# Identificación del problema y análisis de requerimientos

## Caso de Estudio: Cafetería Alonso

|  |  |
| --- | --- |
| Cliente | Alonso |
| Usuario | Clientes de la cafetería de Alonso |
| Contexto del problema | *Don Alonso desea que su cafetería cuente con un software capaz de ofrecer distintos tipos de café en su menú interactivo, con el cual el usuario pueda hacer su pedido. Don Alonso necesita que se puedan guardar 10 bebidas, por lo cual el sistema debe ser capaz de seleccionar el tipo de café, el tipo de leche, el nivel de azúcar y si desea agregar crema o no. Finalmente, se debería calcular el precio total del pedido.* |
| Requerimientos funcionales | RF1- El sistema permite ordenar una bebida  RF2- El sistema permite guardar las bebidas  RF3- El sistema permite consultar el precio total |
| Requerimientos no funcionales | RNF1- El sistema debe poder almacenar 10 bebidas  RNF2- El sistema debe tener un menú interactivo |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador y nombre | *RF1- El sistema permite ordenar una bebida* | | | |
| Resumen | *El sistema pide al usuario el precio de la bebida, el tipo de café, tipo de leche, tipo de azúcar ry si desea agregar crema a su bebida o no. Finalmente, si el proceso ha sido exitoso el sistema retornara si la bebida fue guardada o no.* | | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | | **Condición valores válidos** |
| precio | double | | *No puede ser una cantidad nula* |
| tipo\_cafe | tipo\_cafe | | *Es un dato de tipo enumeración y el usuario puede elegir entre café orgánico o café tradicional* |
| tipo\_leche | tipo\_leche | | *Dato de tipo enumeración y el usuario puede elegir entre leche de almendras, deslactosada o entera* |
| tipo\_azucar | tipo\_azucar | | *Dato de tipo enumeración y el usuario elige si desea azúcar, neutro o dulce* |
| crema | boolean | | *El usuario elige si desea crema o no* |
| Resultado o Postcondición | El registro de la bebida fue exitoso | | | |
| Salidas | **Nombre salida** | | **Tipo de dato** | **Formato** |
| registro | | boolean | *Se devuelve un true o false en el método y si el resultado es true entonces devuelve un mensaje diciendo que el proceso se realizó con éxito* |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador y nombre | *RF2- El sistema permite guardar las bebidas* | | | |
| Resumen | *Se ingresa un dato de tipo café como objeto y se almacena en un array de 10 posiciones, si el registro fue exitoso se comunicara al usuario con un mensaje el éxito de su proceso.* | | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | | **Condición valores válidos** |
| cafe | Cafe | | *[Condición de la entrada. Ej. números enteros como mínimo 5 dígitos y máximo 15]* |
| Resultado o Postcondición | La bebida se almacenó con éxito | | | |
| Salidas | **Nombre salida** | | **Tipo de dato** | **Formato** |
| registro | | boolean | *Se devuelve un true o false en el método y si el resultado es true entonces devuelve un mensaje diciendo que el proceso se realizó con éxito* |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador y nombre | *RF3- El sistema permite consultar el precio total* | | | |
| Resumen | *EL sistema recibe el precio de cada producto que compro el cliente para finalmente imprimir cual es la cantidad total a pagar* | | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | | **Condición valores válidos** |
| precio | double | | *No debe ser una cantidad negativa* |
| Resultado o Postcondición | El precio total se calculó de manera exitosa | | | |
| Salidas | **Nombre salida** | | **Tipo de dato** | **Formato** |
| precio\_total | | double | *Al terminar de sumar todos los precios, el método retorna el precio total* |