



Universidad Tecnológica Nacional

Facultad Regional Avellaneda

Trabajo Practico Nº 3

2ºE

2º cuatrimestre 2021

Aplicación de manejo de Usuarios Asociación Pablo Besson

Alumna: Yañez Evelyn Gabriela

OBJETIVO

El Objetivo del desarrollo es realizar una aplicación para la Asociación Pablo Besson que le permita a los Coordinadores y Psicólogos que trabajan en la misma realizar el manejo de información de los Usuarios que asisten y realizar distintos análisis con la misma.

ESPECIFICACIÓN FUNCIONAL

Apertura y Cierre de Turnos:

La aplicación contara un sistema de cierre de y apertura de turnos que le permitirá al coordinador o psicólogo encargado de realizar la carga de asistencias controlar que estas se hayan realizado en su totalidad. Este proceso de cierre está corriendo cada dos segundos de manera secundaria en la aplicación, realizando las validaciones correspondientes al cierre.

Apertura de turno:

- Si no existe ningún turno abierto entonces se realizará la apertura del turno correspondiente a la fecha actual.
- Si existe un turno cargado y este se encuentra cerrado entonces se realizará la apertura del día siguiente al último cerrado.

Cierre de turno:

Para realizar el cierre de un turno se deberá dar una de las siguientes condiciones:

- Todos los usuarios registrados para el grupo correspondiente al turno que se encuentra abierto deben tener un registro de asistencia cargado y la fecha de calendario actual es mayor al turno que se quiere cerrar.
- No hay un grupo activo para la fecha correspondiente al turno que se encuentra abierto y la fecha de calendario actual es mayor al turno que se quiere cerrar.

Observaciones:

- Cuando no sea posible realizar el cierre de turno debido a la falta de carga de registros saltara una alerta en el menú principal y al hacer clic sobre la misma se podrá acceder a una pantalla de carga donde se encuentren todos los registros de asistencias faltantes y estos podrán se cargados a la vez.
- Una vez realizada la carga de asistencias mencionada en el punto anterior ya estarán dadas las condiciones para el cierre de turno, lo que implicara que el mismo se cierre la próxima de vez que se ejecute el proceso de cierre.

Perfiles:

La aplicación contara con dos perfiles de ingreso:

- Administrador:
- Empleado:

El empleado podrá:

- Dar de alta un usuario.
- Listar Usuarios.
- Agregar una asistencia.
- Editar Asistencia.
- Importar asistencias (Desde un XML o Json). (*)
- Filtrar asistencias por grupo y fecha.
- Exportar los listados de asistencia filtrados (csv o txt)
- Generar reportes y exportarlos (csv, txt, XML o json).

El administrador podrá:

- Realizar las acciones que realizar el empleado.
- Ver los listados de empleados.
- Ver los listados de administradores.

(*) Al importar asistencias se tendrán en cuenta las siguientes reglas:

- Si el listado de asistencias a importar posee una asistencia ya cargada entonces esta no se cargará, pero no detendrá la carga del resto del listado.
- Si el listado posee una asistencia correspondiente a un usuario que no está registrado entonces esta no se cargará, pero no detendrá la carga del resto del listado.

Reportes generados:

Se podrán generar reportes de dos tipos, reportes de filtros y reportes por porcentaje, que le permitirán al operador/a o psicólogo/a realizar un análisis de los datos en el sistema.

Reportes de Estadístico:

Se calcularán y mostrar en pantalla las siguientes estadísticas:

- Causa de Ingreso que caracteriza a cada grupo y el porcentaje de sus integrantes que posee esta causa de ingreso.
- Número de causas de ingreso que posee el usuario con más denuncias. En caso de haber más de un usuario con la cantidad máxima de denuncias, se retornará el que posea el mayor número de causas de ingreso.
- Grupo o grupos que posean algún integrante con la mayor cantidad de denuncias..

Reportes de filtro:

Se podrá filtrar el listado de usuarios por:

- Fecha de ingreso desde y fecha hasta.
- Cantidad de denuncias desde y hasta.
- Motivos de ingreso.

Reportes de porcentajes:

Se podrá generar reportes por porcentajes de usuarios/asistencias en:

- Intervalos de cantidad de denuncias registradas.
- Intervalos de cantidad de meses que el usuario lleva ingresado en la asociación.
- Agrupaciones por causas de ingreso (*).
- Agrupaciones por tipos de asistencia, determinando el grupo al que pertenece el listado de asistencias a analizar.

(*) El total de los porcentajes de usuarios por cada una de las causas de ingreso no debe ser necesariamente del 100% debido a que un usuario puede tener más de una causa de ingreso.

ESPECIFICACIÓN DE APLICACIÓN DE TEMAS EVALUADOS

En esta sección se indicará donde se aplicaron cada uno de los temas obligatorios para esta entrega.

- **Bases de datos:**

Se creó la clase estática DB para establecer y configurar la conexión a la base de datos.

Se crearon las clases AsistenciaDB, TurnoDB y UsuarioDB para generar y ejecutar las queries de inserción, selección y actualización según corresponda.

Se crearon las tablas asistencias, usuarios, delitos y turnos con las relaciones correspondientes.

Observación: se generó un script con la estructura e información de la base de datos en la carpeta TP4 con el nombre TPFinal_EvelynYanez.

- **Delegados y expresiones lambda:**

Se creó el delegado CierreAperturaTurno en la clase CierreTurno.

Se utilizaron expresiones lambda en el método ControlarCierre de la clase CierreTurno y en los métodos Filtrar, BuscarListadoDni y FiltrarCausaDelIngreso de la clase UsuarioControlador.

Se utilizaron expresiones lambda en los métodos CalcularCantidadCausaIngreso,

CalcularCausaDelIngresoUsuarioConMasDenuncias y

CaracterizacionGruposPorCausaDelIngreso en la clase FrmReporteEstadistico.

- **Hilos:**

Se utilizó una tarea en el método ControlarCierre de la clase CierreTurno.

Se utilizaron tareas en el formulario FrmReporteEstadistico.

- **Eventos:**

Se crearon y utilizaron los eventos CambioTurno y AsistenciasFaltantes de la clase CierreTurno.

- **Métodos de extensión:**

Se utilizan métodos de extensión para extender la clase List y Datetime en las clases ListExtension y DateTimeExtension respectivamente.

En la clase PersonaControlador se utiliza el método de extensión BuscarDniQueContenga para extender las listas genéricas que heredan o son del tipo Persona.

En la clase EmpleadoControlador se utiliza el método de extensión FiltrarPorTipoEmpleado para extender las listas que heredan del tipo Persona y filtrar por un tipo genérico que herede o sea de la clase Empleado.

En la clase EnumExtension se utiliza el método de extensión ObtenerDescripcion para obtener la descripción de un enumerado en caso de que este la tenga.