

## Trabajo práctico Final – Laboratorio de programación II

### **Funcionalidad**

La presente aplicación pretende simular la atención al público de un centro médico.

En la pantalla principal se ofrecerá una planilla con los turnos reservados para el día de la fecha donde se podrá filtrar según el estado de los mismos (Pendiente – Ausente – Atendido – Espera). Todos los turnos del día iniciarán en estado “Pendiente”, y en la medida que vayan ingresando los pacientes se cambiará manualmente en el estado que corresponda. Los pacientes, médicos y turnos fueron importados automáticamente desde archivos en formato .Json a partir de la ejecución del programa.

También incluye un menú donde podremos visualizar:

#### **Turnos:**

- Alta: Se podrá registrar un turno a partir del DNI del paciente (registrado previamente), DNI del médico, fecha y hora del turno. Ambos textbox de DNI sugieren DNIs ya registrados con valores de entre 20.000.000 - 44.000.000. Los DNI de médicos se podrán filtrar por especialidad
- Ver turnos: Se mostrara una planilla con todos los turnos cargados tanto del día de la fecha como pasados y futuros. Se podrá exportar en formato .Json en la carpeta Datos del Escritorio.

#### **Pacientes:**

- Alta: Se podrá registrar un paciente en el sistema.
- Ver Pacientes: Se mostrará una planilla con los datos de todos los pacientes. Se podrá exportar en formato .Json en la carpeta Datos creada en el Escritorio.

#### **Especialidades:**

- Se podrá visualizar el listado de todas las especialidades y médicos disponibles en la clínica. También se dispone de un botón para exportar la planilla en formato .Json

#### **Ingreso:**

- Registrar: En este apartado se visualizarán todos los turnos en estado “Pendiente”. Aquí se podrá cambiar el estado del turno a “Espera” si el paciente vino o “Ausente” en caso de no asistir.
- Atender: Todos los turnos marcados como “Espera” en la vista anterior serán los que se ofrecerán en la planilla de este apartado para, finalmente, pasar a estado “Atendido”.

Cuando la aplicación se cierre guardará un archivo en formato .Json con todos los turnos y pacientes nuevos para su próxima ejecución en la carpeta bin del proyecto.

### **Implementaciones:**

#### **Excepciones:**

- ListaVacíaException: Se arroja en el método CambiarEstados() en el formulario FrmIngresarRegistro: Se maneja en el evento Click del botón RegistrarTurno.
- ArgumentoNoValidoException: Se arroja en los setters de las propiedades cuyos atributos correspondan validar.
- NoDisponibleException: Se arroja cuando el horario seleccionado no está disponible.
- NoEncontradoException: Se arroja cuando un elemento no se encuentra en las listas.

#### **Pruebas unitarias:**

- BuscarTurnoTest: Se prueba funcionalidad del método BuscarTurno de la clase Clínica.
- AgregarListadoTest: Se prueba funcionalidad del método genérico AgregarListado de la clase Clínica

#### **Tipos Genéricos:**

- Se utilizan métodos genéricos: AgregarListado e Importar de la clase Clínica
- Se utilizan dos clases genéricas: ArchivosJson y ArchivosXml

#### **Interfaces:**

- IListable: Se utiliza para otorgar métodos necesarios para listar y guardar en los registros. Se implementa en las clases Persona y Turno

#### **Archivos y Serialización:**

- FrmPrincipal\_Load: Se utiliza el método Importar() que incluye el Leer() de la clase ArchivosJson
- FrmListarTurnos: Se utilizan los métodos Escribir() de ArchivosJson y ArchivosXml para exportar información en los respectivos formatos.