

CLUB VOLEIBOL LYNX: SEGUNDA ENTREGA

Betancur Izquierdo Simón, Enciso Cabana Natalia,
Montaña Rodríguez David Alejandro, García Ramírez Luisa María
e-mail: natalia.cabana@urosario.edu.co,
simon.izquierdo@urosario.edu.co,
david.montana@urosario.edu.co,
luisam.garcia@urosario.edu.co

RESUMEN

Este proyecto se basa en un club de voleibol “Lynx” que enfrenta problemas de desorganización y falta de registros, lo que afecta la gestión de jugadores, entrenadores y horarios. Para solucionar esta situación se implementará un sistema de base de datos que optimice la administración del club permitiendo así, un manejo eficiente de la información personal, las mensualidades, horarios de entrenamiento y torneos. Adicionalmente, gracias a una metodología Kanban y modelos entidad-relación, este proyecto mejorará la estructura organizativa del club facilitando su crecimiento y operatividad.

1 FASE DE ANÁLISIS NECESIDAD:

Crear una base de datos que le permita al club de voleibol “Lynx” un manejo más eficiente y apropiado de la información que sostiene como club deportivo.

CONTEXTUALIZACIÓN:

El club de voleibol “Lynx”, fundado en el año 2020, ha venido enfrentando retos organizativos y administrativos desde un reciente cambio en su equipo directivo. En la actualidad, los procesos de recolección y manejo de información se llevan a cabo de forma rudimentaria: se envían archivos de Excel a los acudientes para que ellos mismos los llenen con los datos de los deportistas, como nombres, edades, EPS, entre otros. Esta información luego se centraliza de manera manual y muchas veces queda incompleta y desactualizada, esto por esto por lo que el club no ha podido expandirse.

PROBLEMÁTICA:

¿Cómo puede el club de voleibol “Lynx” organizar mejor su información y facilitar la gestión de jugadores, entrenadores, horarios y torneos, a través de un sistema de bases de datos que le permita tener de forma clara?

2 FASE DE PLANEACIÓN:

Objetivo General:

Crear una base de datos relacional (SQL) y una base de datos no relacional (NoSQL) para el club de voleibol “Lynx”, para que se pueda mejorar la organización y acceso a la información de deportistas, entrenadores, horarios, mensualidades y torneos.

Objetivos Específicos:

1. Analizar las necesidades del club de voleibol “Lynx” y definir el problema, los objetivos, las posibles soluciones a hacer, esto es necesario para estructurar la información actual.
2. Diseñar todo el sistema de datos que se utilizara, incluyendo los modelos como el MER y el modelo relacional, los requisitos del sistema y los diagramas que nos ayuden a estructurar bien la información.
3. Implementar una base de datos en SQL y otra en NoSQL, asegurando el correcto funcionamiento por medio de pruebas que confirmen la información insertada se guarde, consulte y se organice bien.

Alcance:

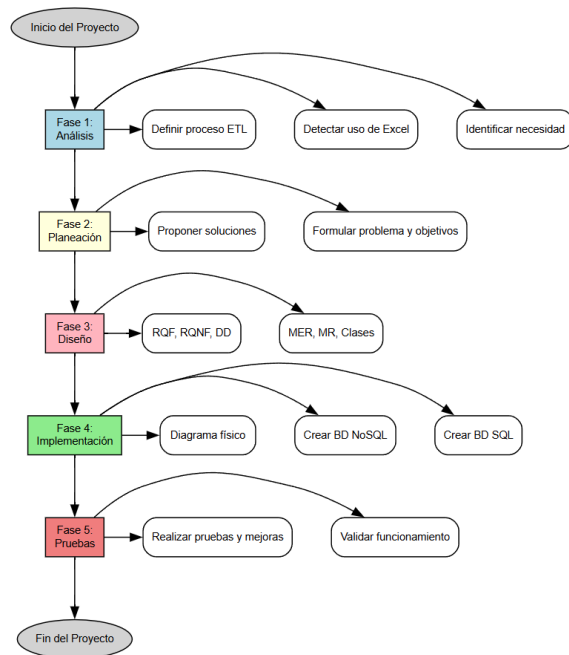
Con este proyecto se logrará suplir la necesidad del club de voleibol “Lynx” mediante las siguientes implementaciones en su base de datos. Este se llevará a cabo en un lapso de 4 meses, desde el inicio del semestre académico y concluirá al final de este, con la participación de 4 estudiantes que estarán en las tareas

de análisis, diseño, implementación y pruebas.

1. Información entrenadores y deportistas: Este es un registro de datos personales, EPS, categoría, horarios y la posición de los deportistas.
2. Mensualidades de los jugadores:
3. Partidos y fechas importantes para el club: Registro histórico de los partidos jugados y que se jugaran, con sus respectivos resultados.
4. Equipo (posiciones y disponibilidad): Posiciones de cada jugador y como es esta asignada dependiendo de la disponibilidad que este.

Todo esto se realizará mediante bases de datos en MySQL WorkBench, donde se hará la creación de bases de datos, con sus respectivos campos, relaciones, etc.

Diagrama Solución de Flujo:



Generado en: <https://dreampuf.github.io/GraphvizOnline/>

Metodología KanBan:

La metodología que se aplicó fue KanBan, con esto primero creamos un Trello, donde se definirían las tareas que se harían para llevar a cabo el proyecto de bases de datos.



Imagen 1 sacada de: <https://trello.com/h/nqCocWYy/daily> & <https://trello.com/h/ACtTYEih/backlog-general> Tablero Kanban

Para este caso, hemos hecho reuniones, no han sido seguidas, pero han sido concisas para lograr el proyecto, con esta metodología hemos podido organizarnos para proponernos una meta para la reunión y cumplirla.

Levantamiento de Información:

Principalmente para el levantamiento de información fue mediante una entrevista, cada integrante conoció al dueño del club y se le hicieron preguntas sobre el club, como:

- ¿Cómo maneja los datos del club?
- ¿Están organizados los datos del club?
- ¿Desde cuándo se presentó el desorden de los datos?

Luego de esto, el cliente nos explicó que por un cambio de administración todos los datos entraron en un desorden completo. Es por esto, por lo que decidimos crear una base de datos para que Lynx tenga un orden a la hora de registrar nuevos deportistas que ingresen al club.

A partir de esto, identificamos los Stakeholders, y diferenciamos quienes eran los que tenían mayor influencia y menor influencia, e identificamos Stakeholders como:

- Dueño del club.
- Entrenadores.
- Deportistas.
- Acudientes.

Con esto se concluyó que el club no contaba con un sistema eficiente para las bases de datos que ellos manejan, lo que dificultaba la gestión, el seguimiento de mensualidad, la organización de partidos y disponibilidad de jugadores.

3 STAKE HOLDERS

Dentro del proyecto se definen las entidades externas e internas del club, quienes influyen en la importancia del proyecto llamados StakeHolders:

Externos:

- Acudientes
- Enfermera

Internos:

- Dueño (Administrador)
- Deportistas
- Entrenadores

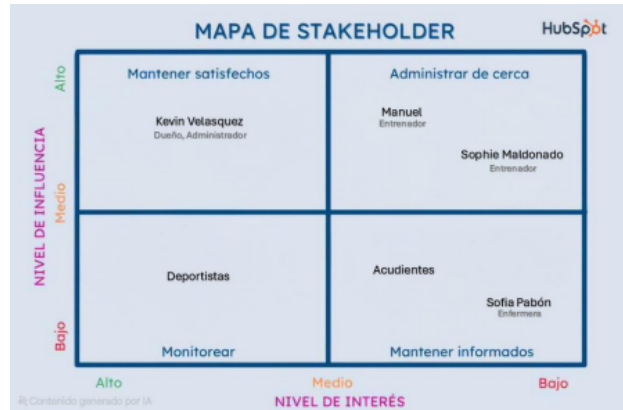


Imagen 2 StakeHolders

4 DISEÑO

En este apartado nos adentramos a la fase del diseño, donde podremos generar tanto requisitos funcionales, como no funcionales, también podremos generar el diagrama de clases, los modelos MER y MR.

A continuación, se encuentran los requisitos funcionales y cada uno de estos acompañados de su historia de usuario.

Cada historia de usuario tendrá un costo de **1 punto** de esfuerzo, esto equivale a **una hora de trabajo para el RQE**.

Requisitos Funcionales (RQF):

CÓDIGO	REQUISITO FUNCIONAL
RQF001	Nombre: Buscar nombre de un deportista
	Descripción: El sistema permitirá al deportista hacer una búsqueda de su nombre y así, obtener la información que necesite o desee solicitar en ese preciso momento. Además, tanto el administrador como el entrenador también podrán buscar a algún deportista por su nombre en el sistema.
	Usuarios: Deportista, Administrador y Entrenador.

HU001:

Nombre: Buscar nombre de un deportista

Historia:

Yo como deportista quiero tener acceso a mi propio nombre dentro de la base de datos diseñada para el club, con el fin de buscar alguna información básica acerca de mí.

Yo como administrador quiero tener acceso a los nombres

de los deportistas que hacen parte de mi club con el fin de obtener información generalizada de ellos.

Yo como entrenador quiero tener acceso al nombre de los deportistas que pertenecen a las clases que dictó dentro del club con el fin de tener información acerca de ellos y conocerlos.

Criterios de Aceptación:

- Registrar el nombre del deportista
- Consultar nombre del deportista

Puntos de esfuerzo: 1*

CÓDIGO	REQUISITO FUNCIONAL
RQF002	Nombre: Buscar identificación de un deportista Descripción: El sistema permitirá al deportista hacer una búsqueda de su identificación y así, obtener información más precisa acerca de todo su rendimiento y lo que la base de datos le permita consultar con base en su id. Además, tanto el administrador como el entrenador también podrán buscar a algún deportista y su información por su id en el sistema. Usuarios: Deportista, Administrador y Entrenador.

HU002:

Nombre: Buscar identificación de un deportista

Historia:

Yo como deportista quiero tener acceso a mi propia identificación (id) dentro de la base de datos diseñada para el club, con el fin de buscar información detallada acerca de mí,

como mi categoría, mi eps, mis entrenadores, etc.

Yo como administrador quiero tener acceso a las identificaciones de los deportistas que hacen parte de mi club con el fin de obtener información detallada acerca de ellos.

Yo como entrenador quiero tener acceso a la identificación de los deportistas que pertenecen al club con el fin de tener información detallada acerca de ellos, como por ejemplo su edad para saber si hacen parte de la categoría que tengo encargada, etc.

Criterios de Aceptación:

- Registrar id del deportista
- Consultar id del deportista

Puntos de esfuerzo: 1*

CÓDIGO	REQUISITO FUNCIONAL
RQF003	Nombre: Buscar nombre de un entrenador Descripción: El sistema permitirá al entrenador hacer una búsqueda de su nombre y obtener información suya dentro del club, como sus horarios, a que categoría entrena, etc. Además, tanto el administrador como los deportistas podrán buscar el nombre de algún entrenador en el sistema para obtener alguna información sobre él. Usuarios: Entrenador, Administrador y Deportistas.

HU003:

Nombre: Buscar nombre de un entrenador

Historia:

Yo como entrenador quiero tener acceso a mi propio

nombre dentro de la base de datos diseñada para el club, con el fin de buscar información básica acerca de mí.

Yo como administrador quiero tener acceso a los nombres de los entrenadores que hacen parte de mi club con el fin de obtener información generalizada acerca de ellos y conocerlos.

Yo como deportista quiero tener acceso al nombre del entrenador que me será asignado según mi categoría.

Criterios de Aceptación:

- Registrar nombre del entrenador
- Consultar nombre del entrenador

Puntos de esfuerzo: 1*

CÓDIGO	REQUISITO FUNCIONAL
RQF004	Nombre: Buscar identificación de un entrenador Descripción: El sistema permitirá al entrenador hacer una búsqueda de su identificación y obtener información más precisa acerca de todo su trabajo dentro del club y lo que la base de datos le permita consultar con base en su id. Además, el administrador podrá buscar a algún entrenador y su información por su id en el sistema. Usuarios: Entrenador y Administrador

HU004:

Nombre: Buscar nombre de un entrenador

Historia:

Yo como entrenador quiero tener acceso a mi propia identificación (id) dentro de la base de datos diseñada para el club, con el fin de buscar información detallada acerca de mí, como la categoría que me fue asignada para dictar clase, mi nombre, mis deportistas, etc.

Yo como administrador quiero tener acceso a las identificaciones de los entrenadores que hacen parte de mi club con el fin de obtener información detallada acerca de ellos.

Criterios de Aceptación:

- Registrar nombre del entrenador
- Consultar nombre del entrenador

Puntos de esfuerzo: 1*

CÓDIGO	REQUISITO FUNCIONAL
RQF005	Nombre: Buscar categoría a la que pertenece un deportista Descripción: El sistema permitirá al deportista buscar a qué categoría pertenece, por ejemplo: infantil, menores, juvenil o mayores. Además, el administrador y los entrenadores tendrán acceso a esta información. Usuarios: Deportista, Administrador y Entrenador.

HU005:

Nombre: Buscar categoría a la que pertenece un deportista

Historia:

Yo como deportista quiero tener acceso a mi categoría

dentro del club, con el fin de saber en qué categoría tendré que entrenar y jugar; infantil, menores, juvenil o mayores. Yo como administrador quiero tener acceso a las categorías de los deportistas que hacen parte de mi club con el fin de obtener información detallada acerca de con qué entrenadores tendrán que llevar sus entrenamientos según dicha categoría. Yo como entrenador quiero tener acceso a la categoría en que los deportistas juegan, pues así sabré a quienes tendré que dictarles clase.

Criterios de Aceptación:

- Registrar categoría del deportista
- Consultar categoría del deportista

Puntos de esfuerzo: 1*

CÓDIGO	REQUISITO FUNCIONAL
RQF006	<p>Nombre: Buscar categoría a la que pertenece un entrenador.</p> <p>Descripción: El sistema permitirá al entrenador buscar a qué categoría debe de dictar clase dentro del club seguido por estas categorías: infantil, menores, juvenil y mayores. Además, el administrador y los deportistas tendrán acceso a esta información.</p> <p>Usuarios: Deportista, Administrador y Entrenador.</p>

HU006:

Nombre: Buscar categoría a la que pertenece un entrenador

Historia:

Yo como entrenador quiero tener acceso a mi categoría dentro del club, con el fin de saber en qué categoría tendré que dictar clase: infantil, menores, juvenil y mayores. Yo como administrador quiero tener acceso a las categorías de los entrenadores que hacen parte de mi club con el fin de obtener información detallada acerca de a qué deportistas tendrán que entrenar según dicha categoría. Yo como deportista quiero tener acceso a la categoría en que los entrenadores dictan sus clases, pues así sabré con quién tendré clase.

Criterios de Aceptación:

- Registrar categoría del entrenador
- Consultar categoría del entrenador

Puntos de esfuerzo: 1*

CÓDIGO	REQUISITO FUNCIONAL
RQF007	<p>Nombre: Buscar posición en la que juega un deportista</p> <p>Descripción: El sistema permitirá al deportista buscar en qué posición debe de jugar dentro de su categoría: centro, libero, punta/auxiliar, opuesto o armadora. Además, el administrador y los entrenadores tendrán acceso a esta información.</p> <p>Usuarios: Deportista, Administrador y Entrenador.</p>

HU007:

Nombre: Buscar posición en la que juega un deportista

Historia:

Yo como deportista quiero tener acceso a la posición en la que debo de jugar dentro del club, con el fin de saber en dónde tendría que ubicarse y que cosas debo de tener en cuenta durante entrenamientos y partidos: centro, libero,

punta/auxiliar, opuesto o armadora.

Yo como administrador quiero tener acceso a las posiciones de los deportistas que hacen parte de mi club con el fin de obtener información detallada acerca de sus posiciones y lo que deben de hacer según estas.

Yo como entrenador quiero tener acceso a la posición en que los deportistas juegan dentro de la categoría en la que dicto clases, pues así sabré cómo será el entrenamiento de cada uno.

Criterios de Aceptación:

- Registrar posición del deportista
- Consultar posición del deportista

Puntos de esfuerzo: 1*

CÓDIGO	REQUISITO FUNCIONAL
RQF008	<p>Nombre: Buscar eps a la que pertenece un deportista</p> <p>Descripción: El sistema permitirá al deportista buscar su eps dentro de la base de datos, esto en caso de una emergencia, pues estará registrado en su id como deportista, además le permitirá renovar esta información para que el administrador del club esté al tanto, pues es una información de gran importancia.</p> <p>Usuarios: Deportistas y Administrador</p>

HU008:

Nombre: Buscar EPS a la que pertenece un deportista

Historia:

Yo como deportista quiero tener acceso a mi propia eps con el fin de poder renovarla cuando sea necesario y para que el club esté alerta en caso de cualquier situación de accidente.

Yo como administrador quiero tener acceso a las eps de los deportistas que hacen parte de mi club con el fin de tener un plan de acción en caso de algún accidente o emergencia, además de que es un requisito del club en general.

Criterios de Aceptación:

- Registrar eps del deportista
- Consultar eps del deportista

Puntos de esfuerzo: 1*

CÓDIGO	REQUISITO FUNCIONAL
RQF009	<p>Nombre: Buscar dorsal de un deportista</p> <p>Descripción: El sistema permitirá al deportista buscar su dorsal (Número de camisa dentro del equipo) dentro de la base de datos, así no habrá 2 deportistas con el mismo número y no generará confusiones dentro del club; el administrador y el entrenador también tendrán esta información.</p> <p>Usuarios: Deportista, Administrador y Entrenador.</p>

HU009:

Nombre: Buscar el dorsal de un deportista

Historia:

Yo como deportista quiero tener acceso a la dorsal que me corresponde en el club y mi categoría, con el fin de saber mi número dentro del equipo ya que es único y también identificarme más fácil tanto a mi como a mis compañeras.

Yo como administrador quiero tener acceso a las dorsales

de los deportistas que hacen parte de mi club con el fin de saber qué número corresponde a cada deportista según su categoría y tener otro método de identificación de deportistas.

Yo como entrenador quiero tener acceso a las dorsales de los deportistas pertenecientes a mi categoría con el fin de saber identificar más fácilmente a cada uno de los deportistas a los que entreno.

Criterios de Aceptación:

- Registrar dorsal del deportista
- Consultar dorsal del deportista

Puntos de esfuerzo: 1*

CÓDIGO	REQUISITO FUNCIONAL
RQF010	Nombre: Buscar la dirección de un deportista
	Descripción: El sistema permitirá al administrador buscar la dirección de los deportistas del club. El deportista también tiene acceso a esta información.
	Usuarios: Deportistas y Administrador

HU010:

Nombre: Buscar la dirección de un deportista

Historia:

Yo como deportista quiero tener acceso a mi propia dirección con el fin de poder renovarla cuando sea necesario y para que el administrador esté alerta en caso de cualquier situación extemporánea.

Como administrador quiero tener acceso a las direcciones de los deportistas que hacen parte de mi club con el fin de tener un plan de acción en caso de alguna situación extemporánea.

Criterios de Aceptación:

- Registrar dirección del deportista
- Consultar dirección del deportista

Puntos de esfuerzo: 1*

CÓDIGO	REQUISITO FUNCIONAL
RQF011	Nombre: Buscar edad de un deportista
	Descripción: El sistema permitirá al deportista buscar su edad dentro del sistema, cosa que también tendrá decisión sobre su categoría como deportista dentro del club. El administrador y el entrenador también tendrán esta información.
	Usuarios: Deportista, Administrador y Entrenador.

HU011:

Nombre: Buscar edad de un deportista

Historia:

Yo como deportista quiero tener acceso a mi edad con el fin de saber qué categoría puede llegar a corresponderme dentro del club.

Yo como administrador quiero tener acceso a las edades de los deportistas que hacen parte de mi club con el fin de saber la categoría de cada deportista y si pueden participar en el club.

Yo como entrenador quiero tener acceso a las edades de los deportistas con el fin de saber a qué categoría pertenece

cada uno y por ende a quienes tendré que entrenar.

Criterios de Aceptación:

- Registrar edad del deportista
- Consultar edad del deportista

Puntos de esfuerzo: 1*

CÓDIGO	REQUISITO FUNCIONAL
RQF012	Nombre: Buscar nombre del acudiente de un deportista
	Descripción: El sistema permitirá al deportista buscar el nombre de su acudiente en caso de algún accidente o de algún permiso relacionado con el club. Además de esto, el administrador y el entrenador también tendrán esta información.
	Usuarios: Deportista, Administrador y Entrenador.

HU012:

Nombre: Buscar nombre del acudiente de un deportista

Historia:

Yo como deportista quiero tener acceso al nombre de mi acudiente con el fin de poder acudir a ellos en caso de una emergencia o si debo de mostrarlo en algún momento.

Yo como administrador quiero tener acceso a los nombres de los acudientes de los deportistas que hacen parte de mi club con el fin de tener un contacto al cual llamar en caso de emergencia o de alguna información importante acerca del club como sus membresías, partidos próximos, eventos, etc.

Yo como entrenador quiero tener acceso a los nombres de los acudientes de los deportistas que entrenan frecuentemente con el fin de saber a quién llamar o identificar en caso de alguna emergencia o inconveniente con el deportista.

Criterios de Aceptación:

- Registrar nombre del acudiente del deportista
- Consultar nombre del acudiente del deportista

Puntos de esfuerzo: 1*

CÓDIGO	REQUISITO FUNCIONAL
RQF013	Nombre: Buscar id acudiente
	Descripción: El sistema le permite al deportista buscar el id de su acudiente (únicamente el propio), para verificar que siga actualizado y al administrador y entrenador también le permite buscar el id del acudiente del deportista para llenar información en torneo o partidos
	Usuarios: Deportista, entrenador y administrador

HU013:

Nombre: Buscar id acudiente

Historia:

Yo como deportista quiero buscar el id de mi acudiente para verificar que siga actualizado.

Yo como administrador o entrenador quiero buscar el id de los acudientes de cada deportista en caso de ser necesario en planilla de torneos o temas externos

Criterios de Aceptación:

- Es necesario que cada usuario inicie sesión sea deportista, administrador o entrenador para que su acceso sea autenticado asegurando que solo los usuarios con los

permisos adecuados puedan realizar el tipo de búsqueda, es decir el deportista sólo puede ver la información de el mismo, en cambio el entrenador y administrador pueden buscar la el id del acudiente de cada deportista

- Registrar el id y el nombre de cada deportista ya que de esta forma el entrenador y el administrador los pueden buscar

- Registrar el id de cada acudiente de cada deportista

Puntos de esfuerzo: 1*

CÓDIGO	REQUISITO FUNCIONAL
RQF014	Nombre: Buscar teléfono acudiente
	Descripción: El sistema le permite al deportista buscar el teléfono de su acudiente (únicamente el propio), para verificar que siga actualizado y al administrador y entrenador también le permite buscar el teléfono del acudiente del deportista para llenar información en torneos o en caso de emergencias
	Usuarios: Deportista, entrenador y administrador

HU014:

Nombre: Buscar teléfono acudiente

Historia:

Yo como deportista quiero buscar el teléfono de mi acudiente para verificar que siga actualizado, yo como administrador o entrenador quiero buscar el teléfono de los acudientes de cada deportista en caso de ser necesario en planilla de torneos, emergencias o temas externos

Criterios de Aceptación:

- Es necesario que cada usuario inicie sesión sea deportista, administrador o entrenador para que su acceso sea autenticado, asegurando que solo los usuarios con los permisos adecuados puedan realizar el tipo de búsqueda, es decir el deportista sólo puede ver la información de el mismo, en cambio el entrenador y administrador pueden buscar el teléfono del acudiente de cada deportista

- Registrar el id y el nombre de cada deportista ya que de esta forma el entrenador y el administrador los pueden buscar

- Registrar el teléfono de cada acudiente de cada deportista

Puntos de esfuerzo: 1*

CÓDIGO	REQUISITO FUNCIONAL
RQF015	Nombre: Buscar correo acudiente
	Descripción: El sistema le permite al deportista buscar el correo de su acudiente (únicamente el propio), para verificar que siga actualizado y al administrador y entrenador también le permite buscar el correo del acudiente del deportista para llenar información en torneo o partidos
	Usuarios: Deportista, entrenador y administrador

HU015:

Nombre: Buscar correo acudiente

Historia:

Yo como deportista quiero buscar el correo de mi acudiente para verificar que siga actualizado, yo como administrador o entrenador quiero buscar el correo de los acudientes de cada deportista en caso de ser necesario en planilla de torneos, emergencias o temas externos

Criterios de Aceptación:

- Es necesario que cada usuario inicie sesión sea deportista, administrador o entrenador para que su acceso sea autenticado, asegurando que solo los usuarios con los permisos adecuados puedan realizar el tipo de búsqueda, es decir el deportista sólo puede ver la información de el mismo, en cambio el entrenador y administrador pueden buscar el correo del acudiente de cada deportista

- Registrar el id y el nombre de cada deportista ya que de esta forma el entrenador y el administrador los pueden buscar

- Registrar el correo de cada acudiente de cada deportista

Puntos de esfuerzo: 1*

CÓDIGO	REQUISITO FUNCIONAL
RQF016	Nombre: Buscar horario del deportista
	Descripción: El sistema le permite al deportista buscar su horario de entrenamiento (únicamente el propio), para conocer sus días de entreno, a su vez tanto el entrenador como administrador pueden buscar el horario del deportista para conocer o confirmar sus días de entreno
	Usuarios: Deportista, entrenador y administrador

HU016:

Nombre: Buscar horario del deportista

Historia:

Yo como deportista quiero buscar mi horario de entrenamiento para conocer los días de entreno, yo como administrador o entrenador quiero buscar el horario de entrenamiento de cada deportista para conocerlos y/o verificarlos

Criterios de Aceptación:

- Es necesario que cada usuario inicie sesión sea deportista, administrador o entrenador para que su acceso sea autenticado, asegurando que solo los usuarios con los permisos adecuados puedan realizar el tipo de búsqueda, es decir el deportista sólo puede ver la información de el mismo, en cambio el entrenador y administrador pueden buscar el horario de cada deportista

- Registrar el id y el nombre de cada deportista ya que de esta forma el entrenador y el administrador los pueden buscar

- Registrar el horario de entrenamiento de cada deportista

Puntos de esfuerzo: 1*

CÓDIGO	REQUISITO FUNCIONAL
RQF017	Nombre: Buscar horario del entrenador
	Descripción: El sistema le permite al entrenador buscar su horario de entrenamiento (únicamente el propio), para conocer sus días de entreno, a su vez pueden buscar el horario del entrenador para conocer o confirmar sus días de entreno
	Usuarios: Entrenador y administrador

HU017:

Nombre: Buscar horario del entrenador

Historia:

Yo como entrenador quiero buscar mi horario de entrenamiento para conocer los días de entreno, yo como

administrador buscar el horario de entrenamiento de cada entrenador para conocerlos y/o verificarlos

Criterios de Aceptación:

- Es necesario que cada usuario inicie sesión sea deportista, administrador o entrenador para que su acceso sea autenticado, asegurando que solo los usuarios con los permisos adecuados puedan realizar el tipo de búsqueda, es decir los deportistas no puede ver esta información, el entrenador solo puede ver la de él mismo y el administrador pueden buscar el horario de cada entrenador
- Registrar el id y el nombre de cada entrenador ya que de esta forma el administrador los puede buscar
- Registrar el horario de entrenamiento de cada entrenador

Puntos de esfuerzo: 1*

CÓDIGO	REQUISITO FUNCIONAL
RQF018	Nombre: Modificar nombre del deportista
	Descripción: El sistema le permite al administrador modificar el nombre de cada deportista para mantener la base de datos actualizada y en caso de error permitir corregirlo
	Usuarios: Administrador

HU018:

Nombre: Modificar nombre del deportista

Historia:

Yo como administrador quiero modificar el nombre de cada deportista para mantener la base de datos actualizada y en caso de error permitir corregirlo

Criterios de Aceptación:

- Es necesario que cada usuario inicie sesión sea deportista, administrador o entrenador para que su acceso sea autenticado, asegurando que solo los usuarios con los permisos adecuados puedan realizar el tipo de búsqueda, es decir ni el deportista ni el entrenador pueden realizar esta acción solo el administrador puede modificar el nombre de cada deportista
- Registrar el id y el nombre de cada deportista ya que de esta forma el administrador los puede buscar

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF019	Nombre: Modificar id del deportista
	Descripción: El sistema le permite al administrador modificar el id de cada deportista para mantener la base de datos actualizada y en caso de error permitir corregirlo
	Usuarios: Administrador

HU019:

Nombre: Modificar id del deportista

Historia:

Yo como administrador quiero modificar el id de cada deportista para mantener la base de datos actualizada y en caso de error permitir corregirlo

Criterios de Aceptación:

- Es necesario que cada usuario inicie sesión sea deportista, administrador o entrenador para que su acceso sea autenticado, asegurando que solo los usuarios con los permisos adecuados puedan realizar el tipo de búsqueda, es decir ni el deportista ni el entrenador pueden realizar esta acción solo el administrador puede modificar el nombre de cada deportista
- Registrar el id y el nombre de cada deportista ya que de esta forma el administrador los puede buscar

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF020	Nombre: Modificar nombre del entrenador
	Descripción: El sistema le permite al administrador modificar el nombre de cada entrenador para mantener la base de datos actualizada y en caso de error permitir corregirlo
	Usuarios: Administrador

HU020:

Nombre: Modificar nombre del entrenador

Historia:

Yo como administrador quiero modificar el nombre de cada entrenador para mantener la base de datos actualizada y en caso de error permitir corregirlo

Criterios de Aceptación:

- Es necesario que cada usuario inicie sesión sea deportista, administrador o entrenador para que su acceso sea autenticado, asegurando que solo los usuarios con los permisos adecuados puedan realizar el tipo de búsqueda, es decir ni el deportista ni el entrenador pueden realizar esta acción solo el administrador puede modificar el nombre de cada entrenador
- Registrar el id y el nombre de cada entrenador ya que de esta forma el administrador los puede buscar

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF021	Nombre: Modificar id del entrenador
	Descripción: El sistema le permite al administrador modificar el id de cada entrenador para mantener la base de datos actualizada y en caso de error permitir corregirlo
	Usuarios: Administrador

HU021:

Nombre: Modificar id del entrenador

Historia:

Yo como administrador quiero modificar el id de cada entrenador para mantener la base de datos actualizada y en caso de error permitir corregirlo

Criterios de Aceptación:

- Es necesario que cada usuario inicie sesión sea deportista, administrador o entrenador para que su acceso sea autenticado, asegurando que solo los usuarios con los permisos adecuados puedan realizar el tipo de búsqueda, es decir ni el deportista ni el entrenador pueden realizar

esta acción solo el administrador puede modificar el id de cada entrenador

- Registrar el id y el nombre de cada entrenador ya que de esta forma el administrador los puede buscar

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF022	Nombre: Modificar nombre del administrador
	Descripción: El sistema le permite al administrador modificar su nombre para mantener la base de datos actualizada y en caso de error permitir corregirlo
	Usuarios: Administrador

HU022:

Nombre: Modificar nombre del administrador

Historia:

Yo como administrador quiero modificar mi nombre para mantener la base de datos actualizada y en caso de error permitir corregirlo

Criterios de Aceptación:

- Es necesario que cada usuario inicie sesión sea deportista, administrador o entrenador para que su acceso sea autenticado, asegurando que solo los usuarios con los permisos adecuados puedan realizar el tipo de búsqueda, es decir ni el deportista ni el entrenador pueden realizar esta acción solo el administrador puede modificar su nombre
- Registrar el id y el nombre del administrador para que se pueda buscar

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF023	Nombre: Modificar id de administrador
	Descripción: El sistema le permite al administrador modificar su id para mantener la base de datos actualizada y en caso de error permitir corregirlo
	Usuarios: Administrador

HU023:

Nombre: Modificar id del administrador

Historia:

Yo como administrador quiero modificar mi id para mantener la base de datos actualizada y en caso de error permitir corregirlo

Criterios de Aceptación:

- Es necesario que cada usuario inicie sesión sea deportista, administrador o entrenador para que su acceso sea autenticado, asegurando que solo los usuarios con los permisos adecuados puedan realizar el tipo de búsqueda, es decir ni el deportista ni el entrenador pueden realizar esta acción solo el administrador puede modificar su id
- Registrar el id y el nombre del administrador para que se pueda buscar

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF024	Nombre: Modificar categorías deportistas
	Descripción: El sistema le permite al administrador modificar la categoría de cada deportista según su edad (infantil, menores, juvenil y mayores) para dividir categorías según correspondan y al momento de cambio le permita modificar
	Usuarios: Administrador

HU024:

Nombre: Modificar categorías deportistas

Historia:

Yo como administrador quiero modificar la categoría de cada deportista según su edad (infantil, menores, juvenil y mayores) para dividir categorías según correspondan y al momento de cambio le permita modificar

Criterios de Aceptación:

- Es necesario que cada usuario inicie sesión sea deportista, administrador o entrenador para que su acceso sea autenticado, asegurando que solo los usuarios con los permisos adecuados puedan realizar el tipo de búsqueda, es decir ni el deportista ni el entrenador pueden realizar esta acción solo el administrador puede modificar la categoría de cada deportista
- Registrar el id, el nombre y la edad del deportista para que se pueda buscar
- Registrar las categorías según las edades para registrar a cada deportista en una categoría

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF025	Nombre: Encontrar a los deportistas mediante su documento de identidad.
	Descripción: El sistema permitirá al administrador hacer una búsqueda eficiente para encontrar al deportista, con esto que aparezca información de este, como parientes, tipo de sangre, número telefónico, etc.
	Usuarios: Administrador, deportistas.

HU025:

Nombre: Buscar a los deportistas mediante su documento de identidad.

Historia:

- Yo como administrador quiero hacer una búsqueda por documento
- Para tener una base actualizada de datos.

Criterios de Aceptación:

- Crear una interfaz de usuario.
- Validar la información que esté completa.
- Buscar a los deportistas mediante su documento de identidad.

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF026	Nombre: Gestionar competencias futuras en las que se pueda participar. Descripción: El sistema permitirá al administrador hacer una búsqueda para saber qué competencias se tienen en cuenta para participar en un futuro, también el sistema permitirá al administrador ingresar qué competencias se tienen en cuenta para participar y que este salte una alerta cuando se acerque esta. Usuarios: Administrador

HU026:

Nombre: Gestionar competencias futuras en las que se pueda participar.

Historia:

- Yo como administrador quiero hacer un registro y búsqueda por competencia.
- Quiero consultar y agregar competencias en las que se quiera participar en un futuro.
- Para que se pueda tener un registro sobre las competencias que estén cerca a la fecha "actual".

Criterios de Aceptación:

- Crear una interfaz de búsqueda.
- Validar datos de competencias (lugar, fecha, precio).
- Permitir agregar información cuando se quiera agregar una competencia.

Puntos de esfuerzo: 3***

CÓDIGO	REQUISITO FUNCIONAL
RQF027	Nombre: Modificar la categoría del Entrenador. Descripción: El sistema permitirá al administrador modificar la categoría asignada a un entrenador, seleccionando entre las opciones: Infantil, Prejuvenil, Juvenil y Mayores. Usuarios: Administrador, entrenador.

HU027:

Nombre: Modificar la categoría del Entrenador.

Historia:

- Yo como administrador quiero modificar la categoría del entrenador.
- Para que se pueda tener en la base de datos un registro organizado sobre quienes entrenan a cada categoría.

Criterios de Aceptación:

- Registrar la categoría del entrenador.
- Modificar y/o cambiar la categoría.

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF028	Nombre: Modificar la posición del deportista. Descripción: El sistema permitirá al administrador modificar la posición del deportista, cuando sea necesario y el entrenador lo haya notificado, esto para tener un registro más ordenado sobre el equipo. Usuarios: Administrador, deportista.

HU028:

Nombre: Modificar la categoría del Entrenador.

Historia:

- Yo como administrador quiero modificar la categoría del entrenador.
- Para que se pueda tener en la base de datos un registro organizado sobre quienes entrenan a cada categoría.

Criterios de Aceptación:

- Registrar la categoría del entrenador.
- Modificar y/o cambiar la categoría.

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF029	Nombre: Modificar la EPS del deportista. Descripción: El sistema permitirá al administrador modificar la EPS del deportista, siempre y cuando el deportista haya notificado que esta se ha cambiado, así mismo el deportista tiene acceso a ver la información para corroborar que está bien, pero, no podrá modificar información. Usuarios: Administrador, deportista.

HU029:

Nombre: Modificar la EPS del deportista.

Historia:

- Yo como administrador quiero modificar la EPS del deportista.
- Para que se mantenga un registro actualizado de salud del deportista y este garanticen información precisa en el sistema.

Criterios de Aceptación:

- Validar EPS correcta.
- Modificar y/o actualizar información de la EPS en la base de datos.
- Cambiar antigua a nueva EPS.

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF030	Nombre: Modificar la EPS del entrenador. Descripción: El sistema permitirá al administrador modificar la EPS del entrenador, siempre y cuando el entrenador haya notificado que esta se ha cambiado, de la misma manera el entrenador podrá tener acceso, pero este no podrá modificar información. Usuarios: Administrador, entrenador.

HU030:

Nombre: Modificar la EPS del entrenador.

Historia:

- Yo como administrador quiero modificar la EPS del entrenador.
- Para que se pueda tener una información de salud del entrenador más precisa en la base de datos.

Criterios de Aceptación:

- Validar EPS correcta.
- Modificar y/o actualizar información de la EPS en la base de datos.
- Cambiar antigua a nueva EPS.

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF031	Nombre: Modificar la mensualidad del deportista. Descripción: El sistema permitirá al administrador modificar la mensualidad del deportista, esto dependerá en la categoría que se encuentre, pues estos varían dependiendo de las cuatro categorías en las que se encuentre. Usuarios: Administrador, deportista.

HU031:

Nombre: Modificar la mensualidad del deportista.

Historia:

- Yo como administrador quiero modificar la mensualidad del deportista.
- Para asegurar que el registro de cobros esté actualizado y refleje correctamente los valores establecidos por cada deportista.

Criterios de Aceptación:

- Permite seleccionar al deportista cuyo valor de mensualidad se modificará.
- Ingresar o modificar un nuevo valor de mensualidad.

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF032	Nombre: Modificar el Salario del Entrenador. Descripción: El sistema permitirá al administrador modificar el salario del entrenador, este varía dependiendo de la categoría a la que este entrene, pues este varía dependiendo de la categoría a la que el entrenador esté asignado. Usuarios: Administrador, entrenador.

HU032:

Nombre: Modificar el salario del entrenador.

Historia:

- Yo como administrador quiero modificar el salario del entrenador.
- Para asegurar que el registro de pagos este actualizado y refleje correctamente los valores establecidos por cada entrenador.

Criterios de Aceptación:

- Permite seleccionar al entrenador cuyo valor de salario se modificará.
- Ingresar o modificar un nuevo valor de salario.

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF033	Nombre: Modificar el dorsal del deportista. Descripción: El sistema permitirá al administrador modificar el dorsal del deportista, todo esto si el deportista pide el cambio del dorsal, esto ayudará a tener una base de datos más arreglada para organizar los uniformes. Usuarios: Administrador, deportista.

HU033:

Nombre: Modificar el dorsal del deportista.

Historia:

- Yo como administrador quiero modificar el dorsal del deportista.
- Para que se mantenga actualizada la asignación de números de dorsal y con esto se garantice la identificación en competencias establecidas por cada deportista.

Criterios de Aceptación:

- Seleccionar al deportista cuyo dorsal se modificará.
- Se muestra el dorsal actual e histórico del deportista.

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF034	Nombre: Modificar dirección deportista. Descripción: El sistema permitirá al administrador modificar la dirección de residencia del deportista, esto sucederá únicamente si el deportista cambia de lugar de residencia, de ser así tendrá que notificarlo al administrador del club. Usuarios: Administrador, deportista.

HU034:

Nombre: Modificar dirección deportista.

Historia:

- Yo como administrador quiero modificar la dirección del deportista.
- Para mantener actualizada la información de contacto y asegurar que la dirección de la residencia del deportista sea la correcta.

Criterios de Aceptación:

- Seleccionar deportista para modificar la dirección.
- Mostrar la dirección actual.
- Validar direcciones, que sean verídicas.

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF035	Nombre: Modificar edad del deportista.
	Descripción: El sistema permitirá al administrador modificar la edad del deportista, todo esto para saber bien en qué categoría queda el deportista, si llega a cumplir la edad mínima de una categoría y excede otra, este cambiará mediante su edad.
	Usuarios: Administrador, deportista.

HU035:

Nombre: Modificar edad del deportista.

Historia:

- Yo como administrador quiero modificar la edad del deportista.
- Para mantener un registro actualizado de la edad de cada deportista en el sistema.

Criterios de Aceptación:

- Seleccionar deportista para ver la edad actual.
- Ingresar y/o modificar la edad.

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF036	Nombre: Modificar el nombre del acudiente.
	Descripción: El sistema permitirá al administrador modificar el nombre del acudiente, de ser necesario, el deportista tendrá que notificar el cambio.
	Usuarios: Administrador, deportista.

HU036:

Nombre: Modificar el nombre del acudiente.

Historia:

- Yo como administrador quiero modificar el nombre del acudiente.
- Para mantener actualizada la información del acudiente de cada deportista en la base de datos.

Criterios de Aceptación:

- Validar el nombre del acudiente.
- Modificar nombre del acudiente, si es necesario.

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF037	Nombre: Modificar el ID del acudiente.
	Descripción: El sistema permitirá al administrador modificar el ID del acudiente, todo esto tiene que estar junto al dato de nombre, es decir, si se cambia nombre, el ID también tendrá que ser cambiado.
	Usuarios: Administrador, deportista.

HU037:

Nombre: Modificar el ID del acudiente.

Historia:

- Yo como administrador quiero modificar el ID del acudiente.
- Para asegurar un identificador único a cada

deportista que será el acudiente, esto para evitar duplicados de información.

Criterios de Aceptación:

- Validar ID del acudiente.
- Modificar ID del acudiente, si es necesario.

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF038	Nombre: Modificar el teléfono del acudiente.
	Descripción: El sistema permitirá al administrador modificar el teléfono del acudiente, siempre y cuando sea notificado, todo esto para tener el contacto encargado del deportista por si cualquier suceso inesperado que suceda dentro del entrenamiento.
	Usuarios: Administrador

HU038:

Nombre: Modificar el teléfono del acudiente.

Historia:

- Yo como administrador quiero modificar el teléfono del acudiente.
- Para asegurar que la información del acudiente este actualizada, con esto facilitar la comunicación con este, para futuras comunicaciones o reportes respecto al deportista.

Criterios de Aceptación:

- Validar el teléfono del acudiente.
- Modificar teléfono del acudiente, si es necesario.

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF039	Nombre: Modificar correo acudiente.
	Descripción: El administrador de la base de datos podrá realizar cambios en el correo registrado del acudiente de un deportista, para mantener la información actualizada.
	Usuarios: Administrador.

HU039:

Nombre: Modificar correo acudiente.

Historia:

- Yo como administrador quiero modificar el correo del correo electrónico de un acudiente asociado a un deportista.
- Para tener una base actualizada de datos y garantizar una buena comunicación.

Criterios de Aceptación:

- El deportista debe tener un acudiente asociado.
- El sistema debe permitir realizar la búsqueda del deportista por el nombre o id.
- La información del acudiente debe estar completa y debe ser verídica.

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF040	Nombre: Registrar deportista.
	Descripción: La base de datos permitirá al administrador inscribir a un nuevo deportista que ingrese al club, ingresando en la base de datos información como nombre, edad, posición, la información del acudiente, etc.
	Usuarios: Administrador

HU040:

Nombre: Registrar Deportista

Historia:

- Yo como administrador quiero inscribir a un nuevo deportista.
- Para que pueda participar de entrenamientos y torneos.

Criterios de Aceptación:

- El deportista no puede estar ya registrado.
- Creación de la interfaz de administrador.
- Se le debe asignar un horario y entrenador.

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF041	Nombre: Registrar Entrenador
	Descripción: El administrador podrá inscribir a nuevos entrenadores que hagan parte del club de voleibol, es decir registrará toda su información, (nombre, teléfono, correo, etc.), y se les asignará un horario.
	Usuarios: Administrador

HU041:

Nombre: Registrar Entrenador

Historia:

- Yo como administrador quiero registrar un nuevo entrenador.
- Para que pueda dirigir las sesiones de entrenamiento y partidos.

Criterios de Aceptación:

- El sistema debe verificar que no haya otro entrenador con el mismo id.
- El sistema debe permitir que se le asigne un horario.
- Creación de la interfaz de administrador.

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF042	Nombre: Registrar horario
	Descripción: El sistema permitirá la creación de nuevos horarios de entrenamiento, especificando fecha, hora y lugar.
	Usuarios: Administrador

HU042:

Nombre: Registrar Horario

Historia:

- Yo como administrador quiero registrar nuevos horarios.
- Para que tanto entrenadores como deportistas tengan claros los horarios y puedan tener una planificación clara.

Criterios de Aceptación:

- Creación de interfaz de administrador.
- Asignar el horario a un grupo de deportistas y entrenadores.
- Debe validar que los horarios de un entrenador no se crucen.

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF043	Nombre: Modificar horario
	Descripción: El sistema tendrá la opción de modificar horarios de entrenamiento existentes. Es decir, se modificarán la fecha, hora y lugar.
	Usuarios: Administrador

HU043:

Nombre: Modificar horario.

Historia:

- Yo como administrador quiero modificar un horario.
- Para realizar cambios en la planeación cuando se requiera.

Criterios de Aceptación:

- Creación de interfaz de administrador.
- Realizar búsqueda de horarios existentes.
- Debe validar que al realizar un cambio en un horario no hayan conflictos con otros horarios existentes.

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF044	Nombre: Buscar el salario del entrenador
	Descripción: El administrador y el entrenador podrán consultar el salario de un entrenador. La búsqueda puede realizarse por nombre o ID del entrenador.
	Usuarios: Administrador, entrenador

HU044:

Nombre: Buscar salario del entrenador.

Historia:

- Yo como administrador quiero el salario de un entrenador.
- Para gestionar pagos y poder hacer el presupuesto del club.

Criterios de Aceptación:

- Búsqueda del entrenador por id o nombre.
- Solo el administrador y el entrenador asociado pueden tener acceso a esta información.
- Visualización del sueldo base, bonos.

Puntos de esfuerzo: 1**

CÓDIGO	REQUISITO FUNCIONAL
RQF045	Nombre: Buscar horario
	Descripción: El sistema permitirá que tanto deportistas, entrenadores y administrador puedan consultar los horarios de entrenamiento. La búsqueda debe poder realizarse por nombre, categoría o fecha específica, mostrando la información relevante de manera organizada.
	Usuarios: Deportistas, Entrenadores, Administrador

HU045:

Nombre: Buscar Horario.

Historia:

- Yo como deportista o entrenador quiero consultar mi horario de entrenamiento.
- Para poder planear mi día o semana.

Criterios de Aceptación:

- Búsqueda por nombre, categoría o día específico.
- Debe mostrar fecha, hora y lugar.
- El horario se debió haber registrado anteriormente.

Puntos de esfuerzo: 1**

CÓDIGO	REQUISITO FUNCIONAL
RQF046	Nombre: Modificar el horario de un deportista
	Descripción: El administrador podrá cambiar el horario de un deportista, para asignarlo a un nuevo grupo o por el cambio de horario de esa categoría.
	Usuarios: Administrador

HU046:

Nombre: Modificar horario de un deportista.

Historia:

- Yo como administrador quiero realizar cambios en el horario de un deportista.
- Para poder cambiarlo de grupo y ajustar su disponibilidad.

Criterios de Aceptación:

- El deportista debe tener registrado un horario
- La búsqueda se puede realizar por id o nombre del deportista.
- El nuevo horario no debe generar conflictos con otros horarios.

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF047	Nombre: Modificar el horario de un entrenador
	Descripción: El sistema permitirá al administrador realizar cambios en los horarios asignados a un entrenador.
	Usuarios: Administrador

HU047:

Nombre: Modificar el horario de un entrenador.

Historia:

- Yo como administrador quiero realizar cambios en el horario de un entrenador.
- Para poder ajustar la disponibilidad o cambiarlo de grupo.

Criterios de Aceptación:

- El entrenador debe tener un horario ya registrado
- La búsqueda se debe poder realizar por id o nombre.
- No puede haber conflictos entre el nuevo horario y otros ya existentes.

Puntos de esfuerzo: 2**

CÓDIGO	REQUISITO FUNCIONAL
RQF048	Nombre: Buscar horario entrenador
	Descripción: Tanto los entrenadores como el administrador podrán consultar los horarios de entrenamiento asignados a un entrenador específico. La búsqueda se podrá realizar por nombre o ID del entrenador.
	Usuarios: Entrenadores, administrador

HU048:

Nombre: Buscar el horario de un entrenador.

Historia:

- Yo como administrador o entrenador quiero consultar el horario de un entrenador.
- Para poder mejorar la planificación de las sesiones.

Criterios de Aceptación:

- El entrenador debe tener un horario ya registrado.
- La búsqueda se debe poder realizar por id o nombre.
- La consulta debe mostrar la información organizada por: fecha, hora y lugar.

Puntos de esfuerzo: 1**

CÓDIGO	REQUISITO FUNCIONAL
RQF049	Nombre: Buscar la mensualidad de un deportista
	Descripción: La base de datos permitirá a los jugadores y al administrador consultar el precio de la mensualidad que paga el deportista, con el fin de mantener los datos actualizados y evitar problemas de un cobro erróneo.
	Usuarios: Administrador, Deportista

HU049:

Nombre: Buscar la mensualidad de un deportista.

Historia:

- Yo como administrador quiero consultar la mensualidad de un deportista.
- Para poder llevar un control de los pagos de los deportistas, evitar pagos erróneos y poder hacer el presupuesto del club.

Criterios de Aceptación:

- El deportista debe estar activo en el club.
- La búsqueda se debe poder hacer por id o por nombre.
- El sistema debe mostrar el estado del pago (pendiente, pago, atrasado).

Puntos de esfuerzo: 1**

CÓDIGO	REQUISITO FUNCIONAL
RQF050	Nombre: Buscar el salario de un entrenador Descripción: La base de datos permitirá a los entrenadores y al administrador consultar el sueldo del entrenador, con el fin de mantener los datos actualizados. Usuarios: Administrador, Entrenador.

HU050:

Nombre: Buscar el sueldo de un entrenador.

Historia:

- Yo como administrador quiero buscar el sueldo de un entrenador.
- Para gestionar los pagos y llevar un control de los gastos del club.

Criterios de Aceptación:

- El entrenador debe estar activo en el club.
- La búsqueda se debe poder hacer por id o por nombre.
- Solo el administrador puede acceder a esta información.

Puntos de esfuerzo: 1**

CÓDIGO	REQUISITO FUNCIONAL
RQF051	Nombre: Inicio de Sesión Descripción: El sistema permitirá tanto a los deportistas como al administrador dividir la información de la base de datos dependiendo de cómo se registre al momento de inicio de sesión (tipo de usuario). Usuarios: Administrador y deportistas.

HU051:

Nombre: Inicio de Sesión.

Historia:

Yo como administrador quiero consultar una base de datos en donde tenga acceso a la información de mis deportistas dentro del club deportivo y además de esto poder modificar aspectos como los horarios, los deportistas que jugarán en cierto partido, los entrenamientos, etc.

Yo como deportista quiero consultar una base de datos en donde pueda consultar mis horarios, categoría como deportista, partidos y entrenamientos que tengo pendientes durante la semana.

Criterios de Aceptación:

- Registrar el documento de identidad de cada deportista y del administrador para poder lograr un Registro diferencial entre administrador y deportistas.
- Generar un usuario y contraseña por cada deportista y para el administrador.
- Generar una validación de usuario por medio de una contraseña.

Generar la interfaz de inicio de sesión

Puntos de esfuerzo: 4**

Además, tenemos dos requisitos NO FUNCIONALES

Requisitos No Funcionales (RQNF):

CÓDIGO	REQUISITO FUNCIONAL
RQNF001	Nombre: Seguridad de la Información Descripción: El sistema se asegurará de la autenticación de cada usuario y el control de datos.

CÓDIGO	REQUISITO FUNCIONAL
RQNF002	Nombre: Eficiencia en la gestión de datos Descripción: El sistema permitirá realizar consultas rápidas y optimizadas

Diagrama de Clases:

El diagrama de clases que ha surgido con base en las ideas de todos los integrantes, tenemos 6 clases: Usuario, Administrador, Deportista, Entrenador, Evento y Horario.

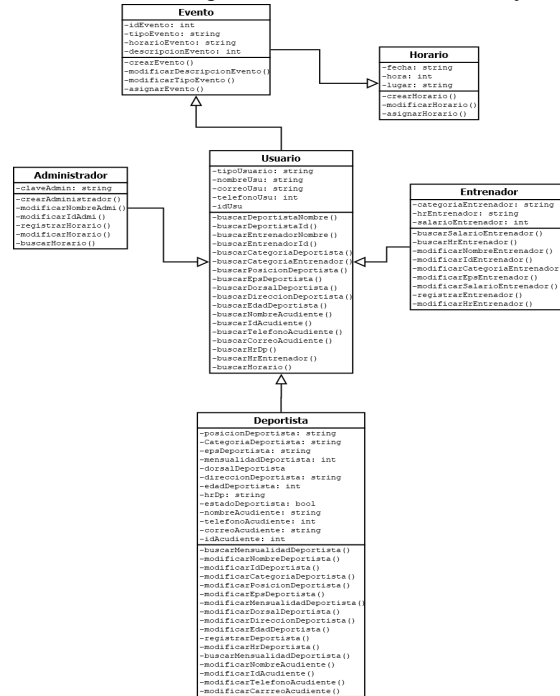


Imagen 3 Diagrama De clases generado en: Dia

Modelo Conceptual:

Image 4 Modelo Conceptual generado en: Dia

Modelo Lógico:

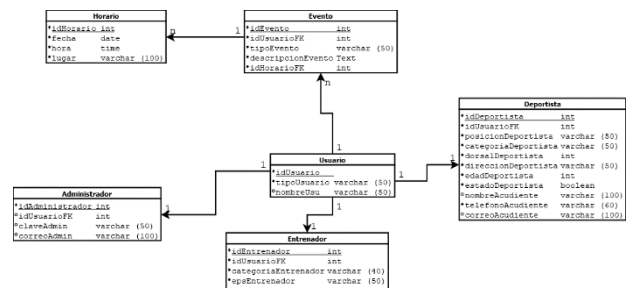


Imagen 5 Modelo Lógico generado en: Dia

6 Modelo Físico:

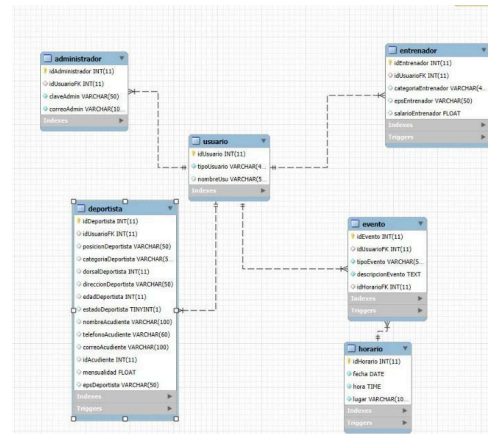


Imagen 6 Modelo Físico Creado en MySQL

Diccionario de Datos:

Administrador				
Nombre Tabla				
Fecha	6/04/2025			
Descripción	tabla con la información de un usuario tipo administrador			
Campo	Tipo de Dato	Tamaño	Restricciones	Descripción
idAdministrador	INT		Primary Key - Auto Incrementable	Llave primaria de administrador
idUsuario	INT		Foreign Key, not null	Llave foránea de Usuario referenciada en administrador
ClaveAdmin	VARCHAR	50 not null		Clave del Administrador
CorreoAdmin	VARCHAR	100 not null		Correo del Administrador

Entrenador				
Nombre Tabla				
Fecha	6/04/2025			
Descripción	tabla con la información de un usuario tipo entrenador			
Campo	Tipo de Dato	Tamaño	Restricciones	Descripción
idEntrenador	int		Primary Key - Auto Incrementable	Llave primaria de entrenador
idUsuarioFK	int		Foreign Key, not null	Llave foránea del id del usuario
categoríaEntrenador	varchar	40 not null		categoría a la que el entrenador da clases
epsEntrenador	varchar	50 not null		Eps a la que está afiliado el entrenador
salario	float			salario del entrenador

Evento				
Nombre Tabla				
Fecha	6/04/2025			
Descripción	tabla con la información de los eventos del club			
Campo	Tipo de Dato	Tamaño	Restricciones	Descripción
idEvento	int		Primary key	id del evento
tipoEvento	varchar	50 not null		tipo de evento
descripciónEvento	text			descripción del evento registrado
idUsuarioFK	int		Foreign key, not null	id del usuario que tiene ese evento
idHorarioFK	int		Foreign key, not null	id del horario que tiene ese evento

Usuario				
Nombre Tabla				
Fecha	6/04/2025			
Descripción	Tabla con la información de los usuarios			
Campo	Tipo de Dato	Tamaño	Restricciones	Descripción
idUsuario	int		Primary key	Llave primaria de cada usuario
tipUsuario	varchar	40 not null		tipo de usuario
nombreUsuario	varchar	50 not null		nombre del Usuario

Deportista				
Nombre Tabla				
Fecha	6/04/2025			
Descripción	tabla con la información de los usuarios tipo deportistas			
Campo	Tipo de Dato	Tamaño	Restricciones	Descripción
idDeportista	int		Primary key, Auto Incrementable	Llave primaria de cada usuario
posicionDeportista	varchar	50 not null		posicion en la que juega el deportista
categoríaDeportista	varchar	50 not null		categoría a la que pertenece el deportista
epsDeportista	varchar	50 not null		eps a la que está afiliado el deportista
mensualidad	float		not null	mensualidad que paga el deportista
dorsal	int			numero de camiseta del jugador
direcciónDeportista	varchar	50 not null		dirección de residencia del deportista
edad	int		not null	edad del deportista
estadoDp	boolean		not null	estado del deportista(activo o inactivo)
nombreAcudiente	varchar	100 null		nombre del acudiente del deportista
telefonoAcudiente	varchar	20 null		telefono del acudiente
correoAcudiente	varchar	100 null		correo del acudiente
identificaciónDeportista	varchar	40 not null		
idUsuarioFK			Foreign Key	Llave foránea de Usuario referenciada en deportista
idEventoFK			Foreign Key	Llave foránea de evento referenciada en deportista

Horario				
Nombre Tabla				
Fecha	6/04/2025			
Descripción	tablas con los horarios relacionados con un evento			
Campo	Tipo de Dato	Tamaño	Restricciones	Descripción
idHorario	int		Primary key	id del horario
fecha	date		not null	fecha del evento
hora	time		not null	hora del evento
lugar	varchar	100 not null		lugar del evento

Imagen 7 Diccionario de Datos

```

create table usuario (
    idusuario INT auto_increment PRIMARY KEY,
    tipousuario VARCHAR(40) NOT NULL,
    nombreusu varchar (50)
);

create table administrador(
    idAdministrador INT auto_increment primary key,
    idusuarioFK int,
    claveAdmin varchar(50) NOT NULL,
    correoAdmin varchar(100) NOT NULL,
    FOREIGN KEY (idusuarioFK) REFERENCES usuario(idusuario) ON DELETE SET NULL ON UPDATE CASCADE
);

create table deportista(
    idDeportista INT AUTO_INCREMENT PRIMARY KEY,
    idusuarioFK int,
    posicionDeportista varchar(50) not null,
    categoriaDeportista varchar(50) not null,
    dorsalDeportista int not null,
    direccionDeportista varchar(50) not null,
    edadDeportista int not null,
    estadoDeportista boolean not null,
);

create table entrenador(
    idEntrenador int auto_increment primary key,
    idusuarioFK int,
    categoriaEntrenador varchar (40),
    epsEntrenador varchar(50),
    salarioEntrenador float,
    FOREIGN KEY (idusuarioFK) REFERENCES usuario(idusuario) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE horario (
    idhorario int auto_increment primary key,
    fecha date NOT NULL,
    hora time NOT NULL,
    lugar varchar(100) NOT NULL
);

CREATE TABLE evento (
    idEvento int auto_increment primary key,
    idusuarioFK int,
    tipoEvento varchar(50) NOT NULL,
    descripcionEvento TEXT NOT NULL,
    idhorarioFK int,
    FOREIGN KEY (idhorarioFK) REFERENCES horario(idhorario) ON DELETE SET NULL ON UPDATE CASCADE,
    FOREIGN KEY (idusuarioFK) REFERENCES usuario(idusuario) ON DELETE SET NULL ON UPDATE CASCADE
);

```

Imagen 8 creación de tablas en MySQL

- Inserciones:

```

-------USUARIO-----
-- Inserciones de deportistas en la tabla usuario
INSERT INTO usuario (idusuario, tipousuario, nombreusu)
VALUES
('Administrador', 'Kevin Velásquez'),
('Entrenador', 'Carlos Andrés Henao'),
('Entrenador', 'Juliana Ramírez Soto'),
('Entrenador', 'Oscar Giovanni López'),
('Entrenador', 'Patricia Hernández Castro'),
('Deportista', 'Carolina Dolores Mederos Silveira'),
('Deportista', 'Elizabeth García Arias'),
('Deportista', 'Luciana Dussan Molina'),
('Deportista', 'Laura Camila Galvis Nieto'),
('Deportista', 'Sofía Alejandra Henao'),
('Deportista', 'Valentina Ríos Castro'),
('Deportista', 'Nicolai Gabriela Pajerito Gómez'),
('Deportista', 'María Alejandra Ortega Cruz'),
('Deportista', 'Danna Isabella García Arias'),
('Deportista', 'Elly Estefanía Velasco Bonilla'),
('Deportista', 'Lina Estefanía Fonseca Sierra'),
('Deportista', 'Nicolai Gabriela Pajerito Gómez'),
('Deportista', 'Alejandro Ramírez Molina'),
('Deportista', 'Santiago Gómez López'),
('Deportista', 'Daniela Torres Sánchez'),
('Deportista', 'Marlene Vargas Ojeda'),
('Deportista', 'César Porras Paula Daniela'),
('Deportista', 'Natalia Cabana Enciso'),
('Deportista', 'María Fernanda Rodríguez'),
('Deportista', 'Valerie Cruz García'),
('Deportista', 'Ana Sofía Forero Berdugo'),

```

Imagen 9 Inserciones en tablas creadas en MySQL

- Consultas:

Creación de Base Relacional

- Creación de tablas:

```

#----- ADD COLUMN -----
-- Añadir columna mensualidad y registro
ALTER TABLE deportista ADD COLUMN mensualidad INT;
-- Añadir columna mensualidad y registro
ALTER TABLE deportista ADD COLUMN mensualidad FLOAT;
-- Añadir columna eps a deportista y registro
ALTER TABLE deportista ADD COLUMN epsdeportista VARCHAR(40);

#----- Búsquedas -----
-- Buscar deportista por nombre
SELECT * FROM usuario WHERE nombreUs LIKE "Daniel%";
-- Buscar deportista por id
SELECT * FROM deportista WHERE iddeportista = 40;
-- Buscar entrenador por nombre
SELECT * FROM usuario WHERE nombreUs LIKE "Oscar%";
-- Buscar entrenador por id
SELECT * FROM entrenador WHERE identrenador = 3;
-- Buscar deportista por categoría
SELECT * FROM deportista WHERE categoriadeportista = "Juvenil";
-- Buscar entrenador por categoría
SELECT * FROM entrenador WHERE categoriadeentrenador = "Infantil";
-- Buscar deportista por posición
SELECT * FROM deportista WHERE posiciondeportista = "Auxiliar";
-- Buscar deportista por el dorsal
SELECT * FROM deportista WHERE epsdeportista = "sanitas";

#----- Modificaciones -----
-- Modificar nombre por nombre
UPDATE usuario SET nombreUs = "Catalina Carroño" WHERE idusuario = 5;
-- Modificar posición del deportista
UPDATE deportista SET posiciondeportista = "Auxiliar" WHERE iddeportista = 39;
-- Modificar eps del deportista
UPDATE deportista SET epsdeportista = "salud total" WHERE iddeportista = 11;
-- Modificar mensualidad
UPDATE deportista SET mensualidad = 9000 WHERE iddeportista = 9;
-- Modificar dorsal
UPDATE deportista SET dorsaldeportista = 79 WHERE iddeportista = 40;
-- Modificar dirección
UPDATE deportista SET direcciondeportista = "Calle 23 #45-78" WHERE iddeportista = 2;
-- Modificar edad
UPDATE deportista SET edaddeportista = 28 WHERE iddeportista = 4;
-- Modificar nombre del entrenador
UPDATE usuario SET nombreUs = "Juan Lopez" WHERE idusuario = (SELECT idusuarioFK FROM entrenador WHERE identrenador = 3);
-- Modificar eps del entrenador
UPDATE entrenador SET epsentrenador = "salud total" WHERE identrenador = 1;
-- Modificar salario
UPDATE entrenador SET salariodentrenador = 18000 WHERE identrenador = 3;
-- Modificar nombre del administrador
UPDATE usuario SET nombreUs = "Kevin Daniel Velasquez" WHERE idusuario = (SELECT idusuarioFK FROM administrador WHERE idadministrador = 1);
-- Modificar ID del administrador (no recomendado)
UPDATE administrador SET idadministrador = 1 WHERE idadministrador = 2;
-- Modificar nombre del acudiente
UPDATE deportista SET nombreacudiente = "Gonzalo Mosquera" WHERE iddeportista = 40;
-- Modificar teléfono del acudiente

#----- Consultas multitabla -----
-- Consultar jugadores que están inscritos a un evento
SELECT
  u.nombreUs AS "Nombre deportista",
  e.tipoEvento AS "Tipo de evento",
  e.descripcion AS "Descripción",
  h.fecha AS "Fecha",
  h.hora AS "Hora",
  h.lugar AS "Lugar",
  FROM deportista d
  JOIN usuario u ON d.idusuarioFK = u.idusuario
  JOIN evento e ON e.idusuarioFK = u.idusuario
  JOIN horario h ON e.idusuarioFK = h.idusuario;

#----- Consultas multitabla -----
-- Consultar jugadores que están inscritos a un evento
SELECT
  u.nombreUs AS "Nombre deportista",
  e.tipoEvento AS "Tipo de evento",
  e.descripcion AS "Descripción",
  h.fecha AS "Fecha",
  h.hora AS "Hora",
  h.lugar AS "Lugar",
  FROM deportista d
  JOIN usuario u ON d.idusuarioFK = u.idusuario
  JOIN evento e ON e.idusuarioFK = u.idusuario
  JOIN horario h ON e.idusuarioFK = h.idusuario;

```

```

#Entrenadores cuyo salario está por encima del promedio general
SELECT
  u.nombreUs AS "Nombre Entrenador",
  e.salarioEntrenador AS "Salario"
FROM entrenador e
JOIN usuario u ON e.idusuarioFK = u.idusuario
WHERE e.salarioEntrenador > (
  SELECT AVG(salarioEntrenador) FROM entrenador
);

#Deportistas cuyo acudiente tiene correo registrado en Gmail
SELECT
  u.nombreUs AS "Nombre Deportista",
  d.correoAcudiente AS "Correo Acudiente"

```

Nombre Entrenador	Salario
Juliana Ramírez Soto	2200000
Juan Lopez	3500000

```

# Consultas vistas
-- Vista 1: Usuarios registrados
CREATE VIEW vista_usuarios_generales AS
SELECT idusuario, tipoUsuario, nombreUs
FROM usuario;

select * from vista_usuarios_generales;

```

idUsuario	tipoUsuario	nombreUs
1	Administrador	Kevin Daniel Velasquez
2	Entrenador	Carlos Andrés Mendoza
3	Entrenador	Juliana Ramírez Soto
4	Entrenador	Oscar Giovanni López
5	Entrenador	Juan Lopez
6	Deportista	Karolina Dolores Mederos Silveira
7	Deportista	Elizabeth Garcia Arias
8	Deportista	Luciana Dussan Molina

```

/*Procedimiento para registrar un usuario*/
DELIMITER //
create procedure RegistrarUsuario(in idusuario int, in tipousuario varchar(40), nombreusu varchar(50))
BEGIN
  insert into usuario (idusuario, tipousuario, nombreusu)
  values(idusuario, tipousuario, nombreusu);
END //
DELIMITER ;

call RegistrarUsuario("Deportista", "Alberto Gamero");

```

Imagen 10 Consultas multitabla creadas en MySQL

7. Arquitectura de base de datos No relacional



Imagen 11 Diseño de arquitectura de la base de datos

8. Mapa de Componentes de Base de datos NoSQL.

1. Usuario

Descripción:

Contiene los datos básicos de todas las personas registradas en el sistema.

Atributo:

- **idUsuario:** Identificador único.
- **tipoUsuario:** Rol del usuario (administrador, entrenador, deportista).
- **nombreUsu:** Nombre completo del usuario.
- **correo:** Correo electrónico.
- **clave:** Contraseña cifrada.

2. Deportista

Descripción:

Almacena los datos específicos de los jugadores del club.

Relación:

Referencia al documento en la colección Usuario.

Atributos:

- **idDeportista:** Identificador del deportista.
- **posicion:** Posición dentro del campo.
- **categoria:** Nombre o ID de la categoría.
- **dorsal:** Número de camiseta.
- **direccion:** Dirección de residencia.
- **edad:** Edad del jugador.
- **estado:** Activo o inactivo.
- **acudiente:** Subdocumento (nombre, teléfono, correo).
- **eps:** EPS del deportista.

3. Entrenador

Descripción:

Contiene información del personal técnico del club.

Relación:

Referencia al documento en la colección Usuario.

- **idEntrenador:** Identificador del entrenador.

- **categoriaAsignada:** Categoría a la que está asignado.
- **eps:** EPS del entrenador.
- **salario:** Ingreso mensual.

4. Categoría

Descripción:

Agrupar a los deportistas por edad o nivel competitivo.

Atributos:

- **idCategoria:** Identificador único.
- **nombreCategoria:** Nombre de la categoría (Ej: Infantil, Juvenil, Mayores)

5. Evento

Descripción:

Registra las actividades del club, como entrenamientos y torneos.

Relación:

Referencia al Usuario organizador y al Horario.

Atributos:

- **idEvento:** Identificador del evento.
- **tipoEvento:** Entrenamiento, partido, reunión, etc.
- **descripcion:** Detalles del evento.
- **horario:** Subdocumento o referencia (fecha, hora, lugar).

6. Horario

Descripción:

Define la fecha, hora y lugar de los eventos del club..

Atributos:

- **idHorario:** Identificador del horario.
- **fecha:** Fecha del evento.
- **hora:** Hora del evento.
- **lugar:** Lugar del evento.

9. Script mongoshell: script mongo shell

<https://docs.google.com/document/d/1ztY64VKcTEs6RRAh044flx9LmQ-rM7oQAo0UQdF5HHU/edit?usp=sharing>

PLANEACIÓN:

a) Modelo de Datos

Por parte del modelo de datos, las entidades se presentan como usuarios que ingresan a la base de datos mediante un

ID y contraseña.

La base de datos se realiza en MongoDB con integración en Node.js, permitiendo la inserción de registros correspondientes a cada categoría dentro del club Lynx tales como deportistas, entrenadores, eventos y horarios.

Las colecciones sugeridas para estructurar el modelo son:

- Usuarios: contiene información básica del administrador y posibles otros usuarios.
- Deportistas: incluye nombre, edad, categoría y disciplina.
- Entrenadores: almacena el nombre y especialidad del entrenador.
- Eventos: registra torneos o actividades con fecha, lugar y participantes.
- Horarios: define día, hora y lugar de entrenamientos por disciplina

b) Distribución y Escalabilidad

Los datos estarán organizados en un servidor gratuito de Mongo Atlas, el cual, permitiendo la conexión desde una ID local del dispositivo autorizado, se podrá manejar la información mediante Visual Studio Code.

Todo el sistema se gestiona en un solo servidor, sin mecanismos de escalado automático ni distribución por clústeres, ya que la solución está diseñada dentro de los límites del entorno gratuito.

c) Índices y Rendimiento

Los índices de rendimiento para nuestra base de datos son dependientes del servidor.

Al ser gratuito, se tiene una restricción de 512MB de almacenamiento máximo, aunque se estima que los datos del club no superarán este límite.

La latencia del servidor oscila entre 90ms y 100ms, lo cual proporciona un tiempo de respuesta aceptable para nuestras operaciones básicas.

El uso de GPU y memoria está restringido por los límites del plan gratuito, y se ve influenciado por la carga que maneja el servidor compartido.

d) Consistencia y Disponibilidad

Las gestiones en la base de datos deben ser realizadas desde la interfaz web de MongoDB Atlas, desde donde el administrador puede ejecutar los cambios necesarios.

MongoDB ofrece consistencia eventual y garantiza integridad a nivel de documento.

No se cuenta con réplicas ni balanceo de carga, ya que se trabaja con un entorno limitado, pero suficiente para la escala del proyecto del club.

e) Seguridad y Acceso

La seguridad y el acceso están restringidos únicamente para el administrador, quien tendrá acceso a todos los datos del club, incluyendo deportistas, entrenadores, eventos, horarios, entre otros.

La autenticación se realiza mediante usuario y contraseña, y la conexión al servidor está protegida mediante control de IP autorizadas y conexión cifrada (SSL) provista por MongoDB Atlas.

Las credenciales deben mantenerse seguras mediante técnicas como hash de contraseñas en el backend de Node.js.

CONCLUSIONES

1. En conclusión, la implementación de una base de datos relacional y no relacional permitió solucionar el problema principal del Club de Voleibol Lynx: El desorden y la falta de control en el registro y consulta de información. Esto gracias al uso de tecnologías como MySQL y MongoDB, logrando de forma significativa mejorar la organización y el acceso a los datos de deportistas, entrenadores, horarios, mensualidades y torneos dentro del club.
2. En conclusión, la realización del proyecto permitió analizar a fondo las necesidades reales del Club Lynx entendiendo los desafíos generados por el cambio de administración y la recolección rudimentaria de datos. Además, a partir de entrevistas, revisión de procesos y definición de stakeholders, se logró delimitar el problema, definir objetivos claros y proponer soluciones viables, en donde este análisis inicial fue clave para estructurar correctamente la información actual y sentar las bases para un sistema organizado..
3. En conclusión, se desarrolló un diseño completo del sistema de datos, integrando modelos como el MER, el modelo lógico, físico y el modelo relacional, así como diagramas de clases y flujos de procesos. También, se establecieron claramente los requisitos funcionales y no funcionales, cubriendo las necesidades operativas del club, en donde, esta etapa de diseño garantizó que el sistema respondiera a los distintos tipos de usuarios, sus roles y necesidades dentro de la estructura del club.
4. En conclusión, la implementación de las bases de datos en SQL y NoSQL fue exitosa, utilizando herramientas modernas como MySQL Workbench, MongoDB Atlas, Node.js y Visual Studio Code. Se realizaron múltiples pruebas que confirmaron la correcta inserción, consulta y organización de los datos, garantizando la funcionalidad del sistema en condiciones reales. Esta fase permitió validar que las estructuras diseñadas cumplieran con los requerimientos establecidos, dando cierre al proyecto y demostrando que el sistema puede llegar a escalar en el futuro.

REFERENCIAS

1. M. P. Barrett, "Node.js: A Platform for Building Scalable Applications," *IEEE Internet Computing*, vol. 22, no. 5, pp. 15-23, Sep. 2018. [Online]. Available: <https://ieeexplore.ieee.org/document/8543038>.
2. MongoDB, Inc., "MongoDB: The Database for Modern Applications," *MongoDB Documentation*, 2020. [Online]. Available: <https://www.mongodb.com/docs/>

ENLACE GitHub:

<https://github.com/Alejom-hub/Proyecto-Ingenieria-de-Datos>

ENLACE Tablero Kanban:

<https://trello.com/b/AcTfYEih/backlog-general>

ENLACE Repositorio Node:

https://github.com/simon1655/basededatosnode/tree/25308fff089b2c2bb1247d083f5c67e29e36b5c4/BD_MongoDB

ANEXOS

I. Evaluación de soluciones propuestas y selección de opción definitiva

Al momento de evaluar las alternativas de solución para el Club de Voleibol Lynx se optó por desarrollar un sistema propio utilizando Node.js y MongoDB, pues estos permiten gestionar de forma clara y segura la información de jugadores, entrenadores, horarios y torneos. Además, esta solución responde eficientemente a los requisitos funcionales definidos: garantiza el acceso controlado a datos personales y médicos y tiene un impacto positivo en términos de salud, seguridad, sostenibilidad ambiental y eficiencia económica, al reducir el uso de papel y centralizar toda la gestión del club en una sola plataforma digital fácil de utilizar para cualquier persona.

II. Evidencias de Gerencia, Evidencias de Diseño, Evidencias de Documentación y Evidencias de Experiencia de usuario e Interfaz gráfica

Estas evidencias se pueden consultar en el siguiente link:

https://docs.google.com/document/d/13cV0kfZODlnJRwwm_vhjrUG_MYdY4lt-Ddg2U_nrGks/edit?tab=t.0