|  |
| --- |
| FINAL project  TF1  LlenaFast |
| Nombre del curso: Diseño de Base de Datos  Sección: SV43  Nombre del profesor: Fidel Eugenio García Rojas  Integrantes:  - Jesús Alessandro Contreras Cucho (U20221A615)  - Chiara Kelly Oliva Cabanillas (U202112430)  - Daryl Alexander Canales Quintana (U202021033)  - André Nicolas Vásquez Castro (U20211B559)  - Renzo D' Ángelo Laguna Belleza (U20211B897)  Integrante 5 Código |



**Universidad Peruana de Ciencias Aplicadas**

Ingeniería de Sistemas y Computación | Ingeniería de Software

CC52 | Ingeniería de Software



Ciclo 2023-01

Registro de Versiones del Informe

|  |  |  |  |
| --- | --- | --- | --- |
| VERSIÓN | FECHA | AUTOR | DESCRIPCIÓN DE MODIFICACIÓN |
| TB1 | 06/04/2023 | TODOS LOS INTEGRANTES DEL GRUPO | NUEVOS PROCEDIMIENTOS, INTRODUCCIÓN DE PARTES DEL TB1. |
| TB2 | 21/04/2023 | TODOS LOS INTEGRANTES DEL GRUPO | NUEVOS PROCEDIMIENTOS, INTRODUCCIÓN DE PARTES DEL TB2. |
| TP1 | 01/05/2023 | TODOS LOS INTEGRANTES DEL GRUPO | NUEVOS PROCEDIMIENTOS, INTRODUCCIÓN DE PARTES DEL TP1. |
| TB3 | 31/05/2023 | TODOS LOS INTEGRANTES DEL GRUPO | NUEVOS PROCEDIMIENTOS, INTRODUCCIÓN DE PARTES DEL TB3. |
| TB4 | 10/06/2023 | TODOS LOS INTEGRANTES DEL GRUPO | NUEVOS PROCEDIMIENTOS, INTRODUCCIÓN DE PARTES DEL TB4. |
| TF1 | 23/06/2023 | TODOS LOS INTEGRANTES DEL GRUPO | NUEVOS PROCEDIMIENTOS, INTRODUCCIÓN DE PARTES DEL TF1. |

CONTENIDO

[Contenido 3](#_Toc98615263)

Capítulo I: Introducción……………………………………………………………………………………………………………………….7

1.1. Startup Profile………………………………………………………………………………………………………………………………7

1.1.1. Descripción de la Startup………………………………………………………………………………………………………..7

1.1.2. Perfiles de integrantes del equipo…………………………………………………………………………………………..8

1.2. Solution Profile…………………………………………………………………………………………………………………………10

1.2.1. Antecedentes y problemática………………………………………………………………………………………………11

1.2.2. Lean UX Process…………………………………………………………………………………………………………………..13

1.2.2.1. Lean UX Problem Statements…………………………………………………………………………………………13

1.2.2.2. Lean UX Assumptions………………………………………………………………………………………………………13

1.2.2.3. Lean UX Hypothesis Statements……………………………………………………………………………………..16

1.2.2.4. Lean UX Canvas………………………………………………………………………………………………………………17

Capítulo II: Requirements Elicitation & Analysis……………………………………………………………………………….19

2.1. Competidores…………………………………………………………………………………………………………………………..19

2.1.1. Análisis competitivo………………………………………………………………………………………………………..19

2.2.2. Estrategias y tácticas frente a competidores………………………………………………………………….20

2.2. Entrevistas…………………………………………………………………………………………………………………………..21

2.2.1. Diseño de entrevistas…………………………………………………………………………………………………….21

2.2.2. Registro de entrevistas………………………………………………………………………………………………….22

2.2.3. Análisis de entrevistas………………………………………………………………………………………………….28

2.3. Needfinding……………………………………………………………………………………………………………………….28

2.3.1. User Personas………………………………………………………………………………………………………………..28

Capítulo III: Database Design………………………………………………………………………………………………..28

3.1. Entities……………………………………………………………………………………………………………………………28

3.2. Atributes………………………………………………………………………………………………………………………28

3.3. Logical Data Model………………………………………………………………………………………………33

3.4. Collections……………………………………………………………………………………………………37

3.5. Schema Validation…………………………………………………………………………………………..38

3.6. Data Model Patterns…………………………………………………………………………………………43

4.1. Relational Database Management System……………………………………………………………………34

4.2. Physical Data Model………………………………………………………………………………………………35

4.3. NonRelational Database…………………………………………………………………………………………48

4.4. Queries…………………………………………………………………………………………………………………63

4.4.1. Relational Database Queries……………………………………………………………………………63

4.4.2. Non-Relational Database Queries……………………………………………………………..66

Conclusiones……………………………………………………………………………………………………………………………………33

Bibliografía……………………………………………………………………………………………………………………………………….33

Anexos……………………………………………………………………………………………………………………………..34

# Student Outcome:

|  |  |  |
| --- | --- | --- |
| Criterio especifico | Acciones realizadas | Conclusiones |
| Diseño de experimentos | **Vásquez Castro, André Nicolas**  **TF1**  Plantear las partes que faltan, como una parte de los Queries.  **Canales Quintana, Daryl Alexander**  **TF1**  Plantear las partes que faltan, como otra parte de los Queries.  **Laguna Belleza, Renzo D’ Angelo**  **TF1**  Plantear las partes que faltan, como el Relational Database Queries.  **Oliva Cabanilla, Chiara Kelly**  **TF1**  Plantear las partes que faltan, como una parte del Non-Relational Database Queries.  **Contreras Cucho, Jesús Alessandro**  **TF1**  Plantear las partes que faltan, como otro parte del Non-Relational Database Queries. | El planteamiento del desarrollo de todas las correcciones y nuevas cosas de los trabajos del TF1. |
| Desarrollo de experimentos | **Vásquez Castro, André Nicolas**  **TF1**  Desarrollar una parte del Queries del TF1.  **Canales Quintana, Daryl Alexander**  **TF1**  Desarrollar otra parte del Queries del TF1.  **Laguna Belleza, Renzo D’ Angelo**  **TF1**  Desarrollar el Relational Database Queries del TF1.  **Oliva Cabanilla, Chiara Kelly**  **TF1**  Desarrollar una parte del Non-Relational Database Queries del TF1.  **Contreras Cucho, Jesús Alessandro**  **TF1**  Desarrollar otra parte del Non-Relational Database Queries del TF1. | El desarrollo de todos estos procesos o requisitos del TF1 es muy bueno para que la aplicación se desarrolle correctamente. |
| Análisis e interpretación de datos/resultados | **Vásquez Castro, André Nicolas**  **TF1**  Se ha podido realizar una parte de los Queries del TF1.  **Canales Quintana, Daryl Alexander**  **TF1**  Se ha podido realizar otra parte de los Queries del TF1  **Laguna Belleza, Renzo D’ Angelo**  **TF1**  Se ha podido realizar el Relational Database Queries del TF1.  **Oliva Cabanilla, Chiara Kelly**  **TF1**  Se ha podido realizar una parte del Non-Relational Database Queries del TF1.  **Contreras Cucho, Jesús Alessandro**  **TF1**  Se ha podido realizar otra parte del Non-Relational Database Queries del TF1. | Todo lo desarrollado del TF1 es un gran avance con nuestra aplicación, de esta forma vamos a hacer las cosas bien con la aplicación. |

CHAPTER

1

Ciclo 2018-02 | Sección XXXX

# Capítulo 1: Introducción

## *Startup Profile*

## Somos la empresa “Gas-Car”, un grupo de estudiantes de la Universidad Peruana de Ciencias Aplicadas, buscamos solucionar uno de los problemas que tiene la personas cada día de su vida: ¿Dónde podemos acudir si se nos acabó el combustible y no hay gasolineras cerca?, estos hechos suelen ocurrir a menudo, hasta es poco probable de que otras personas con sus vehículos puedan ayudarte dándoles de sus gasolinas, lo cual, puede ser incomodo o hasta un momento de desesperación para la mayoría de personas. Para ello, solucionaremos este problema a través de una app que ayudará en la solicitud de vehículos que contienen cualquier tipo de combustible aptos para necesidad de cada vehículo.

### *Descripción de la Startup:*

Nuestro producto llamado “LlenaFast”, se basa mayormente en envíos de combustibles a todos los vehículos en funcionamiento para que, con nuestra aplicación, podamos ayudar a nuestros clientes quitando las preocupaciones de que si su vehículo se queda o no sin combustible suficiente o la preocupación de no encontrar una gasolinera cerca durante sus destinos. Para ello, le brindamos un servicio de mayor calidad con LlenaFast para tener mayor disposición a nuestros usuarios.

Mano sosteniendo un celular en la mano

Descripción generada automáticamente con confianza media

### *Perfil de los Integrantes:*

1. André Nicolas Vásquez Castro - Ingeniería de Sistemas de Información

Mi nombre es André Nicolas Vásquez Castro, nací en el año 2004 en Lima y tengo 18 años. Soy un estudiante de la carrera de Ingeniería de Sistemas de Información. He terminado mis estudios escolares en el colegio Cruz Saco de Los Olivos. Estoy estudiando la carrera de Ingeniería de Sistemas de Información porque me encanta el tema de la tecnología, el funcionamiento de los sistemas y demás. Estoy dispuesto a ayudar a mi grupo con el trabajo y resolver problemas que se presenten en el transcurso de todo este ciclo.



1. Daryl Alexander Canales Quintana – Ingeniería de Sistemas de Información

Mi nombre es Daryl Alexander Canales Quintana, actualmente estoy cursando el 6to ciclo de la carrera de Ingeniería de Sistemas de Información, tengo 21 años.

1. Renzo D’ Angelo Laguna Belleza – Ingeniería de Sistemas de Información

Mi nombre es Renzo D’Angelo Laguna Belleza, tengo 19 años y actualmente estoy cursando el 5to ciclo de la carrera de Ingeniería de Sistemas de Información. Me considero una persona en su mayoría responsable, siempre estoy dispuesto a ayudar con los trabajos en grupo y ayudar a resolver cualquier problema que aparezca durante el transcurso de este.

1. Chiara Kelly Oliva Cabanillas – Ingeniería de Sistemas de Información

Me llamo Chiara Kelly Oliva Cabanillas, soy una estudiante de Ingeniería de Sistemas de Información, estoy cursando el 4to ciclo de la carrera. Actualmente tengo 18 años y me considero una persona responsable que siempre busca apoyar en los proyectos con el objetivo de poder lograr buenos resultados.

1. Un joven sonriendo con una playera de color gris

   Descripción generada automáticamenteJesús Alessandro Contreras Cucho – Ingeniería de Sistemas

Mi nombre es Jesús Contreras, tengo 18 años, nací en la ciudad de Lima, Perú, un 29 de diciembre del 2004. Hoy en día curso la carrera de Ingeniería de Sistemas de Información en la Universidad Peruana de Ciencias Aplicadas (UPC).

## *Solution Profile*

*LlenaFast:*

Nuestra plataforma mostrará los combustibles disponibles dentro de su zona, permitiéndole escoger a su elección cuál mejor le convenga. Y luego permitirá escoger el transporte que este mismo desee. Esta aplicación facilitará el transporte de gasolina en caso de emergencias, así evitando que un auto se quede varado en medio de la carretera.

*Diagrama, Dibujo de ingeniería

Descripción generada automáticamente*

### *Antecedentes y problemática:*

**Técnica 5W-2H:**

En el presente contexto la aplicación de la técnica de las 5W y 2H está orientada a definir cuál es la problemática a la que estará enfocada el startup del equipo. A través de un listado de preguntas iremos resolviendo las interrogantes más importantes presentes en el trabajo.

1. **What (Qué) ¿Qué necesitaremos para implementarlo?**

Necesitaremos recabar información sobre la frecuencia de la falta de combustible durante el viaje de carretera de los vehículos. Esto nos brindará un panorama.

1. **When (Cuándo) ¿Cuándo se aplicará la solución?**

Cuando a algún conductor le sea necesario gasolina recurrirá a nosotros.

1. **Who (Quién) ¿Quién es el beneficiado por la propuesta?**

Los transportistas en general que realizan largos viajes, como los cortos, ya que la falta de gasolina le puede ocurrir a cualquiera del sector transporte.

1. **Why (Por qué) ¿Por qué es nuestro objetivo?**

Porque así conseguiremos que haya mejor servicio en el sector transporte asegurándonos de ser un medio de respaldo ante la falta de gasolina.

1. **How (Cómo) ¿Cómo nos diferenciaremos de la competencia?**

Facilitando la entrega de combustible por delivery de escala vehicular, debido a que no existen competidores exactos para esta rama.

1. **How Much (Cuánto) ¿Cuántas empresas compiten contra nosotros?**

En este rubro tenemos identificado a “Petroleum Service” y “Petrotransp” las cuales brindan un delivery de combustible, pero no con el enfoque que nosotros brindamos una entrega descentralizada de combustible.

Edificio con letrero en la calle

Descripción generada automáticamente con confianza media

### *Lean UX Process:*

#### *Lean UX Problem Statements:*

#### El problema que se tiene como objetivo resolver son las dificultades que pueden surgir con el tiempo del trayecto del delivery, así como las consecuencias que pueden tener las personas al incomodarse por el tiempo de espera y cancelar el pedido. En el Perú, suelen surgir muchos problemas con respecto al tránsito de los vehículos, junto a ellos encontramos los tráficos masivos en las carreteras, lo cual, es problemático también con el gasto del combustible.

#### Estos hechos ocurren en el día a día, como quedarse varado en las pistas o trasladar los vehículos empujándolos hasta una gasolinera que posiblemente pueda estar cerca o no, lo cual, genera una incomodidad a uno mismo. Frente a esta problemática, llegamos a la pregunta:

#### ¿Cómo podemos satisfacer a nuestro cliente con nuestro servicio para reducir el tiempo de envío de combustible al lugar solicitado?

#### *Lean UX Assumptions:*

* Creo que mis clientes necesitan envíos de combustible para no tener preocupación de buscar una gasolinera.
* Estas necesidades se pueden resolver mediante los envíos a autos en movimiento.
* Mis clientes iniciales son mujeres y hombres mayores de 18 que conducen y desean no perder tiempo.
* El valor más importante que un cliente quiere de mi servicio es que sea rápido y eficiente.
* El cliente también puede obtener beneficios adicionales mediante una membresía premium.
* Voy a adquirir la mayoría de los clientes a través del registro de clientes cuando descarguen la app.
* Haré dinero mediante la publicidad a marcas de gasolina en la app, el costo de envío y con las membresías premium.
* Mi competencia principal en el mercado serán otras gasolineras con un servicio rápido.
* Las venceremos debido a la rapidez de nuestros envíos, la inexistencia de cola y los diversos métodos de pago.
* Mi mayor riesgo de servicio es que el pedido demore en llegar a la ubicación del cliente.
* Resolveremos esto a través de la capacitación de nuestros conductores y la muestra del tiempo de llegada a nuestros clientes.

1. **¿Quién es el usuario?**

Conductores de 18 años a más que desean reducir el tiempo al recargar combustible.

1. **¿Dónde encaja nuestro servicio en su trabajo o vida?**

En su vida al momento de conducir, pero ello también involucra el ir al trabajo o trabajar conduciendo.

1. **¿Qué problemas tiene nuestro producto y cómo se pueden resolver?**

Un problema podría ser el tiempo de llegada, ya que el auto está en movimiento, pero se puede solucionar mediante trabajadores en toda la ciudad y con rutas claras para llegar de manera rápida al cliente.

1. **¿Cuándo y cómo es usado nuestro producto?**

Al momento de conducir y ver que no queda mucho combustible, mediante una parada se puede usar de manera rápida.

1. **¿Qué características son importantes?**

Que tenga una interfaz de fácil interacción para que sea aún más rápido el servicio, también que llegue de manera rápida a nuestro cliente.

1. **¿Cómo debe verse nuestro producto y cómo debe comportarse?**

Debe ser de interfaz de fácil interacción y tener un tiempo de respuesta rápido.

**Features:**

* Hacer uso de atajos para hacer que el pedido habitual de un usuario se pueda ejecutar más rápido.
* Hacer que el usuario pueda tener guardados sus métodos de pago para aumentar la rapidez del servicio.
* Poseer dos versiones, una gratuita y una premium, que brinde el retiro de anuncios y quite los costos de envío. Esto permitirá generar ingresos y le dará la posibilidad al usuario de elegir.
* Hacer que la app tenga la opción de programar pedidos para que tener su ubicación y poder tener todo preparado para que su servicio se vea agilizado.

**Business Outcomes:**

* Brindar un medio para reducir el tiempo de recarga de combustible.
* Agilizar el proceso de recarga de combustible con nuestro envío rápido y gasolina de buena calidad.
* Aumentar el número de descargas de nuestro aplicativo.
* Los usuarios se suscriben mediante la membresía premium.
* Generar ingresos mediante la publicidad de marcas de gasolina en la aplicación.
* Popularizar la app para que sea usada por la mayoría de los conductores.
* Reducir el tiempo de recarga de combustible a 15 minutos de llegada con el carro en movimiento.

**User Benefits:**

* Obtener gasolina mientras está en movimiento.
* Reducir el tiempo de recarga de combustible.
* Eliminar la preocupación por buscar una gasolinera.

#### *Lean UX Hypothesis Statements:*

* **Hypothesis 1:**

Creemos que capacitar a nuestros conductores, ayudará a reducir el tiempo de llegada a la ubicación del usuario.

Sabremos que hemos tenido éxito.

Cuando veamos que el tiempo de llegada mostrado en la app sea reducido en un 20%.

* **Hypothesis 2:**

Creemos que incluir publicidad de marcas reconocidas de gasolinas en la aplicación y membresías premium, ayudará a generar más ingresos

Sabremos que hemos tenido éxito.

Cuando veamos que las ganancias de la app hayan aumentado en un 40%.

* **Hypothesis 3:**

Creemos que mejorar la rapidez y eficiencia de nuestro servicio, ayudará a que tengamos más reconocimiento.

Sabremos que esto es cierto.

Cuando veamos que somos la aplicación de delivery de gasolina más reconocida del Perú.

* **Hypothesis 4:**

Creemos que generar un autoguardado de pedidos habituales y ubicación del usuario, ayudará a aumentar la rapidez del uso del servicio.

Sabremos que hemos tenido éxito.

Cuando veamos que el cliente se sienta satisfecho con la rapidez de la interfaz de la aplicación.

#### *Lean UX Canvas*

|  |  |  |
| --- | --- | --- |
| **Business problem** El problema que se tiene como objetivo resolver son las dificultades que pueden surgir con el tiempo del trayecto del delivery, así como las consecuencias que pueden tener las personas al incomodarse por el tiempo de espera y cancelar el pedido. En el Perú, suelen surgir muchos problemas con respecto al tránsito de los vehículos, junto a ellos encontramos los tráficos masivos en las carreteras, lo cual, es problemático también con el gasto del combustible.Estos hechos ocurren en el día a día, como quedarse varado en las pistas o trasladar los vehículos empujándolos hasta una gasolinera que posiblemente pueda estar cerca o no, lo cual, genera una incomodidad a uno mismo. Frente a esta problemática, llegamos a la pregunta:¿Cómo podemos satisfacer a nuestro cliente con nuestro servicio para reducir el tiempo de envío de combustible al lugar solicitado? | **Solution ideas**   * App que permita hacer uso de atajos para hacer que el pedido habitual de un usuario se pueda ejecutar más rápido. * Hacer que la app tenga la opción de programar pedidos para que tener su ubicación y poder tener todo preparado para que su servicio se vea agilizado. | **Business Outcomes**   * Brindar un medio para reducir el tiempo de recarga de combustible. * Agilizar el proceso de recarga de combustible con nuestro envío rápido y gasolina de buena calidad. * Aumentar el número de descargas de nuestro aplicativo. * Los usuarios se suscriben mediante la membresía premium. * Generar ingresos mediante la publicidad de marcas de gasolina en la aplicación. * Popularizar la app para que sea usada por la mayoría de los conductores. * Reducir el tiempo de recarga de combustible a 15 minutos de llegada con el carro en movimiento. |
| **User & Customers**  Personas que son transportistas en general (edad mínima de 18 años) de género masculino y femenino. |  | **User Benefits**   * Obtener gasolina mientras está en movimiento * Reducir el tiempo de recarga de combustible * Eliminar la preocupación por buscar una gasolinera |

## *Segmentos Objetivo:*

## Personas que son transportistas en general (edad mínima de 18 años) de género masculino y femenino.

# Capítulo 2: Requirements Elicitation & Analysis

CHAPTER

2

## *Requirements Elicitation & Analysis*

## *Competidores*

### *Análisis competitivo:*

Vamos a analizar la descripción de nuestra aplicación y la descripción de la aplicación de nuestro competidor:

### Facilito: Aplicativo que te permite obtener combustible al mejor precio, la aplicación también te permite ubicar los grifos que están a tu alrededor o en el destino que elijas, con sus respectivos actualizados.

### LlenaFast: Aplicativo diseñado para Android y iOS que te permite llenar tu vehículo con combustible sin la necesidad de ir a una gasolinera, quitando todas las preocupaciones de los conductores de que si su vehículo se queda sin combustible.

### En las dos descripciones de las aplicaciones que se han analizado, se puede observar características similares entre las dos aplicaciones, pero existen diferencias como que nuestra aplicación llenará combustible a todos los vehículos sin la necesidad de ir a una gasolinera. En un futuro iremos agregando más características interesantes a la aplicación para que pueda hacer frente al competidor y dar satisfacción a nuestros usuarios.

### 

### *Estrategias y tácticas frente a competidores*

Vamos a aplicar diferentes estrategias y tácticas para hacerle frente a nuestros competidores, por ejemplo:

1. Nuestra aplicación será gratis, rápida, y pedirá poco espacio y requisitos para que cualquier usuario pueda descargarlo y usarlo sin problemas.
2. Agregaremos más actualizaciones de características nuevas de seguridad y rendimiento a nuestra aplicación.
3. Haremos publicidad de la aplicación en redes sociales y en las calles para ganar más usuarios.

## *Entrevistas*

### *Diseño de entrevistas*

### Formato para Conductores:

### 1. ¿Cuál es tu nombre, qué edad tienes y a qué te dedicas?

### 2. ¿Cuánto tiempo lleva trabajando como conductor?

### 3. ¿Qué horario tiene de trabajo?

### 4. ¿Cuánto viaja le suele faltar algo para su vehículo?

### 5. ¿Qué tan frecuente se le suele acabar la gasolina en un viaje? Y ¿En qué ocasiones ocurre?

### 6. ¿Lleva gasolina en su vehículo por precaución?

### 7. ¿Cuánta gasolina suele necesitar su vehículo?

### - El entrevistador explica acerca de la aplicación.

### 8. ¿Estaría dispuesto a probar nuestra aplicación? ¿Por qué?

### 9. ¿Qué cree que podría mejorar o añadir a nuestra propuesta?

### *Registro de entrevistas*

**USUARIO 1:**

**Enlace:**

<https://drive.google.com/drive/folders/1Hn97YcfN_jQPrkyvOEHTdG8m6g2hXoar?lfhs=2>

### **A. ¿Cuál es tu nombre, qué edad tienes y a qué te dedicas?**

     Se llama Fabián, tiene 22 años y se desempeña como conductor.

### **B. ¿Cuánto tiempo lleva trabajando como conductor?**

  Nos comenta, que actualmente lleva manejando 4 meses.

### **C. ¿Qué horario tiene de trabajo?**

 Suele trabajar de martes a sábado a partir de las 9 pm.

### **D. ¿Cuánto viaja le suele faltar algo para su vehículo?**

Por lo general no le hace falta nada, porque conduce en lugares más cercanos de su casa.

### **E. ¿Qué tan frecuente se le suele acabar la gasolina en un viaje? Y ¿En qué ocasiones ocurre?**

### Se le suele acabar la gasolina cuando realiza viajes largos por carreteras y cuando conduce por áreas remotas.

### **F. ¿Lleva gasolina en su vehículo por precaución?**

Suele llevar gasolina extra en su vehículo en caso de emergencia, o a veces cuando va conduciendo en áreas remotas donde no se encuentran estacionamientos de servicios cercanos.

### **G. ¿Cuánta gasolina suele necesitar su vehículo?**

### - El entrevistador explica acerca de la aplicación.

### - Gasta alrededor de 3 galones de gasolina diarios.

### **H. ¿Estaría dispuesto a probar nuestra aplicación? ¿Por qué?**

Si estuviera dispuesto, ya que nos comenta que es una propuesta interesante y que no es tan común en otros sitios por lo que sería una buena oportunidad de probar algo nuevo y ver cómo podría mejorar su experiencia como conductor.

**USUARIO 2:**

**Enlace:**

<https://drive.google.com/file/d/1FcUmzlULEK0RC_4ILJ7NsQQVcv5uVu-a/view?usp=sharing>

1. **¿Cuál es tu nombre, tu edad y a que te dedicas?**

Mauricio Lamas, tiene 35 años y se dedica a ser chofer.

1. **¿Cuánto tiempo llevas trabajando como conductor?**

Como chofer lleva 10 años y medio, pero conduciendo 15 años.

1. **¿Cuánto tiempo solías trabajar?**

Semanalmente entre 40 a 45 horas o algunas veces 50, diciendo que suele variar el tiempo, pero siempre en un promedio de 40 a 45 horas.

1. **¿En algún momento usted se llegó a quedar sin combustible en medio de la carretera o estuvo cerca de quedar sin esta?**

Menciona que alguna vez estuvo a punto de quedarse sin combustible durante sus trayectos, esto debido al tráfico de las calles y que varias veces lograba llegar a los grifos usando el tanque de reserva.

1. **¿Suele llevar combustible de repuesto en su vehículo para casos de emergencia?**

 Al inicio menciona que, si solía llegar combustible de repuesto, pero actualmente no debido a que los pasajeros solían quejarse del olor que esta emite. Por eso siempre intenta tener el tanque lleno, aunque muchas veces se le llega a olvidar.

1. **¿Cuánto suele gastar en combustible para el vehículo?**

El entrevistado menciona que, dependiendo la temporada o el mes, mencionando que en promedio la gasta 100 a 150 soles a la semana, pero cuando hace mucho recorrido puede llegar hasta los 200 soles.

1. **¿Estaría dispuesto a probar nuestra aplicación? ¿Por qué?**

El entrevistado está interesado en probar la aplicación, mencionando que puede funcionar para casos de emergencia, porque como menciono esta una experiencia horrible quedarse sin combustible y no tener un grifo cerca.

1. **¿Qué cree que se podría mejorar o añadir a nuestra propuesta?**

Mencionan que sería bueno que los deliverys no solo se concentren en el combustible, sino que también en otros productos básicos o necesarios para el vehículo como el anticongelante, aceite, bombillas para los faros, etc.

**USUARIO 3:**

**Enlace:**

<https://youtu.be/UV4UL_VmXqU>

**A. ¿Cuál es tu nombre, tu edad y a que te dedicas?**

Andrés Valdez, tiene 25 años y se dedica a ser chofer.

**B.  ¿Cuánto tiempo llevas trabajando como conductor?**

Como chofer lleva 5 años trabajando

**C. ¿Cuánto tiempo solías trabajar?**

Normalmente trabajaba en las mañanas y por las noches, 6 dias a la semana

1. **¿En algún momento usted se llegó a quedar sin combustible en medio de la carretera o estuvo cerca de quedar sin esta?**

Menciona que algunas veces se quedaba sin gasolina debido al trayecto tan largo que recorría con su vehículo.

1. **¿Suele llevar combustible de repuesto en su vehículo para casos de emergencia?**

Menciona que, si suele llevar gasolina aparte, para no estar gastando demás en alguna gasolinera y sea mucho más rápido el proceso.

1. ¿**Cuánto suele gastar en combustible para el vehículo?**

100 a 125 soles diario en gasolina es lo que menciona nuestro entrevistado, que gasta diariamente en su trayecto.

1. **¿Estaría dispuesto a probar nuestra aplicación? ¿Por qué?**

El entrevistado está interesado en probar la aplicación, por la razón de que de esa forma ahorraría la preocupación de estar buscando un punto de reabastecimiento de combustible.

1. **¿Qué cree que se podría mejorar o añadir a nuestra propuesta?**

Menciona que sería bueno aplicar un precio cómodo y accesible para los usuarios que estén dispuestos a usar esta aplicación.

### *Análisis de entrevistas*

### Nuestros entrevistados se encuentra entre las edades 18-22 años de los cuales conducen por sus diferentes necesidades que estos poseen, llevan conduciendo menos de 2 años.

### Los entrevistados responden a las preguntas diseñadas por nosotros para saber lo que ellos dicen, es necesario hacer todas estas preguntas para poder mejorar nuestra aplicación y saber las necesidades de los entrevistados.

## *Needfinding*

### *User Personas*

### 

# Capítulo 3: DATA BASE DESIGN

3

CHAPTER

### ***3.1.*** *Entities*

**Entidades de LlenaFast:**

* **Entidad Usuario:** Esta es una entidad que representa a todos los usuarios que requieran acceder a nuestra aplicación.
* **Entidad Financiera:** Esta es una entidad que representa todos los registros contables de la aplicación LlenaFast
* **Entidad Organización:** Esta es una entidad que representa toda la organización de LlenaFast.
* **Entidad Lista\_Usuario:** Esta es una entidad que representa toda la lista de usuarios que se registraron en la aplicación.
* **Entidad Personal:** Esta es una entidad que representa a todos los personales que requieran acceder a la aplicación.
* **Entidad Lista\_Personal:** Esta es una entidad que representa toda la lista de personales que se registraron en la aplicación.
* **Entidad Producto:** Esta es una entidad que representa todos los productos que se registraron en la aplicación.
* **Entidad Lista\_Producto:** Esta es una entidad que representa toda la lista de productos que se registraron en la aplicación.
* **Entidad Comprar\_Producto:** Esta es una entidad que representa los pasos para comprar un producto en la aplicación.
* **Entidad Historial:** Esta es una entidad que representa todo el historial de todos los productos que fueron comprados por el usuario en la aplicación.
* **Entidad Pedido:** Esta es una entidad que representa todos los productos que fueron seleccionados para pedidos.
* **Entidad Ubicación:** Esta es una entidad que representa la ubicación del usuario en tiempo real para la entrega de los productos de la aplicación.

### ***3.2.*** *Atributes*

### **Atributos de las entidades creadas de LleneFast:**

### **Atributos de Usuario:**

### **[PK] ID\_Usuario:** Es el código único que identifica a cada instancia de la entidad, al ser la llave primaria.

### **Nombre\_Usuario:** Es el nombre real del usuario que desee revisar la aplicación.

### **Apellido\_Usuario:** Es el apellido real del usuario que desee revisar la aplicación.

### **Edad\_Usuario:** Es la edad real del usuario que desee revisar la aplicación

### **Teléfono\_Usuario:** Es el teléfono real del usuario que desee revisar la aplicación

### **FechaNacimiento\_Usuario:** Es la fecha de nacimiento del usuario que desee revisar la aplicación

### **DNI\_Usuario:** Es el número de DNI del usuario que desee revisar la aplicación

### **Distrito\_Usuario:** Es el distrito donde vive el usuario.

### **Atributos de Financiera:**

### **[PK] Id\_Financiera:** Es el código único que se usa para identificar a cada instancia de la entidad.

### **RegistroContables\_Financiera:** Se va a poder verificar todos los fondos que presenta la empresa en total.

### **Atributos de Organización:**

### **[PK] ID\_Organización:** Es el código único que se usa para identificar a cada instancia de la entidad.

### **Nombre\_Organización:** Aquí se registra el nombre de la organización.

### **Socios\_Organización:** Aquí se registra a todos los socios de la aplicación.

### **Atributos de Lista\_Usuario:**

### **[PK] ID\_Lista\_Usuario:** Es el código único de Lista Usuario que se usa para identificar a cada instancia de la entidad.

### **FechaRegistro\_Lista\_Usuario:** Aquí se va a presentar la fecha en que se registró por primera vez el usuario en la aplicación.

### **[FK] Usuario\_ID:** Visualización de los usuarios mediante el ID de la lista de usuarios.

### **Atributos de Personal:**

### **[PK] ID\_Personal:** Es el código único que identifica a cada instancia de la entidad, al ser la llave primaria.

### **Nombre\_Personal:** Es el nombre real del personal que desee revisar la aplicación.

### **Apellido\_Personal:** Es el apellido real del personal que desee revisar la aplicación.

### **Edad\_Personal:** Es la edad real del personal que desee revisar la aplicación

### **Teléfono\_Personal:** Es el teléfono real del personal que desee revisar la aplicación

### **FechaNacimiento\_Personal:** Es la fecha de nacimiento del personal que desee revisar la aplicación

### **DNI\_Personal:** Es el número de DNI del personal que desee revisar la aplicación

### **Distrito\_Personal:** Es el distrito donde vive el personal.

### **Atributos de Lista\_Personal**

### **[PK] ID\_Lista\_Personal:** Es el código único de Lista Personal que se usa para identificar a cada instancia de la entidad.

### **FechaRegistro\_Lista\_Personal:** Aquí se va a presentar la fecha en que se registró por primera vez el personal en la aplicación.

### **[FK] Personal\_ID:** Visualización del personal mediante el ID de la lista de personal.

### **Atributos de Producto**

### **[PK] ID\_Producto:** Es el código único de Producto que se usa para identificar a cada instancia de la entidad.

### **Precio\_Producto:** Aquí se va a presentar el precio del producto en la aplicación.

### **Marca\_Producto:** Aquí se va a presentar la marca del producto en la aplicación.

### **Nombre\_Producto:** Aquí se va a poder observar el nombre del producto que el usuario quiera comprar en la aplicación.

### **Cantidad\_Producto:** Aquí se va a presentar la cantidad de productos que el usuario quiera en la aplicación.

### **Descripción\_Producto:** Aquí se va a poder observar la descripción del producto en la aplicación.

### **Atributos de Lista\_Producto**

### **[PK] ID\_Lista\_Producto:** Es el código único de Lista Producto que se usa para identificar a cada instancia de la entidad.

### **VecesComprada\_Lista\_Producto:** Aquí se va a presentar las veces que fue comprado el producto en la aplicación.

### **FechaComprada\_Lista\_Producto:** Aquí se va a presentar la fecha en que fue comprado el producto en la aplicación.

### **TotalPagado\_Lista\_Producto:** Aquí se va a poder observar el total pagado del producto en la aplicación.

### **DirecciónEntregado\_Lista\_Producto:** Aquí se va a presentar la dirección en que fue entregado el producto en la aplicación.

### **[FK] Personal\_ID:** Visualización del personal mediante el ID de la lista de producto.

### **[FK] Producto\_ID:** Visualización del producto mediante el ID de la lista de producto.

### **Atributos de Comprar\_Producto**

### **[PK] ID\_Comprar\_Producto:** Es el código único de Comprar Producto que se usa para identificar a cada instancia de la entidad.

### **Nombre\_Comprar\_Producto:** Aquí se va a llenar el nombre del usuario para comprar el producto en la aplicación.

### **Teléfono\_Comprar\_Producto:** Aquí se va a llenar el teléfono del usuario para comprar el producto en la aplicación.

### **MétodoPago\_Comprar\_Producto:** Aquí se va a seleccionar el método de pago para comprar el producto en la aplicación.

### **CódigoPostal\_Comprar\_Producto:** Aquí se va a presentar la dirección en que fue entregado el producto en la aplicación.

### **[FK] Ubicación\_ID:** Visualización de la ubicación mediante el ID de la compra del producto.

### **[FK] Producto\_ID:** Visualización del producto mediante el ID de la compra del producto.

### **Atributos de Historial**

### **[PK] ID\_Historial:** Es el código único de Historial que se usa para identificar a cada instancia de la entidad.

### **HoraPedido\_Pedido:** Aquí se va a poder observar la hora en que se ha pedido el producto en la aplicación.

### **CantidadCombustible\_Pedido:** Aquí se va a poder observar la cantidad de combustible que se ha pedido en la aplicación.

### **EstadoPedido\_Pedido:** Aquí se va a poder observar el estado del pedido del producto en la aplicación.

### **[FK] Usuario\_ID:** Visualización del usuario mediante el ID del historial.

### **Atributos de Pedido**

### **[PK] ID\_Pedido:** Es el código único de Pedido que se usa para identificar a cada instancia de la entidad.

### **FechaPedido\_Pedido:** Aquí se va a poder ver la fecha del pedido en la aplicación.

### **HoraPedido\_Pedido:** Aquí se va a poder observar la hora en que se ha pedido el producto en la aplicación.

### **CantidadCombustible\_Pedido:** Aquí se va a poder observar la cantidad de combustible que se ha pedido en la aplicación.

### **EstadoPedido\_Pedido:** Aquí se va a poder observar el estado del pedido del producto en la aplicación.

### **Atributos de Ubicación**

### **[PK] ID\_Ubicación:** Es el código único de Ubicación que se usa para identificar a cada instancia de la entidad.

### **NombreUbicación\_Ubicación:** Aquí se va a poder llenar el nombre de la ubicación en la aplicación.

### **DirecciónUbicación\_Ubicación:** Aquí se va a poder llenar la dirección de la ubicación en la aplicación.

### ***3.3.*** *Logical Data Model*

### **Relaciones:**

### **Lista\_Usuario a muchos Usuario:** Quiere decir que hay una lista de usuario para todos los usuarios de la aplicación.

### **Lista\_Personal a muchos Personal:** Quiere decir que hay una lista de personal para todo el personal de la aplicación.

### **Usuario a muchos Pedido:** Quiere decir que hay un usuario que puede tener varios o muchos pedidos de la aplicación.

### **Lista\_Producto a muchos Producto:** Quiere decir que hay una lista de productos para todos los productos que el usuario ha pedido.

### **Compra\_Producto a muchos Producto:** Quiere decir que se puede realizar una compra de producto a muchos productos con tan solo agregar al carrito a varios productos de la aplicación.

### **Pedido a muchos Compra\_Producto:** Quiere decir que se genera un pedido cuando se compra varios productos que se agregaron al carrito de la aplicación.

### **Historial a muchos Usuario:** Quiere decir que hay un historial para cada usuario que se registre en la aplicación.

### **Ubicación a muchos Personal:** Quiere decir que hay un registro de ubicación por cada personal de la aplicación.

### **Ubicación a muchos Usuario:** Quiere decir que hay un registro de ubicación por cada usuario de la aplicación.

### **Organización a muchos Personal:** Quiere decir que en una organización hay muchos personales contratados para atender la aplicación.

### **Organización a muchos Lista\_Personal:** Quiere decir que en una organización hay muchas listas de personales de la aplicación.

### **Organización a muchos Financiera:** Quiere decir que en una organización hay varios estados de cuenta de la aplicación.

### **Organización a muchos Lista\_Usuario:** Quiere decir que en una organización hay muchas listas de usuario de la aplicación.

### **Organización a muchos Usuario:** Quiere decir que en una organización hay varios usuarios que quieren utilizar la aplicación.

### **Organización a muchos Productos:** Quiere decir que en una organización hay muchos productos de la aplicación.

### **Organización a muchos Lista\_Productos:** Quiere decir que en una organización hay muchas listas de productos de la aplicación.

### **Organización a muchos Compra\_Producto:** Quiere decir que en una organización hay muchas compras de productos de la aplicación.

### **Organización a muchos Pedido:** Quiere decir que en una organización hay muchos pedidos de la aplicación.

### **Organización a muchos Ubicación:** Quiere decir que en una organización hay muchas ubicaciones de usuarios de la aplicación.

### **Organización a muchos Historial:** Quiere decir que en una organización hay muchos historiales de usuarios de la aplicación.

### **Financiera a muchos Usuario:** Quiere decir que en un estado de cuenta se podrá visualizar las ganancias por las compras de gasolina de varios usuarios en la aplicación.

### **Financiera a muchos Personal:** Quiere decir que en un estado de cuenta se podrá visualizar todo lo que se debe pagar por los personales contratados de la aplicación.

### **Financiera a una Organización:** Quiere decir que todos los registros contables pertenecen y se guarda en la empresa Gas-Car.

### 

### **Enlace del LucidChart:** [**https://lucid.app/lucidchart/747e596d-cb4c-406d-aa35-480b60fab2cd/edit?viewport\_loc=66%2C154%2C2560%2C1216%2C0\_0&invitationId=inv\_7e225b94-77b0-492f-99d0-b1c3809f7712**](https://lucid.app/lucidchart/747e596d-cb4c-406d-aa35-480b60fab2cd/edit?viewport_loc=66%2C154%2C2560%2C1216%2C0_0&invitationId=inv_7e225b94-77b0-492f-99d0-b1c3809f7712)

### ***3.4.*** *Collections*

### Hemos decidido realizar las siguientes colecciones en el MongoDB, debido a que se está considerando manejar información importante para que se realice la base de datos no relacional.

### 

### 

### Tabla Descripción generada automáticamente

### Aquí se tiene 6 colecciones a diferencia del Physical Data Model, la razón es simple, y es que se está usando el patrón embedded document para que se pueda fusionar algunas de las tablas que se ha creado anteriormente. Por ejemplo:

### En el primer caso se ha fusionado las tablas Productos, Lista\_Producto y Comprar\_Producto por pertenecer al dominio de “productos” y se convierte en colección.

### En el segundo caso pasa lo mismo, se ha fusionado las tablas Usuario, Lista\_Usuario y Pedido por pertenecer al dominio de “usuarios” y se convierte en colección.

### En el tercer caso se ha fusionado las tablas Personal y Lista\_Personal por pertenecer al dominio de “personal” y se convierte en colección.

### Por último, se ha fusionado las tablas Organización y Financiera por pertenecer al dominio de “entidades” y se convierte en colección.

### Diagrama, Esquemático Descripción generada automáticamente

### ***3.5.*** *Schema Validation*

### Estos son nuestros Schema Validation para nuestras Colecciones:

### - Entidades:

### 

### Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico Descripción generada automáticamente

### - Historial:

### Captura de pantalla de computadora Descripción generada automáticamente

### Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico Descripción generada automáticamente

### - Personal:

### Captura de pantalla de computadora Descripción generada automáticamente

### Texto Descripción generada automáticamente

### Interfaz de usuario gráfica, Texto, Correo electrónico Descripción generada automáticamente

### Texto, Aplicación Descripción generada automáticamente

### - Productos:

### Captura de pantalla de computadora Descripción generada automáticamente

### 

### Texto Descripción generada automáticamente

### Texto Descripción generada automáticamente

### Interfaz de usuario gráfica, Texto, Aplicación Descripción generada automáticamente

### Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico Descripción generada automáticamente

### - Ubicación:

### Captura de pantalla de computadora Descripción generada automáticamente

### Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico Descripción generada automáticamente

### - Usuarios:

### Captura de pantalla de computadora Descripción generada automáticamente

### Texto Descripción generada automáticamente

### Texto Descripción generada automáticamente

### Interfaz de usuario gráfica, Texto, Aplicación Descripción generada automáticamente

### Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico Descripción generada automáticamente

### ***3.6.*** *Data Model Patterns*

### Según la situación de organizar los documentos en sus colecciones.

### Se ha predominado el uso de:

### Model one-to-one relationships with embedded documents:

### **Embedded document pattern:**

### Este patrón se utilizó debido a su eficacia en la reducción del número de operaciones de lecturas requeridas para recuperar los datos. Se puede presenciar este patron en varias colecciones, como la colección Personal, colección Usuarios, colección Entidades y colección Productos.

### **Subset pattern:**

### Esto se usa en el trabajo para simplificar algunos inconvenientes del uso del patrón anterior y evitar implementar colecciones innecesarias que ocupan más espacio.

### 

# Capítulo 4: DATABASE IMPLEMENTATION

4

CHAPTER

### ***4.1.*** *Relational Database Management System*

### - Se utilizó el motor de base de datos Microsoft SQL versión 19.0. Estas importantes razones son:

### - SQL mejora la información personal de gestión a gestión de datos del cliente. Una base de datos gestiona la información almacenada. Se convierte en un activo importante de la empresa.

### ✓ Mejora la transferencia de bases de datos y la agilidad de recopilación.

### ✓ Promover un enfoque disciplinado para la gestión de datos.

### ✓ Generar múltiples recursos para realizar una correcta extracción. Transformación y tratamiento de la información empresarial.

### ✓ Brinda soporte estratégico para el manejo de datos no estructurados. Análisis de productividad y contexto de comportamiento Porcentaje de consumidores en internet (análisis).

### ✓ Generar ingresos administrativos solución.

### ✓ Puede hacer crecer su negocio de manera segura según el protocolo de seguridad de información.

### ✓ Realizar indexación y gestionar procesos de modificación de datos Los empleados lo hacen.

### ✓ SQL Server realiza operaciones de restauración y restauración. Información mientras las características del programa están habilitadas.

### 

### ***4.2.*** *Physical Data Model*

### Se le presentará el script de nuestro proyecto desarrollado es el siguiente:

CREATE DATABASE LlenaFastDB;

GO

USE LlenaFastDB;

-- tables

-- Table: Usuario

CREATE TABLE Usuario (

ID\_Usuario int NOT NULL,

Nombre\_Usuario varchar(20) NOT NULL,

Apellido\_Usuario varchar(20) NOT NULL,

Edad\_Usuario int NOT NULL,

Teléfono\_Usuario int NOT NULL,

Fecha\_Nacimiento\_Usuario date NOT NULL,

DNI\_Usuario int NOT NULL,

Distrito\_Usuario varchar(50) NOT NULL,

CONSTRAINT Usuario\_pk PRIMARY KEY (ID\_Usuario)

);

-- Table: Personal

CREATE TABLE Personal (

ID\_Personal int NOT NULL,

Nombre\_Personal varchar(20) NOT NULL,

Apellido\_Personal varchar(20) NOT NULL,

Edad\_Personal int NOT NULL,

Teléfono\_Personal int NOT NULL,

Fecha\_Nacimiento\_Personal date NOT NULL,

DNI\_Personal int NOT NULL,

Distrito\_Personal varchar(50) NOT NULL,

CONSTRAINT Personal\_pk PRIMARY KEY (ID\_Personal)

);

-- Table: Lista\_Usuario

CREATE TABLE Lista\_Usuario (

ID\_Lista\_Usuario int NOT NULL,

FechaRegistro\_Lista\_Usuario date NOT NULL,

Usuario\_ID int NOT NULL,

CONSTRAINT Lista\_Usuario\_pk PRIMARY KEY (ID\_Lista\_Usuario)

);

-- Table: Lista\_Personal

CREATE TABLE Lista\_Personal (

ID\_Lista\_Personal int NOT NULL,

FechaRegistro\_Lista\_Personal date NOT NULL,

Personal\_ID int NOT NULL,

CONSTRAINT Lista\_Personal\_pk PRIMARY KEY (ID\_Lista\_Personal)

);

-- Table: Organizacion

CREATE TABLE Organizacion (

ID\_Organizacion int NOT NULL,

Nombre\_Organizacion varchar(20) NOT NULL,

Socios\_Organizacion varchar(50) NOT NULL,

CONSTRAINT Organizacion\_pk PRIMARY KEY (ID\_Organizacion)

);

-- Table: Financiera

CREATE TABLE Financiera (

ID\_Financiera int NOT NULL,

RegistroContables\_Financiera varchar(300) NOT NULL,

CONSTRAINT Financiera\_pk PRIMARY KEY (ID\_Financiera)

);

-- Table: Producto

CREATE TABLE Producto (

ID\_Producto int NOT NULL,

Precio\_Producto float NOT NULL,

Marca\_Producto varchar(20) NOT NULL,

Nombre\_Producto varchar(20) NOT NULL,

Cantidad\_Producto float NOT NULL,

Descripción\_Producto varchar(100) NOT NULL,

CONSTRAINT Producto\_pk PRIMARY KEY (ID\_Producto)

);

-- Table: Lista\_Producto

CREATE TABLE Lista\_Producto (

ID\_Lista\_Producto int NOT NULL,

VecesComprada\_Lista\_Producto int NOT NULL,

FechaComprada\_Lista\_Producto date NOT NULL,

TotalPagado\_Lista\_Producto float NOT NULL,

DireccionEntregado\_Lista\_Producto varchar(50) NOT NULL,

Personal\_ID int NOT NULL,

Producto\_ID int NOT NULL,

CONSTRAINT Lista\_Producto\_pk PRIMARY KEY (ID\_Lista\_Producto)

);

-- Table: Comprar\_Producto

CREATE TABLE Comprar\_Producto (

ID\_Comprar\_Producto int NOT NULL,

Nombre\_Comprar\_Producto varchar(20) NOT NULL,

Teléfono\_Comprar\_Producto int NOT NULL,

MétodoPago\_Comprar\_Producto varchar(50) NOT NULL,

CódigoPostal\_Comprar\_Producto int NOT NULL,

Ubicación\_ID int NOT NULL,

Producto\_ID int NOT NULL,

CONSTRAINT Comprar\_Producto\_pk PRIMARY KEY (ID\_Comprar\_Producto)

);

-- Table: Ubicación

CREATE TABLE Ubicación (

ID\_Ubicación int NOT NULL,

NombreUbicación\_Ubicación varchar(50) NOT NULL,

DirecciónUbicación\_Ubicación varchar(50) NOT NULL,

CONSTRAINT Ubicación\_pk PRIMARY KEY (ID\_Ubicación)

);

-- Table: Historial

CREATE TABLE Historial (

ID\_Historial int NOT NULL,

HoraPedido\_Pedido time NOT NULL,

CantidadCombustible\_Pedido float NOT NULL,

EstadoPedido\_Pedido varchar(30) NOT NULL,

Usuario\_ID int NOT NULL,

CONSTRAINT Historial\_pk PRIMARY KEY (ID\_Historial)

);

-- Table: Pedido

CREATE TABLE Pedido (

ID\_Pedido int NOT NULL,

FechaPedido\_Pedido date NOT NULL,

HoraPedido\_Pedido time NOT NULL,

CantidadCombustible\_Pedido float NOT NULL,

EstadoPedido\_Pedido varchar(30) NOT NULL,

CONSTRAINT Pedido\_pk PRIMARY KEY (ID\_Pedido)

);

-- foreign keys

-- Reference: Lista\_Usuario\_Usuario (table: Personal)

ALTER TABLE Lista\_Usuario ADD CONSTRAINT Lista\_Usuario\_Usuario

FOREIGN KEY (Usuario\_ID)

REFERENCES Usuario (ID\_Usuario);

-- Reference: Lista\_Personal\_Personal (table: Personal)

ALTER TABLE Lista\_Personal ADD CONSTRAINT Lista\_Personal\_Personal

FOREIGN KEY (Personal\_ID)

REFERENCES Personal (ID\_Personal);

-- Reference: Lista\_Producto\_Personal (table: Lista\_Producto)

ALTER TABLE Lista\_Producto ADD CONSTRAINT Lista\_Producto\_Personal

FOREIGN KEY (Personal\_ID)

REFERENCES Personal (ID\_Personal);

-- Reference: Lista\_Producto\_Producto (table: Lista\_Producto)

ALTER TABLE Lista\_Producto ADD CONSTRAINT Lista\_Producto\_Producto

FOREIGN KEY (Producto\_ID)

REFERENCES Producto (ID\_Producto);

-- Reference: Comprar\_Producto\_Ubicación (table: Comprar\_Producto)

ALTER TABLE Comprar\_Producto ADD CONSTRAINT Comprar\_Producto\_Ubicación

FOREIGN KEY (Ubicación\_ID)

REFERENCES Ubicación (ID\_Ubicación);

-- Reference: Comprar\_Producto\_Producto (table: Comprar\_Producto)

ALTER TABLE Comprar\_Producto ADD CONSTRAINT Comprar\_Producto\_Producto

FOREIGN KEY (Producto\_ID)

REFERENCES Producto (ID\_Producto);

-- Reference: Historial\_Usuario (table: Historial)

ALTER TABLE Historial ADD CONSTRAINT Historial\_Usuario

FOREIGN KEY (Usuario\_ID)

REFERENCES Usuario (ID\_Usuario);

insert into Usuario(ID\_Usuario, Nombre\_Usuario, Apellido\_Usuario, Edad\_Usuario, Teléfono\_Usuario, Fecha\_Nacimiento\_Usuario, DNI\_Usuario, Distrito\_Usuario) values(1, 'Nicolas', 'Vásquez', 19, 952856168, '2004-05-21', 72925420, 'Los Olivos');

insert into Usuario(ID\_Usuario, Nombre\_Usuario, Apellido\_Usuario, Edad\_Usuario, Teléfono\_Usuario, Fecha\_Nacimiento\_Usuario, DNI\_Usuario, Distrito\_Usuario) values(2, 'Chiara', 'Cabanillas', 18, 934764267, '2004-08-28', 73736529, 'Surco');

insert into Usuario(ID\_Usuario, Nombre\_Usuario, Apellido\_Usuario, Edad\_Usuario, Teléfono\_Usuario, Fecha\_Nacimiento\_Usuario, DNI\_Usuario, Distrito\_Usuario) values(3, 'Paolo', 'Hurtado', 20, 964856233, '2004-05-13', 74736530, 'La Molina');

insert into Usuario(ID\_Usuario, Nombre\_Usuario, Apellido\_Usuario, Edad\_Usuario, Teléfono\_Usuario, Fecha\_Nacimiento\_Usuario, DNI\_Usuario, Distrito\_Usuario) values(4, 'Esteban', 'Torres', 21, 956667899, '2004-03-11', 72567420, 'Callao');

insert into Usuario(ID\_Usuario, Nombre\_Usuario, Apellido\_Usuario, Edad\_Usuario, Teléfono\_Usuario, Fecha\_Nacimiento\_Usuario, DNI\_Usuario, Distrito\_Usuario) values(5, 'Valeria', 'González', 22, 912345678, '2001-09-10', 72223344, 'San Borja');

insert into Usuario(ID\_Usuario, Nombre\_Usuario, Apellido\_Usuario, Edad\_Usuario, Teléfono\_Usuario, Fecha\_Nacimiento\_Usuario, DNI\_Usuario, Distrito\_Usuario) values(6, 'Luis', 'Ramírez', 25, 987654321, '1998-07-15', 99887766, 'Miraflores');

insert into Usuario(ID\_Usuario, Nombre\_Usuario, Apellido\_Usuario, Edad\_Usuario, Teléfono\_Usuario, Fecha\_Nacimiento\_Usuario, DNI\_Usuario, Distrito\_Usuario) values(7, 'Fernanda', 'López', 23, 934567890, '2000-11-25', 44556677, 'Barranco');

insert into Usuario(ID\_Usuario, Nombre\_Usuario, Apellido\_Usuario, Edad\_Usuario, Teléfono\_Usuario, Fecha\_Nacimiento\_Usuario, DNI\_Usuario, Distrito\_Usuario) values(8, 'Martín', 'García', 24, 977654321, '1999-06-12', 22334455, 'San Isidro');

insert into Usuario(ID\_Usuario, Nombre\_Usuario, Apellido\_Usuario, Edad\_Usuario, Teléfono\_Usuario, Fecha\_Nacimiento\_Usuario, DNI\_Usuario, Distrito\_Usuario) values(9, 'Carolina', 'Torres', 27, 946789012, '1996-04-30', 99887766, 'Chorrillos');

insert into Usuario(ID\_Usuario, Nombre\_Usuario, Apellido\_Usuario, Edad\_Usuario, Teléfono\_Usuario, Fecha\_Nacimiento\_Usuario, DNI\_Usuario, Distrito\_Usuario) values(10, 'Javier', 'Pérez', 29, 911234567, '1994-08-07', 11223344, 'La Victoria');

insert into Usuario(ID\_Usuario, Nombre\_Usuario, Apellido\_Usuario, Edad\_Usuario, Teléfono\_Usuario, Fecha\_Nacimiento\_Usuario, DNI\_Usuario, Distrito\_Usuario) values(11, 'Ana', 'López', 26, 932456789, '1997-12-18', 44556677, 'Magdalena');

insert into Personal(ID\_Personal, Nombre\_Personal, Apellido\_Personal, Edad\_Personal, Teléfono\_Personal, Fecha\_Nacimiento\_Personal, DNI\_Personal, Distrito\_Personal) values(1, 'Renzo', 'Laguna', 19, 963838114, '2004-05-10', 72756356, 'Independencia');

insert into Personal(ID\_Personal, Nombre\_Personal, Apellido\_Personal, Edad\_Personal, Teléfono\_Personal, Fecha\_Nacimiento\_Personal, DNI\_Personal, Distrito\_Personal) values(2, 'Jesús', 'Cucho', 18, 946273821, '2004-03-15', 77325136, 'San Martin de Porres');

insert into Personal(ID\_Personal, Nombre\_Personal, Apellido\_Personal, Edad\_Personal, Teléfono\_Personal, Fecha\_Nacimiento\_Personal, DNI\_Personal, Distrito\_Personal) values(3, 'Ana', 'García', 20, 963838115, '2003-07-18', 73829127, 'San Borja');

insert into Personal(ID\_Personal, Nombre\_Personal, Apellido\_Personal, Edad\_Personal, Teléfono\_Personal, Fecha\_Nacimiento\_Personal, DNI\_Personal, Distrito\_Personal) values(4, 'Luis', 'Ramírez', 22, 963838116, '2001-02-25', 85937219, 'Miraflores');

insert into Personal(ID\_Personal, Nombre\_Personal, Apellido\_Personal, Edad\_Personal, Teléfono\_Personal, Fecha\_Nacimiento\_Personal, DNI\_Personal, Distrito\_Personal) values(5, 'Carla', 'Herrera', 25, 963838117, '1998-09-12', 91283721, 'La Molina');

insert into Personal(ID\_Personal, Nombre\_Personal, Apellido\_Personal, Edad\_Personal, Teléfono\_Personal, Fecha\_Nacimiento\_Personal, DNI\_Personal, Distrito\_Personal) values(6, 'Miguel', 'Cáceres', 30, 963838118, '1993-04-02', 63918275, 'San Isidro');

insert into Personal(ID\_Personal, Nombre\_Personal, Apellido\_Personal, Edad\_Personal, Teléfono\_Personal, Fecha\_Nacimiento\_Personal, DNI\_Personal, Distrito\_Personal) values(7, 'Paola', 'Vargas', 28, 963838119, '1995-12-08', 52398127, 'Barranco');

insert into Personal(ID\_Personal, Nombre\_Personal, Apellido\_Personal, Edad\_Personal, Teléfono\_Personal, Fecha\_Nacimiento\_Personal, DNI\_Personal, Distrito\_Personal) values(8, 'Martín', 'Torres', 19, 963838120, '2002-06-03', 73198273, 'Chorrillos');

insert into Personal(ID\_Personal, Nombre\_Personal, Apellido\_Personal, Edad\_Personal, Teléfono\_Personal, Fecha\_Nacimiento\_Personal, DNI\_Personal, Distrito\_Personal) values(9, 'Elena', 'Castillo', 21, 963838121, '2000-01-15', 92713528, 'Surco');

insert into Personal(ID\_Personal, Nombre\_Personal, Apellido\_Personal, Edad\_Personal, Teléfono\_Personal, Fecha\_Nacimiento\_Personal, DNI\_Personal, Distrito\_Personal) values(10, 'Alejandro', 'Sánchez', 23, 963838122, '1999-11-20', 62837941, 'Surquillo');

insert into Lista\_Usuario(ID\_Lista\_Usuario, FechaRegistro\_Lista\_Usuario, Usuario\_ID) values(1, '2023-04-23', 1);

insert into Lista\_Usuario(ID\_Lista\_Usuario, FechaRegistro\_Lista\_Usuario, Usuario\_ID) values(2, '2023-04-12', 2);

insert into Lista\_Usuario(ID\_Lista\_Usuario, FechaRegistro\_Lista\_Usuario, Usuario\_ID) values(3, '2023-05-01', 3);

insert into Lista\_Usuario(ID\_Lista\_Usuario, FechaRegistro\_Lista\_Usuario, Usuario\_ID) values(4, '2023-03-28', 4);

insert into Lista\_Usuario(ID\_Lista\_Usuario, FechaRegistro\_Lista\_Usuario, Usuario\_ID) values(5, '2023-06-10', 5);

insert into Lista\_Usuario(ID\_Lista\_Usuario, FechaRegistro\_Lista\_Usuario, Usuario\_ID) values(6, '2023-02-15', 6);

insert into Lista\_Usuario(ID\_Lista\_Usuario, FechaRegistro\_Lista\_Usuario, Usuario\_ID) values(7, '2023-07-05', 7);

insert into Lista\_Usuario(ID\_Lista\_Usuario, FechaRegistro\_Lista\_Usuario, Usuario\_ID) values(8, '2023-01-03', 8);

insert into Lista\_Usuario(ID\_Lista\_Usuario, FechaRegistro\_Lista\_Usuario, Usuario\_ID) values(9, '2023-08-18', 9);

insert into Lista\_Usuario(ID\_Lista\_Usuario, FechaRegistro\_Lista\_Usuario, Usuario\_ID) values(10, '2023-06-30', 10);

insert into Lista\_Usuario(ID\_Lista\_Usuario, FechaRegistro\_Lista\_Usuario, Usuario\_ID) values(11, '2023-04-05', 11);

insert into Lista\_Personal(ID\_Lista\_Personal, FechaRegistro\_Lista\_Personal, Personal\_ID) values(1, '2023-01-01', 1);

insert into Lista\_Personal(ID\_Lista\_Personal, FechaRegistro\_Lista\_Personal, Personal\_ID) values(2, '2023-01-03', 2);

insert into Lista\_Personal(ID\_Lista\_Personal, FechaRegistro\_Lista\_Personal, Personal\_ID) values(3, '2023-01-10', 3);

insert into Lista\_Personal(ID\_Lista\_Personal, FechaRegistro\_Lista\_Personal, Personal\_ID) values(4, '2023-01-15', 4);

insert into Lista\_Personal(ID\_Lista\_Personal, FechaRegistro\_Lista\_Personal, Personal\_ID) values(5, '2023-01-20', 5);

insert into Lista\_Personal(ID\_Lista\_Personal, FechaRegistro\_Lista\_Personal, Personal\_ID) values(6, '2023-01-25', 6);

insert into Lista\_Personal(ID\_Lista\_Personal, FechaRegistro\_Lista\_Personal, Personal\_ID) values(7, '2023-02-02', 7);

insert into Lista\_Personal(ID\_Lista\_Personal, FechaRegistro\_Lista\_Personal, Personal\_ID) values(8, '2023-02-08', 8);

insert into Lista\_Personal(ID\_Lista\_Personal, FechaRegistro\_Lista\_Personal, Personal\_ID) values(9, '2023-02-14', 9);

insert into Lista\_Personal(ID\_Lista\_Personal, FechaRegistro\_Lista\_Personal, Personal\_ID) values(10, '2023-02-20', 10);

insert into Lista\_Personal(ID\_Lista\_Personal, FechaRegistro\_Lista\_Personal, Personal\_ID) values(11, '2023-02-26', 11);

insert into Producto(ID\_Producto, Precio\_Producto, Marca\_Producto, Nombre\_Producto, Cantidad\_Producto, Descripción\_Producto) values(1, 50.00, 'REPSOL', 'REPSOL EFITEC 95', 10, 'Repsol Efitec 95 es un carburante diseñado para satisfacer las necesidades de gasolina.');

insert into Producto(ID\_Producto, Precio\_Producto, Marca\_Producto, Nombre\_Producto, Cantidad\_Producto, Descripción\_Producto) values(2, 50.00, 'PRIMAX', 'PRIMAX 90', 5, 'Primax 90 para todos los vehículos de los transportistas.');

insert into Producto(ID\_Producto, Precio\_Producto, Marca\_Producto, Nombre\_Producto, Cantidad\_Producto, Descripción\_Producto) values(3, 50.00, 'Primax', 'Primax G-Force 95', 10, 'Primax G-Force 95 es una gasolina de alta calidad diseñada para optimizar el rendimiento de tu vehículo.');

insert into Producto(ID\_Producto, Precio\_Producto, Marca\_Producto, Nombre\_Producto, Cantidad\_Producto, Descripción\_Producto) values(4, 55.00, 'Pecsa', 'Pecsa V-Power 97', 15, 'Pecsa V-Power 97 es una gasolina de alto octanaje que brinda mayor potencia y rendimiento a tu vehículo.');

insert into Producto(ID\_Producto, Precio\_Producto, Marca\_Producto, Nombre\_Producto, Cantidad\_Producto, Descripción\_Producto) values(5, 45.00, 'Repsol', 'Repsol Diesel Turbo', 20, 'Repsol Diesel Turbo es un combustible diésel de calidad premium que garantiza una mayor eficiencia en el motor.');

insert into Producto(ID\_Producto, Precio\_Producto, Marca\_Producto, Nombre\_Producto, Cantidad\_Producto, Descripción\_Producto) values(6, 60.00, 'Primax', 'Primax Turbo Gas 98', 12, 'Primax Turbo Gas 98 es una gasolina de alto rendimiento diseñada para motores de alta compresión.');

insert into Producto(ID\_Producto, Precio\_Producto, Marca\_Producto, Nombre\_Producto, Cantidad\_Producto, Descripción\_Producto) values(7, 48.00, 'Repsol', 'Repsol Gasohol 90', 18, 'Repsol Gasohol 90 es una mezcla de gasolina y etanol que ofrece una excelente relación entre calidad y precio.');

insert into Producto(ID\_Producto, Precio\_Producto, Marca\_Producto, Nombre\_Producto, Cantidad\_Producto, Descripción\_Producto) values(7, 52.00, 'Pecsa', 'Pecsa Power Diesel', 25, 'Pecsa Power Diesel es un combustible diésel de alto poder calorífico que garantiza un mejor desempeño en motores de alto rendimiento.');

insert into Producto(ID\_Producto, Precio\_Producto, Marca\_Producto, Nombre\_Producto, Cantidad\_Producto, Descripción\_Producto) values(8, 42.00, 'Primax', 'Primax Gasohol 84', 30, 'Primax Gasohol 84 es una opción económica de combustible que brinda un buen rendimiento para motores de baja cilindrada.');

insert into Producto(ID\_Producto, Precio\_Producto, Marca\_Producto, Nombre\_Producto, Cantidad\_Producto, Descripción\_Producto) values(9, 65.00, 'Repsol', 'Repsol Neotech 100', 8, 'Repsol Neotech 100 es una gasolina de alto octanaje y baja emisión de compuestos nocivos, ideal para motores de alto rendimiento.');

insert into Producto(ID\_Producto, Precio\_Producto, Marca\_Producto, Nombre\_Producto, Cantidad\_Producto, Descripción\_Producto) values(10, 47.00, 'Pecsa', 'Pecsa Gasohol 95', 16, 'Pecsa Gasohol 95 es una mezcla de gasolina y etanol que brinda un buen balance entre rendimiento y eficiencia energética.');

insert into Lista\_Producto(ID\_Lista\_Producto, VecesComprada\_Lista\_Producto, FechaComprada\_Lista\_Producto, TotalPagado\_Lista\_Producto, DireccionEntregado\_Lista\_Producto, Personal\_ID, Producto\_ID) values(1, 2, '2023-11-24', 100.00, 'Avenida Brasil', 1, 1);

insert into Lista\_Producto(ID\_Lista\_Producto, VecesComprada\_Lista\_Producto, FechaComprada\_Lista\_Producto, TotalPagado\_Lista\_Producto, DireccionEntregado\_Lista\_Producto, Personal\_ID, Producto\_ID) values(2, 3, '2023-10-20', 150.00, 'Avenida Elmer Fauccet', 2, 2);

insert into Lista\_Producto(ID\_Lista\_Producto, VecesComprada\_Lista\_Producto, FechaComprada\_Lista\_Producto, TotalPagado\_Lista\_Producto, DireccionEntregado\_Lista\_Producto, Personal\_ID, Producto\_ID) values(3, 1, '2023-11-15', 60.00, 'Avenida Arequipa', 3, 3);

insert into Lista\_Producto(ID\_Lista\_Producto, VecesComprada\_Lista\_Producto, FechaComprada\_Lista\_Producto, TotalPagado\_Lista\_Producto, DireccionEntregado\_Lista\_Producto, Personal\_ID, Producto\_ID) values(4, 4, '2023-12-10', 160.00, 'Avenida Salaverry', 4, 4);

insert into Lista\_Producto(ID\_Lista\_Producto, VecesComprada\_Lista\_Producto, FechaComprada\_Lista\_Producto, TotalPagado\_Lista\_Producto, DireccionEntregado\_Lista\_Producto, Personal\_ID, Producto\_ID) values(5, 2, '2023-11-28', 80.00, 'Avenida Angamos Este', 5, 5);

insert into Lista\_Producto(ID\_Lista\_Producto, VecesComprada\_Lista\_Producto, FechaComprada\_Lista\_Producto, TotalPagado\_Lista\_Producto, DireccionEntregado\_Lista\_Producto, Personal\_ID, Producto\_ID) values(6, 3, '2023-12-03', 120.00, 'Avenida La Marina', 6, 6);

insert into Lista\_Producto(ID\_Lista\_Producto, VecesComprada\_Lista\_Producto, FechaComprada\_Lista\_Producto, TotalPagado\_Lista\_Producto, DireccionEntregado\_Lista\_Producto, Personal\_ID, Producto\_ID) values(7, 1, '2023-11-18', 60.00, 'Avenida Benavides', 7, 7);

insert into Lista\_Producto(ID\_Lista\_Producto, VecesComprada\_Lista\_Producto, FechaComprada\_Lista\_Producto, TotalPagado\_Lista\_Producto, DireccionEntregado\_Lista\_Producto, Personal\_ID, Producto\_ID) values(8, 4, '2023-12-15', 160.00, 'Avenida Pardo', 8, 8);

insert into Lista\_Producto(ID\_Lista\_Producto, VecesComprada\_Lista\_Producto, FechaComprada\_Lista\_Producto, TotalPagado\_Lista\_Producto, DireccionEntregado\_Lista\_Producto, Personal\_ID, Producto\_ID) values(9, 3, '2023-12-07', 120.00, 'Avenida Túpac Amaru', 9, 9);

insert into Lista\_Producto(ID\_Lista\_Producto, VecesComprada\_Lista\_Producto, FechaComprada\_Lista\_Producto, TotalPagado\_Lista\_Producto, DireccionEntregado\_Lista\_Producto, Personal\_ID, Producto\_ID) values(10, 3, '2023-12-05', 120.00, 'Avenida Javier Prado Este', 10, 10);

insert into Lista\_Producto(ID\_Lista\_Producto, VecesComprada\_Lista\_Producto, FechaComprada\_Lista\_Producto, TotalPagado\_Lista\_Producto, DireccionEntregado\_Lista\_Producto, Personal\_ID, Producto\_ID) values(11, 2, '2023-12-12', 90.00, 'Avenida Garcilaso de la Vega', 11, 11);

insert into Ubicación(ID\_Ubicación, NombreUbicación\_Ubicación, DirecciónUbicación\_Ubicación) values(1, 'Gamarra, Los Olivos', 'Gamarra, La Victoria 15018');

insert into Ubicación(ID\_Ubicación, NombreUbicación\_Ubicación, DirecciónUbicación\_Ubicación) values(2, 'Covida, Los Olivos', 'Urb. Covida Los Olivos');

insert into Ubicación(ID\_Ubicación, NombreUbicación\_Ubicación, DirecciónUbicación\_Ubicación) values(3, 'Miraflores, Lima', 'Miraflores, Lima 15074');

insert into Ubicación(ID\_Ubicación, NombreUbicación\_Ubicación, DirecciónUbicación\_Ubicación) values(4, 'San Isidro, Lima', 'San Isidro, Lima 15036');

insert into Ubicación(ID\_Ubicación, NombreUbicación\_Ubicación, DirecciónUbicación\_Ubicación) values(5, 'Barranco, Lima', 'Barranco, Lima 15063');

insert into Ubicación(ID\_Ubicación, NombreUbicación\_Ubicación, DirecciónUbicación\_Ubicación) values(6, 'La Molina, Lima', 'La Molina, Lima 15012');

insert into Ubicación(ID\_Ubicación, NombreUbicación\_Ubicación, DirecciónUbicación\_Ubicación) values(7, 'Surco, Lima', 'Surco, Lima 15023');

insert into Ubicación(ID\_Ubicación, NombreUbicación\_Ubicación, DirecciónUbicación\_Ubicación) values(8, 'San Borja, Lima', 'San Borja, Lima 15037');

insert into Ubicación(ID\_Ubicación, NombreUbicación\_Ubicación, DirecciónUbicación\_Ubicación) values(9, 'Jesús María, Lima', 'Jesús María, Lima 15072');

insert into Ubicación(ID\_Ubicación, NombreUbicación\_Ubicación, DirecciónUbicación\_Ubicación) values(10, 'Magdalena del Mar, Lima', 'Magdalena del Mar, Lima 15086');

insert into Ubicación(ID\_Ubicación, NombreUbicación\_Ubicación, DirecciónUbicación\_Ubicación) values(11, 'Lince, Lima', 'Lince, Lima 15077');

insert into Comprar\_Producto(ID\_Comprar\_Producto, Nombre\_Comprar\_Producto, Teléfono\_Comprar\_Producto, MétodoPago\_Comprar\_Producto, CódigoPostal\_Comprar\_Producto, Ubicación\_ID, Producto\_ID) values(1, 'Nicolas', 952856168, 'Tarjeta de Crédito VISA', 141815, 1, 1);

insert into Comprar\_Producto(ID\_Comprar\_Producto, Nombre\_Comprar\_Producto, Teléfono\_Comprar\_Producto, MétodoPago\_Comprar\_Producto, CódigoPostal\_Comprar\_Producto, Ubicación\_ID, Producto\_ID) values(2, 'Chiara', 934764267, 'Tarjeta de Débito VISA', 157234, 2, 2);

insert into Comprar\_Producto(ID\_Comprar\_Producto, Nombre\_Comprar\_Producto, Teléfono\_Comprar\_Producto, MétodoPago\_Comprar\_Producto, CódigoPostal\_Comprar\_Producto, Ubicación\_ID, Producto\_ID) values(3, 'Paolo', 964856233, 'Transferencia Bancaria', 126543, 3, 3);

insert into Comprar\_Producto(ID\_Comprar\_Producto, Nombre\_Comprar\_Producto, Teléfono\_Comprar\_Producto, MétodoPago\_Comprar\_Producto, CódigoPostal\_Comprar\_Producto, Ubicación\_ID, Producto\_ID) values(4, 'Esteban', 956667899, 'PayPal', 182345, 4, 4);

insert into Comprar\_Producto(ID\_Comprar\_Producto, Nombre\_Comprar\_Producto, Teléfono\_Comprar\_Producto, MétodoPago\_Comprar\_Producto, CódigoPostal\_Comprar\_Producto, Ubicación\_ID, Producto\_ID) values(5, 'Valeria', 912345678, 'Tarjeta de Crédito Mastercard', 132456, 5, 5);

insert into Comprar\_Producto(ID\_Comprar\_Producto, Nombre\_Comprar\_Producto, Teléfono\_Comprar\_Producto, MétodoPago\_Comprar\_Producto, CódigoPostal\_Comprar\_Producto, Ubicación\_ID, Producto\_ID) values(6, 'Luis', 987654321, 'Efectivo', 150987, 6, 6);

insert into Comprar\_Producto(ID\_Comprar\_Producto, Nombre\_Comprar\_Producto, Teléfono\_Comprar\_Producto, MétodoPago\_Comprar\_Producto, CódigoPostal\_Comprar\_Producto, Ubicación\_ID, Producto\_ID) values(7, 'Fernanda', 934567890, 'Tarjeta de Débito Mastercard', 165432, 7, 7);

insert into Comprar\_Producto(ID\_Comprar\_Producto, Nombre\_Comprar\_Producto, Teléfono\_Comprar\_Producto, MétodoPago\_Comprar\_Producto, CódigoPostal\_Comprar\_Producto, Ubicación\_ID, Producto\_ID) values(8, 'Martín', 977654321, 'Transferencia Bancaria', 123456, 8, 8);

insert into Comprar\_Producto(ID\_Comprar\_Producto, Nombre\_Comprar\_Producto, Teléfono\_Comprar\_Producto, MétodoPago\_Comprar\_Producto, CódigoPostal\_Comprar\_Producto, Ubicación\_ID, Producto\_ID) values(9, 'Carolina', 946789012, 'Tarjeta de Crédito American Express', 176543, 9, 9);

insert into Comprar\_Producto(ID\_Comprar\_Producto, Nombre\_Comprar\_Producto, Teléfono\_Comprar\_Producto, MétodoPago\_Comprar\_Producto, CódigoPostal\_Comprar\_Producto, Ubicación\_ID, Producto\_ID) values(10, 'Javier', 911234567, 'PayPal', 189765, 10, 10);

insert into Comprar\_Producto(ID\_Comprar\_Producto, Nombre\_Comprar\_Producto, Teléfono\_Comprar\_Producto, MétodoPago\_Comprar\_Producto, CódigoPostal\_Comprar\_Producto, Ubicación\_ID, Producto\_ID) values(11, 'Ana', 932456789, 'Efectivo', 160987, 11, 11);

insert into Pedido(ID\_Pedido, FechaPedido\_Pedido, HoraPedido\_Pedido, CantidadCombustible\_Pedido, EstadoPedido\_Pedido) values(1, '2023-11-24', '16:38', '95.00', 'Entregado');

insert into Pedido(ID\_Pedido, FechaPedido\_Pedido, HoraPedido\_Pedido, CantidadCombustible\_Pedido, EstadoPedido\_Pedido) values(2, '2023-10-20', '18:23', '90.00', 'Entregado');

insert into Pedido(ID\_Pedido, FechaPedido\_Pedido, HoraPedido\_Pedido, CantidadCombustible\_Pedido, EstadoPedido\_Pedido) values(3, '2023-12-05', '14:12', '85.00', 'Entregado');

insert into Pedido(ID\_Pedido, FechaPedido\_Pedido, HoraPedido\_Pedido, CantidadCombustible\_Pedido, EstadoPedido\_Pedido) values(4, '2023-11-15', '12:45', '80.00', 'Entregado');

insert into Pedido(ID\_Pedido, FechaPedido\_Pedido, HoraPedido\_Pedido, CantidadCombustible\_Pedido, EstadoPedido\_Pedido) values(5, '2023-12-10', '10:30', '75.00', 'Entregado');

insert into Pedido(ID\_Pedido, FechaPedido\_Pedido, HoraPedido\_Pedido, CantidadCombustible\_Pedido, EstadoPedido\_Pedido) values(6, '2023-11-28', '09:15', '70.00', 'Entregado');

insert into Pedido(ID\_Pedido, FechaPedido\_Pedido, HoraPedido\_Pedido, CantidadCombustible\_Pedido, EstadoPedido\_Pedido) values(7, '2023-10-25', '16:50', '65.00', 'Entregado');

insert into Pedido(ID\_Pedido, FechaPedido\_Pedido, HoraPedido\_Pedido, CantidadCombustible\_Pedido, EstadoPedido\_Pedido) values(8, '2023-12-02', '11:20', '60.00', 'Entregado');

insert into Pedido(ID\_Pedido, FechaPedido\_Pedido, HoraPedido\_Pedido, CantidadCombustible\_Pedido, EstadoPedido\_Pedido) values(9, '2023-11-12', '15:05', '55.00', 'Entregado');

insert into Pedido(ID\_Pedido, FechaPedido\_Pedido, HoraPedido\_Pedido, CantidadCombustible\_Pedido, EstadoPedido\_Pedido) values(10, '2023-10-18', '13:40', '50.00', 'Entregado');

insert into Pedido(ID\_Pedido, FechaPedido\_Pedido, HoraPedido\_Pedido, CantidadCombustible\_Pedido, EstadoPedido\_Pedido) values(11, '2023-12-08', '08:55', '45.00', 'Entregado');

insert into Historial(ID\_Historial, HoraPedido\_Pedido, CantidadCombustible\_Pedido, EstadoPedido\_Pedido, Usuario\_ID) values(1, '16:38', '95.00', 'Entregado', 1);

insert into Historial(ID\_Historial, HoraPedido\_Pedido, CantidadCombustible\_Pedido, EstadoPedido\_Pedido, Usuario\_ID) values(2, '18:23', '90.00', 'Entregado', 2);

insert into Historial(ID\_Historial, HoraPedido\_Pedido, CantidadCombustible\_Pedido, EstadoPedido\_Pedido, Usuario\_ID) values(3, '12:45', '85.00', 'Entregado', 3);

insert into Historial(ID\_Historial, HoraPedido\_Pedido, CantidadCombustible\_Pedido, EstadoPedido\_Pedido, Usuario\_ID) values(4, '14:12', '92.50', 'Entregado', 4);

insert into Historial(ID\_Historial, HoraPedido\_Pedido, CantidadCombustible\_Pedido, EstadoPedido\_Pedido, Usuario\_ID) values(5, '10:30', '80.00', 'En Proceso', 5);

insert into Historial(ID\_Historial, HoraPedido\_Pedido, CantidadCombustible\_Pedido, EstadoPedido\_Pedido, Usuario\_ID) values(6, '20:15', '89.50', 'En Proceso', 6);

insert into Historial(ID\_Historial, HoraPedido\_Pedido, CantidadCombustible\_Pedido, EstadoPedido\_Pedido, Usuario\_ID) values(7, '11:50', '88.00', 'En Proceso', 7);

insert into Historial(ID\_Historial, HoraPedido\_Pedido, CantidadCombustible\_Pedido, EstadoPedido\_Pedido, Usuario\_ID) values(8, '19:05', '87.50', 'Pendiente', 8);

insert into Historial(ID\_Historial, HoraPedido\_Pedido, CantidadCombustible\_Pedido, EstadoPedido\_Pedido, Usuario\_ID) values(9, '16:20', '86.00', 'Pendiente', 9);

insert into Historial(ID\_Historial, HoraPedido\_Pedido, CantidadCombustible\_Pedido, EstadoPedido\_Pedido, Usuario\_ID) values(10, '13:15', '94.50', 'Pendiente', 10);

insert into Historial(ID\_Historial, HoraPedido\_Pedido, CantidadCombustible\_Pedido, EstadoPedido\_Pedido, Usuario\_ID) values(11, '17:45', '93.00', 'Pendiente', 11);

insert into Organizacion(ID\_Organizacion, Nombre\_Organizacion, Socios\_Organizacion) values(1, 'GAS-CAR', 'REPSOL, PRIMAX, DIESEL');

insert into Organizacion(ID\_Organizacion, Nombre\_Organizacion, Socios\_Organizacion) values(2, 'GAS-CAR', 'PETROPERÚ, PRIMAX, DIESEL');

insert into Organizacion(ID\_Organizacion, Nombre\_Organizacion, Socios\_Organizacion) values(3, 'GAS-CAR', 'SHELL, PRIMAX, DIESEL');

insert into Organizacion(ID\_Organizacion, Nombre\_Organizacion, Socios\_Organizacion) values(4, 'GAS-CAR', 'REPSOL, YPF, DIESEL');

insert into Organizacion(ID\_Organizacion, Nombre\_Organizacion, Socios\_Organizacion) values(5, 'GAS-CAR', 'EXXON, PRIMAX, DIESEL');

insert into Organizacion(ID\_Organizacion, Nombre\_Organizacion, Socios\_Organizacion) values(6, 'GAS-CAR', 'TOTAL, PRIMAX, DIESEL');

insert into Organizacion(ID\_Organizacion, Nombre\_Organizacion, Socios\_Organizacion) values(7, 'GAS-CAR', 'CHEVRON, PRIMAX, DIESEL');

insert into Organizacion(ID\_Organizacion, Nombre\_Organizacion, Socios\_Organizacion) values(8, 'GAS-CAR', 'ESSO, PRIMAX, DIESEL');

insert into Organizacion(ID\_Organizacion, Nombre\_Organizacion, Socios\_Organizacion) values(9, 'GAS-CAR', 'GULF, PRIMAX, DIESEL');

insert into Organizacion(ID\_Organizacion, Nombre\_Organizacion, Socios\_Organizacion) values(10, 'GAS-CAR', 'CEPSA, PRIMAX, DIESEL');

insert into Financiera(ID\_Financiera, RegistroContables\_Financiera) values(1, 'Gastos de personal, atención al personal, sueldos y salarios, seguridad, gastos en general, ganancias en general');

insert into Financiera(ID\_Financiera, RegistroContables\_Financiera) values(2, 'Inversiones, activos fijos, cuentas por pagar, ingresos financieros, depreciación');

insert into Financiera(ID\_Financiera, RegistroContables\_Financiera) values(3, 'Impuestos, deudas financieras, amortización, cuentas por cobrar, gastos financieros');

insert into Financiera(ID\_Financiera, RegistroContables\_Financiera) values(4, 'Costos de producción, costos indirectos, ventas, ingresos por servicios, proveedores');

insert into Financiera(ID\_Financiera, RegistroContables\_Financiera) values(5, 'Reservas, dividendos, reembolsos, ingresos extraordinarios, costos de administración');

insert into Financiera(ID\_Financiera, RegistroContables\_Financiera) values(6, 'Ingresos por ventas, cuentas bancarias, activos intangibles, gastos de publicidad, utilidades');

insert into Financiera(ID\_Financiera, RegistroContables\_Financiera) values(7, 'Comisiones, inversiones financieras, costos variables, ingresos por alquileres, provisiones');

insert into Financiera(ID\_Financiera, RegistroContables\_Financiera) values(8, 'Pasivos acumulados, gastos financieros, ingresos por intereses, activos líquidos, depreciación acumulada');

insert into Financiera(ID\_Financiera, RegistroContables\_Financiera) values(9, 'Costos de mantenimiento, reembolsos de préstamos, ingresos por arrendamiento, reservas de capital, impuestos diferidos');

insert into Financiera(ID\_Financiera, RegistroContables\_Financiera) values(10, 'Cuentas de ahorro, provisiones para impuestos, ingresos por dividendos, inversiones a largo plazo, gastos operativos');

-- End of file.

-- Validación de campos para la tabla Usuario

ALTER TABLE Usuario

ADD CONSTRAINT CK\_Usuario\_Nombre\_Usuario CHECK (LEN(Nombre\_Usuario) <= 20);

ALTER TABLE Usuario

ADD CONSTRAINT CK\_Usuario\_Apellido\_Usuario CHECK (LEN(Apellido\_Usuario) <= 20);

ALTER TABLE Usuario

ADD CONSTRAINT CK\_Usuario\_Distrito\_Usuario CHECK (LEN(Distrito\_Usuario) <= 50);

-- Validación de campos para la tabla Personal

ALTER TABLE Personal

ADD CONSTRAINT CK\_Personal\_Nombre\_Personal CHECK (LEN(Nombre\_Personal) <= 20);

ALTER TABLE Personal

ADD CONSTRAINT CK\_Personal\_Apellido\_Personal CHECK (LEN(Apellido\_Personal) <= 20);

ALTER TABLE Personal

ADD CONSTRAINT CK\_Personal\_Distrito\_Personal CHECK (LEN(Distrito\_Personal) <= 50);

-- Validación de campos para la tabla Organizacion

ALTER TABLE Organizacion

ADD CONSTRAINT CK\_Organizacion\_Nombre\_Organizacion CHECK (LEN(Nombre\_Organizacion) <= 20);

ALTER TABLE Organizacion

ADD CONSTRAINT CK\_Organizacion\_Socios\_Organizacion CHECK (LEN(Socios\_Organizacion) <= 50);

-- Validación de campos para la tabla Producto

ALTER TABLE Producto

ADD CONSTRAINT CK\_Producto\_Marca\_Producto CHECK (LEN(Marca\_Producto) <= 20);

ALTER TABLE Producto

ADD CONSTRAINT CK\_Producto\_Nombre\_Producto CHECK (LEN(Nombre\_Producto) <= 20);

ALTER TABLE Producto

ADD CONSTRAINT CK\_Producto\_Descripcion\_Producto CHECK (LEN(Descripcion\_Producto) <= 100);

-- Validación de campos para la tabla Lista\_Producto

ALTER TABLE Lista\_Producto

ADD CONSTRAINT CK\_Lista\_Producto\_DireccionEntregado\_Lista\_Producto CHECK (LEN(DireccionEntregado\_Lista\_Producto) <= 50);

-- Validación de campos para la tabla Comprar\_Producto

ALTER TABLE Comprar\_Producto

ADD CONSTRAINT CK\_Comprar\_Producto\_Nombre\_Comprar\_Producto CHECK (LEN(Nombre\_Comprar\_Producto) <= 20);

ALTER TABLE Comprar\_Producto

ADD CONSTRAINT CK\_Comprar\_Producto\_MetodoPago\_Comprar\_Producto CHECK (LEN(MetodoPago\_Comprar\_Producto) <= 50);

-- Validación de campos para la tabla Ubicación

ALTER TABLE Ubicación

ADD CONSTRAINT CK\_Ubicacion\_NombreUbicacion\_Ubicacion CHECK (LEN(NombreUbicacion\_Ubicacion) <= 50);

ALTER TABLE Ubicación

ADD CONSTRAINT CK\_Ubicacion\_DireccionUbicacion\_Ubicacion CHECK (LEN(DireccionUbicacion\_Ubicacion) <= 50);

-- Validación de campos para la tabla Historial

ALTER TABLE Historial

ADD CONSTRAINT CK\_Historial\_EstadoPedido\_Pedido CHECK (LEN(EstadoPedido\_Pedido) <= 30);

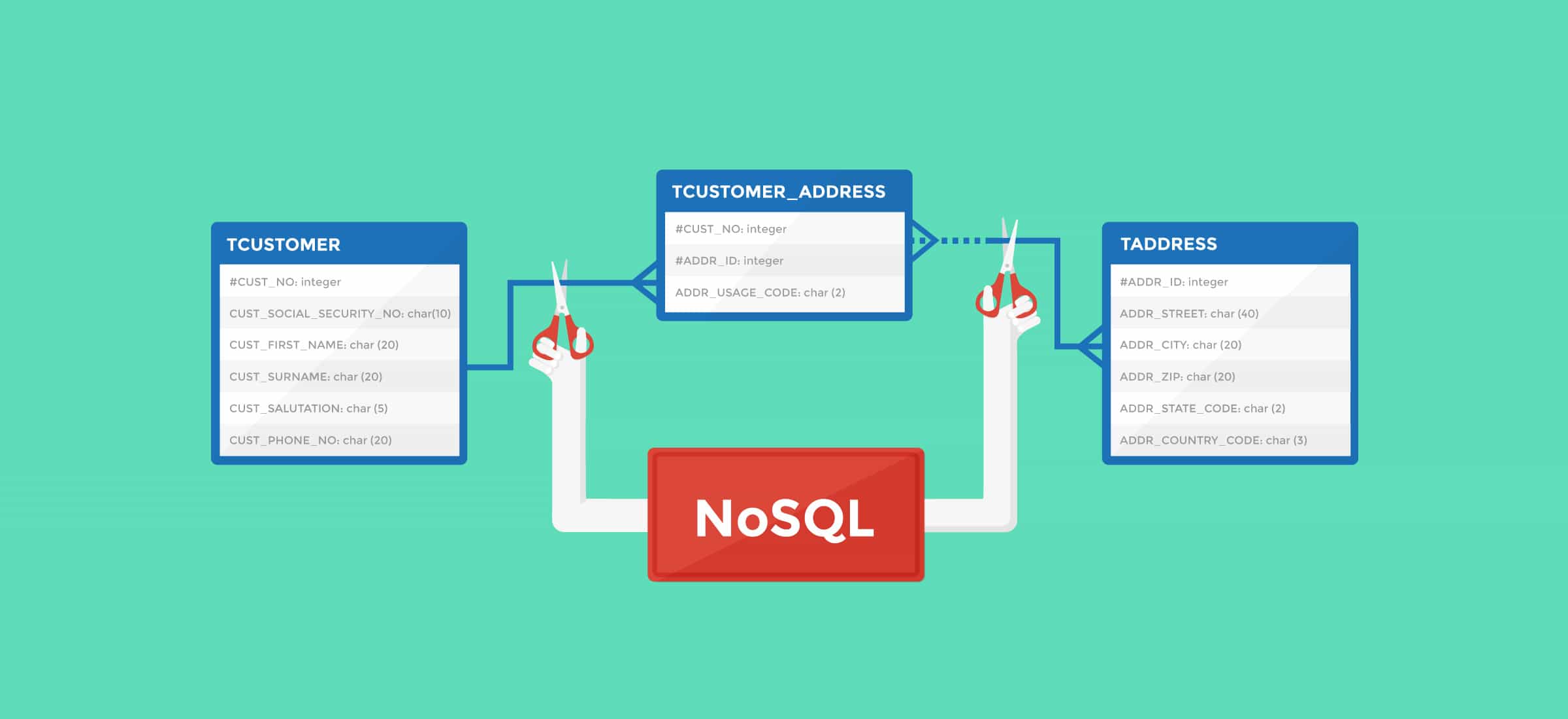
### ***4.3.*** *NonRelational Database*

### Una base de datos no relacional es un sistema de almacenamiento de información. Esto se debe principalmente a que no existe un lenguaje SQL. Las base de datos NoSQL son ampliamente aceptados debido a su facilidad de desarrollo. Gran potencia y rendimiento para una gran experiencia de usuario.

• **Flexibilidad:** Ofrece esquemas flexibles que permiten un desarrollo más rápido e iterativo. El modelo de datos flexible permite bases de datos NoSQL Ideal para datos semiestructurados y no estructurados.

• **Escalabilidad:** Diseñado para escalar utilizando clústeres distribuidos. Se usa hardware en lugar de agregar servidores costosos y robustos para escalar. Varios proveedores de la nube manejan estos procesos en segundo plano, como un servicio totalmente gestionado.

**• Rendimiento:** Optimizado para patrones y modelos de datos específicos. Acceso que permite un mayor rendimiento que intentar acceder funcionalidad similar a las bases de datos relacionales.

**• Funcional:** Proporciona API ricas y tipos de datos. Diseñado específicamente para cada modelo de datos.

A continuación se va a presenter una base de datos no relacional:

**Mongo DB:**

Es una base de datos no relacional escrita para documentos de código abierto y escrito en C++, MongoDB almacena datos en estructuras de datos BSON. Por otra parte proporciona gran escalabilidad y flexibilidad, y un modelo avanzado de consulta e indexación.

Logotipo, nombre de la empresa

Descripción generada automáticamente

**Razón por la que se eligió MongoDB:**

Elegimos MongoDB porque es una de las bases de datos NoSQL más populares. A diferencia de las bases de datos relacionales, no se almacenan datos en una tabla, pero utilizando un archivo plano con formato JSON estándar. Ahora es ampliamente utilizado en diversas aplicaciones. Esto permite integrar MongoDB y con estas aplicaciones es mucho más fácil.

### ***4.4.*** *Queries*

Para ejecutar una consulta, primero completamos la entrada de datos, aquí tenemos bases de datos tanto relacionales como no relacionales. Nuevamente, ayuda mostrar el ejemplo de la base de datos. Contiene datos con los que podemos simular el contenido de una consulta real.

### En este 1er Anexo tenemos el script que nos ha permitido cargar los datos en la base de datos relacional. En este siguiente enlace se puede observar:

### <https://drive.google.com/drive/u/0/folders/1tvQfsckOjsxDI1iygUfxXojHxqM_CyQJ>

### En el 2do Anexo se va a poder encontrar el script que nos va a permitir cargar los datos en la base de datos no relacional. En este siguiente enlace se puede visualizar todos los archivos .json:

### <https://drive.google.com/drive/u/0/folders/1SNaYY9HbkZlvnj5MuOC6TAUmakoF6UeW>

### ***4.4.1.*** *Relational Database Queries*

### - André Nicolas Vásquez Castro (3 consultas)

### **Propósito:** Obtener a todos los usuarios de la tabla “Usuario”.

### SELECT \* FROM Usuario;

### **Propósito:** Seleccionar el nombre y el precio del producto con el valor más alto en la columna "Precio\_Producto" de la tabla "Producto".

SELECT TOP 1 Nombre\_Producto, Precio\_Producto

FROM Producto

### ORDER BY Precio\_Producto DESC;

### **Propósito:** Calcular la suma total de los valores en la columna "TotalPagado\_Lista\_Producto" de la tabla "Lista\_Producto" y asignar ese valor a la "Suma\_Ventas" en el resultado de la consulta.

SELECT SUM(TotalPagado\_Lista\_Producto) AS Suma\_Ventas

### FROM Lista\_Producto;

### - Daryl Alexander Canales Quintana (3 consultas)

### **Propósito:** Obtener todos los registros de la tabla “Usuario”.

### SELECT Nombre\_Usuario, Edad\_Usuario FROM Usuario;

### **Propósito:** Seleccionar el nombre, apellido y fecha de nacimiento de los usuarios de la tabla "Usuario" que tienen una edad superior a 30 años.

SELECT Nombre\_Usuario, Apellido\_Usuario, Fecha\_Nacimiento\_Usuario

FROM Usuario

### WHERE Edad\_Usuario > 30;

### **Propósito:** Realizar una consulta que obtenga el nombre y apellido del personal (tabla "Personal") que tiene la misma edad que un usuario específico (tabla "Usuario"), identificado por el valor de la variable @UsuarioID.

DECLARE @UsuarioID INT

SELECT P.Nombre\_Personal, P.Apellido\_Personal

FROM Personal P

INNER JOIN Usuario U ON P.Edad\_Personal = U.Edad\_Usuario

### WHERE U.ID\_Usuario = @UsuarioID;

### - Renzo D’Angelo Laguna Belleza (3 consultas)

### **Propósito:** Seleccionar todos los registros de la tabla "Producto" donde el valor de la columna "Precio\_Producto" sea mayor que 40.

### SELECT \* FROM Producto WHERE Precio\_Producto > 100;

### **Propósito:** Crear un procedimiento almacenado llamado "ActualizarUsuario" que actualice la información del usuario en la tabla "Usuario".CREATE PROCEDURE ActualizarUsuario

@ID int,

@Nombre varchar(20),

@Apellido varchar(20),

@Edad int,

@Telefono int,

@FechaNacimiento date,

@DNI int,

@Distrito varchar(50)

AS

BEGIN

UPDATE Usuario

SET Nombre\_Usuario = @Nombre, Apellido\_Usuario = @Apellido,

Edad\_Usuario = @Edad, Teléfono\_Usuario = @Telefono,

Fecha\_Nacimiento\_Usuario = @FechaNacimiento, DNI\_Usuario = @DNI,

Distrito\_Usuario = @Distrito

WHERE ID\_Usuario = @ID;

### END;

### **Propósito:** Proporcionar una manera de calcular y obtener el total pagado por un usuario en base a los registros de la tabla "Historial", simplificando el proceso de cálculo y permitiendo su reutilización en consultas y cálculos posteriores.

### CREATE FUNCTION CalcularTotalPagado

(

@IDUsuario int

)

RETURNS FLOAT

AS

BEGIN

DECLARE @TotalPagado FLOAT;

SELECT @TotalPagado = SUM(CantidadCombustible\_Pedido)

FROM Historial

WHERE Usuario\_ID = @IDUsuario;

RETURN @TotalPagado;

### END;

### - Chiara Kelly Oliva Cabanillas (3 consultas)

### **Propósito:** Seleccionar el nombre de todos los productos de la tabla "Producto" y ordenarlos alfabéticamente por su nombre.

### SELECT Nombre\_Producto FROM Producto ORDER BY Nombre\_Producto;

### **Propósito:** Realizar una consulta que obtenga el nombre y apellido de los usuarios de la tabla "Usuario", junto con la cantidad de productos registrados en el historial de cada usuario.

### SELECT U.Nombre\_Usuario, U.Apellido\_Usuario, COUNT(\*) AS Cantidad\_Productos

FROM Usuario U

INNER JOIN Historial H ON U.ID\_Usuario = H.Usuario\_ID

### GROUP BY U.Nombre\_Usuario, U.Apellido\_Usuario;

### **Propósito:** Realizar una consulta que obtenga el nombre y apellido de los usuarios de la tabla "Usuario", junto con el total pagado en todas las listas de productos asociadas a cada usuario.

### SELECT U.Nombre\_Usuario, U.Apellido\_Usuario, SUM(LP.TotalPagado\_Lista\_Producto) AS Total\_Pagado

FROM Usuario U

INNER JOIN Lista\_Usuario LU ON U.ID\_Usuario = LU.Usuario\_ID

INNER JOIN Lista\_Producto LP ON LU.ID\_Lista\_Usuario = LP.ID\_Lista\_Producto

### GROUP BY U.Nombre\_Usuario, U.Apellido\_Usuario;

### - Jesús Alessandro Contreras Cucho (3 consultas)

### **Propósito:** Contar la cantidad total de registros en la tabla "Usuario" y asignar ese valor a la "Cantidad\_Usuarios" en el resultado de la consulta.

### SELECT COUNT(\*) AS Cantidad\_Usuarios FROM Usuario;

### **Propósito:** Realizar una consulta que obtenga el nombre del producto y la marca del producto de la tabla "Producto", pero solo para aquellos productos que estén asociados a usuarios mayores de 25 años.

SELECT P.Nombre\_Producto, P.Marca\_Producto

FROM Producto P

INNER JOIN Lista\_Producto LP ON P.ID\_Producto = LP.Producto\_ID

INNER JOIN Lista\_Usuario LU ON LP.ID\_Lista\_Producto = LU.ID\_Lista\_Usuario

INNER JOIN Usuario U ON LU.Usuario\_ID = U.ID\_Usuario

### WHERE U.Edad\_Usuario > 25;

### **Propósito:** Realizar una consulta que obtenga el nombre del producto y la marca del producto de la tabla "Producto", pero solo para el producto más común en la tabla "Lista\_Producto".

### SELECT TOP 1 P.Nombre\_Producto, P.Marca\_Producto

FROM Producto P

INNER JOIN Lista\_Producto LP ON P.ID\_Producto = LP.Producto\_ID

GROUP BY P.Nombre\_Producto, P.Marca\_Producto

### ORDER BY COUNT(\*) DESC;

### ***4.4.2.*** *Non-Relational Database Queries*

### - André Nicolas Vásquez Castro (2 consultas)

### **Propósito:** Calcular el número total de usuarios registrados en la base de datos.

### db.usuarios.count()

### **Propósito:** Encontrar la edad mínima y máxima de los usuarios registrados.

### db.usuarios.aggregate([

### {

### $group: {

### \_id: null,

### edadMinima: { $min: "$Edad" },

### edadMaxima: { $max: "$Edad" }

### }

### }

### ])

### - Daryl Alexander Canales Quintana (2 consultas)

### **Propósito:** Calcular el promedio de los precios de los productos en la base de datos.

### db.productos.aggregate([

### {

### $group: {

### \_id: null,

### precioPromedio: { $avg: "$Precio" }

### }

### }

### ])

### **Propósito:** Calcular la cantidad total de productos disponibles en stock.

### **db.productos.aggregate([**

### **{**

### **$group: {**

### **\_id: null,**

### **cantidadTotal: { $sum: "$Cantidad" }**

### **}**

### **}**

### **])**

### Renzo D’Angelo Laguna Belleza (2 consultas)

### **Propósito:** Calcular el número de compras realizadas por cada usuario.

### db.usuarios.aggregate([

### {

### $group: {

### \_id: "$Usuario\_ID",

### totalCompras: { $sum: 1 }

### }

### }

### ])

### **Propósito:** Calcular el monto total pagado en todas las compras registradas.

### db.productos.aggregate([

### {

### $group: {

### \_id: null,

### totalPagado: { $sum: "$TotalPagado" }

### }

### }

### ])

### - Chiara Kelly Oliva Cabanillas (2 consultas)

### **Propósito:** Calcular el número de productos comprados por cada usuario

### db.historial.aggregate([

### {

### $group: {

### \_id: "$Usuario\_ID",

### totalProductosComprados: { $sum: "$CantidadCombustible" }

### }

### }

### ])

### **Propósito:** Calcular la cantidad promedio de productos por cada marca en la base de datos.

### db.productos.aggregate([

### {

### $group: {

### \_id: "$Marca",

### cantidadPromedio: { $avg: "$Cantidad" }

### }

### }

### ])

### - Jesús Alessandro Contreras Cucho (2 consultas)

### **Propósito:** Calcular el número de pedidos en cada estado (Ejemplos: “pendiente” o “entregado”).

### db.usuarios.aggregate([

### {

### $group: {

### \_id: "$EstadoPedido",

### totalPedidos: { $sum: 1 }

### }

### }

### ])

### **Propósito:** Encontrar el nombre de la organización que tiene mayor cantidad de socios.

db.entidades.aggregate([

{

$match: {

Socios: { $exists: true, $ne: null } // Filtrar solo documentos con el campo "Socios" existente y no nulo

}

},

{

$project: {

Nombre: 1,

numSocios: { $size: { $split: ["$Socios", ","] } } // Obtener el número de socios para cada documento

}

},

{

$sort: {

numSocios: -1 // Ordenar en orden descendente según el número de socios

}

},

{

$limit: 1 // Obtener solo el documento con el mayor número de socios

}

])

*Recomendación:*

Debería habilitar un botón de “¿Desea cambiar su ubicación?”, para que el usuario vuelva a digitar su ubicación y que no ocurra inconvenientes con el pedido ni con el usuario.

## *Conclusiones y recomendaciones:*

- Todos los procesos fueron importantes para la buena elaboración de la aplicación móvil, esto nos beneficia para que sea más completa y no padezca de ningún error de presente a futuro.

- Todo lo desarrollado en este proyecto nos servirá para conocer todo sobre nuestra aplicación. Así tendremos un mejor conocimiento y manejo de LlenaFast, con el fin de poder satisfacer a nuestros usuarios.

- Hay que recordar que crear una aplicación desde cero no es nada fácil, pues se necesita tomar en cuenta varias cosas en ese transcurso de la creación de la aplicación, es un proceso bastante largo.

## *Bibliografía:*

- Gothelf, J. (9 de setiembre de 2019). Lean UX Canvas V2. Jeff Gothelf.

- Rocket Content. (20 de junio de 2019). Metodología 5w2h: qué es y cómo te ayudará a tomar las acciones correctas para tu empresa. https://rockcontent.com/es/blog/metodologia-5w2h/

Diagrama

Descripción generada automáticamente

## *Anexos:*

## |Gráfico, Gráfico de burbujas Descripción generada automáticamente

## Diagrama Descripción generada automáticamente