Instituto Tecnológico de Costa Rica



Escuela de Computación

Programación Orientada a Objetos

Oscar Víquez Acuña

Daniel Berrocal Ramírez

201049486

Jorge Rojas Aragonés

200969830

San Carlos, 22 de Noviembre de 2013

# Objetivo General:

* Mediante pruebas de implementación de código, aprender las bondades del lenguaje Java y con este proceso familiarizarse con las características de la Programación Orientada a Objetos.

## Objetivos Específicos:

* Aplicar conceptos de diseño de aplicaciones utilizando las características principales del paradigma.
* Consolidar la programación orientada a objetos mediante el lenguaje Java.
* Generar una herramienta que canalice todos los conceptos estudiados del tema.

# Introducción

La orientación a objetos ayuda a tener un modelo más cercano de la realidad de muchos problemas. Mediante un adecuado soporte de abstracciones de datos es posible tener una visión de objetos que tienen operaciones para manipularlos, sin tener acceso a detalles internos y a la información que encapsulan.

Una forma efectiva para implementar polimorfismo y genericidad de tipos en Java es mediante la definición de interfaces que son luego implementadas por diferentes clases. De esta manera un programa puede trabajar con objetos del tipo de la interfaz sin saber específicamente a qué clase pertenecen.

# Descripción del Proyecto:

El proyecto consiste en la elaboración de un sistema que le permita al usuario profesor crear y publicar exámenes para que sean respondidos por los usuarios estudiantes. El sistema debe proveer a los usuarios una interfaz sencilla sin caer en la simplicidad, que facilite su adecuado uso esto se llevara a cabo mediante la implementación de JInternalFrames.

El tema central del proyecto consiste en que el sistema cuente con la capacidad de cargar dinámicamente clases externas al sistema, esto para la creación de preguntas.

## Especificaciones del Sistema:

La solución del sistema requiere de varias características necesarias entre las cuales están:

1. Cargar dinámicamente clases externas.
2. El usuario profesor podrá crear varios exámenes.
3. Cada examen puede contar con varias secciones.
4. Cada sección podrá contar con varias preguntas.
5. El usuario podrá ver la nota obtenida al final del examen.

# Descripción de la Solución:

Para llevar a cabo la elaboración de este proyecto, se pasó por varias etapas de desarrollo:

* Investigación:

En esta etapa se realizó una intensa búsqueda de todo tipo de información que proporciona algún apoyo para la confección del sistema, entre la información que se buscó se encuentra:

* Manejo de JInternalFrames.
* Paso de mensajes entre JInternalFrames.
* Programación de interfaz dinámica.
* Manejo de clases externas.
* Carga dinámica de clases.

* Planteo de la Solución:

En dicha etapa, con toda la información que se recaudó en la etapa de investigación, se procedió a plantear posibles maneras de llegar a una solución, discutiendo, probando, y elaborando pequeñas pruebas de código, mutuamente se llegó a la decisión de cuál posible solución implementar.

* Desarrollo:

Teniendo en mente de qué manera darle solución al problema, se inicia con el desarrollo de la solución, en esta etapa se implementa todo el conocimiento adquirido en clase y  la información recaudada en la etapa de investigación. En la etapa de planteo de la solución se decidió por utilizar diferentes tipos de estructuras para guardar los datos que maneja el sistema. Estos datos se manejan en listas ya que se facilita el guardado y recuperación de los mismos.

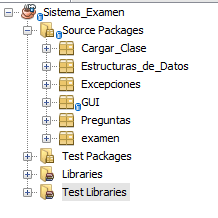
* Prueba:

En esta etapa, se le realizaron distintas pruebas al sistema para buscar todo tipo de errores y buscar posibles soluciones para dichos errores.

**Detalles en de la solución**

La aplicación esta implementada en lenguaje JAVA y todo el proceso de desarrollo y prueba se hizo en el ambiente de desarrollo NetBeans 7.4.

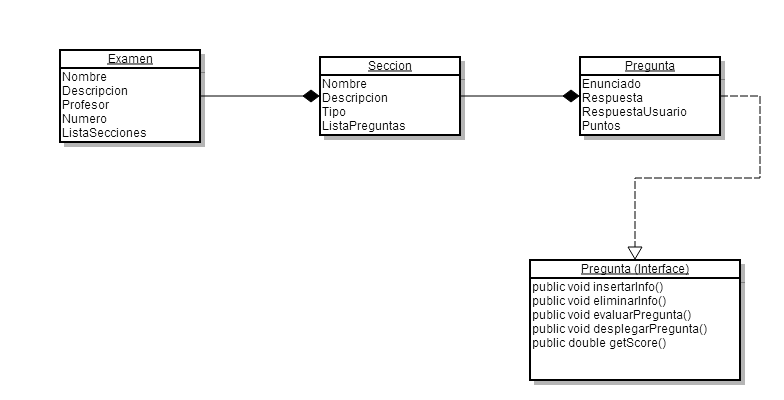
El proyecto está organizado en paquetes para tener un orden específico de las clases.



En el tiempo de desarrollo se implementaron 2 excepciones para evitar fallos al sistema. Estas son:

* ListaFueraDeRangoException.
* NullException.

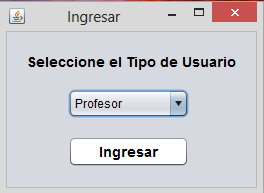
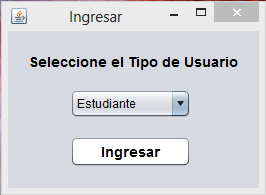
# Diagrama de Clases

****

**Manual de Usuario**

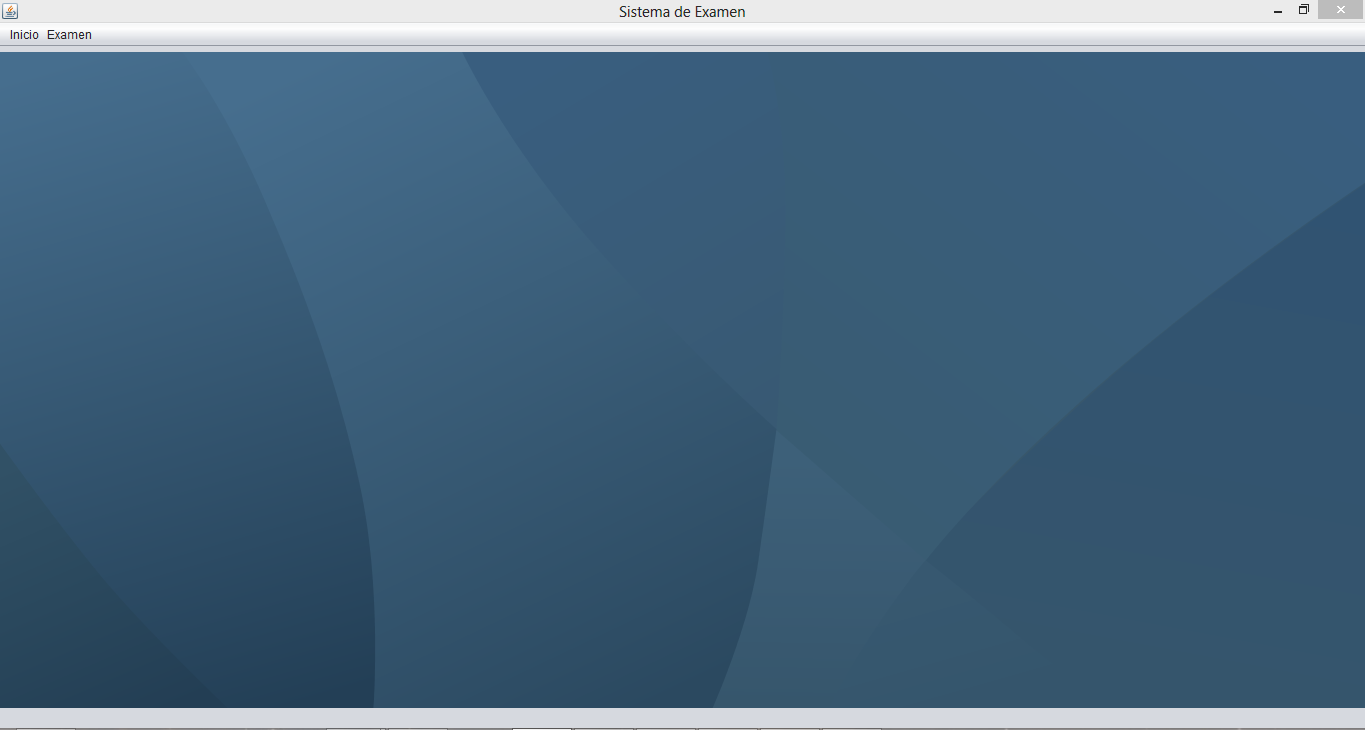
La utilización de la aplicación es sumamente sencilla, para ayudar aún más al usuario se implementó una barra de estado al pie de la aplicación para guiarlo.

Pasos para su correcto uso:

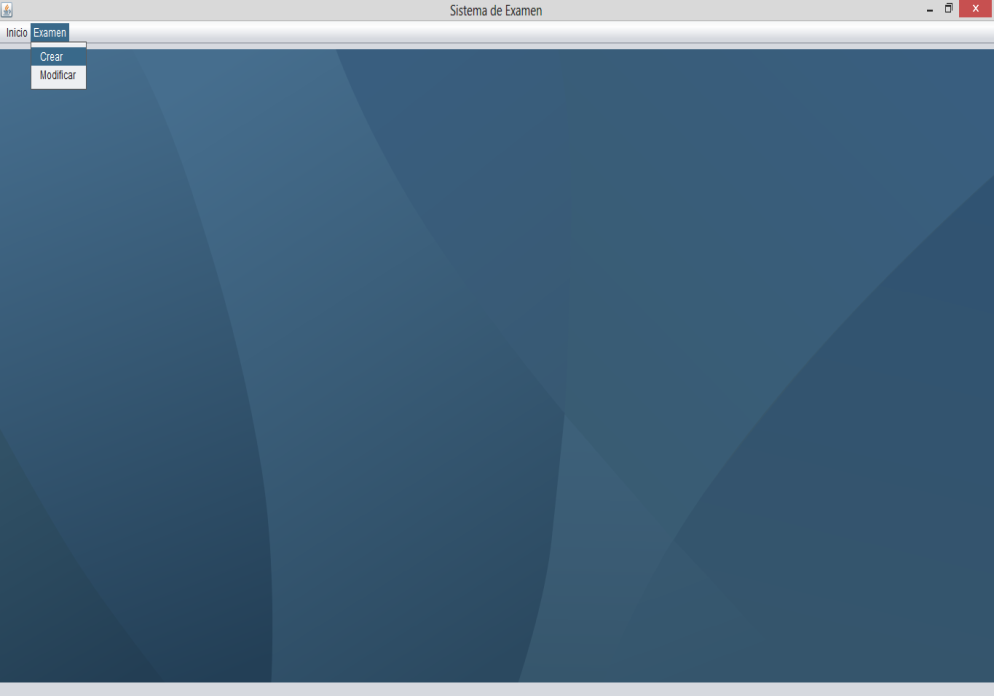
 

La aplicación está diseñada para ser usada por 2 diferentes usuarios, el usuario profesor quien es el que crea los exámenes y el usuario estudiante el cual contestas los mismos.

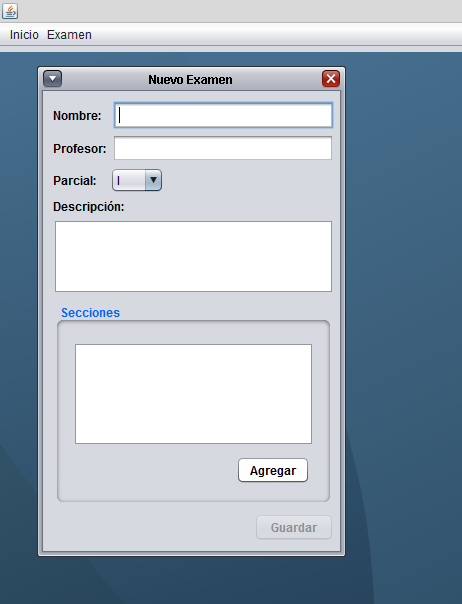
Una vez que el usuario ingresa (sea estudiante o profesor) vera la siguiente pantalla.



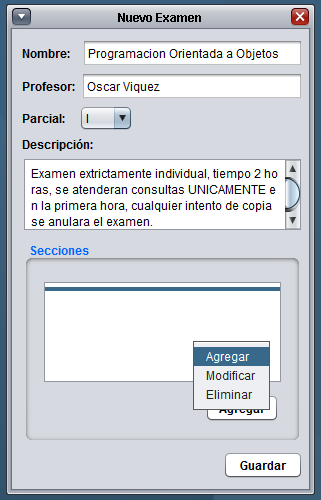
Hasta este momento la interfaz es igual para los 2 usuarios. La diferencia radica en las opciones que tiene sobre los exámenes, ya que el usuario profesor puede crear y modificar uno o varios exámenes y el usuario estudiante solo puede contestar los mismos.



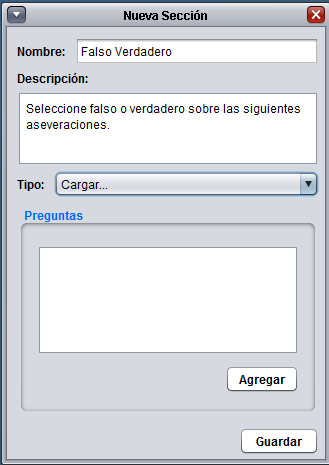
Si el usuario profesor decide crear un nuevo examen se le mostrara la siguiente ventana interna.



Para agregar una nueva sección basta con presionar click derecho, como se muestra en la siguiente imagen, de la misma manera se puede modificar o eliminar una sección del examen.



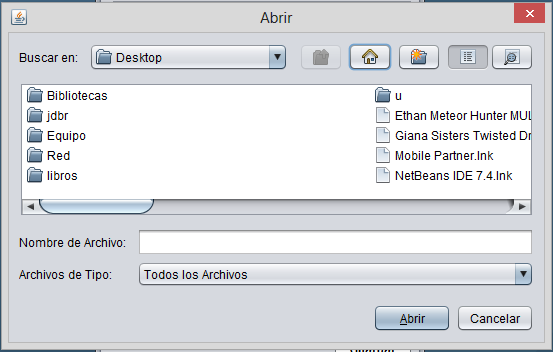
# Al seleccionar la opción de agregar una nueva sección, se mostrara la siguiente ventana interna.



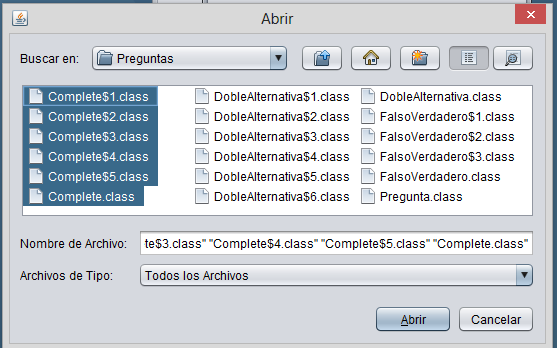
Se llenan los campos con la información correspondiente y se procede a cargar dinámicamente la estructura de las preguntas.

Debe seleccionar la opción de cargar, al hacerlo aparecerá una ventana donde podrá localizar los archivos .class que desee cargar.

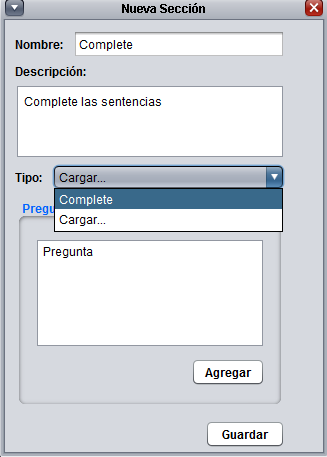
Puede cargar cuantas clases externas se desee.



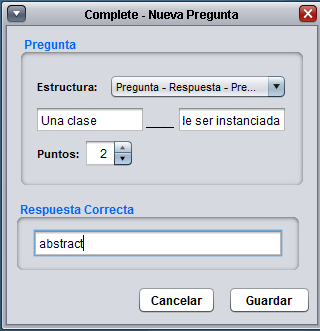
Cuando encuentre los archivos que desea cargar, selecciónelos todos de la siguiente manera.



Una vez cargada la clase, el ComboBox se actualizara y esta aparecerá en el mismo para poder ser elegida.



Al elegir el tipo de pregunta (la recién cargada) y presionar el botón de agregar se levantara la interfaz correspondiente a la misma.

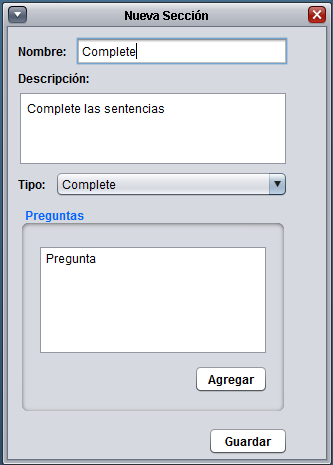


Esta interfaz puede variar mucho dependiendo del tipo de pregunta que se cargue.

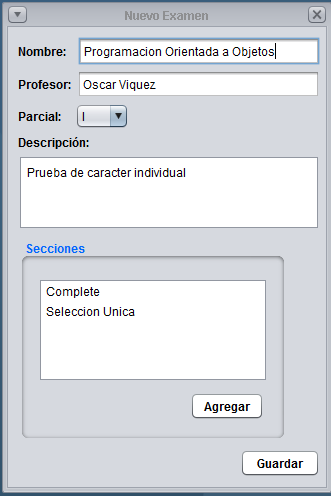
Se editan los campos y la estructura de la misma y se presiona guardar.

Esta podrá ser editada o borrada de la sección si el usuario así lo desea.

# Al finalizar de crear la pregunta y presionar el botón de guardar la ventana se cierra y se actualiza la lista de preguntas de la sección.

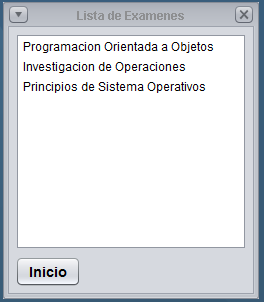


Cuando se terminen de crear o editar todas las preguntas correspondientes a la sección se presiona el botón de guardar, esto actualizara la lista de secciones del examen.

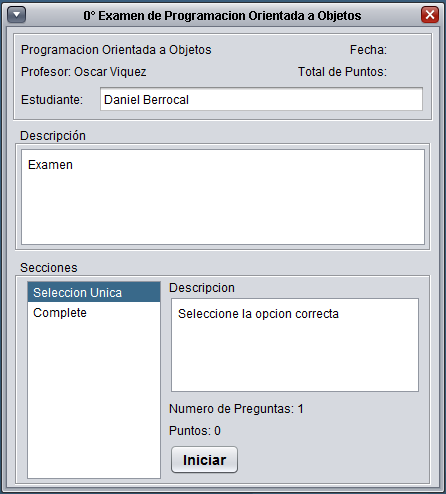


# Luego de haber creado o editado el examen se presiona el botón de guardar esto actualizara la lista de exámenes creados.

Cuando se procede con el cambio de usuario (de profesor a estudiante) la lista de exámenes habilitados para ser respondidos estará actualizada.



El usuario estudiante selecciona el examen que desea contestar y la aplicación le muestra la interfaz necesaria para la evaluación del mismo.



# Conclusión

Durante el desarrollo de este proyecto, se logran rescatar muchos aspectos importantes para cualquier estudiante de la carrera de ingeniería en computación, pero el que más importancia parece tener es, el impacto a nivel académico, que se da cuando se analizan los procesos que lleva que se dan cuando una herramienta facilita las actividades que se realizaban manualmente.

Es importante rescatar el valor académico que tiene la asignación de este tipo de trabajos para el desarrollo como futuros profesionales.

# 

# Bibliografía

Rodriguez, R. V. (n.d.). My Java Zone: JDesktopPane y JInternalFrame. My Java Zone: JDesktopPane y JInternalFrame. Retrieved November 10, 2013, from http://www.myjavazone.com/2010/07/jdesktoppane-y-jinternalframe.html

Meza Martines, J. I. (n.d.). Jorge Ivan Meza Martinez: Implementando la carga dinámica de clases. Retrieved November 21, 2013, from http://blog.jorgeivanmeza.com/2012/04/implementando-la-carga-dinamica-de-clases-en-processing/