

# Modelos Cloud y Análisis de Odoo

## 1. Modelo de servicio en la nube de cada uno

- Odoo Online pertenece al modelo **SaaS (Software como servicio)**. Justificación: la propia web de Odoo dice “Odoo Online is SaaS ... you'll never have to worry about the technical aspects because Odoo takes care of everything”. [Odoo+2Odoo+2](#)
- Odoo.sh pertenece al modelo **PaaS (Plataforma como servicio)**. Justificación: Odoo lo describe como “PaaS (Platform as a Service): Odoo.sh ... we also offer a PaaS ... it allows you to create your cloud-based database whilst having custom developments or third-party apps installed.” [Odoo+2Odoo+2](#)

En resumen:

- Odoo Online → SaaS
  - Odoo.sh → PaaS
- 

## 2. Grado de control del usuario en cada caso

Voy a detallar varios aspectos: instalación, módulos, base de datos, código fuente, copias de seguridad.

### Odoo Online

- Instalación: el usuario no realiza la instalación de la plataforma, se pone en marcha con unos clics y el proveedor se encarga del hosting, infraestructura, mantenimiento. [Odoo](#)
- Módulos: solo se pueden instalar los módulos “certified” por Odoo (aplicaciones oficiales) o apps del repositorio estándar, no se permite instalar módulos de terceros personalizados ni cambiar el código fuente. [Odoo+2Odoo+2](#)
- Base de datos: la base de datos está gestionada por Odoo; hay backups diarios, replicación en múltiples máquinas, el usuario puede en general duplicar bases de datos para pruebas (según el caso) pero no tiene el control profundo del motor o de la infraestructura. [Odoo](#)
- Código fuente: el usuario **no** tiene acceso para modificar el código núcleo del sistema, ni desplegar sus propios módulos personalizados (o al menos no libremente). [Odoo+2Odoo+2](#)
- Copias de seguridad: Odoo se encarga de las copias de seguridad (diarias, replicación en múltiples centros). El usuario puede descargar backups en el interfaz de gestión (en determinados límites) en Odoo Online. [Odoo](#)

# Odoo.sh

- Instalación: la infraestructura ya está montada por Odoo (gestión del hosting/servicios) pero el usuario gana más control porque puede desplegar su propio código, módulos, gestionar ramas, etc. [Odoo+2The Odoo Cloud Platform+2](#)
- Módulos: permite instalar aplicaciones de terceros y módulos personalizados, además de los módulos “core”. Es parte del valor de Odoo.sh. [Odoo](#)
- Base de datos: el usuario puede gestionar entornos de “staging” (pruebas), desarrollo, producción; se integran backups y control más fino. [cybrosys.com+1](#)
- Código fuente: sí permite modificar código, trabajar con GitHub, hacer ramas, CI/CD (integración continua), ver logs, etc. [Odoo+1](#)
- Copias de seguridad: también gestionadas por Odoo, pero con más visibilidad: backups diarios incrementales, replicación, posibilidad de restaurar, entornos de prueba. [The Odoo Cloud Platform+1](#)

En resumen del grado de control:

- Odoo Online → mínimo control técnico/interno: instalación y administración la gestiona Odoo; personalización limitada; módulo restringido; código cerrado para el usuario.
- Odoo.sh → más control: infraestructura gestionada parcialmente, pero tú gestionas el despliegue, módulos personalizados, código; tienes entornos de desarrollo/prueba; aunque sin tener que gestionar servidor hardware desde cero como un “on-premise”.

---

## 3. Tipo de empresa o perfil de usuario que se beneficia más de cada servicio

- Odoo Online: Ideal para **empresas pequeñas o medianas**, con procesos estándar, que no necesitan muchas personalizaciones ni integraciones complejas, que no quieren asumir la carga de infraestructura/servidores ni tener un equipo técnico potente. Por ejemplo: startups, PYMEs con procesos ERP/CRM relativamente “out-of-the-box”, que valoran simplicidad, rapidez de puesta en marcha, costes reducidos, y que aceptan las limitaciones de personalización.
  - Odoo.sh: Más adecuado para **empresas medianas a grandes**, o aquellas que necesitan una solución escalable, con personalizaciones, integraciones de terceros, quizá múltiples entornos (dev/test/prod), con algún equipo técnico o partner que pueda gestionar esas personalizaciones, o con necesidades más específicas de módulo o flujo de negocio. También para empresas que quieren evitar la gestión total de infraestructura (on-premise) pero desean más control que un SaaS puro.
-

## 4. Similitudes y diferencias principales (más allá del precio)

### Similitudes

- Ambos están en la nube — el usuario accede desde Internet sin necesidad de montar localmente la infraestructura (servidores físicos propios) directamente.
- Ambos son propuestas gestionadas por Odoo (en cuanto a hosting, disponibilidad, backups, etc) en mayor o menor medida.
- Ambos facilitan la puesta en marcha más ágil que una instalación on-premise completa.
- Ambos permiten acceder al ERP/CRM de Odoo con módulo estándar (o al menos aplicaciones certificadas) y están pensados para negocio.
- Ambos ofrecen continuidad, soporte de infraestructura (aunque con diferentes niveles) y modelos de actualización.

### Diferencias

- Nivel de personalización: Odoo.sh permite módulos de terceros y código, Odoo Online no.
- Control sobre el entorno: Odoo.sh ofrece entornos dev/test/producción, GitHub, CI/CD, logs, más visibilidad – Odoo Online no.
- Instalación y gestión de infraestructura: en Online prácticamente cero para el cliente, en Odoo.sh aún se gestiona configuración de entorno (workers, almacenamiento, rama de código) aunque Odoo lo soporta.
- Flexibilidad de módulos y código: gran diferencia.
- Escalabilidad técnica / control de rendimiento: Odoo.sh permite escalar recursos, definir número de “workers”, ver logs, etc. En Online tienes menos control en ese sentido. [VentorTech+1](#)
- Versión del software: En Online, generalmente estás sujeto a la versión estable definida por Odoo; en Odoo.sh tienes más control sobre versiones, ramas, etc. [cybrosys.com+1](#)

En resumen: Odoo Online = máximo “manejado”, mínimo intervención; Odoo.sh = más “semi-gestionado”, con mayor control y capacidad de personalización.

---

## 5. En qué se basan técnicamente (infraestructura, contenedores, Git, etc.)

- Odoo.sh: técnica y explícitamente se presenta como un entorno PaaS que incluye: integración con GitHub (repositorio de código), ramas (“branches”), entornos de desarrollo, pruebas automatizadas (CI/CD), logs, shell/SSH en algunos contenedores, backup incremental, replicación de bases de datos, entorno preparado de hosting. Por

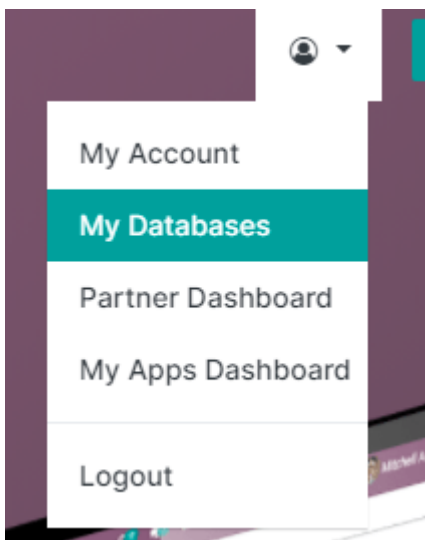
ejemplo, en la web oficial se muestran características como “Database replication”, “Staging servers”, “Your own runbot”, “Shell access” para contenedores. [The Odoo Cloud Platform+1](#)

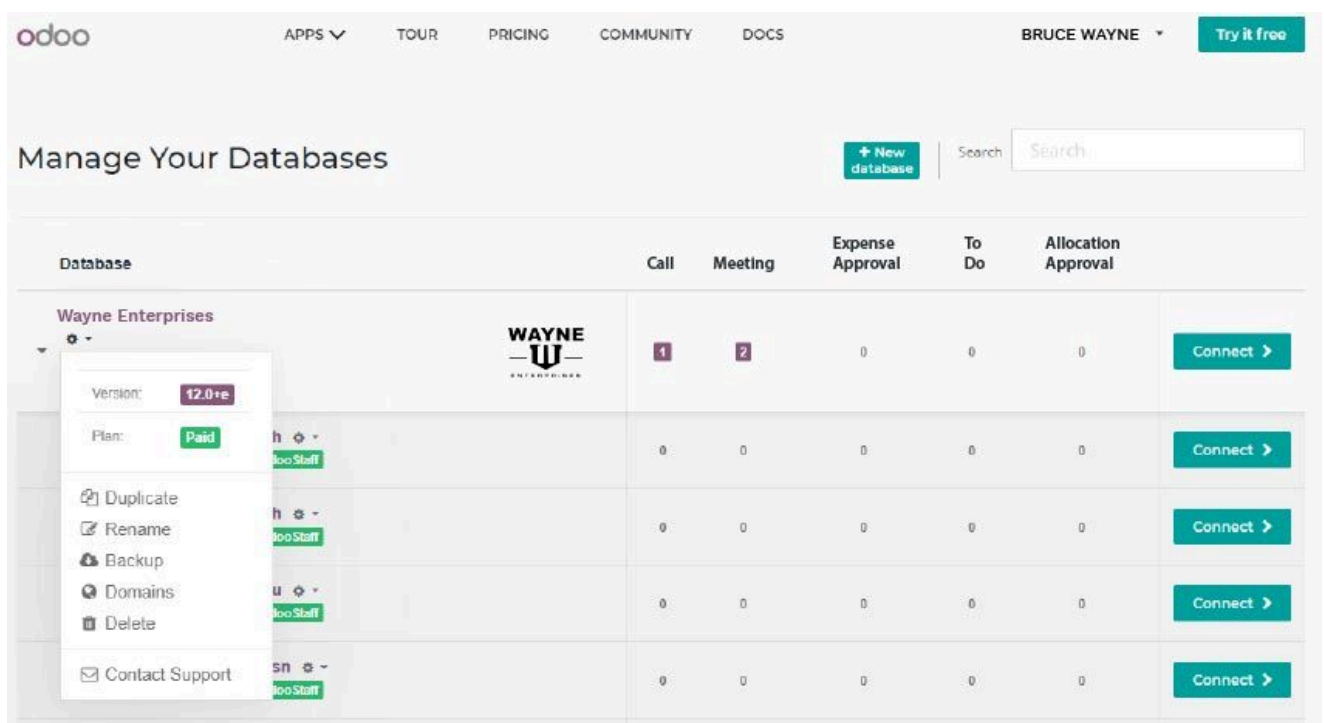
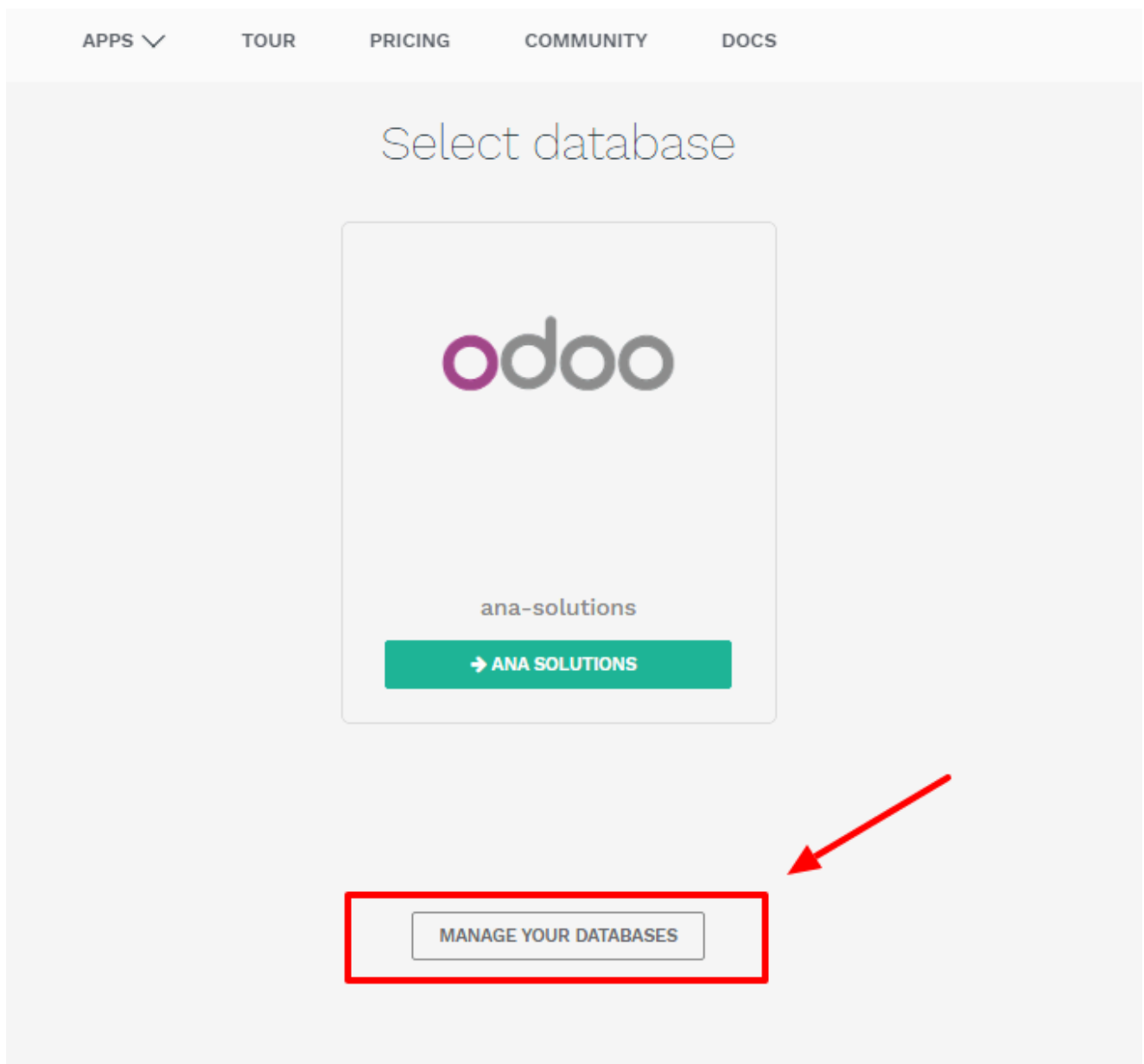
- Esto significa que bajo el capó están contenedores, workers, servicios de ejecución, almacenamiento dedicado, rutas DNS, correo, etc.
  - La arquitectura permite que cada commit o rama del código se despliegue automáticamente, se ejecute test, etc. (modo DevOps).
  - Odoo Online: se basa en un enfoque SaaS puro: infraestructura gestionada por Odoo, backups automáticos y replicación, el cliente no ve la infraestructura, no gestiona servidores, contenedores o workers; fruto de ello se ofrece alta disponibilidad (SLA 99,9 %) y escalabilidad gestionada. [Odoo+1](#)
  - Desde el punto de vista del usuario: en Odoo.sh se puede “tocar” el código, usar Git, hacer módulos personalizados; en Online no.
  - Desde el punto de vista del hosting técnico: ambos probablemente usan virtualización o containers (aunque Odoo no detalla todos los niveles), pero la diferencia real es el grado de control que se da al usuario y la presencia de herramientas de desarrollo (Git, CI/CD, entorno staging) en Odoo.sh.
- 

## 6. Capturas de pantalla de respaldo

Aquí imágenes relevantes:

### Odoo Online





Odoo.sh - Branches | OdooShExample/ExampleRepository | +

https://www.odoo.sh/project/odooexample-example | Zoeken

BETA odoo.sh | Branches | Builds | Status | Settings

odooexample-examplerepository (OdooShExample/ExampleRepository) | OdooShExample

git clone --recurse-submodules --branch master git@github.com:OdooShExample

HISTORY | MAILS | SHELL | LOGS

Today

**OdooShExample** - 15 minutes ago  
FIX: Correct typo in test 7350b83 **Test: Success** ✓ **CONNECT**

**OdooShExample** - 2 hours ago  
INIT: Add demo module with automated test e98c06e **Test: Failed** ❌ **CONNECT**

**GitHub** - 2 hours ago  
Initial commit 7251e0f **Test: Success** ✓ **CONNECT**

Every commit on this branch will be installed, tested and deployed automatically.  
A DNS and mail catcher is setup for you, on odoo.sh, to test outgoing emails.  
Builds in development branches have a short lifespan.

Authorize application | +

GitHub, Inc. (US) | https://github.com/login/oauth/authorize | Zoeken

Authorize Odoo.sh

**Odoo.sh by odoo**  
wants to access your OdooShExample account

**Personal user data**  
Full access

**Repositories**  
Public and private

**Authorize odoo**

Authorizing will redirect to  
<http://www.odoo.sh>

Not owned or operated by GitHub | Created 4 months ago | More than 1K GitHub users

odoo

BranchesBuildsStatusSettingsChangelogDocumentation

main\_live

CloneForkMergeSSHSubmoduleDelete

git clone --recurse-submodules --branch main\_live git@github.com:cybro

Filter branches...

Production

Staging

Development

HISTORYSHELL EDITOR MONITOR LOGS BACKUPS UPGRADE SETTINGS

Prev12345Next

Yesterday

CybroOdooDev - 2 days ago

66e0ce0

Test: Success ✓CONNECT

April 14, 2022

CybroOdooDev - 6 days ago

7d30b7c

Test: Success ✓DROPPED

CybroOdooDev - 6 days ago

2941178

Test: Success ✓DROPPED