

# **El Fogón**

Sistema distribuido para músicos con estado compartido del compás, la  
canción y el repertorio

Segundo Cuatrimestre de 2025

Tutor: Diego Montaldo

Luis Waldman

Padrón: 79279

# Índice

<b>Palabras Clave</b>	<b>4</b>
<b>Abstracto</b>	<b>5</b>
Español . . . . .	5
English . . . . .	5
<b>Agradecimientos</b>	<b>7</b>
<b>1. Introducción</b>	<b>8</b>
1.1. Motivación . . . . .	8
1.2. Objetivos . . . . .	8
1.2.1. Objetivo General . . . . .	8
1.2.2. Objetivos Técnicos . . . . .	9
<b>2. Estado del Arte</b>	<b>10</b>
2.1. Sobre el sonido y la música . . . . .	10
2.2. Herramientas . . . . .	10
2.2.1. Afinadores . . . . .	10
2.2.2. Metrónomos . . . . .	11
2.2.3. Pentagramas y cancioneros . . . . .	11
2.3. Aplicaciones Web Musicales . . . . .	11
2.3.1. Cancioneros . . . . .	11
2.3.2. Reproductores online de video y acordes . . . . .	12
2.3.3. Editores de partituras online . . . . .	12
2.3.4. Reproductores online de audio y video . . . . .	12
2.4. El futuro llega, hace rato... . . . .	12
2.5. Conclusión . . . . .	12
<b>3. Solución Implementada</b>	<b>14</b>
3.1. WWW.FOGON.AR . . . . .	14
3.1.1. tocar . . . . .	15
3.1.2. tocar . . . . .	15
3.2. Arquitectura del Sistema . . . . .	15
3.2.1. Vista Lógica . . . . .	15
3.2.2. Vista de procesos . . . . .	15

3.2.3. Vista Física . . . . .	15
3.2.4. Vista desarrollo . . . . .	16
3.3. Tecnologías Utilizadas . . . . .	16
3.4. Implementación . . . . .	16
<b>4. Metodología Aplicada</b>	<b>17</b>
4.1. Marco Metodológico . . . . .	17
4.2. Proceso de Desarrollo . . . . .	17
4.3. Herramientas de Desarrollo . . . . .	17
4.4. Gestión del Proyecto . . . . .	17
<b>5. Experimentación y Validación</b>	<b>18</b>
5.1. Diseño de Experimentos . . . . .	18
5.2. Casos de Prueba . . . . .	18
5.3. Resultados Obtenidos . . . . .	18
5.4. Análisis de Resultados . . . . .	18
5.5. Validación con Usuarios . . . . .	18
<b>6. Cronogramas</b>	<b>19</b>
6.1. Cronograma Planificado . . . . .	19
6.2. Cronograma Real . . . . .	19
6.3. Desviaciones y Ajustes . . . . .	19
<b>7. Riesgos y Lecciones Aprendidas</b>	<b>20</b>
7.1. Identificación de Riesgos . . . . .	20
7.2. Gestión de Riesgos . . . . .	20
7.3. Problemas Encontrados . . . . .	20
7.4. Lecciones Aprendidas . . . . .	20
<b>8. Impactos Sociales y Ambientales del Proyecto</b>	<b>21</b>
8.1. Impacto Social . . . . .	21
8.2. Impacto Ambiental . . . . .	21
8.3. Responsabilidad Social y Ética . . . . .	21
8.4. Medidas de Mitigación . . . . .	21
<b>9. Desarrollos Futuros</b>	<b>22</b>
9.1. Mejoras Propuestas . . . . .	22

9.2. Funcionalidades Adicionales . . . . .	22
9.3. Escalabilidad . . . . .	22
9.4. Investigación Futura . . . . .	22
9.5. Conclusiones Finales . . . . .	22

## Palabras Clave

- Web
- Vue.js
- WebSocket
- WebRTC
- Diseño responsivo
- Golang
- NUnit
- Playwright
- Sincronización
- Compensación de *jitter*
- Música
- Letras
- Acordes
- Partituras

## Abstracto

### Español

Fogón es un sistema distribuido orientado a facilitar la práctica musical y la sincronización entre músicos. A cada uno le ofrece, desde una aplicación web progresiva, vistas para su instrumento: letras, acordes o partituras. Permite crear listas y editar canciones. También, crear sesiones en la que los participantes comparten el estado de la canción, el repertorio y hasta el compás exacto que están tocando.

El sistema ayuda al aprendizaje y la enseñanza musical al mostrar cómo realizar los acordes en cada instrumento y al permitir afinarlos.

La arquitectura soporta numerosas vistas complejas y responsivas para distintos instrumentos, y permite agregar nuevas vistas para otros instrumentos. Para las partituras emplea la librería VexFlow, mientras que para las vistas de letra y acordes se usó un desarrollo propio.

El principal desafío técnico fue el de la reproducción en dispositivos distribuidos: un "delay" de 20 ms empieza a ser perceptible por el oído humano y la latencia en internet es mayor. Se implementó un protocolo que sincroniza los dispositivos usando WebSocket y WebRTC (con compensación de jitter) a través de un servidor Golang.

Para probar y "debuggear" el sistema de sincronización hubo que desarrollar algunas vistas y controles.

Las pruebas de aceptación se desarrollaron de punta a punta con NUnit y Playwright.

**Palabras Clave:** Web, Vue.js, WebSocket, WebRTC, Diseño responsivo, Golang, NUnit, Playwright, Sincronismo, Compensación de \*jitter\*, Música, Letras, Acordes, Partituras.

### English

Fogón is a distributed system designed to facilitate musical practice and synchronization among musicians. It provides each musician with a progressive web application featuring instrument-specific views: lyrics, chords, or sheet music. It allows users to create playlists and edit songs. Additionally, it enables the creation of sessions where participants share the song state, repertoire, and even the exact bar they are playing.

The system aids in musical learning and teaching by showing how to perform chords on each instrument and allowing for tuning.

The architecture supports numerous complex and responsive views for different instruments, and allows for the addition of new views for other instruments. For sheet music, it employs the VexFlow library, while for lyrics and chord views, a custom development was used.

The main technical challenge was distributed device playback: a delay of 20 ms starts to become perceptible to the human ear, and internet latency is typically higher. A protocol was implemented

that synchronizes devices using WebSocket and WebRTC (with jitter compensation) through a Golang server.

To test and debug the synchronization system, several views and controls had to be developed. End-to-end acceptance tests were developed with NUnit and Playwright.

**Keywords:** Web, Vue.js, WebSocket, WebRTC, Responsive design, Golang, NUnit, Playwright, Synchronization, Jitter compensation, Music, Lyrics, Chords, Sheet Music.

## Agradecimientos

A mi madre y a mi padre,

A la Educación Pública y en particular a las cátedras de Arquitectura de Software, Base de Datos, Introducción a Sistemas Distribuidos y Sistemas Distribuidos de la Universidad de Buenos Aires.

A Pitágoras, a Newton, a Turing y a todos los que asumen la heroica tarea de descubrir y transmitir ciencia.

A los artistas que prefieren estructuras rigurosas.

A las Cadenas de Márkov y la proliferación de IAs.

A vos que estás leyendo esto.



# 1 Introducción

Tanenbaum y Van Steen definen: “Un sistema distribuido es una colección de computadoras independientes que aparece ante sus usuarios como un sistema único y coherente” [?]

La definición coincide con la de un grupo musical que combina armonías, melodías y ritmos de modo que se escuche como un tema único y coherente.

Para hacer esto los músicos se nutren de protocolos y mecanismos de coordinación que les permiten sincronizarse y compartir información.

Los avances en la ciencia y la ingeniería fueron incorporados por los músicos: Pitágoras sintetizó la matemática y la armonía; la imprenta permitió la publicación de partituras; la revolución industrial introdujo el metrónomo de Maelzel.

Desde que existe Internet, circulan archivos con páginas de acordes que evolucionaron a páginas multimedia, archivos MIDI, aplicaciones de edición de partituras, etc.

La aplicación .<sup>El</sup> Fogón”pone a disposición de los músicos estos avances y propone un nuevo enfoque: cada músico puede acceder con su dispositivo a un Estado compartido de compás, canción, acordes, partituras, repertorio, etc.

Esto permite que varios músicos toquen juntos y en sincronía, compartiendo y actualizando información en tiempo real, pero cada uno ve la información de su instrumento.

## 1.1 Motivación

Busca hacer un aporte novedoso a la música desde la informática, incorporando soluciones anteriores y agregando un enfoque novedoso: el estado compartido entre músicos.

Además, busca la alta disponibilidad: los músicos pueden acceder a la aplicación en su dispositivo en cualquier momento, sin necesidad de conexión a la red.

## 1.2 Objetivos

Fogón es una solución dirigida tanto a cantantes y guitarristas aficionados como a músicos de orquestas profesionales: una aplicación en donde puedan buscar letras de acordes y canciones de modo intuitivo y también una herramienta que los ayude a ensayar y crear cosas nuevas.

### 1.2.1 Objetivo General

Cada músico podrá ver en su dispositivo la vista de su instrumento: el cantante, la letra; el guitarrista, los acordes; el pianista, sus partituras. Tendrá autoscroll y subrayado automático del compás actual; podrá editar los tamaños de letra y acordes.

El público podrá ver y editar una variedad de canciones publicadas en el mismo sitio. Si se loguea con su usuario, podrá compartir sus canciones con otros usuarios.

Varios usuarios podrán unirse en una sesión para sincronizar la lista de canciones, la canción que se está reproduciendo y el compás actual: de este modo podrán organizar un ensayo, un concierto o una

noche de karaoke entre amigos.

### **1.2.2 Objetivos Técnicos**

La sincronización del compás en una sesión debe ser exacta cuando un grupo de músicos está tocando: un delay de 20 ms empieza a ser perceptible por el oído humano y la latencia en Internet puede ser mayor. Los distintos dispositivos se conectarán con un servidor Golang e implementarán un protocolo que combine timestamps sincronizados (basados en NTP), buffers adaptativos y compensación del jitter (variación en la latencia) para resolver esto.

El mismo servidor, además, por medio de HTTP intercambia los archivos de las canciones con las aplicaciones.

Todas las vistas deberán poder adaptarse a distintos dispositivos, ser configurables y extensibles: será posible incorporar vistas adicionales, como notación numérica para armónica, tablaturas para guitarra y reproductores multimedia como YouTube o MIDI.

Edición de letra y acordes: también podrán editar las canciones mediante una interfaz intuitiva y accesible; la compleja relación entre las letras y los acordes, que además se agrupan en partes que se repiten según una secuencia, podrá modificarse de una manera natural y sencilla.

Sincronización: varios usuarios logueados podrán unirse en una sesión para sincronizar la lista de canciones, la canción que se está reproduciendo y el estado de la reproducción.

Construir algunas herramientas necesarias para la música, como un afinador que permita afinar distintos instrumentos.

Construir herramientas para probar y "debuggear" el sistema de sincronización desarrollado.

## 2 Estado del Arte

Como es parte nuestro "negocio", describiremos algunos conceptos sobre la naturaleza del sonido y de la música.

Luego, comentaremos algunas tecnologías destacadas que tomamos en el Fogón y sobre las novedades que llegaron con internet. También, repasaremos aplicaciones web con IA que ofrecen servicios relacionados con la música.

Finalmente, explicaremos cómo nuestro enfoque es novedoso y aporta valor, comparado con las herramientas antes mencionadas.

### 2.1 Sobre el sonido y la música

El sonido es una onda mecánica que se propaga a través de un medio elástico, como el aire o el agua. Se caracteriza por propiedades físicas como la frecuencia, la amplitud y el timbre determinando la frecuencia el tono o la nota musical, la amplitud el volumen y otro la calidad del sonido.

Pitágoras (siglo VI a.C.) descubrió que lo que producía sonidos agradables entre cuerdas vibrantes era que tengan relaciones matemáticas. Ej: (2:1)Una octava, (3:2)una quinta justa. (4:3)una cuarta justa.

Esto dio origen al estudio de la armonía: la mayoría de los humanos no podemos determinar la frecuencia a la que vibra una cuerda, pero si podemos distinguir si produce un sonido agradable, y ese sonido se produce cuando hay relaciones matemáticas simples entre las frecuencias de las notas.

Pero no solo se producen sonidos agradables con relaciones matemáticas simples entre cuerdas, hay un elemento fundamental en la música además de armonía y melodía: El ritmo es la repetición de sonidos en intervalos regulares. En la música suelen organizarse de a 2, 3 o 4 tiempos, formando compases.

### 2.2 Herramientas

#### 2.2.1 Afinadores

En el siglo XVII, Marin Mersenne formuló las leyes que gobiernan la vibración de cuerdas tensas, permitiendo una base física para afinar cuerdas y abrió el camino hacia la acústica moderna.

$$f = \frac{1}{2L} \sqrt{\frac{T}{\mu}}$$

donde  $f$  es la frecuencia,  $L$  es la longitud de la cuerda,  $T$  es la tensión aplicada y  $\mu$  es la densidad lineal de masa.

Se afinaba con un diapasón, un metal que siempre sonaba con la misma frecuencia y que se tomaba como nota de referencia.

En 1939 durante la conferencia de Londres científicos y músicos acordaron que la nota La<sub>4</sub> se afinaría a 440 Hz, es decir, 440 vibraciones por segundo.

En 1980 se popularizaron afinadores digitales compactos, portátiles y precisos.

### **2.2.2 Metrónomos**

En 1815 Johann Maelzel patenta el metrónomo mecánico, ganándose el reconocimiento de Beethoven y popularizándose en toda Europa.

En el siglo XX llegaron metrónomos electrónicos, que permiten mayor precisión y funcionalidades adicionales como sonidos personalizables, luces intermitentes y conectividad con otros dispositivos.

### **2.2.3 Pentagramas y cancioneros**

15 Siglos antes que Pitágoras, por el 2000 A.C., en la Mesopotamia se usaban tablillas de arcilla para anotar música, con símbolos que representaban las alturas de las notas. Recién en el siglo IX, el monje benedictino Guido d'Arezzo desarrolló un sistema de notación musical basado en líneas y espacios, que luego evolucionó de 4 a 5 líneas. Pocos años después de la invención de la imprenta, en 1501, Ottaviano Petrucci publica "Harmonice Musices Odhecaton", con las primeras partituras impresas con tipos móviles, permitiendo la difusión masiva de la música escrita. En el siglo XIX se transmitían los acordes folclóricos de forma oral, pero para el siglo XX, en la música popular (tango, rock, jazz) se empieza a usar la notación de acordes sobre la letra de la canción. Se difunden las revistas de acordes y letras, y luego los cancioneros impresos. Con la llegada de internet, surgen diversas páginas y aplicaciones que revisamos en la próxima sección.

## **2.3 Aplicaciones Web Musicales**

Con dispositivos móviles inteligentes, escuchar un sonido y procesarlo para detectar su frecuencia es muy sencillo, igual que repetir un comportamiento cada un período constante de tiempo. Es por eso, que hubo metrónomos y afinadores en cada celular desde el comienzo de este siglo.

Las próximas líneas transcurrirán sobre aplicaciones web que son usadas actualmente por músicos, pero además de señalar solamente sus aportes, se hará foco en sus debilidades.

### **2.3.1 Cancioneros**

Los cancioneros online Reemplazaron a los folletines ofreciendo la diversidad de la web pero copiaron un defecto.

El músico está tocando su instrumento, ajustado perfecto con el metrónomo, cuando llega al último acorde de la página (o de la pantalla) y lo obliga a soltar su instrumento para manipular el cancionero, perdiendo así el ritmo.

Las páginas suelen en su mayoría estar diagramadas en la pantalla pensando en maximizar el espacio para publicidad y no en la usabilidad del músico, por lo que ofrecen poca personalización en la vista. Esto hace que aunque muchas ofrezcan instrucciones sobre cómo se realizan los acordes en el instrumento, o algunos implementen un rudimentario autoscroll, sea poco visible para el músico.

De las que tienen contenido argentino, las principales son lacuerda.net, cifraclub y acordesweb.

Y claro, muestra solo los acordes, para las paginas de acordes son paginas distintas!

### 2.3.2 Reproductores online de video y acordes

Otras aplicaciones web ofrecen videos con acordes sincronizados, como Chordify y Ultimate Guitar, ajustan la reproduccion a videos de YouTube, apenas permite editar los acordes

### 2.3.3 Editores de partituras online

Los editores de partituras online permiten crear, editar y compartir partituras musicales a través de una interfaz web. musescore.com destaca por su buen balance entre contenido gratuito y de pago y su comunidad activa de usuarios,

### 2.3.4 Reproductores online de audio y video

Como YouTube y Spotify, permiten reproducir audio y video en línea, pero no están diseñados para músicos que tocan juntos. Estas aplicaciones suelen permitir compartir una lista de reproducción.

## 2.4 El futuro llega, hace rato...

Al momento de escribir esto, diciembre de 2025, los resultados del uso de la IA vienen avanzado enorme y exponencialmente.

Logran generar todo tipo de contenido, letras y partituras, audios y videos.

Tambien permiten procesar audio para separar los instrumentos y transcribir partituras. Sobresale en ese sentido la aplicacion Moises.ai y Spleeter.

## 2.5 Conclusión

Mostramos una tabla comparativa de las herramientas presentadas:

Característica	Metronomo	Diapasón	Cancioneros	Cancioneros Web	Reprod. video y acordes	Edit. partituras online	Reprod. audio y video	Herramientas Con IA	Fogon.ar
Marca el ritmo?	Sí	No	No	No	No	No	No	Sí	Sí
Afina?	No	Sí	No	No	No	No	No	Sí	Sí
Muestra acordes?	No	No	Sí	Sí	Sí	Sí	No	Sí	Sí
Se actualiza con la canción?	No	No	No	Algunos	Sí	No	No	Sí	Sí
Muestra partitura?	No	No	No	No	No	Sí	No	Sí	Sí
Funciona sin Internet?	Sí	Sí	Sí	No	No	No	No	No	Sí
Permite estados compartidos?	No	No	No	No	No	No	Algunos	No	Sí

Cuadro 1: Comparativa de herramientas y aplicaciones musicales

Se deduce entonces que el estado compartido es un enfoque novedoso para aplicaciones web dedicadas a músicos.

La gran cantidad de tipos de aplicaciones web musicales existentes, cada una con su solución particular, muestra la necesidad de unificar herramientas para mejorar la experiencia del músico.

El criterio de privilegiar el aspecto educativo y de usabilidad está ausente en los cancioneros online y

en la mayoría de las aplicaciones web musicales.

### 3 Solución Implementada

En esta sección está la descripción de la solución implementada y una explicación técnica basada en el modelo de vistas 4+1 de Krutchen.

En la última vista, la vista de escenarios, los escribimos de igual modo que a las pruebas de aceptación en Roqroll.

Por último, incluimos una revisión de las tecnologías utilizadas.

#### 3.1 WWW.FOGON.AR

Es una aplicación progresiva (PWA) y multiplataforma que permite buscar y tocar canciones, y también crear, editar y compartirlas.

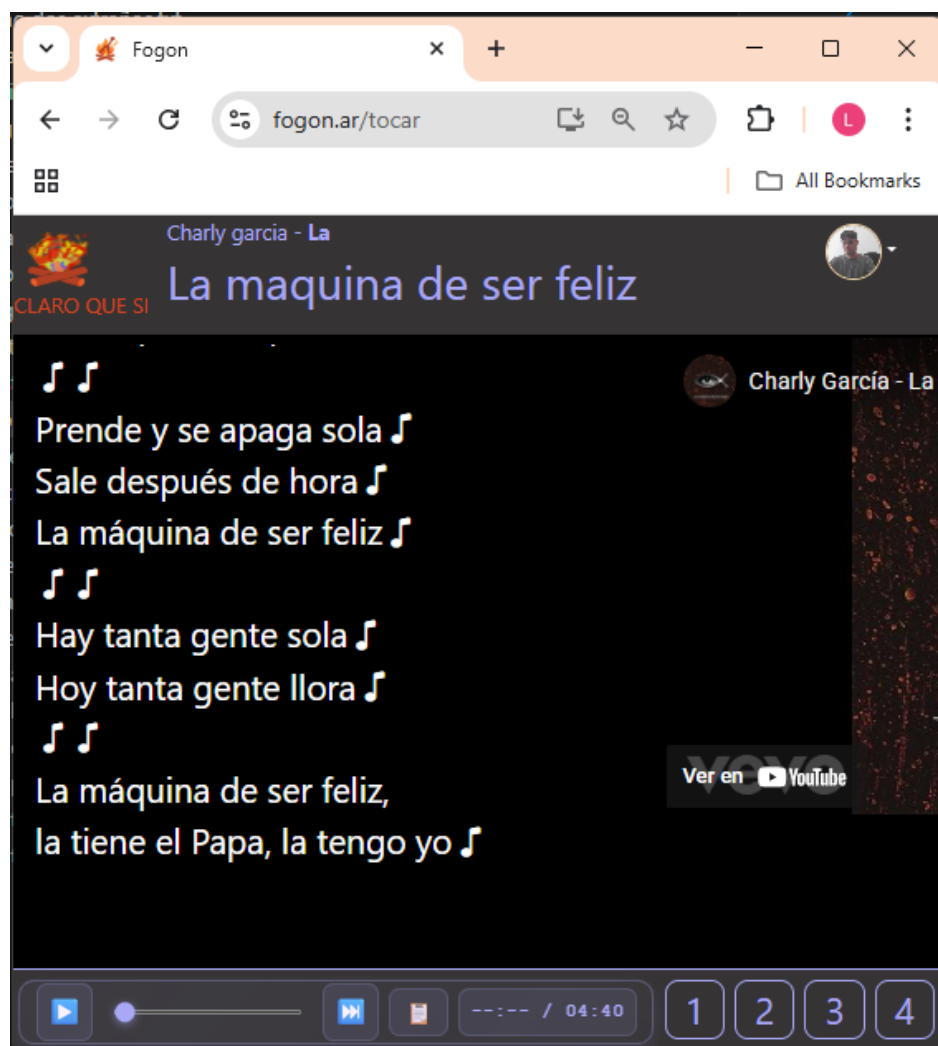


Figura 1: FIG 1: "La maquina de ser feliz", tocando con un video

Además, Permite crear sesiones colaborativas para llevar en un estado compartido el compás y la canción, de modo que cada músico vea las instrucciones para su instrumento, en su dispositivo, de manera sincronizada.

Administra la lista de reproducción y listas en general, guardandolas localmente o en el servidor.

Ofrece tambien otras herramientas utiles para los musicos, como un afinador y cambios automáticos de escala.

### **3.1.1 tocar**

La pagina tocar muestra la vista del instrumento para cada musico y muestra el controlador del tiempo (Como un metronomo).

**vistas**

**buscar**

**editar**

**compartir**

### **3.1.2 tocar**

## **3.2 Arquitectura del Sistema**

Descripción de la arquitectura general del sistema.

### **3.2.1 Vista Lógica**

qué funcionalidades ofrece el sistema

- Componente 1
- Componente 2
- Componente 3

### **3.2.2 Vista de procesos**

cómo se comporta dinámicamente

- Componente 1
- Componente 2
- Componente 3

### **3.2.3 Vista Física**

cómo se despliega en hardware (nodos, servidores, redes).

- Componente 1
- Componente 2
- Componente 3



### 3.2.4 Vista desarrollo

cómo se organiza el software para los desarrolladores (módulos, paquetes, capas).

- Componente 1
- Componente 2
- Componente 3

## 3.3 Tecnologías Utilizadas

Listar y justificar las tecnologías seleccionadas.

## 3.4 Implementación

Describir cómo se implementó la solución propuesta.

Listing 1: Ejemplo de código

```
1 def funcion_ejemplo():  
2     print("Hola, mundo!")  
3     return True
```

Figura 2: Descripción de la figura

## **4 Metodología Aplicada**

Describir la metodología empleada para el desarrollo del trabajo.

### **4.1 Marco Metodológico**

Explicar el marco metodológico utilizado (ágil, cascada, etc.).

### **4.2 Proceso de Desarrollo**

Detallar las etapas del proceso de desarrollo:

1. Análisis de requisitos
2. Diseño de la solución
3. Implementación
4. Pruebas
5. Despliegue

### **4.3 Herramientas de Desarrollo**

- Herramienta 1
- Herramienta 2
- Herramienta 3

### **4.4 Gestión del Proyecto**

Describir cómo se gestionó el proyecto (sprints, reuniones, etc.).

## 5 Experimentación y Validación

Presentar los experimentos realizados y la validación de la solución implementada.

### 5.1 Diseño de Experimentos

Describir el diseño de los experimentos realizados para validar la solución.

### 5.2 Casos de Prueba

Detallar los casos de prueba ejecutados.

### 5.3 Resultados Obtenidos

Presentar los resultados obtenidos del trabajo realizado.

Caso de Prueba	Resultado Esperado	Resultado Obtenido
Prueba 1	Éxito	Éxito
Prueba 2	Éxito	Éxito

Cuadro 2: Resultados de las pruebas realizadas

### 5.4 Análisis de Resultados

Analizar e interpretar los resultados obtenidos.

### 5.5 Validación con Usuarios

Describir el proceso de validación con usuarios finales.

## 6 Cronogramas

Presentar el cronograma de actividades del proyecto.

### 6.1 Cronograma Planificado

Descripción del cronograma inicial planificado para el proyecto.

Actividad	Duración	Período
Análisis de requisitos	2 semanas	Enero
Diseño	3 semanas	Febrero
Implementación	8 semanas	Marzo-Abril
Pruebas	2 semanas	Mayo
Documentación	1 semana	Mayo

Cuadro 3: Cronograma planificado del proyecto

### 6.2 Cronograma Real

Descripción del cronograma real de ejecución del proyecto.

### 6.3 Desviaciones y Ajustes

Análisis de las desviaciones respecto al plan original y los ajustes realizados.

## 7 Riesgos y Lecciones Aprendidas

Análisis de los riesgos identificados durante el proyecto y las lecciones aprendidas.

### 7.1 Identificación de Riesgos

Listar y describir los riesgos identificados al inicio del proyecto.

Riesgo	Probabilidad	Impacto
Riesgo técnico 1	Alta	Alto
Riesgo de recursos	Media	Medio
Riesgo de tiempo	Baja	Alto

Cuadro 4: Matriz de riesgos del proyecto

### 7.2 Gestión de Riesgos

Describir cómo se gestionaron los riesgos durante el proyecto.

### 7.3 Problemas Encontrados

Detallar los problemas principales enfrentados durante el desarrollo.

### 7.4 Lecciones Aprendidas

Compartir las lecciones aprendidas durante la ejecución del proyecto:

- Lección 1
- Lección 2
- Lección 3

## **8 Impactos Sociales y Ambientales del Proyecto**

Análisis de los impactos sociales y ambientales del proyecto desarrollado.

### **8.1 Impacto Social**

Describir el impacto social esperado o generado por el proyecto:

- Beneficios para la comunidad
- Mejoras en la calidad de vida
- Accesibilidad y inclusión

### **8.2 Impacto Ambiental**

Analizar el impacto ambiental del proyecto:

- Consumo de recursos
- Huella de carbono
- Sostenibilidad de la solución

### **8.3 Responsabilidad Social y Ética**

Consideraciones éticas y de responsabilidad social del proyecto.

### **8.4 Medidas de Mitigación**

Describir las medidas implementadas para minimizar impactos negativos.

## **9 Desarrollos Futuros**

Describir posibles extensiones, mejoras y trabajos futuros relacionados con el proyecto.

### **9.1 Mejoras Propuestas**

Listar las mejoras que podrían implementarse en el futuro:

- Mejora 1
- Mejora 2
- Mejora 3

### **9.2 Funcionalidades Adicionales**

Describir funcionalidades adicionales que podrían agregarse.

### **9.3 Escalabilidad**

Analizar cómo el sistema podría escalarse para soportar mayor carga o alcance.

### **9.4 Investigación Futura**

Proponer líneas de investigación futuras relacionadas con el proyecto.

### **9.5 Conclusiones Finales**

Presentar las conclusiones finales del trabajo, resumiendo los principales hallazgos y logros alcanzados.

## Referencias

## Referencias

- [1] A. S. Tanenbaum and M. Van Steen, *Distributed Systems: Principles and Paradigms*, 3rd ed. Boston, MA: Pearson, 2017.