

Iteración 3

Rafael Andrés Forero Alvarado 201514185

Jorleidis Monsalvo Torres 201621171

- 1) Si lo requiere, ajuste el modelo del mundo (modelo conceptual: diagrama de clases UML) propuesto en la iteración 2 que representó el caso de estudio. Indique cuáles clases del modelo del mundo fueron actualizadas o creadas en esta iteración.**

Se modificaron relaciones faltantes como la de pedido con menús y productos, se agregaron las clases equivalencias y mesas con sus respectivas relaciones. Se agregó el atributo id a cliente.

- 2) –a A partir del diseño existente, analice el impacto que representa la introducción de los nuevos requerimientos y restricciones a nivel del modelo conceptual. Realice los cambios necesarios en su modelo relacional para respetar las reglas de negocio y asegurar la calidad del mismo. Tenga en cuenta los comentarios recibidos en la sustentación de la iteración 2. Documente el diseño y las decisiones tomadas para crear los elementos de la base de datos que da el respaldo de persistencia a la aplicación, a partir del modelo conceptual. Incluya un listado con las tablas generadas en la base de datos, utilizando los estándares establecidos, disponibles en la wiki del curso (sección tutorial). Este listado, resultado de una consulta SQL, debe incluir el nombre de la tabla, el nombre y el tipo de dato de sus campos, así como los nombres de restricciones de llaves primarias, llaves foráneas y de chequeo. } Sea claro en mencionar explícitamente los cambios relevantes entre su diseño entregado en la iteración anterior y esta.**

En cuanto al modelo conceptual, tuvimos que agregar nuevas clases, atributos y relaciones. En el modelo relacional, se crearon nuevas llaves foráneas, nuevas tablas.

Bono:

¿Qué diferencias hay en el manejo transaccional por parte de un contenedor de aplicaciones con respecto al manejo dado por parte del programador de la aplicación? Explique claramente las ventajas y desventajas de cada uno de ellos y cómo se refleja en su arquitectura de software y diseño detallado de su RotondAndes.

Contenedor de aplicaciones: las ventajas son la escalabilidad, el acceso seguro y transparente al sistema. Un sistema más manejable, interfaz de administración consistente y simplificada, con una interconectividad más eficientes. Los elementos creados dentro del contenedor se pueden usar o aplicar simultáneamente. Ofrece seguridad, entre otros.

Programador: son muchas las desventajas ya que a un programador le tocaría adaptar el sistema para que funcione en casos extras contemplados en tiempo de diseño.

En la arquitectura de nuestro proyecto esto se ve reflejado en la distribución de los paquetes, el manejo de capas permite una mejor organización del programa y en caso de algún fallo podemos saber exactamente en donde falla y si está relacionado con la base de datos.