

Proyecto Integrado DAW Curso 2024-2025

Aventureros Natos

Manuel Roldán Pérez

Proyecto Integrado Ciclo Superior Desarrollo de Aplicaciones Web

Departamento de Informática – IES Velázquez

Diciembre 2025

IES Velázquez – Departamento de Informática

ÍNDICE

1. Introducción
2. Estudio de Viabilidad
 - 2.1 Sistema Actual
 - 2.2 Sistema Nuevo
 - 2.3 Requisitos del Sistema
 - 2.4 Descripción de la Solución
 - 2.5 Planificación del Proyecto
 - 2.6 Estudio de Costes
3. Análisis del Sistema
 - 3.1 Entorno Tecnológico
 - 3.2 Modelado de Datos
 - 3.3 Identificación de Usuarios
 - 3.4 Subsistemas
 - 3.5 Requisitos por Subsistema
 - 3.6 Diagramas de Análisis
 - 3.7 Interfaces de Usuario
4. Construcción del Sistema
5. Conclusiones
6. Glosario
7. Bibliografía

1. Introducción

Este proyecto, denominado **Aventureros Natos**, consiste en la creación de una plataforma web destinada a la gestión de **ofertas de actividades, excursiones, expediciones, eventos y talleres**, disponibles para distintos tipos de usuarios.

La plataforma permite:

- Publicación de ofertas por parte de ofertantes.
- Visado por parte de gestores.
- Inscripción por parte de demandantes.
- Administración completa por parte de administradores.

El sistema está desarrollado con:

- **PHP** (lógica de servidor)
- **MySQL/MariaDB** (persistencia de datos)
- **HTML5, CSS3 y Bootstrap** (interfaz)
- **Git/GitHub** (control de versiones)
- **VirtualBox + Linux Mint + Apache + MySQL + PHP** (entorno de pruebas)
- **InfinityFree** (hosting público mediante FTP)

Este documento recoge la planificación, análisis, diseño y construcción del sistema.

2. Estudio de Viabilidad

2.1 Descripción del Sistema Actual

Antes del desarrollo del proyecto, **no existía un sistema centralizado** para gestionar actividades turísticas o recreativas bajo roles diferenciados.

Los principales problemas del sistema actual (manual o inexistente) son:

- Falta de un repositorio común de actividades.
- No existe un control del aforo.
- La validación de ofertas es manual.
- No hay trazabilidad ni historial de inscripciones.
- No existe distinción de roles ni permisos.

2.2 Descripción del Sistema Nuevo

El nuevo sistema implementa:

- **Gestión completa de usuarios** con perfiles: administrador, gestor, ofertante, demandante.
- **Publicación de ofertas** (con imágenes y descripciones).
- **Validación de ofertas** por parte de gestores.
- **Inscripción automática** respetando el aforo.
- **Autenticación por roles**.
- **Sistema de correo interno simulado**, generando archivos en /acceso/emails/.
- **Recuperación y verificación de cuenta mediante tokens únicos**.
- **Control de que la nueva contraseña no coincida con la anterior**.
- **Despliegue local y en hosting** mediante archivo variables.php configurable.

2.3 Identificación de Requisitos del Sistema

2.3.1 Requisitos de Información

Tabla: usuarios

- **Descripción:** Almacena los datos de los usuarios registrados en la plataforma.
- **Campos:**
 - id (PK, int, obligatorio)
 - nombre (varchar, obligatorio)
 - email (varchar, obligatorio)
 - perfil_id (FK, obligatorio)
 - password (varchar, obligatorio)
 - activo (bool, obligatorio)
 - token (varchar 60)
 - created_at, updated_at
- **Observaciones:** Contraseñas cifradas con password_hash.

Tabla: perfiles

- **Descripción:** Lista de roles del sistema.
- **Campos:** id, perfil, created_at, updated_at.

Tabla: ofertas

- **Descripción:** Actividades publicadas por ofertantes.
- **Campos:** id, usuario_id, categoria_id, nombre, descripcion, fecha_actividad, aforo, visada, timestamps.

Tabla: categorias

- **Descripción:** Clasifica las ofertas por tipo.
- **Campos:** id, categoria, timestamps

Tabla: solicitudes

- **Descripción:** Inscripciones de usuarios a ofertas.
- **PK compuesta:** oferta_id + usuario_id
- **Campos adicionales:** fecha_solicitud, timestamps

2.3.2 Requisitos Funcionales

RF01 – Registro de usuario
RF02 – Verificación de email
RF03 – Login y autenticación por rol
RF04 – Gestión de ofertas por ofertantes
RF05 – Validación/visado de ofertas por gestores
RF06 – Incripción a ofertas por demandantes
RF07 – Panel de administración de gestión de usuarios
RF08 – Edición del perfil y cambio de contraseña
RF09 – Recuperación de contraseña con token
RF10 – Filtro de ofertas por categoría, fecha, aforo

2.3.3 Otros Requisitos

- Acceso responsive desde móviles.
- Hosting gratuito funcional.
- Base de datos autosincronizable.
- Seguridad mínima:
 - tokens,
 - password_hash,
 - roles protegidos,
 - verificación por email.

2.4 Descripción de la Solución

La solución consiste en una aplicación web modular construida en PHP con vistas independientes, utilizando Bootstrap para la maquetación y MySQL para persistencia. Toda la lógica de conexión se centraliza en variables.php, lo que permite ejecutar el proyecto tanto en local como en hosting.

2.5 Planificación del Proyecto

2.5.1 Equipo de trabajo

- **Manuel Roldán Pérez** – Desarrollador único
- Funciones: análisis, backend, frontend, BD, servidor, despliegue, documentación.

2.5.2 Planificación temporal

- Análisis: 2 semanas
- Diseño BD: 1 semana
- Desarrollo backend: 4 semanas
- Desarrollo frontend: 3 semanas
- Integración + seguridad: 2 semanas
- Pruebas: 2 semanas
- Despliegue: 1 semana

2.6 Estudio de Costes

Entorno gratuito utilizado:

- VirtualBox + Linux Mint (0€)
- InfinityFree hosting (0€)
- MySQL + PHP + Apache gratuitos
- Editor VS Code (0€)
- GitHub (0€)

Total coste económico: **0€**

Coste en horas estimadas: **≈ 160 horas**

3. Análisis del Sistema

3.1 Entorno Tecnológico

Desarrollo

- SO: Windows 11
- VM Linux Mint con Apache + PHP + MySQL
- Editor: Visual Studio Code
- Servidor local: XAMPP / Live Server
- Gestores BD: phpMyAdmin / Adminer

Explotación

- Hosting: InfinityFree
- FTP: FileZilla
- BD: MySQL remoto
- PHP: v7.4+

Archivo variables.php permite intercambiar:

- host local: localhost
- host remoto: sql204.infinityfree.com

3.2 Modelado de Datos

3.2.1 Modelo Entidad–Relación (descripción textual)

Entidades:

- Usuario (1:N) Ofertas
- Usuario (N:N) Ofertas mediante Solicitudes
- Perfiles (1:N) Usuarios
- Categorías (1:N) Ofertas

3.2.2 Esquema Relacional

(Incluye todas las tablas ya enviadas en tu SQL. Lo incluiré íntegro si lo deseas en un apéndice.)

3.2.3 Datos de prueba

Documentados en el SQL recibido.

3.3 Identificación de Usuarios

- **Administrador:** Acceso total con CRUD de cualquier tabla incluido.
- **Gestor:** Visado de ofertas.
- **Ofertante:** Publica actividades.
- **Demandante:** Se inscribe a actividades.
- **Usuario sin registrar:** Solo consulta landing page.

3.4 Subsistemas

1. Subsistema de Autenticación.
2. Subsistema de Gestión de Ofertas.
3. Subsistema de Inscripciones.
4. Subsistema de Administración.

3.5 Requisitos por Subsistema

A continuación se detallan los requisitos funcionales específicos que debe cumplir cada uno de los subsistemas definidos en el análisis del sistema.

3.5.1. Subsistema de Autenticación

Objetivo

Permitir el acceso seguro y controlado al sistema mediante registro, verificación, login y recuperación de credenciales.

Requisitos

SA-01. Registro de usuario

El sistema permitirá registrar nuevos usuarios mediante formulario con validación de campos obligatorios.

SA-02. Verificación mediante email

Tras el registro, se enviará un email simulado con un enlace que incluye un token único para activar la cuenta.

SA-03. Inicio de sesión

El sistema permitirá iniciar sesión introduciendo email y contraseña, verificando la existencia del usuario y su activación.

SA-04. Autenticación por perfil

Cada usuario visualizará únicamente las funcionalidades propias de su rol (administrador, gestor, ofertante o demandante).

SA-05. Cifrado de contraseñas

Todas las contraseñas se almacenarán cifradas mediante password_hash().

SA-06. Recuperación de contraseña

El sistema permitirá solicitar una recuperación mediante email simulado con un token de un solo uso.

SA-07. Cambio de contraseña

El usuario podrá cambiar su contraseña siempre que no coincida con la almacenada previamente.

3.5.2 Subsistema de Gestión de Ofertas

Objetivo

Permitir a los ofertantes publicar actividades y a los gestores controlar su visado.

Requisitos

SO-01. Creación de ofertas

Los ofertantes podrán crear nuevas ofertas introduciendo nombre, categoría, descripción, fecha y aforo.

SO-02. Edición de ofertas

El sistema permitirá modificar ofertas mientras no hayan sido visadas.

SO-03. Visado de ofertas

Los gestores podrán marcar las ofertas como “visadas”, permitiendo su visualización pública.

SO-04. Eliminación de ofertas

Un ofertante podrá eliminar sus ofertas siempre que no cuenten con inscripciones.

SO-05. Listado y filtrado de ofertas

Cualquier usuario podrá filtrar ofertas por categoría, fecha o disponibilidad.

SO-06. Control de aforo

El sistema limitará las inscripciones al número máximo definido por el ofertante.

3.5.3 Subsistema de Inscripciones

Objetivo

Gestionar las solicitudes de participación de demandantes en las actividades.

Requisitos

SI-01. Inscripción en una oferta

El demandante podrá inscribirse en una oferta visada y con aforo disponible.

SI-02. Prevención de inscripciones duplicadas

El sistema no permitirá que un usuario se inscriba más de una vez a la misma oferta (PK compuesta).

SI-03. Visualización de inscripciones propias

El demandante podrá consultar todas sus solicitudes realizadas.

SI-04. Cancelación de solicitudes

El demandante podrá cancelar su inscripción mientras el evento no haya pasado.

SI-05. Control de aforo restante

El sistema actualizará el número de plazas restantes en tiempo real.

3.5.4. Subsistema de Administración

Objetivo

Proveer un panel de control íntegro para el administrador del sistema.

Requisitos

SA-Admin-01. Gestión completa de usuarios

El administrador podrá crear, editar, activar/desactivar y eliminar usuarios.

SA-Admin-02. Gestión de perfiles

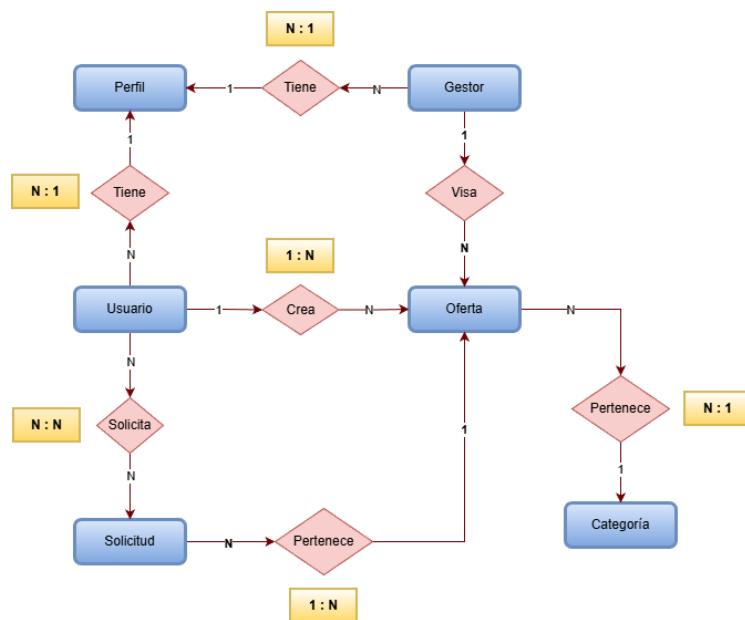
Podrá asignar o modificar los roles de cada usuario.

SA-Admin-03. Gestión de ofertas y solicitudes

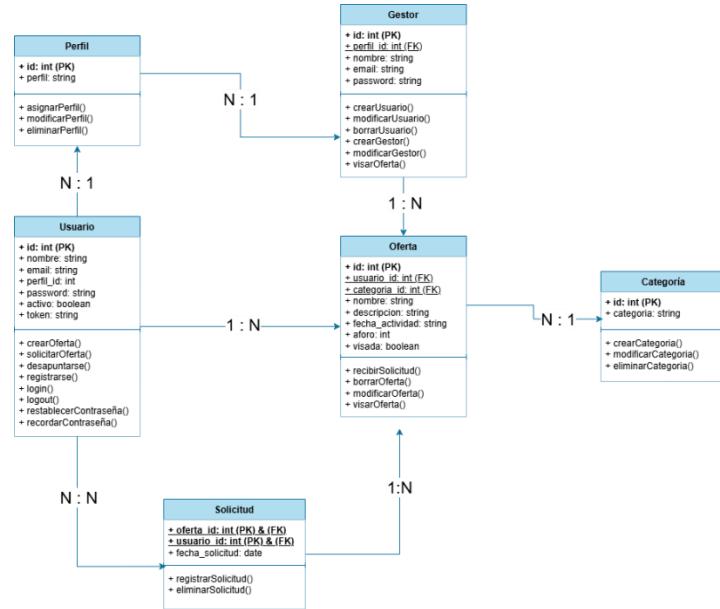
El administrador podrá consultar y modificar cualquier registro de la plataforma.

3.6 Diagramas de Análisis

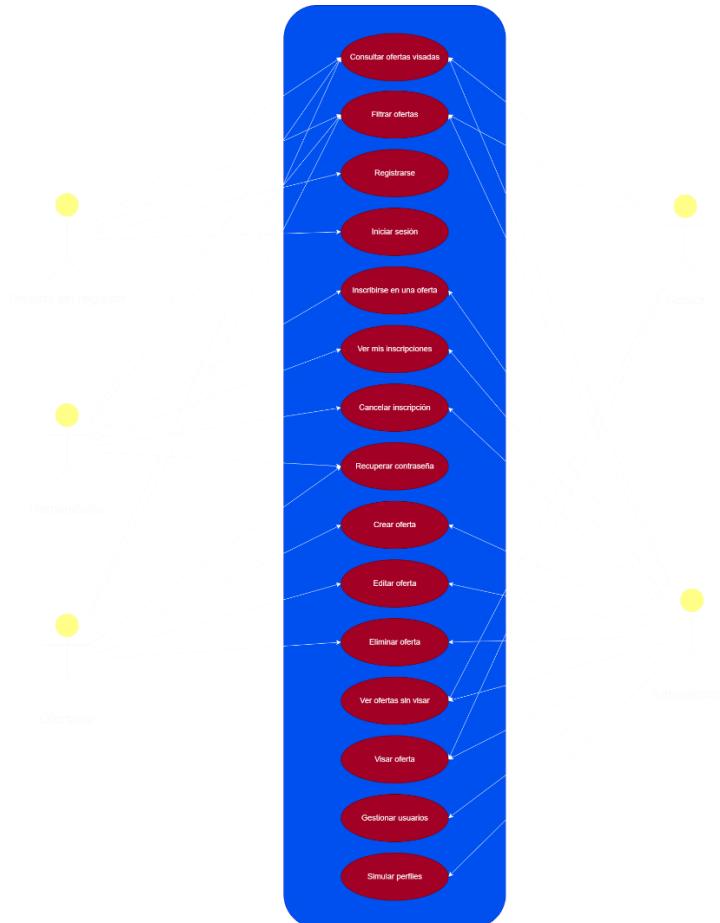
- Modelo Entidad Relación



- Diagrama de clases



- Diagrama de casos de uso



3.7 Definición de Interfaz

Incluye:

- Pantallas de login, registro, listado, paneles según rol, etc.
 - Navegabilidad basada en perfil.
-

4. Construcción del Sistema

Incluye:

- Arquitectura PHP modular
 - Conexión centralizada
 - Sistema de emails simulado en /acceso/emails/
 - Tokens para verificación y recuperación
 - Despliegue por FTP
 - Configuración de hosting y entorno
-

5. Conclusiones

Se ha desarrollado un sistema completo y funcional, desplegado tanto localmente como en Internet, con implementación real de seguridad básica y separación de roles.

6. Glosario

- **Backend:** Parte del sistema que gestiona la lógica del servidor, consultas a la base de datos y control de permisos.
- **Frontend:** Interfaz visible por el usuario; maquetación, estilos y elementos visuales.
- **CRUD:** Conjunto de operaciones básicas sobre datos: Crear, Leer, Actualizar y Borrar.
- **Token:** Código único generado para validar acciones como verificación de cuenta o recuperación de contraseña.
- **Aforo:** Número máximo de plazas disponibles en una actividad.
- **Roles:** Perfiles que definen los permisos: Administrador, Gestor, Ofertante y Demandante.
- **Visado:** Proceso de revisión y validación de ofertas por parte del gestor.
- **Entidad:** Elemento del modelo de datos que representa un objeto real (usuario, oferta, categoría...).
- **Clave Primaria (PK):** Identificador único de cada registro de una tabla.
- **Clave Foránea (FK):** Campo que conecta dos tablas y mantiene la integridad referencial.
- **Hosting:** Servicio donde se aloja la web para que sea accesible por Internet.
- **FTP:** Protocolo utilizado para transferir archivos al hosting.
- **password_hash:** Función de PHP que cifra contraseñas de forma segura.
- **Bootstrap:** Framework CSS utilizado para el diseño de la interfaz.
- **Repositorio (GitHub):** Espacio donde se almacena el código del proyecto y su historial de versiones.
- **phpMyAdmin:** Herramienta gráfica para gestionar bases de datos MySQL.
- **Inscripción:** Acción realizada por un demandante para reservar plaza en una oferta.
- **Visor de ofertas:** Sección donde se muestran las actividades disponibles.

7. Bibliografía

- Documentación PHP
- Documentación MySQL
- Bootstrap
- Hosting InfinityFree
- phpMyAdmin / Adminer