



---

# PIXCHAT

---



Integrantes:

Paloma Osiris Báez Lara  
Alejandro Sánchez Marín  
Juan Reyes María Teresa

12 DE ENERO DE 2025

# Contenido

Introducción .....	2
Requerimientos .....	2
Casos de uso.....	2
Prototipos de UI .....	3
Requerimientos funcionales .....	4
Requerimientos no funcionales .....	4
Diseño.....	5
Diseño Arquitectónico .....	5
Vista de casos de uso .....	5
Vista lógica .....	5
Vista de despliegue .....	6
Descripciones de casos de uso.....	7
Construcción.....	9
Explicación de la solución .....	9
Repositorios de los proyectos .....	10
Estrategia de despliegue .....	10
Conclusiones .....	11

## Introducción

Este documento presenta el desarrollo de PixChat, un sistema que combina un lienzo interactivo para dibujo con un chat en tiempo real, diseñado para fomentar la creatividad y la colaboración entre usuarios. Las funcionalidades principales incluyen dibujar en el lienzo, limpiar o borrar su contenido, insertar figuras como rectángulos y óvalos, elegir colores, descargar el lienzo como imagen, y enviar mensajes en el chat, con la opción de compartir los lienzos creados.

El contenido está organizado en secciones clave: los requerimientos funcionales y no funcionales del sistema, los casos de uso que definen su alcance, los prototipos de la interfaz de usuario, el diseño arquitectónico y lógico, y las descripciones de los casos de uso. Además, se documentan los detalles de construcción, incluyendo los repositorios del proyecto, y la estrategia de despliegue para garantizar su implementación efectiva. Este documento funcionará como guía para el equipo de desarrollo y para cualquier individuo que participe en la creación.

## Requerimientos

### Casos de uso

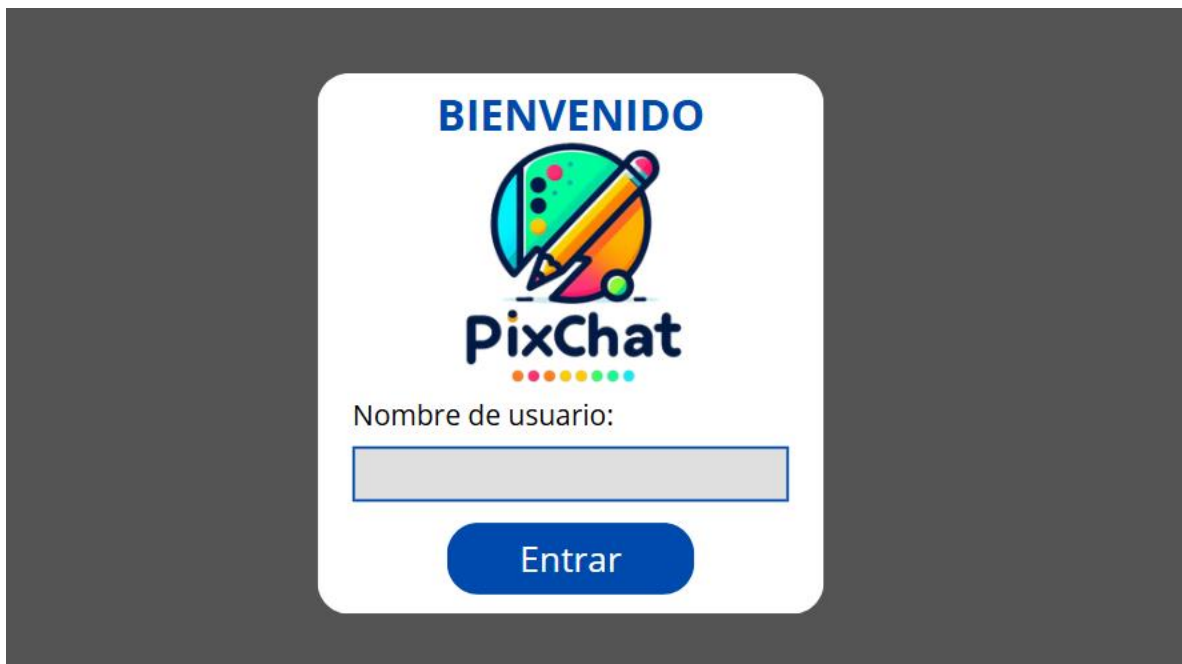
A nivel de análisis se lograron identificar los siguientes casos de uso:

- 1.- Dibujar en lienzo
- 2.- Mandar mensaje en chat

Ambos casos de uso incluyen varios casos de flujos alternos.

## Prototipos de UI

Los prototipos realizados se muestran a continuación:



## Requerimientos funcionales

Estos requisitos detallan las funciones específicas que el sistema debe realizar para satisfacer las necesidades del usuario. A continuación, se presenta una lista de los requisitos funcionales clave para el sistema PixChat:

**RF-01.** El sistema debe permitir al usuario dibujar trazos libres en un lienzo.

**RF-02.** El sistema debe permitir borrar trazos o secciones del lienzo con una herramienta de borrador.

**RF-03.** El sistema debe ofrecer herramientas para crear formas básicas como rectángulos y óvalos.

**RF-04.** El sistema debe permitir al usuario seleccionar colores para dibujar.

**RF-05.** El sistema debe incluir una función para limpiar completamente el lienzo.

**RF-06.** El sistema debe permitir al usuario descargar el lienzo actual como una imagen en formato PNG.

**RF-07.** El sistema debe incluir un chat en tiempo real que permita a los usuarios enviar mensajes de texto entre ellos.

**RF-08.** El sistema debe permitir compartir el lienzo actual con otros usuarios a través del chat.

**RF-09.** El sistema debe diferenciar entre usuarios activos en el chat y mostrar sus nombres o pseudónimos.

**RF-10.** El sistema debe mostrar el historial de conversaciones para los usuarios activos en una sesión.

## Requerimientos no funcionales

Los requisitos no funcionales establecen los criterios de calidad y las condiciones en las que el sistema operará. La siguiente lista proporciona una visión clara de estos aspectos no funcionales que son necesarios para el sistema PixChat.

### Rendimiento:

**RNF – 01.** El sistema debe ser capaz de procesar trazos y herramientas de dibujo en tiempo real sin retrasos perceptibles para el usuario.

**RNF – 02.** El sistema debe permitir que el chat entregue mensajes en menos de 3 segundo en condiciones de conexión estable.

## Usabilidad:

**RNF – 03.** El sistema debe tener una interfaz intuitiva y sencilla de usar para usuarios de cualquier nivel técnico.

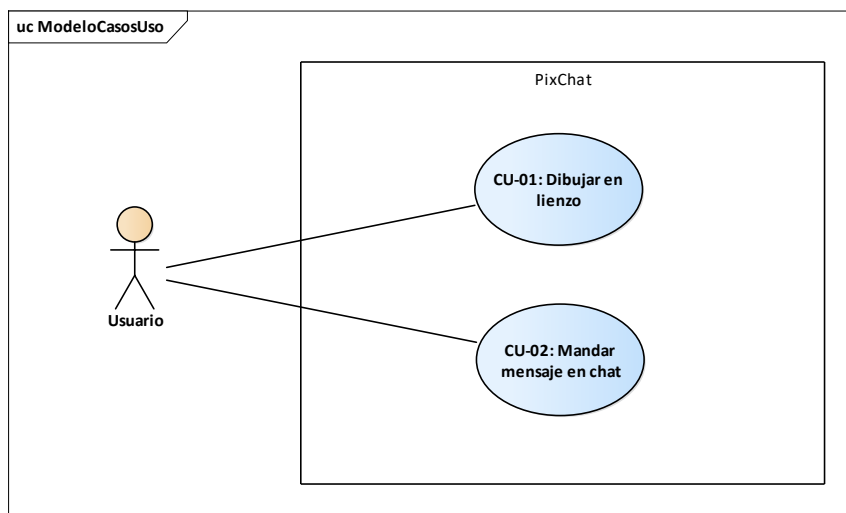
**RNF – 04.** El sistema debe tener botones y herramientas claramente identificados con íconos y etiquetas.

## Diseño

### Diseño Arquitectónico

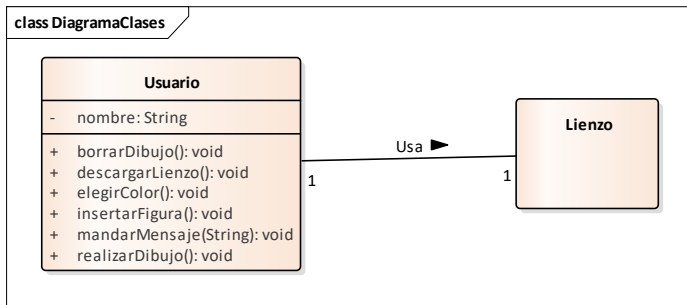
#### Vista de casos de uso

El diagrama de casos de uso muestra la interacción del usuario con el sistema PixChat. Los usuarios pueden dibujar en un lienzo, eso implica todas las funciones de edición básicas como borrar o elegir colores, asimismo, puede interactuar mediante un chat donde además de enviar mensajes a los demás usuarios conectados, puede compartir el lienzo de su dibujo.



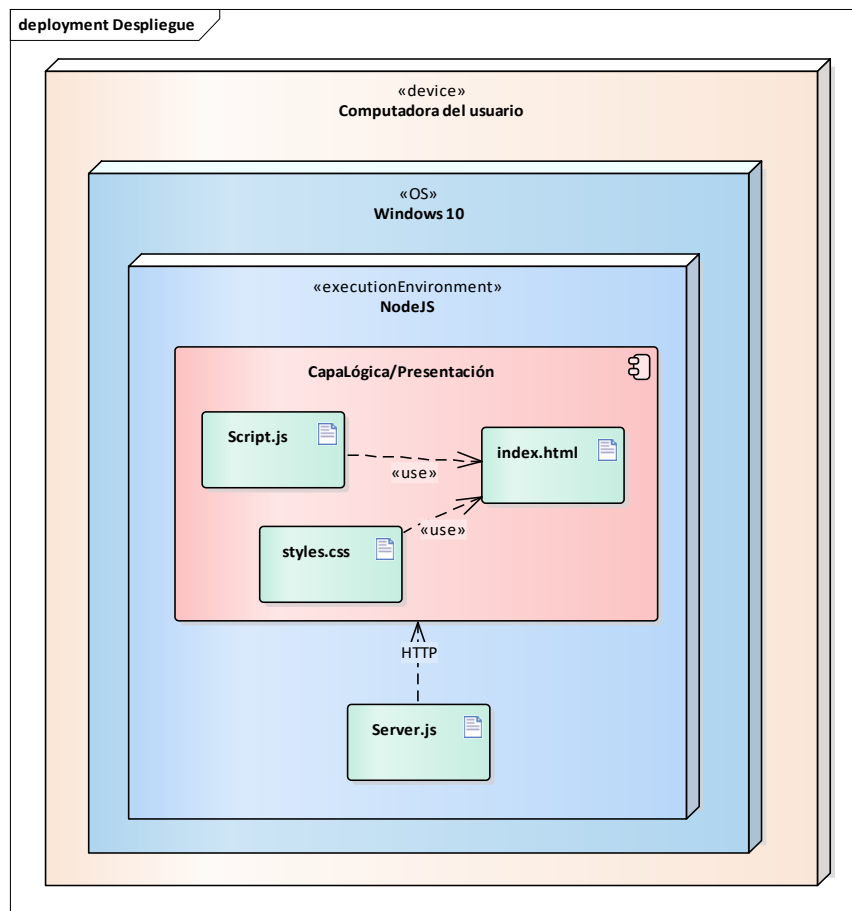
#### Vista lógica

El diagrama de clases de PixChat muestra la estructura y relaciones entre sus principales componentes. Siendo la clase usuario la principal, donde este es capaz de borrar dibujo, descargar lienzo, elegir color, insertar figura, mandar mensaje y realizar dibujo; a la segunda clase que es Lienzo, la relación entre ambas clases es uno a uno, donde un usuario usa un lienzo.



## Vista de despliegue

Este diagrama de despliegue muestra cómo se implementa el sistema PixChat en un entorno cliente-servidor. El sistema se ejecuta en una computadora del usuario con Windows 10 y utiliza Node.js como entorno de ejecución. La capa de presentación incluye los archivos index.html (interfaz principal), styles.css (diseño visual) y Script.js (lógica de la interfaz), los cuales trabajan en conjunto. Por su parte, el archivo Server.js actúa como el servidor que maneja las solicitudes HTTP y proporciona datos a la capa de presentación. La comunicación entre el cliente y el servidor ocurre mediante el protocolo HTTP.



## Descripciones de casos de uso

<b>ID:</b>	CU-01
<b>Nombre del CU:</b>	Dibujar en lienzo
<b>Descripción:</b>	El usuario desea dibujar sobre el canva.
<b>Actor(es):</b>	Usuario (Primario)
<b>Disparador:</b>	El Usuario selecciona el icono del lápiz.
<b>Precondiciones:</b>	PRE-01: El <b>usuario</b> ha ingresado su nombre en el Login.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. PixChat muestra el lienzo y la opción de dibujar en la barra de herramientas.</li> <li>2. El USUARIO realiza un trazo sobre el lienzo.</li> <li>3. PixChat dibuja el trazo que realizo el <b>usuario</b>.</li> <li>4. Fin del caso de uso.</li> </ol>
<b>Flujos Alternos:</b>	<p>FA-01: Borrar parte del dibujo:</p> <ol style="list-style-type: none"> <li>1. El USUARIO, selecciona el icono de la goma.</li> <li>2. PixChat borra la parte del trazo por donde pasa el cursor el <b>usuario</b>.</li> <li>3. Regresa al paso 2 del flujo normal.</li> </ol> <p>FA-02: Insertar rectángulo:</p> <ol style="list-style-type: none"> <li>1. El USUARIO, selecciona el icono del rectángulo.</li> <li>2. PixChat dibuja un rectángulo del tamaño que se define por el cursor del <b>usuario</b>.</li> <li>3. Regresa al paso 2 del flujo normal.</li> </ol> <p>FA-03: Insertar círculo:</p> <ol style="list-style-type: none"> <li>1. El USUARIO, selecciona el icono del círculo.</li> <li>2. PixChat dibuja un círculo del tamaño que se define por el cursor del <b>usuario</b>.</li> <li>3. Regresa al paso 2 del flujo normal.</li> </ol> <p>FA-04: Limpiar lienzo:</p> <ol style="list-style-type: none"> <li>1. El USUARIO, selecciona el icono de eliminar.</li> <li>2. PixChat limpia el lienzo de todos los trazos hechos previamente por el <b>usuario</b>.</li> <li>3. Regresa al paso 2 del flujo normal.</li> </ol> <p>FA-05: Guardar dibujo:</p> <ol style="list-style-type: none"> <li>1. El USUARIO, selecciona el icono de guardar.</li> <li>2. PixChat crea una imagen PNG del lienzo y la descarga automáticamente en el equipo del <b>usuario</b>.</li> <li>3. Regresa al paso 2 del flujo normal.</li> </ol> <p>FA-06: Elegir color:</p> <ol style="list-style-type: none"> <li>1. El USUARIO, selecciona la opción elegir color.</li> <li>2. PixChat muestra un selector de color.</li> <li>3. El USUARIO elije un color.</li> </ol>



	<ol style="list-style-type: none"> <li>PixChat, actualiza el color al seleccionado por el <b>usuario</b>.</li> <li>Regresa al paso 2 del flujo normal.</li> </ol>
<b>Excepciones:</b>	<p>EX-01: No hay conexión con el servidor</p> <ol style="list-style-type: none"> <li>PixChat muestra una alerta con el mensaje “Se perdió la conexión con el servidor_</li> <li>El USUARIO selecciona la opción “Aceptar” de la alerta.</li> <li>PixChat cierra la alerta.</li> <li>Fin del caso de uso.</li> </ol>
<b>Postcondiciones:</b>	POST-01: El dibujo realizado por el usuario queda almacenado temporalmente en el lienzo hasta que se guarde, edite o limpie el lienzo.
<b>Incluye:</b>	N/A
<b>Extiende:</b>	N/A

<b>ID:</b>	CU-02
<b>Nombre del CU:</b>	Mandar mensaje en chat
<b>Descripción:</b>	El usuario desea mandar un mensaje en el chat.
<b>Actor(es):</b>	Usuario (Primario)
<b>Disparador:</b>	El Usuario selecciona el icono de chat.
<b>Precondiciones:</b>	PRE-01: El <b>usuario</b> ha ingresado su nombre en el Login.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>PixChat muestra un chat con espacio para escribir un mensaje y dos opciones, “Enviar mensaje” y “Compartir lienzo”</li> <li>El USUARIO ingresa un mensaje y da clic en “Enviar mensaje”.</li> <li>PixChat envia el mensaje que escribió el <b>usuario</b>, a todos los usuarios conectados.</li> <li>Fin del caso de uso.</li> </ol>
<b>Flujos Alternos:</b>	<p>FA-01: Compartir lienzo:</p> <ol style="list-style-type: none"> <li>El USUARIO, selecciona el icono de compartir lienzo.</li> <li>PixChat manda al chat, una captura actual del lienzo del <b>usuario</b>.</li> <li>Regresa al paso 2 del flujo normal.</li> </ol>
<b>Excepciones:</b>	<p>EX-01: No hay conexión con el servidor</p> <ol style="list-style-type: none"> <li>PixChat muestra una alerta con el mensaje “Se perdió la conexión con el servidor_</li> <li>El USUARIO selecciona la opción “Aceptar” de la alerta.</li> <li>PixChat cierra la alerta.</li> <li>Fin del caso de uso.</li> </ol>
<b>Postcondiciones:</b>	POST-01: El mensaje enviado por el usuario es recibido correctamente por todos los usuarios conectados al chat.
<b>Incluye:</b>	N/A
<b>Extiende:</b>	N/A

# Construcción

## Explicación de la solución

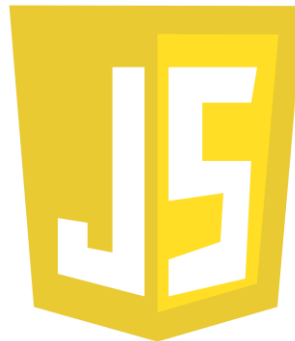
Debido a que se trataba de un proyecto principalmente del lado del cliente, se decidió llevar a cabo la construcción de este sin ningún framework de apoyo, haciendo únicamente de las tecnologías de HTML, CSS Y JavaScript. Se requirió de dependencias que provee el entorno de ejecución de NodeJS como ExpressJs y Socket.io para hacer la comunicación con sockets web. Se decidió hacer todo en el mismo proyecto debido a que es un proyecto pequeño por lo tanto únicamente se separó el cliente y el servidor a nivel de capas, el servidor quedando en la raíz y el cliente en una carpeta.

Para llevar a cabo el despliegue, se hizo uso de Podman el cual no represento dificultades a excepción de la investigación de como exponer más de un puerto.

En general, la construcción de este proyecto no significo un gran problema, pero si ayudo al reforzamiento de los conocimientos.

Tecnologías para el desarrollo empleadas:

**HTML**



**CSS**



## Repositorios de los proyectos

A continuación, se añade el repositorio del proyecto de PixChat.

Cliente web:

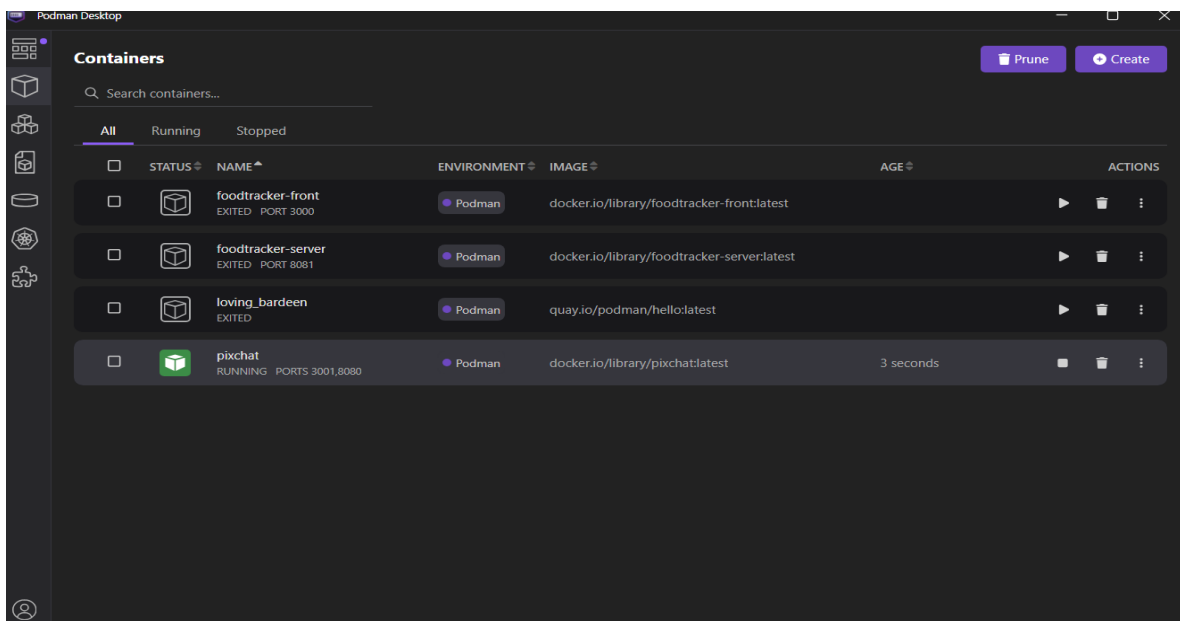
<https://github.com/Alejandrin08/PixChat.git>

## Estrategia de despliegue

La estrategia de despliegue del proyecto se basó en el uso de Podman para la creación y gestión de imágenes de contenedores. Estas imágenes permiten la distribución del proyecto a diferentes usuarios, asegurando la portabilidad y la consistencia del entorno de ejecución de la aplicación. La construcción de las imágenes se realizó localmente mediante el comando `podman build`, utilizando un archivo `Dockerfile` en el proyecto de PixChat, este `Dockerfile` tuvo que ser adaptado para su construcción con NodeJS.

Tras crear el contenedor, la imagen resultante se subió a Docker Hub para facilitar su acceso y uso por parte de otros desarrolladores.

En general, la elección de Podman como base de la estrategia fue adecuada debido a su similitud con Docker, lo que permitió reutilizar conocimientos previos sin una curva de aprendizaje significativa.



## Conclusiones

En conclusión, este proyecto significo el reforzamiento de conocimientos debido a la magnitud de este, al ser en su mayoría un proyecto del lado del cliente no se requirió de un gran esfuerzo. A pesar de esto, consideramos que PixChat cumple con lo requerido y es funcional para llevar a cabo el dibujo y el lograr compartirlos consiguiendo que exista una comunicación entre los clientes. La construcción de este trabajo logro cumplir con los requerimientos definidos y se logró realizar una distribución de este exitosamente con Podman.