

## resumen-temas-1-2

# TEMA 1: SISTEMAS ERP-CRM

## ¿QUE ES UN ERP?

Un ERP es un sistema de gestión global para la empresa. Está compuesto por diferentes módulos y se utilizan en la producción, ventas, compras..

---

## OBJETIVOS DE UN ERP

Los objetivos principales de los sistemas ERP son:

- Optimización de los procesos empresariales.
  - Acceso a la información.
  - Posibilidad de compartir información entre todos los componentes de la organización.
  - Eliminación de datos y operaciones innecesarias de reingeniería.
- 

## PROPOSITO DE UN ERP

El propósito es otorgar apoyo a los clientes del negocio, tiempos rápidos de respuesta a sus problemas, un eficiente manejo de información que permita la toma oportuna de decisiones y disminución de los costos

Los beneficios que puede aportar una herramienta de ERP se resumen en la resolución de los problemas contables, mercantil o fiscal de la empresa.

---

## CARACTERÍSTICAS DE UN ERP

- **Modulares:** Una ventaja es que la funcionalidad de un ERP se encuentra dividida en módulos, los cuales pueden instalarse de acuerdo con los requerimientos del cliente.
- **Configurables:** Los ERP pueden ser configurados por cada empresa con cambios código del software.
- **Especializados:** Un ERP especializado ayuda a la empresa en las áreas más complejas y bajo una estructura de constante evolución.
- **Base de datos** centralizada
- Los componentes **interactúan entre sí** (operaciones)
- Los **datos** que se guardan deben ser consistentes, completos y comunes.
- **Reingeniería de procesos:** Las empresas suelen tener modificar algún proceso para alinearlos con su sistema ERP.

# ÁREAS FUNCIONALES DE UN ERP

Algunos de los módulos funcionales de la ERP son los siguientes:

- **Módulo de Planificación de la Producción:** Un ERP optimiza la utilización de fabricación, recursos materiales y las piezas utilizando los datos de producción y la previsión de ventas.
  - **Módulo de Compras:** Este módulo se utiliza para la automatización y gestión de compra de artículos necesarios para llevar a cabo las tareas.
  - **Módulos de Ventas:** Este módulo automatiza las tareas de ventas, pedidos de clientes, facturación y envío de productos
  - **Contabilidad y Finanzas módulos:** Este módulo interactúa con los otros módulos para recoger los datos financieros.
  - **Módulo de Recursos Humanos:** El módulo maneja la historia del empleados tanto actuales como despedidos.
  - **Módulo de Fabricación:** Este módulo incluye el diseño de productos, listas de materiales, gestión de costos, flujo de trabajo, etc.
  - **Módulo de Marketing:** Ayuda a la generación de actividades de promoción.
- 

## VENTAJAS DE UN ERP

- Aporta **personalización**, **control** y **optimización**.
  - Todo es **modificable**.
  - Cualquier **cambio** que se necesite realizar estará integrado.
  - Se **protegen los datos** que circulan por debajo del ERP.
  - Ayuda a integrar los **controles de calidad** y las distintas fases por las que pasa un negocio.
- 

## DESVENTAJAS DE UN ERP

- Implantar un ERP lleva **tiempo** y requiere de un proceso de mantenimiento
  - Formar a los conlleva una **pérdida** de **tiempo** y **dinero** a la empresa
  - El **coste** de la instalación y mantenimiento es muy elevado
  -
- 

## EJEMPLOS DE ERP

SAGE, Oracle, Microsoft Dynamic 365, SAP, VELNEO, ODOO, OpenBravo...

---

# ¿QUÉ ES UN CRM?

Un CRM se encarga de la estrategia de negocios centrada en el cliente. Sirve para ser más efectivos en la interacción con el cliente para que este mantenga su lealtad con nosotros

---

## COMPONENTES DE UN CRM

El CRM consiste en 10 componentes:

- Funcionalidad de las **ventas** y su **administración**
  - El **telemarketing**
  - El manejo del **tiempo**
  - El **servicio** y **soporte** al cliente
  - El **marketing**
  - El manejo de la **información** para ejecutivos
  - La **integración** del ERP
  - La excelente sincronización de los **datos**
  - El **e-Commerce**
  - El servicio en el campo de **ventas**.
- 

## BENEFICIOS DE UN CRM

- **Retención** de clientes
  - **Marketing** efectivo
  - Crear inteligentes oportunidades de **venta cruzada**
  - Incorporación más fácil de **nuevos productos**
- 

## CANALES DE COMUNICACIÓN SISTEMAS CRM

- Las **RRSS** y el **correo electrónico** son los medios más usados. Podemos incluir aquí publicidad en internet.
  - Los **Call Center**.
- 

## Factores de éxito en un CRM

- **Determinar** las **funciones** que se desean automatizar
- Obtener el **soporte** y **compromiso** de los niveles altos de la compañía
- Emplear inteligentemente la **tecnología**

- Involucrar a los usuarios en la **construcción del sistema**
  - Realiza un **prototipo del sistema**
  - Capacita a los **usuarios**
  - Motiva al personal que lo **utilizará**
  - **Administra** el sistema desde dentro
  - Mantén un comité administrativo del sistema para **dudas** o **sugerencias**
- 

## FALLOS EN UN SISTEMA CRM

- Pensar que la **tecnología** es la solución.
  - **Falta de apoyo** por parte de la dirección debido a la falta de conocimiento de las oportunidades que el CRM ofrece
  - No existe **pasión por el cliente** en la cultura de la organización
  - **Retorno de la inversión** poco claro debido a que no es un sector maduro y existe un desconocimiento generalizado sobre su ROI.
  - Falta de **visión** y **estrategia**
  - No redefinir los **procesos**.
  - Mala calidad de los **datos** e **información**.
  - Problemas con la **integración**.
  - No gestionar correctamente el **cambio**.
  - Poca implantación de **CRM analítico**
  - Inmadurez del **mercado**
- 

## TEMA 2: IMPLANTACIÓN, INSTALACIÓN Y CONFIGURACIÓN DE UN SGE

### LICENCIAS DE SOFTWARE PROPIETARIAS/PRIVATIVAS

Se conoce como Software privativo a aquél cuyo **uso**, **redistribución** o **modificación** esta **prohibido** o necesita una autorización. Para que el usuario pueda usar el software, este tiene que aceptar las licencias

---

### LICENCIAS DE SOFTWARE PROPIETARIO. MICROSOFT

Existen tres tipos de licencias en Microsoft:

- **Licencias OEM:** Licencia que viene **incluida en los equipos** cuando se compran.
- **Licencias Retail:** Licencias obtenidas **mediante la web**, útiles para usuarios que requieren **menos de 5 licencias**.

- **Licencias por Volumen** útiles para grandes empresas, centros educativos, etc.
- 

## LICENCIAS SOFTWARE LIBRES O DE CÓDIGO ABIERTO

Estas licencias se incluyen en dos categorías:

- **Licencias permisivas** Aquellas cuyo objetivo es tener los mínimos requisitos.
  - Aquellas cuyo objetivo **preserva las libertades** que son dadas.
- 

## GENERAL PUBLIC LICENSE (GPL)

Esta licencia tiene como objetivo dar a todos los usuarios libertad indefinida en el uso, estudio y modificación del software. Las modificaciones hechas y redistribuidas por el usuario final debe incluir el código fuente de estas, y la licencia de cualquier trabajo derivado no puede incluir ninguna restricción adicional.

---

## LICENCIAS DE SOFTWARE LIBRE PERMISIVAS

**La licencia BSD:** Dan ilimitados permisos de uso, estudio y modificación privada del software, e incluye unos requisitos mínimos de redistribución.

**La licencia MIT:** Esta licencia impone muy pocas limitaciones en la reutilización.

---

## FREE SOFTWARE FOUNDATION

Se dedica a eliminar las restricciones sobre la copia, redistribución, entendimiento y modificación de programas de computadoras.

---

## SOFTWARE LIBRE

El Software libre es un software que da la libertad a los usuarios de ejecutarlo en cualquier propósito como también estudiarlo, modificarlo y distribuir el software original y las versiones adaptadas a partir de él. Los derechos de estudio y modificación del software implica que el código fuente del resultado también será accesible por otros usuarios

---

## DEFINICION Y LAS CUATRO LIBERTADES

Para que un software pertenezca al software libre deben cumplirse las 4 libertades siguientes:

- **Libertad 0:** Libertad de ejecutar el programa cuando se desea.
  - **Libertad 1:** Libertad de estudiar cómo funciona el programa y cambiarlo para que haga lo que quieras. El acceso al código fuente es obligatorio.
  - **Libertad 2:** Libertad de redistribuir copias para ayudar a su prójimo.
  - **Libertad 3:** Libertad de redistribuir copias de sus versiones modificadas a terceros.
- 

## INICIATIVA DE SOFTWARE LIBRE (Open Source)

Es el término con el que se conoce al software distribuido y desarrollado libremente.

La idea del código abierto se centra en la premisa de que al compartir el código, el programa tiende a ser de calidad superior.

El código abierto ofrece:

- **Acceso al código fuente:** Para modificarlo, corregirlo o añadir más prestaciones.
  - **Gratuidad:** El software puede obtenerse libremente.
  - La posibilidad de evitar monopolios de software propietario: Para no depender de un único fabricante de software.
  - **Un modelo de avance:** Por lo cual la información no se oculta.
- 

## DECALOGO DE CÓDIGO ABIERTO

El código abierto tiene una serie de requisitos:

- **Libre redistribución:** El software debe poder ser regalado o vendido libremente.
  - **Código fuente:** El código fuente debe estar incluido u obtenerse libremente.
  - **Trabajos derivados:** La redistribución de modificaciones debe estar permitida.
  - **Integridad del código fuente del autor:** Las licencias pueden requerir que las modificaciones sean redistribuidas solo como parches.
  - **La licencia no debe discriminar a ninguna persona o grupo:** nadie puede dejarse fuera.
  - **Sin discriminación de áreas de iniciativa:** Los usuarios comerciales no pueden ser excluidos.
  - **Distribución de la licencia:** Deben aplicarse los mismos derechos a todo el que reciba el programa
  - **La licencia no debe ser específica de un producto:** El programa no puede licenciarse solo como parte de una distribución mayor.
  - **La licencia no debe restringir otro software:** La licencia no puede obligar a que algún otro software que sea distribuido con el software abierto deba también ser de código abierto.
  - **La licencia debe ser tecnológicamente neutral:** No debe requerirse la aceptación de la licencia por medio de un acceso por clic de ratón o de otra forma específica del medio de soporte del software. Este decálogo es compatible con las cuatro libertades del software libre.
-

# INFRAESTRUCTURA DE SISTEMAS

**On-premise** e **in-house** ERP proporciona, la **mayor flexibilidad** de adaptación de **paquetes** que las necesidades organizacionales de las corporaciones tienen, pero es costoso.

Las características de las soluciones **CLOUD ERP** con herramientas de desarrollo robustas emergen rápidamente, pero en el modelo de solución **basado en cloud**, se debe estar seguro de encontrar las **necesidades buscadas** y cerciorarse de que tus **datos están a salvo**.

---

## ON-PREMISE ERP SYSTEM

Estos son **instalados** y **ejecutados** en los ordenadores dentro de la **empresa/organización**

---

## SISTEMAS ERP CLOUD

Consiste en ofrecer al usuario un **espacio virtual** en el que puede disponer de las versiones más actualizadas de hardware y software.

Existen tres tipos:

- **Infraestructura** como servicio
  - **Plataforma** como servicio
  - **Software** como servicio
- 

## Infraestructura como servicio (IaaS)

En el que el usuario **contrata** únicamente las **infraestructuras** tecnológicas sobre las que instala sus plataformas y aplicaciones. El usuario tiene el **control** total sobre las **plataformas y aplicaciones**, pero **no** tiene ningún **control** sobre las **infraestructuras**.

---

## PLATAFORMA COMO SERVICIO (PaaS)

El usuario contrata un servicio que le **permite alojar** y **desarrollar** sus propias aplicaciones en una plataforma que dispone de **herramientas de desarrollo**. El usuario **no** tiene ningún **control** sobre la **plataforma** ni sobre la **infraestructura** pero mantiene el **control** total sobre las sus **aplicaciones**.

---

## SOFTWARE COMO SERVICIO (SaaS)

El usuario **contrata** la utilización de unas determinadas **aplicaciones**, sobre las que únicamente puede ejercer acciones de configuración y parametrización permitidas por el proveedor. El usuario **no** tiene

control sobre la aplicación ni sobre la plataforma ni sobre la infraestructura .

---

## GESTOR DE PROYECTOS

El paso preliminar a cualquier desarrollo de un proyecto es elegir el gestor de proyectos que el equipo de desarrollo junto con el jefe del mismo van a utilizar para el desarrollo completo del proyecto.

---

## METODOLOGÍAS ÁGILES FRENTE A METODOLOGÍAS TRADICIONALES

Metodología tradicional:

- Planes rígidos
- Negociación contractual
- Se valoran los procesos

Metodología ágil:

- Flexibilidad ante el cambio
  - Colaboración con el cliente
  - Se valoran las personas
- 

## MÉTODOLOGÍAS ÁGILES EN LA IMPLANTACION DE SGE

Principales principios ágiles para sistemas de gestión empresarial:

- Asumir simplicidad
  - Acoger cambios con naturalidad
  - Realizar los cambios incrementalmente
  - Valorar el esfuerzo de los participantes en el proyecto
  - Plantear objetivos en las tareas de gestión
  - Aceptar las múltiples vistas del proyecto y consensuar las soluciones
  - Buscar facilidades para que la retroalimentación sea ágil
  - Tener presente la filosofía de working progress
- 

## INTRODUCCION AL MODELO SCRUM

Scrum es una metodología de desarrollo muy simple, que requiere trabajo duro porque no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto.



Se comienza con la visión general del producto, especificando y dando detalle a las funcionalidades que tienen mayor prioridad de desarrollo y que pueden llevarse a cabo en un periodo de tiempo breve normalmente de 30 días. Cada uno de estos periodos de desarrollo es una iteración que finaliza con la producción de un incremento operativo del producto.

Scrum gestiona su evolución a través de reuniones breves diarias en las que todo el equipo revisa el trabajo realizado el día anterior y el previsto para el día siguiente.

---

## CONTROL DE LA EVOLUCIÓN DEL PROYECTO

Scrum controla de forma empírica y adaptable la evolución del proyecto, empleando las siguientes prácticas de la gestión ágil:

- Revisión de las iteraciones
  - Desarrollo incremental
  - Desarrollo evolutivo
  - Auto organización
  - Colaboración
- 

## PRACTICAS DE LA GESTION ÁGIL

### Revisión de las iteraciones:

- Al finalizar cada iteración (normalmente 30 días) se lleva a cabo una revisión con todas las personas implicadas en el proyecto.

#### Desarrollo incremental:

- El desarrollo incremental implica que al final de cada iteración se dispone de una parte del producto operativa que se puede inspeccionar y evaluar.

#### Desarrollo evolutivo:

- Se intenta predecir en las fases iniciales cómo será el producto final aunque las circunstancias obligarán a remodelarlo muchas veces.
- En Scrum se toma a la inestabilidad como una premisa, y se adoptan técnicas de trabajo para permitir esa evolución sin degradar la calidad de la arquitectura que se irá generando durante el desarrollo.

#### Auto-organización:

- En Scrum los equipos son auto-organizados, con margen de decisión suficiente para tomar las decisiones que consideren oportunas.

#### Colaboración:

- Para que funcione la autoorganización como un control eficaz cada miembro del equipo debe colaborar de forma abierta con los demás, según sus capacidades y no según su rol o su puesto.
-

# ELEMENTOS DEL DESARROLLO DE SCRUM

Los elementos de Scrum son:

- **Pila del producto:** Lista de requisitos de usuario que se origina con la visión inicial del producto y va creciendo y evolucionando durante el desarrollo.
  - **Pila del sprint:** Lista de los trabajos que debe realizar el equipo durante el sprint para generar el incremento previsto.
  - **Incremento:** Resultado de cada sprint
  - **Los roles:** Scrum clasifica a las personas que intervienen en el proyecto como propietario, equipo y gestor son los responsables del proyecto, mientras que el resto de interesados.
- 

## VISION GENERALEAL DEL PROCESO

Scrum denomina “sprint” a cada iteración de desarrollo y recomienda realizarlas con duraciones de 30 días

---

# ELEMENTOS DEL DESARROLLO DE SCRUM

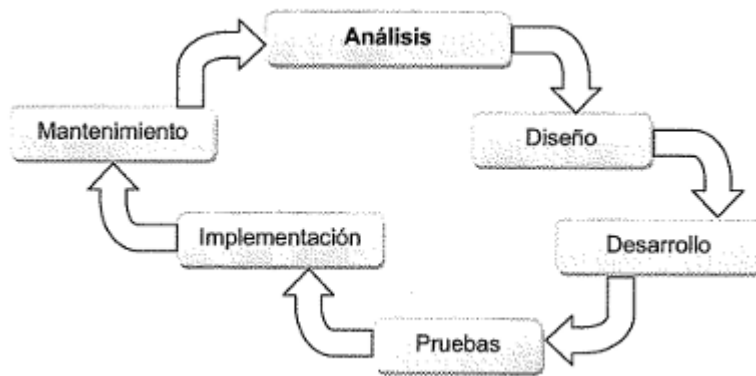
## Las reuniones

- *Planificación de sprint:* Jornada de trabajo previa al inicio de cada sprint en la que se determina cuál va a ser el trabajo y los objetivos que se deben cumplir en esa iteración.
  - *Reunión diaria:* Breve revisión del equipo del trabajo realizado hasta la fecha y la previsión para el día siguiente.
  - *Revisión de sprint:* Análisis y revisión del incremento generado.
  -
- 

## IMPLANTACION DE UN ERP

Una metodología de implantación es un proceso estructurado y metodológico para llevar a buen término el desarrollo.

Una propuesta de metodología de implantación es la adaptada al ciclo de vida clásico del desarrollo de Software.



*Ciclo de vida clásico*

## Fases de implantación:

### Fase 1: Iniciación

#### Subfases

1. Estudiar el ámbito del proyecto.
2. Realizar un estudio de viabilidad económica, técnica y organizativa.
3. Determinar el nivel de cambio del nuevo sistema con respecto al original.
4. Organizar y planear el proyecto.

### Fase 2: Desarrollo

#### Subfases:

1. Análisis detallado.
2. Diseño físico del sistema (hardware).
3. Diseño lógico del sistema (software).
4. Revisión de las previsiones.

### Fase 3: Implementación

**Subfases:**

1. Adquisición del hardware.
2. Desarrollo de software.
3. Plan de pruebas.
4. Documentación.

### Fase 5: Producción y soporte

**Subfases:**

1. Operación normal.
2. Soporte.
3. Mantenimiento.
4. Documentación al cliente.