CAPÍTULO 2

2. CONSTRUCCIÓN DEL MODELO DE SIMULACIÓN DE LA RUTA TRONCAL 7 "ORQUÍDEAS-CENTRO URBANO"

2.1. INTRODUCCIÓN

En el presente capítulo se explicará los aspectos que deben ser tomados en cuenta para realizar la construcción del simulador que modelará el comportamiento de la ruta Troncal 7 "Orquídeas-Centro Urbano". Las etapas para la construcción del simulador, se describen en la sección 2.2, los softwares utilizados se describe en la sección 2.3, la interacción de los softwares utilizados en la construcción del simulador están detallados en la sección 2.4, en la sección 2.5 se describe al software de simulación utilizado-GPSS

World, en la sección 2.6 se describe la codificación de los procesos que simulan las actividades del Sistema de Transportación. Finalmente en la sección 2.7 se describe la base de datos que almacenará la información obtenida del proceso de simulación.

2.2. ETAPAS PARA LA CONSTRUCCIÓN DEL SIMULADOR TRONCAL 7 "ORQUÍDEAS-CENTRO URBANO"

En la primera etapa se debe definir los objetivos de estudio, el cual consiste en realizar un modelo de simulación para el comportamiento del la Ruta Troncal 7 "Orquídeas-Centro Urbano" el cual no solo funcionará en base a las políticas establecidas por el Municipio de Guayaquil, sino también se implementará nuevas políticas de operación. El objetivo de construir el simulador en base a varias políticas de operación servirá para analizar los cambios en el comportamiento de ruta Troncal 7.

La segunda fase constituye en seleccionar el software de simulación en el que se codificará los procedimientos que describen el comportamiento de la ruta Troncal 7. Por las características que posee GPSS World 4.3.5. ha sido el software seleccionado para la construcción del modelo.

El simulador tiene como usuario principal los miembros del Departamento Municipal de Transporte de la M.I. Municipalidad de Guayaquil, los mismos que desconocen el manejo de GPSS World, debido a esto, fue necesario crear un software que posea una interfaz amigable y de fácil manejo.

Es de gran importancia el manejo de los resultados obtenidos en la simulación del sistema de transporte para varias políticas, es por ello que nace la necesidad de almacenar dichos resultados en una base de datos, que tenga la capacidad de controlar grandes cantidades de registros.

2.3. SOFTWARES UTILIZADOS EN EL DESARROLLO DEL SIMULADOR TRONCAL 7 "ORQUIDEAS-CENTRO URBANO"

Con la intención de diseñar un software que cumpla con los requerimientos anteriormente mencionados, se decidió implementar una aplicación que integre varios paquetes de software que cumplan con requisitos específicos, para que una vez integrados se obtenga un Sistema de Simulación de Transporte acorde a las necesidades de los usuarios del sistema. Los paquetes a utilizar y las funciones que éstos cumplen se detalla a continuación:

- **Microsoft Visual Basic 6.0**

Es un software diseñado para facilitar el desarrollo de aplicaciones en un entorno grafico como Windows 98, Windows NT o superior. La principal característica de Visual Basic 6.0 es que permite diseñar entornos amigables y de fácil utilización para todo tipo de aplicaciones.

Con un software amigable y de fácil utilización, los usuarios no familiarizados con el lenguaje de programación de GPSS, no se verán imposibilitados de la utilización del simulador Troncal 7 “Orquídeas-Centro Urbano”.

Los datos requeridos para diseñar la ruta Troncal 7 serán ingresados desde el ambiente gráfico desarrollado bajo Visual Basic. Esta aplicación será la única con la que el usuario va a interactuar, ya que, las demás aplicaciones realizarán sus funciones de forma interna e imperceptible para el usuario.

- **Microsoft SQL Server 2000**

  El lenguaje de gestión de bases de datos más conocido en la actualidad es SQL, el mismo que se define como un lenguaje de consulta y programación de bases de datos. Se lo utiliza para acceder a los datos con el objetivo de consultar, actualizar y gestionar sistemas de bases de datos relacionales.

Este paquete será el motor de base de datos que el sistema utilizará, para el manejo y almacenamiento de los registros relacionados con cada simulación que se realice dentro del sistema.

Se escogió este software debido a que SQL Server maneja eficientemente grandes cantidades de información y ofrece, además, la seguridad y protección que los datos requieren, siendo esto uno de los mayores requerimientos para la selección del software encargado de almacenar información reservada e importante.

- **Minuteman GPSS World 4.3.5**

  Este Software será el encargado de realizar la simulación del modelo que describe el funcionamiento de la Troncal 7. Procesará la información de entrada que describa la ruta de transporte de la Troncal 7 con las políticas de operación que el usuario quiera analizar, luego de realizar la simulación proporcionará los resultados obtenidos.

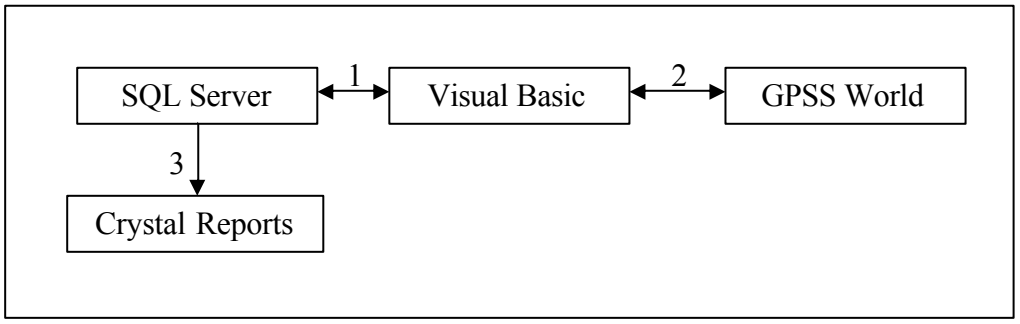
- **Seagate Crystal Reports 8.5**

Crystal Reports es un software especializado en la presentación de reportes impresos, se eligió este software para el manejo de la presentación de resultados por su flexibilidad y facilidad de uso; a pesar de que Visual Basic incluye un módulo para la presentación de reportes, el mismo no posee la capacidad de manejo de información como la que brinda Crystal Reports.

2.4. INTERACCIÓN DE LOS PAQUETES DE SOFTWARE UTILIZADOS EN LA CREACIÓN DEL SIMULADOR

Para que el simulador realice todas las actividades requeridas, es necesario que los cuatro paquetes de software anteriormente mencionados, se encuentren totalmente interrelacionados. La relación que mantiene cada programa se resume en el siguiente esquema:

Grafico 2.1.
Interacción de los Softwares



Las 3 relaciones que existen entre los paquetes de softwares se detalla a continuación:

Relación 1: Visual Basic ⟷ SQL Server

El usuario ingresa todos los parámetros que definirán a la ruta de transporte, es decir, el diseño de la ruta y sus políticas de operación en la aplicación de Visual Basic, luego Visual Basic se encarga de enviar los datos para que sean almacenados en la base de SQL Server. Esta conexión es directa, ya que Visual Basic está diseñado para establecer un enlace directo con motores de Base de Datos como SQL Server mediante objetos especiales conocidos como objetos ADO (ActiveX Data Objects).

Asimismo, SQL Server envía los datos que la aplicación de Visual Basic requiera, de esta forma, el enlace entre ambas aplicaciones es bidireccional y no requiere de ningún paso intermedio para su establecimiento, ya que ambas aplicaciones han sido diseñadas para ser interrelacionadas.

Relación 2: Visual Basic ⟷ GPSS World

Una vez que todos los datos se encuentren ingresados, Visual Basic tiene la tarea de transformar la información recopilada en un lenguaje que GPSS World pueda comprender. No existe una conexión directa que se pueda establecer entre estos dos paquetes,

por lo tanto, se vio la necesidad de hallar un medio al cual ambos programas puedan acceder, dicho medio son los archivos de texto o datastreams (Archivos TXT), estos archivos pueden ser manipulados por ambas aplicaciones y van a servir como medio de transferencia para la comunicación.

Visual Basic transforma a lenguaje de GPSS toda la información recopilada, y adicionalmente escribe este código en archivos de texto, construyendo la estructura total del simulador que GPSS va a ejecutar. Una vez que esta transformación ha concluido Visual Basic envía la orden a GPSS para que la ejecución del modelo comience.

Como no existe una conexión directa entre ambas aplicaciones, no se puede visualizar los resultados obtenidos por GPSS, por lo tanto nuevamente se va a hacer uso de los datastreams para almacenar dichos resultados para que luego Visual Basic haga uso de ellos.

Una vez que la ejecución del modelo ha concluido, Visual Basic lee los archivos de texto de resultados creados por GPSS y los almacena en la base de datos haciendo uso nuevamente de la relación número 1.

La relación también es bidireccional, ya que a pesar de que existe un paso intermedio para la realización de la conexión entre ambas aplicaciones, este paso es realizado en ambas direcciones, cada vez que se realice la simulación del sistema.

Relación 3: SQL Server ⟶ Crystal Reports

Luego que el modelo de simulación ha sido ejecutado y los resultados han sido almacenados en la base de datos SQL Server el usuario tiene la opción de visualizar la información en reportes impresos, para ello, hace uso del software Crystal Reports, el cuál lee la información almacenada en SQL Server. La conexión entre estos dos paquetes es de forma directa, ya que Crystal Reports ya cuenta con los objetos de conexión necesarios para la interacción entre estos dos paquetes de software.

La relación entre estos dos paquetes es unidireccional ya que Crystal Reports presenta la información almacenada en la base de datos, pero no envía ninguna clase de información a la misma.

2.5. SOFTWARE DE SIMULACION - GPSS WORLD

GPSS/PC, General Purpose System Simulation (sistema de simulación de propósitos generales), fue desarrollado por Minuteman Software y es una versión para PC.

GPSS es un lenguaje de diagrama de bloques. Se define una cantidad de tipos básicos de bloques, cada uno de los cuales representa una acción determinada. El usuario debe de describir las actividades del sistema en términos de estos tipos básicos de bloques. Como cuestión de conveniencia en la programación, la mayoría de los lenguajes de simulación establecen una diferencia entre las entidades temporales y permanentes. Las temporales se crean durante la corrida de simulación, avanzan a través del sistema y finalmente salen del mismo. Las permanentes son elementos del sistema que permanecen durante toda la simulación.

GPSS esta orientado a la simulación de sistemas discretos cuya estructura de funcionamiento está basada en el enfoque de la interacción del proceso.

Ventajas de GPSS World

- Puede ser continuamente mejorado.
- Está escrito en un lenguaje de programación y por lo tanto es muy veloz.
- Puede resolver una variedad de problemas en una forma rápida y precisa. Dichos problemas pueden ser de diferente naturaleza

tales como los de ingeniería, industria manufacturera, ciencia y negocios.

- Habiendo sido introducido en 1961 por IBM, ha resistido la prueba de tiempo, mientras que otros lenguajes de programación han fallado.

Políticas Planteadas para el Funcionamiento del Simulador "Troncal 7 Orquídeas-Centro Urbano"

Se implementará un sistema de simulación que permita no sólo predecir el comportamiento de la ruta de transporte con las políticas establecidas por el Municipio, sino también analizar el cambio en el comportamiento de dicha ruta con nuevas políticas de operación.

- Intervalo de salida de los Buses Troncales de la Terminal de Integración

- **Constante**

  Se podrá simular la Troncal 7 para cualquier tiempo de salida constante que el usuario desea analizar.

- **Dependiente del Día**

  Es usuario tiene la opción de analizar el comportamiento de la Troncal 7 con tiempos de salidas de los buses que varían dependiendo del día.

- **Dependiente del Día y la Hora**

  Se podrá simular la Troncal 7 con tiempos de salidas de los buses que varían dependiendo del día y la hora de operación.

- Tiempo de Espera del Bus Troncal en el Paradero Troncal

- **Fijo**

  El bus no podrá extender el tiempo de permanencia en el paradero aún cuando todavía existan personas esperando por subir al bus. Únicamente el bus continuará con su recorrido, cuando la capacidad del bus ha sido utilizada por completo, es decir, cuando ya no hay asientos disponibles ya que sería una pérdida de tiempo que espere a que finalice el tiempo de permanencia si ya no tiene asientos disponibles.

- **No puede irse antes de finalizar el tiempo pero sí puede irse después**

  Si ya no hay personas esperando por subir al bus y todavía no finaliza el tiempo de espera el bus esperará hasta que finalice el tiempo de espera para continuar con su recorrido, pero si al finalizar el tiempo de espera en el paradero y todavía hay personas esperando por subir, el bus extenderá su tiempo de espera en el paradero hasta que no exista ninguna persona esperando por subir al bus. El bus continuará con el recorrido

aún cuando todavía no finaliza el tiempo de espera en el caso de que ya no existan asientos disponibles en el bus.

- **Puede irse antes de finalizar el tiempo pero no puede irse después**

  Si todavía no ha finalizado el tiempo de espera en el paradero y no existe nadie esperando por subir, el bus dá por finalizado su tiempo de espera y continúa con el recorrido. Si todavía hay personas esperando por subir y el tiempo de espera ya ha finalizado el bus no extenderá su tiempo de espera y continuará con su recorrido.

- **Puede irse antes y después de finalizar el tiempo de espera**

  Si todavía no ha finalizado el tiempo de espera en el paradero y no existe nadie esperando por subir, el bus continúa con el recorrido, y además si al finalizar el tiempo de espera y todavía hay personas esperando por subir el bus extiende su tiempo de espera hasta que no haya nadie en el paradero esperando por subir.

- ***Comportamiento de las Colas de Espera de los Pasajeros en los Paraderos Troncales***

  - **Aleatoria**

En los paraderos no existirán filas de espera ya que el ascenso de las personas al bus es independiente del orden en que llegan las personas y por lo tanto no habría necesidad que las personas formen fila.

- **FIFO (Primeros en Llegar primeros en ser atendido)**

  La subida de las personas al bus depende de su llegada al paradero, es decir, primeros en llegar primeros en subir al bus, por lo tanto existe una cola física de la espera de las personas en el paradero.

- Ascenso y Descenso de las Personas del Bus Troncal

- **Ascensos y Descensos Independientes**

  Las personas pueden subir y bajar por cualquiera de las 4 puertas que tienen los buses troncales.

- **Ascensos y Descensos Dependientes**

  Primero se realiza las bajadas de las personas del bus por las 8 puertas disponibles y luego se realiza las subidas igualmente por cualquiera de las 8 puertas que tienen los buses troncales.

2.6. CODIFICACIÓN DE PROCESOS EN GPSS WORLD

Los diversos procesos que realiza la Troncal 7 poseen formatos estándar los cuales serán modificados de acuerdo a la información

que el usuario ingrese para la construcción del simulador y además de acuerdo a las políticas de operación de la ruta. Los procesos que simulará GPSS son los siguientes:

1. Simulación de llegadas de pasajeros a los paraderos
2. Simulación del funcionamiento de los semáforos
3. Simulación de la llegada de los pasajeros en los buses alimentadores
4. Simulación del reloj que da por terminada la simulación
5. Simulación del funcionamiento de los paraderos

Simulación de llegadas de pasajeros a los paraderos

En esta simulación debemos tener en cuenta la política de colas de espera de pasajeros en los paraderos que el usuario ha establecido; las opciones para esta política son:

- Colas FIFO o PEPS (Primero en entrar, primero en salir)
- Colas Aleatorias

Colas FIFO o PEPS

La explicación del funcionamiento de las sentencias se encuentra en el anexo.

```
                        INITIAL         $T7_SVL_P1_ORDEN,0
                        GENERATE    FN$T7_FUN_P1_LLEGADAS
T7_LBL_P1_CALCORDEN     EST E       CH$T7_QUE_P1_PASAJEROS,0,T7_LBL_P1_SAVE
                        SAVEVALUE   T7_SVL_P1_ORDEN,0
T7_LBL_P1_SAVE          SAVEVALUE   T7_SVL_P1_ORDEN+,1
                        ASSIGN      ORDEN,X$T7_SVL_P1_ORDEN
                        TEST E      CH$T7_QUE_P1_PASAJEROS,0,T7_LBL_P1_REENTER
                        GATE LR     T7_LGS_P1_SWITCH,T7_LBL_P1
T7_LBL_P1_REENTER       LINK        T7_QUE_P1_PASAJEROS,P$ORDEN
```

Con el Savevalue X$T7_SVL_P1_ORDEN se controla el orden ó el índice del pasajero dentro del userchain. Este contador se va incrementando a medida que ingresan las personas al paradero, es decir, cuando entra la primera persona el savevalue especificado toma el valor de 1 y este se asigna como parámetro a la transacción pasajero para que lo utilice como índice de ordenamiento al momento que entra al userchain, la segunda transacción hará que el savevalue se incremente y tome el valor de 2 y repite el proceso anterior.

Una vez que las transacciones pasajeros ya han sido transferidas al bus, es decir, ya han abandonado el userchain, y en el paradero no hay gente, eso quiere decir que el primer pasajero que llegue al userchain será nuevamente el pasajero 1, por lo tanto, el savevalue X$T7_SVL_P1_ORDEN es reseteado cada vez que el userchain se vacía, esto nos asegura, que el valor de X$T7_SVL_P1_ORDEN no será excesivamente alto lo cual generará menos consumo de memoria.

La secuencia del código especificado anteriormente es la siguiente:

1. Se inicializa el valor del savevalue X$T7_SVL_P1_ORDEN a 0.
2. El bloque GENERATE genera la transacción pasajero con una distribución específica.
3. La transacción prueba si el userchain que simula el paradero está vacío
4. Si la pregunta anterior es verdadera entonces el Savevalue X$T7_SVL_P1_ORDEN se resetea a 1
5. Si la pregunta es falsa entonces el valor del X$T7_SVL_P1_ORDEN se incrementa en una unidad.
6. Se asigna el valor del X$T7_SVL_P1_ORDEN al parámetro ORDEN de la transacción
7. Luego la transacción pregunta si es que el userchain se encuentra vacío para saber si es ó no el primero de la cola, porque si es el primero de la cola, debe preguntar si hay algún bus en el paradero para transferirse directamente al bus y no pasar por la cola, pero si no es primero de la cola, adquiere la posición indicada por su parámetro ORDEN para su ubicación en el userchain.

Cola Aleatoria

	GENERATE	FN$T7_FUN_P1_LLEGADAS
T7_LBL_P1_CALCORDEN	ASSIGN	ORDEN,(CALCULAR_ORDEN(V$T7_VAR_P1_MEDIA,RN1))
	TEST E	CH$T7_QUE_P1_PASAJEROS,0,T7_LBL_P1_REENTER
	GATE LR	T7_LGS_P1_SWITCH,T7_LBL_P1
T7_LBL_P1_REENTER	LINK	T7_QUE_P1_PASAJEROS,P$ORDEN

A diferencia de la política FIFO, la política Aleatoria difiere con respecto al valor que adquiere el parámetro, el cual le indica el orden que le corresponde en la cadena userchain, ya que este ya no es almacenado en un savevalue e incrementado secuencialmente, sino que este valor es calculado para cada transacción de manera aleatoria, y este es el que le indicará la posición o ubicación dentro de la cadena userchain.

La secuencia del código especificado anteriormente es la siguiente:

1. El bloque GENERATE genera la transacción pasajero con una distribución de probabilidad específica.
2. Se asigna al parámetro orden de la transacción una posición aleatoria en base a un procedimiento PLUS que retornará un número aleatorio entre 1 y el número de transacciones que tenga el userchain para saber en que posición ubicarla.
3. La transacción pregunta si es que el userchain se encuentra vacío para saber si es o no el primero de la cola, porque si es el primero de la cola, debe preguntar si hay algún bus en el paradero para transferirse directamente al bus y no ingresar a la cola innecesariamente, pero si no es primero de la cola, adquiere la posición indicada por su parámetro ORDEN para su ubicación en el userchain.

Simulación del funcionamiento de los Semáforos

Para controlar la simulación de los semáforos debemos identificar dos tipos de controles:

- Control de Cambio de estado del Semáforo
- Control de la Llegada del Bus al Semáforo

Control de Cambio de estado del Semáforo

```
                GENERATE        ,,MX$T7_MTX_SEMAF(1,3),1,50
T7_LBL_CICLO1   LOGIC R         T7_LGS_SEMAFORO1
                ADVANCE         MX$T7_MTX_SEMAF(1,1)
                LOGIC S         T7_LGS_SEMAFORO1
                ADVANCE         MX$T7_MTX_SEMAF(1,2)
                TRANSFER        ,T7_LBL_CICLO1
```

Para la simulación de semáforos utilizamos la entidad LOGICSWITCH, donde especificamos que el estado SET del logicswitch simula el estado verde, y el estado RESET simula el estado rojo.

El cambio del semáforo se lo realiza con una sola transacción que se encuentra ciclada dentro de un mismo proceso que altera los estados del semáforo, al iniciar la simulación la transacción sale con un determinado tiempo de retraso (especificado en la 3 columna de la matriz de tiempos) debido a la diferencia de los tiempos de inicio de operación de los semáforos, ya que de no considerar este

retraso, estaríamos diciendo que al inicio de operación todos los semáforos empiezan en un mismo estado y a un mismo tiempo.

La secuencia del código especificado anteriormente es la siguiente:

1. El bloque GENERATE genera una transacción cuando el reloj marque el tiempo de retraso especificado por el elemento de la matriz MX$T7_MTX_SEMAF(1,3), cabe recalcar que es una única transacción la que se genera y que esta tiene la mayor prioridad (50) ya que en caso de que un bus llegue a un paradero y le pregunte por su estado pero al mismo tiempo el semáforo debe cambiar su estado, la transacción que se debe realizar primero debe ser la del cambio del semáforo de otro modo se puede dar la posibilidad de que el bus cruce cuando el semáforo este en luz roja, lo cual es incorrecto.
2. Una vez que la transacción ha sido generada, la transacción entra al bloque LOGIC R cambiando el valor de logicswitch, es decir el valor del semáforo, a estado RESET (estado rojo).
3. Luego se demora el flujo de la transacción con un bloque ADVANCE, el cual simula el tiempo de permanencia del semáforo en luz en roja especificado por la primera columna de la matriz.

4. Luego la transacción entra al bloque LOGIC S cambiando el valor de logicswitch, es decir el valor del semáforo, a estado SET (estado verde).
5. Luego se simula la permanencia del semáforo en luz verde por medio del bloque ADVANCE. Los valores del semáforo en luz verde están especificados en la segunda columna de la matriz.
6. Luego se transfiere la transacción al paso 2. Este ciclo se repite independientemente a los demás procesos que se realicen en la simulación.

Control de llegada del Bus Troncal al Semáforo

```
QUEUE       T_QUE_SEMAFORO1
GATE LS     T_LGS_SEMAFORO1
DEPART      T_QUE_SEMAFORO1
```

Esta secuencia de bloques simula la llegada de una transacción bus a un semáforo, cuando un bus tiene como siguiente objeto en su secuencia de objetos un semáforo, la transacción bus debe preguntarle al semáforo en que estado se encuentra, en caso de que esté en estado verde la transacción bus continua con su recorrido, caso contrario el flujo de la transacción se detiene hasta que el semáforo cambia a luz verde.

La secuencia del código especificado anteriormente es la siguiente:

1. El tiempo que la transacción bus espera a que el semáforo lo deje continuar, se controla mediante una entidad QUEUE, por eso antes de que la transacción pregunte por el estado del semáforo, la transacción ingresa al bloque QUEUE y registra el tiempo actual.
2. Luego la transacción pregunta si es que el semáforo se encuentra en estado verde (SET) para así poder continuar. Si el semáforo se encuentra en estado rojo (RESET) éste le bloquea la entrada y detiene su flujo hasta que el estado cambie.
3. Una vez que la transacción bus ha cruzado el semáforo, el bloque DEPART registra el tiempo actual, para que en conjunto con el tiempo que marcó en el ingreso al Bloque QUEUE se calcule la diferencia y la registre como tiempo de permanencia en espera de que el semáforo permita continuar con el flujo.

Simulación de la llegada de los pasajeros en los buses alimentadores

```
                        GENERATE  FN$T7_FUN_P1_TIEMPOS_ALIMENTADORAS
                        ASSIGN    T7_PAR_P1_PASALIMENT,FN$T7_FUN_P1_LLEG_PERS_ALIM
                        SPLIT     (P$T7_PAR_P1_PASALIMENT-1),T7_LBL_P1_TTALIMENT,T7_PAR_P1_TTID
T7_LBL_P1_TTALIMENT     TRANSFER  ,T7_LBL_P1_CALCORDEN
```

Esta secuencia de bloques simula la llegada de los buses alimentadores al paradero, donde cada alimentadora tiene cierta

distribución de probabilidad, la cual indicará el número de pasajeros que deja en el paradero.

Cada vez que una transacción bus alimentador llega, se asigna al parámetro de la transacción el número de personas que debe dejar, (si la función de distribución le dice que bajen 40 pasajeros, estos 40 son asignados al parámetro para luego simular la bajada), debido a que el único bloque que puede crear transacciones es el bloque GENERATE como éste bloque no puede recibir una transacción, entonces para crear las transacciones de los pasajeros que se bajan del bus alimentador se utiliza un bloque SPLIT, el cual va a realizar copias de la transacción bus alimentador par luego transferirlas al paradero convirtiendo la copia de la transacción bus alimentador en transacción pasajero.

La secuencia del código especificado anteriormente es la siguiente:

1. Una transacción sale del bloque GENERATE con cierta distribución de probabilidad.
2. Luego asigna al parámetro T7_PAR_P1_PASALIMENT, el número de pasajeros que debe dejar dependiendo de lo que indique la distribución de probabilidad.
3. Luego crea las copias que luego serán convertidas en transacciones pasajeros mediante el comando SPLIT, debido a que el comando SPLIT crea copias de transacciones pero no

elimina la transacción original de donde fueron creadas las copias, entonces se toma a la transacción original también como transacción pasajero, por lo tanto se la debe incluir en el número de pasajeros que debe dejar la alimentadora. Es por esto que el bloque SPLIT crea n-1 copias de la transacción donde n es el número de pasajeros que indica el parámetro.

4. Luego se transfiere tanto el original como las copias hacia el paradero

Simulación del Reloj que dá por terminada la simulación

```
GENERATE        3600
TERMINATE          1
```

Toda simulación en GPSS se controla mediante el SNA Global llamado TG1, cuando el valor del TG1 es cero se dá por finalizada la simulación. El valor de este SNA es asignado por el comando START, es decir que si corremos una simulación con un START 19, este valor de 19 es enviado al TG1, y el único comando que tiene la capacidad de decrementar el valor del TG1 es el bloque TERMINATE pero con su operando "**A**" diferente de cero.

Una iteración de nuestra simulación comprenderá la operación de la troncal 7 desde las 5 de la mañana hasta las 12 de la noche, es decir, una iteración de nuestra simulación terminará cuando se haya ejecutado el modelo durante 19 horas de operación que en

segundos equivale a 68400 segundos. Es por esto que para el control del tiempo de terminación de nuestra simulación se utilizó el bloque GENERATE para crear transacciones cada 3600 segundos (cada hora), la única función de esta transacción es entrar a un bloque TERMINATE 1, es decir, que cada transacción que se genera cada hora decrementará en una unidad al TG1, por lo tanto, si queremos simular 19 horas para cada iteración bastará con ejecutar un comando START con parámetro “A” de 19.

La secuencia del código especificado anteriormente es la siguiente:

1. Un transacción sale del bloque GENERATE cada 3600 segundos
2. Esta transacción decrementa el valor del TG1 en una unidad

Simulación del funcionamiento de los paraderos

Tabla 1
Combinaciones de Políticas de los Paraderos

Política de Ascensos y Descensos del bus	Política de Espera del Bus en el Paradero
Subidas independiente de las bajadas	Fijo-No puede exceder el tiempo de espera
	Fijo-Si puede exceder el tiempo de espera
	Flexible-No puede exceder el tiempo de espera
	Flexible-Si puede exceder el tiempo de espera
Primero bajan, luego suben al bus	Fijo-No puede exceder el tiempo de espera
	Fijo-Si puede exceder el tiempo de espera
	Flexible-No puede exceder el tiempo de espera
	Flexible-Si puede exceder el tiempo de espera

El código de los Paraderos depende de las distintas combinaciones que existen para el funcionamiento de los mismos:

A pesar de las diferentes combinaciones, una parte del código funciona indistintamente de la combinación en la que se esté trabajando, es por esto que primero se detallará esta primera parte, para luego especificar el cambio que sufre el código cuando la política cambia:

```
                        DEPART      T7_QUE_P1_TACUM
                        QUEUE       T7_QUE_P1_GENERAL
                        QUEUE       T7_QUE_P1_ESPERA
                        ENTER       T7_STO_P1_ESPACIO
                        DEPART      T7_QUE_P1_ESPERA
                        TEST NE     S$T7_STO_P1_CAPACIDAD,0,T7_LBL_P1_ET7
                        LEAVE       T7_STO_P1_CAPACIDAD,S$T7_STO_P1_CAPACIDAD
T7_LBL_P1_ET7           ENTER       T7_STO_P1_CAPACIDAD,P$T7_PAR_CAPACTUAL
                        SAVEVALUE   T7_SAV_P1_TOTAL,P$T7_PAR_TCAPACIDAD
                        LOGIC R     T7_LGS_P1_END_SUB
                        LOGIC R     T7_LGS_P1_SWITCH
                        LOGIC R     T7_LGS_P1_TIME_IS_UP
                        ASSIGN      T7_PAR_P1_BAJADAS,(PR_CALCULAR_BAJADAS(P$T7_PAR_CAPACTUAL,FN$T7_FUN_P1_BAJADAS))
                        TEST NE     P$T7_PAR_P1_BAJADAS,0,T7_LBL_P1_SPLIT
                        LEAVE       T7_STO_P1_CAPACIDAD,P$T7_PAR_P1_BAJADAS
T7_LBL_P1_SPLIT         SPLIT       2,T7_LBL_P1_DESTINOS,T7_PAR_P1_ID
T7_LBL_P1_DESTINOS      TRANSFER    FN,T7_FUN_P1_DESTINOS1
```

En esta secuencia de bloque se simula la llegada de la transacción bus al paradero, cuando el bus llega al paradero debe preguntar si puede entrar al paradero ya que los paraderos solo pueden albergar un bus a la vez, y si este espacio se encuentra ocupado por otra transacción bus, entonces la transacción tendrá que esperar a que el almacenamiento se desocupe para poder ingresar, para contabilizar el tiempo que el bus debe esperar hasta que se desocupe dicho almacenamiento se utiliza el bloque QUEUE.

Existen tres indicadores propios de cada paradero los cuales funcionan como variables booleanas (verdadero o falso), en GPSS se simulan estas variables mediante las entidades LOGICSWITCHES las cuales se detallan a continuación:

T7_LGS_P1_END_SUB

Cuando el logicswitch está en estado RESET indica que la bajada de pasajeros todavía no ha terminado. Si esta en estado SET indica que la baja de pasajeros ya termino.

T7_LGS_P1_TIME_IS_UP

Si el logicswitch está en estado RESET el tiempo que debe permanecer el bus en el paradero todavia no ha concluido. Si esta en estado SET indica que el tiempo que debe permanecer el bus en el paradero ya concluyo.

T7_LGS_P1_SWITCH

Si esta en estado RESET la transacción pasajeros la cual se encuentra en el paradero todavía no pueden subir al bus.

Si esta en estado SET la transacción pasajeros la cual se encuentra en el paradero ya pueden iniciar su ascenso al bus.

Este switch es muy importante ya que es el medio de comunicación de la transacción bus con la transacción pasajero, ya que este switch

es el encargado de indicar que el avance de los pasajeros al bus puede iniciar.

Todos estos indicador son fundamentales para la identificación del tipo de política de operación del paradero.

Cuando una transacción bus entra al paradero los 3 procesos inician su operación de forma simultánea e independiente (depende de la politica).

1. El control del tiempo que el bus permanece en el paradero
2. El control de los pasajeros que bajan del bus.
3. El control de los pasajeros que suben al bus.

Debido a que la capacidad utilizada del bus depende de cada bus, es decir, es un atributo propio de cada transacción, el cual se va ir trasladando para su manipulación de paradero en paradero, no se puede utilizar una variable global para controlar esta capacidad, ya que no se sabe a ciencia cierta cuantos buses se va a tener en circulación, sino que más bien se utiliza un parámetro para almacenar esta capacidad utilizada, sabiendo que los parámetros son atributos propios de cada transacción.

Al decir que 3 procesos deben empezar a funcionar desde el momento en que la transacción bus llegue al paradero, y que cada

proceso es independiente uno del otro, entonces una sola transacción no puede controlar dicha simulación, sino que se utiliza un bloque SPLIT para crear dos copias de esta transacción y así poder asignar una transacción para cada proceso; al llevar a cabo dicha división surge un nuevo problema que es que cada transacción (la original y las 2 copias) van a tener un parámetro que indique la capacidad utilizada del bus, y si cada proceso es independiente entonces cada proceso alterará independientemente el valor de su parámetro, cuando en realidad se está hablando de un mismo bus, para evitar este inconveniente antes de hacer la división de la transacción bus, se transfiere el valor de la capacidad actual a un almacenamiento temporal ("storage") para que las tres transacciones en lugar de modificar su parámetro, éstas modifiquen un almacenamiento en común, para evitar así incoherencias en el manejo de los datos. Una vez que los tres procesos han sido concluidos, las tres transacciones vuelven a ensamblarse para convertirse en una sola, el valor del almacenamiento temporal es devuelto al parámetro de la transacción bus.

La secuencia del código especificado anteriormente es la siguiente:

1. Liberar la Entidad QUEUE T7_QUE_P1_TACUM, esta entidad nos dará las estadísticas del tiempo que demoró el bus en llegar al paradero desde la Terminal de Integración.

2. Entrar a la Entidad QUEUE T7_QUE_P1_GENERAL, la cual nos va a dar las estadísticas del tiempo total que permaneció el bus en el paradero incluido el tiempo de espera para poder entrar al paradero.
3. Entrar a la Entidad QUEUE T7_QUE_P1_ ESPERA, la cual nos va a dar estadísticas del tiempo de espera para poder entrar al paradero.
4. El bus intenta Ingresar al paradero si el almacenamiento (de capacidad 1) T7_STO_P1_ESPACIO se encuentra ocupado este niega la entrada a la transacción la cual tendrá que esperar hasta que el espacio este desocupado, si el almacenamiento está desocupado el bus ingresa al paradero.
5. Se libera la entidad QUEUE T7_QUE_P1_ESPERA
6. Se inicializa el almacenamiento temporal y se asigna el valor del parámetro que indica la capacidad utilizada a dicho almacenamiento.
7. Se coloca en estado RESET todos los indicadores del paradero.
8. Se decrementa del almacenamiento temporal el valor que retorne la función de distribución que indica el número de bajadas de personas en el paradero para ese paradero en particular y para esa hora en particular.

9. Como ya mencionamos anteriormente los 3 procesos se deben iniciar es por esto que con el Bloque SPLIT se crean dos copias y cada una de las tres transacciones es transferida a su proceso correspondiente.

Subidas Independiente de las Bajadas

Tiempo de Espera del bus es Fijo No puede Exceder

- Ascenso y Descenso son procesos independientes
- El bus no puede irse antes ni puede irse después

```
T7_LBL_P1_ESPBUS1     LOGIC R  T7_LGS_P1_TIME_IS_UP
                      QUEUE        T7_QUE_P1_TIEMPO_ESPERA
                      ASSIGN       T7_PAR_P1_TIEMPO_BUS,FN$T7_FUN_P1_ESPERAS
T7_LBL_P1_LOOP        ADVANCE      1
                      TEST NE      (S$T7_STO_P1_CAPACIDAD),X$T7_SAV_P1_TOTAL,T7_LBL_P1_END_TIME
T7_LBL_P1_TLOOP       LOOP         T7_PAR_P1_TIEMPO_BUS,T7_LBL_P1_LOOP
T7_LBL_P1_END_TIME    GATE LS      T7_LGS_P1_END_SUB
                      LOGIC S      T7_LGS_P1_TIME_IS_UP
                      LOGIC R      T7_LGS_P1_SWITCH
                      DEPART       T7_QUE_P1_TIEMPO_ESPERA
T7_LBL_P1_SALT7       TRANSFER     ,T7_LBL_P1_ENSAMBLAJE
T7_LBL_P1_BAJADAS     LOGIC R      T7_LGS_P1_END_SUB
                      QUEUE        T7_QUE_P1_TIME_BAJADAS
                      TEST NE      P$T7_PAR_P1_BAJADAS,0,T7_LBL_P1_FINBAJ
                      ADVANCE      (CALCULAR_TIEMPO(4,V$T7_VAR_TIEMPO_BAJADA,P$T7_PAR_P1_BAJADAS))
T7_LBL_P1_FINBAJ      DEPART       T7_QUE_P1_TIME_BAJADAS
                      OGIC S       T7_LGS_P1_END_SUB
                      TRANSFER     ,T7_LBL_P1_ENSAMBLAJE
T7_LBL_P1_SUBIDAS     ST L     (S$T7_STO_P1_CAPACIDAD),X$T7_SAV_P1_TOTAL,T7_LBL_P1_JUMP1
                      QUEUE        T7_QUE_P1_TIEMPO_SUBIDAS
                      LOGIC S      T7_LGS_P1_SWITCH
                      NLINK        T7_QUE_P1_PASAJEROS,T7_LBL_P1,1
                      GATE LR      T7_LGS_P1_SWITCH
                      GATE SE      T7_STO_P1_DOOR
                      DEPART       T7_QUE_P1_TIEMPO_SUBIDAS
T7_LBL_P1_JUMP1       TRANSFER     ,T7_LBL_P1_ENSAMBLAJE
```

En esta secuencia de bloques se simulan los tres procesos anteriormente especificados los cuales varían dependiendo del tipo de política seleccionada, para esta combinación en particular se tiene que el ascenso y descenso de pasajeros son independientes,

por ello podemos observar que el proceso de subidas no espera a que el proceso de bajadas termine. El tiempo que el bus debe permanecer en el paradero es fijo el único caso que permitiría irse al bus antes del tiempo establecido es cuando la capacidad del bus haya llegado a su máxima ocupación.

Tiempo de Espera del bus es Fijo Si puede Exceder

- Ascenso y Descenso Independientes
- El bus troncal no puede irse antes pero si puede irse después

```
T7_LBL_P1_ESPBUS1  LOGIC R     T7_LGS_P1_TIME_IS_UP
                   QUEUE       T7_QUE_P1_TIEMPO_ESPERA
                   ASSIGN      T7_PAR_P1_TIEMPO_BUS,FN$T7_FUN_P1_ESPERAS
 T7_LBL_P1_LOOP    ADVANCE  1
                   TEST  NE    (S$T7_STO_P1_CAPACIDAD),X$T7_SAV_P1_TOTAL,T7_LBL_P1_END_TIME
 T7_LBL_P1_TLOOP   LOOP        T7_PAR_P1_TIEMPO_BUS,T7_LBL_P1_LOOP
                   GATE LS     T7_LGS_P1_END_SUB
 T7_LBL_P1_TESTS   TEST NE     CH$T7_QUE_P1_PASAJEROS,0,T7_LBL_P1_END_TIME
                   TEST NE     (S$T7_STO_P1_CAPACIDAD),X$T7_SAV_P1_TOTAL,T7_LBL_P1_END_TIME
                   ADVANCE   1
                   TRANSFER  ,T7_LBL_P1_TESTS
 T7_LBL_P1_END_TIME LOGIC S    T7_LGS_P1_TIME_IS_UP
                    LOGIC R    T7_LGS_P1_SWITCH
                    DEPART     T7_QUE_P1_TIEMPO_ESPERA
 T7_LBL_P1_SALT7    TRANSFER   T7_LBL_P1_ENSAMBLAJE
 T7_LBL_P1_BAJADAS  LOGIC R    T7_LGS_P1_END_SUB
                    QUEUE      T7_QUE_P1_TIME_BAJADAS
                    TEST NE    P$T7_PAR_P1_BAJADAS,0,T7_LBL_P1_FINBAJ
                    ADVANCE    (CALCULAR_TIEMPO(4,V$T7_VAR_TIEMPO_BAJADA,P$T7_PAR_P1_BAJADAS))
 T7_LBL_P1_FINBAJ   DEPART     7_QUE_P1_TIME_BAJADAS
                    LOGIC S    T7_LGS_P1_END_SUB
                    TRANSFER   T7_LBL_P1_ENSAMBLAJE
 T7_LBL_P1_SUBIDAS  TEST L     (S$T7_STO_P1_CAPACIDAD),X$T7_SAV_P1_TOTAL,T7_LBL_P1_JUMP1
                    QUEUE      T7_QUE_P1_TIEMPO_SUBIDAS
                    LOGIC S    T7_LGS_P1_SWITCH
                    UNLINK     T7_QUE_P1_PASAJEROS,T7_LBL_P1,1
                    GATE LR    T7_LGS_P1_SWITCH
                    GATE SE    T7_STO_P1_DOOR
                    DEPART     T7_QUE_P1_TIEMPO_SUBIDAS
 T7_LBL_P1_JUMP1    TRANSFER   T7_LBL_P1_ENSAMBLAJE
```

En esta secuencia de bloques se simulan los tres procesos explicados anteriormente, los cuales varían dependiendo del tipo de

política seleccionada, para esta combinación en particular se tiene que el ascenso y descenso de pasajeros son independientes, por ello podemos observar que el proceso de subidas no espera a que el proceso de bajadas termine.

El tiempo que el bus debe permanecer en el paradero es fijo y por lo tanto el bus no puede irse antes del tiempo especificado, pero si todavía existen pasajeros en cola al finalizar el tiempo establecido el bus espera a que éstas personas suban por tanto puede irse después del tiempo especificado.

En esta y en las demás políticas el único caso en que el bus se vaya antes del tiempo establecido es cuando la capacidad del bus haya llegado a su máxima ocupación, es decir, cuando los 180 asientos del bus estén ocupados.

Tiempo de Espera del bus es Flexible No puede Exceder

- Ascenso y Descenso Independientes
- El bus troncal si puede irse antes pero no puede irse después

```
T7_LBL_P1_ESPBUS1  LOGIC R     T7_LGS_P1_TIME_IS_UP
                   QUEUE       T7_QUE_P1_TIEMPO_ESPERA
                   ASSIGN      T7_PAR_P1_TIEMPO_BUS,FN$T7_FUN_P1_ESPERAS
T7_LBL_P1_LOOP     ADVANCE     1
                   TEST NE     P$T7_PAR_P1_TIEMPO_BUS,1,T7_LBL_P1_TLOOP
                   GATE LS     T7_LGS_P1_END_SUB,T7_LBL_P1_TLOOP
                   TEST NE     CH$T7_QUE_P1_PASAJEROS,0,T7_LBL_P1_END_TIME
                   TEST NE     (S$T7_STO_P1_CAPACIDAD),X$T7_SAV_P1_TOTAL,T7_LBL_P1_END_TIME
T7_LBL_P1_TLOOP    LOOP        T7_PAR_P1_TIEMPO_BUS,T7_LBL_P1_LOOP
T7_LBL_P1_END_TIME GATE LS     T7_LGS_P1_END_SUB
                   LOGIC S     T7_LGS_P1_TIME_IS_UP
                   LOGIC R     T7_LGS_P1_SWITCH
                   DEPART      T7_QUE_P1_TIEMPO_ESPERA
T7_LBL_P1_SALT7    TRANSFER ,T7_LBL_P1_ENSAMBLAJE
T7_LBL_P1_BAJADAS  LOGIC R     T7_LGS_P1_END_SUB
                   QUEUE       T7_QUE_P1_TIME_BAJADAS
                   ADVANCE     (CALCULAR_TIEMPO(4,V$T7_VAR_TIEMPO_BAJADA,P$T7_PAR_P1_BAJADAS))
T7_LBL_P1_FINBAJ   DEPART      T7_QUE_P1_TIME_BAJADAS
                   LOGIC S     T7_LGS_P1_END_SUB
                   TRANSFER  ,T7_LBL_P1_ENSAMBLAJE
T7_LBL_P1_SUBIDAS  TEST L      (S$T7_STO_P1_CAPACIDAD),X$T7_SAV_P1_TOTAL,T7_LBL_P1_JUMP1
                   QUEUE       T7_QUE_P1_TIEMPO_SUBIDAS
                   LOGIC S     T7_LGS_P1_SWITCH
                   UNLINK      T7_QUE_P1_PASAJEROS,T7_LBL_P1,1
                   GATE LR     T7_LGS_P1_SWITCH
                   GATE SE     T7_STO_P1_DOOR
                   DEPART      T7_QUE_P1_TIEMPO_SUBIDAS
T7_LBL_P1_JUMP1    TRANSFER  ,T7_LBL_P1_ENSAMBLAJE
```

Para esta combinación en particular se tiene que el ascenso y descenso de pasajeros son independientes, por ello el proceso de subidas no espera a que el proceso de bajadas termine. El tiempo que el bus debe permanecer en el paradero es flexible, lo cual indica que el bus puede irse antes del tiempo especificado en caso de que no haya pasajeros en cola, pero una vez finalizado el tiempo máximo de espera aunque existan pasajeros en cola el bus no espera a que estas personas suban por tanto no puede irse después del tiempo especificado.

Tiempo de Espera del bus es Flexible Si puede Exceder

- Ascenso y Descenso Independientes

- El bus troncal puede irse antes y también puede irse después de finalizar el tiempo de espera

```
T7_LBL_P1_ESPBUS1  LOGIC R       T7_LGS_P1_TIME_IS_UP
                   QUEUE         T7_QUE_P1_TIEMPO_ESPERA
                   GATE LS       T7_LGS_P1_END_SUB
T7_LBL_P1_TESTS    TEST NE       CH$T7_QUE_P1_PASAJEROS,0,T7_LBL_P1_END_TIME
                   TEST NE       (S$T7_STO_P1_CAPACIDAD),X$T7_SAV_P1_TOTAL,T7_LBL_P1_END_TIME
                   ADVANCE       1
                   TRANSFER      ,T7_LBL_P1_TESTS
T7_LBL_P1_END_TIME LOGIC S       T7_LGS_P1_TIME_IS_UP
                   LOGIC R       T7_LGS_P1_SWITCH
                   DEPART        T7_QUE_P1_TIEMPO_ESPERA
T7_LBL_P1_SALT7    TRANSFER ,T7_LBL_P1_ENSAMBLAJE
T7_LBL_P1_BAJADAS  LOGIC R       T7_LGS_P1_END_SUB
                   QUEUE         T7_QUE_P1_TIME_BAJADAS
                   TEST NE       P$T7_PAR_P1_BAJADAS,0,T7_LBL_P1_FINBAJ
                   ADVANCE       (CALCULAR_TIEMPO(4,V$T7_VAR_TIEMPO_BAJADA,P$T7_PAR_P1_BAJADAS))
T7_LBL_P1_FINBAJ   DEPART        T7_QUE_P1_TIME_BAJADAS
                   LOGIC S       T7_LGS_P1_END_SUB
                   TRANSFER      ,T7_LBL_P1_ENSAMBLAJE
T7_LBL_P1_SUBIDAS  TEST L        (S$T7_STO_P1_CAPACIDAD),X$T7_SAV_P1_TOTAL,T7_LBL_P1_JUMP1
                   QUEUE         T7_QUE_P1_TIEMPO_SUBIDAS
                   LOGIC S       T7_LGS_P1_SWITCH
                   UNLINK        T7_QUE_P1_PASAJEROS,T7_LBL_P1,1
                   GATE LR       T7_LGS_P1_SWITCH
                   GATE SE       T7_STO_P1_DOOR
                   DEPART        T7_QUE_P1_TIEMPO_SUBIDAS
T7_LBL_P1_JUMP1    TRANSFER ,T7_LBL_P1_ENSAMBLAJE
```

El ascenso y descenso de pasajeros son independientes, lo cual indica que el proceso de subidas no espera a que el proceso de bajadas termine, es decir, las personas empiezan a subir y a bajar al mismo tiempo.

El tiempo que el bus debe permanecer en el paradero es flexible, y por lo tanto el bus puede irse antes en el caso de que no haya pasajeros en cola, y además, puede irse después esto se dá cuando ya ha finalizado el tiempo de espera y todavía están pasajeros en cola esperando por subir al bus.

Primero bajan luego suben

Tiempo de Espera del bus es Fijo No puede Exceder

- Primero los descensos después los ascensos
- El bus troncal no puede irse antes ni tampoco pueden irse después

```
T7_LBL_P1_ESPBUS1  LOGIC R      T7_LGS_P1_TIME_IS_UP
                   QUEUE        T7_QUE_P1_TIEMPO_ESPERA
                   ASSIGN       T7_PAR_P1_TIEMPO_BUS,FN$T7_FUN_P1_ESPERAS
T7_LBL_P1_LOOP     ADVANCE   1
                   TEST NE      (S$T7_STO_P1_CAPACIDAD),X$T7_SAV_P1_TOTAL,T7_LBL_P1_END_TIME
T7_LBL_P1_TLOOP    LOOP         T7_PAR_P1_TIEMPO_BUS,T7_LBL_P1_LOOP
T7_LBL_P1_END_TIME GATE LS      T7_LGS_P1_END_SUB
                   LOGIC S      T7_LGS_P1_TIME_IS_UP
                   LOGIC R      T7_LGS_P1_SWITCH
                   DEPART       T7_QUE_P1_TIEMPO_ESPERA
T7_LBL_P1_SALT7    TRANSFER  ,T7_LBL_P1_ENSAMBLAJE
T7_LBL_P1_BAJADAS  LOGIC R      T7_LGS_P1_END_SUB
                   QUEUE        T7_QUE_P1_TIME_BAJADAS
                   TEST NE      P$T7_PAR_P1_BAJADAS,0,T7_LBL_P1_FINBAJ
                   ADVANCE      (CALCULAR_TIEMPO(8,V$T7_VAR_TIEMPO_BAJADA,P$T7_PAR_P1_BAJADAS))
T7_LBL_P1_FINBAJ   DEPART       T7_QUE_P1_TIME_BAJADAS
                   LOGIC S      T7_LGS_P1_END_SUB
                   TRANSFER  T7_LBL_P1_ENSAMBLAJE
T7_LBL_P1_SUBIDAS  GATE LS      T7_LGS_P1_END_SUB
                   TEST L       (S$T7_STO_P1_CAPACIDAD),X$T7_SAV_P1_TOTAL,T7_LBL_P1_JUMP1
                   QUEUE        T7_QUE_P1_TIEMPO_SUBIDAS
                   LOGIC S      T7_LGS_P1_SWITCH
                   UNLINK       T7_QUE_P1_PASAJEROS,T7_LBL_P1,1
                   GATE LR      T7_LGS_P1_SWITCH
                   GATE SE      T7_STO_P1_DOOR
                   DEPART       T7_QUE_P1_TIEMPO_SUBIDAS
T7_LBL_P1_JUMP1    TRANSFER ,T7_LBL_P1_ENSAMBLAJE
```

El ascenso y descenso de pasajeros son dependientes, es decir, el proceso de ascenso de pasajeros debe esperar a que el proceso de descenso haya finalizado, por ello podemos observar que el proceso de subidas espera a que el proceso de bajadas termine con un GATE al Logicswitch.

También debemos recalcar que para este tipo de políticas la capacidad de las puertas del bus se incrementa al doble, ya que los

procesos de ascenso y descenso se realizan de manera independiente.

El tiempo que el bus debe permanecer en el paradero es fijo, y por lo tanto el bus no puede irse antes, ni después.

Tiempo de Espera del bus es Fijo Si puede Exceder

- Primero Ascensos luego descensos
- El bus troncal no puede irse antes pero si puede irse después

```
T7_LBL_P1_ESPBUS1    LOGIC R     T7_LGS_P1_TIME_IS
QUEUE                T7_QUE_P1_TIEMPO_ESPERA
                     ASSIGN      T7_PAR_P1_TIEMPO_BUS,FN$T7_FUN_P1_ESPERAS
T7_LBL_P1_LOOP       ADVANCE     1
                     TEST NE     (S$T7_STO_P1_CAPACIDAD),X$T7_SAV_P1_TOTAL,T7_LBL_P1_END_TIME
T7_LBL_P1_TLOOP      LOOP        T7_PAR_P1_TIEMPO_BUS,T7_LBL_P1_LOOP
                     GATE LS     T7_LGS_P1_END_SUB
T7_LBL_P1_TESTS      TEST NE     CH$T7_QUE_P1_PASAJEROS,0,T7_LBL_P1_END_TIME
                     TEST NE     (S$T7_STO_P1_CAPACIDAD),X$T7_SAV_P1_TOTAL,T7_LBL_P1_END_TIME
                     ADVANCE     1
                     TRANSFER    T7_LBL_P1_TESTS
T7_LBL_P1_END_TIME   LOGIC S     T7_LGS_P1_TIME_IS_UP


LOGIC R              T7_LGS_P1_SWITCH
                     DEPART      7_QUE_P1_TIEMPO_ESPERA
T7_LBL_P1_SALT7      TRANSFER    ,T7_LBL_P1_ENSAMBLAJE

T7_LBL_P1_BAJADAS    LOGIC R     T7_LGS_P1_END_SUB
                     QUEUE       T7_QUE_P1_TIME_BAJADAS
                     TEST NE     P$T7_PAR_P1_BAJADAS,0,T7_LBL_P1_FINBAJ
                     ADVANCE     (CALCULAR_TIEMPO(4,V$T7_VAR_TIEMPO_BAJADA,P$T7_PAR_P1_BAJADAS))

T7_LBL_P1_FINBAJ     DEPART      T7_QUE_P1_TIME_BAJADAS

                     LOGIC S             T7_LGS_P1_END_SUB
                     TRANSFER    ,T7_LBL_P1_ENSAMBLAJE
T7_LBL_P1_SUBIDAS    GATE LS     T7_LGS_P1_END_SUB
                     TEST L              (S$T7_STO_P1_CAPACIDAD), X$T7_SAV_P1_TOTAL,T7_LBL_P1_JUMP1
                     QUEUE       T7_QUE_P1_TIEMPO_SUBIDA
                     LOGIC S     T7_LGS_P1_SWITCH
                     UNLINK      T7_QUE_P1_PASAJEROS,T7_LBL_P1,1
                     GATE LR     T7_LGS_P1_SWITCH
                     GATE SE     T7_STO_P1_DOOR
                     DEPART      T7_QUE_P1_TIEMPO_SUBIDAS
T7_LBL_P1_JUMP1      TRANSFER    ,T7_LBL_P1_ENSAMBLAJE
```

Primero se realizan las bajadas de las personas al bus, para luego realizar las subidas.

El tiempo que el bus debe permanecer en el paradero es fijo, y por lo tanto el bus no puede irse antes del tiempo especificado, pero si todavía existen pasajeros en cola al finalizar el tiempo establecido el bus espera a que estas personas suban por tanto puede irse después del tiempo especificado.

Tiempo de Espera del bus es Flexible No puede Exceder

- Ascensos y descensos dependientes
- Si puede irse antes pero no puede irse después

```
T7_LBL_P1_ESPBUS1   LOGIC R      T7_LGS_P1_TIME_IS_UP
                    QUEUE        T7_QUE_P1_TIEMPO_ESPE
                    ASSIGN       T7_PAR_P1_TIEMPO_BUS,FN$T7_FUN_P1_ESPERAS
T7_LBL_P1_LOOP      ADVANCE      1
TEST NE             P$T7_PAR_P1_TIEMPO_BUS,1,T7_LBL_P1_TLOOP
                    GATE LS      T7_LGS_P1_END_SUB,T7_LBL_P1_TLOOP
                    TEST NE      CH$T7_QUE_P1_PASAJEROS,0,T7_LBL_P1_END_TIME
                    TEST NE      (S$T7_STO_P1_CAPACIDAD),X$T7_SAV_P1_TOTAL,T7_LBL_P1_END_TIME
T7_LBL_P1_TLOOP     LOOP         T7_PAR_P1_TIEMPO_BUS,T7_LBL_P1_LOOP
T7_LBL_P1_END_TIME  GATE LS      T7_LGS_P1_END_SUB
                    LOGIC S      T7_LGS_P1_TIME_IS_UP
                    LOGIC R      T7_LGS_P1_SWITCH
                    DEPART       T7_QUE_P1_TIEMPO_ESPERA
T7_LBL_P1_SALT7     TRANSFER     ,T7_LBL_P1_ENSAMBLAJE
T7_LBL_P1_BAJADAS   LOGIC R      T7_LGS_P1_END_SUB
                    QUEUE        T7_QUE_P1_TIME_BAJADAS
                    TEST NE      P$T7_PAR_P1_BAJADAS,0,T7_LBL_P1_FINBAJ
                    ADVANCE      (CALCULAR_TIEMPO(8,V$T7_VAR_TIEMPO_BAJADA,P$T7_PAR_P1_BAJADAS))
T7_LBL_P1_FINBAJ    DEPART       T7_QUE_P1_TIME_BAJADAS
                    LOGIC S      T7_LGS_P1_END_SUB
                    TRANSFER ,T7_LBL_P1_ENSAMBLAJE
T7_LBL_P1_SUBIDAS   GATE LS      T7_LGS_P1_END_SUB
                    TEST L       (S$T7_STO_P1_CAPACIDAD),X$T7_SAV_P1_TOTAL,T7_LBL_P1_JUMP1
                    QUEUE        T7_QUE_P1_TIEMPO_SUBIDAS
                    LOGIC S      T7_LGS_P1_SWITCH
                    UNLINK       T7_QUE_P1_PASAJEROS,T7_LBL_P1,1
                    GATE LR      T7_LGS_P1_SWITCH
                    GATE SE      T7_STO_P1_DOOR
                    DEPART       T7_QUE_P1_TIEMPO_SUB IDAS
T7_LBL_P1_JUMP1     TRANSFER ,T7_LBL_P1_ENSAMBLAJE
```

Las subidas y bajadas funcionan igual que en la combinación anterior. La política de espera del bus en el paradero indica que el bus puede irse antes del tiempo especificado en caso de que no haya pasajeros en cola, pero también puede irse después de que haya finalizado el tiempo de espera, se dá este caso cuando todavía existen pasajeros en cola esperando por subir al bus.

Segunda Parte

Esta última secuencia de bloques es la misma para todas las combinaciones anteriormente descritas.

```
T7_LBL_P1                GATE LS      T7_LGS_P1_SWITCH,T7_LBL_P1_REENTER
                         GATE SNF     T7_STO_P1_DOOR
                         ENTER        T7_STO_P1_CAPACIDAD,1
                         GATE LR      T7_LGS_P1_TIME_IS_UP,T7_LBL_P1_CAMBIO
                         TEST L       S$T7_STO_P1_CAPACIDAD,X$T7_SAV_P1_TOTAL,T7_LBL_P1_CAMBIO2
                         ENTER        T7_STO_P1_DOOR
                         QUEUE        T7_QUE_P1_SUBIDAS
                         UNLINK       T7_QUE_P1_PASAJEROS,T7_LBL_P1,1
                         ADVANCE      V$T7_VAR_TIEMPO_SUBIDA
                         LEAVE        T7_STO_P1_DOOR
                         TERMINATE
T7_LBL_P1_CAMBIO         LOGIC R      T7_LGS_P1_SWITCH
                         LEAVE        T7_STO_P1_CAPACIDAD,1
                         TRANSFER     ,T7_LBL_P1_REENTER
T7_LBL_P1_CAMBIO2        LOGIC R      T7_LGS_P1_SWITCH
                         ENTER        T7_STO_P1_DOOR
                         QUEUE        T7_QUE_P1_SUBIDAS
                         ADVANCE      V$T7_VAR_TIEMPO_SUBIDA
                         LEAVE        T7_STO_P1_DOOR
                         TERMINATE
T7_LBL_P1_ENSAMBLAJE     ASSEMBLE     3
                         ASSIGN       T7_PAR_CAPACTUAL,S$T7_STO_P1_CAPACIDAD
                         DEPART       T7_QUE_P1_SUBIDAS,Q$T7_QUE_P1_SUBIDAS
                         LEAVE        T7_STO_P1_ESPACIO
                         DEPART       T7_QUE_P1_GENERAL
```

Los procesos que se realizan son los siguientes:

1. Se simula la transferencia de los pasajeros desde la cola de espera en el paradero, hacia el bus troncal.

2. Se ensambla las 3 transacciones de cada uno de los procesos para formar una única transacción bus.
3. Se transfiere el valor del almacenamiento temporal al parámetro de la transacción bus.
4. Se libera el espacio del paradero y se continua con la secuencia

2.7. DISEÑO DE LA BASE DE DATOS

Una base de datos es una colección de información almacenada en un soporte informático. Los datos estarán integrados de tal forma que su estructura refleja las interrelaciones y restricciones existentes en el mundo real.

Claves

Son atributos que identifican a cada registro almacenado. Una tabla puede tener más de una clave. Las cuales pueden ser primaria o secundaria.

Clave primaria

Es aquella clave que el usuario escogerá para identificar los registros en una tabla.

Clave secundaria

Son aquellas claves que permiten obtener información de otras tablas en las cuales ésta clave secundaria opera como clave primaria.

DESCRIPCIÓN DE TABLAS

Las tablas que forman parte de la base de dato en la cual se almacenará la información requerida por el simulador y los resultados obtenidos en la simulación del Sistema METROVÍA Troncal 7, se detallan a continuación:

Tabla: *simulaciones*

Se entiende por "simulación" a cada diseño de ruta de transporte que el usuario crea. Por ejemplo un diseño de ruta de la troncal 7 puede estar conformado por 24 paraderos, 8 semáforos y con unas determinadas políticas de operación, y luego hacer otro diseño 30 paraderos, 10 semáforos y con otras políticas de funcionamiento de la ruta.

Estos dos diseños son independientes entre sí de tal forma que se puede evaluar individualmente cada resultado.

Tabla 2
Campos de la tabla *simulaciones*

Nombre	Clave	Tipo de Dato	Longitud
Id	primaria	numeric	9
nombre	-	varchar	200
fecha	-	datetime	8

id : Código secuencial, identificador único de cada registro.

nombre : Nombre del modelo de simulación que se va a crear.

fecha : Fecha de la creación del modelo de simulación por defecto se guarda la fecha del sistema.

Los registros de las demás tablas estarán relacionadas con la tabla *simulaciones,* ya que todos los parámetros que se definan en el modelo guardan relación con el modelo de simulación al que pertenecen.

Tabla: *dias*

Tabla 3
Campos de la tabla *días*

Nombre	Clave	Tipo de Dato	Longitud
Id	primaria	numeric	9
descripcion	-	varchar	20

id : código secuencial, identificador único de cada registro.

descripcion: nombre del día

La tabla *dias* almacena los días de la semana por código, de esta forma se hace más rápida la búsqueda de los registros que pertenecen a un determinado día, además la tabla sirve como referencia para las claves foráneas de las demás tablas. Los datos que almacenará la tabla son los siguientes.

Tabla 4
Elementos de la tabla *días*

id	descripción
1	lunes

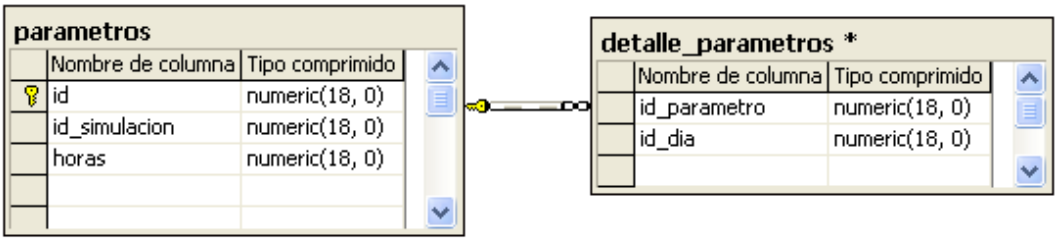
2	martes
3	miércoles
4	jueves
5	viernes
6	sábado
7	domingo

Relación Maestra – Detalle

Tablas: *parametros* con *detalle_parametros*

Las tablas que se muestran a continuación poseen la relación *maestra – detalle*, la tabla *parametros* es la tabla maestra, y la tabla *detalle_parametros* es la tabla detalle.

Tabla 5
Relación de la tabla *parametros* con *detalle_parametros*



A continuación se describe la funcionalidad de cada tabla.

Tabla: *parametros*

La tabla *parametros* almacena el número de horas de operación de un determinado modelo de ruta, por ejemplo, si se ingresa el valor de 6 la ruta operará desde las 5:00 a.m. hasta las 11:00 a.m.

La tabla *parametros* trabaja en conjunto con la tabla *detalle_parametros*, la relación que existe entre ambas tablas es de uno a muchos, un registro de la tabla *parametros* puede tener varios registros en la tabla *detalle_parametros.* Esta relación permite almacenar los días que va a funcionar el modelo de simulación.

Tabla 6
Campos de la tabla *parametros*

Nombre	Clave	Tipo de Dato	Longitud
id	primaria	numeric	9
id_simulacion	secundaria	numeric	9
hora	-	numeric	9

id : código secuencial, identificador único de cada registro.

id_simulacion: clave foránea que hace referencia a la simulación con la que se está trabajando

hora : número de horas que va a funcionar la Troncal 7

Tabla: *detalle_parametros*

En esta tabla se especifica los días que va funcionar la troncal 7, la tabla tiene dos claves foráneas, la primera id_parametro hace referencia a la tabla *parametros* que es su tabla maestra en la cual se especifica las horas que va a funcionar la troncal, la segunda clave foránea id_dia hace referencia a la tabla *dias* la cual contiene los días que va a funcionar la troncal.

Tabla 7
Campos de la tabla *detalle_parametros*

Nombre	Clave	Tipo de Dato	Longitud
id_parametro	secundaria	numeric	9
id_dia	secundaria	numeric	9

id_parametro: clave foránea que hace referencia a la tabla *parametros* la cual contiene el número de horas que va a funcionar la troncal 7.

id_dia : clave foránea que hace referencia al código del día que va a funcionar la troncal 7.

Esta tabla no hace referencia al identificador de la simulación con la que se está trabajando, esto se debe a que como esta tabla esta relacionada a la tabla *parametros* mediante id_parametro, este campo sirve como puente hacia la tabla *parametros* para determinar a que simulación pertenece el registro, con este procedimiento se evita la redundancia de datos.

Tabla: *semaforos*

Esta tabla contiene la descripción de los semáforos que operarán en el modelo de la ruta Troncal 7.

Tabla 8
Campos de la tabla *semáforos*

Nombre	Clave	Tipo de Dato	Longitud

Id	primaria	numeric	9
id_simulacion	secundaria	numeric	9
numero	-	numeric	9
descripción	-	varchar	200

id : código único secuencial

id_simulacion : clave foránea que hace referencia a la simulación con la que se está trabajando

numero : identificador de los semáforos de acuerdo a la simulación con la que se está trabajando.

descripcion : dirección del semáforo

A pesar de que los semáforos están identificados por su clave única id, es importante que el semáforo tenga otro identificador, ya que el campo id identifica a cada semáforo pero por ser este de tipo secuencial generará identificadores indistintamente de la simulación en que se trabaja lo que dificultaría la identificación de los semáforos por parte del usuario.

En el campo descripción se establece la dirección en la cual el semáforo se encuentra ubicado.

Tabla: *tipos_paraderos*

Esta tabla servirá para identificar la ubicación del paradero en calles de una vía o doble vía.

Teniendo esta descripción de los paraderos se podrá saber la utilización del espacio físico que comparten los paraderos de ida y regreso.

Tabla 9
Campos de la tabla *tipos_paraderos*

Nombre	Clave	Tipo de Dato	Longitud	Longitud
id	primaria	numeric	9	9
descripcion	-	varchar	200	200

id : código secuencial único que identifica el tipo de ubicación de los paraderos.

descripcion : descripción del tipo de paradero (una via, dos vias)

Tabla: *paraderos*

La tabla paraderos almacenará la descripción de los datos principales que identifica a cada uno de los paraderos que compone la ruta Troncal 7.

Tabla 10
Campos de la tabla *paraderos*

Nombre	Clave	Tipo de Dato	Longitud
id	primaria	numeric	9
id_simulacion	secundaria	numeric	9
id_tipo	secundaria	numeric	9
numero	-	numeric	9
capacidad	-	numeric	9
direccion	-	varchar	200
terminal	-	numeric	9

id : código secuencial único del paradero

id_simulacion : identificador de la simulación con la que se está trabajando

id_tipo : ubicación del paradero en calles de una vía o doble vía

numero : código que identifica a cada paradero de acuerdo al modelo de simulación con el que se está trabajando.

capacidad : número de personas que puede albergar el paradero

direccion : descripción de la ubicación del paradero

terminal : describe si el paradero funciona como terminal de transferencia

La tabla *paraderos* está relacionada con el modelo o diseño a simular por medio del campo id_simulacion, también guarda relación con la tabla *tipos_paraderos* la misma que indica si el paradero se encuentra en calles de "una via" o "dos vias".

A pesar de que los paraderos están identificados por su clave única id, es importante que el paradero tenga otro identificador, ya que el campo id identifica al paradero pero por ser este de tipo secuencial generará identificadores indistintamente de la simulación en que se trabaja lo que dificultaría la identificación de los paraderos por parte del usuario.

El campo capacidad guardará la capacidad máxima que tiene el paradero para recibir a las personas. Este campo es puramente informativo ya que no afecta el modelo de simulación. La M.I. Municipalidad de Guayaquil no ha realizado estudios para determinar la capacidad de los paraderos, es por ello que en este campo se le asignará la capacidad de 100. La dirección en que se encuentra ubicado el paradero se lo especifica en el campo dirección, y el campo terminal es un campo “booleano” (verdadero o falso), que nos indicará si el terminal funcionará como terminal de transferencia para otras rutas; este campo también es informativo y no afectará el modelo de simulación ya que en el simulador actual se pretende analizar únicamente el comportamiento de la ruta troncal 7 independientemente del funcionamiento de las demás rutas.

Tabla: *secuencia_objetos*

Esta tabla almacena la secuencia de los paraderos, semáforos y terminal de integración que componen la ruta 7. Además se especificará la distancia que existe entre un objeto y otro, así también se indicará el tiempo medio y la desviación en segundos que tarda un bus troncal en movilizarse de un objeto a otro.

Tabla 11
Campos de la tabla *secuencia_objetos*

Nombre	Clave	Tipo de Dato	Longitud

id_simulacion	secundaria	numeric	9
id_objeto	secundaria	numeric	9
tipo_objeto	-	varchar	50
orden	-	numeric	9
distancia	-	decimal	9
tiempo	-	decimal	9
desviacion	-	decimal	9

id_simulacion : identificador de la simulación con la que se está trabajando

id_objeto : código de objeto

tipo_objeto : almacena el tipo de objeto paradero o semáforo (“S” o “P”) respectivamente

orden : identificador del orden que le toca al paradero o semáforo en la ruta

distancia : distancia en mts

tiempo : tiempo medio que tarda el bus troncal en llegar al objeto de destino teniendo como partida el objeto anterior.

desviacion : desviación del tiempo que toma el bus en llegar al objeto de destino teniendo como partida el objeto anterior.

Los registros en esta tabla dependen de la simulación con la que se está trabajando por medio del campo id_simulación. El campo id_objeto indica el código del objeto siendo el objeto un paradero o un semáforo; debemos tener en cuenta que debido a que la tabla

paraderos y la tabla semáforos son dos tablas independientes, puede darse el caso de que dos objetos de diferente tipo tengan el mismo ID, es por esto que para evitar la inconsistencia se utiliza el campo tipo_objeto que indicará si se trata de un paradero o de un semáforo.

Por medio del campo orden se especifica la secuencia que tienen los objetos que conforma la ruta Troncal 7.

Tabla: *tiempos_semaforos*

La tabla tiempos semáforos almacena los tiempos de permanencia del semáforo estado rojo y verde, el tiempo del semáforo en estado amarillo se lo suma al tiempo del semáforo en estado rojo.

Tabla 12
Campos de la tabla *tiempos_semaforos*

Nombre	Clave	Tipo de Dato	Longitud
id_semaforo	secundaria	numeric	9
tiempo_verde	-	numeric	9
tiempo_rojo	-	numeric	9
retraso	-	numeric	9

id_semaforo : código del semáforo con el que se va a trabajar

tiempo_verde : tiempo en segundos de permanencia del semáforo en estado verde

tiempo_rojo : tiempo en segundos de permanencia del semáforo en estado rojo

retraso : tiempo de retraso para iniciar la operación en estado verde

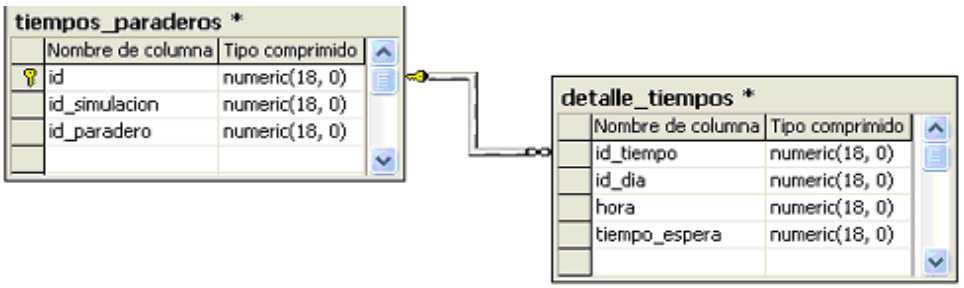
Se utiliza el campo retraso para sincronizar el inicio de operación de los semáforos, como sucede en el sistema real, cuando los semáforos empiezan a trabajar no todos empiezan en un mismo tiempo, este retraso servirá para que el semáforo espere en estado rojo y luego continúe al estado verde. Únicamente esto funciona al inicio de la simulación ya que luego los semáforos adoptan el tiempo en rojo correspondiente.

Relación Maestra – Detalle

Tablas: *tiempos_paraderos* con *detalle_tiempos*

Las tablas que se muestran a continuación poseen la relación *maestra – detalle*, la tabla *tiempos_paraderos* es la tabla maestra, y la tabla *detalle_tiempos* es la tabla detalle.

Tabla 13
Relación de la tabla *tiempos_paraderos* con *detalle_tiempos*



La información que almacenarán estas tablas se detallan a continuación:

Tabla: *tiempos_paraderos*

Esta tabla servirá para almacenar los datos que identificarán la simulación con la que se está trabajando y los paraderos a los cuales se les ingresará la información referente al tiempo que los buses troncales deben permanecer en cada paradero.

Tabla 14
Campos de la tabla *tiempos_paraderos*

Nombre	Clave	Tipo de Dato	Longitud
id	primaria	Numeric	9
id_simulacion	secundaria	Numeric	9
id_paradero	secundaria	Numeric	9

id : código secuencial identificador de la tabla maestra

id_simulacion : referencia a la simulación con la que se esta bajando

id_paradero : paradero al cual se le van asignar los datos

El enlace entre la tabla maestra y la tabla detalle es a través del campo id de la tabla tiempos_paraderos.

El campo id_simulacion servirá para reconocer la simulación con la que se está trabajando.

Tabla: *detalle_tiempos*

La tabla *detalle_tiempos* servirá para almacenar el tiempo que los buses troncales permanecerán en los paraderos de acuerdo al día y la hora en que esta operando la troncal.

Tabla 15
Campos de la tabla *detalle_tiempos*

Nombre	Clave	Tipo de Dato	Longitud
id_tiempo	Secundaria	numeric	9
id_dia	Secundaria	numeric	9
hora	-	numeric	9
tiempo_espera	-	numeric	9

id_tiempo : código que hace referencia a la tabla maestra

id_dia : código que identifica el día

hora : hora a la que se le va asignar el tiempo

tiempo_espera : tiempo de permanencia del bus en el paradero (segundos)

El campo id_tiempo hace referencia al id de la tabla tiempos_paraderos que sirve como enlace entre ambas tablas.

El campo id_dia y hora hace referencia al día y la hora en que se va a guardar el tiempo de espera del bus en cada paradero, dicho tiempo será almacenado en el campo tiempo_espera.

Tabla: *salidas_buses*

En esta tabla se almacenará los tiempos entre salidas de los buses desde la terminal de integración, este tiempo puede ser ingresado para un día y hora específico.

Tabla 16
Campos de la tabla *salidas_buses*

Nombre	Clave	Tipo de Dato	Longitud
id_simulacion	secundaria	numeric	9

id_dia	secundaria	numeric	9
hora	-	numeric	9
t_entre_salida	-	numeric	9

id_simulacion : código que hace referencia a la simulación activa

id_dia : código del día

hora : hora del modelo

t_entre_salida : tiempo en segundos entre salidas de buses

Al igual que en las tablas anteriores, ésta tabla debe hacer referencia a un determinado modelo de simulación, esto se logra por medio del campo id_simulación.

El campo id_dia y el campo hora identifican el día y la hora a la que se le va asignar el tiempo entre salidas de los buses, el cual va a ser almacenado en el campo t_entre_salida.

Tabla: *politica*

La tabla politica tiene como propósito almacenar las políticas con las cuales el modelo de simulación va a operar, estas políticas se detallan a continuación:

Políticas de Cola de Espera en los Paraderos

Las colas de espera pueden ser de dos comportamientos.

1. Colas Aleatorias, es decir, no existan colas en los paraderos y por lo tanto el orden de llegadas de las personas al paradero es independiente del orden de subidas al bus.

2. Colas FIFO, el orden en que las personas suban al bus dependerá del orden de llegada a los paraderos.

Políticas de Ascenso y Descenso

También se han diseñado dos escenarios con respecto a las subidas y bajadas de las personas del bus.

1. Las subidas y las bajadas de los buses son independientes, existirán 8 puertas por las cuales 4 han sido asignadas para las personas y 4 para las bajadas.
2. Las personas para subir al bus, deben esperar primero que bajen las personas, es decir primero se realizan los descensos y luego los ascensos.

Politicas de Espera de los buses

Se ha diseñado el simulador para 4 diferentes comportamientos de espera de los buses troncales.

1. **Fijo-No puede exceder el tiempo máximo de espera**

   Al bus troncal se le asigna un tiempo de espera en el paradero con esta opción el bus deberá permanecer el tiempo que se le ha asignado, el único caso en que el bus se puede ir antes de que haya finalizado este tiempo de espera es cuando el bus está lleno y por lo tanto ya no tiene capacidad lo cual indica que ya están ocupados los 180 asientos del bus.

2. **Fijo-Sí puede exceder el tiempo máximo de espera**

Con esta opción el bus troncal no se puede ir antes de finalizar el tiempo de espera pero sí se puede ir después del tiempo asignado, esto es en el caso de que en el paradero todavía hay personas y la capacidad del bus todavía está disponible. El único caso en que el bus se puede ir antes del tiempo asignado es cuando el bus ya no tenga capacidad.

3. **Flexible - No puede exceder tiempo Máximo de espera**

   Con esta opción el bus troncal no se podrá ir después del tiempo asignado pero sí se puede ir antes, ya que se puede dar el caso de que ya no hayan personas en el paradero y la espera del bus va a ser innecesaria. También el bus se podrá ir antes de finalizar el tiempo de espera cuando tenga su capacidad ocupada por completo.

4. **Flexible - Sí puede exceder tiempo Máximo de espera**

   Si en el paradero ya no están personas entonces el bus finaliza su tiempo de espera y continúa con su recorrido, además si al finalizar el tiempo de espera todavía hay personas en el paradero el bus espera hasta que la cola de espera quede en cero, también se podrá ir antes el bus cuando la capacidad este ocupada en su totalidad.

Tabla 17
Campos de la tabla *politica*

Nombre	Clave	Tipo de Dato	Longitud

Id	Primaria	numeric	9
id_simulacion	Secundaria	numeric	9
politica_cola	-	numeric	9
politica_asc_desc	-	numeric	9
politica_espera_paradero	-	numeric	9
descripcion	-	varchar	200

id : código secuencial identificador de cada registro

id_simulacion : código que hace referencia a la simulación activa

politica_cola : política de colas de espera en los paraderos

politica_asc_desc : política de ascenso y descenso de pasajeros

politica_espera_paradero : política de espera de buses en paraderos

descripcion : datos adicionales que se deseen ingresar como observaciones

El campo id_simulacion hace referencia al modelo de simulación con que se está trabajando.

El campo politica_cola identifica la política de las colas de espera en los paraderos, la identificación de cada política es la siguiente:

Tabla 18
Elementos del campo *politica_colas*

id	descripción
0	FIFO primero en entrar primero en salir

1	Aleatoria

El campo politica_asc_desc identifica la política de ascensos y descensos de pasajeros en el paradero.

Tabla 19
Elementos del campo *politica_asc_desc*

id	descripción
0	Primero descensos, luego ascensos
1	Ascensos y descensos independientes

El campo política_espera_paradero identifica la política de espera de los buses en el paradero, la identificación de cada política es la siguiente:

Tabla 20
Elementos del campo *política_espera_paradero*

id	descripción
0	Fijo-No puede exceder
1	Fijo-Si puede exceder
2	Flexible-No puede exceder
3	Flexible-Si puede exceder

Tabla: *tipo_distribucion*

La tabla servirá para almacenar los diversos tipos de distribución que se van ha utilizar a lo largo de la simulación. El usuario deberá ingresar distribuciones para los siguientes parámetros:

1. Tiempo entre llegada de pasajeros al paradero (segundos)
2. Tiempo entre llegadas de los buses alimentadores al paradero (segundos)

3. Número de pasajeros que los buses troncales dejan en el paradero
4. Número de pasajeros que el bus alimentador deja en el paradero

Tabla 21
Campos de la tabla *tipo_distribucion*

Nombre	Clave	Tipo de Dato	Longitud
id	primaria	numeric	9
descripcion	-	varchar	200

id : código secuencial que identifica el tipo de distribución

descripcion : nombre del tipo de distribución

La tabla contendrá los siguientes registros:

Tabla 22
Registros de la tabla *tipo_distribucion*

id	descripción
1	Tiempo entre llegada de pasajeros al paradero
2	Tiempo entre llegadas de los buses alimentadores al paradero
3	Número de pasajeros que los buses troncales dejan en el paradero
4	Número de pasajeros que el bus alimentador deja en el paradero

Tabla: *tipo_funcion*

Cuando el usuario utilice funciones empíricas la tabla almacena el tipo de función ya sea discreta o continua para dicha distribución empírica. Cuando el usuario utilice un tipo de distribución diferente a los que se encuentran en la lista de funciones establecidas para operar en el simulador, el usuario deberá especificar una función

empírica de tipo continua e ingresar la mayor cantidad de pares ordenados posibles para así el simulador utilice estos pares para crear una curva que se aproxime a la verdadera función, de esta manera el simulador utilizará esta curva para simular la distribución requerida por el usuario que no se encuentra en la lista que ofrece el diseño del simulador.

Tabla 23
Campos de la tabla *tipo_funcion*

Nombre	Clave	Tipo de Dato	Longitud
Id	primaria	Numeric	9
descripción	-	Varchar	200

id : código secuencial

descripcion : nombre de la función

La tabla contendrá los siguientes registros.

Tabla 24
Registros de la tabla *tipo_funcion*

id	descripción
1	Discreta
2	Contínua

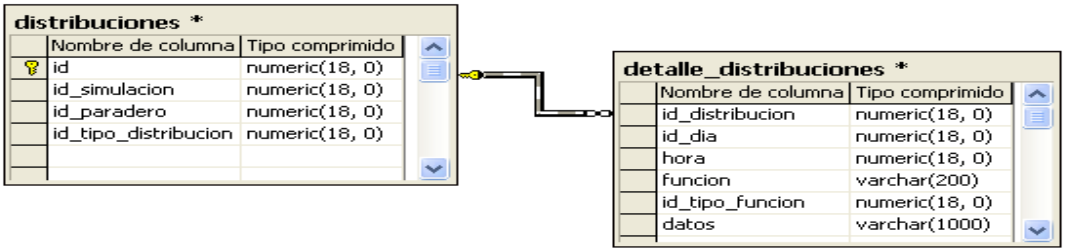
Relación Maestra – Detalle

Tablas: *distribuciones* con *detalle_ distribuciones*

La tabla maestra es la tabla *distribuciones* y la tabla detalle es la tabla *detalle_distribuciones*. La funcionalidad básica de este conjunto de tablas es la de almacenar los datos referentes a los distintos tipos

de distribuciones de probabilidad relacionadas con cada paradero para un determinado día y hora específico.

Tabla 25
Relación de la tabla *distribuciones* con *detalle_distribuciones*



La descripción de cada tabla se muestra a continuación:

Tabla: *distribuciones*

La tabla *distribuciones* contendrá los datos relacionados con la simulación que se está trabajando, el paradero al cual se van a especificar una distribución de probabilidad específica y el tipo de distribución la que puede ser el tiempo entre llegada de las personas al paradero ó el tiempo entre llegada de los buses troncales etc. La tabla *detalle_distribuciones* almacenará el dia, la hora, los datos y los demás parámetros respectivos para este paradero en particular.

Tabla 26
Campos de la tabla *distribuciones*

Nombre	Clave	Tipo de Dato	Longitud
Id	primaria	numeric	9
id_simulacion	secundaria	numeric	9
id_paradero	secundaria	numeric	9
id_tipo_distribucion	secundaria	numeric	9

id : código secuencial identificador de cada registro

id_simulacion : código que hace referencia a la simulación activa

id_paradero : código que hace referencia al paradero al cual se ingresarán los datos

id_tipo_distribucion : código que identifica el tipo de distribución al cual pertenecerán la distribución que se ingresará.

El campo id_simulacion el cual esta relacionado con la tabla simulaciones hace referencia a la simulación con la que se está trabajando, el campo id_paradero identificará el paradero al que le corresponde la distribución ingresada; finalmente el campo id_tipo_distribución identificará cuál de los cuatro tipos de distribuciones mencionados anteriormente es el que vamos a ingresar.

Tabla: *detalle_distribuciones*

Esta tabla almacenar información referente al días, la hora, el tipo de distribución, el tipo de función en el caso de que sea empírica y los datos correspondientes a la distribución de probabilidad.

Tabla 27
Campos de la tabla *detalle_distribuciones*

Nombre	Clave	Tipo de Dato	Longitud
id_distribucion	secundaria	numeric	9
id_dia	secundaria	numeric	9
id_tipo_funcion	secundaria	numeric	9
hora	-	numeric	9
funcion	-	varchar	200
datos	-	varchar	1000

id_distribucion : código que identifica el tipo de distribución a utilizar

id_dia : código que identifica el día al cual se ingresará los datos

id_tipo_funcion : código identificador del tipo de función continua o discreta, en el caso de que se utilice funciones empíricas

hora : hora a la cual pertenecen los datos ingresados

funcion : datos correspondientes a la función empírica

datos : datos correspondientes a las funciones que no son empíricas y que se encuentran establecidas en una lista prediseñada

Todos estos datos deben ser ingresados para un paradero específico, y para ello se utiliza la clave foránea id_distribución que mediante su relación con la tabla maestra *distribuciones* se establece el paradero en el que se está trabajando.

Los campos id_dia y hora, identifican el día y la hora a la que la distribución de probabilidad pertenece.

El campo funcion identifica el tipo de distribución de probabilidad que va a ser asignada a ese paradero en un día y hora especifico, este campo puede tener los siguientes valores:

Tabla 28
Registros del campo *funcion*

id	descripción
1	Constante
2	Uniforme Discreta
3	Uniforme Continua
4	Exponencial
5	Poisson

El campo id_tipo_funcion solamente será llenado en caso de que el campo funcion tenga el valor de “Empírica”, en este campo se ingresará los valores de acuerdo a la tabla tipo_función la cual identifica a 1 (Discreta) y 2 (Continua).

Los parámetros del tipo de distribución escogida (Exponencial, normal, empirica, etc.), serán almacenados en el campo datos; cabe recalcar que debido a que las diferentes distribuciones mencionadas requieren de parámetros diferentes, la forma en que el campo datos almacenará estos datos variará de acuerdo al tipo de distribución escogida, de la siguiente manera:

Tabla 29
Parámetros de la Distribución de Probabilidad

Distribución de Probabilidad	Parámetros
Constante	Media
Uniforme Discreta	Media , Desviación
Uniforme Continua	Media , Desviación
Exponencial	Media
Poisson	Media
Normal	Media , Desviación

Los pares ordenados de la función empírica que el usuario haya ingresado se almacenan separados por una "/", mientras que los componentes "x" y "y" que componen el par ordenado son separados por una ","

A continuación se describe la forma en que se almacenan los pares ordenados:

Empírica x1,y1/x2,y2/x3,y3/…../xn,yn

Diagrama Entidad-Relación

En el siguiente diagrama se podrá apreciar todas las tablas unidas unas con otras de forma relacional.

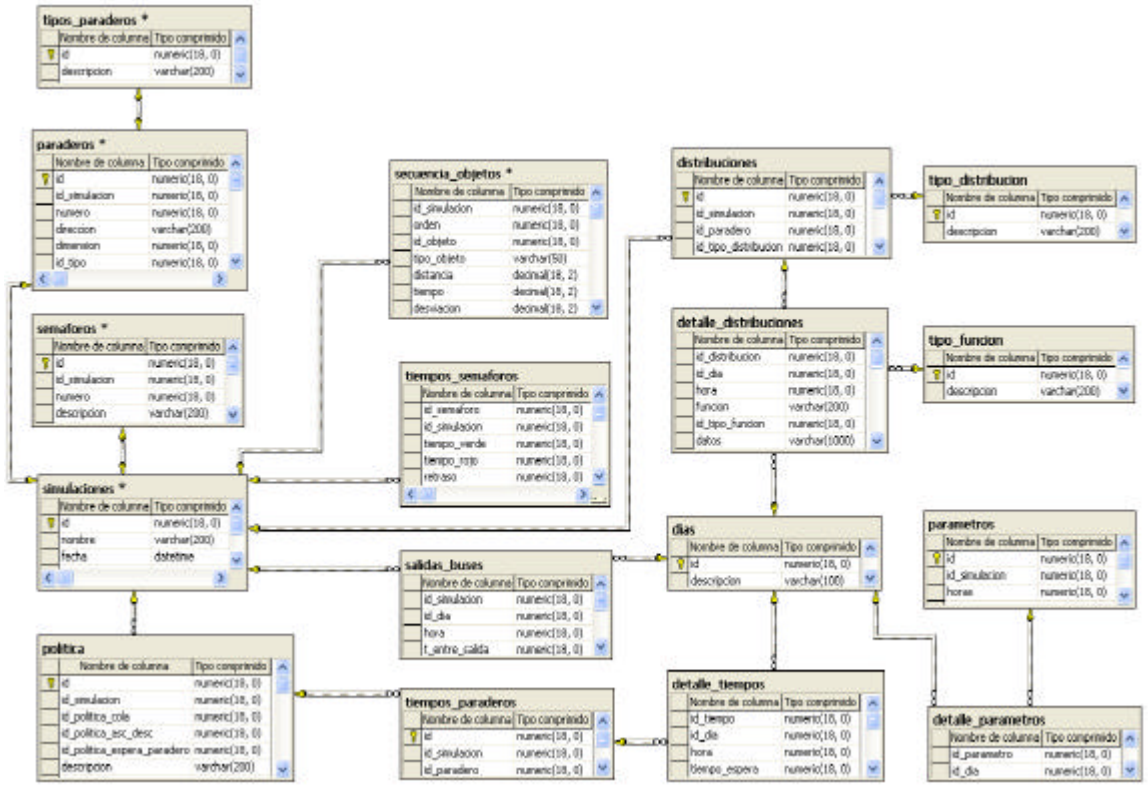



Gráfico 2.2.
Diagrama Entidad-Relación