

**UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO**  
**FACULTAD DE INGENIERÍA EN CIENCIAS**  
**DE LA COMPUTACIÓN Y TELECOMUNICACIONES**



**2DO PARCIAL**  
**FLASH LEARN IA Y EXAMMASTER**

**INTEGRANTES:**

- |                                 |           |
|---------------------------------|-----------|
| • Sahonero Salas Jose Alejandro | 218048513 |
| • Salazar Vargas Guido          | 220029938 |
| • Valdez Payllo Maily Celina    | 219125041 |

**MATERIA:** Ingeniería de Software 1

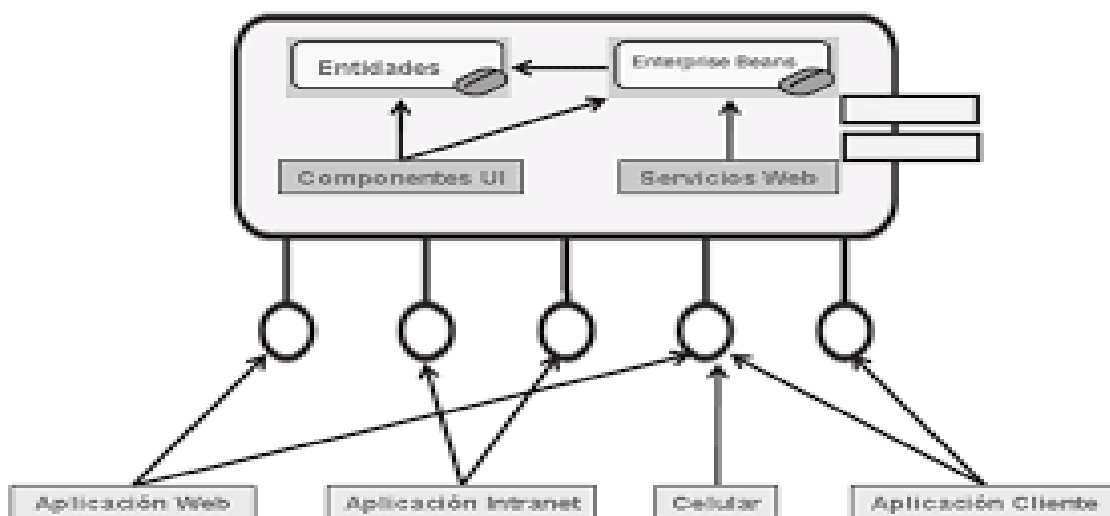
**DOCENTE:** Ing. Martínez Canedo Rolando Antonio

Noviembre - 2024  
Santa Cruz de la Sierra – Bolivia

**CAPÍTULO I**

## 1. Fundamentación Teórica

### 1.1. Desarrollo basado en componentes



El desarrollo basado en componentes consiste en construir aplicaciones mediante ensamblado de módulos software reutilizables, que han sido diseñados previamente independientemente de las aplicaciones en las que van a ser utilizados. Es decir, los componentes pueden ser que los desarrolle otro equipo u otras personas distintas a las que los ensamblan. Incluso suelen utilizarse componentes desarrollados por otras empresas o componentes genéricos.

El modularidad se logra al dividir el sistema en módulos independientes y cohesivos, lo que facilita la comprensión, el mantenimiento y la escalabilidad del software. Cada componente puede ser desarrollado, probado y mejorado de manera independiente, lo que agiliza el proceso de desarrollo y reduce la complejidad del sistema en general.

En cuanto a la eficiencia en el desarrollo, este enfoque permite a los desarrolladores trabajar de forma concurrente en diferentes componentes, lo que acelera el proceso de creación del software. Además, al reutilizar componentes ya probados y optimizados, se reduce la necesidad de desarrollar código desde cero, lo que ahorra tiempo y recursos. En resumen, el enfoque basado en componentes promueve la reutilización de código, la modularidad y la eficiencia en el desarrollo al descomponer el sistema en partes más pequeñas y funcionales, que pueden ser fácilmente compartidas, adaptadas y mejoradas.

### 1.2. Software Libre

«Software libre» es el software que respeta la libertad de los usuarios y la comunidad. A grandes rasgos, significa que los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Es decir, el «software

libre» es una cuestión de libertad, no de precio. En inglés, a veces en lugar de «free software» decimos «libre software», empleando ese adjetivo francés o español, derivado de «libertad», para mostrar que no queremos decir que el software es gratuito.

### **Las cuatro libertades esenciales**

Un programa es software libre si los usuarios tienen las cuatro libertades esenciales:

La libertad de ejecutar el programa como se desee, con cualquier propósito (libertad 0).

La libertad de estudiar cómo funciona el programa, y cambiarlo para que haga lo que se desee (libertad 1). El acceso al código fuente es una condición necesaria para ello.

La libertad de redistribuir copias para ayudar a otros (libertad 2).

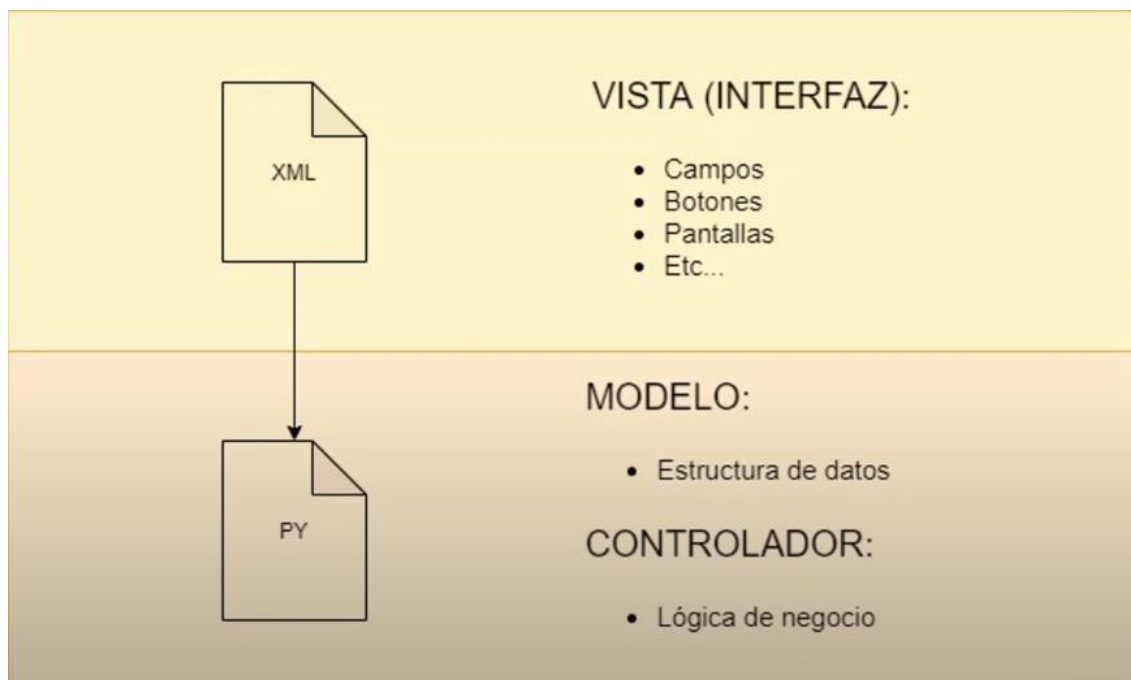
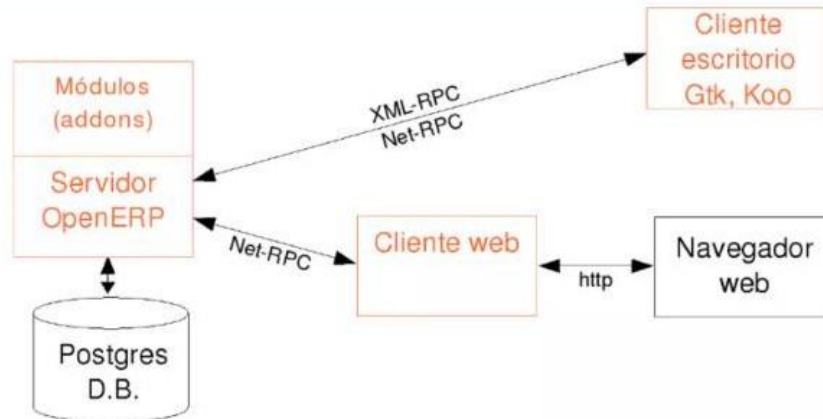
La libertad de distribuir copias de sus versiones modificadas a terceros (libertad 3). Esto le permite ofrecer a toda la comunidad la oportunidad de beneficiarse de las modificaciones. El acceso al código fuente es una condición necesaria para ello.

Un programa es software libre si otorga a los usuarios todas estas libertades de manera adecuada. De lo contrario no es libre. Existen diversos esquemas de distribución que no son libres, y si bien podemos distinguirlos en base a cuánto les falta para llegar a ser libres, nosotros los consideramos contrarios a la ética a todos por igual.

En cualquier circunstancia, estas libertades deben aplicarse a todo código que pensemos utilizar hacer que otros utilicen. Tomemos por ejemplo un programa A que automáticamente ejecuta un programa B para que realice alguna tarea. Si se tiene la intención de distribuir A tal cual, esto implica que los usuarios necesitarán B, de modo que es necesario considerar si tanto A como B son libres. No obstante, si se piensa modificar A para que no haga uso de B, solo A debe ser libre; B no es relevante en este caso.

### 1.3.Desarrollo Arquitectura del Software

## Arquitectura



Odoo sigue una arquitectura de varios niveles, lo que significa que la presentación, lógica empresarial y el almacenamiento de datos están separados. El nivel de presentación es una combinación de HTML5, JavaScript y CSS. El nivel lógico está escrito exclusivamente en Python, mientras que el nivel de datos solo admite PostgreSQL como RDBMS. Dependiendo del alcance de su módulo, el desarrollo de Odoo se puede realizar en cualquiera de estos niveles.

Tanto las extensiones de servidor como de cliente, están empaquetadas como módulos que se cargan opcionalmente en una base de datos.

Los módulos de Odoo pueden agregar una nueva lógica empresarial a un sistema Odoo o alterar y ampliar la lógica existente. Se puede crear un módulo para agregar las reglas de contabilidad de su país al soporte de contabilidad genérico de Odoo, mientras que un módulo diferente puede agregar soporte para la visualización en tiempo real de una flota de autobuses. En Odoo, el comienzo y el fin es mediante módulos.

#### **1.4.Odoo**

##### **1.4.1. Versiones**

Odoo lanza una nueva versión mayor anualmente. Algunas versiones importantes son:

**Odoo 8:** Primer gran lanzamiento con muchas aplicaciones nuevas.

**Odoo 9:** Introducción de la edición Enterprise.

**Odoo 10:** Mejora en la interfaz de usuario y nuevas aplicaciones.

**Odoo 11:** Mejoras de rendimiento y nuevas características.

**Odoo 12:** Mayor enfoque en la interfaz de usuario y nuevas funcionalidades.

**Odoo 13:** Introducción de aplicaciones como el módulo de eLearning y mejoras en la gestión de inventarios.

**Odoo 14:** Lanzado en octubre de 2020, introdujo mejoras en rendimiento y nuevas aplicaciones como el constructor de sitios web.

**Odoo 15:** Lanzado en octubre de 2021, incluyó mejoras en el módulo de CRM, POS, y nuevas aplicaciones como la hoja de tiempo.

**Odoo 16:** Lanzado en octubre de 2022, enfocado en la mejora de la experiencia de usuario y funcionalidades avanzadas para manufactura y comercio electrónico.

**Odoo 17:** La última versión, con mejoras en la UI/UX, rendimiento y nuevas funcionalidades.

##### **1.4.2. Ediciones**

**Odoo Community Edition (CE):** Versión gratuita y de código abierto. Incluye funcionalidades básicas y es mantenida por la comunidad.

**Odoo Enterprise Edition (EE):** Versión de pago con funcionalidades avanzadas y soporte técnico oficial de Odoo S.A. Incluye características adicionales no presentes en la versión Community.

##### **1.4.3. Arquitectura del Software (Odoo)**

Odoo (anteriormente OpenERP) es un software de planificación de recursos empresariales (ERP) de código abierto que tiene una arquitectura modular y flexible. Algunos aspectos clave de la arquitectura de Odoo incluyen:

1. **Diseño modular:** Odoo se construye utilizando una arquitectura modular, donde cada área funcional (p. ej., contabilidad, inventario, CRM) se implementa como un módulo separado. Esto permite a los usuarios instalar solo los módulos que necesitan, lo que hace que el sistema sea ligero y personalizable.
2. **Patrón Modelo-Vista-Controlador (MVC):** Odoo sigue el patrón arquitectónico Modelo-Vista-Controlador (MVC), que separa la lógica de la aplicación en tres componentes interconectados: el modelo (acceso a datos), la vista (interfaz de usuario) y el controlador (lógica de la aplicación).
3. **Diseño orientado a objetos:** El núcleo de Odoo se construye utilizando un enfoque orientado a objetos, donde todas las entidades empresariales se representan como objetos con sus propios atributos y comportamientos.
4. **Arquitectura basada en la web:** Odoo es una aplicación web a la que se puede acceder a través de un navegador web. Esto permite una implementación sencilla y compatibilidad multiplataforma.
5. **Extensible y personalizable:** El diseño modular de Odoo y el uso de Python y XML hacen que el sistema sea altamente extensible y personalizable. Los desarrolladores pueden crear nuevos módulos o modificar los existentes para satisfacer requisitos empresariales específicos.
6. **Capa de base de datos:** Odoo utiliza PostgreSQL como su base de datos principal, lo que proporciona características avanzadas y escalabilidad para aplicaciones empresariales.
7. **Arquitectura multiinquilino:** Odoo admite una arquitectura multiinquilino, lo que permite a varios clientes compartir la misma instancia de Odoo mientras mantienen el aislamiento y la seguridad de los datos.
8. **Internacionalización y localización:** Odoo está diseñado para ser internacionalizado, con soporte para varios idiomas, monedas y regulaciones contables y fiscales regionales.

#### 1.4.4. Estructura

**Core:** Contiene los módulos fundamentales que proporcionan la funcionalidad básica del sistema.

**Add-ons:** Módulos adicionales que pueden ser instalados para ampliar la funcionalidad del sistema.

**Database:** Almacena los datos del sistema, generalmente utilizando PostgreSQL.

**ORM (Object-Relational Mapping):** Permite interactuar con la base de datos utilizando objetos Python.

**Business Logic:** Contiene las reglas de negocio y lógica aplicativa.

**Views:** Definiciones de vistas (formularios, listas, gráficos) que determinan cómo se presentan los datos a los usuarios.

**Controllers:** Gestionan las interacciones entre las vistas y los modelos.

**API:** Odoo ofrece una API que permite interactuar con el sistema desde aplicaciones externas.

#### 1.4.5. Store

Odoo Store es la plataforma oficial para la adquisición de aplicaciones y módulos adicionales para Odoo, un sistema de gestión empresarial (ERP) ampliamente utilizado. Odoo es conocido por su flexibilidad y por ser altamente personalizable, permitiendo a las empresas adaptar el software a sus necesidades específicas. La tienda Odoo Store facilita este proceso al proporcionar un amplio catálogo de aplicaciones desarrolladas tanto por la comunidad de usuarios como por Odoo S.A., la compañía detrás de Odoo.

En Odoo Store, los usuarios pueden encontrar aplicaciones que cubren una variedad de funcionalidades, como la contabilidad, la gestión de inventarios, la ventas, el marketing, la gestión de proyectos, entre otras. Estas aplicaciones están diseñadas para integrarse sin problemas con el sistema principal de Odoo, ofreciendo características adicionales que pueden ser necesarias para optimizar los procesos empresariales.

Una de las ventajas de Odoo Store es que permite a las empresas explorar y probar diferentes módulos antes de realizar una compra, facilitando así una toma de decisiones informada. Además, las aplicaciones disponibles en Odoo Store suelen ser actualizadas regularmente para mantener la compatibilidad con las versiones más recientes de Odoo, asegurando que las empresas puedan beneficiarse de las últimas mejoras y correcciones de errores.

#### 1.4.6. Estandarización para módulos nuevos

## 1. Estructura del Módulo:

- `__init__.py`: Inicializa el módulo.
- `__manifest__.py`: Archivo de configuración del módulo, define las dependencias, datos, y vistas.
- `models/`: Contiene los modelos de datos (archivos Python).
- `views/`: Contiene las vistas XML.
- `security/`: Define los permisos de acceso.
- `data/`: Archivos de datos que deben ser cargados, como datos demo.
- `static/`: Archivos estáticos como CSS y JS.

## 2. Definición de Modelos (models):

```
python
from odoo import models, fields, api

class MyModel(models.Model):
    __name = 'my.module'
    __description = 'My Module'

    name = fields.Char(string='Name',
        required=True)
    description =
        fields.Text(string='Description')
```

## 3. Definición de Vistas (views):

```
xml
<odoo>
    <record id="view_my_module_form" model="ir.ui.view">
        <field name="name">my.module.form</field>
        <field name="model">my.module</field>
        <field name="arch" type="xml">
            <form>
                <group>
                    <field name="name"/>
                    <field name="description"/>
                </group>
            </form>
        </field>
    </record>
```



```
</group>
</form>
</field>
</record>
</odoo>
```

#### 4. Seguridad (security):

csv

```
id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
access_my_module,my.module,model_my_module,base.group_user,1,1,1,1
```

#### 1.4.7. Como ser socio de Odoo

Para convertirse en un partner (dealer) de Odoo, se deben seguir estos pasos:

##### 1. Contactar a Odoo:

- Rellenar el formulario de solicitud en el sitio web de Odoo.

##### 2. Elegir un Nivel de Asociación:

- **Ready Partner:** Para empresas que comienzan a trabajar con Odoo.
- **Silver Partner:** Para empresas con experiencia y un historial de implementación de Odoo.
- **Gold Partner:** Para empresas con una experiencia significativa y un gran volumen de implementaciones exitosas.

##### 3. Cumplir con los Requisitos:

- **Capacitación:** Completar la formación requerida por Odoo.
- **Clientes:** Tener una cantidad mínima de implementaciones exitosas.
- **Certificaciones:** Obtener las certificaciones necesarias.

##### 4. Firmar el Contrato de Partner:

- Firmar un contrato de partnership con Odoo.

## 1.5.ORM

Odoo utiliza su propio ORM (Object-Relational Mapping), que permite a los desarrolladores interactuar con la base de datos de manera abstracta. Este ORM facilita la creación, actualización y eliminación de registros en la base de datos sin necesidad de escribir consultas SQL directamente.

### Características del ORM de Odoo:

**Declaración de Modelos:** Los modelos se declaran como clases en Python.

**Campos:** Diversos tipos de campos como Char, Integer, Float, Many2one, One2many, etc.

**APIs:** Decoradores como @api.model, @api.multi, @api.depends para gestionar la lógica del negocio y las dependencias.

**Herencia:** Soporte para herencia clásica y polimórfica.

**Composición:** Relaciones entre modelos usando claves foráneas (Many2one) y colecciones (One2many, Many2many).

## 1.6.Docker

El software de TI de Docker es tecnología en contenedores que permite la creación y el uso de contenedores Linux.

La comunidad Docker de open source trabaja para mejorar estas tecnologías y beneficiar a todos los usuarios de forma gratuita.

La empresa, Docker Inc., desarrolla el trabajo de la comunidad Docker, lo hace más seguro y comparte estos avances con el resto de la comunidad. También respalda las tecnologías mejoradas y reforzadas para clientes empresariales.

Con DOCKER, usted podrá utilizar los contenedores como máquinas virtuales muy livianas y modulares, y obtendrá tal flexibilidad que podrá crearlos, implementarlos, copiarlos y pasarlos de un entorno a otro.

### ¿Cómo funciona Docker?

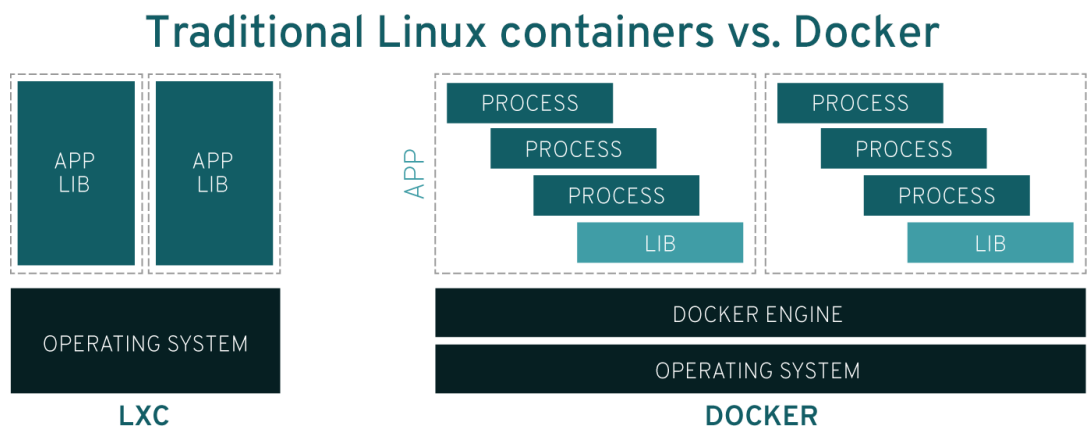
La tecnología Docker usa el kernel de Linux y funciones de este, como Cgroups y namespaces, para segregar procesos y que puedan ejecutarse de manera independiente. Esta independencia es la intención de los contenedores, la capacidad de ejecutar varios procesos y aplicaciones separados unos de los otros para hacer un mejor uso de su infraestructura, y mantener la seguridad que tendría con sistemas separados.

Las herramientas del contenedor, como Docker, ofrecen un modelo de implementación basado en imágenes.

Esto permite a la aplicación en un conjunto de servicios, con todas sus dependencias en varios entornos. Docker también automatiza la implementación de la aplicación (o los conjuntos combinados de procesos que constituyen una aplicación) en este entorno de contenedores.

Estas herramientas creadas a partir de los contenedores Linux (lo cual los vuelve únicos y fáciles de usar) les otorgan a los usuarios un acceso sin precedentes a las aplicaciones, la habilidad de implementar rápidamente, controlar las versiones y su distribución.

Los contenedores de Linux tradicionales usan un sistema init que puede gestionar varios procesos. Esto significa que aplicaciones enteras pueden ejecutarse como una. La tecnología Docker pretende que las aplicaciones se dividan en sus procesos separados y ofrece las herramientas para hacerlo. Este enfoque granular tiene sus ventajas.



#### 1.6.1. Ventajas

##### Modularidad

El enfoque Docker para la creación de contenedores se centra en la capacidad de tomar una parte de una aplicación, actualizarla o repararla, sin tomar innecesariamente toda la aplicación. Además de estos enfoques basados en microservicios, puede compartir procesos entre varias aplicaciones de la misma forma que funciona la arquitectura orientada a servicios (SOA).

##### Capas y Control de Versión de la Imagen

Cada archivo de imagen de Docker está conformado por una serie de capas. Estas capas se combinan en una única imagen. Una capa se crea cuando la imagen cambia. Cada vez que un usuario especifica un comando, como ejecutar o copiar, se crea una nueva capa.

Docker reutiliza estas capas para construir nuevos contenedores. Lo cual hace mucho más rápido el proceso de construcción. Los cambios intermedios se comparten

entre imágenes, mejorando aún más la velocidad, el tamaño y la eficiencia. El control de versión es inherente a la creación de capas. Cada vez que hay un cambio nuevo, existe básicamente un registro de cambios integrado con control total de las imágenes de contenedor.

## **Restauración**

Probablemente la mejor parte de la creación de capas es la capacidad de restaurar. Toda imagen tiene capas. ¿No le gusta la iteración actual de una imagen? Restáurela a la versión anterior. Esto soporta un enfoque de desarrollo ágil y ayuda a hacer realidad la integración e implementación continuas (CI/CD) desde una perspectiva de herramientas.

## **Implementación Rápida**

Solía demorar días desarrollar un nuevo hardware, ejecutarlo, proveer y facilitar. Y el nivel de esfuerzo y sobrecarga era extenuante. Los contenedores basados en Docker pueden reducir el tiempo de implementación a segundos. Al crear un contenedor para cada proceso, puede compartir rápidamente los procesos similares con nuevas aplicaciones.

Y, debido a que un Sistema Operativo no necesita iniciarse para agregar o mover un contenedor. Los tiempos de implementación son sustancialmente inferiores. Asimismo, con la velocidad de implementación, puede crear y borrar datos creados por sus contenedores sin preocupación, de forma fácil y accesible.

Entonces, la tecnología Docker es un enfoque más granular y controlable, basado en microservicios, que coloca mayor valor en la eficiencia.

## **¿Existen Desventajas en el uso de Docker?**

En sí mismo, Docker es muy bueno para gestionar contenedores únicos.

Cuando comienza a usar cada vez más contenedores y aplicaciones en contenedores. Es muy difícil hacer la separación en cientos de piezas, la administración y la orquestación. Finalmente, debe retroceder y agrupar los contenedores para proporcionar servicios (de red, de seguridad, de telemetría, etc.) en todos los contenedores. Aquí es cuando aparecen los Kubernetes.

Con Docker no se obtiene la misma funcionalidad tipo UNIX que se obtiene con los contenedores Linux tradicionales. Por ejemplo, no se obtiene la capacidad para usar procesos como cron o syslog en el contenedor, junto con la aplicación. Del mismo modo, existen ciertos límites, como la limpieza de los procesos después de terminar con los procesos hijo.

## **Evolución y Mejora Continua**

Asimismo, existen otros subsistemas y dispositivos de Linux que no tienen espacios de nombres. Estos incluyen SELinux, Cgroups y dispositivos `/dev/sd*`. Esto implica que, si un atacante obtiene control de estos subsistemas, el host se ve comprometido. Para permanecer liviano, compartir el kernel del host con contenedores abre la posibilidad de vulnerabilidad. Esto difiere de las máquinas virtuales, las cuales están fuertemente separadas del sistema host.

El daemon del Docker también puede ser una preocupación en materia de seguridad. Para usar y ejecutar los contenedores Docker, es muy probable que utilice el daemon del Docker, un tiempo de ejecución persistente para contenedores. El daemon del Docker requiere privilegios de acceso a la raíz del sistema.

Por lo que se debe prestar especial atención a quiénes obtienen acceso a este proceso y en dónde reside este. Por ejemplo, un daemon local tiene una superficie de ataque más pequeña que uno que se encuentra en un sitio más público.

### **1.7. XML**

El lenguaje de marcado extensible (XML) permite definir y almacenar datos de forma compartible. XML admite el intercambio de información entre sistemas de computación, como sitios web, bases de datos y aplicaciones de terceros. Las reglas predefinidas facilitan la transmisión de datos como archivos XML a través de cualquier red, ya que el destinatario puede usar esas reglas para leer los datos de forma precisa y eficiente.

El lenguaje de marcado extensible (XML) es un lenguaje de marcado que proporciona reglas para definir cualquier dato. A diferencia de otros lenguajes de programación, XML no puede realizar operaciones de computación por sí mismo. En cambio, se puede implementar cualquier software o lenguaje de programación para la administración estructurada de datos.

#### **Beneficios**

##### **Respaldo para las transacciones interempresariales**

Cuando una empresa vende un bien o servicio a otra empresa, las dos empresas necesitan intercambiar información como el costo, las especificaciones y los plazos de entrega. Con el lenguaje de marcado extensible (XML), pueden compartir toda la información necesaria electrónicamente y cerrar negocios complejos de forma automática, sin intervención humana.

##### **Conservación de la integridad de los datos**

XML le permite transferir datos junto con la descripción de los datos, lo que evita la pérdida de la integridad de los datos. Puede usar esta información descriptiva para hacer lo siguiente:

**Verificar la precisión de los datos.**

Personalizar automáticamente la presentación de datos para diferentes usuarios.

Almacenar datos de forma coherente en múltiples plataformas.

**Mejora de la eficiencia de búsqueda**

Los programas de computación, como los motores de búsqueda, pueden ordenar y categorizar archivos XML de forma más eficiente y precisa que otros tipos de documentos. Por ejemplo, la palabra marca puede ser un sustantivo o un verbo.

Basándose en las etiquetas XML, los motores de búsqueda pueden categorizar con precisión marca para resultados de búsqueda relevantes. Por lo tanto, XML ayuda a las computadoras a interpretar el lenguaje natural de manera más eficiente.

**Diseño de aplicaciones flexibles**

Con XML, puede actualizar o modificar cómodamente el diseño de su aplicación. Muchas tecnologías, especialmente las más nuevas, vienen con compatibilidad con XML incorporada. Pueden leer y procesar automáticamente los archivos de datos XML para que pueda realizar cambios sin tener que volver a formatear toda la base de datos.

**1.1. *Python***

En términos técnicos, Python es un lenguaje de programación de alto nivel, orientado a objetos, con una semántica dinámica integrada, principalmente para el desarrollo web y de aplicaciones informáticas.

Es muy atractivo en el campo del Desarrollo Rápido de Aplicaciones (RAD) porque ofrece tipificación dinámica y opciones de encuadernación dinámicas.

Python es relativamente simple, por lo que es fácil de aprender, ya que requiere una sintaxis única que se centra en la legibilidad. Los desarrolladores pueden leer y traducir el código Python mucho más fácilmente que otros lenguajes.

Por tanto, esto reduce el costo de mantenimiento y de desarrollo del programa porque permite

que los equipos trabajen en colaboración sin barreras significativas de lenguaje y experimentación.

Además, soporta el uso de módulos y paquetes, lo que significa que los programas pueden ser diseñados en un estilo modular y el código puede ser reutilizado en varios proyectos. Una vez se ha desarrollado un módulo o paquete, se puede escalar para su uso en otros proyectos, y es fácil de importar o exportar.

En definitiva, es un lenguaje de programación relativamente fácil de aprender, y las herramientas necesarias están disponibles para todos de forma gratuita. Esto hace que sea accesible para casi todo el mundo. Si dispones de tiempo para aprender, conseguirás crear esos proyectos que tienes en mente

### **¿Para qué se usa y qué se puede programar con Python?**

Python es un lenguaje de programación de propósito general, que es otra forma de decir que puede ser usado para casi todo. Lo más importante es que se trata de un lenguaje interpretado, lo que significa que el código escrito no se traduce realmente a un formato legible por el ordenador en tiempo de ejecución.

Este tipo de lenguaje también se conoce como «lenguaje de scripting» porque inicialmente fue pensado para ser usado en proyectos sencillos.

El concepto de «lenguaje de scripting» ha cambiado considerablemente desde su creación, porque ahora se utiliza Python para programar grandes aplicaciones de estilo comercial, en lugar de sólo las simples aplicaciones comunes.

En definitiva, sabes que el lenguaje debe ser realmente importante cuando se encarga de impulsar un sistema bursátil. De hecho, la NASA lo utiliza cuando programan sus equipos y maquinaria espacial.

También, puede ser usado para procesar texto, mostrar números o imágenes, resolver ecuaciones científicas y guardar datos.

En resumen, se utiliza entre bastidores para procesar un montón de elementos que podrías necesitar o encontrar en tu(s) dispositivo(s), incluido el móvil.

## ¿Cómo funciona Python?

El lenguaje de programación Python utiliza módulos de código que son intercambiables en lugar de una larga lista de instrucciones que era estándar para los lenguajes de programación funcional.

### **El intérprete de Python realiza las siguientes tareas para ejecutar un programa:**

**Paso 1:** El intérprete lee un código o instrucción python. Luego verifica que la instrucción esté bien formateada, es decir, comprueba la sintaxis de cada línea. Si encuentra algún error, detiene inmediatamente la traducción y muestra un mensaje de error.

**Paso 2:** Si no hay ningún error, es decir, si la instrucción o el código python está bien formateado, el intérprete lo traduce a su forma equivalente en un lenguaje intermedio llamado «código Byte». Así, después de la ejecución exitosa de la escritura del código python, se traduce completamente en código Byte.

**Paso 3:** El código del byte se envía a la Máquina Virtual Python, donde de nuevo se ejecuta el código del byte en PVM. Si se produce un error durante esta ejecución, ésta se detiene con un mensaje de error.

## **1. Introducción**

### **1.1 Descripción del Problema**

### **1.2 Objetivo General**

Desarrollar un software innovador que transforme la experiencia de estudio en módulos educativos, mejorando la automatización educativa. Este sistema tiene como objetivo revolucionar la forma en que los estudiantes puedan desarrollar sus capacidades de aprendizaje de una forma más eficiente y eficaz, donde los padres de familia puedan estar informados de las actividades educativas por parte del colegio y de las reuniones o actividades académicas.

### **1.3 Objetivo Especifico**

- Analizar exhaustivamente las necesidades de los módulos educativos y los módulos que desempeñan para la educación y formación de los estudiantes.
- Identificar las principales dificultades que enfrentan los colegios al inicio de la gestión escolar y su desarrollo académico.
- Diseñar una arquitectura de software flexible y escalable que permita la integración fluida de todas las funcionalidades requeridas.
- Implementar soluciones innovadoras basadas en tecnologías de procesamiento con IA para automatizar el proceso y Análisis de documentos para fines académicos.
- Incorporar herramientas avanzadas de automatización y digitalización



- Realizar pruebas exhaustivas de usabilidad y rendimiento para garantizar que el software cumpla con los requerimientos y satisfaga las necesidades de todos los usuarios que interactúen con el sistema

## **CAPITULO II**

### **2. Fundamentación Teórica**

#### **2.1 Metodología Scrum**

Scrum, un proceso ágil ampliamente utilizado en el desarrollo de productos, especialmente en el ámbito del software, destaca por su enfoque en plazos ajustados, requisitos complejos y singularidad. El equipo trabaja en ciclos cortos, denominados sprints, para entregar productos de alta calidad. Fundamentado en transparencia, inspección y adaptación, Scrum promueve la colaboración entre los miembros del equipo.

##### **2.1.1. cuando se utiliza?**

Scrum resulta ideal en proyectos de desarrollo de software que demandan flexibilidad y adaptabilidad a cambios. También es efectivo en proyectos con incertidumbre y complejidad, donde los requisitos no están definidos claramente. Además, es una opción sólida para equipos distribuidos geográficamente, facilitando la colaboración y comunicación continua.

##### **2.1.2. El equipo Scrum**

El equipo Scrum es esencial y se compone de tres roles principales: Dueño de Producto, Equipo de Desarrollo y Scrum Master. El Dueño de Producto maximiza el valor del producto, el Equipo de Desarrollo crea incrementos terminados en cada sprint, y el Scrum Master asegura el seguimiento del proceso y elimina obstáculos. Todos son multifuncionales y autoorganizados, permitiendo colaboración efectiva para alcanzar objetivos.

##### **2.1.3. Eventos del Scrum**

Scrum establece eventos predefinidos para impulsar la regularidad en el desarrollo. Además del Sprint, que es el contenedor principal, incluye eventos clave como Planificación del Sprint, Reunión Diaria de Scrum, Revisión del Sprint y Retrospectiva del Sprint. Cada evento tiene una duración máxima y un objetivo específico para maximizar la transparencia y la adaptación en el proceso. La falta de alguno de estos eventos puede reducir la transparencia y desaprovechar oportunidades para inspeccionar y adaptarse.

##### **2.1.4. Arquitectura de Odoo**

Odoo, anteriormente conocido como OpenERP, es un software de planificación de recursos empresariales (ERP) de código abierto, con una arquitectura modular y flexible que permite

adaptar el sistema a las necesidades de cualquier organización. Algunos aspectos clave de su arquitectura son:

**Diseño Modular:** La arquitectura modular de Odoo permite que cada área funcional (como contabilidad, inventario, CRM, etc.) esté implementada en módulos separados. Los usuarios pueden instalar solo los módulos que necesitan, haciendo que el sistema sea más ligero y adaptable.

**Patrón Modelo-Vista-Controlador (MVC):** Odoo sigue el patrón de arquitectura MVC, que separa la lógica de la aplicación en tres componentes interconectados:

**Modelo:** Gestiona el acceso y almacenamiento de datos.

**Vista:** Representa la interfaz de usuario.

**Controlador:** Contiene la lógica de la aplicación que gestiona las interacciones del usuario y las operaciones de negocio.

**Diseño Orientado a Objetos:** El núcleo de Odoo se construye con un enfoque orientado a objetos, donde todas las entidades de negocio se representan como objetos con atributos y comportamientos específicos.

**Arquitectura Basada en la Web:** Odoo es una aplicación basada en la web, accesible a través de cualquier navegador. Esto facilita la implementación y la compatibilidad en diferentes plataformas.

**Extensible y Personalizable:** Gracias a su diseño modular y el uso de Python y XML, Odoo es altamente extensible y personalizable. Los desarrolladores pueden crear nuevos módulos o modificar los existentes para adaptarse a requisitos específicos.

**Capa de Base de Datos:** Odoo utiliza PostgreSQL como base de datos principal, proporcionando funciones avanzadas y escalabilidad para aplicaciones a nivel empresarial.

**Arquitectura Multi-tenant:** Odoo admite una arquitectura multi-tenant (multiusuario), permitiendo que múltiples clientes compartan la misma instancia de Odoo mientras se mantienen la seguridad y el aislamiento de los datos.

**Internacionalización y Localización:** Odoo está diseñado para ser internacionalizado, con soporte para múltiples idiomas, monedas y normativas fiscales y contables regionales.

En general, la arquitectura de Odoo está diseñada para ser flexible, escalable y personalizable, lo que la convierte en una opción popular para empresas de todos los tamaños y sectores.

### **3. Alcance**

#### **3.1 Gestionar Usuarios**

#### **3.2 Gestionar Administrativo**

#### **3.3 Gestionar Académica**

- 3.4 Gestionar Curso
- 3.5 Gestionar Estudiantes
- 3.6 Visualizar el numero de Inscritos por nivel
- 3.7 Gestión Grados
- 3.8 Visualizar horarios de materias con el docente
- 3.9 Visualizar padres u apoderados
- 3.10 Gestionar Roles
- 3.11 Gestionar Comunicados
- 3.12 Gestionar Actividades Académicas
- 3.13 Gestionar Asistencia
- 3.14 Gestionar Eventos
- 3.15 Gestionar matricula
- 3.16 Gestionar Materias
- 3.17 Gestionar pagos
- 3.18 Gestionar notificaciones
- 3.19 Gestionar documentos
- 3.20 Gestión de Educativa

## 4. Sprint 1

### 4.1 Sprint Backlog

Product Backlog				
Proyecto		Software para la Generación de Subtítulos y Traducción		
Product Owner		Guido Salazar Vargas		
Versión		1.0	Fecha	18-04-2022
ID	Rol	Funcionalidad	Razón/Resultado	P
1	Administrador	Gestión de Usuarios	Gestionar los usuarios que interactúan con el sistema.	Alta
2	Administrador	Gestión de Profesores	Los Profesores tienen sus propios accesos dentro del sistema .	Alta
3	Administrador	Gestión de Alumnos	Los alumnos cumplen su función dentro del sistema.	Alta
4	Profesor	Servicio de Carga de Archivo	Subir un archivo para ser procesado por el sistema.	Media
5	Profesor	Servicio de Creación de Exámenes	Automatizar la creación de exámenes.	Media

6	Profesor	Servicio de Personalización de Exámenes	La personalización de los exámenes se ajusta al contenido del docente	Media
7	Profesor	Servicio de dificultad de exámenes	Los docentes pueden definir la dificultad de las preguntas	Media
8	Profesor	Servicio de Exámenes Bimestrales Completos	Se adapta a cubrir los temarios de la gestión	Media
9	Estudiante	Servicio de Aprendizaje rápido y eficaz	Separar los servicios de subtitulación y traducción.	Media
10	Estudiante	Mejora en la retención de información	Gestionar los archivos procesados por el cliente	Baja
11	Estudiante	Servicio de personalización practica	El estudiante podrá organizarse en los temas que requiera mas practica	Baja
12	Estudiante	Servicio optimizado de estudio	El estudiante podrá aprender de manera	Media

Sprint Backlog
Gestión de Usuarios
Gestión de Profesores
Implementación de creación de Exámenes
Servicio de Carga de archivo

Sprint Backlog - Tareas					
<b>Objetivo:</b> Desarrollar el entorno básico para el funcionamiento del sistema.					
ID	Tarea	Tipo	Estimación	Responsable	Estado
1	Diseñar la base de datos	Diseño	3 Días	Alejandro	Completado
2	Desarrollar Gestionar Usuario	Diseño	3 Días	Guido, Maily	Completado
3	Gestión de perfiles de usuario	Gestión	1 Día	Alejandro	Completado
4	Gestión de Control de roles y permisos	Diseño	1 Día	Maily, Guido	Completado
5	Registro de usuarios nuevos	Implementación	3 Días	Guido	Completado
6	Gestión de clases y grupos	Diseño	2 Días	Alejandro	Completado

7	Implementar la Gestión de Planes	Implementación	3 Día	Alejandro	Completado
8	Gestión de clases y grupos	Implementación	3 Días	Alejandro, Guido	Completado
9	Gestión de reuniones con padres	Diseño	2 Días	Alejandro	Incompleto
10	Subida de material didáctico	Implementación	3 Días	Alejandro	Incompleto

#### 4.2 Historias de Usuario

Gestión de Profesores	
<b>ID</b> 1	<b>Descripción:</b> Gestionar las funcionalidades que tiene el profesor
<b>Prioridad</b> Alta	<b>Estimación</b> 4 días
<b>Criterios de Aceptación:</b> El profesor una vez registrado accede al sistema	
<b>Desarrollares a Cargo:</b> Salazar Vargas Guido Sahonero Salas Jose Alejandro	

Implementación de creación de Exámenes	
<b>ID</b> 3	<b>Descripción:</b> Se crearán los exámenes a partir del programa analítico
<b>Prioridad</b> Alta	<b>Estimación</b> 3 días
<b>Criterios de Aceptación:</b> Se podrá crear exámenes a partir del formato PDF con el avance de cada asignatura	
<b>Desarrollares a Cargo:</b> Salazar Vargas Guido Sahonero Salas Jose Alejandro	

Servicio de Carga de Archivo	
<b>ID</b> 4	<b>Descripción:</b> Subir un archivo para ser procesado en el sistema.
<b>Prioridad</b> Media	<b>Estimación</b> 5 días
<b>Criterios de Aceptación:</b> El estudiante sube un archivo	
<b>Desarrollares a Cargo:</b> Valdez Payllo Maily Celina	

### 4.3 Sprint Review

Sprint Review				
Inicio	Duración	Actividad	Descripción	Responsable
22:00	5 min	<b>Objetivo del Sprint</b>	Desarrollar el entorno básico para el funcionamiento del sistema	Product Owner
22:05	10 min	<b>Revisión de los requisitos principales</b>	Gestión de Usuarios. Gestión de Profesores. Implementación de creación de exámenes Servicio de carga de archivos	Product Owner
22:15	20 min	<b>Estado del Sprint</b>	<b>PLAN VS REALITY</b> El equipo tuvo reuniones presenciales y remotas de forma continua para la planificación y el desarrollo del Software	Scrum Master
22:35	20 min	<b>Demostración</b>	Los requisitos para el desarrollo de este Sprint, fueron probados y aceptados por los miembros del equipo, cumpliendo con todas las funcionalidades establecidas previamente.	Equipo
22:55	10 min	<b>Feedback</b>	Para el siguiente Sprint se tiene planeado implementar todas aquellas funcionalidades relacionadas con el Servicio Speech to Text.	Scrum Master
23:05	5 min	<b>Conclusiones</b>	Se lograron implementar las funcionalidades básicas para el funcionamiento del sistema.	Scrum Master

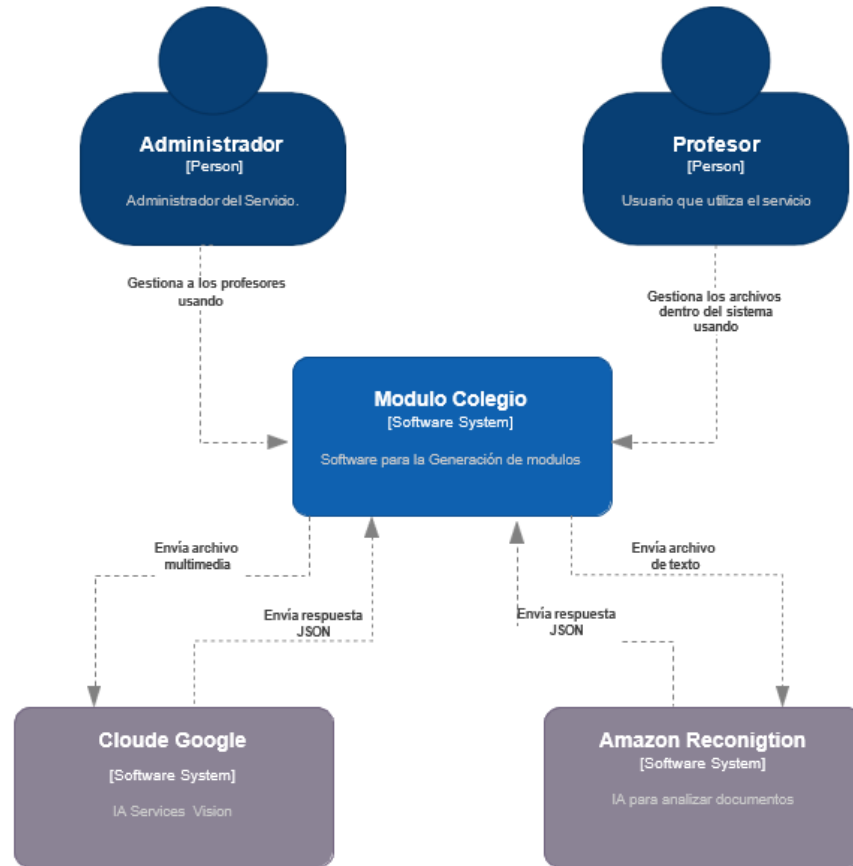
### 4.4 Sprint Retrospective

Después de la reunión, se concluyó que:

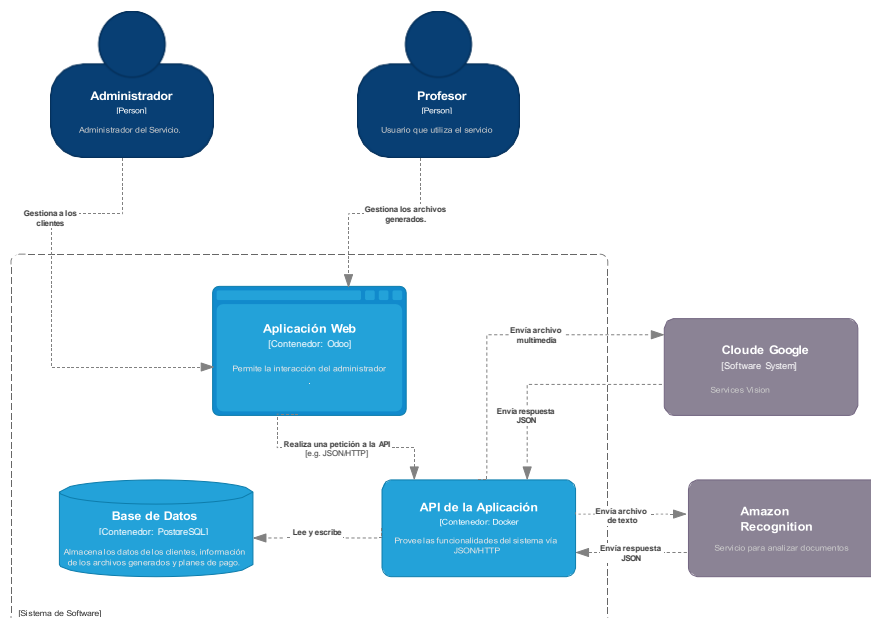
- 1.- el equipo pudo concluir satisfactoriamente sus ideales
- 2.- el equipo logro llegar a un acuerdo
- 3.- se ve el entusiasmo por parte del equipo para el desarrollo del Software

## 4.5 Modelos

### 4.5.1. Diagramas de Contexto

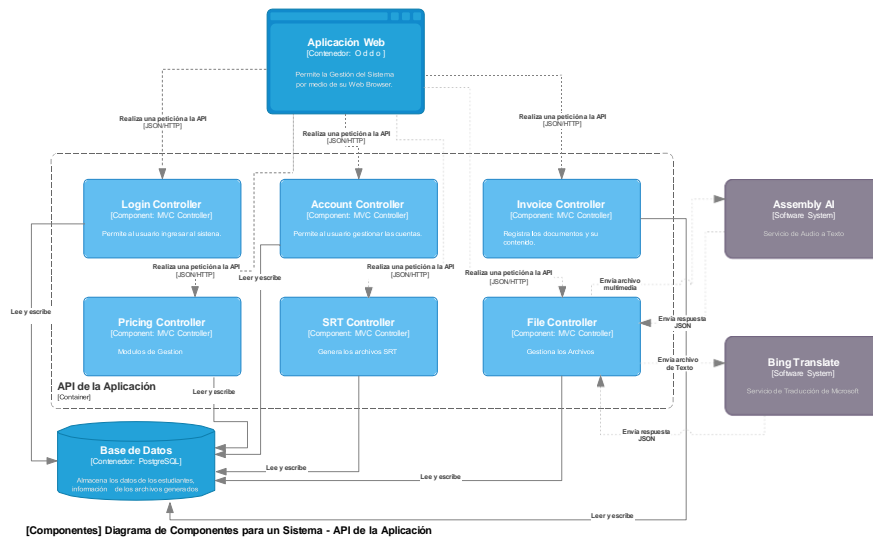


### 4.5.2. Diagrama de Contenedores

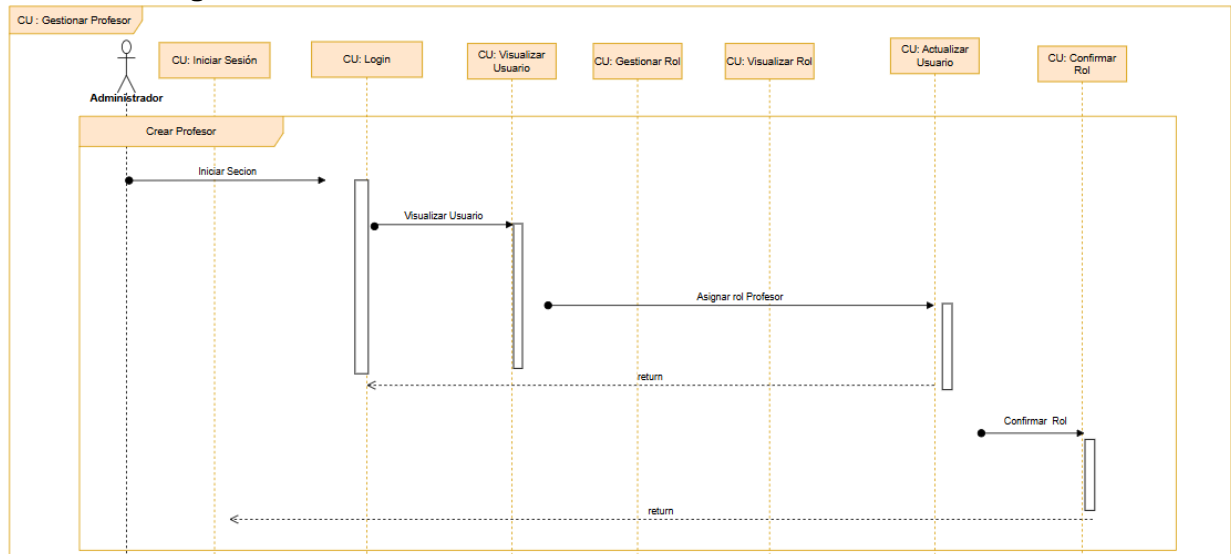




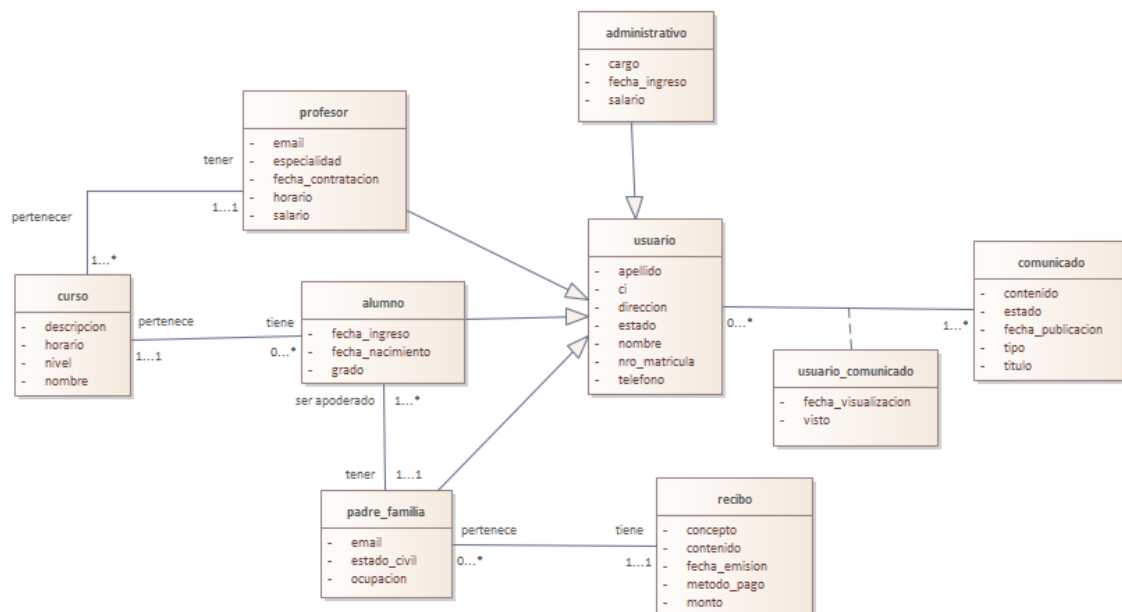
### 4.5.3. Diagrama de Componentes



#### 4.5.4. Diagrama de Secuencia



#### 4.5.5. Modelo de Datos



## 5. Sprint 2

### 5.1 Sprint Backlog

Product Backlog				
Proyecto		Software para la Generación de Subtítulos y Traducción		
Product Owner				
Versión		1.0	Fecha	10-05-2022
ID	Rol	Funcionalidad	Razón/Resultado	P
1	Administración	Gestión de reuniones con padres	Gestionar las reuniones con los padres	Alta
2	Administración	Comunicación con alumnos y padres	Gestión de comunicados a los padres de alumnos	Alta
3	Cliente	Subida de material didáctico	Subir documentos tipo PDF	Alta
4	Cliente	Visualización de calificaciones	Registrar el tiempo de inicio y final en que cada palabra es mencionada.	Media
5	Cliente	Consulta de asistencia	Devolver un texto en formato SRT.	Media
6	Cliente	Comunicación con profesores	Realizar la transcripción de un audio.	Media
7	Cliente	Entrega de trabajos	Traducir a algún idioma los archivos de texto generados por el sistema.	Media
8	Cliente	Acceso a material didáctico	Separar los servicios de subtitulación y traducción.	Media
9	Cliente	Visualización de horarios	Subir un archivo para ser procesado por el sistema.	Baja
10	Cliente	Recepción de notificaciones académicas	Gestionar los archivos procesados por el cliente.	Baja

### 5.2 Historias de Usuario

Gestión de Profesores
-----------------------

<b>ID</b> 1	<b>Descripción:</b> Gestionar las funcionalidades que tiene el profesor
<b>Prioridad</b> Alta	<b>Estimación</b> 4 días
<b>Criterios de Aceptación:</b> El profesor una vez registrado accede al sistema	
<b>Desarrollares a Cargo:</b> Salazar Vargas Guido Sahonero Salas Jose Alejandro	

Gestión de Estudiantes	
<b>ID</b> 3	<b>Descripción:</b> Se le asignara el rol a los estudiantes
<b>Prioridad</b> Alta	<b>Estimación</b> 3 días
<b>Criterios de Aceptación:</b>  Se podrá enviar notificaciones a los estudiantes de las actividades	
<b>Desarrollares a Cargo:</b>  Valdez Payllo Mialy Celina	

### 5.3 Sprint Review

Sprint Review				
Inicio	Duración	Actividad	Descripción	Responsable
22:00	5 min	<b>Objetivo del Sprint</b>	Desarrollar el entorno básico para el funcionamiento del sistema	Product Owner
22:05	10 min	<b>Revisión de los requisitos principales</b>	Gestión de Usuarios. Gestión de Profesores. Implementación de creación de exámenes Servicio de carga de archivos	Product Owner
22:15	20 min	<b>Estado del Sprint</b>	<b>PLAN VS REALITY</b> El equipo tuvo reuniones presenciales y remotas de forma continua para la planificación y el desarrollo del Software	Scrum Master
22:35	20 min	<b>Demostración</b>	Los requisitos para el desarrollo de este Sprint, fueron probados y aceptados por los miembros del equipo, cumpliendo con todas las funcionalidades establecidas previamente.	Equipo

22:55	10 min	<b>Feedback</b>	Para el siguiente Sprint se tiene planeado implementar todas aquellas funcionalidades relacionadas con el Servicio Speech to Text.	Scrum Master
23:05	5 min	<b>Conclusiones</b>	Se lograron implementar las funcionalidades básicas para el funcionamiento del sistema.	Scrum Master

#### 5.4 Sprint Retrospective

Después de la reunión, se concluyó que:

- 1.- el equipo pudo concluir satisfactoriamente sus ideales
- 2.- el equipo logro llegar a un acuerdo
- 3.- se ve el entusiasmo por parte del equipo para el desarrollo del Software

### 6. Sprint 3

#### 6.1 Sprint Backlog

Product Backlog				
Proyecto		Software para la Generación de Subtítulos y Traducción		
Product Owner				
Versión		1.0	Fecha	10-05-2022
ID	Rol	Funcionalidad	Razón/Resultado	P
1	Administración	Gestión de reuniones con padres	Gestionar las reuniones con los padres	Alta
2	Administración	Comunicación con alumnos y padres	Gestión de comunicados a los padres de alumnos	Alta
3	Cliente	Subida de material didáctico	Subir documentos tipo PDF	Alta
4	Cliente	Visualización de calificaciones	Registrar el tiempo de inicio y final en que cada palabra es mencionada.	Media
5	Cliente	Consulta de asistencia	Devolver un texto en formato SRT.	Media
6	Cliente	Comunicación con profesores	Realizar la transcripción de un audio.	Media
7	Cliente	Entrega de trabajos	Traducir a algún idioma los archivos de texto generados por el sistema.	Media

8	Cliente	Acceso a material didáctico	Separar los servicios de subtítulo y traducción.	Media
9	Cliente	Visualización de horarios	Subir un archivo para ser procesado por el sistema.	Baja
10	Cliente	Recepción de notificaciones académicas	Gestionar los archivos procesados por el cliente.	Baja

## 6.2 Historias de Usuario

Gestión de Profesores	
<b>ID</b> 1	<b>Descripción:</b> Gestionar las funcionalidades que tiene el profesor
<b>Prioridad</b> Alta	<b>Estimación</b> 4 días
<b>Criterios de Aceptación:</b> El profesor una vez registrado accede al sistema	
<b>Desarrollares a Cargo:</b> Salazar Vargas Guido Sahonero Salas Jose Alejandro	

Gestión de Estudiantes	
<b>ID</b> 1	<b>Descripción:</b> Visualizar los grados , alumnos y/o apoderados
<b>Prioridad</b> Alta	<b>Estimación</b> 4 días
<b>Criterios de Aceptación:</b> Ya registrado el estudiante podrá ver su grado y materias	
<b>Desarrollares a Cargo:</b> Salazar Vargas Guido Sahonero Salas Jose Alejandro	

## 6.3 Sprint Review

Sprint Review				
Inicio	Duración	Actividad	Descripción	Responsable

22:00	5 min	<b>Objetivo del Sprint</b>	Desarrollar el entorno básico para el funcionamiento del sistema	Product Owner
22:05	10 min	<b>Revisión de los requisitos principales</b>	Gestión de Usuarios. Gestión de Profesores. Implementación de creación de exámenes Servicio de carga de archivos	Product Owner
22:15	20 min	<b>Estado del Sprint</b>	<b>PLAN VS REALITY</b> El equipo tuvo reuniones presenciales y remotas de forma continua para la planificación y el desarrollo del Software	Scrum Master
22:35	20 min	<b>Demostración</b>	Los requisitos para el desarrollo de este Sprint, fueron probados y aceptados por los miembros del equipo, cumpliendo con todas las funcionalidades establecidas previamente.	Equipo
22:55	10 min	<b>Feedback</b>	Para el siguiente Sprint se tiene planeado implementar todas aquellas funcionalidades relacionadas con el Servicio Speech to Text.	Scrum Master
23:05	5 min	<b>Conclusiones</b>	Se lograron implementar las funcionalidades básicas para el funcionamiento del sistema.	Scrum Master

#### 6.4 Sprint Retrospective

Después de la reunión, se concluyó que:

- 1.- el equipo pudo concluir satisfactoriamente sus ideales
- 2.- el equipo logro llegar a un acuerdo
- 3.- se ve el entusiasmo por parte del equipo para el desarrollo del Software

### 7. Sprint 4

#### 7.1 Sprint Backlog

Product Backlog			
<b>Proyecto</b>	Software para la Generación de Subtítulos y Traducción		
<b>Product Owner</b>			
<b>Versión</b>	1.0	<b>Fecha</b>	10-05-2022

ID	Rol	Funcionalidad	Razón/Resultado	P
1	Administración	Gestión de reuniones con padres	Gestionar las reuniones con los padres	Alta
2	Administración	Comunicación con alumnos y padres	Gestión de comunicados a los padres de alumnos	Alta
3	Cliente	Subida de material didáctico	Subir documentos tipo PDF	Alta
4	Cliente	Visualización de calificaciones	Visualizar calificaciones de estudiantes	Media
5	Cliente	Consulta de asistencia	Consultar asistencia de estudiantes	Media
6	Cliente	Comunicación con profesores	Comunicación con profesores con padres	Media
7	Cliente	Recepción de notificaciones	Notificaciones tipo push	Media
8	Cliente	Gestión de reuniones con profesores	Reunión profesores , tutores o apoderados	Media
9	Cliente	Gestión de matrículas	Gestión de pagos	Baja
10	Cliente	Gestión de personal docente	Creación de nuevos docentes en el software	Baja

## 7.2 Historias de Usuario

Gestión de Profesores	
<b>ID</b> 1	<b>Descripción:</b> Gestionar las funcionalidades que tiene el profesor
<b>Prioridad</b> Alta	<b>Estimación</b> 4 días
<b>Criterios de Aceptación:</b> El profesor una vez registrado accede al sistema	



<b>Desarrollares a Cargo:</b> <div>Salazar Vargas Guido</div> <div>Sahonero Salas Jose Alejandro</div>
-----------------------------------------------------------------------------------------------------------

Gestión Académica	
<b>ID</b> 1	<b>Descripción:</b> Gestionar las funciones académicas, Profesores, materias y horarios
<b>Prioridad</b> Alta	<b>Estimación</b> 12 días
<b>Criterios de Aceptación:</b> El profesor una vez registrado accede al sistema ve las materias y horarios	
<b>Desarrollares a Cargo:</b> <div>Salazar Vargas Guido</div> <div>Sahonero Salas Jose Alejandro</div>	

Gestión de Comunicaciones	
<b>ID</b> 1	<b>Descripción:</b> Gestionar los comunicados con destinatarios
<b>Prioridad</b> Alta	<b>Estimación</b> 12 días
<b>Criterios de Aceptación:</b> Se seleccionara el tipo de comunicado y los destinatarios para el comunicado	
<b>Desarrollares a Cargo:</b> <div>Salazar Vargas Guido</div> <div>Valdez Payllo Maily Celina</div>	

### 7.3 Sprint Review

Sprint Review				
Inicio	Duración	Actividad	Descripción	Responsable

22:00	5 min	<b>Objetivo del Sprint</b>	Desarrollar el entorno básico para el funcionamiento del sistema	Product Owner
22:05	10 min	<b>Revisión de los requisitos principales</b>	Gestión de Usuarios. Gestión de Profesores. Implementación de creación de exámenes Servicio de carga de archivos	Product Owner
22:15	20 min	<b>Estado del Sprint</b>	<b>PLAN VS REALITY</b> El equipo tuvo reuniones presenciales y remotas de forma continua para la planificación y el desarrollo del Software	Scrum Master
22:35	20 min	<b>Demostración</b>	Los requisitos para el desarrollo de este Sprint, fueron probados y aceptados por los miembros del equipo, cumpliendo con todas las funcionalidades establecidas previamente.	Equipo
22:55	10 min	<b>Feedback</b>	Para el siguiente Sprint se tiene planeado implementar todas aquellas funcionalidades relacionadas con el Servicio Speech to Text.	Scrum Master
23:05	5 min	<b>Conclusiones</b>	Se lograron implementar las funcionalidades básicas para el funcionamiento del sistema.	Scrum Master

#### 7.4 Sprint Retrospective

Después de la reunión, se concluyó que:

- 1.- el equipo pudo concluir satisfactoriamente sus ideales
- 2.- el equipo logro llegar a un acuerdo
- 3.- se ve el entusiasmo por parte del equipo para el desarrollo del Software

### CAPITULO III

#### GUIA PARA USUARIOS

##### 3.1 Manual de Usuario

[https://youtu.be/vTi2wtS\\_pEg](https://youtu.be/vTi2wtS_pEg)



### 3.2 Videos demostrativos

#### Configuración de Odoo en Localhost

<https://www.youtube.com/watch?v=ZF9iE2btmKw>



#### Odoo Deploy in Digital Ocean

<https://youtu.be/iffWfej6N7E?si=Wgil0DsUICqq55PS>



## 8. Bibliografía

Gracia Peña, R. (2013). Gestión de proyectos con metodologías ágiles.  
Ingenieria.de.software.enfoque.practico.7ed.Pressman.

Fundamentos de la ingeniería del Software Lic. César Ariel Briano 20232daEdicion  
Araujo, O. L. F. & Lorena, O. (2014). Desarrollo de una nueva metodología derivada  
del COCOMO II agregando pautas de calidad.

## **ANEXO**

Repositorio GitHub del proyecto de Odoo:

<https://github.com/GuidoSV7/odooingsw1>





