

Conceptualización

A continuación, se presentan los principales cambios en el proyecto “SimuladorHorario”, que es el proyecto de la solución que contiene toda la funcionalidad del programa, y la conceptualización de “VistasSimuladorHorario”, que es el proyecto que contiene toda la interfaz gráfica de la solución.

SIMULADOR USUARIO

- Se deja de usar casi toda la clase program, que contiene toda la funcionalidad en consola. Se siguen usando algunos métodos que servían para cosas puntuales del programa (ninguna relevancia en la funcionalidad)
- Los menús de Gestor y PlataformaUsuario se eliminan ya que estos eran en consola y ya no se necesitan. Además, los métodos que requieren que el usuario ingrese información ahora piden la información como parámetro. Estos parámetros son entregados mediante la interfaz gráfica.
- Se elimina la interfaz IHorarioMostrable ya que esta interfaz solo servía para mostrar la información de las clases Horario y Agenda a través de la consola.
- Se agregan a la clase Aplicación las funciones SerializeAll() y DeSerializeAll() que están encargadas de serializar la información relevante para tener persistencia de información cuando el programa finalice.

VISTAS SIMULADOR HORARIO

En esta sección existen 4 tipos de archivos:

Controladores: Estos se encargan de las suscripciones de los eventos del Form correspondiente. Todos reciben en su constructor a su Form correspondiente.

Al momento de crear un objeto de esta clase se suscriben todos los métodos correspondientes a sus eventos correspondientes, los cuales se encuentran en la form correspondiente. Reciben como parámetro la Form que les corresponde. Los controladores existentes son:

- AgregarEventoController
- CrearCursoController
- EditarPerfilController
- EliminarCursoController
- GestorController
- InscribirCursoController
- LeerCursoController
- LoginController
- PlataformaEstudianteController
- RegistrarseController

Forms y sus códigos fuentes: Estas son las interfaces gráficas como tal, en estas clases se define la disposición de los botones, textboxes, labels, etc.... Que sean necesarios para la ventana correspondiente. En su código poseen las clases de los ítems de la form y los eventos que hacen que cuando se clickee un botón, se cambie la ventana a la ventana deseada. Estos son los eventos a los cuales se suscriben las funciones del controlador de la form correspondiente. Las forms de la solución son:

- **AgregarEventoForm:** Aquí si agregan los eventos de los cursos que crea el administrador mediante el gestor. Se accede desde **CrearCursoForm**.
- **CrearCursoForm:** Aquí, el administrador puede crear cursos, posee un botón a **AgregarEventoForm**. Se accede desde **GestorForm**.
- **EditarPerfilForm:** Aquí se puede editar el perfil del estudiante. Se accede desde **PlataformaEstudianteForm**.
- **EliminarCursoForm:** Aquí, el administrador puede eliminar un curso. Se accede desde **GestorForm**.
- **InscribirCursoForm:** Aquí, el estudiante puede inscribir un curso para que se muestre en el horario. Se accede desde **PlataformaEstudianteForm**.
- **LeerCursoForm:** Aquí, el administrador puede obtener la información detallada de un curso en específico. Se accede desde **GestorForm**.
- **LoginForm :** Esta es la primera ventana del programa, aquí se inicia sesión o se accede a la ventana de registro. Si el usuario ingresado es un estudiante, se accede a la plataforma de estudiante; si es administrador, se accede al gestor.
- **PlataformaEstudianteForm:** Esta es la ventana principal para el estudiante, aquí puede ver su agenda y su horario. Además, a través de menús desplegables puede acceder a todas las funcionalidades hechas para él, como editar su perfil, agregar cursos inscritos, eliminar cursos inscritos, agregar eventos etc... Se accede desde **LoginForm**.
- **RegistrarseForm:** Esta es la ventana donde un estudiante puede registrarse en el sistema. Se accede desde **LoginForm**.

EventArgs: En nuestro programa, todos los eventos son definidos con el delegate **EventHandler**, es por esto que, para los eventos que necesiten argumentos, definimos una clase *nombredelaventanaEventArgs*, en la cual se definen los argumentos que toma el evento. Las clases son las siguientes:

- **IniciarSesionEventArgs:** Define un **Usuario** como argumento.
- **InscribirCursoEventArgs:** Define un **string** con el nombre del curso a inscribir.