Universidad Tecnológica Nacional
Facultad Regional Avellaneda

1° Parcial Lab II - 2°D - 2°Cuatri 2023

Características de la fábrica

Nos han encomendado la tarea de diseñar un sistema en .NET utilizando Windows Forms para poder administrar un negocio

Desarrollar funcionalidades de un sistema para una fabrica de produccion:

- Se deberá poder agregar productos (al menos 2 diferentes).

- Darle el enfoque que crean correcto, teniendo en cuenta que el sistema tiene que describir algún proceso de dicha fábrica (no compras, no ventas, fabricación).

- En la fábrica trabajan empleados y supervisores.

Requerimientos del sistema

1. **Administrar usuarios**: debe poder diferenciar un operario de un supervisor.

    a. Operario.

        i. El operario deberá poder gestionar la línea de producción.

    b. Supervisor.

        i. Deberá poder realizar las mismas tareas que el operario.
        ii. Podrá acceder a ver la información de todos los operarios registrados en la aplicación.
        iii. Agregar nuevos productos al inventario y rellenar el stock de materia prima nuevamente.

SE TOMO ACCION DEJANDO QUE EL OPERARIO PUEDA PRODUCCIR Y COCINAR SEGÚN CORRESPONDA EN EL PROGRAMA Y EL ADMINISTRADOR PUEDE SETEAR MATERIA PRIMA INGRESANDO CON METODOS A LA CLASE ESTATICA ENCARGADA DE ESTO QUE SE LLAMA STOCK, TAMBIEN PUDIENDO LISTAR LOS USUARIOS QUE ESTEN REGISTRADOS EN LA APLIACION (HARCODE 4 PARA QUE SE VEA LA FUNCIONALIDAD)

2. **Administrar el inventario**:
    a. El sistema deberá poder informar a quien esté logueado como viene de materia prima la fábrica. Debe tener alguna sección donde se pueda ver el estado del stock y se pueda notar de forma ágil que algún material está por acabarse.

LOS USUARIOS TIENEN UN ID UNICO PARA OPERARIO ES OP+DNI Y PARA SUPERVISOR ES SU+DNI ES DECIR NO PUEDE HABER DOS USUARIOS IGUALES.

A tener en cuenta

- *La aplicación deberá contar con una interfaz clara y operativa. Deberá ser lo más profesional posible.*

Condiciones de aprobación

1. *Los 2 proyectos (Biblioteca de clases y Formularios) deben cumplir las reglas de estilo:*
    - *Código correctamente comentado con Summary*
    - *Nombres de controles con los estilos correspondientes.*
    - *Los formularios no pueden tener ni color ni nombre ni icono por defecto.*

2. *Deben estar presentes las funcionalidades requeridas anteriormente.*

3. *La aplicación debe ser intuitiva y debe además estar distribuida en secciones de fácil navegación y acceso. Deben haber al menos 3 formularios diferentes.*

CONTIENE 4 FORMS (SUPERVISOR ---LOGIN---OPERARIO---COCINA)

4. *Deben ser utilizados todos los temas mencionados a continuación:*
    - *Herencia*
    - *Sobrecarga de constructores, ¿operadores y métodos*
    - *Propriedades e Indexadores*
    - *Colecciones (al menos 2 colecciones diferentes)*
    - *Enumerados*
    - *Conversiones Implícitas y Explícitas*
    - *Formularios modal.*
    - *Clases estáticas*
    - *Polimorfismo (clases abstractas, métodos abstractos, métodos virtuales*

5. *No debe generar ningún tipo de error (ni de compilación, ni de ejecución).*

6. *Se deben respetar los 4 pilares de POO.*

7. *Deben existir botones que completen los datos de un supervisor o de un operario automáticamente para loguear la app (No loguearse automáticamente, sino completar la información en los textbox correspondientes para agilizar el ingreso de información).*

8. *La aplicación deberá tener hardcodeada información previamente para poder probarla sin cargar nada a mano.*

9. *Serán valorados a la hora de evaluar la creatividad que se aplique al programa en cuanto al*

diseño de la app.

10. *Debe estar bien documentado y acompañado por un documento PDF que explique brevemente la funcionalidad pretendida.*

11. *En dicho PDF Deberá indicarse por cada tema utilizado del punto 4 de este documento (Tener en cuenta que la segunda parte de este parcial también va a ser incluida en este*
documento), y marcar donde se encuentra su implementación, a fines de facilitar la corrección.

12. *Si DOS o Más proyectos se parecen en lo más mínimo, serán*

desaprobados, 13. Quienes no cumplan TODAS estas condiciones, no serán

evaluados

Generar una solución nombrada como. Apellido.Nombre.Parcial

Formato de entrega

El examen será entregado el día martes **16 de Octubre.**.
Por medio de un form de google se pedirá que registren el repositorio de github donde se encontrará ubicado el código. El mismo debe configurarse como **privado** y agregar los usuarios de los docentes para poder ver el código.

El repositorio deberá ser reportado por única vez en este formulario: Link del formulario.

Defensa del parcial

Aquellos parciales que estén al borde de aprobar o desaprobar, o que tengan nota de promoción deberán realizar la defensa del parcial de forma oral con cámara web prendida (de no tenerla,utilizar la cámara del teléfono celular).

Con el fin de probar que los temas están comprendidos y que el desarrollo fue hecho por el/la alumno/a, la instancia de defensa podrá incluir:

- Preguntas técnicas y/o funcionales sobre el desarrollo presentado.
- Modificación de una parte del código en vivo a criterio del docente.
- Preguntas teóricas sobre qué cambios deberían de realizarse si se quisiera contemplar algún otro escenario.

La defensa permitirá:

- Subir,bajar o nivelar la nota del alumno.

- Desaprobación total del mismo (por más que esté correctamente desarrollado) en caso de que el alumn@ no demuestre conocimiento sobre los temas aplicados/desarrollados planteando la duda si fue hecho por otra persona.

Consultas sobre el parcial.

Se responderán consultas sobre el parcial durante las clases. De existir necesidad de hacer alguna consulta en otro momento de la semana,no se garantizará una respuesta veloz ya que la misma quedará sujeta a disponibilidad del docente y/o sus ayudantes. Todas las preguntas deberán realizarse por **DISCORD como único medio y sin excepción**.

○ *Herencia*

Se aplica una clase abstracta de producto, donde no se puede instanciar y derivan de ella 3 clases (BIZCOCHUELO, GELATINA Y FLAN). También se aplico herencia en cuanto a la clase usuarios derivando de ella operario y de operario supervisor.

○ *Sobrecarga de constructores, ¿operadores y métodos*

Se sobrecargo el método == para poder validar que no haya dos usuarios iguales no vi útil realizar otro tipo de sobrecarga.

○ *Propriedades e Indexadores*

Se utilizaron las propiedades de las variables estáticas privadas para poder setear stock y retornar la cantidad de ese stock.

○ *Colecciones (al menos 2 colecciones diferentes)*

Se utilizaron listas para cargar los usuarios, y también se utilizaron diccionarios, para poder manejar el tema de sabores ya que eran varios pensé que era mejor utilizar un diccionario clave string valor int para poder usar lo sabores string y su cantidad int.

○ *Enumerados*

Se utilizo para tipo de sabores general y tipos de usuarios.

○*Conversiones Implícitas y Explícitas*

Se castea a int los numericupdowm utilizados en máquinas y cocinas para poder utilizar su valor numérico. (explicita) implícita se genera una instancia de supervisor asignándola a una de operario. Ya que supervisor es herencia el operario pasa a ser funcionalmente un supervisor.

○ *Formularios modal.*

Se utilizaron formularios modales para todo el programa por medios de avisos.

○ *Clases estáticas*

Se utilizo clase estática STOCK para poder manejar las cantidades de materias primas.

○ *Polimorfismo (clases abstractas, métodos abstractos, métodos virtuales*

Se utilizó la clase abstracta producto derivando de ella otras 3 y también se generaron métodos abstractos para que las derivadas estén obligados a utilizarlo