

Proyecto Dual 1ºDAM



Alumno: José Manuel Sánchez Rosal.
Empresa: Evirom Soluciones en la Nube S.L.
Curso Escolar: 2024/2025

Índice.

Índice.....	2
1. Aclaraciones.....	2
2. Programación RA8.....	3
3. Programación RA9.....	8
4. Base de Datos RA3.....	11
5. Base de Datos RA7. (NOSQL).....	13
6. Entornos de Desarrollo RA3.....	17
7. Lenguaje de Marcas RA7.....	24

1. Aclaraciones.

El presente documento tiene como objetivo reflejar los Resultados de Aprendizaje (R.A.) de cada asignatura así como sus diferentes Criterios de Evaluación (C.E.) y la forma en la que se han abordado con el trabajo realizado por el Alumno en la Empresa.

Los apartados se han ido respondiendo bien con una pequeña descripción y alguna captura si se han tratado en la realización del proyecto, o se han respondido de forma teórica.

Cabe destacar que algunos se han respondido de forma conjunta.

Por otro lado se han utilizado las siguientes herramientas para el desarrollo del Proyecto:

- Documentación: Google Docs.
- FrontEnd: HTML5 y CSS3
- BackEnd: XAMPP Control Panel, Apache, MySQL, phpMyAdmin.
- IDE: Visual Studio Code.
- Control de versiones: GitLab y GitHub.

A continuación se muestran los diferentes RA's clasificados por Asignatura.

2. Programación RA8.

Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información:

a) Se han identificado las características de las bases de datos orientadas a objetos:

- **Encapsulamiento:** Los datos y los métodos que los manipulan se almacenan juntos, como en la programación orientada a objetos.
- **Herencia:** Permite que una clase hija herede atributos y métodos de una clase padre, facilitando la reutilización y organización de los datos.
- **Polimorfismo:** Un mismo método puede tener distintos comportamientos según el objeto que lo utilice.
- **Identidad de los objetos:** Cada objeto tiene un identificador único, lo que permite diferenciarlo de otros, incluso si tienen los mismos datos.
- **Persistencia:** Los objetos pueden mantenerse almacenados en la base de datos sin perder su estado, incluso después de que el programa deje de ejecutarse.
- **Integridad referencial sin claves foráneas:** Se mantienen relaciones directas entre objetos mediante referencias, sin necesidad de claves primarias y foráneas como en bases de datos relacionales.
- **Consultas basadas en objetos:** Se utilizan lenguajes de consulta como **OQL (Object Query Language)**, similares a SQL pero adaptados a objetos.
- **Soporte para transacciones:** Garantiza la confiabilidad de las operaciones mediante las propiedades **ACID** (Atomicidad, Consistencia, Aislamiento y Durabilidad).
- **Soporte para estructuras complejas:** Permiten almacenar objetos anidados, listas, árboles y grafos de manera natural sin necesidad de normalización.
- **Menor impedancia entre modelo de datos y programación:** No es necesario convertir los datos a formato tabular, ya que se almacenan tal como se manejan en la programación orientada a objetos.

Algunos ejemplos de bases de datos orientadas a objetos son: db4o, ObjectDB, Versant y GemStone.

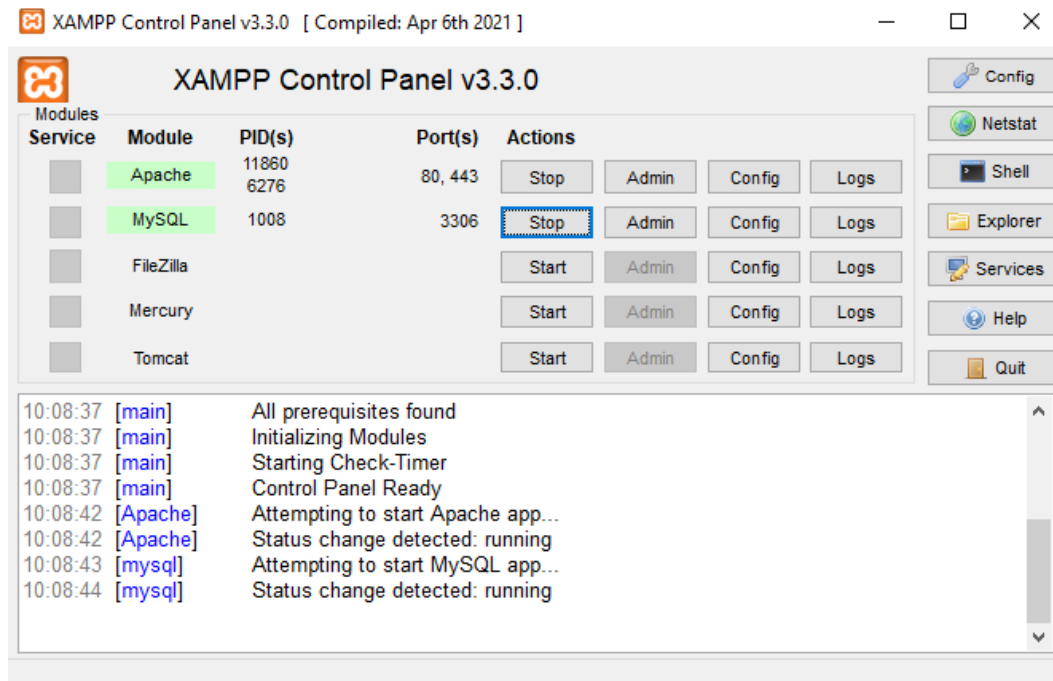
b) Se ha analizado su aplicación en el desarrollo de aplicaciones mediante lenguajes orientados a objetos:

Las bases de datos se integran perfectamente con lenguajes de programación orientados a objetos, permitiendo un desarrollo más fluido y natural de apps:

- **Menor impedancia entre código y datos:** No es necesario transformar los objetos en tablas (como en bases de datos relacionales), ya que se almacenan en la base de datos tal como se manejan en el código.
- **Uso directo de objetos:** Se pueden guardar y recuperar objetos sin necesidad de convertirlos a formatos como JSON o SQL.
- **Soporte para lenguajes orientados a objetos:** Se integran fácilmente con **Java, C#, Python y C++**, permitiendo almacenar objetos sin modificar su estructura.
- **Consultas mediante OQL (Object Query Language):** Se pueden hacer consultas avanzadas sobre los atributos de los objetos sin necesidad de utilizar SQL.
- **Compatibilidad con ORM (Object-Relational Mapping):** Herramientas como Hibernate (Java) o Entity Framework (C#) permiten usar bases de datos relacionales como si fueran orientadas a objetos.

c) Se han instalado sistemas gestores de bases de datos orientadas a objetos:

En nuestro caso se ha procedido a instalar la en control panel de XAMPP. **XAMPP** es un paquete de software que permite crear un **servidor local** en tu computadora para desarrollar y probar aplicaciones web. Incluye los componentes esenciales para ejecutar sitios web y aplicaciones dinámicas sin necesidad de un servidor en la nube.



d) Se han clasificado y analizado los distintos métodos soportados por los sistemas gestores para la gestión de la información almacenada:

Los SGBD permiten gestionar información con métodos de almacenamiento, manipulación, seguridad y optimización, según las necesidades del sistema:

1. Almacenamiento de datos:

- **Archivos:** CSV, JSON, XML.
- **Bases de datos relacionales (SQL):** MySQL, PostgreSQL.
- **Bases de datos NoSQL:** MongoDB, Redis.
- **Bases de datos orientadas a objetos:** db4o, ObjectDB.

2. Manipulación de datos

- **SQL y OQL** para consultas.
- **CRUD** (Create, Read, Update, Delete).
- **Procedimientos almacenados y triggers.**
- **ORMs** (Hibernate, Entity Framework).

3. Seguridad y control

- Autenticación y roles de usuario.
- Cifrado de datos.
- Copias de seguridad.

4. Optimización y rendimiento

- Índices, normalización y replicación para mejorar velocidad y disponibilidad.

e) Se han creado bases de datos y las estructuras necesarias para el almacenamiento de objetos:

Se ha creado una base de datos en phpmyadmin mediante lenguaje SQL:

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
<input type="checkbox"/> categorías	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8mb4_general_ci	16.0 KB	-
<input type="checkbox"/> fabricantes	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8mb4_general_ci	16.0 KB	-
<input type="checkbox"/> productos	Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	utf8mb4_general_ci	48.0 KB	-
3 tablas	Número de filas	10	InnoDB	utf8mb4_general_ci	80.0 KB	0 B

		id_categoria	nombre_categoria
<input type="checkbox"/>	Editar Copiar Borrar	1	Moviles
<input type="checkbox"/>	Editar Copiar Borrar	2	Tablets
<input type="checkbox"/>	Editar Copiar Borrar	3	Portatiles

		id_fabricante	nombre_fabricante	correo_fabricante	telefono_fabricante
<input type="checkbox"/>	Editar Copiar Borrar	1	Samsung	samsung@gmail.com	111111111
<input type="checkbox"/>	Editar Copiar Borrar	2	Xiaomi	xiaomi@gmail.com	222222222
<input type="checkbox"/>	Editar Copiar Borrar	3	Apple	apple@gmail.com	333333333

		id_producto	nombre_producto	descripcion_producto	cantidad_producto	id_fabricante	id_categoria
<input type="checkbox"/>	Editar Copiar Borrar	1	Samsung Galaxy S24 Ultra	Está equipado con una cuádruple cámara de hasta 20...	20	1	1
<input type="checkbox"/>	Editar Copiar Borrar	3	Apple MacBook Air 13"	CPU de 10 núcleos con 4 núcleos de rendimiento y 6...	52	3	3
<input type="checkbox"/>	Editar Copiar Borrar	17	Xiaomi Redmi Pad 15 PRO 5G	Gran pantalla 2,5K de 12,1" envolvente con cuidado...	66	2	2
<input type="checkbox"/>	Editar Copiar Borrar	20	Pavillion 3D	Portátil de 22" pantalla ultra 3D de alta veloci...	77	2	3

f) Se han programado aplicaciones que almacenen objetos en las bases de datos creadas:

```
$sql = "INSERT INTO productos (nombre_producto, descripcion_producto, cantidad_producto, id_fabricante, id_categoria)
VALUES (:nombre, :descripcion, :cantidad, :id_fabricante, :id_categoria)";
```

- g) Se han realizado programas para recuperar, actualizar y eliminar objetos de las bases de datos:

```
$sql = "UPDATE productos SET nombre_producto = :nombre, descripcion_producto = :descripcion,
cantidad_producto = :cantidad, id_fabricante = :id_fabricante, id_categoria = :id_categoria WHERE
id_producto = :id_producto";
```

```
$sql = "DELETE FROM productos WHERE id_producto = :id_producto";
$stmt = $pdo->prepare($sql);
if ($stmt->execute(['id_producto' => $id_producto])) {
    echo "Producto eliminado correctamente.";
    header("Location: index.php");
} else {
    echo "Error al eliminar el producto.";
}
```

- h) Se han realizado programas para almacenar y gestionar tipos de datos estructurados, compuestos y relacionados:

Se realiza CRUD completo para el index.php para poder crear un producto, actualizarlo y borrarlo mediante lenguaje php.

3. Programación RA9.

Gestiona la información almacenada en bases de datos relacionales manteniendo la integridad y consistencia de los datos:

- a) **Se han identificado las características y métodos de acceso a sistemas gestores de bases de datos:**

Mediante la instalación de xampp y ejecución de PHPmyadmin para ejecutar servidor local y poder crear una nueva BBDD:

El screenshot muestra la interfaz de phpMyAdmin. En la parte superior, hay una barra de navegación con pestañas como Estructura, SQL, Buscar, etc. Debajo, hay un campo de búsqueda con el texto 'Que contengan la palabra:'. En la parte inferior, hay una tabla con los detalles de las bases de datos y tablas.

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
<input type="checkbox"/> categorias	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8mb4_general_ci	16.0 KB	-
<input type="checkbox"/> fabricantes	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8mb4_general_ci	16.0 KB	-
<input type="checkbox"/> productos	Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	utf8mb4_general_ci	48.0 KB	-
3 tablas	Número de filas	10	InnoDB	utf8mb4_general_ci	80.0 KB	0 B

En la parte inferior, hay un botón 'Seleccionar todo' y un menú desplegable 'Para los elementos que están marcados:'.

- b) **Se han programado conexiones con bases de datos:**

Se ha programado en lenguaje php la conexión de un archivo a una base de datos en phpmyadmin:

```
$host = "localhost";
$dbname = "bbdd_catalogo_jmsr"; // ⚠ Asegúrate de que este es el nombre correcto en MySQL
$username = "root";
$password = "";

try {
    // Crear la conexión con PDO
    $pdo = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8", $username, $password);

    // Configurar PDO para que lance excepciones en caso de error
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    echo "Conexión establecida";
} catch (PDOException $e) {
    die("Error de conexión: " . $e->getMessage());
}
```

c) Se ha escrito código para almacenar información en bases de datos:

Mediante el INSERT en SQL introducimos un nuevo producto en nuestra tabla "productos":

✓ 1 fila insertada. (La consulta tardó 0,0004 segundos.)

```
INSERT INTO `productos`(`id_producto`, `nombre_producto`, `descripcion_producto`, `cantidad_producto`,
`id_fabricante`, `id_categoria`) VALUES ('20','Pavillion 3D','Portátil de 22" pantalla ultra 3D de alta
velocidad de transmisión de datos','20','1', '1');
```

[[Editar en línea](#)] [[Editar](#)] [[Crear código PHP](#)]

d) Se ha escrito código para recuperar y mostrar información almacenada en bases de datos:

```
// Obtener fabricantes y categorías para los formularios
$fabricantes = $pdo->query("SELECT * FROM fabricantes")->fetchAll(PDO::FETCH_ASSOC);
$categorias = $pdo->query("SELECT * FROM categorias")->fetchAll(PDO::FETCH_ASSOC);
```

localhost / 127.0.0.1 / bbdd_cat x CRUD catalogo x +

localhost/proyecto-jmsr/index.php

Conexión establecida

Lista de Productos

Nombre	Descripción	Cantidad	Fabricante	Categoría	Acciones
Samsung Galaxy S24 Ultra	Está equipado con una cuádruple cámara de hasta 200 MPX que garantiza fotos claras y detalladas, y una cámara frontal de 12 MPX para selfies perfectas. Su pantalla de 6,8" y la potencia de la red 5G hacen del Samsung Galaxy S24 Ultra una elección inigualable para los amantes de la tecnología.	20	Samsung	Moviles	Editar Eliminar
Apple MacBook Air 13"	CPU de 10 núcleos con 4 núcleos de rendimiento y 6 de eficiencia. GPU de 10 núcleos. Trazado de rayos acelerado por hardware. Neural Engine de 16 núcleos. 120 GB/s de ancho de banda de memoria.	52	Apple	Portatiles	Editar Eliminar
Xiaomi Redmi Pad 15 PRO 5G	Gran pantalla 2,5K de 12,1" envolvente con cuidado de la vista Da rienda suelta a tu imaginación Xiaomi HyperOS proporciona una experiencia fluida insignia La plataforma móvil Snapdragon® 7s Gen 2 ofrece un rendimiento eficiente Batería de gran capacidad de 10 000 mAh (typ)	66	Xiaomi	Tablets	Editar Eliminar
Pavillion 3D	Portátil de 22" pantalla ultra 3D de alta velocidad de transmisión de datos	77	Xiaomi	Portatiles	Editar Eliminar

Agregar Producto

Nombre:

Descripción:

Cantidad:

Fabricante:

Categoría:

- e) Se han efectuado borrados y modificaciones sobre la información almacenada:

```
// Eliminar un producto
if (isset($_GET['accion']) && $_GET['accion'] == 'eliminar') {
    $id_producto = $_GET['id_producto'];

    $sql = "DELETE FROM productos WHERE id_producto = :id_producto";
    $stmt = $pdo->prepare($sql);
    if ($stmt->execute(['id_producto' => $id_producto])) {
        echo "Producto eliminado correctamente.";
        header("Location: index.php");
    } else {
        echo "Error al eliminar el producto.";
    }
}
```

- f) Se han creado aplicaciones que muestran la información almacenada en bases de datos:
- g) Se han creado aplicaciones para gestionar la información presente en bases de datos:

4. Base de Datos RA3.

Consulta la información almacenada en una base de datos empleando asistentes, herramientas gráficas y el lenguaje de manipulación de datos:

- a) Se han identificado las herramientas y sentencias para realizar consultas:

```
// Obtener todos los productos
$productos = $pdo->query("SELECT * FROM productos")->fetchAll(PDO::FETCH_ASSOC);
```

- b) Se han realizado consultas simples sobre una tabla:

```
// Obtener fabricantes y categorías para los formularios
$fabricantes = $pdo->query("SELECT * FROM fabricantes")->fetchAll(PDO::FETCH_ASSOC);
$categorias = $pdo->query("SELECT * FROM categorias")->fetchAll(PDO::FETCH_ASSOC);
```

- c) Se han realizado consultas sobre el contenido de varias tablas mediante composiciones internas:

```
SELECT nombre_producto, nombre_fabricante, nombre_categoria FROM productos p INNER JOIN
fabricantes f ON p.id_fabricante = f.id_fabricante INNER JOIN categorias c ON
p.id_fabricante = c.id_categoria;
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo | Número de filas: 25 ▼ Filtrar filas: Ordenar según la

Opciones extra

nombre_producto	nombre_fabricante	nombre_categoria
Samsung Galaxy S24 Ultra	Samsung	Moviles
Xiaomi Redmi Pad Pro 5G	Xiaomi	Tablets
Apple MacBook Air 13	Apple	Portatiles

- d) Se han realizado consultas sobre el contenido de varias tablas mediante composiciones externas:

Usando JOIN para mostrar nombre_fabricante y categoría en lugar de ID en el CRUD:

```
todos los productos
$pdo->query("SELECT p.id_producto, p.nombre_producto, p.descripcion_producto, p.cantidad_producto, f.nombre_fabricante, c.nombre_categoria FROM productos p JOIN
f ON p.id_fabricante = f.id_fabricante JOIN categorias c ON p.id_categoria = c.id_categoria")->fetchAll(PDO::FETCH_ASSOC);
```

- e) Se han realizado consultas resumen:

Se ha realizado búsqueda de cantidad total de "productos" de nuestra tabla:

Su consulta se ejecutó con éxito.

```
SELECT COUNT(*) AS total_productos FROM productos;
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

Opciones extra

total_productos
3

Se busca cantidad total de la suma de todos nuestros productos:

✓ Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0003 segundos.)

```
SELECT SUM(cantidad_producto) AS total_stock FROM productos;
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo | Número de filas: 25 ▼ Filtrar filas:

Opciones extra

total_stock
152

f) Se han realizado consultas con subconsultas:

Vamos a realizar una subconsulta, por ejemplo, de mostrar el producto que más tengamos en stock:

```
SELECT nombre_producto FROM productos WHERE cantidad_producto = (SELECT MAX(cantidad_producto) FROM productos);
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo | Número de filas: 25 ▼ Filtrar filas:

Opciones extra

nombre_producto
Xiaomi Redmi Pad 15 PRO 5G

☐ Editar ☐ Copiar ☐ Borrar

g) Se han realizado consultas que implican múltiples selecciones:

Las consultas en SQL que implican **múltiples selecciones** pueden lograrse mediante el uso de Union y Join, y subconsultas (f). Estas técnicas permiten combinar datos de varias tablas o realizar selecciones avanzadas.

Método	Descripción
UNION	Combina resultados de múltiples consultas (sin duplicados).
JOIN	Une datos de varias tablas relacionadas.
Subconsultas	Permiten seleccionar datos en base a otra consulta anidada.
EXISTS	Filtra resultados basados en la existencia de datos en otra tabla.

h) Se han aplicado criterios de optimización de consultas:

- Evitar el uso del SELECT *, seleccionar únicamente las columnas necesarias para la búsqueda.
- Crear índices en Foreign Keys para que las búsquedas sean más rápidas.
- Usar EXISTS en lugar de IN para subconsultas.
- Uso de JOIN para subconsultas.
- Aplicar filtros WHERE en lugar de HAVING.
- Uso de LIMIT para reducir la carga de resultados a mostrar.

5. Base de Datos RA7. (NOSQL)

Gestiona la información almacenada en bases de datos no relacionales, evaluando y utilizando las posibilidades que proporciona el sistema gestor:

a) Se han caracterizado las bases de datos no relacionales:










b) Se han evaluado los principales tipos de bases de datos no relacionales:

1. **Modelo de datos flexible:** No requieren esquemas fijos, lo que permite almacenar datos estructurados, semiestructurados y no estructurados.
 2. **Escalabilidad horizontal:** Se diseñan para distribuir datos entre múltiples servidores, facilitando el escalamiento y el rendimiento en grandes volúmenes de información.
 3. **Alta disponibilidad y tolerancia a fallos:** Se enfocan en la replicación y distribución de datos para evitar puntos únicos de fallo.
 4. **Optimización para operaciones específicas:** Son eficientes en lectura/escritura de grandes volúmenes de datos, lo que las hace ideales para aplicaciones web, análisis de Big Data e IoT.
 5. **Diversidad de modelos de almacenamiento:** Existen diferentes tipos de bases de datos NoSQL según la estructura de los datos y las necesidades del sistema.
- Tipos de Bases de Datos NoSQL:
 - **Basadas en Documentos:**
 - i) Almacenan datos en documentos JSON, BSON o XML.
 - ii) Ejemplo: **MongoDB, CouchDB.**
 - iii) Uso: Aplicaciones web y sistemas con datos semiestructurados.
 - **Basadas en Clave-Valor:**

- iv) Guardan información en pares clave-valor, similar a los diccionarios en programación.
- v) Ejemplo: **Redis, DynamoDB.**
- vi) Uso: Caché de datos, sesiones de usuario y procesamiento en tiempo real.
- **Basadas en Columnas:**
 - vii) Organizan los datos en columnas en lugar de filas, optimizando consultas sobre grandes volúmenes de datos.
 - viii) Ejemplo: **Apache Cassandra, HBase.**
 - ix) Uso: Big Data, análisis de datos y registros masivos.
- **Basadas en Grafos:**
 - x) Diseñadas para modelar relaciones entre entidades a través de nodos y aristas.
 - xi) Ejemplo: **Neo4j, ArangoDB.**
 - xii) Uso: Redes sociales, recomendaciones y análisis de conexiones.

c) **Se han identificado los elementos utilizados en estas bases de datos no relacionales:**

En nuestro caso, los elementos que se han utilizado para guardar la información es a través de tablas (entidades) filas (registros) y columnas (atributos):

← T →				id_categoria	nombre_categoria
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	1	Moviles
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	2	Tablets
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	3	Portatiles

d) Se han identificado distintas formas de gestión de la información según el tipo de base de datos no relacionales:

Mediante tablas (entidades):

	id_fabricante	nombre_fabricante	correo_fabricante	telefono_fabricante
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	1	Samsung	samsung@gmail.com	111111111
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	2	Xiaomi	xiaomi@gmail.com	222222222
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	3	Apple	apple@gmail.com	333333333

e) Se han utilizado las herramientas del sistema gestor para la gestión de la información almacenada:

Se utiliza el lenguaje SQL para gestionar consultas sobre nuestra base de datos:

```
1 SELECT `id_fabricante`, `nombre_fabricante`, `correo_fabricante`, `telefono_fabricante`
2 FROM `fabricantes`
3 WHERE `nombre_fabricante` = 'Samsung';
```

1. Bases de Datos Basadas en Documentos:

- Documento: Unidad básica de almacenamiento, generalmente en formato JSON, BSON o XML.
- Colección: Conjunto de documentos similares (equivalente a una tabla en bases de datos relacionales).
- Clave: Identificador único dentro de un documento.
- Índices: Optimizan la búsqueda de documentos.
- Consultas Agregadas: Permiten realizar operaciones como filtrado, agrupación y transformación de datos.

```
{
  "_id": "123",
  "nombre": "Smartphone X",
  "fabricante": "TechCorp",
  "precio": 599.99,
  "especificaciones": {
    "pantalla": "6.5 pulgadas",
    "bateria": "4000mAh"
  }
}
```


2. Bases de Datos Basadas en Clave-Valor

- Clave: Identificador único para acceder a un valor.
- Valor: Puede ser un número, cadena, objeto o estructura compleja.
- Espacio de Claves: Conjunto de todas las claves almacenadas.
- TTL (Time-To-Live): Tiempo de vida de un dato antes de ser eliminado automáticamente.

```
SET usuario:1001 "Juan Pérez"
GET usuario:1001
```

3. Bases de Datos Basadas en Columnas

- Familia de Columnas: Agrupación de columnas relacionadas (similar a una tabla).
- Clave de Fila: Identificador único de una fila de datos.
- Supercolumna: Conjunto de columnas anidadas dentro de una familia de columnas.
- Nodo: Servidor que almacena una parte de los datos en un sistema distribuido.

```
CREATE TABLE productos (  
  id UUID PRIMARY KEY,  
  nombre TEXT,  
  precio DECIMAL,  
  categoria TEXT  
);
```

4. Bases de Datos Basadas en Grafos

- Nodo: Representa una entidad (por ejemplo, una persona o un producto).
- Arista: Representa una relación entre nodos.
- Etiqueta: Categoriza los nodos o relaciones.
- Propiedad: Atributos asociados a nodos y relaciones.

- Consulta con Lenguaje Cypher (en Neo4j):

```
CREATE (p:Persona {nombre: "Carlos", edad: 30})
CREATE (c:Ciudad {nombre: "Madrid"})
CREATE (p)-[:VIVE_EN]->(c)
```

6. Entornos de Desarrollo RA3.

Verifica el funcionamiento de programas diseñando y realizando pruebas.

a) Se han identificado los diferentes tipos de pruebas.

Las pruebas se pueden clasificar de diferentes maneras según su objetivo, nivel, automatización y metodología utilizada:

1. Según su Propósito

- Pruebas Funcionales: Verifican que el software cumple con los requisitos especificados.
- Pruebas No Funcionales: Evalúan aspectos como rendimiento, seguridad, accesibilidad y usabilidad.

2. Según la Etapa del Desarrollo

- Pruebas Unitarias: Se testean módulos o funciones individuales del código.
- Pruebas de Integración: Se evalúa la comunicación entre diferentes módulos del sistema.
- Pruebas del Sistema: Se verifica el comportamiento del software en su conjunto.
- Pruebas de Aceptación: Se realizan para validar con el usuario final antes del despliegue.

3. Según la Técnica Utilizada

- Caja Negra: Evalúan la funcionalidad sin conocer el código interno.
- Caja Blanca: Se revisa la lógica interna del código y su estructura.

- **Caja Gris:** Combina las técnicas de caja negra y caja blanca.

4. Según su Automatización

- **Pruebas Manuales:** Ejecutadas por testers sin herramientas automatizadas.
- **Pruebas Automatizadas:** Se emplean herramientas para realizar pruebas repetitivas.

Dependiendo del entorno de desarrollo y el lenguaje de programación utilizado, se pueden emplear diferentes herramientas para llevar a cabo las pruebas:

- **JUnit** (Java) → Para pruebas unitarias en aplicaciones Java.
- **Mockito** (Java) → Para pruebas de integración con mocks.
- **Espresso** (Android) → Para pruebas de UI en aplicaciones móviles.
- **Selenium** → Para pruebas de aplicaciones web automatizadas.
- **Postman** → Para probar API REST.
- **JMeter** → Para pruebas de rendimiento.

b) Se han definido casos de prueba.

En nuestro caso, hemos añadido a nuestra librería el Junit 5, y hemos realizado pruebas de funcionalidad de nuestra clase calculadora, y sus módulos suma, resta, producto y cociente, todos ellos pasados satisfactoriamente:

The screenshot shows an IDE with a project named 'JavaPoo'. The left sidebar shows the project structure, including a package named 'PruebasJunit' containing 'FuncionCalculador' and 'FuncionCalculadorTest'. The main editor displays the code for 'FuncionCalculadorTest.java'.

```

package PruebasJunit;
import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.Test;

public class FuncionCalculadorTest {

    @Test
    void testSuma() {
        FuncionCalculador calc = new FuncionCalculador( a: 5, b: 3);
        assertEquals( expected: 8, calc.sumar());
    }

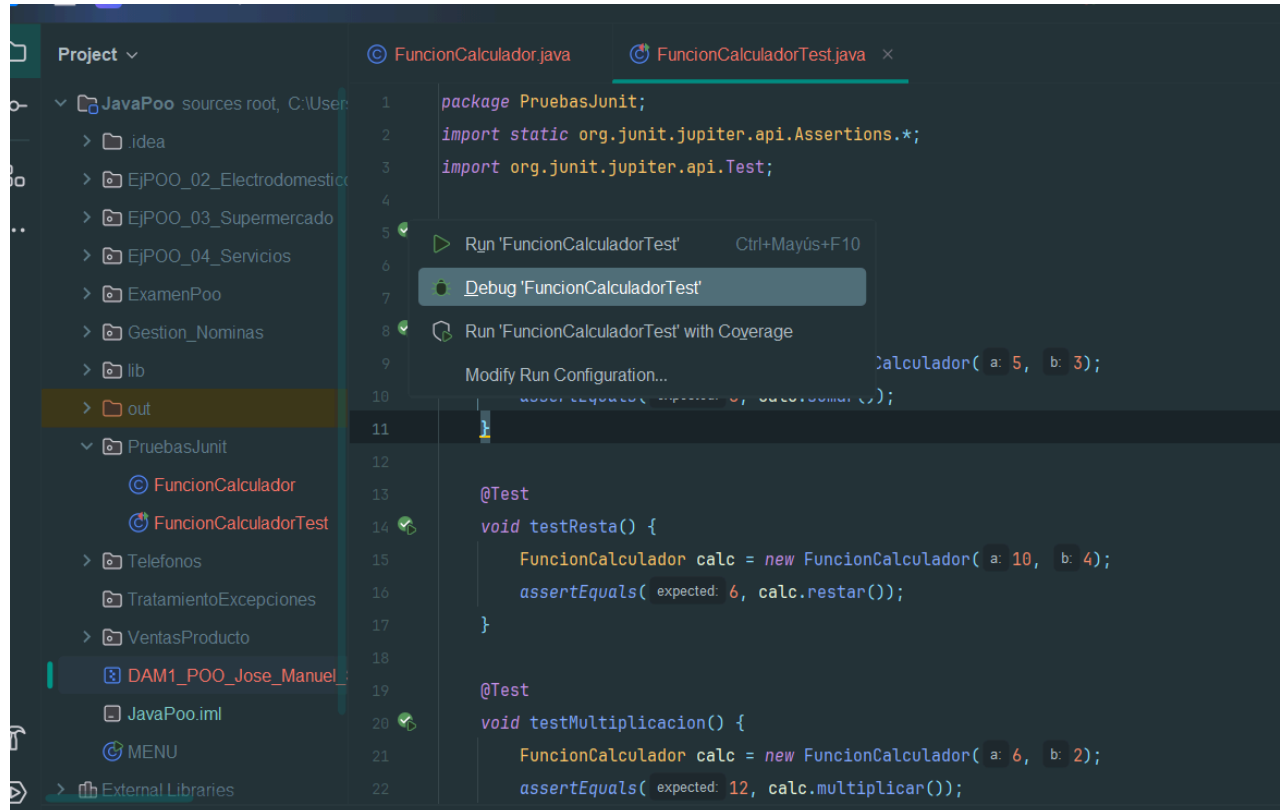
    @Test
    void testResta() {
        FuncionCalculador calc = new FuncionCalculador( a: 10, b: 4);
        assertEquals( expected: 6, calc.restar());
    }

    @Test
    void testMultiplicacion() {
        FuncionCalculador calc = new FuncionCalculador( a: 6, b: 2);
        assertEquals( expected: 12, calc.multiplicar());
    }
  
```

At the bottom, the 'Run' tab shows the execution of 'FuncionCalculadorTest.testDivisionPorCero'. The test passed successfully in 19 ms. The output console shows the command: 'C:\Users\José\.jdk\openjdk-23.0.2\bin\java.exe ...'.

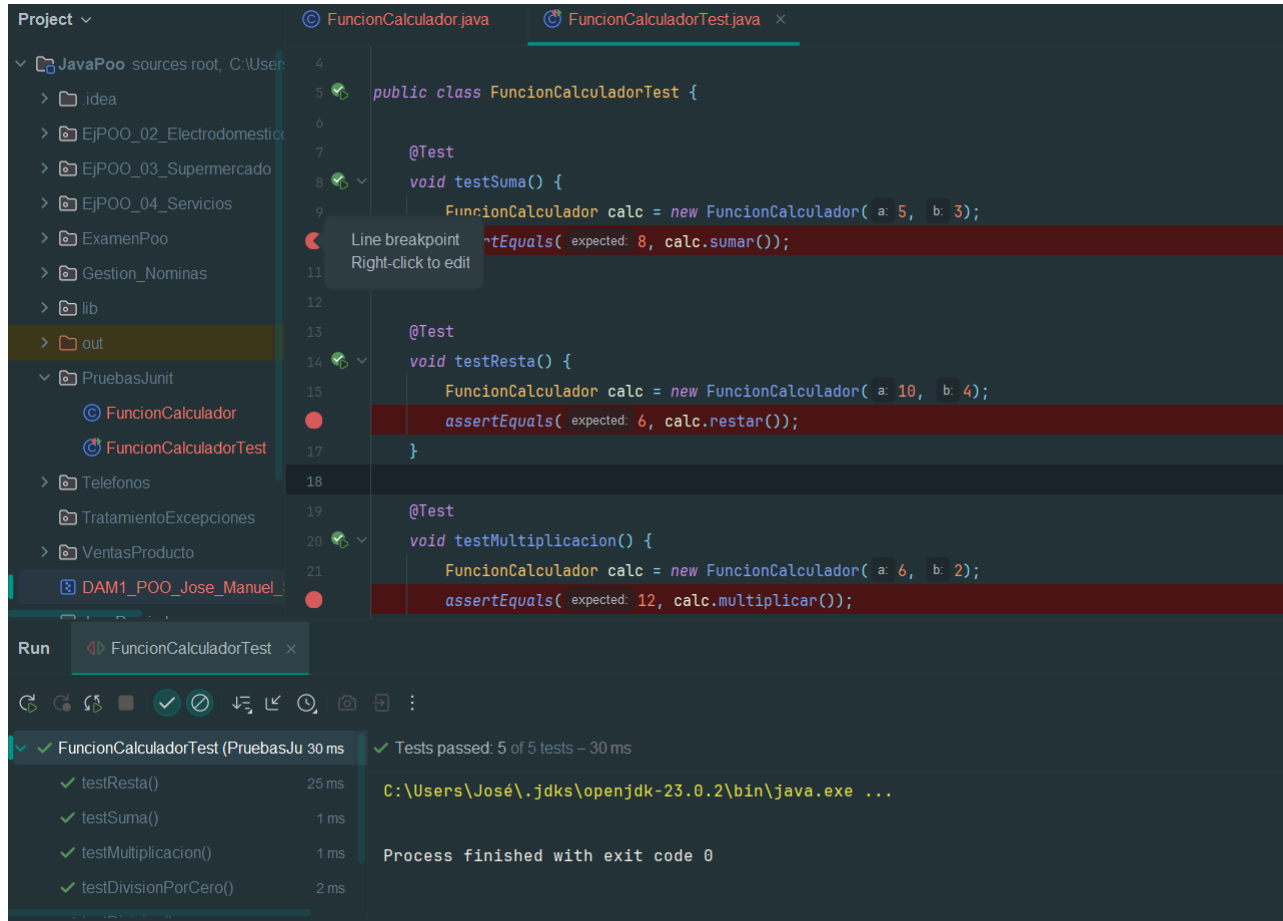
c) Se han identificado las herramientas de depuración y prueba de aplicaciones ofrecidas por el entorno de desarrollo:

Se utiliza el debugger del propio entorno IntelliJ IDEA para realizar búsqueda de fallos y errores en el código:



- d) Se han utilizado herramientas de depuración para definir puntos de ruptura y seguimiento:

Se añaden puntos de ruptura para saber en esa misma línea el valor y el comportamiento del programa:



- e) Se han utilizado las herramientas de depuración para examinar y modificar el comportamiento de un programa en tiempo de ejecución:

Se cambia valor en punto de ruptura en tiempo de ejecución y el programa nos lanza error de fallo de test en la función suma:

```

public class FuncionCalculadorTest {

    @Test
    void testSuma() {
        FuncionCalculador calc = new FuncionCalculador( a: 10, b: 3);
        assertEquals( expected: 8, calc.sumar());
    }
}

```

Run **FuncionCalculadorTest** x

Tests failed: 1, passed: 4 of 5 tests – 34 ms

Test	Duration	Status
testResta()	26 ms	Passed
testSuma()	4 ms	Failed
testMultiplicacion()	1 ms	Passed
testDivisionPorCero()	2 ms	Passed
testDivision()	1 ms	Passed

org.opentest4j.AssertionFailedError:
Expected :8
Actual :13
<Click to see difference>

- f) Se ha documentado el plan de pruebas.

No realizado.

- g) Se han efectuado pruebas unitarias de clases y funciones.

En apartados anteriores se han definido casos de prueba de funcionalidad de la clase calculadora.

- h) Se han efectuado pruebas de integración, de sistema y de aceptación. NO
i) Se han implementado pruebas automáticas. NO
j) Se han documentado las incidencias detectadas:

En mi caso, una de las incidencias detectadas era en el desarrollo del catálogo web, es que tenía dos directorios con el mismo nombre en diferentes ubicaciones. Entonces los cambios que se producían en el IDE Visual Studio Code en el archivo .css, no reflejaba los cambios.

k) Se han aplicado normas de calidad a los procedimientos de desarrollo de software.

Durante el desarrollo del software se han intentado seguir unas normas de calidad del producto:

ISO/IEC 25000 (SQuaRE):

- **Funcionalidad** (cumple con los requisitos).
- **Fiabilidad** (baja tasa de fallos).
- **Usabilidad** (fácil de usar).
- **Eficiencia** (buen rendimiento).
- **Mantenibilidad** (fácil de modificar).
- **Portabilidad** (se puede ejecutar en diferentes plataformas).

l) Se han realizado medidas de calidad sobre el software desarrollado.

Se realizan pruebas de funcionalidad y rendimiento a la clase calculadora.

m) Se han utilizado dobles de prueba para aislar los componentes durante las pruebas.

NO realizado.

7. Lenguaje de Marcas RA7.

Opera sistemas empresariales de gestión de información realizando tareas de importación, integración, aseguramiento y extracción de la información.

a) Se han identificado los principales sistemas de gestión empresarial.

Los principales sistemas de gestión empresarial de la información son:

- ERP (Enterprise Resource Planning): SAP, Odoo, Microsoft Dynamics 365.
- CRM (Customer Relationship Management): Salestore, HubSpot, Zoho CRM.
- SCM (Supply Chain Management): Oracle SCM, SAP SCM.
- BI: (Business Intelligence): Power BI, Tableau.

b) Se han reconocido las ventajas de los sistemas de gestión de información empresariales.

A continuación se muestran las principales ventajas de los mismos:

- Centralización de datos y procesos.
- Reducción de errores y duplicación de la información.
- Optimización en la toma de decisiones mediante herramientas de análisis de datos.
- Automatización de tareas y mejora considerable de la productividad.

c) Se han evaluado las características de las principales aplicaciones de gestión empresarial.

- Interfaz intuitiva y accesible.
- Capacidad e integración de las mismas con otras herramientas.
- Seguridad y control de accesos.
- Flexibilidad y personalización.

d) Se han instalado aplicaciones de gestión de la información empresarial:

Evisane es un software orientado a empresas que se dedican al control de plagas y sanidad ambiental. Desarrollado enteramente en la nube para que todos los usuarios puedan trabajar de forma simultánea en un entorno simple, rápido e intuitivo.

Se da de alta una nueva empresa:

Añadir nueva empresa

USUARIO: J.miguel25 EMAIL: miguelangel@outlook

CONFIRMA LA CONTRASEÑA DE TU CUENTA

DATOS EMPRESA

NOMBRE EMPRESA: NOMBRE COMERCIAL: NIF:

DIRECCIÓN: CP:

PAIS: España PROYINANCIA: POBLACIÓN:

WEB: TELÉFONO: MÓVIL:

EMAIL: miguelangel@outlook DIRECCIÓN: TIPO DE IMPUESTO: IVA

ENLACE REDES SOCIALES

FACEBOOK: TWITTER: GOOGLE:

LINKEDIN: INSTAGRAM: YOUTUBE:

USUARIOS: MÓDULOS: OTROS DATOS

DATOS USUARIO 1

NOMBRE: APELLIDOS: EMAIL:

D.N.I.: LUGAR Y FECHA DE CURSO DE LEGIONELLA:

DATOS USUARIO 2

NOMBRE: APELLIDOS: EMAIL:

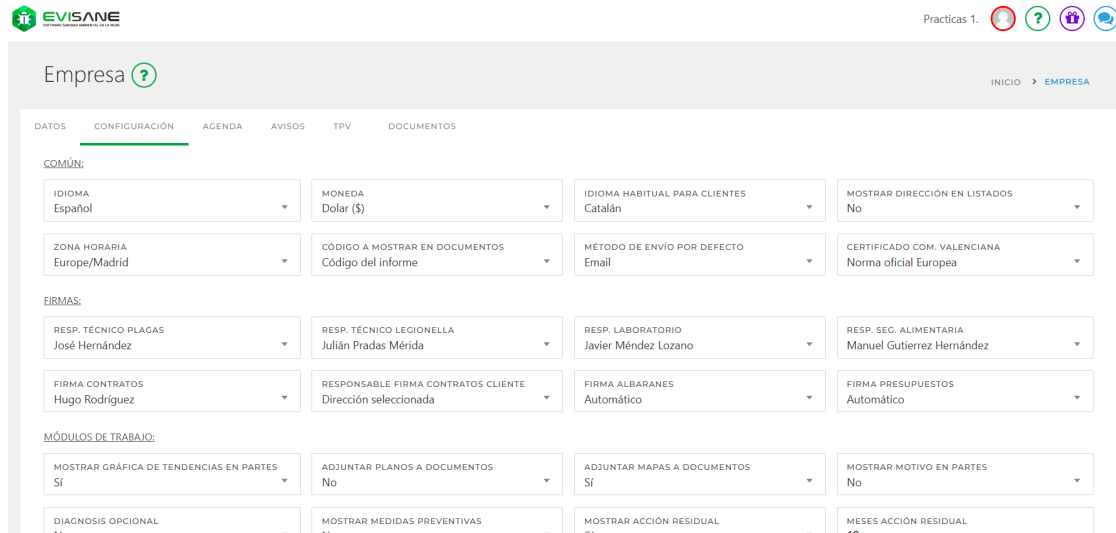
D.N.I.: LUGAR Y FECHA DE CURSO DE LEGIONELLA:

Añadir usuario

Crear Cancelar

e) Se han configurado y administrado las aplicaciones:

Modificamos la configuración desde la app web, apartado configuración:



Empresa ?

INICIO > EMPRESA

DATOS CONFIGURACIÓN AGENDA AVISOS TPV DOCUMENTOS

COMUN:

IDIOMA Español	MONEDA Dolar (\$)	IDIOMA HABITUAL PARA CLIENTES Catalán	MOSTRAR DIRECCIÓN EN LISTADOS No
ZONA HORARIA Europe/Madrid	CÓDIGO A MOSTRAR EN DOCUMENTOS Código del informe	MÉTODO DE ENVÍO POR DEFECTO Email	CERTIFICADO COM. VALENCIANA Norma oficial Europea

FIRMAS:

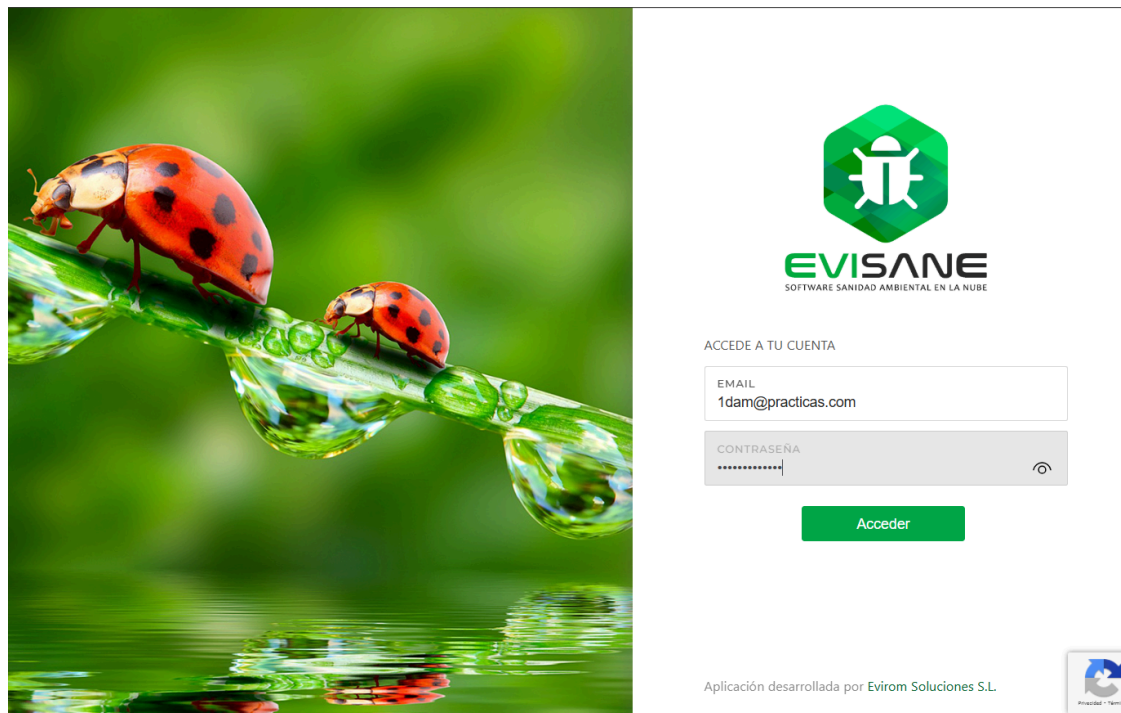
RESP. TÉCNICO PLAGAS José Hernández	RESP. TÉCNICO LEGIONELLA Julián Pradas Mérida	RESP. LABORATORIO Javier Méndez Lozano	RESP. SEG. ALIMENTARIA Manuel Gutierrez Hernández
FIRMA CONTRATOS Hugo Rodríguez	RESPONSABLE FIRMA CONTRATOS CLIENTE Dirección seleccionada	FIRMA ALBARANES Automático	FIRMA PRESUPUESTOS Automático

MÓDULOS DE TRABAJO:

MOSTRAR GRÁFICA DE TENDENCIAS EN PARTES Sí	ADJUNTAR PLANOS A DOCUMENTOS No	ADJUNTAR MAPAS A DOCUMENTOS Sí	MOSTRAR MOTIVO EN PARTES No
DIAGNOSIS OPCIONAL No	MOSTRAR MEDIDAS PREVENTIVAS Mostrar	MOSTRAR ACCIÓN RESIDUAL C	MESES ACCIÓN RESIDUAL 12

f) Se han establecido y verificado mecanismos de acceso seguro a la información.

Login acceso a usuarios:



EVISANE
SOFTWARE SANIDAD AMBIENTAL EN LA NUBE

ACCEDE A TU CUENTA

EMAIL
1dam@practicas.com

CONTRASEÑA

Acceder

Aplicación desarrollada por Evirom Soluciones S.L.

Privacidad | Términos

g) Se han generado informes:

Se realiza nuevo registro o reporte:

Añadir - Reportes

DATOS GENERALES

CÓDIGO

8

FECHA

18-03-2025

CLIENTE

Selecciona cliente

DIRECCIÓN

LIBRO

Selecciona plan / registros

TÍTULO DEL INFORME

Reporte de mediciones

FECHA INICIO

18-12-2024

FECHA FIN

18-03-2025

PERIODO

3 últimos meses

FECHA Y HORA

Deseada

MEDIDO POR

Cliente (Medición)

CONTROL

Todos

CÓDIGO

No hay valores

PARÁMETRO

Todos

USUARIO

Todos

RESULTADOS

Dentro de máximos y mínimos...

SEDE ASOCIADA

Fincas Pérez SL (Arteixo)

FIRMADO POR

Practicas 1 DAM

DENOMINACIÓN

Origen

RESPONSABLE

JM

TELÉFONO

465768798

DETALLES (OPCIONAL)

↶ ↷

Párrafo

Fuente de sist...

12pt

B *I* U ~~S~~

≡

≡

≡

≡

✍

📎

📄

🔗

🔗

🔗

...

h) Se han realizado procedimientos de extracción de información para su tratamiento e incorporación a diversos sistemas:


Se genera documento Excel con la información requerida:

[illegible]

i) Se han elaborado documentos relativos a la explotación de la aplicación:

Se elabora un nuevo registro de actividad:

Modificar - Libro de registro

CÓDIGO 202400028	FECHA INICIO 06-09-2024	FECHA FIN 05-09-2025	Duración: 1 año Cambiar	DESCRIPCIÓN Mantenimiento de instalaciones de Hotel X
CLIENTE Mantecados El Jameo (Mantecados El Jameo S.L.)	DIRECCIÓN [Casa cultura] Avinguda de Vila-Real 12, 12006 Castelló de la Plat...	SEDE ASOCIADA BioControl S.L. (Cañada Rosal)	CONTRATO ASOCIADO Ninguno	
<p>PLANOS</p> <div><p>PRUEBA</p><p>Ver plano</p></div> <p>Crear nuevo plano</p>				

Planificador

Filtro - Medido por	X	Filtro - Técnico asignado	X	Filtro - Tipo control	X	Añadir controles
---------------------	---	---------------------------	---	-----------------------	---	------------------