

## Objetivos

### Unidad 1: Pruebas Automáticas y Tipos de Excepción

La actividad planteada en esta hoja de trabajo contribuye al desarrollo del siguiente objetivo específico:

- OE1.1 Reconocer el mecanismo de manejo de excepciones señalando las implicaciones de la propagación versus el control.
- OE1.2 Usar e implementar distintos tipos de excepción como parte de un programa, de manera que sea posible clasificar los tipos de error que se pueden presentar y asociarles en el programa distintas maneras de recuperarse ante el problema.
- OE1.3 Diseñar pruebas unitarias automáticas que permitan validar el adecuado funcionamiento de las operaciones del sistema desarrolladas para soportar los requerimientos funcionales.
- OE1.4 Desarrollar las clases y los métodos necesarios para implementar las pruebas unitarias automáticas, que ayudan a comprobar el correcto funcionamiento de un programa.

## Enunciado

El administrador del minimercado *Mi Barrio Te Quiere* necesita un programa que permita registrar el documento de las personas que ingresan a su establecimiento cada día. **El programa debe tener un menú muy sencillo con tres opciones, la primera debe permitir registrar el ingreso de una nueva persona, la segunda debe permitir consultar la cantidad de personas que han *intentado* ingresar al minimercado y la tercera debe permitir salir del programa.**

Registrar el ingreso de una persona implica ingresar únicamente el **tipo** de documento de identificación (que puede ser: TI -Tarjeta de Identidad-, CC -Cédula de Ciudadanía-, PP -Pasaporte- o CE -Cédula de Extranjería-) y el **número** del documento de identificación. De acuerdo con la última normativa del gobierno local, las personas cuyo documento tiene el **penúltimo** número par, pueden salir si el día del mes es un número impar, y viceversa. Para ingresar, la persona debe tener un tipo de documento que no sea TI, pues debe ser mayor de edad.

El programa debe registrar solamente la información de tipo y número del documento de cada una de las personas que finalmente pudieron ingresar (porque cumplieron con las condiciones). De las personas que no cumplieron con las condiciones para el ingreso no se deben almacenar esos datos. En cambio, si se debe almacenar la **cantidad** total de personas que intentaron ingresar (incluyendo las que se pudieron registrar y las que no). Por tanto, cada vez que se intenta registrar a alguien, sea exitoso o no el ingreso, se debe aumentar el contador de personas que intentaron ingresar al minimercado.

El programa debe manejar como una Excepción, cada una de las dos (2) condiciones que impiden que una persona pueda ingresar al minimercado.

Debe diseñar y probar únicamente el método del modelo encargado de registrar el ingreso de una persona. Es una sola prueba, pero debe haber varios casos para dicha prueba, incluyendo los casos en que (1) se puede registrar correctamente, (2) no puede registrar porque es un menor de edad, y (3) no se puede registrar porque no le corresponde el día de acuerdo con el número de su cédula y el día del mes.

Usted debe entregar los siguientes artefactos y condiciones:

1. **[25%]** Diseño completo del diagrama de clases, incluyendo el paquete de excepciones y pruebas.
2. **[15%]** Diseño de todos los casos de la única prueba, incluyendo los escenarios.
3. **[50%]** Implementación completa y correcta del modelo, las excepciones, la ui y las pruebas.
4. **[10%]** Usted debe entregar el enlace del repositorio en GitHub o GitLab con los elementos anteriores. Su repositorio debe tener al menos 3 commits con diferencia de 1 hora entre cada uno de ellos.

**Importante:** su repositorio debe ser privado hasta la hora y fecha de entrega, después de la cual usted debe hacerlo privado para que pueda ser revisado.

La rúbrica con la que se evaluará esta prueba se encuentra en el [listado de notas de seguimientos](#) en la pestaña SegSem2.