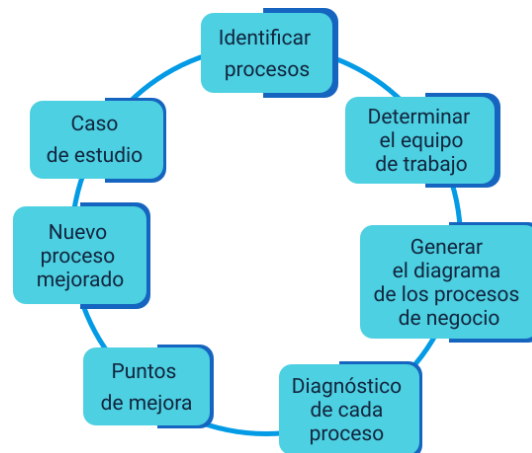


**Nombre:** Danny Julián Perilla Mikán  
**Programa:** ADSO  
**Evidencia:** GA1-220501092-AA1-EV02  
**Título:** Identificación de procesos organizacionales

### Introducción:

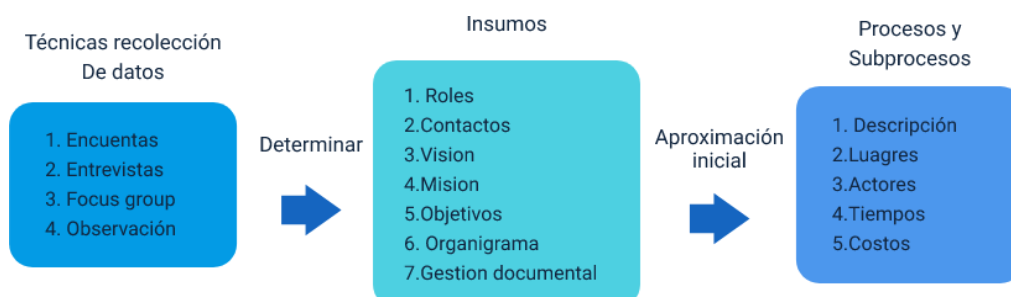
Aplicando la Teoría General de los Sistemas, identificamos el conjunto de elementos que intervienen en un equipo de desarrollo de software: las relaciones y actores responsables. En el análisis de los procesos a nivel de negocio, usamos el siguiente ciclo:



En este ciclo, vamos a desarrollar el primer paso: **identificar procesos**. Para realizar una excelente identificación de procesos se recomienda realizar las siguientes actividades en dos escenarios:

1. Antes: se debe obtener acceso de información sobre la organización, para antes de llegar a esta, se investigue sobre el contexto de negocio en el que se mueve la empresa.
2. Cuando exista cercanía, acceso o contacto con la entidad, planear entonces la recolección de información

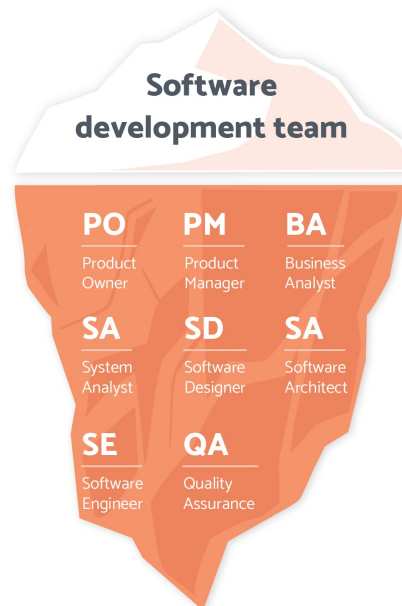
Como resultado de estas técnicas se deben establecer claramente y como mínimo lo siguientes ítems: roles de los intervinientes, información de contactos con responsabilidades, misión, visión, objetivos estratégicos, estructura organizacional un organigrama de la entidad su distribución de dependencias u oficinas en donde se puede ubicar de manera ágil a los responsables que seguramente se convertirán en futuros miembros de los equipos de trabajo donde se presentará interacción (Sinergia laboral), ver el siguiente esquema.



## Desarrollo:

Una estructura típica de equipo de desarrollo de software incluye los siguientes roles:

1. A business analyst (un analista de negocios) **BA**,
2. A product owner (un propietario del producto) **PO**,
3. A product manager (un gerente de proyecto) **PM**,
4. A software designer (un diseñador de software) **SD**,
5. A software architect (un arquitecto de software) **SA**,
6. Software engineers (ingenieros de software) **SE**,
  - a. Front-end engineers
  - b. Back-end engineers
  - c. Mobile engineers
7. Test engineer or QA engineer (ingenieros de pruebas de software) **TE/QA**.
8. DevOps Engineer (Ingeniero de DevOps).



Fuente: <https://clockwise.software/blog/software-development-team-structure/>

© Clockwise Software

### 1. Business Analyst:

- Comprende los procesos comerciales del cliente.
- Traduce las necesidades comerciales del cliente en requisitos.
- Analiza los flujos de trabajo del cliente y recopila comentarios de las partes interesadas para ayudar al cliente a definir cómo se verán sus deseos y alinear la visión del cliente con lo que está produciendo el equipo de desarrollo. Traduce una idea de producto abstracta en un conjunto de requisitos tangibles. Enriquece al equipo de desarrollo del producto con una comprensión profunda de los procesos comerciales desde varias perspectivas y la capacidad de dar forma a un producto de software que genere el máximo valor comercial.

## **2. Product Owner:**

- Responsable de la visión y evolución del producto.
- Garantiza que el producto final cumpla con los requisitos del cliente.
- Un propietario del producto tiene más responsabilidad en el éxito del producto que cualquier otro miembro del equipo de desarrollo. Equilibrando las necesidades comerciales y las tendencias del mercado, definen una estrategia comercial, dan forma a la visión del producto, aseguran que satisfaga las necesidades del cliente y gestionan una lista de productos. Asociado principalmente con entornos ágiles, el propietario del producto debe trabajar bien en escenarios donde los requisitos y flujos de trabajo cambian con frecuencia.

## **3. Product Manager:**

- Asegura que un producto o parte de él se entregue a tiempo y dentro del presupuesto.
- Gestiona y motiva al equipo de desarrollo de software.
- En proyectos secuenciales, un gerente de proyecto es responsable de distribuir tareas entre los miembros del equipo, planificar actividades de trabajo y actualizar el estado del proyecto. En proyectos ágiles, donde se enfoca en la autoadministración, la transparencia y la propiedad compartida, un gerente de proyectos establece la visión del producto, mantiene la transparencia, fomenta la comunicación, busca mejoras en el proceso de desarrollo y garantiza que el equipo entregue más valor con cada iteración.

## **4. Software Designer UX/UI:**

- Transforma la visión del producto en diseños amigables para el usuario.
- Crea recorridos del usuario para la mejor experiencia del usuario y las tasas de conversión más altas.
- Un diseñador de UX se involucra en actividades como investigación de usuarios, desarrollo de personajes, diseño de arquitectura de información, creación de wireframes, prototipos, etc. Un diseñador de UI, por su parte, diseña interfaces intuitivas, fáciles de usar y agradables a la vista. Un diseñador UX/UI te acompañaría durante todo el ciclo de desarrollo, ayudándote a alcanzar objetivos comerciales a través de experiencias de usuario funcionales y atractivas, así como analizando, evaluando y mejorando esas experiencias con el tiempo.

## **5. Software Architect:**

- Diseña la arquitectura de software de alto nivel.
- Selecciona herramientas y plataformas apropiadas para implementar la visión del producto.

- Establece estándares de calidad de código y realiza revisiones de código. Un arquitecto de software, nivel experto en ingeniería de software, toma decisiones de diseño ejecutivas en un equipo de desarrollo de aplicaciones. Si estás lidiando con un producto de software con requisitos complejos o software heredado que requiere cambios profundos, necesitarás un arquitecto de software. Un arquitecto de software decide cómo deben comunicarse los servicios y las bases de datos, cómo deben funcionar las integraciones y cómo asegurar que el producto sea seguro y estable.

## **6. Software Engineers:**

- Ingenia y estabiliza el producto.
- Resuelve cualquier problema técnico que surja durante el ciclo de desarrollo.
- Un desarrollador de software realiza el trabajo real y codifica una aplicación. Y al igual que una aplicación tiene una interfaz de usuario y una parte del servidor, hay desarrolladores de front-end y back-end. Los desarrolladores de front-end crean la parte de una aplicación con la que los usuarios interactúan, asegurándose de que la aplicación ofrezca una experiencia igualmente fluida a todos, independientemente del dispositivo, plataforma o sistema operativo. Los desarrolladores de back-end, por su parte, implementan el núcleo de una aplicación, sus algoritmos y lógica comercial.

## **7. Test Engineer or QA Engineer:**

- Asegura que una aplicación funcione según los requisitos.
- Identifica defectos funcionales y no funcionales.
- El trabajo de un ingeniero de pruebas de software es verificar si una aplicación cumple con los requisitos, tanto funcionales como no funcionales. Para verificar ambos, los ingenieros de pruebas ejecutan diversas pruebas, seguidas de analizar los resultados de las pruebas y reportar sobre la calidad de la aplicación.

## **8. Ingeniero de DevOps:**

- Facilita la cooperación entre los equipos de desarrollo y operaciones.
- Construye pipelines CI/CD para una entrega más rápida.
- Incluso en entornos ágiles, los equipos de desarrollo y operaciones pueden estar separados. Los ingenieros de DevOps sirven como un enlace entre los dos equipos, unificando y automatizando el proceso de entrega de software y ayudando a equilibrar la introducción de cambios rápidos y mantener una aplicación estable. Trabajando junto con desarrolladores de software, administradores de sistemas y personal operativo, los ingenieros de DevOps supervisan y facilitan las versiones de código sobre una base CI/CD.

Referencias:

1. <https://www.campusmvp.es/recursos/post/que-es-un-desarrollador-web-front-end-back-end-full-stack.aspx>
2. <https://itrexgroup.com/blog/software-development-team-structure/>
3. <https://clockwise.software/blog/software-development-team-structure/>