EVIDENCIA DE PRODUCTO: GA1-220501092-AA1-EV03 MAPA DE PROCESOS DEL SOFTWARE A CONSTRUIR

P4. Para la creación de mapas de procesos, existen varias herramientas TIC que pueden ayudarte a diseñar y documentar flujos de trabajo. Algunas opciones populares incluyen:
Herramientas en Línea y Software de Diagramación

1. **Microsoft Visio** – Ideal para diagramas de flujo y mapas de procesos empresariales.
2. **Lucidchart** – Herramienta basada en la nube con colaboración en tiempo real.
3. **Bizagi Modeler** – Especializado en modelado de procesos de negocio con notación BPMN.
4. **Draw.io** (diagrams.net) – Gratuita y fácil de usar para crear diagramas de flujo.
5. **ARIS Express** – Enfocado en procesos empresariales con una interfaz intuitiva.

Suites Empresariales con Modelado de Procesos

6. Adonis BPM – Plataforma profesional para la gestión de procesos de negocio.
7. Signavio – Herramienta avanzada para análisis y optimización de procesos.
8. IBM Blueworks Live – Software para la modelación y automatización de procesos.

Herramientas de Código Abierto

9. Camunda – Permite modelado y ejecución de procesos BPMN.
10. Modelio – Compatible con BPMN y UML, útil para análisis de procesos.

Si buscas algo gratuito y fácil de usar, Draw.io o Lucidchart pueden ser buenas opciones. Para proyectos empresariales, herramientas como Bizagi o Visio ofrecen funciones más avanzadas.

P9. Para establecer una **gestión de procesos** **dentro de una organización, se tienen que identificar todos los procesos que se incluirían en cada uno de los tres grandes bloques: estratégicos, operativos y de soporte. A partir de aquí, se elabora el mapa de procesos que ayudará a identificar las interrelaciones de cada uno de ellos.**
Hacer un **mapa de procesos** de la empresa es una tarea compleja que puede ser clave en la elaboración de un plan estratégico corporativo, pero ¿qué es exactamente un mapa de procesos? ¿Qué aporta a la empresa?

El mapa de procesos **es la representación gráfica**, o sea el diagrama, de la interrelación existente entre todos los procesos y subprocesos de la empresa. Dicho así, puede parecer fácil de realizar, pero la verdad es que suele ser una tarea ardua que implica a los diferentes departamentos de la empresa.

El objetivo de este mapa es conocer de forma muy detallada y profunda el funcionamiento de los procesos y actividades en los que la empresa está involucrada. Lo suele realizar un equipo con responsables de todos los departamentos, que se ocupan de identificar los procesos, tanto los principales como los secundarios.

PROCESO
Explicación Simple:
💡 "Un proceso es un conjunto de pasos que se siguen para lograr un resultado."
Así como para hacer una limonada sigues pasos (exprimir limones, agregar agua y azúcar, mezclar), en cualquier tarea o actividad hay un orden lógico de acciones para obtener un resultado final.

EL MAPA DE PROCESOS Un Mapa de Procesos es como un mapa de ruta que muestra cómo se conectan y relacionan los diferentes procesos dentro de un sistema de gestión.
En resumen, un mapa de procesos nos ayuda a ver cómo todas las partes del sistema trabajan juntas para lograr un objetivo.

Ejemplo práctico:
Aplicado al software:
Si queremos hacer una app móvil, el mapa de procesos incluiría:
1 Análisis: ¿Qué necesita el usuario?
2 Diseño: Crear bocetos de la app.
3 Desarrollo: Programar la app.
4 Pruebas: Revisar errores.
5 Lanzamiento: Publicarla y dar soporte.

P10. Mapa de procesos:

Es una guía que ayuda a entender cómo se conectan y organizan los procesos para construir un software de manera eficiente

Ejemplo: Crear una App de Chat

1 Entrada: ¿Qué necesita el usuario?
🔷 Enviar y recibir mensajes en tiempo real.
2 Procesos:
🔷 **Análisis →** ¿Qué funciones tendrá la app? (mensajes, emojis, fotos, etc.)
🔷 **Diseño →** Cómo se verá la app (colores, botones, interfaz).
🔷 **Desarrollo →** Programar las funciones (enviar mensajes, notificaciones).
🔷 **Pruebas →** Revisar y corregir errores antes de lanzarla.
3 Salida:
✅ La app de chat funcionando, lista para que los usuarios la descarguen y usen.

Explicación simple
Un mapa de procesos es como una receta de cocina para desarrollar software."
Así como una receta indica los ingredientes (entradas), los pasos a seguir (procesos) y el platillo final (salida), un mapa de procesos muestra cómo se crean las diferentes partes de un software, desde el inicio hasta el final.

P11. "Pasos para la Elaboración de un Mapa de Procesos":

Se trata de una serie de pasos que ayudan a estructurar los procesos de una organización, lo que es fundamental para su correcta documentación y optimización.

1. Identificar los procesos
Primero, se debe hacer una lista de todas las actividades y tareas que realiza la organización o el software.
➡️ Es como hacer un inventario de “todo lo que se hace” dentro del negocio.

⚪ **2. Nombrar los procesos**
Después de identificar las actividades, se les da un nombre corto y claro a cada proceso.
➡️ Por ejemplo: *Gestión de pedidos*, *Control de inventario*, *Atención al cliente*, etc.

🟡 **3. Seleccionar los procesos**
De todos los procesos identificados, se eligen solo los más importantes o los que realmente aportan valor al objetivo del proyecto.
➡️ Es decir, quedarse solo con los que ayudan a mejorar el servicio o cumplir las metas.

🔵 **4. Agrupar los procesos según su tipo**
Una vez seleccionados, se clasifican en tres grupos:

- Estratégicos: Los que guían y dirigen (ejemplo: planificación del negocio).
- Operativos: Los que producen el resultado principal (ejemplo: preparación de pedidos, gestión de ventas).
- De apoyo o soporte: Los que ayudan a que los otros funcionen bien (ejemplo: recursos humanos, mantenimiento del sistema).
  ➡️ Esto permite entender el rol que cumple cada proceso.

🟢 **5. Establecer la secuencia e interacción**
Finalmente, se ordenan los procesos en un diagrama o mapa, mostrando cómo se relacionan y en qué orden se realizan.
➡️ Así se puede ver el "camino" que sigue el trabajo dentro de la organización, desde el inicio hasta la entrega del servicio o producto.

P12.¿Cómo convertir esto en un mapa de procesos gráfico?
Para representar un **mapa de procesos** visualmente, se pueden utilizar diagramas de flujo o modelos BPMN. Una forma común es organizar los procesos en tres niveles:

- **Procesos estratégicos** (definen el rumbo de la empresa).
- **Procesos operativos** (los que generan valor directo).
- **Procesos de soporte** (apoyan la operación).

Esto puede representarse mediante **bloques o flechas** que muestren la relación y secuencia de cada proceso.

la **clasificación de los procesos dentro de una organización**. Se divide en tres categorías principales:

1. **Procesos Estratégicos**: Son aquellos que establecen la dirección de la empresa, alineados con su visión y misión. Ejemplo: planificación estratégica, gestión de calidad.
2. **Procesos Operativos**: Son los que generan valor directamente para el cliente y representan el núcleo del negocio. Ejemplo: producción, ventas, atención al cliente.
3. **Procesos de Apoyo**: Brindan soporte a los procesos operativos para garantizar su correcto funcionamiento. Ejemplo: recursos humanos, tecnología, contabilidad.

Esta clasificación es fundamental para el **diseño de un mapa de procesos** y la gestión eficiente de una organización.

P13. Crear un mapa de procesos

Los procesos de una organización se dividen en 3 grandes grupos: **procesos estratégicos**, **procesos operativos** y **procesos de soporte**. A partir de aquí, se identifican los procesos más "pequeños" que formarían cada gran grupo. La mejor forma para entenderlo es mediante un ejemplo:

"Una empresa distribuidora de bebidas refrescantes decide gestionar su empresa mediante procesos*. Para ello tiene que identificar todos los procesos en los cuales se dividirá la compañía para una gestión eficiente. A continuación, algunos ejemplos de los procesos que se encontrarían dentro de cada gran bloque":*



🔷 **Procesos Estratégicos**

Son los que **marcan la dirección y los objetivos de la empresa**.
Incluyen todas las decisiones que definen cómo se va a operar, crecer y mantenerse competitiva.
Ejemplos mostrados en la imagen:

- **Planificación estratégica:** define los objetivos y estrategias del negocio.
- **Gestión de las alianzas:** busca y mantiene relaciones con proveedores o socios.
- **Políticas empresariales:** establecen normas y procedimientos internos.
- **Políticas financieras:** controlan los recursos económicos y las inversiones.

➡️ Estos procesos ayudan a **garantizar la sostenibilidad y el crecimiento** de la empresa.

🟩 **Procesos Operativos**

Son los **procesos principales** que permiten que el producto (bebidas refrescantes) llegue al cliente.
Representan las **actividades diarias del negocio**.
Ejemplos en la imagen:

- **Gestión de las compras:** adquisición de materias primas o productos.
- **Recepción y control del producto:** verificación de calidad y cantidad de lo recibido.
- **Gestión de los pedidos:** procesamiento y distribución de pedidos a los clientes.
- **Gestión de los gastos de viaje:** control de costos del personal en desplazamiento.

➡️ Estos procesos **producen directamente el valor** que percibe el cliente.

🟠 **Procesos de Soporte o Apoyo**

Son los que **brindan el soporte necesario** para que los procesos estratégicos y operativos funcionen correctamente.
Ejemplos en la imagen:

- **Informática:** mantenimiento de sistemas y equipos.
- **Formación del personal:** capacitación continua de los empleados.
- **Mantenimiento de las instalaciones:** cuidado y reparación de infraestructura.
- **Reputación empresarial:** gestión de la imagen y comunicación de la marca.

➡️ Aunque no generan ingresos directos, **son esenciales para mantener la eficiencia** y calidad del servicio.

P14. *Actividad de aprendizaje GA1-220501092-AA2: identificar conceptos básicos de teoría general de sistemas y enfoque sistémico*

Esta actividad se centra en la identificación de fuentes de requisitos.

Duración: 12 horas.

Materiales de formación: para el desarrollo de esta actividad es importante la lectura y análisis del material de formación: “**Ingeniería de requisitos”.**

P19. Ingeniería de requisitos

La **Ingeniería de Requisitos** es una de las primeras y más importantes etapas dentro del desarrollo de software.
Su objetivo principal es **identificar, analizar, documentar y validar** las necesidades del cliente o usuario para garantizar que el producto final cumpla con sus expectativas y con los objetivos del negocio.

1 **Ciclo de vida del software:**

también conocido como (SDLC o Systems Development Life Cycle), es el proceso que se sigue para construir y hacer evolucionar un determinado software. El ciclo de vida permite iniciar una serie de fases mediante las cuales se procede a la validación y al desarrollo del software garantizando que se cumplan los requisitos para la aplicación y verificación de los procedimientos de desarrollo; para ello, se utilizan métodos del ciclo del software, que indican distintos pasos a seguir para el desarrollo de un producto.

A continuación, se indican cuáles son los elementos que integran un ciclo de vida:

Fases:
Una fase es un conjunto de actividades relacionadas con un objetivo en el desarrollo del proyecto. Se construye agrupando tareas (actividades elementales) que pueden compartir un tramo determinado del tiempo de vida de un proyecto. La agrupación temporal de tareas impone requisitos temporales correspondientes a la asignación de recursos (humanos, financieros o materiales).

Entregables:
Son los productos intermedios que generan las fases. Pueden ser materiales o inmateriales (documentos, software). Los entregables permiten evaluar la marcha del proyecto mediante comprobaciones de su adecuación o no a los requisitos funcionales y de condiciones de realización previamente establecido.

P20. 1.1 Fases

Las fases del modelo de ciclo del software son: planificación, análisis, diseño, implementación, pruebas y mantenimiento, las cuales se describen en la siguiente figura.

1.FASE PLANEACION:
En esta primera fase se realiza el planteamiento del problema, se definen alcances y objetivos del software.

2. FASE ANÁLISIS (definición de requisitos):
Esta fase busca definir los requisitos que son los que dirigirán el desarrollo del proyecto de software.
Objetivos: conocer los requisitos, asegurar que los requisitos son alcanzables, formalizar acuerdo con el cliente

3.FASE DISEÑO:
En esta fase se estudian posibles opciones de implementación para el software que hay que construir, estructura general del mismo.
Objetivos: Identificar soluciones tecnológicas, asignar recursos materiales, proponer identificar y seleccionar, establecer métodos de validación, ajustar especificaciones.

4. FASE PRUEBAS:
Esta fase busca detectar fallos cometidos en las etapas anteriores para corregirlos.
Objetivos: Realizar los ajustes necesarios para corregir posibles errores o inconsistencias.

5. FASE MANTENIMIENTO:

En esta fase se realizan tres puntos referenciados: mantenimiento correctivo, mantenimiento adaptativo y mantenimiento perfectivo.
Objetivos: Operación asegurar que el uso del proyecto es el que se pretendía, mantenimiento.

P21. Paradigmas de los modelos de ciclo de vida del software

Con la finalidad de proporcionar una metodología común entre el cliente y la empresa de software, se utilizan los modelos de ciclos de vida o paradigmas de desarrollo de software para plasmar las etapas y la documentación necesaria, de manera que cada fase se valide antes de continuar con la siguiente.

Un modelo de ciclo de vida de software es una vista de las actividades que ocurren durante el desarrollo de software e intenta determinar el orden de las etapas involucradas y los criterios de transición asociados entre estas.

Existen modelos preestablecidos con los cuales se pude elaborar un proyecto; a continuación, se mencionan los diferentes paradigmas de modelos de ciclo de vida para desarrollar software.

Paradigma tradicional
Los paradigmas tradicionales se identifican, fundamentalmente, por ser lineales, es decir se trata de completar cada proceso de principio a fin hasta que quede listo para avanzar a la segunda fase del ciclo del software.

Si bien es verdad que las metodologías actuales están basadas en lo que fueron los paradigmas tradicionales, hoy en día se ha evolucionado, sin embargo, los paradigmas tradicionales se mantienen.
Paradigma orientado a objetos:
Las etapas de desarrollo de software en el paradigma orientado a objetos, se conforma principalmente por la creación de clases, análisis de requisitos y el diseño. Con este paradigma se pretende que el código fuente sea reutilizable para otros proyectos.
Paradigma de desarrollo ágil
El objetivo de este paradigma es el desarrollo de proyectos en poco tiempo, se simplifican procesos tediosos, se agilizan las fases del desarrollo y las interacciones se hacen en corto tiempo.

Una de las principales diferencias con los paradigmas anteriores es que el cliente se ve involucrado en el proyecto durante el desarrollo de este, así, el cliente sugiere mejoras, propone ideas y se mantiene al tanto del desarrollo del producto, a diferencia del paradigma tradicional y el orientado objeto donde el cliente únicamente está al principio.

A continuación, se revisan los modelos del paradigma tradicional más utilizados.
1.1 Modelo en cascada
En su concepción básica, cada una de las actividades genera, como salidas, productos y modelos que son utilizados como entradas para el proceso subsiguiente; esto supone que una actividad debe terminarse (por lo menos, en algún grado) para empezar la siguiente.
1.2 Modelo espiral

se basa en una serie de ciclos repetitivos para ir ganando madurez en el producto final. El espiral se repite las veces que sea necesario hasta que el cliente o el usuario obtenga la satisfacción de sus necesidades.
1.3 Modelo iterativo o por prototipos
Este modelo consiste en un procedimiento que permite al equipo de desarrollo diseñar y analizar una aplicación que represente el sistema que será implementado (McCracken y Jackson, 1982).

P22. Fase de definición de requisitos

Esta fase tiene como propósito **comprender en detalle las necesidades del cliente o usuario**, los objetivos del negocio y las funcionalidades que deberá cumplir el sistema de software. Es una etapa fundamental dentro de la Ingeniería de Requisitos, ya que de ella depende la calidad y precisión de todo el desarrollo posterior.

Actividades y artefactos de la fase de definición de requisitos

Fase	Actividades	Artefactos (resultados o entregables)
Análisis (definición de requisitos)	**Definición del alcance del proyecto**	Documento que delimita lo que el sistema hará y lo que no hará.
	Identificación del negocio	Modelo del negocio (representación de los procesos actuales y necesidades del cliente).
	Toma de requerimientos	Análisis y realización de **casos de uso**, entrevistas, encuestas o reuniones con el cliente.
	Estudio de procesos de negocio	Modelo de procesos y actividades del negocio (diagramas BPMN o de flujo).
	Calendarización del proyecto	**Cronograma del proyecto**, donde se definen tiempos, responsables y etapas de ejecución.

P23. Requisitos

los requisitos comunican las expectativas de los consumidores de productos software; de otra parte, los requisitos pueden ser obvios o estar ocultos, conocidos o desconocidos, esperados o inesperados, desde el punto de vista del cliente.

- Establecen el alcance del trabajo subsecuente, pueden definir estrategias de desarrollo, riesgos, tomar decisiones de negocio (viabilidad de negocio), de proyecto (tiempo, recursos), de sistema (arquitectura).

- Indican al equipo del proyecto qué requieren los usuarios (necesidades de negocio).

- El éxito o fracaso de un proyecto está altamente influenciado por la calidad de los requisitos y el proceso para gestionarlos durante el desarrollo de un producto.

Clasificación:
1Requerimientos de usuario
Son declaraciones, en lenguaje natural y en diagramas, de los servicios que se espera que el sistema proporcione y de las restricciones bajo las cuales debe funcionar.

2Requerimientos de sistema

Estos requerimientos establecen con detalle las funciones, servicios y restricciones operativas del sistema. El documento de requerimientos del sistema deberá ser preciso, y definir exactamente lo que se va a desarrollar.

En la siguiente clasificación se observa la que se da a los requerimientos del sistema, la cual se encuentra dividida con base en lo que se va a describir, las clasificaciones son:

2.1 Requerimientos funcionales

Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que este debe reaccionar a entradas particulares; o también pueden declarar explícitamente lo que el sistema no debe hacer.

2.2 Requerimientos no funcionales

Son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Dentro de estos requerimientos se encuentra todo lo referente a la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento.

La imagen muestra una **tabla comparativa** que diferencia entre los **requisitos funcionales** y los **requisitos no funcionales** de un sistema de software.

🟧 **Requisitos Funcionales**

Son las funciones específicas que el sistema debe realizar; es decir, **qué hace** el software.

En la tabla se presentan dos ejemplos:

1. **“Se debe ingresar cédula, nombre y teléfono de cada cliente.”**
   → Indica que el sistema debe permitir registrar la información básica de los clientes.
2. **“Se requiere un listado de clientes por zona.”**
   → El sistema debe generar reportes o listados organizados por zona geográfica.

👉 Estos requisitos describen las **acciones o servicios concretos** que el sistema debe ofrecer al usuario.

🟧 **Requisitos No Funcionales**

Describen las **condiciones o características de calidad** bajo las cuales el sistema debe operar; es decir, **cómo lo hace**.

Ejemplos en la tabla:

1. **“Las consultas deben resolverse en menos de 3 segundos.”**
   → Define un parámetro de rendimiento o velocidad del sistema.

2. **“El lenguaje de programación debe ser Java.”**
   → Especifica una restricción técnica sobre la implementación.

👉 Estos requisitos aseguran que el sistema tenga **buena eficiencia, confiabilidad y cumplimiento técnico**.

🧩 **Conclusión**

La imagen permite entender la diferencia clave entre ambos tipos de requisitos:

- Los **funcionales** definen las **acciones del sistema**.
- Los **no funcionales** definen las **características de desempeño y calidad**.

Ingeniería de requisitos

El término IR “ingeniería de requisitos” ha surgido para englobar los procesos de desarrollo y gestión de requisitos en el ciclo de vida del software, el primer término (ingeniería) se enfoca en las actividades de obtención, análisis, especificación y validación de los requisitos que permitirá alcanzar los objetivos del negocio y el segundo (requisitos) está centrado en la administración de los mismos y tiene como propósito central la gestión de los cambios y la trazabilidad, de esta forma la IR proporciona el mecanismo apropiado para:

- Entender lo que el cliente quiere.
- Analizar las necesidades.
- Evaluar la factibilidad.
- Negociar una solución razonable.
- Especificar la solución sin ambigüedades.
- Validar la especificación.
- Administrar los requisitos conforme éstos se transforman en un sistema operacional.