



APELLIDO Y NOMBRE DEL ALUMNO:

PROFESOR CON QUIEN CURSÓ:

FINAL DE DISEÑO DE SISTEMAS – 16 DE MAYO DE 2016

El Ministerio de Transporte de Ciudad República ha encarado un proceso de modernización hace varios años. En una primera etapa del proceso se llevó a cabo el registro de las empresas de transporte público de pasajeros, indicando los medios que opera, los recorridos que cada medio tiene habilitados (en esa primera etapa se registraron las cabeceras y algunos puntos de interés cercanos al recorrido) y los horarios asignados entre cabeceras a lo largo del día. Esta primera etapa ha conformado un software denominado RTR (Registro de Transporte de Ciudad República).

Actualmente se encuentra iniciando la segunda etapa con importantes mejoras. Así como la primera etapa tenía objetivos de índole administrativo, esta segunda etapa apunta a brindar información al usuario. El objetivo principal de esta etapa es generar una nueva App Mobile denominada *República Transporte* y cuya funcionalidad estrella es proporcionar un estimado de los tiempos de arribo de las unidades de transporte a las paradas/estaciones. Para esto se deben incorporar al sistema puntos intermedios en el recorrido que permitirá constituirlos como nodos para analizar tiempos de recorridos entre nodos, definir el tiempo estándar de recorrido entre nodos y definir factores de retraso: semáforos, tránsito en la calle, cruce de vías, retraso en las paradas/estaciones por exceso de pasajeros.

El Ministerio entiende que los factores de retraso pueden ser configurados de manera compleja y en el futuro se espera reemplazar el cálculo del índice de retraso por un nuevo componente. Por el momento la información de los factores de retraso es definida por cada empresa de transporte o por cada línea que opera la empresa (esta decisión queda a criterio de cada empresa) e implica definir para cada enlace entre nodos un porcentaje de retraso generado por cada uno de los factores mencionados en el párrafo anterior. Estos factores constituyen, conjuntamente, el IR (Índice de Retraso). Aún no está claro (y probablemente no lo esté hasta el final del proyecto) cómo se conjugan estos factores para generar el IR, por lo cual deberá diseñar la solución teniendo en cuenta esta situación y, además, teniendo en cuenta que, según ha informado el Ministerio de Transporte, dicho algoritmo podría ser modificado en reiteradas oportunidades (en virtud de “ajustarlo” según la experiencia de los resultados).

Todo usuario de la aplicación podrá solicitar la notificación de arribo del transporte deseado con una anticipación de minutos preconfigurada por el propio usuario. Esta notificación se realizará mediante un envío de correo electrónico a la dirección que el usuario hubiera solicitado y mediante un mensaje al móvil.

Nos han solicitado formar parte del equipo de Diseño que lleva a cabo el Backend (o sea, no la App Mobile sino el Backend sobre el cual se sostiene la App Mobile).

Entonces el alcance de la etapa actual es:

- Brindar al usuario la información sobre el horario de llegada a determinado punto de un medio de transporte
- Realizar las notificaciones pertinentes
- Altas, Bajas, Modificaciones de la información requerida para los puntos anteriores
- Configuración de factores de retraso e índice de retraso



APELLIDO Y NOMBRE DEL ALUMNO:

PROFESOR CON QUIEN CURSÓ:

Además se han definido las siguientes restricciones y requisitos no funcionales:

- El desarrollo vinculado a esta segunda etapa debe utilizar el software RTR, pero operará como un componente externo que se vincula con el mismo
- Se persistirá sobre un modelo de datos relacional sobre un motor Oracle
- Se desarrollará en Java¹
- Se requiere llevar registro del historial de acciones sobre la aplicación por parte de los usuarios, indicando qué acción realizó el usuario, cuál fue el usuario y en qué fecha y hora fue realizada la acción
- La tasa de fallas máxima permitida es de 0,001%²

1. MODELADO DE CLASES

1.1. ¿Cómo diseñaría la solución respecto al envío del email? ¿Qué ocurriría si la tecnología de envío de email cambiara?³. Justifique considerando los principios del diseño.

1.2. El Ministerio de Transporte se encuentra discutiendo con el Ministerio de Tecnología el siguiente aspecto: ¿Cree que sería factible desligar a la aplicación del envío del email y darle esta responsabilidad a un Centro de

¹ Tener en cuenta que en Java existe una clase Date, cuyo uso se requiere.

La clase Date se utiliza para representar una fecha y una hora. Se pueden manipular el día, mes, año, día de la semana, horas minutos y segundos.

La clase Date maneja fechas y horas UTC (Universal Time Coordinated), aunque su precisión depende del host en que se esté ejecutando la Máquina Virtual Java. Hay veces en que se indica el tiempo en términos GMT (Greenwich Mean Time), que es equivalente al Tiempo Universal, UT (Universal Time). GMT es el nombre civil para el estándar y UT es el nombre científico para ese mismo estándar. Y, la distinción entre UTC y UT es que UTC se basa en un reloj atómico y UT está basado en observaciones astronómicas, pero la diferencia existente entre los dos es bien poca.

Esta clase posee varios constructores posibles: Date(); Date(año,mes,día); Date(año,mes,día,horas,minutos); Date(año,mes,día,horas,minutos,segundos); Date(String); Date(long)

Y sus métodos públicos son:

getTime(): Devuelve el número de milisegundos transcurrido desde el 1 de Enero de 1970, a las 00:00:00 GMT
setTime(long): Fija la fecha a partir del número de milisegundos transcurrido desde el 1 de Enero de 1970, a las 00:00:00 GMT
before(Date): Comprueba si una fecha es anterior a la especificada
after(Date): Comprueba si una fecha es posterior a la especificada
equals(Object): Compara dos fechas. El resultado es true, si y sólo si el argumento no es nulo y los objetos coinciden a nivel de milisegundos
toString(): Crea la representación canónica de la fecha, de la forma "Fri Oct 3 21:23:45 GMT 1997"
getDate(): Devuelve la fecha actual

² Tener en cuenta que en Ciudad República las conexiones de Internet suelen tener inconvenientes

³ Puede pensarlo en cualquier tecnología, pero si quiere pensar en un ejemplo concreto supongo el uso de JavaMail. JavaMail proporciona un marco independiente de la plataforma y del protocolo para construir aplicaciones de correo y mensajería. Es una API que proporciona clases para el tratamiento de un sistema de correo. ¿Qué ocurriría si decidiéramos utilizar JavaMail y posteriormente cambiarla por otra API?



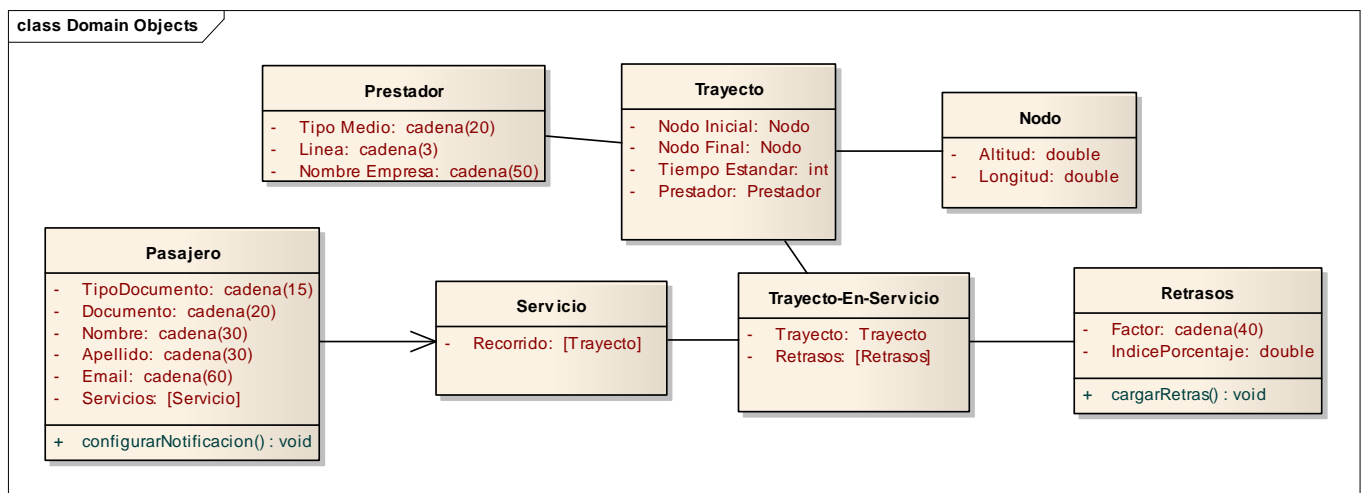
APELLIDO Y NOMBRE DEL ALUMNO:

PROFESOR CON QUIEN CURSÓ:

Notificación (que sería desarrollado por el propio Ministerio de Tecnología) que reciba de diferentes aplicaciones mensajes y tenga como responsabilidad única enviar emails? ¿Qué ventajas y desventajas percibe? Justifique considerando los principios del diseño.

1.3. ¿Qué decisión en el diseño implica que “en el futuro se espera reemplazar el cálculo del índice de retraso por un nuevo componente” tal como se afirma en el texto?

1.4. Diseñe el modelo de clases de diseño considerando el siguiente modelo preliminar obtenido en la etapa de análisis y teniendo en cuenta el texto presentado y las observaciones de los puntos anteriores (1.1; 1.2; 1.3). Incluya atributos y métodos.



1.5. ¿Qué cambios implicaría en el diseño si usted supiera que la información acerca del retraso asociado a los semáforos será entregada a partir de archivos en diversos formatos por el Centro de Monitoreo de Semáforos de Ciudad República que pueden ser importados desde República Transporte? Suponga que la información puede ser generada en varios formatos a la vez (por ejemplo, porque son generados desde diferentes aplicaciones del Centro)

2. MODELADO DE DATOS

Para el modelo de clases realizado en 1.4 realice el modelo de datos. Justifique bajo las premisas de la técnica de mapeo objetos-relacional.

3. ARQUITECTURA

Explique los aspectos del diseño arquitectónico de su software en vinculación con el software RTR. ¿Cómo resolvería la arquitectura considerando la tasa de fallas solicitada y las características de las conexiones en Ciudad República.