# Identificación del problema y análisis de requerimientos

## Caso de Estudio:

|  |  |
| --- | --- |
| Cliente | Don Alonso |
| Usuario | Cliente |
| Contexto del problema | Don Alonso es un apasionado del café que ha decidido montar una cafetería en su barrio, esto con el objetivo de traer la promesa del café artesanal a los lugares más cotidianos. Para esto don Alonso ha construido un menú que ofrece distintos tipos de bebidas de café y a su vez distintos tipos de ingredientes cuando la bebida lo requiere.  A pesar de tener una visión tradicional, don Alonso también apoya las visiones modernas y quiere integrar una herramienta tecnológica a su negocio, en este caso está pensando en la oportunidad de que su negocio tenga un menú interactivo en el que el usuario pueda ordenar su pedido directamente.  Como lo que requiere don Alonso se trata apenas de un prototipo, la idea es que el sistema pueda guardar 10 bebidas. (Bebidas que ya van a estar disponibles cuando el usuario acceda al menú) |
| Requerimientos funcionales | RF1 Seleccionar el número de bebidas a pedir  RF2 Ordenar la bebida  RF3 calcular el total del pedido |
| Requerimientos no funcionales | Los requerimientos no funcionales del sistema para la cafetería de Don Alonso podrían ser los siguientes:  1. Rendimiento\*\*: El sistema debe ser capaz de manejar múltiples pedidos simultáneamente sin ralentizaciones o bloqueos.  2. Disponibilidad\*\*: El sistema debe estar disponible durante el horario de trabajo de la cafetería.  3.Usabilidad\*\*: El sistema debe ser fácil de usar para los clientes, con una interfaz intuitiva y amigable.  4. \*\*Seguridad\*\*: Los datos del cliente deben ser almacenados de forma segura y no deben ser accesibles a terceros no autorizados.  5. \*\*Compatibilidad\*\*: El sistema debe ser compatible con varios dispositivos y navegadores para permitir a los clientes realizar pedidos desde su dispositivo preferido.  6. \*\*Escalabilidad\*\*: Aunque inicialmente el sistema sólo necesita almacenar 10 bebidas, debe ser capaz de escalar para almacenar más elementos en el menú en el futuro.  7. Mantenibilidad: El sistema debe ser fácil de mantener y actualizar, permitiendo la adición de nuevas características o la modificación de las existentes con el mínimo esfuerzo.  8. Fiabilidad: El sistema debe ser fiable y preciso al calcular el precio total del pedido del cliente.  Estos son solo algunos ejemplos y podrían haber más requerimientos no funcionales dependiendo de las necesidades específicas del negocio de Don Alonso. |
|  |  |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador y nombre | *[RF1-añadir bebida]* | | | |
| Resumen | *Dentro de la aplicación estará la función principal, que cumple con que el usuario pueda ordenar su pedido directamente desde el programa indicando entradas como:*   * *Seleccionar el número de bebidas a pedir* * *Organizar la bebida, para esto el usuario puede seleccionar el tipo de café: ORGÁNICO, TRADICIONAL; El tipo de leche (si el usuario lo desea con este ingrediente) ALMENDRAS, DESLACTOSADA, ENTERA; El nivel de azúcar con el que lo desea: SIN AZÚCAR, NEUTRO, DULCE. Finalmente, si desea agregar crema o no.* | | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | | **Condición valores válidos** |
| bebidasCantidad | int | | *Valido en un rango de 1..10* |
| tipoCafe | Enum | | El usuario solo puede elegir entre : ORGANICO Y TRADICIONAL |
|  | tipoLeche | Enum | | El usuario solo puede elegir entre:  *ALMENDRAS, DESLACTOSADA, ENTERA* |
|  | nivelAzucar | Enum | | El usuario solo puede elegir entre:  *SIN AZÚCAR, NEUTRO, DULCE.* |
|  | crema | boolean | | El usuario solo puede elegir entre:  *si desea agregar crema o no* |
| Resultado o Postcondición | La bebida será registrada con los datos ingresada por el usuario, y dependiendo de la cantidad de estas, se calculara el precio. | | | |
| Salidas | **Nombre salida** | | **Tipo de dato** | **Formato** |
| mensaje | | toString() | “su bebida ha sido registrada” + (bebida) |
|
|
|
|