MUTUAL

Este programa está pensado para poder gestionar los turnos que otorga una mutual a sus afiliados, para poder sacar turnos para consultar a los distintos médicos disponibles. Se pueden dar de alta, baja y modificar la información de los afiliados, y de las atenciones solicitadas.

Proyecto Entidades -> Biblioteca de clases:

Aquí se definen las entidades necesarias para el funcionamiento de la aplicación. Se utiliza herencia, clases abstractas y se define la interfaz IAcciones, la cual contiene el método mostrar datos que posteriormente implementan las entidades Afiliado y Profesional.

También se define la excepción SerializacionException, la cual se utiliza para capturar errores que puedan darse en el proceso de serialización, tanto a JSON como a XML.

Para la mencionada serialización, se definen las clases SerializacionJSON y SerializacionXML, las cuales contienen métodos tanto para serializar como deserializar archivos. Éstos métodos usan tipos Genéricos como objetos a la hora de ejecutarse, es decir que reciben o devuelven objetos genéricos según corresponda.

La clase AccesoDatos se encarga de administrar el acceso a la base de datos, definiendo los parámetros de conexión en su constructor -estático- y luego abriendo/cerrando dicha conexión en cada método según corresponda. Éstos métodos se encargan de leer, agregar o borrar registros en la DB.

Proyecto UI -> Formularios:

FormLogin: Form básico para loguearse al sistema. Los datos de ingreso son “Admin”, “admin”, pero se ha incluido un botón para completarlos automáticamente.

FormPrincipal: Es el Form principal del programa, desde el cual se accede a todas las funciones del mismo.

FormAfiliados: ABM de Afiliados del sistema. También permite exportar la grilla tanto a JSON como a XML.

FormAltaAfiliado: Form para cargar manualmente a un nuevo afiliado.

FormProfesionales: Form para ver a los profesionales de la Mutual, la carga en la grilla de los mismos se realiza con una Task que puede ser cancelada en cualquier momento, o lo será al terminar de cargar todos los profesionales disponibles. Se crea un hilo paralelo que se encarga de realizar esta carga, dejando 2 segundos de tiempo entre cada carga de profesional realizada.

FormAtencionesDiarias: Form para cargar atenciones aleatorias para el día de la fecha, el botón de carga disparará un evento que cargará una nueva atención. Se podrá cargar atenciones diarias hasta llegar al límite establecido por la Mutual: 10. El evento turnosCompletos se define en la clase estática Mutual, y se invoca en el método AregarNuevaAtencion de dicha clase, al alcanzar el límite de atenciones.

FormAtencionesFuturas: Form para agendar atenciones manualmente. Estas atenciones se guardan en la DB y no tienen relación con las atenciones diarias.

FormCargarAtencion: Form para cargar manualmente una nueva atención futura.