

BUBBLEBOX

Manual técnico del sistema

Integrantes:

Juan Andrés Toro Blandón

Mariana Andrea Nisperuza Puerta

Yenifer Tamayo Villa

Centro tecnológico del mobiliario Itagüí

Fecha:

25/02/2025

CONTENIDO

PRESENTACIÓN	5
RESUMEN.....	6
OBJETIVO	7
FINALIDAD DEL MANUAL.....	7
INTRODUCCIÓN	8
ASPECTOS TECNICOS	9
HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO	10
DIAGRAMAS DE MODELAMIENTO.....	11
DIAGRAMA DE CLASES.....	11
DIAGRAMA DE CASOS DE USO	13
DICCIONARIO DE DATOS	16
ASPECTO TECNICO DEL DESARROLLO DEL SISTEMA.....	¡Error! Marcador no definido.
MODIFICACIÓN LOCAL.....	¡Error! Marcador no definido.
REQUERIMIENTOS DEL SOFTWARE	¡Error! Marcador no definido.
REQUISITOS MINIMOS.....	¡Error! Marcador no definido.
BIBLIOGRAFIA	26

TABLA DE FIGURAS

<i>Ilustración 1 Diagrama de clases</i>	11
<i>Ilustración 2 Diagrama caso de uso (usuario)</i>	13
<i>Ilustración 3 Diagrama casos de uso (Administrador)</i>	14
<i>Ilustración 4 Diagrama casos de uso (notificaciones)</i>	14
<i>Ilustración 5 Pagina web descarga NodeJS</i>	¡Error! Marcador no definido.
<i>Ilustración 6 Pagina web descarga Git</i>	¡Error! Marcador no definido.
<i>Ilustración 7 Pagina web descarga VSCODE</i>	¡Error! Marcador no definido.
<i>Ilustración 8 Configuración clave SSH github part1</i>	¡Error! Marcador no definido.
<i>Ilustración 9 Clonación repositorio</i>	¡Error! Marcador no definido.
<i>Ilustración 10 Instalación dependencias part1</i>	¡Error! Marcador no definido.
<i>Ilustración 11 Instalación dependencias part2</i>	¡Error! Marcador no definido.
<i>Ilustración 12 Instalación dependencias part3</i>	¡Error! Marcador no definido.
<i>Ilustración 13 Instalación dependencias part4</i>	¡Error! Marcador no definido.

LISTA DE TABLAS

<i>Tabla 1 Diccionario de datos modelo Usuarios</i>	16
<i>Tabla 2 Diccionario de datos modelo publicaciones</i>	17
<i>Tabla 3 Diccionario de datos modelo comunidades</i>	17
<i>Tabla 4 Diccionario de datos modelo usuarios_comunidades</i>	18
<i>Tabla 5 Diccionario de datos modelo reels</i>	18
<i>Tabla 6 Diccionario de datos modelo historias</i>	18
<i>Tabla 7 Diccionario de datos modelo vistas_historias</i>	19
<i>Tabla 8 Diccionario de datos modelo reacciones</i>	19
<i>Tabla 9 Diccionario de datos modelo comentarios</i>	20
<i>Tabla 10 Diccionario de datos modelo amistades</i>	20
<i>Tabla 11 Diccionario de datos modelo mensajes</i>	21
<i>Tabla 12 Diccionario de datos modelo grupos</i>	21
<i>Tabla 13 Diccionario de datos modelo miembros_grupos</i>	21
<i>Tabla 14 Diccionario de datos modelo mensajes_grupos</i>	22
<i>Tabla 15 Diccionario de datos modelo notificaciones</i>	22
<i>Tabla 16 Diccionario de datos modelo respuestas_comentarios_publicaciones</i>	23
<i>Tabla 17 Diccionario de datos modelo respuestas_comentarios_reels</i>	23
<i>Tabla 18 Diccionario de datos modelo backups</i>	23
<i>Tabla 19 Diccionario de datos modelo backups_restores</i>	24
<i>Tabla 20 Diccionario de datos modelo reportes</i>	24
<i>Tabla 21 Diccionario de datos modelo intereses</i>	25
<i>Tabla 22 Diccionario de datos modelo configuraciones_usuario</i>	25

PRESENTACIÓN

El avance de la tecnología ha revolucionado la manera en que las personas se comunican, interactúan y colaboran en distintos ámbitos. En este contexto, Bubblebox surge como una red social innovadora diseñada para facilitar la conexión entre individuos, sin importar su experiencia o nivel de conocimientos, brindando un espacio ideal para el intercambio de ideas, la colaboración en proyectos y el aprendizaje mutuo.

Con el propósito de garantizar un uso eficiente y una gestión adecuada de la plataforma, se ha desarrollado el presente manual, el cual proporciona toda la información necesaria para la instalación, mantenimiento y exploración de Bubblebox. Este documento está dirigido tanto a desarrolladores como a administradores de la plataforma, brindando una referencia completa que permite comprender la arquitectura del sistema y realizar modificaciones o mejoras según sea necesario.

El manual ofrece una guía detallada sobre la estructura del software, lo que facilita a los desarrolladores que trabajan con ReactJS y NodeJS comprender el funcionamiento de la plataforma y llevar a cabo tareas de desarrollo y optimización de manera eficiente. Asimismo, se proporciona información relevante sobre las tecnologías utilizadas, los diagramas de modelado de datos y la configuración del entorno de desarrollo, aspectos fundamentales para la evolución y mejora continua del sistema.

La documentación aquí presentada tiene como objetivo garantizar la sostenibilidad y escalabilidad de Bubblebox, permitiendo que su comunidad de usuarios disfrute de una experiencia fluida, segura y enriquecedora. Se espera que este manual sirva como una herramienta de referencia clave para todos aquellos que contribuyan al crecimiento y fortalecimiento de la plataforma.

RESUMEN

Este manual proporciona una descripción detallada de los aspectos técnicos e informáticos del software **Bubblebox**, con el objetivo de explicar su estructura a quienes deseen administrarlo, editarlo o configurarlo.

El documento está organizado en secciones que abarcan las herramientas utilizadas en el desarrollo del software, incluyendo una explicación paso a paso de su implementación. Además, se detallan las distintas funcionalidades de la plataforma, especificando los requisitos de hardware y software necesarios para su correcto funcionamiento.

Asimismo, se ofrecen recomendaciones para el uso adecuado del sistema de información, garantizando su eficiencia, seguridad y escalabilidad.

OBJETIVO

Dar a conocer el uso adecuado del software Bubblebox, proporcionando una descripción detallada e ilustrada de los componentes y funcionalidades que garantizan el correcto funcionamiento del sistema de información.

FINALIDAD DEL MANUAL

Este manual técnico tiene como finalidad instruir a las personas que deseen administrar, editar o configurar el software Bubblebox, brindando las herramientas y conocimientos necesarios para su correcta gestión y mantenimiento.

INTRODUCCIÓN

El presente manual ha sido elaborado con el propósito de proporcionar una guía detallada sobre el software Bubblebox, abordando sus aspectos técnicos para que cualquier persona encargada de su administración, edición o configuración pueda hacerlo de manera eficiente y apropiada.

El documento está estructurado en diferentes secciones que permiten comprender a profundidad el funcionamiento del sistema de información:

- **Aspectos Teóricos:** Se presentan conceptos, definiciones y explicaciones de los componentes del software desde una perspectiva teórica, facilitando el entendimiento del sistema y sus herramientas.
- **Diagramas de Modelamiento:** Incluye diagramas e ilustraciones que representan visualmente el funcionamiento del aplicativo.
- **Aspecto Técnico del Desarrollo del Sistema:** Proporciona información detallada sobre los componentes del software desde un enfoque técnico, abarcando el almacenamiento de datos, la estructura de desarrollo y recomendaciones para su correcto uso.
- **Requerimientos del Software:** Especifica los requisitos básicos de hardware y software necesarios para el correcto funcionamiento de Bubblebox.

ASPECTOS TECNICOS

El software Bubblebox ha sido desarrollado con el objetivo de facilitar la conexión entre personas, fomentando la interacción y el aprendizaje colaborativo. Para garantizar su correcto funcionamiento, es fundamental comprender los aspectos técnicos que conforman su estructura y asegurar su adecuado mantenimiento.

Se recomienda que este manual sea utilizado únicamente por aquellas personas encargadas de administrar, editar o configurar Bubblebox, con el fin de preservar la integridad y seguridad de los datos almacenados en la base de datos, evitando posibles usos indebidos de la información.

A continuación, se detallan los principales aspectos técnicos del sistema:

- **Arquitectura del Software:** Bubblebox es una aplicación web desarrollada con ReactJS para el frontend y Node.js para el backend, utilizando MySQL como sistema de gestión de bases de datos.
- **Seguridad y Accesibilidad:** Se han implementado medidas de seguridad como autenticación de usuarios, encriptación de contraseñas y control de accesos para proteger la información almacenada en la base de datos.
- **Gestión de Datos:** La base de datos ha sido diseñada para almacenar información de usuarios, publicaciones, interacciones y demás datos relevantes para el funcionamiento de la plataforma.
- **Requisitos de Hardware y Software:** Se especifican las configuraciones mínimas y recomendadas necesarias para ejecutar el software de manera óptima.
- **Mantenimiento y Actualización:** Se brindan recomendaciones para la actualización del sistema, resolución de problemas y optimización del rendimiento.

HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO

En esta sección se procede a explicar las herramientas informáticas empleadas para el desarrollo del aplicativo:

VISUAL STUDIO CODE

VSCode es un editor de código ligero, multiplataforma y altamente extensible, desarrollado por Microsoft. Ofrece soporte integrado para múltiples lenguajes de programación, resaltado de sintaxis, autocompletado inteligente y depuración avanzada. Además, permite la instalación de extensiones para mejorar la productividad, como herramientas para ReactJS, Node.js, Git y control de versiones.

GIT Y GITHUB

Git es un sistema de control de versiones distribuido que permite gestionar el código fuente de manera eficiente, facilitando el seguimiento de cambios y la colaboración entre desarrolladores. GitHub, por su parte, es una plataforma basada en la nube que proporciona repositorios remotos para almacenar proyectos, colaborar en código, realizar revisiones y gestionar versiones de software de manera organizada y segura.

POSTMAN

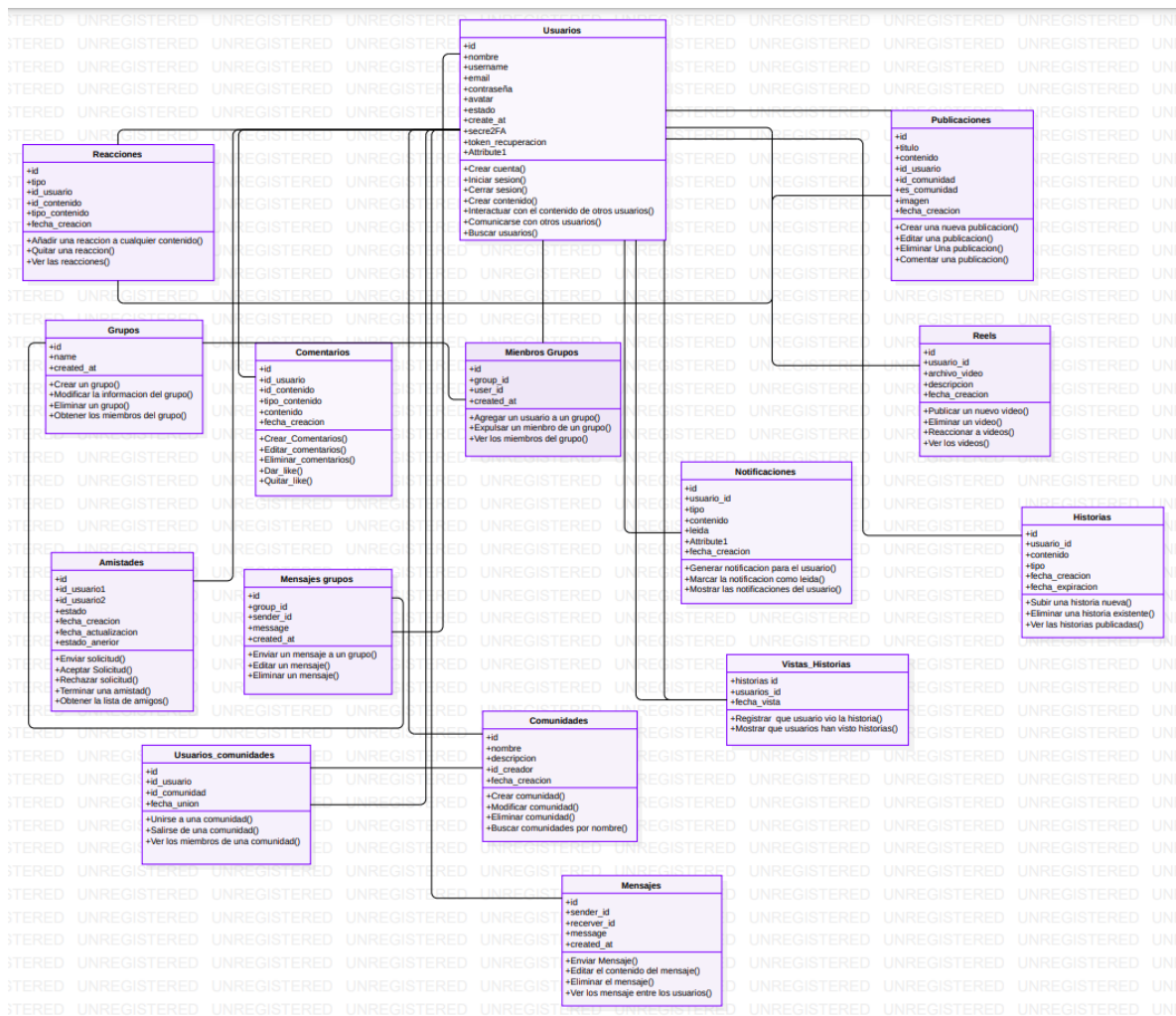
Postman es una herramienta utilizada para el desarrollo y prueba de APIs. Permite enviar solicitudes HTTP, visualizar respuestas, gestionar entornos y automatizar pruebas, facilitando la comunicación entre el frontend y el backend. Es ampliamente utilizada en el desarrollo de aplicaciones web y RESTful APIs, ya que simplifica la verificación del correcto funcionamiento de los servicios.

DIAGRAMAS DE MODELAMIENTO

DIAGRAMA DE CLASES

El diagrama de clases está compuesto de las entidades y atributos que se crearon para el almacenamiento de datos del software.

Ilustración 1 Diagrama de clases



En la ilustración 1, se detalla cada una de las entidades (modelos) utilizadas para el almacenamiento de datos del aplicativo, especificando sus relaciones y las funciones que cada una desempeña dentro del sistema.

USUARIOS: Representa a los usuarios registrados en la plataforma. Almacena información como nombre, correo electrónico, contraseña y estado. Además, contiene funciones para crear, actualizar y buscar usuarios, así como para gestionar sus interacciones con otros usuarios y contenidos.

PUBLICACIONES: Modelo encargado de almacenar las publicaciones de los usuarios. Contiene atributos como el ID del usuario, comunidad asociada y el contenido de la publicación. Permite la creación, edición y eliminación de publicaciones, además de gestionar los comentarios.

REACCIONES: Permite a los usuarios reaccionar a distintos tipos de contenido dentro de la plataforma, como publicaciones y comentarios. Almacena el tipo de reacción y a qué contenido está asociada.

COMENTARIOS: Modelo destinado al almacenamiento de comentarios en publicaciones u otros elementos. Contiene la información del usuario que comenta, el contenido del comentario y la publicación a la que pertenece.

GRUPOS: Representa los grupos dentro del sistema. Contiene el nombre del grupo y permite la gestión de la creación y administración de los mismos.

MIEMBROS_GRUPOS: Almacena la relación entre los usuarios y los grupos a los que pertenecen, permitiendo agregar o eliminar usuarios de un grupo.

MENSAJES_GRUPOS: Maneja la mensajería dentro de los grupos, permitiendo el envío y recepción de mensajes entre los miembros.

MENSAJES: Representa los mensajes privados entre usuarios. Contiene el remitente, destinatario y contenido del mensaje.

AMISTADES: Modelo que gestiona las solicitudes y relaciones de amistad entre los usuarios, permitiendo enviar solicitudes, aceptar o rechazar amigos.

COMUNIDADES: Representa las comunidades dentro de la red social. Almacena el nombre, descripción y fecha de creación, permitiendo a los usuarios unirse o salir de ellas.

USUARIOS_COMUNIDADES: Modelo intermedio que relaciona a los usuarios con las comunidades a las que pertenecen.

NOTIFICACIONES: Almacena las notificaciones generadas dentro del sistema, permitiendo a los usuarios estar informados de eventos importantes como reacciones, comentarios o solicitudes de amistad.

HISTORIAS: Representa las historias publicadas por los usuarios, almacenando el contenido y su tiempo de duración.

VISTAS_HISTORIAS: Modelo que rastrea qué usuarios han visto una historia específica.

REELS: Contiene información sobre los videos cortos creados y compartidos por los usuarios dentro de la plataforma.

DIAGRAMA DE CASOS DE USO

En las siguientes ilustraciones, se detallan los papeles que desempeñan los diferentes actores en relación con el aplicativo Bubblebox. Estos diagramas muestran las interacciones que cada usuario puede realizar dentro del sistema, permitiendo visualizar las funcionalidades disponibles para cada uno.

Ilustración 2 Diagrama caso de uso (usuario)

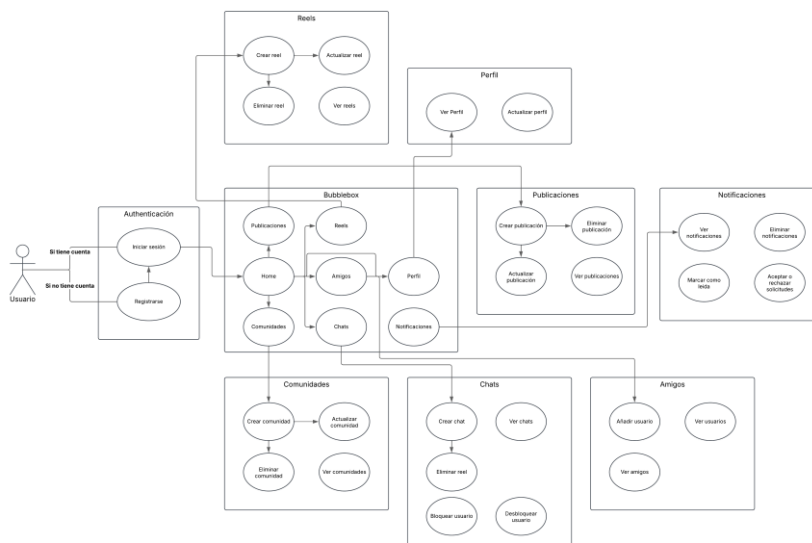


Ilustración 3 Diagrama casos de uso (Administrador)

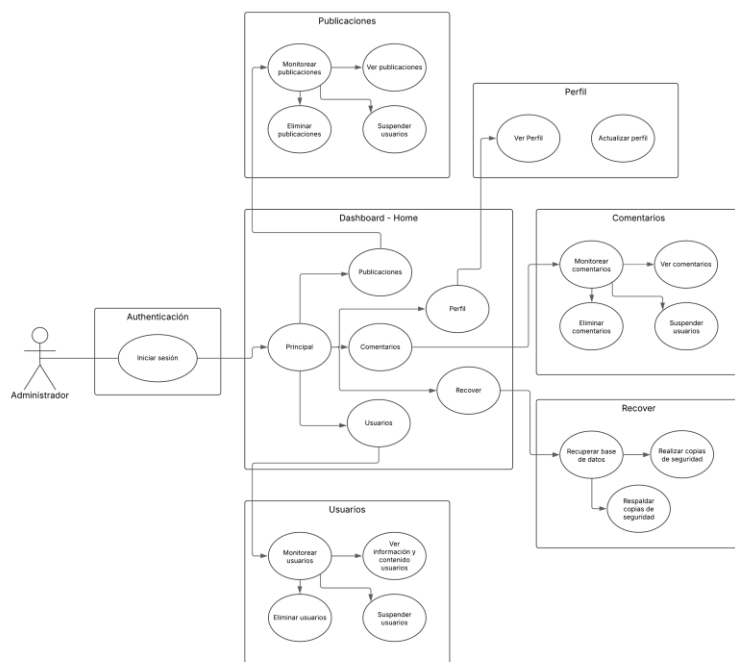
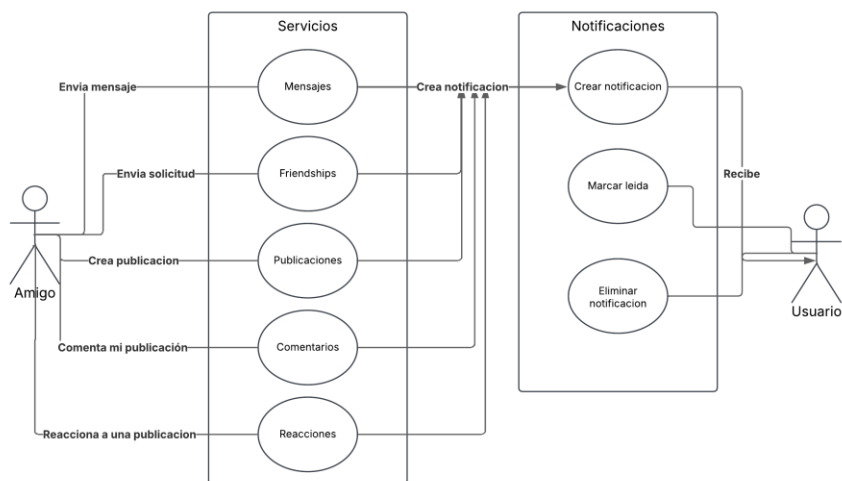


Ilustración 4 Diagrama casos de uso (notificaciones)



USUARIO: Es el actor principal del sistema. Puede registrarse, iniciar sesión, crear publicaciones, reaccionar a contenido, comentar, unirse a comunidades, enviar mensajes y visualizar publicaciones o historias de otros usuarios.

ADMINISTRADOR: Posee permisos avanzados dentro del sistema. Puede gestionar usuarios, moderar contenido, eliminar publicaciones inapropiadas y administrar las comunidades para mantener un entorno seguro dentro de la plataforma.

MODERADOR DE COMUNIDAD: Tiene la capacidad de gestionar publicaciones dentro de las comunidades, aceptar o rechazar miembros, y eliminar contenido que infrinja las normas establecidas en cada grupo.

SISTEMA DE NOTIFICACIONES: Genera alertas para los usuarios sobre nuevas interacciones, como solicitudes de amistad, reacciones a publicaciones, nuevos mensajes en grupos o menciones en comentarios.

DICCIONARIO DE DATOS

Para el almacenamiento de datos del software, se definen los campos necesarios para cada una de las entidades relacionadas con el aplicativo.

Tabla 1 Diccionario de datos modelo Usuarios

USUARIOS				
Nombre Campo	Tipo dato	PK/INDEX(FK)	Tamaño	Descripción
id	int	PK, AI		Identificador de usuario
nombre	varchar		250	Nombre real del usuario
username	varchar		250	Nombre ficticio que usara en el aplicativo
email	varchar	UNIQUE	250	Correo electrónico del usuario
contraseña	varchar		250	Contraseña que usara para iniciar sesión
avatar	varchar		250	Imagen de perfil del usuario
estado	enum		("conectado", "desconectado", "suspendido")	Identificador si esta conectado o desconectado en el aplicativo
lastSeen	timestamp			Ultima conexion
descripcion_usuario	text			Descripción libre que pondrá el usuario para expresarse
created_at	timestamp			Fecha de creación de la cuenta del usuario
secret2FA	varchar		255	Código que utiliza la autenticación de doble factor
token_recuperacion	varchar		255	Token de recuperación que se le envía al usuario a través de correo para realizar cambio de contraseña
token_expiracion	bigint			Token expiración que tiene el token de recuperación una vez este expire se eliminara el token de recuperación
rol	enum		("usuario", "administrador")	Rol del usuario dentro del sistema

Tabla 2 Diccionario de datos modelo publicaciones

PUBLICACIONES				
Nombre Campo	Tipo dato	PK/INDEX(FK)	Tamaño	Descripción
id	int	PK / AI		Identificador de publicación
titulo	varchar		255	Título de la publicación
contenido	text			Contenido que tendrá la publicación
id_usuario	int	INDEX (FK)		Llave foránea que relaciona la tabla usuarios
id_comunidad	int	INDEX (FK)		Llave foránea que relaciona la tabla comunidades
es_comunidad	tinyint		1	Booleano dependiendo si la publicación se hizo en una comunidad o no
imagen	varchar		255	Imagen que contiene la publicación
fecha_creacion	timestamp			Fecha de creación de la publicación

Tabla 3 Diccionario de datos modelo comunidades

COMUNIDADES				
Nombre Campo	Tipo dato	PK / INDEX(FK)	Tamaño	Descripción
id	int	PK / AI		Identificador de la comunidad
nombre	varchar		255	Nombre de la comunidad
descripcion	text			Descripción de la comunidad
id_creador	int	INDEX (FK)		Llave foránea que contiene el id del creador de la comunidad
imagen	varchar		255	Imagen de la comunidad
tipo_privacidad	enum		publica, privada, suspendido	Privacidad de comunidad
motivo	varchar		255	Motivo de la suspensión
duracion	int			Duración de la suspensión
fecha_fin_suspension	timestamp			Fecha de la suspensión
fecha_creacion	timestamp			Fecha de creación de la comunidad

Tabla 4 Diccionario de datos modelo usuarios_comunidades

USUARIOS COMUNIDADES				
Nombre Campo	Tipo dato	PK / INDEX (FK)	Tamaño	Descripción
id	int	PK / AI		Identificador de relación
id_usuario	int	INDEX (FK)		Llave foránea que relaciona la tabla usuarios
id_comunidad	int	INDEX (FK)		Llave foránea que relaciona la tabla comunidades
fecha_union	timestamp			Fecha de creación de la relación

Tabla 5 Diccionario de datos modelo reels

REELS				
Nombre Campo	Tipo dato	PK / INDEX(FK)	Tamaño	Descripción
id	int	PK / AI		Identificador del reel
usuario_id	int	INDEX (FK)		Llave foránea que relaciona la tabla usuarios
archivo_video	varchar		255	Video que contiene el reel
descripcion	text			Descripción del video
fecha_creacion	timestamp			Fecha de creación del reel

Tabla 6 Diccionario de datos modelo historias

HISTORIAS				
Nombre Campo	Tipo dato	PK / INDEX (FK)	Tamaño	Descripción
id	int	PK / AI		Identificador de historia
usuario_id	int	INDEX (FK)		Llave foránea que relaciona la tabla usuarios
contenido	text			Contenido que tendrá la historia
tipo	varchar		255	Tipo de contenido
fecha_creacion	timestamp			Fecha de creación de la historia
fecha_expiracion	timestamp			Fecha expiración de la historia

Tabla 7 Diccionario de datos modelo vistas_historias

VISTAS_HISTORIAS				
Nombre Campo	Tipo dato	PK / INDEX (FK)	Tamaño	Descripción
historia_id	int	INDEX (FK)		Llave foránea que relaciona la tabla historias
usuario_id	int	INDEX (FK)		Llave foránea que relaciona la tabla usuarios
fecha_vista	timestamp			Fecha de vista

Tabla 8 Diccionario de datos modelo reacciones

REACCIONES				
Nombre Campo	Tipo dato	PK / INDEX (FK)	Tamaño	Descripción
id	int	PK / AI		Identificador de reacción
tipo	varchar		50	Tipo de reacción
id_usuario	int	INDEX (FK)		Llave foránea que relaciona la tabla usuarios
id_contenido	int	INDEX (FK)		Llave foránea que relaciona contenido (publicación, etc)
tipo_contenido	enum		Publicación, reel, historia	tipo de contenido
fecha_creacion	timestamp			Fecha creación de la reacción

Tabla 9 Diccionario de datos modelo comentarios

COMENTARIOS				
Nombre Campo	Tipo dato	PK / INDEX(FK)	Tamaño	Descripción
id	int	PK / AI		Identificador de comentario
id_usuario	int	INDEX (FK)		Llave foránea que relaciona la tabla usuarios
id_contenido	int	FK		Llave foránea que relaciona contenido (publicación, etc)
tipo_contenido	enum		Publicación, reel, historia	tipo de contenido
contenido	text			Contenido del comentario
fecha_creacion	timestamp			Fecha creación del comentario

Tabla 10 Diccionario de datos modelo amistades

AMISTADES				
Nombre Campo	Tipo dato	PK / INDEX (FK)	Tamaño	Descripción
id	int	PK / AI		Identificador de amistades
id_usuario1	int	INDEX (FK)		Llave foránea que relaciona la tabla usuarios
id_usuario2	int	INDEX (FK)		Llave foránea que relaciona la tabla usuarios
estado	enum		Aceptada, rechazada, bloqueado, pendiente	Estado de la solicitud de amistad
fecha_creacion	timestamp			Fecha creación de la amistad
fecha_actualizacion	timestamp			Fecha actualización del estado de la amistad
estado_anterior	varchar		50	Estado anterior de la solicitud de amistad

Tabla 11 Diccionario de datos modelo mensajes

MESSAGES				
Nombre Campo	Tipo dato	PK / INDEX (FK)	Tamaño	Descripción
id	int	PK / AI		Identificador de mensaje
sender_id	int	INDEX (FK)		Llave foránea que relaciona la tabla usuarios
receiver_id	int	INDEX (FK)		Llave foránea que relaciona la tabla usuarios
message	text			Contenido del mensaje
audio_path	varchar		255	Ruta audio
duration	varchar		10	Duración del audio
read_status	tinyint		1	Estado de leído
created_at	timestamp			Fecha creación del mensaje

Tabla 12 Diccionario de datos modelo grupos

GRUPOS				
Nombre Campo	Tipo dato	PK / INDEX (FK)	Tamaño	Descripción
id	int	PK / AI		Identificador del grupo
id_creador	int	INDEX (FK)		Id del creador
name	varchar		255	Nombre del grupo
descripcion	varchar		255	Descripción del grupo
imagen	varchar		255	Imagen de perfil del grupo
created_at	timestamp			Fecha creación del grupo

Tabla 13 Diccionario de datos modelo miembros_grupos

MIEMBROS_GRUPOS				
Nombre Campo	Tipo dato	PK / INDEX (FK)	Tamaño	Descripción
id	int	PK / AI		Identificador de relación
group_id	int	INDEX (FK)		Llave foránea que relaciona la tabla grupos
user_id	int	INDEX (FK)		Llave foránea que relaciona la tabla usuarios
created_at	timestamp			Fecha creación de la relación

Tabla 14 Diccionario de datos modelo mensajes_grupos

MENSAJES_GRUPOS				
Nombre Campo	Tipo dato	PK / INDEX (FK)	Tamaño	Descripción
id	int	PK / AI		Identificador de mensaje
group_id	int	INDEX (FK)		Llave foránea que relaciona la tabla grupo
sender_id	int	INDEX (FK)		Llave foránea que relaciona la tabla usuarios
message	text			Contenido del mensaje
created_at	timestamp			Fecha creación del mensaje

Tabla 15 Diccionario de datos modelo notificaciones

NOTIFICACIONES				
Nombre Campo	Tipo dato	PK / INDEX (FK)	Tamaño	Descripción
id	int	PK / AI		Identificador de la notificacion
usuario_id	int	INDEX (FK)		Llave foránea que relaciona la tabla usuarios
tipo	varchar		50	Tipo de notificacion
contenido	text			Contenido de la notificacion
leida	tinyint		1	Estado de la notificación si esta leida o no
referencia_id	int	INDEX (FK)		Llave foránea que relaciona con la tabla amistades
fecha_creacion	timestamp			Fecha creación del grupo

Tabla 16 Diccionario de datos modelo respuestas_comentarios_publicaciones

RESPUESTAS_COMENTARIOS_PUBLICACIONES				
Nombre Campo	Tipo dato	PK / INDEX (FK)	Tamaño	Descripción
id	int	PK / AI		Identificador de la respuesta
id_usuario	int	INDEX (FK)		Llave foránea que relaciona la tabla usuarios
id_comentario	int	INDEX (FK)		Llave foránea que relaciona la tabla comentarios
contenido	text			Contenido de la respuesta
fecha_creacion	timestamp			Fecha creación del grupo

Tabla 17 Diccionario de datos modelo respuestas_comentarios_reels

RESPUESTAS_COMENTARIOS_REELS				
Nombre Campo	Tipo dato	PK / INDEX (FK)	Tamaño	Descripción
id	int	PK / AI		Identificador de la respuesta
id_usuario	int	INDEX (FK)		Llave foránea que relaciona la tabla usuarios
id_comentario	int	INDEX (FK)		Llave foránea que relaciona la tabla comentarios
contenido	text			Contenido de la respuesta
fecha_creacion	timestamp			Fecha creación del grupo

Tabla 18 Diccionario de datos modelo backups

BACKUPS				
Nombre Campo	Tipo dato	PK / INDEX (FK)	Tamaño	Descripción
id	int	PK / AI		Identificador del backup
filename	varchar		255	Nombre del archivo
created_at	datetime			Fecha de creación
created_by	int			Id usuario quien lo creo

Tabla 19 Diccionario de datos modelo backups_restores

BACKUPS_RESTORES				
Nombre Campo	Tipo dato	PK / INDEX (FK)	Tamaño	Descripción
id	int	PK / AI		Identificador de la restauración
backup_filename	varchar		255	Nombre del archivo
restored_at	datetime			Fecha de restauración
restored_by	int			Id usuario quien lo restauro

Tabla 20 Diccionario de datos modelo reportes

REPORTES				
Nombre Campo	Tipo dato	PK / FK	Tamaño	Descripción
id	int	PK / AI		Identificador de reporte
tipo_reporte	<u>enum</u>		publicacion, reel, comentario, comunidad, usuario, contenido	Tipo de reporte
id_contenido	int	INDEX (FK)		Llave foránea que relaciona las tablas de contenido
id_usuario_reportante	int	INDEX (FK)		Llave foránea que relaciona la tabla usuarios
motivo	varchar		255	Motivo del reporte
descripcion	text			Descripción del reporte
estado	enum		pendiente, revisado, resuelto, rechazado	Estado del reporte
fecha_reporte	timestamp			Fecha de creación del reporte
fecha_resolucion	timestamp			Fecha de solución
id_admin_resolucion	int	INDEX (FK)		Llave foránea que relaciona la tabla usuarios
accion_tomada	varchar		255	Acción tomada contra la persona infractora

Tabla 21 Diccionario de datos modelo intereses

INTERESES				
Nombre Campo	Tipo dato	PK / INDEX (FK)	Tamaño	Descripción
id	int	PK / AI		Identificador del interes
user_id	int	INDEX (FK)		Llave foránea que relaciona la tabla usuarios
interes	varchar		255	Descripción del interes
created_at	timestamp			Fecha de creación

Tabla 22 Diccionario de datos modelo configuraciones_usuario

CONFIGURACIONES_USUARIO				
Nombre Campo	Tipo dato	PK / FK	Tamaño	Descripción
id	int	PK / AI		Identificador de las configuraciones
usuario_id	int	INDEX (FK)		Llave foránea que relaciona la tabla usuarios
privacidad	varchar		20	Descripcion del tipo de privacidad del perfil
notificaciones	tinyint		1	Booleando para ver si las notificaciones están activas o no
audio_enable	tinyint		1	Booleano para ver si el usuario tiene permiso en la app de grabar audio
files_access_enable	tinyint		1	Booleano para ver si el usuario tiene permisos a la app para acceder a sus archivos
online_visibility	tinyint		1	Booleano para ver si el usuario tiene permitido que otros usuarios lo vean en línea o no
idioma	varchar		5	Idioma de la app para el usuario
autoplay_videos	tinyint		1	Si el usuario tiene permitido que los videos se reproduzcan automáticamente
fecha_creacion	timestamp			Fecha de creación
fecha_actualización	timestamp			Fecha actualización

BIBLIOGRAFIA

Creadores. (s.f.). *Casos de uso*. Obtenido de

https://drive.google.com/file/d/10b6DKxRV12o4wqP_lQp_k--eQbdJtdS7/view?usp=drive_link.

Creadores. (s.f.). *Diagrama DB*. Obtenido de

https://drive.google.com/file/d/1RXXntxyeuKzXWeMktxHqJVwQXaYUubver/view?usp=drive_link.

Creadores. (s.f.). *Diagrama de componentes*. Obtenido de

https://drive.google.com/file/d/1S52UBY8Yr2FuuKzIEAbPH5m1NBQ8jvEi/view?usp=drive_link.

Creadores. (s.f.). *Diagrama interfaces*. Obtenido de

https://drive.google.com/file/d/1YQTrBrEzbtQW-W0a0PR1XiAPsPe8McE4/view?usp=drive_link.

Creadores. (s.f.). *Diagrama microservicios*. Obtenido de

https://drive.google.com/file/d/1QnU2LZs0YijjRaTCIVl4DqN0lqykDwMt/view?usp=drive_link.

Creadores. (s.f.). *Diagrama secuencia*. Obtenido de https://drive.google.com/file/d/1-Oq0EDLIFJ9ZGV083HGh-mNAjyNsdlZt/view?usp=drive_link.

Creadores. (s.f.). *Script esquema DB*. Obtenido de

https://drive.google.com/file/d/12X7PM0FIWDqEgRpScavU5rB7o-r5xcX_/view?usp=drive_link.

documentation, G. (s.f.). *GitHub*. Obtenido de <https://docs.github.com>.

Foundation, O. (s.f.). *Node.js*. Obtenido de <https://nodejs.org/es>.

Microsoft. (s.f.). *Visual Studio Code*. Obtenido de <https://code.visualstudio.com>.

