

“Manejo transaccional de información”

Juan Sebastián Díaz Serrano, Sergio Guzmán Mayorga

Iteración 3

Universidad de los Andes, Bogotá, Colombia

{js.diaz, s.guzmanm}@uniandes.edu.co

Fecha de presentación: Octubre 29 de 2017

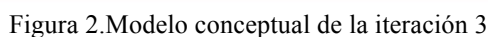
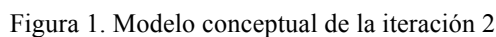
Tabla de contenido

1	Introducción	1
2	Análisis	2
3	Diseño de la aplicación	3
3.1	Impacto de nuevos requerimientos	3
3.2	Modelo relacional	3
3.3	Documentación de cada tabla seguida por de su forma normal	4
3.4	Cambios sustanciales de la Iteración 2 a la Iteración 3	18
3.5	Documentación de la lógica de requerimientos nuevos y transaccionales	18
4	Construcción de la aplicación	22
4.1	RF9	22
4.2	RF10	22
4.3	RF11	22
4.4	RF12	22
4.5	RF13	22
4.6	RF14	22
4.7	RF15	22
4.8	RF16	23
4.9	RF17	23
4.10	RFC7	23
4.11	RFC8	23
4.12	RNF3	23
5	Consideraciones adicionales	24
6	Análisis de resultados	24
6.1	Aprendizajes y logros	24
6.2	Conclusiones	24
7	Bibliografía	24

1 Introducción

En el presente documento se presentan los puntos clave para el desarrollo de las actividades planteadas en [1]. Para cumplir con esto, cada sección del presente texto hará referencia a las tareas a cumplir en el documento referenciado, a algunos puntos que pensamos que vale la pena aclarar respecto al uso de requerimientos, o a conclusiones y observaciones que se hayan presentado a lo largo de la iteración. Como se sabe que no se va a ser posible ver todos los

2 Análisis



3 Diseño de la aplicación

3.1 Impacto de nuevos requerimientos

Dentro del impacto total de los requerimientos sobre nuestro antiguo modelo se encuentran:

- La necesidad de tener mesas como entidades registradas en cada zona.
- Proveer que cualquier pedido pueda tener sustituciones anexas, que deben persistirse en una base de datos.
- Identificar cuando una cuenta está pagada o no, de forma más eficiente que el hecho de revisar si todos sus productos se entregaron o no.
- La inclusión de que una cuenta puede o no estar asociada a una mesa.

3.2 Modelo relacional

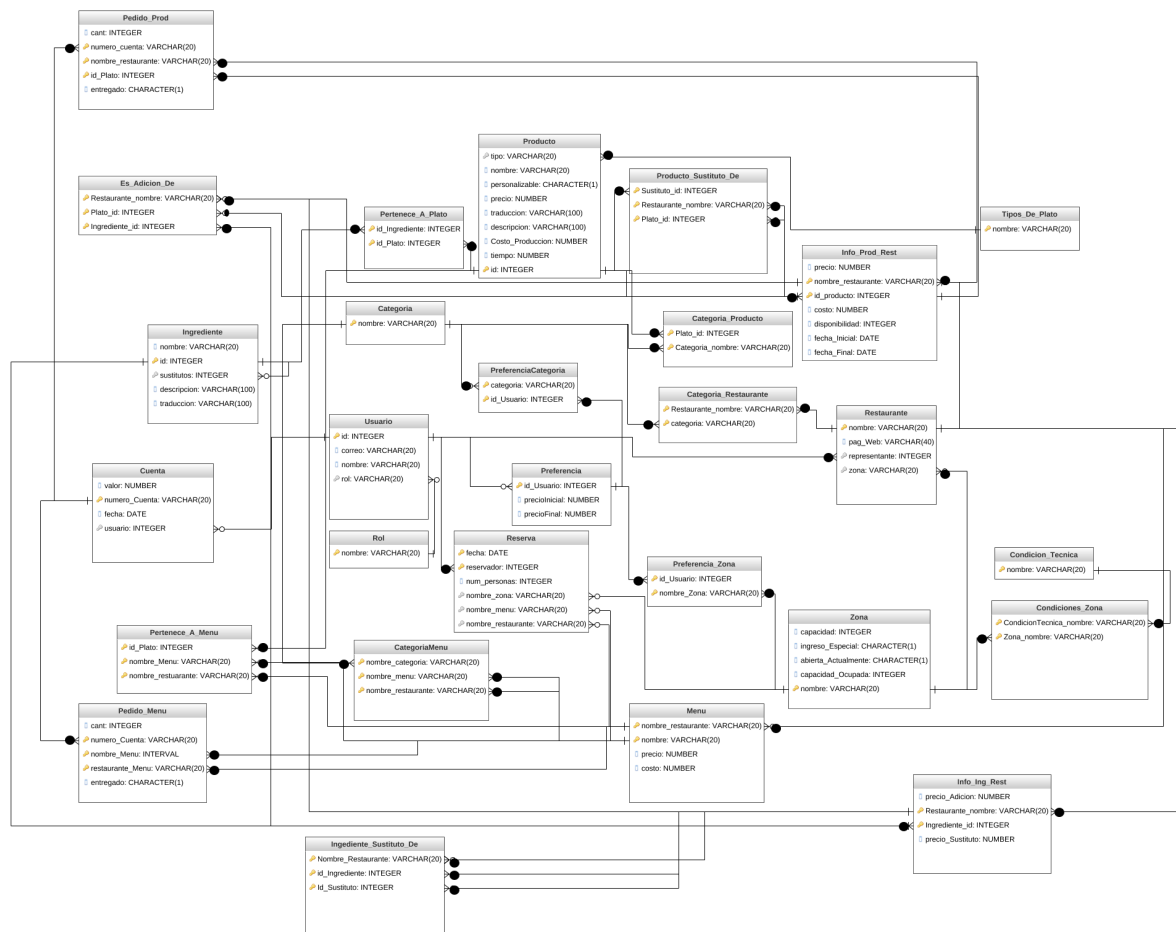


Figura 4. Modelo lógico de la iteración 2

NOMBRE	VARCHAR(20)	Nombre del tipo	PK, CK
--------	-------------	-----------------	--------

Figura 7. Tabla de tipos de plato

- DEPENDENCIAS: {}
- LLAVES: {NOMBRE}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al solo haber un atributo
- 3NF: No hay dependencias transitivas entre no primos al solo haber un atributo.
- BCNF: No hay dependencias no triviales entre primos al solo haber un atributo.

Nombre	INGREDIENTE		
Descripción	Tabla que representa los ingredientes del sistema		
Llave	PK (ID)		
Atributo	Tipo	Descripción	Restricción
NOMBRE	VARCHAR(20)	Nombre del ingrediente	CK
ID	INTEGER	Id del ingrediente	PK
DESCRIPCION	VARCHAR(20)	Descripción del ingrediente	CK
TRADUCCION	VARCHAR(20)	Traducción del ingrediente	CK

Figura 8. Tabla de ingrediente

- DEPENDENCIAS: {ID->(NOMBRE,DESCRIPCION,TRADUCCION)}
- LLAVES: {ID}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al solo haber una dependencia.
- 3NF: No hay dependencias transitivas entre no primos al solo haber una dependencia.
- BCNF: No hay dependencias no triviales entre primos al solo existir un primo.

Nombre	PRODUCTO		
Descripción	Tabla que representa los productos del sistema		
Llave	PK (ID), FK(TIPO _{TIPOSDPLATO.NOMBRE})		
Atributo	Tipo	Descripción	Restricción
NOMBRE	VARCHAR(50)	Nombre del producto	CK
ID	INTEGER	Id del producto	PK
DESCRIPCION	VARCHAR(50)	Descripción del producto	CK
TRADUCCION	VARCHAR(50)	Traducción del producto	CK
TIPO	VARCHAR(20)	Tipo del producto	FK,CK
PERSONALIZABLE	CHAR(1)	Si se puede personalizar el producto	CK

Figura 9. Tabla de producto

- DEPENDENCIAS: {ID->(NOMBRE,DESCRIPCION,TRADUCCION, TIPO, PERSONALIZABLE)}
- LLAVES: {ID}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al solo haber una dependencia.
- 3NF: No hay dependencias transitivas entre no primos al solo haber una dependencia.
- BCNF: No hay dependencias no triviales entre primos al solo existir un primo.

Nombre	CATEGORIA		
Descripción	Tabla que representa las categorías del sistema		
Llave	PK(NOMBRE)		
Atributo	Tipo	Descripción	Restricción
NOMBRE	VARCHAR(20)	Nombre de la categoría	CK

Figura 10. Tabla de Categoría

- DEPENDENCIAS: {}
- LLAVES: {NOMBRE}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al solo haber un atributo
- 3NF: No hay dependencias transitivas entre no primos al solo haber un atributo.
- BCNF: No hay dependencias no triviales entre primos al solo haber un atributo.

Nombre	CONDICIONTECNICA		
Descripción	Tabla que representa las condiciones técnicas del sistema		
Llave	PK (NOMBRE)		
Atributo	Tipo	Descripción	Restricción
NOMBRE	VARCHAR(20)	Nombre de la condicion	CK

Figura 11. Tabla de Condición Técnica

- DEPENDENCIAS: {}
- LLAVES: {NOMBRE}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al solo haber un atributo
- 3NF: No hay dependencias transitivas entre no primos al solo haber un atributo.
- BCNF: No hay dependencias no triviales entre primos al solo haber un atributo.

Nombre	PERTENECE_A_PLATO		
Descripción	Tabla que representa los productos del sistema en relación con sus ingredientes		
Llave	PK (ID_PLATO, ID_INGREDIENTE), FK(ID_PLATO_PRODUCTO.ID, ID:INGREDIENTE INGREDIENTE.ID)		
Atributo	Tipo	Descripción	Restricción
ID_PLATO	INTEGER	Id del plato	PK,FK
ID_INGREDIENTE	INTEGER	Id del ingrediente	PK.FK

Figura 12. Tabla de Pertenece_A_Plato

- DEPENDENCIAS: {}
- LLAVES: {(ID_PLATO, ID_INGREDIENTE)}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al no haber dependencias.
- 3NF: No hay dependencias transitivas entre no primos al no haber dependencias.
- BCNF: No hay dependencias no triviales entre primos al no haber dependencias que los conecte directamente.

Nombre	USUARIO		
Descripción	Tabla que representa los usuarios del sistema		
Llave	PK(ID), FK(ROL_ROL.NOMBRE)		
Atributo	Tipo	Descripción	Restricción

NOMBRE	VARCHAR(50)	Nombre del usuario	CK
ID	INTEGER	Id del usuario	PK, CK
CORREO	VARCHAR(100)	Correo del usuario	
ROL	VARCHAR(20)	Rol del usuario	FK, CK

Figura 13. Tabla de Usuario

- DEPENDENCIAS: {ID->(NOMBRE,CORREO,ROL), CORREO->(ID,NOMBRE,ROL)}
- LLAVES: {ID,CORREO}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al tener atributos únicos como determinadores (a la izquierda).
- 3NF: No hay dependencias transitivas entre no primos al no haber dependencias que usen no primos como determinadores (a la izquierda).
- BCNF: Solo hay dependencias triviales entre primos ya que uno determina al otro de forma individual, pero esto no involucra que se sobrepase de esta forma normal.

Nombre	ZONA		
Descripción	Tabla que representa las zonas del sistema		
Llave	PK (NOMBRE)		
Atributo	Tipo	Descripción	Restricción
NOMBRE	VARCHAR(20)	Nombre de la zona	PK
CAPACIDAD	INTEGER	Capacidad de la zona	CK
INGRESOESPECIAL	CHAR(1)	Si es de ingreso especial o no	CK
ABIERTA ACTUALMENTE	CHAR(1)	Si está abierta actualmente o no	CK
CAPACIDAD OCUPADA	INTEGER	Capacidad ocupada de la zona	CK

Figura 14. Tabla de Zona

- DEPENDENCIAS: {NOMBRE->(CAPACIDAD,INGRESOESPECIAL,ABIERTA ACTUALMENTE,CAPACIDAD OCUPADA)}
- LLAVES {NOMBRE}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al solo haber una dependencia.
- 3NF: No hay dependencias transitivas entre no primos al solo haber una dependencia.
- BCNF: No hay dependencias no triviales entre primos al solo haber una dependencia.

Nombre	MESA		
Descripción	Tabla que representa las mesas del sistema		
Llave	PK (ID), FK(ZONA_ZONA.NOMBRE)		
Atributo	Tipo	Descripción	Restricción
ZONA	VARCHAR(20)	Nombre de la zona a la que pertenece	FK,CK
ID	INTEGER	Id de la mesa	PK
CAPACIDAD	INTEGER	Capacidad de la mesa	CK
CAPACIDAD OCUPADA	INTEGER	Capacidad ocupada de la mesa	CK

Figura 15. Tabla de Mesa

- DEPENDENCIAS: {ID->(ZONA,ID,CAPACIDAD,CAPACIDAD OCUPADA)}

- LLAVES{ID}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al solo haber una dependencia.
- 3NF: No hay dependencias transitivas entre no primos al solo haber una dependencia.
- BCNF: No hay dependencias no triviales entre primos al solo haber una dependencia.

Nombre	RESTAURANTE		
Descripción	Tabla que representa los restaurantes del sistema		
Llave	PK(NOMBRE),FK(NOMBRE_ZONA _{ZONA.NOMBRE} , ID_REPRESENTANTE _{USUARIO.ID})		
Atributo	Tipo	Descripción	Restricción
NOMBRE	VARCHAR(20)	Nombre del restaurante	PK
ID_REPRESENTANTE	INTEGER	Id del representante	FK,CK
PAG_WEB	VARCHAR(40)	Página web del restaurante	
NOMBRE_ZONA	VARCHAR(20)	Nombre de la zona a la que pertenece	FK,CK

Figura 16. Tabla de Restaurante

- DEPENDENCIAS: {NOMBRE-> (ID_REPRESENTANTE, PAG_WEB, NOMBRE_ZONA)}
- LLAVES{NOMBRE}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al solo haber una dependencia.
- 3NF: No hay dependencias transitivas entre no primos al solo haber una dependencia.
- BCNF: No hay dependencias no triviales entre primos al solo haber una dependencia.

Nombre	CONDICIONZONA		
Descripción	Tabla que representa la relación de condiciones con zonas.		
Llave	PK(CONDICIONTECNICANOMBRE, ZONANOMBRE), FK(CONDICIONTECNICANOMBRE _{CONDICIONTECNICA.NOMBRE} , ZONANOMBRE _{ZONA.NOMBRE})		
Atributo	Tipo	Descripción	Restricción
CONDICIONTECNICANOMBRE	VARCHAR(20)	Nombre de la condición técnica	PK,FK
ZONANOMBRE	VARCHAR(20)	Nombre de la zona	PK,FK

Figura 17. Tabla de CondicionZona

- DEPENDENCIAS: {}
- LLAVES: {(CONDICIONTECNICANOMBRE, ZONANOMBRE)}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al no haber ni una dependencia.
- 3NF: No hay dependencias transitivas entre no primos al no haber ni una dependencia.
- BCNF: No hay dependencias no triviales entre primos al no haber una dependencia que los conecte directamente.

Nombre	SUSTITUTOS_PRODUCTO		
Descripción	Posibles sustitutos de un producto		
Llave	PK(ID_PRODUCTO, ID_SUSTITUTO, NOMBRE_RESTAURANTE), FK(ID_PRODUCTO INFO_PROD_REST.ID_PRODUCTO, ID_SUSTITUTO INFO_PROD_REST.ID_PRODUCTO, NOMBRE_RESTAURANTE INFO_PROD_REST.NOMBRE_RESTAURANTE)		
Atributo	Tipo	Descripción	Restricción
ID_PRODUCTO	INTEGER	Id del producto	PK, FK, CK
ID_SUSTITUTO	INTEGER	Id del sustituto del producto	PK, FK, CK
NOMBRE_RESTAURANTE	VARCHAR(20)	Nombre del restaurante del sustituto	PK, FK

Figura 18. Tabla de Sustitutos_Producto

- DEPENDENCIAS: {}
- LLAVES: {(ID_PRODUCTO, ID_SUSTITUTO, NOMBRE_RESTAURANTE)}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al no haber ni una dependencia.
- 3NF: No hay dependencias transitivas entre no primos al no haber ni una dependencia.
- BCNF: No hay dependencias no triviales entre primos al no haber ni una dependencia que los conecte directamente.

Nombre	SUSTITUTOS_INGREDIENTE		
Descripción	Describe los sustitutos de ingredientes por restaurante		
Llave	PK(ID_INGREDIENTE, ID_SUSTITUTO, NOMBRE_RESTAURANTE), FK(ID_INGREDIENTE INFO_ING_REST.ID_INGREDIENTE, ID_SUSTITUTO INFO_ING_REST.ID_INGREDIENTE, NOMBRE_RESTAURANTE INFO_ING_REST.NOMBRE_RESTAURANTE)		
Atributo	Tipo	Descripción	Restricción
ID_PRODUCTO	INTEGER	Id del ingrediente	PK, FK, CK
ID_SUSTITUTO	INTEGER	Id del sustituto del ingrediente	PK, FK, CK
NOMBRE_RESTAURANTE	VARCHAR(20)	Nombre del restaurante del sustituto	PK, FK

Figura 19. Tabla de Sustitutos_Ingrediente

- DEPENDENCIAS: {}
- LLAVES: {(ID_PRODUCTO, ID_SUSTITUTO, NOMBRE_RESTAURANTE)}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al no haber ni una dependencia.
- 3NF: No hay dependencias transitivas entre no primos al no haber ni una dependencia.
- BCNF: No hay dependencias no triviales entre primos al no haber ni una dependencia que los conecte directamente.

Nombre	SUSTITUCION_PRODUCTO		
Descripción	Posibles sustitutos de un producto en una orden		
Llave	PK(ID_ORIGINAL, ID_SUSTITUTO, NOMBRE_RESTAURANTE, NUMERO_CUENTA, NOMBRE_MENU), FK(ID_ORIGINAL PRODUCTO.ID, ID_SUSTITUTO PRODUCTO.ID, NOMBRE_RESTAURANTE PEDIDO_MENU.NOMBRE_RESTAURANTE, NOMBRE_MENU PEDIDO_MENU.NOMBRE_MENU, NUMERO_CUENTA PEDIDO_MENU.NUMERO_CUENTA)		
Atributo	Tipo	Descripción	Restricción

ID_ORIGINAL	INTEGER	Id del producto	PK,FK
ID_SUSTITUTO	INTEGER	Id del sustituto del producto	PK,FK
NUMERO_CUENTA	VARCHAR(20)	Número de la cuenta	PK,FK
NOMBRE_MENU	VARCHAR(20)	Nombre del menu donde se sustituye	PK,FK
NOMBRE_RESTAURANTE	VARCHAR(20)	Nombre del restaurante	PK,FK

Figura 20. Tabla de Sustitucion_Producto

- DEPENDENCIAS: {}
- LLAVES: {(ID_ORIGINAL, ID_SUSTITUTO, NUMERO_CUENTA, NOMBRE_MENU, NOMBRE_RESTAURANTE)}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al no haber ni una dependencia.
- 3NF: No hay dependencias transitivas entre no primos al no haber ni una dependencia.
- BCNF: No hay dependencias no triviales entre primos al no haber ni una dependencia que los conecte directamente.

Nombre	SUSTITUCION_INGREDIENTE		
Descripción	Posibles sustitutos de un producto en una orden		
Llave	PK(ID_ORIGINAL, ID_SUSTITUTO, NOMBRE_RESTAURANTE, NUMERO_CUENTA, ID_PRODUCTO), FK(ID_ORIGINAL_PRODUCTO.ID, ID_SUSTITUTO_PRODUCTO.ID, NOMBRE_RESTAURANTE_PEDIDO_PROD.NOMBRE_RESTAURANTE, ID_PRODUCTO_PEDIDO_PROD.ID_PRODUCTO, NUMERO_CUENTA_PEDIDO_PROD.NUMERO_CUENTA)		
Atributo	Tipo	Descripción	Restricción
ID_ORIGINAL	INTEGER	Id del ingrediente	PK,FK
ID_SUSTITUTO	INTEGER	Id del sustituto del ingrediente	PK,FK
NUMERO_CUENTA	VARCHAR(20)	Número de la cuenta	PK,FK
ID_PRODUCTO	INTEGER	Id del producto de los ingredientes	PK,FK
NOMBRE_RESTAURANTE	VARCHAR(20)	Nombre del restaurante	PK,FK

Figura 21. Tabla de Sustitucion_Ingrediente

- DEPENDENCIAS: {}
- LLAVES: {(ID_PRODUCTO, ID_ORIGINAL, ID_SUSTITUTO, NUMERO_CUENTA, NOMBRE_RESTAURANTE)}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al no haber ni una dependencia.
- 3NF: No hay dependencias transitivas entre no primos al no haber ni una dependencia.
- BCNF: No hay dependencias no triviales entre primos al no haber ni una dependencia que los conecte directamente.

Nombre	SUSTITUCION_INGREDIENTE_EN_PRODUCTO		
Descripción	Posibles sustitutos de un producto en una orden		
Llave	PK(ID_ORIGINAL, ID_SUSTITUTO, NOMBRE_RESTAURANTE, NUMERO_CUENTA, NOMBRE_MENU, ID_PRODUCTO), FK(ID_ORIGINAL_INGREDIENTE.ID, ID_SUSTITUTO_INGREDIENTE.ID, NOMBRE_RESTAURANTE_PEDIDO_MENU.NOMBRE_RESTAURANTE, NOMBRE_MENU_PEDIDO_MENU.NOMBRE_MENU, NUMERO_CUENTA_PEDIDO_MENU.NUMERO_CUENTA, ID_PRODUCTO_PRODUCTO.ID)		
Atributo	Tipo	Descripción	Restricción

ID_ORIGINAL	INTEGER	Id del ingrediente	PK,FK
ID_SUSTITUTO	INTEGER	Id del sustituto del ingrediente	PK,FK
NUMERO_CUENTA	VARCHAR(20)	Número de la cuenta	PK,FK
NOMBRE_MENU	VARCHAR(20)	Nombre del menu donde se sustituye	PK,FK
NOMBRE_RESTAURANTE	VARCHAR(20)	Nombre del restaurante	PK,FK
ID_PRODUCTO	INTEGER	Id del producto al que hacen referencia	PK,FK

Figura 22. Tabla de Sustitucion_Ingrediente_En_Producto

- LLAVES: {(ID_PRODUCTO, ID_ORIGINAL, ID_SUSTITUTO, NUMERO_CUENTA, NOMBRE_RESTAURANTE, NOMBRE_MENU)}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al no haber ni una dependencia.
- 3NF: No hay dependencias transitivas entre no primos al no haber ni una dependencia.
- BCNF: No hay dependencias no triviales entre primos al no haber ni una dependencia que los conecte directamente.

Nombre	PREFERENCIA		
Descripción	Preferencias de clientes		
Llave	PK(IDUSUARIO), FK(IDUSUARIO_USUARIO.ID)		
Atributo	Tipo	Descripción	Restricción
IDUSUARIO	INTEGER	Id del usuario	PK,FK
PRECIOINICIAL	INTEGER	Precio inicial	CK
PRECIOFINAL	VARCHAR(20)	Precio final	CK

Figura 23. Tabla de Preferencia

- DEPENDENCIAS: {IDUSUARIO-> (PRECIOINICIAL,PRECIOFINAL)}
- LLAVES {IDUSUARIO}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al solo haber una dependencia.
- 3NF: No hay dependencias transitivas entre no primos al solo haber una dependencia.
- BCNF: No hay dependencias no triviales entre primos al solo haber una dependencia.

Nombre	INFO_PROD_REST		
Descripción	Información de un producto de un restaurante		
Llave	PK(ID_PRODUCTO, NOMBRE_RESTAURANTE) FK(ID_PRODUCTO_PRODUCTO.ID, NOMBRE_RESTAURANTE_RESTAURANTE.NOMBRE)		
Atributo	Tipo	Descripción	Restricción
ID_PRODUCTO	INTEGER	Id del producto	PK,FK
NOMBRE_RESTAURANTE	VARCHAR(20)	Nombre del restaurante	PK,FK
PRECIO	NUMBER	Precio	CK
COSTO	NUMBER	Costo	CK
DISPONIBILIDAD	NUMBER	Disponibilidad del producto	CK
FECHA_INICIO	DATE	Fecha de inicio de servicio	CK
FECHA_FIN	DATE	Fecha de fin de servicio	CK
CANTIDAD_MAXIMA	NUMBER	Cantidad máxima del producto	CK

Figura 24. Tabla de Info_Prod_Rest

- DEPENDENCIAS: {(ID_PRODUCTO,NOMBRE_RESTAURANTE)->(PRECIO,COSTO,DISPONIBILIDAD,FECHA_INICIO,FECHA_FIN,CANTIDAD_MAXIMA)}
- LLAVES {(ID_PRODUCTO,NOMBRE_RESTAURANTE)}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al solo haber una dependencia.
- 3NF: No hay dependencias transitivas entre no primos al solo haber una dependencia.
- BCNF: No hay dependencias no triviales entre primos al solo haber una dependencia.

Nombre	INFO_ING_REST		
Descripción	Información de los ingredientes de un restaurante		
Llave	PK(ID_INGREDIENTE,NOMBRE_RESTAURANTE), FK(ID_INGREDIENTE_INGREDIENTE.ID, NOMBRE_RESTAURANTE RESTAURANTE.NOMBRE)		
Atributo	Tipo	Descripción	Restricción
ID_INGREDIENTE	INTEGER	Id del ingrediente	PK,FK
NOMBRE_RESTAURANTE	VARCHAR(20)	Nombre del restaurante	PK,FK
PRECIO_ADICION	NUMBER	Precio si es adición	
PRECIO_SUSTITUTO	NUMBER	Precio si es sustituto	

Figura 25. Tabla de Info_Ing_Rest

- DEPENDENCIAS: {(ID_INGREDIENTE,NOMBRE_RESTAURANTE)->(PRECIO_ADICION, PRECIO_SUSTITUTO)}
- LLAVES {(ID_INGREDIENTE,NOMBRE_RESTAURANTE)}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al solo haber una dependencia.
- 3NF: No hay dependencias transitivas entre no primos al solo haber una dependencia.
- BCNF: No hay dependencias no triviales entre primos al solo haber una dependencia.

Nombre	MENU		
Descripción	Información de los menús del sistema		
Llave	PK(NOMBRE, NOMBRE_RESTAURANTE), FK(NOMBRE_RESTAURANTE RESTAURANTE.NOMBRE)		
Atributo	Tipo	Descripción	Restricción
NOMBRE	VARCHAR(20)	Nombre del menú	PK,FK
NOMBRE_RESTAURANTE	VARCHAR(20)	Nombre del restaurante	PK,FK
PRECIO	NUMBER	Precio del menú	CK
COSTO	NUMBER	Costo del menú	

Figura 26. Tabla de Menu

- DEPENDENCIAS: {(NOMBRE,NOMBRE_RESTAURANTE)->(PRECIO,COSTO)}
- LLAVES {(NOMBRE,NOMBRE_RESTAURANTE)}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al solo haber una dependencia.

- 3NF: No hay dependencias transitivas entre no primos al solo haber una dependencia.
- BCNF: No hay dependencias no triviales entre primos al solo haber una dependencia.

Nombre	PERTENECE_A_MENU		
Descripción	Tabla donde se ve qué productos pertenecen a qué menú		
Llave	PK(NOMBRE_MENU, ID_PLATO, NOMBRE_RESTAURANTE), FK(NOMBRE_MENU MENU.NOMBRE, ID_PLATO PRODUCTO.ID, RESTAURANTE MENU.NOMBRE RESTAURANTE)		
Atributo	Tipo	Descripción	Restricción
NOMBRE_MENU	VARCHAR(20)	Nombre del menú	PK,FK
ID_PLATO	INTEGER	Id del plato	PK,FK
NOMBRE_RESTAURANTE	VARCHAR(20)	Nombre del restaurante	PK,FK

Figura 27. Tabla de Pertenece_A_Menu

- DEPENDENCIAS: {}
- LLAVES: {(NOMBRE_MENU, ID_PLATO, NOMBRE_RESTAURANTE)}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al no haber ni una dependencia.
- 3NF: No hay dependencias transitivas entre no primos al no haber ni una dependencia.
- BCNF: No hay dependencias no triviales entre primos al no haber ni una dependencia que los conecte directamente.

Nombre	CATEGORIA_MENU		
Descripción	Información de las categorías de los menús		
Llave	PK(NOMBRE_CATEGORIA, NOMBRE_MENU, NOMBRE_RESTAURANTE), FK(NOMBRE_CATEGORIA CATEGORIA.NOMBRE, NOMBRE_MENU MENU.NOMBRE, NOMBRE_RESTAURANTE MENU.NOMBRE RESTAURANTE)		
Atributo	Tipo	Descripción	Restricción
NOMBRE_CATEGORIA	VARCHAR(50)	Nombre de la categoría	PK,FK
NOMBRE_MENU	VARCHAR(20)	Nombre del menú	PK,FK
NOMBRE_RESTAURANTE	CVARCHAR(20)	Nombre del restaurante	PK,FK

Figura 28. Tabla de Categoria_Menu

- DEPENDENCIAS: {}
- LLAVES: {(NOMBRE_CATEGORIA, NOMBRE_MENU, NOMBRE_RESTAURANTE)}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al no haber ni una dependencia.
- 3NF: No hay dependencias transitivas entre no primos al no haber ni una dependencia.
- BCNF: No hay dependencias no triviales entre primos al no haber ni una dependencia que los conecte directamente.

Nombre	CATEGORIA_RESTAURANTE		
Descripción	Información de las categorías de los restaurantes		

Llave	PK(NOMBRE_CATEGORIA, NOMBRE_RESTAURANTE), FK(NOMBRE_CATEGORIA_CATEGORIA.NOMBRE, NOMBRE_RESTAURANTE_RESTAURANTE.NOMBRE)		
Atributo	Tipo	Descripción	Restricción
NOMBRE_CATEGORIA	VARCHAR(50)	Nombre de la categoría	PK,FK
NOMBRE_RESTAURANTE	CVARCHAR(20)	Nombre del restaurante	PK,FK

Figura 29. Tabla de Categoria_Restaurante

- DEPENDENCIAS: {}
- LLAVES: {(NOMBRE_CATEGORIA, NOMBRE_RESTAURANTE)}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al no haber ni una dependencia.
- 3NF: No hay dependencias transitivas entre no primos al no haber ni una dependencia.
- BCNF: No hay dependencias no triviales entre primos al no haber ni una dependencia que los conecte directamente.

Nombre	CATEGORIA_PRODUCTO		
Descripción	Información de las categorías de los productos		
Llave	PK(NOMBRE_CATEGORIA, ID_PRODUCTO), FK(NOMBRE_CATEGORIA_CATEGORIA.NOMBRE, ID_PRODUCTO_PRODUCTO.ID)		
Atributo	Tipo	Descripción	Restricción
NOMBRE_CATEGORIA	VARCHAR(50)	Nombre de la categoría	PK,FK
ID_PRODUCTO	INTEGER	Id del producto	PK,FK

Figura 30. Tabla de Categoria_Producto

- DEPENDENCIAS: {}
- LLAVES: {(NOMBRE_CATEGORIA, ID_PRODUCTO)}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al no haber ni una dependencia.
- 3NF: No hay dependencias transitivas entre no primos al no haber ni una dependencia.
- BCNF: No hay dependencias no triviales entre primos al no haber ni una dependencia que los conecte directamente.

Nombre	CUENTA		
Descripción	Información de las cuentas del sistema		
Llave	PK(NUMEROCUENTA), FK(IDUSUARIO_USUARIO.ID, MESA_MESA.ID)		
Atributo	Tipo	Descripción	Restricción
VALOR	NUMBER	Valor de la cuenta	CK
NUMEROCUENTA	VARCHAR(20)	Número de cuenta	PK,CK
FECHA	DATE	Fecha de la cuenta	CK
IDUSUARIO	INTEGER	Id del usuario	FK
MESA	INTEGER	Id de la mesa	FK
PAGADA	CHAR(1)	Si está pagada o no	CK

Figura 31. Tabla de Cuenta

- DEPENDENCIAS: {(NUMEROCUENTA)->(VALOR,FECHA,IDUSUARIO,MESA,PAGADA)}

- LLAVES{(NUMEROCUENTA)}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al solo haber una dependencia.
- 3NF: No hay dependencias transitivas entre no primos al solo haber una dependencia.
- BCNF: No hay dependencias no triviales entre primos al solo haber una dependencia.

Nombre	RESERVA		
Descripción	Información de las reservas en el sistema		
Llave	PK(FECHA,ID_RESERVADOR), FK(ID_RESERVADOR _{USUARIO.ID} , NOMBRE_ZONA _{ZONA.NOMBRE} , NOMBRE_MENU _{MENU.NOMBRE} NOMBRE_RESTAURANTE _{MENU.NOMBRE_RESTAURANTE})		
Atributo	Tipo	Descripción	Restricción
FECHA	DATE	Fecha de la reserva	PK
ID_RESERVADOR	INTEGER	Id del reservador	PK,FK
NUM_PERSONAS	INTEGER	Número de personas	
NOMBRE_ZONA	VARCHAR(20)	Nombre de la zona	FK
NOMBRE_MENU	VARCHAR(20)	Nombre del menú	FK
NOMBRE_RESTAURANTE	VARCHAR(20)	Nombre del restaurante	FK

Figura 32. Tabla de Reserva

- DEPENDENCIAS: {(FECHA, ID_RESERVADOR)->(NUM_PERSONAS, NOMBRE_ZONA, NOMBRE_MENU, NOMBRE_RESTAURANTE)}
- LLAVES {(FECHA, ID_RESERVADOR)}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al solo haber una dependencia.
- 3NF: No hay dependencias transitivas entre no primos al solo haber una dependencia.
- BCNF: No hay dependencias no triviales entre primos al solo haber una dependencia.

Nombre	PEDIDO_MENU		
Descripción	Información del pedido de un menú		
Llave	PK(NUMERO_CUENTA, NOMBRE_MENU, NOMBRE_RESTAURANTE), FK(NUMERO_CUENTA _{CUENTA.NUMEROCUENTA} , NOMBRE_MENU _{MENU.NOMBRE} NOMBRE_RESTAURANTE _{MENU.NOMBRE_RESTAURANTE})		
Atributo	Tipo	Descripción	Restricción
NUMERO_CUENTA	VARCHAR(20)	Número de la cuenta	PK,FK
NOMBRE_MENU	VARCHAR(20)	Nombre del menú	PK,FK
NOMBRE_RESTAURANTE	VARCHAR(20)	Nombre del restaurante	PK,FK
CANTIDAD	INTEGER	Cantidad de producto	CK
ENTREGADO	CHAR(1)	Si ha sido entregado o no	CK

Figura 33. Tabla de Pedido_Menu

- DEPENDENCIAS: {(NUMERO_CUENTA, NOMBRE_MENU, NOMBRE_RESTAURANTE)->(CANTIDAD,ENTREGADO)}

- LLAVES{(NUMERO_CUENTA, NOMBRE_RESTAURANTE), NOMBRE_MENU,
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al solo haber una dependencia.
- 3NF: No hay dependencias transitivas entre no primos al solo haber una dependencia.
- BCNF: No hay dependencias no triviales entre primos al solo haber una dependencia.

Nombre	PEDIDO_PROD		
Descripción	Información de pedidos de productos		
Llave	PK(NUMERO_CUENTA, ID_PRODUCTO, NOMBRE_RESTAURANTE), FK(NUMERO_CUENTA_CUENTA.NUMEROCUENTA, ID_PRODUCTO_INFO_PROD_REST.ID_PRODUCTO, NOMBRE_RESTAURANTE_INFO_PROD_REST.NOMBRE_RESTAURANTE)		
Atributo	Tipo	Descripción	Restricción
NUMERO_CUENTA	VARCHAR(20)	Número de la cuenta	PK,FK
ID_PRODUCTO	INTEGER	Id del producto	PK,FK
NOMBRE_RESTAURANTE	VARCHAR(20)	Nombre del restaurante	PK,FK
CANTIDAD	INTEGER	Cantidad de producto	CK
ENTREGADO	CHAR(1)	Si ha sido entregado o no	CK

Figura 34. Tabla de Pedido_Prod

- DEPENDENCIAS: {(NUMERO_CUENTA, NOMBRE_RESTAURANTE)->(CANTIDAD,ENTREGADO)} ID_PRODUCTO,
- LLAVES{(NUMERO_CUENTA, NOMBRE_RESTAURANTE), ID_PRODUCTO,
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al solo haber una dependencia.
- 3NF: No hay dependencias transitivas entre no primos al solo haber una dependencia.
- BCNF: No hay dependencias no triviales entre primos al solo haber una dependencia.

Nombre	PREFERENCIAZONA		
Descripción	Información de las preferencias de un usuario por zona		
Llave	PK(IDUSUARIO, NOMBREZONA), FK(IDUSUARIO_PREFERENCIA.IDUSUARIO, NOMBREZONA_ZONA.NOMBRE)		
Atributo	Tipo	Descripción	Restricción
IDUSUARIO	INTEGER	Id del usuario	PK,FK
NOMBREZONA	VARCHAR(20)	Nombre de la zona	PK,FK

Figura 35. Tabla de PreferenciaZona

- DEPENDENCIAS: {}
- LLAVES{(IDUSUARIO,NOMBREZONA)}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al no haber ni una dependencia.
- 3NF: No hay dependencias transitivas entre no primos al no haber ni una dependencia.

- BCNF: No hay dependencias no triviales entre primos al no haber ni una dependencia que los conecte directamente.

Nombre	PREFERENCIACATEGORIA		
Descripción	Información de la preferencias de un usuario por categoría		
Llave	PK(IDUSUARIO, NOMBRECATEGORIA FK(IDUSUARIO PREFERENCIA.IDUSUARIO, NOMBRECATEGORIA CATEGORIA.NOMBRE)		
Atributo	Tipo	Descripción	Restricción
IDUSUARIO	INTEGER	Id del usuario	PK,FK
NOMBRECATEGORIA	VARCHAR(20)	Nombre de la categoría	PK,FK

Figura 36. Tabla de PreferenciaCategoria

- DEPENDENCIAS: {}
- LLAVES: {(IDUSUARIO, NOMBRECATEGORIA)}
- 1NF: No existen atributos multivalor
- 2NF: No hay dependencias parciales al no haber ni una dependencia.
- 3NF: No hay dependencias transitivas entre no primos al no haber ni una dependencia.
- BCNF: No hay dependencias no triviales entre primos al no haber ni una dependencia que los conecte directamente.

Por último, se muestra a continuación la consulta SQL pedida con el resultado obtenido. Cabe destacar que el resultado tiene 249 resultados, por lo que se adjunta una imagen parcial del resultado, y se adjunta en la carpeta docs (CONSULTA2A.pdf o CONSULTA"A.xls) todo lo obtenido:

```
SELECT NOMBRETABLA, NOMBRECOL, DATA_TYPE AS TIPODATO, NOMBRECONSTRAINTCOLUMNA,
NOMBRETABLAREFFK
FROM
  (SELECT T.TABLE_NAME AS NOMBRETABLA, T.COLUMN_NAME AS NOMBRECOL,
T.CONSTRAINT_NAME AS NOMBRECONSTRAINTCOLUMNA
  FROM ALL_CONS_COLUMNS T
 WHERE T.OWNER LIKE 'ISIS2304A061720')
NATURAL FULL OUTER JOIN
  (SELECT A.TABLE_NAME AS NOMBRETABLA,C.COLUMN_NAME AS NOMBRECOL,B.TABLE_NAME AS
NOMBRETABLAREFFK,
  A.CONSTRAINT_NAME AS NOMBRECONSTRAINTCOLUMNA
  FROM ALL_CONSTRAINTS A, ALL_CONSTRAINTS B, ALL_CONS_COLUMNS C
 WHERE A.OWNER LIKE 'ISIS2304A061720' AND A.CONSTRAINT_NAME LIKE 'FK%' AND
A.R_CONSTRAINT_NAME=B.CONSTRAINT_NAME AND
  B.CONSTRAINT_NAME LIKE 'PK%' AND C.CONSTRAINT_NAME LIKE A.CONSTRAINT_NAME),
ALL_TAB_COLUMNS
WHERE NOMBRECONSTRAINTCOLUMNA NOT LIKE 'BIN%' AND NOMBRETABLA LIKE TABLE_NAME AND
NOMBRECOL LIKE COLUMN_NAME AND NOMBRETABLA NOT IN ('RESERVAS','SILLASRESERVAS')
ORDER BY NOMBRETABLA ASC, NOMBRECOL ASC, NOMBRECONSTRAINTCOLUMNA ASC;
```

NOMBRETABLA	NOMBRECOL	TIPODATO	NOMBRECONSTRAINTCOLUMNA	NOMBRETABLAFFK
1 CATEGORIA	NOMBRE	VARCHAR2	PK_CATEGORIA	(null)
2 CATEGORIA_MENU	NOMBRE_CATEGORIA	VARCHAR2	CATEGORIA_DE_MENU_EXISTE	(null)
3 CATEGORIA_MENU	NOMBRE_CATEGORIA	VARCHAR2	FK_CATEGORIA_DEL_MENU	CATEGORIA
4 CATEGORIA_MENU	NOMBRE_CATEGORIA	VARCHAR2	PK_CATEGORIAMENU	(null)
5 CATEGORIA_MENU	NOMBRE_MENU	VARCHAR2	FK_MENU_DE_LA_CATEGORIA	MENU
6 CATEGORIA_MENU	NOMBRE_MENU	VARCHAR2	NOMBRE_DE_MENU_EXISTE	(null)
7 CATEGORIA_MENU	NOMBRE_MENU	VARCHAR2	PK_CATEGORIAMENU	(null)
8 CATEGORIA_MENU	NOMBRE_RESTAURANTE	VARCHAR2	FK_MENU_DE_LA_CATEGORIA	MENU
9 CATEGORIA_MENU	NOMBRE_RESTAURANTE	VARCHAR2	NOMBRE_DE_RESTAURANTE_EXISTE	(null)
10 CATEGORIA_MENU	NOMBRE_RESTAURANTE	VARCHAR2	PK_CATEGORIAMENU	(null)
11 CATEGORIA_PRODUCTO	ID_PRODUCTO	NUMBER	FK_PRODUCTO_DE_CATEGORIA	PRODUCTO
12 CATEGORIA_PRODUCTO	ID_PRODUCTO	NUMBER	PK_CATEGORIA_PRODUCTO	(null)
13 CATEGORIA_PRODUCTO	ID_PRODUCTO	NUMBER	PRODUCTO_DE_CATEGORIA_EXISTE	(null)
14 CATEGORIA_PRODUCTO	NOMBRE_CATEGORIA	VARCHAR2	CATEGORIA_DE_PRODUCTO_EXISTE	(null)
15 CATEGORIA_PRODUCTO	NOMBRE_CATEGORIA	VARCHAR2	FK_CATEGORIA_DE_PRODUCTO	CATEGORIA
16 CATEGORIA_PRODUCTO	NOMBRE_CATEGORIA	VARCHAR2	PK_CATEGORIA_PRODUCTO	(null)
17 CATEGORIA RESTAURANTE	NOMBRE_CATEGORIA	VARCHAR2	CATEGORIA DE REST EXISTE	(null)

Figura 37. Tabla parcial de la consulta SQL

3.4 Cambios sustanciales de la Iteración 2 a la Iteración 3

- Inclusión de la clase `Mesa` con sus relaciones correspondientes.
- Inclusión de un atributo booleano para saber si la cuenta está pagada o no.
- Vinculación de la capacidad ocupada de una mesa a la zona en la que pertenece.
- Creación de las clases `SustitucionIngrediente`, `SustitucionProducto` y `SustitucionIngredienteProducto` con sus respectivas relaciones, incluyendo la conexión con las clases `PedidoMenu` y `PedidoProducto`.

3.5 Documentación de la lógica de requerimientos nuevos y transaccionales

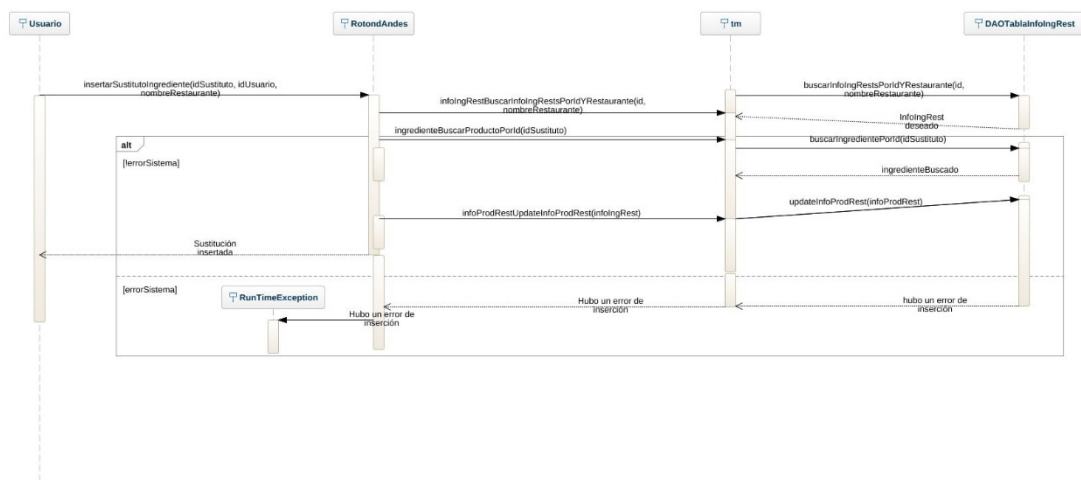


Figura 38. Diagrama secuencial para el requerimiento 11

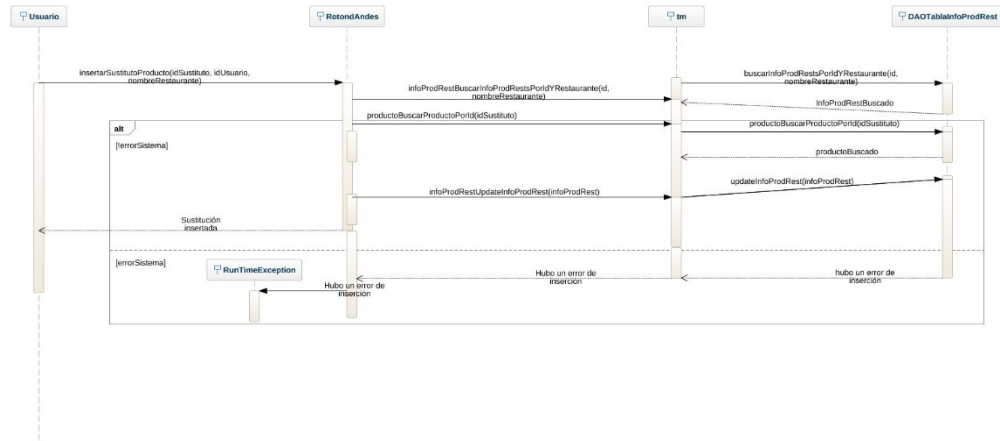


Figura 39. Diagrama secuencial para el requerimiento 12.

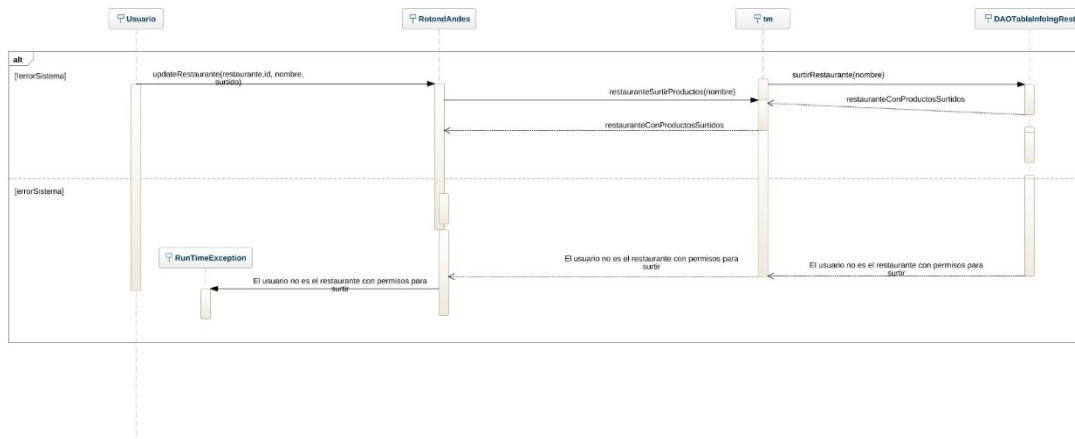


Figura 40. Diagrama secuencial para el requerimiento 13.

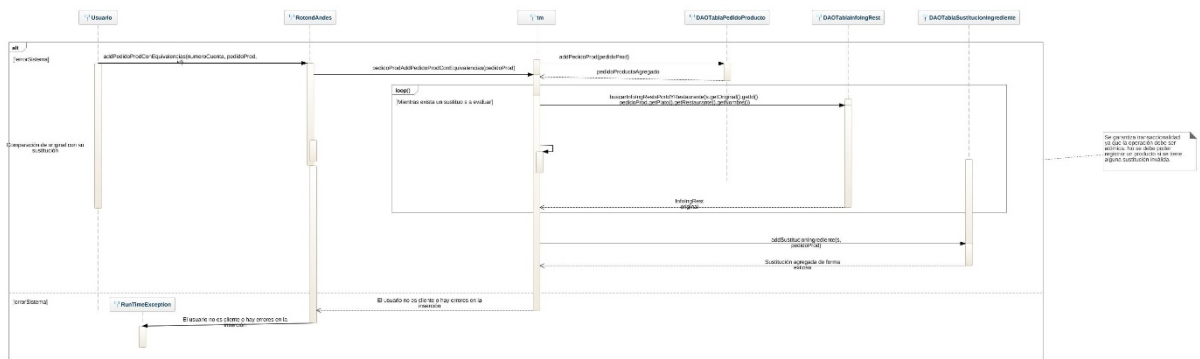


Figura 41. Diagrama secuencial para el requerimiento 14: Agregar pedido de producto con sustituciones de ingredientes.

Para este requerimiento se usa transaccionalidad al tenerse una propiedad de atomicidad: No se puede permitir mantener una sustitución agregada si hay alguna otra que es inválida.



Figura 42. Diagrama secuencial para el requerimiento 14: Agregar pedido de menú con sustituciones de productos y de ingredientes.

Para este requerimiento se usa transaccionalidad al tenerse una propiedad de atomicidad: No se puede permitir mantener una sustitución agregada si hay alguna otra que es inválida.

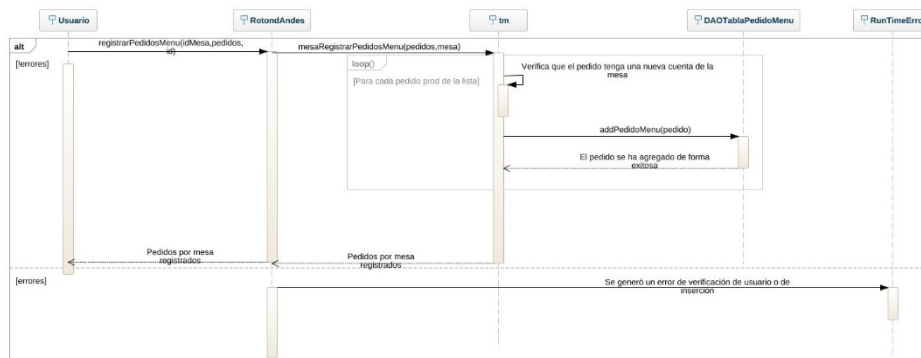


Figura 43. Diagrama secuencial para el requerimiento 15: Registro de menús.

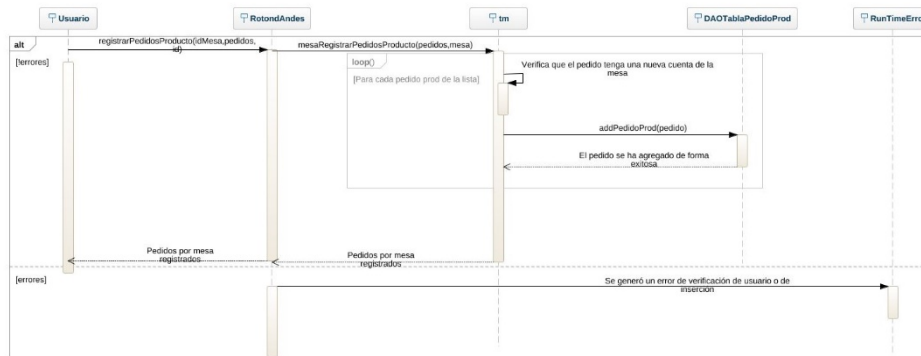


Figura 44. Diagrama secuencial para el requerimiento 15: Registro de productos.

Para este requerimiento se toma la decisión de usar el nivel SQL Read Committed para permitir que a la hora de registrar un producto o un servicio se vean todas las transacciones que han hecho commit o rollback en la base de datos.

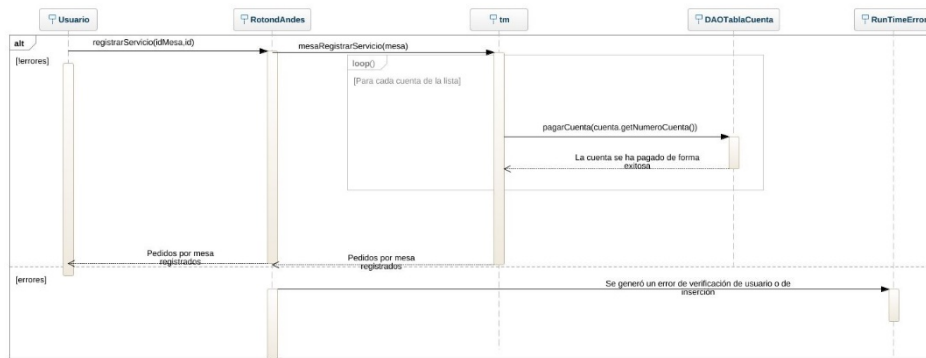


Figura 46. Diagrama secuencial para el requerimiento 16

Para este requerimiento se toma la decisión de usar el nivel SQL Read Committed para permitir que a la hora de registrar un producto o un servicio se vean todas las transacciones que han hecho commit o rollback en la base de datos. A su vez, se garantiza la operación de atomicidad de no pagar unas cuentas y otras no.

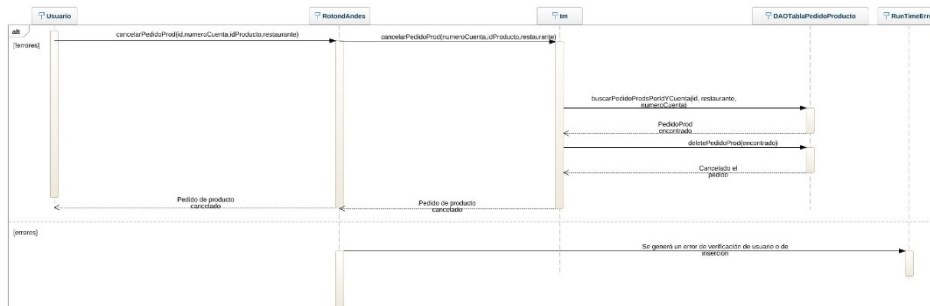


Figura 47. Diagrama secuencial para el requerimiento 17: Producto

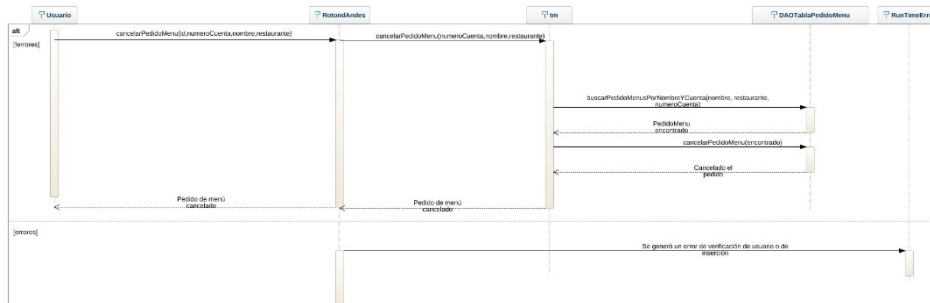


Figura 48. Diagrama secuencial para el requerimiento 17: Menú

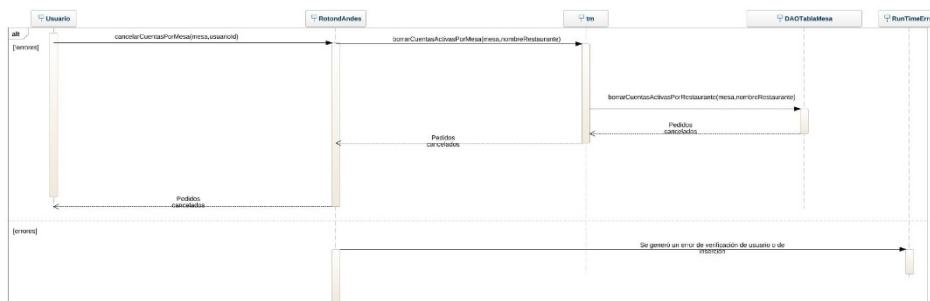


Figura 49. Diagrama secuencial para el requerimiento 17: Mesa

Para este requerimiento se utilizó la atomicidad de las transacciones para que no se cancelen cuentas de forma parcial.

4 Construcción de la aplicación

En esta sección se presenta una breve descripción de cómo se realizaron los requerimientos introducidos en esta nueva iteración. Las pruebas de los mismos se encuentran en colecciones de *Postman* como se sugiere en [1], junto con casos de éxito y fallo en un archivo *Excel*.

4.1 RF9

Para este requerimiento, cada vez que se agregaba un producto o un menú al pedido se verificaba la disponibilidad y se actualizaba el precio de la cuenta dependiendo del valor y cantidad dados.

4.2 RF10

Para este requerimiento se tuvo la decisión de coger cada producto y menú encontrado en la cuenta y verificar si era posible pagarlo o no con la disponibilidad actual del restaurante; si lo anterior fallaba de alguna manera se guardaba en una lista todos los artículos comprados que no se tuvieran en el momento y se le mostraba al usuario el listado respectivo. Con este listado se puede decidir actualizar el pedido, dejar todo cómo está o , cuando se ofrezca la posibilidad de utilizar sustitutos, sustituir el producto dado por alguna de sus posibles opciones.

4.3 RF11

Para este requerimiento se permitió añadir las equivalencias de dos maneras: o mediante un PUT a la tabla de `InfoIngRest` el cual permiten modificar la lista de sustituciones que está guardada en ella, o mediante un POST en el cual se pueden asociar dos ingredientes, conociendo sus identificadores. Borrar la sustitución solo se permitió a través del PUT.

4.4 RF12

Para este requerimiento se permitió añadir las equivalencias de dos maneras: o mediante un PUT a la tabla de `InfoProdRest` el cual permiten modificar la lista de sustituciones que está guardada en ella, o mediante un POST en el cual se pueden asociar dos productos, conociendo sus identificadores. Borrar la sustitución solo se permitió a través del PUT.

4.5 RF13

Se pretende que cuando se actualice un restaurante se indique si se quiere surtir de nuevo los productos o no. En caso tal de que se desee hacer esto, se valida que solo lo pueda hacer un usuario dueño del restaurante objetivo. Al final, cuando se obtiene la información del restaurante se puede ver la reacomodación de datos de forma clara.

4.6 RF14

Este requerimiento se dividió en dos servicios REST diferentes, uno para ordenar productos, y otro para ordenar menús. Al ordenar el producto toca añadir en una lista las sustituciones deseadas, al ordenar el menú se puede sustituir un producto de él por otro, o sustituir un ingrediente dentro de alguno de los productos que componen el menú. Ambas cosas pueden hacerse añadiendo la lista de sustituciones al cuerpo de la petición.

4.7 RF15

Este requerimiento hace uso del requerimiento RF9 para ordenar varios pedidos. Todos deben pertenecer a la misma mesa, aunque pueden ser de diferentes cuentas. El ordenar los pedidos se hace de manera atómica, es decir, o se hacen todos los pedidos o ninguno. Cabe aclarar que esto también se hace a partir de dos servicios, uno para ordenar muchos productos, y otro para ordenar muchos menús.

4.8 RF16

En este requerimiento hace uso del requerimiento RF10 para tomar el servicio de todas las cuentas de una mesa, lo cual se hace de manera atómica, si no hay disponibilidad de algún pedido no se lleva a cabo el servicio de ninguno.

4.9 RF17

En general, para este requerimiento se usó la opción de borrar para las clases `PedidoProd` y `PedidoMenu` por lo que si uno es el restaurante, solo puede borrarlos del sistema antes de haberlos entregado. En el caso de una mesa se permite que el usuario de tipo `Restaurante` intente eliminar todas los pedidos que pertenecen únicamente a su local dentro de la misma mesa.

4.10 RFC7

Para este requerimiento se sigue la ruta "<http://localhost:8080/VideoAndes/rest/usuarios/productos>", además se crearon las siguientes clases en el paquete `rfc`:

- `ContenedoraClienteProducto`: Muestra la información de un cliente (usando su identificación), dicha información se encuentra en la clase `ContenedoraMesaMenuCliente`.
- `ContenedoraMesaMenuCliente`: Representa a través de un booleano si el menú tiene o no una mesa asignada, y después muestra un listado de información expresado en la clase `ContenedoraMenuCliente`.
- `ContenedoraMenuCliente`: Tiene el nombre del restaurante, el nombre del menú y la información relevante del producto. En caso tal de que se haya consumido el producto por fuera del menú, el nombre del menú no existe. La información del producto se expresa en la clase `ProductoInformativo`.
- `ProductoInformativo`: Solo tiene un nombre, una descripción, tipo, traducción e identificación.

En esencia, dependiendo del usuario que lo solicite, se muestra la información de un cliente (el que realiza la consulta) o de todos los clientes del sistema (si el usuario que lo solicita es un administrador).

4.11 RFC8

Para este requerimiento se sigue la ruta "<http://localhost:8080/VideoAndes/rest/restaurantes/informacion-financiera>", además se crearon las siguientes clases en el paquete `rfc`:

- `ContenedoraRestauranteInfoFinanciera`: Contiene la información financiera de un restaurante particular.
- `InformacionFinanciera`: Es la información financiera de un restaurante. Contiene el producto al que hace referencia (Id de un plato o nombre del menú), la cantidad que se ha vendido a cuentas con o sin cliente, y el total de ventas en ambos casos.

Dependiendo de la identificación del usuario que llega como parámetro se escoge si mostrar toda la información de todos los restaurantes, o si la de uno solo (el restaurante del que el usuario sería dueño). Esta información se almacena en las clases ya descritas.

4.12 RNF3

Para probar el mantenimiento de la concurrencia, además de tener el uso de `commit` y `rollback`, y de usar un nivel de aislamiento `READ COMMITTED`, se realizó una carpeta de pruebas en el archivo *Postman* llamado "PruebasConcurrencia". En este se encuentran tres carpetas:

- **Misma sesión**: Se intenta probar la concurrencia de dos pedidos simulados en una sola sesión donde se agota la cantidad de uno de los productos, y cuando se va a hacer otro pedido sobre el mismo ítem se genera un error.

- Sesión 1: Carpeta que se corre antes de la carpeta “Sesión 2”, y es usada para surtir el restaurante llamado “Skinte”. Se supone que después de haberse surtido el restaurante, se inicia la prueba de la carpeta “Sesión 2”, y uno de los registros de pedido con valor “3000” en alguna de las dos sesiones debe fallar debido a la falta de disponibilidad.
- Sesión 2: Pruebas que se corren después de haberse surtido el restaurante “Skinte” en la carpeta “Sesión 1”.

5 Consideraciones adicionales

Dentro de la carpeta `data` se tienen las siguientes consideraciones:

- Se tienen los archivos de pruebas de *Postman* en la carpeta `pruebas`. Antes de ejecutarlas se debe correr el archivo `PrePostman.sql`.
- En el archivo `BorradoEInserciones.sql` se tienen los comandos para borrar todos los datos existentes en la base de datos y volver a poblarla en caso tal de que se corrompan de alguna manera los datos del sistema.
- En el archivo `DBSETUPFULL.sql` se tienen los esquemas de las tablas para reconstruir la base de datos dado el caso en que haya la necesidad de hacerlo.
- En la carpeta `inicializacion` están todas las inserciones en tablas utilizadas en la población de la base de datos. Además, las tablas de equivalencias se encuentran en el archivo `Prepostman.sql`.
- La ruta se sigue llamando “VideoAndes” de acuerdo con el ejemplo de clase, dado que no se encontró la forma de cambiar el nombre a “RotondAndes” como se pretendía.

6 Análisis de resultados

6.1 Aprendizajes y logros

A lo largo del desarrollo de la iteración 3 se aprendió a refinar un sistema transaccional básico con el uso de las propiedades ACID.

6.2 Conclusiones

- Se debe saber determinar las operaciones transaccionales del sistema porque hay varias que requieren ser o todo o nada.
- SQL es una herramienta poderosa que permite realizar consultas complejas sobre modelos de datos gigantes.
- La interconexión entre clases permite la funcionalidad completa de un sistema transaccional dado.
- Toda decisión de diseño se debe reflejar en la base de datos planteada.
- Las propiedades ACID ayudan a determinar el manejo sobre la concurrencia de la aplicación, permitiendo definir diferentes niveles de aislamiento, dependiendo de las necesidades del programador. Para esta iteración se usó el modo `READ COMMITED` al ver que solo se debía garantizar que las transacciones pudieran ver quien había realizado o no un `commit` durante las diferentes transacciones de la aplicación.

7 Bibliografía

1. Universidad de los Andes. *Caso de estudio - v3*. [En línea] Universidad de los Andes. [Citado el: 4 de Octubre de 2017.] https://sicuapplus.uniandes.edu.co/bbcswebdav/pid-1918605-dt-content-rid-20597579_1/courses/UN_201720_ISIS2304/isis2304-172-CasoEstudio-RotondAndes-v3%281%29.pdf