

Entrega Final: Documentación

Documentación Final en la página 11



ITESO, Universidad
Jesuita de Guadalajara

- **Fecha de entrega** Martes a las 23:59
- **Puntos** 10
- **Entregando** un cuadro de entrada de texto o una carga de archivo
- **Tipos de archivo** docx y pdf
- **Disponible** hasta el 28 de feb en 23:59
- **Github:** https://github.com/EIChicoYetix/API_BetsNFriends/tree/entrega_4

Objetivo: Crear la documentación inicial del proyecto integrador.

Descripción:

Generar un documento en Word o PDF para describir el proyecto integrador, para explicar en qué consiste el proyecto, cuál es el problema que se desea resolver, cómo se desea resolver y con qué herramientas.

Título

Bets n' friends

Descripción

La idea principal del proyecto consiste en realizar una red social donde puedas chatear con tus amigos y, del mismo modo, realizar apuestas de manera “friendly” con los partidos más importantes de la semana. Se tendrá un score de cuanto quieres apostar y el otro usuario indicará si quiere aumentar la apuesta o igualarla, así hasta que ambos lleguen a un acuerdo. Se planea implementar mecánicas o estadísticas para fomentar la interacción entre los usuarios.

Integrantes

- Andre Yahir Gonzalez Cuevas
- Carlos Emiliano Rodríguez Núñez

Stack de Tecnologías a utilizar (librerías, frameworks, base de datos, integración con servicios de 3ros, cloud, etc.)

Librerías bases para realizar una API cómo npm, node, nodemon, docker, git y github para el manejo de versiones y avances del proyecto, enfocado en git kraken , mongoose, express, el software de postman, los servicios de AWS, passport

El manejo de base de datos con los usuarios se realizará a través de MongoDB.

Listar y describir los roles de usuarios

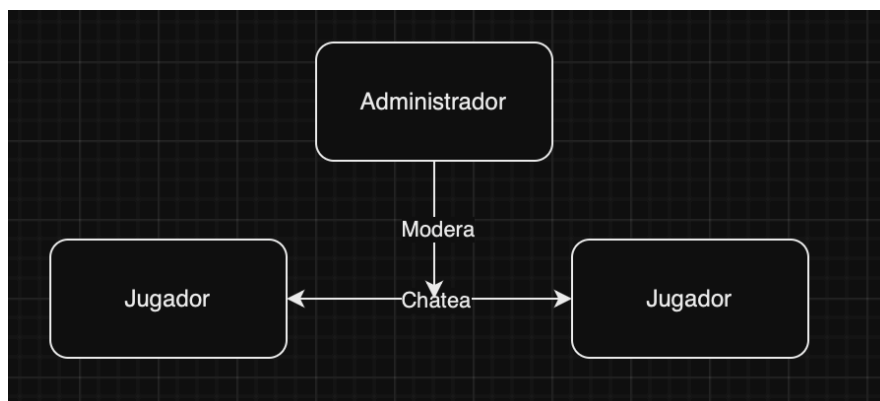
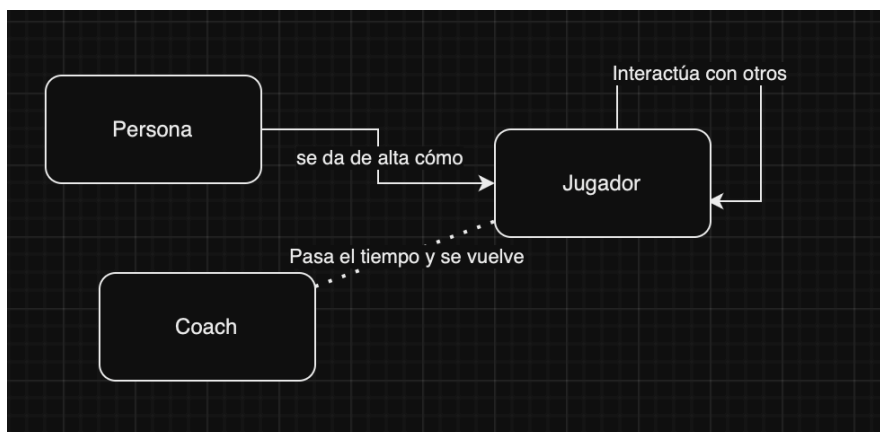
Planeamos que los roles sean en base al acceso que se puede tener, serían los siguientes:

- Administrador: Tiene acceso a todo el proyecto y puede agregar partidos, acceder al código y ver todos los usuarios que existen, así como tomar acciones ante las cuentas registradas en caso de que vea comportamiento o acciones que infrinjan las normas de la página.
- Jugadores: Usuarios que pueden interactuar con otros “jugadores”, conversar con ellos, apostar, ver partidos y enviar nuevas solicitudes de amistad.
- Coach: Jugadores que tiene cierto tiempo en la plataforma puede solicitar ser coach y poder agregar partidos

Tareas principales o flujos principales de cada usuario

- Administrador: Mantener una sana convivencia entre los usuarios y prevenir que se utilice para otros propósitos.
- Jugadores: Apostar, tener más amigos y divertirse.
- Coach: Dar el ejemplo de un buen apostador y agregar partidos de la semana.

Diagramas: Incluir por lo menos 3 diagramas de secuencia o flujo que describan algunos de los procesos de la aplicación.



Describir las entidades que se van a administrar dentro de la aplicación y las acciones que se realizarán sobre cada una de estas. Ejemplo: Usuario: Registro como cliente, iniciar sesión, ver y editar perfil, etc. etc.

- Usuario:

Registro: Permitirá a los usuarios crear una cuenta en la plataforma proporcionando información básica como nombre, correo electrónico y contraseña.

Iniciar sesión: Permite a los usuarios acceder a sus cuentas utilizando sus credenciales.

Ver perfil: Los usuarios pueden ver su información personal, incluidos detalles como nombre, imagen de perfil, amigos, etc.

Editar perfil: Los usuarios pueden actualizar su información personal, cambiar su foto de perfil, actualizar la contraseña, etc.

Agregar amigos: Permite a los usuarios enviar solicitudes de amistad y aceptar solicitudes de otros usuarios para conectarse y chatear.

Chat: Los usuarios pueden comunicarse con sus amigos a través de un sistema de mensajería instantánea integrado en la plataforma.

- Partido:

Agregar partidos: Los administradores y los coaches pueden agregar nuevos partidos a la plataforma, proporcionando detalles como equipos involucrados, fecha, hora, etc.

Ver partido: Los usuarios pueden ver información sobre los partidos disponibles, incluidos detalles como equipos, fecha, hora y apuestas realizadas.

Apostar: Los jugadores pueden realizar apuestas en los partidos disponibles, especificando la cantidad que desean apostar y seleccionando su opción de apuesta.

Seguir partido: Los usuarios pueden seguir en tiempo real los resultados de los partidos en los que han apostado y recibir actualizaciones sobre el progreso del juego.

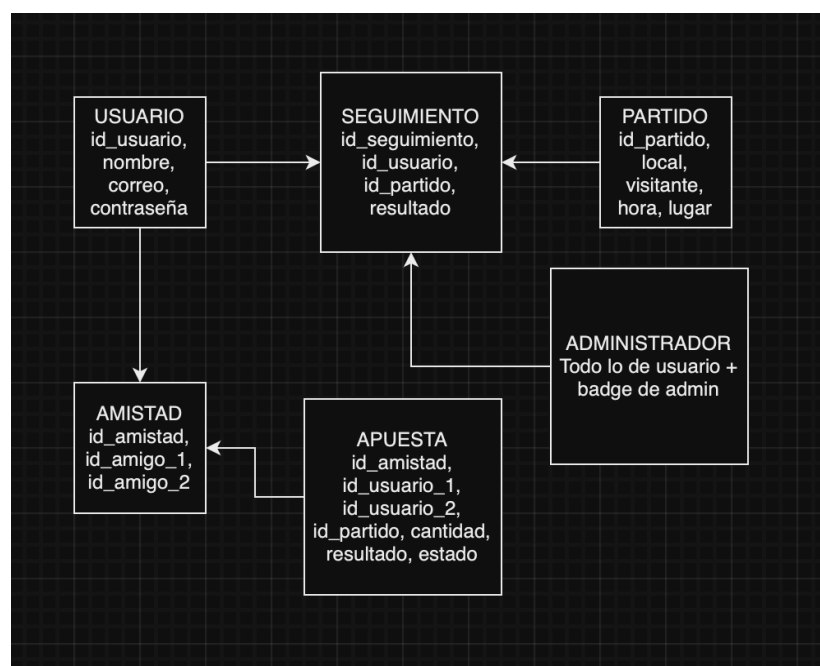
- Administrador:

Monitorear actividad: Los administradores pueden monitorear la actividad de los usuarios, revisar las apuestas realizadas, verificar el comportamiento de los usuarios, entre otras.

Gestionar usuarios: Los administradores tienen la capacidad de ver, editar o eliminar cuentas de usuarios según sea necesario.

Gestionar partidos: Los administradores pueden agregar, editar o eliminar partidos según la programación y relevancia de los eventos deportivos.

Generar un Diagrama de Entidad-Relación de acuerdo al punto anterior



Alcance (temas a cubrir de acuerdo a la guía de aprendizaje) y breve descripción de éste:

- Se planea usar conocimientos previos sobre html y css para el front end y las páginas que se requieran crear.
- Para el testing se planea usar Git como base para el CI/CD, donde se pueda hacer un seguimiento de las versiones del proyecto.
- Para la verificación con google se planea usar passport desde visual studio code donde se verifique el correo y contraseña.
- Para completar el perfil de los usuarios se requerirá subir una foto, la cual funcione como foto de usuario.
- Se requerirá una plataforma para poder chatear en tiempo real con los usuarios así como la dada de alta o baja para las apuestas en vivo.

Objetivo: Desarrollar una aplicación web o de escritorio utilizando herramientas y frameworks modernos de JavaScript			
Aprendizajes Esperados	Actividades en clase	Actividades independiente (tareas)	Productos e indicadores de evaluación
MVC y Estructuras de código enfocadas al backend	Profesor: expondrá los conceptos, tecnologías, la sintaxis básica y ejemplos. Grupal: Realizar ejemplos Independiente: Se dejan pequeños retos para probar que se comprende el tema y prácticas guiadas	El alumno debe aplicar sus conocimientos de HTML y CSS para dar estructura y estilo a un documento de acuerdo a las especificaciones solicitadas por el profesor.	Tareas entregadas cumpliendo los requerimientos solicitados por el profesor.
Frameworks para el Backend		Resolver problemas en plataformas como HackerRank	Se realizarán preguntas en clase y prácticas que muestren conocimiento del tema visto.
JavaScript Avanzado		Resolver prácticas guiados	El proyecto debe aplicar estos conocimientos.
Bundler como webpack junto con herramientas como TypeScript, handlebars			
Herramientas de backend: axios, Passport, multer, entre otras		Realizar ejercicios y prácticas guiadas que resuelvan un problema acotado	

Aprendizajes Esperados	Actividades en clase	Actividades independiente	Productos e indicadores de evaluación
Paradigma Orientado a Objeto y Patrones de diseño: Modelo Vista Controlador	Profesor: expondrá los conceptos, la estructura de carpetas y como se aterrizan en una aplicación. Grupal: Realizar ejemplos y prácticas Independiente: Se dejan pequeños retos para probar que se comprende el tema	Tarea y práctica relacionadas.	Tareas entregadas cumpliendo los requerimientos solicitados por el profesor.
Testing e integración continua	Profesor: explica los conceptos, el lenguaje para testing y CI/CD Grupal: Ejemplos y prácticas	Resolver práctica guiada	Se realizarán preguntas en clase y/o prácticas que muestren conocimiento del tema visto.
Plataformas para despliegue de aplicaciones	Profesor: explica los conceptos y como se aplica y desarrolla Grupal: Ejemplos y prácticas	Ejercicios y prácticas donde implica el despliegue de una aplicación del backend	El proyecto debe aplicar estos conocimientos.

Entrega 3: Passport, Archivos y Pruebas:

Esta entrega consiste en la creación de los endpoints, middlewares, controladores y modelos necesarios para incorporar:

- Autenticación mediante la cuenta de Google (Passport)
- Carga de archivos en S3
- Pruebas automatizadas

Carga de archivos en S3. El archivo /middlewares/upload-s3.ts contiene lo requerido para poder realizar un post con el botón que se encuentra al correr localhost 3000 pero hay un error en a la hora de correr npm run dev.

Amazon S3

Account snapshot

All AWS Regions

View Storage Lens dashboard

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

General purpose buckets

Directory buckets

General purpose buckets (2)

Info

All AWS Regions

Refresh

Copy ARN

Empty

Delete

Create bucket

Buckets are containers for data stored in S3.

Find buckets by name

< 1 > ⚙

	Name	AWS Region	IAM Access Analyzer	Creation date
<input type="radio"/>	bucketemmet	US East (Ohio) us-east-2	View analyzer for us-east-2	April 19, 2024, 22:34:13 (UTC-06:00)
<input type="radio"/>	iteso-p2024	US East (Ohio) us-east-2	View analyzer for us-east-2	April 19, 2024, 21:10:14 (UTC-06:00)

IAM > Users

Users (3)

Info

Refresh

Delete

Create user

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

Search

< 1 > ⚙

<input type="checkbox"/>	User name	Path	Group	Last activity	MFA	Password age	Conso
<input type="checkbox"/>	bucketemmet	/	1	-	-	-	-
<input type="checkbox"/>	iteso-p2024	/	0	-	-	-	-
<input type="checkbox"/>	superadmin	/	1	-	-	-	-

Parece que no se conecta de manera correcta al bucket, aunque lo intentamos de mil maneras.

```
MONGODB_URI=mongodb+srv://emi_owner:proyecto1desarrollo@proyecto1.shytxv.mongodb.net/
GOOGLE_ID=316466451928-ccb2vcjri9v08714krklmj641renqamd.apps.googleusercontent.com
GOOGLE_SECRET=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
GOOGLE_CALLBACK_URL=http://localhost:3000/google/callback
SECRET_KEY=helloworld
S3_ACCESS_KEY=AKIATCKAQFBIA3QBCRBL
S3_SECRET_KEY=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
S3_REGION=us-east2
S3_BUCKET_NAME=bucketemmet
```

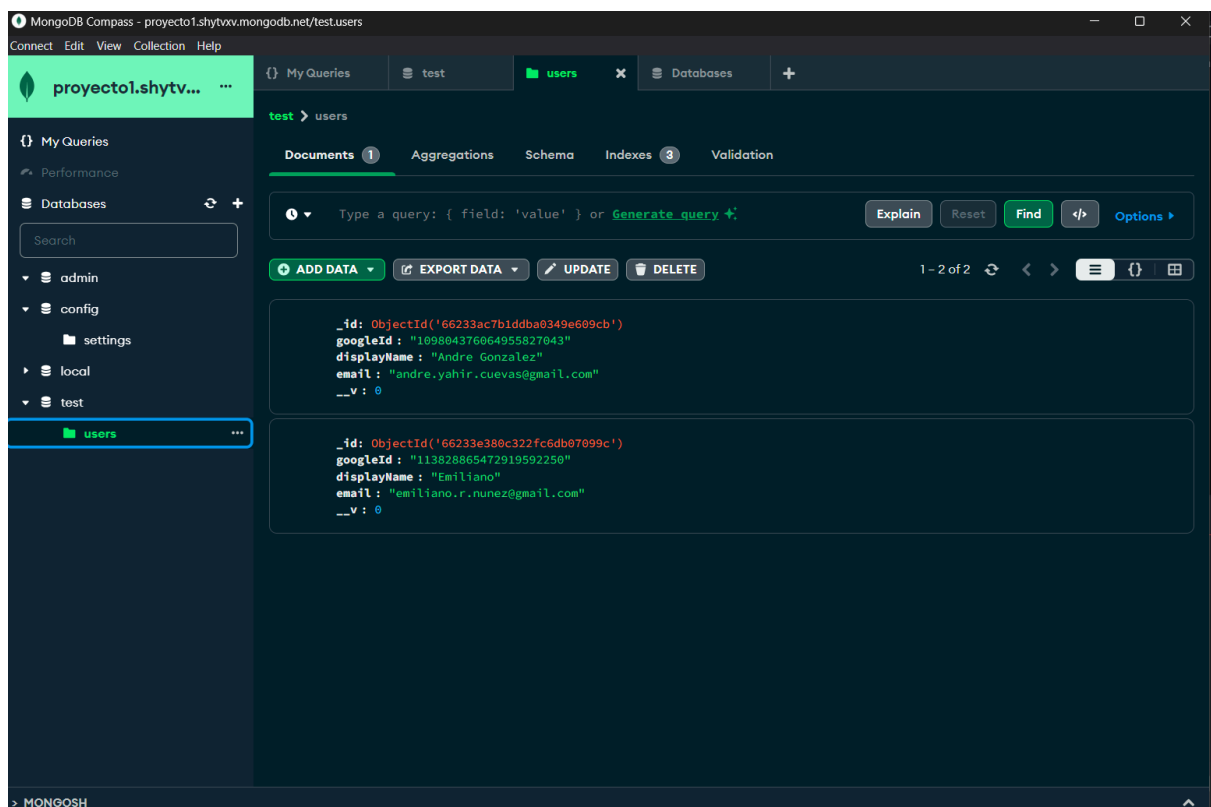
Inicio de sesión con google.

Iniciar Sesión

Iniciar Sesión

Iniciar Sesión con Google

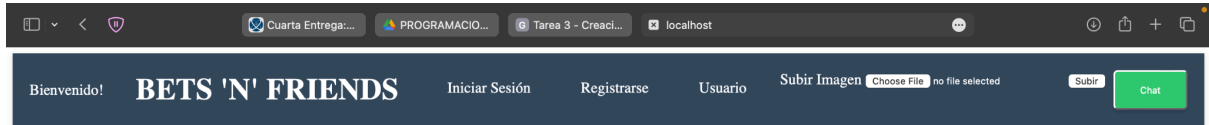
El archivo `/middlewares/google-auth.ts` permite dar de alta al usuario desde la cuenta de google a la hora de iniciar sesión. Vimos varios videos de youtube y solo nos aparecen los usuarios en esta interfaz, seguiremos intentando para que se conecte bien a la carpeta `/test` y se muestren correctamente en MongoDB.



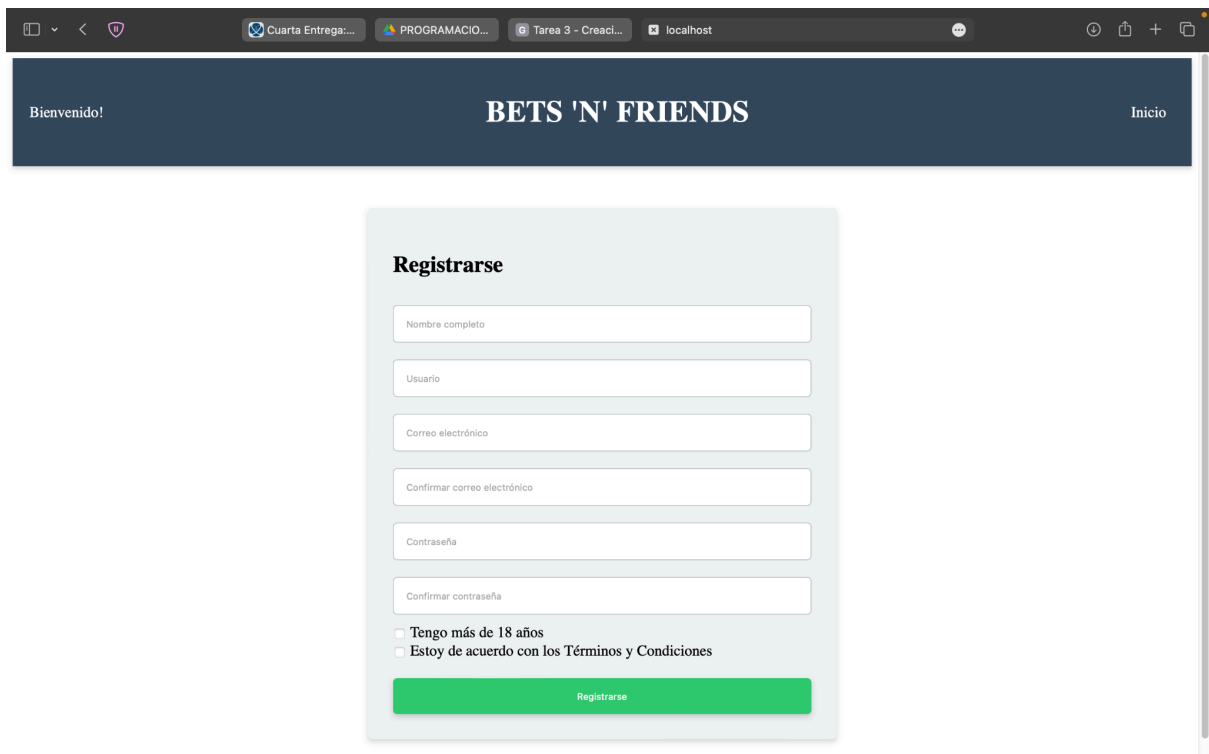
Entrega 4: Socket.io

Implementar comunicación en tiempo real mediante Sockets.

Después de abrir localhost:3000/inicio



Puedes elegir o Registrarte



O iniciar sesión directamente, ya sea con tu cuenta o con google.

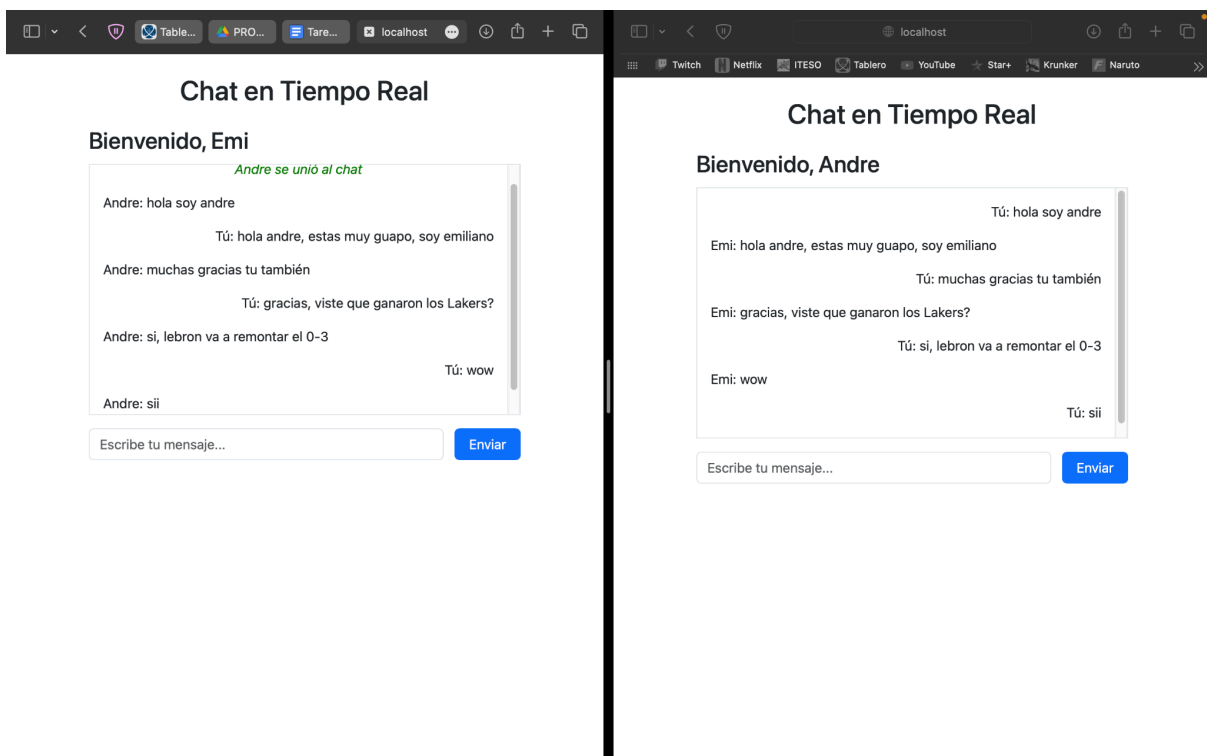


Iniciar Sesión

Iniciar Sesión

Iniciar Sesión con Google

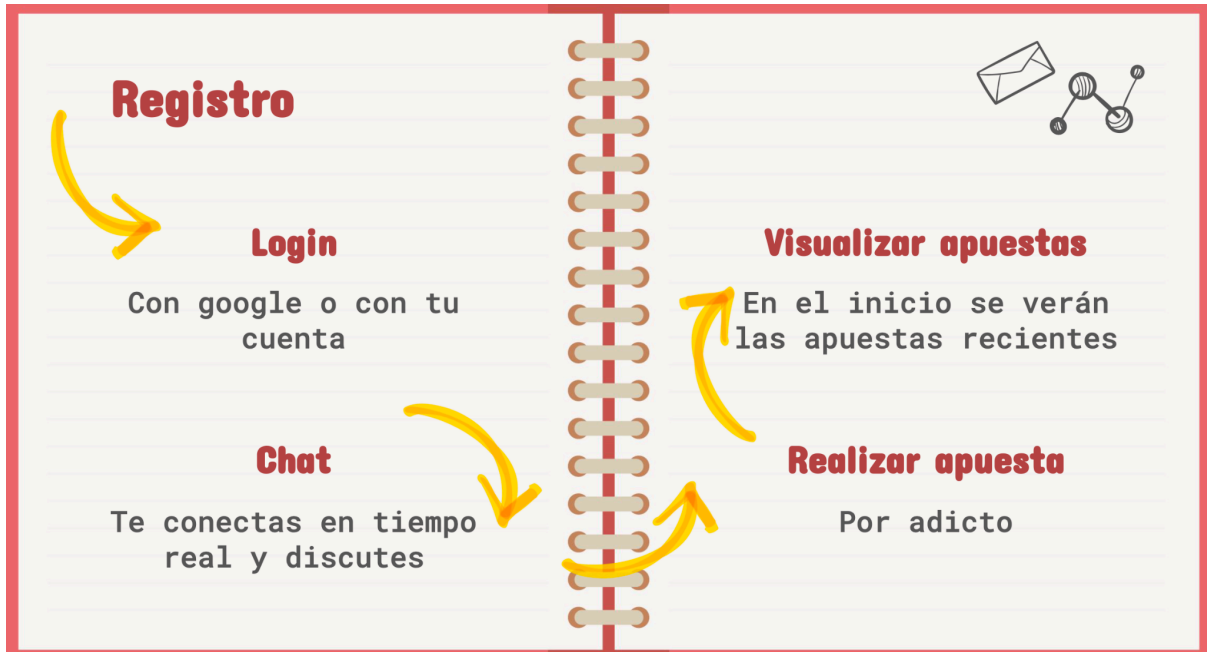
Seguido de esto, seleccionas el botón de chat y puedes tener una conversación en tiempo real.



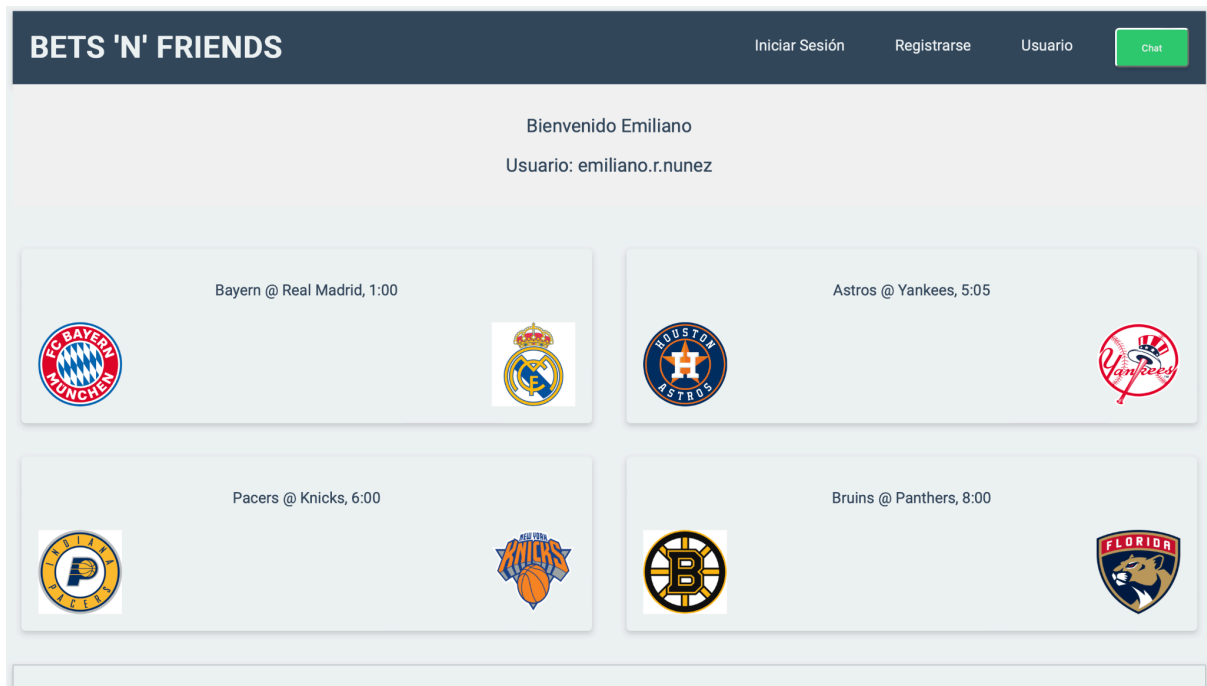
```
Conexión a la base de datos exitosa
[nodemon] restarting due to changes...
[nodemon] starting `ts-node src/index.ts`
El servidor está corriendo en el puerto 3000
Conexión a la base de datos exitosa
A new user connected
A new user connected
```

Versión Final:

La versión final de nuestro proyecto nos permite seguir el siguiente flujo de trabajo.



Esto se puede realizar desde nuestra página web donde puedes presionar el botón de registrarse para crear una cuenta desde cero. Después de esto, puedes realizar el inicio de sesión con tu cuenta o con google. Para finalizar, te vas a la sección de chat donde puedes platicar con tus amigos y realizar apuestas de manera amigable en tiempo real.



En la parte de chat se pueden realizar apuestas que se guardan en un archivo.txt el cuál tiene conexión con mongodb para tener registro de quién, cuándo y a qué equipo se le apostó.

Estas apuestas se pueden ver en el inicio de la página, junto con la de los demás usuarios, al guardarse las apuestas, la página se actualiza conforme a nuevas apuestas salgan y son separadas por el ID del partido al que pertenezcan.

