

Academic Schedule

INTEGRANTES

Diego Fernando Hidalgo Lopez

Brian Stiven Romero Restrepo

UNIVERSIDAD ICESI

ALGORITMOS Y PROGRAMACIÓN II

DEPARTAMENTO DE TIC

CALI - VALLE DEL CAUCA

2021

Justificación

Consideramos que dos personas es la cantidad indicada para llevar a cabo los objetivos propuestos a alcanzar en el presente proyecto. Primero, por la complejidad que pueden tener algunos objetivos a desarrollar y segundo, porque de esta manera se logra realizar una distribución adecuada de las tareas a realizar para alcanzar los plazos de entrega.

Enunciado

Los estudiantes actuales requieren de gestionar una gran cantidad de información referente a los cursos que están tomando. Hacer esto en papel puede llegar a ser ineficiente debido a que podemos olvidar los eventos que hemos planeado o porque se podría necesitar el uso de grandes cantidades de papel. Dado esto, la mejor opción es llevar un registro de los cursos que se están tomando usando medios digitales. Para cumplir con esta necesidad, se plantea el desarrollo de un programa que permita gestionar la información de los cursos y las notas, y adicional a esto que brinde herramientas que permitan crear planes de estudio para cumplir de manera satisfactoria los objetivos propuestos por cada curso.

Para el uso del sistema se debe crear un usuario desde el cual se podrá acceder a todas las opciones que ofrece el programa. El usuario cuenta con un nombre, un apellido, un nombre de usuario único, una contraseña y una foto de perfil.

El programa debe permitir la gestión de cursos, esto es: crear un curso, actualizar la información de los cursos y eliminar un curso en particular. Un curso cuenta con un nombre, una cantidad de créditos, los días y horarios en que se toma, un conjunto de calificaciones y un estado (MATRICULADO o CANCELADO). A su vez se requiere que se gestione la información de las calificaciones de cada curso. Esto implica agregar una nueva nota a un curso, actualizar la información de la nota y eliminarla. Una nota cuenta con una descripción, un valor numérico y peso (porcentaje).

El programa debe estar en la capacidad de crear un plan de estudios, actualizar su información y eliminarlo. Este plan de estudios es una forma de designar un espacio de tiempo para enfocarse a estudiar unos temas en específico y obtener unas metas planeadas en el momento de programar el plan de estudios. En este orden de ideas, un plan de estudios cuenta con un día en el que se va a realizar, una hora de inicio y una hora de finalización, los temas a estudiar durante ese tiempo, un curso al que están relacionados los temas de estudio, una breve descripción y unas metas a lograr durante el tiempo definido.

Se deben enviar notificaciones de los eventos cuando estén próximos a ocurrir. La notificación incluirá una breve descripción del evento a ocurrir. El usuario debe tener la opción para configurar el momento en que se debe enviar la notificación antes del evento (por ejemplo 10 minutos antes).

Se le debe dar la opción al usuario de mostrar en una gráfica el rendimiento por curso basado en la nota.

Se debe permitir importar información de nuevos cursos a agregar al sistema desde un archivo de texto plano.

El programa debe estar en la capacidad de exportar información referente al rendimiento de los cursos en un archivo en la ubicación seleccionada por el usuario. Este archivo debe incluir la información de los cursos referente al nombre y las calificaciones. Adicional a esto se incluye la nota calculada hasta el momento de exportar el archivo tanto de los cursos como del promedio ponderado.

La información del programa debe guardarse en archivos serializados de manera automática y transparente para el usuario.

Requerimientos funcionales

Para el correcto cumplimiento de los requerimientos que se proponen para este proyecto, el sistema debe de estar en la capacidad de:

RF1: Gestionar cursos con un nombre, días en los que toma el curso, las hora en la que ve el curso, la cantidad de créditos, un conjunto de notas con su peso (en porcentaje) y un estado (Matriculado o Cancelado)

- **RF1.1** – Se debe poder crear un nuevo curso.
- **RF1.2** – Se debe poder modificar un curso existente.
- **RF1.3** – Se debe poder cambiar el estado de un curso.
- **RF1.4** – Se debe poder gestionar las calificaciones del curso. Las calificaciones tienen una breve descripción, un valor numérico y un peso.
- **RF1.4.1** - Se debe poder agregar una nueva calificación a un curso.
- **RF1.4.2** - Se debe poder modificar las calificación de un curso.
- **RF1.4.3** - Se debe poder eliminar las calificación de un curso.

RF2: Gestionar usuarios con un nombre, apellido, nombre de usuario, contraseña y foto de perfil.

- **RF2.1** – Se debe poder crear un nuevo usuario.
- **RF2.2** – Se deben poder modificar los datos de un usuario.
- **RF3.3** - Se debe poder eliminar un usuario del sistema.

RF3: Presentar en una gráfica el rendimiento por curso basado en la nota.

- **RF3.1** - Se debe poder ver la gráfica comparativa entre las notas de todos los cursos

RF4: Crear un plan de estudio para un día específico con un curso relacionado, hora de inicio, hora de finalización, título, descripción y metas de estudio.

- **RF4.1** - Se debe poder crear un nuevo plan de estudio
- **RF4.2** - Se debe poder modificar los datos de un plan de estudios en específico
- **RF4.3** - Se debe poder eliminar un plan de estudios.

RF5: Gestionar eventos de las clases con la fecha del evento, título y descripción.

- **RF5.1** - Se debe poder crear un nuevo evento.
- **RF5.2** - Se debe poder modificar los datos de un evento.
- **RF5.3** - Se debe poder eliminar un evento.
- **RF5.4** - Se debe poder mostrar la información de los eventos en un calendario.

RF6: Enviar notificaciones de los eventos próximos. Estableciendo el momento en el que debe enviarse.

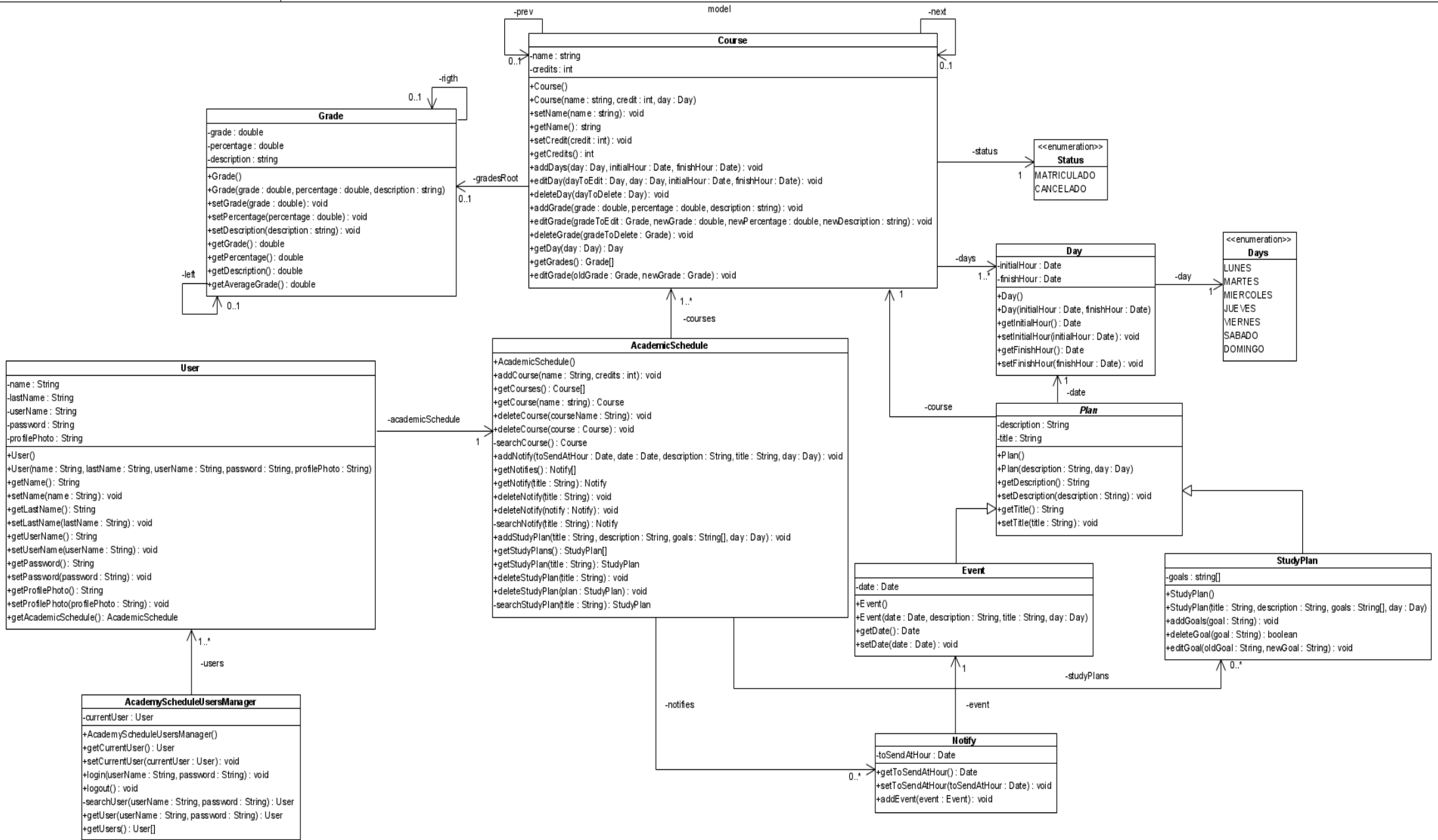
- **RF6.1** - Se debe poder programar una nueva notificación.
- **RF6.2** - Se debe mostrar la notificación en el momento en que fue programada. La notificación incluye un título y una breve descripción del evento al que está relacionada.
- **RF6.3** - Se debe poder modificar la hora en para la que fue programada una notificación.
- **RF6.4** - Se debe poder desactivar la notificación para un evento en específico.

RF7: Exportar información sobre los cursos a un archivo de texto.

- **RF7.2** - El archivo debe contener el nombre del curso, las notas, la nota calculada hasta el momento y el promedio ponderado hasta el momento de exportar el archivo.
- **RF7.1** - El archivo debe guardarse en la ubicación especificada por el usuario.

RF8: Importar información sobre nuevos cursos a agregar desde un archivo de texto plano.

RF9: Guardar la información en archivos de texto planos serializados. Este proceso se hace de forma automática.



Bienvenido

Ingresar

Registrate



Ingresar

Registrarse

Agregar un curso

+

+

+

Estado

▼

Matriculado

Cancelado

Agregar

Crear plan de estudio

+

Cursos

▼

+

+

Agregar meta

Crear

Cursos actuales

Curso 1

Curso 2

Curso 3

Curso n

Información curso 1

Curso 1

+

Martes, miercoles

+

10:00

+

12:00

+

Matriculado

+

Nota
4.0

Porcentaje
20%

+

Modificar

Plan de estudio 1

18/05/2021



Curso 1

14:00



18:00



Una descripción

Meta 1

Agregar meta

Crear evento

fecha de inicio



fecha final



Hora de inicio



Hora final



Descripción...

Enviar notificación

Crear

Modificar evento

fecha de inicio



fecha final



Hora de inicio



Hora final



Descripción...

Enviar notificación

Crear

Configuración de los escenarios

Nombre	Clase	Escenario
setUpScenary1	AcademicScheduleUsersManager	Se crea un objeto de la clase AcademicScheduleUsersManager con una lista de usuarios vacía
Nombre	Clase	Escenario
setUpScenary2	AcademicScheduleUsersManager	Se crea un objeto de la clase AcademicScheduleUsersManager con 3 usuarios registrados 1. name = "Diego" lastName = "Hidalgo" userName = "DiegoH" password = "diegofer" photoPath = "..." 2. name = "Brian" lastName = "Romero" userName = "BrianR" password = "brianromero" photoPath = "..." 3. name = "Juan" lastName = "Hernandez" userName = "JuanR" password = "juanhernandez" photoPath = "..."
Nombre	Clase	Escenario
setUpScenary3	AcademicScheduleUsersManager	Se crea un objeto de la clase AcademicScheduleUsersManager con 1 usuario registrado y se ingresa al sistema 1. name = "Diego" lastName = "Hidalgo" userName = "DiegoH" password = "diegofer" photoPath = "..."
Nombre	Clase	Escenario
setUpScenary4	AcademicSchedule	Se crea un objeto de la clase AcademicSchedule con las listas vacías y el curso en null
Nombre	Clase	Escenario
setUpScenary5	AcademicSchedule	Se crea un objeto de la clase AcademicSchedule con un curso registrado

		1. name = "APO II" credits = 3 days = {"Lunes", "Jueves"} initHours = {"14:00","14:00"} finishHours = {"15:00","15:00"}
Nombre	Clase	Escenario
setUpScenary6	AcademicSchedule	Se crea un objeto de la clase AcademicSchedule con un curso y una notificación registrados. 1. name = "APO II" credits = 3 days = {"Lunes", "Jueves"} initHours = {"14:00","14:00"} finishHours = {"15:00","15:00"} 2. toSendAtHour = "13:00" description = "Recordatorio de clase" title = "Recordatorio" initHour = "14:00" finHour = "15:00" day = "Lunes"
Nombre	Clase	Escenario
setUpScenary7	AcademicSchedule	Se crea un objeto de la clase AcademicSchedule con un curso y un plan de estudio registrado 1. name = "APO II" credits = 3 days = {"Lunes", "Jueves"} initHours = {"14:00","14:00"} finishHours = {"15:00","15:00"} 2. title = "Plan de estudios" description = "Prueba de plan de estudios" goals = {"1","2","3"} day = "Lunes" init = "08:00" fin = "10:00"

Diseño de los casos de prueba

Objetivo de la prueba: Verificar que se registra exitosamente un usuario				
Clase	Método	Escenario	Valores de Entrada	Resultado
AcademicSchedule UsersManager	addUser	setUpScenary1	name = "Diego" lastName = "Hidalgo" userName = "DiegoH" password = "diegofer" photoPath = "..."	True. Se agrega un nuevo usuario y la cantidad de usuarios es 1
Objetivo de la prueba: Verificar que no se permite el registro de usuarios con el mismo nombre				
Clase	Método	Escenario	Valores de Entrada	Resultado
AcademicSchedule UsersManager	addUser	setUpScenary2	name = "Diego" lastName = "Hidalgo" userName = "DiegoH" password = "diegofer" photoPath = "..."	True. Se lanza la excepción y la lista de usuarios sigue siendo 3
Objetivo de la prueba: Verificar el correcto ingreso al sistema de un usuario				
Clase	Método	Escenario	Valores de Entrada	Resultado
AcademicSchedule UsersManager	logIn	setUpScenary2	userName: "DiegoH" password: "diegofer"	True. Se ingresa al sistema y los datos del usuario en el sistema son los dados en la entrada
Objetivo de la prueba: Verificar que no se permite el ingreso al sistema con credenciales incorrectas				
Clase	Método	Escenario	Valores de Entrada	Resultado
AcademicSchedule UsersManager	logIn	setUpScenary2	userName: "DiegoH" password: "1234567"	True. Se lanza la excepción. El usuario actualmente en el sistema es null.
Objetivo de la prueba: Verificar la validación de campos vacíos en una serie de cadenas de texto				
Clase	Método	Escenario	Valores de Entrada	Resultado
AcademicSchedule UsersManager	verifyBlankChars	setUpScenary1	{"a","b","c"."d"."e"."f" "}	True. Se verifica que las cadenas dadas no contienen solo espacios.
Objetivo de la prueba: Verificar la validación de campos vacíos en una serie de cadenas de texto				
Clase	Método	Escenario	Valores de Entrada	Resultado
AcademicSchedule UsersManager	verifyBlankChars	setUpScenary1	{" ", " ", " ", "ab "}	True. Se verifica que

				por lo menos una de las cadenas dadas contiene solo espacios en blanco
Objetivo de la prueba: Verificar el correcto cambio de datos de un usuario				
Clase	Método	Escenario	Valores de Entrada	Resultado
AcademicSchedule UsersManager	changeUser	setUpScenary2	name = "Juan" lastName = "Perez" userName = "JuanPe" password = "1234567" photoPath = "..."	True. Los datos del usuario se actualizan y coinciden con los datos dados en la entrada.
Objetivo de la prueba: Verificar que se elimina correctamente un usuario del sistema				
Clase	Método	Escenario	Valores de Entrada	Resultado
AcademicSchedule UsersManager	deleteUser	setUpScenary2		True. Se elimina correctamente el usuario. La cantidad de usuarios es ahora de 2.
Objetivo de la prueba: Verificar el correcto cambio de contraseña de un usuario				
Clase	Método	Escenario	Valores de Entrada	Resultado
AcademicSchedule UsersManager	changeUserPassword	setUpScenary2	userName: "DiegoH" newPassword: "diegofer12345"	True. Se cambia la contraseña del usuario indicado por la contraseña dada.
Objetivo de la prueba: Verificar que se agrega correctamente un curso				
Clase	Método	Escenario	Valores de Entrada	Resultado
AcademicSchedule	addCourse	setUpScenary4	name = "Algebra lineal" credits = 3 days = {"Lunes", "Jueves", "Viernes"} initHours = {"14:00", "14:00", "14:00"} finishHours = {"15:00", "15:00", "15:00"} 0"	True. Se agrega un nuevo curso con los datos dados. La cantidad de cursos es ahora de 1.
Objetivo de la prueba: Verificar que se agrega correctamente un curso				

Clase	Método	Escenario	Valores de Entrada	Resultado
AcademicSchedule	addCourse	setUpScenary5	name = “Algebra lineal” credits = 3 days = {“Lunes”, “Jueves”, “Viernes”} initHours = {“14:00”, “14:00”, “14:00”} finishHours = {“15:00”, “15:00”, “15:00”}	True. Se agrega un nuevo curso, la cantidad de cursos es ahora de 2.
Objetivo de la prueba: Verificar que se elimina correctamente un curso				
Clase	Método	Escenario	Valores de Entrada	Resultado
AcademicSchedule	deleteCourse	setUpScenary5	Se pasa el objeto del primer curso planteado en el escenario.	True. Se elimina el curso indicado. El atributo firstCourse es ahora null.
Objetivo de la prueba: Verificar que se agrega una notificación correctamente				
Clase	Método	Escenario	Valores de Entrada	Resultado
AcademicSchedule	addNotify	setUpScenary5	toSendAtHour = “14:00” description = “Recordatorio” title = “Recordatorio” initHour = “16:00” finHour = “18:00” day = “Martes”	True. Se agrega una nueva notificación con los datos dados.
Objetivo de la prueba: Verificar que se elimina una notificación correctamente				
Clase	Método	Escenario	Valores de Entrada	Resultado
AcademicSchedule	deleteNotify	setUpScenary6	Se pasa el objeto de la notificación planteada en el escenario	True. Se elimina la notificación indicada. La cantidad de notificaciones es ahora de 0.
Objetivo de la prueba: Verificar que se agrega un plan de estudio correctamente				
Clase	Método	Escenario	Valores de Entrada	Resultado
AcademicSchedule	addStudyPlan	setUpScenary5	title = “Prueba” description = “Prueba” goals = {“A”, “B”, “C”, “D”} day = “Viernes”	True. Se agrega un nuevo plan de estudios con los datos

			init = "10:00" fin = "13:00"	dados. La cantidad de planes es ahora de 1.
Objetivo de la prueba: Verificar que se elimina un plan de estudio correctamente				
Clase	Método	Escenario	Valores de Entrada	Resultado
AcademicSchedule	deleteStudyPlan	setUpScenary7	Se pasa el objeto del plan de estudios planeado en el escenario	True. Se elimina el plan indicado. La cantidad de planes de estudio es ahora de 0.