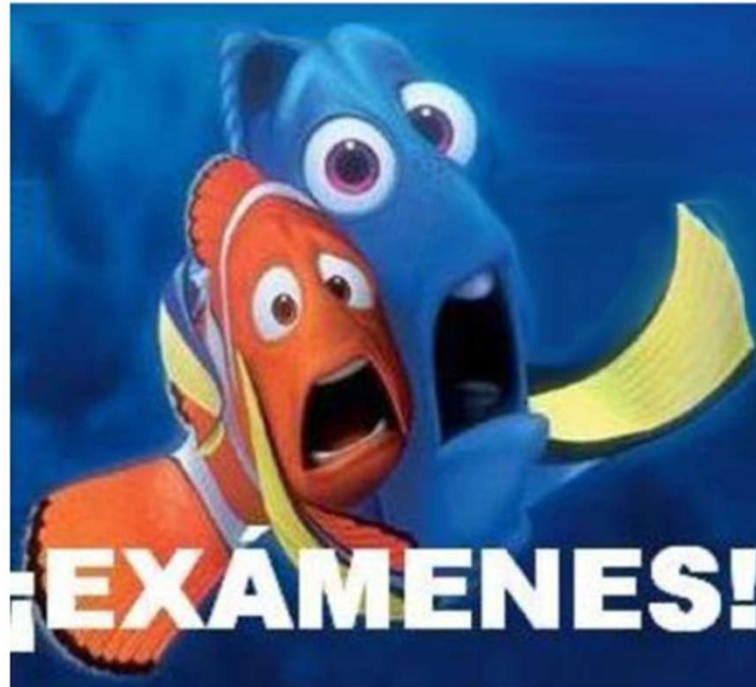


PARCIAL TIO 2024 LUNES 28 DE OCTUBRE 15HS AMBAS COMISIONES

**INCLUYE TODOS
LOS TEMAS VISTOS
EN LAS CLASES
TEÓRICAS Y
PRÁCTICAS**



**Examen
escrito
TRAER
DNI**

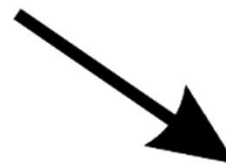
TECNOLOGIA

INFORMACION

ORGANIZACION



conjunto de
personas
con
objetivos
comunes



ELEMENTOS:
objetivos metas
RRHH
RRMATERIALES
INFORMACION

**CULTURA
ORGANIZACIONAL**

**SINERGIA
1+1=3**

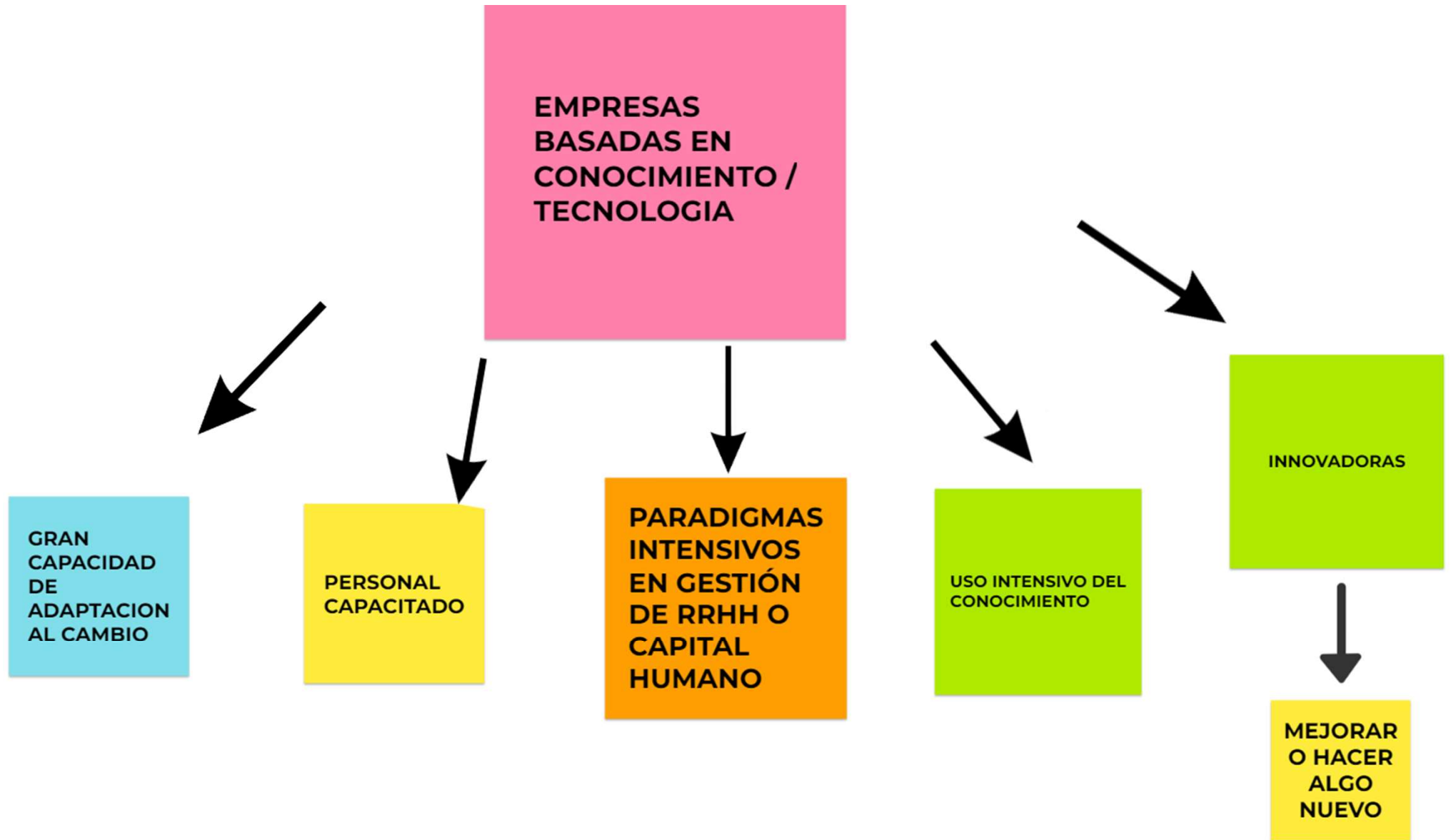
**ORGANIGRAMA
GRAFICO CON
DIVISION DEL
TRABAJO**

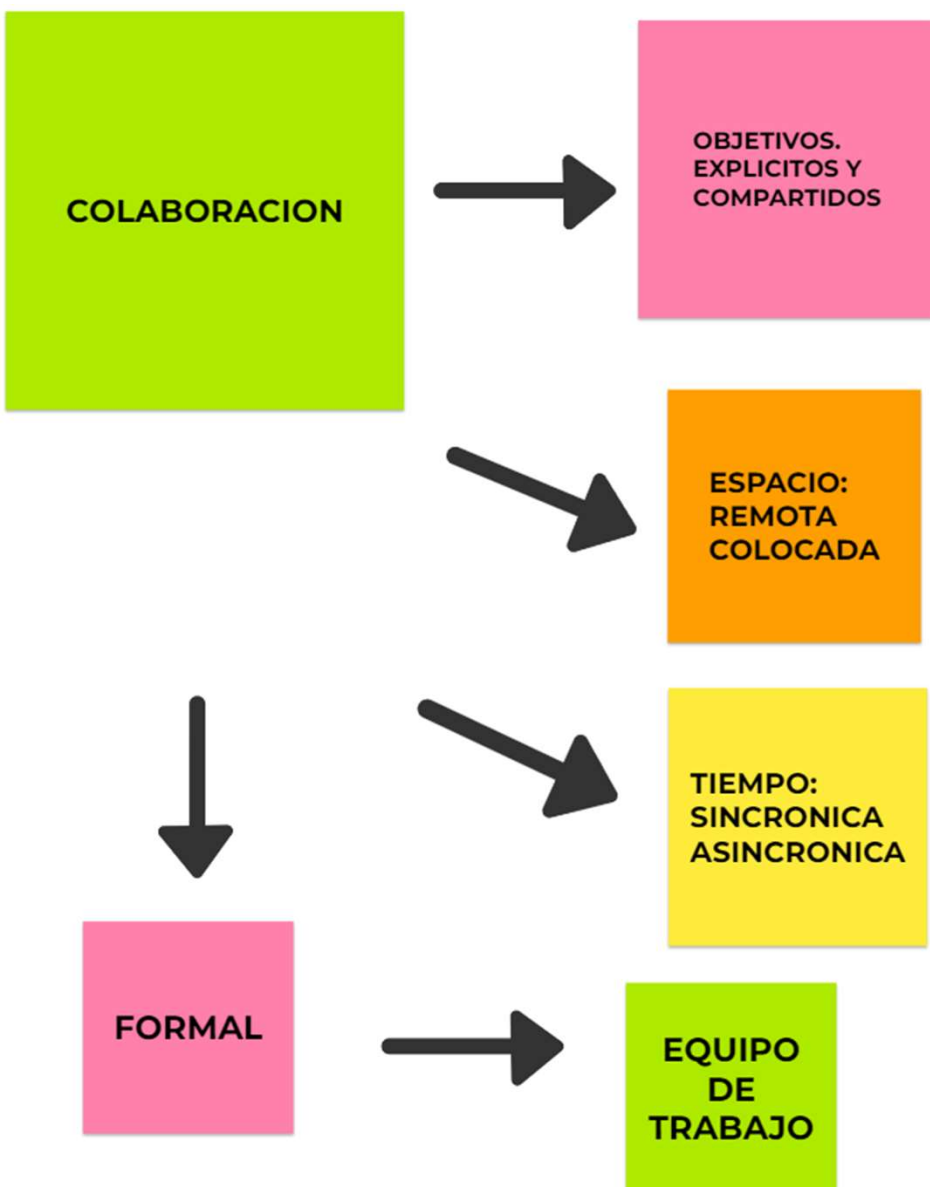
**División
del
trabajo**

**HORIZONTAL:
ESPECIALIDAD-
DIVISION DE
TAREAS**

**VERTICAL:
JERARQUIAS**





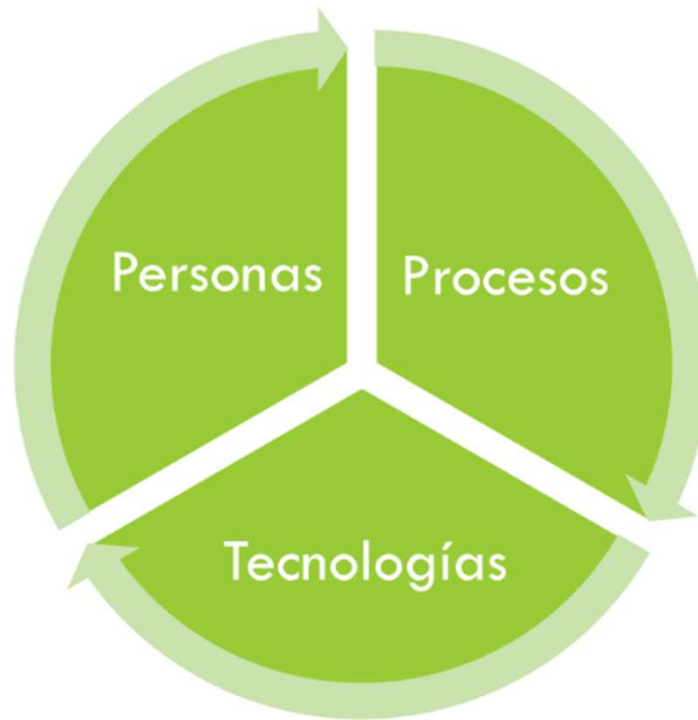


**SOFT
SKILLS**

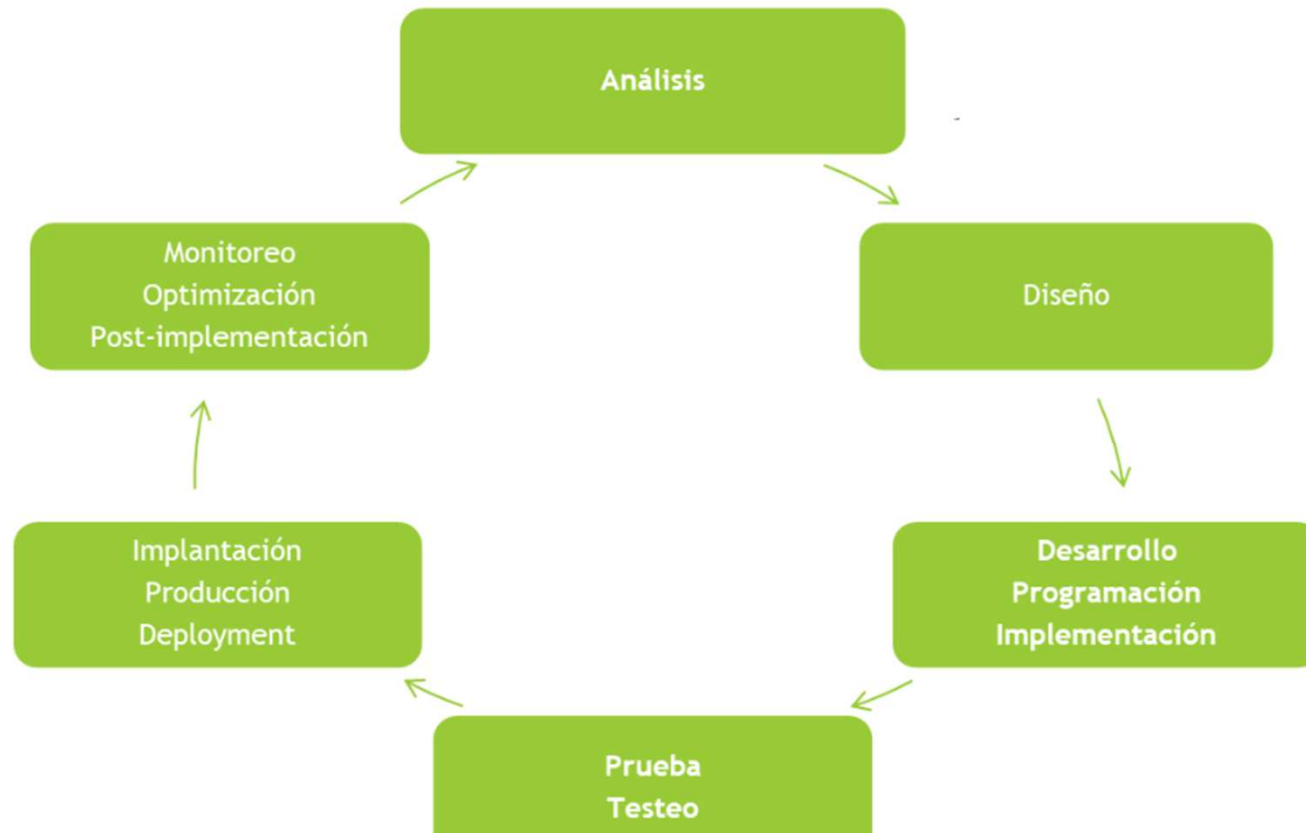
**EQUIPOS
DE
TRABAJO**

CONFLICTOS

**ELEMENTOS
CLAVES PARA
EL
DESARROLLO
DE
SOFTWARE**



CICLO DE VIDA TRADICIONAL DEL DESARROLLO DE SOFTWARE



**METODOLOGIA
CONJUNTO DE
METODOS PARA
CUBRIR CADA UNA
DE LAS
ACTIVIDADES DEL
CICLO DEL
DESARROLLO**

**METODOLOGIAS
ESTRUCTURADAS**

**METODOLOGIAS
ORIENTADAS A
OBJETOS**

Objeto como unidad básica
que encapsula datos y acciones.
Pueden comunicarse entre sí.
- son reutilizables, se agrupan
y utilizan herencia.
- Utilizado para enfoque híbrido
entre ciclo de vida tradicional
y prototipos.

RUP

1. Inicio
2. elaboración
3. construcción
4. transición

**MÉTODOS
AGILES**

-casi no genera
documentación
- Proceso no secuencial.
partes clave del desarrollo
se realizan en paralelo
- proceso de creación de
sistemas funcionales en
tiempo muy corto

**REINGENIERIA
DEL
SOFTWARE**

Metodología que ataca el problema del envejecimiento del software reciclandolo
-Se utiliza la reingeniería para extraer la información de diseño y programación de los sistemas existentes siguiendo los pasos:

Desarrollo de frameworks ->
Reusabilidad.

- Ingeniería inversa
- Modificación del diseño y especificaciones de programa.
- Ingeniería hacia adelante

**ROL:
RESPONSABILIDAD**

**FRONT
END**

BACKEND

**FULL
STACK**

DESARROLLADOR

ANALISTA

ARQUITECTO

INGENIERO



**GESTION DE
CONFIGURACION
DEL SOFTWARE**

Se refiere a las prácticas y herramientas transversales al desarrollo de software (a los requerimientos o el diseño en sí del programa)



Trazabilidad

Reproducibilidad de
releases

COMUNICACION

Interacción,
coordinación e
integración del
trabajo de los
diferentes
miembros del
equipo

ESTRATEGIAS O PRÁCTICAS PARA LA CREACIÓN DE SISTEMAS

DESARROLLO INTERNO

- tradicional
- low-code
- no-code (pensado para usuarios sin conocimientos en programación)

Contratación de Empresas de Desarrollo o Terciarización (Outsourcing)

COMPRA DE PAQUETES/LICENCIAS

Colaboración con Startups o Innovación Abierta



Saas

**BASADO
EN LA
NUBE**

**SENSACION
DE
CAUTIVIDAD
CON EL
PROVEEDOR**

**MENORES
COSTOS**

**CONEXION A
INTERNET**

**ACTUALIZACIONES
Y SOPORTE RAPIDO**

SEGURIDAD

**CLASIFICACION DEL
SOFTWARE**

```
graph TD; A[CLASIFICACION DEL SOFTWARE] --> B[SOFTWARE DE SISTEMA]; A --> C[SOFTWARE DE PROGRAMACION]; A --> D[SOFTWARE DE APLICACION];
```

The diagram illustrates the classification of software into three categories. At the top is a yellow box labeled 'CLASIFICACION DEL SOFTWARE'. Three arrows point downwards from this box to three separate colored boxes below: a blue box for 'SOFTWARE DE SISTEMA', a pink box for 'SOFTWARE DE PROGRAMACION', and an orange box for 'SOFTWARE DE APLICACION'.

**SOFTWARE
DE SISTEMA**

**SOFTWARE DE
PROGRAMACION**

**SOFTWARE
DE
APLICACION**

**SOFTWARE
DE
APLICACION**

```
graph TD; A[SOFTWARE DE APLICACION] --> B[SOFTWARE DE PRODUCTIVIDAD]; A --> C[SOFTWARE DE ESPECIALIDAD]; A --> D[SOFTWARE EDUCATIVO]; A --> E[SOFTWARE DE ENTRETENIMIENTO]; A --> F[SOFTWARE PARA LA CASA O DOMOTICA];
```

**SOFTWARE DE
PRODUCTIVIDAD**

**SOFTWARE DE
ESPECIALIDAD**

**SOFTWARE
EDUCATIVO**

**SOFTWARE DE
ENTRETENIMIENTO**

**SOFTWARE
PARA LA
CASA O
DOMOTICA**

Para licenciar a un software debemos pensar en el fin de este, si es comercial, el software no se vende "se licencia". Si es es con fines no comerciales se usa para limitar o restringir la forma de uso, modificacion y distribucion. Para proteger la propiedad intelectual o para desligarse de implicaciones legal

Patente de software

Se consideran como "expresion de una idea", por lo que está bajo el regimen de derecho de autor. Los programas de computación no son considerados invenciones. Sin embargo, los programas de computación pueden ser patentables si poseen un efecto técnico.

propietario

No hay forma de acceder al código fuente. Solamente lo conocen los creadores.

código libre

Proporciona la libertad de ejecutar el programa para cualquier proposito. Estudiar el funcionamiento del programa y adaptarlo a sus necesidades.

Redistribuir copias.

Mejorar el programa y poner estas mejoras a disposicion del público, para beneficio de toda la comunidad.

Código abierto

hablar de Open Source es hablar de software libre, pero de forma amigable a las empresas.

No plantea cuestiones éticas

La organización dedicada al código abierto es la Open Source Initiative (OSI)

CopyRight y CopyLeft

Conjunto de derechos exclusivos garantizados por un gobierno o autoridad al inventor de un nuevo producto susceptible de ser explotado industrialmente para el bien del solicitante por un periodo de tiempo limitado

copyrigh

Derecho legal sobre el software

Protege al autor del software sobre los derechos de copia, modificación y distribución no autorizada.

copyleft

Restricción sobre licencias OS

Permite la copia, modificacion y distribucion con la condicion de que se mantenga libre

Protege al software libre de convertirse en privativo propagando el mismo tipo de licencias

licencias virales