

**TECNOLÓGICO DE COSTA RICA  
SEDE SAN CARLOS  
ESCUELA DE COMPUTACIÓN  
PROGRAMACIÓN ORIENTADA A OBJETOS**



**SEGUNDA TAREA PROGRAMADA:  
SISTEMA GESTOR DE EXÁMENES**

**PRESENTADO POR:**

**DANIEL MURILLO PORRAS    200854763**

**SANTA CLARA, SAN CARLOS  
22 DE NOVIEMBRE DE 2013**

## Contenido

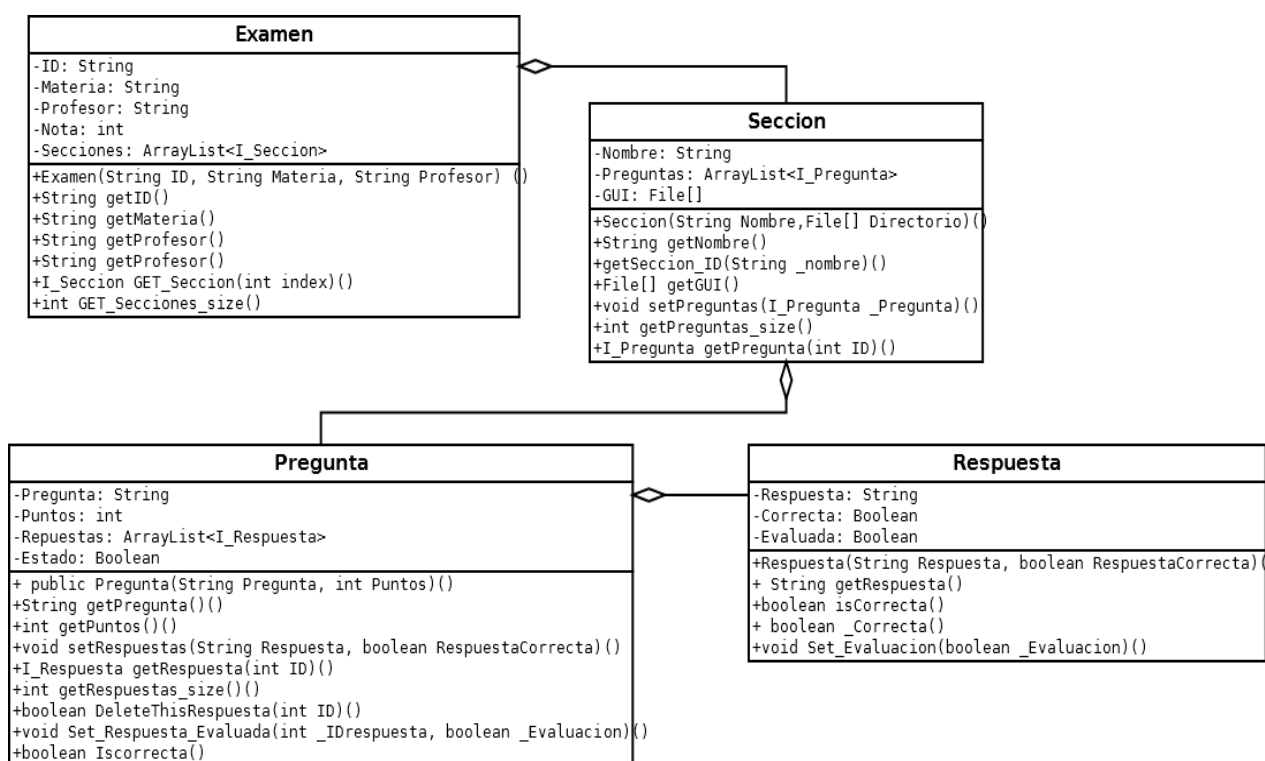
Descripción del Problema: .....	3
Solución del Problema: .....	3
Diagramas y Estructuras:.....	<b>¡Error! Marcador no definido.</b>
Algoritmo de Ford/Fulkerson : .....	<b>¡Error! Marcador no definido.</b>
Algoritmo de Dinic : .....	<b>¡Error! Marcador no definido.</b>
Diagramas de Flujo:.....	<b>¡Error! Marcador no definido.</b>
Análisis de Resultados: .....	<b>¡Error! Marcador no definido.</b>
Conclusiones .....	7
Recomendaciones: .....	<b>¡Error! Marcador no definido.</b>
Bibliografía .....	<b>¡Error! Marcador no definido.</b>

## Descripción del Problema:

Se solicita la implementación de un sistema capaz de soportar la carga de archivos de tipo java class, el cual permitirá la creación de exámenes de forma dinámica durante el tiempo de ejecución. Además dichos exámenes deberán soportar ser evaluados usando el prototipo de interfaz que se haya definido durante la creación y además estos deberán retornar la calificación obtenida.

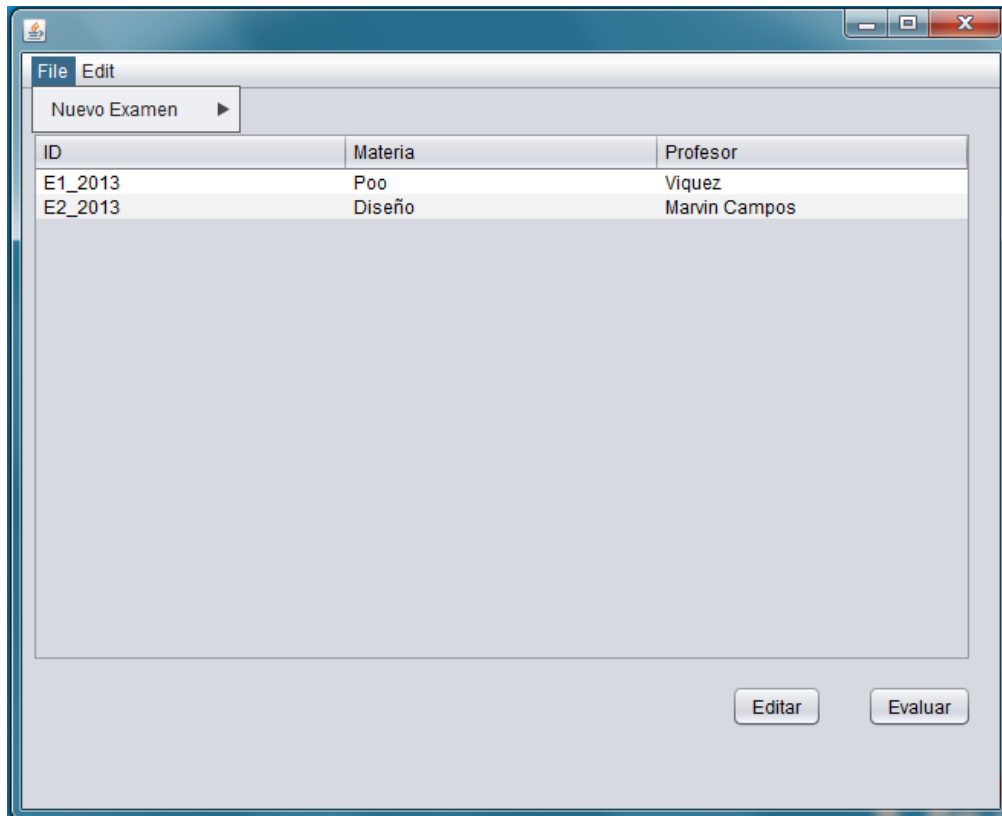
## Solución del Problema:

Mediante el uso de **Java Classloader**( Java Runtime Environment ) el cual carga dinámicamente clases Java en la Java Virtual Machine, se cargaran dinámica cada una de los prototipos de interfaz que el usuario desee, además se usa la extensión **ArrayList** para realizar el almacenamiento de todos los objetos resultantes. Apolándose de los interface para organizar la programación, obligar a que ciertas clases utilicen los mismos métodos y sobre todo a establecer relaciones entre clases que no estén relacionadas.

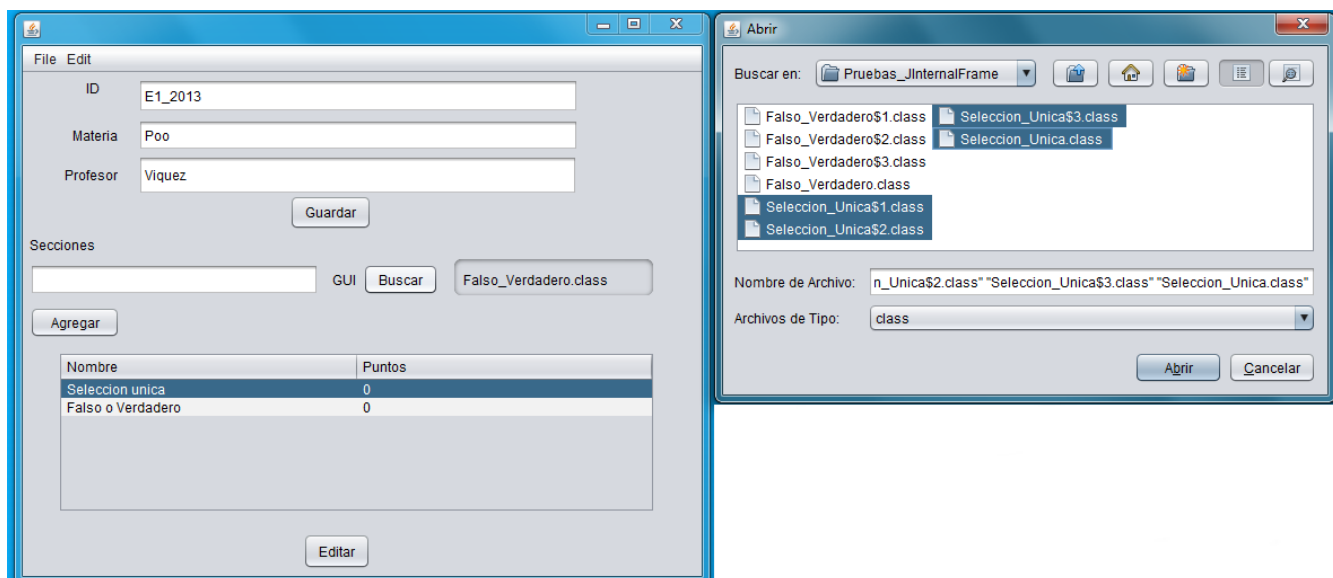


## Manual de Usuario:

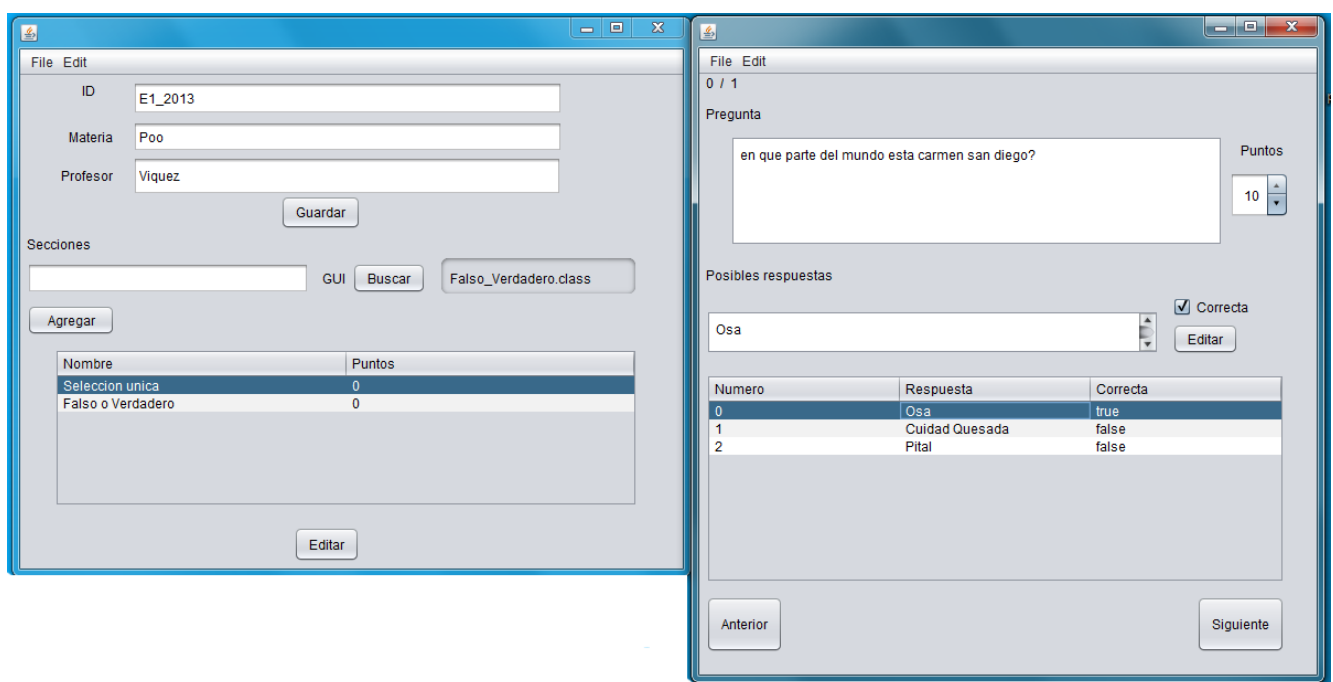
El sistema presenta una ventana de inicio en la cual el usuario puede seleccionar cualquiera de exámenes que estén debidamente creados o bien crear un nuevo examen.



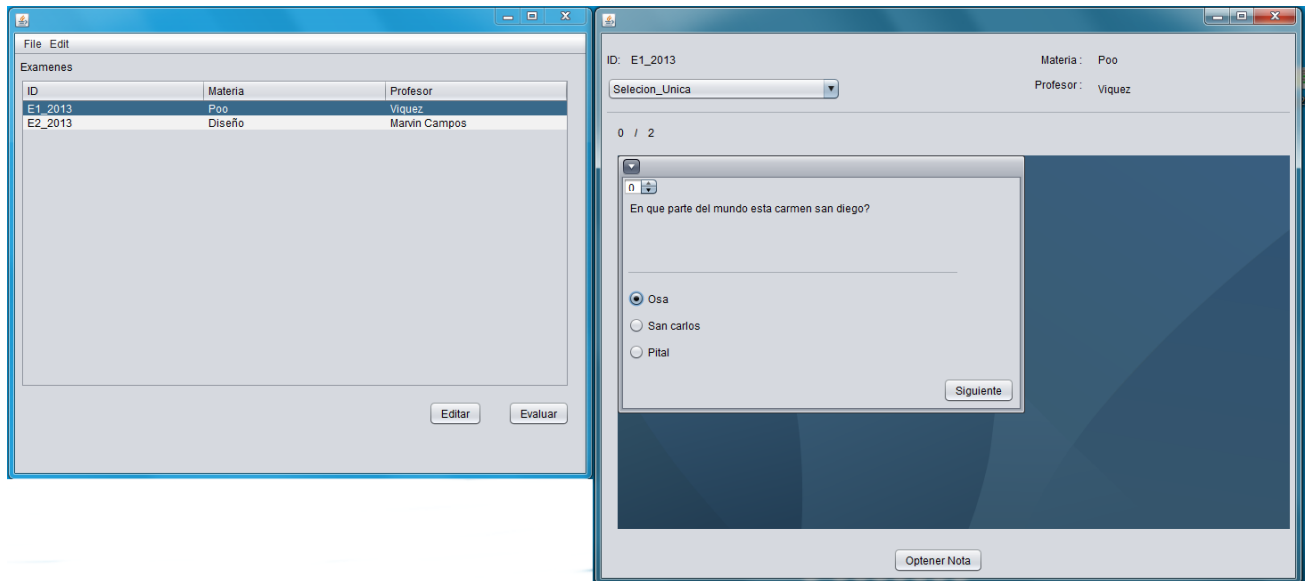
En caso que el usuario escoja la opción de crear un nuevo examen, el sistema le presentara una ventana donde ingresando cada uno de los datos solicitados podrá crear el examen y las secciones que este contendrá; para cada una de las secciones se deberá especificar la interfaz que se desea utilizar simplemente buscándola dentro de los directorios del sistema.



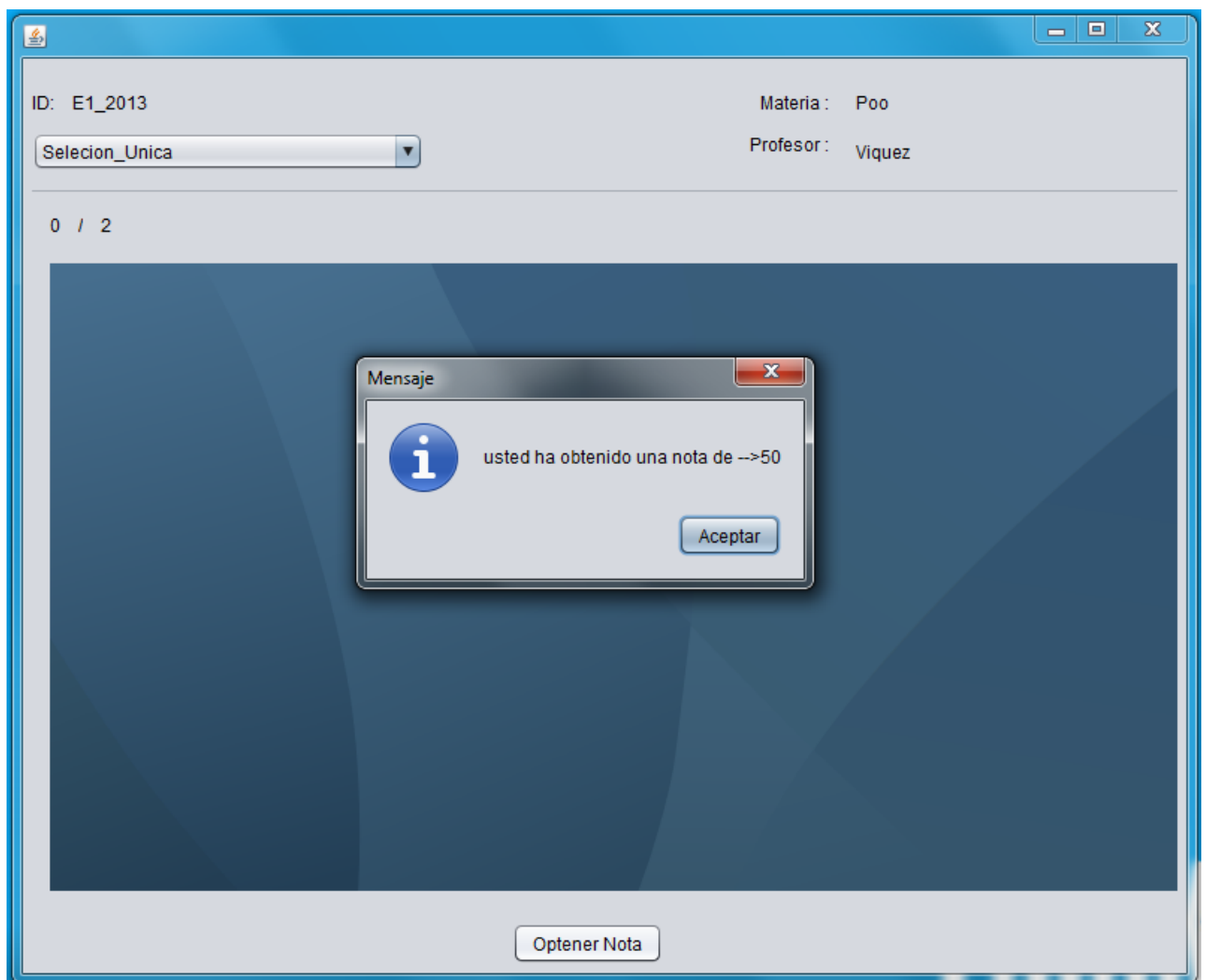
Una vez definidas las secciones que contendrá el examen, el usuario podrá ingresar todas aquellas preguntas que desee y una vez finalizado solo deberá cerrar la ventana y realizar el mismo proceso para cada una de las secciones.



En caso que se quiera resolver un examen, bastara con seleccionarlo y este automáticamente le cargara cada una de las secciones que este contenga usando la interfaz gráfica que se solicitó.



Una vez finalizado la evaluación presionado la opción de “obtener nota “ mostrara la nota obtenida.



## Conclusiones

Durante el transcurso del curso se ha enseñado sobre las ventajas de la programación orientada a objetos y el futuro que esta tecnología representa que reúne las ventajas de la herencia favoreciendo el mantenimiento y la extensión de las aplicaciones. ¿Por qué? Porque al definir interfaces permitimos la existencia de variables polimórficas y la invocación polimórfica de métodos, ya que un aspecto fundamental de las interfaces es separar la especificación de una clase (qué hace) de la implementación (cómo lo hace) lo que da lugar a programas más robustos y con menos errores.

## Bibliografía

- Ferraz, J. (s.f.). *Desmitificando los ClassLoaders de Java*. Obtenido de <http://www.masterdlabs.es/desmitificando-los-classloaders-de-java/>
- oracle. (s.f.). *Class ClassLoader*. Obtenido de <http://docs.oracle.com/javase/6/docs/api/java/lang/ClassLoader.html>
- wikibooks. (s.f.). *Programación en Java* . Obtenido de Interfaces: [http://es.wikibooks.org/wiki/Programaci%C3%B3n\\_en\\_Java/Interfaces](http://es.wikibooks.org/wiki/Programaci%C3%B3n_en_Java/Interfaces)