

# Trabajo Práctico Final Programación y Laboratorio 3.

*"Laboratorio 3 Learning System"*

```
1 public class Main {  
2  
3     public static void main(String[] args) {  
4         System.out.println("Trabajo Práctico final");  
5     }  
6 }  
7
```

## **Alumnos**

- Herrada, Emilio Álvaro
- Valverde, Gustavo Martín

## **Asignatura**

- Programación 3
- Laboratorio 3

## **Ciclo Lectivo**

- 2019

## **Universidad**

- U.T.N. MDP

## Contenido

1.Resumen.....	3
2.Proceso de Desarrollo. ....	3
3.Informe Técnico. ....	3
Administrador: .....	3
Creación de usuarios .....	3
Creación de cursos .....	3
Instructor.....	4
Creación y asignación de preguntas a los cursos .....	4
Asignación de estudiantes.....	4
Remoción de estudiantes asignados a un curso .....	4
Estudiante .....	4
Visualizar cursos disponibles.....	4
Completar curso.....	4
Obtener su promedio .....	4
Cursos.....	4
Preguntas .....	5
Verdadero o Falso .....	5
Por valor .....	5
Opción múltiple.....	5
Otras clases .....	5
Funcionalidades pendientes .....	5
Manual de usuario .....	6
Administrador .....	6
Estudiante .....	6
Instructor.....	7
Links:.....	8

## 1.Resumen.

El Presente trabajo consta de un sistema programado íntegramente en JAVA, cuya finalidad es generar una plataforma educativa virtual.

Existen tres tipos de usuarios hasta el momento, Administradores, Instructores y Estudiantes, donde cada uno tiene diferentes libertades a la hora de utilizar el programa.

Los usuarios de tipo Administrador tienen la potestad de crear usuarios del tipo Instructor y Estudiante, como así también de crear nuevos Cursos.

Los usuarios de tipo Instructor están autorizados para agregar preguntas a los Cursos y para inscribir usuarios del tipo Estudiante a los mismos, así como también eliminarlos de ser necesario.

Finalmente los usuarios de tipo Estudiante tienen permitido resolver los cursos y obtener calificaciones en base a su rendimiento.

Los datos generados por el código, se almacenan en archivos de texto.

## 2.Proceso de Desarrollo.

Inicialmente se definió el objetivo del sistema y los límites del mismo, en base a esta información se generó el pertinente diagrama UML (Anexado al presente trabajo). Posteriormente se realizó la división de tareas y responsabilidades.

Una vez delimitados las actividades que debe realizar cada integrante del grupo, se procedió a la realización del código del sistema. Una vez terminadas las tareas individuales, se integraron los códigos.

Finalmente se realizaron testeos de funcionalidad y se corrigieron errores que surgieron a la hora de unificar código.

## 3.Informe Técnico.

Una vez definidas las responsabilidades de cada tipo de usuario, se generó la funcionalidad correspondiente a cada uno de ellos.

A continuación, se describe cómo funciona cada tipo de usuario y como se relacionan entre sí.

### Administrador:

El administrador es el encargado de la creación de todo lo que compete al sistema. Esto quiere decir, que para crear estudiantes, instructores y/o cursos, se requiere la intervención de un administrador.

Actualmente, el sistema no permite crear otros administradores, por lo que es necesario crear manualmente al Administrador para poder empezar a utilizar el sistema.

### Creación de usuarios

La creación, tanto de estudiantes como instructores, es sencilla. Únicamente se requiere una serie de datos básicos y una contraseña.

### Creación de cursos

La creación de cursos solo requiere conocer el nombre del curso y el ID de un instructor a cargo del mismo. Cada curso posee un único instructor.

## Instructor

El instructor es quien está a cargo de los cursos. Las funcionalidades que posee son las siguientes.

### Creación y asignación de preguntas a los cursos

El instructor tiene la capacidad de crear y asignar tantas preguntas como desee a un curso específico. El instructor no puede crear ni asignar preguntas a un curso en el cual no sea titular.

### Asignación de estudiantes

El instructor puede asignar estudiantes a los cursos. Estos, deben existir y no estar asignados al mismo. Al igual que en el caso anterior, no puede agregar estudiantes a cursos que no le pertenezcan. Además, no se pueden agregar estudiantes a cursos que aún no posean preguntas.

### Remoción de estudiantes asignados a un curso

Así como puede agregar, el instructor tiene la potestad de remover estudiantes asignados a un curso de su propiedad. Si el estudiante ya tenía una calificación en dicho curso, esta también será removida, permitiendo volver a asignar al estudiante dentro del curso para repetir su evaluación.

## Estudiante

Es el objetivo de este sistema. Es quien va a ser asignado a cursos para su futura evaluación. Este, posee la siguiente funcionalidad.

### Visualizar cursos disponibles

Se considera disponible a todo curso, en el cual el estudiante está inscripto, pero aún no realizó su evaluación. El estudiante puede visualizar los cursos que tiene disponibles para su realización. Una vez completados, no puede volver a verlos.

### Completar curso

El estudiante puede completar un curso en el cual está inscripto. En el proceso, deberá responder todas y cada una de las preguntas que posea dicho curso. Una vez finalizado el mismo, obtendrá una calificación.

### Obtener su promedio

El estudiante puede acceder a su promedio. Este es el resultado obtenido en base a todos los cursos que ha realizado.

## Cursos

Es la estructura que nuclea tanto a Administradores, Estudiantes e Instructores.

En el curso se almacenarán:

- Una lista de estudiantes inscriptos
- Un instructor a cargo del curso
- Una lista de preguntas
- Una lista de calificaciones.

## Preguntas

Existen 3 tipos de preguntas, Verdadero o Falso, Por valor y Opción Múltiple. Cada pregunta retorna verdadero (en caso de ser afirmativa la respuesta) y falso (en caso de ser incorrecta la respuesta)

### Verdadero o Falso

El usuario debe responder 1 para verdadero y 2 para falso.

### Por valor

Este tipo de pregunta, recibe un valor por teclado y debe coincidir con la respuesta indicada por el instructor.

### Opción múltiple

El usuario recibe una pregunta y 3 opciones, pudiendo elegir cualquiera de ellas.

## Otras clases

Para la realización de este sistema, además se utilizaron otros tipos de clases. Las detallamos a continuación:

- Clase genérica ListMgmt
  - Esta clase se utiliza para todo el manejo de listas, siendo prácticamente similar a lo que puede hacer las colecciones de tipo List, su utilización está dada en base a poder utilizar lo aprendido en clase.
- Clase genérica de Almacenamiento en archivos
  - Esta clase, entregada por la cátedra, fue modificada para poder utilizar nuestra clase ListMgmt como método de entrada.
- Interfaz de respuesta de preguntas
  - En este caso, la interfaz no era necesaria. El único método que envía para su utilización, en principio se heredaba de una súper clase pero, al igual que en el caso de ListMgmt, se forzó su utilización.
- Clases estáticas para los menú
  - Estas clases, se utilizan para generar los menús y su interacción con el usuario final. Son estáticas, para no tener que instanciarlas.

## Funcionalidades pendientes

El sistema es bastante acotado en su funcionalidad. Estamos al tanto de esto, pero se desarrolló una estructura lo suficientemente escalable, como para poder implementar nuevas funcionalidades, como pueden ser:

- Modificación de preguntas
- Listados extra
- Ver nota del curso realizado en el momento
- Obtener promedio de curso
- Modificar calificación
- Etc.

## Manual de usuario

Para la generación desde 0 (sin archivos creados) de un administrador, se debe realizar lo siguiente:

1. Quitar el comentario multi-línea que inicia en la línea 20 y finaliza en la línea 27 de nuestra clase Main.
2. Comentar desde la línea 30 hasta la 35.
3. Iniciar el sistema y realizar el login (DNI "99", Password "12345")
4. Cerrar el sistema
5. Revertir los cambios en los pasos 1 y 2.

Listo, nuestro usuario Admin ha sido creado y persistido.

## Administrador

Una vez realizado el login con el administrador:

1. Crear Estudiantes
  - a. Crea un nuevo estudiante, el cual no está asignado a ningún curso
2. Crear Instructores
  - a. Crea un nuevo instructor, el cual no es titular de ningún curso
3. Crear Cursos
  - a. Crea un nuevo curso, el cual no tiene estudiantes ni instructores asignados a él.

Cabe aclarar, que para realizar cualquiera de estas acciones, se solicitarán datos a ingresar por teclado desde la consola.

## Estudiante

Una vez realizado el login con el estudiante:

1. Ver cursos pendientes
  - a. Se mostrará la lista de cursos pendientes que posee el estudiante
2. Completar un curso
  - a. El estudiante puede completar alguno de los cursos que posee pendientes. Se le solicitará el ID del mismo.
3. Ver promedio
  - a. Se le mostrará el promedio basado en las calificaciones obtenidas en los cursos ya realizados.

## Instructor

Una vez realizado el login con el instructor:

1. Crear y asignar preguntas a un curso
  - a. Se le solicitará el nombre del curso al que le desea crear y asignar preguntas. El curso debe estar bajo su titularidad.
  - b. Se le preguntará que tipo de pregunta desea crear. Debe seleccionar el tipo por teclado. Un curso puede tener preguntas de distinto tipo.
  - c. Debe completar los campos que le solicite el sistema
  - d. Al finalizar la creación de la pregunta, el sistema le preguntará si desea generar otra pregunta para el mismo curso. Puede crear tantas preguntas como quiera, sin importar el tipo.
2. Asignar alumnos a cursos
  - a. Se le solicita tanto el nombre del curso, como el DNI del usuario a agregar.  
Aclaración: No se pueden agregar estudiantes a cursos que no posean preguntas, como tampoco se pueden asignar otros instructores como si fueran alumnos.
  - b. Solo puede asignar estudiantes a cursos que estén bajo su titularidad.
3. Eliminar alumnos de cursos.
  - a. Bajo el mismo concepto que el punto anterior, se le solicitará el nombre del curso y el DNI del estudiante.
  - b. Si el estudiante ya poseía una calificación en dicho curso, esta también será removida.
  - c. No se pueden remover estudiantes que no estén inscriptos en el curso.

Links:

- Trello: <https://trello.com/b/6XrM5MBz/herrada-valverde>
- GITHUB: [https://github.com/DrFunkMDQ/TP-Final-Laboratorio3-Herrada\\_Valverde](https://github.com/DrFunkMDQ/TP-Final-Laboratorio3-Herrada_Valverde)