Iteración 2

Juan Cañizarez, Daniel del Castillo A.

ISIS2304 – Sistemas Transaccionales

Universidad de los Andes, Bogotá, Colombia

{je.canizarez, d.delcastillo}@uniandes.edu.co

Fecha de presentación: 3 de noviembre de 2019

Tabla de contenido

Contents

Análisis y cambios

A continuación, se presenta el análisis de los requerimientos transaccionales de la aplicación y los cambios que se efectuaron sobre esta con respecto a la primera iteración.

Análisis

Para cumplir con las características ACID (atomicidad, consistencia, aislamiento y durabilidad) del sistema transaccional, se implementó el mayor nivel de aislamiento posible. Cada transacción termina en commit o rollback, y así, se garantiza la atomicidad. La consistencia de mantiene debido a las fuertes restricciones únicas y de llaves foráneas que no permiten la introducción de datos corruptos o modificaciones directas de los identificadores sobre las tablas (dado que se generan automáticamente por el sistema, y no se introducen por el usuario o administrador). El aislamiento se consigue por medio del nivel de aislamiento predeterminado por el SMBD, que para las necesidades funcionales del proyecto, es suficiente, dado que la gran mayoría de información no requiere de actualización (e.g., cambios de nombres, IPS, servicios, entre otros) y si sí, se ejercen por el mismo administrador. Finalmente, la durabilidad se garantiza por medio del diseño relacional de datos (dado que en ningún momento se es necesario eliminar una tupla). Este conjunto de decisiones garantiza la transaccionalidad del sistema.

Cambios

Se cambió la tabla de servicios para que tuviera un estado y además registrara la identificación de la IPS para poder diferenciar capacidades y horario. Esto, ya que con la tabla intermedia de prestanServicio la consulta resultaba bastante costosa. El estado se incluyó para poder satisfacer los nuevos requerimientos funcionales relacionados con la introducción a la creación de campañas.

Se incluyó una interfaz gráfica en la aplicación para simplificar la introducción de información, así como para visualizar de manera más organizada la información. El diseño es básico y enfocado hacia la funcionalidad, más que el diseño; sin embargo, se planea para el futuro volver la aplicación atractiva para los usuarios que la manejen y personalizada dependiendo del rol del usuario que la acceda.

Lógica de nuevos requerimientos

Para todo el manejo de la creación de una campaña, se crearon dos nuevas tablas: campaña y serviciosCampana. La primera se encarga de guardar un identificador asignado por el sistema, un nombre y el identificador del organizador. Como las campañas son únicas, se colocó esta restricción sobre el nombre de la campaña. Finalmente, se colocó una restricción de llave foránea para indicar el afiliado organizador de la campaña. Así, la sentencia encargada de genera esta tabla se observa en la Figura 1.

```
-- Tabla de campaña:
CREATE TABLE Campana(
Id NUMBER GENERATED ALWAYS AS IDENTITY,
Nombre VARCHAR2(255 BYTE) NOT NULL,
IdOrganizador NUMBER,
CONSTRAINT CMP_Unique UNIQUE (Nombre),
CONSTRAINT CMP_PK PRIMARY KEY (Id),
CONSTRAINT CMP_IdOrganizador_FK FOREIGN KEY (IdOrganizador) REFERENCES Afiliado(documento));
```

Figura 1 sentencia encargada de generar la tabla Campana

Para el manejo de los servicios que presta la campaña, se creó la tabla intermedia serviciosCampana encargada de relacionar un servicio prestado por una IPS con una campaña. Así, tiene dos atributos, el identificador del tipo de servicio que se encuentra en la campaña, y el identificador de la campaña como tal, ambas con restricciones de llave foránea. Así, la sentencia encargada de generar esta tabla de observa en la Figura 2.

```
-- Tabla de servicios de campaña:
CREATE TABLE ServiciosCampana(
idTipoServicio NUMBER,
idCampana NUMBER,
CantidadReservada NUMBER NOT NULL,
CONSTRAINT ServiciosCampana_PK PRIMARY KEY (idTipoServicio, idCampana),
CONSTRAINT CampanaServicio_FK FOREIGN KEY (idTipoServicio) REFERENCES TipoServicio(Id),
CONSTRAINT CampanaId_FK FOREIGN KEY (idCampana) REFERENCES Campana(Id));
```

Figura 2 sentencia encargada de genera la tabla serviciosCampana

Con la introducción de estas dos tablas se ejercen las consultas con respecto a los requerimientos de consulta. Así, se procede a definir el proceso lógico para estas:

- *Requerimiento de consulta RFC6*: se obtiene la capacidad de los servicios prestados por la IPS y, por medio de la información almacenada en la tabla de servicios, se mira cuál es el índice de uso como porcentaje a través de la división del uso sobre la capacidad.
- *Requerimiento de consulta RFC7*: por medio de la tabla de citas y servicios se hace uso del producto cartesiano para obtener los usuarios que hayan recibido más de 12 de servicios de salud de 3 tipos de servicios diferentes (se diferencian los tipos de servicios por medio del identificador único que se almacena en la tabla de servicios, correspondiente a la tabla de TipoServicio).
- *Requerimiento de consulta RFC8*: por medio de la tabla de citas y servicios se hace uso del producto cartesiano para obtener los servicios que hayan sido solicitado menos de 3 veces a la semana. Esto, por medio de un contador del identificador del servicio en tabla de citas que permite saber su uso y la información relacionada a la fecha.