

# OpenERP

Raúl Reguillo  
Cristian Del Cerro  
Laura del Río  
Javier García



Escuela  
Superior  
de Informática

- Introducción
- Estado del Arte
- Marco Tecnológico
- Diseño de la Herramienta
- Pruebas
- Conclusiones



- ¿Qué son los “ERP”?



- Bases de ERP

- Optimiza los procesos empresariales

- Acceso a la información

- Compartir toda la información



- Metas de ERP

- ¿Qué es OpenERP?
- Bases de OpenERP
- Optimiza los procesos empresariales
- Acceso a la información
- Compartir toda la información de la organización



- Modular
- Configurable
- Base de Datos Centralizada
- Interacción entre Componentes
- Captura de Datos
- Reingeniería de Procesos



# Introducción – Ventajas vs Desventajas

↑ Personalización a medida

↓ Factor humano

↑ Integración modular de  
componentes

↓ Requisitos

↑ Seguridad

↓ Coste

↓ Cuellos de botella

## SAP ERP

SAP AG es una multinacional del software de origen alemán, reconocida por su implicación en el software empresarial. Los productos SAP se centran en las ERP, siendo SAPERP su producto principal y se divide en las siguientes soluciones: SAP ERP Finanzas, SAP ERP manejo de capital humano y SAP ERP operaciones.



## Desarrollo ABAP

El desarrollo en SAP se basa en los lenguajes C, C++ y ABAP/4 (Advanced Business Application Programming). Este último desarrollado por la misma compañía como lenguaje de alto nivel específico con el que se pueden generar dos tipos de programas ejecutables: informes o módulos para cualquier herramienta SAP.

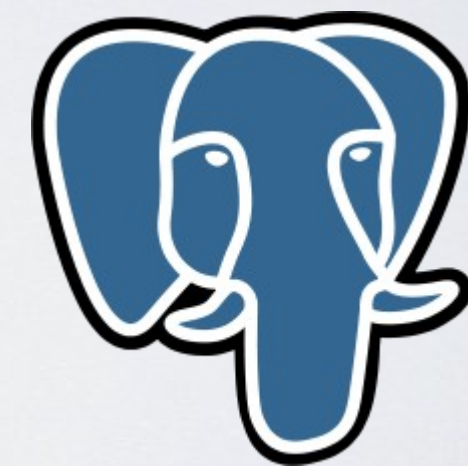
## OpenERP

OpenERP es una alternativa ERP de código abierto y está compuesto por módulos que abarcan ámbitos tales como: administración de ventas, gestión de proyectos, sistemas de gestión de almacenes, contabilidad analítica y financiera, gestión de recursos humanos, gestión de inventario, campañas de marketing, et.

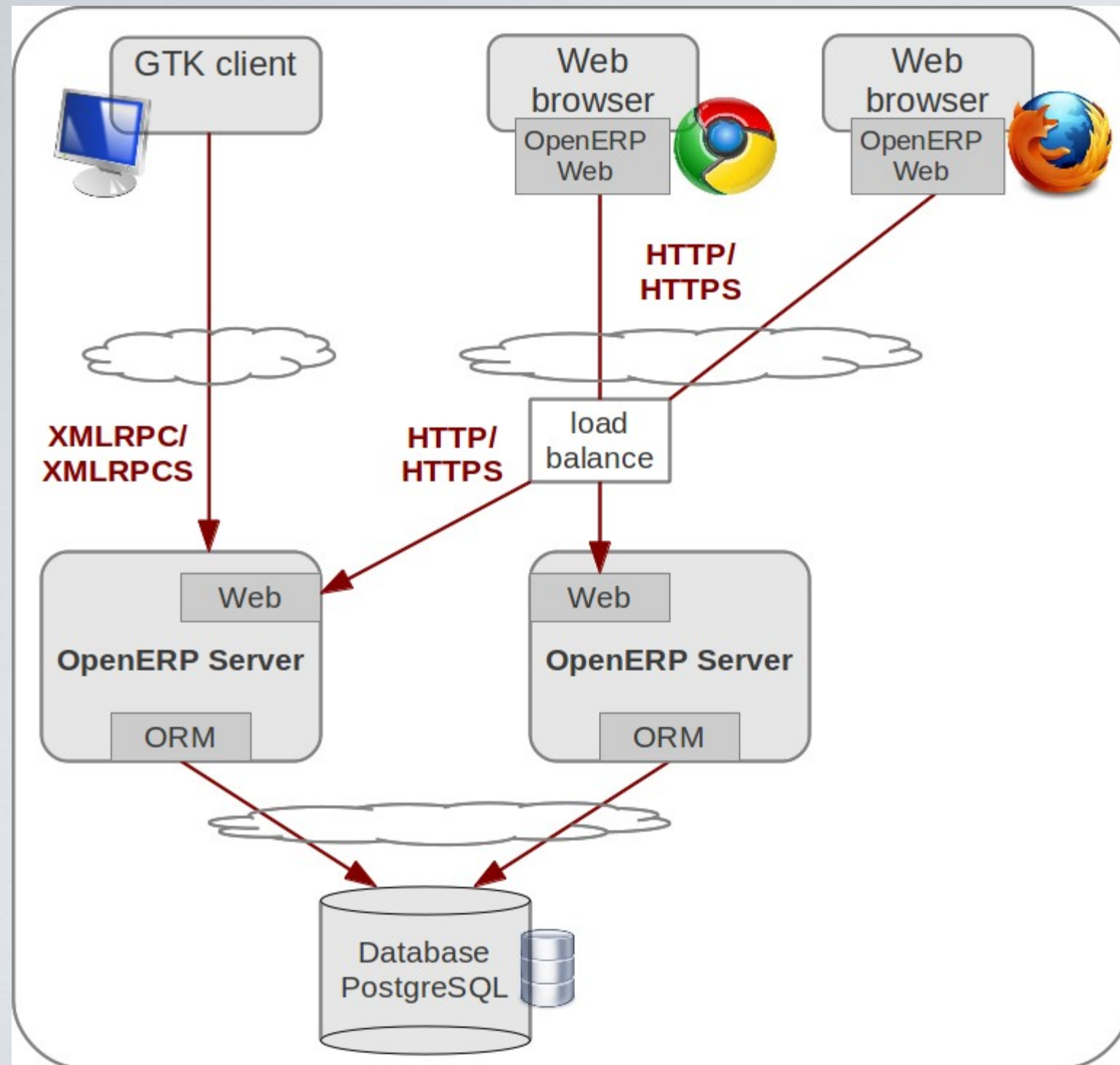


## Desarrollo

OpenERP hace uso de Python como lenguaje de programación y PostgreSQL como sistema gestor de bases de datos. Python es usado tanto para programar módulos como para la propia programación del sistema.







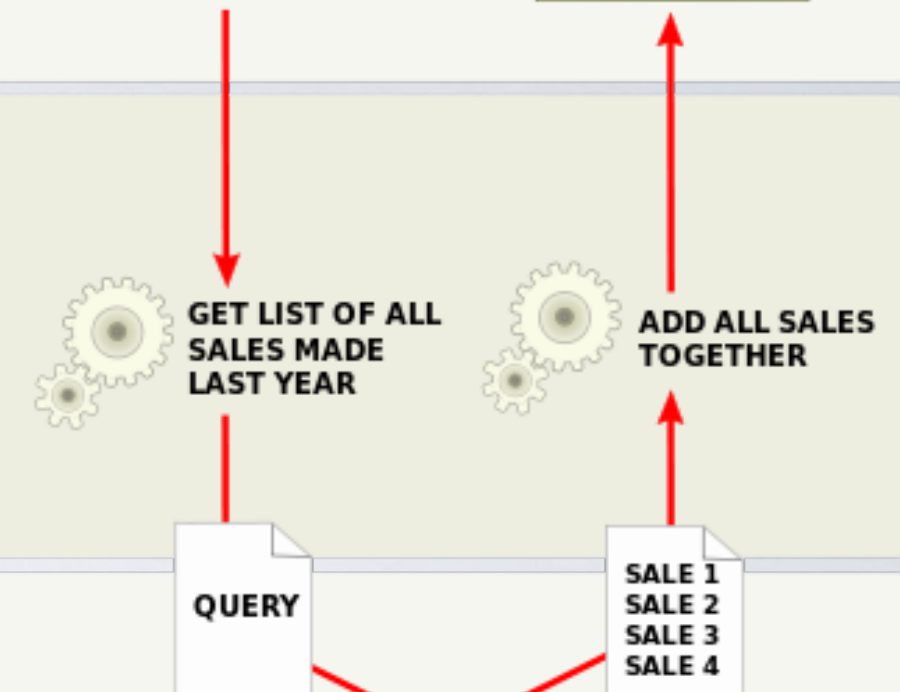
## Presentation tier

The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.



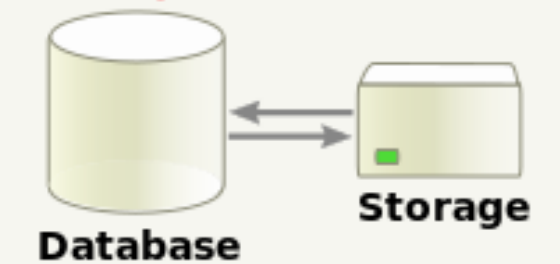
## Logic tier

This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.



## Data tier

Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.



## PostgreSQL

Contiene todas las bases de datos de usará OpenERP, contienen desde los datos de la aplicación hasta la mayoría de los elementos configuración del mismos OpenERP. También existe la posibilidad de utilizar bases de datos distribuidas.



## Servidor OpenERP

Contiene toda la lógica de negocio y se encarga de que OpenERP funcione correctamente. Una capa del servidor se encarga de comunicar con la base de datos PostgreSQL, el motor ORM. Otra capa permite la comunicación entre el servidor y el navegador web, la capa web.

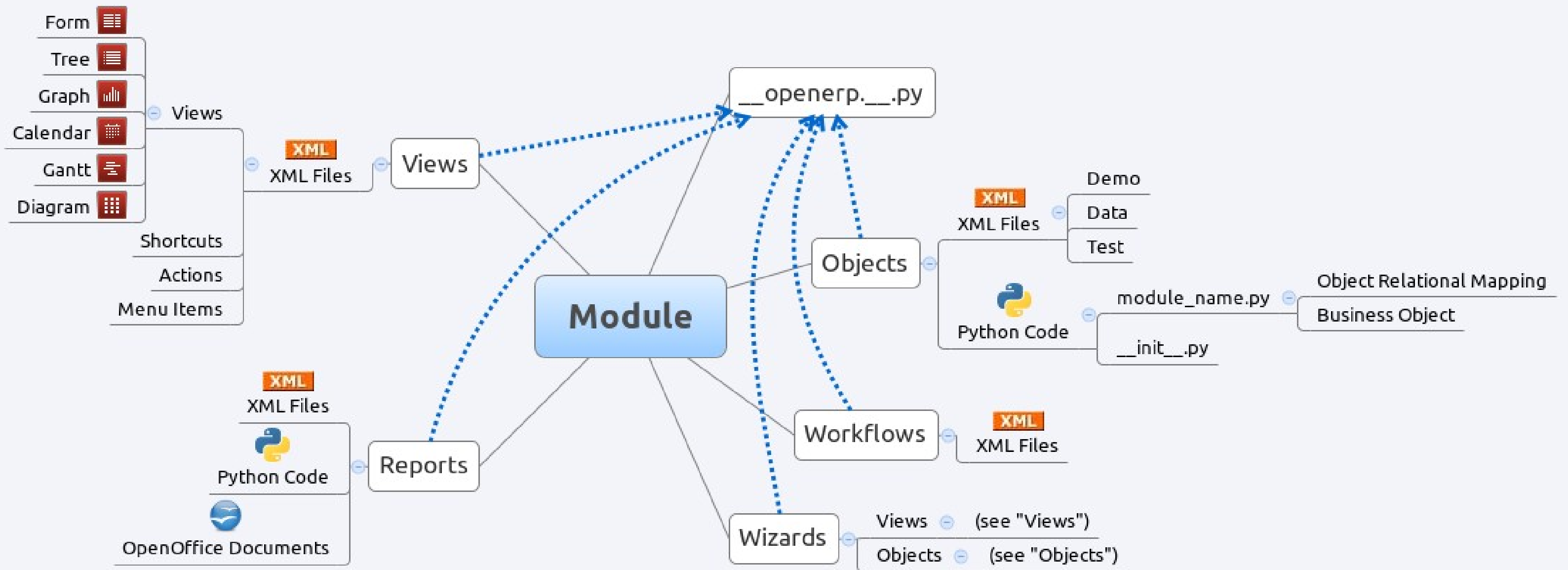


## Aplicación JavaScript

Actúa de cliente ejecutándose en un navegador web.









- `__init__.py`

```
import clientesbugginv2
```

- `__openerp__.py`

```
{  
    ' name ': ' Sistema Gestion Buggin and Debuggin ',  
    ' version ': ' 0.1 ',  
    ' category ': ' Tools ',  
    ' description ': """ Modulo para la gestion de nuestra empresa . """ ,  
    ' author ': ' Buggin and Debuggin ',  
    ' website ': ' www.BugginAndDebuggin.com ',  
    ' depends ': [ ' base '],  
    ' update_xml ': [ ' clientesbugginv2_view . xml '],  
    ' installable ': True ,  
}
```

## ● clientesbugginv2.py

```
from osv import osv, fields
```

```
class cliente(osv.osv):
```

```
    _name = "clientesbugginv2.cliente"
```

```
    _rec_name = 'email'
```

```
    _description = "Clientes"
```

```
    _columns = {
```

```
        'nombre': fields.char('Nombre', size=64),
```

```
        'apellidos': fields.char('Apellidos', size=64),
```

```
        'email': fields.char('Correo electronico', size=64),
```

```
        'telefono': fields.char('Telefono', size=20),
```

```
        'domicilio': fields.char('Domicilio', size=64),
```

```
        'localidad': fields.char('Localidad', size=64),
```

```
        'provincia': fields.many2one('res.country.state', 'Provincia', ondelete='set null'),
```

```
        'pais': fields.many2one('res.country', 'Pais', ondelete='set null')
```

```
    }
```

```
cliente()
```



- clientesbugginv2.py

```
class ventas(osv.osv):  
    _name = "clientesbugginv2.ventas"  
    _description = "Gestion de ventas"  
    _columns = {  
        'fecha': fields.date('Fecha de recepcion', size=64),  
        'cliente': fields.many2one('clientesbugginv2.cliente', 'cliente', ondelete='cascade'),  
        'precio': fields.float('Precio', required=True, digits=(14,4))  
    }  
  
ventas()
```



Escuela  
Superior  
de Informática