

# 1. Concepto general del sistema

El proyecto es una **plataforma de conversaciones anónimas** entre:

- **Clientes** (usuarios anónimos)
- **Admin** (una única cuenta o varias cuentas autorizadas)

La comunicación sigue el patrón de un **sistema de tickets**, donde cada conversación:

- Tiene un **código único**
- Tiene **un estado**
- Tiene **mensajes**
- Puede ser reabierta, pausada o cerrada
- No requiere crear un usuario global, solo un **nickname y contraseña para la conversación**

El usuario no se registra:

👉 *solo entra a una conversación concreta mediante su código y la contraseña asociada.*

---

## 2. Flujo completo del usuario

### Pantalla de bienvenida

Botones:

- Continuar conversación
  - Crear conversación
- 

### Crear conversación

1. El usuario rellena:
    - *Motivo* (breve)
    - *Descripción detallada*
  2. Luego ingresa:
    - *Nickname* (opcional, se genera uno si se deja vacío)
    - *Contraseña*
  3. Se crea un ticket:
    - Estado inicial: "**espera**"
    - Se genera un **ID único**
  4. El ID se muestra al usuario, con opción de:
    - Descargar un PDF/archivo con el código
    - Copiarlo al portapapeles
- 

## Continuar conversación

Formulario:

- **Código del ticket**
- **Contraseña**

Si coinciden → acceso a la conversación.

---

## Dentro de la conversación

Elementos:

- Estado visible (espera / activo / pausado / cerrado)

- Mensajes cronológicos
- Adjuntar archivos
- Botón “Salir”

El admin:

- Ve todos los tickets
  - Puede cambiar estados
  - Puede responder
- 

## 3. Arquitectura técnica

### 3.1 Frontend (cliente web)

- **HTML5**
- **CSS3**
- **JavaScript puro** (por ahora)
- Eventualmente podrías ampliarlo a:
  - WebSockets (para mensajes en tiempo real)
  - IndexedDB o LocalStorage para guardar el código de la conversación

### 3.2 Backend

- **PHP** (REST API con controladores organizados)
- Puede aprovechar librerías como:
  - JWT (si quieras sesión del admin)
  - Slim / Lumen si quieras microframework (opcional)

### 3.3 Base de datos

- MySQL

### 3.4 Comunicación

- Peticiones REST
- Enviar/recibir mensajes JSON
- Para mensajes en tiempo real:
  - Polling simple (cada X segundos)
  - O WebSockets más adelante

### 3.5 Almacenamiento de archivos

- Ficheros subidos:
    - Se guardan en `/uploads/`
    - Se relacionan con mensajes por ID
- 

## 4. Estructura de la base de datos

Aquí tienes un diseño que funciona y es fácil de implementar:

---

### TABLA: conversations

Campo	Tipo	Descripción
id	VARCHAR(12) PK	Identificador único público (ej: "A3F9D12EBC4")
nickname	VARCHAR(50)	Nombre del cliente

password_has	VARCHAR(255) h	Hash de la contraseña
title	VARCHAR(100)	Motivo
description	TEXT	Descripción del usuario
status	ENUM('espera', 'activo', 'pausado', 'cerrado')	Estado
created_at	DATETIME	Fecha creación
updated_at	DATETIME	Última actualización

---

## TABLA: messages

Campo	Tipo	Descripción
id	INT PK AI	ID del mensaje
conversation_id	VARCHAR(12) FK	Ticket al que pertenece
sender	ENUM('cliente', 'admin')	Quién envió el msg
content	TEXT	Texto del mensaje
created_at	DATETIME	Fecha/hora
file_path	VARCHAR(255)	Ruta del archivo adjunto (nullable)

---

## TABLA: admins

(para controlar login del admin)

Campo	Tipo
id	INT PK AI
username	VARCHAR(50)
password_has	VARCHAR(255) h

---

## 5. Seguridad

- Contraseñas **siempre hashed** con `password_hash()`
  - Validar acceso a cada ticket por:
    - ID + contraseña
  - Limitar tamaño de archivos
  - Filtrar HTML para evitar XSS
  - API separada por roles:
    - Endpoints cliente
    - Endpoints admin
- 

## 6. Páginas necesarias

### Cliente

- `index.html` — Bienvenida
- `create.html` — Crear conversación
- `access.html` — Continuar conversación
- `conversation.html` — Chat + archivos

### Admin

- `admin_login.html`
- `admin_dashboard.html`
- `admin_conversation.html`

### API PHP

- `/api/createConversation.php`
  - `/api/accessConversation.php`
  - `/api/getMessages.php`
  - `/api/sendMessage.php`
  - `/api/changeStatus.php`
  - `/api/uploadFile.php`
  - `/api/adminLogin.php`
- 

## 7. Funcionalidades extra para más adelante

- Notificaciones con WebSockets
  - Cifrar archivos en el servidor
  - Autogenerar nickname tipo:  
`user-37492`
  - Descarga del PDF con:
    - Código del ticket
    - Contraseña
    - Instrucciones de acceso
  - Vista previa de archivos en el chat
- 

## 8. Resumen de la idea refinada

Estás creando un sistema que:

- Permite **crear tickets anónimos** con un nickname y contraseña
- Genera un **ID único descargable**
- Permite a un administrador manejar **todas las conversaciones**
- Permite subir archivos
- Funciona con **PHP + REST + MySQL + JS**

Es un **chat asíncrono**, no un chat en tiempo real (al menos por ahora).



## 9. Panel de Administración (lado del admin)

Los **administradores** son usuarios reales del sistema (a diferencia de los clientes, que son anónimos). Son los encargados de gestionar, responder y supervisar todas las conversaciones abiertas por los usuarios.

Los admins son **todo-poderosos** dentro del sistema:

- Pueden ver **todas las conversaciones**
  - Pueden **entrar** en cualquier conversación
  - Pueden **cambiar el estado** de cualquier ticket
  - Pueden **responder mensajes**
  - Pueden **subir archivos**
  - Pueden **cerrar, pausar, reabrir** y controlar el flujo de las conversaciones
  - Pueden **eliminar mensajes** o información sensible si se requiere
- 

### 9.1. Autenticación del administrador

Los administradores sí requieren un proceso de login.

La autenticación se compone de:

- **Nombre de usuario**
- **Contraseña**
- *Opcional:* Token JWT para persistir la sesión (si el backend lo usa)

La tabla `admins` almacena sus credenciales.

#### Flujo de inicio de sesión

1. El admin accede a `admin_login.html`.

2. Introduce credenciales.
  3. La API valida:
    - Usuario existe
    - Contraseña coincide (password\_verify)
  4. Si es correcto:
    - Se genera sesión o token
    - Se redirige al *Panel de Control*
- 

## 9.2. Vista general del panel del admin

El panel principal funciona como un sistema de **gestión de tickets**.

 **Página: admin\_dashboard.html**

**Elementos principales:**

 **Listado de conversaciones**

Una tabla o lista con todos los tickets:

ID	Nickname	Estado	Última actividad	Acciones
A8D9F2...	user-3023	espera	hace 5 min	Ver
F92CDE...	anónimo	activo	hace 10 seg	Ver
X192LQ...	terminado	hace 2 días	Ver	

Características:

- Ordenar por estado, fecha o ID
- Filtros (solo activos, solo en espera, cerrados...)
- Indicadores visuales:
  -  espera
  -  activo
  -  cerrado
  -  pausado



### Panel lateral (opcional)

- Número de conversaciones en cada estado
  - Acceso al perfil del admin
  - Botón de cerrar sesión
- 

## 9.3. Página de gestión de una conversación

`admin_conversation.html?id=[ID]`

Al entrar en un ticket, el admin ve:

---



### 1. Información del ticket

- ID del ticket
- Nickname del cliente
- Estado actual

- Fecha de creación
- Última actualización
- Descripción original del usuario

Ejemplo de cabecera:

Ticket: A8D9F2C1B92

Usuario: shadow-2381

Estado: Activo

Motivo: Problemas personales

Creado: 2025-11-20 19:24

---

## ★ 2. Control de estado

El admin puede cambiar el estado mediante un menú:

- Espera → Activo
- Activo → Pausado
- Pausado → Activo
- Activo → Cerrado

Botones presentados claramente:

- **Activar**
  - **Pausar**
  - **Cerrar ticket**
  - **Reabrir**
-

## ★ 3. Zona de mensajes (chat)

La parte principal es un chat con formato tipo WhatsApp/Discord:

### Mensajes del usuario (cliente)

A la izquierda, color suave.

### Mensajes del admin

A la derecha, color distinto.

### Funciones:

- Autorefresh o WebSockets para nuevos mensajes
  - Adjuntar archivos:
    - Imágenes
    - PDF
    - TXT
  - Ver archivos enviados por el cliente
  - Eliminar o editar mensajes del admin (si quieres añadir esa opción)
- 

## ★ 4. Escribir un mensaje

Un área de texto + botón:

-  “Enviar mensaje”
  -  “Adjuntar archivo”
- 

## ★ 5. Historial del ticket

Opcional, pero útil.

- Cambios de estado
- Conexiones del usuario
- Envío de archivos

Ejemplo:

```
[19:22] El usuario creó el ticket.  
[19:25] El admin cambió el estado a ACTIVO.  
[19:26] El usuario envió un archivo: foto.png
```

---

## 9.4. Funcionalidades clave del admin

### ✓ Ver todas las conversaciones

Incluyendo las cerradas, archivadas o pausadas.

### ✓ Tomar control de cualquier conversación

El admin siempre tiene acceso total.

### ✓ Cambiar cualquier estado

Este control es exclusivo del administrador.

### ✓ Enviar mensajes y archivos

Pueden subir documentos o capturas.

### ✓ Filtrar, ordenar y buscar tickets

Herramientas para manejar decenas o cientos de conversaciones.

### ✓ Cerrar una conversación

Cuando ya no requiere más intervención.

---

## 9.5. Seguridad en la sección de admin

- Acceso restringido mediante login
  - Todas las páginas admin requieren verificar token/sesión
  - Los endpoints se protegen con roles:
    - Cliente solo accede a su conversación
    - Admin accede a todas
  - Validación estricta de archivos
  - Registro de logs básicos (opcional):
    - Cambios de estado
    - Accesos
    - Respuestas del admin
- 

## 9.6. Cómo se ven visualmente las páginas del admin

### Login (admin\_login.html)

- Minimalista
  - Fondo gris suave
  - Card centrada con:
    - Usuario
    - Contraseña
    - Botón “Entrar”
-

## **Panel (admin\_dashboard.html)**

Estilo tipo gestor:

- Barra lateral oscura
    - Dashboard
    - Tickets en espera
    - Tickets activos
    - Tickets cerrados
    - Configuración
    - Cerrar sesión
  - Parte central clara con tabla de tickets
  - Búsqueda arriba
  - Filtros por estado
- 

## **Vista de conversación (admin\_conversation.html)**

Diseño muy intuitivo:

- Columna izquierda:
  - Info del ticket
  - Estados
- Columna derecha:
  - Chat completo
  - Caja de mensajes
  - Adjuntar archivo

Estética tipo:

- Slack / Discord / WhatsApp Web

Muy usable para escribir rápido y gestionar mucho contenido.