin1_general_ci NOT NULL default '',  
  `descripcion` varchar(50) collate latin1_general_ci default NULL,  
  PRIMARY KEY (`id_tipo_nota`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=6 ;  
  
--  
-- Volcar la base de datos para la tabla `tipo_nota`  
--  
  
INSERT INTO `tipo_nota` VALUES (1, 'Proyecto', NULL);  
INSERT INTO `tipo_nota` VALUES (2, 'AT-Incidencia', NULL);  
INSERT INTO `tipo_nota` VALUES (3, 'AT-consulta', NULL);  
INSERT INTO `tipo_nota` VALUES (4, 'AT-Peticion', NULL);  
INSERT INTO `tipo_nota` VALUES (5, 'Iguala', 'Iguala');
```

## usuarios

```
--  
-- Estructura de tabla para la tabla `usuarios`  
--  
  
CREATE TABLE IF NOT EXISTS `usuarios` (  
  `id_usuario` int(10) NOT NULL auto_increment,  
  `nombre` varchar(20) collate latin1_general_ci NOT NULL default '',  
  `cliente` int(10) NOT NULL default '0',  
  `telefono` int(15) NOT NULL default '0',  
  `correo_electronico` varchar(50) collate latin1_general_ci default NULL,  
  `equipo` varchar(50) collate latin1_general_ci default NULL,  
  `tipo_equipo` varchar(50) collate latin1_general_ci default NULL,  
  `sistema_equipo` varchar(50) collate latin1_general_ci default NULL,  
  PRIMARY KEY (`id_usuario`),  
  KEY `PK_TERCERO` (`cliente`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=1 ;  
  
--  
-- Volcar la base de datos para la tabla `usuarios`  
--
```

---

## Filtros para las tablas anteriores:

```
--
-- Filtros para las tablas descargadas (dump)
--

--
-- Filtros para la tabla `albaran_entrada`
--
ALTER TABLE `albaran_entrada`
  ADD CONSTRAINT `albaran_entrada_ibfk_1` FOREIGN KEY (`id_tercero`) REFERENCES `clientes` (`id_cliente`) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE `albaran_entrada`
  ADD CONSTRAINT `albaran_entrada_ibfk_2` FOREIGN KEY (`personal_recepcion`) REFERENCES `personal` (`id_personal`) ON DELETE NO ACTION ON UPDATE NO ACTION;

--
-- Filtros para la tabla `albaran_salida`
--
ALTER TABLE `albaran_salida`
  ADD CONSTRAINT `albaran_salida_ibfk_1` FOREIGN KEY (`id_tercero`) REFERENCES `terceros` (`ID_tercero`) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE `albaran_salida`
  ADD CONSTRAINT `albaran_salida_ibfk_2` FOREIGN KEY (`personal_expedido`) REFERENCES `personal` (`id_personal`) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE `albaran_salida`
  ADD CONSTRAINT `albaran_salida_ibfk_3` FOREIGN KEY (`personal_entregado`) REFERENCES `personal` (`id_personal`) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE `albaran_salida`
  ADD CONSTRAINT `albaran_salida_ibfk_4` FOREIGN KEY (`pedido`) REFERENCES `pedidos` (`cod_pedido`) ON DELETE NO ACTION ON UPDATE NO ACTION;

--
-- Filtros para la tabla `facturas_recibidas`
--
ALTER TABLE `facturas_recibidas`
  ADD CONSTRAINT `facturas_recibidas_ibfk_1` FOREIGN KEY (`id_tercero`) REFERENCES `clientes` (`id_cliente`) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE `facturas_recibidas`
  ADD CONSTRAINT `facturas_recibidas_ibfk_2` FOREIGN KEY (`tecnico`) REFERENCES `personal` (`id_personal`) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE `facturas_recibidas`
  ADD CONSTRAINT `facturas_recibidas_ibfk_3` FOREIGN KEY (`personal_verificado`) REFERENCES `personal` (`id_personal`) ON DELETE NO ACTION ON UPDATE NO ACTION;

--
-- Filtros para la tabla `lineas_otros`
--
ALTER TABLE `lineas_otros`
  ADD CONSTRAINT `lineas_otros_ibfk_1` FOREIGN KEY (`num_reg_fact`) REFERENCES `facturas_recibidas` (`num_reg`) ON DELETE NO ACTION ON UPDATE NO ACTION;

--
-- Filtros para la tabla `material`
--
ALTER TABLE `material`
  ADD CONSTRAINT `material_ibfk_1` FOREIGN KEY (`id_alb_in`) REFERENCES `albaran_entrada` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE `material`
  ADD CONSTRAINT `material_ibfk_2` FOREIGN KEY (`id_alb_out`) REFERENCES `albaran_salida` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE `material`
  ADD CONSTRAINT `material_ibfk_3` FOREIGN KEY (`num_reg_fact`) REFERENCES `facturas_recibidas` (`num_reg`) ON DELETE NO ACTION ON UPDATE NO ACTION;

--
-- Filtros para la tabla `menus`
--
ALTER TABLE `menus`
  ADD CONSTRAINT `menus_ibfk_1` FOREIGN KEY (`grupo`) REFERENCES `menu_principal` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION;
```

```
--
-- Filtros para la tabla `nota`
--
ALTER TABLE `nota`
  ADD CONSTRAINT `nota_ibfk_1` FOREIGN KEY (`usuario_cliente`) REFERENCES `usuarios` (`id_usuario`) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE `nota`
  ADD CONSTRAINT `nota_ibfk_2` FOREIGN KEY (`prioridad_cliente`) REFERENCES `prioridad` (`id_prioridad`) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE `nota`
  ADD CONSTRAINT `nota_ibfk_3` FOREIGN KEY (`personal_recepcion`) REFERENCES `personal` (`id_personal`) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE `nota`
  ADD CONSTRAINT `nota_ibfk_4` FOREIGN KEY (`tipo_nota_inicial`) REFERENCES `tipo_nota` (`id_tipo_nota`) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE `nota`
  ADD CONSTRAINT `nota_ibfk_5` FOREIGN KEY (`tipo_cierre`) REFERENCES `tipo_cierre` (`id_tipo_cierre`) ON DELETE NO ACTION ON UPDATE NO ACTION;
--
-- Filtros para la tabla `pedidos`
--
ALTER TABLE `pedidos`
  ADD CONSTRAINT `pedidos_ibfk_1` FOREIGN KEY (`clase_id`) REFERENCES `pedidos` (`clase_id`) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE `pedidos`
  ADD CONSTRAINT `pedidos_ibfk_2` FOREIGN KEY (`estado`) REFERENCES `pedidos_estado` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE `pedidos`
  ADD CONSTRAINT `pedidos_ibfk_3` FOREIGN KEY (`cliente`) REFERENCES `clientes` (`id_cliente`) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE `pedidos`
  ADD CONSTRAINT `pedidos_ibfk_4` FOREIGN KEY (`responsable`) REFERENCES `usuarios` (`id_usuario`) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE `pedidos`
  ADD CONSTRAINT `pedidos_ibfk_5` FOREIGN KEY (`tipo_id`) REFERENCES `pedidos_tipo` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE `pedidos`
  ADD CONSTRAINT `pedidos_ibfk_6` FOREIGN KEY (`tipo_pedido_id`) REFERENCES `pedidos_soporte` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION;
--
-- Filtros para la tabla `perfil_accion`
--
ALTER TABLE `perfil_accion`
  ADD CONSTRAINT `perfil_accion_ibfk_1` FOREIGN KEY (`perfil`) REFERENCES `perfiles` (`id_perfil`) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE `perfil_accion`
  ADD CONSTRAINT `perfil_accion_ibfk_2` FOREIGN KEY (`accion`) REFERENCES `acciones` (`id_accion`) ON DELETE NO ACTION ON UPDATE NO ACTION;
--
-- Filtros para la tabla `perfil_menu`
--
ALTER TABLE `perfil_menu`
  ADD CONSTRAINT `perfil_menu_ibfk_1` FOREIGN KEY (`perfil`) REFERENCES `perfiles` (`id_perfil`) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE `perfil_menu`
  ADD CONSTRAINT `perfil_menu_ibfk_2` FOREIGN KEY (`accion_menu`) REFERENCES `menus` (`id_menu`) ON DELETE NO ACTION ON UPDATE NO ACTION;
--
-- Filtros para la tabla `perfil_personal`
--
ALTER TABLE `perfil_personal`
  ADD CONSTRAINT `perfil_personal_ibfk_1` FOREIGN KEY (`perf_id`) REFERENCES `perfil_personal` (`perf_id`) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE `perfil_personal`
  ADD CONSTRAINT `perfil_personal_ibfk_2` FOREIGN KEY (`personal_id`) REFERENCES `personal` (`id_personal`) ON DELETE NO ACTION ON UPDATE NO ACTION;
--
-- Filtros para la tabla `resolucion`
--
ALTER TABLE `resolucion`
  ADD CONSTRAINT `resolucion_ibfk_1` FOREIGN KEY (`nota`) REFERENCES `nota` (`id_nota`) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE `resolucion`
  ADD CONSTRAINT `resolucion_ibfk_2` FOREIGN KEY (`persona_asignada`) REFERENCES `personal` (`id_personal`) ON DELETE NO ACTION ON UPDATE NO ACTION;
```

```
--
-- Filtros para la tabla `usuarios`
--
ALTER TABLE `usuarios`
  ADD CONSTRAINT `usuarios_ibfk_1` FOREIGN KEY (`cliente`) REFERENCES `terceros` (`ID_tercero`) ON DELETE NO ACTION ON UPDATE NO ACTION;
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es">
  <head>
    <title>e-REdING. </title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <meta name="description" content="Buscador de proyectos ETSI" />
    <meta name="author" content="Daniel Callejo"/>
    <meta name="author" content="Federico L&acute;pez"/>
    <meta name="author" content="Juli&acute;n P&acute;rez"/>
    <meta name="author" content="Jos&eacute; Mar&iacute;a Vidal Vidal"/>
    <base href="https://biblus.us.es/bibing/proyectos/" /><link rel="shortcut icon" href="/favicon.ico" />
    <link rel="stylesheet" href="/publico/css/estructura.css" type="text/css" />
    <link rel="stylesheet" href="/publico/css/bootstrap.css" type="text/css" media="screen"/>
    <link rel="stylesheet" href="/publico/css/enlaces-capas.css" type="text/css" />
    <link rel="stylesheet" href="/publico/css/pestanas.css" type="text/css" />
    <link rel="stylesheet" href="/publico/css/formularios.css" type="text/css" />
    <link rel="stylesheet" href="/publico/css/lightbox.css" type="text/css" />
    <link rel="stylesheet" href="/publico/css/autocompleter.css" type="text/css" />
    <link rel="stylesheet" href="/publico/css/resultados.css" type="text/css" />
    <link rel="stylesheet" href="/publico/css/milista.css" type="text/css" />
    <link rel="stylesheet" href="/publico/css/explorador.css" type="text/css" />
    <link rel="stylesheet" href="/publico/css/ultimos.css" type="text/css" />
    <link rel="stylesheet" href="/publico/css/contactar.css" type="text/css" />
    <!-- <link rel="stylesheet" href="/css/print.css" type="text/css" media="print" /> -->
    <script src="/publico/javascript/jquery-1.9.1.min.js" type="text/javascript"></script>
    <script src="/publico/javascript/bootstrap.js" type="text/javascript"></script>

    <script src="/publico/javascript/mootools-1.2.4-core.js" type="text/javascript"></script>
    <script src="/publico/javascript/mootools-1.2.4.2-more.js" type="text/javascript"></script>
    <script src="/publico/javascript/targetbl.js" type="text/javascript"></script>
    <script src="/publico/javascript/mensajes.js" type="text/javascript"></script>
    <script src="/publico/javascript/form.js" type="text/javascript"></script>
    <script src="/publico/javascript/autocompleter/Observer.js" type="text/javascript"></script>
    <script src="/publico/javascript/autocompleter/Autocompleter.js" type="text/javascript"></script>
    <script src="/publico/javascript/autocompleter/Autocompleter.Request.js" type="text/javascript"></script>
    <script src="/publico/javascript/domready.js" type="text/javascript"></script>

    <!-- Google tag (gtag.js) -->
    <script async src="https://www.googletagmanager.com/gtag/js?id=G-2HP38G406P"></script>
    <script>
      window.dataLayer = window.dataLayer || [];
      function gtag(){dataLayer.push(arguments);}
      gtag('js', new Date());

      gtag('config', 'G-2HP38G406P');
    </script>
```

## 4. Exposición de Credenciales en Archivos de Configuración

```
site:portal-institucional.edu filetype:php | filetype:xml |
filetype:ini "db_password" | "database"
```

### Descripción del Hallazgo:

Se identificaron archivos de configuración que contenían credenciales de acceso a bases de datos. Estos archivos, al estar accesibles públicamente, representan un riesgo crítico, ya que

permiten a un atacante conectarse directamente a las bases de datos.

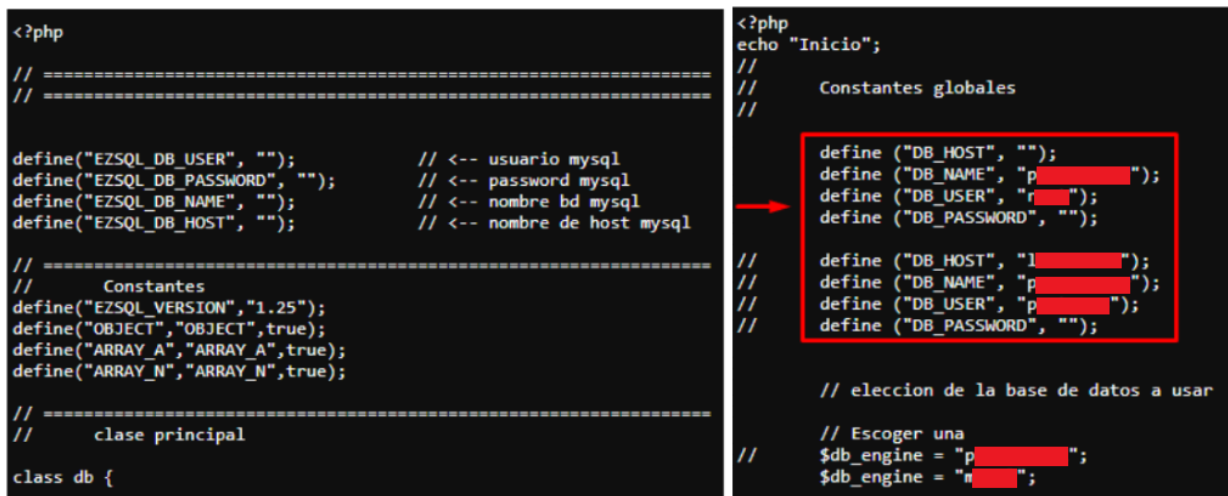
### Impacto Potencial:

- Acceso No Autorizado: Un atacante podría obtener acceso completo a la base de datos, manipular o extraer información sensible.
- Compromiso de la Infraestructura: Si las credenciales se reutilizan en otros sistemas, el riesgo podría extenderse más allá del portal Biblus.

### Recomendación:

- Mover las credenciales a archivos seguros fuera del directorio público.
- Implementar variables de entorno o un gestor seguro de secretos para almacenar credenciales.

### Credenciales para acceso a la base de datos expuestas en varios archivos



```
<?php
// =====
// =====

define("EZSQL_DB_USER", "");           // <-- usuario mysql
define("EZSQL_DB_PASSWORD", "");       // <-- password mysql
define("EZSQL_DB_NAME", "");           // <-- nombre bd mysql
define("EZSQL_DB_HOST", "");           // <-- nombre de host mysql

// =====
//      Constantes
define("EZSQL_VERSION", "1.25");
define("OBJECT", "OBJECT", true);
define("ARRAY_A", "ARRAY_A", true);
define("ARRAY_N", "ARRAY_N", true);

// =====
//      clase principal
class db {
```

```
<?php
echo "Inicio";
//
//      Constantes globales
//
define ("DB_HOST", "");
define ("DB_NAME", "p[REDACTED]");
define ("DB_USER", "r[REDACTED]");
define ("DB_PASSWORD", "");

define ("DB_HOST", "l[REDACTED]");
define ("DB_NAME", "p[REDACTED]");
define ("DB_USER", "p[REDACTED]");
define ("DB_PASSWORD", "");

// eleccion de la base de datos a usar

// Escoger una
$db_engine = "p[REDACTED]";
$db_engine = "m[REDACTED]";
```

```

//DeleteNote($note_id);

$user_id="aa";
$task_id=0;
$title="Nota r (new2)";
$text="Nota r (new2)";
AddNote($user_id, $task_id, $title, $text);
echo "3 ok";

$note_id=3;
GetNote($note_id, $title, $text, $owner, $task_id);
echo "4 ok";

$from_id="aa";
$to_id="3f";
$subject="Nuevo mensaje";
$message="Nuevo mensaje";
NewMessage($from_id, $to_id, $subject, $message);
echo "5 ok";

//DeleteMessage($message_id, $user_id);

$message_id=3;
GetMessage($message_id,$subject,$from_id,$message,$creation_date);
echo "6 ok";

$user_id="aa";
GetNodeLeafFromTree($user_id,$node,$leaf);
echo "7 ok";

$user_id="aa";
GetUserInfo($user_id,$name,$can_add_user,$active,$username,$password,$email);
echo "8 ok";

$node="r";
$leaf=0;
GetSlaves($node,$leaf,$slaves);
echo "9 ok";

$user_id="aa";
ListMessages($user_id, $arr_msgid, $arr_subj);
echo "10 ok";

$user_id="aa";
$task_id=0;
ListNotes($user_id, $task_id, $arr_noteid, $arr_title);
echo "11 ok";

//EnumTasks_Report($start_node, $callback, $filter);

//EnumTasks($start_node,$callback);

$task_id=22;
GetTask($task_id, $title, $state, $node, $leaf, $slave_id,
        $master_id, $due_date, $initial_date,
        $state_feedback, $description, $priority);
echo "12 ok";

```

```

<?php
//Toma los valores enviados por la aplicacion
$Nomb = "pg";
$Direc = "pg";
define("DB_HOST", "1"); //Nombre del host
define("DB_USER", "r"); //Nombre usuario de la base de datos
define("DB_PASSWORD", "r"); //Contraseña de la base de datos
define("DB_DATABASE", "e"); //Nombre de la base de datos.
//Conectamos con la base de datos
$con = mysql_connect(DB_HOST, DB_USER, DB_PASSWORD);
//Selección base de datos
mysql_select_db(DB_DATABASE);

$result = mysql_query("SELECT * FROM usuarios
WHERE username = '$Nomb' and direccion = '$Direc'") or die("Error en la consulta SQL");
//Comprueba que la dupla usuario-mail es correcta
if (!$result){
    //Si no existe dicha dupla, devuelve error
    echo "false";
    //Si existe, se envia mail.
}else{
    $row = mysql_fetch_array ( $result );
    echo $row [ "passw" ];
}

mysql_close();
//Codifica resultado en JSON
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es">
<head>
<title>e-REDING.</title>

```

```

<?php
//
//  Constantes globales
//
//
define ("DB_HOST", "1");
define ("DB_NAME", "p");
define ("DB_USER", "r");
define ("DB_PASSWORD", "r");

define ("DB_HOST", "");
define ("DB_NAME", "p");
define ("DB_USER", "p");
define ("DB_PASSWORD", "r");

// eleccion de la base de datos a usar

// Escoger una
$db_engine = "r";
//
$db_engine = "r";
//
$db_engine = "r";

// Lenguaje por defecto
$default_language = "r";

```





---

### Descripción del Hallazgo:

Durante la búsqueda accidental, se encontraron archivos de configuración internos que contenían referencias explícitas a nombres de usuario, contraseñas e IPs. Estos archivos, además de exponer información sensible, revelaron prácticas de codificación inseguras que podrían ser explotadas por actores malintencionados.

```
<?php

require "inc/init.php";
require "inc/small_functions.php";

$post_vars = array('username','password','language');
include "import_vars_post.inc";

// valida los datos

$username = addslashes(strip_slashes($username));
$username = addslashes(strip_slashes($username));
$password = addslashes(strip_slashes($password));

// Si no hay ningun usuario definido, solicita un nuevo usuario root
if (!is_users()) header ("Location: first_time.php");

// Verifica la identidad de usuario

if ($username) {
    $i++;
    if (AuthUser($username, $password, $user_id)) {
        setcookie("user_id", $user_id);
        setcookie("language", $language);
        setcookie("last_language", $language, time()+3600*24*7); // dura una semana
        header("Location: home.php");
        exit;
    }
}
```

Datos del HTML también descritos en el mismo documento descubierto

```
> BASES DE DATOS</h4>

<b> Raúl</b>
```

---

## 1. Falta de Validación y Saneamiento de Entradas

```
// valida los datos

$name = addslashes(strip_slashes($name));
$username = addslashes(strip_slashes($username));
$password = addslashes(strip_slashes($password));
$password2 = addslashes(strip_slashes($password2));
$email = addslashes(strip_slashes($email));

// Puede el usuario añadir a otros?
if (getUserInfo($user_id,$d,$can_add_user,$d,$d,$d,$d)) {
    // Si no puede, se le envia a home
    if ($can_add_user != 1) Header ("Location: home.php");
} else {
    // El usuario no existe
    // Se le manda a la pantalla de login
    Header ("Location: index.php");
}
```

El análisis del código mostró que, aunque se utilizan las funciones `AddSlashes` y `StripSlashes` para manejar las entradas de usuario, estas no son suficientes para proteger contra ataques de inyección SQL. Las entradas no validadas adecuadamente permiten que un atacante inyecte comandos maliciosos directamente en las consultas SQL, comprometiendo la integridad y seguridad de la base de datos.

### Impacto Potencial:

- **Riesgo de Inyección SQL:** Una validación insuficiente podría permitir consultas arbitrarias a la base de datos, posibilitando la exfiltración o manipulación de datos sensibles.
  - **Compromiso del Sistema:** Accesos no autorizados a la base de datos podrían derivar en la alteración de información crítica o la ejecución de comandos maliciosos.
-

---

**Recomendación:**

- Sustituir el uso de AddSlashes y StripSlashes por consultas preparadas o funciones específicas de escape de SQL (e.g., PDO o mysqli en PHP).
- Implementar filtros de entrada que validen y saneen los datos recibidos antes de procesarlos.

**2. Manejo Inseguro de Cookies**

```
// Verifica la identidad de usuario

if ($username) {
    $i++;
    if (AuthUser($username, $password, $user_id)) {
        setcookie("user_id", $user_id);
        setcookie("language", $language);
        setcookie("last_language", $language, time()+3600*24*7); // dura una semana
        Header("Location: home.php");
        exit;
    }
}
```

El código identificado establece las cookies user\_id, language y last\_language sin utilizar atributos de seguridad adicionales, como HttpOnly o Secure.

- HttpOnly: Evita el acceso a las cookies desde scripts del lado del cliente, reduciendo el riesgo de robo mediante ataques de Cross-Site Scripting (XSS).
- Secure: Garantiza que las cookies solo se envíen a través de conexiones cifradas (HTTPS), evitando la exposición en canales no seguros.

**Impacto Potencial:**

- Riesgo de Robo de Sesiones: Si un atacante logra explotar una vulnerabilidad XSS, podría acceder a las cookies y secuestrar sesiones de usuarios.
-

- 
- **Manipulación de Preferencias:** Las cookies sin protección pueden ser alteradas, lo cual podría desencadenar comportamientos no deseados en la aplicación.

**Recomendación:**

- Establecer todas las cookies con los atributos HttpOnly, Secure y SameSite.
- Revisar el uso de cookies sensibles y minimizar su uso en contextos críticos de seguridad.

**3. Uso de Variables Globales**

```
<?php

require "inc/init.php";
require "inc/small_functions.php";

$post_vars = array(
    'n', 'u', 'p', 'p', 'email', 'cancel'
);

include "import_vars_post.inc";

#$get_vars = array('task_id', 'note_id');
#include "import_vars_get.inc";
```

El código utiliza variables globales, como `$post_vars` y `$get_vars`, para gestionar las entradas del usuario. El uso de estas variables sin una correcta sanitización podría llevar a conflictos en el código y a vulnerabilidades de seguridad.

**Impacto Potencial:**

- **Dificultad en el Mantenimiento:** Las variables globales pueden ser modificadas desde cualquier parte del código, lo cual complica la depuración y el control de flujo.
-

- 
- **Riesgo de Inyección de Datos:** Si las variables no son filtradas correctamente, un atacante podría inyectar datos maliciosos en el flujo de la aplicación.

#### **Recomendación:**

- Sustituir las variables globales por arreglos asociativos seguros (\$\_POST, \$\_GET, \$\_SESSION).
- Implementar funciones específicas para la obtención de parámetros, asegurando una capa adicional de validación.

#### **4. Control de Permisos Débil**

```
// Puede el usuario añadir a otros?
if (GetUserInfo($user_id,$d,$can_add_user,$d,$d,$d,$d)) {
    // Si no puede, se le envia a home
    if ($can_add_user != 1) Header ("Location: home.php");
} else {
    // El usuario no existe
    // Se le manda a la pantalla de login
    Header ("Location: index.php");
}
```

El código realiza una verificación básica de permisos mediante la función `GetUserInfo`, sin aplicar controles adicionales que garanticen una segmentación adecuada de roles y permisos.

#### **Impacto Potencial:**

- **Escalada de Privilegios:** Usuarios con permisos limitados podrían acceder a funcionalidades restringidas.
- **Riesgo de Acciones No Autorizadas:** Falta de controles robustos podría permitir que un usuario estándar añada o modifique otros usuarios sin supervisión.

#### **Recomendación:**

---

- Implementar un sistema de roles y permisos más granular (e.g., RBAC: Role-Based Access Control).
- Revisar los permisos de cada acción crítica y aplicar el principio de privilegios mínimos.

## 5. Creación de Usuarios Administradores sin Seguridad Adecuada

```
// verifica si las claves coinciden  
if ("$password" != "$password2") $error = $strThepasswordsdontmatch;  
  
# Si se obtienen los parametros sin ningun mensaje de error  
# se salvan y se vuelve a la pantalla de login  
# De otra forma solicita de nuevo otro usuario root  
  
if ($name && $username && !$error) {  
    $id = CreateUser('r', '', $username, $password, $name, '', '1');  
    setcookie ("user_id", "$id");  
    Header ("Location: home.php");  
} else {
```

Se identificó un proceso de creación de usuarios con privilegios de administrador ('r\*\*\*\*\*') sin realizar las debidas verificaciones de seguridad. Además, las contraseñas se almacenan en texto plano, lo cual es una práctica extremadamente insegura.

### Impacto Potencial:

- Acceso Administrativo Ilegítimo: Cualquier usuario podría registrarse como administrador sin pasar por controles de seguridad.
- Riesgo de Brechas de Datos: En caso de un ataque, las contraseñas en texto plano pueden ser explotadas fácilmente.

---

**Recomendación:**

- Implementar validaciones estrictas al crear cuentas con privilegios elevados, incluyendo autenticación multifactor (MFA).
- Utilizar algoritmos de hashing seguro, como bcrypt, Argon2 o PBKDF2, para almacenar contraseñas.
- Realizar revisiones periódicas de los registros de administración para identificar accesos sospechosos.

## 4. Análisis de Riesgos

### *1. Exposición de Archivos SQL*

**Descripción del Riesgo:**

La exposición pública de archivos SQL permite a cualquier usuario acceder a consultas complejas, lo que podría derivar en la obtención de información sensible, como direcciones IP, marcas de tiempo y potencialmente datos personales o registros de autenticación.

**Impacto:**

Alto: La divulgación de información sensible podría facilitar ataques dirigidos, como inyecciones SQL, robo de identidad y explotación de vulnerabilidades del sistema.

**Probabilidad:**

Media: La exposición de archivos a través de consultas específicas sugiere una falta de control en el acceso a archivos

---



---

internos, lo que podría ser aprovechado por un atacante con conocimientos técnicos.

**Medidas Recomendadas:**

- Restringir el acceso a archivos SQL mediante autenticación robusta.
- Utilizar procedimientos almacenados en lugar de consultas directas accesibles desde interfaces públicas.

## ***2. Falta de Validación de Entradas y Riesgo de Inyección SQL***

**Descripción del Riesgo:**

El uso inadecuado de funciones como AddSlashes y StripSlashes sin una validación robusta expone la aplicación a ataques de inyección SQL, donde un atacante podría manipular las consultas para acceder o modificar datos críticos.

**Impacto:**

Muy Alto: Un ataque exitoso podría comprometer la integridad de la base de datos, permitir la filtración de información sensible o incluso otorgar acceso administrativo no autorizado.

**Probabilidad:**

Alta: La falta de consultas preparadas y la ausencia de funciones específicas para el saneamiento de entradas aumentan significativamente la probabilidad de explotación.

---

---

**Medidas Recomendadas:**

- Implementar consultas preparadas y funciones de escape específicas para cada motor de base de datos utilizado.
- Establecer filtros de entrada que validen el tipo, longitud y formato de los datos recibidos.

### **3. Manejo Inseguro de Cookies**

**Descripción del Riesgo:**

Las cookies críticas (`user_id`, `language`, `last_language`) carecen de atributos de seguridad como `HttpOnly` y `Secure`, lo cual podría permitir su manipulación o robo mediante ataques de Cross-Site Scripting (XSS).

**Impacto:**

Alto: Un atacante podría secuestrar sesiones de usuario, acceder a información personal o realizar acciones no autorizadas en nombre del usuario afectado.

**Probabilidad:**

Media: La explotación de esta vulnerabilidad requiere combinarla con un ataque XSS, pero la ausencia de medidas de seguridad incrementa el riesgo.

**Medidas Recomendadas:**

- Configurar las cookies con los atributos `HttpOnly`, `Secure` y `SameSite`.
  - Revisar el manejo de sesiones y aplicar controles adicionales para la validación de la identidad del usuario.
-

---

## **4. Creación de Usuarios Administradores sin Verificaciones Adecuadas**

### **Descripción del Riesgo:**

El código permite la creación de usuarios con privilegios de administrador ('r\*\*\*\*\*') sin verificaciones estrictas, exponiendo la aplicación a posibles accesos no autorizados y a una escalada de privilegios.

### **Impacto:**

Crítico: Cualquier usuario podría registrarse como administrador, comprometiendo por completo la seguridad del sistema y facilitando la ejecución de acciones maliciosas.

### **Probabilidad:**

Alta: La falta de controles en la creación de cuentas administrativas facilita la explotación de esta vulnerabilidad, especialmente si el formulario de registro es público.

### **Medidas Recomendadas:**

- Implementar autenticación multifactor (MFA) para cuentas administrativas.
  - Aplicar un control de roles y permisos más granular, verificando la legitimidad de cada solicitud de creación de administradores.
-

---

## **5. Almacenamiento de Contraseñas en Texto Plano**

### **Descripción del Riesgo:**

El almacenamiento de contraseñas sin cifrado (texto plano) representa una grave vulnerabilidad, ya que en caso de una brecha de datos, las credenciales podrían ser utilizadas directamente sin necesidad de decodificación.

### **Impacto:**

Muy Alto: Si un atacante accede a la base de datos, podría comprometer las cuentas de usuario, realizar ataques de credential stuffing o acceder a otros sistemas donde los usuarios reutilicen las mismas credenciales.

### **Probabilidad:**

Alta: Cualquier acceso no autorizado a la base de datos expondría de inmediato las contraseñas almacenadas sin protección.

### **Medidas Recomendadas:**

- Almacenar las contraseñas utilizando algoritmos de hashing seguro, como bcrypt, Argon2 o PBKDF2.
  - Implementar una política de renovación de contraseñas y notificar a los usuarios en caso de una posible brecha.
-

---

### Evaluación General del Riesgo

Vulnerabilidad	Impacto	Probabilidad	Nivel de Riesgo
Exposición de archivos SQL	Alto	Media	Alto
Inyección SQL por validación débil	Muy Alto	Alta	Crítico
Manejo inseguro de cookies	Alto	Media	Alto
Creación de administradores insegura	Crítico	Alta	Crítico
Almacenamiento de contraseñas en claro	Muy Alto	Alta	Crítico

El análisis muestra que varias vulnerabilidades identificadas presentan un riesgo crítico para la seguridad del sitio web. Se recomienda priorizar las acciones correctivas, especialmente en aspectos relacionados con la autenticación, el control de accesos y la protección de datos sensibles.

## 5. Recomendaciones. Formas de Mitigación o Corrección

Con base en el análisis de riesgos realizado, se proponen las siguientes recomendaciones para mitigar las vulnerabilidades identificadas en el sitio web institucional. Estas acciones tienen como objetivo reforzar la seguridad de la plataforma, proteger la información sensible y prevenir posibles incidentes de seguridad.

---

---

## 1. Revisión y Eliminación de Archivos Sensibles

- **Eliminar Archivos Expuestos:** Revisar y eliminar de inmediato los archivos sensibles detectados mediante las consultas avanzadas (Google Dorks), especialmente aquellos que contienen credenciales, estructuras de bases de datos o información de autenticación.
- **Control de Acceso a Archivos:** Configurar el servidor web para evitar la exposición pública de archivos de configuración y bases de datos, estableciendo reglas de acceso específicas en el .htaccess o mediante políticas de seguridad a nivel de servidor.
- **Realizar una Auditoría Completa:** Revisar todos los directorios accesibles públicamente para identificar posibles archivos expuestos adicionales.

## 2. Fortalecer la Autenticación y Autorización

- **Autenticación Multifactor (MFA):** Implementar un segundo factor de autenticación para accesos administrativos y acciones críticas dentro de la plataforma.
  - **Control de Roles y Permisos:** Aplicar un sistema de control basado en roles (RBAC: Role-Based Access Control) para segmentar adecuadamente los permisos y limitar el acceso solo a las funciones necesarias para cada perfil de usuario.
  - **Revisar las Políticas de Contraseñas:** Establecer políticas estrictas que incluyan la complejidad de contraseñas, la rotación periódica y la detección de contraseñas comprometidas en bases de datos públicas de fugas.
-

---

### 3. Uso de Cifrado para Proteger Datos Sensibles

- **Cifrado de Contraseñas:** Utilizar algoritmos de hashing seguro como bcrypt, Argon2 o PBKDF2 para almacenar las contraseñas en la base de datos.
- **Cifrado de Datos en Tránsito:** Implementar el protocolo HTTPS en todo el sitio utilizando un certificado TLS válido para asegurar que la información sensible se transmita de forma cifrada.
- **Cifrado de Datos en Reposo:** Considerar el uso de cifrado en la base de datos para campos críticos, especialmente aquellos que contienen datos personales o financieros.

### 4. Mejoras en la Configuración del Servidor Web

- **Deshabilitar la Exploración de Directorios:** Asegurarse de que el servidor web no permita listar el contenido de directorios públicos.
  - **Establecer Cabeceras de Seguridad:** Configurar cabeceras HTTP como:
    - Content-Security-Policy (CSP): Para evitar ataques XSS.
    - X-Frame-Options: Para prevenir ataques de clickjacking.
    - Strict-Transport-Security (HSTS): Para forzar la navegación segura mediante HTTPS.
  - **Configuración de Permisos en Archivos:** Asegurarse de que los archivos sensibles tengan los permisos adecuados (chmod 640) y que solo los procesos necesarios puedan acceder a ellos.
-

---

## 5. Implementar Monitorización y Auditoría de Seguridad

- **Registros de Acceso:** Activar el registro detallado de accesos al servidor y a la base de datos, incluyendo eventos críticos como inicios de sesión fallidos o modificaciones de privilegios.
- **Sistemas de Detección de Intrusiones (IDS):** Implementar herramientas como OSSEC, Snort o Wazuh para identificar actividades sospechosas en tiempo real.
- **Revisión Periódica de Logs:** Establecer un proceso para analizar regularmente los registros en busca de patrones anómalos o indicios de actividad maliciosa.

## 6. Políticas de Seguridad y Capacitación del Personal

- **Establecer Políticas Claras:** Desarrollar políticas de seguridad que incluyan la gestión de credenciales, el uso adecuado de datos sensibles y la respuesta a incidentes de seguridad.
  - **Capacitación Continua:** Proporcionar formación al personal técnico y administrativo sobre las mejores prácticas de seguridad, incluyendo la identificación de amenazas comunes como phishing y malware.
  - **Simulacros de Seguridad:** Realizar ejercicios prácticos para evaluar la respuesta del personal ante potenciales incidentes de seguridad.
-



---

## 7. Realizar Pruebas de Penetración Regulares

- **Pentesting Periódico:** Contratar servicios de pruebas de penetración para identificar nuevas vulnerabilidades antes de que puedan ser explotadas.
- **Análisis Automatizado de Seguridad:** Utilizar herramientas como OWASP ZAP, Burp Suite o Nessus para realizar análisis automáticos de seguridad en el sitio web.
- **Validar la Implementación de las Medidas de Seguridad:** Revisar los resultados de las pruebas para asegurarse de que las recomendaciones implementadas son efectivas y que no se introdujeron nuevas vulnerabilidades durante el proceso de corrección.

## 7. Conclusión

El presente informe documenta los hallazgos fortuitos relacionados con la posible exposición de información sensible en un sitio web institucional. Durante una búsqueda rutinaria, se identificaron diversas vulnerabilidades que, de no ser mitigadas, podrían comprometer la confidencialidad, integridad y disponibilidad de los datos alojados en la plataforma.

Entre las principales vulnerabilidades detectadas destacan:

La exposición de archivos críticos , como bases de datos SQL y archivos de configuración con credenciales sensibles.

Prácticas de programación inseguras , como la falta de validación de entradas y el uso de contraseñas en texto plano.

Deficiencias en el manejo de permisos y autenticación , lo que podría permitir accesos no autorizados y escaladas de privilegios.

El análisis de riesgos determinó que varias de estas vulnerabilidades presentan un nivel de riesgo crítico, especialmente aquellas relacionadas con la seguridad de las credenciales y la protección de datos personales.

---

---

Para mitigar estos riesgos, se proponen recomendaciones detalladas que incluyen:

- La eliminación de archivos expuestos.
- La implementación de controles de autenticación y autorización más robustos.
- La adopción de buenas prácticas de seguridad en el desarrollo y configuración de la plataforma.

Es crucial actuar de manera proactiva para cerrar las brechas de seguridad detectadas. Se recomienda realizar una revisión urgente del sitio web y aplicar las medidas correctivas indicadas para garantizar la seguridad de la información y proteger tanto a los usuarios como a la propia entidad involucrada.

## 8. Contacto

Quedo a la completa disposición de cualquier departamento o persona encargada que desee contactar conmigo, para ello facilitaré mis datos de contacto:

Gmail: [maroldandh@gmail.com](mailto:maroldandh@gmail.com)

Teléfono: +34 722163479

LinkedIn: <https://www.linkedin.com/in/mardh>

---