

ARQUITECTURA DE SOFTWARE

JUAN PABLO BORJA ESPITIA - 20202020091
JULIAN DAVID ROJAS CALLEJAS - 20182020153
KEVIN NICOLAS SIERRA GÓNZALEZ- 20182020151

5 de febrero de 2025

Índice general

I ORGANIZACIÓN	7
1. Método y Lenguaje	9
1.1. Introducción	9
1.2. ADM	10
1.3. Archimate	10
2. Empresa	11
2.1. Introducción	11
2.2. Nombre	12
2.3. Misión	12
2.4. Visión	12
2.5. Objetivos Organizacionales	12
2.6. Requerimientos Organizacionales	12
2.7. Servicios	13
2.8. Procesos	13
2.9. Productos	14
II ARQUITECTURA	15
3. Capa Motivacional	17
3.1. Introducción	17
3.2. Vista de Stakeholder	18
3.3. Vista de Realización de Objetivos	18
3.4. Vista de Contribución de Objetivos	19
3.5. Vista de Principios	19
3.6. Vista de Realización de Requerimientos	20
4. Capa Estratégica	21
4.1. Introducción	21
4.2. Vista de Mapa de Capacidad	22
4.3. Vista de Realización de Resultado	23
4.4. Vista de Mapa de Recurso	23
4.5. Vista de Flujo de Valor	24
5. Capa organizacional	25
5.1. Introducción	25
5.2. Vista de Organización	26
5.3. Vista de cooperación de actor	27
5.4. Vista de función de negocio	28
5.5. Vista de proceso de negocio	29
5.6. Vista de cooperación de proceso de negocio	30
5.7. Vista de producto	31

6. Capa de Aplicación	33
6.1. Introducción	33
6.2. Vista de comportamiento de la aplicación	34
6.3. vista de cooperación de la aplicación	35
6.4. vista de estructura de la aplicación	35
7. Capa de Tecnología	37
7.1. Introducción	37
7.2. vista de tecnología	38
7.3. vista de uso de la tecnología	38
7.4. vista de estructura de la información	39
7.5. vista de Despliegue e Implementación	39
8. Capa de Migración	41
8.1. Introducción	41
8.2. vista de proyecto	42
8.3. vista de migración	43
8.4. vista de Implementación/Migración	43
III PATRONES	45
9. Patrones Creacionales	47
9.1. Patrón Singleton	47
9.1.1. Problema	47
9.1.2. Estructura	47
9.1.3. Funcionamiento	47
9.1.4. Implementación	47
9.1.5. Observaciones	48
9.2. Patrón Factory	49
9.2.1. Problema	49
9.2.2. Estructura	49
9.2.3. Funcionamiento	49
9.2.4. Implementación	49
9.2.5. Observaciones	50
10.Estructurales	51
10.1. Patrón Proxy	51
10.1.1. Problema	51
10.1.2. Estructura	51
10.1.3. Funcionamiento	51
10.1.4. Implementación	52
10.1.5. Observaciones	52
10.2. Patrón Decorator	53
10.2.1. Problema	53
10.2.2. Estructura	53
10.2.3. Funcionamiento	54
10.2.4. Implementación	54
10.2.5. Observaciones	54
11.Patrones de Comportamiento	55
11.1. Patrón Comando	55
11.1.1. Problema	55
11.1.2. Estructura	55
11.1.3. Funcionamiento	55
11.1.4. Implementación	56
11.1.5. Observaciones	56

<i>ÍNDICE GENERAL</i>	5
IV CIERRE	57
Referencias	59

Parte I

ORGANIZACIÓN

Capítulo 1

Método y Lenguaje

1.1. Introducción

En el mundo de la arquitectura empresarial, los métodos y lenguajes juegan un papel crucial en la estructura y gestión de las organizaciones. El Método-Lenguaje con ADM y ArchiMate es una combinación poderosa utilizada para diseñar, planificar y gestionar arquitecturas empresariales de manera efectiva. Cuando se combinan ADM y ArchiMate, se obtiene un enfoque robusto para la arquitectura empresarial. ADM proporciona la metodología y el proceso para desarrollar y gestionar la arquitectura, mientras que ArchiMate ofrece el lenguaje y las herramientas para modelar y visualizar esta arquitectura. Juntos, estos componentes permiten a las organizaciones construir arquitecturas empresariales que sean efectivas, comprensibles y alineadas con sus objetivos estratégicos.

1.2. ADM

ADM es un componente central del marco TOGAF (The Open Group Architecture Framework). TOGAF es un estándar de la industria para el desarrollo de arquitecturas empresariales. El ADM proporciona una metodología estructurada y un conjunto de fases para la creación y gestión de arquitecturas.

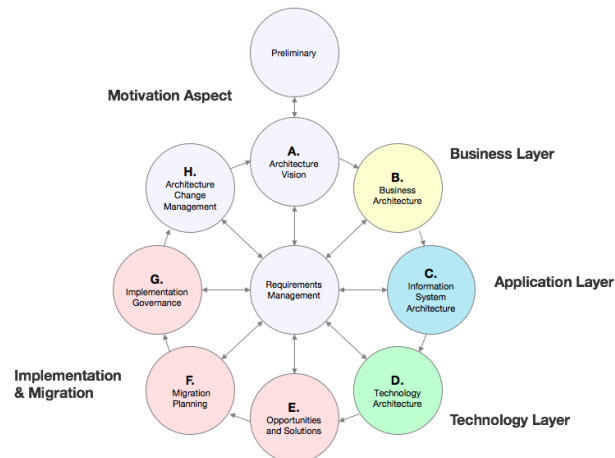


Figura 1.1: ADM [4, 3, 2]

1.3. Archimate

ArchiMate es un lenguaje de modelado para la arquitectura empresarial que complementa el ADM. Desarrollado por The Open Group, ArchiMate proporciona un marco visual para representar las diferentes capas y aspectos de una arquitectura empresarial.

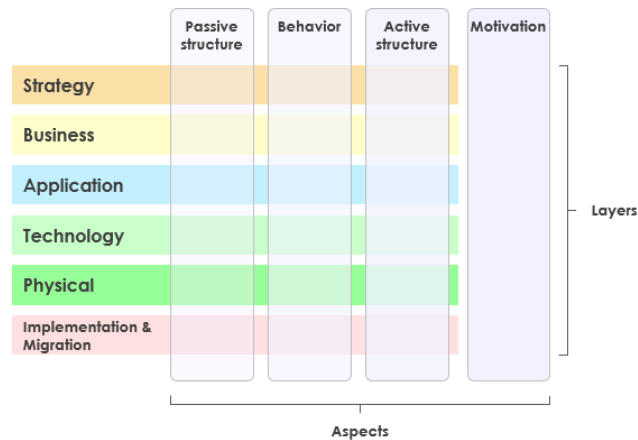


Figura 1.2: Archimate

Capítulo 2

Empresa

2.1. Introducción

ContaSync Pro es una startup tecnológica creada con el propósito de transformar la gestión contable de las pequeñas y medianas empresas (PYMES) en Colombia, enfrentando los desafíos del mundo globalizado. Aprovechando las últimas tecnologías emergentes, como la inteligencia artificial, el análisis de datos y la automatización de procesos, ContaSync Pro ofrece soluciones personalizadas que no solo mejoran la eficiencia operativa, sino que también garantizan la precisión y la integridad de la información financiera, alineándose con las normativas locales e internacionales. Nuestro objetivo es empoderar a las PYMES colombianas, proporcionándoles herramientas que les permitan competir en un mercado cada vez más digital y globalizado, asegurando al mismo tiempo su sostenibilidad y crecimiento.

2.2. Nombre

ContaSync Pro

2.3. Misión

Transformar la gestión financiera de las pequeñas y medianas empresas (PYMES) en Colombia mediante la provisión de soluciones tecnológicas avanzadas que automatizan y optimizan los procesos contables. A través de un enfoque centrado en la innovación continua, la seguridad de la información y la satisfacción del cliente, buscamos empoderar a las PYMES para que logren un control financiero preciso, eficiente y alineado con las normativas locales e internacionales. Nuestro compromiso es asegurar que cada cliente no solo mantenga su competitividad en un mercado globalizado, sino que también esté preparado para crecer y expandirse en el futuro.

2.4. Visión

Ser la empresa líder en América Latina en soluciones de automatización contable para PYMES, reconocida por nuestra innovación constante, fiabilidad y capacidad para adaptar las tecnologías emergentes a las necesidades cambiantes del mercado. Para 2030, aspiramos a ser un referente en la transformación digital del sector financiero empresarial, habiendo establecido una presencia significativa en mercados internacionales, y logrando que nuestras soluciones sean indispensables para la gestión financiera eficiente y segura en un mundo cada vez más interconectado.

2.5. Objetivos Organizacionales

- **Optimizar los procesos administrativos :** Implementaremos un sistema de gestión financiera automatizado que garantice la integridad y precisión de los datos contables, nomina, inventario y contabilidad, cuyo resultado esperado es la reducción de la pérdida de recursos.
- **Adoptar un 60 % de nuestras soluciones en el mercado de PYMES en Colombia en 5 años:** Lanzaremos una campaña de expansión nacional, estableciendo alianzas con cámaras de comercio locales y asociaciones de PYMES para promover la adopción de ContaSync Pro. Además, nos enfocaremos en ofrecer versiones escalables de nuestras soluciones para diferentes sectores industriales dentro del país.
- **Gestión de Nómina Eficiente :** Desarrollar un módulo de gestión de nómina que automatice el cálculo de salarios, deducciones, y pagos, asegurando cumplimiento con las leyes laborales.
- **Innovar de manera continua con la incorporación de nuevas funcionalidades basadas en tecnologías emergentes:** Cada año, incorporaremos al menos 3 funcionalidades nuevas en nuestro software, utilizando inteligencia artificial, análisis predictivo y tecnologías móviles, asegurando que nuestras soluciones permanezcan a la vanguardia del sector.
- **Control y Gestión de Inventario :** Implementar un sistema de gestión de inventario que automatice el seguimiento de existencias, órdenes de compra, y reabastecimiento.
- **Cumplimiento Normativo y Seguridad de Datos :** Garantizar que el sistema cumpla con las regulaciones locales y nacionales en materia de contabilidad y protección de datos, asegurando la seguridad y confidencialidad de la información.

2.6. Requerimientos Organizacionales

- Cumplimiento estricto de las regulaciones gubernamentales, especialmente la Ley 43 de 1990.
- Mantener un equipo altamente capacitado en normativas contables vigentes y análisis financiero.
- Implementar y mantener rigurosos protocolos de seguridad de datos.

- Desarrollar una cultura organizacional basada en la ética, integridad y mejora continua.
- Establecer un programa de actualización profesional continua para todo el personal.
- Contruir una solución escalable y flexible que se adapte a las necesidades particulares de las pymes.

2.7. Servicios

- Automatización de procesos contables para PYMES: Implementamos soluciones que automatizan la entrada de datos, generación de reportes fiscales y otros procesos contables clave, mejorando la eficiencia y precisión.
- Consultoría en implementación de sistemas contables automatizados: Ofrecemos asesoría integral en la evaluación previa, personalización e implementación de sistemas contables, así como soporte post-implementación para asegurar una integración efectiva.
- Capacitación en el uso de herramientas de automatización contable: Proporcionamos formación completa a través de talleres en línea, sesiones presenciales y tutoriales interactivos, para asegurar que los usuarios aprovechen al máximo las herramientas de automatización.
- Auditoría y optimización de procesos financieros: Realizamos auditorías exhaustivas y optimizamos procesos financieros para mejorar la eficiencia y garantizar la conformidad con las regulaciones vigentes.
- Soporte técnico 24/7 para usuarios de nuestras soluciones: Ofrecemos asistencia continua a través de múltiples canales (teléfono, chat, correo electrónico), abordando problemas técnicos y proporcionando ayuda en el uso del software.
- Actualización y Mantenimiento del Software: Implementamos un servicio de mantenimiento proactivo, que incluye actualizaciones regulares y parches de seguridad para mantener el software en óptimas condiciones.

2.8. Procesos

- Desarrollo y actualización continua de software de automatización contable: Trabajamos en la mejora constante del software, implementando nuevas funcionalidades y correcciones según las necesidades del mercado y el feedback de los usuarios.
- Implementación personalizada de soluciones para cada cliente: Adaptamos nuestras soluciones a las necesidades específicas de cada cliente, realizando un análisis detallado y personalizando la implementación para garantizar el mejor ajuste posible.
- Monitoreo y análisis de la eficiencia de los procesos automatizados: Evaluamos continuamente la eficiencia de los procesos automatizados utilizando herramientas de monitoreo avanzadas y métricas clave para asegurar el rendimiento óptimo.
- Gestión de la seguridad y integridad de los datos financieros: Implementamos medidas robustas de seguridad, como encriptación de datos, control de acceso y políticas de respaldo, para proteger la información financiera y personal.
- Capacitación y soporte continuo a clientes: Ofrecemos capacitación continua y soporte adicional para asegurar que nuestros clientes se mantengan actualizados y resuelvan cualquier duda o problema que surja.
- Evaluación de Satisfacción del Cliente: Implementamos encuestas de satisfacción y procesos de retroalimentación para medir la satisfacción del cliente y realizar mejoras continuas basadas en sus comentarios.

2.9. Productos

- ContaSync Core: Software Base de Automatización Contable para PYMES, Una solución integral que cubre todas las necesidades contables básicas, incluyendo la gestión de transacciones, conciliaciones y reportes financieros.
- ContaSync CxC: Módulo Especializado en Gestión de Cuentas por Cobrar, Un módulo diseñado para optimizar la gestión de cuentas por cobrar, incluyendo seguimiento de pagos, generación de recordatorios y análisis de la antigüedad de saldos.
- ContaSync Nómina: Sistema Automatizado de Gestión de Nómina y Recursos Humanos, Ofrece funciones avanzadas para el cálculo de salarios, gestión de beneficios, y generación de reportes fiscales, asegurando cumplimiento con las regulaciones laborales.
- ContaSync Analítica: Herramienta de Análisis Financiero y Generación de Reportes, Permite realizar análisis detallados y generar informes financieros personalizados, facilitando una mejor toma de decisiones estratégicas.
- ContaSync Mobile: Aplicación Móvil para Acceso y Gestión Contable en Tiempo Real, Ofrece acceso remoto y gestión de datos contables desde cualquier lugar, mejorando la flexibilidad y la capacidad de respuesta.

Parte II

ARQUITECTURA

Capítulo 3

Capa Motivacional

3.1. Introducción

La capa motivacional proporciona una visión del porqué detrás de las decisiones tomadas dentro de una empresa, que se enfoca en las razones o motivos subyacentes a dichas decisiones y los factores que impulsan a estos cambios dentro de la empresa. Todo esto con el propósito de ayudar a alinear las estrategias y los objetivos con las decisiones de arquitectura empresarial.

3.2. Vista de Stakeholder

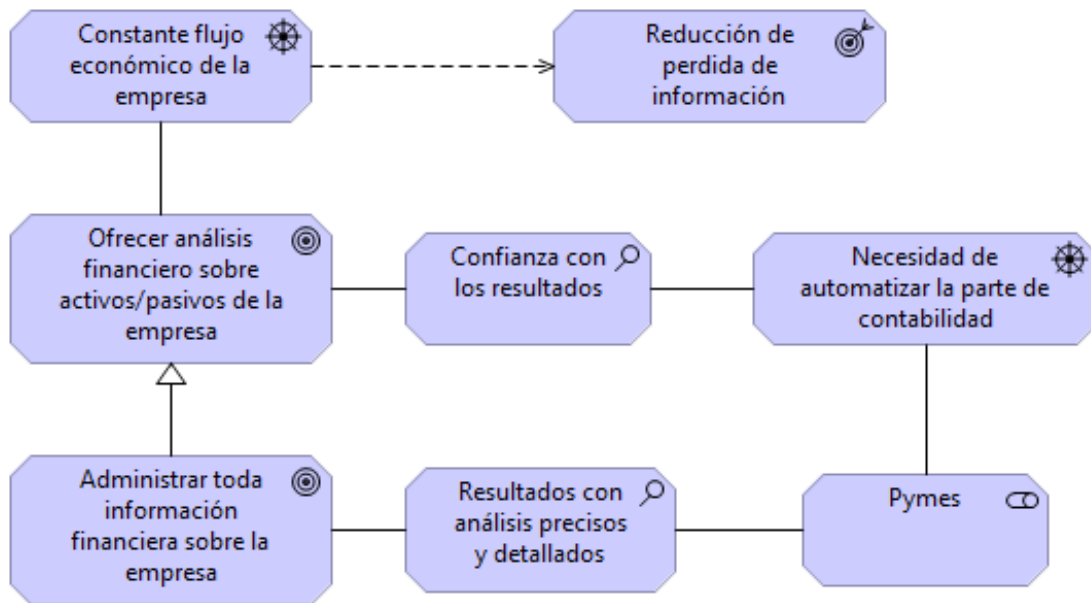


Figura 3.1: Diagrama de Vista de Stakeholder

3.3. Vista de Realización de Objetivos

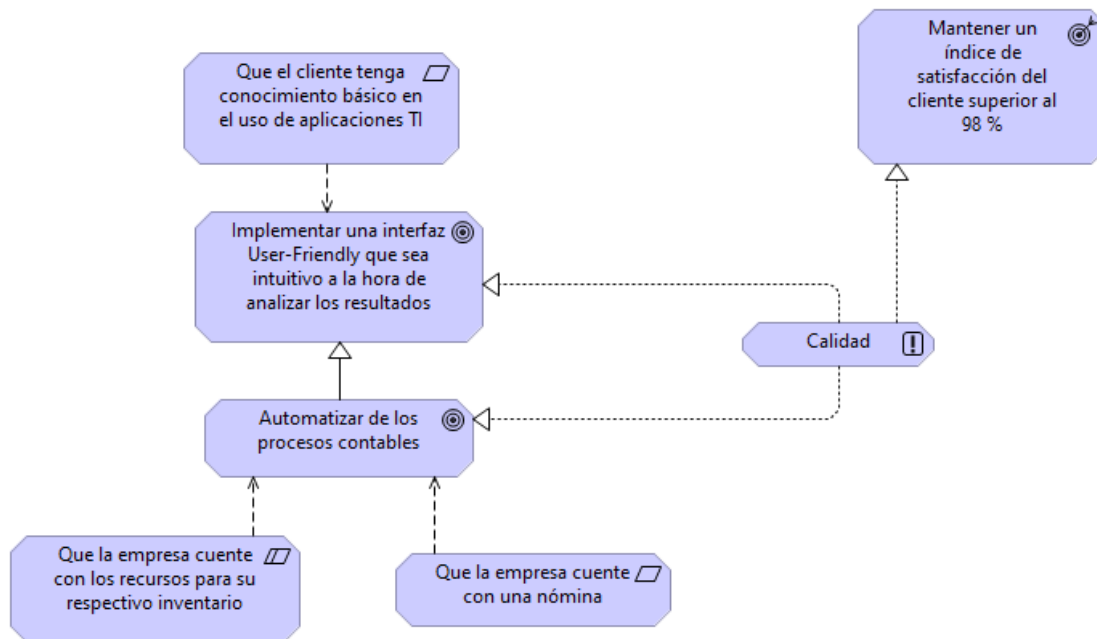


Figura 3.2: Diagrama de Vista de Realización de Objetivos

3.4. Vista de Contribución de Objetivos

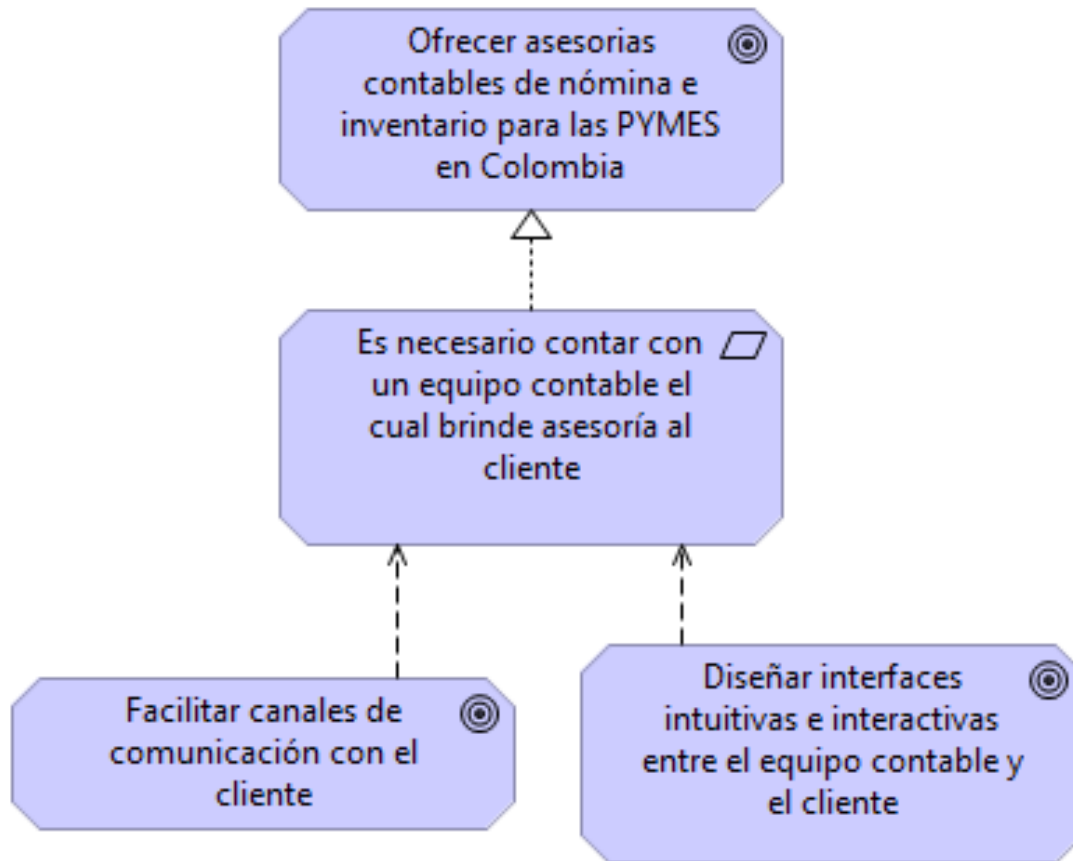


Figura 3.3: Diagrama de Vista de Contribución de Objetivos

3.5. Vista de Principios

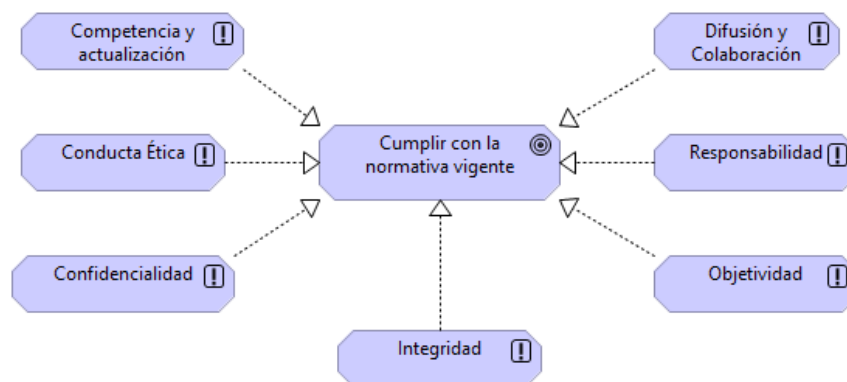


Figura 3.4: Diagrama de Vista de Principios

3.6. Vista de Realización de Requerimientos

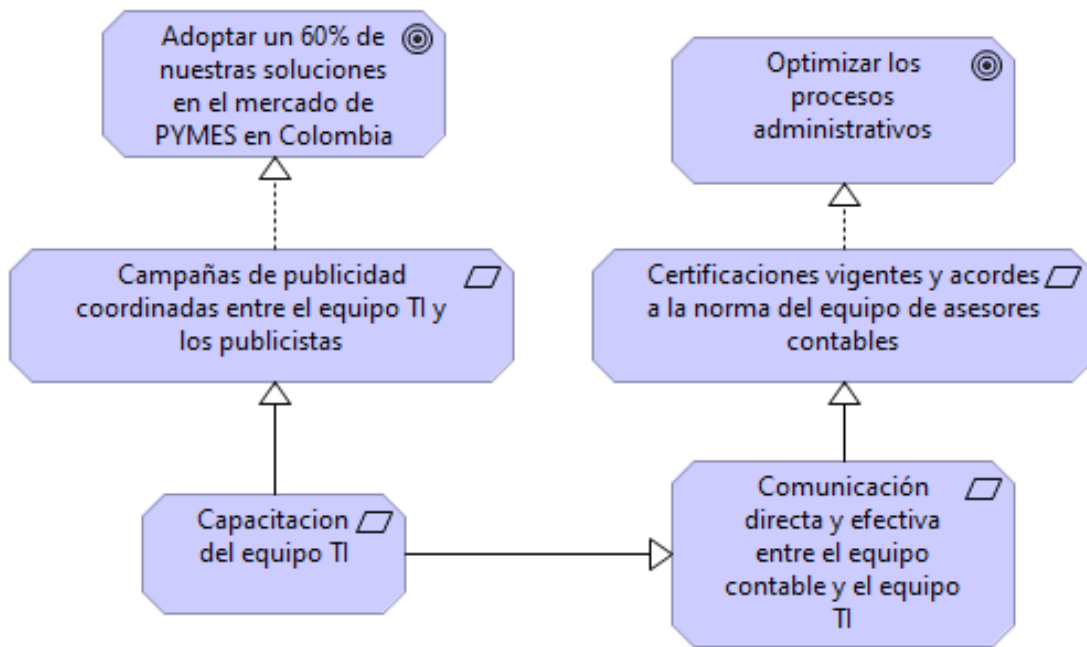


Figura 3.5: Diagrama de Vista de Realización de Requerimientos

Capítulo 4

Capa Estratégica

4.1. Introducción

La capa estratégica es un punto de vista que se enfoca en la planificación y gestión de negocios, tecnología y proyectos. En términos generales, se refiere al nivel de análisis y toma de decisiones que está enfocado en los objetivos a largo plazo y la dirección general de una organización.

4.2. Vista de Mapa de Capacidad

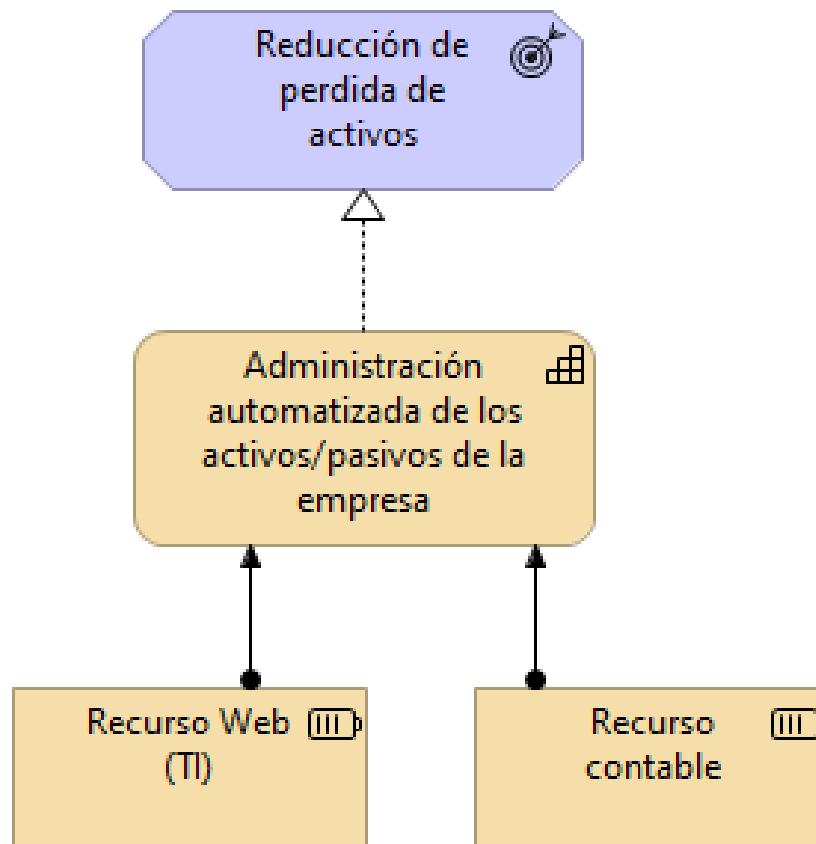


Figura 4.1: Diagrama de Vista de Mapa de Capacidad

4.3. Vista de Realización de Resultado

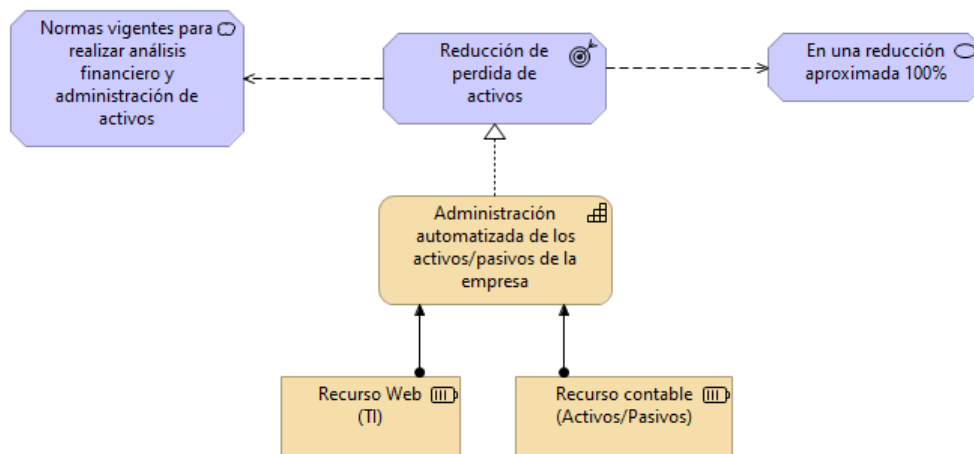


Figura 4.2: Diagrama de Vista de Realización de Resultado

4.4. Vista de Mapa de Recurso

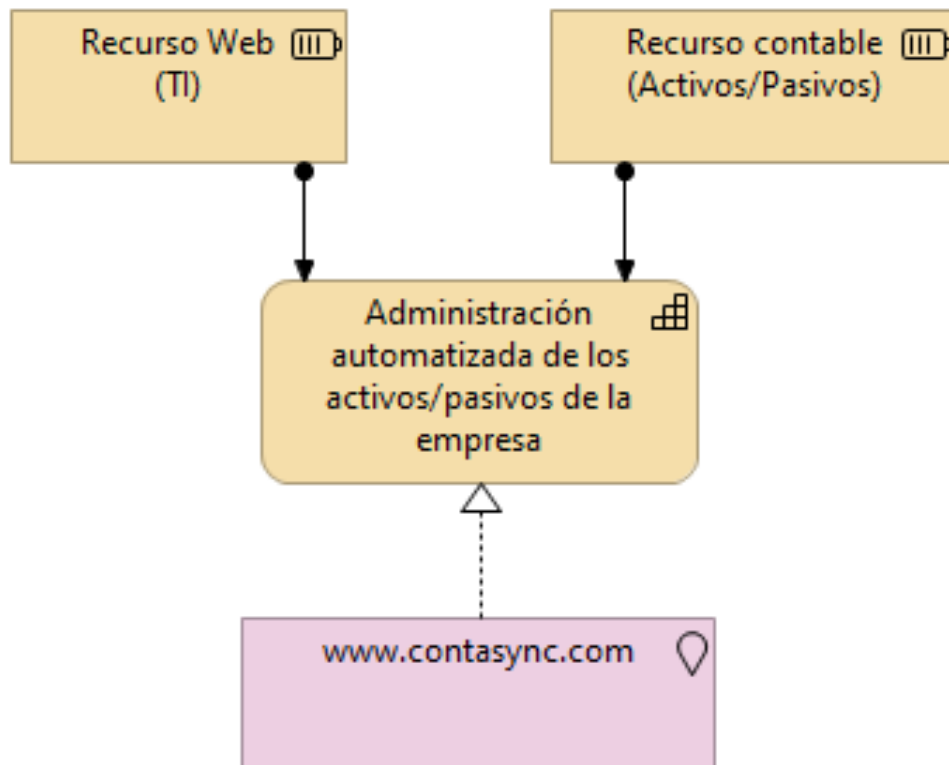


Figura 4.3: Diagrama de Vista de Mapa de Recurso

4.5. Vista de Flujo de Valor

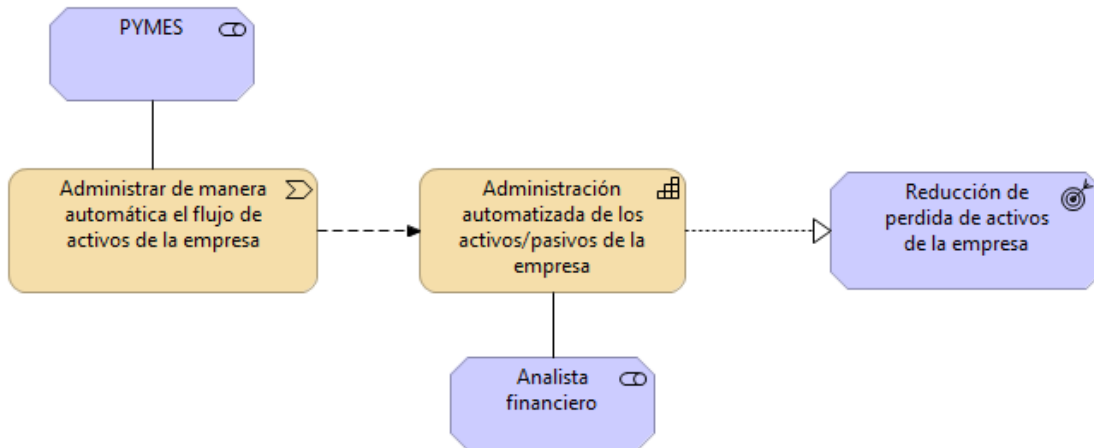


Figura 4.4: Diagrama de Vista de Flujo de Valor

Capítulo 5

Capa organizacional

5.1. Introducción

El modelo de negocio de *Soluciones ContaSync* está diseñado y alineado conforme a la capa de negocio de la arquitectura empresarial ArchiMate 3.1, lo que permite estructurar claramente las interrelaciones entre las diferentes partes de la organización, los procesos de negocio y los servicios ofrecidos a las PYMES. Este enfoque facilita la automatización y optimización de los procesos contables para garantizar la mitigación de la pérdida de información financiera y el cumplimiento normativo.

El modelo de negocio está estructurado alrededor de actores clave como los supermercados, textilerías y restaurantes (dentro del mercado objetivo de PYMES) que utilizan los servicios de ContaSync Pro para mejorar su gestión financiera. Para ello, se implementan procesos centralizados de administración de cuentas por cobrar, activos, pasivos y nómina, basados en regulaciones gubernamentales, como la Ley 43 de 1990, y estándares internacionales.

El modelo sigue una estrategia clara de automatización de procesos contables, lo que no solo optimiza la operatividad, sino que reduce el riesgo de errores humanos y pérdidas de información. Este proceso de automatización se alinea con las normas contables vigentes en Colombia, asegurando que las empresas cumplan con los requerimientos regulatorios mientras optimizan sus operaciones.

Además, el modelo incluye componentes de soporte clave como la integridad, objetividad, responsabilidad, y confidencialidad, lo que garantiza un servicio confiable y ético para los usuarios. La empresa también está comprometida con la actualización y mejora continua, ofreciendo a los clientes una plataforma adaptable y flexible a los cambios del mercado y de las normativas gubernamentales.

Este modelo de negocio está diseñado para ser escalable y expandible, permitiendo su crecimiento en otras industrias y regiones, así como la integración con tecnologías emergentes, como el análisis de datos avanzados y la inteligencia artificial. El objetivo final es que *ContaSync Pro* se convierta en un socio estratégico de las PYMES en su proceso de transformación digital, asegurando su sostenibilidad financiera a largo plazo.

5.2. Vista de Organización

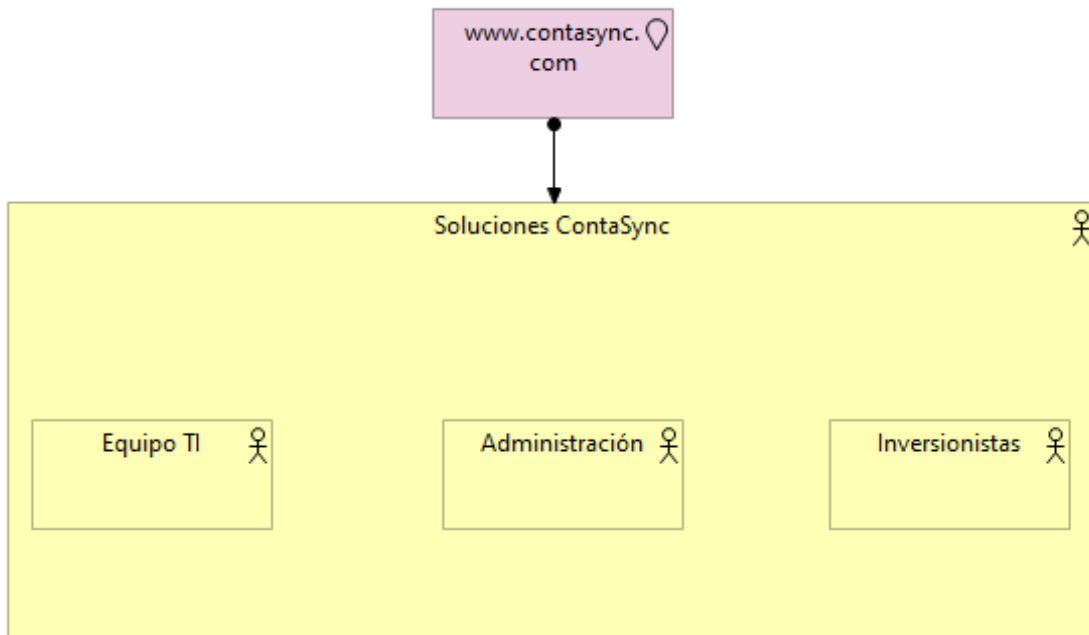


Figura 5.1: Diagrama de Vista de la organización

5.3. Vista de cooperación de actor

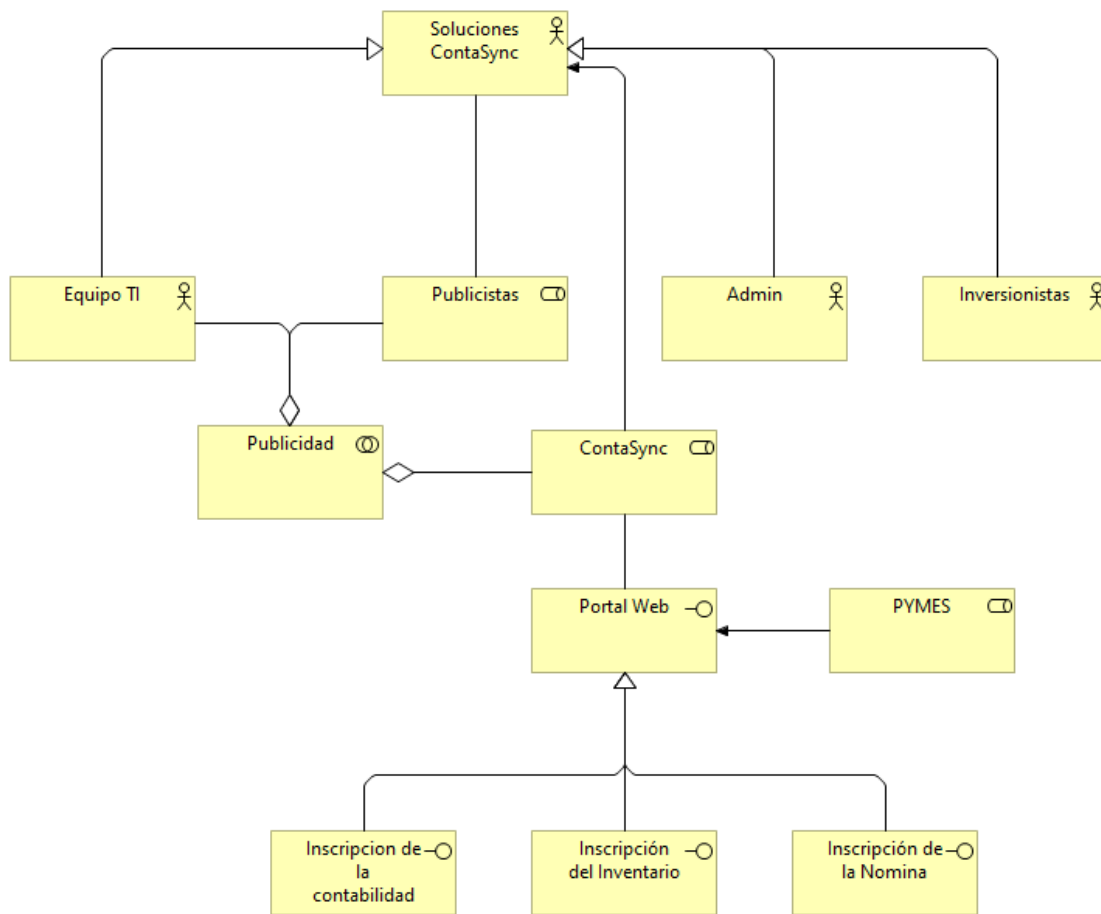


Figura 5.2: Diagrama de Vista de cooperación de actor

5.4. Vista de función de negocio

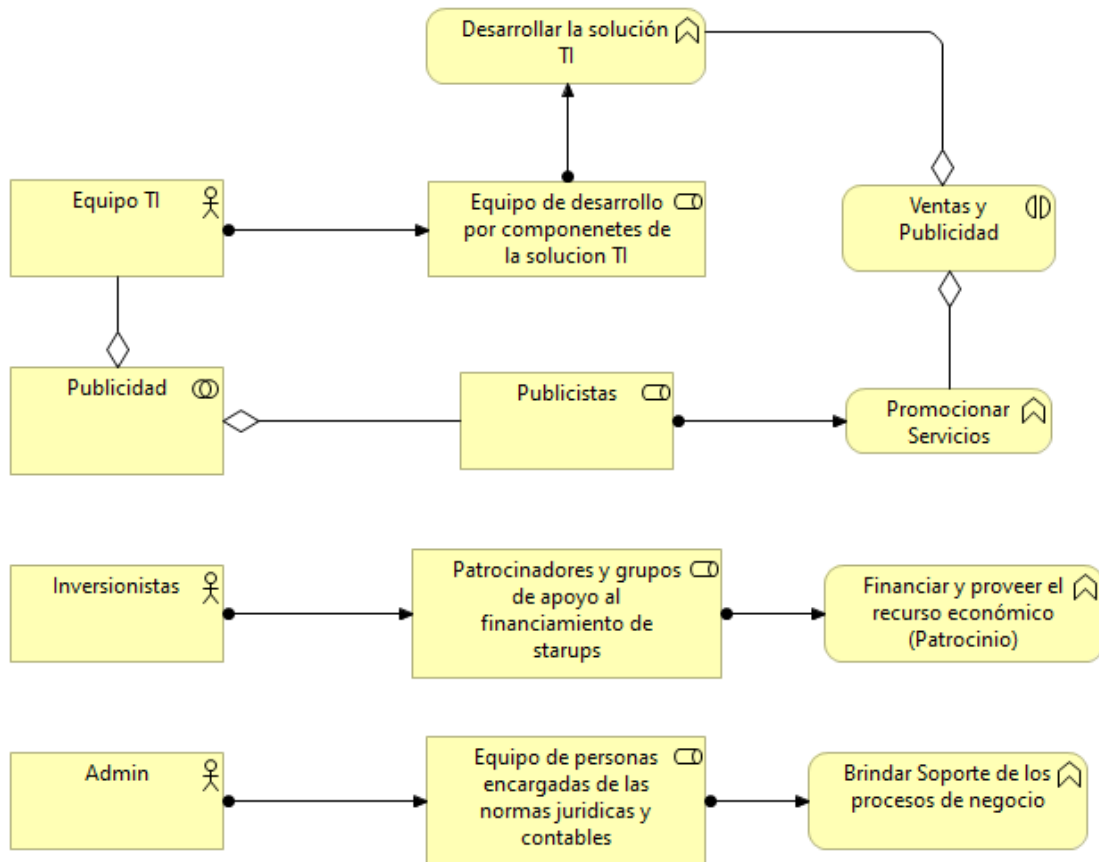


Figura 5.3: Diagrama de Vista de función de negocio

5.5. Vista de proceso de negocio

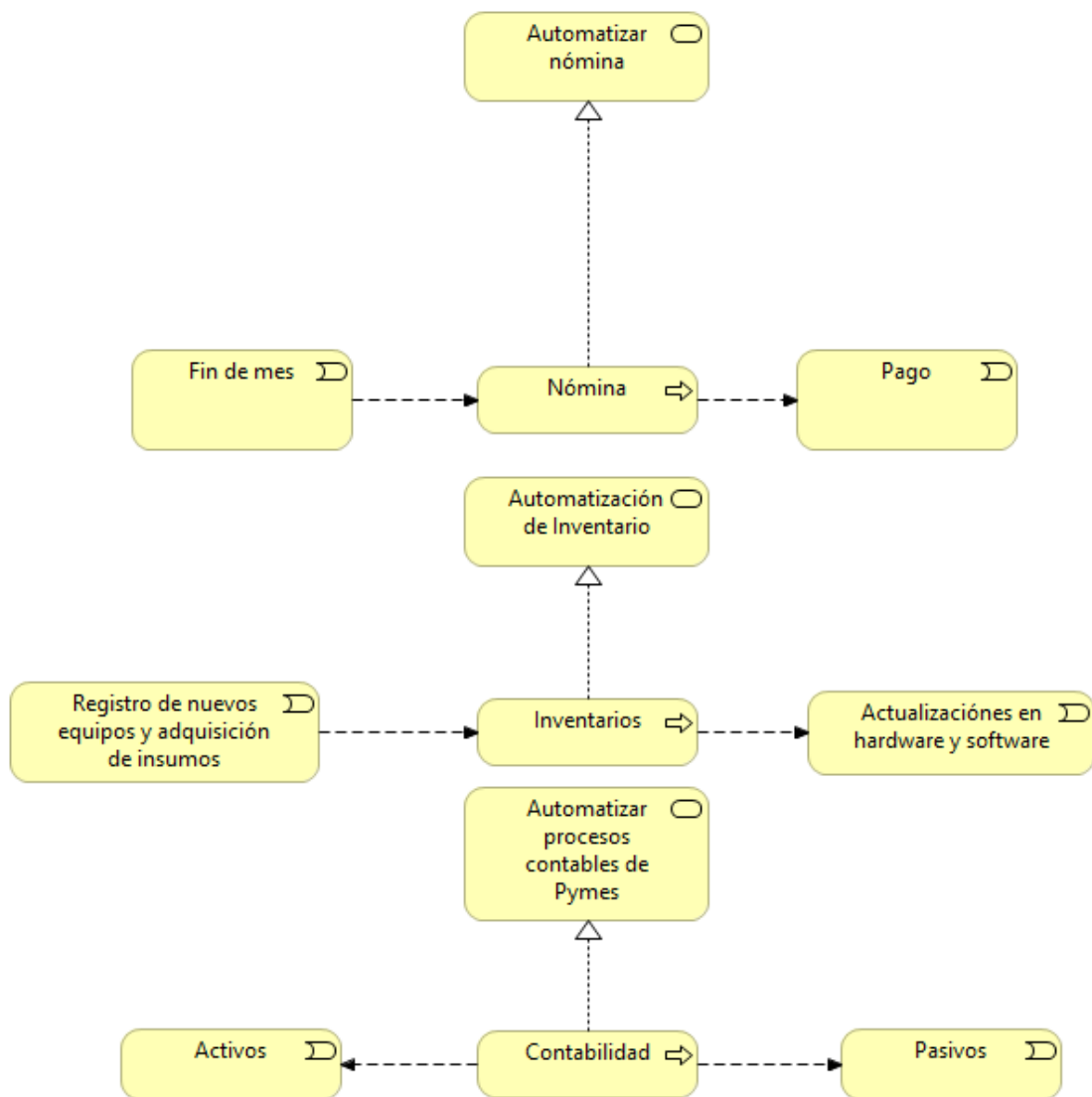


Figura 5.4: Diagrama de Vista de proceso de negocio

5.6. Vista de cooperación de proceso de negocio

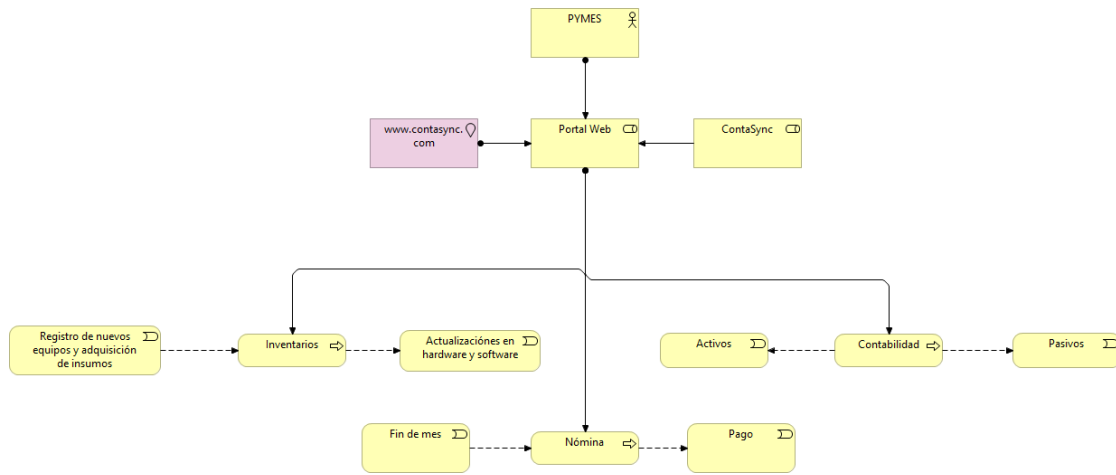


Figura 5.5: Diagrama de Vista de cooperación de proceso de negocio

5.7. Vista de producto

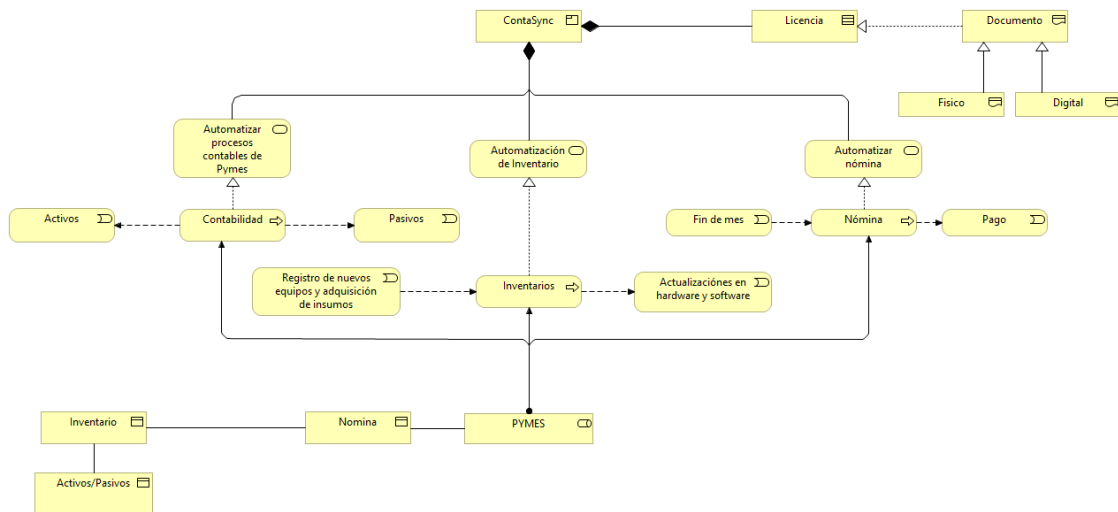


Figura 5.6: Diagrama de Vista de producto

Capítulo 6

Capa de Aplicación

6.1. Introducción

La capa de aplicación en ContaSync Pro define los sistemas y servicios tecnológicos que soportan los procesos de negocio, facilitando la automatización y optimización de la gestión contable para las PYMES en Colombia. Esta capa actúa como un puente entre la capa de negocio y la capa tecnológica, proporcionando funcionalidades clave que permiten la gestión eficiente de datos financieros, la integración con normativas contables locales e internacionales y la mejora de la experiencia del usuario.

En esta capa, se modelan los componentes de software que componen la plataforma ContaSync Pro, tales como módulos de contabilidad, gestión de nómina, cuentas por cobrar, analítica financiera y acceso móvil. Estos sistemas están diseñados para garantizar la interoperabilidad, escalabilidad y seguridad de la información, asegurando que las empresas puedan gestionar sus operaciones contables de manera precisa y confiable.

Adicionalmente, la capa de aplicación soporta la integración con servicios externos, como entidades bancarias, sistemas fiscales y plataformas de terceros, mediante API y estándares de comunicación seguros. De esta manera, ContaSync Pro no solo optimiza la gestión contable, sino que también se convierte en un ecosistema digital que facilita la toma de decisiones estratégicas y el cumplimiento normativo.

6.2. Vista de comportamiento de la aplicación

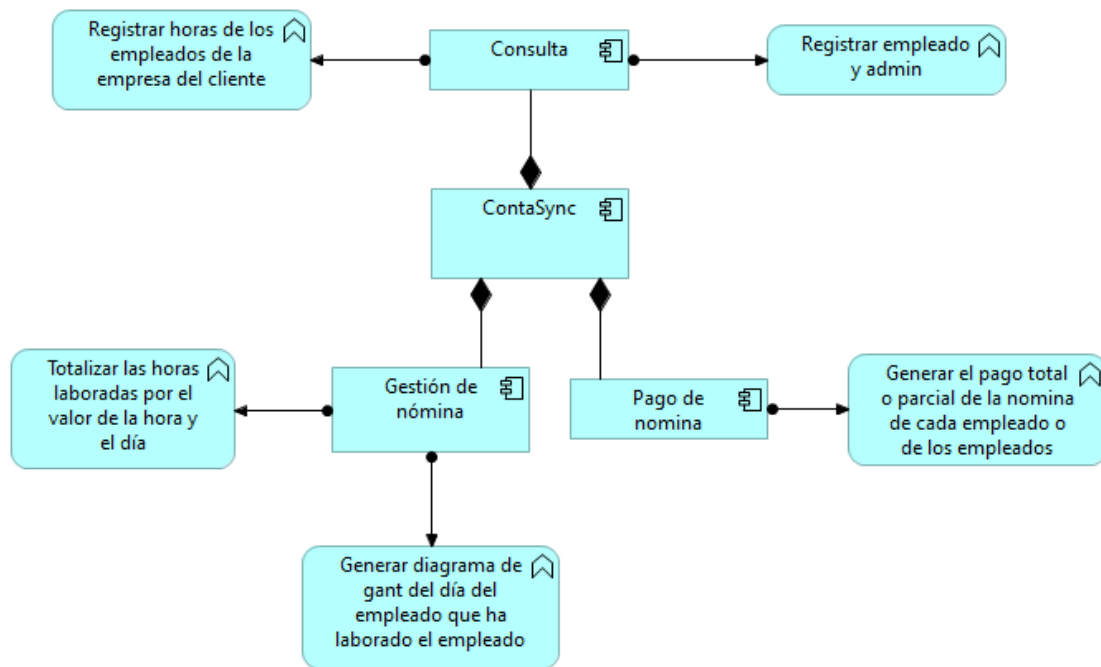


Figura 6.1: Diagrama de comportamiento de la aplicación

6.3. vista de cooperación de la aplicación

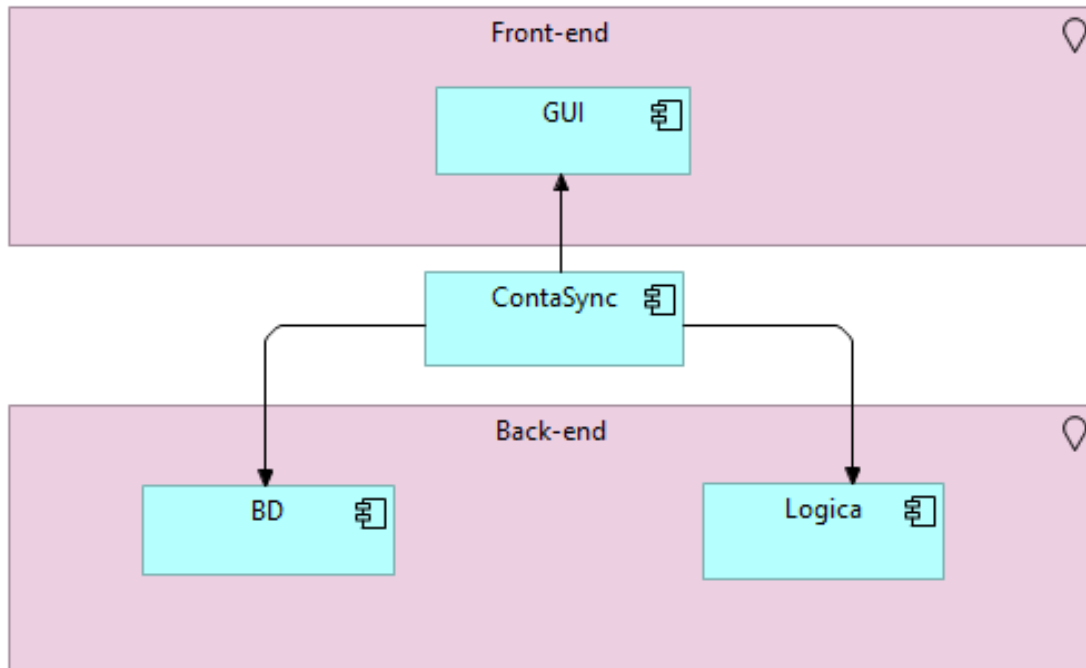


Figura 6.2: Diagrama de cooperación de la aplicación

6.4. vista de estructura de la aplicación

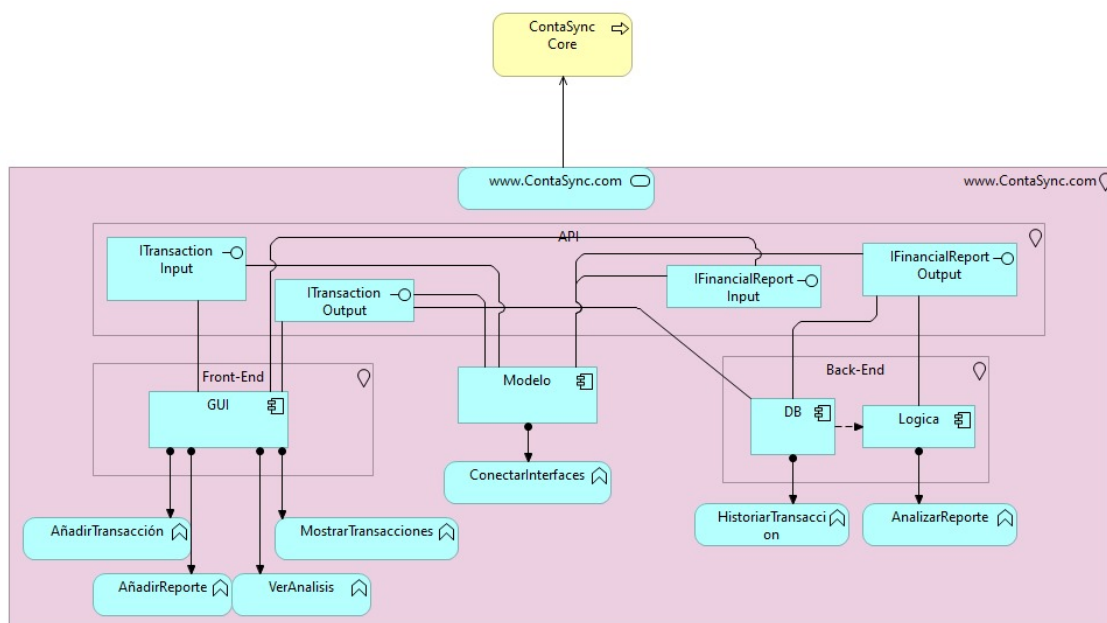


Figura 6.3: Diagrama de la estructura de la aplicación

Capítulo 7

Capa de Tecnología

7.1. Introducción

La capa tecnológica en ContaSync Pro proporciona la infraestructura y los recursos tecnológicos necesarios para soportar la operación eficiente y segura de los sistemas y servicios de la capa de aplicación. Esta capa abarca los componentes de hardware, redes, almacenamiento y plataformas de ejecución que garantizan el desempeño, la disponibilidad y la escalabilidad de la solución.

Dentro de esta capa, se incluyen los servidores, bases de datos, servicios en la nube y mecanismos de comunicación que permiten la ejecución de las aplicaciones de ContaSync Pro con altos estándares de rendimiento y seguridad. La arquitectura tecnológica está diseñada para asegurar la continuidad del negocio, soportando altos volúmenes de transacciones contables y financieras sin comprometer la integridad de los datos.

Adicionalmente, la capa tecnológica implementa estrategias de seguridad informática como encriptación de datos, autenticación multifactor y monitoreo en tiempo real, garantizando la protección de la información sensible de las PYMES. También se establecen protocolos de integración con otros sistemas empresariales y entidades externas mediante interfaces seguras y estándares de interoperabilidad.

Gracias a una infraestructura flexible y escalable basada en tecnologías emergentes como la computación en la nube, la virtualización y la inteligencia artificial, ContaSync Pro está preparado para adaptarse a las necesidades cambiantes del mercado y garantizar un servicio confiable y eficiente a sus usuarios.

7.2. vista de tecnología

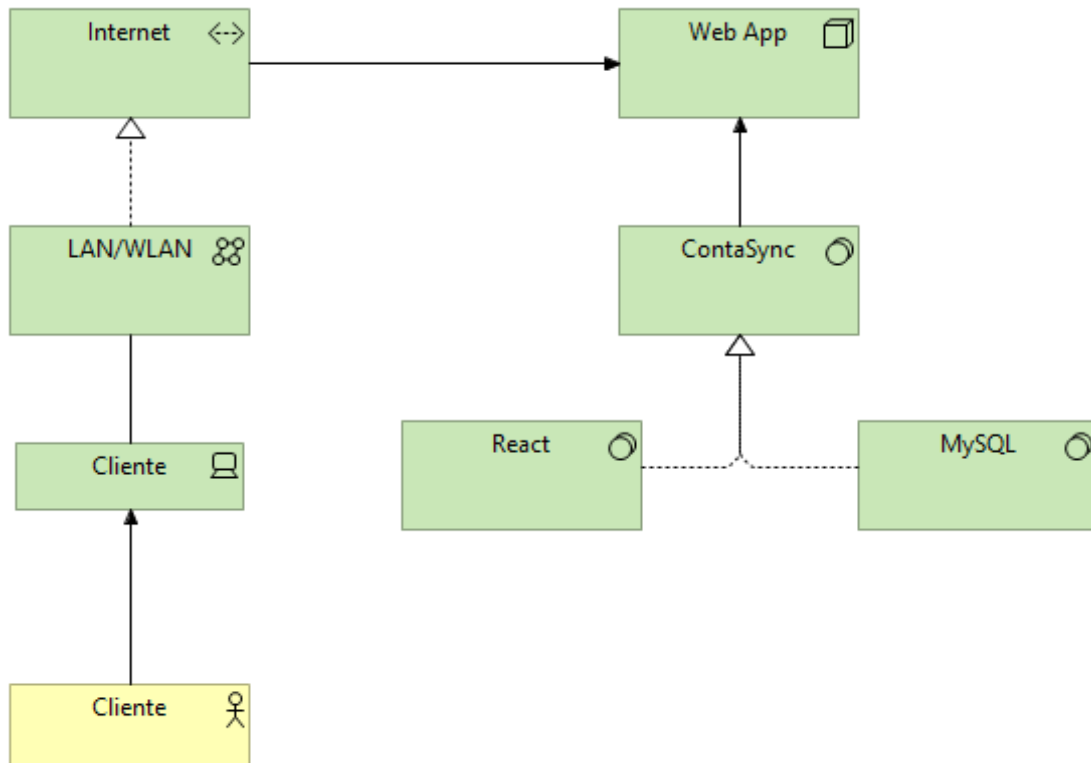


Figura 7.1: Diagrama de la tecnología

7.3. vista de uso de la tecnología

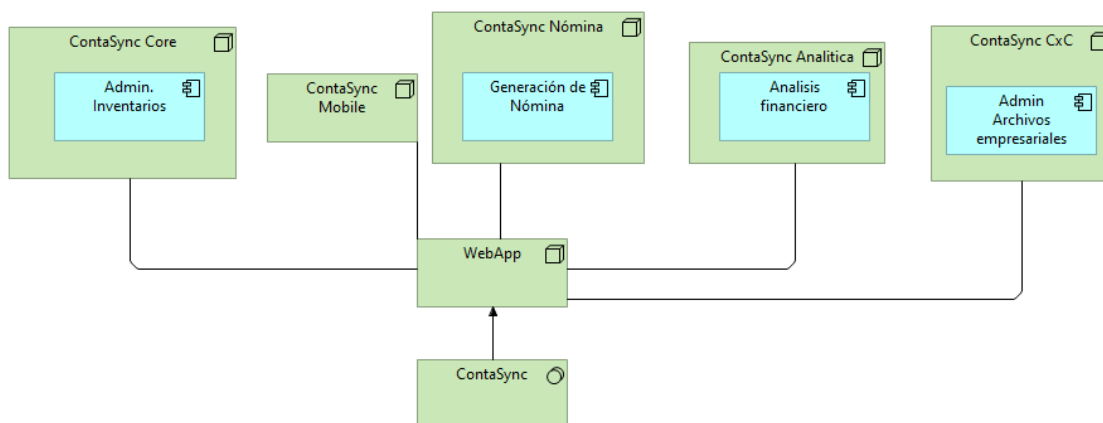


Figura 7.2: Diagrama de uso de la tecnología

7.4. vista de estructura de la información

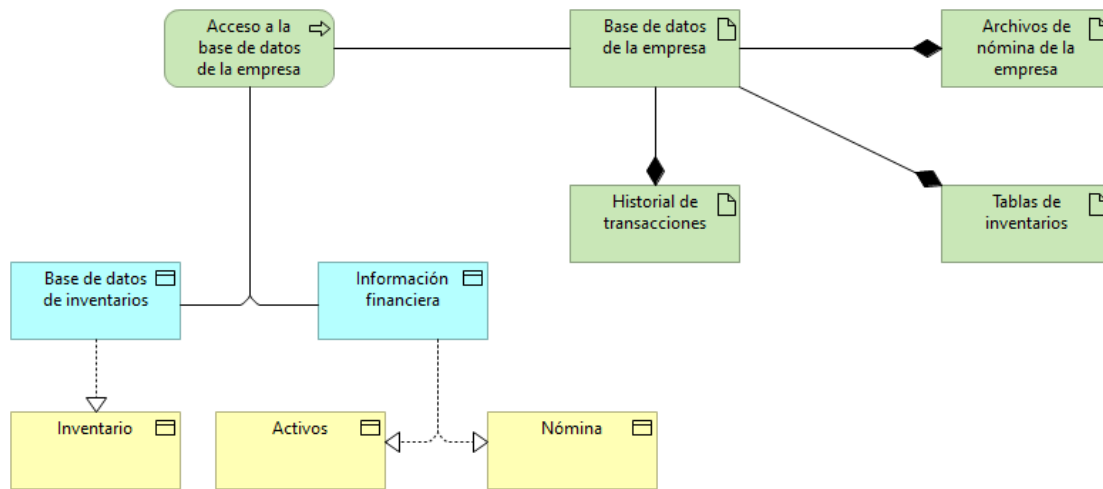


Figura 7.3: Diagrama de la estructura de la información

7.5. vista de Despliegue e Implementación

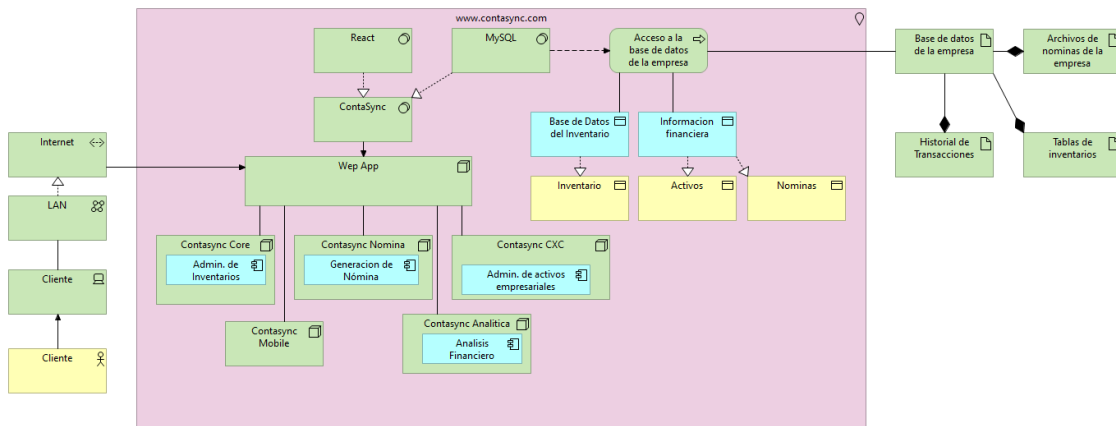


Figura 7.4: Diagrama del Despliegue e Implementación

Capítulo 8

Capa de Migración

8.1. Introducción

La capa de migración en ContaSync Pro define los procesos, estrategias y herramientas utilizadas para trasladar datos, sistemas y aplicaciones desde infraestructuras o soluciones preexistentes a la nueva plataforma de automatización contable. Su propósito es garantizar una transición segura, eficiente y sin interrupciones, minimizando riesgos y asegurando la integridad de la información contable y financiera de las PYMES.

Esta capa abarca diferentes tipos de migración, incluyendo la migración de datos desde sistemas contables tradicionales, la transición de infraestructuras locales a entornos en la nube y la adaptación de aplicaciones a nuevas tecnologías. Se aplican metodologías avanzadas para la extracción, transformación y carga (ETL) de datos, asegurando la compatibilidad y consistencia de la información en el nuevo ecosistema digital.

Adicionalmente, la capa de migración contempla procesos de validación, pruebas y optimización post-migración para garantizar que los sistemas continúen operando con precisión y eficiencia. Se implementan estrategias de respaldo y recuperación para mitigar posibles fallos y garantizar la continuidad del negocio durante la transición.

Gracias a un enfoque estructurado y basado en mejores prácticas, ContaSync Pro permite a las PYMES adoptar soluciones innovadoras sin comprometer la seguridad ni la disponibilidad de sus operaciones, facilitando una evolución tecnológica progresiva y confiable.

8.2. vista de proyecto

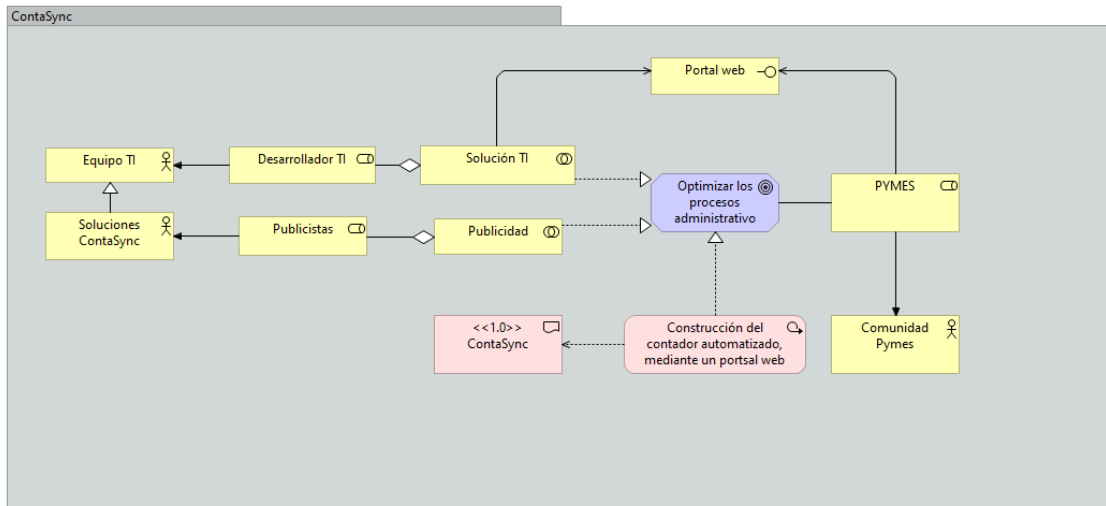


Figura 8.1: Diagrama del proyecto

8.3. vista de migración

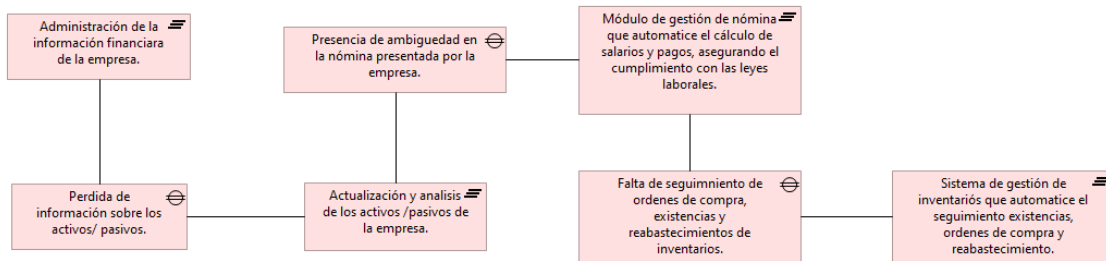


Figura 8.2: Diagrama de Migración

8.4. vista de Implementación/Migración

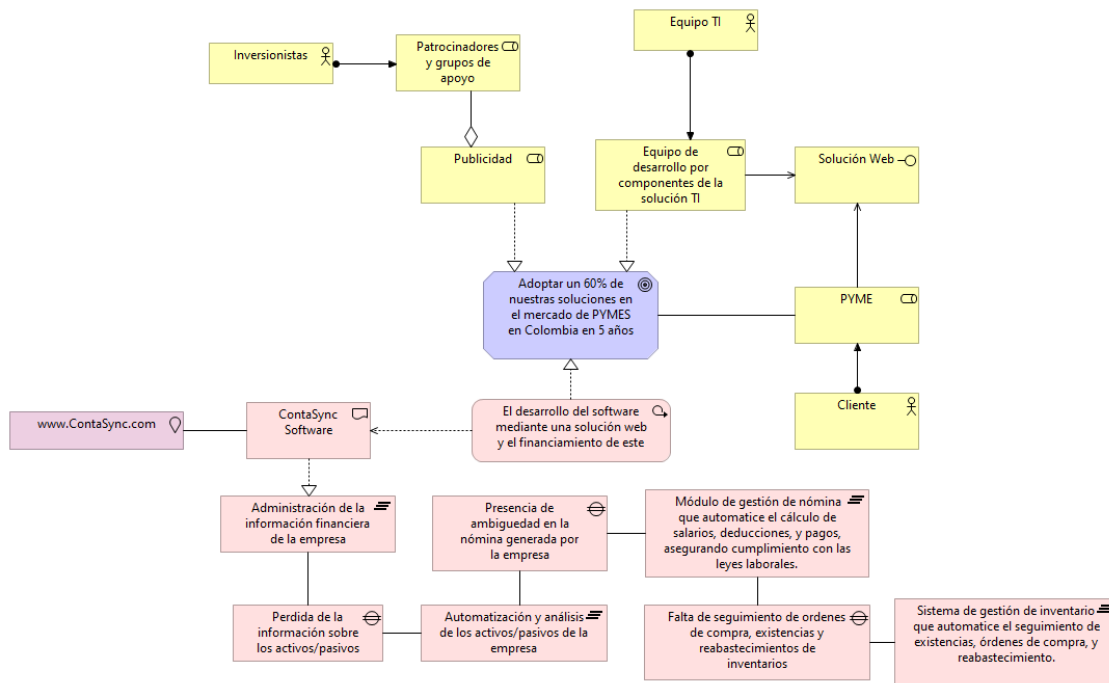


Figura 8.3: Diagrama de Implementación

Parte III

PATRONES

Capítulo 9

Patrones Creacionales

9.1. Patrón Singleton

9.1.1. Problema

En algunas aplicaciones, es necesario garantizar que una clase tenga una única instancia globalmente accesible. Esto es común en la gestión de recursos compartidos como conexiones a bases de datos, registros de logs o manejadores de configuración. Sin un control adecuado, múltiples instancias podrían generar inconsistencias, aumentar el uso de memoria y afectar el rendimiento.

9.1.2. Estructura

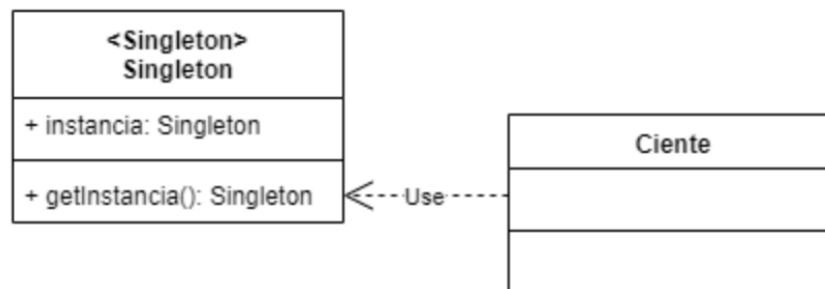


Figura 9.1: Estructura del patrón Singleton en diagrama de clases

9.1.3. Funcionamiento

Cuando un cliente necesita usar la clase Singleton, llama al método `getInstancia()`. Si la instancia no existe, se crea y se almacena. En llamadas posteriores, se devuelve la misma instancia sin volver a crearla. Esto asegura que todos los clientes compartan el mismo objeto, evitando duplicaciones y garantizando un único punto de acceso.

9.1.4. Implementación

Para el caso de el aplicativo ContaSync Nomina, usamos el patrón singleton para implementar una sola instancia que permite a toda la aplicación, conectarse a la base de datos sin que se deba instanciar esta clase mas de una vez, evitando mayor uso de memoria por usuario ya que la conexión a base de datos se usa en repetidas ocasiones durante sus ejecución, ya sea para registrar las horas de un empleado o para realizar queries de usuarios desde la dashboard de administrador (Vea Figura acontinuación)

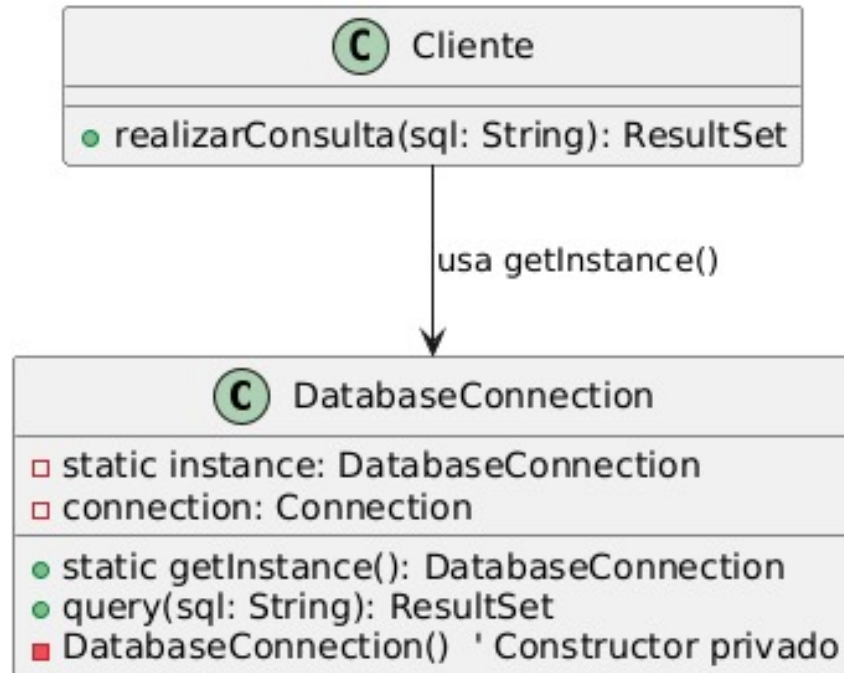


Figura 9.2: Implementación del patrón Singleton en contaSync

9.1.5. Observaciones

- Si la instancia nunca se usa, puede generar un uso innecesario de memoria. Para evitar esto, en algunos casos es preferible usar una instancia bajo demanda (Lazy Initialization) en lugar de una creada desde el inicio (Eager Initialization).[1]
- La implementación del Singleton en DatabaseConnection garantiza que solo exista una conexión única a la base de datos en toda la aplicación. Esto evita la sobrecarga de múltiples conexiones abiertas, optimizando el rendimiento y el uso de recursos.
- Como DatabaseConnection es una instancia única y global, puede ser difícil de mockear o reemplazar en pruebas. Para futuros avances de esta solución lo esperable sería mejorar la testabilidad donde se tendría planeado modificar el diseño para permitir una inyección de dependencias, en lugar de depender directamente de la clase DatabaseConnection.

9.2. Patrón Factory

9.2.1. Problema

En una aplicación que maneja distintos tipos de usuarios (Como el caso de ContaSync Nomina donde usa dos tipos de usuarios Empleado y Admin), la creación de objetos de manera manual puede generar código repetitivo y acoplamiento innecesario. Si se desea agregar más tipos de usuarios en el futuro, modificar directamente el código de instanciación puede ser riesgoso y difícil de mantener.

9.2.2. Estructura

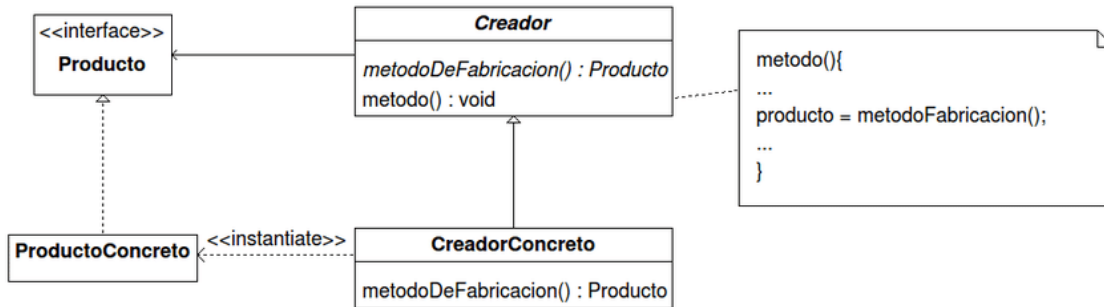


Figura 9.3: Estructura del patrón Factory en diagrama de clases

9.2.3. Funcionamiento

Cuando un cliente necesita crear un usuario, en lugar de instanciar directamente **Empleado** o **Admin**, llama a `UsuarioFactory.crearUsuario(Empleado)` o `UsuarioFactory.crearUsuario(Admin)`, esto nos permite que el código del cliente no dependa directamente de las clases concretas, sino solo de la interfaz **Usuario**.

9.2.4. Implementación

En el caso de ContaSync Nomina, se crea una fabrica de usuarios que permite crear dos tipos de usuarios que, en el contexto de una empresa, seria el supervisor y el empleado, donde el supervisor podrá monitorear a los empleados, y los empleados hacer uso del registro de tiempo para que se puedan registrar en sus horas laborales a la hora de calcular la nomina.

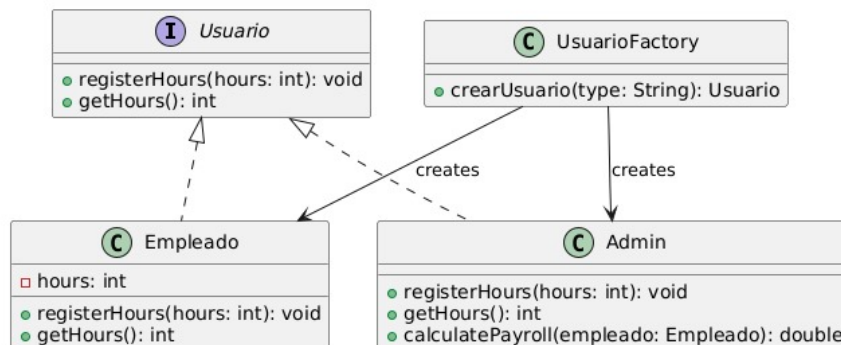


Figura 9.4: Implementación del patrón Factory en contaSync

9.2.5. Observaciones

- Si se necesita agregar un nuevo tipo de usuario (Practicante, Gerent), solo se modifica UsuarioFactory, sin tocar el código donde se crean los usuarios. Esto hace que la aplicación sea más fácil de escalar.
- Si solo existen dos tipos de usuarios (Empleado y Admin) y no se planean más en el futuro, el uso de UsuarioFactory puede ser innecesario. En estos casos, puede ser más eficiente instanciar los objetos directamente.
- El patrón Factory puede combinarse con el Singleton, por ejemplo, si la fábrica necesita compartir ciertos recursos como una conexión a la base de datos (DatabaseConnection). También se puede integrar con Prototype, si se requiere la clonación de usuarios existentes.

Capítulo 10

Estructurales

10.1. Patrón Proxy

10.1.1. Problema

En algunas situaciones, acceder directamente a un objeto puede ser costoso o inseguro. Por ejemplo, si los usuarios interactúan con el sistema sin restricciones, pueden modificar su información sin control provocando brechas de seguridad o filtración de información valiosa. En otras palabras, Patrón Proxy actúa como un intermediario que controla el acceso al objeto real como lo es el usuario, gracias a este patrón, nos permite:

- Restringir o monitorear el acceso a ciertos métodos.
- Diferir la creación del objeto real hasta que sea realmente necesario.
- Mejorar el rendimiento en operaciones costosas o repetitivas.

10.1.2. Estructura

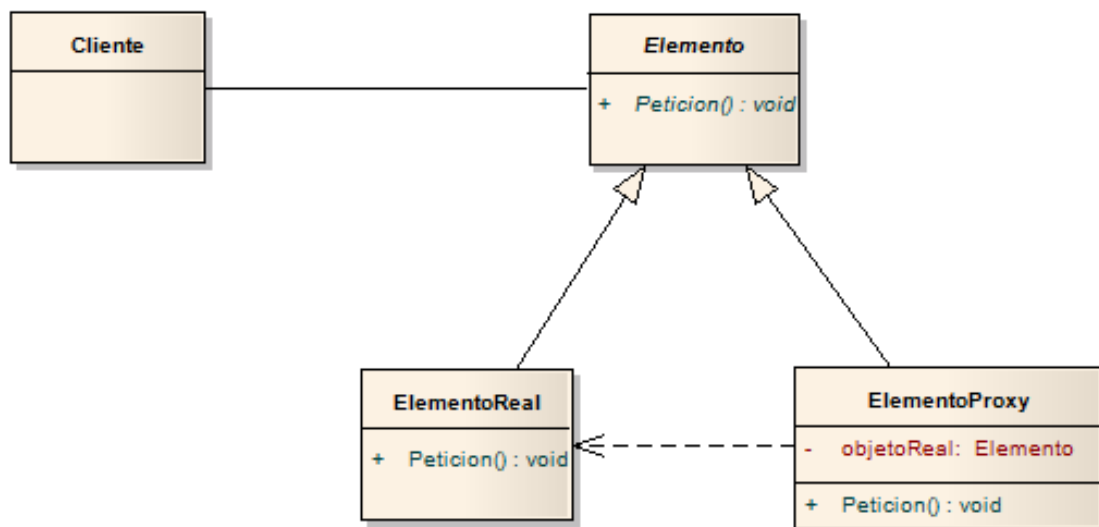


Figura 10.1: Estructura del patrón Proxy en diagrama de clases

10.1.3. Funcionamiento

Cuando el cliente necesita usar un usuario (Empleado o Admin), en lugar de acceder directamente a la instancia real, interactúa con UsuarioProxy.

Al crear un `UsuarioProxy`, se le pasa una instancia real (`Empleado` o `Admin`). Cuando se llama a un método, el Proxy podemos:

- Permitir el acceso y delegar la llamada al objeto real.
- Restringir la operación si el usuario no tiene permisos.
- Registrar acciones antes o después de ejecutar la operación.

10.1.4. Implementación

Al i

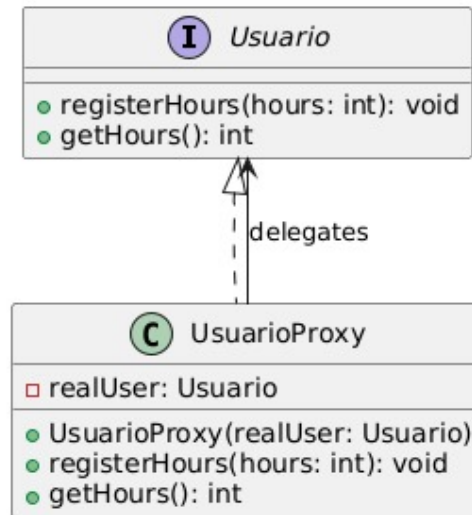


Figura 10.2: Implementación del patrón Proxy en ContaSync

10.1.5. Observaciones

- `UsuarioProxy` puede verificar si un usuario tiene permisos antes de ejecutar acciones sensibles, como registrar horas o calcular nóminas.
- Se puede registrar todas las operaciones realizadas por los usuarios sin modificar `Empleado` o `Admin`, facilitando la auditoría de cambios.
- Si `Usuario` requiere una consulta costosa a la base de datos (`DatabaseConnection`), el Proxy puede diferir la carga del objeto real hasta que realmente se necesite, optimizando recursos.
- Si el control de acceso no es una necesidad importante en el sistema, el uso de un Proxy puede ser innecesario y hacer el código más complejo sin aportar un beneficio real.

10.2. Patrón Decorator

10.2.1. Problema

En sistemas donde los objetos pueden necesitar comportamientos adicionales de forma dinámica, el uso de herencia tradicional puede ser limitante y poco flexible. En este caso, los usuarios (Empleado y Admin) pueden necesitar cálculos adicionales sobre sus horas trabajadas, como el cálculo de horas extras. En lugar de modificar directamente estas clases o crear múltiples subclases, el Patrón Decorator permite añadir funcionalidades sin alterar la estructura base.

10.2.2. Estructura

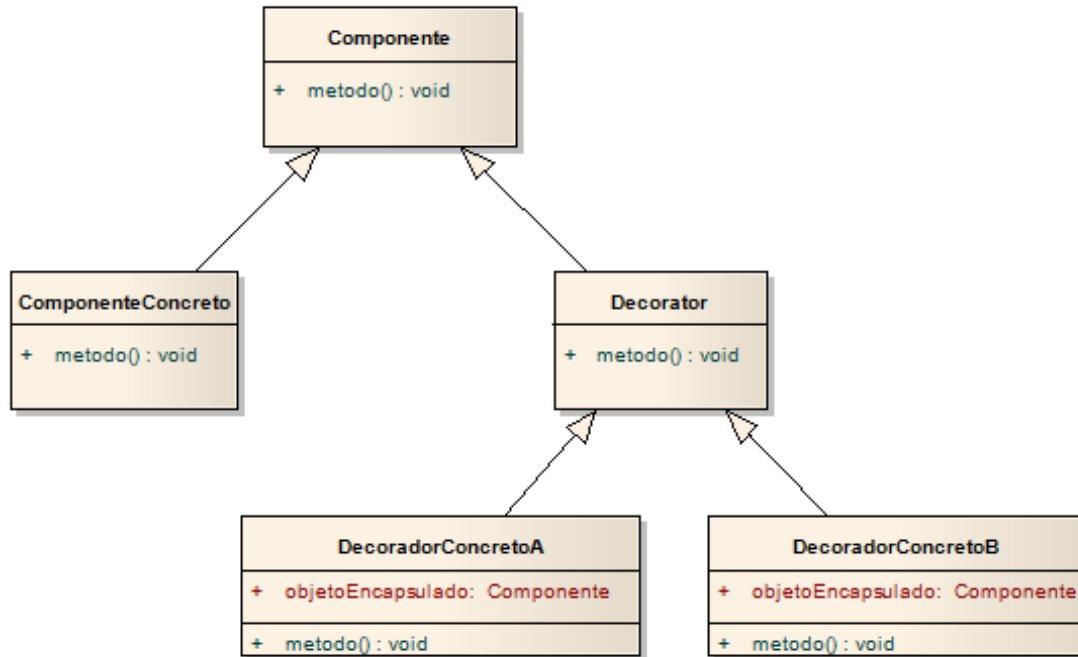


Figura 10.3: Estructura del patrón Decorator en diagrama de clases

10.2.3. Funcionamiento

Cuando un usuario (Empleado o Admin) necesita una funcionalidad adicional, como el cálculo de horas extras, en lugar de modificar su clase directamente, se envuelve en un decorador. Primero, se crea un objeto Empleado normal y se utiliza tal cual. Si en algún momento se requiere calcular sus horas extras, en lugar de modificar la clase Empleado, se pasa este objeto a un decorador como OvertimePayrollDecorator.

10.2.4. Implementación

El decorador toma el usuario original y añade la funcionalidad adicional sin alterar su comportamiento base. Cuando se llama a `getHours()`, el decorador puede modificar o complementar la respuesta del objeto real, sumando horas extras según corresponda. Esto permite una gran flexibilidad, ya que se pueden combinar múltiples decoradores sin necesidad de cambiar el código original del objeto

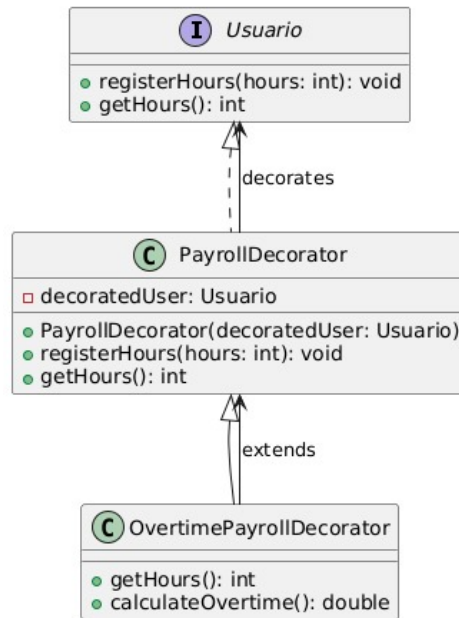


Figura 10.4: Implementación del patrón Decorator en ContaSync

10.2.5. Observaciones

El uso de decoradores en la aplicación mejora la modularidad, ya que permite agregar funcionalidades específicas sin tocar la estructura de Empleado o Admin. Además, evita la proliferación de subclases que combinarían diferentes comportamientos, lo que facilitaría la evolución del código en el futuro. Sin embargo, al anidar múltiples decoradores, el código puede volverse más difícil de seguir, por lo que es recomendable documentar bien su uso y asegurarse de que la estructura no se vuelva innecesariamente compleja.

Capítulo 11

Patrones de Comportamiento

11.1. Patrón Comando

11.1.1. Problema

En muchas aplicaciones, es necesario desacoplar el código que emite una solicitud del código que la ejecuta. Esto es especialmente útil cuando se necesitan funcionalidades como deshacer acciones, ejecutar comandos en cola o registrar operaciones.

11.1.2. Estructura

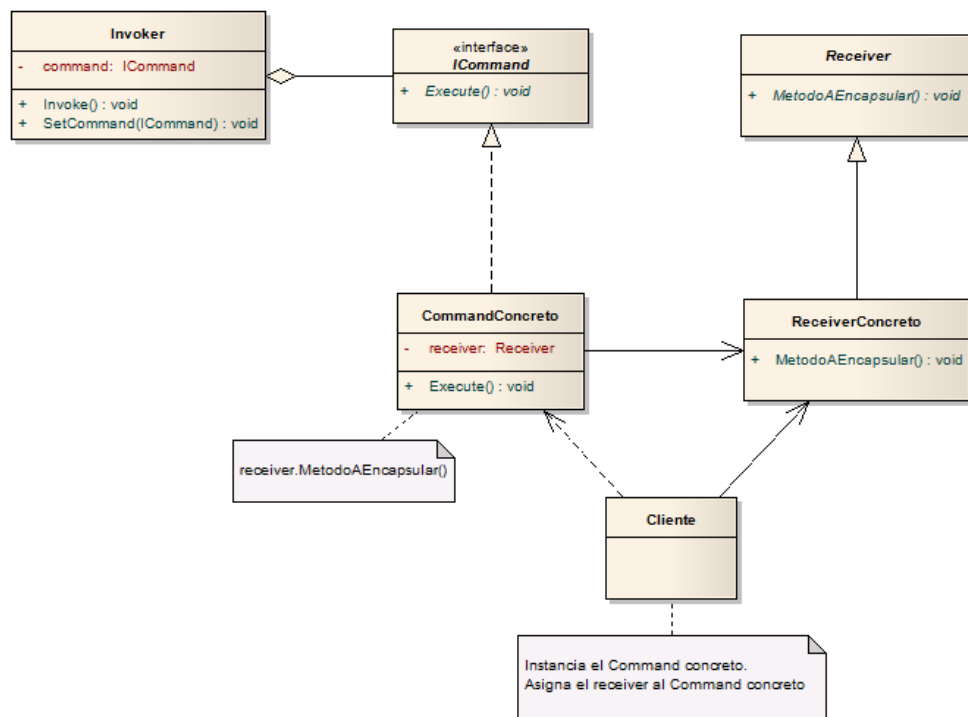


Figura 11.1: Estructura del patrón Singleton en diagrama de clases

11.1.3. Funcionamiento

Cuando se necesita calcular la nómina de un empleado, en lugar de llamar directamente a un método en Admin, se crea un objeto `CalculatePayrollCommand` y se le pasa el Admin y el Empleado correspondientes.

Luego, este comando se ejecuta llamando a su método `execute()`, que delega la operación de cálculo en Admin.

Este enfoque permite almacenar comandos en una lista para ejecución diferida, registrar operaciones realizadas o incluso implementar la opción de deshacer cálculos en el futuro. Además, el patrón hace que el sistema sea más flexible, ya que nuevos comandos pueden ser añadidos sin modificar la estructura base de Admin o Empleado.

11.1.4. Implementación

En este caso, Admin tiene la capacidad de calcular la nómina de un Empleado, pero acoplar directamente este cálculo dentro de Admin puede dificultar la reutilización del código y hacer que el sistema sea menos flexible. En su lugar, el Patrón Command encapsula esta operación en un objeto independiente (CalculatePayrollCommand), permitiendo mayor control sobre su ejecución.

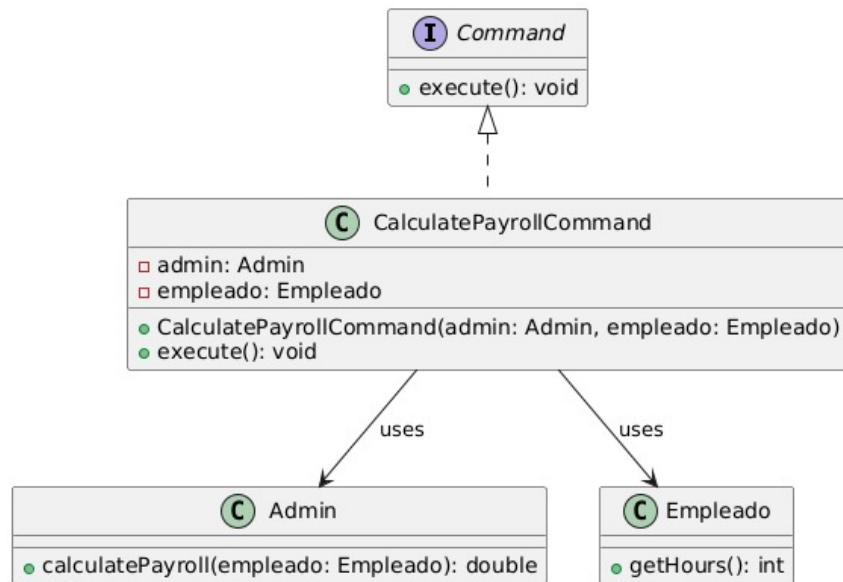


Figura 11.2: Implementación del patrón Comando en contaSync

11.1.5. Observaciones

El uso de comandos en este sistema proporciona un alto grado de flexibilidad y reutilización de código. Se pueden agregar nuevas operaciones sin modificar las clases de usuario, lo que mejora la mantenibilidad. También permite extender funcionalidades en el futuro, como un historial de comandos o una cola de procesamiento. Sin embargo, si el sistema no requiere funcionalidades avanzadas como deshacer o ejecución en cola, el uso del patrón Command podría añadir una capa innecesaria de complejidad.

Parte IV

CIERRE

Referencias

Articulos

- [1] <https://refactoring.guru/es/design-patterns/singleton>. En: ().
- [2] Wenming Liu y Mengfan Bai. “Research on the Application of Fret Pattern in Furniture Intelligent Design”. En: (2020), págs. 209-213. DOI: 10.1109/ICID52250.2020.00051.
- [3] Nadia Nahar y Kazi Sakib. “ACDPR: A Recommendation System for the Creational Design Patterns Using Anti-patterns”. En: 4 (2016), págs. 4-7. DOI: 10.1109/SANER.2016.44.
- [4] Na Wang, Shiwei Wu y Xiaochen Yin. “Research on Parametric Design of Decorative Patterns Based on Color Composition Model;Subtitle;Take Continuous Geometric Patterns as A Case Study”. En: (2020), págs. 338-342. DOI: 10.1109/ICIDDT52279.2020.00067.