|  |  |
| --- | --- |
| **Instituto Tecnológico de Costa Rica**  **Ingeniería en Computación**  **Sede Regional San Carlos** | Tercer y Cuarta Tareas Programadas  Lenguajes de Programación |
| Prof. Oscar Víquez Acuña. | Paradigmas Lógico y OO |

**Objetivos**

* Aprender detalles de implementación en varios lenguajes y familiarizarse con la Programación Lógica y Orientada a Objetos.
* Aplicar conceptos de diseño de aplicaciones utilizando las características principales vistas en clase para de los paradigmas involucrados en el proyecto.
* Generar una herramienta que canalice todos los conceptos estudiados del tema.

**Descripción del proyecto**

La coordinación de la Carrera de Ingeniería en Computación, con sede en San Carlos, se quiere encargar de su propia confección de la guía de horarios de clase de los cursos que la carrera imparte en las aulas/laboratorios que tiene disponibles. Este proceso lejos de ser manual, debe ser automático de manera que se diseñe e implemente un software que permita automatizar esta gestión.

El principal objetivo para la coordinación, es la generación automática de los horarios sin choques, a partir de determinadas informaciones como:

Las asignaturas que se estarán impartiendo cada semestre.

Las asignaciones de profesor para cada asignatura a impartir.

Las aulas disponibles para impartir las clases con sus características.

De cada profesor interesa conocer su nombre, cédula, disponibilidad de horario, cursos que imparte.

Las asignaturas tienen un nombre, un tipo de asignatura (normal o laboratorio – se usa para determinar el tipo de aula) y un número de créditos y el semestre al que pertenecen. Así mismo, para cada asignatura se podría definir para cada periodo en particular, si se quiere separar en horarios de un día 4 horas o en dos días de 2 horas. Nótese que las asignaturas no tienen información del profesor que las imparte puesto que podría ser variante.

Las aulas tienen un nombre, número de aula, capacidad, tipo de aula.

Los posibles semestres son ocho, esto es particularmente de interés para saber a cuál semestre pertenece cada asignatura y para la impresión gráfica del resultado. Así mismo se deben estipular un número de asignaciones o cursos por semestre.

Los días de clase van de lunes a viernes, y las horas posibles son de 7:00 a 11:30 y de 12:30 a 16:00. Las asignaturas se imparten en bloques de 50 minutos contemplando además las pausas entre asignaciones de 5 minutos.

Toda la información anterior con datos reales de los cursos de esta carrera, debe ser analizada y estructurada para ser cargada en la base de datos de conocimiento de Prolog para el posterior cálculo de horarios de la forma en que lo determine cada programador. Es altamente probable que exista un manejo dinámico de datos para cada consulta que se quiera hacer.

Orientación a Objetos

El backend para el cálculo automático de guías de horarios se realizará en Prolog, sin embargo, todo el mantenimiento y despliegue de los datos e información deberá realizarse en un lenguaje Orientado a Objetos de su preferencia con el debido enlace con en interprete de Prolog que decida utilizarse para el proyecto.

Se debe implementar siguiendo los principios de diseño de lenguajes de programación Orientados a Objetos dando énfasis en el uso correcto de encapsulamiento y ocultamiento de información. Así mismo será requisito obligatorio implementar al menos tres patrones de diseño de Orientación a Objetos no triviales con la respectiva justificación para su uso en la documentación

**Detalle de Interfaz de Usuario**

El programador debe desarrollar los mecanismos adecuados para administrar y mostrar la información del sistema en una interfaz que permita realizar lo siguiente:

1. Administrar los datos (inserción, borrado y modificación) para cursos, profesores y aulas de manera que cuando el sistema arranque dicha información se cargue de algún almacenamiento que se defina (se recomienda alguna base de datos simple que no requiera de configuración para efectos de revisión ágil – por ejemplo SQLite <https://www.sqlite.org/index.html>). Debe considerarse que algunos de los datos se cargan una única vez y que podrían eventualmente eliminarse o modificarse, pero algunos otros elementos que puedan ser de interés se solicitarán y almacenarán por cada semestre y puede que no sean de almacenamiento permanente. Queda a criterio del programador.
2. Asociación de profesores por semestre a cursos a partir de la disponibilidad de horario del profesor y los cursos que imparte cada uno de ellos. CONSULTA EN BACKEND DE PROLOG.
3. Despliegue de datos del punto 1 y punto 2 inclusive, de manera de listado eficiente y fácil de manejar. Se recomienda una especie de interfaz con despliegue único y botones de siguiente y anterior. Si se quiere utilizar imágenes, se puede hacer.
4. Consultas de horarios posibles (sin choques) a partir de una lista de cursos que se seleccionen (aún cuando no correspondan al semestre). Si se seleccionan muchos cursos y no es posible encontrar una solución debe indicarse. Esta búsqueda no toma en cuenta profesores. Si el resultado de la búsqueda muestra diferentes escenarios, deben mostrarse gráficamente de una forma amigable (solo se puede ver un resultado a la vez); si la cantidad es muy extensa, podría limitarse a un número de despliegues. CONSULTA EN BACKEND DE PROLOG.
5. El sistema también debe permitir la consulta de posibles horarios por semestre (también sin choques) independientemente del profesor que sea que imparta los cursos. Quiere decir que el sistema debe buscar las opciones de horarios para cursos, aulas y profesores posibles siguiendo la restricción de que sea para el semestre seleccionado. Todos los elementos necesarios para esta búsqueda deben haber sido suministrados o suministrarse en el momento de la consulta (por ejemplo los mencionados en el punto 2). Igualmente, si el resultado de la búsqueda muestra diferentes escenarios, deben mostrarse gráficamente de una forma amigable (solo se puede ver un resultado a la vez); si la cantidad es muy extensa, podría limitarse a un número de despliegues. CONSULTA EN BACKEND DE PROLOG.

**Notas Finales para la Tarea:**

* La tarea es individual. Bajo ninguna circunstancia se permitirán ***COPIAS*** de trabajos.
* Documentación: un único video con el mismo formato de los trabajos anteriores. Debe considerar aspectos técnicos de implementación de manera descriptiva general y un manual de usuario/instalación simple.
* Fecha de entrega para la tarea aplicable para los proyectos de Prolog y Java: Viernes 31 de Julio de 2020.
* Habrá documentos de evaluación individuales para cada paradigma.
* Hora de entrega 10:00 pm por el Tec-Digital.