**Identificación del problema y análisis de requerimientos**

**Caso de Estudio : Cafeteria**

|  |  |
| --- | --- |
| **Cliente** | **Don Alonso** |
| **Usuario** | Usuarios de la cafetería de Don Alonso |
| **Contexto del problema** | *Don Alonso ha decidido abrir una cafetería en su barrio. Quiere ofrecer bebidas de café personalizables y modernizar su negocio mediante un menú interactivo. La idea es que el sistema pueda gestionar hasta 10 bebidas predefinidas. Al acceder al menú, los usuarios pueden seleccionar el número de bebidas a pedir y personalizar cada bebida según sus preferencias, incluyendo el tipo de café, tipo de leche, nivel de azúcar y si desean agregar crema donde también se debe mostrar el precio total del pedido.* |
| **Requerimientos funcionales** | RF1: Ordenar bebida. |
| **Requerimientos no funcionales** | Usabilidad  Escalabilidad  Rendimiento  Interfaz fácil de usar |

|  |  |  |  |
| --- | --- | --- | --- |
| Identificador y nombre | *[RF1-Ordenar bebida]* | | |
| Resumen | *el usuario podra personalizar su cafe a la hora de ordenar.* | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | **Condición valores válidos** |
| numeroDeBebidas | Int | *enteros entre [1-10]* |
| tipoDeCafe | Enum | campo valido (“Orgánico” o “tradicional”) |
|  | tipoDeLeche | Enum | Campo valido  (“Almendras”, ”Deslactosada”, ”Entera” o ”Ninguna”) |
|  | nivelDeAzúcar | Enum | Campo valido  (“Sin azúcar”, ” Neutro”, ”Dulce” o “ninguna”) |
|  | agregarCrema | Boolean | True = Agregar  False = No agregar |
| Resultado o Postcondición | Personalización con éxito | | |
| Salidas | **Nombre salida** | **Tipo de dato** | **Formato** |
| confirmacionDelPedido | String | Mensaje con especificaciones de la bebida |
| precioTotal | Double | Valor precio final |
|  |  |  |