



DISEÑO **BASE DE DATOS**

GRUPO #1

INTEGRANTES

Efrain Edwin Baret Garcia 2024-0366

Harold Rafael Dominguez De Leon 2024-0354

Isaac Rafael Fermin Llauger 2024-0397

Lisbeth Arlette Jimenez Cambero 2024-0367

Marcos Eliezer Mendez Montero 2024-0361



INTRODUCCIÓN

Este proyecto busca desarrollar una base de datos para el equipo de baloncesto Golden State Warriors, con el objetivo de centralizar y gestionar de manera eficiente la información clave del equipo. Esto permitirá al personal técnico y administrativo tomar decisiones basadas en datos, optimizando tanto el rendimiento del equipo como la administración de recursos.

ANÁLISIS DE REQUISITO



Usuarios de la base de datos

Entrenadores
Analistas de rendimiento
Directivos y administrativo
Preparadores físicos



Requerimientos funcionales

Gestión de jugadores
Estadísticas de partidos
Historial de lesiones
Rendimiento de equipo
Generación de reportes



Requerimientos no funcionales

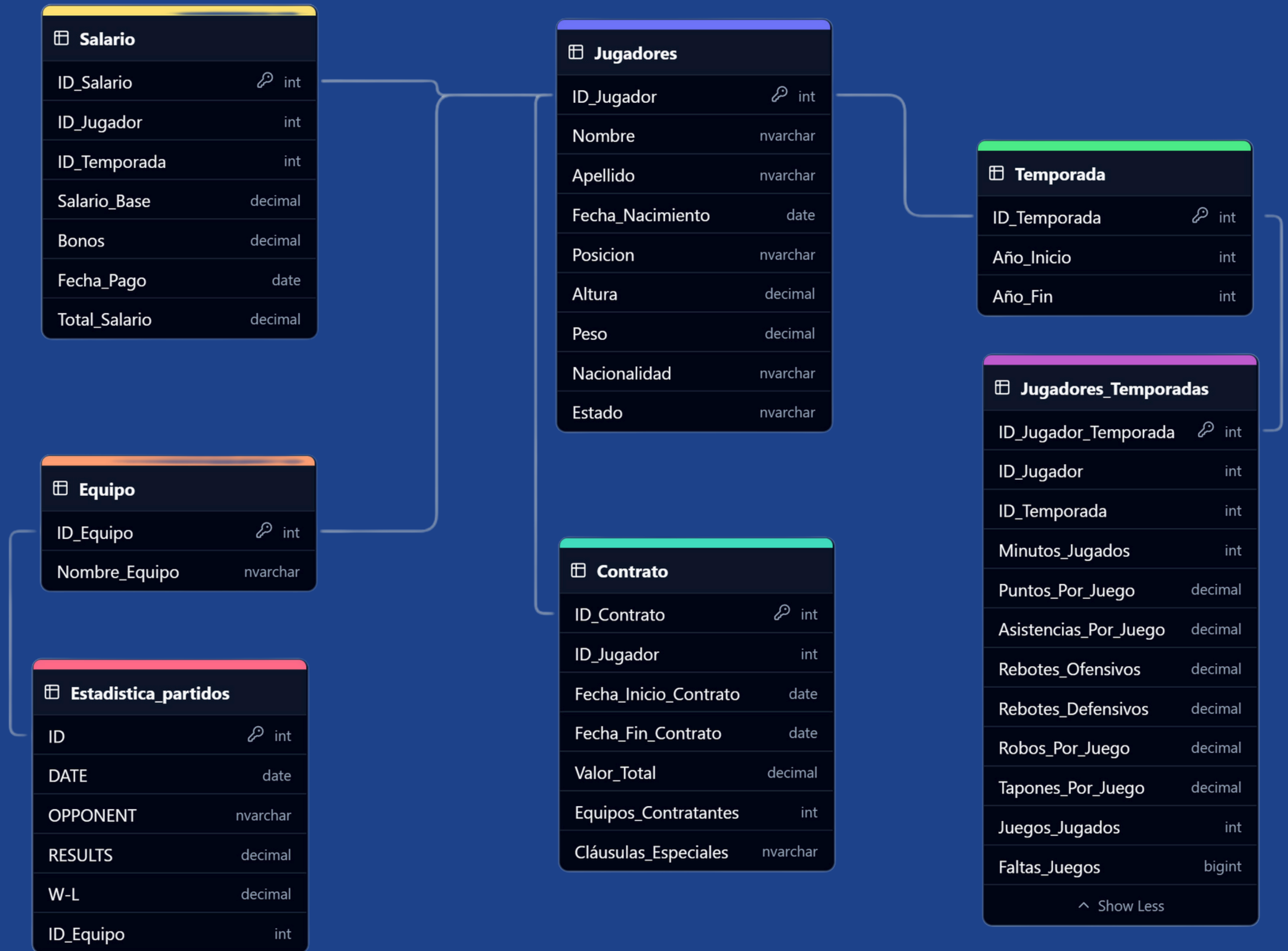
Rendimiento
Seguridad
Escalabilidad
Disponibilidad



Otros requerimientos

Gestos de base de datos
Plataforma
Backups regulares
Actualización de datos

DIAGRAMA ER



TIPOS DE DATOS

| Jugadores | |
|------------------|----------|
| ID_Jugador | int |
| Nombre | nvarchar |
| Apellido | nvarchar |
| Fecha_Nacimiento | date |
| Posicion | nvarchar |
| Altura | decimal |
| Peso | decimal |
| Nacionalidad | nvarchar |
| Estado | nvarchar |

| Temporada | |
|--------------|-----|
| ID_Temporada | int |
| Año_Inicio | int |
| Año_Fin | int |

| Equipo | |
|---------------|----------|
| ID_Equipo | int |
| Nombre_Equipo | nvarchar |

TIPOS DE DATOS

Estadistica_partidos


| | |
|-----------|----------|
| ID | int |
| DATE | date |
| OPPONENT | nvarchar |
| RESULTS | decimal |
| W-L | decimal |
| ID_Equipo | int |


Jugadores_Temporadas

| | |
|-----------------------|---------|
| ID_Jugador_Temporada | int |
| ID_Jugador | int |
| ID_Temporada | int |
| Minutos_Jugados | int |
| Puntos_Por_Juego | decimal |
| Asistencias_Por_Juego | decimal |
| Rebotes_Ofensivos | decimal |
| Rebotes_Defensivos | decimal |
| Robos_Por_Juego | decimal |
| Tapones_Por_Juego | decimal |
| Juegos_Jugados | int |
| Faltas_Juegos | bigint |

^ Show Less

TIPOS DE DATOS

| Contrato | | |
|-----------------------|---|----------|
| ID_Contrato |  | int |
| ID_Jugador | | int |
| Fecha_Inicio_Contrato | | date |
| Fecha_Fin_Contrato | | date |
| Valor_Total | | decimal |
| Equipos_Contratantes | | int |
| Cláusulas_Especiales | | nvarchar |

| Salario | | |
|---------------|---|---------|
| ID_Salario |  | int |
| ID_Jugador | | int |
| ID_Temporada | | int |
| Salario_Base | | decimal |
| Bonos | | decimal |
| Fecha_Pago | | date |
| Total_Salario | | decimal |

MODELO FÍSICO

El modelo físico de la base de datos proporciona una representación detallada de cómo se almacenan y gestionan los datos en el sistema, incluyendo el diseño de las tablas, la estructura de los datos y las optimizaciones realizadas para garantizar un rendimiento eficiente



El modelo físico de la base de datos incluye:

*Esquema de la Base de Datos
Scripts SQL*

Decisiones de Almacenamiento y Optimización

Golden_StateDB

- Database Diagrams
- Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.Contrato (1)
 - Columns
 - ID_Contrato (PK, int, not null)
 - ID_Jugador (FK, int, null)
 - Inicio_Contrato (date, null)
 - Fin_Contrato (date, null)
 - Valor_Total (decimal(15,2), null)
 - Equipo_Contratantes (FK, int, not null)
 - Cláusulas_Especiales (varchar(50), null)
 - Keys
 - Constraints
 - Triggers
 - Indexes
 - Statistics
 - dbo.Equipo (1)
 - Columns
 - ID_Equipo (PK, int, not null)
 - Nombre_Equipo (varchar(50), null)
 - Keys
 - Constraints
 - Triggers
 - Indexes
 - Statistics
 - dbo.Estadistica_partidos
 - Columns
 - Keys
 - Constraints
 - Triggers
 - Indexes
 - Statistics
 - dbo.Salario (3)
 - Columns
 - ID_Salario (PK, int, not null)
 - ID_Jugador (FK, int, null)
 - ID_Temporada (FK, int, null)
 - Salario (decimal(10,2), null)
 - Bonos (decimal(10,2), null)

```

SET ID = 31944
WHERE ID = 31965 AND OPPONENT = 'Utah'

ALTER TABLE Estadistica_partidos
ADD CONSTRAINT PK_ID_Estadistica PRIMARY KEY (ID)

select * from [Equipo (1)]

UPDATE Estadistica_partidos
SET ID_Equipo = 1569
where ID_Equipo IS NULL

ALTER TABLE Estadistica_partidos
ALTER COLUMN ID_Equipo int not null

UPDATE [Contrato (1)]
SET Equipo_Contratantes = 1569
WHERE Equipo_Contratantes = 1

ALTER TABLE [Contrato (1)]
ALTER COLUMN Equipo_Contratantes int not null

ALTER TABLE [Contrato (1)]
ADD CONSTRAINT FK_Equipo_Contratantes FOREIGN KEY (Equipo_Contratantes)
REFERENCES [Equipo (1)] (ID_Equipo)

ALTER TABLE Estadistica_partidos
ADD CONSTRAINT FK_ID_Equipo_Equipo FOREIGN KEY (ID_Equipo)
REFERENCES [Equipo (1)] (ID_Equipo)

UPDATE [Contrato (1)]
SET Inicio_Contrato = Inicio_Contrato + '-1'
where LEN(Inicio_Contrato) = 7

UPDATE [Contrato (1)]
SET Fin_Contrato = Fin_Contrato + '-1'
where LEN(Fin_Contrato) = 7

ALTER TABLE [Contrato (1)]
ALTER COLUMN Fin_Contrato DATE

select * from [Contrato (1)]

```

83 %

Messages

Commands completed successfully.

Completion time: 2024-10-22T11:38:20.9632608-04:00

83 %

Query executed successfully

BACKUP Y RECUPERACIÓN



BACKUPS DE LOGS DE TRANSACCIONES
FRECUENCIA: DESPUÉS DE CADA PARTIDO

DESCRIPCION: RESPALDA LAS
TRANSACCIONES QUE HAN OCURRIDO
DESDE EL ÚLTIMO BACKUP DE LOG.

RETENCION: MANTÉN LOS ÚLTIMOS 7
DÍAS DE LOGS.

BACKUP Y RECUPERACIÓN



BACKUPS COMPLETOS (FULL BACKUP)

FRECUENCIA: UNA VEZ POR SEMANA

DESCRIPCIÓN: CAPTURA TODA LA BASE DE DATOS EN UN SOLO ARCHIVO DE RESPALDO.

RETENCIÓN: MANTÉN LOS ÚLTIMOS 4 BACKUPS COMPLETOS

BACKUP Y RECUPERACIÓN



ALMACENAMIENTO DE BACKUPS

UBICACIÓN PRIMARIA: ALMACENA LOS BACKUPS EN UN SERVIDOR PRIVADO DEL EQUIPO O UNIDAD DE RED LOCAL CON ESPACIO DEDICADO.

UBICACIÓN SECUNDARIA: USA ALMACENAMIENTO EN LA NUBE O UN SERVIDOR EN UNA UBICACIÓN FÍSICA DISTINTA PARA UNA COPIA SECUNDARIA EN CASO DE DESASTRE.

PASOS PARA RECUPERACIÓN EN CASO DE FALLOS

El plan para manejar problemas en la base de datos se divide en cuatro pasos clave para garantizar que los datos se mantengan seguros y completos:

1

**Identificación del
Problema**

2

**Proceso de
Recuperación**

3

**Pruebas y
Verificaciones**

4

**Documentación y
Reporte**

IMPLEMENTACION

La implementación técnica de la base de datos en SQL fue un proceso estructurado que incluyó la creación de tablas, la inserción de datos, la configuración de la seguridad y la realización de pruebas para garantizar su funcionalidad y eficiencia



CONCLUSIÓN

La creación de una base de datos para el equipo representa un paso fundamental hacia la optimización de su gestión y rendimiento. Este sistema no solo permitirá almacenar y organizar información crucial sobre jugadores, partidos y estadísticas, sino que también proporcionará herramientas analíticas que faciliten la toma de decisiones estratégicas. Al centralizar los datos, se logrará un acceso más ágil y eficiente, lo que es vital para los entrenadores y analistas que buscan mejorar el rendimiento del equipo.





¡GRACIAS

**POR SU
ATENCIÓN!**



Instituto Tecnológico de las Américas (ITLA)

Asignatura:

Base de datos I

Carrera:

Analítica y Ciencia de Datos

Integrantes:

Efrain Edwin Baret Garcia 2024-0366

Harold Rafael Dominguez De Leon 2024-0354

Isaac Rafael Fermin Llauger 2024-0397

Lisbeth Arlette Jimenez Cambero 2024-0367

Marcos Eliezer Mendez Montero 2024-0361

Maestro:

José Aquino

Tema:

Diseño y Creación de Base de Datos

RESUMEN EJECUTIVO

El presente proyecto tiene como objetivo la creación de una base de datos que centralice y optimice la gestión de información para el equipo de baloncesto **Golden State**. Esta base de datos permite almacenar y acceder de manera eficiente a datos clave relacionados con jugadores, partidos, estadísticas y otros elementos relevantes para el funcionamiento del equipo.

La base de datos facilitará la toma de decisiones estratégicas al ofrecer una estructura organizada y accesible para el análisis de rendimiento de los jugadores, la planificación de partidos y la creación de estrategias que permitan sacar lo mejor de cada jugador enfocándose en sus fortalezas y debilidades. Asimismo, el sistema permitirá el seguimiento histórico de estadísticas individuales y colectivas, lo que contribuirá al desarrollo de estrategias de juego más efectivas.

Entre las principales funcionalidades destacan:

- Registro y gestión de los perfiles de los jugadores.
- Almacenamiento y consulta de estadísticas de rendimiento por temporada y partido.
- Historial de partidos jugados, incluyendo resultados y métricas clave.

Otro aspecto importante de este proyecto es su capacidad de generar reportes personalizados en función de las necesidades del equipo. Los usuarios podrán obtener informes detallados sobre el rendimiento de los jugadores en una temporada específica, la evolución de un jugador a lo largo del tiempo, o el desempeño del equipo en una determinada competición. Estos reportes serán fundamentales para la toma de decisiones basadas en datos, permitiendo que los entrenadores y directores deportivos ajusten sus estrategias en función del análisis de las estadísticas disponibles.

Este sistema busca mejorar la eficiencia administrativa y deportiva, ofreciendo una solución robusta y escalable que podrá ser ampliada según las necesidades futuras del equipo.

Al centralizar la información en un único sistema de gestión de datos, el equipo podrá acceder a datos precisos y actualizados en cualquier momento, lo que facilitará tanto la planificación a corto plazo como el análisis estratégico a largo plazo. Con una estructura escalable y flexible, la base de datos está preparada para crecer junto con el equipo, adaptándose a sus necesidades futuras y ofreciendo un soporte continuo para mejorar su rendimiento y éxito.

INTRODUCCIÓN

Para este proyecto vamos a realizar un caso de uso, en el cual fuimos contactados por el equipo técnico y estadístico del **Golden State** Warriors, que es un reconocido equipo de básquetbol de la NBA, nos contactaron con la idea de un proyecto que consiste en la creación de una base de datos para el equipo de básquetbol. El objetivo es almacenar, gestionar y analizar información relevante del equipo, incluyendo datos de jugadores, partidos, estadísticas, entrenadores, y más.

Esta base de datos servirá como un recurso integral para el equipo técnico y administrativo, facilitando la toma de decisiones basada en datos.

El objetivo principal es desarrollar un sistema de base de datos que permita el almacenamiento, consulta y actualización eficiente de los datos relacionados con el equipo. Esto incluye el seguimiento de las estadísticas individuales de los jugadores, los resultados de los partidos, y los registros históricos del equipo, con el fin de optimizar el rendimiento del equipo y gestionar los recursos de manera efectiva.

El proyecto incluirá los siguientes módulos:

- Información de los jugadores (nombre, posición, estadísticas individuales, etc.).
- Registro de partidos (resultados, estadísticas del partido, rivales).
- Historial de campeonatos y trofeos.
- Consultas estadísticas y generación de reportes para el análisis.

La creación de esta base de datos es necesaria para centralizar toda la información relevante del equipo en un solo sistema. Esto viene a resolver un problema que tenía el departamento técnico del equipo a la hora de elaborar mediciones y reportes de desempeño ya que esta base de datos permitirá una gestión más eficiente de los datos, un acceso más rápido a la información crucial para la toma de decisiones, y la posibilidad de realizar análisis detallados para mejorar el rendimiento tanto de los jugadores como del equipo en general. Además, un sistema organizado y bien estructurado ayudará a mantener un historial completo y accesible, lo cual es fundamental para la planificación estratégica del equipo.

Este sistema también servirá para preservar el historial del equipo, facilitando el acceso a registros pasados que son importantes para la planificación estratégica futura. Al digitalizar y estructurar estos datos, se asegura que la información sea fácilmente accesible y mantenida a lo largo del tiempo, contribuyendo así a la eficiencia operativa y al éxito deportivo del equipo

ANÁLISIS DE REQUISITO

Los requisitos planteados por parte del equipo técnico de Golden States Warriors son los siguientes.

1-Usuarios de la Base de Datos.

La base de Datos será utilizada por diferentes personas cada una de las cuales tendrá diferentes niveles de acceso a la misma, los principales usuarios serán:

Entrenadores: Consultarán la base de datos para obtener información sobre el rendimiento de los jugadores y estadísticas de partidos.

Analistas de rendimiento: Realizarán consultas avanzadas sobre las estadísticas de jugadores y equipo, generando reportes para la toma de decisiones.

Directivos y administrativos: Tendrán acceso a reportes sobre la salud financiera y rendimiento global del equipo, basados en las métricas de rendimiento deportivo.

Preparadores físicos: Consultarán la base de datos para revisar el historial de lesiones y seguimiento del estado físico de los jugadores

Cada tipo de usuario tendrá un acceso específico según su rol, determinado por políticas de control de acceso, los usuarios mencionados son usuarios que no serán del departamento técnico, sino que usarán la base de datos para ver reporte del equipo.

2-Requerimientos Funcionales

La base de datos debe de cumplir con una serie de requerimientos funcionales que ayuden a la gestión de los datos del equipo y poder satisfacer las necesidades de los usuarios del sistema.

Gestión de jugadores: La base de datos debe permitir registrar, modificar y eliminar información sobre jugadores, incluyendo su nombre, posición, estatura, peso, y estado (activo o inactivo).

Estadísticas de partidos: Se deben almacenar estadísticas detalladas de cada partido, como puntos, asistencias, rebotes, y minutos jugados por cada jugador.

Historial de lesiones: El sistema debe permitir registrar y consultar información sobre lesiones de los jugadores, así como las fechas estimadas de recuperación.

Rendimiento de equipo: Los usuarios deben poder consultar el rendimiento general del equipo, incluyendo partidos ganados, perdidos, puntos acumulados, entre otros.

Generación de reportes: Se deben generar reportes de rendimiento individual de los jugadores y reportes acumulados de las estadísticas del equipo por temporada.

3. Requerimientos No Funcionales

Además de las funcionalidades básicas, la base de datos debe cumplir con ciertos criterios de rendimiento y seguridad:

Rendimiento: La base de datos debe ser capaz de manejar consultas rápidas sobre grandes volúmenes de datos, especialmente cuando se consulten estadísticas acumuladas por temporadas completas.

Seguridad: Se implementará un control de acceso basado en roles, asegurando que solo los usuarios autorizados puedan acceder o modificar la información.

Escalabilidad: La base de datos debe ser capaz de soportar la inclusión de nuevas temporadas, jugadores y estadísticas sin comprometer el rendimiento.

Disponibilidad: El sistema debe garantizar un alto nivel de disponibilidad, minimizando el tiempo de inactividad durante las operaciones de mantenimiento o actualizaciones

4-Otros requerimientos

Gestor de base de datos: como gestor usaremos Microsoft SQL Server debido a su capacidad de manejo de grandes volúmenes de datos, consultas complejas y relativamente fácil implementación en comparación con otros Gestores.

Plataforma: Después de creada la base de datos, estará a disponibilidad del equipo si alojarla en servidores propios por un tema de seguridad o alojarla en servidores en la nube.

Backups regulares: Se implementarán copias de seguridad diarias automáticas para evitar la pérdida de información en caso de fallos.

Actualización de datos: Se establecerán procedimientos para la actualización de la base de datos al final de cada temporada, asegurando que se puedan agregar nuevos jugadores, temporadas, y partidos.

MODELO CONCEPTUAL

Entidades:

- **Jugador:** Representa a un jugador de baloncesto.
- **Temporada:** Define un período de juego específico.
- **Contrato:** Detalla los términos del acuerdo entre un jugador y un equipo.
- **Equipo:** Representa a una organización de baloncesto.
- **Jugadores_Temporada:** Relaciona a un jugador con una temporada específica, permitiendo el seguimiento de su desempeño a lo largo del tiempo.
- **Salario:** Registra el salario de un jugador en una temporada determinada.
- **Estadísticas_Jugador:** Contiene las estadísticas de un jugador en una temporada.
- **Estadísticas_partido:** contiene las estadísticas de los partidos en una temporada

Relaciones:

- **Jugador tiene una relación uno a muchos con:**
- **Jugadores_Temporada:** Un jugador puede participar en múltiples temporadas.

- **Salario:** Un jugador puede tener diferentes salarios a lo largo de su carrera.
- **Estadísticas_Jugador:** Un jugador acumula estadísticas en cada temporada.
- **Estadística_partido:** Los equipos acumulan estadística en cada partido jugado durante la temporada.
- **Contrato:** Un jugador puede tener múltiples contratos a lo largo de su carrera.
- **Temporada tiene una relación uno a muchos con:**
- **Jugadores_Temporada:** Una temporada involucra a múltiples jugadores.
- **Salario:** En cada temporada se registran los salarios de los jugadores.
- **Estadísticas_Jugador:** Las estadísticas se registran por temporada.
- **Contrato tiene una relación uno a muchos con Equipo:**
- Un equipo puede tener múltiples contratos con diferentes jugadores, pero un contrato pertenece a un solo equipo.

DIAGRAMA ER

Entidades:

1. Jugador:

Atributos:

ID_Jugador (PK)

Nombre

Apellido

Fecha_Nacimiento

Posición

Altura

Peso

Nacionalidad

Estado

2. Temporada:

Atributos:

ID_Temporada (PK)

Año_Inicio

Año_Fin

3. Jugadores_Temporada (Relación):

Atributos:

ID_Jugador_Temporada (PK)

ID_Jugador (FK) (de Jugador)

ID_Temporada (FK) (de Temporada)

Juegos_Jugados

Minutos_Jugados

Puntos_Por_Juego

Rebotes_Ofensivos

Rebotes_Defensivos

Asistencias_Por_Juego

Robos_Por_Juego

Tapones_Por_Juego

Perdidas_por_partido

Faltas_por_partido

4. Salario:

Atributos:

ID_Salario (PK)

ID_Jugador (FK) (de Jugador)

ID_Temporada (FK) (de Temporada)

Salario_Base

Bonos

Fecha_Pago

Total_Salario

5. Contrato:

Atributos:

ID_Contrato (PK)

ID_Jugador (FK) (de Jugador)

Fecha_Inicio_Contrato

Fecha_Fin_Contrato

Valor_Total

Equipos_Contratantes (FK hacia Equipo)

Cláusulas_Especiales

6. Equipo (Entidad externa relacionada con contratos):

Atributos:

ID_Equipo (PK)

Nombre_Equipo

7-Estadistica_partido:

Id(pk)

Date

Opponent

Result

W-L

Id_equipo (fk)

Relaciones:

Jugador y Temporada tienen una relación muchos a muchos a través de la entidad Jugadores_Temporada.

Jugador y Temporada están relacionados a través de la tabla Salario, ya que los jugadores tienen diferentes salarios por temporada.

Jugador y Equipo están relacionados por la tabla Contratos, donde un jugador puede tener uno o más contratos con diferentes equipos.

Temporada y partido: Tienen una relación de muchos a muchos, ya que puede haber varias temporadas con varios partidos.

Tipos de datos de cada tabla

Tabla: Jugadores

ID_Jugador: INT (PK) — Un número entero para identificar al jugador.

Nombre: NVARCHAR (50) o VARCHAR (50) — Texto para almacenar el nombre del jugador (longitud de hasta 50 caracteres).

Apellido: NVARCHAR (50) o VARCHAR (50) — Texto para almacenar el apellido del jugador.

Fecha Nacimiento: DATE — Para almacenar la fecha de nacimiento.

Posición: NVARCHAR (20) o VARCHAR (20) — Texto que indica la posición del jugador (Ej.: "Portero", "Delantero").

Altura: DECIMAL (5,2) — Altura en pies, con 2 decimales (Ej.: 1.85).

Peso: DECIMAL (5,2) — Peso en libras, con 2 decimales (Ej.: 75.50).

Nacionalidad: NVARCHAR (50) o VARCHAR (50) — Texto que indica la nacionalidad del jugador.

Estado: NVARCHAR (15) o VARCHAR (15) — Texto para indicar si el jugador está activo, lesionado, suspendido o retirado.

Tabla: Temporadas

ID_Temporada: INT (PK) — Un número entero para identificar la temporada.

Año_Inicio: YEAR o INT (4) — Año de inicio de la temporada.

Año_Fin: YEAR o INT (4) — Año de finalización de la temporada.

Tabla: Jugadores_Temporada

ID_Jugador_Temporada: INT (PK) — Identificador único para la relación entre jugador y temporada.

ID_Jugador: INT (FK) — Referencia al ID del jugador.

ID_Temporada: INT (FK) — Referencia al ID de la temporada.

Juegos_Jugados: INT — Número de juegos jugados en la temporada.

Minutos_Jugados: INT — Minutos jugados en la temporada.

Puntos_Por_Juego: DECIMAL (5,2) — Promedio de puntos por juego.

Rebotes_Ofensivo: DECIMAL (5,2) — Promedio de rebotes ofensivo por juego.

Rebotes_Defensivos: DECIMAL (5,2) — Promedio de rebotes defensivos por juego.

Asistencias_Por_Juego: DECIMAL (5,2) — Promedio de asistencias por juego.

Robos_Por_Juego: DECIMAL (5,2) — Promedio de robos por juego.

Tapones_Por_Juego: DECIMAL (5,2) — Promedio de tapones por juego.

Perdidas_por_partido: DECIMAL (5,2) — Promedio de pérdidas de balón por partido

Faltas_por_partido: DECIMAL (5,2) — Promedio de faltas por partido

Tabla: Salarios

ID_Salario: INT (PK) — Identificador único del salario.

ID_Jugador: INT (FK) — Referencia al ID del jugador.

ID_Temporada: INT (FK) — Referencia al ID de la temporada.

Salario_Base: DECIMAL (10,2) — Salario base del jugador en esa temporada.

Bonos: DECIMAL (10,2) — Bonos adicionales del jugador.

Fecha_Pago: DATE — Fecha en la que se realiza el pago.

Total_Salario: DECIMAL (12,2) — Total del salario, incluyendo bonos.

Tabla: Contratos

ID_Contrato: INT (PK) — Identificador único del contrato.

ID_Jugador: INT (FK) — Referencia al ID del jugador.

Fecha_Inicio_Contrato: DATE — Fecha de inicio del contrato.

Fecha_Fin_Contrato: DATE — Fecha de fin del contrato.

Valor_Total: DECIMAL (15,2) — Valor total del contrato.

Equipos_Contratantes: INT (FK hacia la tabla Equipos) — Identificador del equipo contratante.

Cláusulas_Especiales: TEXT o NVARCHAR (255) — Detalles sobre cláusulas especiales en el contrato, si las hay.

Tabla: Equipos (referida en Contratos)

ID_Equipo: INT (PK) — Identificador único del equipo.

Nombre_Equipo: NVARCHAR (100) o VARCHAR (100) — Nombre del equipo.

Tabla: Estadistica_partidos:

ID: int

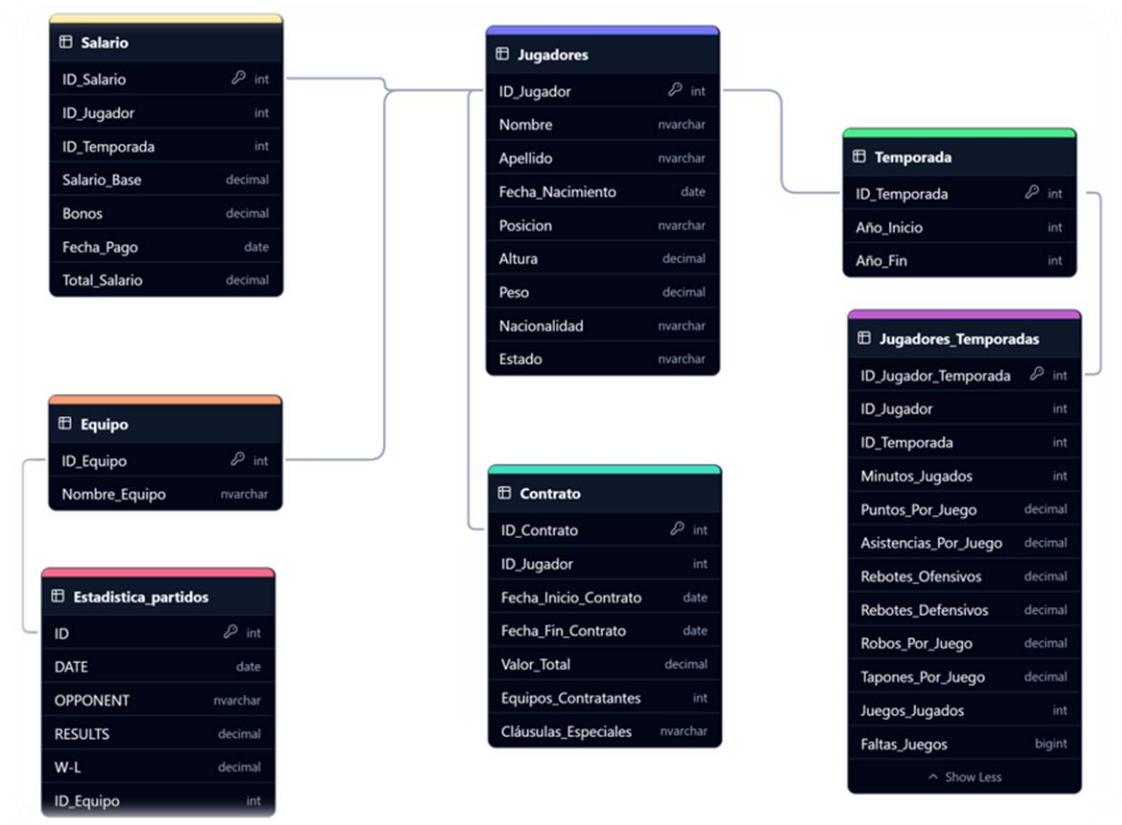
Date: Date

Opponent: Varchar (50)

Result: Varchar (50)

W-L: Varchar (50)

Id_equipo: int



MODELO FÍSICO

Esquema de la base de datos en SQL servers management studio (tablas, tipos de datos, restricciones, etc.).

Scripts SQL utilizados para crear las tablas, vistas, índices, y cualquier otro objeto de base de datos.

Explicación de las decisiones sobre el almacenamiento y optimización.

```
ALTER TABLE [Temporada (2)]
```

```
ALTER COLUMN ID_Temporada INT NOT NULL
```

```
ALTER TABLE [Temporada (2)]
```

```
ADD CONSTRAINT PK_Id_Temporada
```

Primary key (ID_Temporada)

```
select * from [Tabla Jugadores-Temporada (1)]
```

```
ALTER TABLE [Tabla Jugadores-Temporada (1)]
```

```
ALTER COLUMN ID_Jugador_Temporada INT NOT NULL
```

```
ALTER TABLE [Tabla Jugadores-Temporada (1)]
```

```
ADD CONSTRAINT PK_Id_Jugador_Temporada
```

Primary key (ID_Jugador_Temporada)

```
select * from [Salario (3)]
```

```
ALTER TABLE[Salario (3)]
```

```
ALTER COLUMN ID_Salario INT NOT NULL
```

```
ALTER TABLE [Salario (3)]
```

```
ADD CONSTRAINT PK_Id_Salario
```

```
Primary key (ID_Salario)
```

```
select * from [Equipo (1)]
```

```
ALTER TABLE[Equipo (1)]
```

```
ALTER COLUMN ID_Equipo INT NOT NULL
```

```
ALTER TABLE [Equipo (1)]
```

```
ADD CONSTRAINT PK_ID_Equipo
```

```
Primary key (ID_Equipo)
```

```
select * from [Tabla Jugadores (2)]
```

```
ALTER TABLE[Tabla Jugadores (2)]
```

```
ALTER COLUMN ID_Jugador INT NOT NULL
```

```
ALTER TABLE [Tabla Jugadores (2)]
```

```
ADD CONSTRAINT PK_ID_Jugador
```

```
Primary key (ID_Jugador )
```

---Claves foraneas--

--Jugadores temporada--

```
ALTER TABLE [Tabla Jugadores-Temporada (1)]
```

```
add constraint FK_ID_Jugador FOREIGN KEY (ID_Jugadores)
```

```
REFERENCES [Tabla Jugadores (2)] (ID_Jugador)
```

```
ALTER TABLE [Tabla Jugadores-Temporada (1)]
```

```
ALTER COLUMN ID_Jugadores INT
```

```
ALTER TABLE [Tabla Jugadores-Temporada (1)]
```

```
add constraint FK_ID_Temporada FOREIGN KEY (ID_Temporada)
```

```
REFERENCES [Temporada (2)] (ID_Temporada)
```

```
ALTER TABLE [Tabla Jugadores-Temporada (1)]
```

```
ALTER COLUMN Id_Temporada INT
```

```
exec sp_rename '[Tabla Jugadores-Temporada  
(1)].ID_Temperada','ID_Temporada','COLUMN'
```

```
select * from [Tabla Jugadores-Temporada (1)]
```

--salario--

```
ALTER TABLE [Salario (3)]
```

```
ALTER COLUMN ID_Jugador INT
```

```
ALTER TABLE [Salario (3)]
```

```
ALTER COLUMN ID_Temporada INT
```



```
ALTER TABLE [Salario (3)]
```

```
ADD CONSTRAINT FK_ID_Jugador_Salario FOREIGN KEY (ID_Jugador)
```

```
REFERENCES [Tabla Jugadores (2)] (ID_Jugador)
```

```
ALTER TABLE [Salario (3)]
```

```
ADD CONSTRAINT FK_ID_Temporada_Salario FOREIGN KEY (ID_Temporada)
```

```
REFERENCES [Temporada (2)](ID_Temporada)
```

--Contrato--

```
ALTER TABLE [Contrato (1)]
```

```
ADD CONSTRAINT FK_ID_Jugador_Contrato FOREIGN KEY (ID_Jugador)
```

```
REFERENCES [Tabla Jugadores (2)] (ID_Jugador)
```

```
ALTER TABLE [Contrato (1)]
```

```
ALTER COLUMN ID_Jugador INT
```

```
select * from [Contrato (1)]
```

```
select * from [Equipo (1)]
```

```
ALTER TABLE [Contrato (1)]
```

```
ALTER COLUMN Valor_Total DECIMAL(15,2)
```

```
ALTER TABLE [Contrato (1)]
```

```
ALTER COLUMN Inicio_Contrato DATE
```

```
update [Contrato (1)]
```

```
set Valor_Total = REPLACE(Valor_Total,',','')
```

```
UPDATE Estadistica_partidos
```

SET ID = 31944

WHERE ID = 31965 AND OPPONENT = 'Utah'

ALTER TABLE Estadistica_partidos

ADD CONSTRAINT PK_ID_Estadistica PRIMARY KEY (ID)

select * from [Equipo (1)]

UPDATE Estadistica_partidos

SET ID_Equipo = 1569

where ID_Equipo IS NULL

ALTER TABLE Estadistica_partidos

ALTER COLUMN ID_Equipo int not null

UPDATE [Contrato (1)]

SET Equipo_Contratantes = 1569

WHERE Equipo_Contratantes = 1

ALTER TABLE [Contrato (1)]

ALTER COLUMN Equipo_Contratantes int not null

ALTER TABLE [Contrato (1)]

ADD CONSTRAINT FK_Equipo_Contratantes FOREIGN KEY (Equipo_Contratantes)

REFERENCES [Equipo (1)] (ID_Equipo)

ALTER TABLE Estadistica_partidos

ADD CONSTRAINT FK_ID_Equipo_Equipo FOREIGN KEY (ID_Equipo)

REFERENCES [Equipo (1)] (ID_Equipo)

```
UPDATE [Contrato (1)]  
SET Inicio_Contrato = Inicio_Contrato + '-1'  
where LEN(Inicio_Contrato) = 7
```

```
UPDATE [Contrato (1)]  
SET Fin_Contrato = Fin_Contrato + '-1'  
where LEN(Fin_Contrato) = 7
```

```
ALTER TABLE [Contrato (1)]  
ALTER COLUMN Fin_Contrato DATE
```

```
select * from [Contrato (1)]
```

```
select * from [Salario (3)]
```

```
ALTER TABLE [Salario (3)]  
ALTER COLUMN Total_salario DECIMAL(10,2)
```

```
UPDATE [Salario (3)]  
SET Total_salario = REPLACE(Total_salario,'$', '')
```

```
UPDATE [Salario (3)]  
SET Total_salario = REPLACE(Total_salario,',', '')
```

```
ALTER TABLE [Salario (3)]  
ALTER COLUMN Fecha_pago DATE
```

```
select * from [Tabla Jugadores (2)]
```

```
ALTER TABLE [Tabla Jugadores (2)]
```

```
ALTER COLUMN Altura DECIMAL(5,2)
```

```
UPDATE [Tabla Jugadores (2)]
```

```
SET Altura = REPLACE(Altura, ',', '.')
```

```
EXEC sp_rename '[Tabla Jugadores (2)].Altura','Altura_FT','COLUMN'
```

```
select * from [Tabla Jugadores-Temporada (1)]
```

```
ALTER TABLE [Tabla Jugadores-Temporada (1)]
```

```
ALTER COLUMN GP int
```

```
ALTER TABLE [Tabla Jugadores-Temporada (1)]
```

```
ALTER COLUMN MIN Decimal(5,2)
```

```
ALTER TABLE [Tabla Jugadores-Temporada (1)]
```

```
ALTER COLUMN PTS Decimal(5,2)
```

```
ALTER TABLE [Tabla Jugadores-Temporada (1)]
```

```
ALTER COLUMN [OR] Decimal(5,2)
```

```
ALTER TABLE [Tabla Jugadores-Temporada (1)]
```

```
ALTER COLUMN DR Decimal(5,2)
```

```
ALTER TABLE [Tabla Jugadores-Temporada (1)]
```

```
ALTER COLUMN AST Decimal(5,2)
```

```
ALTER TABLE [Tabla Jugadores-Temporada (1)]
```

```
ALTER COLUMN STL Decimal(5,2)
```

```
ALTER TABLE [Tabla Jugadores-Temporada (1)]
```

```
ALTER COLUMN [TO] Decimal(5,2)
```

```
ALTER TABLE [Tabla Jugadores-Temporada (1)]
```

```
ALTER COLUMN [PF] Decimal(5,2)
```

```
SELECT STL
```

```
FROM [Tabla Jugadores-Temporada (1)]
```

```
WHERE ISNUMERIC(STL)= 0
```

```
UPDATE [Tabla Jugadores-Temporada (1)]
```

```
SET STL = NULL
```

```
WHERE TRY_CONVERT(DECIMAL(10, 2), STL) = 0;
```

--AgregarJugadorConContrato--

```
CREATE PROCEDURE Agregar_Jugador_Con_Contrato
```

```
@Nombre NVARCHAR(50),
```

```
@Apellido NVARCHAR(50),
```

```
@FechaNacimiento DATE,
```

```
@Posicion NVARCHAR(20),
```

```
@Altura DECIMAL(5,2),
```

```
@PesoEnLBS DECIMAL(5,2),
```

```
@Nacionalidad NVARCHAR(50),
```

```
@Estado NVARCHAR(15),
```

```
@FechaInicioContrato DATE,
```

```
@FechaFinContrato DATE,
```

```
@ValorTotal DECIMAL(15,2),
```

```
@EquipoContratante INT
```

AS

BEGIN

-- Insertar el jugador

INSERT INTO [Tabla Jugadores]

(Nombre, Apellido, Fecha_de_Nacimiento, POS, Altura_FT, Peso_En_LBS, Nacionalidad, Estado)

VALUES

(@Nombre, @Apellido, @FechaNacimiento, @Posicion, @Altura, @PesoEnLBS, @Nacionalidad, @Estado);

-- Obtener el ID del jugador insertado

DECLARE @IdJugador INT = SCOPE_IDENTITY();

-- Insertar el contrato

INSERT INTO [Contrato]

(ID_Jugador, Inicio_Contrato, Fn_Contrato, Valor_Total, Equipo_Contratantes)

VALUES

(@IdJugador, @FechaInicioContrato, @FechaFinContrato, @ValorTotal, @EquipoContratante);

END

--ActualizarSalarioJugador--

```
CREATE PROCEDURE Actualizar_Salario_Jugador
@ID_Jugador INT,
@ID_Temporada INT,
@NuevoSalario DECIMAL(10,2),
@NuevosBonos DECIMAL(10,2)
AS
BEGIN
UPDATE [Salario]
SET Salario = @NuevoSalario,
Bonos = @NuevosBonos,
Total_Salario = @NuevoSalario + @NuevosBonos
WHERE ID_Jugador = @ID_Jugador
AND ID_Temporada = @ID_Temporada;
END
```

----RegistrarEstadisticasPartido-----

CREATE PROCEDURE RegistrarPartido

@ID INT,

@Fecha DATE,

@Oponente NVARCHAR(50),

@Resultado NVARCHAR(20),

@Win_Lose NVARCHAR(10),

@ID_Equipo INT

AS

BEGIN

INSERT INTO Estadistica_partidos

(ID, DATE, OPPONENT, RESULT, Win_Lose, ID_Equipo)

VALUES

(@ID, @Fecha, @Oponente, @Resultado, @Win_Lose, @ID_Equipo);

END

select * from Estadistica_partidos

---EliminarContratoJugador--

CREATE PROCEDURE EliminarContratoJugador

@ID_Contrato INT

AS

BEGIN

DELETE FROM Contrato WHERE ID_Contrato = @ID_Contrato;

END

select * from [Contrato (1)]

--Reporte de estadística de una temporada--

```
CREATE PROCEDURE ObtenerEstadisticasTemporada
@ID_Jugador INT,
@ID_Temporada INT
AS
BEGIN
-- Seleccionar las estadísticas de la tabla [Tabla Jugadores-Temporada]
SELECT *
FROM [Tabla Jugadores-Temporada]
WHERE ID_Jugadores = @ID_Jugador AND ID_Temporada = @ID_Temporada;
END
```

--Vistas--

```
CREATE VIEW Vista_JugadoresActivos AS
SELECT [Tabla Jugadores (2)].Nombre, [Tabla Jugadores (2)].Apellido, [Tabla Jugadores (2)].POS AS Posición, [Equipo (1)].Nombre_Equipo
FROM [Tabla Jugadores (2)]
INNER JOIN [Contrato (1)] ON [Tabla Jugadores (2)].ID_Jugador = [Contrato (1)].ID_Jugador
INNER JOIN [Equipo (1)] ON [Contrato (1)].Equipo_Contratantes = [Equipo (1)].ID_Equipo
WHERE [Tabla Jugadores (2)].Estado = 'Activo';
```

```
select * from [Contrato (1)]
```

```
select * from [Equipo (1)]
```

```
select * from [Tabla Jugadores (2)]
```

--Vista_EstadisticasTemporada--

```
CREATE VIEW Vista_EstadisticasTemporada AS

SELECT

[Tabla Jugadores (2)].Nombre,

[Tabla Jugadores (2)].Apellido,

[Tabla Jugadores-Temporada (1)].ID_Temporada AS Temporada,

[Tabla Jugadores-Temporada (1)].PTS AS Puntos_Temporada,

[Tabla Jugadores-Temporada (1)].AST AS Asistencias_Temporada,

[Tabla Jugadores-Temporada (1)].DR AS Rebotes_Temporada,

[Tabla Jugadores-Temporada (1)].STL AS Robos_Temporada

FROM [Tabla Jugadores-Temporada (1)]

INNER JOIN [Tabla Jugadores (2)]

ON [Tabla Jugadores-Temporada (1)].ID_Jugadores = [Tabla Jugadores (2)].ID_Jugador;
```

--Vista_SalariosPorTemporada--

```
CREATE VIEW Vista_SalariosPorTemporada AS
SELECT
    [Tabla Jugadores (2)].Nombre,
    [Tabla Jugadores (2)].Apellido,
    [Tabla Jugadores-Temporada (1)].ID_Temporada AS Temporada,
    [Salario (3)].Total_Salario
FROM [Salario (3)]
INNER JOIN [Tabla Jugadores (2)]
    ON [Salario (3)].ID_Jugador = [Tabla Jugadores (2)].ID_Jugador
INNER JOIN [Tabla Jugadores-Temporada (1)]
    ON [Salario (3)].ID_Temporada = [Tabla Jugadores-Temporada (1)].ID_Temporada;
```

--Vista_ContratosActivos--

```
CREATE VIEW Vista_ContratosActivos AS
SELECT
    [Tabla Jugadores (2)].Nombre,
    [Tabla Jugadores (2)].Apellido,
    [Contrato (1)].Inicio_Contrato AS Fecha_Inicio_Contrato,
    [Contrato (1)].Fin_Contrato AS Fecha_Fin_Contrato,
    [Equipo (1)].Nombre_Equipo
FROM [Contrato (1)]
INNER JOIN [Tabla Jugadores (2)]
    ON [Contrato (1)].ID_Jugador = [Tabla Jugadores (2)].ID_Jugador
INNER JOIN [Equipo (1)]
    ON [Contrato (1)].Equipo_Contratantes = [Equipo (1)].ID_Equipo
WHERE [Contrato (1)].Fin_Contrato >= GETDATE();
```

--Vista_PartidosGanados--

```
CREATE VIEW Vista_PartidosGanados AS

SELECT

[Equipo (1)].Nombre_Equipo,

[Estadistica_partidos].DATE,

[Estadistica_partidos].OPPONENT,

[Estadistica_partidos].RESULT

FROM [Estadistica_partidos]

INNER JOIN [Equipo (1)]

ON [Estadistica_partidos].ID_Equipo = [Equipo (1)].ID_Equipo

WHERE [Estadistica_partidos].Win_Lose LIKE 'W%';
```

IMPLEMENTACIÓN

La base de datos para el equipo de basketball Golden State se implementó utilizando SQL Server Management Studio (SSMS). Esta sección detalla el proceso de creación de tablas.

Lo primero que hicimos fue definir los requerimientos necesarios para esta base de datos , los cuales ya lo hemos detallado en este trabajo , luego diseñamos el diagrama de la base de datos cumpliendo con los requerimientos anteriormente especificados.

Creamos la base datos en Sql server , ya teniendo nuestro modelo listo creamos las tablas de manera individual supliendo la data de la página oficial de la NBA , utilizando python y pandas , luego agregamos todas estas tablas a la base de datos y la conectamos mediante claves primarias y foráneas , siguiendo el modelo anteriormente elaborado.

Por último creamos todos los stored procedures necesarios para las tareas rutinarias de la base de datos como la actualización de jugadores o la inserción de nuevos jugadores o contratos, tambien creamos vistas para los reportes de rendimientos de los jugadores por temporadas o los reportes de los contratos en cada temporada, así los analista de desempeño del equipo tendrán rápido acceso a estas informaciones.

Obtencion de Datos

+ Código + Texto

import pandas as pd

[]

url='https://www.espn.com/nba/team/schedule/_/name/gs/season/2024'

df=pd.read_html(url)

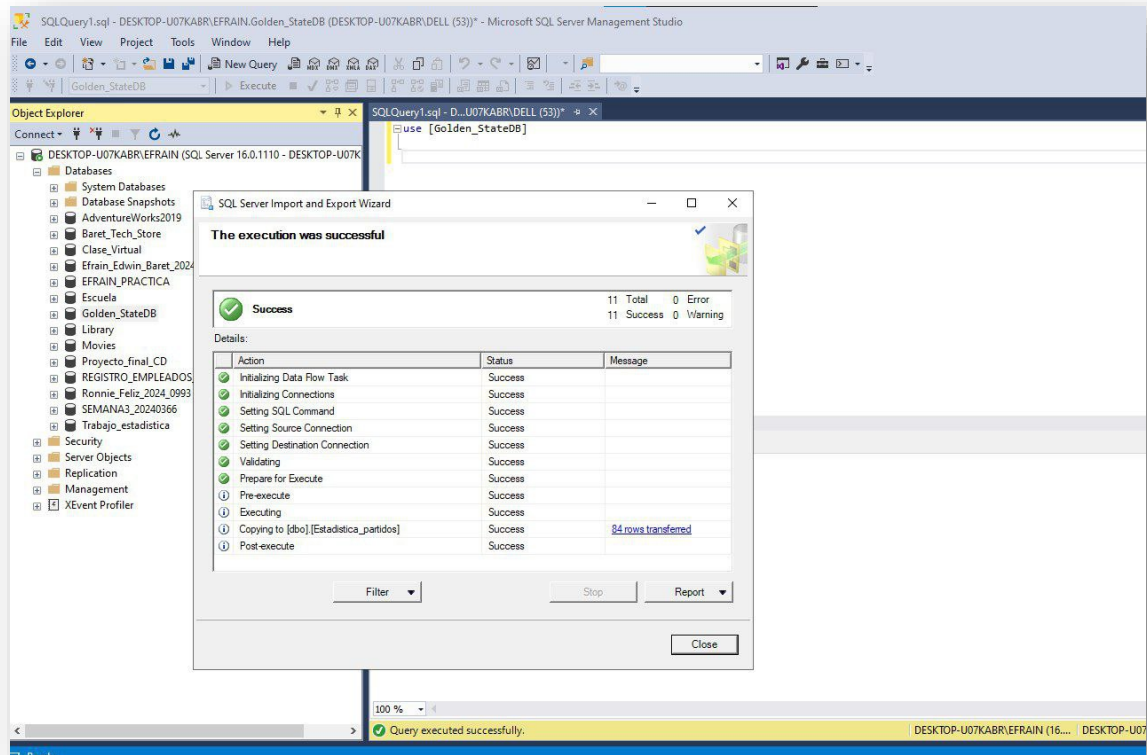
df=df[0]

#eliminar la primera fila

df

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-----|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 0 | Regular Season | Regular Season | Regular Season | Regular Season | Regular Season | Regular Season | Regular Season | Regular Season |
| 1 | DATE | OPPONENT | RESULT | W-L | Hi Points | Hi Rebounds | Hi Assists | NaN |
| 2 | Tue, Oct 24 | vs Phoenix | L108-104 | 0-1 | Curry 27 | Looney 11 | Paul 9 | NaN |
| 3 | Fri, Oct 27 | @ Sacramento | W122-114 | 1-1 | Curry 41 | Looney 12 | Paul 12 | NaN |
| 4 | Sun, Oct 29 | @ Houston | W106-95 | 2-1 | Curry 24 | Looney 11 | Paul 7 | NaN |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 81 | Sun, Apr 7 | vs Utah | W118-110 | 43-35 | Thompson 32 | Kuminga 10 | Paul 9 | NaN |
| 82 | Tue, Apr 9 | @ Los Angeles | W134-120 | 44-35 | Thompson 27 | Curry 7 | Green 10 | NaN |
| 83 | Thu, Apr 11 | @ Portland | W100-92 | 45-35 | Curry 22 | Looney 11 | Curry 8 | NaN |
| 84 | Fri, Apr 12 | vs New Orleans | L114-109 | 45-36 | Curry 33 | Green 12 | Green 11 | NaN |
| 85 | Sun, Apr 14 | vs Utah | W123-116 | 46-36 | Thompson 25 | Podziemski 9 | Kuminga 7 | NaN |

86 rows x 8 columns



CASOS DE USO

1-Análisis de rendimiento de jugadores:

Realizar un seguimiento de las estadísticas de cada jugador durante las temporadas (puntos, rebotes, asistencias, robos, etc.) para evaluar su rendimiento y contribución al equipo

2-Gestión de contratos:

Mantener un registro de los contratos de los jugadores, incluyendo detalles como duración, salarios y cláusulas especiales, lo que facilita la planificación financiera del equipo.

3-Análisis de comparativa de jugadores:

Comparar el rendimiento de diferentes jugadores en base a sus estadísticas, lo que ayuda en decisiones de selección y alineación.

4-Planificación de estrategia de juego:

Utilizar estadísticas históricas para identificar patrones en el rendimiento del equipo y de los oponentes, lo que permite diseñar estrategias más efectivas durante los juegos.

5-Proyecciones de rendimiento:

Utilizar modelos de análisis predictivo para proyectar el rendimiento futuro de los jugadores basándose en su desempeño pasado y en condiciones actuales según los datos.

ESTRATEGIA DE BACKUP DE LA BASE DE DATOS.

Backups Completos (Full Backup):

Frecuencia: Una vez por semana (preferiblemente los fines de semana).

Descripción: Captura toda la base de datos en un solo archivo de respaldo. Esta será la base para otros tipos de backups.

Retención: Mantén los últimos 4 backups completos (1 mes de retención).

Backups de Logs de Transacciones:

Frecuencia: Después de cada partido

Descripción: Respaldar las transacciones que han ocurrido desde el último backup de log. Esto te permite hacer una restauración punto a punto y minimizar la pérdida de datos.

Retención: Mantén los últimos 7 días de logs.

Almacenamiento de los Backups:

Ubicación primaria: Almacena los backups en un servidor privado del equipo o unidad de red local con espacio dedicado.

Ubicación secundaria (off-site): Usa almacenamiento en la nube o un servidor en una ubicación física distinta para una copia secundaria en caso de desastre (por ejemplo, un servicio de Azure o AWS), esto nos asegura que en caso de que le pase algo a los servidores físicos del equipo aun así tengamos la información segura.

Descripción del plan de recuperación en caso de fallos

Como plan de contingencia en caso de alguna falla en nuestra base de datos seguiremos 4 pasos fundamentales para poder asegurar la integridad y seguridad de los datos.

1-Identificación del Problema:

Monitoreamos constantemente la base de datos para detectar fallos como corrupción de datos, errores de hardware o caídas del sistema.

2-Proceso de Recuperación:

Agotaremos varios pasos para el proceso de recuperación:

1. Verificar la última vez que se realizó un backup completo y los logs de transacciones más recientes.
2. Restaurar el último backup completo disponible.
3. Verificar la integridad de los datos y el correcto funcionamiento de la base.

3-Pruebas y Verificaciones:

Validación: Ejecutaremos pruebas para verificar que los datos están completos y la base de datos esté en un estado correcto y consistente.

Monitoreo post-recuperación: Será una obligación monitorear la base de datos después de la recuperación para detectar cualquier anomalía

4-Documentación y Reporte:

Es necesario mantener un registro detallado del proceso de recuperación, incluyendo los tiempos de falla, tiempos de restauración, y cualquier dato o transacción perdida.

Siguiendo esta estrategia se asegura la mínima pérdida de datos y un tiempo de recuperación óptimo en caso de fallos, reconociendo que manejamos datos de alta importancia para el equipo, es necesario no fallar a ninguno de estos criterios de recuperación y seguridad de los datos.

CONCLUSIÓN

En resumen, la implementación de la base de datos para el equipo Golden State Warriors ha logrado cumplir los objetivos planteados, proporcionando una solución efectiva para gestionar información clave sobre jugadores, partidos y estadísticas. Los resultados obtenidos muestran que el sistema facilita el acceso a datos relevantes en tiempo real, optimiza la toma de decisiones basada en métricas precisas, y mejora el análisis de rendimiento tanto a nivel individual como colectivo. La base de datos centralizada ha demostrado ser eficiente y escalable, garantizando flexibilidad para el futuro.

Posibles mejoras:

- Implementar análisis predictivos utilizando inteligencia artificial para predecir el rendimiento de los jugadores o el resultado de los partidos basándose en datos históricos.
- Incorporar una interfaz gráfica más intuitiva para facilitar el uso de la base de datos por parte de los entrenadores y el personal técnico.
- Ampliar la base de datos para incluir información de otros equipos, lo que permitiría realizar análisis comparativos en tiempo real.

Recomendaciones para el futuro:

- Mantener actualizada la base de datos con las estadísticas más recientes y asegurarse de que la seguridad de la información esté acorde con los estándares más actuales.
- Evaluar el rendimiento del sistema periódicamente y ajustar su capacidad para manejar un volumen creciente de datos, a medida que el equipo evoluciona y crece.
- Considerar la integración con plataformas externas de análisis deportivo para obtener una visión más completa de las estrategias y tendencias en la NBA.

Referencias

ESPN. **Golden State Warriors 2023-24 NBA Regular Season Stats**. Consultado el 20 de octubre de 2024. [Golden State Warriors Estadísticas de la temporada regular de la NBA 2023-24 - ESPN](#)

Mockaroo. **Random Data Generator and API Mocking Tool**. Consultado el 20 de octubre de 2024. <https://www.mockaroo.com>

ChartDB. **Diagram 1 Diagram: Visualize Database Schemas**. Consultado el 20 de octubre de 2024. <https://www.chartdb.com>

LearnSQL.es. **Sintaxis SQL**. Consultado el 20 de octubre de 2024. <https://learnsql.es>

Canva. **Herramienta de Diseño Gráfico**. Consultado el 21 de octubre de 2024. <https://www.canva.com>