



Universidad
Tecmilenio®





Seguridad de Base de Datos

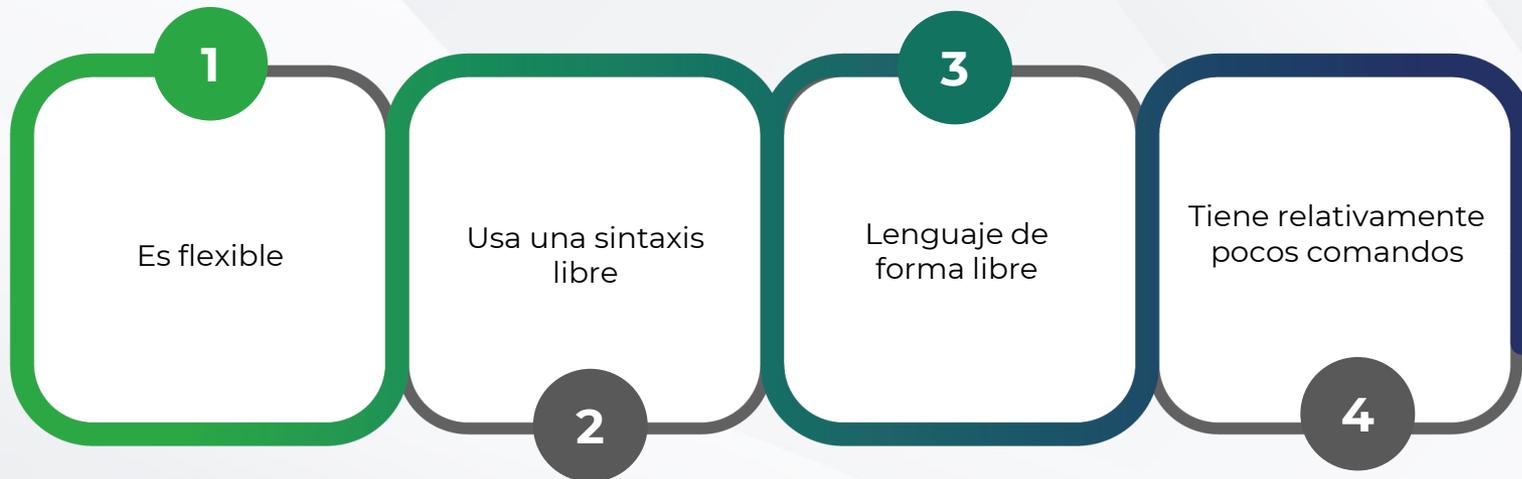
Programación en SQL



SQL es un lenguaje especializado en la comunicación con la base de datos que funciona en otros lenguajes de uso general, mismos que son utilizados para desarrollar las funcionalidades de una aplicación.



Características de SQL



Ventajas de SQL

- Lenguaje de alto nivel y mayor abstracción.
- Usa el mismo lenguaje para estructuras y datos.
- Los programas son portátiles y necesita pocas modificaciones.
- Idioma simple y fácil de aprender.
- Tiene base teórica sólida.
- Es poderoso, ya que procesa conjunto de registros.
- Es independiente en la implementación interna.
- Implementado en diversos DBMS.

```
__tablename__ = "comments" id = db.Column(db.Integer, primary_key=True) author_id = db.Column(db.Integer, db.Fo
users.id') author = relationship("User", back_populates="comments") post_id = db.Column(db.Integer, db.Fo
blog_posts.id') blogpost = relationship("BlogPost", back_populates="comments") text = db.Column(db.Text, n
create_all() gravatar = Gravatar(app, size=100, rating='g', default='retro', force_default=False, force_low
False
er.init_app(app) @login_manager.user_loader
): @wraps(function) def decorated_function(
, **kwargs) return decorated_function @app.
late("index.html", all_posts=posts[::-1]) @
r_form = RegisterForm() if register_user_fo
_name = User.query.filter_by(email=email).f
n redirect(url_for('login')) else: password
'pbkdf2:sha256', salt_length=8) new_user =
ata, password = password, ) db.session.add(
et_all_posts') return render_template("reg
) def login(): login_user_form = LoginForm(
ail.data password = login_user_form.passwor
sh("Email does not exist.") elif not check_
se: login_user(user_name) return redirect(u
_user_form) @app.route('/logout') @login_re
route("/post/<int:post_id>", methods=['GET'
m() post = BlogPost.query.get(post_id) if c
h("You need to login or register to comment
form.body.data, author = current_user, blog
_for("show_post", post_id=post.id)) request
_template("post.html", post=requested_post,
) @app.route("/about") def about(): return render_template("about.html") @app.route("/contact") def contact
render_template("contact.html") @app.route("/new-post", methods=['GET', 'POST']) @login_required @admin_onl
(): form = CreatePostForm() if form.validate_on_submit(): new_post = BlogPost(title=form.title.data, subtit
data, body=form.body.data, img_url=form.img_url.data, author=current_user, date=date.today(), strftime("%B %
```



SQL i - SQL d

- SQL incorporado
- SQL dinámico

DDL

- VDL
- Integridad
- Autorización

Componentes del lenguaje SQL

DCL

- Otorgar
- Revocar
- Almacenar Trans
- Eliminar Trans

DML

- Consultar
- Insertar
- Eliminar
- Consultar



Manipulación de datos en SQL





Con base en la tabla, realiza las tres principales actividades:

- Insertar una tupla.
- Eliminar la tupla de la clave 753.
- Actualizar la tupla de la clave 951 en el campo Ciudad.

Clave	Proveedor	Ciudad	Estado
159	Cielo S.A.	León	Gto
456	Planos S.A.	Tampico	Tamps
753	Atlas S.A.	Saltillo	Coah
808	Seguriventa	Veracruz	Ver
951	Patito S.A.	Morelia	Mich



- El tratamiento de datos estructurados es muy importante, gracias al lenguaje SQL se logra de una manera práctica y rápida.
- SQL es el lenguaje de consulta más utilizado en la industria de bases de datos.



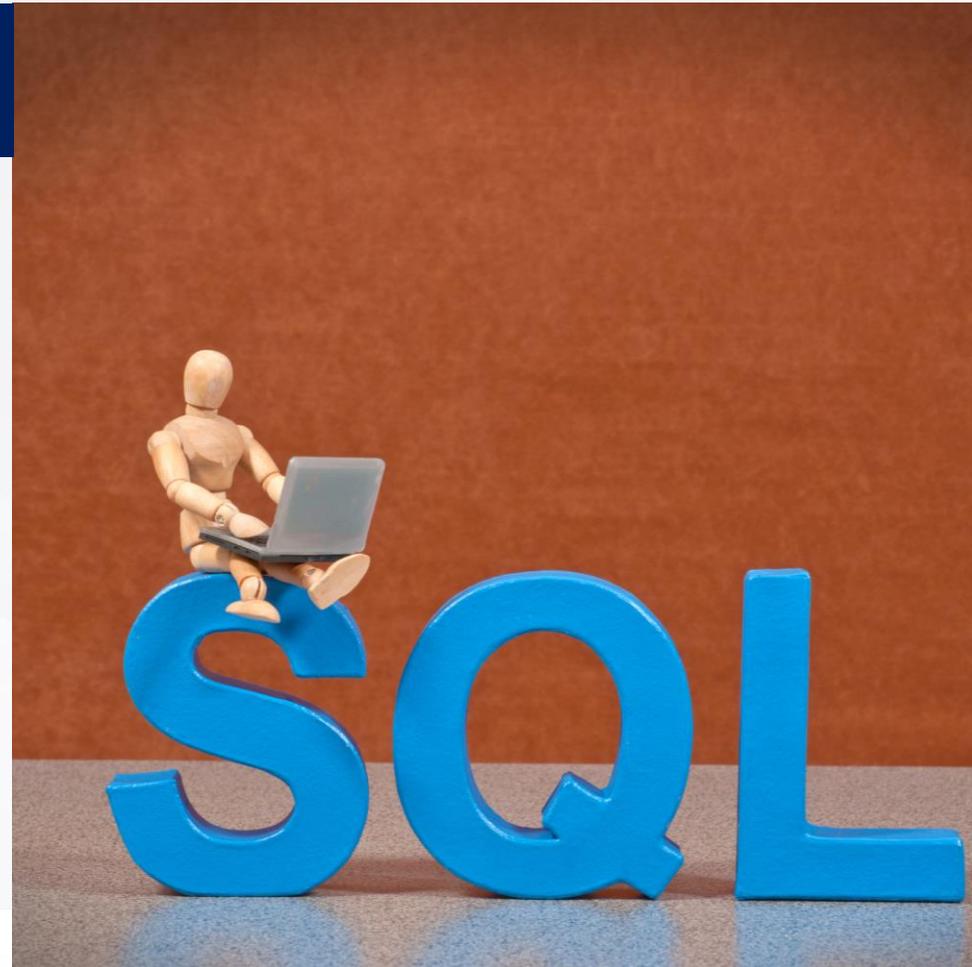
Structured



Query



Language





Universidad
Tecmilenio®



Seguridad de Base de Datos

Asegurar el servidor SQL



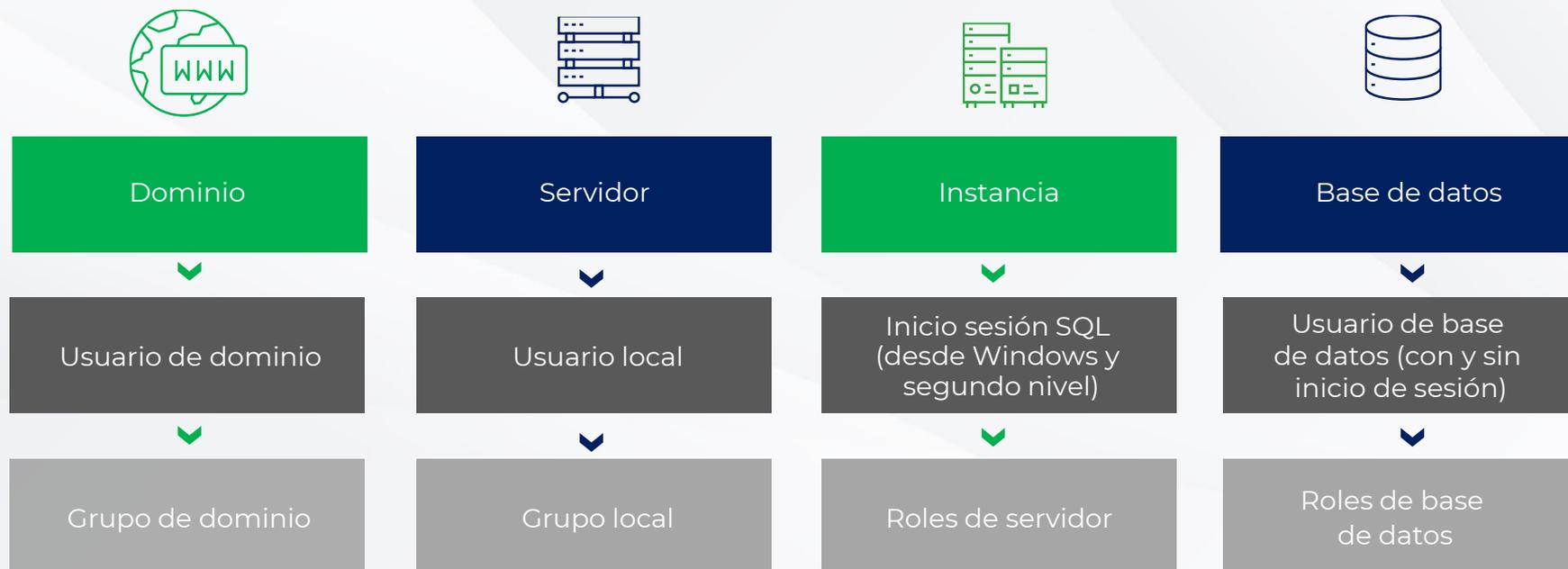


La protección del servidor ha sido un tema de estudio por los expertos, tal es el caso de SQL Server 2017 que implementó un marco de seguridad con el fin de ayudar a los DBA para el manejo de riesgos y amenazas.

La seguridad activa se refiere a la práctica de limitar el acceso de los usuarios a datos y estructuras, con el uso de permisos.



Jerarquía principal de seguridad



Seguridad de nivel de instancia y Logins

- Implica la creación y administración de inicios de sesión, credenciales y roles de servidor.
- Existen dos modos de autenticación en SQL Server:
 - Modo Windows: los usuarios deben autenticarse en el servidor local o en el dominio.
 - Modo mixto: se puede firmar por Windows o por usuario de SQL (conocido como inicio de sesión de segundo nivel).
- La autenticación de modo mixto se aplica en los siguientes casos:
 - Hay aplicaciones heredadas que requieren un inicio de sesión de segundo nivel.
 - Se accede desde fuera del dominio (como un servidor Linux).
 - En entornos donde la seguridad se implementa en el nivel de la aplicación y un único inicio de sesión SQL se conecta al motor de la base de datos.





Roles del servidor y credenciales

Roles de servidor integrados: listos para usar con permisos de nivel de instancia a inicios de sesión que tienen requisitos comunes.

Roles de servidor personalizados: otorgan un conjunto personalizado de permisos a un grupo de inicios de sesión.

Credenciales: proporcionan la capacidad de acceder a los recursos.





Revisa en SQL Server 2019 los roles de servidor fijos y determina cuáles son los tres más apropiados para la organización en la que laboras.





Actualmente, la seguridad basada en roles es inmersa al DBMS.



SQL Server admite dos modos de autenticación: Windows y mixto.



Es recomendable que el usuario se autentique tanto a nivel sistema operativo como a nivel de base de datos.

