

Curso:

IC-2101 Programación Orientada a Objetos

Profesor:

Ing. Carlos Benavides

Ing. Luis Pablo Soto Chaves

Primer proyecto programado

II Semestre, 2018



I. Objetivos del proyecto

Objetivo general:

- Desarrollar una solución de software bajo el paradigma de programación orientada a objetos, haciendo énfasis en el modelado y la aplicación de buenas prácticas.

Objetivos específicos:

- Modelar el dominio del problema mediante un diagrama de clases, identificando claramente clases, atributos, métodos y relaciones.
- Programar una solución de software utilizando el lenguaje de programación Java.
- Desarrollar habilidades de investigación en temas relacionados con:
 - Manipulación de archivos en formato .XML o JSON para lectura/escritura de datos.
 - Envío de correo electrónico y generación de archivos PDF..
 - Interfaces gráficas de usuario.
 - Representación de datos mediante gráficos de barras, circular, entre otros.
 - Servicios de mensajería a dispositivos móviles.
- Reconocer y poner en práctica buenas prácticas/recomendaciones en el diseño de interfaces gráficas de usuario.
- Fomentar el trabajo en equipo, el desarrollo de habilidades de liderazgo, planeamiento, resolución de problemas y comunicación efectiva.

II. Contexto y funcionalidades esperadas

La Unidad de Transportes del Tecnológico de Costa Rica tiene a su cargo la flota vehicular de la Institución y es la encargada de administrar los servicios de transporte.

De forma constante se adquieren vehículos para atender la demanda de transporte institucional. Los datos más importantes para los vehículos son los siguientes: placa, año de fabricación, color, marca, capacidad, el kilometraje, número de vin, la sede de pertenencia del vehículo y un estado que puede asumir solamente 3 posibles valores:

- En servicio. Indica que el vehículo se encuentra disponible para prestar servicios de transporte.
- En mantenimiento. Indica que el vehículo se encuentra recibiendo algún servicio de mantenimiento.
- Fuera de servicio. Indica que el vehículo ya no se utiliza para la prestación de servicios.

Los vehículos son sujetos de servicios de mantenimiento durante su vida útil, siendo estos mantenimientos de tipo preventivo o correctivo. Por lo tanto, para cada vehículo se lleva un listado de todos los servicios de mantenimiento.

Cada servicio de mantenimiento se caracteriza por almacenar el identificador de servicio, la fecha de inicio y finalización, el monto pagado, un detalle de la actividad realizada, el tipo de servicio (preventivo o correctivo) y la empresa que realizó el mantenimiento. El sistema de forma interna, debe gestionar el consecutivo de identificadores de servicio, el mismo debe seguir el formato "ID-SM-XXX". Donde XXX es un consecutivo numérico que inicia en 1.

Con respecto a la empresa que presta servicios de mantenimiento, se debe registrar la razón social, el número de cédula jurídica, teléfono y la dirección (indicada como Provincia, Cantón, Distrito y señas).

Obviamente, la razón de ser de esta unidad, es prestar servicios de transporte a funcionarios y estudiantes de la Institución. Para cada persona que usa el servicio se requiere conocer su nombre completo, cédula, dirección, correo electrónico y teléfono.

Los choferes son los únicos autorizados para manejar los vehículos. De cada chofer se debe indicar su nombre, teléfono, dirección de correo, número de cédula y una lista de las licencias que posee. Para cada licencia se requiere su número, fecha de emisión, tipo y fecha de expiración.

La unidad se mantiene en operación continua todo el año, realizando viajes dentro del territorio nacional. Para cada viaje se debe indicar el punto de salida, la fecha y hora de inicio del viaje así como de finalización, el destino, el chofer asignado, el kilometraje inicial y final, un estado (que será explicado en los requerimientos) y la lista de pasajeros. También es necesario registrar la fecha en que se ingresa la solicitud de viaje. El consecutivo de viajes es de la forma "VIA-XXX".

En este sistema se han identificado los siguientes roles de usuario:

- Administrador
- Secretaria

Para cualquiera de ellos se requiere conocer su nombre completo, departamento o escuela en el cual labora, un nombre de usuario, teléfono y una contraseña.

Actor: Secretaria**Requerimiento 1: Registrar un nuevo pasajero del sistema de transporte**

El sistema debe permitir el registro de nuevos pasajeros que utilicen los servicios de transporte y todos los datos son requeridos.

Aspectos a considerar:

- El sistema no debe permitir la existencia de pasajeros con un mismo número de identificación.
- La aplicación debe tener pre-cargados al menos 10 pasajeros desde el archivo pasajerosDB.XML o JSON

Requerimiento 2: Solicitar servicio de viaje

El sistema debe permitir que se ingrese una solicitud de servicio de transporte (viaje).

Aspectos a considerar:

- Se debe indicar toda la información del viaje, incluyendo la lista de pasajeros. Excepto la asignación del chofer y el vehículo, pues estos datos son asignados por otro actor del sistema. Cuando se crea la solicitud de viaje, el estado del viaje es "En confección"
- Un pasajero no puede ser asignado a varios viajes si los mismos presentan una intersección no vacía en las fechas de inicio y finalización.
- La fecha de inicio de la solicitud del viaje debe realizarse al menos con una antelación de 24 horas de la fecha de ingreso de la solicitud.
- Debe existir al menos un pasajero ingresado para procesar la solicitud de servicio de viaje.
- La aplicación debe almacenar la información de las solicitudes de viaje en un archivo llamado viajesDB.XML o JSON

Requerimiento 3: Listar solicitudes de viaje

El sistema debe permitir que se listen las solicitudes de viaje que un usuario particular ha confeccionado. Los criterios de filtrado son los siguientes:

- Fecha en que se registró la solicitud de viaje (dd/mm/aaaa)

- Por estado de la solicitud de viaje (En confección/Aprobado/Cancelado/No aprobado)
- Por destino del viaje (cadena de caracteres)

Los datos se deben desplegar de forma tabular y corresponden a:

1. ID de la solicitud de viaje
2. Fecha de ingreso al sistema
3. Estado
4. Destino
5. Fecha de inicio del viaje

Aspectos a considerar:

- El sistema debe mostrar todos los registros que cumplan el criterio de búsqueda.
- Solamente deben mostrarse los viajes que un usuario específico ha ingresado al sistema.

Requerimiento 4: Consultar detalle de una solicitud de viaje

El sistema debe permitir que se consulte toda la información relacionada a una solicitud de viaje, incluyendo la lista de pasajeros y el chofer y vehículo asignados (en caso que corresponda). Este requerimiento guarda relación estrecha con el requerimiento número 3.

Aspectos a considerar:

- Si no han sido asignados el chofer y el vehículo, se debe indicar en esos campos "sin asignar".

Requerimiento 5: Cancelar solicitud de viaje

El sistema debe permitir que el usuario cancele la solicitud de un viaje.

Aspectos a considerar:

- Si el estado del viaje es "En confección", el mismo pasa a estado Cancelado.

- Si el estado del viaje es "Aprobado", se debe notificar a los teléfonos de los interesados (chofer y pasajeros) mediante un servicio de mensajería (Telegram, Whatsapp, entre otros). Recuerde, los recursos (chofer y vehículo) asignados pueden ser utilizados en otro viaje si así se quiere.

Actor: Administrador**Requerimiento 6: Registrar un nuevo chofer**

El sistema debe permitir que se ingresen nuevos choferes. Todos los datos del chofer son requeridos y al menos se debe ingresar una licencia.

Aspectos a considerar:

- La fecha de expiración de la licencia no puede ser inferior a la fecha del sistema.
- La aplicación debe tener pre-cargados al menos 5 choferes desde el archivo choferesDB.XML o JSON

Requerimiento 7: Registrar un nuevo vehículo

El sistema debe permitir que se ingrese un nuevo vehículo que estará al servicio de la unidad de transporte. Cuando se registra un nuevo vehículo, su estado por omisión es “en servicio”. Todos los datos del vehículo son requeridos.

Aspectos a considerar:

- No pueden existir vehículos con el mismo número de placa o número de vin (*vehicle identification number*).
- La aplicación debe tener pre-cargados al menos 5 vehículos desde el archivo vehiculosDB.XML o JSON

Requerimiento 8: Listar viajes

El sistema debe permitir que el usuario consulte el listado de solicitudes viaje. Estos deben aparecer de forma tabular y ordenados según la fecha de ingreso de la solicitud de viaje.

Los datos que se deben desplegar de forma tabular y corresponden a:

1. ID de la solicitud de viaje
2. Fecha de ingreso al sistema
3. Estado
4. Destino
5. Escuela o departamento que realiza la solicitud de viaje

Requerimiento 9: Aprobar solicitud de viaje

El sistema debe permitir que el usuario realice la aprobación de una solicitud de viaje, es decir cambiar el estado de la solicitud de viaje a “Aprobado”. Además, en este momento se debe asignar un chofer y un vehículo.

Aspectos a considerar:

- El chofer asignado no puede haber sido asignado a otro viaje dentro de la misma fecha de inicio y finalización.
- El vehículo asignado no puede haber sido asignado a otro viaje dentro de la misma fecha de inicio y finalización.
- La capacidad del vehículo que se asigna no puede ser inferior a la cantidad de viajeros. Debe tomar en cuenta que el chofer incide en la capacidad disponible de un vehículo.
- El sistema debe notificar vía correo electrónico a todos los interesados (chofer, pasajeros y el usuario que creó la solicitud del viaje). Para ello se debe adjuntar en el correo un documento PDF que al menos debe contener los siguientes datos:
 - Nombre del chofer asignado y su teléfono
 - Punto de inicio del viaje, fecha y hora.
 - Lista de los viajeros y sus números de teléfono.

Requerimiento 10: Registrar un nuevo usuario de rol “Secretaria”

El sistema debe permitir que se ingresen nuevos usuarios con el rol de secretaria. Todos los datos son requeridos. El nombre del usuario corresponde al nombre utilizado en el correo electrónico.

Aspectos a considerar:

- La contraseña será generado de forma automática por el sistema siguiente las siguientes políticas:
 - Tamaño de la contraseña, 8 caracteres como mínimo y 12 como máximo.
 - Debe incluir al menos un símbolo especial (! # \$ % & ' () * + , - . / : ;

- Debe incluir números, letras y un símbolo especial
- Las letras deben alternar entre mayúscula y minúscula.
- Se debe notificar al usuario en su cuenta de correo, los datos a enviar corresponden a su nombre de usuario y contraseña.
- La aplicación debe tener pre-cargados al menos 3 usuarios desde el archivo usuariosDB.XML o JSON y los datos deben estar cifrados .
 - El algoritmo de cifrado es decisión del equipo pero debe ser documentado.

Requerimiento 11: Consultar datos

El sistema debe permitir que el usuario realice una serie de consultas sobre los datos almacenados, donde la información de estas consultas se desplegará mediante diferentes formatos de gráficos.

Top 5 de los choferes asignados a viajes → Formato gráfico de barras

Top 5 de los departamentos o escuelas según cantidad de viajes → Formato Gráfico de pastel (circular)

Debe considerar esta funcionalidad como parte del menú de reportes de la aplicación.

RECOMENDACIONES Y CONSIDERACIONES IMPORTANTES

- La aplicación debe tener datos precargados según fue solicitado en algunos requerimientos. Si no es así, parte del tiempo de la revisión se utilizará para este fin. Esto podría provocar que el tiempo no sea suficiente para concluir con la revisión funcional de todos los requerimientos.
- La cuenta del usuario administrador es por omisión administrador y la clave 123Admin!
- La solución de software debe proveer persistencia de datos mediante archivos en formato XML, la estructura interna de cada archivo es decisión del equipo.

- Como recomendación, puede utilizar un menú para organizar las distintas funcionales de la aplicación según el tipo de usuario, es también claro que debe existir un mecanismo de validación de credenciales.

III. Puntos a ser evaluados

1. Correctitud y completitud de la solución computacional. La tabla de aspectos a evaluar será publicada 48 horas antes de la fecha de entrega del proyecto.
2. Entregar un documento formal con los siguientes apartados:
 - a. Portada
Debe incluir:
 1. Nombre del curso
 2. Número de semestre y año lectivo
 3. Nombre de los estudiantes
 4. Números de carnet
 5. Número de tarea programada
 6. Estatus de la entrega :
 - a. [Deplorable|Regular|Buena|Muy Buena|Excelente|Superior]
 - b. Introducción, índice.
 - c. Estrategia de Solución:
 - i. Propuesta de metodología de trabajo por parte de los miembros del equipo para abordaje del problema, establecimiento de tareas o cualquier otra actividad que aporte a la solución del problema.
 - ii. Cronograma de trabajo
 - iii. Diagrama de clases de bajo nivel definitivo propuesto en notación UML, el cual debe ser aportado en formato PDF. Para cada clase debe incluir todos los métodos y atributos que se requieran para dar solución al problema propuesto así como indicar las relaciones que existen entre los objetos. Omita los métodos accesorios en este diagrama a fin de incrementar la legibilidad del diagrama de forma general.
Agregue todas las clases de la lógica de negocios en el paquete correspondiente.

- iv. Justificación de las relaciones establecidas entre los objetos del diagrama, es decir, porqué se decidió por una relación de composición, agregación, dependencia o asociación.
- d. Análisis de Resultados
Deberá elaborar un listado de todas y cada una de las actividades y tareas que deben cubrirse a nivel funcional, para cada una de ellas debe aportar el porcentaje de realización y en caso de no ser el 100% debe justificarse el por qué no se completo el requerimiento.
- e. Enlace al JavaDoc generado (documentación interna de la aplicación con los métodos más relevantes incorporados en los distintos objetos). No es necesario documentar los getters-setters.
- f. Aspectos relevantes y lecciones aprendidas: Debe prepararse un listado de las lecciones aprendidas producto del desarrollo de la tarea programada (al menos 3 por cada estudiante). Las lecciones aprendidas pueden ser de carácter personal y/o técnico que involucre aspectos que han logrado un aprendizaje en temas de investigación, desarrollo de habilidades técnicas y habilidades blandas como trabajo en equipo, comunicación, forma de expresar ideas, etc.
- g. Bitácora de trabajo durante el plazo del proyecto, incluyendo verificaciones realizadas (si existieran) de consultas realizadas con el profesor o asistente. (Minutas)
- h. Bibliografía, fuentes digitales y librerías utilizadas.

IV. Condiciones administrativas

Esta tarea programada se rige por las siguientes condiciones:

1. La tarea debe desarrollarse únicamente mediante una solución netamente Orientada a Objetos, de lo contrario, se asignará nota 0 a todos los integrantes del equipo de trabajo.
2. El desarrollo de la tarea es estrictamente en tríos -a menos que numéricamente no sea posible-.
3. La tarea DEBE implementarse con interfaz gráfica (Swing, Processing, Fx, según la decisión del equipo).
4. Debe cumplir con todo lo indicado en la sección "Puntos a ser evaluados".
5. Deberá entregarse en tiempo y forma según el plazo establecido por el profesor al momento de la lectura de este documento.
6. El lenguaje de programación a utilizar es Java, y el entorno de desarrollo también es decisión del equipo de trabajo.
7. La entrega de la propuesta de solución se realizará mediante la plataforma tecDigital y el archivo comprimido debe contener únicamente 2 carpetas: documentación y solución computacional, en la primera deberá incluir la documentación solicitada y en la segunda los archivos y/o carpetas necesarias para la ejecutar este proyecto.
8. El estándar de codificación que debe aplicar es el recomendado por *Google Java Code Style*.
9. Se cuenta con 3 semanas a partir del día de entrega de la tarea.
 - Entrega de forma digital en el TEC Digital, en la sección de Evaluaciones, Tarea Programa #1.
 - El proceso de revisión será dado a conocer oportunamente.

Nota: El incumplimiento de alguna condición implica una calificación de cero.

V. Referencias a consultar

Envió de correo

https://www.example-code.com/java/smtp_gmailSsl.asp

<https://docs.oracle.com/javase/7/api/javax/mail/package-summary.html>

Parser de archivos XML

<https://docs.oracle.com/javase/8/docs/api/javax/xml/parsers/package-summary.html>

Gráficos (circular, barras, entre otros)

<http://www.jfree.org/jfreechart/>

User interface Design Fundamentals

<http://blog.teamtreehouse.com/10-user-interface-design-fundamentals>

Material de apoyo ubicado en la carpeta compartida del curso.

VI. Anexos

Anexo 1: Manual de Usuario

1. Explicación paso a paso de cómo probar cada uno de las funcionalidades mediante pruebas de funcionalidad (pantallas).

Anexo 2: Plantilla de cronograma de trabajo propuesto

Se adjunta en documento adjunto.

Anexo 3: Minutas de seguimiento

Minuta 2018-S2-XX

Fecha:		Hora de inicio:	Hora de finalización:
Lugar de reunión:			
Asistentes a la reunión: - -			
Ausentes en la reunión: -		Justificaciones: -	
Tema	Fecha límite de cumplimiento	Acuerdos/Pendientes	Responsable asignado