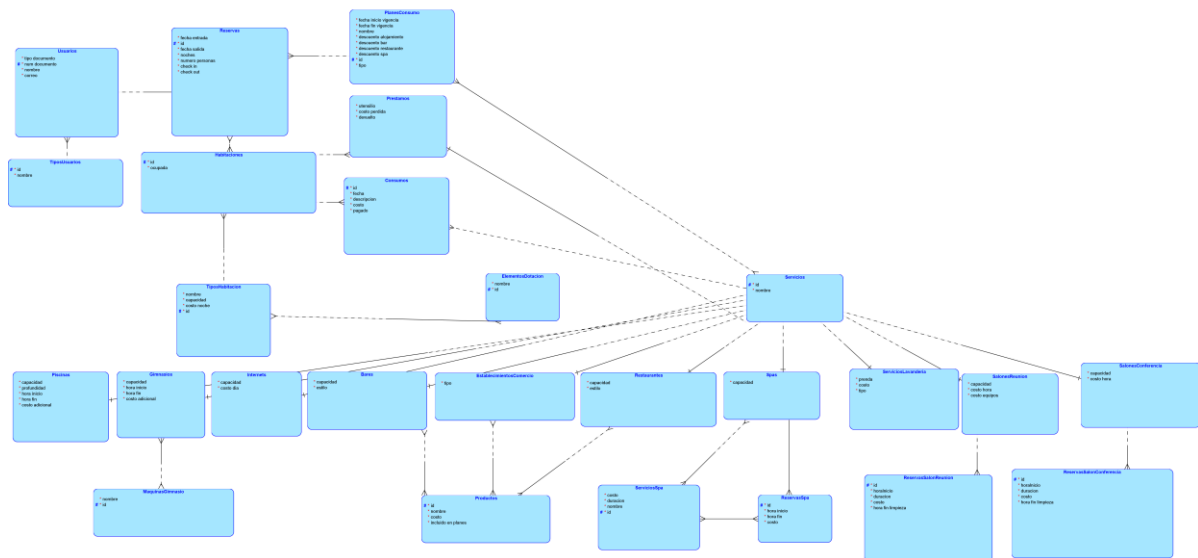


Wilmer Manuel Arévalo González - 202214720 - w.arevalo
Sara Sofía Cárdenas Rodríguez - 202214907 - ss.cardenas
Daniel Felipe Diaz Moreno – 202210733 - d.diazm

[illegible]

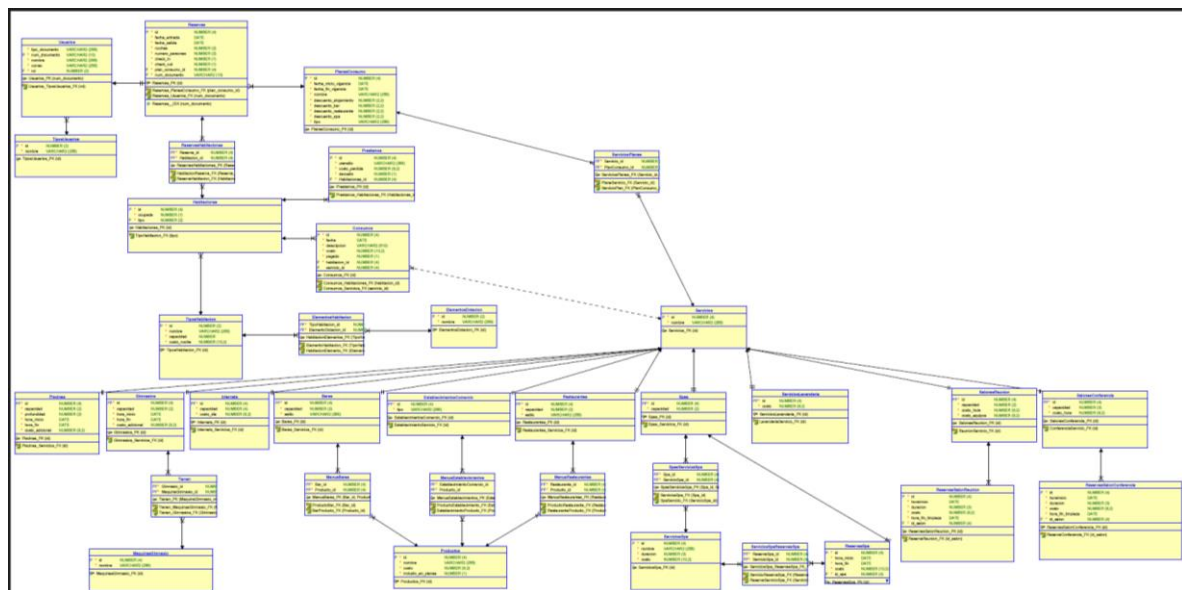
Para obtener una resolución óptima, consulte el archivo UML.pdf ubicado en la carpeta "modelos" dentro del directorio "docs".

MODELO E/R



Para obtener una resolución óptima, consulte el archivo modeloER.pdf ubicado en la carpeta "modelos" dentro del directorio "docs".

MODELO RELACIONAL



Para obtener una resolución óptima, consulte el archivo `modeloRelacional.pdf` ubicado en la carpeta "modelos" dentro del directorio "docs".

RESULTADOS LOGRADOS

Se crearon todas las relaciones necesarias para modelar el modelo del mundo del problema, Esto nos ha permitido satisfacer de manera efectiva cada uno de los requerimientos funcionales establecidos, así como abordar en gran medida los requerimientos no funcionales. Además, la implementación permite prevenir la redundancia de datos y detectar y evitar anomalías durante las operaciones de inserción o eliminación. Esta característica es fundamental para mantener la integridad y la coherencia de la base de datos, asegurando que los datos se almacenen de manera eficiente y sin errores, lo que a su vez contribuye a la robustez y confiabilidad del sistema en su conjunto.

En pocas palabras, se implementaron 35 entities del modelo, 35 repositorios y los controladores necesarios, que operan con front-end y/o consola. Además, las tablas de la base de datos fueron creadas conforme al diseño ya evidenciado.

RESULTADOS NO LOGRADOS

Debido a problemas con la base de datos, no fue posible construir por completo la interfaz gráfica. A pesar de que el proyecto incluye la mayoría de los archivos de la interfaz, no fue posible completarla en su totalidad. Por lo tanto, se decidió que todos los requisitos se ejecuten desde la consola.

Adicionalmente, no se realizaron las pruebas de unicidad, integridad y restricciones.

BALANCE PLAN DE PRUEBAS

A pesar de que no se pudieron especificar pruebas con la lógica de la aplicación, al introducir los archivos de sentencias sql Población, Asociaciones y Borrado, se pudo evidenciar que las inserciones y borrados son los esperados, sin caer en anomalías ni errores de compilación.

SUPUESTOS ADICIONALES DE LAS REGLAS DE NEGOCIO

Todos los servicios del mismo tipo que están incluidos en un plan de consumo específico deben ofrecer el mismo porcentaje de descuento. Por ejemplo, no es permitido que un conjunto de bares ofrezca un descuento del 20% mientras que otro grupo, que forma parte de la misma promoción, ofrezca un descuento del 30%.

Además, una reserva del spa puede incluir 1 o más servicios del spa en cuestión. Los préstamos fueron modelados al mismo nivel que los consumos, ya que no tenía sentido en nuestro modelo que heredaran de la clase Servicio. También decidimos que un cliente pueda tener varias reservas activas. Finalmente, los planes consumo fueron modelados en una sola clase incluyendo los descuentos en los servicios más importantes

LLAVES CANDIDATAS ENCONTRADAS

Definimos id, num_documento, Reserva_id con Habitacion_id, TipoHabitacion_id con ElementoDotacion_id, y Servicio_id con PlanConsumo_id para la parte superior del modelo relacional.

Para los servicios, Gimnasio_id con MaquinaGimnasio_id, Bar_id con Producto_id, EstablecimientoComercio_id con Producto_id, Restaurante_id con Producto_id.

Para el SPA, Spa_id con ServicioSpa_id y ReservaSpa_id con ServicioSpa_id

NIVEL DE NORMALIZACIÓN

Las relaciones presentadas tienen un nivel de normalización Boyce-Codd, lo que garantiza la integridad y eficiencia de la base de datos

FN1: Se cumple ya que ninguna tabla tiene atributos en forma de lista, es decir, todos los atributos son atómicos. Para esto, se usaron clases de asociación y/o FKs

FN2: Se cumple ya que no encontramos dependencias parciales entre atributos primos y no primos dentro de nuestras relaciones. Esto se logró en mayor medida separando entidades desde el UML.

FN3: Se cumple ya que no encontramos dependencias transitivas entre atributos no primos dentro de nuestras relaciones. Esto se logró en mayor medida separando entidades desde el UML.

BCNF: Se cumple ya que en todas las relaciones existe una única llave candidata que se convirtió en PK. Esto se debe a que creamos atributos de id solo cuando era necesario y aprovechamos las reglas de negocio para establecer la PK. Por lo tanto, no existen varias llaves candidatas que se sobrelapen.