Perfecto 🔥 — excelente que te prepares desde ya, eso te dará una ventaja enorme.  
Como el **Día 1 de SENASoft (Desarrollo Libre)** se enfoca en **análisis, planeación y diseño de la solución**, vamos a organizarte una **preparación paso a paso**, con ejemplos, herramientas y recursos que puedes dejar listos **antes de llegar a la competencia**.

**🧭 OBJETIVO DEL DÍA 1 (según el documento)**

📘 **“Análisis de la información y planeación del proyecto. Diseño de la solución propuesta.”**  
➡ En esta etapa, **no se programa todavía**, se entiende el problema, se diseña la solución, y se deja todo listo para comenzar a codificar al día siguiente.

**🧩 TU ROL PRINCIPAL (Analista / Arquitecto / Documentador)**

Te enfocarás en:

1. **Entender el problema y definir requisitos.**
2. **Diseñar diagramas UML.**
3. **Modelar la base de datos.**
4. **Diseñar UX/UI (mockups o wireframes).**
5. **Redactar la documentación inicial.**

**🧠 PLAN DE PREPARACIÓN (antes del día 1)**

**🔹 1. Prepara tu entorno de trabajo**

Asegúrate de tener instaladas:

* **Visual Studio Code**
* **XAMPP (por si piden conexión a MySQL)**
* **MySQL Workbench**
* **Figma** o **Draw.io / Diagrams.net**
* **Lucidchart** o **StarUML** (para diagramas UML)
* **Microsoft Word o Google Docs**

📦 Crea una carpeta con estructura base:

SENASOFT2025/

│

├── docs/

│ ├── Analisis\_Requerimientos.docx

│ ├── Diagramas\_UML/

│ └── Mockups/

│

├── backend/

├── frontend/

└── README.md

**🔹 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.

**🔹 3. Diagramas UML esenciales**

Durante la competencia te pedirán evidencias visuales del análisis.  
Prepara plantillas o practica estos 3:

**🔸 Diagrama de casos de uso (Draw.io)**

* Identifica actores (Usuario, Administrador, Sistema)
* Relaciona con funciones del sistema (Registrar, Consultar, Editar)  
  💡 *Ejemplo: un actor “Cliente” realiza “Hacer pedido”, “Ver historial”, etc.*

**🔸 Diagrama de clases**

* Define las entidades principales (Usuario, Producto, Pedido)
* Relación: 1 a muchos, herencia, composición  
  💡 Usa nombres sencillos y atributos base.

**🔸 Diagrama de secuencia**

* Representa el flujo entre usuario, interfaz, servidor y base de datos.  
  💡 *Ejemplo: Usuario → Frontend → Backend → DB → Respuesta.*

**🔹 4. Modelo de base de datos**

Con **MySQL Workbench** crea o practica un modelo E-R (Entidad-Relación):

Ejemplo rápido:

Usuario (id\_usuario, nombre, correo, contraseña)

Pedido (id\_pedido, id\_usuario, fecha, total)

Detalle\_Pedido (id\_detalle, id\_pedido, producto, cantidad)

💡 *Asegúrate de incluir claves primarias (PK) y foráneas (FK).*

**🔹 5. Diseño de interfaces (UX/UI)**

Usa **Figma** o **Draw.io** para preparar mockups:

📱 Pantallas típicas:

* Login / Registro
* Panel principal
* Formularios
* Listados o reportes

💡 Consejo: no te demores mucho en el color o diseño, céntrate en **la estructura lógica** y **la experiencia del usuario**.

**🔹 6. Documentación base**

Guarda un documento Word llamado **“Diseño de la Solución”**, con este esquema:

**1. Portada**  
**2. Análisis de requerimientos (resumen del punto 2)**  
**3. Diagramas UML**  
**4. Modelo de base de datos (imagen)**  
**5. Mockups UX/UI**  
**6. Propuesta de arquitectura (MVC, API REST)**

**🧩 CHECKLIST PARA MAÑANA**

| **Tarea** | **Estado** |
| --- | --- |
| Instalar herramientas necesarias | RR |
| Crear estructura de carpetas | -- |
| Documento de requerimientos preparado | RR |
| Plantillas UML abiertas (Draw.io o StarUML) | RR |
| Modelo base de datos lista para editar | R |
| Figma o Draw.io abierto con pantallas base | R |
| Plantilla Word “Diseño de la Solución” lista | -Rr |

**🗣️ CONSEJO FINAL**

El **día 1** no se gana por velocidad, sino por **claridad y orden**.  
Tu objetivo es que el jurado vea que:

* Entendiste el problema mejor que otros equipos.
* Tienes una arquitectura clara y coherente.
* Tu equipo tiene una ruta lógica de desarrollo.

💬 Si tú dejas eso sólido, tu compañero podrá desarrollar el código mucho más rápido y sin errores en los siguientes días.