

Semana 4

Modelamiento de Bases de Datos (PRY2204)

Formato de respuesta

| **Nombre estudiante: Luciano Rivera** |  |
| --- | --- |
| **Asignatura: Modelamiento de base de datos** | **Carrera: Desarrollador de Aplicaciones** |
| **Profesor: Josué Oteiza** | **Fecha: 02/09/2024** |

# Descripción de la actividad

En esta cuarta semana realizarás una actividad formativa en parejas por encargo llamada "Modelando y Normalizando datos", donde aplicarás prácticamente los principios de normalización en la creación de modelos de bases de datos, centrándote en las claves primarias y foráneas de normalización (1FN, 2FN y 3FN), así como en las reglas de integridad referencial para la inserción, actualización y eliminación de dato, aplicándolo en el desarrollo y solución de un caso planteado. También aplicarás el reconocimiento de las restricciones de integridad e implementarás lo aprendido para formar modelos de datos normalizados en SQL Data Modeler.

## Instrucciones específicas

Para realizar la actividad formativa de la semana, analizaremos el siguiente caso:

**Contexto:**

La cadena de servicios automotrices en Santiago enfrenta desafíos significativos debido al crecimiento de su clientela y la demanda de sus servicios. La gestión manual de procesos como la toma de datos personales del cliente, el presupuesto de mantenimiento o reparación, y el registro en papel, han comenzado a comprometer la calidad del servicio. Este escenario ha derivado en una sobrecarga laboral para el personal, pérdida de clientes, repuestos y herramientas, y un general desorden administrativo.

Dada esta situación, tu rol como consultor/a es vital para el proceso de automatización que busca implementar la cadena de servicios. Se espera que diseñes un modelo conceptual de negocio que sirva de base para desarrollar una base de datos inicial enfocada en la gestión del servicio automotriz. Este sistema deberá contener toda la información relevante para el negocio, optimizando y mejorando los procesos operativos.

**Los aspectos clave a modelar son los siguientes:**

* Clientes y vehículos: registrar los datos esenciales de los clientes (RUT, nombres, apellidos, domicilio, contactos) y los detalles de sus vehículos (patente, marca, modelo, año, color) para personalizar y agilizar el servicio.
* Ordenes de atención: generar y gestionar órdenes de servicio por cada intervención, detallando el tipo de servicio requerido (mantención, pintura, desabolladura, etc.) y los repuestos utilizados, permitiendo un seguimiento detallado de cada proceso.
* Seguimiento en línea: proporcionar actualizaciones en tiempo real sobre el estado del vehículo (recepción, revisión, control de calidad, disponibilidad, entrega) para mantener informados a los clientes sobre el progreso y la fecha estimada de entrega.
* Validación de Servicios: facilitar la revisión y aprobación de los servicios realizados, incluyendo la mano de obra, los repuestos y los servicios adicionales, para actualizar con precisión la fecha de entrega del vehículo.

La primera etapa del proyecto se centrará exclusivamente en la gestión y automatización de estos procesos, dejando la facturación de servicios para una fase posterior una vez que el modelo actual sea validado y aceptado.

Ahora, para poder realizar la actividad, deberás cumplir con los siguientes pasos:

**Paso 1: Identificación de entidades y atributos**

* Analiza el contexto y la descripción del problema para identificar las entidades clave que formarán parte del modelo. Tendrás que determinar los atributos necesarios para cada entidad.

**Paso 2: Definición de claves**

Asigna una clave primaria a cada entidad identificada, asegurando la unicidad de los registros. Así también, establece las claves foráneas necesarias para relacionar las entidades entre sí.

**Paso 3: Establecimiento de relaciones**

Define claramente cómo se relacionan las entidades entre sí. Por ejemplo, una Orden de Atención está vinculada a un único Cliente y un único Vehículo, pero puede incluir varios Servicios.

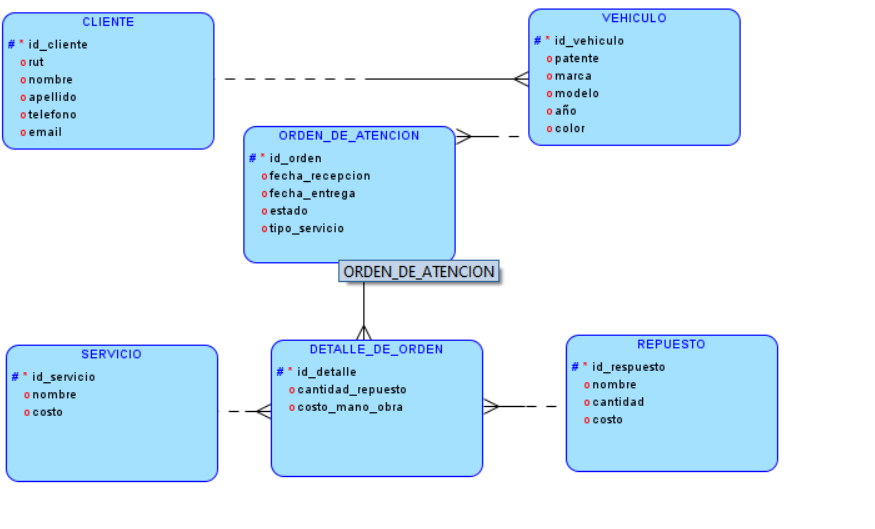
**Paso 4: Normalización**

Aplica las reglas de normalización (1FN, 2FN, 3FN) revisadas durante la semana para estructurar el modelo de manera que se minimice la redundancia y se asegure la integridad de los datos.

**Paso 5:** para realizar el ejercicio, tendrás que utilizar la herramienta Oracle SQL Data Modeler, disponible de descarga a través del siguiente enlace:

[https://www.oracle.com/database/sqldeveloper/technologies/sql-data-modeler/download/](https://www.oracle.com/database/sqldeveloper/technologies/sql-data-modeler/download/ )

Cuando tengas lista tu solución, adjunta una imagen de tu Modelo Entidad-Relación en esta parte:



Solución del proyecto:

* **Entidades del Proyecto**

° CLIENTES

° VEHICULO

° ORDEN DE ATENCION

° SERVICIO

° REPUESTO

° DETALLE DE ORDEN

* **Atributos de la Entidades**

° CLIENTES

. id\_cliente = (\*) clave primaria

. rut

. nombre

. apellido

. domicilio

. teléfono

. email

° VEHICULO

. id\_vehiculo (\*) clave primaria

. patente

. marca

. modelo

. año

. color

. id\_cliente = (\*\*) clave foránea

° ORDEN\_DE\_ATENCION

. id\_orden = (\*\*) clave primaria

. fecha\_recepcion

. fecha\_entrega

. estado

. Id\_vehiculo = (\*\*) clave foránea

. tipo\_servicio

° SERVICIO

. id\_servicio = (\*) clave primaria

. nombre

. costo

° REPUESTO

. id\_repuesto (\*) clave primaria

. nombre

. cantidad

. costo

° DETALLE\_DE\_ORDEN

. id\_detalle = (\*) clave primaria

. id\_orden = (\*\*) clave foránea

. id\_servicio = (\*\*) clave foránea

. id\_repuesto = (\*\*) clave foránea

. cantidad\_repuesto

. costo\_mano\_obra

* **Asignación Claves Primarias**

. CLIENTE = id\_ cliente

. VEHICULO = id\_vehiculo

. ORDEN\_DE\_ATENCION = id\_orden

. SERVICIO = id\_servicio

. REPUESTO = id\_repuesto

. DETALLE\_DE\_ORDEN = id\_detalle

* **Establecer Relaciones**
* **1. CLIENTE - VEHICULO**
* **Relación**: Un cliente puede tener múltiples vehículos.
* **Cardinalidad**: 1 a N (Un cliente tiene 0 a N vehículos)
* **Implementación**: En la entidad **Vehículo**, el atributo **id\_cliente** es una clave foránea que apunta a **id\_cliente** en la entidad **Cliente**.
* **2. VEHICULO – ORDEN\_DE\_ATENCION**
* **Relación**: Un vehículo puede tener múltiples órdenes de atención.
* **Cardinalidad**: 1 a N (Un vehículo tiene 0 a N órdenes de atención)
* **Implementación**: En la entidad **Orden de Atención**, el atributo **id\_vehiculo** es una clave foránea que apunta a **id\_vehiculo** en la entidad **Vehículo**.
* **3. ORDEN\_DE\_ATENCION – DETALLE\_DE\_ORDEN**
* **Relación**: Cada orden de atención puede tener múltiples detalles (servicios y repuestos utilizados).
* **Cardinalidad**: 1 a N (Una orden de atención tiene 1 a N detalles)
* **Implementación**: En la entidad **Detalle de Orden**, el atributo **id\_orden** es una clave foránea que apunta a **id\_orden** en la entidad **Orden de Atención**.
* **4. SERVICIO – DETALLE\_DE\_ORDEN**
* **Relación**: Un servicio puede estar en múltiples detalles de órdenes.
* **Cardinalidad**: 1 a N (Un servicio puede aparecer en 0 a N detalles)
* **Implementación**: En la entidad **Detalle de Orden**, el atributo **id\_servicio** es una clave foránea que apunta a **id\_servicio** en la entidad **Servicio**.
* **5. REPUESTO – DETALLE\_DE\_ORDEN**
* **Relación**: Un repuesto puede estar en múltiples detalles de órdenes.
* **Cardinalidad**: 1 a N (Un repuesto puede aparecer en 0 a N detalles)
* **Implementación**: En la entidad **Detalle de Orden**, el atributo **id\_repuesto** es una clave foránea que apunta a **id\_repuesto** en la entidad **Repuesto**.
* **Determinar la Cardinalidad De Las Relaciones**

**. CLIENTE – VEHICULO = (1:N)**

**. VEHICULO – ORDEN\_DE\_ATENCION = (1:N)**

**. ORDEN\_DE\_ATENCION – DETALLE\_DE\_ORDEN = (1:N)**

**. SERVICIO – DETALLE\_DE\_ORDEN = (1:N)**

**. REPUESTO – DETALLE\_DE\_ORDEN = (1:N)**

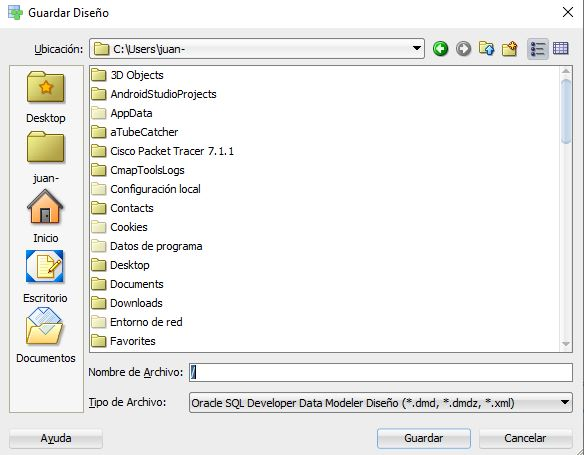
* **Reglas de Normalización**

**Adjunto Excel con el ejemplo en el ava.**

Además, tendrás que descargar el resultado. Para ello, tendrás que hacer clic en la opción Guardar como... del menú Archivo, esto despliega el submenú que se ilustra en la siguiente figura:

**Figura 1**

*Cómo guardar un archivo en SQL*



Nota. Ejemplo de guardado de archivo SQL. Oracle. (s.f.). *Oracle Data Modeler* [Software]. Oracle. <https://www.oracle.com/cl/database/sqldeveloper/technologies/sql-data-modeler/>

**Paso 6:** El archivo descargado desde SQL deberás subirlo al repositorio GitHub. Si no has creado tu cuenta aún, puedes hacerlo a través del siguiente enlace:

<https://github.com/>

Una vez subido el archivo a GitHub, deberás descargar el archivo comprimido .java desde tu repositorio, tal como se muestra en la imagen:

**Figura 2**

*Archivo .raw en GitHub*

Nota. Descarga de archivo desde repositorio GitHub. GitHub (s.f.). *GitHub.* <https://github.com/>

Posteriormente, desde el repositorio, deberás generar un enlace de tu proyecto:

**Figura 3**

Enlace de proyecto GitHub

Nota. Ejemplo de dónde se extrae un enlace en GitHub. GitHub (s.f.). *GitHub.* <https://github.com/>

Deja en este apartado el enlace de tu repositorio GitHub:

**Paso 7:** una vez adjunta tu respuesta y enlace, no olvides comprimir este documento y el archivo .raw y SQL en un archivo .rar, el cual deberás subir al AVA.



Reservados todos los derechos Fundación Instituto Profesional Duoc UC. No se permite copiar, reproducir, reeditar, descargar, publicar, emitir, difundir, de forma total o parcial la presente obra, ni su incorporación a un sistema informático, ni su transmisión en cualquier forma o por cualquier medio (electrónico, mecánico, fotocopia, grabación u otros) sin autorización previa y por escrito de Fundación Instituto Profesional Duoc UC La infracción de dichos derechos puede constituir un delito contra la propiedad intelectual.