



FACULTAD DE POSGRADOS
MAESTRÍA EN GERENCIA DE SISTEMAS Y TECNOLOGÍA
EMPRESARIAL
"FORTALECIMIENTO DE LA FÁBRICA DE SOFTWARE *IN HOUSE*
PARA UNA EMPRESA DE SISTEMAS ERPs"

Profesor Guía

MsC. Germán Pancho

Alumno

Marco Alexander Cárdenas

Año

2020-2021

TABLA DE CONTENIDO

1. INTRODUCCIÓN.....	8
2. FASE PRELIMINAR.....	9
2.1. Contexto.....	9
2.2. <i>Stakeholders</i> y expectativas de valor.....	13
2.3. Organigrama empresarial.....	15
2.3.1. Organización impactada.....	15
2.4. Motivadores.....	17
2.5. Marcos de referencia complementarios.....	19
2.6. Equipo de Arquitectura.....	20
2.6.1. Responsabilidades.....	20
2.7. Catálogo de principios.....	21
2.8. Nivel de madurez de la empresa.....	23
3. VISIONAMIENTO ARQUITECTÓNICO.....	24
3.1. Requerimientos de alto nivel.....	24
3.2. Visionamiento y escenarios de la solución.....	27
3.2.1. Casos de referencia.....	27
3.2.2. Fábrica de software triple A.....	29
3.2.3. Conclusión de visionamiento.....	29
3.3. Análisis de brechas.....	30
3.4. Arquitectura objetivo.....	32
3.4.1. Arquitectura de negocio.....	32
3.4.2. Arquitectura de datos.....	33
3.4.3. Arquitectura de aplicaciones.....	34
3.4.4. Arquitectura de infraestructura base.....	35

4.	ARQUITECTURA DE NEGOCIO	36
4.1.	Arquitectura de negocio actual	36
4.1.1.	Metodología actual de desarrollo de software	37
4.1.2.	Metodología propuesta para desarrollo de software	38
4.1.3.	Análisis de brechas.....	39
4.2.	Arquitectura de negocio objetivo	40
4.2.1.	Procesos	40
4.2.2.	Organización.....	48
4.2.3.	Personal	50
5.	ARQUITECTURA DE APLICACIONES / DATOS.....	57
5.1.	Arquitectura de aplicaciones actual.....	57
5.1.1.	Catálogo de aplicaciones.....	58
5.2.	Arquitectura datos actual	59
5.2.1.	Tipos/fuentes de datos	59
5.3.	Arquitectura de aplicaciones y datos objetivo	61
5.3.1.	Arquitectura de aplicaciones objetivo.....	61
5.3.2.	Arquitectura de datos objetivo.....	63
5.3.3.	Catálogo de aplicaciones objetivo	63
5.3.4.	Análisis de brechas.....	65
6.	ARQUITECTURA DE INFRAESTRUCTURA BASE.....	67
6.1.	Infraestructura base.....	67
6.1.1.	Ambientes de desarrollo de software	68
6.2.	Arquitectura infraestructura base objetivo	71
7.	OPORTUNIDADES Y SOLUCIONES.....	76
7.1.	Arquitectura de negocio	76
7.1.1.	Iniciativa: Implementación de Scrum	76

7.1.2.	Iniciativa: Capacitación al personal del departamento de desarrollo informático en metodología Scrum	89
7.2.	Arquitectura de aplicaciones / Datos.....	92
7.2.1.	Iniciativa: Actualización a versión 14 de Odoos	92
7.2.2.	Iniciativa: Implementación del módulo Scrum en Olympos.....	97
7.2.3.	Iniciativa: Control de actividades procedentes de Scrum en ERP Olympos	102
7.2.4.	Iniciativa: Implementación de SmartGit para control de versionamiento	105
7.3.	Arquitectura de infraestructura base	108
7.3.1.	Iniciativa: Implementación de Servidor AWS	108
7.3.2.	Iniciativa: Reestructuración de entornos de desarrollo e implementación de entornos de integración continua y pre-producción	108
8.	PLAN DE MIGRACIÓN.....	113
8.1.	Análisis de impacto.....	113
8.2.	Fases.....	117
8.3.	Road Map	119
	Bibliografía.....	119

FIGURAS

Figura 1	Fases del método de TOGAF ADM.....	8
Figura 2	Organigrama empresarial	15
Figura 3	Organización impactada.....	16
Figura 4	Business Motivation Model	18
Figura 5	Requerimientos de alto nivel	27
Figura 6	Fábrica 3A.....	29
Figura 7	Análisis de brechas.....	32
Figura 8	Target de arquitectura de negocio	33
Figura 9	Arquitectura de datos.....	33
Figura 10	Arquitectura de aplicaciones y herramientas	34

Figura 11 Arquitectura de infraestructura base	35
Figura 12 Modelo de desarrollo	37
Figura 13 Scrum	38
Figura 14 Análisis de brechas de arquitectura de negocio	39
Figura 15 Fases Scrum.....	41
Figura 16 Fase de Iniciación SRUM	43
Figura 17 Fase de Planificación y Estimación SCRUM	44
Figura 18 Fase de Implementación SRUM	45
Figura 19 Fase de revisión y retrospectiva SCRUM.....	46
Figura 20 Fase de lanzamiento SRUM.....	47
Figura 21 Aplicaciones actuales	58
Figura 22 Tipos de datos.....	59
Figura 23 Objetivo de la Arquitectura de aplicaciones	62
Figura 24 Análisis de la arquitectura de aplicaciones.....	65
Figura 25 Infraestructura actual.....	68
Figura 26 Entornos de desarrollo	70
Figura 28 Arquitectura de infraestructura base	74
Figura 29 Eventos de Scrum.....	80
Figura 29 Precio de nueva versión de Odoos	97
Figura 30 Roles Scrum	98
Figura 32 Control de Sprints	99
Figura 32 Historias de usuarios	100
Figura 33 Reportes.....	101
Figura 34 Tipos de gráficos de evolución de Sprint.....	103
Figura 35 Proyectos	103
Figura 36 Proyectos con todos los detalles	104
Figura 37 Detalles de Backlog	104
Figura 38 Reuniones.....	105
Figura 39 Entornos de desarrollo	111

TABLAS

Tabla 1 Empresas que trabajan con ProtelSoftware S.A	10
Tabla 2 Servicios ERP.....	11
Tabla 3 Número de empleados actual por departamento de la Empresa SW	13
Tabla 4 Stakeholders y expectativas de valor.....	14
Tabla 5 Impacto en la organización.....	17
Tabla 6 Marcos de referencia complementarios para la Empresa SW	19
Tabla 7 Equipo de Arquitectura.....	20
Tabla 8 Matriz Raci.....	21
Tabla 9 Catálogo de Principios	22
Tabla 10 Nivel de madurez de la Empresa	23
Tabla 11 Brechas de visionamiento	31
Tabla 12 Rol actual vs Rol objetivo	52
Tabla 13 Catálogo de aplicaciones objetivo	64
Tabla 14 Brechas arquitectura de aplicaciones- datos	65
Tabla 15 Características de infraestructura base	67
Tabla 16 Características de servidor propuesto	71
Tabla 17 Iniciativas	76
Tabla 18 Impacto	114
Tabla 19 Esfuerzo.....	116
Tabla 20 Fases.....	118

RESUMEN

ProtelSoftware S.A. es una empresa de software, constituida en el Ecuador, que tiene como objetivo la venta, monitoreo y soporte de sistemas informáticos, orientados a la contabilidad y las finanzas.

Actualmente la empresa tiene un sin número de problemas entorno a la fábrica de software. Estos radican en la falta de control de versionamiento, inexistencia de metodologías ágiles y uso de versiones antiguas de Odoo (Plataforma en la cual corren todos los sistemas). Lo que ha generado perdidas de clientes y potenciales compradores, errores en módulos que se encuentran en producción, retraso en la entrega de nuevas soluciones, entre otras.

El objetivo de este trabajo es dar solución a los contratiempos antes mencionados, mediante la aplicación de TOGAF, el cual permite tener una idea clara del estado actual de la fabrica y software y mediante su aplicación obtener una fábrica de calidad estándar.

Para este propósito se implementará de SCRUM, control de versiona miento con GitHub, actualización de Odoo, y la reestructuración de los entornos de desarrollo.

1. INTRODUCCIÓN

El presente proyecto es una alternativa para fortalecer la fábrica de software (*concern*) aplicando la metodología de TOGAF ADM; en la empresa ProtelSoftware S.A.

Este trabajo, consta de 10 fases, ocho de las cuales se implementarán en este ejercicio, a continuación las describimos.

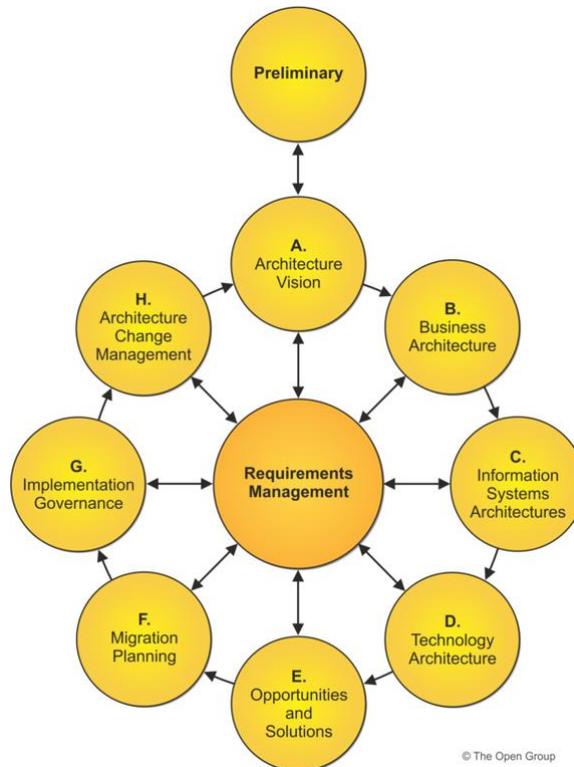


Figura 1 Fases del método de TOGAF ADM

La metodología ADM de Togaf, es una guía para resolver un problema empresarial complejo, se trabajan con metodologías alternas, que aporten a la resolución del problema, siempre orientados a fortalecer el modelo de negocio de una fábrica de software.

Este ejercicio se implementará en la empresa “ProtelSoftware.SA”. dedicada a la “fabricación” o desarrollo de herramientas (módulos) compatibles con un ERP Open Source, además, se plantea una estrategia de “Fortalecimiento de la Fábrica de Software in House.

2. FASE PRELIMINAR

2.1. Contexto

La industria de Software a nivel mundial ha presentado un crecimiento exponencial, busca permanentemente incrementar su productividad, a efectos de contribuir al desarrollo de un sinnúmero de empresas de casi todos los sectores de la economía, he incluso ha incursionado y cambiado el estilo de vida de las personas

A pesar de su importancia, en Ecuador esta industria representa apenas el 0.5% del PIB y ocupa el penúltimo lugar en la región, en términos de exportaciones de servicios de TICs. (Mauro Acebo Plaza, 2017)

En Ecuador la industria de software genera ventas del orden de USD 500 millones por año (0.5% del PIB), con un crecimiento anual de 17%,

Está industria de servicios está constituida por alrededor de 700 empresas y proporcionan empleo a más de 10 mil personas, principalmente desarrolladores (35%) e implementadores (24%). (Mauro Acebo Plaza, 2017))

Este es el caso de ProtelSoftware S.A. empresa familiar con más de 22 años de experiencia, dedicada al análisis, diseño, desarrollo e implementación de sistemas Contable – Financieros.

Se ha especializado en ofrecer soluciones de ERP (Sistemas de planificación de recursos empresariales) de alta calidad, utilizando tecnologías Open Source.

ProtelSoftware S.A. ha alcanzado varias certificaciones, que le permiten alcanzar altos índices de calidad en todos los servicios que ofrece, cuenta con:

- Certificación de Calidad de la Norma ISO 9001:2015 en mayo 2018, con la empresa Bureau Veritas.

- Ocho funcionarios con el Certificado del Programa Internacional en Gerencia de Proyectos, basado en la Metodología del P.M.I., tanto en la Universidad de Asturias, como en la Universidad de Catalunya, de Colombia.

Lo anterior y el manejar Políticas empresariales enfocadas a contratar únicamente profesionales, de al menos tercer nivel, con un alto grado de profesionalismo y experiencia probada, han derivado, en que se sitúe entre el 20% de empresas, a nivel del país, con ventas sobre los \$300,000 anuales.

Cuenta con 33 empleados, distribuidos en el departamento de Desarrollo Informático, Departamento de Implementación, Atención al cliente y en el área administrativa.

ProtelSoftware S.A., opera en varias provincias del Ecuador, dentro de su cartera de clientes podemos mencionar a:

Cliente/Entidad	Provincia
EPMSA	Quito
EPN	Quito
GADs	Pichincha, Ibarra, Otavalo, Jaramijó, San Vicente, Rocafuerte, Manta, GADs, Machala, Gualaquiza, Babahoyo, Flavio Alfaro, Montecristi, Tosagua
ATM	Guayas
Terminal Terrestre	Manta
Empresa Pública de Movilidad	Quito
Patronato Municipal	Santo Domingo

Tabla 1 Empresas que trabajan con ProtelSoftware S.A

Ofrece a sus clientes los siguientes sistemas:

- **Sistema Contable Financiero OLYMPO:** Sistema modular e integrado, que tiene catorce módulos totalmente integrados, tanto para Empresas Privadas como para Empresas Públicas y ONG´s
- **Sistema de Gestión Empresarial ERP OLYMPO:** Es el software de negocios en evolución más rápido en el mundo. OpenERP ahora tiene un conjunto completo de aplicaciones de negocios, cubriendo todas sus necesidades desde Website/Ecommerce hasta manufactura, inventario y contabilidad, todo perfectamente integrado. Es la primera vez en la historia que un editor de software alcanza tal cobertura funcional.

A continuación, se describen los servicios entregados en este sistema

Producto	Servicios
Sistema Contable Financiero ERP	Implementación
	Acompañamiento
	Mantenimiento
	Capitaciones, Instalación y Administración de Servidores (Sistemas Operativos y Bases de Datos) Físicos y Clouds, Desarrollo de Software a la medida.

Tabla 2 Servicios ERP

Estos sistemas permiten:

- Optimizar flujos financieros

- Contar con información oportuna que apoye la toma de las decisiones estratégicas y administrativas
- Mantener mayor control y trazabilidad.
- Permite la integración con clientes y proveedores.
- Es un ERP modular, parametrizable, que puede ser personalizado de acuerdo con las necesidades del cliente.

Todos estos sistemas deben ir evolucionando y adaptándose a las necesidades del cliente, lo que genera un sin número de retos, a ser superados por el departamento de desarrollo informático, estas exigencias permanentes han devenido en que se entreguen sistemas inestables debido a:

- Falta de control de versiones
- Carencia de entornos de prueba
- Falta de documentación y errores en el repositorio centralizado

Estas falencias han impactado en la organización de la siguiente forma:

- Incremento permanente del gasto operativo (OPEX), afectando la rentabilidad de la empresa.
- Insatisfacción en los clientes
- Dificultad en el mantenimiento de soluciones antiguas, soporte e implementación de nuevos módulos.
- Incumplimiento de plazos y cronogramas
- Mala imagen de la empresa

Esta propuesta plantea una reestructuración integral en todas las operaciones fábrica de software in house y procura asegurar el:

- Cumplimiento oportuno de requerimientos

- Fortalecer la calidad y desempeño de los desarrollos
- Minimizar costos
- Incremento sostenido de nuevos proyectos
- Eliminar errores en producción.

2.2. Stakeholders y expectativas de valor

La empresa ProtelSoftware S.A. actualmente cuenta con un total de 33 empleados repartidos por departamentos de la siguiente manera.

Área / Departamento	Cargo	Número de empleados
Gerencia	Gerente General	1
Departamento de Gestión	Coordinador de Gestión	1
	Asistente Contable	1
	Asistente Administrativo	1
	Asistente Comercial	1
	Auxiliar de Servicios	1
Desarrollo Informático	Gerente del área	1
	Coordinador de área	1
	Analista Programador	8
	Técnico Informático	1
	Técnico de Documentación	1
Implementación	Gerente de área	1
	Técnico Implementador	11
	Recepcionista de Servicio al Cliente	1
	Secretario de implementación	2
	Total	33

Tabla 3 Número de empleados actual por departamento de la Empresa SW

La siguiente tabla muestra una visión de la organización, parametrizada en función de los niveles de comprensión y compromiso, tanto actuales como esperados.

DEPARTAMENTO	CARGO	COMPENSIÓN ACTUAL	SE REQUIERE LA COMPENSIÓN	COMPROMISO ACTUAL	COMPROMISO REQUERIDO	EXPECTATIVAS
Gerencia	Gerente General	3	5	4	5	Obtener mayor cantidad de desarrollos al año y bajar los costos operativos
Departamento de gestión	Coordinador de Gestión	2	5	1	4	Obtener soluciones informáticas a menor costo y tiempo
	Asistente Contable	1	2	0	2	Cumplir con los presupuestos de desarrollo
	Asistente Administrativo	1	2	0	2	
	Asistente Comercial	0	2	0	2	Tener mayor cantidad de ofertas comerciales
	Auxiliar de Servicios	0	1	1	1	
Desarrollo informático	Gerente del área	3	5	4	5	Contar con un departamento eficiente, que responda de manera ágil a nuevos desarrollos, soporte y mantenimiento de ERPs
	Coordinador de área	4	5	4	5	Sistemas informáticos, con resultados óptimos y de fácil adaptación. Contar con entornos de desarrollo defonidos
	Analista Programador	4	5	2	5	Desarrollar software de manera ordenada, cumpliendo objetivos y cronogramas
	Técnico Informático	3	5	3	5	Obtener desarrollos de fácil mantenimiento y control de cambios
	Técnico de Documentación	5	5	5	5	Realizar documentación acorde a lo planificado. Sistemas centralizados de información
Implementación	Gerente de área	2	4	2	5	Sistemas informáticos sin errores
	Técnico Implementador	2	4	2	5	Conseguir clientes satisfechos y cumplimiento de cronogramas
	Recepcionista de Servicio al Cliente	2	3	1	2	Conseguir clientes satisfechos
	Secretario de implementación	2	3	1	2	Conseguir clientes satisfechos
ESCALA DE 0 A 5						

■	Nivel alto de comprensión y compromiso
■	Nivel medio de comprensión y compromiso
■	Nivel bajo de comprensión y compromiso

Tabla 4 Stakeholders y expectativas de valor

2.3. Organigrama empresarial

A continuación, se representa la actual estructura organizacional de la empresa

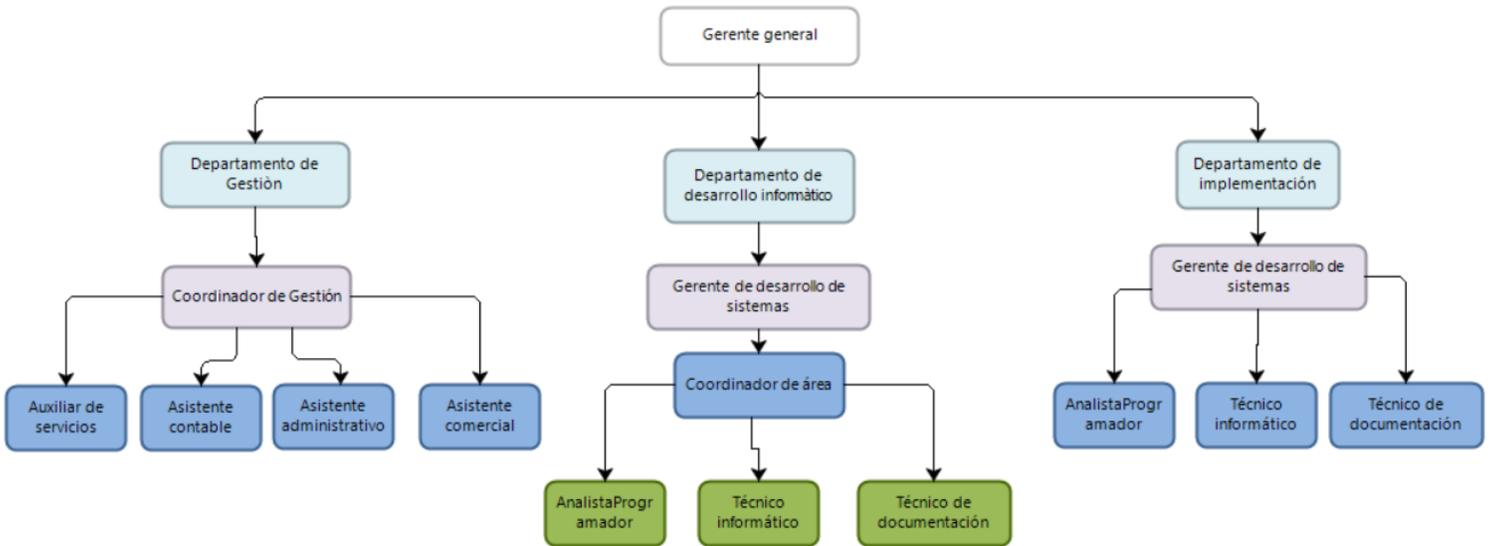


Figura 2 Organigrama empresarial

2.3.1. Organización impactada

El departamento con mayor carga operativa (impacto) es la de desarrollo informático (Color verde), seguidamente con menor impacto (Color azul) los departamentos de gestión e implementación.



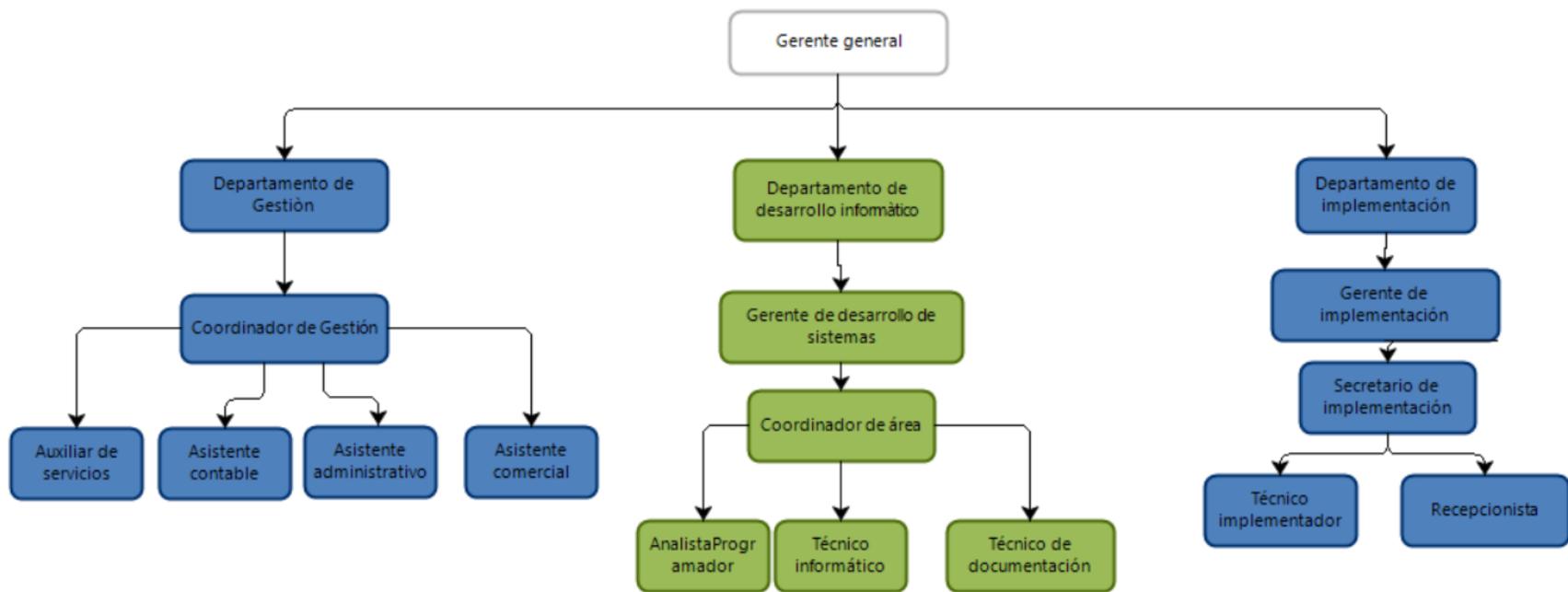


Figura 3 Organización impactada

En la siguiente tabla se describe el impacto y las metas que desea cada una de las áreas de la empresa con la implementación de este proyecto

Departamento	Impacto	Descripción
Gerencia	Alto	<ul style="list-style-type: none"> • Obtener mayor cantidad de módulos para el ERP, que satisfagan las necesidades de los nuevos clientes. • Desarrollo personalizado • Cero errores en producción • Fabrica software eficiente, que genere mayor cantidad de soluciones • Cumplimiento de cronogramas • Satisfacción del cliente.
Departamento de Gestión	Bajo	<ul style="list-style-type: none"> • Soluciones informáticas a menor costo y tiempo. • Optimización de Recursos <i>OpEx</i> y <i>CapEx</i>.
Departamento de desarrollo informático	Alto	<ul style="list-style-type: none"> • Departamento de desarrollo informático eficiente, que utilice metodologías ágiles que permitan. <ul style="list-style-type: none"> ○ Control de versionamiento ○ Manejo de entornos de desarrollo ○ Control de cambios ○ Documentación apropiada ○ Cero errores en entornos de producción. ○ Personalización de nuevos desarrollos o clientes. ○ Cumplimiento de cronogramas.
Departamento de implementación	Bajo	<ul style="list-style-type: none"> • Mayor cantidad de módulos y clientes • Satisfacción del cliente.

Tabla 5 Impacto en la organización

2.4. Motivadores

El uso de la herramienta *Business Motivation Model* (BMM), entendido como un esquema que permite ordenar un plan de negocios en forma lógica, establece los motivos que generan su implementación, basados en el Modelo Motivacional de la Empresa ProtelSoftware S.A. y se determina la relación entre los elementos de cambio, influenciadores alineados a las metas y objetivos actuales de la misma

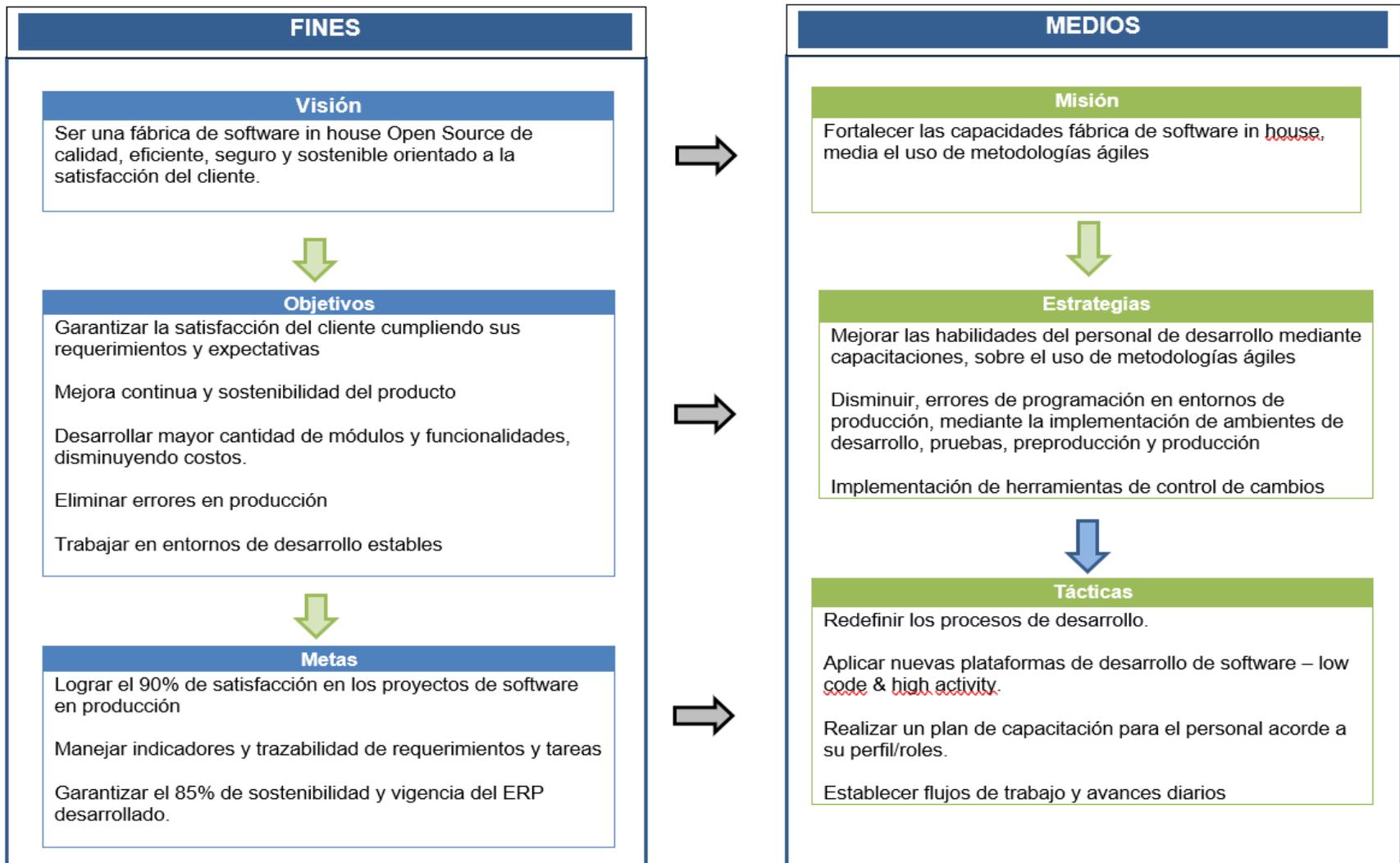


Figura 4 Business Motivation Model

2.5. Marcos de referencia complementarios

Para la ejecución de este proyecto se plantea el uso de a, CMMi, y Scrum como muestra esta gráfica:

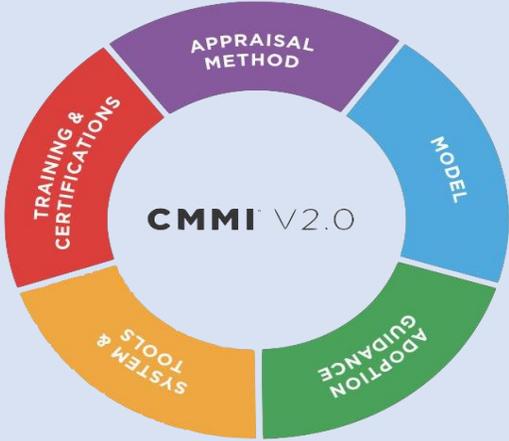
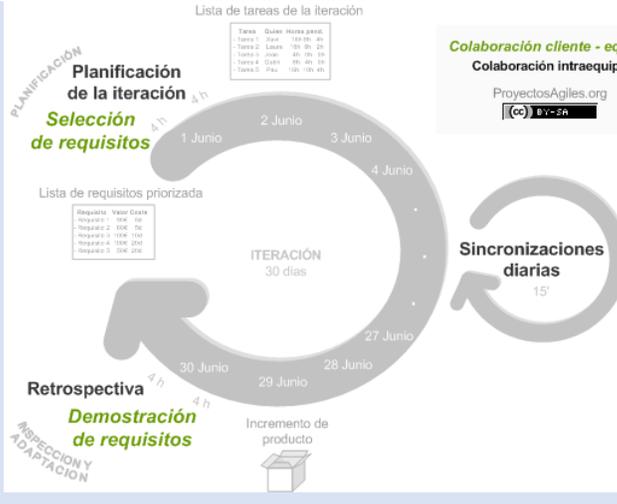
Área	Referencia	Descripción
<p>Departamento de Desarrollo Informático</p>		<p>CMMi es un marco definido como el Modelo de Madurez de Capacidades Integrado que se aplica en una escala del 1 al 5 y que integra disciplinas como sistemas y software en un solo marco de trabajo. (Soft)</p>
<p>Scrum</p>		<p>El Scrum es un proceso que permite el trabajo colaborativo de la organización, permite la productividad, flexibilidad y competitividad (org, s.f.)</p>

Tabla 6 Marcos de referencia complementarios para la Empresa SW

2.6. Equipo de Arquitectura

2.6.1. Responsabilidades

A continuación, se describen las funciones de cada uno de los responsables de la implementación de este proyecto

Rol	Responsable	Descripción de rol
Arquitecto Líder	Marco Cárdenas	Encargado del control y respuesta del proyecto, aplicando la metodología ADM de Togaf
Arquitecto de negocio	Gerente general	Determinar cómo debe cambiar la operación del negocio para apoyar el fortalecimiento de la fábrica de software.
Arquitecto de datos	Gerente de implementación	Trabaja de cerca con el cliente y el “arquitecto de tecnología” siendo capaz de traducir los requisitos empresariales del cliente, en datos manejables
Arquitecto de tecnología	Gerente de desarrollo informático	Establecer un programa para el desarrollo de nuevas soluciones, y acompañamiento continuo.

Tabla 7 Equipo de Arquitectura

2.6.1.1. Matriz Raci

A continuación, se describe la asignación de responsabilidades de cada involucrado con sus actividades y recursos.

Actividades y recursos	Arquitecto Líder	Arquitecto de tecnología	Arquitecto de negocio	Arquitecto de datos
Desarrollo de metodología	RA	I	I	I
Canales de comunicación entre <i>stakeholders</i>	RA	RA	R	I
Fortalecimiento de la fábrica de software	RA	RA	I	I
Desarrollo de software	RA	RA	I	I
Proponer nuevas herramientas	RA	C	I	R
Valor a los datos	RA	A	I	RA
Despliegue de soluciones	RA	RA	I	I
Proponer soluciones tecnológicas	A	RA	I	I
Definir catálogo de productos	A	A	RA	I

Tabla 8 Matriz Raci

2.7. Catálogo de principios

A continuación, se detallan los lineamientos prioritarios que guiarán al Equipo de Arquitectura y servirán para evitar conflictos al momento de la ejecución del trabajo:

Tipo	Nombre	Principio	Justificación	Implicaciones
Gobierno	Alineamiento Estratégico	Desempeñar lo establecido en la AE para alcanzar los objetivos de la empresa	Es importante el compromiso de todos los actores y componentes de la AE para mejorar los resultados	Implementar una nueva metodología de trabajo, capacitaciones internas y socialización
Tecnología	<i>Open source</i>	Los enfoques de desarrollo y las herramientas de apoyo a la fábrica serán basados en <i>Open Source</i>	La mayoría de las empresas donde vamos a implementar son empresas públicas y el presidente anterior la única forma de implementar plataformas de apoyo en empresas del sector público es a través de opciones de software libre	La única opción válida será ubicar la herramienta Odoos de software libre
Tecnología	Incidencias y Tareas	Se generará una incidencia para cada requerimiento del cliente	Tener la posibilidad de intervenir sobre el código para establecer personalizaciones sobre las funcionalidades	Incrementar tareas
Tecnología	Tareas	Solamente el Gerente de Desarrollo puede generar tareas	Revisa el requerimiento y determina la ejecución del mismo	Analiza factibilidad, crea tarea, establece tiempos y verifica ejecución de la tarea
Tecnología	Prácticas ágiles en los procesos	Aseguramiento de la calidad transversalmente	Innovación, permite dar fortalecimiento al servicio	Fortalecer las etapas de diseño, desarrollo, pruebas y soporte.

Tabla 9 Catálogo de Principios

2.8. Nivel de madurez de la empresa

Para continuar con la siguiente fase, es importante realizar la valoración de madurez de la práctica de AE en la organización, para entender mejor las fases, su resultado se muestra en la siguiente matriz:

Elemento de AE/ Nivel de Madurez AE		Nivel 0 No Definido	Nivel 1 <i>Ad Hoc</i>	Nivel 2 En desarrollo	Nivel 3 Definido	Nivel 4 Gestionado	Nivel 5 Monitoreando
1	Procesos de arquitectura	Red	Blanco	Blanco	Verde	Blanco	Blanco
2	Desarrollo de la arquitectura	Red	Blanco	Verde	Blanco	Blanco	Blanco
3	Alineación con el negocio	Red	Blanco	Blanco	Verde	Blanco	Blanco
4	Participación de la alta gerencia	Blanco	Red	Blanco	Verde	Blanco	Blanco
5	Participación de la entidad encargada de las operaciones de TI	Blanco	Red	Verde	Blanco	Blanco	Blanco
6	Comunicación de Arquitectura	Red	Blanco	Verde	Blanco	Blanco	Blanco
7	Seguridad de TI	Red	Blanco	Verde	Blanco	Blanco	Blanco
8	Gobierno de Arquitectura	Red	Blanco	Verde	Blanco	Blanco	Blanco
9	Estrategia de inversión y adquisiciones de TI	Blanco	Red	Blanco	Verde	Blanco	Blanco
10	Talleres de sensibilidad y entrenamiento	Blanco	Blanco	Blanco	Blanco	Verde	Blanco

Red	Estado actual
Verde	Estado deseado

Tabla 10 Nivel de madurez de la Empresa

3.

3. VISIONAMIENTO ARQUITECTÓNICO

Se desarrolla una revisión, de las variables de investigación con el fin de entender las conceptualizaciones y requerimientos que se deben seguir para llegar a un alto nivel en la arquitectura del proyecto.

En esta fase se tomará en consideración las cuatro dimensiones de la Arquitectura Empresarial, de acuerdo con la metodología ADM de Togaf.

- Arquitectura de negocio
- Arquitectura de aplicaciones
- Arquitectura de datos
- Arquitectura de infraestructura

3.1. Requerimientos de alto nivel

Los requerimientos de alto nivel para el fortalecimiento de la fábrica de software en ProtelSoftwareS.A, son los siguientes:

Tecnología

- Control de versionamiento, mediante el uso de herramientas que permitan gestionar los cambios en el código fuente a lo largo del tiempo de desarrollo e implementación.
- Implementación de QA (Control de calidad) que no se limite a detectar fallos, sino a anticiparse y verificar toda la casuística posible para ser validada por todos los implicados.
- Crear ambientes de desarrollo adecuados, tomando en consideración que, en la empresa actualmente demanda la ejecución simultáneamente de múltiples proyectos, con tiempos de puesta en producción exigentes, lo que conlleva a que 8

programadores compartan los mismos ambientes de desarrollo, generando errores en producción.

Se establecerán entornos de:

- Desarrollo
 - Pruebas
 - Pre-Producción
 - Producción
-
- Uso de metodologías ágiles que permitan diseñar y construir un producto correcto, adaptándose de forma flexible e inmediata a las necesidades del cliente
 - Generar de manera correcta y completa las incidencias, requerimientos y/o solicitudes de correcciones en el sistema de gestión empresarial ERP OLYMPO.
 - Mejorar el tratamiento de los datos.

Procesos

- Realizar revisiones y reuniones periódicas en el departamento de desarrollo informático, para definir alcances, avances y problemas
- Generar documentación del software, actualización o cambios
- Retroalimentación de errores, daños o problemas en el transcurso del desarrollo
- Revisiones de versionamiento, código fuente y entornos de desarrollo.
- Gestión de proyectos y requerimientos
- Controlar y planificar de forma ágil los proyectos que tienen gran cantidad de cambios de última hora
- Todo esto generará una reducción significativa de los costos de operación.

Negocio y personas

- Mejorar el perfil de las personas que conforman el departamento de sistemas.
- Capacitación constante al personal del departamento de desarrollo, sobre metodologías ágiles, nuevas plataformas, tecnología, innovación, etc.
- Disminuir costos operativos de desarrollo
- Aumentar propuesta de solución en el portafolio de proyectos de la empresa
- Disminuir tiempos de desarrollo, soporte y mantenimiento
- Revisión gerencial de avances y nuevos desarrollos.

En la siguiente Figura se detalla la interacción entre todos los requerimientos de alto nivel, en relación con:

- Tecnología
- Negocio y personas
- Procesos

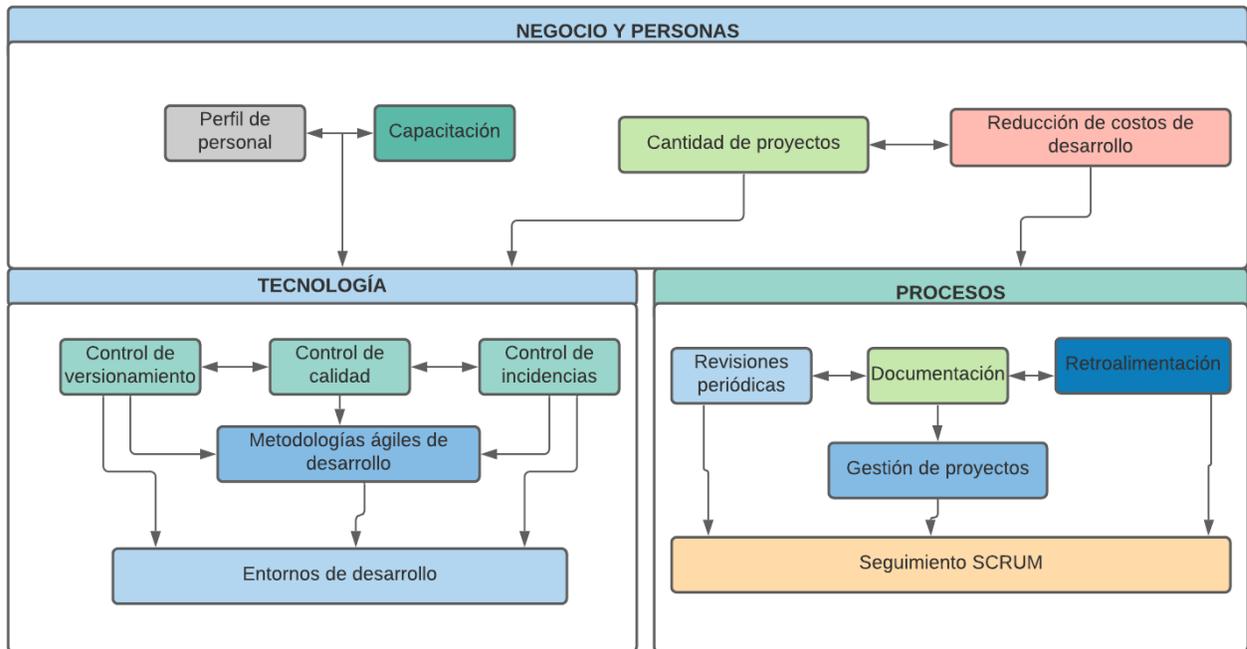


Figura 5 Requerimientos de alto nivel

3.2. Visionamiento y escenarios de la solución

Una vez analizados los diferentes marcos de referencia se optará por la implementación de metodologías ágiles como Scrum, debido a:

- La productividad fue duplicada en grandes proyectos
- Los procesos son optimizados cuando se aplica Scrum
- La cantidad del reproceso se reduce en un 40%.

3.2.1. Casos de referencia

Se tomará como referencia al caso de éxito de Amazon, por el uso de Scrum como metodología ágil de desarrollo.

Una de las principales Prácticas que le permitió a Amazon adoptar Scrum, y que fue clave para el éxito fue la Autonomía en los equipos de trabajo. (Captación, s.f.)

- **Arquitectura orientada a servicios:** Permite lograr equipos y desarrollos modulares.
- **Equipos de dos pizzas:** Los equipos de trabajo son pequeños y están organizados en función de los Servicios de Aplicación. Estos equipos, llamados “Equipos de dos Pizzas” porque es la cantidad de personas que pueden alimentarse con dos pizzas, es lo más parecido a un *Scrum Team* (3 – 8 integrantes), que es ampliamente aceptado como el tamaño óptimo de un equipo de trabajo.
- **Autonomía y *Empowerment*:** Dentro de Scrum es una de las claves para la agilidad. Un equipo autónomo, que pueda tomar sus propias decisiones basadas en su experiencia, sus capacidades, la situación actual, en fin, que pueda ser pragmático, es esencial para lograr Motivación, Compromiso, Colaboración y sobre todo... Agilidad.
- **Enfoque *Pull* por encima de *Push*:** El enfoque *Pull* se refiere a involucrar a las personas, convencerlas y escucharlas.

Amazon en la actualidad, se ha convertido en una empresa exitosa gracias a la aplicación de la metodología AGILES, afirma que entre los elementos claves a utilizar que refuerza dicha metodología se tiene:

- Otorgar a los equipos de trabajo, de una auténtica autonomía
- Desarrollar equipos de trabajo reducido, aproximadamente entre 4 a 6 personas
- Planificar tiempos de desarrollo para cada equipo y su colaboración entre sí
- Capacitaciones constantes
- Atender sugerencias de los equipos, en la aplicación de la metodología ágil

3.2.2. Fábrica de software triple A

A continuación, se muestra un modelo de fábrica de software triple A.

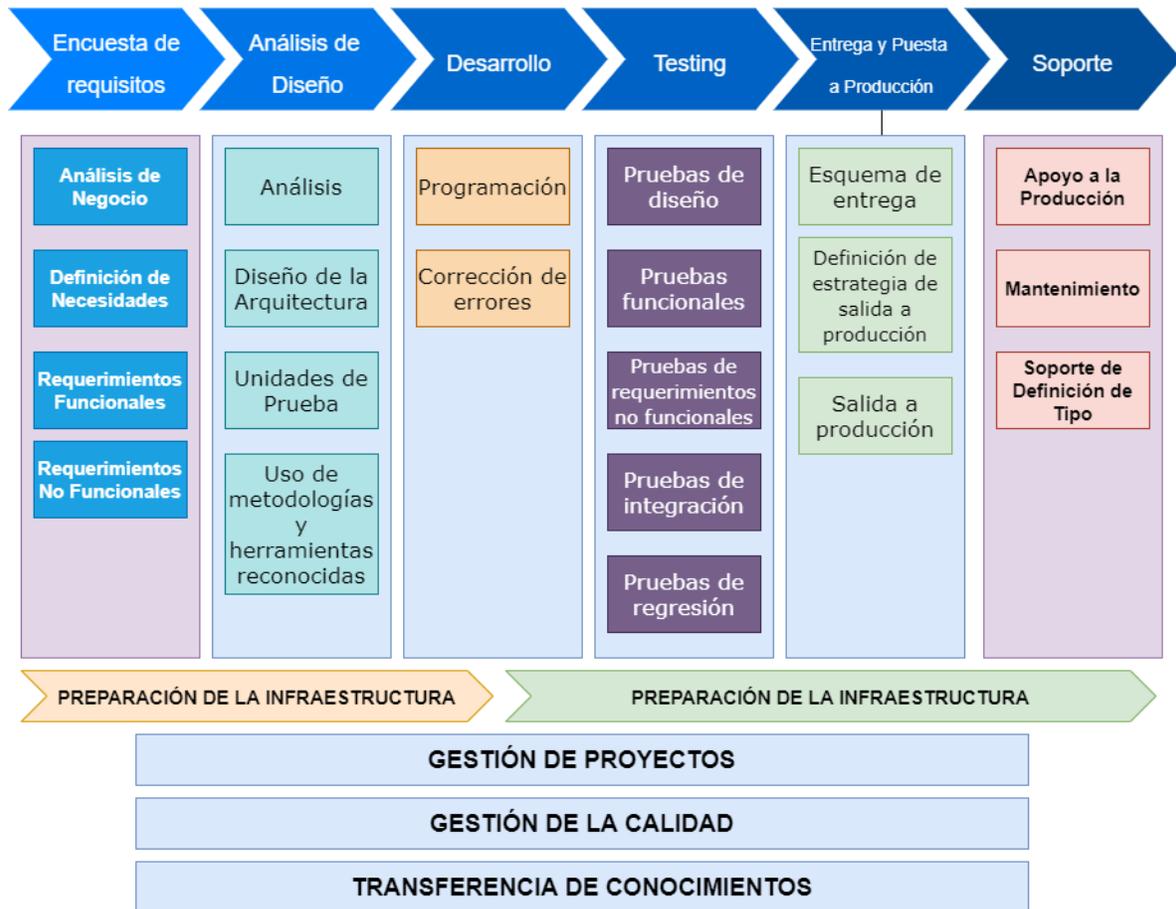


Figura 6 Fábrica 3A

3.2.3. Conclusión de visionamiento

Una vez analizado uno de los casos de éxito más emblemáticos, en el uso de metodologías ágiles es necesario definir el alcance del fortalecimiento de la fábrica software.

Debido a que la empresa es considerada en el entorno empresarial como una Pyme, es suficiente con la implementación de una fábrica de software doble AA.

Misma que deberá contar con la implementación de metodologías ágiles de desarrollo como SCRUM.

3.3. Análisis de brechas

Las áreas de los cuales se generará un esquema de brechas son los siguientes:

- Procesos
- Personas
- Tecnología

Área	Ámbito	Prácticas	Descripción de brechas	Estado actual	Objetivo	Referente
Procesos	Proyecto	Reunión diaria.	Promover reuniones periódicas	2	4	5
		Comunicación entre el equipo	Promover comunicación entre pares	2	5	5
		Revisión continua del avance por estimación del esfuerzo pendiente.	Seguimiento y análisis de resultados	1	4	5
		Desarrollo incremental.	Gestión de Desarrollo de Software	2	5	5
		Entregas en reunión con el cliente.	Gestión de la calidad	2	4	5
		Documentación de desarrollos	Gestión de Ingeniería de Software	4	5	5
		Retroalimentación	SGC - Historias de usuario	3	4	5
		Control del trabajo en proceso	Gestión de Proyectos	3	5	5
Personas	Organización	Equipos autoorganizados.	Metodología de Desarrollo - SCRUM	0	4	5
		Reuniones retrospectivas.	Reuniones gerenciales	2	4	5
		Espacio de trabajo común para el equipo.	Oficina virtual (Teletrabajo)	3	4	5
		Ritmo sostenido (< 40 horas semana).	Sprint - <i>backlogs</i> de Scrum	0	5	5
		Perfil académico del personal acorde a las necesidades de la empresa	SGC -Perfiles de cargo / Plan de capacitación	2	4	5
		Flexibilidad funcional	Promover mejora continua y profesionalismo	3	4	5
Tecnología	Desarrollo de software	Comprobación continua.	SGC - Proceso de DI	2	4	5
		Integración continua.	Uso de tecnología de punta / herramientas	3	4	5
		Programación en parejas.	Scrum - Equipos de programación	0	4	5
		Diseño simple e incremental.	Metodología de Desarrollo - SCRUM	0	4	5
		Estándares de codificación.	MVC - Modelo vista controlador	2	4	5
		Manejo de bases de datos estructurados y no estructurados	Control adecuado de datos	3	4	5
		Control de versionamiento	Uso de Gitlab	0	4	5
		Control de calidad	Promover uso de herramientas de SQA	1	4	5
		Ambientes de desarrollo definidos	Promover uso de contenedores	0	4	5
		Control y trazabilidad de incidencias	ERP - Módulo de Proyectos	1	4	5
		Uso de metodologías ágiles	Scrum	0	4	5

Calificación de
0 a 5

Tabla 11 Brechas de visionamiento



Figura 7 Análisis de brechas

Las brechas actuales, tanto en procesos, personas y tecnología, son sumamente grandes, ya que actualmente no se tiene, ningún tipo de control de versionamiento y metodologías ágiles de desarrollo.

Brechas, que se tiene como objetivo cerrar, con el uso de Scrum o la implementación de software que permita el control de las actividades y planificación del desarrollo de software.

3.4. Arquitectura objetivo

3.4.1. Arquitectura de negocio

Constituye en el diseño operación del negocio, normalmente descrita en términos de capacidades empresariales y capacidades de tecnología de soporte.

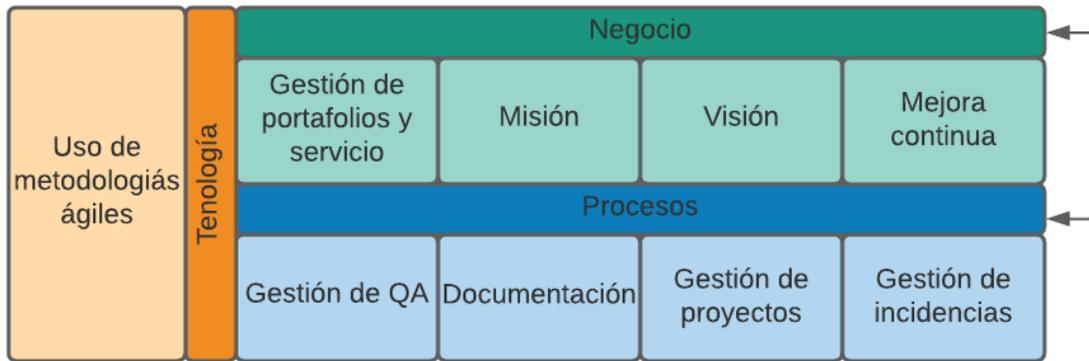


Figura 8 Target de arquitectura de negocio

3.4.2. Arquitectura de datos

Arquitectura de datos son los modelos, políticas, reglas y estándares que nos indican de qué manera tenemos que tratar los datos.

Se pretende dar un mejor tratamiento a datos, estableciendo repositorios de datos estructurados y no estructurados.

Referencia de la imagen: url, etc: <https://blog.gft.com/es/2017/12/05/que-es-una-arquitectura-de-referencia-de-datos-y-que-utilidad-tiene/>



Figura 9 Arquitectura de datos

3.4.3. Arquitectura de aplicaciones

Describe las herramientas que se utilizan para manejar, diseñar y desarrollar software.

La siguiente Figura, representa el estado actual de las aplicaciones y de acuerdo con una codificación de colores, identifica componentes nuevos a implementar, para fortalecer la fábrica de software de ProtelSoftware S.A.

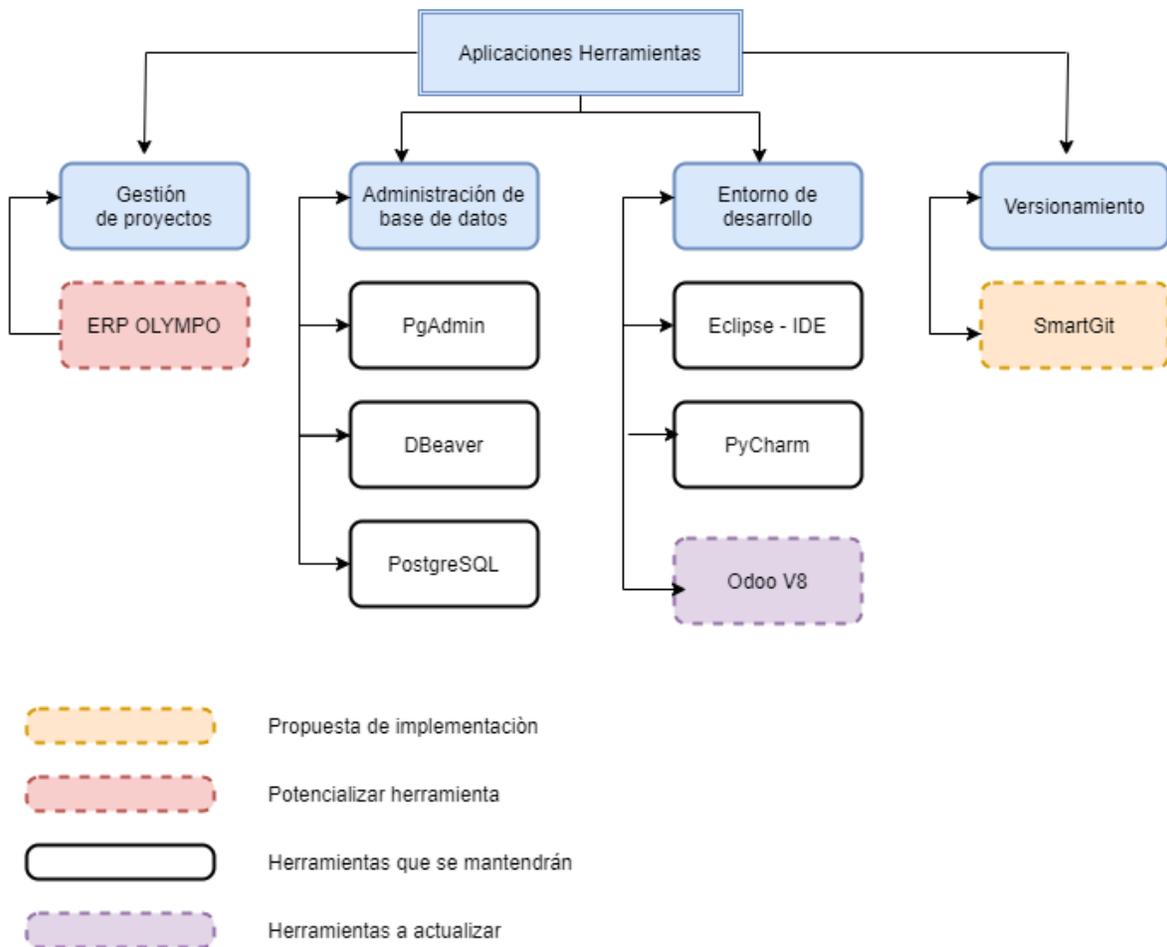


Figura 10 Arquitectura de aplicaciones y herramientas

3.4.4. Arquitectura de infraestructura base

La arquitectura de TI es el diseño de la infraestructura del servicio al usuario. Se pretende manejar la misma infraestructura, potencializando y dando un mejor uso a los equipos existentes.

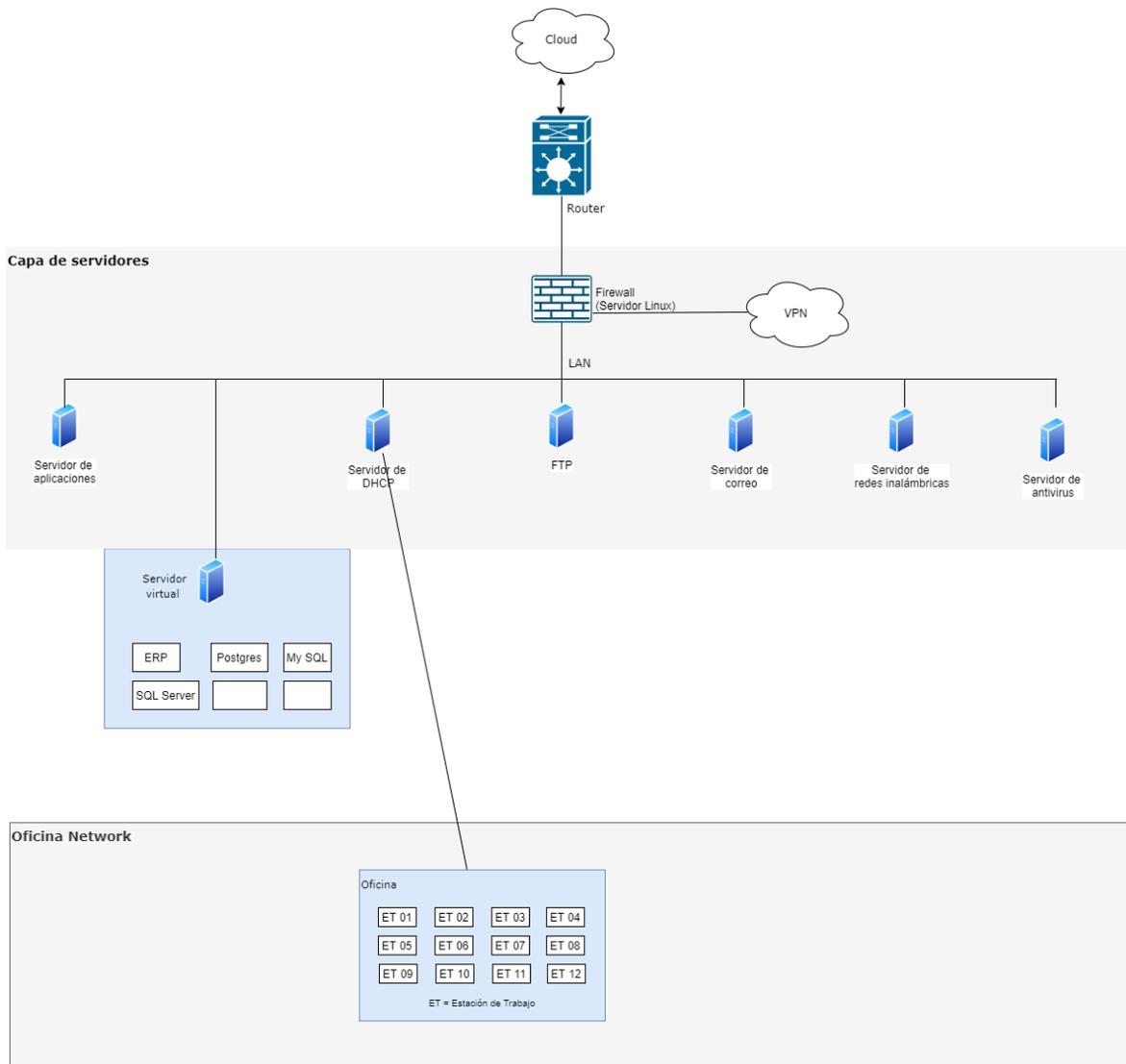


Figura 11 Arquitectura de infraestructura base

4. ARQUITECTURA DE NEGOCIO

4.1. Arquitectura de negocio actual

Actualmente ProtelSoftware S.A maneja una metodología de desarrollo de software en cascada, donde cada desarrollador es responsable de un determinado módulo de ERP.

La metodología de desarrollo actual presenta limitaciones, que constituyen problemas, conforme se describen a continuación:

- Poca flexibilidad al separar el proceso de desarrollo en etapas totalmente distintas.
(SEVILLA)
- Las diferentes etapas, no tienen fronteras definidas, lo que hace que, en ocasiones, se mezclen y compartan información.
- Dificultad para realizar prototipos.
- No permite reutilizar códigos
- Se realizan pruebas sin disponer de una implementación
- Las pruebas son realizadas por el programador
- Se debe tener todos los requisitos al principio. Lo normal es que el cliente no tenga perfectamente definidas las especificaciones del sistema, o puede ser que surjan necesidades imprevistas.
- Si existen errores, es difícil corregirlos
- No se tiene el producto hasta el final, esto quiere decir que:
 - Si se comete un error en la fase de análisis no lo descubrimos hasta la entrega, con el consiguiente gasto inútil de recursos.
 - El cliente no verá resultados hasta el final, con lo que podría inquietarse.
 - No se tienen indicadores fiables del progreso del trabajo.

- Es comparativamente más lento que otros modelos de desarrollo e implica un mayor costo.

4.1.1. Metodología actual de desarrollo de software

El desarrollo en cascada (en inglés, *waterfall model*) es un procedimiento lineal que se caracteriza por dividir los procesos de desarrollo en sucesivas fases de proyecto. Al contrario que en los modelos iterativos, cada una de estas fases se ejecuta tan solo una vez. Los resultados de cada una de las fases sirven como hipótesis de partida para la siguiente. (IONOS, 2019)

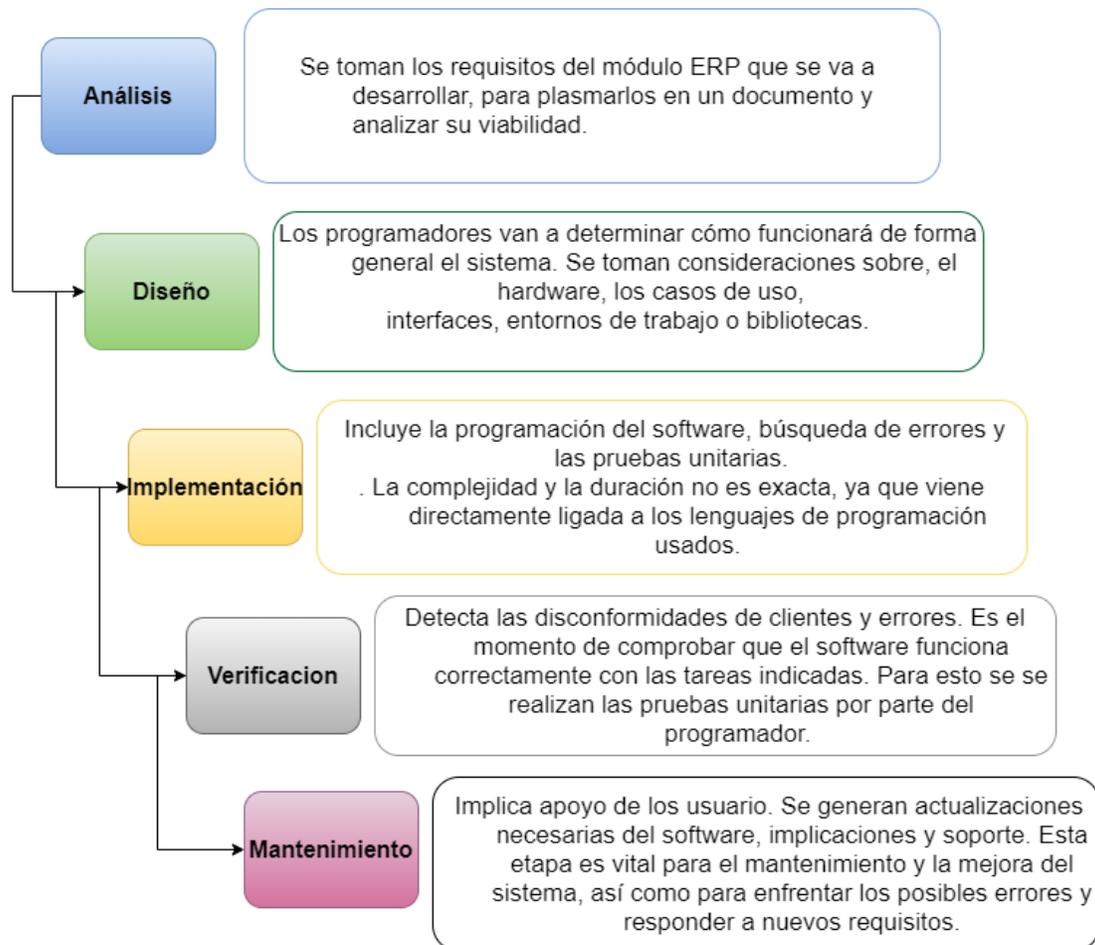


Figura 12 Modelo de desarrollo

4.1.2. Metodología propuesta para desarrollo de software

Se propone la implementación de metodologías ágiles de desarrollo de software, las cuales buscan proporcionar en poco tiempo piezas pequeñas de sistemas de software en funcionamiento para mejorar la satisfacción del cliente. Estas metodologías utilizan enfoques flexibles y el trabajo en equipo para ofrecer mejoras constantes. (Hat, s.f.).

Para fortalecer la fábrica y lograr el propósito de las metodologías ágiles se utilizará Scrum como marco ágil de desarrollo de software.

Este marco de gestión se diseñó para equipos pequeños entre 4 y 9 personas, los cuales dividen su trabajo en acciones que se puedan completar en un período de tiempo uniforme (sprint). Los equipos de Scrum están formados por los miembros, un facilitador y el dueño de un producto. Por lo general, este marco de trabajo se implementa cuando un proyecto grande se puede dividir en *sprints* de dos a cuatro semanas. Además, se centra en ciclos de retroalimentación mediante reuniones donde se realiza una "retrospectiva" del sprint. Su lema no oficial bien podría ser "analice todo y adáptese". (Hat, s.f.)

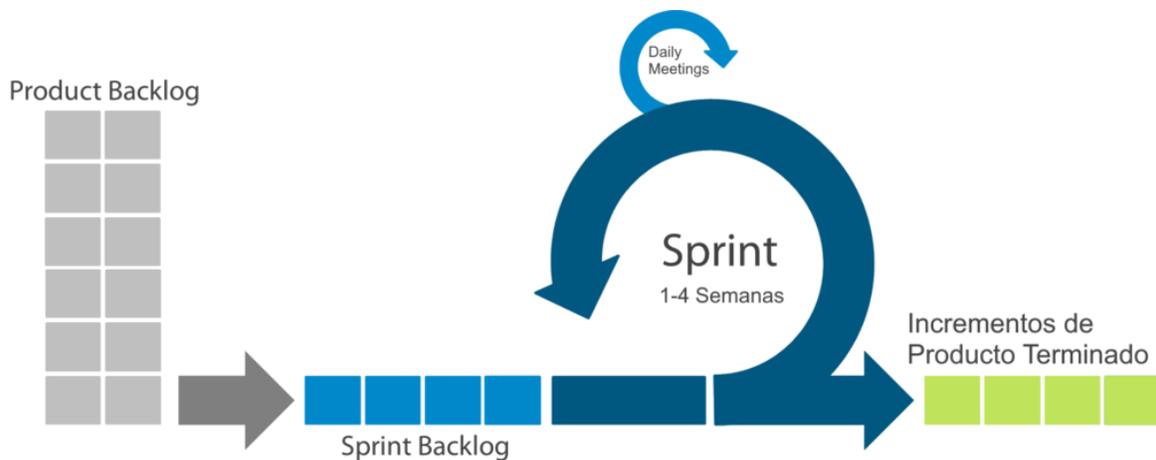


Figura 13 Scrum

4.1.3. Análisis de brechas

En la siguiente tabla se describe una valoración de la situación actual de la fábrica de software, respecto a una visión de agilismo de nuevos procesos basados en Scrum.

Las áreas de los cuales se generará un esquema de brechas son los siguientes:

- Procesos
- Personas
- Tecnología

Para el desglose de las brechas de la arquitectura de negocio ver ANEXO1



Figura 14 Análisis de brechas de arquitectura de negocio

Actualmente la fábrica de software carece de agilidad, lo que repercute en la calidad de: los desarrollos, su implementación, soporte y mantenimiento.

La implementación de metodologías ágiles de desarrollo implica un cambio en la cultura de las personas involucradas en la fábrica de software, en donde el valor proviene del software funcional.

Existe una carencia parcial en el control de todos los procesos actuales y un uso incorrecto de las herramientas implementadas, para lo cual se potencializará el uso del sistema ERP Olympo.

4.2. Arquitectura de negocio objetivo

4.2.1. Procesos

La implementación de Scrum comprende de 19 procesos en 5 fases: (Salazar, 2016)



FASE: INICIACIÓN

Procesos:

1. Crear la visión del proyecto.
2. Identificar al Scrum Master / Stakeholders.
3. Formación del equipo Scrum.
4. Desarrollo de épica (s).
5. Creación de la lista priorizada de pendientes del producto.
6. Realizar el plan de lanzamiento.

FASE: PLANIFICACIÓN Y ESTIMACIÓN

Procesos:

7. Elaborar historias de usuarios.
8. Elaborar, estimar y asignar historias de usuarios.
9. Elaboración de tareas.
10. Estimar tareas.
11. Elaboración de la lista de pendientes del Sprint.

FASE: IMPLEMENTACIÓN

Procesos:

12. Crear entregables.
13. Llevar a cabo el standup diario.
14. Mantenimiento de la lista priorizada de pendientes del producto.

FASE: REVISIÓN Y RETROSPECTIVA

Procesos:

15. Convocar Scrum de Scrums.
16. Demostración y validación del Sprint.
17. Retrospectiva de Sprint.

FASE: LANZAMIENTO

Procesos:

18. Envío de entregables.
19. Retrospectiva del proyecto.

Figura 15 Fases Scrum

i. Iniciación (6 procesos)

En esta fase se crea la visión del Proyecto que sirve de enfoque y dirección de este. Se crean e identifican roles claves del proyecto como el Scrum Master, *Product Owner*, interesados, equipo del proyecto. Así mismo, se define la lista de prioridades o el Product Backlog la cual sirve de base para la elaboración del plan de lanzamiento y tamaño de cada Sprint. (Salazar, 2016)

Procesos

- Crear la visión del proyecto (*Create Project Vision*)
- Identificar al Scrum Máster y a los interesados o socios del proyecto (*Identify Scrum Master and Stakeholder(s)*)
- Formación del equipo Scrum (*Form Equipo Scrum*)
- Desarrollo de épica(s) (*Develop Epic(s)*)
- Creación de la lista priorizada de pendientes del producto (*Create Prioritized Product Backlog*)
- Realizar el plan de lanzamiento (*Conduct Release Planning*)

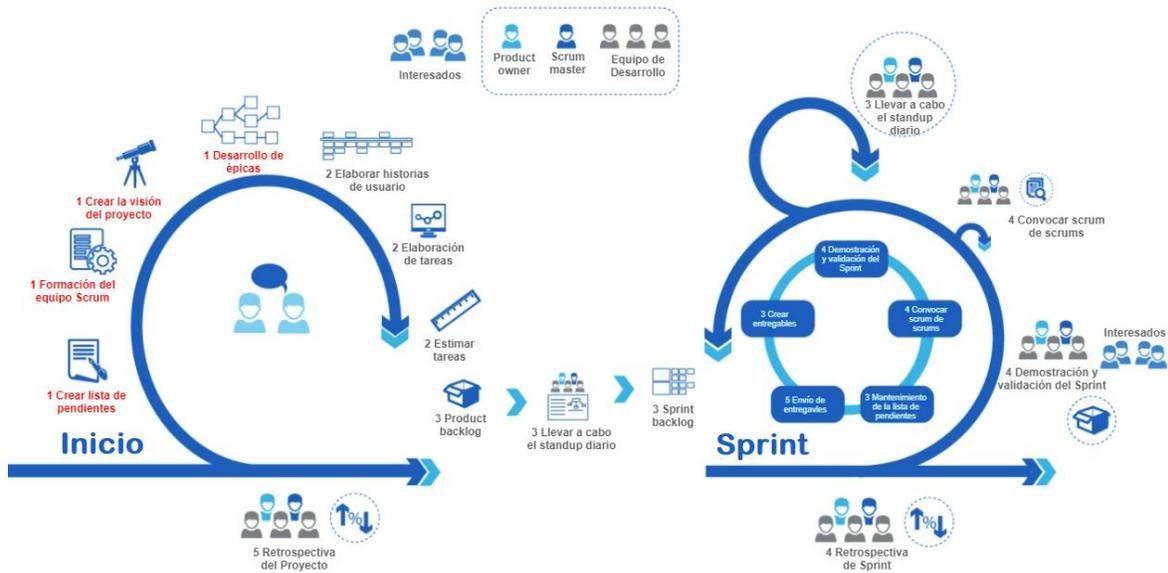


Figura 16 Fase de Iniciación SRUM

ii. Planificación y Estimación (5 procesos):

Aquí se definen los Sprints y las historias de usuarios, se alinean a todo lo que genera valor a la organización y se hacen las estimaciones de tiempo y esfuerzo para cumplirlas, los cuales se traducen en listas de tareas cuyos tiempos de desarrollo se definen en reuniones de equipo correspondientes, así como el proceso de definición del *Sprint Backlog* que contiene todas las tareas que deben completarse en el Sprint. (Salazar, 2016)

Procesos

- Elaborar historias de usuario (*Create User Stories*)
- Aprobar, estimar y asignar historias de usuarios (*Approve, Estimate, and Commit User Stories*).
- Elaboración de tareas (*Create Tasks*)
- Estimar tareas (*Estimate Tasks*)
- Elaboración de la lista de pendientes del Sprint (*Create Sprint Backlog*)

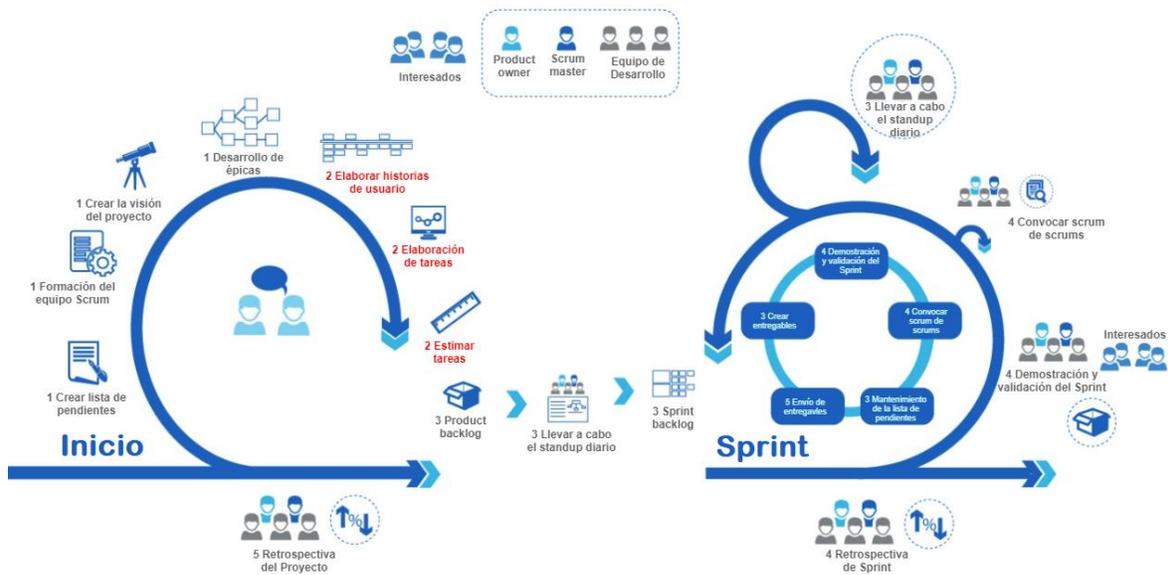


Figura 17 Fase de Planificación y Estimación SCRUM

iii. Implementación (3 procesos):

En esta fase se trabaja en las tareas del Sprint Backlog para crear Sprint *Deliverables*, para ello se utiliza a menudo un *Scrumboard* para realizar el seguimiento del trabajo y de actividades que se llevan a cabo. También, los inconvenientes o problemas que enfrenta el Equipo Scrum se actualizan en un *Impediment Log*. Durante esta fase se realizan las llamadas *Daily Standup Meeting* que son reuniones cortas y eficientes en tiempo donde el equipo da el status de sus actividades diarias y manifiesta cualquier inconveniente que pueda tener. Igualmente se actualiza o revisa la lista de prioridades de pendientes del producto. (Salazar, 2016)

Procesos

- Crear entregables (*Create Deliverables*),
- Llevar a cabo el *standup diario* (*Conduct Daily Standup*)

- Mantenimiento de la lista priorizada de pendientes del producto (*Groom Prioritized Product Backlog*)

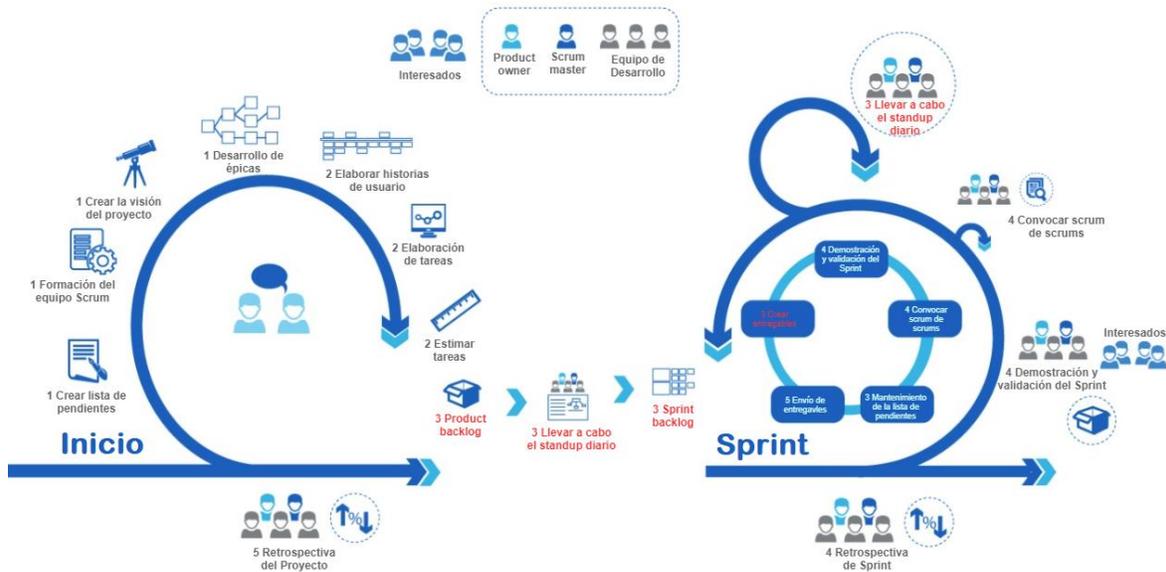


Figura 18 Fase de Implementación SRUM

iv. Revisión y Retrospectiva (3 procesos):

Para proyectos grandes que involucran varios equipos Scrum, se realiza en esta etapa, reuniones que permitan juntar a estos equipos, discutir y revisar avances, dependencias e impedimentos en el desarrollo del proyecto. También en esta etapa se lleva a cabo el proceso donde el Equipo Scrum le demuestra el Sprint *Deliverable* al Propietario del producto y a los Socios relevantes en un *Sprint Review Meeting*. Igualmente, el Scrum Master y el Equipo Scrum se reúnen para discutir las lecciones aprendidas a lo largo del Sprint, información que se documenta como las lecciones aprendidas que pueden aplicarse a los futuros *Sprints*. (Salazar, 2016)

Procesos

- Convocar Scrum de Scrums (*Convene Scrum of Scrums*)
- Demostración y validación del Sprint (*Demonstrate and Validate Sprint*)
- Retrospectiva de Sprint (*Retrospect Sprint*)

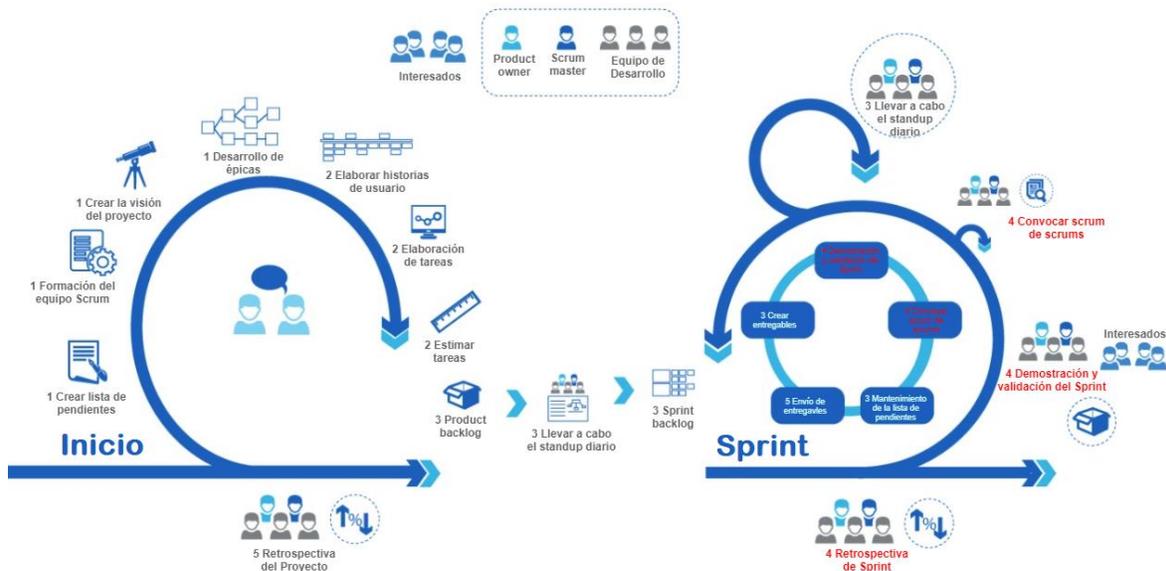


Figura 19 Fase de revisión y retrospectiva SCRUM

v. Lanzamiento (2 procesos):

Finalmente, esta es la fase más esperada por los interesados o socios del proyecto así como del Scrum Master y Equipo Scrum. En esta fase se desarrolla el proceso donde los *Accepted Deliverables* se les entregan a los Socios relevantes. Un acuerdo formal llamado *Working Deliverables Agreement* documenta la finalización con éxito del Sprint. Del mismo modo, se realizan actividades de retrospectiva que permite identificar mejoras y lecciones aprendidas del proyecto.

- Envío de entregables (*Ship Deliverables*)

- Retrospectiva del proyecto (*Retrospect Project*)

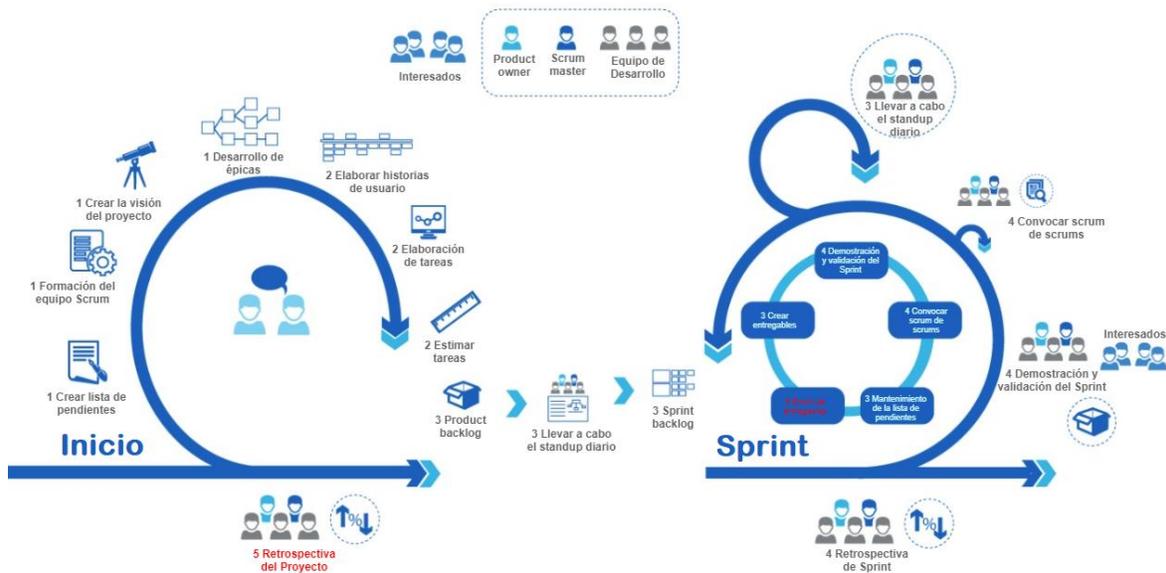


Figura 20 Fase de lanzamiento SRUM

4.2.1.1. Ventajas de la metodología Scrum

La metodología de Scrum tiene un sin número de ventajas, a continuación, se detallan algunas de ellas:

- Scrum es muy fácil de aprender, los roles, hitos y herramientas son claros y tienen un objetivo por lo que es un método muy relacionado con nuestra manera diaria de trabajar.
- El cliente puede comenzar a usar el producto rápidamente.
- Se agiliza el proceso, ya que la entrega de valor es muy frecuente.
- Menor probabilidad de sorpresas o imprevistos, porque el cliente está viendo frecuentemente el proyecto. (Abellán, s.f.)

4.2.1.2. Desventajas de la metodología Scrum

A continuación, se describen limitantes que actualmente tenemos para la implementación de Scrum:

- Aunque Scrum sea fácil de aprender, es muy difícil implementarlo. Esto supone una predisposición y un cambio de cultura de la organización, que debe ir desde los altos mandos hasta los clientes.
- La necesidad de tener equipos multidisciplinares, puede ser un problema, ya que es difícil encontrar personas que sean capaces de hacer todo el trabajo de un equipo.
- El equipo puede tender a realizar el camino más corto para conseguir el objetivo de un sprint, el cual no siempre ofrece resultados de calidad. (Abellán, s.f.)

4.2.2. Organización

Con la implementación de metodologías ágiles de desarrollo de software la organización obtendrá los siguientes resultados:

- **Gestión regular de las expectativas del cliente:** El cliente establece sus expectativas, indicando el valor que le aporta cada requisito del proyecto y cuando espera que esté terminado.

El cliente comprueba de manera regular si se van cumpliendo sus expectativas, da feedback, ya desde el inicio, puede tomar decisiones informadas a partir de resultados objetivos y dirige estos resultados del proyecto, iteración a iteración, hacia su meta. Se ahorra esfuerzo y tiempo al evitar hipótesis. (org, s.f.)

- **Resultados anticipados (“time to market”):** El cliente puede empezar a utilizar los resultados más importantes del proyecto antes de que esté finalizado por completo. Siguiendo la ley de Pareto (el 20% del esfuerzo proporciona el 80% del valor), el cliente puede empezar antes a recuperar su inversión (y/o autofinanciarse) comenzando a utilizar un producto al que sólo le faltan características poco relevantes, puede sacar al mercado un producto antes que su competidor, puede hacer frente a urgencias o nuevas peticiones de clientes, etc.
- **Flexibilidad y adaptación:** De manera regular el cliente redirige el proyecto en función de sus nuevas prioridades, de los cambios en el mercado, de los requisitos completados que le permiten entender mejor el producto, de la velocidad real de desarrollo, etc.

Al final de cada iteración el cliente puede aprovechar la parte de producto completada hasta ese momento para hacer pruebas de concepto con usuarios o consumidores y tomar decisiones en función del resultado obtenido.
- **Retorno de inversión (ROI):** De manera regular, el cliente maximiza el ROI del proyecto. Cuando el beneficio pendiente de obtener es menor que el coste de desarrollo, el cliente puede finalizar el proyecto.
- **Mitigación de riesgos:** Desde la primera iteración el equipo tiene que gestionar los problemas que pueden aparecer en una entrega del proyecto. Al hacer patentes estos riesgos, es posible iniciar su mitigación de manera anticipada. "Si hay que equivocarse o fallar, mejor hacerlo lo antes posible". El feedback temprano permite ahorrar esfuerzo y tiempo en errores técnicos.

La cantidad de riesgo a que se enfrenta el equipo está limitada a los requisitos que se puede desarrollar en una iteración. La complejidad y riesgos del proyecto se dividen de manera natural en iteraciones.

- **Alineamiento entre cliente y equipo:** Los resultados y esfuerzos del proyecto se miden en forma de objetivos y requisitos entregados al negocio. Todos los participantes en el proyecto conocen cuál es el objetivo para. El producto se enriquece con las aportaciones de todos

4.2.3. Personal

4.2.3.1. Personal actual

A continuación, se describen las actividades realizadas por el personal actual en la fábrica de software

- **Gerente de desarrollo de sistemas:** Desarrollar y proponer nuevos productos, mejoras y nuevas tecnologías. Manejar una mejora continua de calidad de los actuales productos, definir y dirigir los recursos humanos y tecnológicos del área, definir los objetivos del área y gestionar proveedores
- **Coordinador desarrollo informático:** Coordinar y distribuir el trabajo del personal, con el propósito de lograr el cumplimiento de los objetivos del área.
Revisar, validar, observar y aceptar los nuevos sistemas desarrollados por los analistas programadores, con el fin de que estos sean implementados

- **Analista programador:** Desarrollar y dar mantenimiento a los sistemas desarrollados por la empresa, de acuerdo con el diseño y análisis realizado, tomando en cuenta la normativa ecuatoriana vigente y a los requerimientos del sistema.
- **Técnico Informático:** Satisfacer las necesidades de actualización, optimización y mantenimiento de todo el sistema informático sobre el que se sustenta la organización, así como de los equipos electrónicos, a fin de garantizar su correcto funcionamiento.
- **Técnico de Documentación:** Elaborar las guías de ayuda administrativa y operativa para los sistemas informáticos de la organización, de cada uno de los módulos que incluyan los mismos.

Cabe recalcar, que actualmente el gerente y coordinador del departamento cuentan con conocimiento y certificación *Scrum*. En relación con los analistas, estos tienen conocimientos básicos de la metodología

4.2.3.2. Personal objetivo

Se mantendrá el personal actual, sin embargo, se capacitará a los analistas programadores y mandos medios del departamento de desarrollo informático en metodología ágiles de desarrollo (*Scrum*).

A continuación, se muestra los roles actuales y los objetivos del personal de sistemas.

Rol actual	Rol objetivo
Gerente de desarrollo informático	Producto Owner
Coordinador desarrollo informático	SCRUM Master
Analista programador	Equipo de desarrollo
Técnico Informático	Técnico Informático
Técnico de Documentación	Técnico de Documentación

Tabla 12 Rol actual vs Rol objetivo

Las funciones que debe cumplir el personal, se describen a continuación:

- **Product Owner:** Se lo define “como responsable de maximizar el valor del producto y del trabajo del *Development Team*” (Herranz, 2016) una mala selección de este conducirá al fracaso del proyecto.

Este rol lo tomará el gerente del departamento de desarrollo informático
- **SCRUM Master:** “Se encarga de la correcta aplicación del método” (Brice, 2018) su principal objetivo es ayudar al equipo y corregir las dificultades de la organización. Es una persona a tiempo completo o parcial que facilite y sea coach de Scrum.

Este rol será ejercido por el coordinador del departamento
- **Equipo de desarrollo:** “Son expertos en el ámbito de la aplicación, responsables de la entrega de elementos *backlog*” (Nader, 2019)

Se conformará 2 equipos de cuatro analistas cada uno, de acuerdo con el conocimiento en relación con los módulos vigentes en el ERP.

- **Técnico Informático:** Dar soporte a los equipos Scrum
- **Técnico de Documentación:** Validar información generada en cada interacción y control de trazabilidad de desarrollo en módulo de Scrum en Olympo.

4.2.3.3. Características del equipo de desarrollo

Para ir alcanzando un estado de alto rendimiento, este equipo debe poseer el siguiente perfil: (Proyectosagiles.org, s.f.):

- **Autónomo y multidisciplinario:** Debe ser ágil y debe desarrollar un producto “tangible” de forma iterativa, para obtener *feedback* en intervalos cortos de tiempo sobre si el producto genera en el cliente el resultado esperado. Para ello, idealmente el equipo no debe depender de otros, pues generaría esperas y retrasos en la entrega de valor (necesidad de coordinación, integración e incluso falta de *ownership*), con lo cual, lo fundamental es contar con equipos multidisciplinarios y autónomos. (Proyectosagiles.org, s.f.)
- **Estable y dedicado:** Los equipos ágiles trabajan en iniciativas estratégicas (que se mantienen en el tiempo) o bien van repitiendo tipos de proyectos similares (si es necesario, van ajustando su misión como equipo). Un equipo de alto rendimiento cuesta tiempo conseguirlo (después hay que aprovechar esa performance), es un activo fundamental para la empresa y es su *building* block básico, lo cual deriva en el interés por mantenerlo por encima de desmantelarlo.

Dado que el equipo debe ser estable, sus miembros deben cambiar lo mínimo posible, para poder aprovechar el esfuerzo que les ha costado construir sus relaciones interpersonales, engranarse y establecer en plan de trabajo.

Los miembros del equipo deben estar dedicados a tiempo completo a esa iniciativa o proyecto, para evitar dañar su productividad los cambios o cumplimiento de otras tareas en otros proyectos no es adecuado. (Proyectosagiles.org, s.f.)

- **Se auto-organizan:** Tienen responsabilidad compartida y piensan juntos:

Para crear sinergias, que les permitirá conseguir el resultado requerido, en el menor tiempo posible.

Para mejorar su proceso de trabajo y el sistema que les rodea, reflexionando regularmente (por ejemplo, haciendo Retrospectivas) e introduciendo acciones de mejora en su *backlog* de tareas a realizar. (Proyectosagiles.org, s.f.)

La manera de conseguir esto se basa en compartir información y en la confianza entre los miembros asignados a este proyecto, para ello, deben realiza de manera conjunta las siguientes actividades:

- Seleccionar los requisitos que prevén completar en una iteración, de forma que estén preparados para ser entregados al cliente.
- Estimar la complejidad de cada requisito en la lista de requisitos priorizada del producto o proyecto.
- En la reunión de planificación de la iteración, decidir cómo va a realizar su trabajo:
- Seleccionar los requisitos que pueden completar en cada iteración, realizando al cliente las preguntas necesarias.

- Identificar todas las tareas necesarias para completar cada requisito.
 - Estimar el esfuerzo necesario para realizar cada tarea.
 - Cada miembro del equipo se autoasigna a las tareas (notar que en una tarea puede haber más de una persona para aumentar su calidad, para aprender entre pares, etc.).
 - Durante la iteración: trabajar de manera conjunta para conseguir los objetivos de la iteración. Cada especialista puede liderar el trabajo en su área y el resto colaboran si es necesario para poder completar un requisito (dado que se hace un esfuerzo en que sean *T-shaped*).
- **Al finalizar la iteración:**
 - Revisar con el cliente los resultados de la iteración y los requisitos completados.
 - Hacer una retrospectiva la final de cada iteración para mejorar de forma continua su manera de trabajar.
- **Están motivados:** Las personas motivadas hacen cosas increíbles y, por otro lado, un equipo no motivado difícilmente desarrollará un gran producto o estará comprometido con proporcionar un gran servicio (lo cual lleva a clientes / usuarios finales a los que se mejora su vida y están más contentos, que es lo que necesita la empresa).
 - **Están sentados juntos:** Todos los miembros del equipo trabajan en la misma localización física, para poder maximizar la comunicación entre ellos mediante conversaciones cara a cara, diagramas en pizarras blancas, etc.
- De esta manera se minimizan otros canales de comunicación menos eficientes, que hacen que las tareas se transformen en un “pasa pelota” o que hacen perder el tiempo

en el establecimiento de la comunicación (como cuando se llama repetidas veces por teléfono cuando la persona no está en su puesto). (Proyectosagiles.org, s.f.)

- En el caso de equipos remotos, es conveniente garantizar un tiempo de calidad cara a cara, como mínimo:
 - 1-2 semanas para el inception inicial del proyecto e integración “social” de los miembros del equipo.
 - 2 o más días por iteración, para actividades como *Refinements*, Retrospectivas o Planificaciones de iteración.

- **Interaccionan con *Stakeholders*:** Este equipo interacciona frecuentemente con los *Stakeholders* para compartir conocimiento de negocio, alinear prioridades y medir los resultados del valor entregado al cliente. (Proyectosagiles.org, s.f.)

5. ARQUITECTURA DE APLICACIONES / DATOS

5.1. Arquitectura de aplicaciones actual

La arquitectura actual de aplicaciones se encuentra conformada por programas englobados en grupos funcionales de aplicación, descritos a continuación:

- **Gestión de proyectos:** Son sistemas que permiten mejorar la relación entre productividad, costes, administración, y tiempo de producción. Para esto se utiliza ERP Olympo.

Sin embargo, en el departamento de desarrollo informáticos, la trazabilidad, documentación y control de nuevos desarrollos y su soporte, es llevado de manera manual. Muy poco se registra en el módulo de proyectos de la plataforma Olympo.

- **Administración de bases de datos:** Administra las tecnologías de la información y la comunicación. Para esto se utiliza:

- PgAdmin
- DBeaver
- PostgreSQL

- **Entornos de desarrollo de software:** Son todas las aplicaciones informáticas que proporcionan servicios integrales para facilitarle al programador el desarrollo de software. Para esto se utiliza:

- Eclipse
- PyCharm
- Odo V8

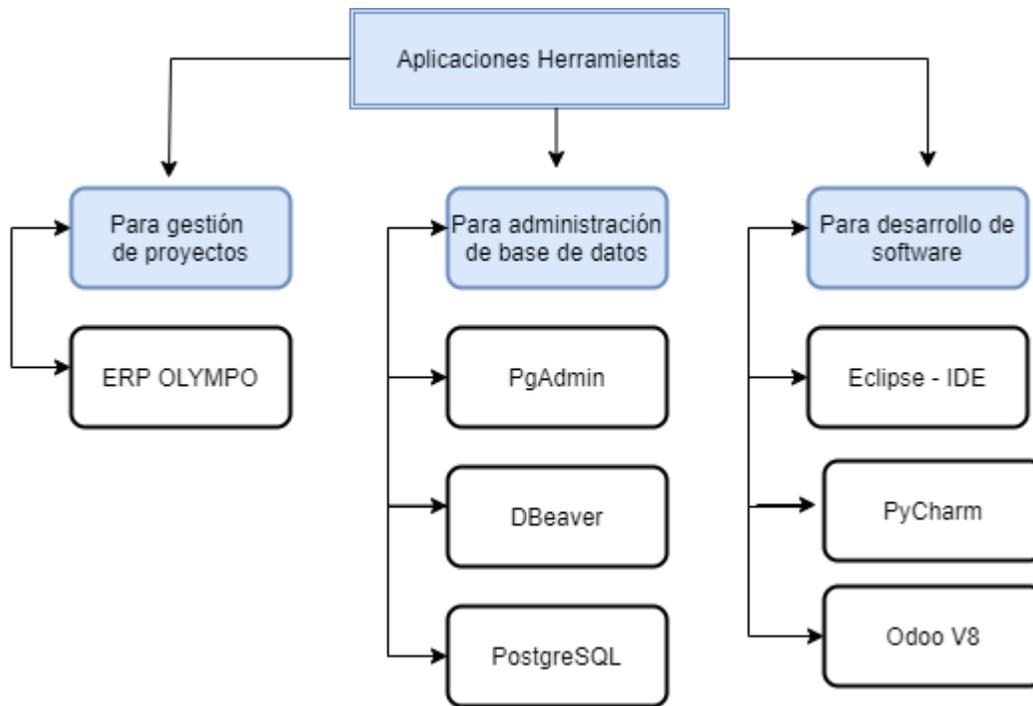


Figura 21 Aplicaciones actuales

5.1.1. Catálogo de aplicaciones

Ver ANEXO2, el cual se describe una breve descripción de todos los programas utilizados por la fábrica de software actualmente.

5.2. Arquitectura datos actual

Actualmente se maneja dos tipos de data:

5.2.1. Tipos/fuentes de datos

Actualmente en la empresa se maneja dos tipos de data:

- Datos estructurados
- Datos no estructurados



Figura 22 Tipos de datos

5.2.1.1. Datos estructurados

Es la información que se encuentra en las bases de datos relacionales.

Para gestionar esta información se utiliza Postgre SQL diseñado para administrar, recuperar y cargar información de los diferentes módulos del ERP Olympo (Administración Financiera, Presupuestos, Gestión de Inventario, Activos Fijos, -Garantías, Rol de Pagos, Proyectos, etc), desarrollado bajo Odoo.

La base de datos del sistema ERP almacena tablas, vistas de diferentes tablas y otros elementos como funciones o disparadores que realizan operaciones sobre los datos.

Se puede acceder a la base de datos de OpenERP directamente. Para ello solo es necesario conectar con ella a través de un cliente de base de datos pgAdmin.

Aproximadamente el tráfico que se tiene en la base de datos es de 300 usuarios simultáneos, dependiendo del módulo gestionado.

- **Estructura de la base de datos.**

Cada módulo crea una base de datos automáticamente. Según se instalen los diferentes módulos, el número de tablas se multiplicará por 3 o por 4.

Por defecto, cada una de las tablas está nombrada según al módulo al que pertenece.

5.2.1.1. Datos no estructurados

La característica principal de este tipo de datos, generalmente binarios, no poseen una estructura interna identificable. Se toma ficheros mediante el uso de *paths* e identificadores, organizados a través de una base de datos no relacional, sin límite de peso.

Los archivos que se almacenan son:

- Documentos PDF.
- Documentos Word.
- Correos electrónicos.
- Hojas de cálculo.
- Vídeos y audios.

Corresponde a toda la información de: informes, actas, manuales, videos, fotos y documentación generada por la fábrica de software.

5.3. Arquitectura de aplicaciones y datos objetivo

5.3.1. Arquitectura de aplicaciones objetivo

Debido a que se va a utilizar metodologías ágiles de desarrollo, es necesario actualizar Odoo, plataforma ERP donde corre actualmente los módulos de Olympo.

Actualmente se trabaja en la versión 8 la cual es muy limitada, y posee los siguientes problemas: (Cabrera, octopus, 2020)

- Al trabajar con un equipo Scrum no permitiría el desarrollo de un mismo módulo por parte de varios programadores simultáneamente.
- El *Front End* (Parte del software que interactúa con los usuarios), es poco amigable con el usuario, en muchos casos complicada.
- Las herramientas de inteligencia de negocios son limitadas para las expectativas de los clientes
- Poca flexibilidad de migración
- No cuenta con notificaciones inteligentes, etc

Mejoras de Odoo en su última versión posee:

- Mejoras en la App de Sitio Web
- CRM adaptativo
- Control de inventarios mejorado
- Control de gastos, ahora con OCR
- Inteligencia de negocios

Se utilizará el módulo de "Odoo Scrum" en Olympo en su módulo de Scrum con el cual se tendrá:

- Listas de requisitos priorizados (historias de usuario)

- Control de *Sprints*
- Seguimiento de tareas
- Pruebas de aceptación, etc.

De esta manera se dará uso al módulo de proyectos existente en Olympto, mejorando la gestión y trazabilidad de tareas resultantes de Scrum

Para el control de versionamiento se utilizará *SmartGit*, el cual permite comparar o fusionar archivos, editarlos uno al lado del otro, resolviendo conflictos de fusión utilizando el solucionador de conflictos.

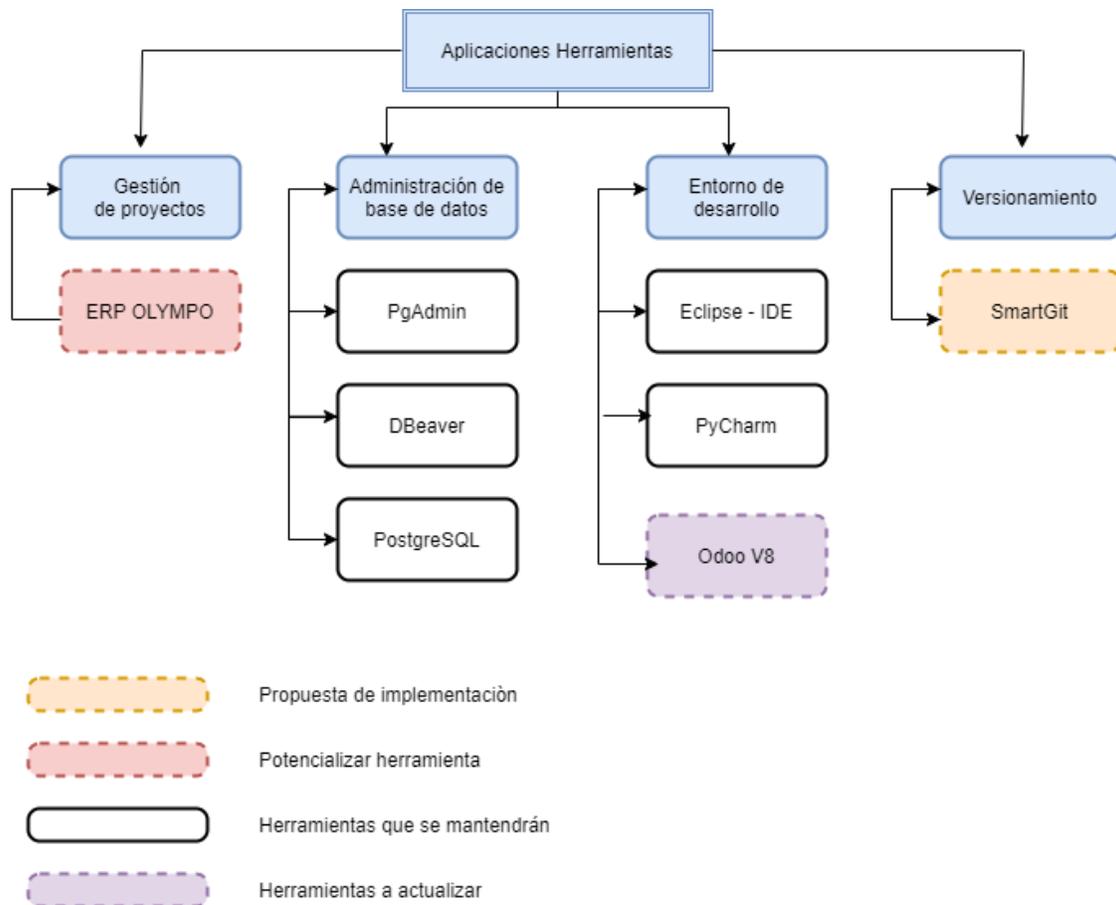


Figura 23 Objetivo de la Arquitectura de aplicaciones

En la Figura anterior podemos visualizar los programas que van a ser implementados en color naranja, los que se potencializaran de color rojo, los que se deben actualizar en color violeta y en blanco los que no se toparán.

5.3.2. Arquitectura de datos objetivo

Se mantendrá la estructura de datos actual, debido a la inversión (tiempo) entregada a la misma. Tomando en cuenta que la actualización de Odoo no genera implicaciones en las bases relacionales existentes.

5.3.3. Catálogo de aplicaciones objetivo

Procesos	Programas	Descripción	Características	Imagen
Versionamiento	SmartGit	<p>Potente cliente Git GUI tiene una interfaz inteligente que se ve y funciona igual en diferentes plataformas.</p> <p>Tiene una función de vista única donde puedes ver tu índice, árbol de trabajo y comandos, todo en la ventana de registro.</p>	<ul style="list-style-type: none"> -El cliente tiene la misma interfaz de usuario para Windows, macOS y Linux. -Oculta la complejidad de Git para el usuario. -Permite realizar acciones como <i>merge</i> y <i>commit</i> de forma gráfica. -Incluye cliente de línea de comandos para Git (Windows y macOS). -Git-Flow. -Cliente SSH. -Comparador de archivos. -Combinador de archivos (resuelve conflictos de sincronización). -La licencia cuesta US\$ 59 (la licencia más reducida) al año. 	
Desarrollo de software	Odoo V14	<p>Es un software de ERP integrado. Cuenta con una versión "comunitaria" de código abierto bajo licencia LGPLv3 y una versión empresarial bajo licencia comercial que complementa la edición comunitaria con características y servicios comerciales y desarrollada por la empresa belga Odoo S.A. El fabricante declara su producto como una alternativa de código abierto a SAP ERP y Microsoft Dynamics. La compañía tiene sucursales en varias partes del mundo.</p>	<p>Mejoras</p> <p>Moderno sistema de ERP: con experiencia de usuario mejorada</p> <p>CRM automatizado</p> <p>Complementos para Gmail y Outlook</p> <p>Lista de tareas pendientes que te ahorrará tiempo</p> <p>Creador de sitios web renovado</p> <p>Herramientas de inteligencia empresarial</p> <p>Vista de hoja de cálculo</p> <p>Ajustes en compras, inventario y manufactura</p>	

Tabla 13 Catálogo de aplicaciones objetivo

5.3.4. Análisis de brechas

Herramientas	Iniciativa de cierre	Estado actual	Objetivo	Referente
ERP Olympos	Uso de módulos nuevos de Odoov14 de gestión de metodologías ágiles	3	4	5
PgAdmin		5	5	5
DBeaver		5	5	5
PostgreSQL		5	5	5
Eclipse		5	5	5
PyCharm		5	5	5
Odoov8	Actualizar herramienta a su última versión	3	5	5

Tabla 14 Brechas arquitectura de aplicaciones- datos

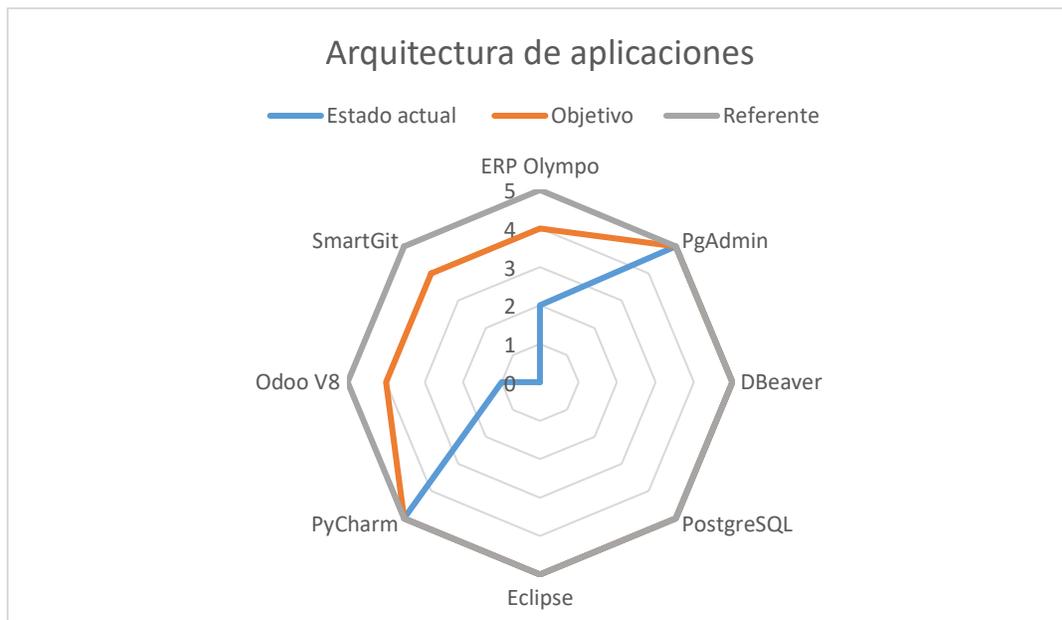


Figura 24 Análisis de la arquitectura de aplicaciones

Es necesario actualizar la versión de Odoo, ya que esta carece de funcionalidades, que actualmente los clientes necesitan.

Tomando en consideración, que Olympo corre bajo el entorno Odoo y debido a que se quiere reestructurar el departamento de sistemas, fortaleciendo la fábrica de software mediante Scrum, es imperativo la actualización y de esta manera mejorar ERP Olympo.

Uno de los problemas más grandes de la empresa, radica en el versionamiento,

Por lo tanto con la implementación de SmartGit se van a eliminar estos errores, tomando el control de proyecto, disminuyendo los encontrados en producción, generados por la implementación de versiones antiguas.

6. ARQUITECTURA DE INFRAESTRUCTURA BASE

Actualmente ProtelSoftware mantiene la siguiente infraestructura, misma que irá potencializándose conforme crezca el requerimiento del negocio y las necesidades de la fábrica de software

6.1. Infraestructura base

A continuación, se detallan las características de servidores, computadores y equipos pertenecientes a la fábrica de software.

Equipo	Tipo	Estado	SO	Procesador	Memoria RAM	Disco Duro	Detalle de uso
Firewall	Clon	Físico	Linux	Intel CORE 2 Quad 2.66 ghZ	4 GB	500 GB SSD	Está destinado como: VPN, DHCP Server, Firewall y Servidor de Intrusión
Servidor de Aplicaciones	HP	Físico	Linux	Intel XEON 2.40 GHz	32 GB	2 TB	Destinado para el manejo de aplicaciones web
Servidor FTP	HP	Físico	Linux	Intel XEON 2.40 GHz	32 GB	2 TB	Destinado para Gestión Documental
Servidor de BDDs	HP	Físico	Windows	Intel XEON 2.40 GHz	32 GB	2 TB	Destinado como gestor de bases de datos
Servidor de Pruebas	HP	Físico	Windows	Intel XEON 2.40 GHz	32 GB	2 TB	Destinado como gestor de bases de datos para pruebas locales
Access Point 1 y 2	Aruba	Físico		100 users	2.4 a 5 GHz		Destinado para conexiones inalámbricas de la red interna
Desarrollador	Dell / HP	Físico	Linux	Core i7	16 GB	500 GB SSD	9na y 10ma Generación para desarrollo de software
Implementador	Asus / Dell / Toshiba / HP	Físico	Windows	Core i7	8 GB	1 TB	9na y 10ma Generación para la implementación de las herramientas del ERP
Otros	Asus / Dell / Toshiba / HP	Físico	Windows	Core i5 / Core i7	8 GB	1 TB	9na y 10ma Generación para operaciones administrativas de las herramientas del ERP dentro de la empresa.

Tabla 15 Características de infraestructura base

En la siguiente Figura, podemos ver el despliegue de los equipos antes descritos

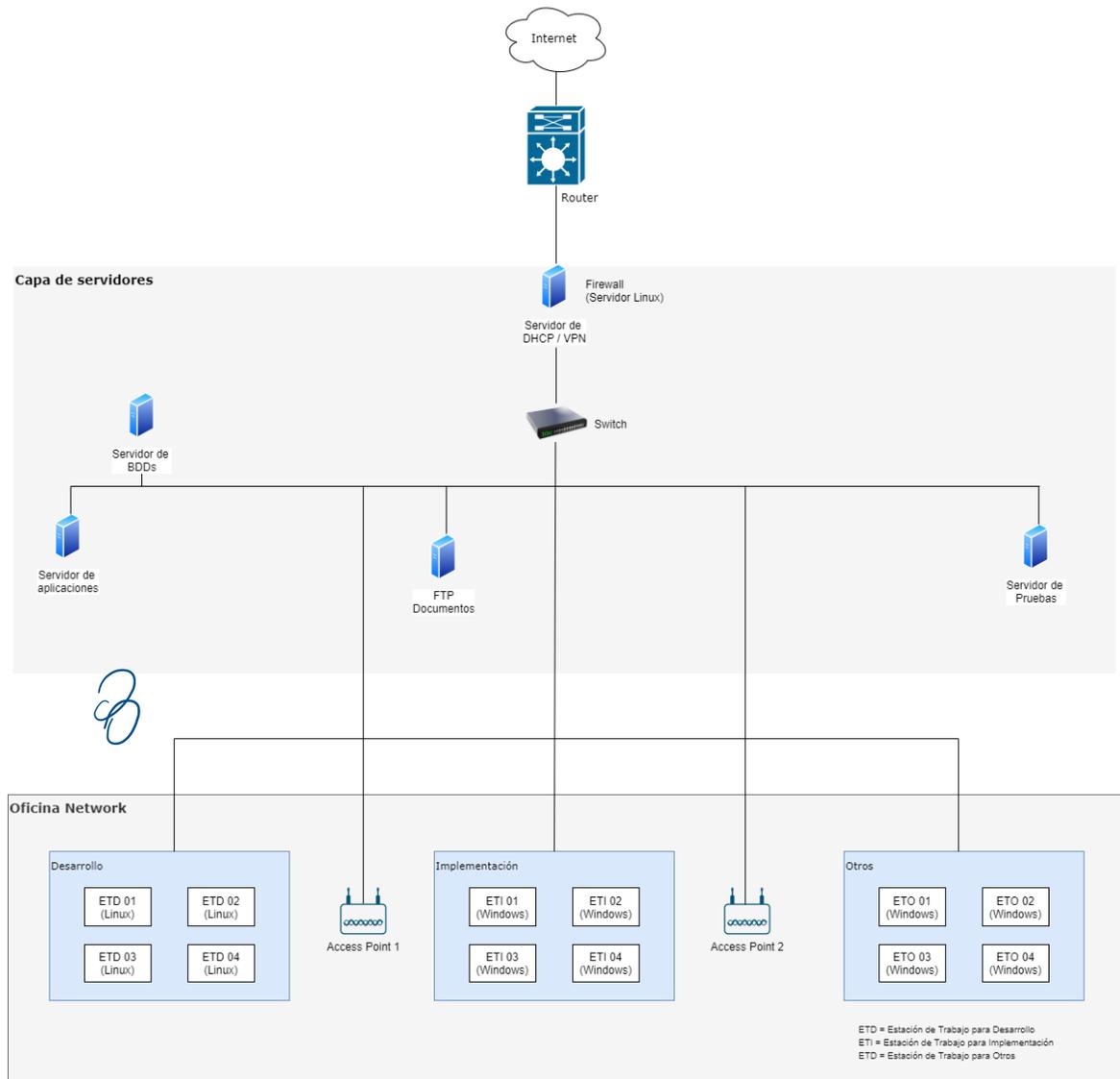


Figura 25 Infraestructura actual

6.1.1. Ambientes de desarrollo de software

Actualmente la fábrica de software mantiene 3 entornos para desarrollar software

- **Ambiente de Desarrollo:** Este es el lugar donde se realizan los cambios solicitados por el cliente, las nuevas ideas, o ajustes. Teóricamente nada de lo que se realice acá

afectará realmente la aplicación de software. Sin embargo, esto no siempre ocurre, debido al error y falta de control en el versionamiento. En este ambiente se suelen hacer cambios directos a la base de datos como agregar o quitar columnas, relación entre tablas y datos, nuevas tablas, cambiar los modelos de la base de datos entre otros. La información que se trabaja acá siempre va a terminar siendo muy incoherente con la realidad. (Ortizvivas, 2017)

Actualmente este entorno carece de las características de producción

- **Ambiente de Pruebas:** Normalmente es un lugar que tiene casi los mismos datos del ambiente productivo, pero que se le implementan los cambios realizados en desarrollo. Estos cambios deben haber pasado por las pruebas iniciales y se trasladan a pruebas para ver su integración. En este ambiente es donde el usuario final intenta comprobar que lo que pidió y contrastar con la realidad. Lo que suele pasar es que se da cuenta que algunos efectos que no pensaban terminan ocurriendo, por lo que se solicitan nuevos ajustes. Si se presenta una falla que el desarrollador no visualizó se reporta y se corrige. Todo debe quedar debidamente ajustado antes de realizar el cambio final.

Por lo general muchas veces se realizan pruebas directamente en producción, lo que limita y genera errores en producción. (Ortizvivas, 2017)

- **Ambiente de Producción:** En este ambiente es donde ocurren todas las transacciones de los usuarios en caliente. Dependiendo de que haga el software puede ser de mucho o poco movimiento, pero al final es la realidad de los usuarios. Los cambios que se han solicitado se recomienda realizarlos en horas de poco o ningún acceso. De hecho, se debe intentar pasar la mayor cantidad de cambios para que en el ambiente de pruebas funcionen correctamente. De todos los ambientes de

trabajo para software este es el más delicado, ya que si algo falla, podría darse una de esas famosas “caídas del sistema”.

Muchas veces se presentan fallos que no se presentan en pruebas porque cada usuario maneja la información de forma diferente. Esto se conoce como errores de usuario y por lo general son los que saturan el soporte. El equipo de soporte debe detectar la falla y replicarla en el ambiente de pruebas. Luego debe solicitar la corrección para que en el ambiente de desarrollo se realicen los cambios y pasar por todo el proceso nuevamente. (Ortizvivas, 2017)

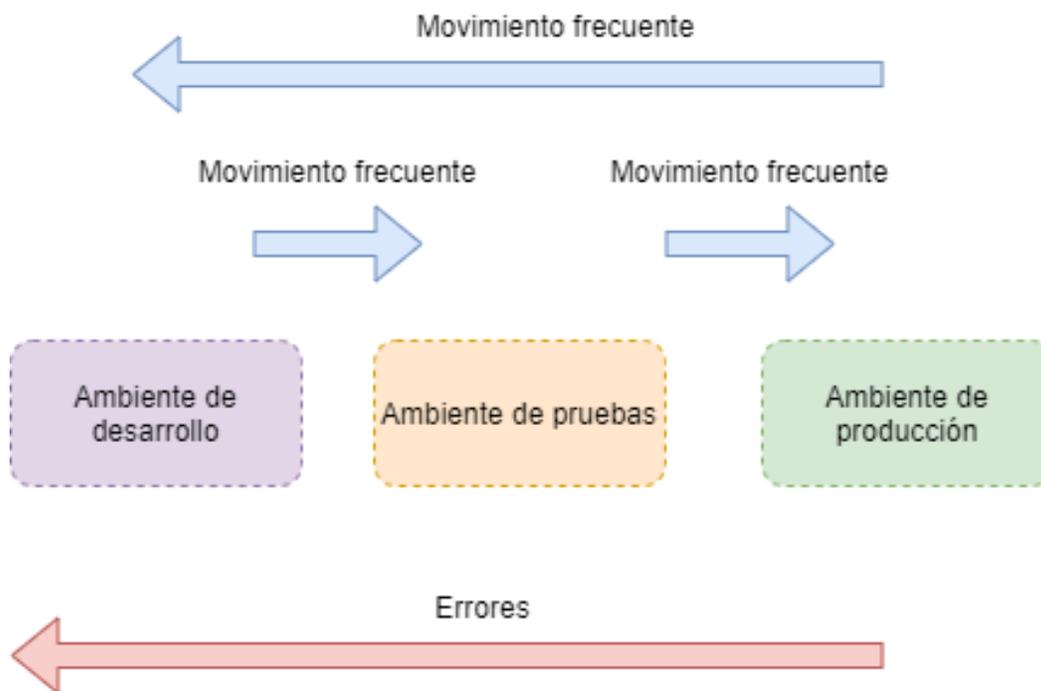


Figura 26 Entornos de desarrollo

Como se puede observar en la Figura anterior los movimientos de información entre cada uno de los entornos es muy frecuente, al no tener control de versionamiento y calidad,

se generarán módulos en producción con inconsistencias y errores, que derivarían a volver a empezar.

6.2. Arquitectura infraestructura base objetivo

Debido al uso de SCRUM, es necesario el despliegue a producción mediante la entrega continua. Entrega continua como la disciplina de desarrollo de software que tiene como objetivo conseguir un despliegue a producción rápido, frecuente y reproducible.

Usando patrones, técnicas y buenas prácticas se consigue la construcción de un software, que se realice de tal forma que en cualquier momento se pueda actualizar su versión en el entorno productivo. Se propone la implementación de un servidor que permita la implementación de entornos de integración continua y de preproducción.

A continuación, se muestran las características del servidor propuesto, en el cual se incorporan algunos servicios como AWS y GIT.

Equipo	Tipo	Estado	Sistema Operativo	Procesador	Memoria RAM	Disco Duro	Detalle de uso
Servidor AWS	WS	Nube	Ubuntu	Intel XEON 2.64 Hz	64 GB	2 TB	Está destinado para ambientes de pruebas/producción internos y externos

Tabla 16 Características de servidor propuesto

Se implementarán los siguientes ambientes de desarrollo:

- **Entorno de desarrollo:** Corresponde básicamente a la máquina de cada desarrollador de tal forma que varias personas pueden estar trabajando en un mismo proyecto a la vez sin molestarte.

Para minimizar incidencias en etapas posteriores este entorno dispondrá de software y una configuración lo más parecida al entorno de producción. Por ejemplo, se eliminará situaciones como desarrollar sobre odoo v8 y que el entorno de producción se encuentre corriendo odoo 14

Solo cuando el desarrollador completa un código perfectamente funcional e integrable en el entorno de pre-producción y tras realizar las pruebas necesarias para asegurar que el software desarrollado tiene la estabilidad suficiente se podrá pasar al entorno de integración continua. (Ahierro, s.f.)

- **Entorno de integración continua:** Este entorno cumple un triple objetivo:
 - Integrar el trabajo de los diferentes desarrolladores en un repositorio central, dando como resultado una versión del código actualizada y consolidada.
 - Automatizar las pruebas de integración y su validación antes de ser movido al siguiente entorno. (Ahierro, s.f.)
 - Enviar el código al siguiente entorno si las pruebas han sido superadas satisfactoriamente.
 - Para este propósito se utilizará SmartGit. Una vez consolidado el código utilizamos los hooks para ejecutar las pruebas definidas, notificar los resultados y enviar a preproducción si es preciso.
- **Entorno de pre-producción:** Una vez superadas las pruebas de integración en el entorno de integración continua, el código será movido al entorno de preproducción. Aquí se realizarán las pruebas de validación al conjunto del software, teniendo como objetivo localizar cualquier error antes de llegar al entorno de producción y evitar así los problemas derivados de ellos.

- **Entorno de producción:** Entorno dónde se verán las virtudes y defectos del proyecto, el objeto por el que será valorado la fábrica de software. Será el ambiente donde se encontrará el código fuente listo para la entrega de mejoras o requerimientos nuevos.
- **Las pruebas:** Uno de los aspectos más significativos de este circuito es que el código fuente no podrá avanzar al próximo ambiente mientras las pruebas no sean satisfactorias. Lo más habitual es realizar test unitarios en el entorno de desarrollo, test de integración en el entorno de integración continua y pruebas de validación sobre el conjunto de la aplicación en el entorno de preproducción.

A continuación, se muestra como estará la infraestructura objetivo tomando en consideración los entornos de desarrollo antes descritos.

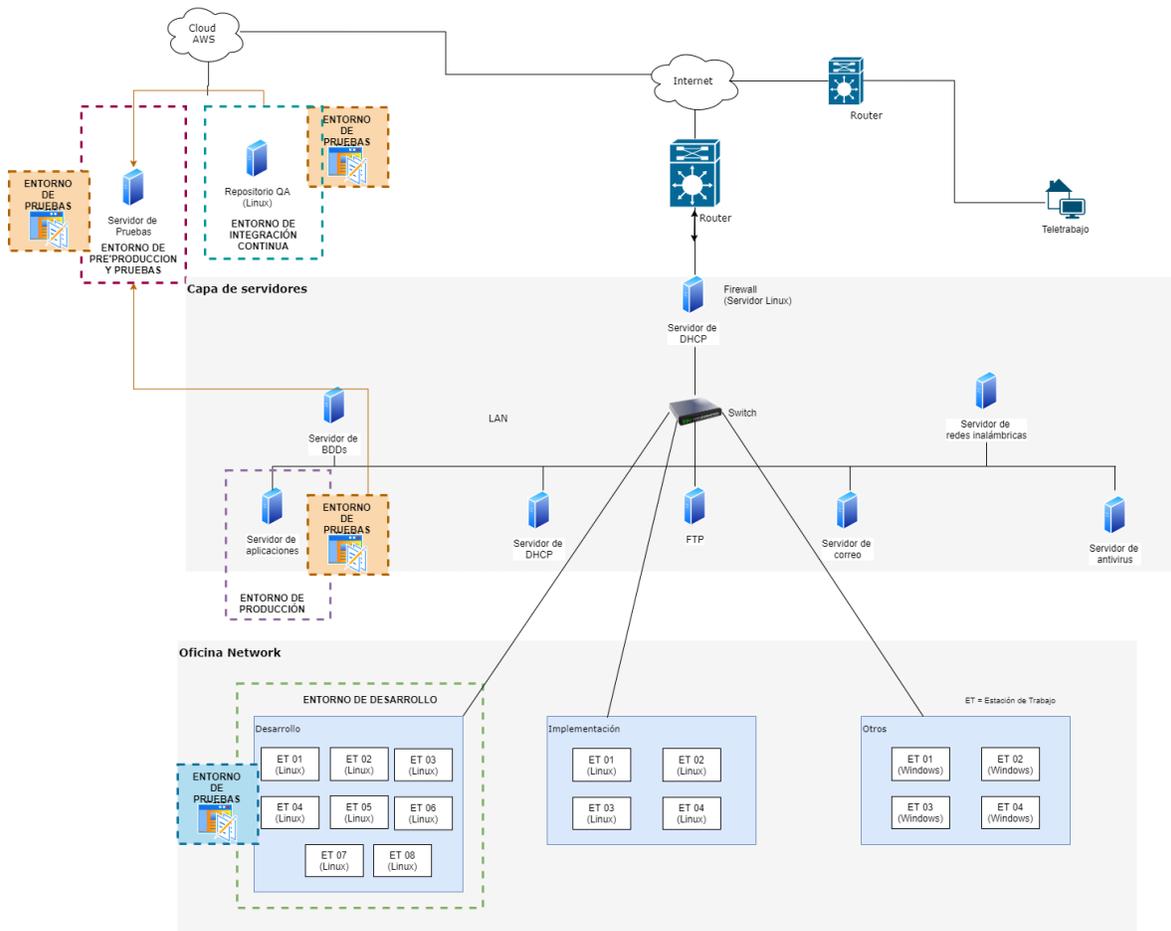


Figura 27 Arquitectura de infraestructura base

En la anterior Figura se muestra la red empresarial a implementarse, tomando en consideración que, en el entorno de desarrollo, las pruebas serán unitarias (Color turquesa), en el computador de cada desarrollador.

En el entorno de producción, preproducción e integración continua (Color naranja), todas las pruebas serán realizadas en el servidor de pruebas, información que será subida a la nube en repositorios GIT.

El desarrollo dentro de la fábrica de software se origina en las estaciones de trabajo de los programadores, una vez establecidos los procesos internos, quienes se encargan de subir

el código fuente a las ramas correspondientes del servidor GIT, utilizando la estructura de red interna definida, pasando por un switch, salen al exterior a través de un firewall para finalmente llegar al GIT. Para el caso del teletrabajo se conectan directamente desde su propio router.

El asegurador de la calidad (QA) accede a los servidores de Amazon Web Services (AWS) para realizar las pruebas correspondientes; ya sea desde la red interna de la empresa con la misma estructura de los desarrolladores o directamente desde su casa cuando aplique teletrabajo.

Adicionalmente dentro de la red interna de la empresa existe el servidor DHCP encargado de asignar las IPs a todas las estaciones de trabajo dentro de la empresa, el servidor FTP utilizado para el almacenamiento de información no estructurada que se manejan en los distintos procesos dentro de la empresa, el servidor de antivirus es el encargado de proteger la red y las estaciones de ataques de virus, el servidor de correo mediante el cual se gestionan los emails y chats de la empresa, el servidor de aplicaciones en el cual se encuentra el sistema de gestión empresarial ERP Olympe para la gestión interna de la oficina.

el firewall se encarga de establecer las políticas internas de la red. De igual forma se encuentran varias estaciones de trabajo adicionales correspondientes a otros departamentos medulares o de apoyo.

De igual forma se debe mencionar que existe un alto tráfico desde la red interna hacia los servidores de AWS y GIT según lo establecido en el proceso de desarrollo informático.

7. OPORTUNIDADES Y SOLUCIONES

Se recopila todas las iniciativas presentadas en el transcurso de las arquitecturas analizadas.

En la siguiente tabla se muestran todas las iniciativas de acuerdo con su arquitectura, las cuales se detallarán a profundidad en los siguientes apartados.

Arquitectura	Iniciativa
Negocio	Implementación de Scrum
	Capacitación al personal del departamento de desarrollo informático en metodología Scrum
Arquitectura de aplicaciones / Datos	Actualización a versión 14 de Odoo
	Implementación del módulo scrum en Olympo
	Control de actividades procedentes de Scrum en ERP Olympo
	Implementación de SmartGit para control de versionamiento
Arquitectura de infraestructura base	Implementación de servidor Servidor AWS
	Reestructuración de entornos de desarrollo e implementación de entornos de integración continua y pre-producción

Tabla 17 Iniciativas

7.1. Arquitectura de negocio

7.1.1. Iniciativa: Implementación de Scrum

Esta iniciativa propone la implementación de SCRUM, el cual consta de las siguientes fases: (software, 2020)

a) Iniciación

La primera fase se encarga de estudiar y analizar el proyecto identificando las necesidades básicas del sprint.

Las preguntas para hacer en la fase de inicio son:

¿Qué quiero?

¿Cómo lo quiero?

¿Cuándo lo quiero?

b) Planificación y Estimación

La clave para llevar una buena administración de los proyectos es hacer una planificación y estimación del sprint, lo que te ayudará a establecer metas fijas y a cumplir con los plazos.

Tal vez esta sea la fase más importante del proyecto, donde se delegará las tareas correspondientes a cada grupo, hacer las estimaciones de tiempos de entrega, así como crear una lista ordenada para clasificar el trabajo según su prioridad.

c) Implementación

Es decir, la sala de reuniones donde se discute el sprint y se explora cómo optimizar el trabajo de cada grupo Scrum para darle forma definitiva al proyecto.

En la fase de implementación o desarrollo no deberían hacerse cambios innecesarios de última hora (se supone que para evitarlo existe una fase de planificación).

d) Revisión y Retrospectiva

Una vez que ya todo está maquetado e implementado, se realiza la revisión del proceso, que no es más que la autocrítica o evaluación interna del grupo respecto a su propio trabajo.

Es importante sumar opiniones constructivas y aportar soluciones viables.

e) Lanzamiento

Es el desenlace del proyecto y entrega del producto

7.1.1.1. Puntos Clave del Scrum

Estos son algunos puntos claves del Scrum que muchos equipos consideran atractivos:

- Es auto organizado y se enfoca en los equipos.
- No hay necesidad de grandes cantidades de documentación, en vez de eso, hay puntos, historias y tareas muy precisas.
- Los equipos multifuncionales funcionan como una sola unidad.
- Comunicación cercana y mucha interacción.
- Tiene un ritmo definido y repetitivo para completar los trabajos a un máximo de 30 días.
- En lugar de intentar hacer todo al mismo tiempo, Scrum ayuda a completar una pequeña parte del todo en un intervalo dado.
- Se confía en la habilidad de los individuos y su capacidad es conocida antes de que se comprometan con nada.

7.1.1.2. Consideraciones que tomar con el uso de Scrum

- El propietario de un producto crea una lista de prioridades.
- Durante la planificación del sprint, el equipo toma una pequeña parte de la lista de prioridades y decide cómo implementar esos elementos.
- El equipo tiene un cierto tiempo (un sprint que usualmente dura de dos a cuatro semanas) para completar este trabajo, pero se reúne cada día para evaluar el proceso (Scrum diario).
- Durante el camino, el Scrum Master mantiene al equipo enfocado en la meta.
- Al final del Sprint, el trabajo debe estar potencialmente enviable: listo para el cliente.
- El sprint finaliza con un análisis y retrospectiva.
- A medida que el siguiente Sprint comienza, el equipo selecciona otro pedazo de los productos en la lista y comienza a trabajar nuevamente.
- Esto continúa hasta que el proyecto se considere completado, ya sea por dejar de trabajar en el mismo (plazo, presupuesto, etc.) o por haberse completado la lista completa.

7.1.1.3. Eventos de Scrum

El Sprint es un contenedor para todos los demás eventos. Cada evento en Scrum es una oportunidad formal para inspeccionar y adaptar los artefactos Scrum. Estos eventos están diseñados específicamente para permitir la transparencia requerida. No operar cualquier evento según lo prescrito da como resultado la pérdida de oportunidades para inspeccionar y adaptar. Los eventos se utilizan en Scrum para crear regularidad y minimizar la necesidad de reuniones no definidas en Scrum.

De manera óptima, todos los eventos se llevan a cabo al mismo tiempo y en el mismo lugar para reducir la complejidad. (Scrum, 2020)

- Sprint
- Sprint planning
- Scrum diario
- Retrospectiva del Sprint

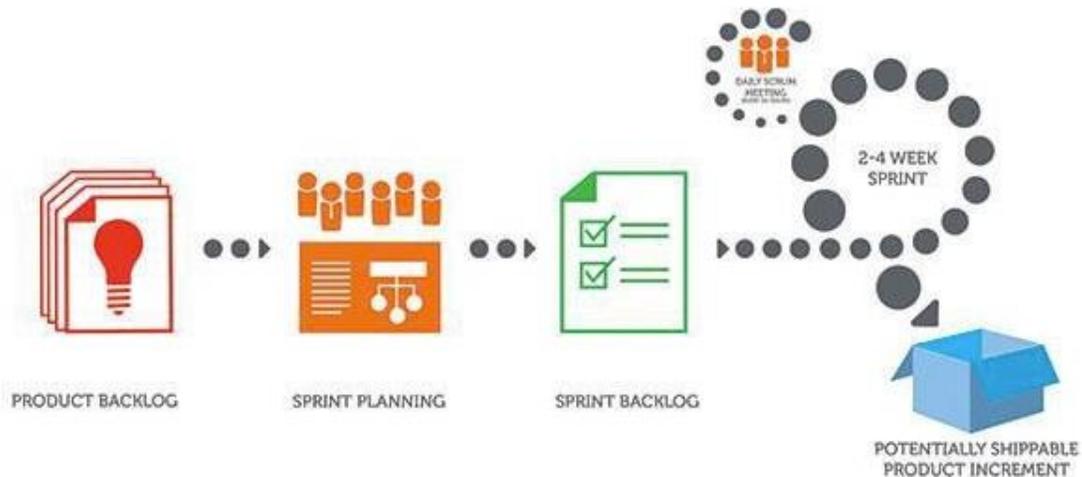


Figura 28 Eventos de Scrum

7.1.1.3.1. Sprint

Los *sprints* son el latido del corazón de Scrum, donde las ideas se convierten en valor.

Son eventos de duración fija de un mes o menos para crear coherencia. Un nuevo Sprint comienza inmediatamente después de la conclusión del Sprint anterior.

Todo el trabajo necesario para lograr el objetivo del producto, incluida la planificación de Sprint, Scrums diarios, Revisión de Sprint y Retrospectiva de Sprint, ocurre dentro de *Sprints*. (Scrum, 2020)

Durante el Sprint:

- No se realizan cambios que pongan en peligro el Sprint Goal;

- La calidad no disminuye;
- La Pila de Producto se refina según sea necesario; y,
- El alcance se puede aclarar y renegociar con el propietario del producto a medida que se aprende más.

Los *sprints* permiten la previsibilidad, al garantizar la inspección y adaptación del progreso hacia un objetivo al menos cada mes calendario. Cuando el horizonte de un Sprint es demasiado largo, el Objetivo del Sprint puede volverse inválido, la complejidad y el riesgo pueden aumentar. Se pueden emplear *Sprints* más cortos para generar más ciclos de aprendizaje y limitar el riesgo de costo y esfuerzo a un marco de tiempo menor. Cada Sprint puede considerarse un proyecto corto.

Un Sprint podría cancelarse si el objetivo de Sprint se vuelve obsoleto. Solo el propietario del producto tiene la autoridad para cancelar el Sprint.

7.1.1.3.2. Sprint planning

Sprint Planning inicia el Sprint al establecer el trabajo que se realizará para el Sprint. Este plan resultante es creado por el trabajo colaborativo de todo el Equipo Scrum.

El Product Owner se asegura de que los asistentes estén preparados para discutir los elementos más importantes del Product Backlog y cómo se asignan al objetivo del sprint. El equipo Scrum también puede invitar a otras personas a asistir a *Sprint Planning* para brindar asesoramiento. (Scrum, 2020)

Sprint Planning aborda los siguientes temas:

Tema uno: ¿Por qué es valioso este Sprint?

El *Product Owner* propone cómo el producto podría incrementar su valor y utilidad en el Sprint actual. Todo el equipo Scrum luego colabora para definir un *Sprint Goal* que comunica por qué el Sprint es valioso para las partes interesadas. El objetivo del Sprint se debe finalizar antes de que finalice la planificación del Sprint.

Tema dos: ¿Qué se puede hacer en este Sprint?

A través de una conversación con el propietario del producto, los desarrolladores seleccionan elementos de la lista de trabajos pendientes del producto para incluirlos en el Sprint actual. El Equipo Scrum puede refinar estos elementos durante este proceso, lo que aumenta la comprensión y la confianza.

Seleccionar cuánto se puede completar dentro de un Sprint puede ser un desafío. Sin embargo, cuanto más sepan los Desarrolladores sobre su desempeño pasado, más confiados estarán en sus pronósticos de Sprint, futuros.

Tema tres: ¿Cómo se realizará el trabajo elegido?

Para cada elemento de la Lista de Producto seleccionado, los desarrolladores planifican el trabajo necesario para crear un Incremento que cumpla con la definición de Terminado. Esto se hace a menudo descomponiendo los elementos de la Lista de Producto en elementos de trabajo más pequeños de un día o menos. La forma en que se hace esto queda a criterio exclusivo de los desarrolladores. Nadie más les dice cómo convertir los elementos de la Lista de Producto en Incrementos de valor.

El objetivo del Sprint, los elementos del Backlog de productos seleccionados para el Sprint, más el plan para entregarlos, se denominan juntos Backlog del Sprint.

La planificación del Sprint tiene un límite de tiempo de un máximo de ocho horas para un Sprint de un mes. Para *Sprints* más cortos, el evento suele ser más pequeño.

7.1.1.3.3. Scrum diario

El propósito del Daily Scrum es inspeccionar el progreso hacia el objetivo de Sprint y adaptar el Sprint Backlog según sea necesario, ajustando el próximo trabajo.

El Daily Scrum es un evento de 15 minutos para los desarrolladores del equipo Scrum. Para reducir la complejidad, se lleva a cabo a la misma hora y en el mismo lugar todos los días hábiles del Sprint. Si el Product Owner o Scrum Master están trabajando activamente en elementos del Sprint Backlog, participan como Desarrolladores.

Los desarrolladores pueden seleccionar la estructura y las técnicas que deseen, siempre que su Scrum diario se centre en el progreso hacia el objetivo y produzca un plan viable para el siguiente día de trabajo. Esto crea enfoque y mejora la autogestión.

Los Scrums diarios mejoran las comunicaciones, identifican impedimentos, promueven la toma de decisiones rápida y en consecuencia, eliminan la necesidad de otras reuniones.

El Scrum diario no es el único momento en el que los desarrolladores pueden ajustar su plan. A menudo se reúnen a lo largo del día para discusiones más detalladas sobre cómo adaptar o volver a planificar el resto del trabajo del Sprint.

Revisión de Sprint

El propósito de Sprint Review es inspeccionar el resultado del Sprint y determinar futuras adaptaciones. El equipo Scrum presenta los resultados de su trabajo a las partes interesadas y se discute el progreso hacia el objetivo del producto.

Durante el evento, el equipo Scrum y las partes interesadas revisan lo que se logró en el Sprint y lo que ha cambiado en su entorno. Con base en esta información, los asistentes colaboran sobre qué hacer a continuación. La pila de producto también se puede ajustar para encontrar nuevas oportunidades. La revisión de Sprint es una sesión de trabajo y el equipo Scrum debe evitar limitarla a una presentación.

La revisión del Sprint es el penúltimo evento del Sprint y tiene un límite de tiempo de un máximo de cuatro horas para un Sprint de un mes. Para *Sprints* más pequeños, el evento suele ser más corto. (Scrum, 2020)

7.1.1.3.4. Retrospectiva del Sprint

El propósito de la Retrospectiva de Sprint es planificar formas de aumentar la calidad y la eficacia.

El equipo scrum inspecciona cómo fue el último Sprint con respecto a las personas, las interacciones, los procesos, las herramientas y su definición de terminado. Los elementos inspeccionados a menudo varían según el ámbito del trabajo. Se identifican los supuestos que los llevaron por mal camino y se exploran sus orígenes. El equipo scrum analiza qué salió bien durante el Sprint, qué problemas encontró y cómo se resolvieron (o no) esos problemas.

El equipo scrum identifica los cambios más útiles para mejorar su efectividad. Las mejoras más impactantes se abordan lo antes posible. Incluso pueden agregarse al Sprint Backlog para el próximo Sprint.

La Retrospectiva del Sprint concluye el Sprint. Tiene un límite de tiempo de un máximo de tres horas para un Sprint de un mes. Para *Sprints* más cortos, el evento suele ser más pequeño. (Scrum, 2020)

7.1.1.4. Artefactos Scrum

Los artefactos de Scrum representan trabajo o valor. Están diseñados para maximizar la transparencia de la información clave. Por lo tanto, todos los que los inspeccionan tienen la misma base para la adaptación.

Cada artefacto contiene un compromiso que garantiza la generación de información que mejore la transparencia y el enfoque frente al cual se pueda medir el progreso (Scrum, 2020):

- Para el *Product Backlog* es el Objetivo del producto.
- Para el *Sprint Backlog* es el Objetivo del sprint.
- Para el Incremento es la Definición de Terminado.

Estos compromisos existen para reforzar el empirismo y los valores de Scrum para el equipo Scrum y sus partes interesadas.

7.1.1.4.1. Product Backlog

El *Product Backlog* es una lista emergente y ordenada de lo que se necesita para mejorar el producto. Es la única fuente de trabajo realizada por el equipo Scrum

Los elementos del Backlog de productos que puede realizar el equipo dentro de un Sprint se consideran listos para ser seleccionados en un evento de *Sprint Planning*. Suelen adquirir este grado de transparencia después de las actividades de refinamiento. El refinamiento de la Lista de Producto es el acto de dividir y definir aún más los elementos de la Lista de Producto en elementos más pequeños y precisos. Esta es una actividad continua para agregar detalles, como una descripción, orden y tamaño. Los atributos a menudo varían según el ámbito del trabajo.

Los Desarrolladores que realizarán el trabajo son responsables del dimensionamiento. El propietario del producto puede influir en los desarrolladores ayudándolos a comprender y seleccionar las compensaciones. (Scrum, 2020)

Objetivo del producto

El objetivo del producto describe un estado futuro del producto que puede servir como un objetivo para que el equipo Scrum planifique. El objetivo del producto está en la lista de trabajos pendientes del producto. El resto del Product Backlog surge para definir "qué" cumplirá con el Product Goal.

Un producto es un vehículo para ofrecer valor. Tiene un límite claro, partes interesadas conocidas, usuarios o clientes bien definidos.

Es el objetivo a largo plazo del equipo Scrum. Deben cumplir (o abandonar) un objetivo antes de asumir el siguiente.

7.1.1.4.2. Sprint Backlog

El Sprint Backlog se compone del objetivo del sprint (por qué), el conjunto de elementos del Product Backlog seleccionados para el Sprint (qué), así como un plan procesable para entregar el Incremento (cómo).

El Sprint Backlog es un plan realizado por y para los desarrolladores. Es una imagen muy visible y en tiempo real del trabajo que los desarrolladores planean realizar durante el Sprint para lograr el objetivo del Sprint. En consecuencia, el Sprint Backlog se actualiza a lo largo del Sprint a medida que se aprende más. Debe tener suficientes detalles para que puedan inspeccionar su progreso en el daily Scrum.

Objetivo del Sprint

Es el único objetivo del Sprint. Aunque es un compromiso de los desarrolladores, proporciona flexibilidad en términos del trabajo exacto necesario para lograrlo, este también crea coherencia y enfoque, alentando al Equipo Scrum a trabajar juntos en lugar de en iniciativas separadas.

Se crea durante el evento Sprint Planning y luego se agrega al Sprint Backlog. Mientras los desarrolladores trabajan durante el Sprint, tienen en mente el Objetivo del Sprint. Si el trabajo resulta ser diferente de lo que esperaban, colaboran con el Product Owner para negociar el alcance del Sprint Backlog dentro del Sprint sin afectar el Sprint Goal. (Scrum, 2020)

7.1.1.4.3. Incremento

Un incremento es un trampolín concreto hacia el objetivo del producto. Cada Incremento se suma a todos los Incrementos anteriores y se verifica minuciosamente, lo que garantiza que

todos los Incrementos funcionen juntos. Para proporcionar valor, el Incremento debe ser utilizable.

Se pueden crear múltiples incrementos dentro de un Sprint. La suma de los Incrementos se presenta en la Sprint Review apoyando así el empirismo. Sin embargo, se puede entregar un Incremento a las partes interesadas antes del final del Sprint. La Revisión de Sprint nunca debe considerarse una puerta para liberar valor.

Objetivo del producto terminado

La Definición de Terminado es una descripción formal del estado del Incremento cuando cumple con las medidas de calidad requeridas para el producto.

En el momento en que un elemento de la Pila de Producto cumple con la Definición de Terminado, nace un Incremento.

La Definición de Terminado crea transparencia al proporcionar a todos un entendimiento compartido de qué trabajo se completó como parte del Incremento. Si un artículo de la Lista de Producto no cumple con la Definición de Terminado, no se puede publicar ni presentar en la Revisión de Sprint. En su lugar, vuelve al Product Backlog para su consideración futura.

Si la Definición de Terminado para un incremento es parte de los estándares de la organización, todos los Equipos Scrum deben seguirla como mínimo. Si no es un estándar organizacional, el Equipo Scrum debe crear una Definición de Hecho apropiada para el producto.

Los desarrolladores deben ajustarse a la definición de Terminado. Si hay varios Equipos Scrum trabajando juntos en un producto, deben definir y cumplir mutuamente la misma Definición de Hecho. (Scrum, 2020)

7.1.1.5. Implementación de Scrum

- a. Elegir un responsable de producto.
- b. Elegir al equipo, de 4 a 9 personas.
- c. Elegir un Scrum máster.
- d. Elaborar y priorizar el backlog.
- e. Hacer una estimación afinada de la lista de objetivos pendientes.
- f. Planificar el primer sprint valorando la cantidad de trabajo.
- g. Hacer que el trabajo se haga visible.
- h. Hacer reunión diaria: misma hora, mismo lugar, máximo 15 minutos.
- i. Al final del sprint, revisar los resultados.
- j. Buscar mejora continua de la productividad.

7.1.1.6. Características de la iniciativa

- **Personal:** 10 integrantes del departamento de desarrollo informático (*Product Owner*, Scrum Máster y equipo de desarrollo), Líder del proyecto
- **Costo referencial:** 5000\$
- **Tiempo aproximado:** 1 mes

7.1.2. Iniciativa: Capacitación al personal del departamento de desarrollo informático en metodología Scrum

El objetivo de esta iniciativa es lograr que el equipo Scrum obtenga la certificación "Professional Scrum Developer"

Para este propósito se pretende realizar la capacitación en Scrum.org (scrum.org, 2021)

7.1.2.1. Objetivos del Curso

- Experimentar una colaboración real entre los desarrolladores, el propietario del producto y Scrum
- Dominar la búsqueda de construir y entregar un producto valioso y de alta calidad.
- Crear y entregar software funcional aplicando prácticas modernas de ingeniería ágil
- Comprender la sinergia entre los diversos elementos de Scrum y las prácticas complementarias.

7.1.2.2. Temas del curso

- El marco de Scrum
- Trabajar dentro de un equipo Scrum
- Definición de Terminado
- Prácticas de gestión de trabajos pendientes y funciones de segmentación
- Calidad del código y deuda técnica
- Prácticas ágiles de arquitectura
- Desarrollo basado en pruebas
- Programación en pareja
- Pruebas ágiles y otras prácticas para garantizar la calidad

7.1.2.3. Características de la iniciativa

- Clase privada
- Tecnologías específicas de .NET y Java
- Curso práctico de 3 días en el que los estudiantes experimentan cómo entregar software de calidad utilizando Scrum con prácticas Agile
- Para el desarrollo de software (APS-SD)
- Certificación Professional Scrum Developer I (PSD I)
- Precio de certificación 200,00 \$
- Puntaje de aprobación: 85%
- Límite de tiempo: 60 minutos
- Número de preguntas: 80
- Formato: opción múltiple, respuesta múltiple y verdadero / falso
- Dificultad: intermedia
- Idioma: solo inglés
- Precio aproximado total : 1600\$
- Personal del departamento de desarrollo informático y líder del proyecto para monitoreo de cumplimiento

7.1.2.4. Detalle de certificación



Si aprueba la evaluación PSD, recibirá la certificación "PSD" reconocida por la industria, junto con un logotipo PSD que puede utilizar para identificar su logro. Además, su nombre aparecerá en Scrum.org.

A diferencia de otras certificaciones de Scrum que solo requieren asistencia a clase, la certificación de Scrum.org requiere una puntuación mínima en una evaluación rigurosa en línea. Esto le da fuerza a la certificación Scrum.org y asegura que tiene un valor real en el mercado.

7.2. Arquitectura de aplicaciones / Datos

7.2.1. Iniciativa: Actualización a versión 14 de Odoo

Para la actualización de Odoo es necesario tomar en consideración:

- Migración a través de una instalación de Odoo
- Migración de la base de datos de Odoo

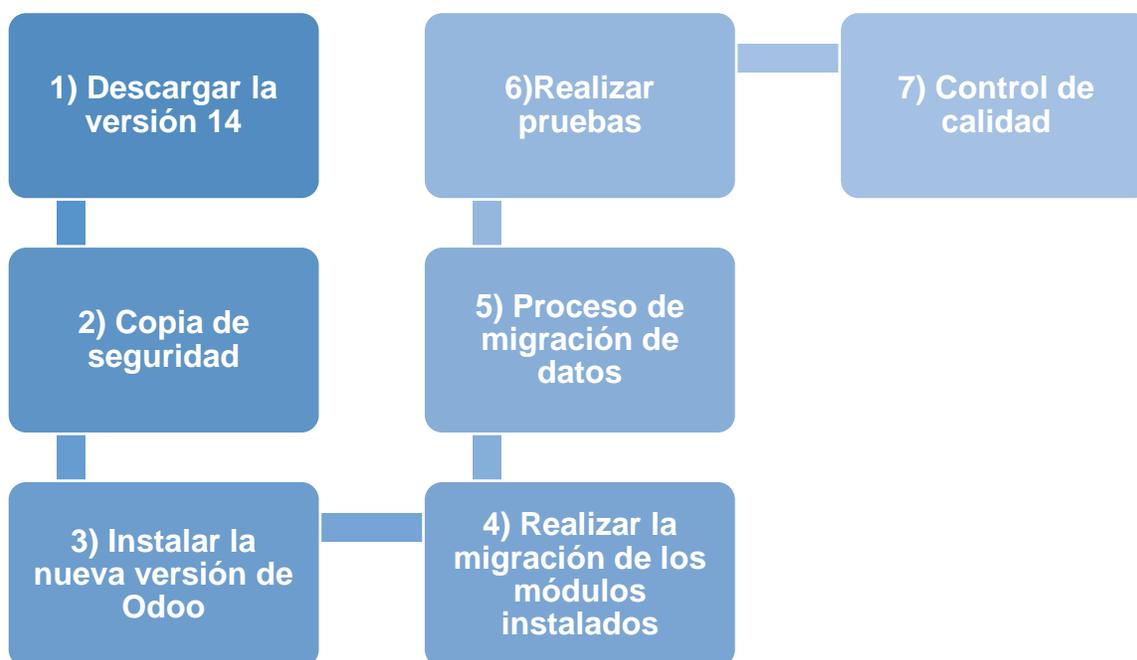
7.2.1.1. Plan de migración dinámica

La actualización se realizará por medio del archivo de instalación

Los datos almacenados no sufrirán ninguna alteración siempre y cuando no sea desinstalado el motor de la base de datos de Odoo (PostgreSQL).

En esta migración se moverán los datos dinámicos (datos que se relacionan con otros datos) y datos estáticos. (Cabrera, Octopus, 2020)

Para este propósito se seguirán los siguientes pasos



i. Paso1: Descargar la versión 14 de Odoo

Obtener una nueva versión de Odoo es a través del enlace incluido en el correo de confirmación de compra de la licencia de Odoo Enterprise. (Cabrera, Octopus, 2020)

ii. Paso2: Copia de seguridad

Realizar el proceso de actualización de Odoo es muy seguro y no debería generar ningún cambio en la base de datos. Pero para evitar cualquier error inesperado se recomienda realizar una copia de seguridad de la base de datos antes de realizar el proceso de actualización. Es recomendable que esta copia de seguridad de la base de datos sea almacenada en un lugar seguro, de ser posible en otra computadora. (Cabrera, Octopus, 2020)

iii. Paso3: Instalar la nueva versión de Odoo

A través del paquete de instalación. Para ello dirigirse al sitio oficial de Odoo y descargar la nueva versión del paquete de instalación correspondiente e instalarlo en tu servidor.

Una vez instalado el nuevo paquete de instalación se debe reiniciar el servicio de Odoo o reiniciar el servidor para que todo funcione correctamente. (Cabrera, octopus, 2020)

iv. Paso4: Realizar la migración de los módulos instalados

Módulos centrales: Estos son los módulos predeterminados que ofrece Odoo cuando se instala Odoo (Odoo Inventario, Odoo Ventas, Odoo Comercio Electrónico, Odoo Contabilidad, entre otros).

Tener en cuenta que algunos módulos han sido modificados, eliminados o movidos de una versión a otra de Odoo. razón por la cual es necesario realizar un análisis de todos los módulos instalados en la versión actual de Odoo y su compatibilidad con la nueva versión. (Cabrera, octopus, 2020)

Módulos principales: Estos son los módulos a los que se le realizó algún tipo de personalización (Módulos: Presupuesto, Control de Activos, Garantías, Gestión de vehículos, Viáticos, Administración financiera, Gestión Inventario, Talento Humano (Vacaciones, Asistencia, Roles de pago), Recaudaciones, Comercial Agua). Antes de realizar la migración de este tipo de módulos debe realizarse un análisis para determinar su compatibilidad con la nueva versión de Odoo que se instalará.

v. Paso5: Proceso de migración de datos

Copiar los datos en una base de datos de prueba y realizar pruebas de control de calidad y una vez que son obtenidos resultados satisfactorios en cada una de ellas se realiza la migración de la información de la base de datos de producción. (Cabrera, Octopus, 2020)

Cargar base de datos

Cargar la base del servidor de pruebas.

La plataforma de actualización acepta una gran cantidad de diferentes formatos:

PostgreSQL volcado de encargo, PostgreSQL tar-formato , sql texto plano o comprimido sql texto plano .

Actualizar y probar base de datos

Una vez que se recibe la base de datos, se ejecuta el proceso de actualización y se prueba

Si el proceso de actualización finalizó sin ningún problema, recibirá un correo electrónico dentro de unas horas, con un enlace donde puede descargar su base de datos actualizada,

Si el proceso de actualización no puede finalizar automáticamente, recibirá un correo electrónico en unas pocas horas, explicando que el proceso de actualización encontró algunas dificultades y que es necesaria una intervención manual.

vi. Paso6: Realizar pruebas

Una vez que se realiza el proceso de migración de datos deben realizarse pruebas para comprobar que dicho proceso se ejecutó de forma correcta. A continuación, se detalla los

dos tipos de pruebas que se realizan una vez culminado el proceso de migración de datos:
(Cabrera, octopus, 2020)

Prueba de humo: este tipo de prueba se le realiza a la estructura y a la interfaz del software para comprobar y solucionar la ocurrencia de errores inesperados en su interfaz y funcionamiento.

Prueba de escenario: este tipo de pruebas se realizan a petición del cliente y se encargan de comprobar el funcionamiento del software en diferentes escenarios y situaciones y de implementar soluciones a los diferentes errores que se puedan identificar.

vii. Paso7: Control de calidad

Realizar el control de calidad asegura que todos los flujos de la nueva versión instalada funcionen correctamente. En este proceso se verifica que en cada operación se obtenga de la base de datos la información correcta y que no se muestren campos irrelevantes en las diferentes vistas ofrecidas al usuario. (Cabrera, octopus, 2020)

7.2.1.2. Características de la iniciativa

- **Tiempo de implementación:** 2 meses
- **Personal:** 3 desarrolladores, 1 Scrum Master, 1 *Product Owner*, Líder de proyecto
- **Precio aproximado:** 4000\$ anuales. (Precio Odoo14)
- **Precio total aproximado:** 12600\$(Sueldos)+4000=16600\$ (Dos mes)

Se debe tomar en cuenta que estos valores serán una inversión y regresaran a la empresa ya, que forman parte de un nuevo módulo a la venta desde el ERP Olympo Se considera el sueldo de los desarrolladores de 1000\$, product owner 1500\$ y Scrum Master 1800\$

Elija sus Aplicaciones

Aplicación	Precio / mes	Estado
CRM	\$8.00 USD	✓
Facturación	\$4.00 USD	✓
Ventas	\$4.00 USD	✓
Web	\$8.00 USD	✓
Comercio electrónico	\$4.00 USD	○
Punto de venta	\$8.00 USD	○
Contabilidad	\$8.00 USD	✓
Proyecto	\$8.00 USD	○
Inventario	\$12.00 USD	✓
Fabricación	\$16.00 USD	✓
Compra	\$4.00 USD	○
Hojas de horas	\$4.00 USD	○
Marketing electrónico	\$4.00 USD	✓
Gastos	\$4.00 USD	✓
Eventos	\$4.00 USD	✓
Ausencias	\$4.00 USD	✓
Contratación	\$4.00 USD	✓
Valoración	\$4.00 USD	✓
Suscripciones	\$8.00 USD	✓
Firmar	\$8.00 USD	✓
Mantenimiento	\$8.00 USD	✓
Calidad	\$8.00 USD	✓
Studio	\$24.00 USD	○
Servicio de asistencia	\$8.00 USD	✓
Product Lifecycle Manag...	\$8.00 USD	○
Citas	\$4.00 USD	○
Automatización de mark...	\$16.00 USD	✓
Documentos	\$8.00 USD	✓
Nodos IoT	\$16.00 USD / month / per box	✓
Aprobaciones	\$4.00 USD	○
Consolidation	\$16.00 USD	✓
Recomendación de empl...	\$4.00 USD	○
Servicio de campo	\$8.00 USD	○
Planificación	\$4.00 USD	✓
Alquiler	\$8.00 USD	○
Marketing Social	\$12.00 USD	✓
Aprendizaje en línea	\$8.00 USD	✓

	Anualmente	Mensual
20 Usuarios		\$160.00 USD
Descuento para usuarios ⁽¹⁾		-\$40.00 USD
25 Aplicaciones		\$200.00 USD
1 Nodos IoT		\$16.00 USD
Total / mes ⁽²⁾		\$336.00 USD
⁽²⁾ Facturado anualmente:		\$4,032.00 USD

PRUEBE AHORA
Prueba gratuita de 15 días

COMPRE AHORA

⁽¹⁾ Los nuevos clientes obtienen un descuento en la cantidad inicial de usuarios adquiridos. (\$6.00 USD en lugar de \$8.00 USD).

Figura 29 Precio de nueva versión de Odoo

7.2.2. Iniciativa: Implementación del módulo Scrum en Olympo

Se pretende implementar en Odoo los módulos de, Project Scrum Management y Project Scrum Portal Agile, mismos que serán adaptados en el ERP Olympo

7.2.2.1. Project Scrum Managment

Este módulo implementa todos los conceptos definidos por la Metodología de Gestión de Proyectos Scrum. Esta aplicación respeta el protocolo scrum.org y tiene las siguientes características: (Drapeau, s.f.)

Roles del proyecto

- Scrum Master
- Dueño del producto
- Equipo de desarrollo

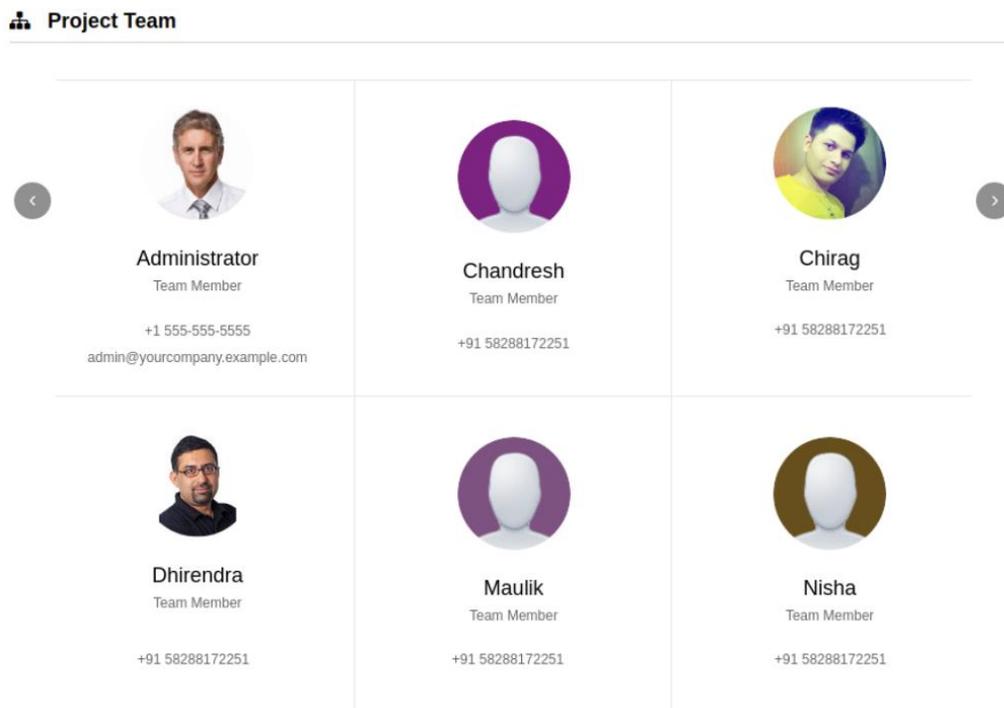


Figura 30 Roles Scrum

Sprints

- date_start y date_stop
- Standup reuniones para cada usuario del equipo
- Revisión de Sprint
- Sprint retrospectiva
- Velocidad planificada (escribe la velocidad deseada y se muestra en la Tabla de velocidad de Sprint)
- Velocidad efectiva (es calculada por todas las historias de usuarios afectadas por el sprint)
- Velocidad efectiva realizada (se calcula mediante todas las historias de usuarios realizadas y se muestra en la Tabla de velocidad de Sprint)

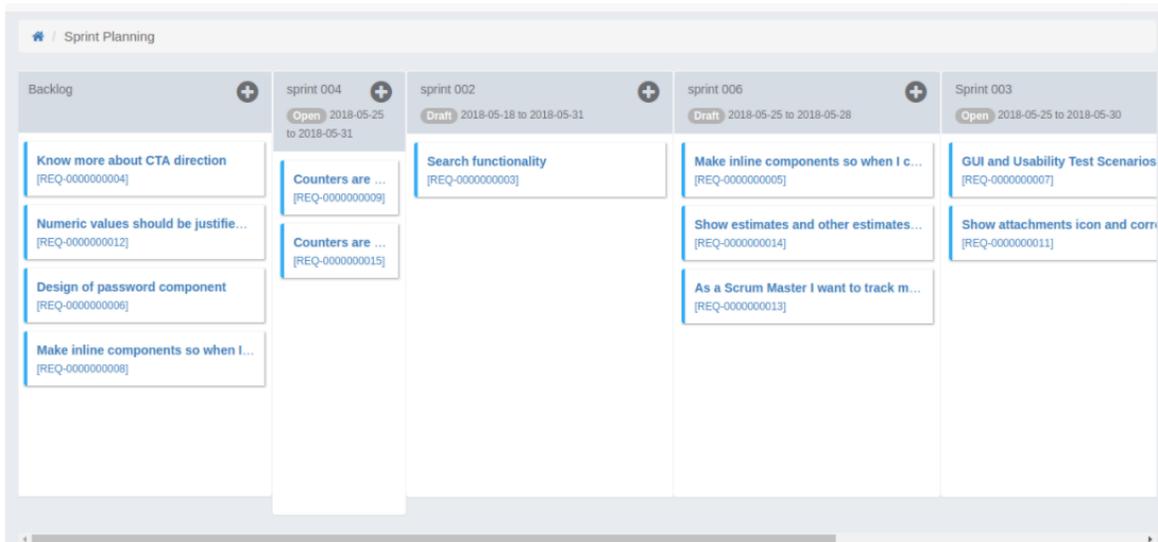


Figura 31 Control de Sprints

Product Backlog (historias de usuarios)

- Nueva vista kanban
- Date_open y date_done
- Puntos de complejidad de la historia
- Área de texto para describir la aceptación de las pruebas

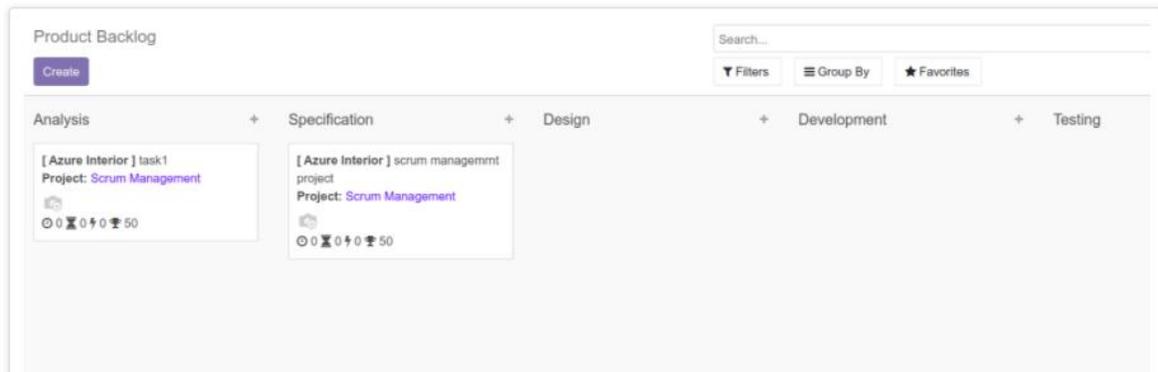


Figura 32 Historias de usuarios

Mostrar gráficos

- Gráfico de reducción (basado en puntos de la historia)
- Velocidad de *Sprints* (para cada proyecto Scrum)

Gestión de proyectos

- Hoja de ruta a mediano y largo plazo
- Clientes / solicitudes funcionales VS técnicas

Reportes

- Cuadro de incendio
- Gráfico de barras

- Gráfico de línea
- Gráfico circular

Los proyectos y tareas de Scrum heredan de los proyectos y tareas reales, por lo que puede continuar trabajando en tareas normales que también incluirán tareas de proyectos de Scrum.

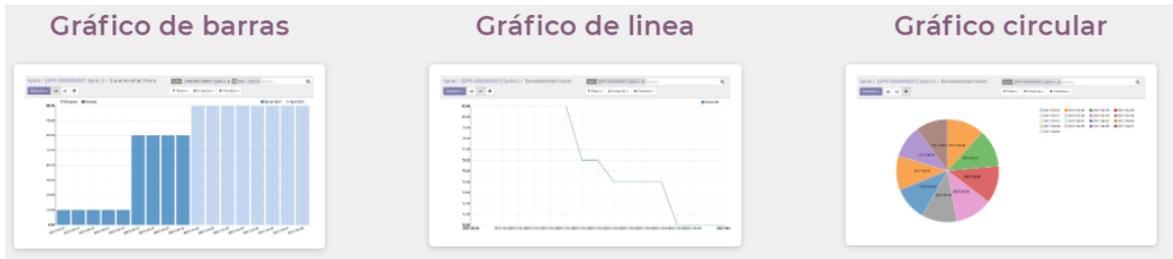


Figura 33 Reportes

7.2.2.2. Implementación de módulo en ERP Olympo

Se instalará el módulo en el ERP actual Olympo, adaptándolo a las necesidades y características, siguiendo estos pasos:

- Descargar desde el repositorio. Se descargará un archivo .zip se debe descomprimir, el resultado será una o varios directorios con el nombre del módulo.
- Luego desde el administrador de Odoo se debe activar las opciones técnicas
- Iniciar sesión como administrador
- Elegir el usuario administrador, marcar la opción de características técnicas y guardar.
- Capacitar al personal de desarrollo, sobre el uso y configuración del módulo
- Adaptar el módulo a Olympo.

7.2.2.3. Características de la iniciativa

- **Tiempo de implementación:** 15 días
- **Personal:** 2 desarrolladores, 1 *Product Owner*, líder del proyecto
- **Precio aproximado de módulo:** 172.42\$
- **Precio de instalación en Olympo:** 2672\$.

Se debe tomar en cuenta que estos valores serán una inversión y regresaran a la empresa ya, que forman parte de un nuevo módulo a la venta desde el ERP Olympo Se considera el sueldo de los desarrolladores de 1000\$ y del product owner 1500\$

7.2.3. Iniciativa: Control de actividades procedentes de Scrum en ERP Olympo

Todas las actividades procedentes de Scrum serán establecidas en el módulo de Scrum de Olympo.

Donde se encuentran las siguientes opciones:

- Gráfico de Sprint Burndown
- Gestión de *Sprints*
- Retrospectiva de *Sprints*
- Estado de la tarea
- Estado del problema
- Carga de trabajo del equipo Scrum
- Detalles de la tarea

Usuarios

- Equipo Scrum
- Scrum Master

- Product Owner
- Gestores de proyectos

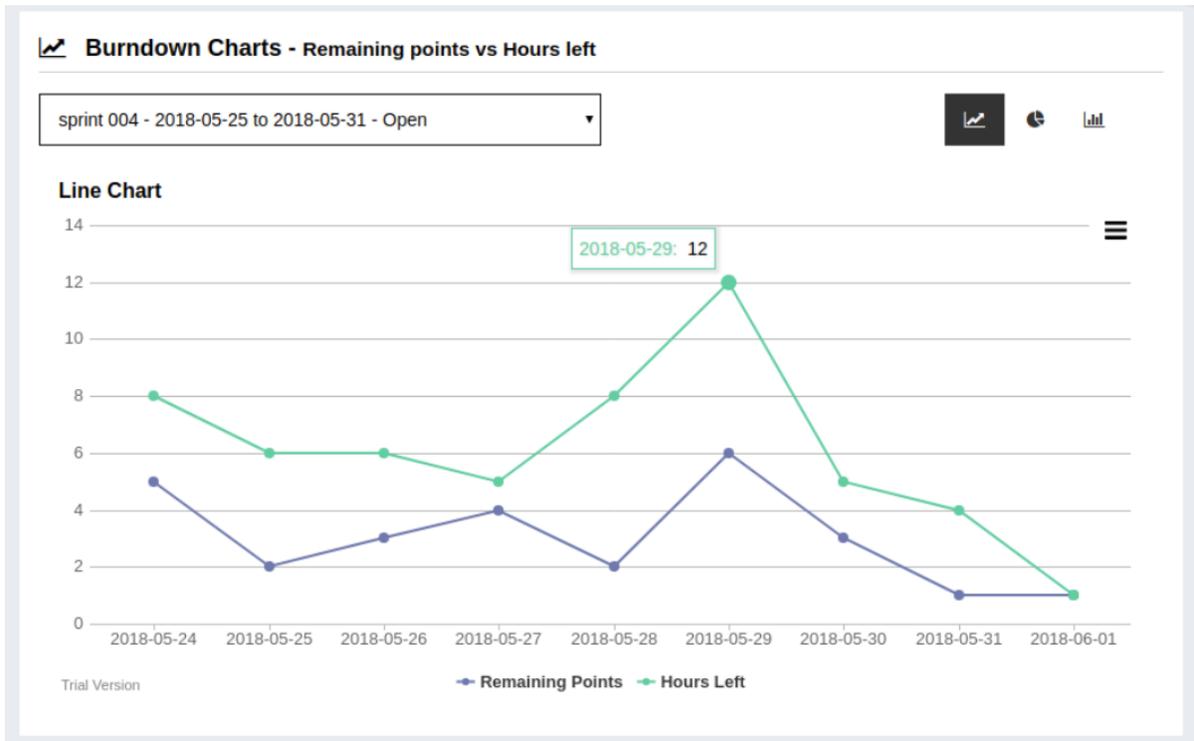


Figura 34 Tipos de gráficos de evolución de Sprint

My Projects

Newest

Name	Tasks	Issues
Website for Sales & WMS	6 Tasks	1 Issues
Research & Development	4 Tasks	3 Bugs
Website Design Templates	3 Designs	7 Bugs
E-Learning Integration	9 Trainings	1 Feedbacks
Data Import/Export Plugin	6 Tasks	0 Issues

Figura 35 Proyectos

Project - Website for Sales & WMS

Customer
YourCompany, Administrator
✉ admin@yourcompany.example.com

6 Tasks 1 Issues

Message and communication history

 Write a message...

Send

Figura 36 Proyectos con todos los detalles

Search functionality

Details General Tasks Notes

REQ-0000000003

Name
Search functionality

Tags
NeedAssistance Feature request Experiment Usability PBCK

Project
E-Learning Integration

Release
R3

Sprint
[SPR-0000000002] sprint 002

Author
Administrator

Start Date
2018-05-23

End Date
2018-05-31

Responsible
Administrator

Application Date
2018-05-30

Who
Chief Technical Officer

Listener
ASUSTeK

Wants
There are many variations of passages of Lorem Ipsum available, but the majority have suffered alteration in some form, by injected humour, or randomised words which don't look even slightly believable.

For
There are many variations of passages of Lorem Ipsum available, but the majority have suffered alteration in some form, by injected humour, or randomised words which don't look even slightly believable.

Proof of acceptance
There are many variations of passages of Lorem Ipsum available, but the majority have suffered alteration in some form, by injected humour, or randomised words which don't look even slightly believable.

Figura 37 Detalles de Backlog

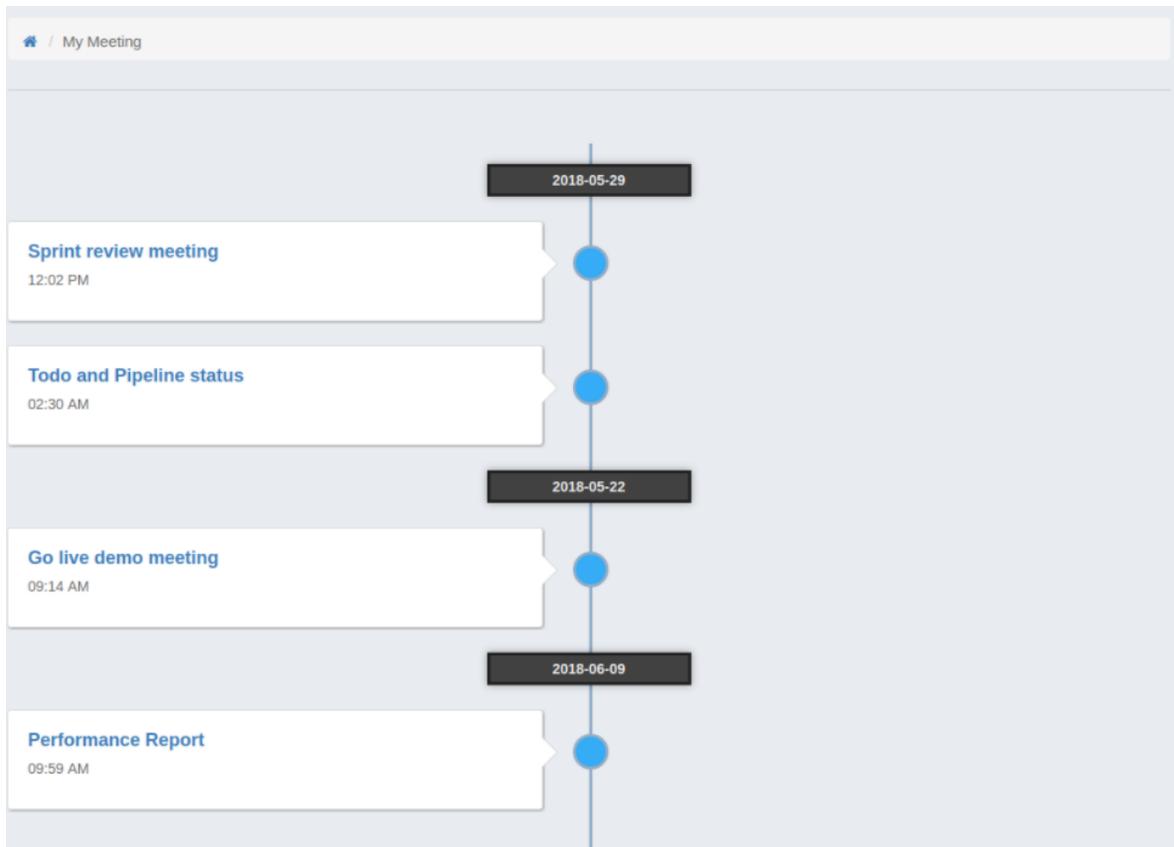


Figura 38 Reuniones

7.2.4. Iniciativa: Implementación de SmartGit para control de versionamiento

El control de versiones es una de las tareas fundamentales para la administración de los diferentes módulos y desarrollos. Surge de la necesidad de mantener y llevar control del código que se va programando, conservando sus distintos estados. Es absolutamente necesario para el trabajo en equipo, ya que existe la necesidad inherente de trabajar en distintas ramas al mismo tiempo, introduciendo cambios a los programas, tanto en la rama de desarrollo como en producción. (Desarrolloweb.com, 2014)

7.2.4.1. Características para la implementación

Usuarios

- Equipo Scrum
- Scrum Master
- Product Owner
- Equipo de desarrollo
- Líder de proyecto

Características

- Comparar el código de un archivo, de modo que se pueda ver las diferencias entre versiones
- Restaurar versiones antiguas
- Fusionar cambios entre distintas versiones
- Trabajar con distintas ramas de un proyecto, por ejemplo, la de producción y desarrollo
- Precio 319\$ (Licencia perpetua)

Sistemas operativos

- Windows
- macOS
- Linux.

Licencias

Se usa una licencia de SmartGit en tantas máquinas y sistemas operativos como desee.

Se adquirirá 8 licencias

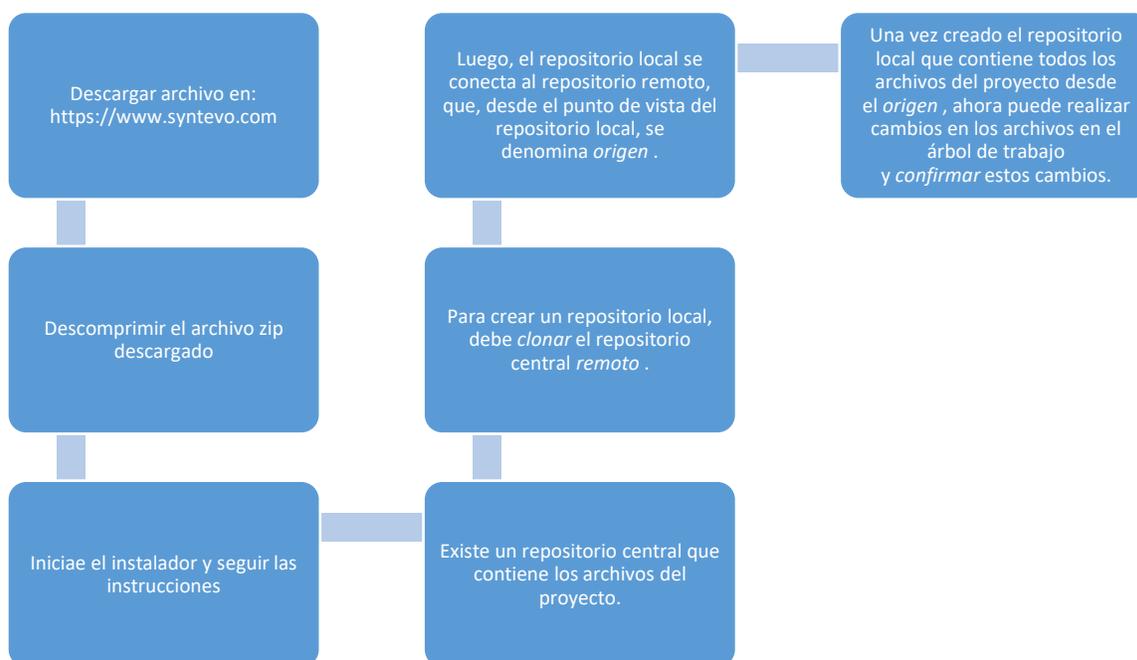
Incluye:

- **Licencias de SmartSynchronize:** El archivo (y el directorio) de SmartGit se compara como una herramienta independiente (SmartGit, 2021)
- **Complemento de revisiones distribuidas:** Revisión de código local, sin conexión y rápida

Precio aproximado

71,92 USD / mes por 8 licencias

7.2.4.2. Implementación de SmartGit



7.3. Arquitectura de infraestructura base

7.3.1. Iniciativa: Implementación de Servidor AWS

Se pretende la implementación de un servidor en la nube AWS, inicialmente será de recursos básicos y conforme siga creciendo la demanda ira incrementándose.

7.3.1.1. Características para la implementación

- **Equipo:** Servidor AWS
- **Tipo** : WS
- **Estado:** Nube
- **Sistema Operativo** : Ubuntu
- **Procesador** : Intel XEON 2.64 Hz
- **Memoria RAM:** 64 GB
- **Disco Duro:** 2 TB
- **Detalle de uso:** Destinado para ambientes de pruebas/producción internos y externos

Tipo de pago

- **Pago por uso:** El pago por uso le permite adaptarse con facilidad a las cambiantes necesidades de la empresa sin comprometerse a dedicar presupuestos excesivos y, además, mejorar la capacidad de respuesta ante los cambios.

7.3.2. Iniciativa: Reestructuración de entornos de desarrollo e implementación de entornos de integración continua y pre-producción

Para la implementación de esta iniciativa es necesario definir los factores que pueden influir a la hora de desarrollar un módulo, los procedimientos a manejarse de acuerdo con el

tamaño del proyecto, la complejidad del software, las tecnologías utilizadas para el desarrollo, la cantidad de desarrolladores implicados en el proyecto, la metodología de desarrollo elegida o la criticidad del software.

7.3.2.1. Entorno de desarrollo

Este entorno está localizado en el computador de cada programador, tomando en consideración que deberán contar con SmartGit para subir cambios y mantener un control del versionamiento, de esta forma varias personas pueden estar trabajando en un mismo proyecto a la vez sin inconvenientes.

Para minimizar incidencias en etapas posteriores este entorno dispondrá de software y configuración lo más parecida al entorno de producción.

Solo cuando el desarrollador completa un código perfectamente funcional e integrable en el entorno de pre-producción y tras realizar las pruebas necesarias para asegurar que el software desarrollado tiene la estabilidad suficiente se podrá pasar al entorno de integración continua.

7.3.2.2. Entorno de integración continua

Este entorno cumple un triple objetivo:

- Integrar el trabajo de los diferentes desarrolladores en un repositorio central, dando como resultado una versión del código actualizada y consolidada.
- Automatizar las pruebas de integración y su validación antes de ser movido al siguiente entorno.

- Enviar el código al siguiente entorno si las pruebas han sido superadas satisfactoriamente.

Para este propósito se implementará SmartGit

7.3.2.3. Pre-producción

Aquí se realizarán las pruebas de validación al conjunto del software, teniendo como objetivo localizar cualquier error antes de llegar al entorno de producción y evitar así los problemas derivados de ellos.

Este entorno lo utiliza el cliente para probar la aplicación e informar sobre los errores que ha encontrado o las características que le faltan a la aplicación.

De aquí extraeremos las impresiones del cliente y localizaremos de una forma temprana posibles carencias en los requisitos iniciales, en el diseño o en su implementación.

Se pueden dar varias situaciones:

- Que todo salga maravillosamente bien, improbable pero posible.
- Que una nueva actualización en el entorno esté o no relacionada con la funcionalidad que estamos probando, provoque errores y demos la validación por fallida.
- Que congelemos la actualización del entorno anterior incurriendo en molestias y dificultades extra para el equipo de desarrollo y retrasando las fechas del proyecto.

7.3.2.4. Entorno de pruebas

Uno de los aspectos más significativos de este circuito es que cada entorno debe tener su propio tipo validación y por lo tanto de pruebas.

Se realizarán pruebas unitarias en el entorno de desarrollo. Estas consisten en test que realiza cada desarrollador conforme avanza con el módulo.

Las pruebas en el entorno de preproducción son en relación con el conjunto de las aplicaciones generadas por uno de los individuos del equipo Scrum

Los test en el entorno de integración continua, serán validadas por SmartGit, ya que en este entorno se une los desarrollos conformados por los diferentes integrantes del equipo Scrum.

7.3.2.5. Entorno de producción

Este es el entorno en el que se ejecuta la aplicación que utilizan los usuarios finales.

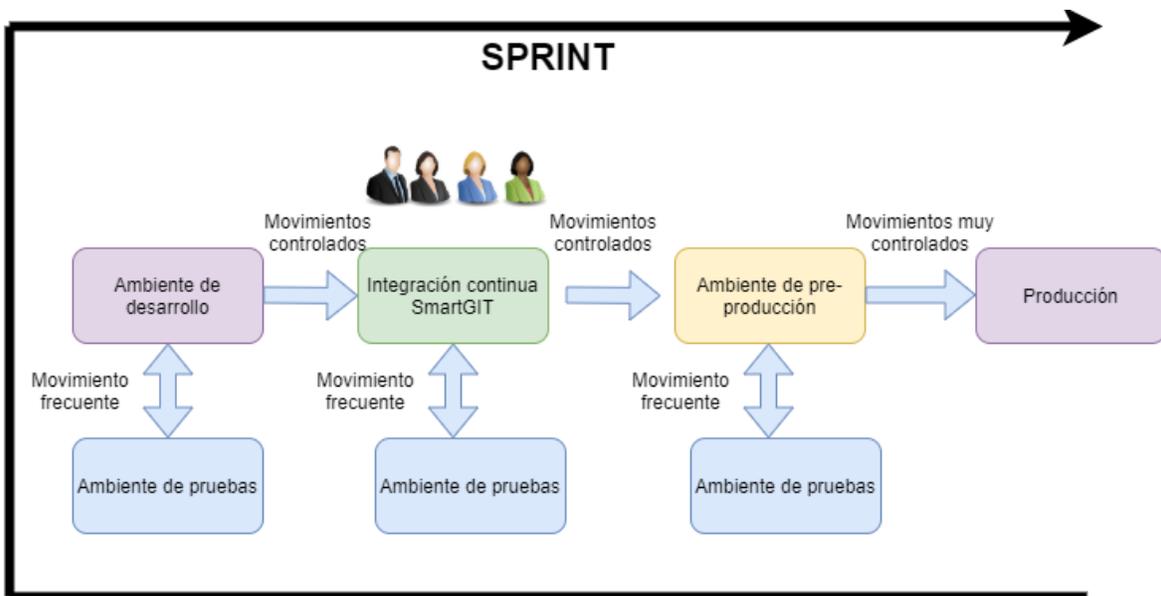


Figura 39 Entornos de desarrollo

Cabe recalcar que esta validación será iterativa en cada sprint, como se observa en la Figura anterior.

Tomando en consideración que en el ambiente de integración continua se unen todos los desarrollos que comprende el sprint, generados por cada uno de los individuos pertenecientes al equipo Scrum, donde la información viaja de manera frecuente entre los diferentes ambientes y pruebas, entre desarrollo, integración y preproducción, la información viaja de manera controlada y a producción muy controlada, ya que se pretende evitar los cambios en caliente.

7.3.2.6. Características de iniciativa

- **Tiempo de implementación:** 1 semana
- **Personal:** 3 desarrolladores, 1 Scrum Master, 1 *Product Owner*, Líder de proyecto
- **Precio aproximado:** Depende de la capacidad de los servidores
- **Precio total aproximado:** 1000\$

8. PLAN DE MIGRACIÓN

El plan de migración permite priorizar las iniciativas, de acuerdo con el impacto del proyecto en la empresa, el esfuerzo que representa su implementación y su necesidad inherente de implementarla.

8.1. Análisis de impacto

En la siguiente tabla se muestra las iniciativas, de acuerdo con las diferentes arquitecturas y sus objetivos

Se encuentra valorado de 0 a 2 puntos el impacto de la iniciativa, tomando una escala entre 0 a 0.7 se considera una iniciativa baja (Color naranja), de 0.7 a 1.4 una iniciativa media (Color amarillo) y de 1.4 a 2 una iniciativa alta (Color verde).

No	Arquitectura	Id	Proyecto	Habilitante	Objetivos							valoración cualitativa	Iniciativa en marcha	Impacto
					16%	12%	14%	18%	20%	10%	10%			
					Sostenibilidad de Servicios de TI.	Contribuir a la experiencia del cliente final	Eficiencia procesos, integración IT, gestión de conocimiento y efectividad laboral	Servicios de TI ágiles y formalizados con aporte a solución de problemas empresariales	Plataforma TI segura, resiliente e interoperable	Desarrollos estables y sin errores	Control de versionamiento			
1	Negocio	P1	Implementación de Scrum	SI								1.90	NO	Alto
2	Negocio	P2	Capacitación al personal del departamento de desarrollo informático en metodología Scrum	SI								1.38	NO	Medio
3	Aplicaciones / Datos	P3	Actualización a versión 14 de Odo	SI								1.80	NO	Alto
4	Aplicaciones / Datos	P4	Implementación del módulo Scrum en Olympo									1.74	NO	Alto
5	Aplicaciones / Datos	P5	Control de actividades procedentes de Scrum en ERP Olympo									1.32	NO	Medio
6	Aplicaciones / Datos	P6	Implementación de SmartGit para control de versionamiento									1.20	NO	Medio
7	Infraestructura base	P7	Implementación de Servidor AWS									0.50	NO	Bajo
8	Infraestructura base	P8	Reestructuración de entornos de desarrollo e implementación de entornos de integración continua y preproducción									1.70	NO	Alto

Escala de Impacto	
Bajo: entre 0 - 0,7	
Medio: entre 0,7 y 1,4	
Alto: entre 1,4 y 2	

Tabla 18 Impacto

Como conclusión del análisis según el impacto, se obtiene que las iniciativas deberán ser implementadas de acuerdo con su impacto en el siguiente orden:

- a. P1: Implementación de Scrum
- b. P3: Actualización a versión 14 de Odoon
- c. P4: Implementación del módulo Scrum en Olympo
- d. P8: Reestructuración de entornos de desarrollo e implementación de entornos de integración continua y pre-producción
- e. P2: Capacitación al personal del departamento de desarrollo informático en metodología Scrum
- f. P5: Control de actividades procedentes de Scrum en ERP Olympo
- g. P6: Implementación de SmartGit para control de versionamiento
- h. P7: Implementación de Servidor AWS

Análisis de esfuerzo

En la siguiente tabla se muestra las iniciativas vs el esfuerzo, en relación con la inversión económica, la complejidad de la iniciativa y la capacidad de departamento de desarrollo informático.

Se encuentra valorado de 0 a 3 puntos el esfuerzo de la iniciativa:

Tomando una escala de:

- 1 esfuerzo bajo (Circulo verde),
- 2 esfuerzo medio (Triangulo amarillo)
- 3 esfuerzo alto (Rombo rojo).

En relación con la inversión económica se considera, un esfuerzo bajo entre 1000\$ a 3000\$, esfuerzo medio de 3000\$ a 7000\$ y un esfuerzo alto de 7000\$ a 20000\$

No	Arquitectura	Id	Proyecto	Criterios Esfuerzo			Suma ponderada	Esfuerzo	
				40%	30%	30%			
				En Marcha	Recursos Económicos	Complejidad	Capacidad TI		
1	Negocio	P1	Implementación de Scrum	NO	2	3	3	2.60	Alto
2	Negocio	P2	Capacitación al personal del departamento de desarrollo informático en metodología Scrum	NO	1	2	3	1.90	Medio
3	Aplicaciones / Datos	P3	Actualización a versión 14 de Odoo	NO	3	3	1	2.40	Alto
4	Aplicaciones / Datos	P4	Implementación del módulo Scrum en Olympo	NO	2	3	2	2.30	Alto
5	Aplicaciones / Datos	P5	Control de actividades procedentes de Scrum en ERP Olympo	NO	1	1	1	1.00	Bajo
6	Aplicaciones / Datos	P6	Implementación de SmartGit para control de versionamiento	NO	1	1	1	1.00	Bajo
7	Infraestructura base	P7	Implementación de Servidor AWS	NO	1	2	1	1.30	Bajo
8	Infraestructura base	P8	Reestructuración de entornos de desarrollo e implementación de entornos de integración continua y pre-producción	NO	1	2	1	1.30	Bajo

Escala de Esfuerzo
Bajo: entre 1 - 1,7
Medio: entre 1,7 y 2,4
Alto: entre 2,4 y 3

Costos
Bajo: entre 1000\$ - 3000\$
Medio: entre 3000\$ y 7000\$
Alto: entre 7000\$ y 2000\$

Tabla 19 Esfuerzo

Como conclusión del análisis según el esfuerzo, se obtiene que las iniciativas deberán ser implementadas de acuerdo con su esfuerzo en el siguiente orden:

- a. P1: Implementación de Scrum
- b. P3: Actualización a versión 14 de Odoon
- c. P4: Implementación del módulo Scrum en Olympon
- d. P2: Capacitación al personal del departamento de desarrollo informático en metodología Scrum
- e. P8: Reestructuración de entornos de desarrollo e implementación de entornos de integración continua y pre-producción
- f. P7: Implementación de Servidor AWS
- g. P5: Control de actividades procedentes de Scrum en ERP Olympon
- h. P6: Implementación de SmartGit para control de versionamiento

8.2. Fases

Una vez analizado el impacto y el esfuerzo se implementará las iniciativas en el siguiente orden

No	Arquitectura	Id	Proyectos	Habilitante	Impacto	Esfuerzo	Prioridad	Fase
1	Negocio	P1	Implementación de Scrum	SI	Alto	Alto	SI	1
2	Negocio	P2	Capacitación al personal del departamento de desarrollo informático en metodología Scrum	SI	Medio	Medio	SI	4
3	Aplicaciones / Datos	P3	Actualización a versión 14 de Odoos	SI	Alto	Alto	SI	1
4	Aplicaciones / Datos	P4	Implementación del módulo Scrum en Olympos	0	Alto	Alto		2
5	Aplicaciones / Datos	P5	Control de actividades procedentes de Scrum en ERP Olympos	0	Medio	Bajo		6
6	Aplicaciones / Datos	P6	Implementación de SmartGit para control de versionamiento	0	Medio	Bajo		5
7	Infraestructura base	P7	Implementación de Servidor AWS	0	Bajo	Bajo		7
8	Infraestructura base	P8	Reestructuración de entornos de desarrollo e implementación de entornos de integración continua y pre-producción	0	Alto	Bajo		3

Tabla 20 Fases

Como conclusión del análisis se tiene que implementar las iniciativas de la siguiente manera:

- a. **Fase1:** P1 Implementación de Scrum
- b. **Fase1:** P3 Actualización a versión 14 de Odoos
- c. **Fase2:** P4 Implementación del módulo Scrum en Olympos
- d. **Fase3:** P8 Reestructuración de entornos de desarrollo e implementación de entornos de integración continua y pre-producción
- e. **Fase4:** P2 Capacitación al personal del departamento de desarrollo informático en metodología Scrum
- f. **Fase5:** P6 Implementación de SmartGit para control de versionamiento

- g. **Fase6:** P5 Control de actividades procedentes de Scrum en ERP Olympo
- h. **Fase7:** P7: Implementación de Servidor AWS

8.3. Road Map

En el ANEXO3 de detallan las actividades a realizar para la implementación del presente Capstone

Bibliografía

- Abellán, E. (s.f.). *Wearemarketing*. Obtenido de <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html>
- Ahierro. (s.f.). *ah*. Obtenido de <https://blog.ahierro.es/entornos-en-el-desarrollo-de-software/>
- Anda, N. (30 de agosto de 2018). *Entendiendo la Industria 4.0 y su contexto en México*. Obtenido de <https://www.factor.mx/portal/entendiendo-la-industria-4-0/>
- Andreas, K. (s.f.). Obtenido de web.archive.org/web/20181121191205/https://www.sciencedirect.com/science/article/pii/S0007681
- Andreas, S. (s.f.). Obtenido de www.researchgate.net/publication/307620022_A_Maturity_Model_for_Assessing_Industry_4_0_Readiness_and_Maturity_of_Manufacturing_Enterprises
- Arsys. (13 de Junio de 2018). *Arsys*. Obtenido de <https://www.arsys.es/blog/soluciones/postgresql-servidores/>
- Blancarte, O. (s.f.). Introducción a la Arquitectura de software. En I. a. la. (Business Motivation Model). *Business Motivation Model*.
- Cabrera, Y. (05 de Julio de 2020). *octopus*. Obtenido de <https://www.octopus.es/odoo-14-lista-de-novedades-y-cambios-para-este-2020/>
- Cabrera, Y. (06 de 12 de 2020). *Octopus*. Obtenido de <https://www.octopus.es/migracion-update-de-odoo-cuando-y-por-que-hacerla/>
- CALENDAMAIA. (10 de Enero de 2014). *Eclipse IDE*. Obtenido de <https://www.genbeta.com/desarrollo/eclipse-ide>

- Capgemini. (s.f.). Obtenido de <https://www.capgemini.com/pt-en/2018/09/industry-4-0-maturity-model-mirroring-today-to-sprint-into-the-future/#>
- Captación, R. (s.f.). *Caso de Éxito con Scrum: Amazon*. Obtenido de <https://www.redcapacitacion.tv/noticias/caso-de-exito-con-scrum-amazon/>
- CMMI . (s.f.). *Tutorialspoint.com* . Obtenido de https://www.tutorialspoint.com/es/cmmi/cmmi_maturity_levels.htm
- COBIT 2019. (2014). *E-commerce 2014*. Essex: Pearson. Obtenido de <https://www.isaca.org/E-Commerce>
- Departament de Ciència Enginyeria Nàutiques. (2004). *El futuro tecnológico de las Terminales Marítimas de Vehículos: La integración de sus sistemas de información*. Barcelona, España: UPC.
- Desarrolloweb.com. (03 de Junio de 2014). *Introducción a Git*. Obtenido de <https://desarrolloweb.com/articulos/introduccion-git-github.html>
- Desdelinux. (s.f.). *Darkcrist* . Obtenido de <https://blog.desdelinux.net/dbever-una-excelente-herramienta-para-la-gestion-de-diferentes-db/>
- Drapeau, D. (s.f.). *Serpent Consulting*. Obtenido de https://apps.odoo.com/apps/modules/14.0/project_scrum_agile/
- Espino. (s.f.). Obtenido de <http://www.v-espino.com/~chema/daw1/tutoriales/postgres/pgadmin1.htm>
- Hat, R. (s.f.). *Red Hat*. Obtenido de <https://www.redhat.com/es/devops/what-is-agile-methodology>
- Herranz, R. (2016). *Despegar con Scrum*. Madrid España: Lulu.
- IONOS. (11 de Marzo de 2019). *IONOS*. Obtenido de <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>
- Jetbrains. (s.f.). Obtenido de <https://www.jetbrains.com/es-es/pycharm/features/>
- Joyanes, L. (2017). *La cuarta revolución industrial. Industria 4.0* . Bogotá: : Alfaomega Colombiana S.A. Obtenido de Joyanes Aguilar, L. (2017). *Industria 4.0 LA CUARTA REVOLUCIÓN INDUSTRIAL*. Bogotá: Alfaomega Colombiana S.A.
- Mauro Acebo Plaza, i. d. (2017). *ESTUDIOS INDUSTRIALES ORIENTACIÓN ESTRATÉGICA PARA LA TOMA DE DECISIONES* . Jonathan Quijano - UGA Diseño y Publicidad ESPOL.
- Mazo, R. (2018). *Guía para la adopción industrial de líneas de productos de software*. Medellín, Colombia: Universidad EAFIT.
- Nader, K. &. (2019). *Los Fundamentos de Agile Scrum*. Van Haren.
- odoo. (s.f.). *odoo*. Obtenido de https://www.odoo.com/es_ES/?utm_source=google&utm_medium=cpc&utm_campaign=Branding-Spanish_Brand-SP-

Odo&utm_term=odoo&utm_matchtype=e&utm_device=c&utm_gclid=Cj0KCQiAj9iBBhCJARIsAE9qRtAN9Tkug7K7prDdt2z3ZI124pz7gEag5LFRiXvQ8YldqaCHUbh7wJlaAh2VEALw_wcB&gclid=

org, P. á. (s.f.). *Proyectos ágiles org*. Obtenido de <https://proyectosagiles.org/que-es-scrum/>

Ortizvivas. (10 de 07 de 2017). *Ortizvivas*. Obtenido de <https://ortizvivas.com/blog/ambientes-de-trabajo-para-software/>

Proyectosagiles.org. (s.f.). *Equipo de desarrollo (Development Team)*. Obtenido de <https://proyectosagiles.org/equipo-team/>

Python, E. d. (s.f.). *Escuela de Python*. Obtenido de <https://www.esuelapython.com/pycharm-uno-de-los-mejores-ide-para-python/>

Salazar, A. (16 de Octubre de 2016). *Procesos de SCRUM*. Obtenido de <http://www.prozessgroup.com/procesos-de-scrum/#:~:text=Los%20procesos%20de%20Scrum%20corresponden,dentro%20de%20un%20proyecto%20Scrum.&text=En%20cada%20proceso%2C%20algunas%20entradas,de%20la%20naturaleza%20del%20proyecto.>

Scrum, L. G. (2020). *ScrumGuides*. Obtenido de <https://scrumguides.org/scrum-guide.html>

scrum.org. (2021). Obtenido de <https://www.scrum.org/courses/applying-professional-scrum-training>

SEVILLA, D. D. (s.f.). *PROBLEMÁTICA DE LOS PROYECTOS SOFTWARE*.

SmartGit. (2021). *Syntevo*. Obtenido de <https://www.syntevo.com/smartgit/purchase/?billing-interval=monthly&distributed-reviews-addon=0>

Soft, A. (s.f.). *El Modelo CMMI*. Monterrey.

software, C. (11 de 12 de 2020). *Juan Oliveros*. Obtenido de <https://blog.comparasoftware.com/fases-metodologia-scrum/>

TOGAF® Versión 9.1 . (2013). *Guía de Bolsillo*. Van Haren.

Tutorialpoint. (2020). Obtenido de www.tutorialspoint.com/es/cmmi/cmmi_maturity_levels.htm

Valderrama, J. (1999). *Información Tecnológica. CIT, 400*.