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, 13 de Octubre de 2013

Contenido

[Objetivo General: 3](#_Toc369220102)

[Objetivos Específicos: 3](#_Toc369220103)

[Introducción 4](#_Toc369220104)

[Descripción del Proyecto: 5](#_Toc369220105)

[Especificaciones del Sistema: 5](#_Toc369220106)

[Alcances: 6](#_Toc369220107)

[Descripción de la Solución: 7](#_Toc369220108)

[Estructura 8](#_Toc369220109)

[Diagrama de Clases 9](#_Toc369220110)

# 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.
* 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:

La confección de los horarios de clase de los cursos del plan de la carrera de Ingeniería en Computación está a cargo de la coordinación de dicha carrera, la confección de dichos horarios se ha llevado a cabo manualmente desde hace bastante tiempo atrás, se ha hecho la solicitud de diseñar y elaborar un sistema que lleve a cabo este proceso de forma automática, esto para agilizar y no sobre cargar las labores de la coordinación de la carrera.

La coordinación desea generar automáticamente los horarios a partir de determinadas fuentes de fuentes de información tales como:

* Las asignaturas de que consta cada semestre.
* Las asignaturas a impartir por cada profesor.
* Las aulas disponibles para impartir las clases.

De cada profesor interesa conocer su nombre, cédula y horario, así como el departamento al que pertenece. Cada departamento tiene un nombre. Las asignaturas tienen un nombre y un número de créditos.

Acerca de las asignaturas (cursos) existen 2 tipos; las teóricas y las prácticas; de igual manera existen dos tipos de aulas para cada tipo de asignatura.

## Especificaciones del Sistema:

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

1. El sistema debe realizar una validación del usuario que desee ingresar (Login).
2. Consultas de horarios según el semestre a escoger y por coordinador, profesor o estudiante.
3. El sistema también debe permitir la consulta de horarios por parte de los alumnos para que puedan ver los detalles asociados a un curso en particular (tipo curso, aulas, características en general).

# Alcances:

Durante el desarrollo de la aplicación se busca cumplir con los siguientes alcances.

1. Cada profesor tiene un nombre, cédula y horario, y pertenece a un departamento/unidad.
2. Cada departamento tiene un nombre.
3. Cada asignatura tiene un nombre y un número de créditos (1 crédito equivale a 3 horas).
4. Las asignaturas pertenecen a una Carrera ( enfocado a la carrera de computación, diferenciar los cursos de computación con los cursos de otros departamentos como ciencias y letras)
5. Existen dos tipos de asignaturas: teóricas y prácticas.
6. Las teóricas tienen una página Web.
7. Las prácticas tienen material de apoyo y sistema operativo a emplear.
8. Existen dos tipos de aulas: para teoría y para laboratorio (asignaturas prácticas).
9. Cada aula tiene un nombre, número de aula, ubicación y capacidad.
10. Las aulas de teoría pueden tener aire acondicionado y varios equipo multimedia
11. Existen 8 semestres, pero podrían ser más o menos.
12. Los semestre pueden ser par o impar
13. Cada semestre tiene una cantidad de asignaciones el cual puede variar
14. Las asignaturas pueden pertenecer a cualquier semestre sin importar el semestre en el que están estipuladas por el programa de la carrera.
15. Los días de clase van de lunes a viernes.
16. El horario de clases es de  7:00 a 11:30 y de 12:30 a 16:00, puede haber horarios nocturnos.
17. Las asignaturas se imparten en bloques de 50 minutos contemplando además las pausas entre asignaciones de 5 minutos.
18. Existen 3 tipos de usuarios: coordinador, profesor y estudiante
19. Los profesores deben introducir la información de las asignaturas que imparten.
20. Los profesores definen cuales son los horarios que tienen disponibles para impartir las asignaturas que les corresponde.
21. El horario generado se crea a partir del semestre seleccionado por el coordinador
22. El horario se genera a partir de los cursos del semestre, los horarios de disponibilidad del profesor y las aulas disponibles.
23. Los estudiantes solo deben poder consultar horarios de los cursos del semestre que generó el coordinador, tipo curso, aula correspondiente, etc.
24. Si el horario del semestre seleccionado por el estudiante-profesor no ha sido generado por el coordinador, esta opción mostrará “Horario no disponible”.
25. Deberá implementarse un Login que permite verificar si es un estudiante, profesor o coordinador quien accede al sistema.

# 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 los puntos que se investigaron, se encontró información sobre algoritmos que elaboraban la confección de horarios de clases automáticamente, estos algoritmos son llamados “Algoritmos Genéticos”, estos utilizan la combinación de diferentes técnicas para llegar a la solución. Estos algoritmos entran en el área de inteligencia artificial por lo cual resulto complicado interpretarlos.

(Ejemplo:

* http://jupiter.utm.mx/~tesis\_dig/6557.pdf
* http://www.latindex.ucr.ac.cr/mate-18-2/matematica-18-2-02.pdf)

* 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.

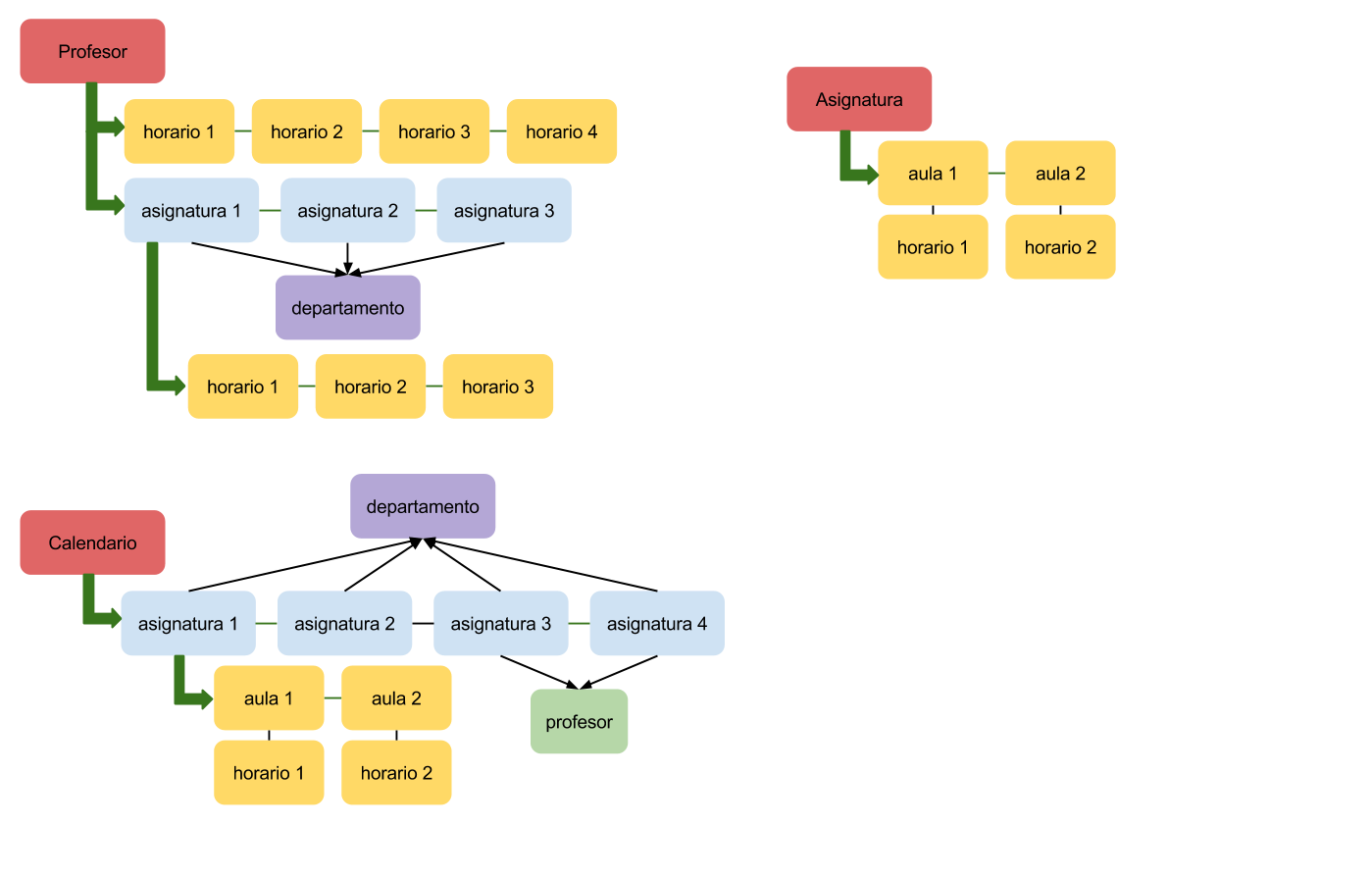
* Desarrollo:

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 cargados desde los archivos. Estas estructuras se guardan en listas que facilitan el guardado y recuperación de los datos.

* Prueba:

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

# Estructura



# Diagrama de Clases

