



SISPRO

El sistema pretende simular la producción de una industria gráfica, con sus supervisores y operarios, quienes se encargan del del manejo del stock y de la línea de producción, respectivamente. En dicha producción nos podemos encontrar con libros, cuadernillos, boletas y demás.

La empresa cuenta con un stock, el cual puede incrementar o disminuir en base a los trabajos completados por el operario. Una de las tareas del supervisor es el agregar más suministros al Stock desde un mercado, para que el operario pueda seguir con la producción.

El menú del supervisor cuenta con una interfaz desde la cual puede crear, modificar y eliminar usuarios que se encuentren en el sistema. Y también con una pestaña de configuración desde la cual puede modificar el fondo de los formularios.

A continuación, indico donde se implementa lo establecido en el punto 4:

- **Herencia:** Podemos ver una relación de herencia en la clase Empresa, siendo esta la clase base, la cual deriva a Administración, la cual deriva a Stock.
- **Sobrecarga de métodos:** Esta aplicado en la clase Stock y en la clase Operación.
- **Propiedades:** En mayor medida esta aplicado en Stock, sino también se puede encontrar en Empresa o en Usuario.
- **Colecciones:** Utilizo List<> para la lista de usuarios y Array para la contraseña.
- **Conversiones Explícitas:** Utilizado mucho en Operación, para la validación de datos ingresados o para Parsear datos del DataGrid.
- **Clases Estáticas:** La clase Operación es estática.
- **Polimorfismo:** La clase Empresa es abstracta.
- **Excepciones:** En FrmConfiguracion, Archivo, en las clases DAO, en Visual para el ManejoDataGridOperario y del CRUD.
- **Archivos y Serialización:** En txt se guarda el log de errores, en JSON se guardan las configs, rutas de los fondos e iconos y en XML se guardan el texto de los elementos del FrmLogin y FrmRegistro. Todos se encuentran en la clase Archivo.
- **Generics:** Se aplica en el método `SerializarXML<T>`.
- **Interfaces:** IMensaje y ILoginUsuario.
- **Test Unitario:** Tienen su respectivo proyecto llamado Testing.
- **Base de Datos:** Tanto la lista de usuarios como el stock están en sus respectivas tablas en una base de datos.
- **Delegados:** Delegado Comparison<> utilizado con función lambda en Administracion para el ordenamiento de la lista de pedidos. Delegado propio aplicado en Administracion para LogErrores, también en FrmLogin y en FrmConfiguracion un delegado de tipo Action.
- **Eventos:** EventoLogError y EventoLoginUsuario en administración. Manejo de eventos predeterminados en FrmConfiguracion. Tarea asincrónica en `CargarArchivo()` en FrmConfiguracion.