

Administración de proyectos

Repasso

Objetivos de la administración de proyectos

- Entregar el software en el tiempo acordado
- Mantener el presupuesto
- Cumplir con las funcionalidades solicitadas
- Mantener un equipo de desarrollo óptimo



Actividades del administrador de proyectos

1. Planeación del proyecto
2. Informes
3. Gestión de riesgo
4. Gestión de personal
5. Creación de propuestas





Retos de los proyectos de software

- El producto del trabajo suele ser intangible
- Proyectos grandes pueden ser inmanejables
 - Tecnologías cambiantes
 - Mucha experiencia en lógica de negocio
- Los proyectos son sumamente cambiantes
- Los proyectos se adaptan a la organización
- Los proyectos suelen ser innovadores para una organización.



Gestión de
Personal

“

*El personal es el activo más
importante de una empresa*



Administrador de proyecto y la gestión del personal

- La organización debe obtener el mejor rendimiento posible de su inversión en personal.
- Se debe estar al tanto de problemas potenciales en el personal.



Conocimiento vs. Habilidades humanas

- Los ingenieros de software tienen el conocimiento técnico para saber qué se necesita.
- Los ingenieros de software tienen la capacidad de entrenar al personal.
- Los ingenieros de software con frecuencia carecen de habilidades para motivar a un equipo.
- Los ingenieros de software con frecuencia carecen de habilidades para dirigir.
- Los ingenieros de software muchas veces no pueden delegar funciones.
- Los ingenieros de software no siempre pueden comunicar problemas de manera asertiva.

Responsabilidades del administrador de proyectos de software:

- La carga de trabajo va acorde a las habilidades y experiencia del empleado
- Respeto a los trabajadores y sus intereses
- Se mantiene motivado al empleado
 - Reconocimiento de su esfuerzo
 - Se le dan oportunidades de crecimiento
 - Se le dan oportunidades de desarrollo personal



Factores críticos

- Consistencia
- Respeto
- Inclusión
- Honestidad





Motivación del personal

Un colaborador motivado contribuirá con lo mejor de sus habilidades.

Trabajadores motivados!

- Motivación significa organizar el trabajo y el ambiente laboral para alentar a los individuos a desempeñarse tan efectivamente como sea posible.
- Si las personas no están motivadas, no estarán interesadas en la actividad que realizan.



Necesidades de autorrealización para la motivación

Necesidades de autorrealización
Necesidades de estima
Necesidades sociales
Necesidades de seguridad
Necesidades fisiológicas



Maslow 1954.



Identificar personalidades del equipo

- Personas orientada a tareas
- Personas orientada hacia sí misma
- Personas orientadas a la interacción



Modelo de
madurez de
gestión de personal

Modelo de Madurez de Gestión de Personal

- “Aumentar la rapidez con la cual las organizaciones de software acometen las aplicaciones cada vez más complejas al ayudar a atraer, aumentar, motivar, desplegar y retener el talento necesario para mejorar su capacidad de desarrollo de software”

CUR94

URL: <https://www.redalyc.org/pdf/4259/425942453008.pdf>

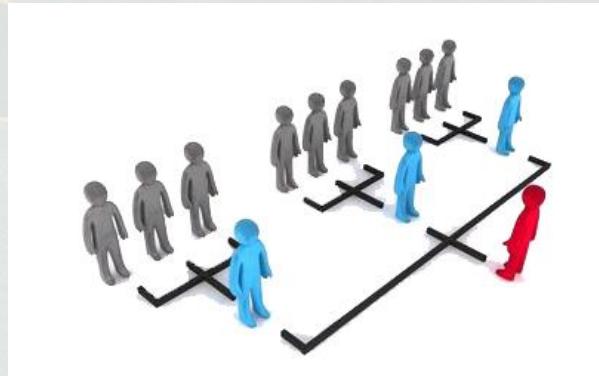
Modelo de Madurez de Gestión de Personal

- La necesidad de contar con personal para el desarrollo del software altamente preparado y motivado se viene discutiendo desde los años 60
- El MMCGP es compañero del modelo de madurez de la capacidad software, que guía a las organizaciones en la creación de un proceso de software maduro.

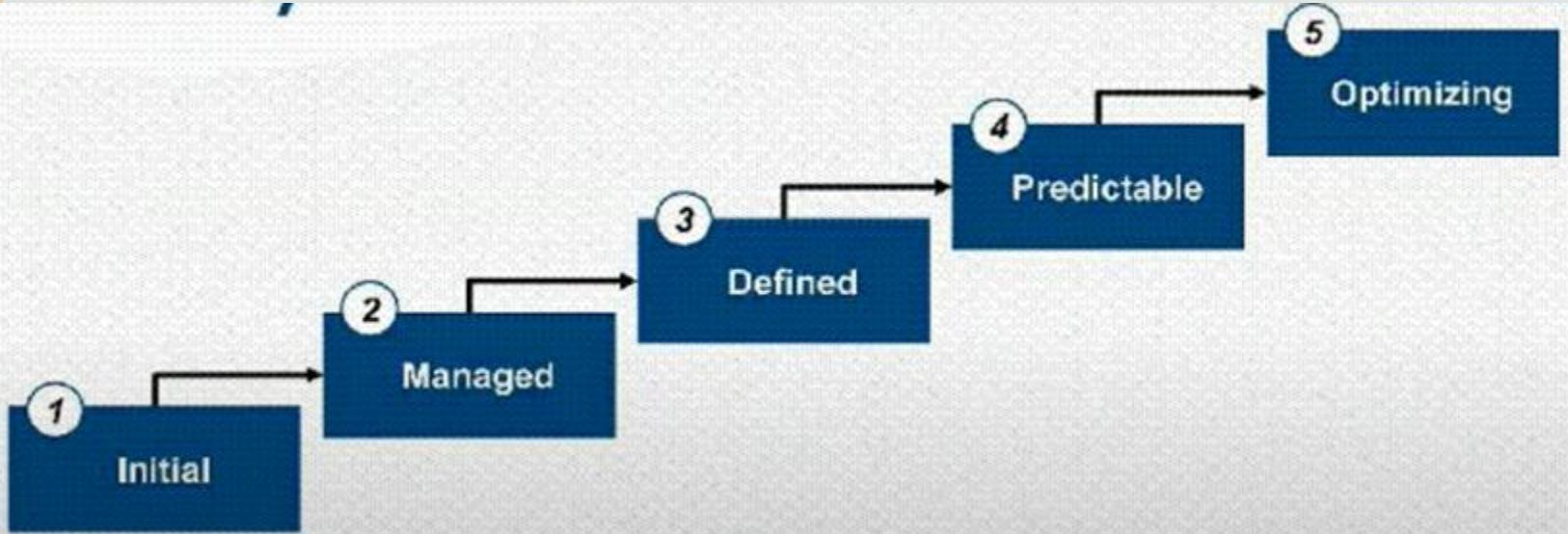


Áreas Clave

- Reclutamiento
- Selección
- Gestión de rendimiento
- Entrenamiento
- Retribución
- Desarrollo de carrera
- Diseño de organización
- Desarrollo de la cultura de equipo



Modelo de Madurez de Gestión de Personal



Reclutamiento y Selección

- Buscar el equilibrio de habilidades técnicas y personalidades.
- Es importante que todos puedan trabajar en conjunto.
- Es importante que un nuevo miembro esté dispuesto a aprender.
- Intentar ver más allá.

Proceso de selección

- Evaluar parte técnica.
- Evaluar personalidad.
- Evaluar compatibilidad en el equipo.
- Evaluar habilidades blandas.



Entrenamiento

- Cómo funciona el negocio?
- Procesos internos de la empresa
- Procesos tecnológicos
- Buenas prácticas
- Nuevas tecnologías
- Nuevas atribuciones



Gestión de Rendimiento

- Rae: Producto o utilidad que rinde o da alguien o algo.
- Conocer la eficiencia de producción por factor y recurso empleado brindará información útil sobre la gestión de recursos en la empresa.
- Necesidad de acciones correctivas
- Reconocer el buen trabajo



Factores que impactan el rendimiento del personal

- Factores técnicos
- Factores de la organización
- Factores Productivos
- Factores Personales
- Factores de Liderazgo



Calculando el rendimiento

- Evaluación de desempeño
 - Habilidades
 - Cualidades
 - Necesidades en el equipo

Intentar hacer objetivo algo que es muy subjetivo



A Definir

- Criterios de medición = punto de comparación
- Indicadores = aspectos/acciones de productividad a medir
- Establecer un sistema de ponderación
- Definir período de tiempo

Métodos para medir el rendimiento

- Rendimiento laboral = (Productos o Servicios Producidos) / (Recursos Utilizados)

A collage of mathematical diagrams and formulas:

- Top left: A right triangle with vertices A and B on a coordinate system. The hypotenuse is labeled $y = a(x-b)2+c$. The area is $SA = \frac{4}{3}\pi r^2$.
- Top center: A circle with radius r and angle θ . The formula $V = \frac{4}{3}\pi r^3$ is shown.
- Top right: A right-angled triangle with legs a and b , hypotenuse c , and angles $d=360^\circ$ and $e=180^\circ$. The formula $(\frac{a}{b})^e = \frac{a^e}{b^e}$ is given.
- Middle left: A diagram of a cone with radius r and height k . The formula $2x^2 + 3x + 4 = y$ is shown.
- Middle center: A logarithmic plot with the equation $\log_a 1 = 0$.
- Middle right: A diagram of a cylinder with radius r and height k . The formula $\sqrt{100 + c^2} = \pm 10 = c$ is given.
- Bottom left: A diagram of a rectangular prism with dimensions l, w, h . The formula $SA = 2lw + 2lh + 2wh$ is shown.
- Bottom center: A diagram of a unit circle with radius r and angle θ . The formula $\sin^2 y + \cos^2 y = 1$ is given.
- Bottom right: A diagram of a right-angled triangle ABC with hypotenuse c . The Pythagorean theorem $c^2 = a^2 + b^2$ is shown.

Other formulas and calculations include:
- $AB = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
- $4^{\frac{3}{2}} = 2\sqrt{4}$
- $\log_e(\frac{a}{b}) = \log_e a - \log_e b$
- $\sum_{k=0}^n k = \frac{1}{2}n(n+1)$
- $\pi \approx 3.14$
- $\sin 30^\circ = \frac{1}{2}$
- $\sin 45^\circ = \frac{1}{\sqrt{2}}$
- $\sin 60^\circ = \frac{\sqrt{3}}{2}$
- $a^b \cdot a^c = a^{b+c}$
- $(a-b-c)^2 = a^2 + b^2 + c^2 - 2ab - 2bc - 2ca$
- $\sin x = a(-x) + b = -(ax-b)$
- $y = \sin x$
- $y = ax^2 + bx + c$
- $A = \frac{1}{2}ar + \frac{1}{2}br + \frac{1}{2}cr$
- $\tan 60^\circ = \sqrt{3}$
- $C = 2\pi r$
- $\tan 30^\circ = \frac{1}{\sqrt{3}}$
- $a^3 + b^3 = (a+b)(a^2 - ab + b^2)$
- $(8^3)^3 = 8^{3 \times 3} = 8^9$
- $\left(\frac{2}{3}\right)^3 = \left(\frac{3}{2}\right)^{-3} = \frac{1}{27}$
- $A = sr$
- $r = \frac{A}{s}$
- $\tan 30^\circ = \frac{1}{\sqrt{3}}$

Ejemplo

- Supongamos que la compañía fabrica 40 botellas de whisky cada día, con un valor de 80 euros por cada una. El empleado en cuestión labora 4 horas al día, percibiendo una tarifa de 15 euros por hora, y embotella 4 botellas diarias.
- La fórmula para evaluar el rendimiento laboral del empleado es la siguiente: Rendimiento laboral del empleado = (Cantidad de productos que el empleado produce x Precio unitario del producto) / (Horas trabajadas diariamente x Tarifa horaria).

Rendimiento vs Productividad

- **Rendimiento:** obtener resultados deseables para una compañía
- **Productividad:** acciones que se llevan a cabo para obtener resultados

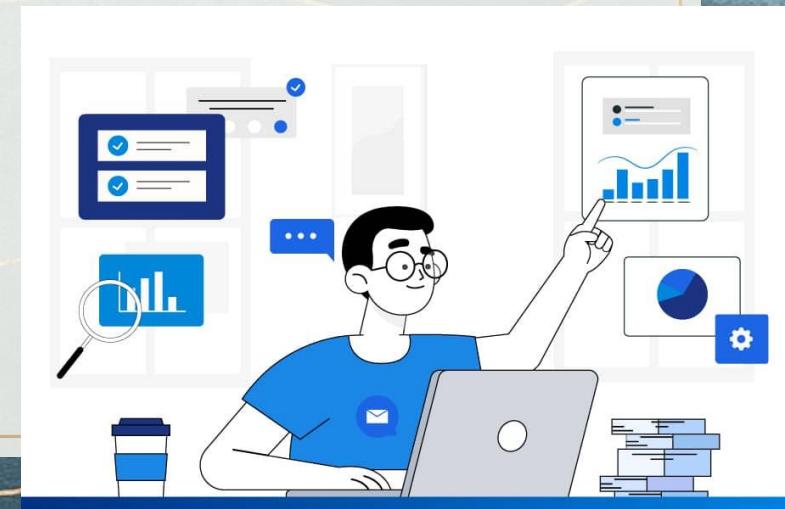
Para el software...

- Se requiere algo de intuición y otros factores
- Habilidades blandas
- Trabajo en equipo
- Capacidad de adaptarse
- Capacidades técnicas
- Lenguaje de programación específico
- Creatividad



Algunos métodos

- Método de la puntuación
- Descripción
- Reunión
- Autoevaluación
- Evaluación del compañero
- Evaluación 360
- Retroalimentación del cliente
- Encuestas
- Híbrido



Otro Ejemplo

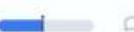
	Media esperada	Equipo de diseño	Carmen Torres	Consuelo Nava...	David Rodríguez	Jimena Menén...	Laura Ruiz
Soft Skills	4.6	3.4	3.2	4.6	2.9	2.8	2.9
Trabajo en equipo	4.1	4.0	2.4	4.7	2.8	4.5	2.9
Capacidad de adaptación	3.2	0.0	2.1	4.3	1.9	4.9	3.1
Technical Skills	3.5	4.7	4.8	4.5	1.0	3.2	4.1

Otro ejemplo

Evaluado
Maria Del Pilar Quintanilla Pérez
Marketing - Técnico - Granada - Director de Marketing.

Evaluador principal
Antonio Flores Quevado:
Peso: 100%
Nota final: 75 %
Finalizado

Valoración final
Balance (Gap)
+33% 
Ha obtenido un 29% de lo que se podía recibir

Competencias	Tipo	Requerido	Obtenido	Balance
Automotivación	Competencias Generales	Nivel 2 - 3	3 - Óptimo	
Trabajo en equipo	Competencias Generales	Nivel 3 - 3	2 - Regular	
Calidad	Competencias Generales	Nivel 2 - 4	3 - Alto	
Adaptación al cambio	Competencias Generales	Nivel 3 - 3	3 - Óptimo	
Lealtad	Valores	Nivel 3 - 4	3 - Se podría mejorar	
Integridad	Valores	Nivel 3 - 4	4 - Óptimo	
Compromiso con la empresa	Valores	Nivel 3 - 3	2 - Regular	
Comunicación fluida	Empresa y cliente	Nivel 2 - 3	3 - Óptimo	
Orientación al cliente	Empresa y cliente	Nivel 2 - 3	3 - Óptimo	

Acciones Correctivas

- No todo está perdido
- Entrenamientos de ser necesarios
- Llamada de atención
- Acompañamiento sobre algún tema en específico
- Ajustes en la actividad o proyecto
- Crear conciencia



Retribución

- Bono de productividad
- Aumento
- Promoción
- Libertades dentro del equipo
- Convenio entre la empresa y el empleado

Desarrollo de Carrera

- Los miembros del equipo se vuelven más y más expertos.
- La tecnología es cambiante
- Es más y más fácil adquirir conocimiento
- Las generaciones actuales buscan superarse
- Es un tipo de retribución
- Es conveniente para la empresa



Diseño de organización

- Deben Existir procesos bien definidos
- Los miembros del equipo deben contar con el equipo necesario para hacer su trabajo
- Los miembros del equipo deben contar con información necesaria para realizar su trabajo
- Deben tener claras las asignaciones laborales





Trabajo en Equipo

Trabajo en Equipo

- Un grupo demasiado grande no puede trabajar en equipo.
- **Trabajar en grupo:** los miembros son independientes entre sí y tienen responsabilidades individuales.
- **Trabajo en equipo:** los miembros colaboran en conjunto con sinergia para resolver problemas.

Trabajo en equipo

- En general, un equipo no debería exceder las 10 personas.
- Deberían estudiar el proyecto de forma grupal.
- Todos deberían reunirse y tener lluvia de ideas.
- Todos deberían aportar en la estimación de tiempos.
- Todos deberían participar en la planificación.
- Si surge un problema, todos deberían ser parte de la solución.
- Todos deberían compartir el conocimiento adquirido

Desarrollo de la cultura de equipo

- Se deben promover actividades de integración de equipos.
- Se debe promover que los miembros del equipo socialicen para que se lleven bien.
- Se debe reconocer el esfuerzo en conjunto.
- Se debe facilitar la comunicación entre los miembros del equipo.
- Reconocer la importancia de cada rol en el equipo.



Importancia de la cultura de equipo

- Confianza entre los miembros del equipo
- Distribuir adecuadamente las habilidades de cada miembro para aplicarlas a un problema
- El conocimiento debe compartirse





Liderazgo

Líderes de equipo

- Motivación: Producir a la mayor capacidad posible
- Organización: adecuar procesos o inventar nuevos
- Ideas o Innovación: habilidad para alentar a las personas a crear aún cuando deben trabajar bajo límites establecidos.

“

*Un líder es aquel que sabe a
dónde quiere ir, se levanta y
va...*

John Erskine

Gestor de proyectos – Rasgos Clave

- Resolución de problemas
- Dotes de Gestión
- Incentivar
- Influencia y fomento de la cultura de equipo



Paradigmas organizacionales

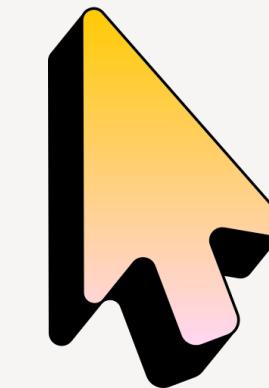
- **Paradigma Cerrado:** jerarquía tradicional de autoridad, trabajan mejor cuando producen software similar al anterior.
- **Paradigma Aleatorio:** depende de la iniciativa individual de los miembros del equipo.
- **Paradigma Abierto:** involucra mucha innovación a la hora de resolver problemas. Es un trabajo sumamente colaborativo.
- **Paradigma sincrónico:** se comparte un problema y se organiza a los miembros para trabajar en partes del problema con poca comunicación entre ellos.

Integración y Entrega continua

AGENDA

Desarrollo ágil

Reutilización del software



Entregas continuas

Integración continua

Desarrollo ágil



Desarrollo ágil

- Entorno Cambiante
- Agilizar el proceso de desarrollo



Métodos ágiles

- Desarrollo Incremental
- Normalmente al final de una iteración hay una liberación en un entorno productivo.



CI - Continuous Integration

Integración de código

Integración de Cambios

Divide y vencerás

Verificación de cambios

Trabajo en Equipo

Colaboración sin prepotencia

Estrategia del cambio

Trabajo a distancia



CD - Continuous Delivery

Dividir Entregas en algo más pequeño

El proceso de instalación se realiza 1 vez

Enfoque en tareas específicas

Ahorro de tiempo

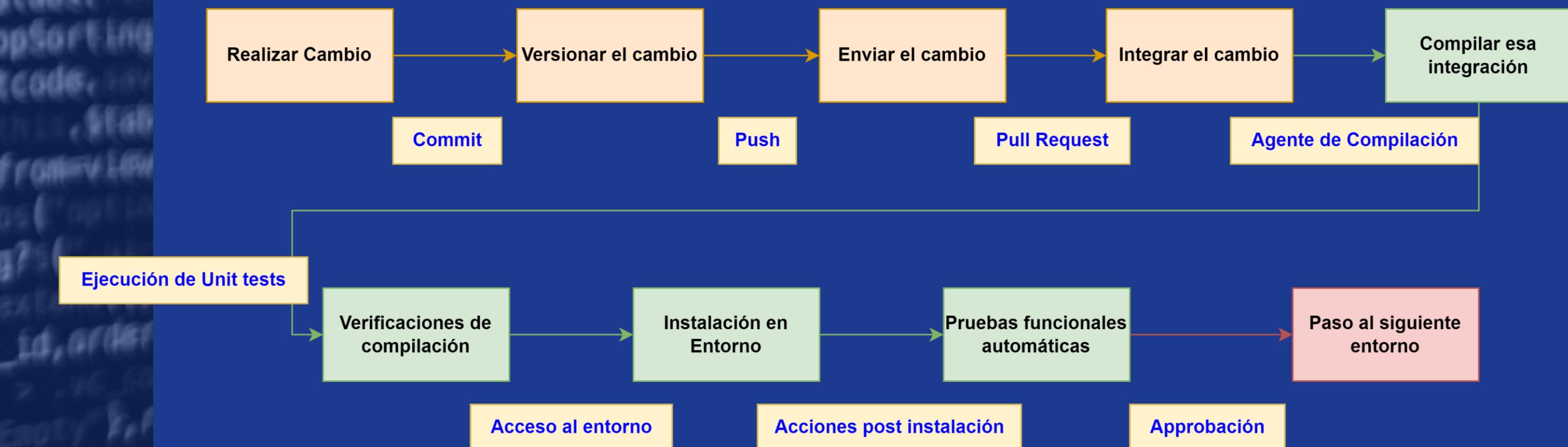
Verificaciones automáticas

Trabajo a Distancia

Agrega valor al cliente



Tareas en el proceso:



Otras Tareas / Políticas

Compilar código enviado

Code Coverage

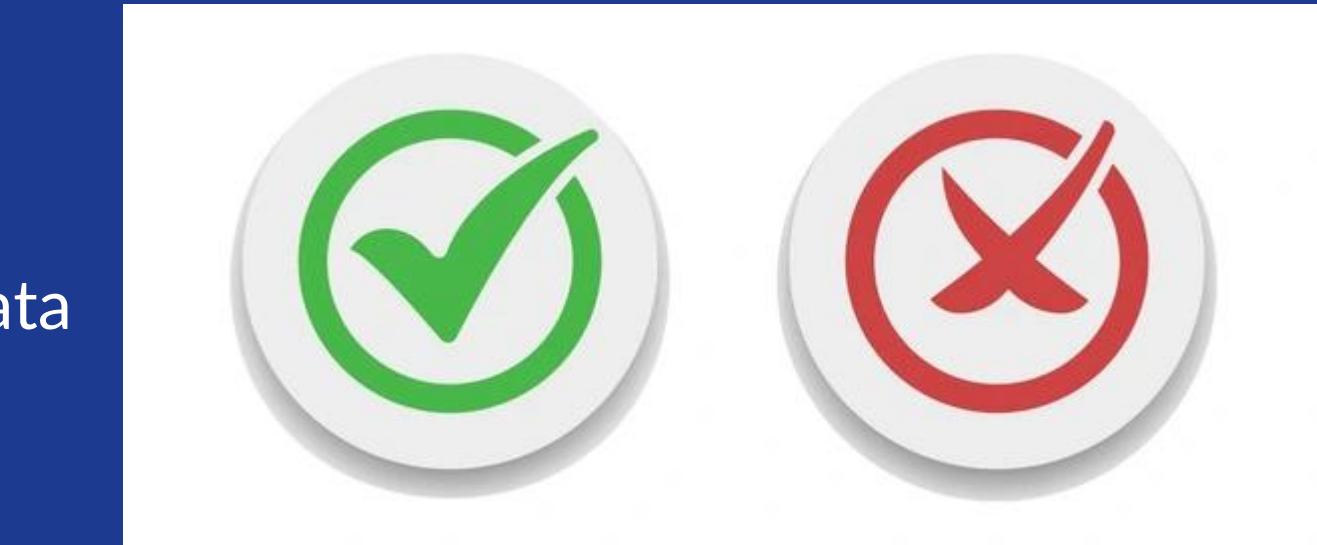
Publicar actualización utilizando un servicio interno ya en un entorno

Aprobación de pruebas unitarias

Pruebas funcionales automáticas

Ejecución de scripts de actualización de data

Backups antes de instalar



Beneficios del proceso definido:

- Reducción del riesgo
- Roles bien definidos
- Rollback en caso de emergencia
- Ahorro de tiempo
- Evitar “Re-Work”
- Trabajo en paralelo

Reparaciones en el camino

Tomar la versión correcta

Realizar los cambios o reparaciones pertinentes

Integrar los cambios

Compilar

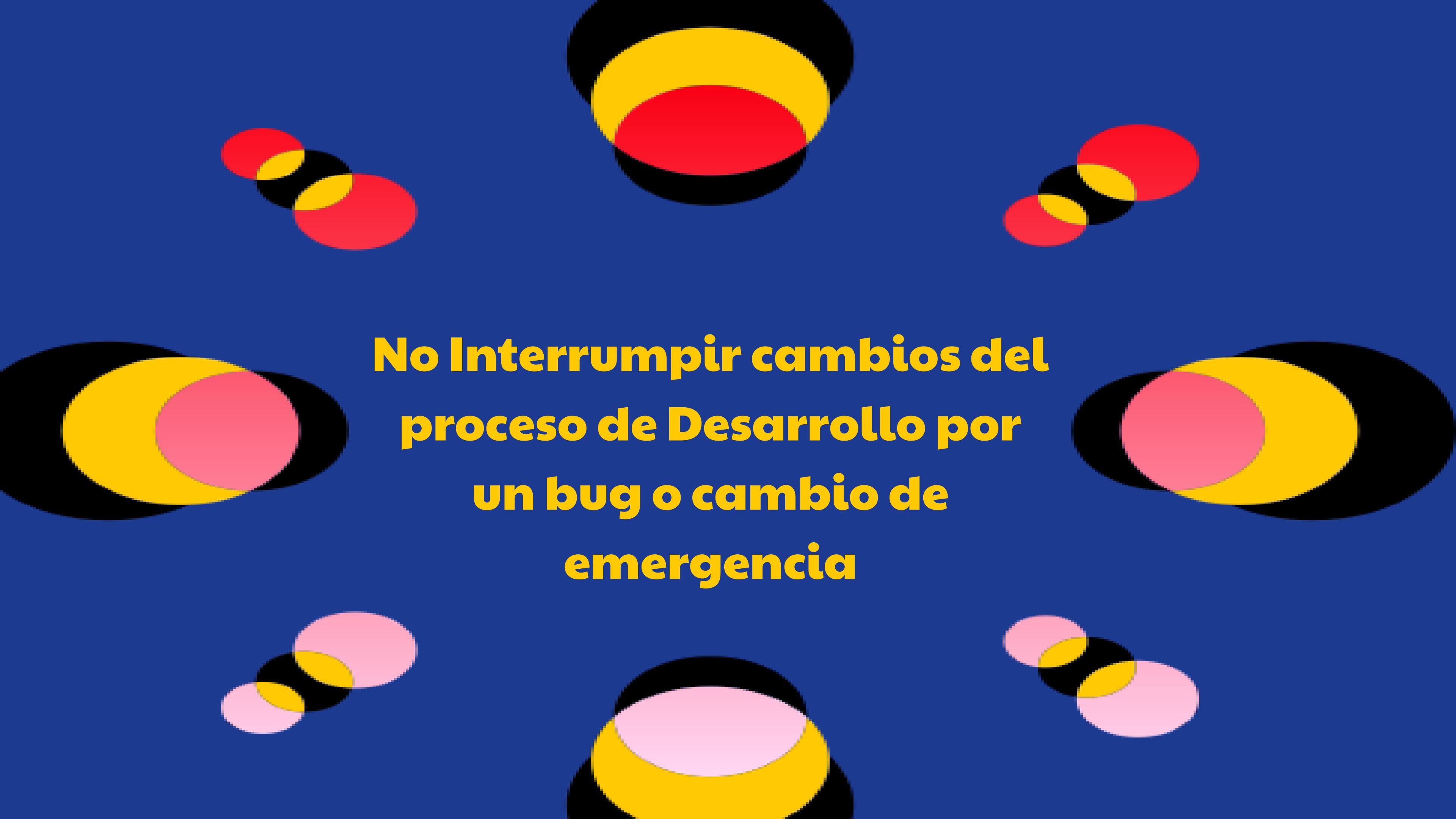
Instalar en entorno de pruebas

Verificar

Enviar al entorno productivo o de operación

Pruebas de Seguridad





**No Interrumpir cambios del
proceso de Desarrollo por
un bug o cambio de
emergencia**

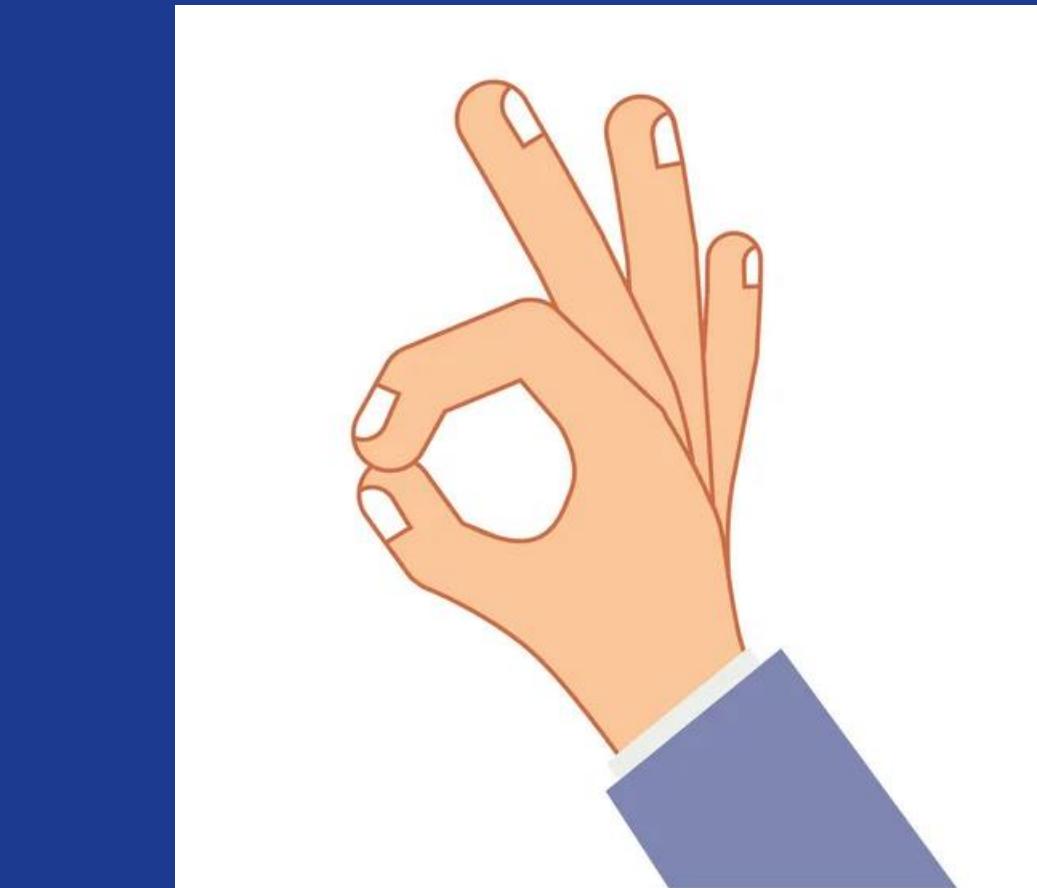
Reducción de riesgo

Verificar no dejar corrupto el proyecto

Establecer estándares mínimos de calidad

Establecer estándares mínimos de seguridad

No depender de acciones manuales



Partes del proceso



Code Management

Integración de código

Buenas prácticas:

- Tomar siempre la última versión antes de hacer un cambio.
- Hacer un merge local antes de subir cambios.
- Crear un Pull Request por Cambio
- Integrar cada commit a una tarea asignada o ticket
- Designar a alguien para la revisión de código

Opciones de manejo de versiones

Solo código y dependencias

Solo código

Código y Ejecutables



Agente de Compilación

Es un recurso de infraestructura que contiene un software especializado que puede realizar la compilación del código en un lenguaje en particular.

El agente se encarga de descargar el código de la rama solicitada y compilarla o interpretarla con la versión que se haya configurado.

El agente de compilación puede también encargarse de descargar dependencias, tanto de repositorios públicos como privados si así se configura.

El agente de compilación debe poder compilar múltiples proyectos de una misma solución.

Plugins y otros agentes

Para realizar otras tareas necesitamos otros tipos de agentes.

No necesariamente son agentes sino plugins sobre la herramienta que se esté utilizando.

Muchas tareas de estos plugins o agentes adicionales pueden realizarse en una máquina o servidor virtual dedicado a esto.

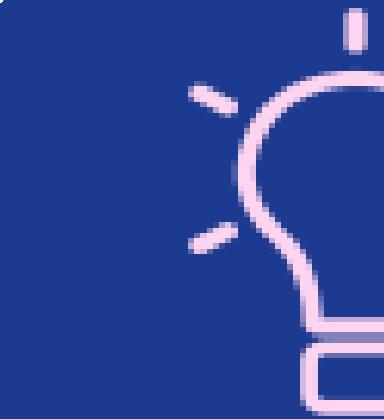
sonarQube



SMARTBEAR
ReadyAPI

Pero ... ¿A qué llamamos agente?

Es una máquina, con sistema operativo o una versión optimizada del mismo, que puede ser tanto física como virtual y contiene todo el software necesario para compilar una solución.



JVM, .NetFramework, GIT, Nuget, Maven, Programa para ejecutar pruebas, scanner de seguridad, etc.



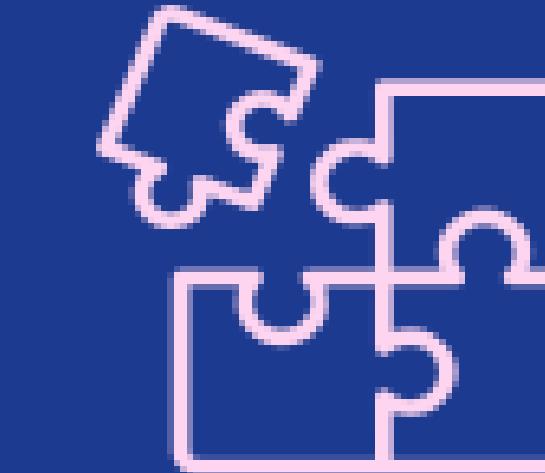
Reutilización de código

Reutilización

En la ingeniería de Software la reutilización es una estrategia en la que se engrana el proceso de desarrollo para reutilizar software existente.

Fue una respuesta a las demandas de reducir costos de producción y mantenimiento

Software más rápido de desarrollar

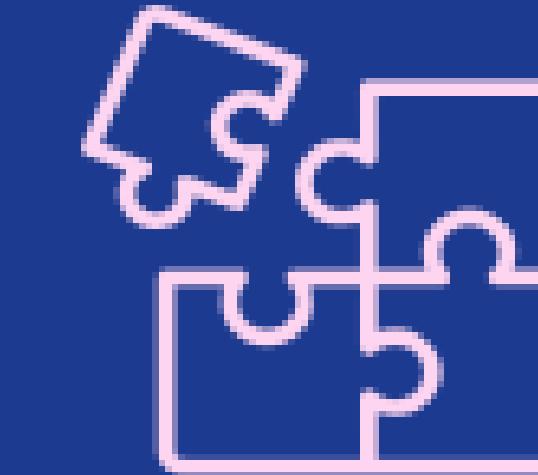


Ejemplos de Reutilización

Reutilizar todo un sistema de aplicación

Reutilización de componentes

Reutilización de objetos y funciones



Ventajas de la reutilización

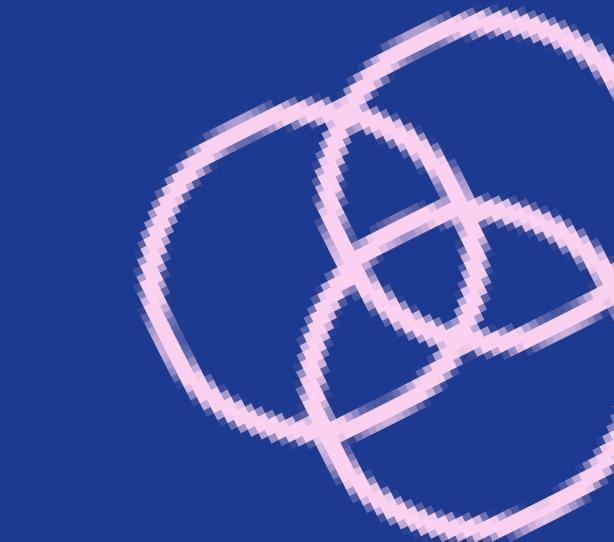
Confiabilidad

Reduce el riesgo de proceso

Uso efectivo de especialistas

Cumplimiento de estándares

Desarrollo Acelerado



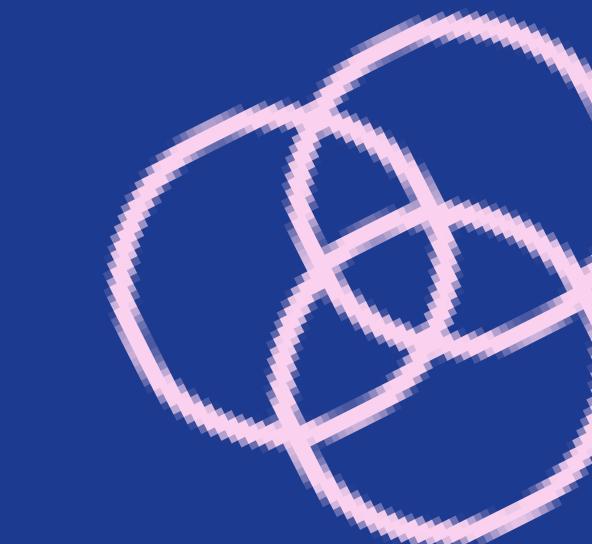
Desventajas de la reutilización

Es posible que el código fuente no esté disponible

Falta de apoyo

“No se inventó aquí”

Adaptación y mantenimiento de componentes



A tomar en cuenta en la reutilización

Calendario de Desarrollo

Vida esperada del software

Antecedentes, habilidades y experiencia del equipo de desarrollo

La criticidad del software y sus requerimientos no funcionales

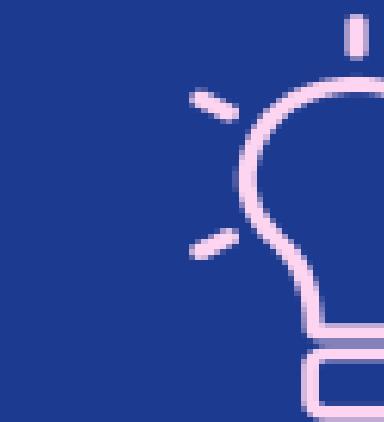
Dominio de la aplicación y tecnologías

Licenciamiento

Frameworks de aplicación

Es una estructura genérica que se extiende para crear una aplicación o subsistema más específico.

Un conjunto integrado de artefactos de software que colaboran en facilitar una arquitectura de reutilización para una familia de aplicaciones relacionadas.



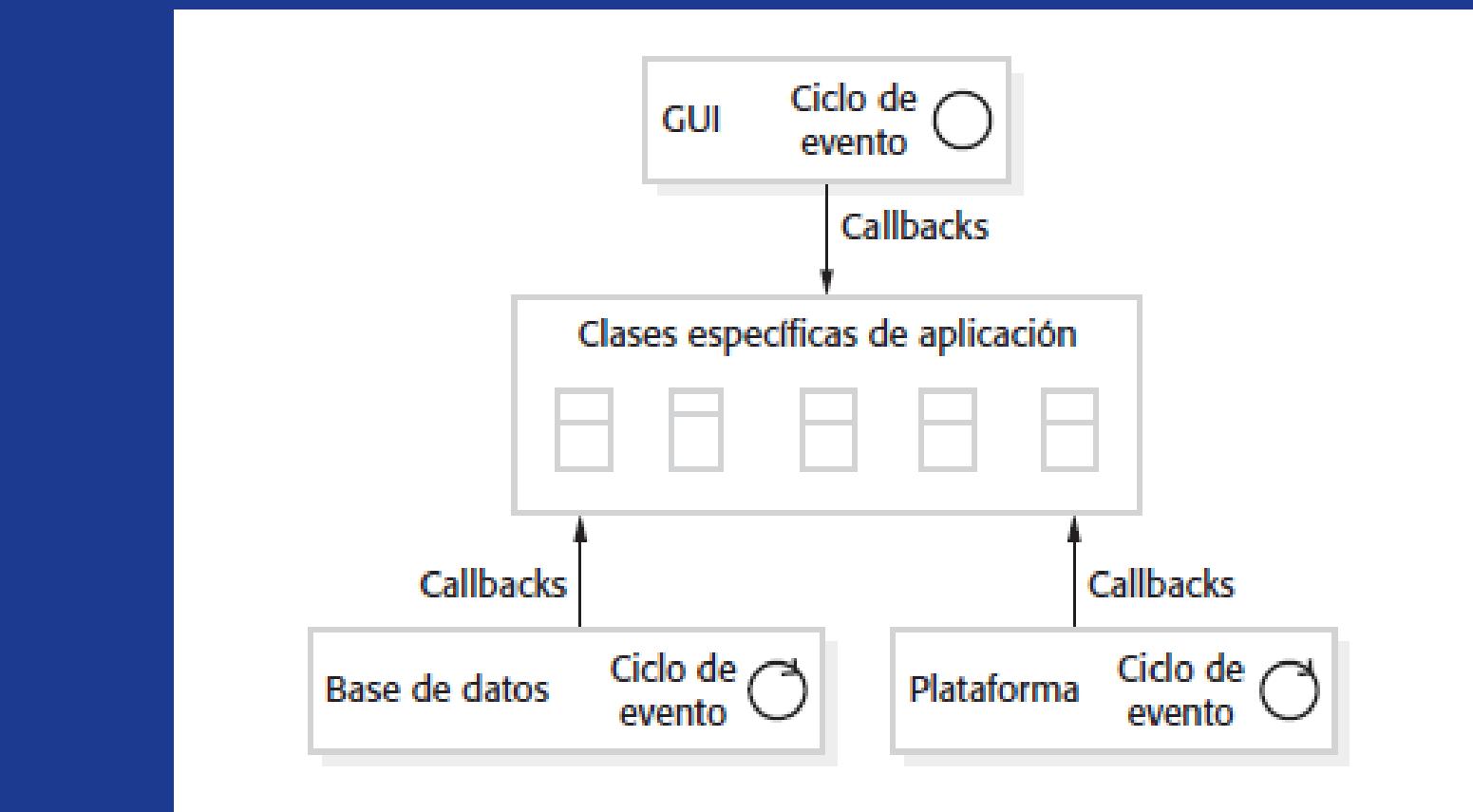
Frameworks de aplicación web WAF

Construcción de sitios dinámicos

Surgen en 1980

Fácil mantenimiento

Buenas prácticas



Frameworks de Software

Se apoyan en características orientadas a objetos o también en interfaces de software, utilizan conceptos como herencia y polimorfismo.

Se enfocan en brindar apoyo técnico antes que dominio específico.

Línea de productos de Software

Es un conjunto de aplicaciones con una arquitectura en común y componentes compartidos.

Esto puede implicar la configuración de algunos componentes, o implementar componentes adicionales para modificar el producto de acuerdo con los requerimientos.

Línea de productos de Software

Una línea de software podría dar soporte a una interfaz de hardware.

Las líneas de productos de hardware constituyen una familia de aplicaciones relacionadas, propiedad de la misma organización.

Especializaciones

Especialización de plataforma

Especialización de entorno

Especialización funcional

Especialización de proceso



Sistemas de Gestión



Sistemas de Gestión o COTS

Sistemas genéricos

Se especializan en un tipo de empresa o actividad empresarial

Son específicos de dominio

Incorporan suposiciones internas y buenas prácticas

Sistemas de Gestión o COTS

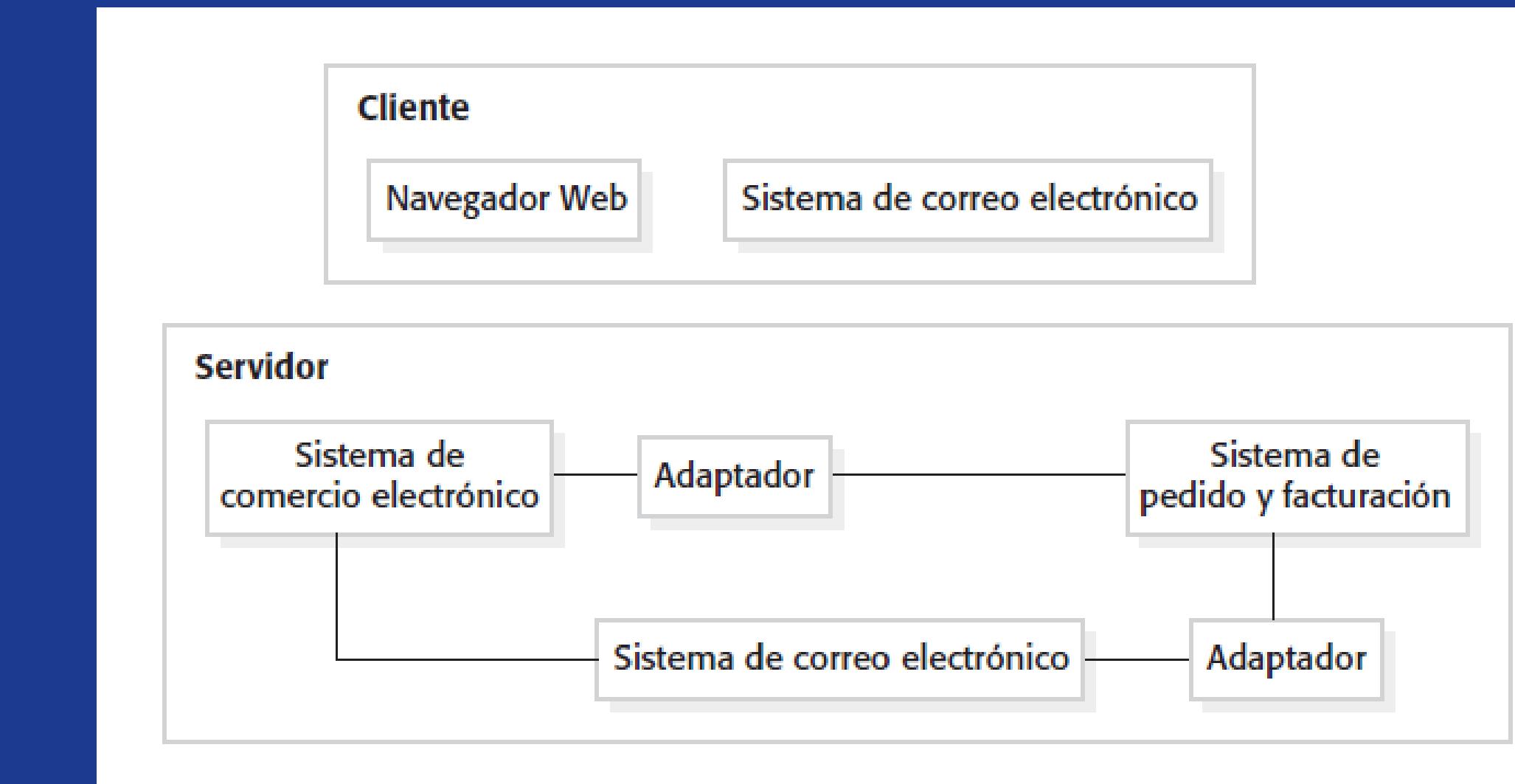
Se pueden utilizar cuando existe un sistema en el mercado que cumple todas las necesidades.

Pueden interactuar con otros sistemas internos a través de interfaces.

Estos sistemas pueden integrarse unos con otros

Ejemplo

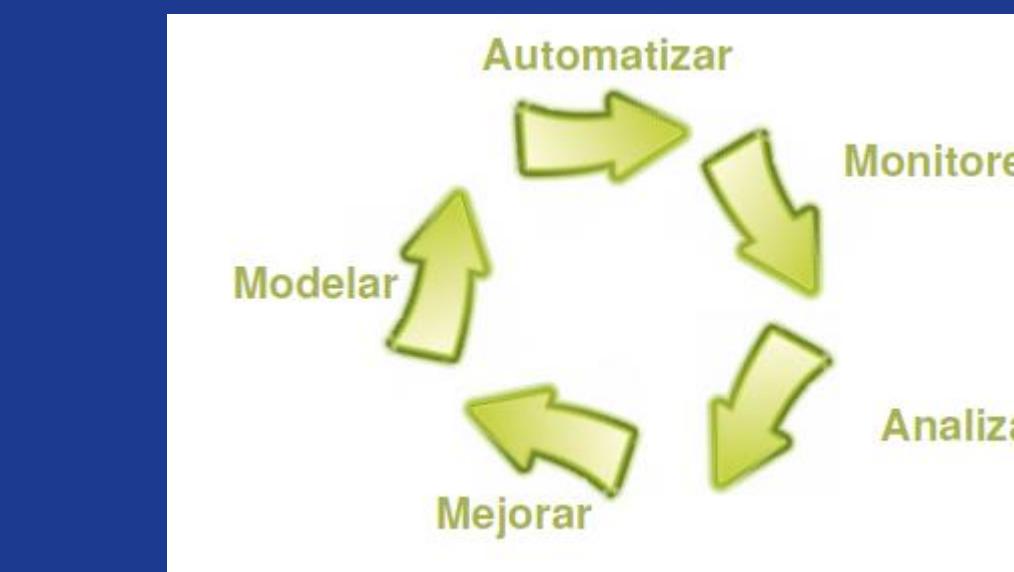
Sistema de compras:



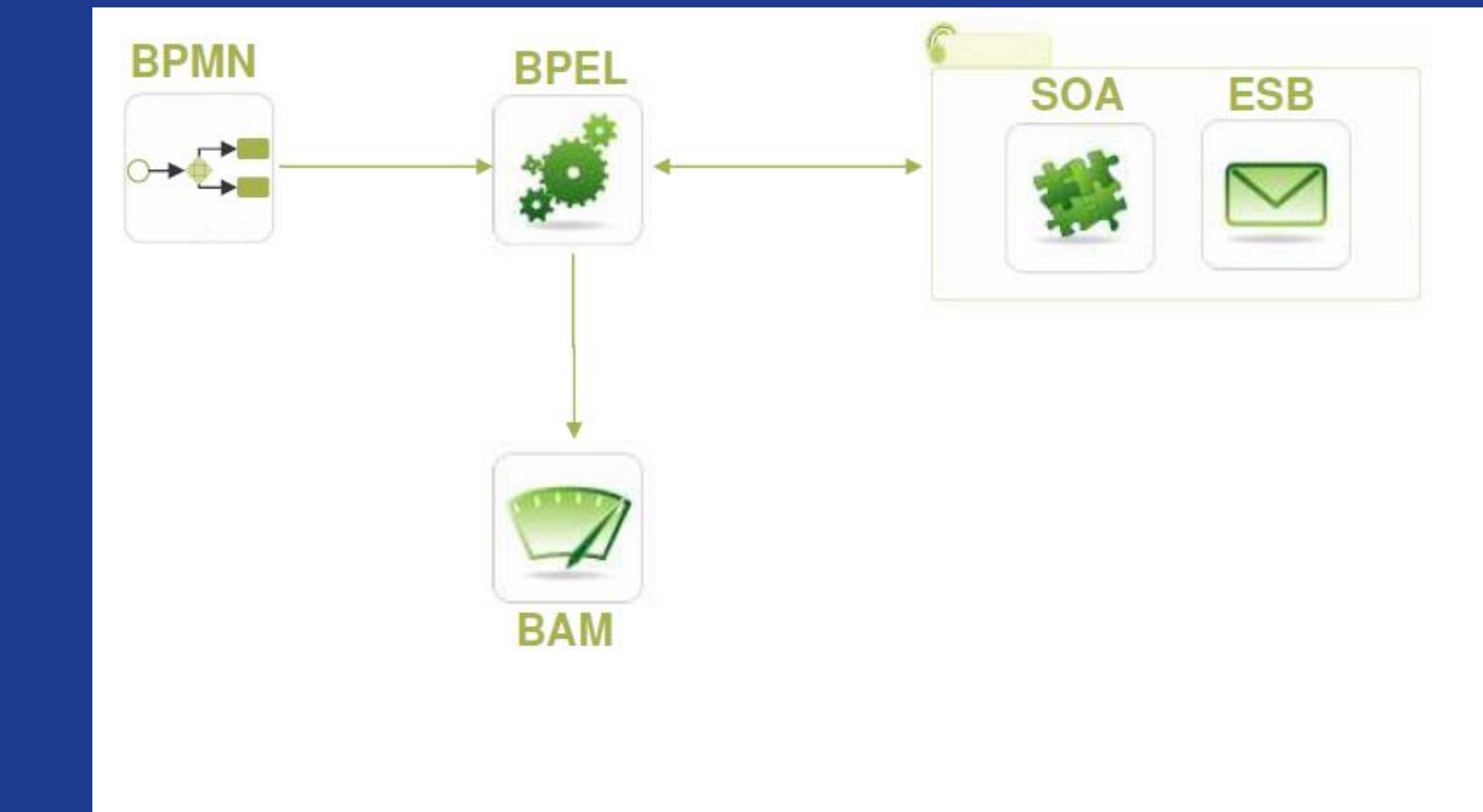
BPM

Business Process Management

Permite diseñar, implementar, probar, simular, ejecutar y modificar procesos de negocios en una organización



BPM



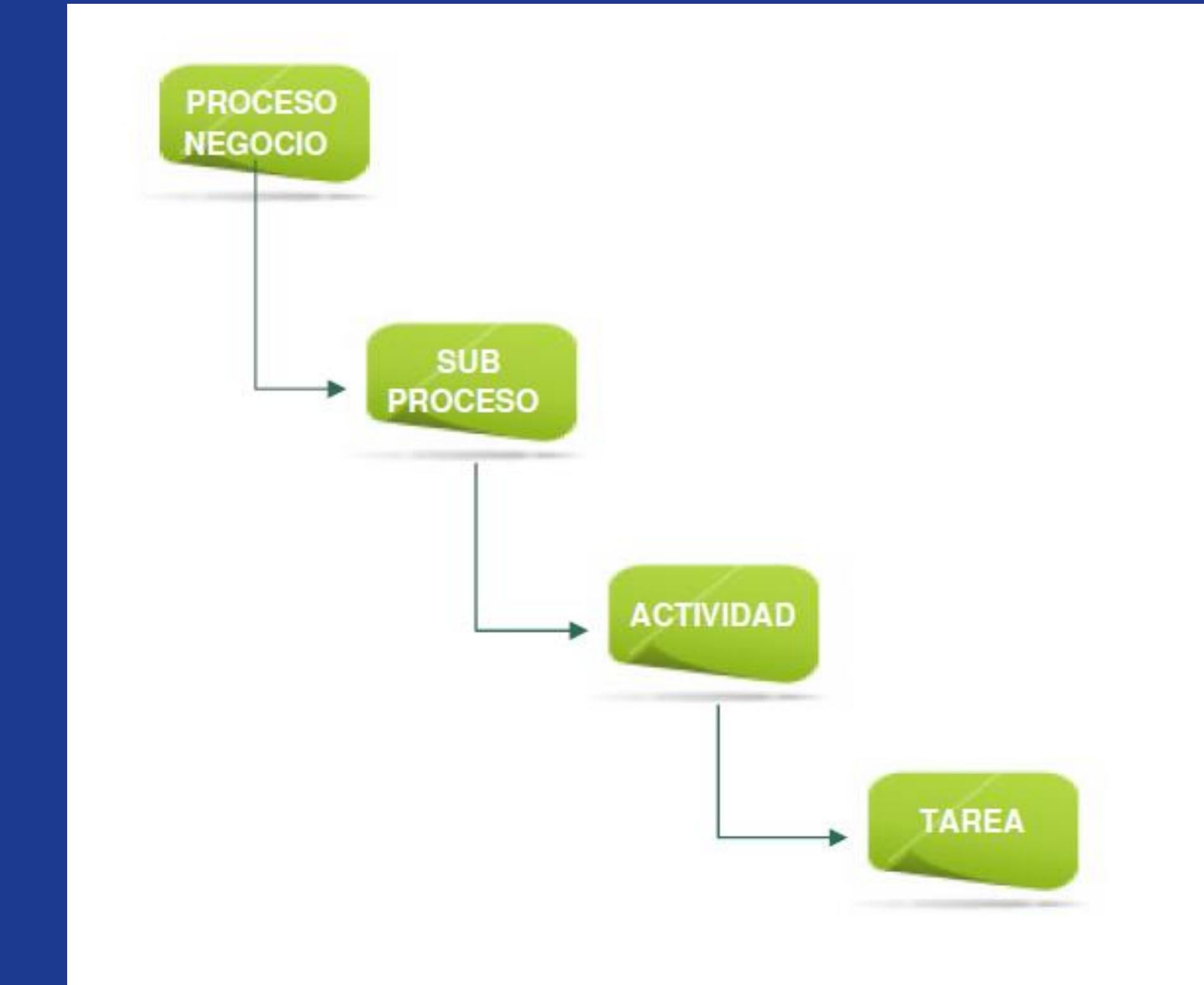
BPM

BPMN: Estándar para modelado de procesos de negocio

BPEL: Lenguaje estándar para ejecutar procesos de negocio

BAM: permite el monitoreo de actividades de negocio
utilizando indicadores clave o KPIs

BPM



ERP

Sistema a gran escala

Apoyo a empresas para acciones:

Pedidos

Facturación

Manejo de Inventarios

Proyecciones de producción

ERP - Módulos

Compras

Ventas

Inventario

Facturación

Contabilidad

Clientes

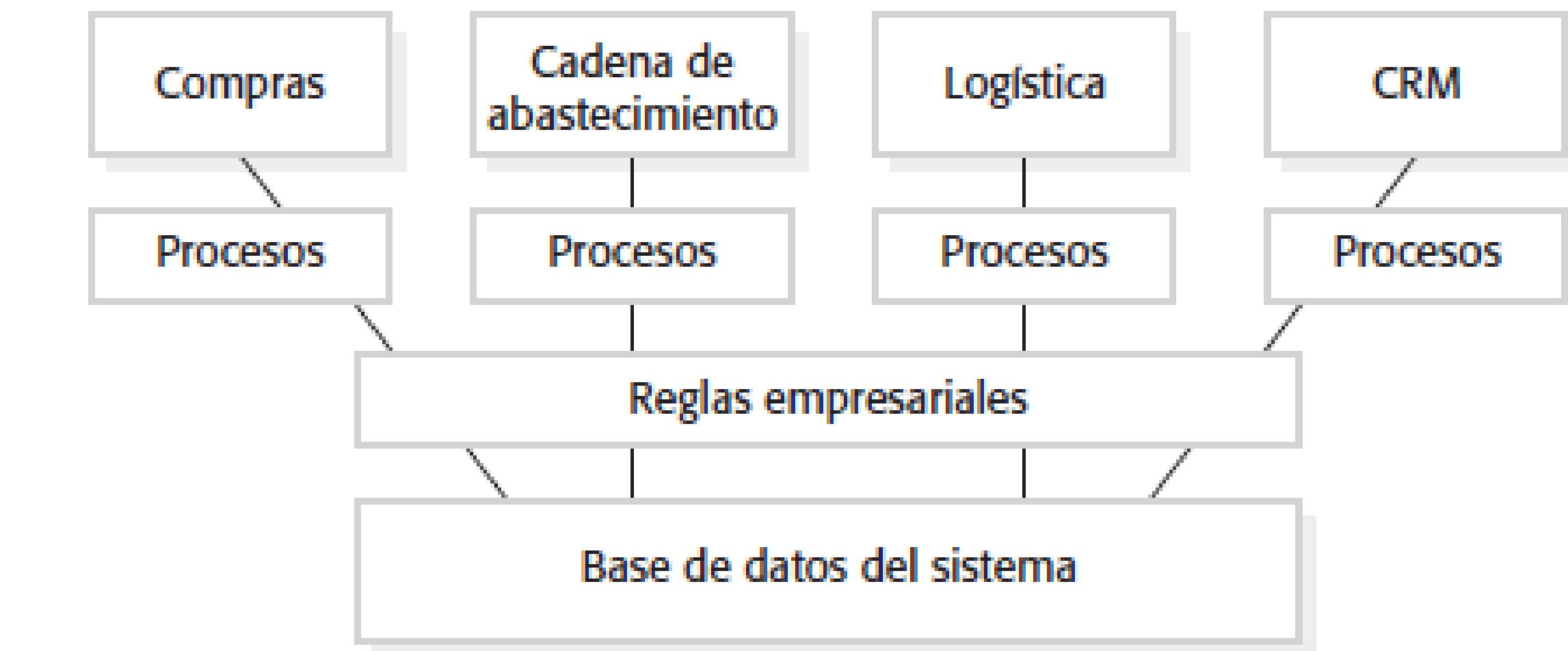
Recursos humanos / Planilla

Marketing

ERP - otros módulos



ERP



CRM

Estrategias de negocios centradas en el cliente

Para empresas cuya estrategia está destinada a lograr
identificar y administrar las relaciones en aquellas cuentas o
clientes más valiosos

Ciclo de vida del cliente



CRM - Módulos

Ventas

Telemarketing

Manejo del tiempo

Servicio y soporte al cliente

Mercadotecnia

Manejo de información para ejecutivos

E-Commerce

CRM - Módulos



CRM - Ejemplo



SCM

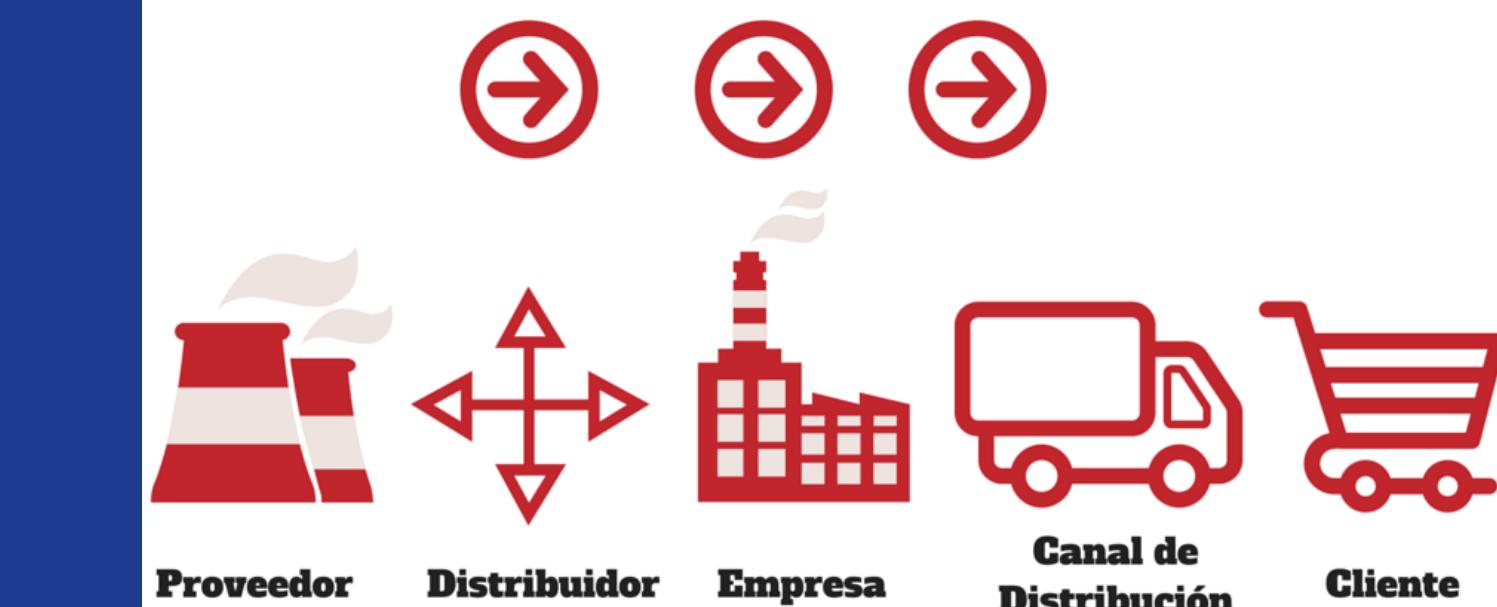
Red de medios de distribución e instalaciones para:

Obtener materia prima

Transformar materia prima

Distribución de materia prima

La Cadena de Suministro



1. Suministro

Cómo, dónde y cuándo conseguimos la materia prima para que la empresa fabrique un producto.



2. Fabricación

Es cómo convertir la materia prima en productos terminados



3. Distribución

Hacer llegar el producto terminado a las manos del consumidor.



Objetivos o especialización

Entregar los productos a tiempo

Realizar un balance de operaciones adecuado

Proveer de todas las variedades de productos de la empresa en todo momento



SCM - Módulos

Aprovisionamiento estacional

Planificación

Gestión de pedidos

Gestión de pedidos de emergencia

Traslado de los productos

Gestión de inventarios

Gestión de garantías

Otros sistemas de gestión



```
128
129
130 }
131 ion-mail{
132   background: url(..../img/mailico.png) no-repeat center;
133   display: inline-block;
134   width: 12px;
135   height: 14px;
136   float: left;
137   margin: 2px 7px 0 0;
138 }
139 ion-phone{
140   background: url(..../img/phoneico.png) no-repeat center;
141   display: inline-block;
142   width: 20px;
143   height: 18px;
144   float: left;
145   margin: 2px 7px 0 0;
146 }
```

(root@private:/var/folders/11/q7013v687zg3c905dra/pamr0000gp/T/8c96fb21-8e7e-4a01-bb5f-804a1b85eb11/antir/css/style.css)

SGA

Sistema de gestión de almacenes

Los CSM y ERPs cubren este sistema

Su especialización es conseguir un mayor nivel de automatización en los almacenes.

Fomenta el uso de lectores de códigos de barras

Sistemas de reconocimiento de voz

Terminales para fácil actualización de estado

MRP

Manufacturing resource planning

Controla la disponibilidad de recursos y materiales

Se enfoca en compras y calendarios de distribución para su almacenamiento

APS

Advanced planning and Scheduling

Planificación de producción en base a pedidos

En algunos casos solo se necesita gestionar la logística para los procesos de producción

PLM

Product Life Management

Administrar productos y generar innovación sobre los mismos

Procesos creativos para la creación de productos

DMS

Sistema de gestión Documental

Procesamiento de datos respecto a distintos procesos de la empresa

Enfoque en adjuntar los documentos

Gestión de contratos

Gestión de pedidos

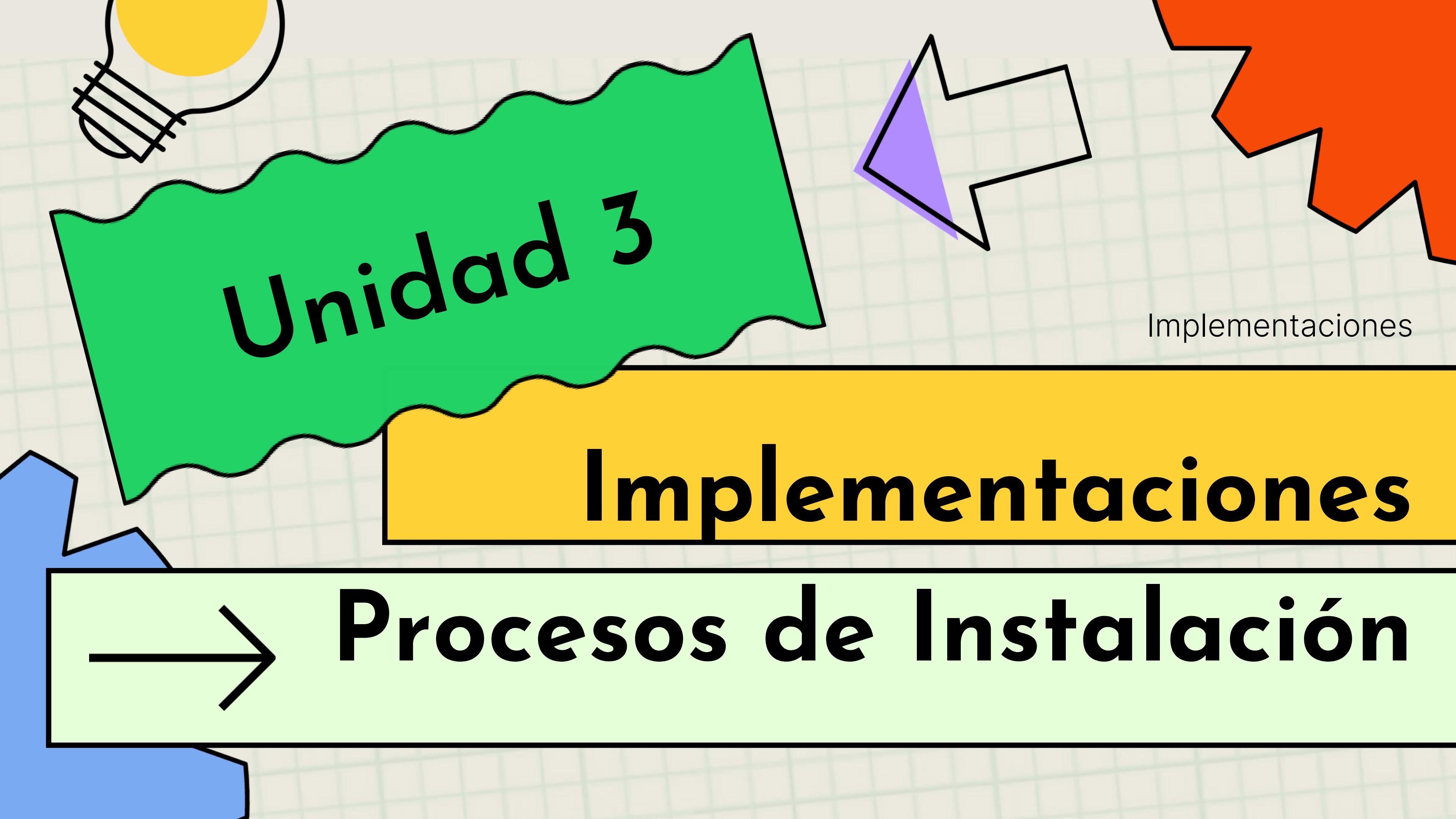
Documentos de aseguradoras

B.I.

Business intelligence

Análisis de la información disponible

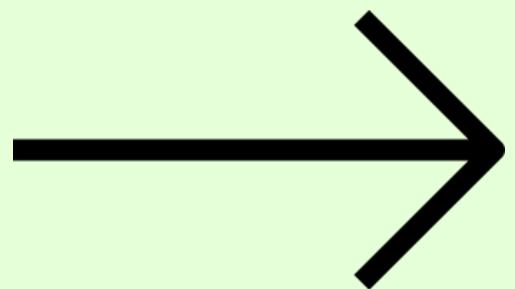
Ayuda únicamente en las tomas de decisiones



Unidad 3

Implementaciones

Implementaciones



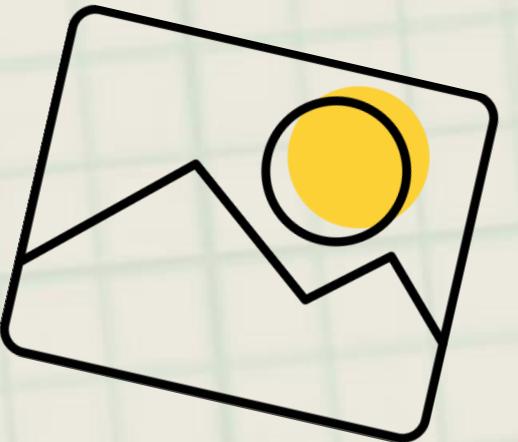
Procesos de Instalación

Implementación

- Una etapa crítica en la ingeniería de software es la implementación.
- La implementación es crear una versión ejecutable y lista para utilizarse del software

Conflictos de implementación

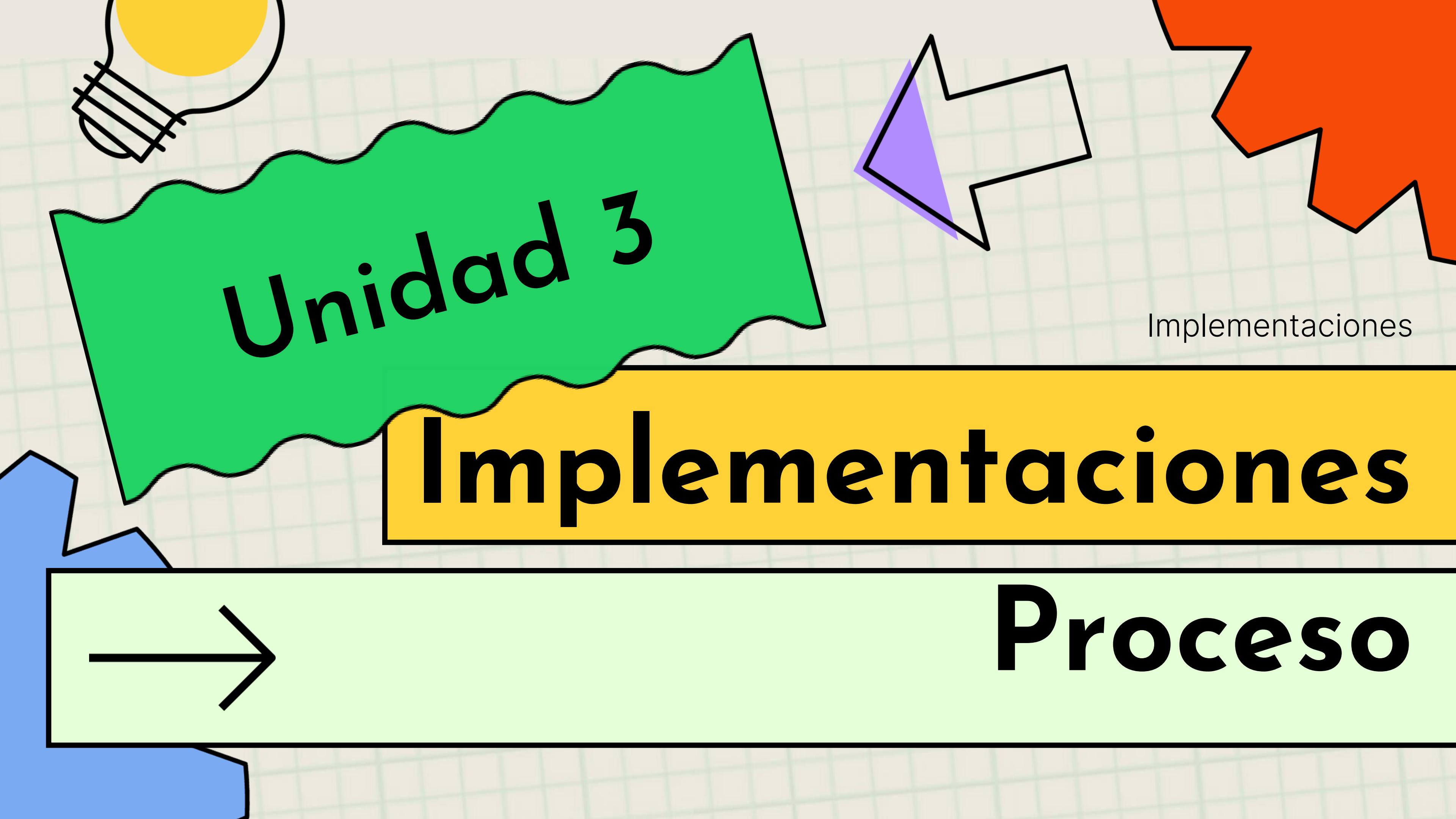
La puesta en producción significa el éxito?



Implementación

- La implementación implica además, configurar el software para su utilización.
- También se corrigen fallos y posibles vulnerabilidades.
- Revisión y asignación de licencias.

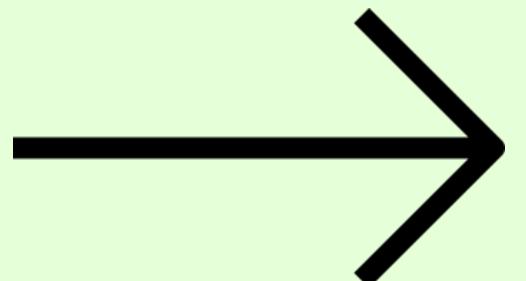




Unidad 3

Implementaciones

Proceso



1. Implementación

- Es cuando un producto de software se instala sin modificaciones en su código.
- Es una de las fases más complejas del proceso.

1. Implementación

- Compilación
- Instalación
- Copia de archivos
- Personalización de configuraciones.



1.1 Compilación

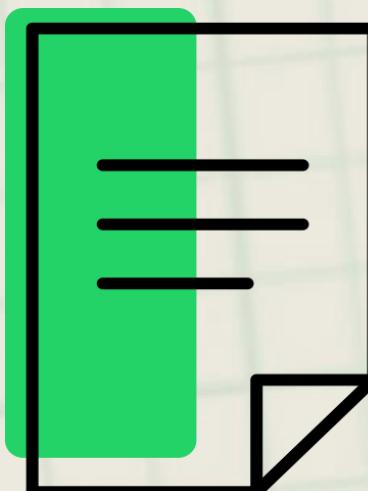
- Es traducir el código fuente a lenguaje de máquina.
- Toma en cuenta también la generación de un programa llamado “instalador” para poder ser implantado en el sistema operativo correspondiente.

1.2 Personalización

- Consiste en adaptar el producto de software a los requerimientos del usuario final.
- Esto mediante el proceso de “Parametrización” donde se ajustan peculiaridades del software para funcionar en un entorno productivo

1.3 Migración de datos

- Consiste en trasladar los datos que el cliente desea incorporar.
- Es posible que se tenga que pasar por un proceso de transformación de datos.



Unidad 3

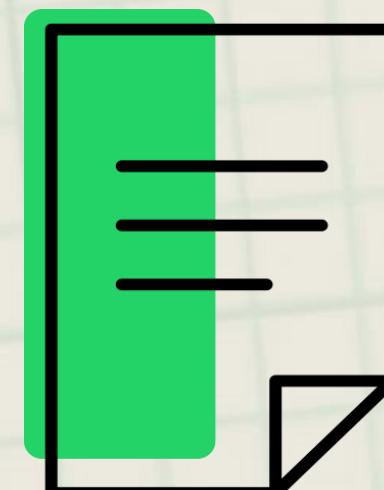
Implementaciones

Implementaciones

→ Administración de configuración

Administrar Configuraciones

- También se denomina CM por configuration management.
- Gestiona las políticas, procesos y herramientas para administrar sistemas de software que son cambiantes.



Administrar Configuraciones

- Una de sus principales funciones es la de no perder de vista los cambios realizados y versiones de los componentes que se incorporan.
- Cuando se trabaja con múltiples versiones en uso, es necesario hacer correcciones o adaptaciones en versiones específicas.
- No siempre los usuarios deben deshacerse del software antiguo para usar el que resuelve 1 problema.



1. Administrar el cambio

- Significa dar seguimiento a las peticiones de cambios del software.
- Decidir si los cambios son necesarios o no.



2. Gestión de versiones

- Es el seguimiento de las versiones tanto en código como de liberables generados para el código.
- También su objetivo es garantizar que los cambios hechos por diferentes personas o equipos no interfieran entre sí.
(Integración continua)
- Tener conocimiento de quiénes utilizan qué versión.
- Discontinuar versiones en específico.



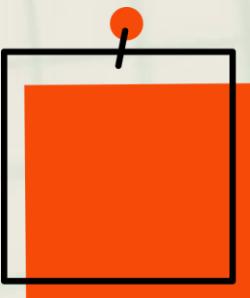
2. Gestión de versiones

1. Tomar la rama principal hacia el espacio de trabajo.
2. Construir los cambios
3. Ejecutar pruebas automatizadas luego de efectuar cambios
4. Integrar los cambios
5. Compilar el sistema
6. Realizar pruebas
7. Si todas las pruebas pasan se integran los cambios a la rama principal



3. Gestión de entregas

- Implica prepararnos para la entrega final al cliente, esto no solo se refiere a los instaladores sino a todo lo necesario como: documentos de recibido, seguimiento de pruebas de UAT, etc.

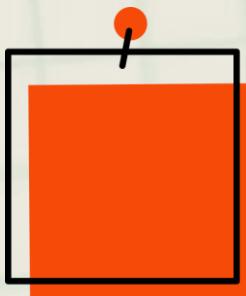


3. Gestión de entregas

Existen diferentes tipos de entregas:

- Entrega mayor
- Entrega menor

X.X.X



4. Gestión de personalizaciones

- Cuando se crea un software para más de un “cliente” suele ser genérico y tiende a ser configurable.
- No todos los clientes o unidades necesitan acceso a lo mismo.
- No todos los clientes o unidades utilizarán los mismos módulos.
- Muchas veces como parte de la instalación existen integraciones con otros sistemas e incluso hardware.

¿Cómo gestionar la configuración?

Herramientas de Integración de código y control de versiones

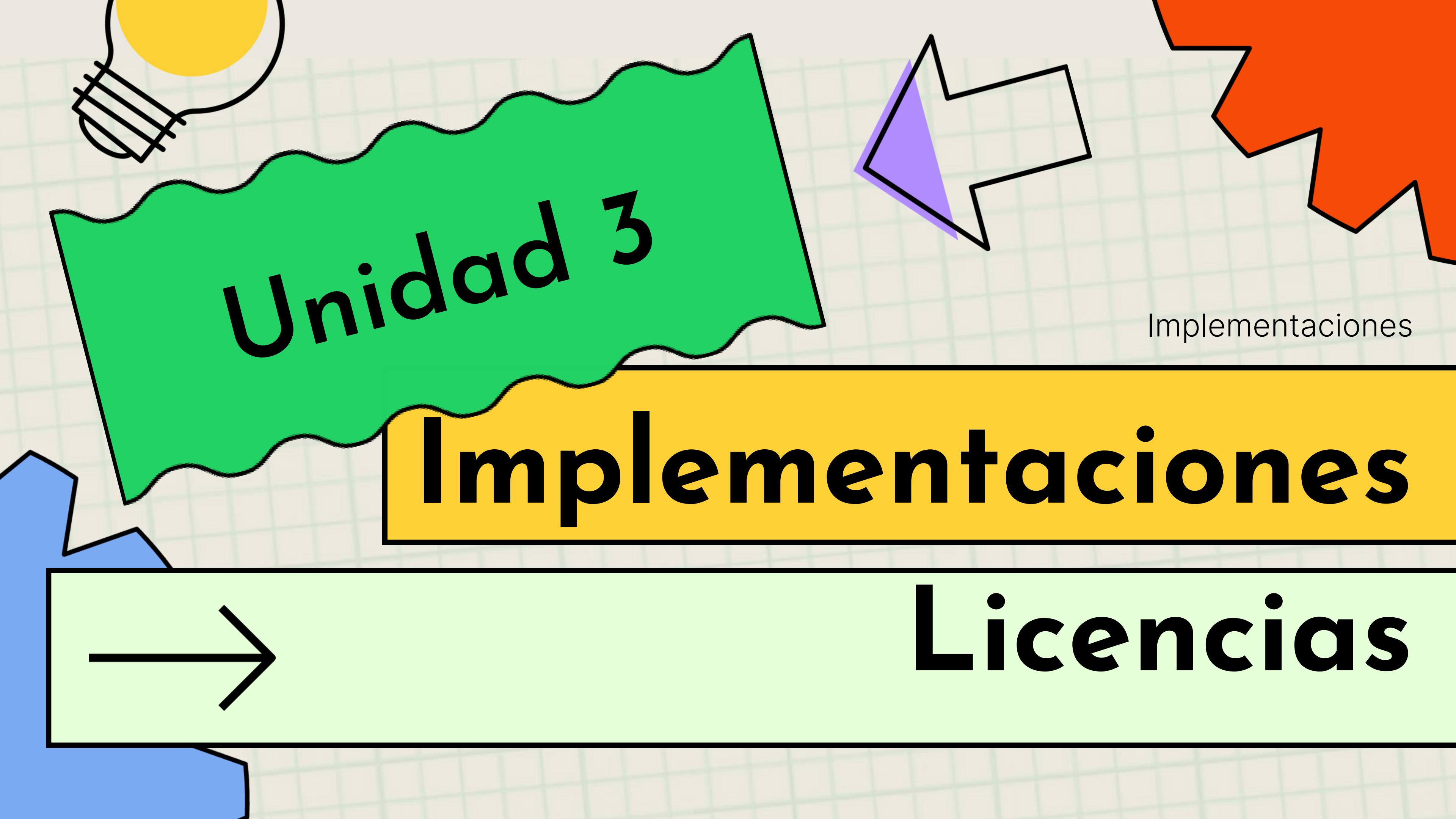
Herramientas de compilación de software

Políticas para el manejo de versiones de entregables

Políticas para el manejo de ramas

Documentación:

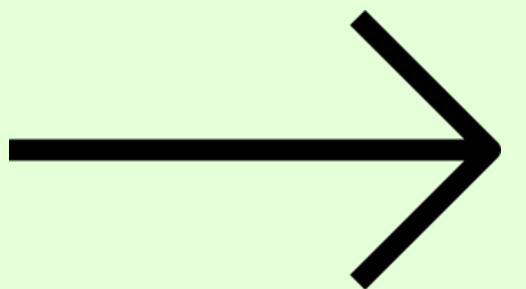
- Documentos de los pasos para la instalación
- Documentos de acuerdo entre el cliente y el equipo
- Documentos solicitando información o configuraciones previas necesarias.



Unidad 3

Implementaciones

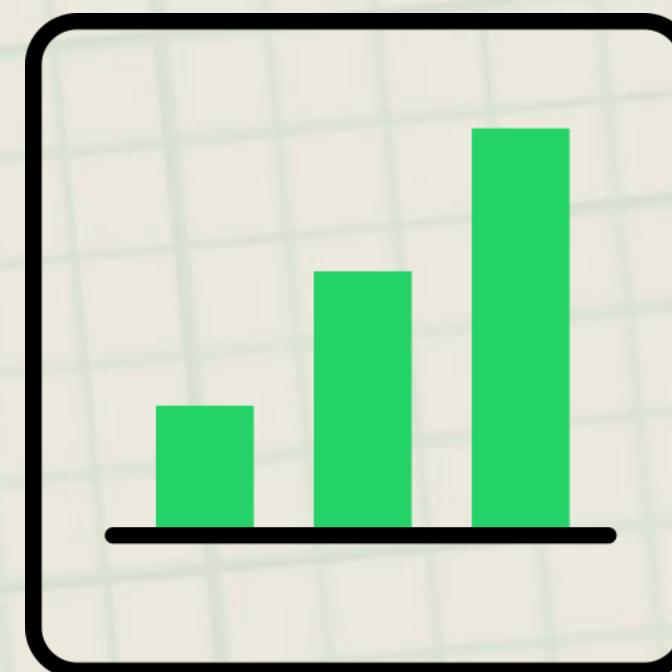
Implementaciones



Licencias

Licencias

- El costo de un software puede ser mucho menor en caso el cliente no requiera el código fuente del producto del software.
- Existen muchas condiciones a la hora de adquirir software.



Software propietario

- Es la utilización del software sin acceso a la propiedad del bien.
- Existen múltiples restricciones del creador y se limita a que el usuario no realizará cambios en el mismo.



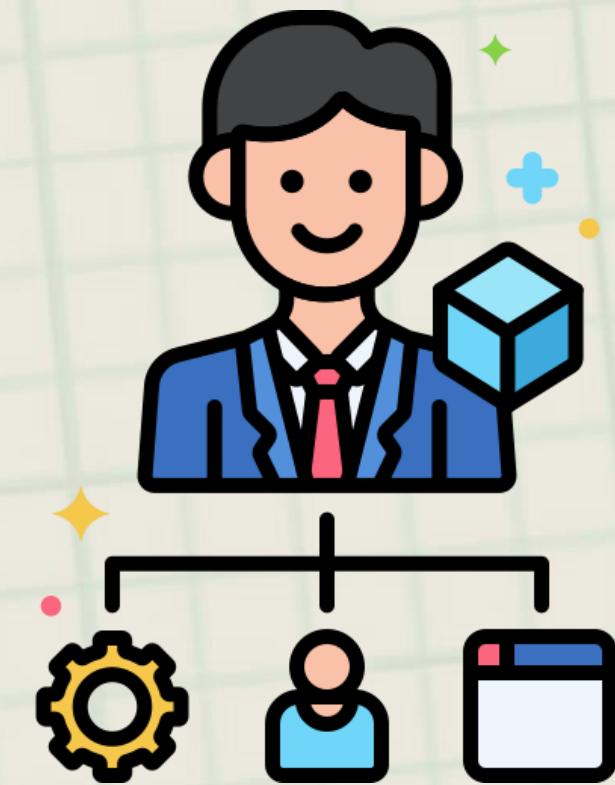
Software propietario

- Llave en mano: es el software propietario que provee acceso al código fuente para entenderlo, sin embargo puede estar sujeto a no cambiarse.
- No llave en mano: el usuario tendrá acceso únicamente al producto de software sin acceso a su código.



Software libre

- Es el software que se puede utilizar, copiar, redistribuir, entender, modificar y mejorar el software.
- No necesariamente se puede re-vender este software, debe respetar las mismas libertades del código original.



Software libre

- La libertad de ejecutar el programa como se deseé, con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa, y cambiarlo para que haga lo que usted quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello.
- La libertad de redistribuir copias para ayudar a otros (libertad 2).
- La libertad de distribuir copias de sus versiones modificadas a terceros (libertad 3). Esto le permite ofrecer a toda la comunidad la oportunidad de beneficiarse de las modificaciones. El acceso al código fuente es una condición necesaria para ello.



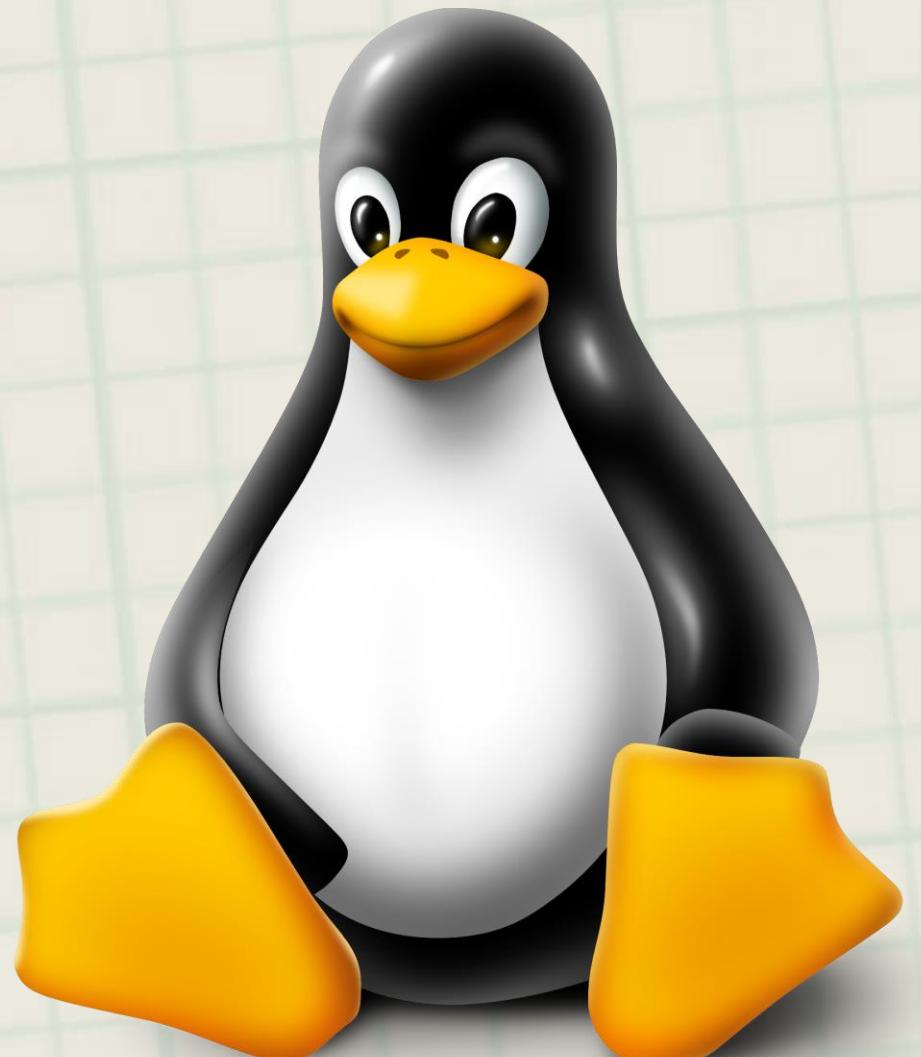
Software Código abierto

- Acceso al Código y modificaciones pero no se obliga que el SW derivado mantenga las libertades del Código original. Copiar, usar, distribuir y modificar libremente, pero debe incluir copyright del desarrollador original y colaboradores.



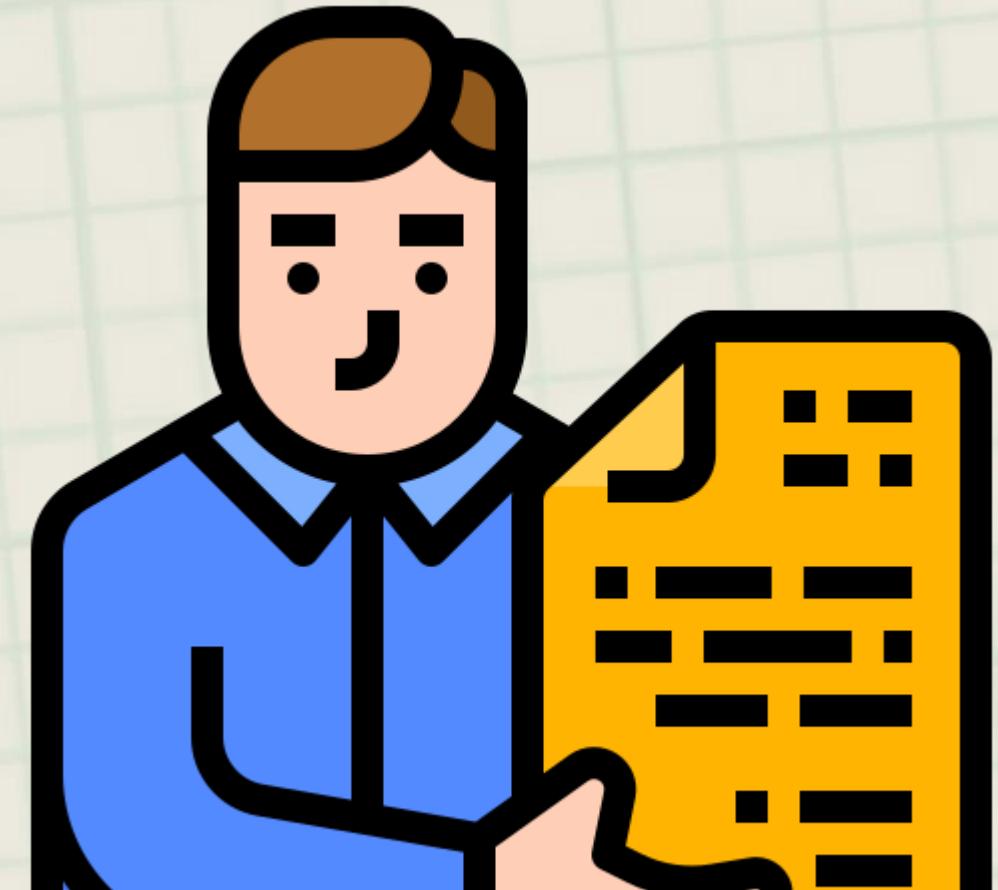
Software Código abierto

- **Licencia pública GNU:** también llamada licencia recíproca.
- Si se utiliza el software de Código Abierto bajo la licencia GPL, el producto de esto debe ser también de código abierto.



Software Código abierto

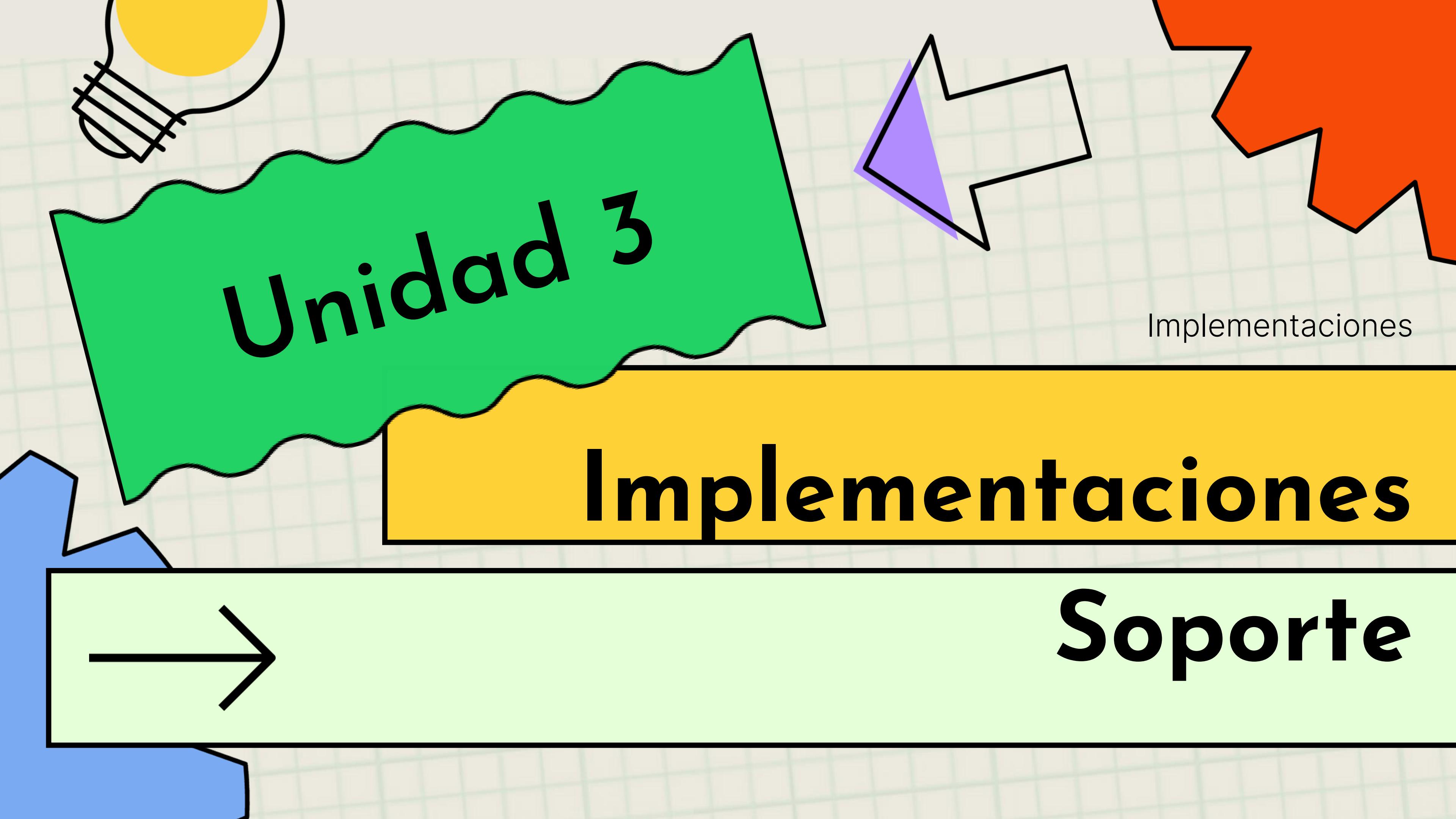
- **Licencia pública menos general GNU:** Se pueden escribir componentes basados en el código abierto sin tener que publicar éste código.
- Pero en caso de cambiar código del software de código abierto, este sí debe publicarse.



Software Código abierto

- **Licencia Berkeley Standard Distribution:** Es una licencia no recíproca.
- No se está obligado a publicar modificaciones en el código abierto.
- Se puede incluir en software propietario de venta, producto de la reutilización de componentes de software de código abierto

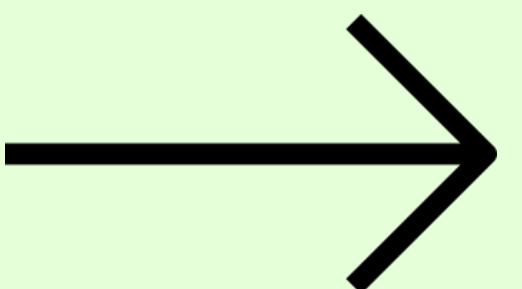




Unidad 3

Implementaciones

Implementaciones



Soporte

Acuerdos de Soporte

- Promover el uso del software y resolución de problemas tiene un costo.
- Existen diferentes estrategias y niveles de compromiso.



Acuerdos de Soporte

Componentes

- Normalmente va ligado a un tipo de licencia.
- Es un soporte más específico
- Se necesita evaluar si el cambio debe realmente hacerse.
- Suele tomar tiempo
- Existen foros al respecto

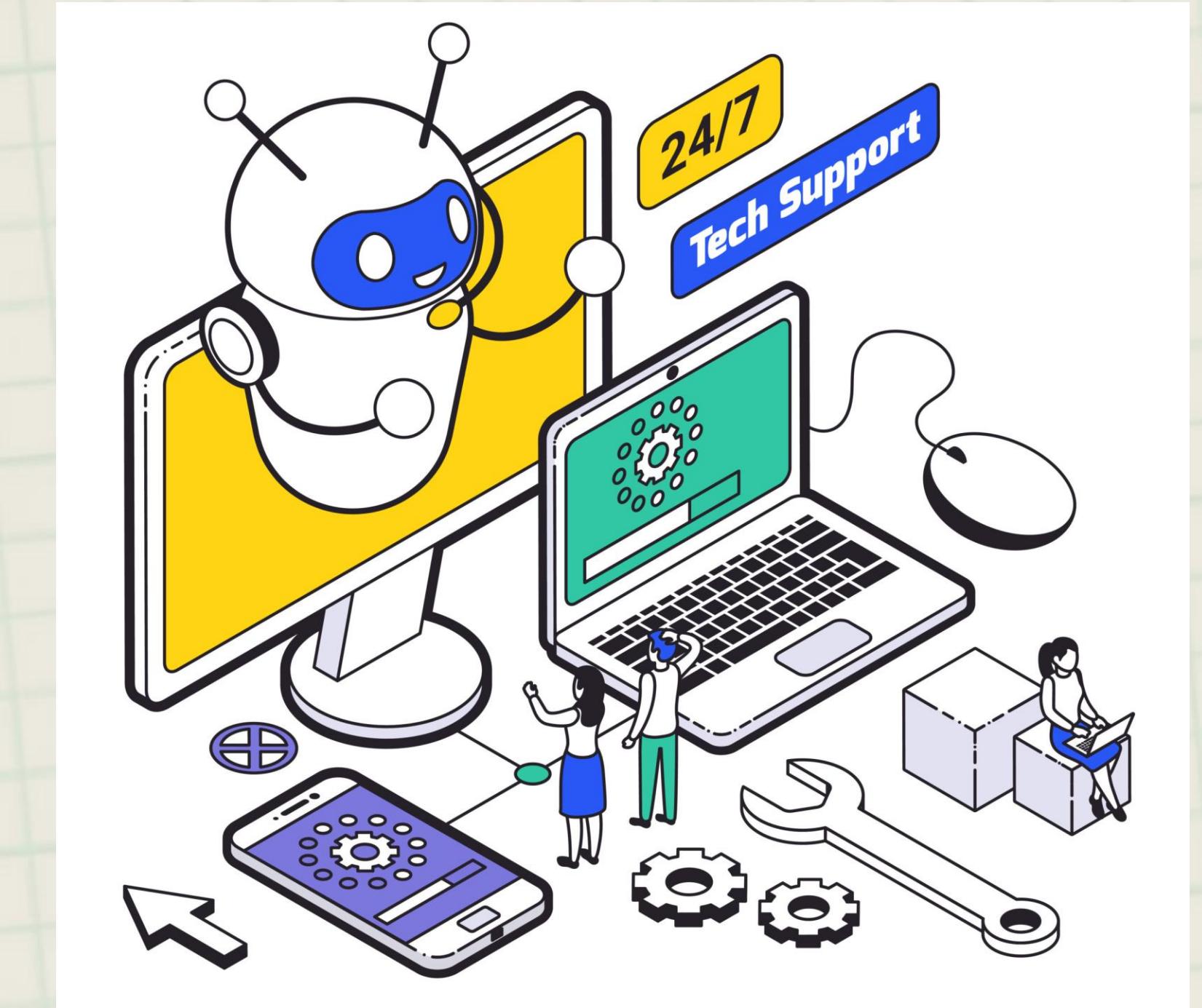
Acuerdos de Soporte

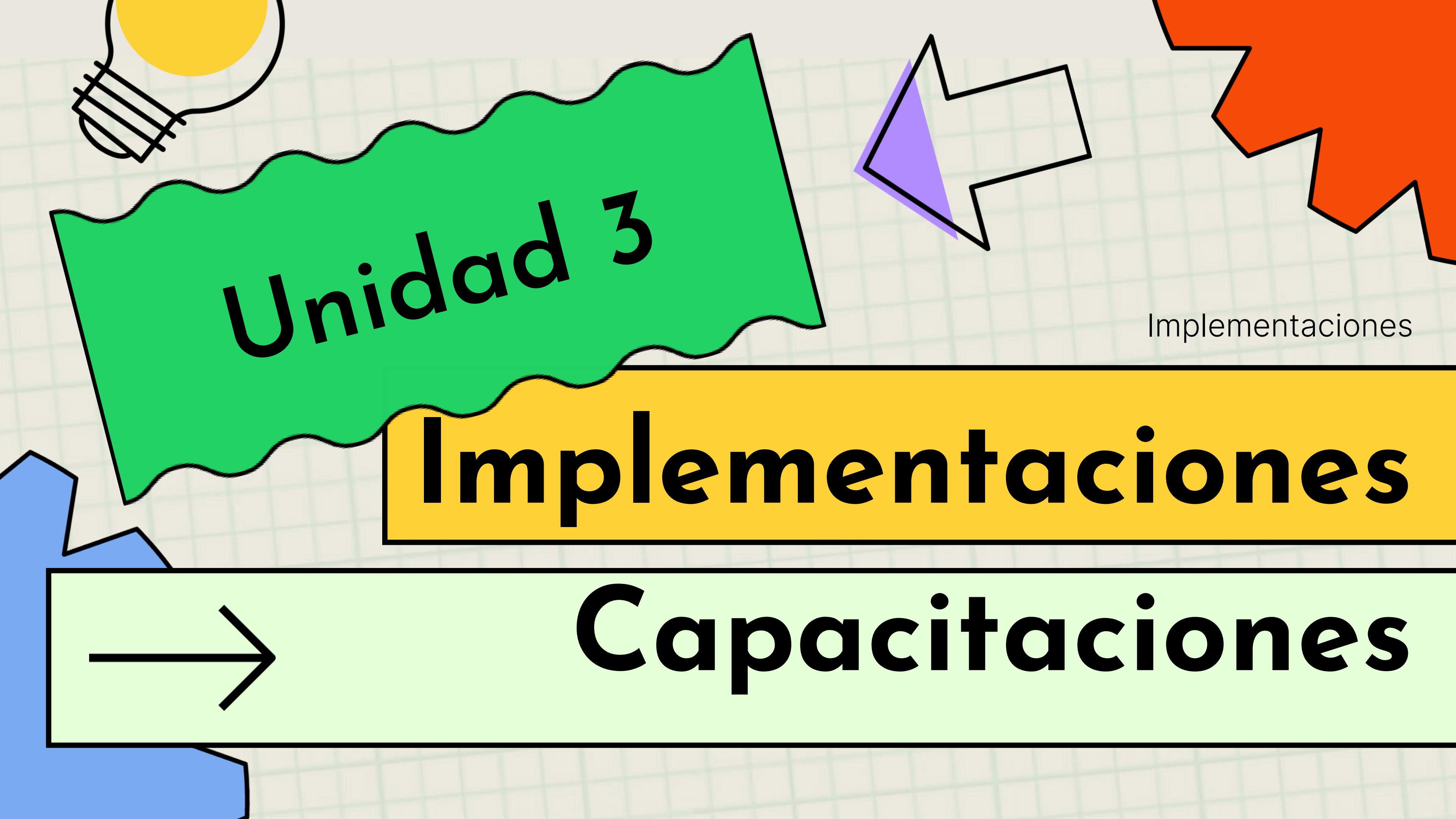
Aplicaciones/Sistemas

- Existen acuerdos de Servicio
- Se crean tickets para su respectiva revisión
- En caso de ser un bug, debe repararse en la versión correspondiente
- Software como servicio requiere más soporte

Tipos de Soporte

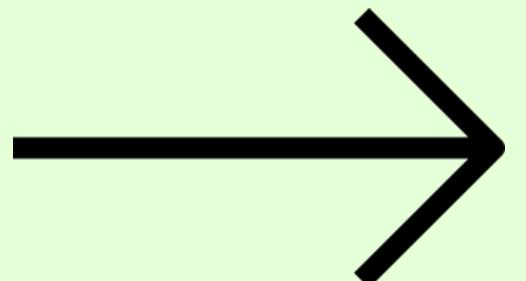
- Preventivo
- Adaptativo
- Correctivo





Unidad 3

Implementaciones



Capacitaciones

Capacitar

- Es el acompañamiento que se les da a los usuarios para la correcta utilización de un software.

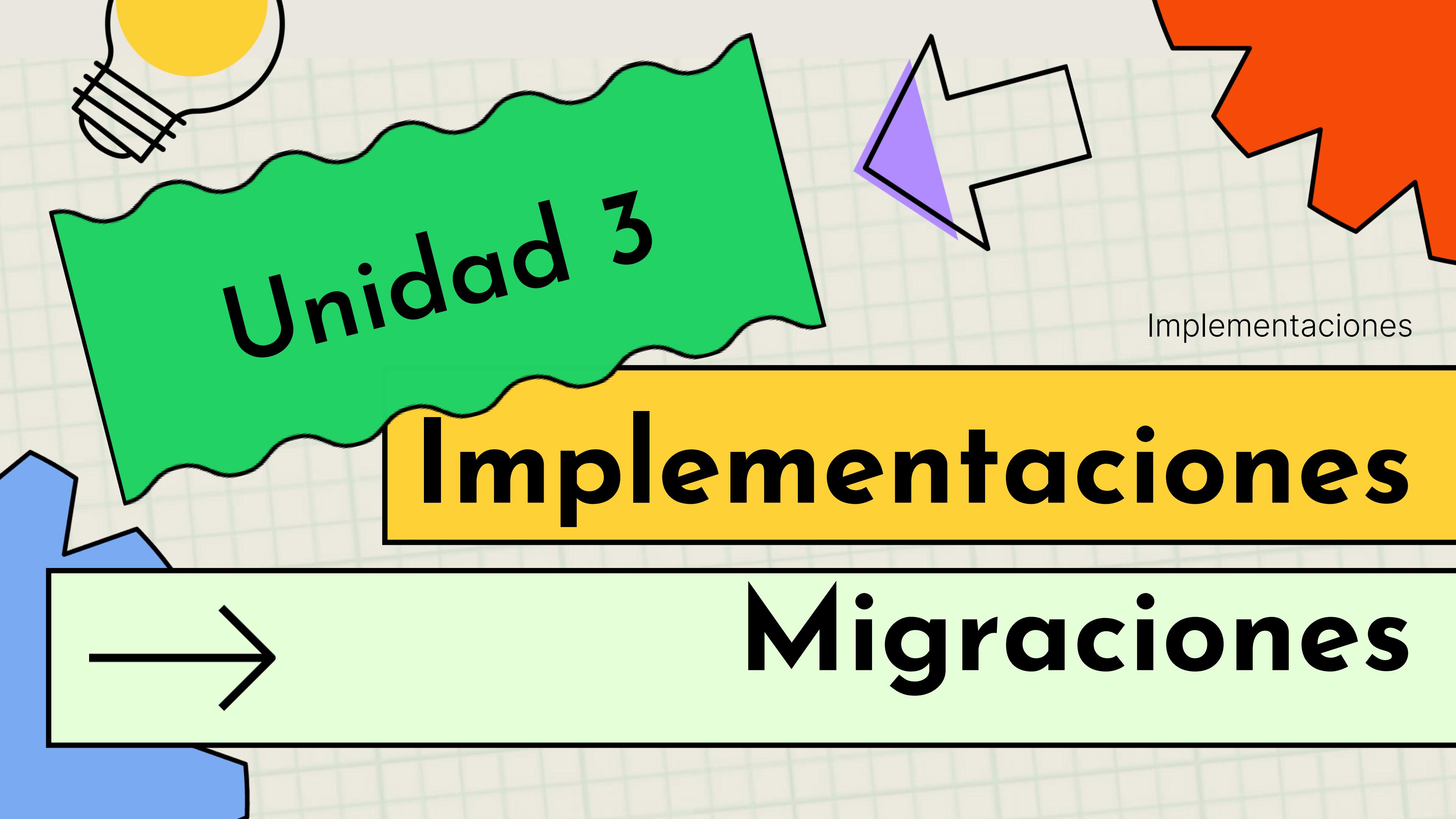


Capacitar

- Se debe identificar las necesidades de la capacitación, si se necesitan conocimientos previos en algún tema en particular.
- Se deben esclarecer los objetivos de la capacitación, qué se quiere lograr.
- Se establece también un alcance de la capacitación, qué es y no responsabilidad del equipo.

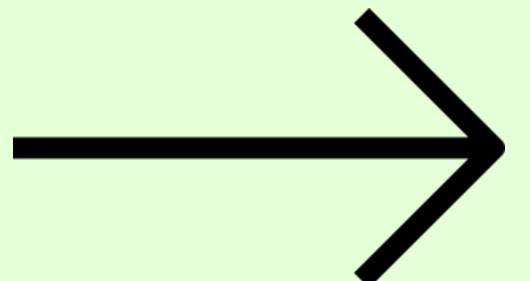
Capacitar

- Se establece si habrá un entorno en particular para la capacitación o será utilizando el entorno productivo.
- Si se utilizará el nuevo sistema junto al sistema anterior
- El soporte puede ser parte de la capacitación.
- Debe tener un cronograma y ser gradual.
- Más allá de enseñar a cómo utilizar un sistema, se debe acompañar en caso exista algún problema.



Unidad 3

Implementaciones



Migraciones

Migrar:

- Migrar es transferir datos de un sistema a otro.
- Esta transmisión puede implicar transformar los datos.
- Puede surgir porque se va a reemplazar un sistema, se dará mantenimiento de infraestructura, se cambiará de proveedor de infraestructura, etc.

Migrar:

- De bases de datos
- Archivos
- Configuraciones
- Código
- Catálogos de aplicación



Formas para migrar datos:

- Utilizando un software basado en matriz
- Utilizando un software basado en host
- Uso de Dispositivos de red

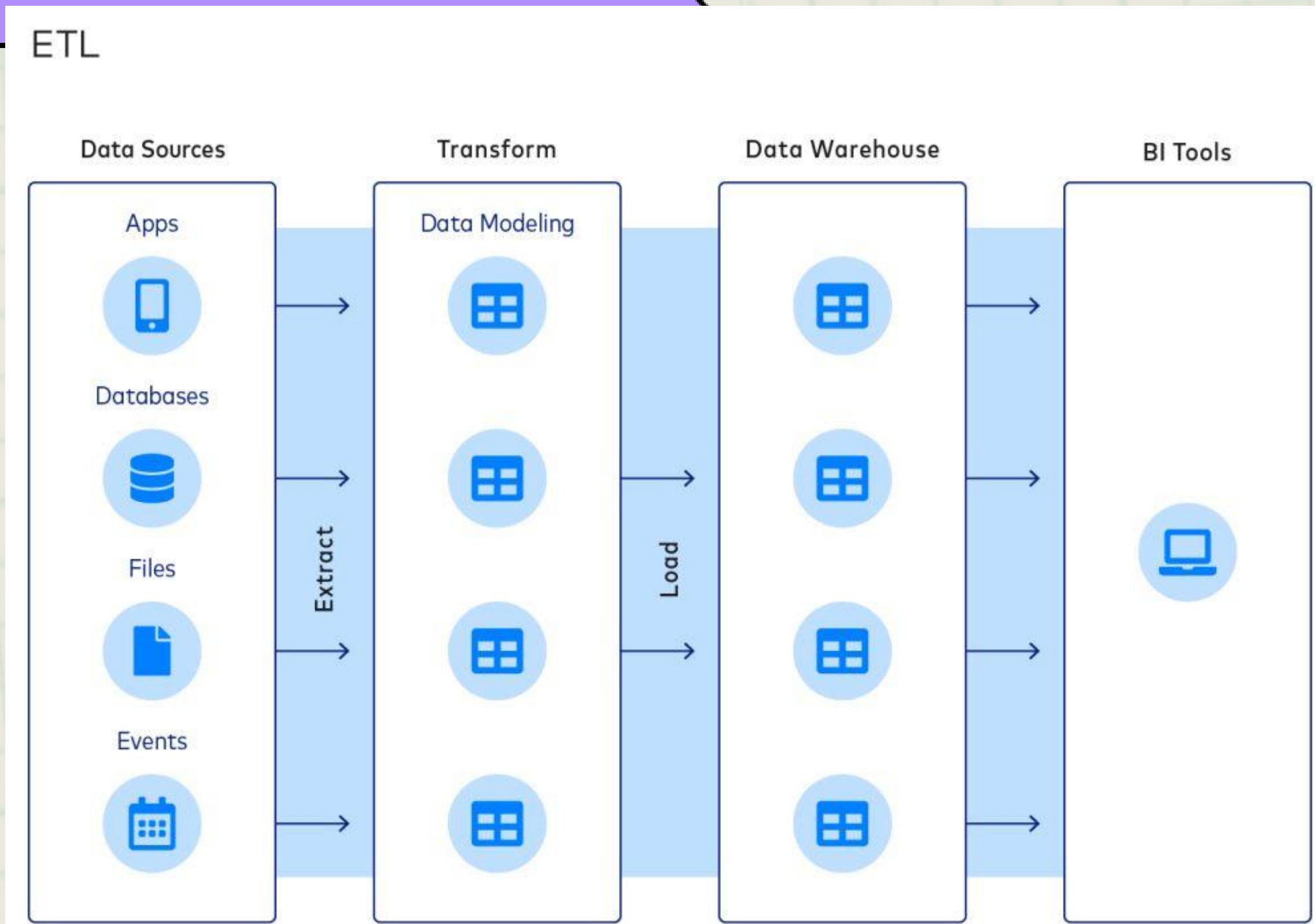
Factores a tomar en cuenta:

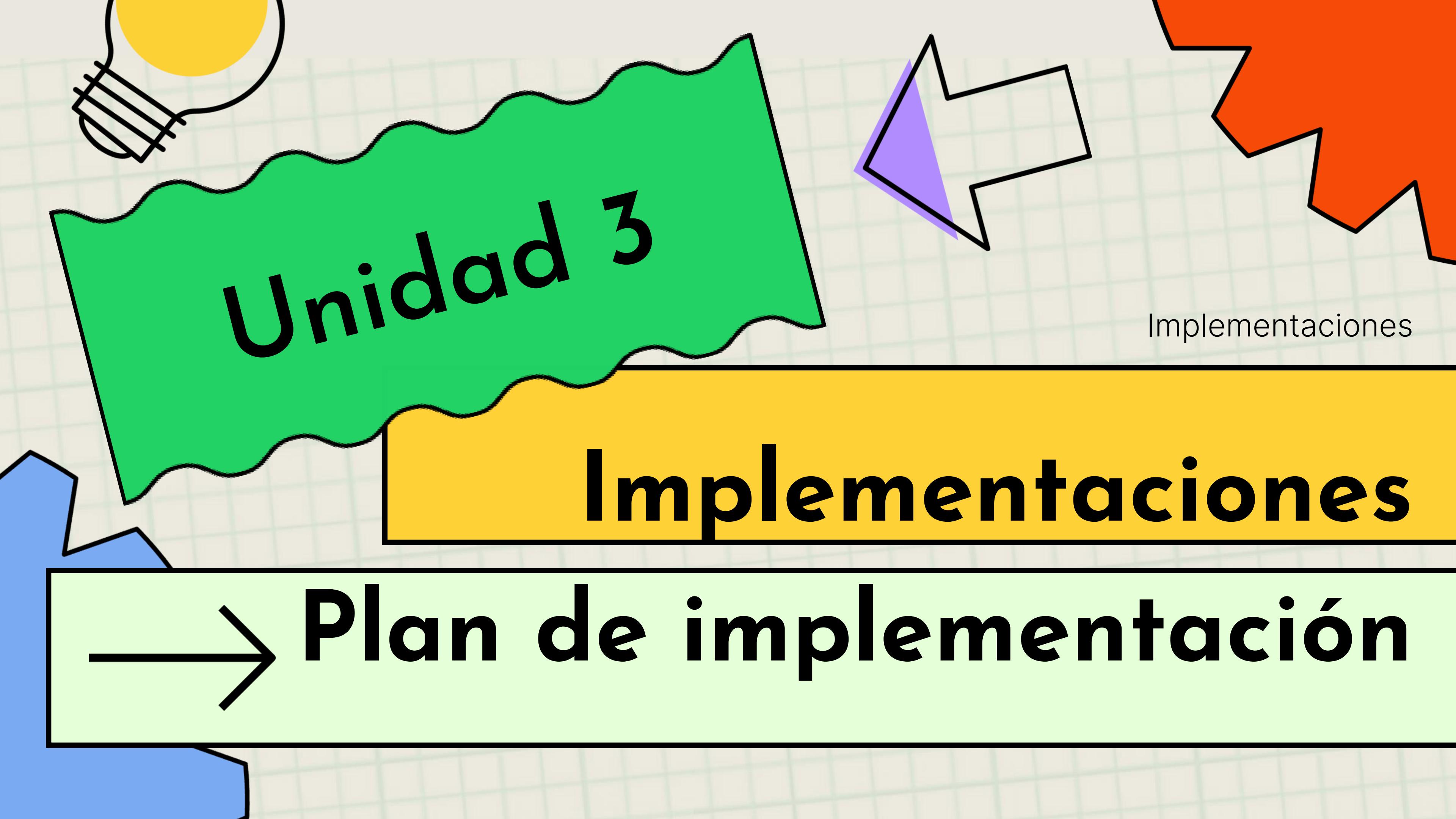
- Tiempo necesario para la migración
- Cantidad de tiempo de inactividad del sistema
- Riesgos para el negocio

Minimizando el riesgo

- Clasificar los datos en base a su criticidad
- Aplicar procesos ETL
- Deben existir políticas de migración
- Deben existir pruebas luego de la migración.

Herramientas ETL





Unidad 3

Implementaciones

Implementaciones

→ Plan de implementación

Plan de implementación

- Es un documento que contendrá la estrategia de instalación del producto de software
- Debe contener las prioridades y plazos de tiempo

Plan de implementación

- Prioridades
- Riesgos
- Plan de Instalación
- Plan de Capacitación
- Plan de soporte
- Plan de mitigación de riesgos
- Finalización de acompañamiento

