

**Logística de Eventos**

**Manual Técnico**



**Grupo Desarrollador**

Evelin Nicol Amador Gutiérrez

Diego Alejandro Pinzón Álvarez

David Santiago Moreno Campos

Johan Alejandro Benavides Castro

Nury Gabriela Navarrete Vargas

2025

## Contenido

Presentación .....	4
Resumen.....	4
Objetivo.....	5
Finalidad del manual.....	5
DIAGRAMAS DE MODELAMIENTO .....	5
ASPECTO TÉCNICO DEL DESARROLLO DEL SISTEMA .....	5
REQUERIMIENTOS DEL SOFTWARE.....	6
HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO .....	6
1.1.1. JavaScript.....	6
1.1.2. Visual Studio Code.....	7
1.1.3. Git .....	7
1.1.4. GitHub .....	8
1.1.5. React, Bootstrap y CSS .....	8
1.1.6. MySQL .....	9
1.1.7. Express.....	9
1.1.8. Node.js.....	9
2.1. DIAGRAMA DE CLASES .....	10
USUARIO .....	11
CLIENTE.....	11
ADMIN.....	11
EVENTO .....	11
RECURSO.....	11

RECURSOS–EVENTOS .....	12
SOLICITUD .....	12
CITAS.....	12
CONTRATO .....	12
ENCUESTA .....	12
PREGUNTAS.....	12
RESPUESTAS.....	13
GALERÍA.....	13
ARCHIVO MULTIMEDIA .....	13
COMENTARIO.....	13
NOTIFICACIÓN.....	13
GUEST (INVITADO) .....	13
3. ASPECTO TÉCNICO DEL DESARROLLO DEL SISTEMA .....	19
4. REQUERIMIENTOS DEL SOFTWARE .....	20
4.1.2. Node.js .....	20
4.1.2. MySQL/XAMPP .....	20
4.1.3. IDE para JavaScript .....	21
4.1.4. IDE para Bases de Datos .....	22
4.2. REQUISITOS MÍNIMOS .....	23

## Presentación

El presente manual técnico ha sido elaborado con el propósito de describir el funcionamiento general del software Happy Art Events y proporcionar la información necesaria para su correcta instalación, uso y mantenimiento. Su finalidad es servir como una guía clara para cualquier persona que necesite comprender cómo está organizado el sistema y de qué manera se desarrollan sus principales procesos.

En este documento se explica la estructura del software, sus componentes más importantes y el propósito de cada una de sus partes. Este manual busca contribuir al entendimiento técnico del proyecto y garantizar que futuras mejoras o ajustes puedan realizarse de manera ordenada y con pleno conocimiento de su funcionamiento interno.

## Resumen

El propósito de este manual técnico es describir el aplicativo **HAPPY-ART-EVENTS** y presentar la estructura del sistema para el personal que desee gestionarlo, modificarlo o configurarlo.

La información se encuentra organizada según las herramientas empleadas en el desarrollo de la aplicación, acompañada de una breve explicación paso a paso.

La aplicación web cuenta con diversas funciones necesarias para su funcionamiento, y en este manual se detalla cómo se ejecuta cada una de ellas para asegurar el correcto desempeño del sistema.

## **Objetivo**

Proporcionar una guía técnica que permita comprender de manera clara la estructura, los componentes y las funciones principales del sistema, con el fin de asegurar su uso adecuado, su correcta operación y facilitar cualquier proceso de actualización, mantenimiento o mejora futura.

## **Finalidad del manual**

Orientar a cualquier persona que necesite administrar, editar o configurar el software, proporcionando las herramientas y la información necesaria para realizar estos procesos de manera adecuada.

## **ASPECTOS TEÓRICOS**

En esta sección se explican, de manera simple, los elementos y funciones que forman parte del aplicativo. El objetivo es que el lector entienda cómo está organizado el sistema, cómo funciona cada componente y qué herramientas intervienen en su gestión diaria.

## **DIAGRAMAS DE MODELAMIENTO**

Esta sección incluye diagramas y representaciones visuales que ayudan a entender cómo está estructurado el sistema y cómo interactúan sus componentes. Estos modelos permiten visualizar de forma clara el funcionamiento interno del aplicativo y la relación entre sus diferentes elementos.

## **ASPECTO TÉCNICO DEL DESARROLLO DEL SISTEMA**

Esta sección explica al lector los elementos principales que intervienen en el desarrollo del sistema desde una perspectiva técnica. Aquí se detalla cómo se organiza la estructura del aplicativo, la manera en que se manejan y almacenan los datos, y las consideraciones necesarias para su uso

adecuado. El objetivo es ofrecer una visión clara del funcionamiento interno del sistema y brindar recomendaciones que permitan administrarlo, actualizarlo y mantenerlo de forma correcta.

## **REQUERIMIENTOS DEL SOFTWARE**

En esta sección se describen los requerimientos básicos necesarios para que el aplicativo funcione correctamente. Aquí se especifican los recursos, configuraciones y condiciones mínimas que debe cumplir el entorno donde se instale el sistema, garantizando así su ejecución adecuada.

## **ASPECTOS TÉCNICOS**

El aplicativo está diseñado para facilitar la organización y actualización del portafolio de eventos mediante una plataforma digital gestionable. Para mantener su funcionamiento adecuado, se recomienda que este manual sea utilizado únicamente por personal autorizado encargado de administrar, editar o configurar el sistema.

El software integra funciones relacionadas con la gestión de contenido, servicios y comunicación con los clientes, por lo que su manipulación requiere cuidado para mantener la integridad y seguridad de la información registrada.

## **HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO**

En esta sección se describen las herramientas informáticas empleadas para el desarrollo del aplicativo:

### **1.1.1. JavaScript**

JavaScript es un lenguaje de programación interpretado, ampliamente utilizado para el desarrollo de aplicaciones web debido a su flexibilidad y dinamismo. Permite crear funcionalidades interactivas dentro de páginas web y desarrollar tanto el front-end como el back-end mediante entornos como Node.js.

Para el desarrollo de *Happy Art Events*, JavaScript se utilizó como lenguaje principal debido a su eficiencia, facilidad de integración con otras tecnologías y su amplio ecosistema de librerías. Gracias a su sintaxis ligera y su capacidad para manejar eventos de forma asíncrona, facilitó la construcción de funciones dinámicas y procesos internos del aplicativo.

### 1.1.2. Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft, reconocido por su ligereza y su extensa gama de extensiones que facilitan el desarrollo de software. Cuenta con autocompletado inteligente, depuración integrada, soporte para múltiples lenguajes y una interfaz intuitiva.

Para *Happy Art Events*, se empleó Visual Studio Code por su capacidad de integrar herramientas como linters, gestores de paquetes y extensiones necesarias para el desarrollo en JavaScript, Express y MySQL. Su sistema de resaltado de sintaxis y sus notificaciones de errores facilitaron la escritura y organización del código. La versión utilizada fue la gratuita, la cual ofrece todas las funcionalidades necesarias sin limitaciones para el desarrollo del aplicativo.

### 1.1.3. Git

Git es un sistema de control de versiones distribuido que permite gestionar el historial de cambios de un proyecto. Ofrece herramientas para crear ramas, fusionar código y trabajar de manera colaborativa manteniendo la integridad del software.

Durante el desarrollo de *Happy Art Events* se utilizó Git para registrar los diferentes cambios mediante commits, facilitando la trazabilidad del código y permitiendo trabajar de forma ordenada entre los miembros del equipo. Gracias a Git se pudo mantener control de versiones y realizar pruebas sin afectar la rama principal del proyecto.

#### **1.1.4. GitHub**

GitHub es una plataforma de alojamiento de repositorios basada en Git, que facilita la colaboración entre desarrolladores y el control de versiones en proyectos de software. Permite realizar forks, gestionar issues, revisar pull requests y mantener repositorios públicos o privados. Para *Happy Art Events*, se utilizó un repositorio privado en GitHub donde se almacenó el código fuente del proyecto. Desde esta plataforma se realizaron commits, ramas y revisiones de código, garantizando un trabajo colaborativo eficiente. GitHub también facilitó la sincronización del proyecto con servicios externos y la integración continua.

#### **1.1.5. React, Bootstrap y CSS**

React es una biblioteca de JavaScript utilizada para construir interfaces de usuario dinámicas mediante componentes reutilizables. Su manejo eficiente del DOM y su arquitectura basada en estados permite desarrollar aplicaciones rápidas y escalables.

Bootstrap es un framework CSS que proporciona estilos predefinidos, sistemas de rejilla y componentes responsivos, facilitando el diseño de interfaces modernas sin necesidad de codificar todo desde cero.

CSS (Cascading Style Sheets) se empleó para personalizar estilos, colores, animaciones y diseños específicos no cubiertos por Bootstrap.

En *Happy Art Events*, estas herramientas se utilizaron en conjunto para construir la interfaz



visual del aplicativo, logrando un diseño atractivo, ordenado y adaptable a diferentes dispositivos.

#### **1.1.6. MySQL**

MySQL es un sistema de gestión de bases de datos relacional que permite almacenar, consultar y administrar grandes volúmenes de información de manera estructurada. Ofrece rendimiento, seguridad y facilidad de integración con múltiples lenguajes de programación. Para *Happy Art Events*, MySQL se utilizó como base de datos principal, encargada de gestionar la información del sistema, tablas de usuarios, registros y relaciones necesarias para el funcionamiento del aplicativo. Su compatibilidad con Node.js y Express facilitó la conexión entre el servidor y los datos almacenados.

#### **1.1.7. Express**

Express es un framework minimalista para Node.js que permite crear aplicaciones web y API de forma ágil y estructurada. Ofrece herramientas para gestionar rutas, peticiones HTTP, middleware y controladores.

En el desarrollo de *Happy Art Events*, Express se empleó para construir el servidor del aplicativo, definir las rutas del sistema, gestionar solicitudes y brindar una comunicación eficiente entre la base de datos MySQL y el front-end. Su simplicidad y robustez lo convierten en una herramienta ideal para aplicaciones escalables.

#### **1.1.8. Node.js**

Node.js es un entorno de ejecución basado en el motor V8 de Google Chrome que permite ejecutar JavaScript del lado del servidor. Su arquitectura orientada a eventos y su modelo asíncrono lo

convierten en una herramienta eficiente para manejar múltiples solicitudes simultáneas.

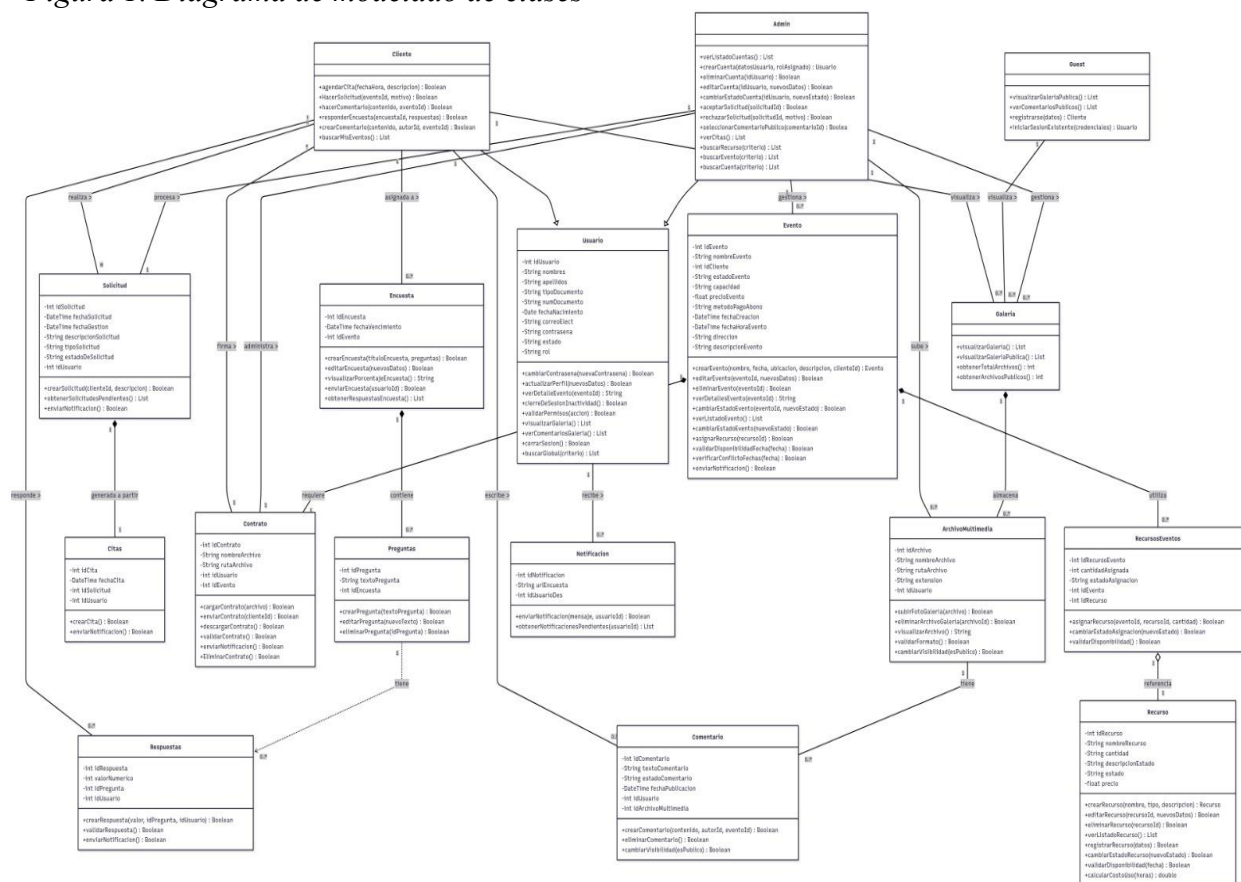
Para *Happy Art Events*, Node.js fue fundamental en la construcción del backend, permitiendo integrar Express, conectarse con MySQL y gestionar la lógica del servidor de manera rápida y escalable.

## DIAGRAMAS DE MODELAMIENTO

### 2.1. DIAGRAMA DE CLASES

El diagrama de clases está formado por las entidades y atributos definidos para el almacenamiento de datos del sistema. Su función es mostrar cómo se estructura la información dentro del aplicativo y la relación que existe entre cada componente, permitiendo entender de manera visual la base del modelo de datos.

Figura 1. Diagrama de modelado de clases



**USUARIO**

Almacena los datos básicos de las personas que acceden al sistema, incluyendo información personal, credenciales y rol asignado. Permite realizar acciones como actualizar perfil, gestionar su sesión, visualizar información y acceder a funciones según sus permisos.

**CLIENTE**

Representa a los usuarios que solicitan o participan en eventos. Puede agendar citas, enviar solicitudes, comentar, responder encuestas y consultar información relacionada con los eventos que tiene asociados.

**ADMIN**

Encargado de gestionar las cuentas del sistema y supervisar los procesos administrativos. Puede crear, editar o eliminar usuarios, gestionar solicitudes, administrar comentarios públicos, revisar eventos, citas y recursos, entre otras funciones de control.

**EVENTO**

Contiene toda la información relacionada con cada evento, como nombre, fecha, ubicación, estado, precio y detalles generales. Permite crear, editar, eliminar y validar la disponibilidad de fechas, además de asignar recursos y enviar notificaciones.

**RECURSO**

Incluye los elementos, materiales o servicios que pueden ser asignados a un evento. Cada recurso cuenta con un estado, disponibilidad y costo. El sistema permite crearlos, modificarlos, calcular su uso y validar su disponibilidad.

**RECURSOS–EVENTOS**

Relaciona los recursos con los eventos, indicando la cantidad asignada y el estado de dicha asignación. Garantiza que los recursos solicitados estén disponibles y gestionados correctamente.

**SOLICITUD**

Almacena las solicitudes creadas por los clientes, registrando su tipo, estado y fechas de gestión. También permite identificar solicitudes pendientes y enviar notificaciones asociadas.

**CITAS**

Registra las citas generadas a partir de una solicitud, incluyendo fecha y los usuarios involucrados. Permite crear y notificar sobre citas programadas.

**CONTRATO**

Gestiona los documentos contractuales relacionados con usuarios y eventos. Permite cargar, enviar, validar, descargar y eliminar contratos dentro del sistema.

**ENCUESTA**

Representa las encuestas asociadas a un evento, incluyendo su fecha de vencimiento y las preguntas que contiene. Permite crear, editar, enviar y visualizar resultados.

**PREGUNTAS**

Incluye el contenido de cada pregunta de una encuesta. Permite crearlas o eliminarlas.

## **RESPUESTAS**

Registra las respuestas enviadas por los usuarios a una encuesta. Cada respuesta incluye el valor, la pregunta a la que pertenece y el usuario que la contestó.

## **GALERÍA**

Permite la visualización de los archivos cargados al sistema. Gestiona la cantidad total de archivos y los que se encuentran visibles al público.

## **ARCHIVO MULTIMEDIA**

Almacena fotografías, imágenes o archivos subidos por usuarios. Permite validar formatos, cambiar visibilidad, visualizar contenido o eliminar archivos.

## **COMENTARIO**

Registra los comentarios realizados sobre archivos o eventos. Permite crearlos, eliminarlos y definir su visibilidad dentro del sistema.

## **NOTIFICACIÓN**

Contiene los datos de las notificaciones enviadas a los usuarios, incluyendo enlaces y destinatarios. Permite consultar notificaciones pendientes.

## **GUEST (INVITADO)**

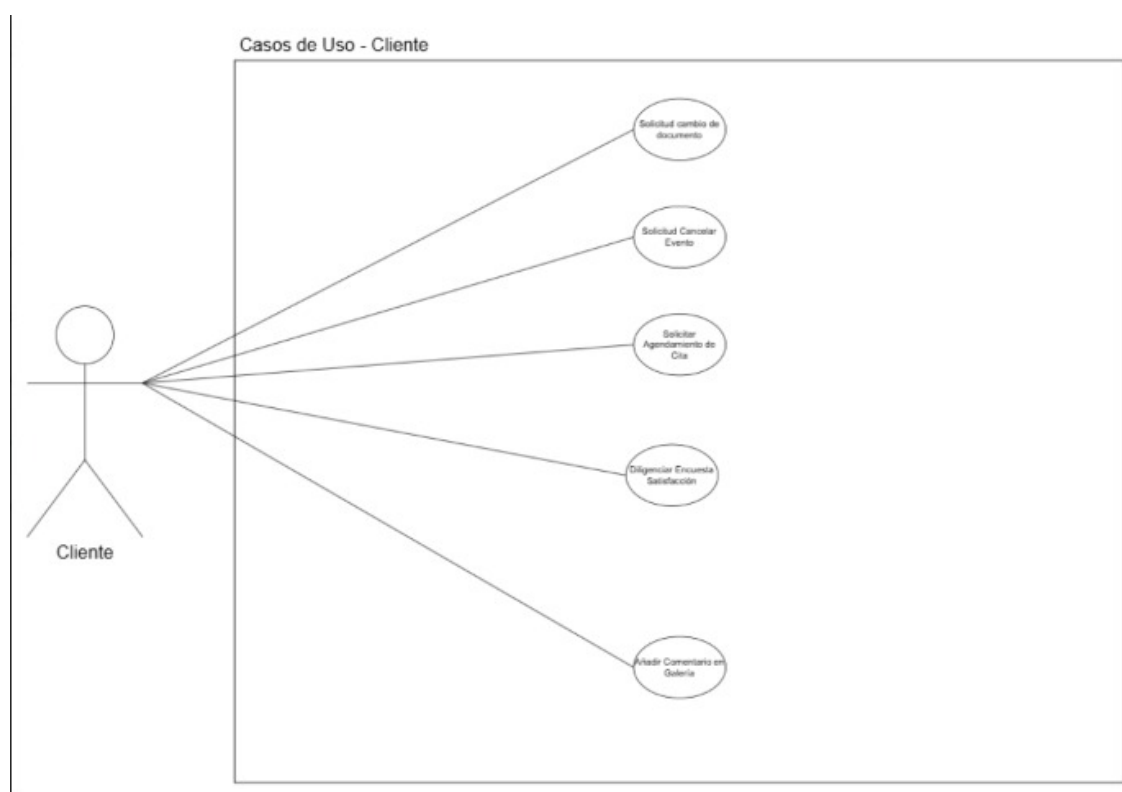
Representa a los visitantes que no están registrados. Puede visualizar contenido público, consultar comentarios y registrarse o iniciar sesión.

## 2.2. DIAGRAMA DE CASOS DE USO

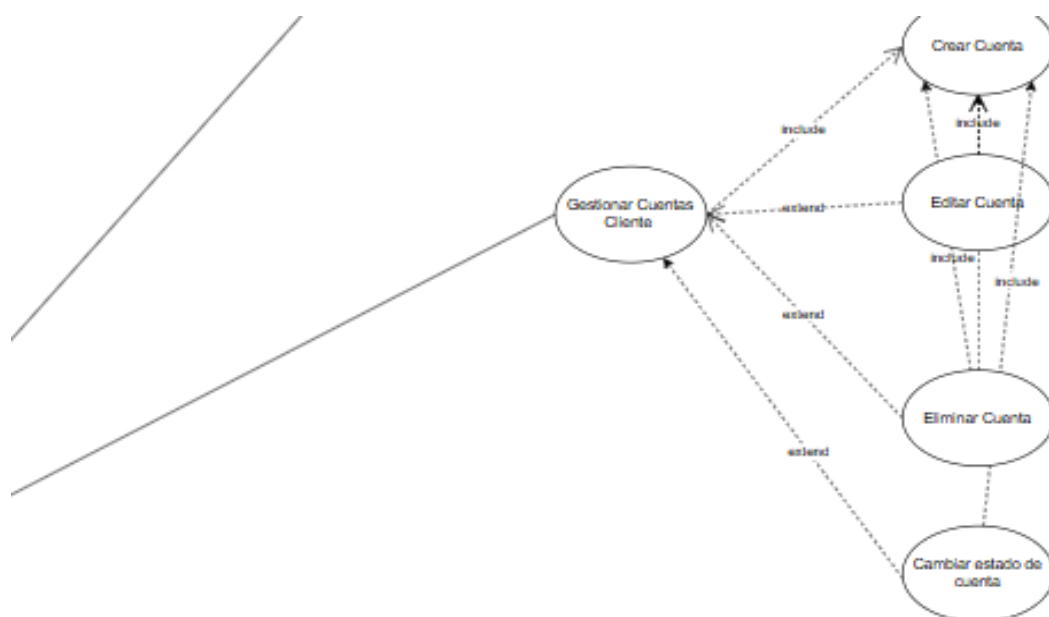
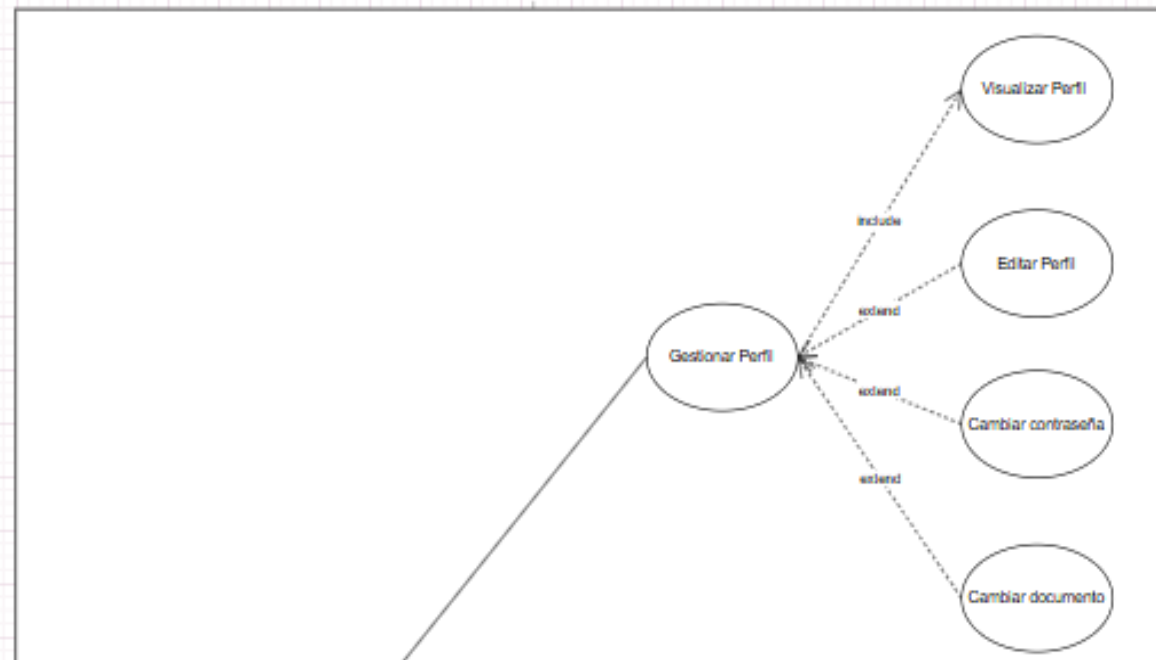
El diagrama de casos de uso muestra las acciones principales que pueden realizar los diferentes actores que interactúan con el sistema. En él se representan las funcionalidades disponibles para cada tipo de usuario y la relación que tienen con el aplicativo.

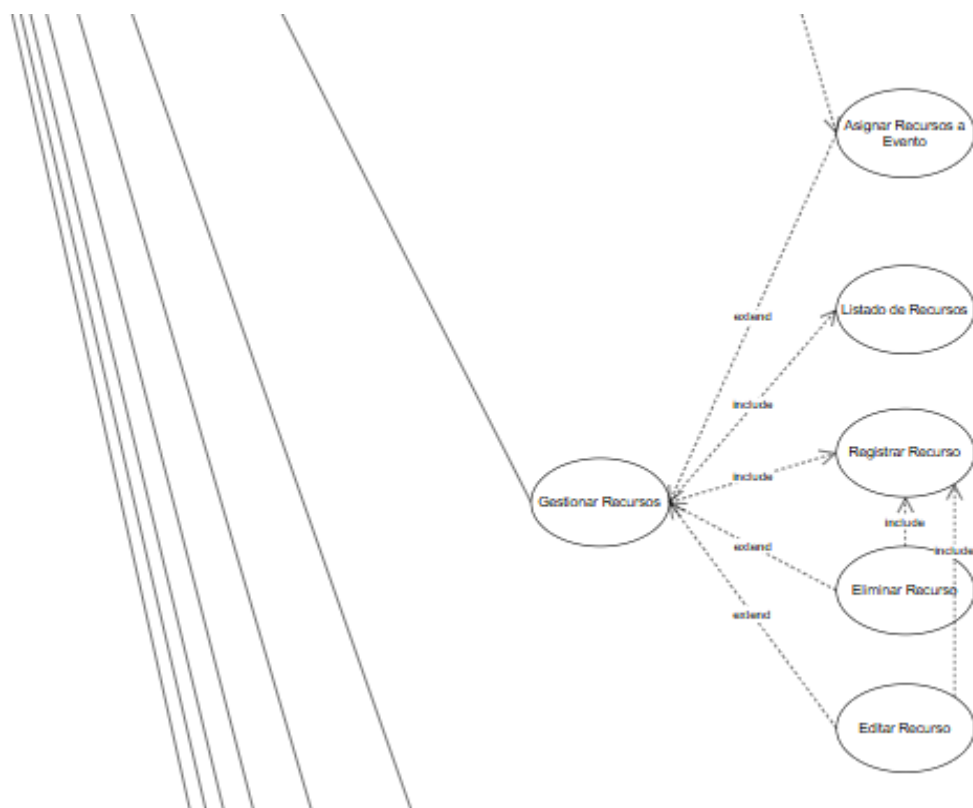
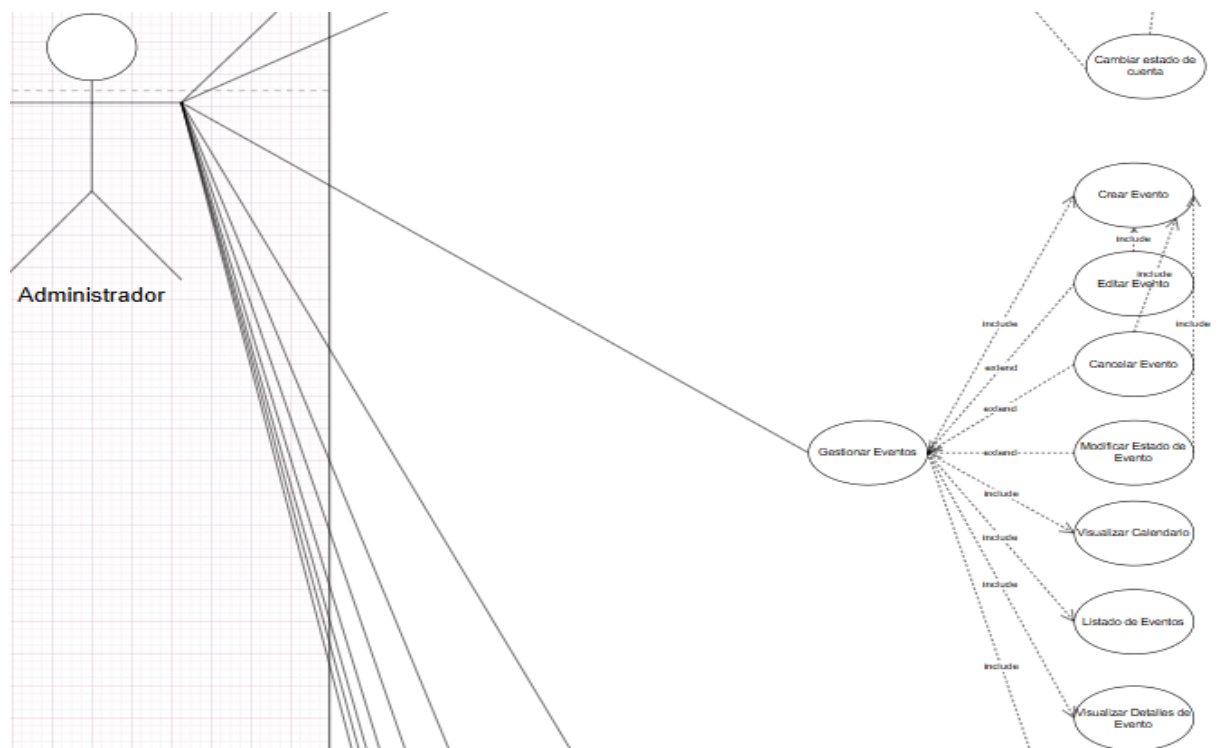
Cada actor posee un conjunto de tareas específicas, como gestionar eventos, administrar recursos, visualizar contenido o realizar solicitudes, lo que permite comprender de manera clara cómo participa cada uno en el funcionamiento general del sistema.

Figura 2. Diagrama de casos de uso

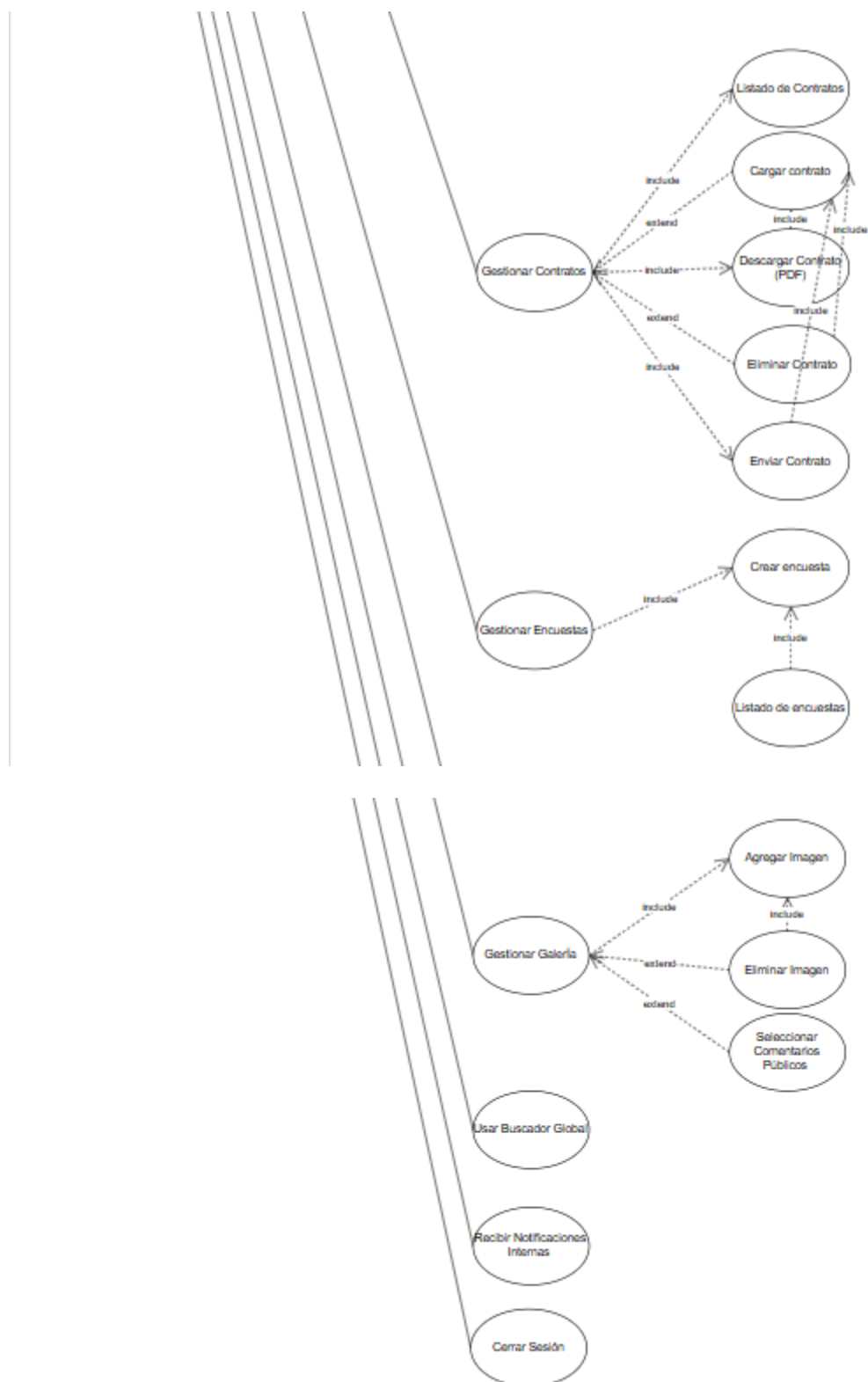


## Casos de Uso - Administrador









## 2.3. DICCIONARIO DE DATOS

User		
Columna	Tipo de Dato	Restricciones
UserId	INT	PK AUTO_INCREMENT
Names	VARCHAR(255)	NOT NULL
DocumentType	ENUM(CC,CE,PP)	NOT NULL
DocumentNumber	VARCHAR(20)	UNIQUE NOT NULL
BirthDate	DATE	NOT NULL
Email	VARCHAR(255)	UNIQUE NOT NULL
Password	VARCHAR(255)	NOT NULL
Status	ENUM(active,inactive)	DEFAULT active
Role	ENUM(user,admin)	DEFAULT user
Photo	VARCHAR(255)	Ninguna
EventResources		
Columna	Tipo de Dato	Restricciones
EventResourceId	INT	PK AUTO_INCREMENT
AssignedQuantity	INT	NOT NULL
AssignmentStatus	ENUM(reserved,assigned,returned)	DEFAULT reserved
EventId	INT	FK Events.EventId
ResourceId	INT	FK Resources.ResourceId
Prices	FLOAT	NOT NULL
Questions		
Columna	Tipo de Dato	Restricciones
QuestionId	INT	PK AUTO_INCREMENT
QuestionText	TEXT	NOT NULL
Comments		
Columna	Tipo de Dato	Restricciones
CommentId	INT	PK AUTO_INCREMENT
CommentText	TEXT	NOT NULL
CommentStatus	ENUM(pending,selected,rejected)	DEFAULT pending
PublicationDate	DATETIME	DEFAULT CURRENT_TIMESTAMP
UserId	INT	FK User.UserId
MultimediaFileId	INT	FK MultimediaFile.FileId SET NULL
Promotions		
Columna	Tipo de Dato	Restricciones
PromotionId	INT	PK AUTO_INCREMENT
TitleProm	VARCHAR(255)	Ninguna
DescriptionProm	VARCHAR(255)	Ninguna
Price	DECIMAL(10,2)	Ninguna
PasswordReset		
Columna	Tipo de Dato	Restricciones
Email	VARCHAR(255)	PK
Code	VARCHAR(10)	NOT NULL
CreatedAt	DATETIME	DEFAULT CURRENT_TIMESTAMP
SendAttempts	TINYINT	DEFAULT 1
LastSendAttempt	DATETIME	DEFAULT CURRENT_TIMESTAMP

Events		
Columna	Tipo de Dato	Restricciones
EventId	INT	PK AUTO_INCREMENT
EventName	VARCHAR(255)	NOT NULL
ClientId	INT	FK User.UserId NOT NULL
EventStatus	ENUM(In_planning,In_execution,Co	DEFAULT In_planning
Capacity	INT	NOT NULL
EventPrice	FLOAT	NOT NULL
AdvancePaymentM	ENUM(Cash,Transfer,Card)	
CreationDate	DATETIME	DEFAULT CURRENT_TIMESTAMP
EventDateTime	DATETIME	NOT NULL
Address	VARCHAR(255)	NOT NULL
EventDescription	TEXT	
ContractRoute	VARCHAR(100)	
ContractNumber	INT	

Requests		
Columna	Tipo de Dato	Restricciones
RequestId	INT	PK AUTO_INCREMENT
RequestDate	DATETIME	DEFAULT CURRENT_TIMESTAMP
ManagementDate	DATETIME	
RequestDescription	TEXT	NOT NULL
RequestType	ENUM(schedule_appointment,cance	NOT NULL
RequestStatus	ENUM(pending,approved,rejected)	DEFAULT pending
UserId	INT	FK User.UserId
EventId	INT	FK Events.EventId SET NULL

Answers		
Columna	Tipo de Dato	Restricciones
AnswerId	INT	PK AUTO_INCREMENT
NumericValue	INT	NOT NULL
EventId	INT	FK Events.EventId
UserId	INT	FK User.UserId
QuestionId	INT	FK Questions.QuestionId
UNIQUE	(EventId,UserId,QuestionId)	UNIQUE constraint

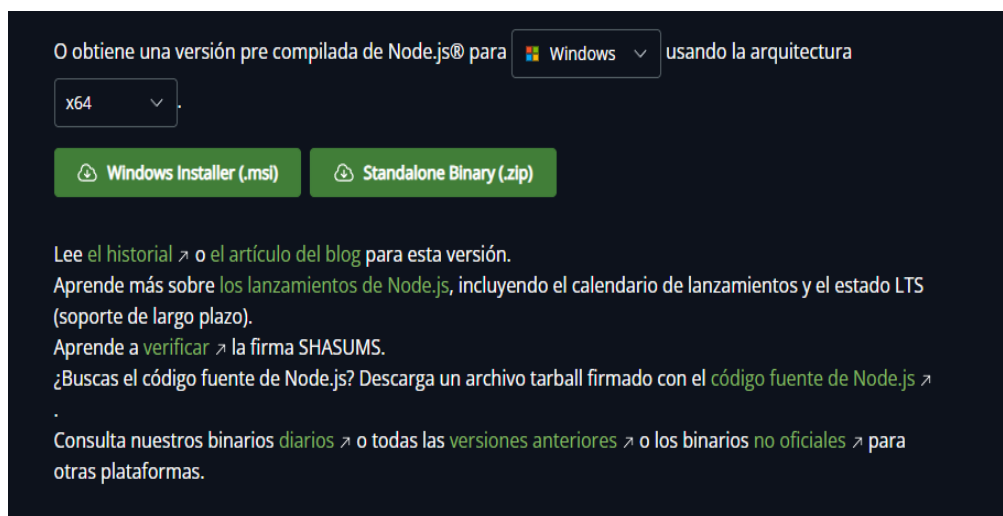
### 3. ASPECTO TÉCNICO DEL DESARROLLO DEL SISTEMA

En esta sección se describe de forma general los aspectos técnicos que intervienen en el desarrollo del sistema, incluyendo las herramientas necesarias y el entorno básico para realizar ajustes o modificaciones. Su propósito es orientar al lector sobre los elementos fundamentales que debe conocer para gestionar o actualizar el aplicativo de manera adecuada y organizada.

## 4. REQUERIMIENTOS DEL SOFTWARE

### 4.1.2. Node.js

Para instalar Node.js, primero entra a su página oficial y descarga la versión estable recomendada para tu sistema operativo (Windows, macOS o Linux). Una vez descargado el instalador, ejecútalo y sigue los pasos que aparecen en pantalla. Al finalizar, Node.js quedará instalado junto con npm (el gestor de paquetes). Puedes comprobarlo abriendo la terminal y escribiendo `node -v` para ver la versión instalada.




### 4.1.2. MySQL/XAMPP

La forma más sencilla de instalar MySQL es usando XAMPP, un paquete que incluye Apache, PHP y MySQL. Descarga XAMPP desde su página oficial, ejecuta el instalador y selecciona los componentes que quieras (asegúrate de marcar MySQL). Una vez instalado, abre el panel de control de XAMPP y activa el servicio de MySQL. Desde ahí podrás gestionar tu base de datos y acceder a phpMyAdmin para trabajar con ella de manera visual.

## Download

XAMPP is an easy to install Apache distribution containing MariaDB, PHP, and Perl. Just download and start the installer. It's that easy. Installers created using InstallBuilder.


**XAMPP for Windows 8.0.30, 8.1.25 & 8.2.12**

Version		Checksum	Size
8.0.30 / PHP 8.0.30	<a href="#">What's Included?</a>	<a href="#">md5</a> <a href="#">sha1</a>	<a href="#">Download (64 bit)</a> 144 Mb
8.1.25 / PHP 8.1.25	<a href="#">What's Included?</a>	<a href="#">md5</a> <a href="#">sha1</a>	<a href="#">Download (64 bit)</a> 148 Mb
8.2.12 / PHP 8.2.12	<a href="#">What's Included?</a>	<a href="#">md5</a> <a href="#">sha1</a>	<a href="#">Download (64 bit)</a> 149 Mb

[Requirements](#)
[More Downloads »](#)

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms [here](#).

### Documentation/FAQs

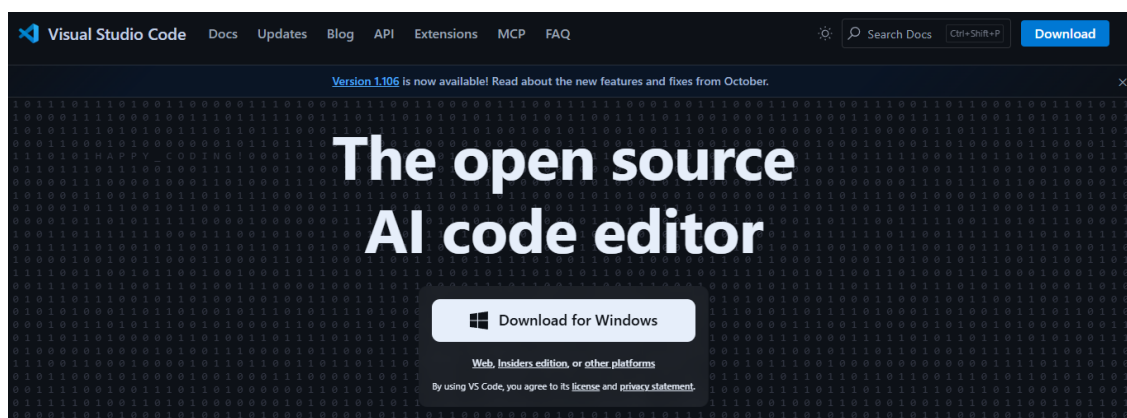
There is no real manual or handbook for XAMPP. We wrote the documentation in the form of FAQs. Have a burning question that's not answered here? Try the [Forums](#) or [Stack Overflow](#).

- [Linux FAQs](#)
- [Windows FAQs](#)
- [OS X FAQs](#)


**XAMPP for Browsers & 8.3.12**


### 4.1.3. IDE para JavaScript

Un IDE enfocado en JavaScript como Visual Studio Code es muy fácil de instalar. Ve a la página oficial de VS Code, descarga el instalador según tu sistema operativo y ejecútalo. Una vez abierto, puedes personalizarlo instalando extensiones como *ESLint* o *Prettier* para mejorar tu flujo de trabajo con JavaScript. Es ligero, rápido y pensado para programar con comodidad.



#### 4.1.4. IDE para Bases de Datos

Para trabajar con bases de datos, existen IDEs especializados como MySQL Workbench. Descárgalo desde la página oficial de MySQL, instala el programa y ábrelo. Desde ahí podrás conectarte a tu servidor MySQL (local o remoto) ingresando usuario, contraseña y puerto. Este entorno te permite crear, modificar y consultar bases de datos de manera gráfica, sin necesidad de escribir todos los comandos manualmente.

[General Availability \(GA\) Releases](#) [Archives](#) 

## MySQL Workbench 8.0.45

Select Operating System:

Microsoft Windows ▼


**Recommended Download:**

### MySQL Installer

for Windows

**All MySQL Products. For All Windows Platforms.  
In One Package.**


Starting with MySQL 5.6 the MySQL Installer package replaces the standalone MSI packages.



**Windows (x86, 32 & 64-bit), MySQL Installer MSI** [Go to Download Page >](#)

**Other Downloads:**

<b>Windows (x86, 64-bit), MSI Installer</b>	8.0.45	252.3M	<a href="#">Download</a>
(mysql-workbench-community-8.0.45-winx64.msi)		MD5: 7374733ebabd22f124bbc10d936b4eb8   <a href="#">Signature</a>	

 We suggest that you use the [MD5 checksums](#) and [GnuPG signatures](#) to verify the integrity of the packages you download.

## 4.2. REQUISITOS MÍNIMOS

- **Sistema Operativo:** Windows 10
- **Procesador:** Intel Celeron / AMD equivalente
- **Memoria RAM:** 2 GB
- **Almacenamiento:** 1 GB de espacio disponible
- **Resolución de pantalla:** 1280 × 720 píxeles
- **Periféricos:** Teclado y ratón (obligatorios)