



# UDN: TRAINING

## PLAN DE

## SEGURIDAD

UNIVERSIDAD TECNOLÓGICA DE  
XICOTEPEC DE JUÁREZ

MATERIA: ADMINISTRACION DE BASES  
DE DATOS

DOCENTE: M.T.I. MARCO ANTONIO  
RAMÍREZ HERNÁNDEZ

INTEGRANTES:

- MARIO GUTIÉRREZ ROSALES 210115
- MARCO ANTONIO MORALES RIVERA 210615
- SURI JAZMÍN PENA LIRA 210644
- JORGE CRUZ CAZARES 210458

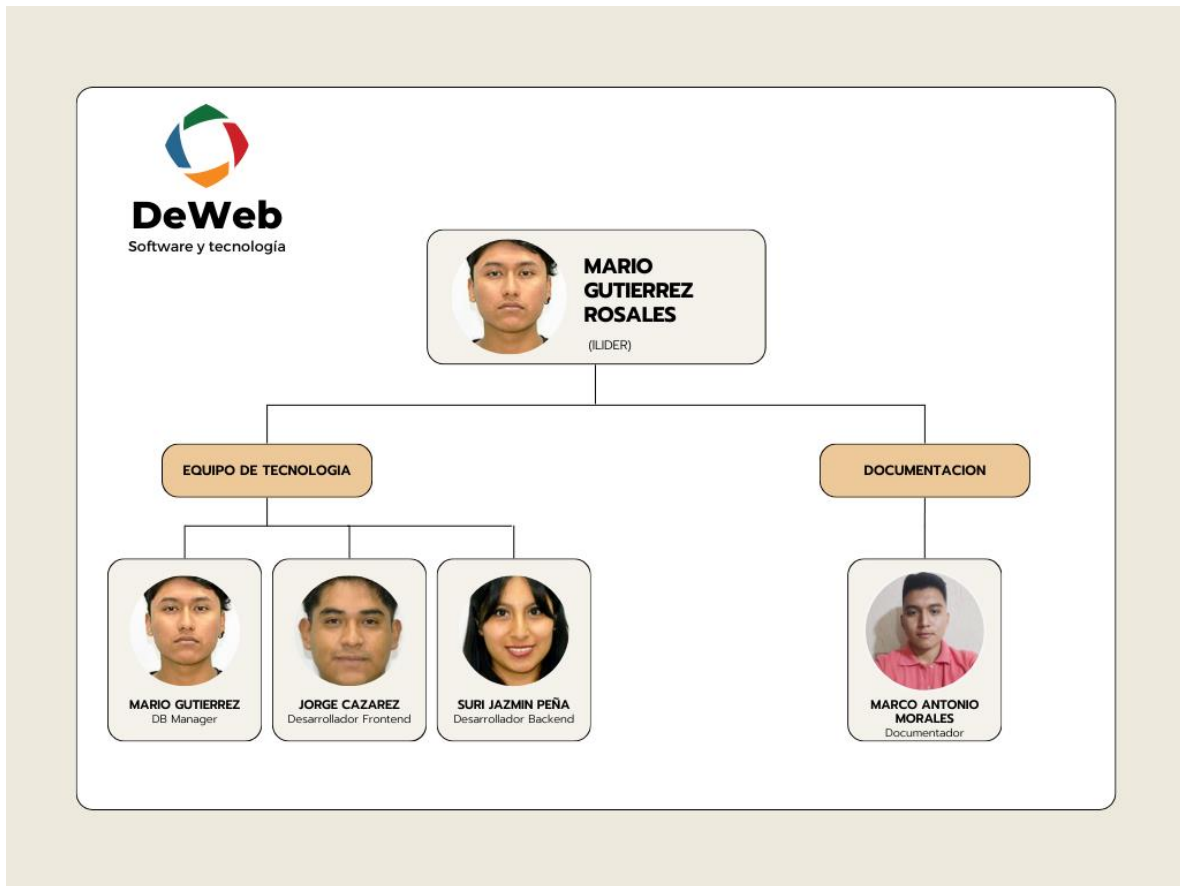
## OBJETIVO GENERAL

Desarrollar un módulo de entrenamiento dentro del sistema web de un gimnasio. Este módulo estará destinado a optimizar la gestión integral de la institución, brindando una plataforma digital que mejore la experiencia tanto de los empleados como de los miembros. Además, se buscará proporcionar herramientas que faciliten la creación, seguimiento y personalización de programas de entrenamiento, rutinas de ejercicios y sesiones específicas para los clientes del gimnasio todo esto con el apoyo de instructores y el uso de una dieta adecuada, promoviendo así la adopción de hábitos saludables y una mejor interacción entre el personal y los usuarios.

## OBJETIVOS ESPECÍFICOS

- **Inicio de Sesión:** Diseñar una interfaz de inicio de sesión intuitiva y atractiva, segura que permita a los usuarios acceder al sistema web del gimnasio. Siguiendo el diseño presentado en el mockup.
- **Módulo de Creación de Rutinas:** Desarrollar un formulario de creación de rutinas que permita a los instructores agregar nuevos ejercicios, establecer repeticiones, series y descansos, validando los datos ingresados para evitar errores y garantizar la coherencia de las rutinas creadas.
- **Módulo de Lectura de Rutinas:** Diseñar una interfaz limpia y fácil de entender que muestre las rutinas existentes de forma clara y ordenada, permitiendo a los usuarios ver detalles específicos de cada rutina, como ejercicios incluidos, series y repeticiones.
- **Módulo de Actualización de Rutinas:** Habilitar la edición de rutinas existentes, permitiendo a los instructores realizar cambios en los ejercicios, repeticiones y otros detalles según sea necesario.
- **Módulo de Eliminación de Rutinas:** Integrar una función de eliminación que permita a los instructores eliminar rutinas obsoletas o incorrectas de manera segura y eficiente.
- **Módulo de Programas Saludables:** Diseñar una página que muestre información detallada sobre los programas saludables ofrecidos por el gimnasio, siguiendo el diseño presentado en el mockup. Mostrar tablas con datos relevantes, como nombres de programas, descripciones y beneficios.
- **Dashboard de Seguimiento:** Desarrollar un dashboard interactivo que proporcione a los usuarios una visión general de su progreso en el gimnasio. Incluir gráficos y estadísticas que muestren datos clave, como el rendimiento en las rutinas, la frecuencia de entrenamiento y el cumplimiento de los programas saludables.

## ORGANIGRAMA DE EQUIPO



Nombre Completo	Cargo
Mario Gutierrez Rosales	Lider/DB Manager
Marco Antonio Morales Rivera	Documentador
Jorge Cruz Cazarez	Desarrollador Backend
Suri Jazmin Peña Lira	Desarrollador Frontend

## ROLES

La creación de roles en una base de datos es crucial, ya que ofrecen un mecanismo robusto para la gestión de la seguridad. Facilitan la administración de los usuarios y aseguran la adherencia a las normativas. Los roles permiten un método adaptable y efectivo para regular el acceso a la información y las capacidades de la base de datos.

Nuestra BD cuenta con los siguientes roles:

- **Usuario**

Persona que no es miembro, pero funciona como visitante dentro del gimnasio.

- **Instructor**

Es el responsable de supervisar y dirigir los entrenamientos dentro del gimnasio, así mismo crear, editar y eliminar rutinas.

- **Miembro**

Persona que cuenta con una membresía, con acceso al gimnasio y funciones del sistema.

- **Desarrollador**

Encargado de gestionar el sitio del gimnasio enfocado a la base de datos.

### Creación de roles en SQL

```
CREATE ROLE desarrollador;  
CREATE ROLE instructor;  
CREATE ROLE miembros;  
CREATE ROLE usuario;
```

✓	89	17:32:06	CREATE ROLE desarrollador	0 row(s) affected	0.031 sec
✓	90	17:32:08	CREATE ROLE instructor	0 row(s) affected	0.000 sec
✓	91	17:32:09	CREATE ROLE miembros	0 row(s) affected	0.000 sec
✓	92	17:32:09	CREATE ROLE usuario	0 row(s) affected	0.000 sec

## USUARIOS

Los usuarios de una base de datos desempeñan una variedad de funciones esenciales, desde el acceso y la manipulación hasta la administración de los datos almacenados. Por esta razón, resulta fundamental establecer con precisión el rol específico que se asignará a cada usuario, para mantener la organización y la seguridad de la base de datos.

🔧 Creación de usuarios y asignación de roles:

```
CREATE USER 'mario'@'%' IDENTIFIED BY 'mario123';
CREATE USER 'marco'@'%' IDENTIFIED BY 'toni123';
CREATE USER 'suri'@'%' IDENTIFIED BY 'jaz123';
CREATE USER 'jorge'@'%' IDENTIFIED BY 'jorge123';
```

✓	93	17:32:11	CREATE USER 'mario'@'%' IDENTIFIED BY 'mario123'	0 row(s) affected	0.016 sec
✓	94	17:32:12	CREATE USER 'marco'@'%' IDENTIFIED BY 'toni123'	0 row(s) affected	0.015 sec
✓	95	17:32:13	CREATE USER 'suri'@'%' IDENTIFIED BY 'jaz123'	0 row(s) affected	0.000 sec
✓	96	17:32:13	CREATE USER 'jorge'@'%' IDENTIFIED BY 'jorge123'	0 row(s) affected	0.000 sec

## PRIVILEGIOS

En una base de datos, los privilegios juegan un papel fundamental para asegurar la gobernanza de las operaciones efectuadas en las distintas tablas. Además, reflejan la integridad de cada usuario, lo que conlleva a la prevención de fallos y al establecimiento de la responsabilidad sobre las actividades ejecutadas por los usuarios.

🔧 Asignación de privilegios:

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, ALTER, DROP ON bd_gimnasio_210115.* TO 'mario'@'%' ;
GRANT SELECT, INSERT, UPDATE, CREATE, DROP, DELETE ON bd_gimnasio_210115.* TO 'marco'@'%' ;
GRANT SELECT, CREATE, UPDATE, DROP ON bd_gimnasio_210115.* TO 'suri'@'%' ;
GRANT SELECT ON bd_gimnasio_210115.* TO 'jorge'@'%' ;
```

✓	97	17:32:16	GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, ALTER, DROP ON bd...	0 row(s) affected	0.000 sec
✓	98	17:32:17	GRANT SELECT, INSERT, UPDATE, CREATE, DROP, DELETE ON bd_gimnasi...	0 row(s) affected	0.000 sec
✓	99	17:32:18	GRANT SELECT, CREATE, UPDATE, DROP ON bd_gimnasio_210115.* TO 'suri'...	0 row(s) affected	0.000 sec
✓	100	17:32:19	GRANT SELECT ON bd_gimnasio_210115.* TO 'jorge'@'%'	0 row(s) affected	0.016 sec

## 🚦 Creación de **roles** con privilegios en **NoSQL**:

```
> db.createRole({
  role: "admin_seguimiento",
  privileges: [
    { resource: { db: "seguimiento_programa", collection: "" }, actions: ["anyAction"] }
  ],
  roles: []
})
< { ok: 1 }
> db.createRole({
  role: "escritura",
  privileges: [
    { resource: { db: "seguimiento_programa", collection: "" }, actions: ["insert", "update", "remove"] }
  ],
  roles: []
})
< { ok: 1 }
> db.createRole({
  role: "lectura",
  privileges: [
    { resource: { db: "seguimiento_programa", collection: "" }, actions: ["find"] }
  ],
  roles: []
})
< { ok: 1 }
```

## 🚦 Creación de **usuarios** y asignación de **roles** en **NoSQL**:

```
> db.createUser({
  user: "desarrollador",
  pwd: "1234",
  roles: ["admin_seguimiento"]
})
< { ok: 1 }
> db.createUser({
  user: "instructor",
  pwd: "1234",
  roles: ["escritura"]
})
< { ok: 1 }
> db.createUser({
  user: "usuario",
  pwd: "1234",
  roles: ["lectura"]
})
< { ok: 1 }
> db.createUser({
  user: "miembro",
  pwd: "1234",
  roles: ["escritura"]
})
< { ok: 1 }
```

## CALENDARIO DE RESPALDOS

Para la unidad de negocio **TRAINING** tenemos en cuenta las siguientes tablas de nuestra base de datos **SQL**:

- Programas Saludables
- Detalle Programas Saludables
- Rutinas
- Rutinas Ejercicios
- Ejercicios
- Sesiones Rutinas

Para la base de datos **NOSQL** tenemos en cuenta un único documento:

- Seguimiento Programa

Los datos de nuestra base tanto sql y nosql se respaldan de la siguiente manera:

1. **Copia de Seguridad Completa (Manual):** Esta es la forma más sencilla y completa de respaldo. Incluye todos los datos en la base de datos, así como parte del registro de transacciones para asegurar la consistencia de la base de datos.
2. **Copia de Seguridad Diferencial:** Solo respalda los cambios realizados desde la última copia de seguridad completa. Esto reduce el tiempo y el espacio de almacenamiento necesario para los respaldos.
3. **Copia de Seguridad de Registro de Transacciones:** Específica para bases de datos con un modelo de recuperación completa o con registro de operaciones masivas. Permite restaurar la base de datos a un punto específico en el tiempo.
4. **Automatización de Respaldos:** Utilizar herramientas como SQL Server Management Studio, T-SQL, o PowerShell para automatizar el proceso de respaldo.
5. **Almacenamiento Seguro:** Guardar las copias de seguridad en una ubicación segura y fuera del sitio para protegerse contra pérdidas de datos catastróficas.

Es de suma importancia elegir el respaldo o método que mejor se adapte a las necesidades y asegurarse de que cada base de datos sea respaldada exitosamente y a tiempo.

Los respaldos mencionados anteriormente deben ser aplicados o considerados para nuestras tablas de mayor relevancia dentro del esquema de trabajo, las tablas a considerar son:

- Programas Saludables
- Rutinas
- Ejercicios
- Sesiones Rutinas

Respecto a las demás tablas sobrantes el respaldo es importante, pero, se puede realizar periódicamente.

La localización de los backups puede variar según la configuración y las preferencias del administrador de la base de datos. Algunas de las opciones comunes para almacenar los backups son:

1. **Disco Local:** Se puede almacenar los backups en el mismo servidor donde reside la BD, aunque no es recomendable por cuestiones de seguridad y disponibilidad.
2. **Disco Remoto:** Utilizar una ubicación de red para almacenar los backups.
3. **Unidades de Cinta:** Para bases de datos muy grandes o para cumplir con políticas de retención a largo plazo, se pueden utilizar unidades de cinta físicamente conectadas al servidor.
4. **Almacenamiento en la Nube:** Servicios como Azure Blob Storage ofrecen una opción segura y escalable para almacenar backups fuera del sitio.
5. **Memoria USB:** Las memorias USB son portátiles y fáciles de transportar, lo que puede ser conveniente para trasladar backups físicamente.



## Evaluación de respaldos

Es esencial realizar evaluaciones regulares de cada método de respaldo para confirmar la legibilidad de los datos. En ocasiones, se producen respaldos que resultan ser ilegibles por distintos motivos. El problema radica en que, a menudo, la ilegibilidad de estos respaldos pasa inadvertida hasta que ocurre una pérdida de datos y se requiere su restauración.

Diversos factores pueden causar este problema, incluyendo desajustes en los cabezales de las unidades de cinta, configuraciones erróneas en el software de respaldo o errores humanos. Independientemente de la causa, la ausencia de revisiones regulares impide garantizar la creación de respaldos fiables de los cuales se puedan recuperar los datos en el futuro.

## TABLAS:

