

Arquitectura de aplicaciones

Lo muy básico que todo ejecutivo debería conocer



¿Cómo se organizan los distintos
elementos de software detrás de
una aplicación de software?

Contenido




- ▶ Tipos de arquitectura de aplicaciones
 - ▶ ¿Cómo las clasificamos?
 - ▶ Ventajas y desventajas de cada modelo
- ▶ Arquitecturas cliente/servidor y middleware
 - ▶ Middleware, APIs y plataformas de integración
 - ▶ Arquitecturas multi-nivel
- ▶ Arquitectura de aplicaciones y el SW Stack
 - ▶ ¿Dónde se instala cada elemento de la arquitectura?
 - ▶ Contenedores

Arquitectura de Aplicaciones

¿De qué estamos hablando?

La forma en que las funciones básicas de las aplicaciones están distribuidas entre los clientes y los servidores.

Funciones

- 1 ▶ Almacenamiento de datos
- 2 ▶ Acceso a datos 
- 3 ▶ Lógica de aplicación 
- 4 ▶ Lógica de presentación 

Arquitectura de Aplicaciones

3 modelos generales

Almacenamiento y
acceso a datos

Lógica de
aplicación

Lógica de
presentación

Arquitectura de Aplicaciones

3 modelos generales

Almacenamiento y
acceso a datos

Lógica de
aplicación

Lógica de
presentación

Basada en servidor

Mainframes o nube con clientes livianos



Servidor



Servidor



Cliente

Arquitectura de Aplicaciones

3 modelos generales

Almacenamiento y
acceso a datos

Lógica de
aplicación

Lógica de
presentación

Basada en servidor

Mainframes o nube con clientes livianos



Servidor



Servidor



Cliente

Cliente/Servidor

Software instalado en servidores y clientes



Servidor



Servidor



Cliente



Cliente

Arquitectura de Aplicaciones

3 modelos generales

Almacenamiento y
acceso a datos

Lógica de
aplicación

Lógica de
presentación

Basada en servidor

Mainframes o nube con clientes livianos



Servidor



Servidor



Cliente

Cliente/Servidor

Software instalado en servidores y clientes



Servidor



Servidor



Cliente



Cliente

Basada en cliente

Stand-alone o P2P



Cliente



Cliente



Cliente

Basada en servidor



Ejemplos

- ▶ Mainframes
- ▶ Aplicaciones web
- ▶ SaaS

Ventajas

- ▶ Administración centralizada
- ▶ Aplicaciones disponibles desde cualquier cliente con acceso a servidor.

Desventajas (mainframe)

- ▶ Servidor = cuello de botella
- ▶ Inversión fuerte
- ▶ Difícil empatar oferta de recursos con demanda.

Basada en clientes



Ejemplos

- ▶ Procesadores de texto y hojas de cálculo instaladas en laptops o PCs.
- ▶ Entornos P2P

Ventajas

- ▶ Menor requerimientos de capital.
- ▶ Fácil empatar oferta de recursos con demanda.

Desventajas

- ▶ Administración descentralizada.
- ▶ Dificultad para sincronizar datos.

Cliente / Servidor



Ejemplos

- ▶ Versiones con clientes de...
- ▶ ERP
- ▶ BI
- ▶ Correo

Ventajas

- ▶ Procesamiento distribuido.
- ▶ Relativamente fácil escalar donde se necesite.
- ▶ Permite HW y SW de distintas marcas y proveedores trabajando juntos.

Desventajas

- ▶ Mayor complejidad
- ▶ Se requiere de *middleware*.

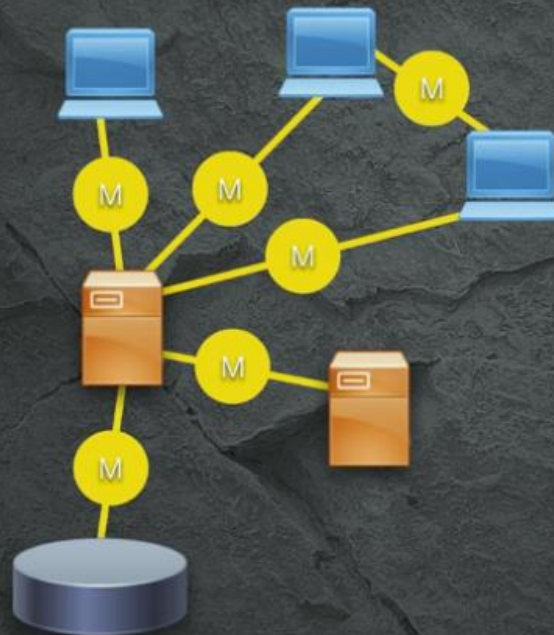


Arquitectura cliente/servidor y Middleware

Middleware

El pegamento que une las partes

Elemento de software que conecta a 2 aplicaciones de software independientes.



- ▶ Método estándar para intercambiar mensajes.
- ▶ Aísla cambios en un elemento de la aplicación de otros (ej., usuario pasa de Windows a Mac y se conecta al mismo servidor sin ajustes en servidor).

Middleware

Ejemplos de interfaces

- ▶ Distributed Computing Environment (DCE)
- ▶ Common Object Request Broker Architecture (CORBA)
- ▶ Open Database Connectivity (ODBC)
- ▶ Java Database Connectivity (JDBC)

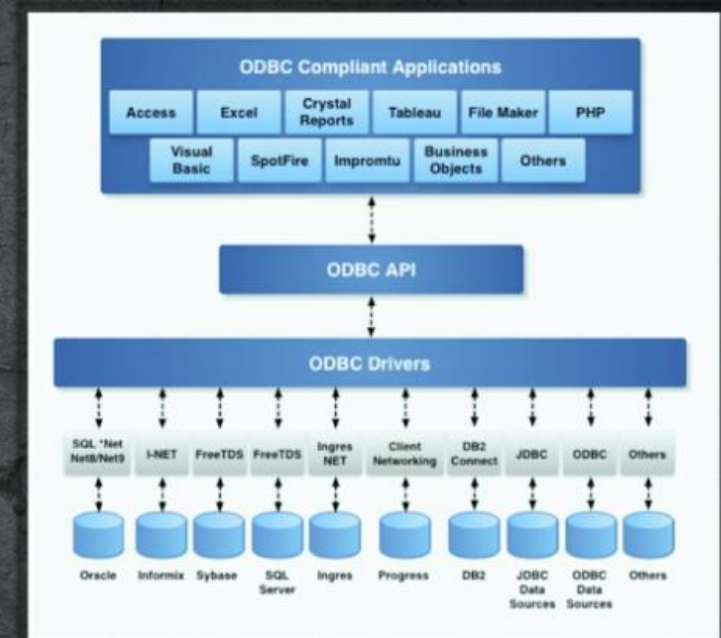
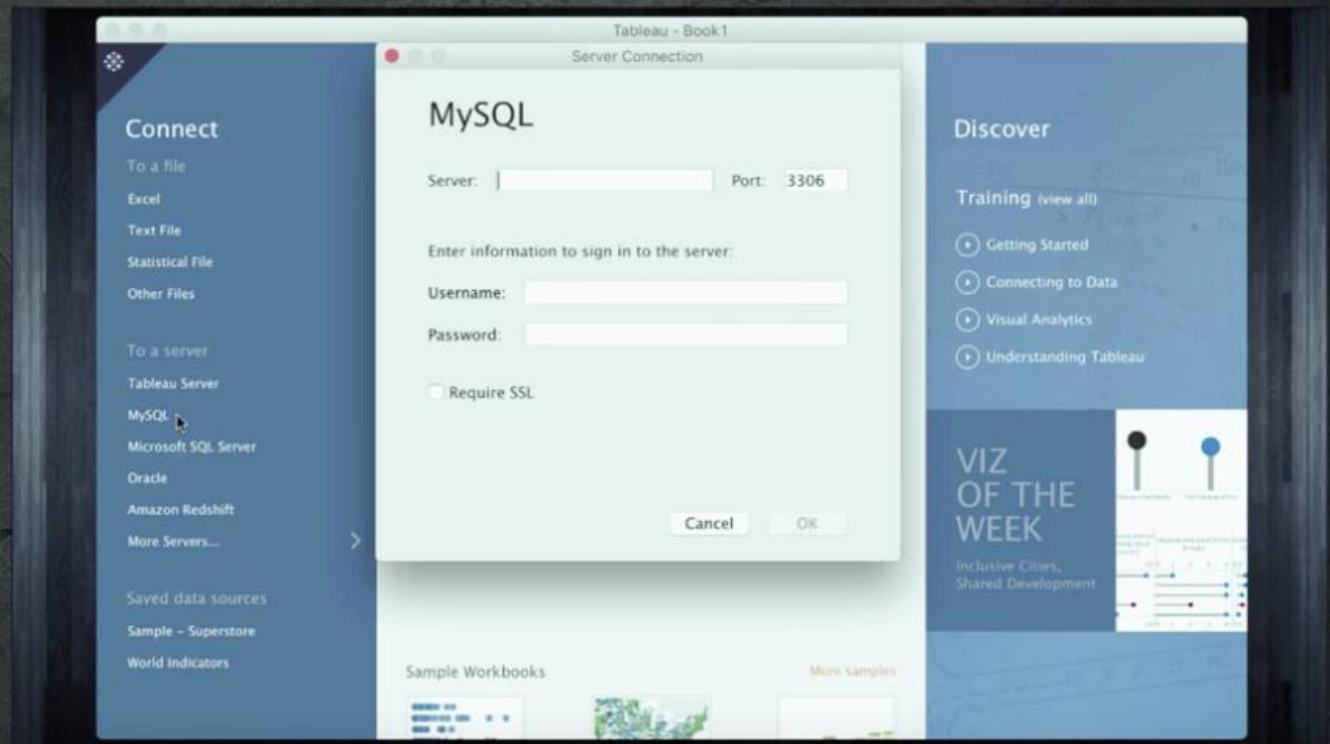


Diagrama de OpenLink Software

Extrayendo datos

Conectándose a una base de datos

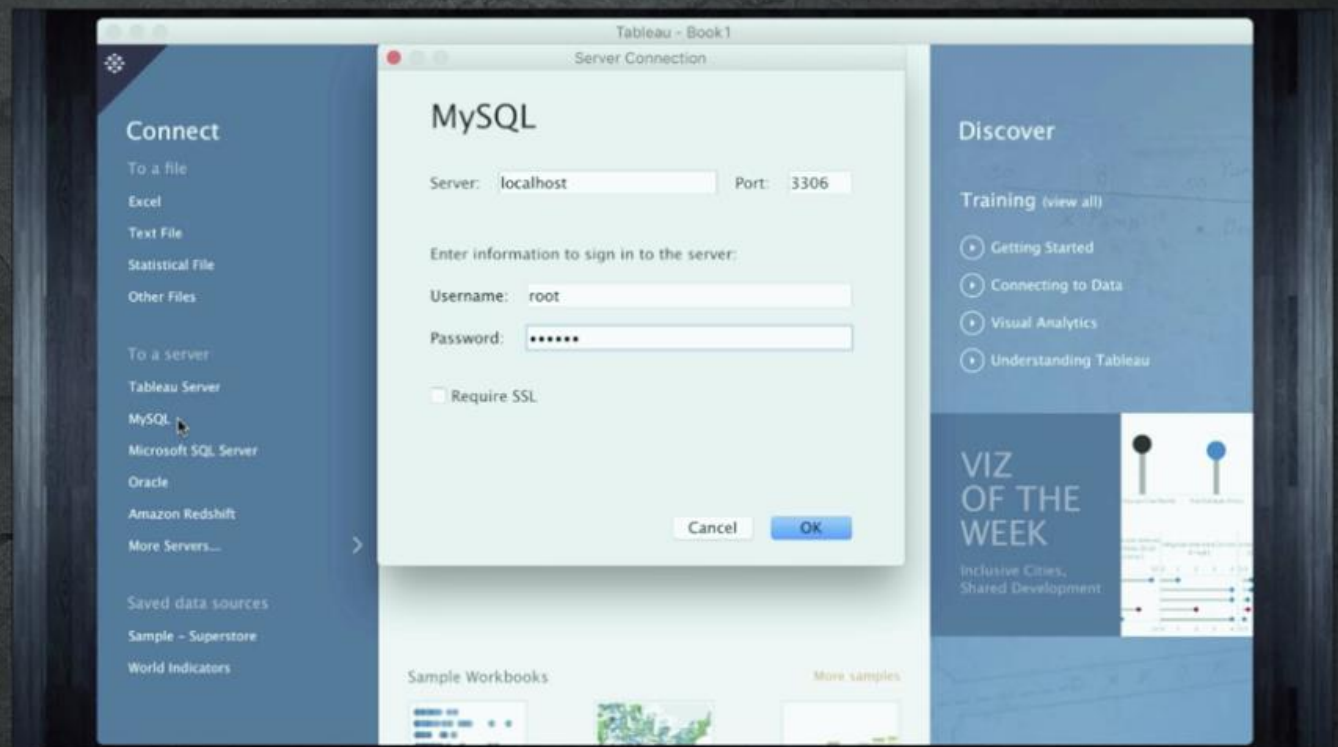
► Tableau



Extrayendo datos

Conectándose a una base de datos

► Tableau



Extrayendo datos

Conectándose a una base de datos

► Tableau

The screenshot shows the Tableau interface with the following details:

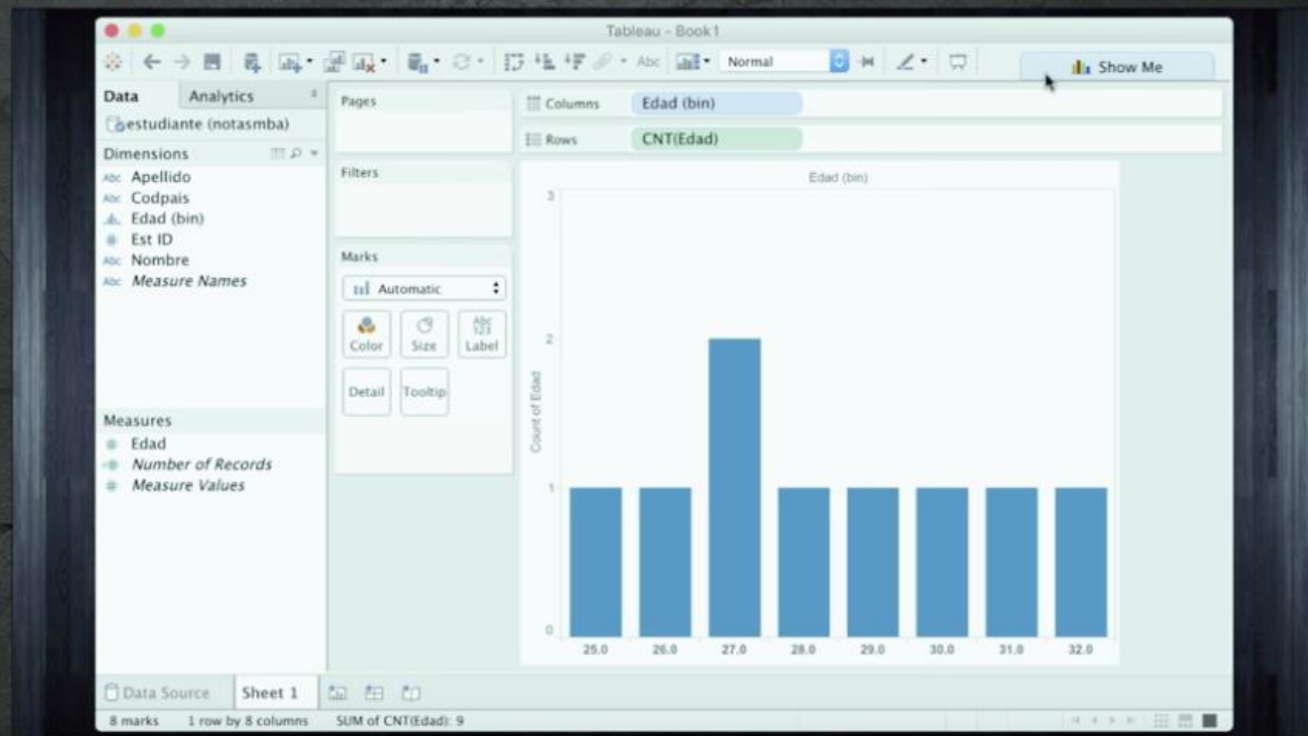
- Tableau - Book1** (Title bar)
- Connection:** ☒ Live ☐ Extract
- Filters:** 0 | Add...
- Server:** localhost
- Database:** notasmba
- Table:** estudiante (notasmba)
- Table Structure:** Est ID, Nombre, Apellido, Codpais, Edad
- Data Rows:** 9 rows displayed
- Buttons:** Copy, Show aliases, Show hidden fields
- Bottom Bar:** Data Source, Sheet 1, Go to Worksheet

Est ID	Nombre	Apellido	Codpais	Edad
1	Alfredo	Álvarez	CRC	27
2	Beatriz	Bosa	NIC	25
3	Carlos	Castilla	PER	30
4	Daniela	Díaz	Castilla ECU	29
5	Eduardo	Elizondo	PAN	32
6	Fernanda	Flores	GUA	26
7	Carolina	Cómac	SAI	37

Extrayendo datos

Conectándose a una base de datos

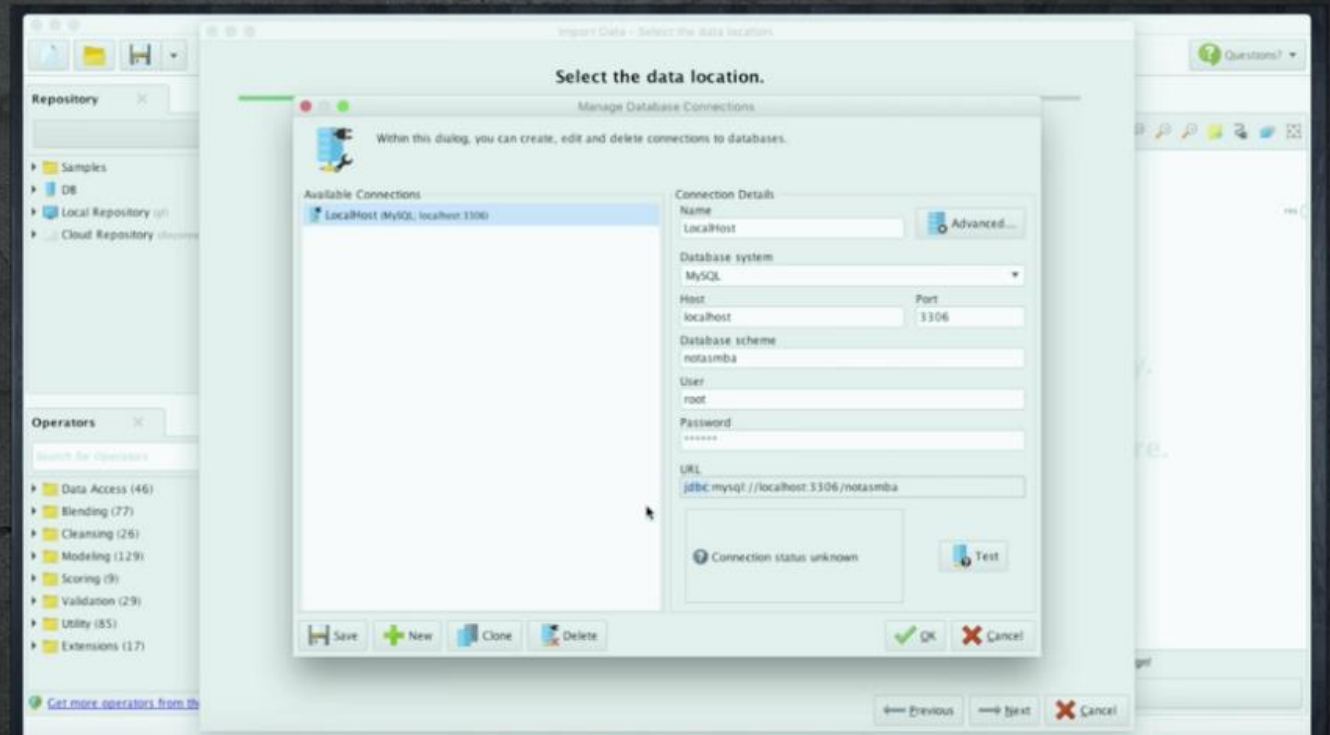
► Tableau



Extrayendo datos

Conectándose a una base de datos

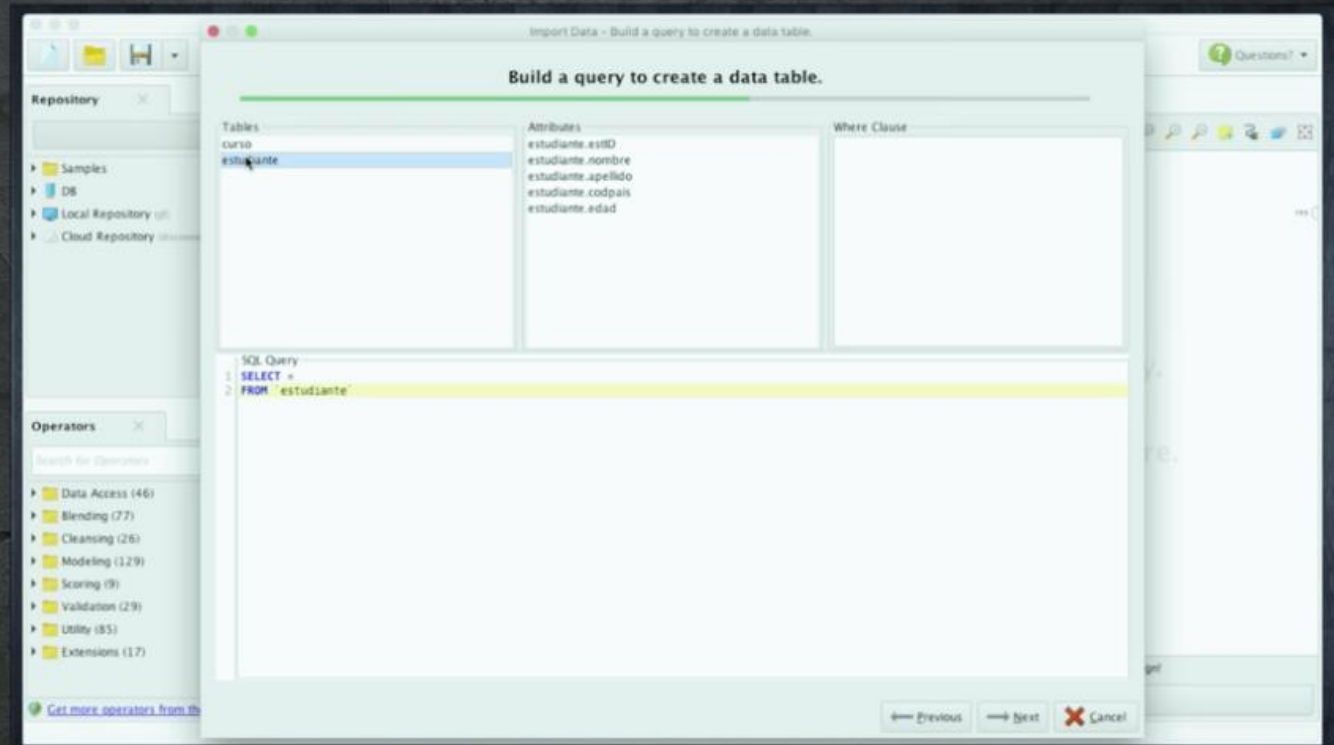
- ▶ Tableau
- ▶ RapidMiner



Extrayendo datos

Conectándose a una base de datos

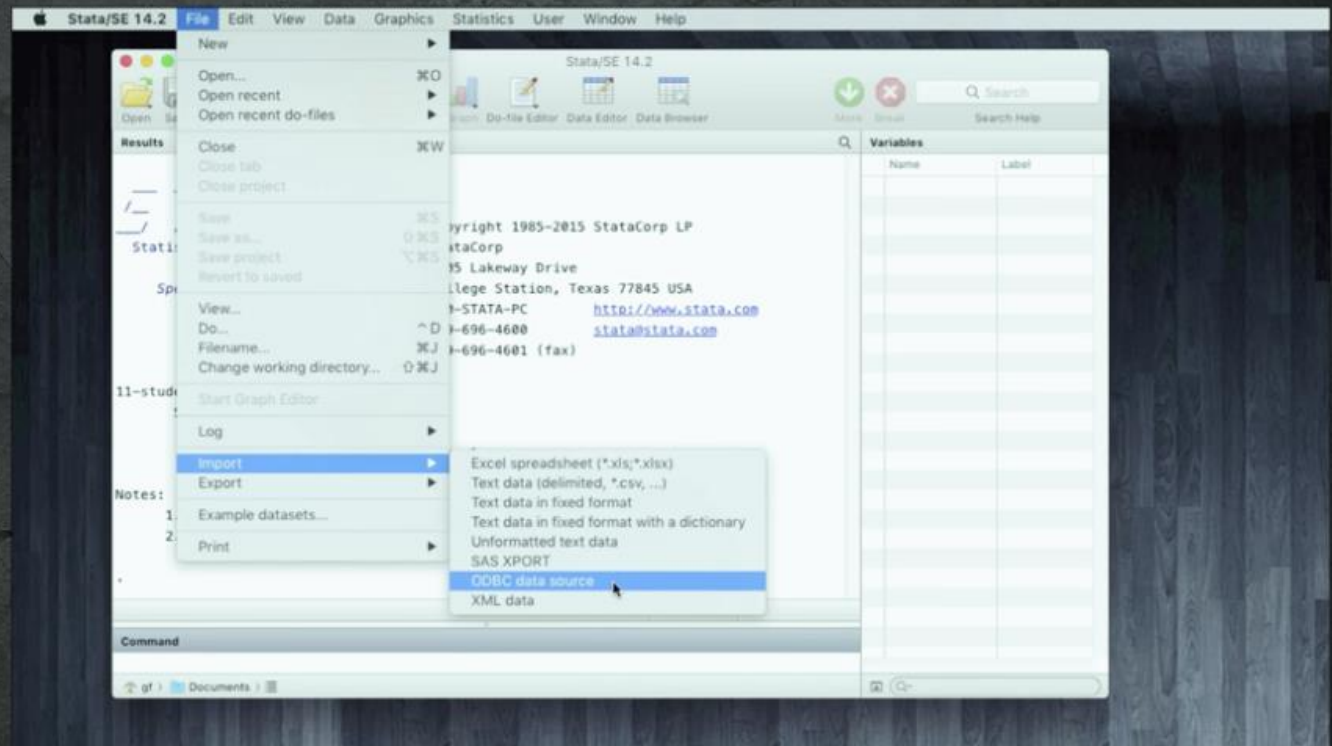
- ▶ Tableau
- ▶ RapidMiner



Extrayendo datos

Conectándose a una base de datos

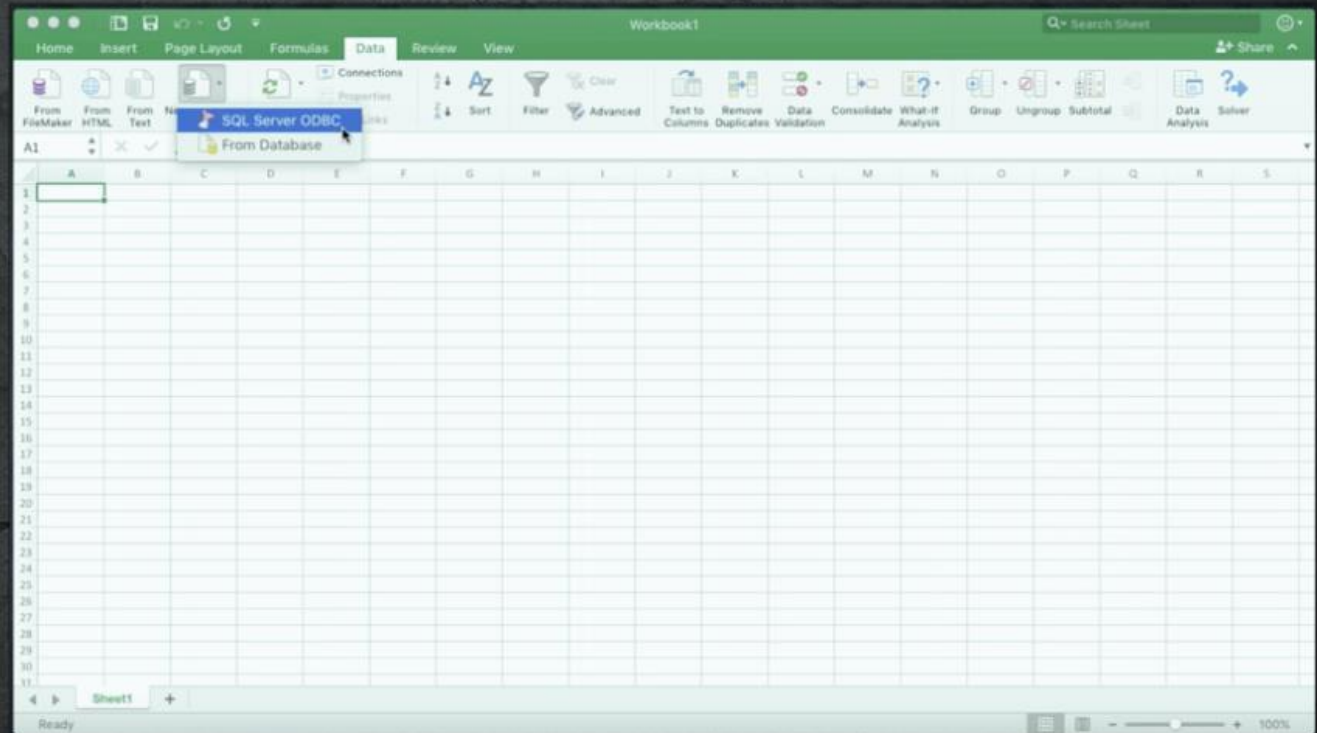
- ▶ Tableau
- ▶ RapidMiner
- ▶ Stata



Extrayendo datos

Conectándose a una base de datos

- ▶ Tableau
- ▶ RapidMiner
- ▶ Stata
- ▶ Excel



Arquitecturas multi-nivel

Multi-tier, multi-capas, o multi-nivel

La lógica de la aplicación se distribuye entre más de 2 clientes o servidores.

► 2-tier:

- 1 cliente y 1 servidor
- Popular en LANs sencillas



► 3-tier:

- 1 cliente y 2 servidores

Arquitecturas multi-nivel

Multi-tier, multi-capas, o multi-nivel

La lógica de la aplicación se distribuye entre más de 2 clientes o servidores.

► 2-tier:

- 1 cliente y 1 servidor
- Popular en LANs sencillas



► 3-tier:

- 1 cliente y 2 servidores



Arquitecturas multi-nivel

Multi-tier, multi-capas, o multi-nivel

La lógica de la aplicación se distribuye entre más de 2 clientes o servidores.

► n-tier:

► Más de 3 elementos



Arquitecturas multi-nivel

Multi-tier, multi-capas, o multi-nivel

La lógica de la aplicación se distribuye entre más de 2 clientes o servidores.

► n-tier:

► Más de 3 elementos



Ventajas

- Distribución de cargas
- Fácil de escalar
- Flexibilidad

Desventajas

- Complejidad
- Difícil determinar causa de fallos

APIs

Application Program Interface

Interfaz expuesta o ofertada por una aplicación de software para que otra hable con ella.



Micrófono

Fotos

Cámara

GPS

Sistema operativo (Android o iOS)

Hardware de teléfono

APIs

Application Program Interface

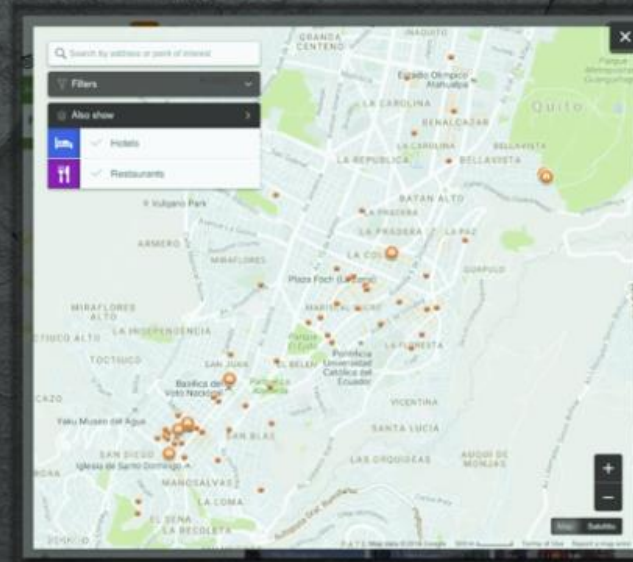
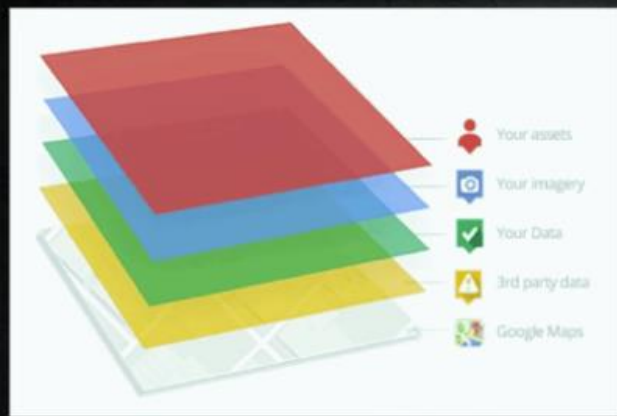
Interfaz expuesta o ofertada por una aplicación de software para que otra hable con ella.



APIs

Application Program Interface

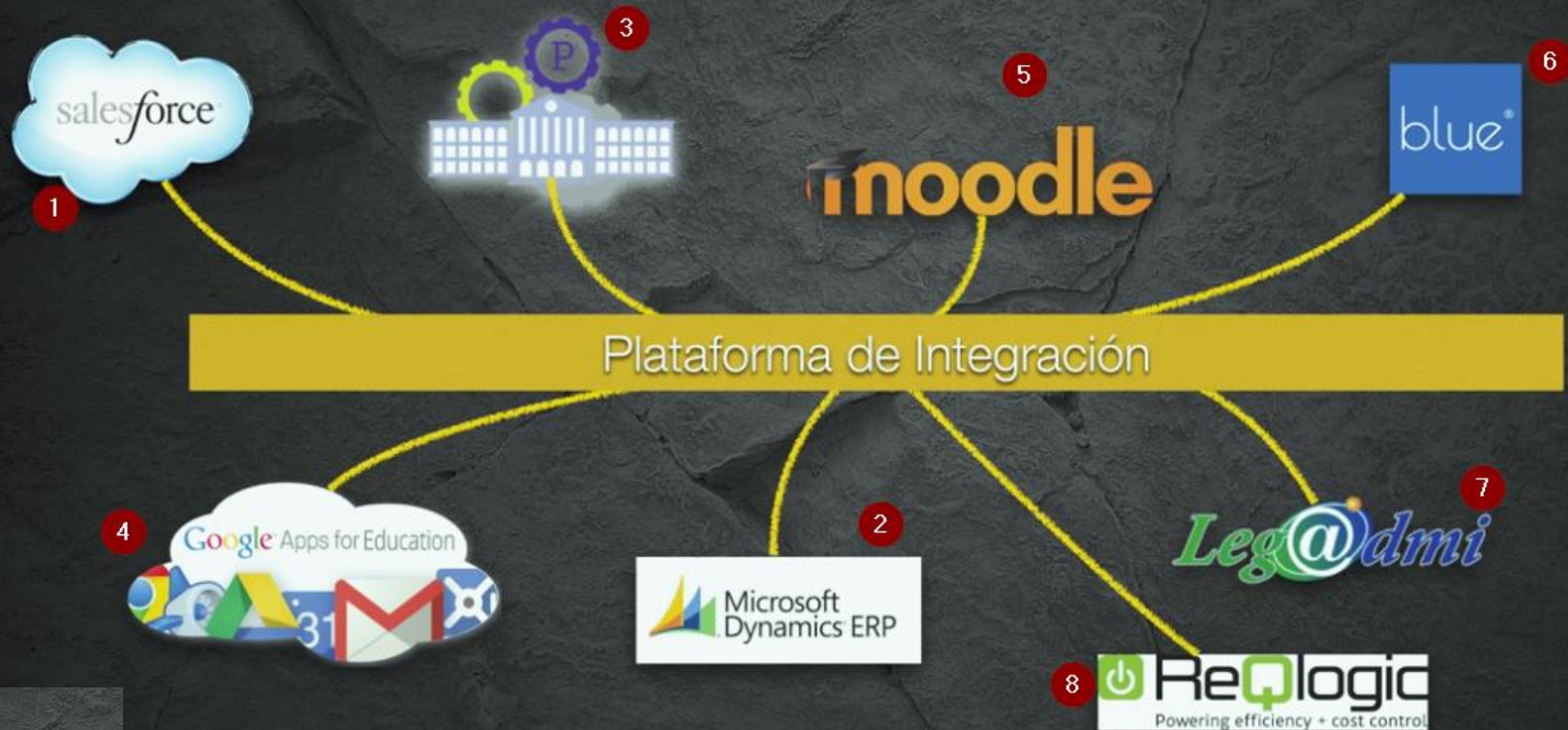
Interfaz expuesta o ofertada por una aplicación de software para que otra hable con ella.



- ▶ Geo-localización
- ▶ Direcciones
- ▶ Información de lugares

Plataforma de integración

Paquetes de software diseñados para integrar sistemas



The background is a dark, charcoal-colored surface with a rough, stone-like texture. There are several vertical and diagonal cracks or fissures running across the surface, giving it a rugged appearance. The lighting is slightly uneven, with some areas appearing darker than others, emphasizing the texture.

Arquitectura de aplicaciones y el Software Stack

Software Stack

“Todo el SW necesario para que corra”



LAMP Stack

Virtualización

“Servidores dentro de servidores”

Página web

Apache

PhP (WordPress)

MySQL ver. A

Linux

Hardware

Manufactura

MSFT .NET

MSFT SQL Server

Windows

Hardware

Mercadeo

Java

MySQL ver. B

Linux

Hardware

Ventas

Apache

Python

Maria DB

Linux

Hardware



Virtualización

“Servidores dentro de servidores”



Contenedores

“Empaquemos sólo lo estrictamente necesario”



Página web



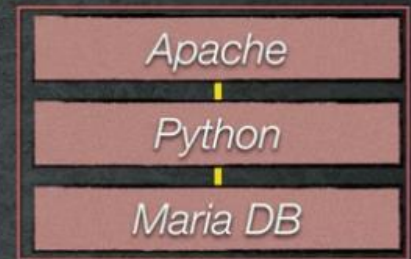
Manufactura



Mercadeo



Ventas



Sistema de contenedores (Docker)

Sistema operativo (Linux)

Hardware físico o real

Comentarios...