**🔹 2. Practica cómo hacer el levantamiento de requisitos**

Cuando te den el reto, sigue este esquema rápido:

**🧾 Documento: “Análisis de Requerimientos”**

**Secciones esenciales:**

1. **Nombre del proyecto:** *(ej. Sistema de gestión de pedidos web)*
2. **Descripción general del problema.**
3. **Objetivo general y específicos.**
4. **Requisitos funcionales:** (qué debe hacer el sistema)
   * RF1: El sistema debe permitir registrar usuarios.
   * RF2: El sistema debe generar reportes de ventas.
5. **Requisitos no funcionales:** (rendimiento, seguridad, usabilidad, etc.)
6. **Alcance:** Qué cubrirá y qué no cubrirá.
7. **Tecnologías a usar:** Node.js, MySQL, React, etc.

👉 Practica redactar esto en formato limpio, con viñetas, sin frases largas.

**🧩 OBJETIVO DEL LEVANTAMIENTO DE REQUISITOS**

👉 Obtener una **comprensión clara, precisa y verificable** de lo que el sistema debe hacer y cómo debe comportarse.  
Esto se traduce en **documentos y diagramas** que describen:

* Qué necesita el cliente.
* Qué hace el sistema.
* Qué límites o condiciones existen.

En SENASoft, esto demuestra **pensamiento analítico, comunicación técnica y organización.**

**🧱 ESTRUCTURA DEL DOCUMENTO “Análisis de Requerimientos”**

Lo ideal es usar un formato tipo informe o ficha técnica (Word).  
Aquí va la **estructura con ejemplos reales** 👇

**🔹 1. Información general del proyecto**

**Ejemplo:**

Nombre del proyecto:\*\* Sistema Web de Gestión de Pedidos\*\*

Competencia:\*\* SENASoft 2025 – Desarrollo Libre\*\*

Integrantes:\*\* Charles (Analista / Arquitecto), Andrés (Desarrollador)

Fecha:\*\* 21 de octubre de 2025

**🔹 2. Descripción general del problema**

Explica en tus palabras qué problema se busca solucionar.

**Ejemplo:**

Actualmente, la empresa “X” maneja sus pedidos y clientes de forma manual mediante hojas de cálculo, lo que genera errores y pérdida de información.  
Se requiere una aplicación web que permita registrar clientes, gestionar pedidos, calcular totales y generar reportes.

💡 Consejo: escribe en **una o dos frases claras**, sin tecnicismos.

**🔹 3. Objetivo general y específicos**

**Objetivo general:**

Desarrollar una aplicación web que permita gestionar clientes y pedidos de manera eficiente, reduciendo errores y mejorando la trazabilidad de la información.

**Objetivos específicos:**

* Permitir el registro, consulta y edición de clientes.
* Permitir la creación y seguimiento de pedidos.
* Generar reportes de ventas filtrados por fechas o clientes.
* Controlar accesos mediante autenticación de usuarios.

💡 Aquí demuestras que entendiste el alcance completo.

**🔹 4. Alcance del proyecto**

El sistema cubrirá la gestión de usuarios, clientes, pedidos y reportes.  
No incluirá pasarelas de pago ni notificaciones automáticas (fuera del alcance).

💡 Define lo que **sí** y **no** hará el sistema.  
Esto evita malentendidos con los evaluadores.

**🔹 5. Requisitos funcionales (RF)**

Estos describen **lo que el sistema hace** (acciones o funciones).

🧾 Ejemplo (usa numeración clara):

| **ID** | **Requisito funcional** | **Descripción** |
| --- | --- | --- |
| RF1 | Registro de usuarios | El sistema permitirá registrar nuevos usuarios con nombre, correo y contraseña. |
| RF2 | Inicio de sesión | El sistema validará credenciales antes de permitir el acceso. |
| RF3 | Registro de clientes | El sistema permitirá crear, modificar y eliminar clientes. |
| RF4 | Gestión de pedidos | El usuario podrá crear un pedido con múltiples productos. |
| RF5 | Reportes | El sistema generará reportes de ventas filtrados por fecha. |

💡 Clave: que sean **verificables, numerados y claros** (acción + resultado esperado).

**🔹 6. Requisitos no funcionales (RNF)**

Estos definen **cómo** debe comportarse el sistema (rendimiento, seguridad, compatibilidad, etc.).

| **ID** | **Requisito no funcional** | **Descripción** |
| --- | --- | --- |
| RNF1 | Usabilidad | La interfaz será intuitiva y responsive. |
| RNF2 | Rendimiento | Las consultas no deben tardar más de 2 segundos. |
| RNF3 | Seguridad | Las contraseñas serán almacenadas cifradas. |
| RNF4 | Compatibilidad | Compatible con navegadores modernos (Chrome, Edge, Firefox). |
| RNF5 | Mantenibilidad | El código seguirá una estructura modular y comentada. |

💡 Usa siempre verbos medibles (“será”, “deberá”) en lugar de “debería”.

**🔹 7. Reglas de negocio**

Son condiciones o restricciones que vienen del funcionamiento real.

**Ejemplos:**

* No se pueden registrar dos usuarios con el mismo correo.
* Un pedido solo puede crearse si el cliente está activo.
* Los reportes de ventas deben mostrar totales con IVA incluido.

💡 Esto lo puedes deducir del contexto o inventar lógicamente en la competencia.

**🔹 8. Casos de uso (resumen textual)**

Antes de dibujar los diagramas UML, define brevemente los casos principales.

**Ejemplo:**

| **ID** | **Caso de uso** | **Actor** | **Descripción** |
| --- | --- | --- | --- |
| CU1 | Registrar cliente | Administrador | Permite agregar un nuevo cliente al sistema. |
| CU2 | Crear pedido | Vendedor | Registra un pedido con productos asociados. |
| CU3 | Generar reporte | Administrador | Muestra estadísticas de ventas por rango de fechas. |

💡 Este cuadro te servirá como guía para el **diagrama de casos de uso UML**.

**🔹 9. Tecnologías propuestas**

**Front-end:** React.js, Tailwind CSS  
**Back-end:** Node.js, Express  
**Base de datos:** MySQL  
**Control de versiones:** GitHub  
**Diseño UX/UI:** Figma  
**Documentación:** Word / Draw.io / Lucidchart

💡 Esto debe alinearse con lo permitido por el documento de SENASoft.

**🔹 10. Glosario de términos (opcional)**

Si usas términos técnicos, explica brevemente qué significan:

* **API REST:** Interfaz que permite la comunicación entre frontend y backend mediante peticiones HTTP.
* **CRUD:** Conjunto de operaciones básicas: Crear, Leer, Actualizar, Eliminar.
* **Mockup:** Boceto visual de una pantalla o interfaz.

**🧠 TÉCNICAS PARA LEVANTAR REQUISITOS (simulando entrevista con cliente)**

Aunque en SENASoft no hay un cliente real, pueden darte un enunciado del tipo:

“Una empresa desea un sistema para gestionar sus productos y empleados.”

Tú debes:

1. **Identificar los actores.**
   * ¿Quiénes usarán el sistema? (ej. admin, empleado, cliente)
2. **Preguntarte:**
   * ¿Qué tareas realiza cada uno?
   * ¿Qué datos necesita manejar?
   * ¿Qué problema principal intenta resolver?
3. **Traducir las respuestas en requisitos.**

💬 Ejemplo mental:

“El cliente necesita ver reportes.” → RF: El sistema generará reportes por fecha y categoría.

**📄 ESTRUCTURA LISTA PARA COPIAR (resumen tipo plantilla)**

# Análisis de Requerimientos

## 1. Información general

## 2. Descripción del problema

## 3. Objetivo general y específicos

## 4. Alcance

## 5. Requisitos funcionales

## 6. Requisitos no funcionales

## 7. Reglas de negocio

## 8. Casos de uso (resumen)

## 9. Tecnologías propuestas

## 10. Glosario (opcional)