

PLAN DE SEGURIDAD

Integrantes:

Adán Salas Galván M-190698.
Maximiliano Amador Peña M-210857.
Oscar Osiel Paredes Castillo M-210661.

Abril 2024

OBJETIVO GENERAL.

Llevar a cabo un análisis exhaustivo de las necesidades y expectativas de los usuarios del sitio web de un gimnasio, con especial énfasis en el área de atención al cliente. A partir de este análisis, se pretende diseñar e implementar una serie de módulos y funcionalidades innovadoras que no solo mejoren la experiencia general del usuario al interactuar con la plataforma en línea, sino que también fortalezcan la relación cliente gimnasio. Además, se busca optimizar los procesos de atención al cliente, desde la primera interacción hasta la resolución de consultas y reclamos, con el fin último de elevar los estándares de calidad del servicio ofrecido por el gimnasio.

OBJETIVOS ESPECIFICOS.

Investigar las mejores prácticas y tendencias en el diseño y funcionalidad de sitios web dentro de la industria del fitness, enfocándose en la atención al cliente, para obtener información que guíen el desarrollo de los módulos propuestos.

Diseñar módulos interactivos que permitan a los clientes reservar clases, consultar horarios, recibir notificaciones personalizadas sobre promociones y eventos, y acceder a contenido exclusivo sobre salud y bienestar.

Crear una interfaz intuitiva y accesible para el sitio web del gimnasio, que facilite la navegación y el acceso a la información relevante sobre servicios, horarios, tarifas y promociones, con especial atención a la usabilidad desde dispositivos móviles.

Integrar herramientas de análisis de datos y seguimiento de conversiones para evaluar el rendimiento de campañas de marketing digital, así como el impacto de las estrategias de retención de clientes y promociones específicas en la participación y la lealtad del cliente.

Implementar un sistema de reservas para servicios adicionales, como entrenamientos personales, sesiones de fisioterapia y alquiler de espacios para eventos, que permita a los clientes programar citas de manera conveniente y garantice la disponibilidad de recursos y personal.

Desarrollar un sistema de gestión de reclamaciones en línea que permita a los clientes informar sobre problemas, quejas o sugerencias, y hacer un seguimiento del estado de resolución de forma transparente y eficiente, con notificaciones automáticas y mecanismos de retroalimentación para garantizar la satisfacción del cliente en todo momento.

Crear un espacio virtual para la comunidad del gimnasio donde los clientes puedan interactuar entre sí, compartir experiencias, organizar eventos y participar en desafíos y competencias, con el objetivo de fomentar un sentido de pertenencia y camaradería.

Implementar un sistema de soporte y atención al cliente en tiempo real, que incluya chat en vivo, correos electrónicos automatizados y una base de conocimientos, para resolver consultas y problemas de manera rápida y eficiente.

Integrar funcionalidades de seguimiento y análisis de datos para evaluar el rendimiento de los módulos propuestos, monitorear el nivel de satisfacción del cliente y realizar ajustes basados en métricas objetivas.

Evaluar periódicamente la efectividad de los módulos implementados mediante análisis de datos, encuestas de satisfacción del cliente y revisiones internas, con el objetivo de identificar áreas de mejora continua y mantener la relevancia del sitio web dentro del contexto del mercado del fitness.

Establecer protocolos y procedimientos claros para la gestión de la información personal de los clientes, garantizando el cumplimiento de las regulaciones de privacidad y protección de datos en línea, así como la confidencialidad y seguridad de la información.

ROLES

BASE DE DATOS SQL

Cliente

Rol asignado para que los clientes (usuarios finales) interactúen con la base de datos de forma limitada y segura, solo pudiendo hacer consultas de su propia información.

Creación

```
Database changed
mysql> create role cliente;
Query OK, 0 rows affected (0.05 sec)

mysql>
```

Concesión de privilegios

```
mysql> GRANT select on bd_gimnasio_210857.* to cliente;
Query OK, 0 rows affected (0.01 sec)
```

Privilegios

- SELECT

Administrador

Rol asignado para que los administradores puedan interactuar con datos de clientes y empleados consultando, insertando, actualizando, alterando y borrando datos de clientes y empleados permitiendo correcciones fáciles y simples de datos almacenados.

Creación

```
mysql> create role administrador;
Query OK, 0 rows affected (0.01 sec)
```

Concesión de privilegios

```
mysql> GRANT SELECT, INSERT, UPDATE, ALTER, DELETE ON bd_gimnasio_210857.* TO administrador;  
Query OK, 0 rows affected (0.02 sec)
```

Privilegios

- SELECT
- INSERT
- UPDATE
- ALTER
- DELETE

Desarrollador

Rol asignado para los empleados encargados del mantenimiento y desarrollo del sitio web, otorgándole todos los permisos necesarios sobre la base de datos para su experimentación, correcta conexión con el API y adaptación para nuevas necesidades que generan con el pasar de los años.

Creación

```
mysql> create role desarrollador;  
Query OK, 0 rows affected (0.02 sec)
```

Concesión de privilegios

```
mysql> GRANT ALL ON bd_gimnasio_210857.* TO desarrollador;  
Query OK, 0 rows affected (0.03 sec)  
mysql>
```

Privilegios

- TODOS

Empleado

Rol asignado por defecto a los empleados permitiendo que, puedan hacer consultas, insertar, y actualizar datos sobre la base de datos.

Creación

```
mysql> create role empleado;
Query OK, 0 rows affected (0.01 sec)
```

Concesión de privilegios

```
mysql> GRANT SELECT, INSERT, UPDATE ON bd_gimnasio_210857.* TO empleado;
Query OK, 0 rows affected (0.01 sec)
```

Privilegios

- SELECT
- INSERT
- UPDATE

BASE DE DATOS NoSQL

Cliente

Se crea el rol cliente y se le asignan los privilegios a los que podrá tener acceso desde el momento de su creación.

```
db.createRole({
  role: "cliente",
  privileges: [
    { resource: { db: "bd_gimnasio", collection: "" }, actions: ["find", "listCollections"] }
  ],
  roles: []
})
< { ok: 1 }
```

Empleado

```
db.createRole({
  role: "empleado",
  privileges: [
    { resource: { db: "bd_gimnasio", collection: "" }, actions: ["find", "insert", "update"] }
  ],
  roles: []
})
< { ok: 1 }
```

Administrador

```
db.createRole({
  role: "administrador",
  privileges: [
    { resource: { db: "bd_gimnasio", collection: "" }, actions: ["find", "insert", "update", "remove", "createIndex", "dropIndex", "dropCollection", "collMod"] }
  ],
  roles: []
})
{ ok: 1 }
```

Desarrollador

```
db.createRole({
  role: "desarrollador",
  privileges: [],
  roles: [
    { role: "dbOwner", db: "bd_gimnasio" }
  ]
})
< { ok: 1 }
```

USUARIOS

BASE DE DATOS SQL

Creación de Usuarios

Nombre de Usuario: Adan.

Contraseña: Adan123.

```
mysql> create user 'Adan'@'%' identified by 'Adan123';
Query OK, 0 rows affected (0.01 sec)
```

Nombre de Usuario: Maximiliano.

Contraseña: Max123.

```
mysql> create user 'Maximiliano'@'%' identified by 'Max123';
Query OK, 0 rows affected (0.01 sec)
```

Nombre de Usuario: Osiel.

Contraseña: Osiel123.

```
mysql> create user 'Osiel'@'%' identified by 'Osiel123';
Query OK, 0 rows affected (0.03 sec)
```

Asignación de rol

Asignar rol administrador a Maximiliano

```
mysql> GRANT administrador TO Maximiliano;  
Query OK, 0 rows affected (0.01 sec)
```

Asignar rol empleado y cliente a Adan

```
mysql> GRANT empleado, cliente TO Adan;  
Query OK, 0 rows affected (0.01 sec)
```

Asignar rol desarrollador a Osiel

```
mysql> GRANT desarrollador TO Osiel;  
Query OK, 0 rows affected (0.01 sec)
```

BASE DE DATOS NoSQL

Se crean los roles para la base de datos NoSQL

Creación de usuarios

Al momento de crear los usuarios ya se le pueden asignar los roles ya creados anteriormente.

Usuario Maximiliano:

```
> db.createUser({  
  user: "Maximiliano",  
  pwd: "Max123",  
  roles: ["administrador"]  
})  
< { ok: 1 }
```

Usuario Osiel:

```
> db.createUser({  
  user: "Osiel",  
  pwd: "Osiel123",  
  roles: ["desarrollador"]  
})  
< { ok: 1 }
```


Usuario Adan:

```
> db.createUser({
  user: "Adan",
  pwd: "Adan123",
  roles: ["empleado","cliente"]
})
< { ok: 1 }
```

CALENDARIO DE RESPALDOS

BASE DE DATOS SQL

TABLAS DELEGADAS:

PROPIAS (OWNER)

1. Quejas_Sugerencias
2. Servicios_Cliente
3. Evaluaciones_Servicios
4. Preguntas
5. Servicios_Usuarios
6. Servicios_Sucursal

DE LECTURA (READER)

1. Persona
2. Empleados
3. Miembros
4. Sucursales
5. Instructores
6. Servicios_Sucursales

JERARQUIA Y HORARIO DE RESPALDOS

1 VEZ AL DÍA

Quejas_Sugerencias----- 3:00 A. M.

Servicios_Cliente-----3:00 A. M.

SINTAXIS

```
BACKUP_DIR=/c:/Proyecto/Gimnasio
```

```
BACKUP_FILE="$BACKUP_DIR/$DB_NAME_$(date +%Y%m%d_%H%M%S').sql"
```

```
mysqldump -h $DB_HOST -u $DB_USER -p$DB_PASS $DB_NAME quejas_sugerencias  
servicio_al_cliente > $BACKUP_FILE
```

Se programa de manera que se haga periodicamente

```
0 3 * * * /C:/Documentos/Proyecto/respaldo_mysql_gimnasio.sh
```

1 VEZ CADA 2 DÍAS

Servicios_Sucursal-----3:00 A. M.

Preguntas-----3:00 A. M.

SINTAXIS

```
BACKUP_DIR=/c:/Proyecto/Gimnasio
```

```
BACKUP_FILE="$BACKUP_DIR/$DB_NAME_$(date +%Y%m%d_%H%M%S').sql"
```

```
mysqldump -h $DB_HOST -u $DB_USER -p$DB_PASS $DB_NAME servicios_sucursal  
pregunta
```

Se programa de manera que se haga periodicamente

```
0 3 */2 * * /C:/Documentos/Proyecto/respaldo_mysql_gimnasio.sh
```

1 VEZ A LA SEMANA

Servicios_Usuarios-----3:00 A. M.

Evaluaciones_Servicios-----3:00 A. M.

SINTAXIS

```
BACKUP_DIR=/c:/Proyecto/Gimnasio
```

```
BACKUP_FILE="$BACKUP_DIR/$DB_NAME_$(date +%Y%m%d_%H%M%S').sql"
```

```
mysqldump -h $DB_HOST -u $DB_USER -p$DB_PASS $DB_NAME servicios_usuarios  
evaluaciones_servicios > $BACKUP_FILE
```

Se programa de manera que se haga periodicamente

```
0 3 * * 1 /C:/Documentos/Proyecto/respaldo_mysql_gimnasio.sh
```

BASE DE DATOS NoSQL

Al ser solo una colección en la base de datos, se vio conveniente que su respaldo sea diario para conservar siempre la información.

Tabla:

Seguimiento_queja ----- 3:00 AM

SINTAXIS

```
BACKUP_DIR=/c:/Proyecto/Gimnasio
```

```
BACKUP_FOLDER="$BACKUP_DIR/gimnasio_$(date +%Y%m%d_%H%M%S')"
```

```
mongodump --db gimnasio --collection seguimiento_queja --out $BACKUP_FOLDER
```

```
0 3 * * * /ruta/del/script/respaldo_mongodb_gimnasio.sh 0 3 * * 1
```