



INSTITUTO TECNOLÓGICO DE COSTA RICA

ÁREA ACADÉMICA DE INGENIRÍA EN COMPUTACIÓN

Tarea programada II

PROYECTO BUSCR

ELABORADO POR:

Kevin Méndez Arce

Fernando Ugarte Ugalde

Carlos Valerio Méndez

PROFESOR:

Carlos Álvarez González

II SEMESTRE 2016

Contenido

Introducción	2
Descripción del problema.....	3
Solución del problema.....	4
Diagrama E/R	5
Diagrama del modelo relacional	6
Conclusión	7
Conclusión final	9
Bibliografía	11

Introducción

El transporte público es una de las mejores soluciones para combatir contra la contaminación ambiental, el tráfico de automóviles disminuye y además de ser ideal para la movilidad de personas en zonas de alta concentración de ciudadanos. Sin embargo este sistema tiene varias fallas, como por ejemplo la repetición intencional de rutas de buses, la falta de terminales en muchas de las empresas autobuseras, inconveniencias a la hora de pagar el viaje y la incapacidad de revisar las rutas en tiempo real. Dadas estas desventajas el Gobierno de Costa Rica decidió realizar una aplicación para contrarrestar al problema dado. En este documento se citan y detallan los diseños de entidad relación, entidad relación extendido y el diagrama de clases de la aplicación que se creó para la solución del problema mencionado, además se detallan los datos utilizados para este proyecto.

Descripción del problema

El transporte público, al ser una opción tan popular en cuestión de movilidad civil, lleva a muchos problemas referentes a la enorme cantidad de medios en la calle. El transporte público en la gran área metropolitana se constituye de autobuses y estos son los que presentan la principal cantidad de problemas presentados por los transportes públicos. En primer lugar, existe una gran cantidad de rutas diferentes que los usuarios pueden tomar, pero que sin embargo atraviesan muchos puntos similares, lo que generalmente causa congestionamientos por la gran cantidad de buses en una misma calle, haciendo paradas en diferentes puntos. Por otro lado, no existe una sectorización inteligente del transporte, es decir, todas las rutas de autobuses llegan a un mismo punto: San José. Son pocas las empresas autobuseras que cuentan con una terminal adecuada, por lo que acaban parqueando sus vehículos en la vía pública, causando congestionamientos. La calidad del servicio a los usuarios es mala, siempre tienen que cargar dinero en efectivo para pagar. Además, no cuentan con un punto central donde consultar las rutas, ni ver en tiempo real cuánto hace falta para que el pase por la estación más cercana.

Solución del problema

El gobierno costarricense ha identificado la necesidad de contar con una plataforma tecnológica que por medio de ciertas características contrarresten estos problemas. A continuación se mencionan los 3 problemas principales y su solución respectiva.

Rutas repetidas. La aplicación CRBus tendrá el módulo de compañía, donde las empresas ingresan los datos de las rutas donde ofrecen el servicio y con los autobuses con los que cuenta. Esto cederá más información a las empresas nuevas o ya existentes de buses y evitará que creen rutas repetidas a las que ya hay.

Incomodidad al portar efectivo. La aplicación CRBus tendrá el módulo de usuario donde estos podrán consultar su saldo y toda la información de su cuenta. Con la ayuda de este módulo es posible cargar puntos a la cuenta de cada usuario y pagar el viaje al boardar un autobús.

Falta de proveedor de información a tiempo real. La aplicación CRBus tendrá el módulo de autobús. Este módulo proveerá información de la geolocalización y el tiempo de espera de uno o más buses seleccionados.

Además la aplicación CRBus tendrá el módulo de reportes, este módulo permitirá diversos tipos de consultas que brindan información importante sobre las empresas, usuarios y rutas. En este módulo se puede verificar las rutas más frecuentes por los buses, las rutas replicadas que existen, la cantidad de usuarios que viajan por ruta y empresa y finalmente el histórico de las tarifas de los buses.

Diagrama E/R

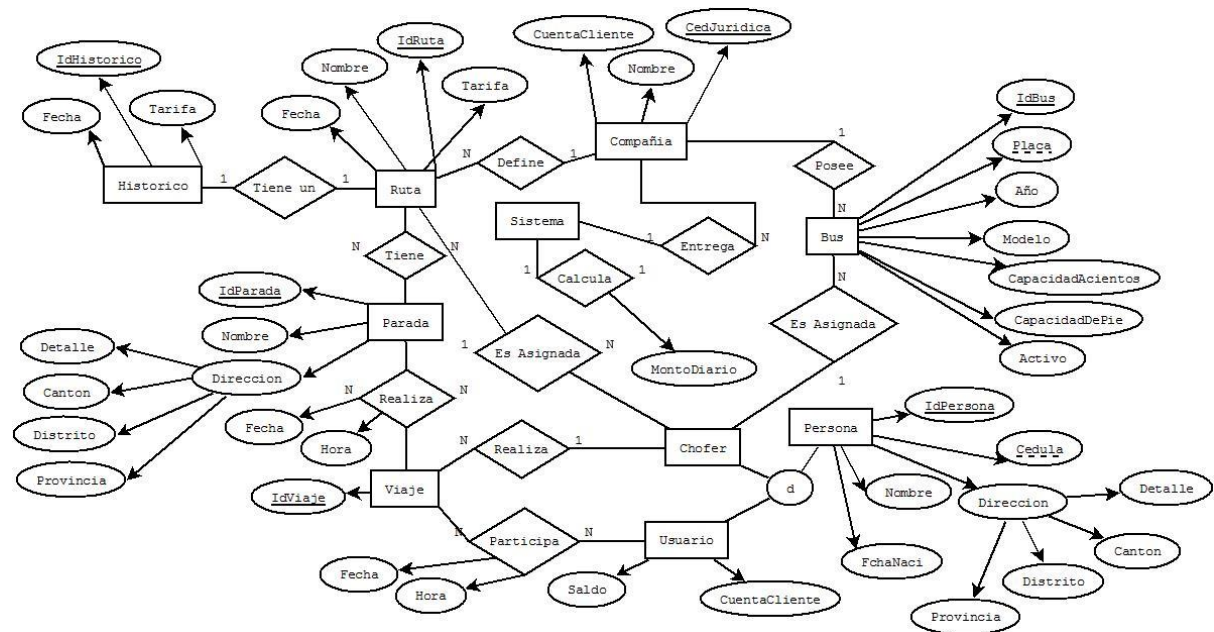
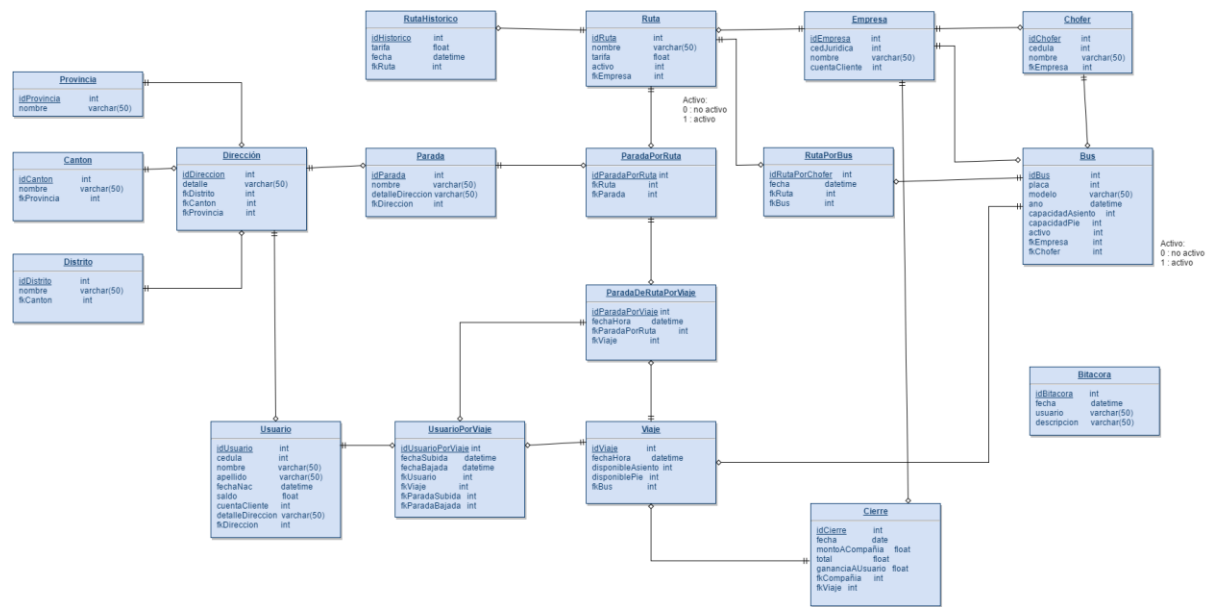


Diagrama del modelo relacional



Conclusión

El control de tiempo es esencial para la completitud de las tareas programadas, se considera que en este caso se empezó temprano a trabajar con el diseño, sin embargo nuestro trabajo no avanzó después de completar esta primera parte, esto por la presión y atención que requirieron los otros cursos. Aun con este contratiempo fue posible alcanzar el ritmo que se requirió para completar el proyecto.

A continuación se citan detalladas conclusiones de las partes primordiales de la tarea programada.

1. Documentación

La documentación se encuentra completa, en esta se detalla muy bien el problema, las soluciones y los diagramas que se realizaron para la creación de la aplicación, el diccionario de datos que se considera una parte esencial para la comprensión de la base de datos y se decidió manejarla por un archivo por aparte adjunto a este, esto por la facilidad de creación y manejo. En esta se detalla todos los atributos/campos utilizados para las conexiones entre tablas y así hacer posible la comunicación de los datos. En esta tabla se mencionan y explican las siguientes características: ID del campo, Nombre del campo, Tabla, Tamaño del campo, Tipo del campo, Formato del campo, Descripción, Ejemplo. Notemos que cuando se menciona cambio, refiere al campo de la tabla donde se encuentran los datos.

1.1 Modelos de entidad/relación, relacional y diagrama de clases

También parte de esta documentación se encuentran incluidos los modelos (entidad/relación y relacional) junto con el diagrama de clases, estos elementos se consideran los más difíciles en cuanto al desarrollo de aplicaciones se refiere, ya que, del diseño depende el 100% del proceso para completar la aplicación, afortunadamente le prestamos mucha atención al diseño y hacer casos de prueba en papel, esto para asegurar de que el diseño funcione en conjunto con las consultas que se solicitaban, esta parte es importante ya que la eficiencia es parte del alcance del proyecto.

2. Creación de las tablas de la base de datos

La creación de las tablas en la base de datos fue sencillo en términos de tiempo y esfuerzo, gracias a los datos proporcionados por las clases, el diseño bien implementado y la comunicación entre los miembros del equipo.

3. Datos básicos o datos de prueba

Los datos de prueba fueron un reto para nosotros, ya que se solicitaban cerca de 2, 000,000 datos de prueba distribuidos en 17 tablas, los datos más desafiantes de crear fueron los viajes, esto porque se solicitaban 1, 000, 000 de datos, muchos de los programas que se utilizaron en la primera tarea programada no permitía la creación de más de 1, 000 datos de prueba, por lo que muchos se tuvieron que hacer por medio del merge de Windows.

Afortunadamente los datos de Costa Rica, así como las provincias, cantones y distritos ya se tenían de la primera tarea programada. El proceso fue largo y tedioso, pero logramos conseguir métodos de obtención por medio de expresiones regulares.

4. Inserción de datos de prueba

Esta sección fue una de las más que se invirtió tiempo, porque era en la parte en la que se utilizaban todos los procedimientos almacenados hechos para la inserción de los datos y la comunicación de las tablas para la obtención de más datos. Los datos se manejaron por medio de archivos separados por coma y se ingresaron por un método llamado 'bulk' o inserción en masa en español, que consiste en insertar todos los datos "al mismo tiempo". Este método es muy efectivo para inserciones en volumen y lo utilizamos para todas las inserciones. Sin embargo en la inserción de los viajes el procedimiento duró 20 horas en completarse. Se piensa que se pudo mejorar el proceso para más eficiencia.

5. Código en Procedimientos Almacenados

Esta parte fue una de las más desafiantes de toda la tarea programada, se tuvo que poner en práctica lo aprendido en clase, y de experiencias anteriores, aunque las presentaciones fueron de gran ayuda, fue un reto poder crear los procedimientos para que trabajaran juntos con otros y hacer que la base de datos trabajara como una sola e incluso se encontraban errores difíciles de solucionar durante la última fase, la creación de la aplicación.

6. Aplicación

La última fase de la tarea programada fue un reto para todo el equipo ya que se tuvo que trabajar en los procedimientos almacenados ya creados y en los que muchos se produjeron errores. Además aprender lenguajes de gran curva de aprendizaje que permitían el manejo de la interfaz y la invocación de los procedimientos almacenados SQL.

Conclusión final

La experiencia final fue difícil, sin embargo necesaria, a pesar de que pasamos muchas noches sin dormir, se aprendió mucho acerca de diferentes partes de las bases de datos, su diseño, su funcionamiento interno, los datos de prueba, la creación de aplicaciones que la utilicen y seguridad sobre la base de datos.

Lecciones aprendidas Durante todo el proyecto programado, se creció y avanzó mucho como persona y como profesional, es importante tener en cuenta que el objetivo se cumplió a continuación se citarán varios puntos de lecciones aprendidas durante la creación de la tarea programada.

- **Tiempo.** El tiempo es lo que más afecta a un proyecto, sin embargo la se tuvo mucho tiempo para la creación de la tarea programada, teniendo esto en cuenta, empezamos a trabajar muy tarde la segunda parte de la base de datos. Es muy importante tener en cuenta el tiempo que se tiene para la finalización de la tarea.
- **Modelo entidad/relación.** El modelo entidad relación es la base para crear un buen diseño para la base de datos por esta razón se tuvo mucho cuidado al realizarlo, esto se premió a la hora de la creación de la base de datos, ya que facilitó entender el problema, además facilitó la creación del modelo relacional.
- **Modelo Relacional.** Similar al caso anterior el modelo relacional es esencial para la creación no solo de la base de datos, sino también para la creación de los datos de prueba. A la hora de convertir el modelo de entidad/relación al modelo relacional, afortunadamente los errores fueron arreglados antes de la creación de la base de datos, gracias a las continuas revisiones hechas.
- **Variables significativas.** Las variables significativas fueron una de las cosas más importantes implementadas en la tarea programada, algunas variables no poseían nombres significativos y obligaban al programador a realizar trabajo innecesario (revisar el diseño, revisar otro procedimiento almacenado, etc). Sin embargo cuando las variables poseían nombres significativos, era mucho más sencillo programar los procedimientos almacenados para la conexión entre tablas y la conexión de datos.
- **Creación de tablas.** A la hora de la creación de tablas es necesario crearlas en un orden específico, si no se hace así, las creaciones de llaves foráneas creará un error, ya que no pueden referenciar a tablas que no están creadas. Es importante

tener en cuenta esto ya que se podrá ahorrar mucho tiempo valioso en la creación de la base de datos.

- Creación de datos de prueba. En esta tarea programada se tuvo que crear alrededor de 1, 000,000 de datos únicamente para una tabla. Esto generó muchos problemas con los programas de modificación de texto ya que los archivos son muy grandes, hay que tener cuidado con los datos que se tienen que agregar y poseer un archivo de excepciones con el cuál guiarse para evitar cometer estos errores.
- Inserción de datos de prueba. A la hora de la inserción de los datos de prueba se invirtió mucho tiempo, ya que el archivo de viajes era muy grande. Aunque se piensa que hay mejores métodos para la inserción de los datos, se manejó muy bien el trabajo. Es importante tener en cuenta los diferentes tipos de métodos que hay y las ventajas y desventajas de cada uno de estos.
- Código en procedimientos almacenados. Se debe tener en cuenta, a la hora de la creación de procedimientos almacenados, el orden y el tipo de los parámetros, ya que los triggers no funcionaron a la hora de insertar ciertos datos, afortunadamente no fue un error difícil de solucionar.
- Aplicación. Es importante conocer la dificultad de los lenguajes de programación antes de elegirlos para la creación de la aplicación, esto ahorra mucho tiempo y facilita el trabajo para optimizarlo.
- Comunicación entre miembros del equipo. Por último es importante tener buena comunicación con el equipo de trabajo, mala comunicación con el equipo de trabajo puede generar trabajo duplicado o trabajo innecesario.

Bibliografía

CSS - Tutorial (s. f.). Recuperado de <http://getbootstrap.com/css/#tables>

Jeansalle, A. (s. f.). Curso PHP Recuperado de <https://www.youtube.com/watch?v=rWxCegISWs&list=PLnPcl8Zt45gVLewfwaAB4HKDIKSE9MYbw&index=4>

Martos navarro, F. (2006, 16 de Febrero). Editorial Mad, S.L. (primera ed.) Manuscrito no publicado.

Microsoft (s. f.). BULK INSERT Recuperado de <https://msdn.microsoft.com/en-us/library/ms188365.aspx>

Microsoft (s. f.). CREATE TRIGGER Recuperado de <https://msdn.microsoft.com/en-us/library/ms189799.aspx>

Natividad, R. (s. f.). 18. Curso de Desarrollo Web con HTML5 y CSS3 - Listas [ESTRUCTURA Y SEMÁNTICA] - YouTube Recuperado de https://www.youtube.com/watch?v=YsryNA4ut78&list=PLLeLFRUi2pkjBj4rhoWCt2_-TnDN8BNYr1&index=18

PHP - Línea de Código (s. f.). Recuperado de <http://lineadecodigo.com/categoria/php/>

PHP - Tutorial (s. f.). Recuperado de https://www.youtube.com/playlist?list=PLHvcSLtKxkO567BDSwRmjV_nClw0JR6TT

PHP: Listado de Funciones y Métodos - Manual (s. f.). Recuperado de <https://secure.php.net/manual/es/indexes.functions.php>

portalucam (s. f.). Recuperado de <https://www.youtube.com/user/portalucam>

Web, M. (s. f.). PHP - Registro y Sesiones de Usuario Recuperado de <https://www.youtube.com/playlist?list=PLHvcSLtKxkO6pR7ouxmGdY9ooSn7n9PL>