



Proyecto ALM System

Sumario

Proyecto ALM System.....	1
Descripción del desarrollo.....	2
Aspectos destacados.....	2
Tecnologías utilizadas.....	2
Pruebas.....	2
Optimización del rendimiento.....	2
Problemas identificados y soluciones aplicadas.....	3
Base de datos.....	4
Errores de compatibilidad.....	4
Funcionalidades pendientes y mejoras futuras.....	4
Guía de instalación y configuración.....	4
Requisitos previos.....	4
Instalación automática.....	5
Instalación manual.....	5
Pruebas y verificación.....	6
Pruebas unitarias.....	6
Pruebas de integración.....	7
Consideraciones adicionales a la prueba de integración.....	8
Despliegue en producción.....	8
Observaciones finales.....	8

Descripción del desarrollo

El presente documento detalla el proceso de desarrollo del proyecto **ALM System**, abarcando las tecnologías empleadas, las principales dificultades encontradas, las soluciones adoptadas y las posibles mejoras futuras. Asimismo, se proporciona una guía de instalación, pruebas y despliegue para facilitar su implementación en diferentes entornos.

Aspectos destacados

- **Tecnologías utilizadas**
 - **Frontend:** HTML5, CSS3, Bootstrap 5.3, Font Awesome 5, Google Fonts.
 - **Backend:** PHP con arquitectura **Modelo-Vista-Controlador (MVC)**.
 - **Cliente-Servidor:** Enrutador propio que gestiona las rutas hacia los controladores correspondientes.
 - **Generación de Documentos:** mPDF para generación dinámica de archivos PDF.
- **Pruebas**
 - Implementación de pruebas unitarias utilizando **PHPUnit**.
- **Optimización del rendimiento**
 - Evaluación de tiempos de carga y eficiencia mediante **GTmetrix** y **PageSpeed Insights**.

GTmetrix



PageSpeed Insights

Informe del 20 abr 2025, 14:05:43

http://bautistasebastiao.ct.ws/PROYECTO/Tarea03_4

Analizar

Móvil

Ordenador

Mostrando resultados de la URL: http://bautistasebastiao.ct.ws/PROYECTO/Tarea03_4/

[Analizar con la URL original](#)



Descubre lo que experimentan tus usuarios reales

No hay datos



Diagnostica problemas de rendimiento

98

Rendimiento

72

Accesibilidad

79

Prácticas
recomendadas

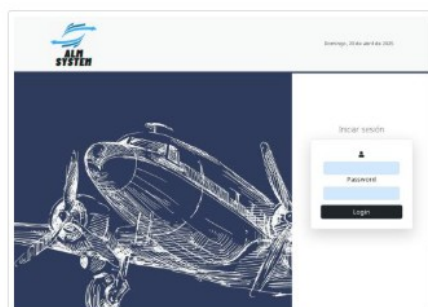
82

SEO

98

Rendimiento

Los valores son estimaciones y pueden variar. La [puntuación del rendimiento se calcula](#) directamente a partir de estas métricas. [Ver calculadora.](#)



Problemas identificados y soluciones aplicadas

- **Redundancia de datos en distintas tablas:**
 - Se soluciona mediante una tabla intermedia “*solicitud_material*” que gestiona las relaciones entre solicitudes, materiales y rutas de transporte.
- **Eliminación de registros huérfanos:**
 - Implementación de un **TRIGGER** en la base de datos para eliminar automáticamente las solicitudes que carecen de materiales asociados. Esta implementación no se puede llevar a cabo en producción debido a las limitaciones del hosting InfinityFree.

- **Consumo de recursos al utilizar API REST (OSRM):**
 - Para el cálculo óptimo de distancias y tiempos de rutas se ha recurrido a la API pública de OSRM que nos ofrece la distancia por carretera entre dos coordenadas. Aunque el uso de la API puede ralentizar el proceso en comparación con el uso de variables locales, se priorizó la obtención de datos actualizados en tiempo real.

Base de datos

- La base de datos fue diseñada para evitar redundancia asegurando la integridad referencial. Las relaciones entre las tablas permiten gestionar las solicitudes de materiales y su correspondiente transporte.
- El archivo “*BaseDatos.sql*” proporciona el esquema y los datos de muestra necesarios para la correcta inicialización del sistema.

Errores de compatibilidad

- Se identificaron diferencias de comportamiento entre versiones de PHP. Para asegurar la estabilidad, se fijaron versiones específicas de las dependencias en el archivo “*composer.json*”, garantizando así la compatibilidad total con PHP 8.2.4.

Funcionalidades pendientes y mejoras futuras

- **Funcionalidades en desarrollo:**
 - Implementación de un sistema de solicitudes basado en criterios de **prioridad**, donde se gestionaría la rapidez de la ruta en función de la urgencia.
- **Mejoras previstas:**
 - Creación de una tabla de vehículos disponibles, incorporando atributos como consumo de combustible, capacidad de carga y velocidad.
 - Optimización de la asignación de materiales en función del vehículo más apropiado y la urgencia de la solicitud.

Guía de instalación y configuración

Requisitos previos

- **Software:**
 - **XAMPP 8.2.4+**, que debe incluir Apache, PHP, MariaDB (MySQL) y phpMyAdmin.
- **Extensiones PHP necesarias:**
 - **IMPORTANTE:** Dentro del servidor **Apache** de XAMPP, es necesario habilitar las siguientes extensiones en el fichero **php.ini**:

curl, fileinfo, gd, intl, mbstring, mysqli, pdo_mysql.

- **Composer** para la gestión de dependencias.
- **Git** para descarga del proyecto.
- **Dependencias adicionales:**
 - **mPDF** para la generación de informes en PDF.

Instalación automática

Existen dos ficheros ejecutables para Windows y Linux respectivamente, que permiten descargar el proyecto completo desde GitHub, lo mueve al directorio *htdocs* de XAMPP y crea la base de datos con datos de muestra.

Sistema **Windows**:

- Doble click sobre el fichero *install_win.bat*.

Sistema **Linux**:

- Dar permisos de ejecución al fichero *install_linux.sh*:

```
chmod +x install_linux.sh
```

- Ejecutar el fichero:

```
./install_linux.sh
```

Instalación manual

1. Clonar el repositorio:

```
git clone https://github.com/jbauseb/PROYECTO.git
```

2. Mover el proyecto a la carpeta **htdocs** de XAMPP.
3. Otorgar permisos de lectura, escritura y ejecución a todos los ficheros y carpetas, si fuera necesario.
4. Acceder a la carpeta “*PROYECTO/Tarea03_4*” e instalar las dependencias:

```
composer install
```
5. Configurar “*config.php*” para adaptarlo al entorno deseado (local o hosting). Por defecto está habilitado el código para despliegue local.
6. Iniciar los servicios de Apache y MySQL desde XAMPP.
7. Importar el archivo *BaseDatos.sql* mediante phpMyAdmin.
8. Acceder a la aplicación:

```
http://localhost/PROYECTO/Tarea03\_4
```

Pruebas y verificación

- Acceder al sistema en http://localhost/PROYECTO/Tarea03_4 para validar su correcto funcionamiento.

Ejemplos de acceso:

Usuario	Password	Rol
Anaopez	1234	Administrador
Elisaramos	1234	Gestor
Marioramirez	1234	Técnico

- Todos los usuarios pueden consultarse en la base de datos, dentro de la tabla “empleado”.

Para testear las pruebas unitarias y de integración seguiremos estos pasos:

1. Mediante phpMyAdmin crearemos una base de datos de prueba, que se encuentra en el fichero *tests/BaseDatosTest.sql*, la cual contiene los mismos datos y relaciones que la original.
2. Instalar PHPUnit (si no se dispone)
composer require --dev phpunit/phpunit
3. Ejecutar los tests desde el directorio *Tarea03_4*:

vendor/bin/phpunit tests/Unit

vendor/bin/phpunit tests/Integration

Pruebas unitarias

Estas pruebas se han desarrollado sobre los cuatro modelos del proyecto: *MaterialModelTest*, *RequestModelTest*, *RouteModelTest* y *UserModelTest*, con los siguientes resultados:

MaterialModelTest

Prueba	Tipo	Entradas	Salida esperada	Resultado
testGetAllMaterials	Unitario	---	Lista de materiales	No vacío
testSearchMaterialEncontrado	Unitario	id_material = 6 + datos correctos	Material encontrado	No vacío
testSearchMaterialNoEncontrado	Unitario	id_material = -1	Material no encontrado	Vacío
testSelectMaterialExistente	Unitario	id_material = 1	Material recuperado	No vacío
testInsertMaterial	Unitario	pn = 'P5432' + datos correctos	Inserción correcta	True
testDeleteMaterial	Unitario	id_material = 16	Borrado correcto	True
testUpdateMaterial	Unitario	id_material = 20 + datos correctos	Actualización correcta	True

RequestModelTest

Prueba	Tipo	Entradas	Salida esperada	Resultado
testGetAllRequests	Unitario	---	Lista de solicitudes	No vacío
testAddMaterialToRequest	Unitario	id_solicitud = 1 + datos correctos	Inserción correcta	True
testUpdateMaterialToRequest	Unitario	id_solicitud = 1 + datos correctos	Actualización correcta	True
testGetRequestDetails	Unitario	id_solicitud = -1	Solicitud no encontrada	Vacío
testFinalizaRequest	Unitario	id_solicitud = 1	Finalización correcta	True
testDeleteRequest	Unitario	id_solicitud = 1	Borrado correcto	True

RouteModelTest

Prueba	Tipo	Entradas	Salida esperada	Resultado
testGetAllRoutes	Unitario	---	Al inicio, sin rutas	Vacío
testInsertRoute	Unitario	Origen = Sevilla Destino = Zaragoza	Id nueva ruta	No vacío
testFinalizeRoute	Unitario	id_ruta = 1	Finalización correcta	True

UserModelTest

Prueba	Tipo	Entradas	Salida esperada	Resultado
testGetAllUsers	Unitario	---	Lista de empleados	No vacío
TestInsertUser	Unitario	nombre = null (dato incorrecto)	Error de usuario	Nulo
TestDeleteUser	Unitario	id_usuario = 1	Eliminación correcta	True

Pruebas de integración

Se ha desarrollado una **prueba de integración** para el sistema, donde se simulan distintos escenarios completos que combinan todas las entidades (usuarios, materiales, solicitudes y rutas). El objetivo es asegurar que los **modelos** y las **relaciones entre ellos** funcionan correctamente.

Prueba	Tipo	Entradas	Salida esperada
testCrearSolicitudConMateriales	Integración	Nuevo usuario. Nuevo material. Nueva solicitud asociada al usuario. Se añade material a la solicitud en la cantidad de 10	Nombre del empleado es correcto. La solicitud tiene asignado un material. La cantidad de material es la indicada
testCrearRutaYCambiarEstadoSolicitud	Integración	Nuevo usuario. Nuevo material. Nueva solicitud en estado “pendiente” por defecto. Nueva ruta “en tránsito”. Se añade material y ruta a la solicitud.	El estado de la solicitud es igual a “en tránsito”.
testEliminaMaterialDisparaTrigger EliminaSolicitud	Integración	Nuevo usuario. Nuevo material. Nueva solicitud. id_ruta = 1. Se añade material y ruta a la solicitud. Cantidad de material de actualiza a 0 (se elimina de	Solicitud eliminada cuando no tiene materiales (vacío)

		la solicitud).	
testSolicitudReduceStockMaterial	Integración	Nuevo usuario. Nuevo material. Nueva solicitud. id_ruta = 1. Se añade material y ruta a la solicitud. El stock del material se reduce a 10.	El nuevo stock del material es igual a 10.
testFinalizarSolicitud	Integración	Nuevo usuario. Nuevo material. Nueva solicitud. id_ruta = 1. Se añade material y ruta a la solicitud. El estado de la solicitud se pasa a "finalizada".	El estado de la solicitud es igual a "finalizada". La fecha de finalización no es nulo. La hora de finalización no es nulo.

Consideraciones adicionales a la prueba de integración

- Cada prueba abre una **transacción** al comenzar y realiza un **rollback** al terminar, para asegurar que no quedan datos persistentes en la base de datos de pruebas.
- Para garantizar una prueba de integración completa, se utilizan los métodos reales de los **modelos** (MaterialModel, RequestModel, RouteModel, UserModel).
- Los datos usados en las pruebas son **ficticios** para no depender de ningún estado previo de la base de datos.

Despliegue en producción

- El proyecto fue desplegado en el servicio de hosting gratuito **InfinityFree**.
- La carga de archivos se realizó mediante **Filezilla 3.66.5**.
- La base de datos fue creada utilizando el fichero *BaseDatos.sql* a través de phpMyAdmin de InfinityFree.
- El archivo *config.php* fue ajustado para el entorno de producción.
- Acceso al proyecto desplegado:

http://bautistasebastiao.ct.ws/PROYECTO/Tarea03_4/

Observaciones finales

La implementación de **ALM System** se diseñó buscando modularidad del código, eficiencia en la gestión de recursos y facilidad de mantenimiento gracias al patrón Modelo-Vista-Controlador.