

**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---



**FACULTAD DE INGENIERÍA**

**Desarrollo de software para  
la implementación del  
sistema de cobro Telepeaje**

**INFORME DE ACTIVIDADES PROFESIONALES**

Que para obtener el título de  
**Ingeniero en computación**

**P R E S E N T A**

Efrén Aldair Rosales Gómez

**ASESOR DE INFORME**

Mtro. Gerardo Gabriel Carrasco Zúñiga



**Ciudad Universitaria, Cd. Mx., 2021**

# Índice

<b>Agradecimientos .....</b>	<b>4</b>
<b>Introducción .....</b>	<b>5</b>
<b>Objetivo .....</b>	<b>6</b>
<b>Antecedentes.....</b>	<b>6</b>
<b>Peaje.....</b>	<b>6</b>
<b>Cobro Electrónico de Peajes.....</b>	<b>7</b>
<b>Telepeaje En México .....</b>	<b>8</b>
<b>Operadores de Telepeaje en México.....</b>	<b>9</b>
<b>Definición del Problema .....</b>	<b>10</b>
<b>Metodología.....</b>	<b>13</b>
<b>Metodologías Ágiles .....</b>	<b>13</b>
<b>Programación Extrema (XP).....</b>	<b>14</b>
<b>Metodología Scrum .....</b>	<b>15</b>
<b>Selección de la metodología.....</b>	<b>17</b>
<b>Principales Tecnologías Utilizadas.....</b>	<b>19</b>
<b>Oracle Golden Gate .....</b>	<b>21</b>
<b>Tipo de replicación implementada. ....</b>	<b>23</b>
<b>Arquitectura .....</b>	<b>25</b>
<b>Sistema de cobro.....</b>	<b>27</b>
<b>Centralización de información.....</b>	<b>27</b>
<b>Envío de transacciones.....</b>	<b>29</b>
<b>Generación lista interoperable.....</b>	<b>30</b>
<b>CRON.....</b>	<b>31</b>
<b>Configuración de servidores .....</b>	<b>32</b>
<b>RAID.....</b>	<b>33</b>
<b>Clúster de servidores.....</b>	<b>34</b>

<b><i>SPLIT BRAIN</i></b> .....	<b>38</b>
<b><i>Disponibilidad</i></b> .....	<b>39</b>
<b><i>Back Office</i></b> .....	<b>42</b>
<b><i>Desarrollo BackOffice</i></b> .....	<b>43</b>
<b><i>Oracle Call Interface</i></b> .....	<b>44</b>
<b><i>Optimización de consultas</i></b> .....	<b>45</b>
<b><i>Bloqueos</i></b> .....	<b>52</b>
<b><i>Antifraude</i></b> .....	<b>56</b>
<b><i>Zona horaria</i></b> .....	<b>61</b>
<b><i>Conclusiones</i></b> .....	<b>62</b>
<b><i>Anexos</i></b> .....	<b>64</b>
<b><i>Fuentes</i></b> .....	<b>73</b>

## Agradecimientos

Le doy gracias a mis padres por apoyarme en todo momento, por los valores que me han inculcado, y por haberme dado la oportunidad de tener una excelente educación en el transcurso de mi vida. Sobre todo, por ser un excelente ejemplo de vida a seguir.

Le agradezco la confianza, apoyo y dedicación de tiempo a mi asesor de titulación el Ingeniero Gerardo Gabriel Carrasco Zuñiga.

A mis compañeros por haber compartido conmigo sus conocimientos y sobre todo su amistad.

## Introducción

En el presente trabajo pretendo describir las actividades y habilidades aplicadas en un ambiente profesional de trabajo, en el área de desarrollo de software y base de datos en relación con los sistemas de cobro de Telepeaje del país, el cual consiste como una alternativa electrónica para poder sustituir el pago tradicional del peaje en los distintos tramos carreteros del país.

Por otra parte, este informe se comprende de un periodo de tiempo de un año de trabajo, durante este periodo como lo mencione anteriormente he podido realizar una serie de actividades relacionadas con el desarrollo de software que serán descritas a lo largo de este informe. Las actividades que realicé me permitieron aplicar los conocimientos, técnicas, metodologías y habilidades adquiridas en la facultad en la carrera de ingeniería en computación, por lo tanto, estos conocimientos fueron fundamentales y útiles para poder asumir los retos inherentes implicados en un ambiente profesional de trabajo.

Egrese de la facultad en el año 2018 y posteriormente me integre a una empresa española en el mismo año, formando parte del equipo de desarrollo de software y configuración de servidores. Durante este periodo de tiempo comencé realizando actividades únicamente de programación, sin embargo, posteriormente me desempeñé tareas como configuración e instalación de servidores, migración de base de datos y puesta en marcha de proyectos.

A lo largo de mis actividades profesionales participe en varios proyectos en los cuales tuve que viajar a distintas partes del país, incluso también he trabajado fuera de México. Igualmente me relacione con distintas personas que desempeñan tareas independientes de temas técnicos, es decir realizan temas administrativos o logísticos para la elaboración de algún proyecto.

Durante el periodo de tiempo de un año de experiencia profesional en cada uno de los viajes y proyectos que realice me enfrente a distintos retos que requieren de una solución rápida debido a que existían factores como la pérdida de recursos económicos.

Como resultado de la constante actividad profesional en la que me he desempeñado, he adquirido prácticas y conocimientos que me permiten desarrollarme como un mejor profesional, es por esto que decidí realizar en este informe para plasmar los retos que se superé para lograr el objetivo de un proyecto real.

## Objetivo

Describir las actividades realizadas en un ambiente profesional de trabajo, para el desarrollo de software y las optimizaciones de procesos implicados en el cobro electrónico del peaje en el país para red de autopistas del Fondo Nacional de Autopistas (FONADIN).

## Antecedentes

### Peaje

El concepto de peaje se conoce como el pago efectuado por el derecho de transitar por un espacio, en la actualidad esta idea se aplica para todas las carreteras del país, es decir, se aplica una tarifa que se debe de cobrar a los medios de transporte terrestres como derecho de utilizar la infraestructura por la cual circularon.

Estrictamente existen dos tipos de peaje, marítimo y vial. Para el caso del peaje vial, se divide en dos principales tipos.

- Peaje Abierto: Consiste en que un tramo carretero posee una o varias plazas de cobro colocadas a cierta distancia para cobrar la tarifa correspondiente y existe la posibilidad de poder salir de la carretera sin pagar una tarifa extra.
- Peaje Cerrado: Cuando un automovilista ingresa a al tramo carretero se registra su entrada y al momento de realizar la salida se cobra la longitud recorrida y no existen paradas intermedias.

## Cobro Electrónico de Peajes

El cobro electrónico de peaje consiste es un sistema que permite realizar el pago de la tarifa de peaje sin la necesidad de usar efectivo, dicho pago se realiza por medio de una infraestructura tecnología de comunicación permitiendo efectuar un pago de manera automática sin que el vehículo se tenga que detener por completo asegurando una velocidad y así permitiendo un ahorro de tiempo para los usuarios que deban de realizar el pago del peaje en una carretera, este modelo sistema es conocido como Telepeaje o Freeflow.

Para que una plaza de cobro pueda implementar este método de pago electrónico debe de partir del concepto básico de la arquitectura del sistema de Telepeaje, dicho principio consiste en una serie de hardware que involucra en una antena y un dispositivo denominado TAG\*<sup>1</sup>.

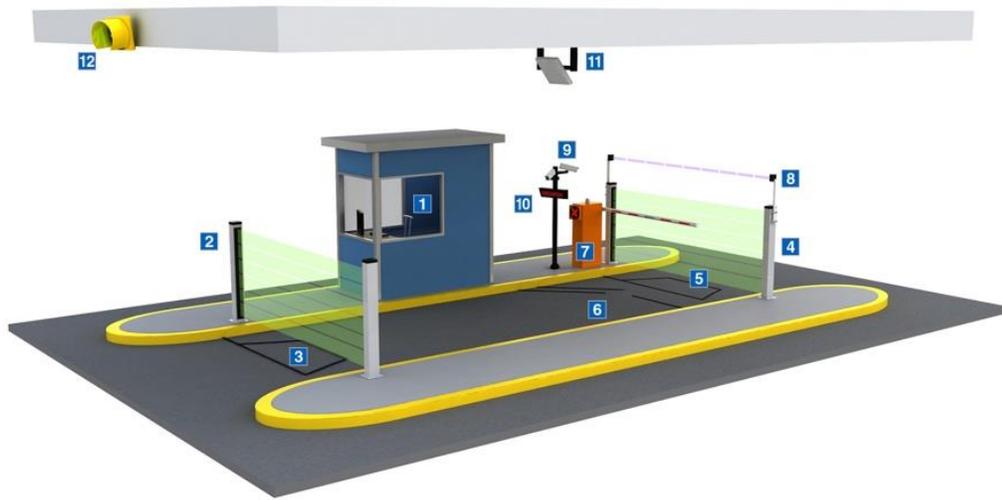
La antena es colocada a cierta altura con una inclinación sobre un poste a la entrada o en la zona de cobro de un carril de la plaza de cobro. El objetivo de la colocación de la antena es poder activar el TAG que está dentro del vehículo para poder realizar la lectura y posteriormente abrir la barrera de salida para permitir el paso del vehículo, realizando el pago de manera automática. A continuación, enunciaré los elementos del carril para facilitar la comprensión del lector.

### Elementos del carril.

1. Cabina de cobrador.
2. Cortina fotoeléctrica de entrada (preclasificación).
3. Peanas de preclasificación.
4. Cortina fotoeléctrica salida (post-clasificación).
5. Bucle de presencia.
6. Peanas de post-clasificación.
7. Barrera.
8. Fococélula de Altura.
9. Cámara de grabación.
10. Display de mensaje.
11. Antena.
12. Marquesina.

---

<sup>1</sup> La palabra TAG no es explícitamente un acrónimo técnico, sin embargo, es una palabra en inglés que puede tener distintos significados, por ejemplo: etiqueta, placa o marca que sirve para identificar algo en específico.



*Ilustración 1. Arquitectura de carril.*

## Telepeaje En México

Los sistemas de Telepeaje se han hecho una parte importante en la vida de los automovilistas, no solo por su uso en las autopistas urbanas sino en los diferentes tramos carreteros, en donde su uso permite el ahorro de tiempo al contar con uno o más carriles exclusivos con este método de pago electrónico.

El sistema de Telepeaje surge a partir de las deficiencias del modelo tradicional de peaje vial que únicamente poseía el método de pago en efectivo, originando una problemática de tiempo para los automovilistas debido a que este provocaba largas filas y generaba grandes tiempos de espera que necesitaba de una solución, por ello surge la necesidad de otro método de pago que permita el ahorro de tiempo para poder realizar un cruce en las plazas de cobro en los tramos carreteros y así poder reducir las filas al momento de pagar. Debo de mencionar, además que en el modelo de Telepeaje están involucradas distintas empresas con sus propias reglas de negocio que se deben de relacionar con el objetivo de intercambiar información de clientes para que el proceso del método de pago electrónico del Telepeaje sea completamente transparente para el automovilista. Dicho concepto de intercambio de información es conocido como interoperabilidad.

## Operadores de Telepeaje en México

El presente informe está basado en la red de autopistas de FONADIN, la cual es una concesión que otorgo el gobierno federal por conducto de la secretaria de comunicaciones y transportes STC, para construir, operar, explotar, conservar y mantener 45 caminos y 3 puentes mismos que tienen una longitud de 4,224,518 kilómetros de los cuales actualmente están en operación 3,935,014 están en operación y 289,504 en construcción. A lo largo de la red de transporte FONADIN distintas empresas están involucradas para suministrar el método de pago electrónico con un dispositivo TAG.

Cada empresa de Telepeaje o también conocido como operador de Telepeaje interoperable (OTI) ofrece a los automovilistas poder adquirir un dispositivo TAG como un identificador único para poder utilizarlo en distintas redes de autopistas dentro del territorio nacional, sin embargo, para poder ofrecer este servicio se requiere de un proveedor de Telepeaje que pueda homologar, concentrar y distribuir los datos de cada OTI.

El proveedor de Telepeaje tiene como finalidad principal la distribución y actualización de los datos del identificador único que corresponde a cada uno de los automovilistas de cada operador de Telepeaje para que cuando realicen el cruce en una de plazas de cobro del país lo puedan realizar de manera automatizada.

Las empresas que participan en el Telepeaje para la red de FONADIN en el país son PASE, VIAPASS, TEDISA y CAPUFE.

Por otra parte, el proveedor de Telepeaje además de distribuir, concentrar y homologar los datos de cada OTI debe aplicar distintas reglas de negocio, que permitan clasificar a los tipos de clientes, y además simultáneamente el proveedor debe de recaudar cada una de las transacciones asociadas a cada cruce que realice un automovilista en las plazas de cobro. La recaudación de los datos de cada cruce tiene la finalidad de devolver el estado actual de cada cliente a su correspondiente OTI para poder realizar el cobro correspondiente al TAG.

Estos son los principales y más importantes procesos para poder ofrecer al automovilista un servicio de calidad que asegure que sus datos se estén actualizados constantemente y garantizar que podrá hacer uso de este método de pago en alguna plaza de cobro del país. Quiero agregar también otro punto a este contexto, que el 1° de agosto del 2014 Caminos y Puentes Federales (CAPUFE), inició operaciones con un antiguo operador de

Telepeaje con 122 plazas de cobro que administra el organismo y el 11 de agosto de ese mismo año comenzó a nivel nacional la interoperabilidad<sup>2</sup>.

## Definición del Problema

Una de las necesidades más significativas para el sistema de Telepeaje es el concepto de interoperabilidad, debido a que es un proceso por el cual el automovilista no se debe preocupar, es decir debe de ser un procedimiento completamente transparente al momento de realizar el pago electrónico en alguna plaza de cobro con un dispositivo TAG, por lo tanto el mejorar la interoperabilidad permitirá que los automovilistas tengan una mayor oferta para poder consumir y utilizar este método de pago en las plazas de cobro del país.

Por consiguiente, si la interoperabilidad se rige por una serie de procedimientos, reglas y una arquitectura que cumpla con las necesidades y las expectativas para brindar un sistema de calidad, los automovilistas se beneficiarán del Telepeaje, permitiendo un control de la trazabilidad de sus transacciones de cada cliente y por otra parte podrán convivir distintos operadores de Telepeaje en un solo sistema de manera fiable, pero sobre todo el ahorro de tiempo al momento de realizar un cruce sobre una plaza de cobro.

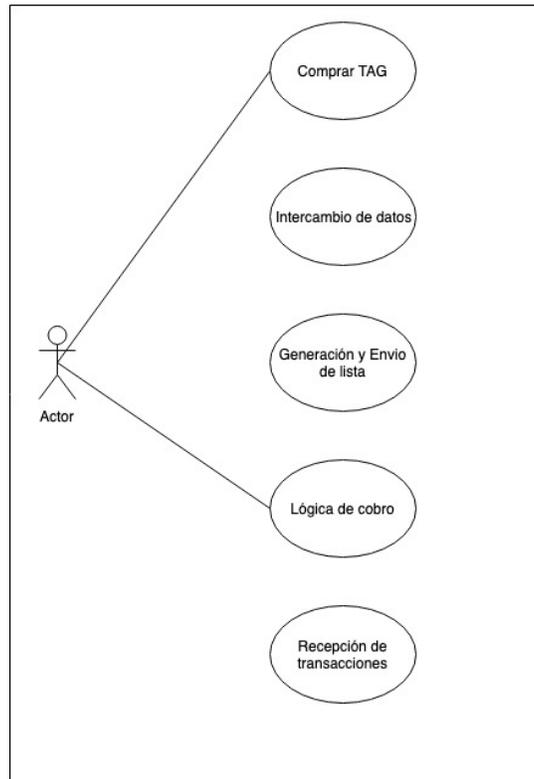
Por otra parte, el sistema de Telepeaje tiene una característica inherente para su correcto funcionamiento la cual consiste en la administración y la atención a clientes como una tarea esencial para que se pueda hacer un uso óptimo de la tecnología instalada en cada plaza de cobro, es otras palabras la administración del sistema de Telepeaje se realiza con una serie de herramientas que están disponibles dentro de un desarrollo de software definido como BackOffice, el cual se describirá más adelante. Sin embargo, como introducción es una herramienta de software dedicado para poder realizar el análisis de datos, generación de informes estadísticos, administración de clientes y monitoreo de tiempo real de las transacciones de cada automovilista.

Por lo tanto, como lo mencione anteriormente la optimización del objetivo de la interoperabilidad permitirá ofrecer un sistema de calidad que satisfaga las necesidades de los automovilistas y además poder cumplir con el desempeño esperado por el organismo encargado de la administración del Telepeaje que utilizará la arquitectura del sistema.

A continuación, el siguiente caso de uso explico la lógica del flujo de la información para poder realizar el cobro de Telepeaje interoperable por medio de un TAG de algún OTI, el cual este compuesto de distintos procesos, en este caso de uso se ejemplifican para facilitar la introducción al lector.

---

<sup>2</sup> Interoperabilidad. Concepto que radica en poder intercambiar y relacionar la información de los clientes de distintos operadores de Telepeaje con el propósito de que el usuario pueda realizar el pago electrónico de peaje con un solo TAG y con un solo contrato, en la mayoría de las autopistas o tramos carreteros.



*Ilustración 2. Caso de uso flujo de la información para el cobro de un cruce*

1. **Comprar TAG:** Este proceso es independiente de la lógica de cobro a nivel del carril. El flujo de información comienza cuando el usuario realiza la compra y activación de un TAG de algún operador de Telepeaje, cada operador tiene sus medios particulares para la venta de los dispositivos TAG, así como la recepción de depósitos de dinero que realizará cada automovilista para poder tener saldo en su dispositivo que acaban de adquirir.
2. **Intercambio de datos:** Para este proceso intervienen dos actores que son el proveedor y el operador de Telepeaje. El objetivo de intercambiar las transacciones de los clientes de cada OTI es poder concentrar y homologar los saldos y datos de cada TAG así el proveedor puede comenzar con los procesos de interoperabilidad.

El intercambio de información se realiza por medio del protocolo FTP (se describirá más adelante su funcionamiento), cada proveedor comparte las transacciones de sus clientes en archivos con XML con los datos necesarios, como saldo, número de TAG, tipo de TAG y clase vehicular asociada.

Cabe señalar que existe la posibilidad que en un intercambio de información exista inconsistencia entre los datos por ambas partes, debido a las diferentes reglas de negocio, o la interpretación de datos como puede ser formatos para el manejo de números. Por lo tanto, es importante homologar los datos para que tengan el mismo significado para los dos sentidos de intercambio.

- 3. Generación y envío de lista:** Este proceso depende plenamente del proveedor de Telepeaje. Consiste en un proceso que se ejecuta de manera automatizada el cual se encarga de aplicar las reglas de negocio que define CAPUFE para poder crear la lista de clientes validos que podrán realizar cruces en la red de autopistas de FONADIN. Este proceso tiene como finalidad generar como salida una lista de clientes validos que se enviará a cada plaza de cobro, dicho archivo es conocido como lista blanca. Dentro de la lista blanca están aplicadas distintas reglas negocio, sirva de ejemplo, clientes exentos, clientes con descuentos, exclusión de clientes fraudulentos, etcétera.
  
- 4. Lógica de cobro:** La lógica de cobro consiste en un proceso que implica la interacción con distintos tipos de hardware para realizar el cobro de forma automática sin la necesidad de efectivo, se necesita al menos de una antena que permita leer el TAG. Sin embargo este no es el único hardware que está implicado al momento de realizar un cobro, también existe un sistema de clasificación vehicular conocido como preclasificación el cuál por medio de hardware colocado estratégicamente al inicio de un carril, permite reconocer que clase de vehículo es el que está entrando, es decir puede identificar la diferencia entre una motocicleta, automóvil o un camión y en base a esta clasificación identificada se puede saber qué tipo de tarifa aplica al vehículo y además si el TAG tiene el saldo mínimo correspondiente a la clase vehicular, permitirá el paso y realizará el pago de manera automática. Cabe señalar que el proceso de clasificación no es el objetivo de este informe depende otros factores y la arquitectura de hardware que puede variar sin embargo el principio de clasificación más adelante explicaré, por otra parte, el proceso de la lectura de la antena es un módulo de software que se explicará de forma genérica.
  
- 5. Recepción de transacciones:** Por último, este proceso se encarga de recolectar y almacenar las transacciones que los usuarios realizaron después de haber hecho cruces en alguna plaza de cobro. El almacenamiento de los cruces se hace sobre una base de datos local en cada plaza de cobro, cabe señalar que cada base de datos está instalada en una configuración de servidores que se explicará más adelante y que realiza una replicación a una base de datos central la cual almacena y concentra la información de todas las plazas de cobro de la red de autopistas de FONADIN.

## Metodología

Para poder desarrollar un proyecto de este tipo, es importante realizar un análisis sistemático que permita estructurar y planear los procesos involucrados en el desarrollo de software y el flujo de la planificación en el tiempo. Para ello existen distintos tipos de metodologías que nos permiten gestionar la administración de un proyecto.

Sin embargo, para el desarrollo del software de este proyecto que tiene como alcance este informe se aplicó una metodología en específico debido a que existen variables como el tiempo y el constante cambio de requerimientos que afectan directamente en el desarrollo del software. Por ello expondré algunas metodologías que se utilizan para la gestión de proyectos de software, comparando las características de cada una y al final elegiré la que más se ajuste a la definición del problema que involucra este trabajo.

## Metodologías Ágiles

Las metodologías ágiles son una buena elección cuando se trabaja con requisitos desconocidos y variables. Si existen requisitos claros, existe una gran posibilidad de tener un diseño estable y de seguir un proceso totalmente planificado, es decir que el objetivo del proyecto no vaya a variar en tiempo ni en recursos.

Este tipo de metodologías ágiles proporcionan una serie de pautas y principios junto a técnicas prácticas que pueden orientar la dirección en la que se dirige el proyecto, permitiendo estandarizar procesos para poder generar módulos funcionales de software.

A continuación, describiré los tres tipos de metodologías ágiles más populares para la gestión de proyectos de software.

## Programación Extrema (XP)

La programación extrema propuesta por Kent Beck<sup>3</sup> se compone de cuatro fases en las cuales existen procesos internos que permiten definir de manera clara cada uno de los roles involucrados en el desarrollo del software.

La primera fase se conoce como planeación la cual consta en definir las historias de usuario, es decir es una técnica en la cual el cliente describe brevemente las funcionalidades que el sistema debe cumplir para poder satisfacer sus requerimientos, el objetivo de la primer fase de planeación es poder determinar las necesidades y los aspectos inherentes del proyecto, como la prioridad, la complejidad, el riesgo y la estimación del tiempo necesario poder generar un entregable o lanzamiento de los requerimientos del cliente.

Durante la planeación se realizan reuniones entre el cliente y el equipo de desarrollo para realizar un plan de trabajo en base a las historias de usuario recabadas, la finalidad de esta fase es poder establecer, agrupar y definir el orden en que serán entregadas las historias de usuario, esta fase es corta.

Para la segunda fase de diseño se implementará las historias de usuario con un enfoque lo más simple posible. La finalidad de realizar un diseño simple radica optimizar los recursos como el tiempo y valor del trabajo para en el desarrollo del software, debido a que en el futuro se puede complicar el desarrollo del software debido a un mal análisis al inicio del proyecto.

La tercera fase es la codificación, en esta fase se desarrollan las historias de usuario y se traducen a funcionalidades específicas de programación para poder generar un entregable al cliente, sin embargo, como las historias de usuario no tienen el suficiente detalle de los requerimientos es importante que el cliente tenga un rol activo en el desarrollo dando una retroalimentación sobre cada entregable realizado para así poder cumplir con sus necesidades y sus expectativas.

Por último, en la fase de pruebas, se entregan módulos funcionales y sin errores, que han sido expuestos a un entorno de desarrollo conocido como QA<sup>4</sup> para poder asegurar la calidad de la funcionalidad añadida al software. Durante esta fase ya no se realizan desarrollos funcionales, pero pueden ser necesarias tareas de ajustes mínimos.

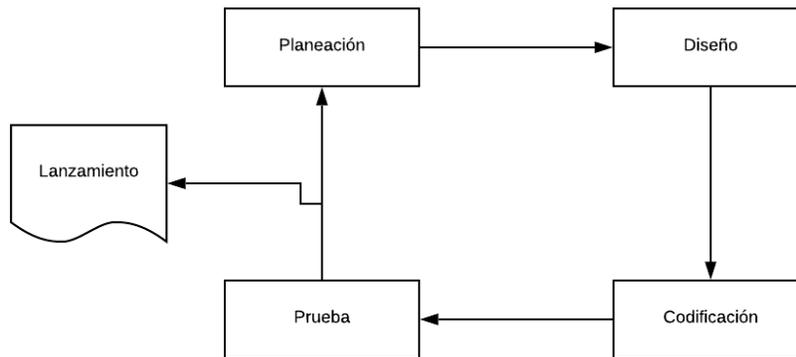
Por otra parte, el ciclo de vida de un proyecto desarrollado con esta metodología ofrece un ciclo de vida dinámico debido a que en muchos casos los clientes no son capaces de definir claramente los requerimientos al inicio del proyecto, por este motivo los ciclos de

---

<sup>3</sup> Kent Beck. Nació el 31 de marzo de 1961, es un ingeniero de software estadounidense, uno de los creadores de la metodología ágil programación extrema para el desarrollo de software.

<sup>4</sup> Por sus siglas en inglés Quality Assurance. Es un ambiente de pruebas con las mismas características que el ambiente de producción

desarrollo son cortos y son conocidos como iteraciones, los cuales consisten en generar entregables funcionales al finalizar cada iteración.



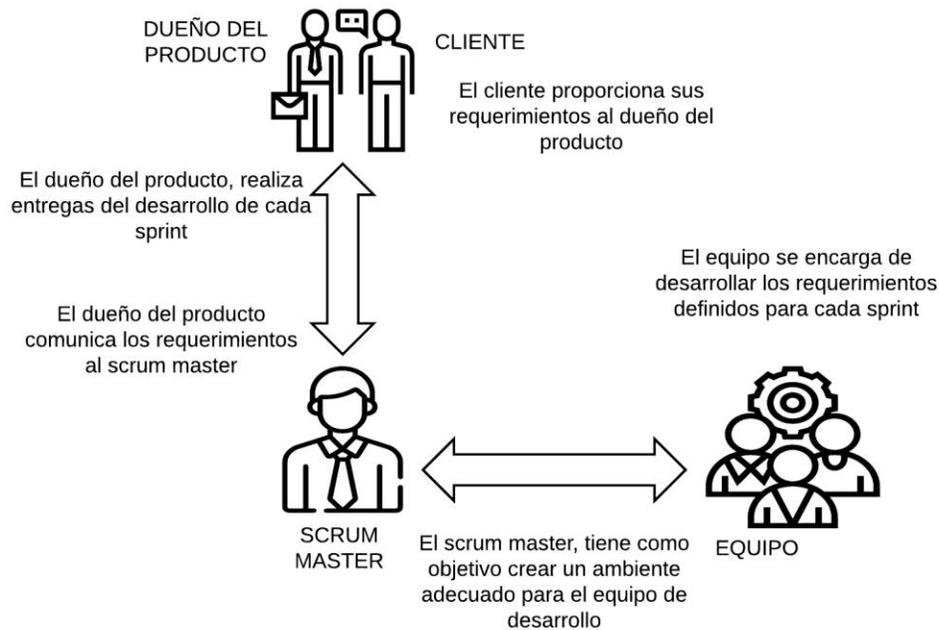
*Ilustración 3. Esquema Programación XP*

## Metodología Scrum

El concepto de la metodología Scrum consiste en distintos roles, eventos y reglas necesarios para un correcto funcionamiento, en el cual cada componente tiene un propósito en específico durante el desarrollo del proyecto. Esta metodología permite aplicar tareas y técnicas por cada rol, para poder definir de manera clara cada responsabilidad y así poder generar un producto final.

El equipo scrum está formado por los roles de ScrumMaster, Product Owner y el Equipo de desarrollo (Development Team).

- Scrum Master: Este rol es el más importante debido a que actúa como el intermediario entre el dueño del producto y como el líder del equipo de desarrollo, filtrando todas las actividades pertinentes de cada entregable y reorganizando los requerimientos. Además, debe de asegurar que los miembros del equipo sigan correctamente los valores y normas de Scrum.
- Product Owner (Dueño del producto): Es una única persona que interactúa con el cliente (Stakeholder) recibiendo los requerimientos solicitados para el desarrollo del proyecto, debe de comprender las necesidades del cliente y del equipo de desarrollo. Además, hace entrega de los módulos desarrollados en cada sprint.
- Development Team (Equipo): Es un grupo de personas multifuncionales que tiene como responsabilidad desarrollar los requerimientos proporcionados por el ScrumMaster.



*Ilustración 4. Roles Scrum.*

La entrega del proyecto se hace por medio de una serie de iteraciones también conocidas como sprints, en cada iteración se crean módulos funcionales para que el cliente pueda conocer el avance del proyecto.

El sprint está compuesto de los siguientes elementos:

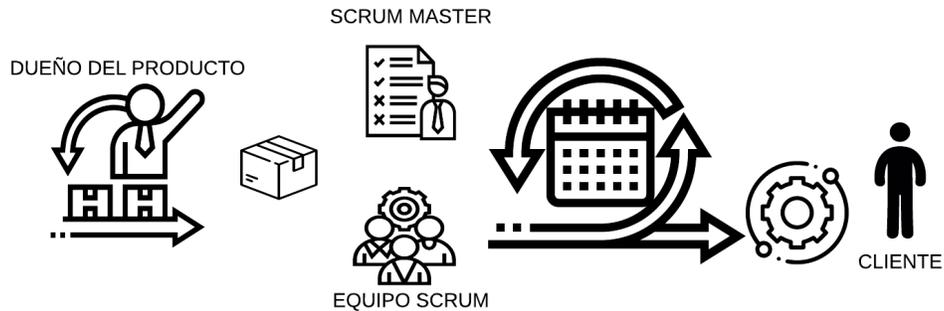
**Planeación del Sprint:** Es una reunión que tiene como objetivo elaborar un plan de trabajo para definir cómo se va a lograr desarrollar los entregables de cada sprint.

**Daily Srum:** También es una reunión del equipo de desarrollo, sin embargo, es una reunión corta que se realiza cada día para explicar lo que se ha logrado desde la última reunión, lo que se hará y los obstáculos que se han presentado en el sprint.

**Revisión del Sprint:** Esta etapa se realiza cuando un sprint ha finalizado, en el cual el dueño del producto interviene y revisa el producto que se generó.

**Retrospectiva del Sprint:** Es una reunión del equipo Srum para tomar conclusiones y generar una retroalimentación de los procesos bien aplicados y los que no.

En la siguiente ilustración se describe el flujo de la metodología SRUM.



*Ilustración 5. Metodología SCRUM*

## Selección de la metodología

Para la elección de la metodología es importante tener de manera clara los factores implícitos del proyecto como el riesgo, el tiempo disponible, los recursos humanos y económicos, el tener claro estos factores facilita la elección de una metodología, sin embargo, es importante mencionar que la elección de una forma de trabajo no es garantía del éxito de un proyecto. Después de presentar las metodologías populares de la actualidad, he decidido aplicar una la metodología de programación extrema que explicaré a continuación.

En la siguiente tabla realizo una comparación de las principales características de cada metodología antes propuestas y posteriormente explicaré porque la programación extrema se adapta a las necesidades del proyecto.

<b>SRUM</b>	<b>PROGRAMACIÓN EXTREMA</b>
Las iteraciones de entrega son de dos a cuatro semanas y se conocen como sprint	Las iteraciones son de una a tres semanas
Al finalizar un sprint, las funcionalidades que se han entregado y que han tenido la conformidad del cliente ya no se vuelven a tocar en ningún momento, porque ha sido aprobado.	Las tareas que se van terminando en las diferentes entregas al cliente, son susceptibles a modificaciones durante el transcurso de todo el proyecto, incluso después de que funcionen correctamente.
El equipo de desarrollo trata de seguir la prioridad de las tareas definidas, sin embargo, si identifican que es mejor modificar la prioridad para el desarrollo, pueden hacerlo.	El equipo de desarrollo sigue estrictamente el orden de prioridad que tiene cada tarea que ha sido definido por el cliente
La metodología SCRUM tiene un enfoque adaptable en la administración de proyectos en general.	En cambio, la programación extrema se centra en la programación o creación de un producto de software.
Cada miembro del equipo de desarrollo trabaja con un enfoque individual.	Los miembros del equipo de desarrollo trabajan en conjunto con el resto de sus compañeros.

*Tabla 1. Características De Metodologías.*

La principal característica del proyecto por la cual he decidido utilizar una metodología ágil de programación extrema es el constante cambio de los módulos entregables que posteriormente de haber sido generados seguía existiendo un desconocimiento a nuevas necesidades que se originan por parte del cliente, debido a que la naturaleza del sistema involucra distintos actores que influyen en la operación del Telepeaje además que se deben de generar resultados a corto plazo para no incrementar atrasos en la generación de saldos de los automovilistas.

Por lo tanto, la elección de la metodología de programación extrema es en base a la constante modificación del software y la generación de resultados en periodo de tiempo corto.

Como lo mencione anteriormente se dispone de un tiempo corto para la entrega de resultados hacia el cliente y para los automovilistas, la principal consecuencia de generar incumplimientos debido a prolongación de los resultados consiste que en pérdidas económicas y sobre todo una afectación directa en términos de servicio para cada plaza del cobro del país, por lo tanto, una manera de agilizar la generación de entregables es con iteraciones más cortas que es una característica de la programación extrema.

Dicho lo anterior estos dos factores influyen de manera directa en el desarrollo del software, porque posteriormente de haber finalizado un entregable existe la posibilidad de realizar modificaciones a lo que se ha aprobado y que es completamente funcional. Sirva de ejemplo los fraudes que realizan algunas personas para evitar pagar la tarifa correspondiente al transitar alguna carretera del país, debido a que generalmente existen usuarios que buscan de manera constante nuevos métodos para evitar realizar un pago del peaje, es por esto por lo que el software siempre se estará adaptando a las necesidades del proyecto.

Por eso es por lo que después de cada iteración el equipo de desarrollo debe de estar preparado para trabajar en tiempos cortos donde los requerimientos del cliente pueden llegar a ser ambiguos debido a que no los expresa de forma precisa sus necesidades, esto puede llevar a una pérdida de recursos debido a que se tiene de rehacer el trabajo o en el peor de los casos volver a empezar, porque no se cumplen con las expectativas del cliente.

## Principales Tecnologías Utilizadas

Posteriormente de definir los requerimientos del cliente habría que analizar las herramientas necesarias que se adaptan al desarrollo del software para poder lograr los objetivos, por ello a continuación mencionaré las principales tecnologías y sus características para ser utilizadas para el desarrollo del software.

**Base de datos.** El componente principal para el desarrollo del sistema es la base de datos debido a que en la actualidad es imprescindible tener un software para el manejo de información porque sin esta herramienta no se podría almacenar, organizar y procesar las transacciones generadas del sistema de Telepeaje.

Existen distintos tipos de modelos para el diseño de una base de datos, por ejemplo, base de datos relacionales, no relacionales, columnares, en memoria (in memory).

Sin embargo, para la implementación de la base de datos se optó por elegir el modelo relacional, debido a las condiciones del proyecto porque la información está distribuida en diferentes puntos y necesita ser centralizada en tiempo real, es decir se necesita de una replicación de datos, es por esto por lo que el modelo relacional se adapta a este tipo de necesidades.

Por otro lado, existen otros tipos de gestores de bases de datos con los cuales se puede trabajar el modelo entidad relación, sin embargo, se utilizó el gestor Oracle debido a que cuenta con una herramienta de software dedicada a la replicación de datos.

Dicha herramienta es un componente fundamental para intercambiar información en tiempo real entre las distintas plazas de cobro y así poder tener los datos de los automovilistas en tiempo real.

**Replicación de base de datos.** Este concepto consiste principalmente en el intercambio de información entre distintas bases de datos a través de una red que implícitamente genera ventajas importantes en su aplicación las ventajas son las siguientes:

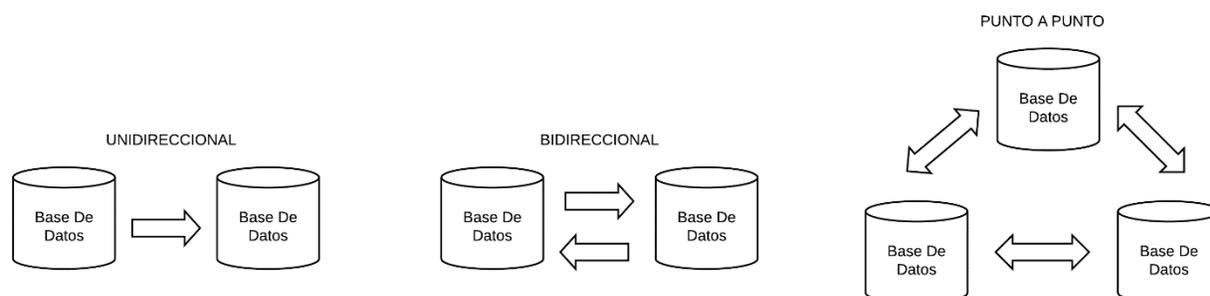
**Disponibilidad:** Esta característica permite garantizar el acceso a la información en caso de que algún servidor presente una falla de cualquier tipo, por ejemplo, de comunicación o una falla de hardware, debido a que la información se replica a distintas partes y puede ser utilizada desde algún otro punto de acceso remoto.

**Respaldo:** Al realizar una replicación los datos son distribuidos en uno o más servidores generando de manera implícita distintas copias que poseen las últimas actualizaciones de la información generada, permitiendo que en algún siniestro se pueda disponer de una copia.

**Tiempo Real:** Este es el punto clave y más importante para las necesidades del proyecto, debido a que los datos se encuentran actualizados de manera prácticamente inmediata, asegurando que las transacciones ejecutadas y los movimientos de los clientes sean aplicados sin importar en lugar donde origine.

## Oracle Golden Gate

Oracle Golden Gate es una herramienta de software que proporciona el intercambio y la manipulación de datos en tiempo real, permitiendo generar una alta disponibilidad en los datos, existen distintas configuraciones de topologías para poder implementar la replicación de datos las cuales describiré en esta sección.

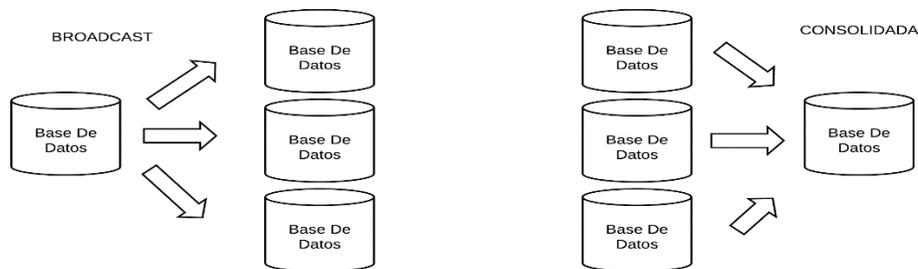


*Ilustración 6. Replicación Unidireccional, Bidireccional y Punto a Punto.*

**Unidireccional:** Este tipo de replicación unidireccional es la configuración más sencilla debido a que su objetivo es copiar las transacciones de una base de datos principal a una base de datos secundaria, con la característica que las únicas modificaciones que se replican son las de la base de datos primaria, es decir si se realiza una modificación en la información de la base de datos secundaria, la base primaria no se verá afectada.

**Bidireccional:** La replicación bidireccional permite enviar todas las transacciones entre ambos sentidos, por ejemplo, principal-secundario o secundario-principal. Cabe mencionar que esta topología se presenta únicamente cuando existen dos servidores.

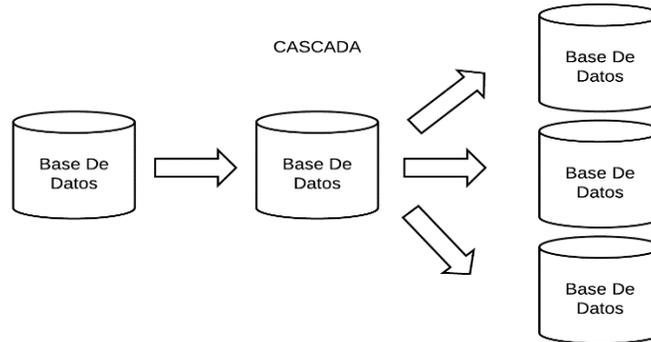
**Punto a punto:** Esta topología parte del principio de la replicación bidireccional, que consiste en enviar las transacciones a cada uno de los servidores disponibles con la única diferencia que existen más de dos servidores, las transacciones replicadas en cada servidor permiten generar una alta disponibilidad en los datos ya que los cambios de los datos se aplican en tiempo real.



*Ilustración 7. Replicación Broadcast y Consolidada.*

**Broadcast:** Esta replicación consiste en un servidor central en cual se almacenan los datos y deben de ser distribuidos por medio de la replicación a distintos servidores secundarios en un único sentido principal-secundario. Las modificaciones de los datos de los servidores secundarios no deben de afectar al servidor central, es decir los servidores secundarios tienen como objetivo únicamente la lectura de información en tiempo real.

**Consolidada:** En la replicación consolidada existen distintos servidores que están replicando las transacciones de los datos a un único servidor con el objetivo de centralizar la información. Como se ha mencionado anteriormente el sentido de la replicación depende de la topología, pero en este caso la replicación va en sentido secundario-primario.



*Ilustración 8. Replicación Cascada.*

**Cascada:** La replicación en cascada consiste en el mismo principio de la replicación Broadcast con la diferencia del concepto de escalabilidad, dicho de otra manera, existe un servidor intermedio se tiene como propósito crear un nodo entre el servidor principal y el resto de los servidores donde se replican los datos, esto con el objetivo de generar un incremento en la seguridad de los datos y del sistema, es decir crear un honeypot.<sup>5</sup>

## Tipo de replicación implementada.

El tipo de replicación elegida para la implementación del proyecto es una combinación del tipo Broadcast y Consolidada, porque el flujo de la información debe de ser bidireccional entre cada plaza de cobro y el centro de gestión. Por lo tanto, al ser bidireccional se genera una difusión de cada transacción de los automovilistas cuando realizan un cruce o cuando realizan algún deposito a su TAG de su operador de telepeaje y por lo tanto así se puede tener un control en tiempo real de lo que ocurre en cada tramo carretero.

El siguiente ejemplo sirve para explicar un caso real de la replicación de la base de datos.

Supongamos que un automovilista realiza un cruce en la plaza de cobro A, posteriormente de hacerlo esa transacción ha sido registrada en la base de datos de la plaza y después de ser almacenada, debe de ser procesada para poder realizar el cobro del saldo correspondiente a la clase vehicular sobre el TAG del automovilista, por lo tanto, la

<sup>5</sup> Honeypot. Es un método estratégico, diseñado para para llamar la atención de posibles ataques maliciosos y así poder engañarlos para desviar el objetivo principal del ataque.

transacción debe de ser replicada en el sentido plaza-gestión. Ya en el centro de gestión, se aplican las reglas de negocio para actualizar el nuevo saldo del cliente.

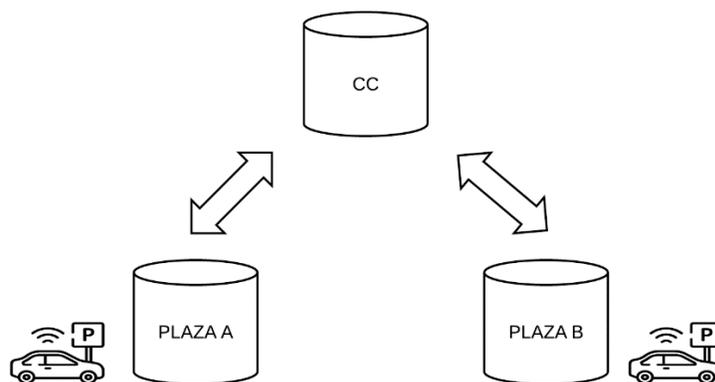
De modo que cuando el automovilista realice un cruce por la plaza de cobro B tenga una actualización de su saldo, en base al saldo cobrado del cruce previamente realizado en la plaza A.

Cabe mencionar que la actualización del saldo del automovilista es aplicada para cualquier plaza de cobro. Sin embargo, para facilitar el ejemplo anterior solo se ejemplificaron dos plazas de cobro.

En la siguiente ilustración se observa gráficamente el ejemplo anterior:

Centro de Gestión - Plaza de Cobro: Broadcast

Plaza de Cobro – Centro de Gestión: Consolidada



*Ilustración 9. Aplicación de replicación.*

Por otra parte, otro ejemplo del uso de la replicación bidireccional es el caso cuando un cliente realiza un depósito en su TAG.

Para poder realizar el depósito se debe de efectuar por medio de algún método de recarga de saldo que disponga el OTI, para esto el automovilista realiza la actualización de saldo de distintas formas, por ejemplo, por transferencia bancaria, uso de aplicación o depósitos en algún establecimiento. Posteriormente el saldo actualizado se debe de aplicar en todas las plazas de cobro, por lo tanto, la actualización de esta información debe de seguir la lógica de la replicación en sentido gestión-plaza.

## Sistema Operativo.

La elección del sistema operativo que se instaló en todos los servidores de cada plaza y centro de gestión depende directamente de la replicación de la base de datos debido a que el motor de la base es Oracle 18 y la herramienta de replicación es Gold Gate por lo tanto para un funcionamiento óptimo en base a la documentación de Oracle<sup>6</sup> se recomienda utilizar el sistema operativo Oracle Linux, el cual es una distribución basada en Red Hat.

Además de que el motor de la base de datos Oracle esta desarrollado en este sistema operativo las principales ventajas de Oracle Linux son es que está construido a partir del código fuente de RHEL (Red Hat Enterprise Linux) que es un sistema que presenta un conjunto de aplicaciones que permiten el aumento de disponibilidad para las aplicaciones en cuanto a la infraestructura requerida.

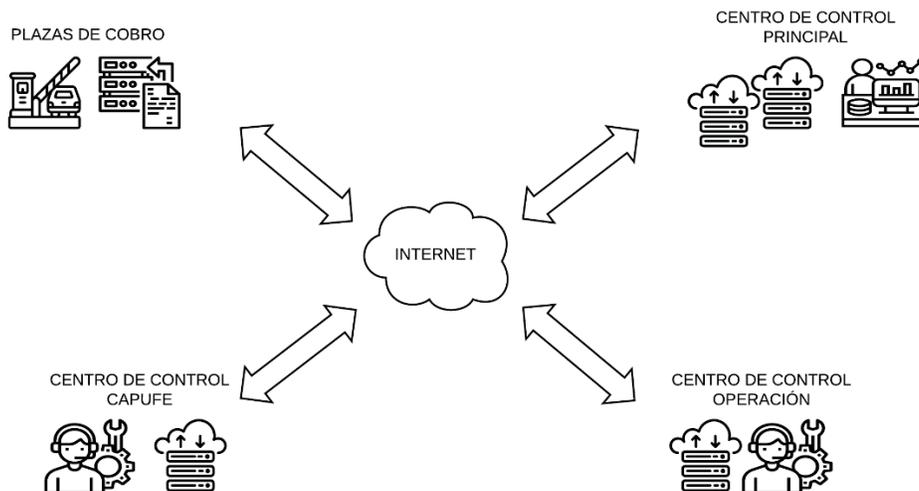
Oracle Linux se diferencia de RHEL por que incluye el denominado Unbreakable Enterprise Kernel, el cual esta específicamente optimizado para obtener mayor estabilidad y rendimiento, pero manteniendo la compatibilidad con la mayoría de las aplicaciones comerciales para el desarrollo de software, el principal objetivo de este Kernel es brindar una mayor seguridad, mejorar las llamadas al sistema y las mejoras de las librerías en el lenguaje C para acelerar las aplicaciones en el tiempo de procesamiento de consultas.

## Arquitectura

Como lo he mencionado anteriormente para que el sistema de Telepeaje funcione de acuerdo con las expectativas de las necesidades se requiere de una arquitectura que garantice la disponibilidad de la información en tiempo real en los distintos tramos carreteros para que los automovilistas puedan hacer uso del dispositivo TAG. Por lo tanto, describiré la arquitectura principal del sistema de Telepeaje.

---

<sup>6</sup> <https://www.oracle.com/a/ocom/docs/linux-for-oracle-database-wp-2068570.pdf>



*Ilustración 10. Arquitectura.*

**Plaza de Cobro:** La plaza de cobro es el sitio donde los automovilistas hacen uso del dispositivo TAG realizando cruces en los carriles exclusivos de Telepeaje. Posteriormente de haber realizado un cruce en la plaza, los datos asociados a cada transacción pasan a almacenarse en una base de datos relacional que está instalada localmente en cada plaza, más adelante estos datos serán replicados en tiempo real hacia distintas bases de datos principales que tiene como objetivo centralizar las transacciones de todos los tramos carreteros.

**Centro de Gestión Principal:** En este punto se centralizan todas las transacciones de cada una de las plazas de cobro de los diferentes tramos carreteros, por medio de la replicación de datos. El objetivo de este punto es verificar que las transacciones se hayan cobrado de manera correcta en base a la categoría del vehículo que cruzó por cada plaza, además funciona como medio de comunicación con los diferentes operadores de Telepeaje para el intercambio de información.

**Centro de Gestión CAPUFE:** En el centro de gestión de CAPUFE de igual manera se centralizan todas las transacciones de cada una de las plazas de cobro para realizar el mismo proceso, verificar y comprobar que los cruces hayan sido cobrados de manera correcta de acuerdo con la categoría vehicular.

**Centro de Control Operación:** También definido como el centro de operación aquí se reciben las transacciones de las plazas de cobro el principal el objetivo de este centro es poder dar seguimiento y soporte a las aclaraciones por parte de los clientes y también poder supervisar el estado de las plazas de cobro.

Cabe mencionar que estos cuatro principales módulos se rigen por la replicación de base de datos previamente seleccionada. Por lo tanto, una transacción que se aplique en cualquier centro o plaza de cobro se verá reflejada para el resto de los actores.

## Sistema de cobro

Para comprender de forma clara el funcionamiento de la infraestructura instalada para poner en marcha el sistema de Telepeaje, es necesario explicar los diferentes procesos y sus tareas específicas que tiene cada uno para que el sistema funcione de manera ideal.

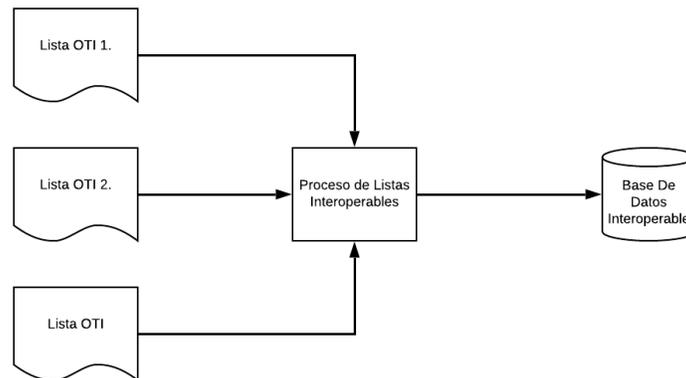
Cabe mencionar que estos procesos deben de ser monitoreados, las veinticuatro horas del día para poder ofrecer al cliente una alta disponibilidad del sistema.

El monitoreo del sistema es una tarea independiente del desarrollo del software, sin embargo, es importante mencionarla debido que permite ofrecer al cliente una solución en tiempo real de cualquier eventualidad que pueda surgir dentro de los procesos que explicaré a continuación.

## Centralización de información.

Tomando como punto de partida el caso de uso de la sección *Definición del problema*, existe un proceso definido como “*Intercambio de datos*”, el cual tiene como objetivo el almacenamiento de la información de todos los clientes, independientemente del tipo de OTI, es por ello por lo que debe de existir un mecanismo que realice esta tarea de manera automatizada.

El concepto del proceso de centralización se puede comprender mejor con la siguiente ilustración. Cada operador de Telepeaje interoperable comparte una lista con los datos de cada uno de sus clientes para ser almacenada y centralizada.



*Ilustración 11. Esquema de Centralización de datos.*

## Recepción de información con OTIS

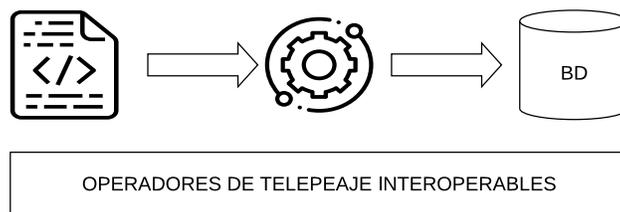
Para poder centralizar la información de los clientes de cada Operador de Telepeaje Interoperable, se debe de homologar los tipos de datos previamente a ser almacenados, esto con la finalidad de poder trabajar en el mismo formato de información y evitar confusiones o inconsistencia en los datos, por ejemplo, las categorías vehiculares pueden variar en terminología dependiendo de cada OTI.

### Proceso de listas Interoperables:

Este mecanismo recibe como entrada las listas que envía cada Operador de Telepeaje por medio de un Web Service<sup>7</sup> donde se deposita archivos XML con los datos asociados que le pertenecen a cada cliente, los principales datos son: Número de serie del TAG, saldo, fecha de depósito, tipo de TAG y clase vehicular.

Los XML pasan a ser homologados por un proceso automático que analiza el archivo XML y posteriormente inserta o actualiza en la base de datos interoperable la información de cada automovilista, con el objetivo que toda la información entrante pueda ser tratada de forma igual sin importar de origen de cada Operador.

<sup>7</sup> Webs Service. Método de comunicación entre varias máquinas conectadas por medio de una red con el objetivo de intercambiar información.

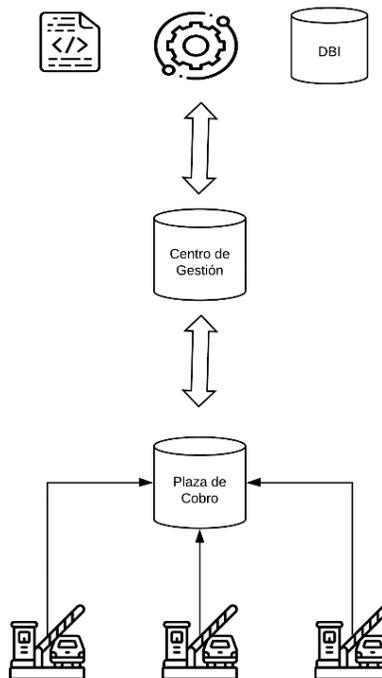


*Ilustración 12. Intercambio de listas.*

## Envío de transacciones

Por otra parte, los Operadores de Telepeaje Interoperables deben de recibir cada una de las actualizaciones que se realizan en los diferentes tramos carreteros del país al momento que cada automovilista hace uso de su TAG para poder realizar el cobro del saldo correspondiente a cada uno de sus clientes.

Para efectuar el envío de las transacciones que realiza cada cliente en cualquier carril, el flujo de las transacciones comienza en la base de datos, que está instalada en cada plaza de cobro y posteriormente es replicado hasta el centro de gestión, en este nivel el OTI se encarga de recuperar las últimas actualizaciones de sus clientes.



*Ilustración 13. Envío de transacciones.*

El proceso de envío de transacciones se efectúa también de manera automatizada para agilizar el cobro del TAG, consiste en generar archivos XML independientes para cada Operador de Telepeaje Interoperable que contiene las últimas transacciones de cada uno de sus clientes, dichos archivos son tomados por cada OTI para ser procesados y realizar el cobro correspondiente.

## Generación lista interoperable

Otro de los procesos principales para la arquitectura del sistema, es definido como la generación de lista interoperable o lista blanca, ya que sin esta lista no se podría realizar el cobro electrónico. Este proceso consiste en tomar los datos de todos los clientes almacenados por el proceso de intercambio de listas con las últimas actualizaciones para generar una única lista de clientes que será utilizada en tiempo real en todas las plazas de cobro, para que los automovilistas puedan cruzar por algún carril. En otras palabras, la lista interoperable contiene las últimas actualizaciones de saldos de los clientes.

De modo que cuando un automovilista este en un carril, previamente para realizar el pago electrónico, el hardware del carril por medio de la lectura de la antena pueda consultar el saldo del TAG dentro de la lista interoperable al momento del cruce. El resultado de la lectura de la lista interoperable dictaminará si el saldo del TAG del cliente es lo suficiente para poder realizar el pago del peaje, en caso de que el saldo se insuficiente, no se permitirá cruzar al automovilista en la plaza de cobro.

La generación de la lista interoperable se realiza al nivel de cada una de las plazas de cobro, por medio de un proceso que se ejecuta en un determinado tiempo de manera automática, es decir, cada lista interoperable se actualiza constantemente con las ultimas transacciones de saldos que están en la base de datos, que son enviadas por medio de la replicación de datos.

## CRON

Debido a que la generación de la lista interoperable, el envío y recepción de transacciones es una tarea que se debe de ejecutar de manera periódica y automática, por consiguiente, se debe de tener una administración para la ejecución independiente de cada proceso, por otra parte, como lo mencione anteriormente el desarrollo del software se realizó sobre una distribución de Linux en consecuencia para este tipo de sistemas operativos existe una herramienta que permite dar una solución a la problemática mencionada.

CRON es un software administrador que permite regular la ejecución de procesos de manera automática para los sistemas operativos Linux, por ejemplo, facilita la ejecución de una o varias tareas en un determinado tiempo tales como cada minuto, una vez a la semana, o una vez al mes, etc.

La configuración de las tareas programadas está definida el archivo crontab<sup>8</sup> que tiene la siguiente estructura:

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/
```

---

<sup>8</sup> El archivo de configuración crontab se encuentra en la ruta /etc/crond.d/crontab o /etc/crontab y puede variar dependiendo de la distribución del sistema operativo Linux.

```
.----- minuto (0-59)
| .----- hora (0-23)
| | .----- día del mes (1-31)
| | | .----- mes (1-12) o jan,feb,mar,apr,may,jun,jul... (meses)
| | | | .---- día de la semana sun,mon,tue,wed,thu,fri,sat
| | | | |
* * * * * comando o tarea a ejecutar
```

Tabla 2. Estructura del archivo crontab.

Como se observa en el ejemplo anterior, el archivo de configuración tiene la característica de poder definir la fecha y hora de cuando se pretende ejecutar una tarea.

Por lo tanto, para poder programar la ejecución de una tarea se debe de agregar los parámetros de minuto, hora, día del mes, mes o día de la semana.

Se ejecutará la tarea cada 5 minutos.

```
5* * * * /home/usuario/scripts/tarea.sh
```

Se actualizará los repositorios del sistema cada domingo a las 10 AM

```
00 10 * * 0 root apt-get -y update Usuario root
```

Para administrar la ejecución automática de tareas en un sistema Linux es de gran utilidad la herramienta CRON.

### Configuración de servidores

Por otra parte, la ejecución de todos los procesos de software involucrados en el desarrollo del sistema a nivel de la plaza de cobro y del centro de gestión, necesitan de recursos de hardware adecuados a las necesidades del proyecto, es por ello por lo que los servidores tienen una función importante en la arquitectura del sistema ya que sin los recursos necesarios el desempeño del software podría ser precario para las necesidades del proyecto, por consiguiente en esta parte describiré la configuración necesaria para la puesta en producción de los servidores instalados.

Uno de los principales retos a resolver para un sistema que debe de tener una alta disponibilidad, es poder asegurar que el sistema podrá tolerar en gran medida posibles fallas como causas físicas, naturales, de hardware o fallos en el suministro eléctrico. Es por esto por lo que existen configuraciones de hardware y software para evitar contingencias de este tipo, que afectan la disponibilidad y el funcionamiento del sistema de Telepeaje, a continuación, describiré las principales herramientas para poder actuar de forma automática a un posible fallo.

## RAID

Es común que un servidor llegue a presentar alguna falla en uno o varios discos duros de almacenamiento debido a que prácticamente estos dispositivos están propensos a un mayor desgaste y una serie de factores que pueden afectar su rendimiento, de modo que, existe una probabilidad de un problema, provocando pérdida de datos y sobre todo una indisponibilidad del sistema.

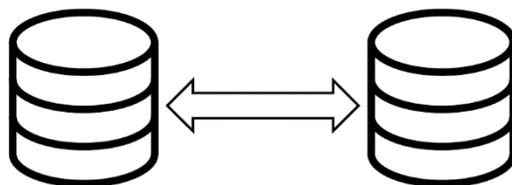
Es por esto por lo que existe una tecnología definida como RAID<sup>9</sup> que consiste en un proceso de combinar o unir varios discos duros con el objetivo que funcionen de manera coordinada formando una única unidad de almacenamiento. Esta configuración permite la tolerancia a las fallas más comunes de los discos duros, como sectores dañados o problemas mecánicos, debido a que se crea una redundancia de discos generando que la información este replicada en los diferentes discos de la configuración.

Existen diferentes tipos de configuraciones RAID, que poseen características de rendimiento específicas dependiendo de las necesidades requeridas, sin embargo, para la aplicación de este proyecto se implementaron las configuraciones RAID 1 y RAID 5 las cuales se explicarán a continuación.

**RAID 1.** Para poder realizar esta configuración se necesita de al menos dos discos para el almacenamiento, la principal característica de este arreglo es que se copian todos los datos en ambos discos, de modo que, si el primer disco llegase a fallar el sistema seguirá trabajando con el segundo disco sin pérdida de información. Esta configuración es óptima para la instalación del sistema operativo.

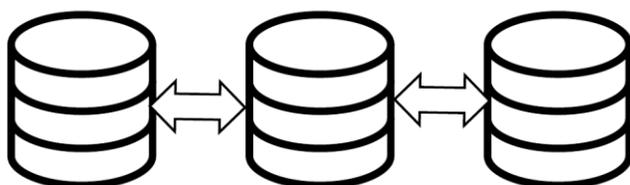
---

<sup>9</sup> Por sus siglas en ingles Redundant Array of Independent Disks.



*Ilustración 14. RAID 1.*

**RAID 5.** En esta configuración se necesita un mínimo de tres discos, utiliza la paridad para recuperar los datos. Se dividen los datos por bloques en los diferentes discos, de forma que, si hay un fallo en uno de ellos, esa parte de los datos se recupera con los datos almacenados en el resto de los discos, permitiendo el funcionamiento de manera normal. Esta configuración solo es tolerante a la falla de un disco.



*Ilustración 15. RAID 5.*

## Clúster de servidores.

Otro mecanismo que permite actuar de forma automática contra la posibilidad de alguna falla es definido como clúster que consiste en la unión de varios servidores que funcionan como si fuera uno solo, el concepto de clúster permite compartir los recursos de hardware y software con la finalidad de crear una alta disponibilidad del sistema.

Como lo mencione anteriormente, el sistema operativo fue una distribución de Linux, sobre el cual se implementó todo el desarrolló todo software, es por esto por lo que a continuación explicaré la creación de un clúster para este tipo de sistema operativo.

La herramienta dedicada a la tarea creación y configuración del clúster es DRBD<sup>10</sup> o por sus siglas en ingles *Distributed Replicated Block Device* que se basa en una distribución de recursos por medio de la red local. El principio de este software es permitir la replicación los datos entre alguna partición definida en cada servidor, con el objetivo poder tener la misma información en ambos servidores, es decir crear de alguna manera un servidor que funcione como espejo.

Para que el software DRBD funcione de acuerdo a lo necesitado además se debe de instalar y configurar otros dos paquetes de software, las cuales son:

- Corosync: Que se encargará de la comunicación entre los dos nodos del clúster.
- Pacemaker. Administra y gestionará los recursos software de cada clúster.

El primer paso para poder configurar el clúster es instalar los paquetes del software DRBD necesarios en el sistema operativo de cada servidor, por lo tanto:

```
yum install -y kmod-drbd84 drbd84-utils
```

Posteriormente de haber instalado los paquetes, se procederá a la configuración del software DRBD para que ambos servidores compartan los recursos de hardware. El archivo de configuración se puede encontrar en:

```
vim /etc/drbd.d/global_common.conf
```

```
global {
  usage-count yes;
}
common {
  net {
    protocol C;
  }
}
```

---

<sup>10</sup> DRBD Documentación oficial <http://oss.linbit.com/drbd>

En esta configuración es importante definir el protocolo que se utilizará para la replicación de datos, por lo tanto:

- **protocol A:** Protocolo de replicación asíncrono; Es el más utilizado para escenarios donde la replicación es a una larga distancia, en este protocolo para el intercambio de datos sobre el nodo primario y el secundario se debe de considerar terminadas las operaciones escritura para poder replicar los datos al nodo secundario.
- **protocol B:** Protocolo de replicación síncrono para servidores a largas distancias, en este protocolo las operaciones locales de escritura se dan como terminadas en cuanto el proceso de escritura comienza.
- **protocol C:** Comúnmente es empleado para nodos que se encuentran a una corta distancia en una red local, es el protocolo síncrono más utilizado en las configuraciones DRBD.

La elección del protocolo de replicación está directamente determinada por la seguridad de la información y la latencia de la red.

Por otra parte, se debe crear un archivo donde se definen los recursos que se compartirán

```
vim /etc/drbd.d/config_cluster.res
```

```
resource test {
    on hostname1 {
        device /dev/drbd0;
        disk /dev/sdb1;
        meta-disk internal;
        address 192.168.0.101:7789;
    }
    on hostname2 {
        device /dev/drbd0;
        disk /dev/sdb1;
        meta-disk internal;
        address 192.168.0.102:7789;
    }
}
```

Donde:

- **on hostname:** En este parámetro definimos el nombre del servidor sobre el cual serán aplicadas las configuraciones del clúster.
- **test:** Es el nombre del nuevo recurso, en este ejemplo se definió el nombre 'test'
- **device /dev/drbd0:** Especifica el nuevo bloque virtual del dispositivo para el almacenamiento que será administrado por DRBD
- **disk /dev/sdb1:** Es donde se define el dispositivo o la partición que posee cada servidor que será utilizado por DRBD para compartir el recurso de memoria.
- **meta-disk:** Define donde DRBD almacenara los metadatos, utilizar el parámetro 'internal' significa que almacenaran los metadatos en el mismo físico que los datos de producción.
- **address:** Se especifica la dirección IP y el puerto de cada nodo para establecer la comunicación

Una vez que se tenga la configuración anterior en cada nodo del clúster, se debe de tener definida la partición de memoria o el disco duro que funcionará para intercambiar los datos entre cada servidor, en otras palabras, cada servidor tendrá físicamente su espacio de memoria de manera independiente de modo que las particiones funcionarán en conjunto para que tener los mismos datos en cada nodo.

Por lo tanto, se deben de formatear cada partición. En este caso utilicé la herramienta fdisk

```
fdisk /dev/sdb
```

Después se procederá a iniciar la configuración en ambos nodos e iniciar el servicio de DRBD

```
drbdadm create-md config_cluster
```

Hasta este punto, el software DRBD está listo para operar, sin embargo, debemos indicar que nodo funcionará como maestro para la sincronización de discos. Por lo tanto, el siguiente comando solo se debe de ejecutar sobre el nodo que funcionará como maestro.

```
drbdadm primary --force test
```

Para comprobar es estado del correcto funcionamiento la configuración se utilizar.

```
drbdadm status test
```

Posteriormente de la comprobación se debe de iniciar la sincronización de discos, para esto se debe de establecer el formato del sistema de archivos en el dispositivo de almacenamiento.

```
mkfs -t ext4 /dev/drbd0
```

Una vez que la sincronización de discos haya finalizado, el recurso memoria está disponible para ambos nodos, por lo tanto, al realizar alguna operación de escritura sobre el nodo principal se replicará al nodo secundario permitiendo generar redundancia en la información y así poder actuar en caso de alguna falla.

## Split Brain

A pesar de ser el clúster de servidores un instrumento para actuar en situaciones de contingencia existe una desventaja para este tipo de configuración, debido a que cuando un clúster funciona normalmente un servidor siempre toma el puesto de nodo maestro y por lo tanto el servidor esclavo o pasivo detecta que ya existe previamente un nodo que asume el rol de como nodo activo.

Sin embargo, existen ocasiones donde los servidores no detectan el estado actual de los nodos del clúster y en consecuencia intentan asumir el rol de nodos activo esto provoca un error definido como Split Brain<sup>11</sup>.

Este error generará que al final el sistema no esté disponible debido a que ...

Para poder solucionar esta falla se debe de intervenir de manera manual para poder determinar que nodo funcionara como maestro, como se menciona anteriormente el software implementado para la ceración del clúster es DRBD, por lo tanto, para aplicar la solución al problema se puede realizar de la siguiente manera.

---

<sup>11</sup> Split Brain. Por su traducción en ingles puede definirse como "Cerebro Dividido" es un concepto que se aplica cuando en un clúster de servidor dos o más nodos asumen el rol de nodo maestro.

Posteriormente de definir que nodo quedara como secundario, se debe de ejecutar los siguiente sobre el servidor previamente definido para descartar los datos:

```
drbdadm secondary all  
drbdadm disconnect all  
drbdadm -- --discard-my-data connect all
```

Una ves hecho esto, se debe de ejecutar lo siguiente, pero para el nodo que asumirá el rol de maestro.

```
drbdadm primary all drbdadm disconnect all drbdadm connect all
```

## Disponibilidad

Dado que he mencionado de manera continua el concepto de disponibilidad de servidores tiene una importancia sumamente considerable debido a que alguna falla afectará la operación del Telepeaje y en consecuencia también impactará directamente a los automovilistas.

Es por esto por lo que en caso de falla se debe de asumir la responsabilidad de la afectación a los clientes, esto con el objetivo de ofrecer un servicio de calidad, actualmente en la mayoría de cualquier tipo de proyecto existe el termino de penalización que consiste en aplicar una infracción en caso de no cumplir con los estándares de desempeño, dicho en otras palabras, una penalización aplica cuando una autoridad determina un incumplimiento con base a lo establecido en un contrato. Así que para poder aplicar una penalización debe de existir un método para calcular un incumplimiento.

Por lo tanto, en esta sección explicaré un ejemplo totalmente ajeno al proyecto para poder calcular el tiempo y porcentaje de disponibilidad de servidores. Quiero mencionar que este conceto es importante ya que se adapta y se aplica a la mayoría de cualquier tipo de proyecto.

Los sistemas particularmente no tan simples para poder ser modelados, debido a que la naturaleza del sistema hará que este compuesto de múltiples componentes y cada uno ellos con un con distinto nivel de prioridad por componente.

Así que para poder cuantificar el concepto de disponibilidad existe un acuerdo de nivel de servicio conocido como SLA<sup>12</sup> que consiste en una herramienta que ayuda al cliente y al proveedor de servicios para definir el nivel acordado de calidad de un servicio. La aplicación de este concepto depende del tipo de sistema, por ejemplo:

Para un sistema no crítico puede definirse como un SLA de 8x5, es decir 8 horas por 5 días, o para un sistema con una alta criticidad puede ser un SLA de 24x365, es decir las 24 horas de todo el año. Por lo tanto, se puede calcular la disponibilidad con la siguiente fórmula:

$$Disponibilidad = \frac{A - B}{A} \times 100$$

Donde:

A = Horas comprometidas de disponibilidad en este caso 24x365 = 8,760 horas

B = Número de horas fuera de servicio. Por ejemplo, 15 horas por alguna falla y 9 horas por mantenimiento

$$Disponibilidad = \frac{8760 - 24}{8760} \times 100$$

$$Disponibilidad = 99.7626 \%$$

Cabe mencionar que el porcentaje de cumplimiento definido depende directamente del acuerdo entre ambas partes del cliente y el proveedor de servicios.

---

<sup>12</sup> SLA por siglas en inglés Service Level Agreement es un acuerdo que está constituido por el proveedor de servicios y su cliente para establecer aspectos como el tiempo de respuesta, disponibilidad y personal asignado para el servicio.

La siguiente tabla muestra el detalle del tiempo indisponible que se debe de tener para un sistema critico de 24x7 es decir para todo un año.

<b>Disponibilidad</b>	<b>Tiempo Fuera Año</b>	<b>Tiempo Fuera Mes</b>	<b>Tiempo Fuera Día</b>
<b>99.99%</b>	52.6 min	4.4 min	8.6 s
<b>99.95%</b>	4.4 horas	21.9 min	43.8 s
<b>99.90%</b>	8.8 horas	43.8 min	1.46 min
<b>99.50%</b>	1.8 días	3.66 horas	7.22 min
<b>99%</b>	3.7 días	7.3 horas	14.4 min
<b>98%</b>	7.3 días	14.6 horas	28.8 min
<b>95%</b>	18.3 días	36.5 horas	1.2 horas
<b>90%</b>	36.5 días	73 horas	2.4 horas

Alcanzar un porcentaje muy alto de disponibilidad es prácticamente un reto muy complejo debido a que existen factores externos que pueden influir en el momento de una falla. Además, la capacidad de poder resolver algún problema depende de factores como:

- La complejidad de la falla
- La disponibilidad del personal para la solución
- La comunicación entre el personal.

Por otra parte, para poder resolver los puntos anteriores y así mejorar la actuación frente a una falla, lo recomendable es elaborar un plan de contingencias donde se defina la prioridad de los componentes del sistema que son más propensos a una falla.

También otra forma de poder actuar es generar un historial de las soluciones a fallas anteriores con el objetivo de saber de mejor manera la causa de la falla y su correspondiente solución.

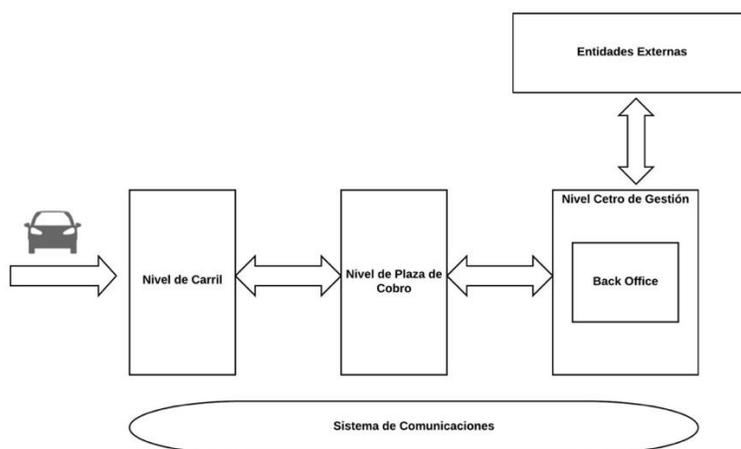
Existen distintas estrategias tecnológicas como las que he mencionado en secciones anteriores, en conclusión, tener un plan de contingencias es algo imprescindible al momento de la implementación de cualquier tipo de sistema

## Back Office

Dentro del sistema de Telepeaje existen procesos que no son automáticos y que se deben de administrar de alguna forma, es por ello existe una herramienta de software definida como Back Office que tiene como objetivo el control de las tareas implicadas como son, la gestión de clientes, generación de informes estadísticos, trazabilidad de la información, atención a clientes y la visualización de los datos en tiempo real.

El Back Office recibe como entrada de información, los datos asociados a cada cruce que realizan los automovilistas en las plazas de cobro, este software posee la funcionalidad de gestionar y organizar todos los datos de las plazas de cobro.

Como se puede observar en la siguiente ilustración, existen entidades externas que consultan los datos del Back Office para tareas administrativas, por ejemplo, las aclaraciones con los operadores interoperables de Telepeaje o atención al cliente.



*Ilustración 16. Arquitectura Back Office.*

Esta herramienta de software está diseñada en base a las necesidades de la administración del sistema de Telepeaje.

## Desarrollo BackOffice

El desarrollo del Back Office está formado por distintas herramientas de software que trabajan en conjunto para poder gestionar las tareas necesarias para la operación del sistema de Telepeaje.

Las herramientas trabajan sobre un esquema de programación conocido como modelo vista controlador (MVC) que a su vez utiliza la base de datos relacional Oracle, a continuación, describiré las ventajas de este modelo de programación.

Empezaré describiendo las características que se obtienen por trabajar con un modelo MVC, este tipo de diseño de programación, como su nombre lo indica se compone de tres principales conceptos, que permite separar el Frontend<sup>13</sup> del Backend<sup>14</sup>.

El modelo es el encargado de la lógica de la aplicación y del control de los datos. Las Vistas permiten mostrar al usuario lo que obtiene el modelo a través del controlador. Por otra parte, el Controlador es el encargado de gestionar las peticiones del usuario y procesarlas invocando al modelo.

Habría que mencionar que existen distintos lenguajes de programación que se adaptan a la arquitectura MVC, sin embargo, para este caso se utilizó principalmente el lenguaje PHP y otros como Javascript.

Una de las principales ventajas de trabajar con el modelo MVC, es que facilita el mantenimiento del software permitiendo la escalabilidad de las funcionalidades que tiene el sistema.

Por otra parte, para el diseño de la base de datos que se utilizó, es el modelo entidad relación que favorece para la abstracción de la aplicación de las reglas de negocio definidas para la implementación del sistema de Telepeaje. Sin embargo, el utilizar el modelo entidad relación tiene algunas limitaciones para el volumen de información que se maneja en el Back Office, debido a que existen millones de clientes que realizan transacciones diarias en cada momento.

Por consiguiente, una limitación importante a resolver fue el manejo y la optimización de consultas en la base de datos, debido a que una consulta mal planeada puede ocasionar algún tipo de bloqueo en la base de datos y por lo tanto afectar el rendimiento del sistema.

---

<sup>13</sup> Frontend. Es el diseño de la estructura del sitio web que utiliza el usuario final.

<sup>14</sup> Backend. Conjunto de procesos dentro de desarrollo web que se encarga de la lógica de la aplicación.

Antes de comenzar a explicar los problemas de base datos, quiero mencionar que estos temas de optimización de consultas y bloqueos son asuntos de estudio que este informe no comprende, debido a que son temas relativamente extensos y no es el objetivo de este informe. Solamente me enfocaré en la problemática real a resolver.

## Oracle Call Interface

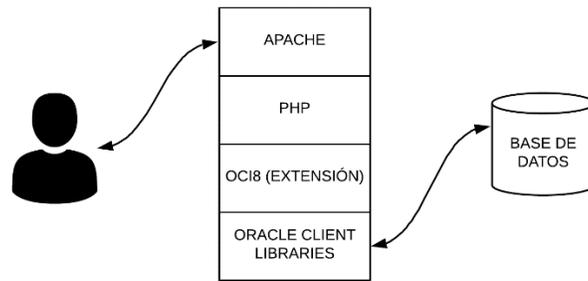
Debido a que el motor de la base de datos es Oracle y el lenguaje de programación utilizado es PHP por lo tanto se necesita de una extensión para el manejo de transacciones hacia la base de datos.

La extensión necesaria para establecer la conexión con la base de datos es conocida por sus siglas OCI (Oracle Call Interface), la cual admite sentencias SQL y PL/SQL, es decir por medio de esta librería se puede realizar el control de transacciones con la base de datos utilizando el lenguaje de programación PHP.

Los requerimientos necesarios para la utilización de esta librería son:

- Oracle Instant Client que es un conjunto de librerías que permiten que los programas se conecten con instancias de Oracle Database locales o remotas.
- Servidor APACHE
- Interprete de lenguaje de programación PHP

La siguiente ilustración es un esquema de la estructura de las librerías implicadas para el manejo de las transacciones hacia la base de datos.



*Ilustración 17. Arquitectura Librerías Oracle*

## Optimización de consultas

Es bastante común que el volumen de los datos incremente a millones de registros debido a la cantidad de clientes involucrados en un sistema tan robusto, por lo tanto, es muy probable que el volumen o cantidad de información afecte el rendimiento de la base de datos.

Por lo tanto, durante el desarrollo de software es conveniente que al utilizar el lenguaje SQL se use de la manera más óptima posible, para reducir los tiempos de espera al trabajar sobre un volumen grande de datos. Por esta razón, es importante conocer el concepto de un plan de ejecución para poder identificar y comprender los posibles factores que afectan el rendimiento de la base de datos.

El plan de ejecución es una serie de pasos ordenados que crea el optimizador de la base de datos antes de ejecutar una consulta. Para poder visualizar el plan de ejecución basta con anteponer a la sentencia SQL la siguiente instrucción.

```
EXPLAIN PLAN FOR select * from dual
```

El resultado de anteponer esta instrucción a la consulta permitirá guardar el plan de ejecución en una tabla definida como “PLAN TABLE”, por lo tanto, bastará con realizar una consulta sobre esta tabla de la siguiente manera.

```
SELECT * FROM PLAN TABLE
```

El siguiente ejemplo tiene como objetivo demostrar el funcionamiento de un plan de ejecución para comprender el costo y los recursos implicados para poder procesar una consulta SQL.

```

EXPLAIN PLAN FOR SELECT
  l.plaza_id AS PLAZA_ID,
  l.lane_id AS LANE_ID,
  l.lane_by_sense AS LANE_BY_SENSE,
  ( CASE WHEN a2.HOURS IS NULL THEN '0' ELSE a2.HOURS END ) AS HOURS,
  ( CASE WHEN a2.TRANSITS IS NULL THEN 0 ELSE a2.TRANSITS END ) AS TRANSITS
FROM
  pt_lanes l
  LEFT JOIN (
    SELECT
      lom.PLAZA_ID AS PLAZA_ID,
      lom.lane_id AS LANE_ID
    FROM
      PT_LANE_BLOCKS lom
    JOIN (
      SELECT
        lom1.PLAZA_ID,
        lom1.lane_id, max( lom1.date_open_s ) AS DATE_OPEN_S
      FROM PT_LANE_BLOCKS lom1
      GROUP BY lom1.PLAZA_ID, lom1.lane_id
    )
      laux ON lom.PLAZA_ID = laux.PLAZA_ID
      AND lom.lane_id = laux.lane_id
      AND lom.date_open_s = laux.date_open_s
    LEFT JOIN pt_transits t ON lom.PLAZA_ID = t.PLAZA_ID AND lom.lane_id = t.lane_id AND
lom.lane_block_id = t.lane_block_id
    WHERE
      t.date_s >= 1588334400
      and t.date_s <= 1588377599
      AND lom.PLAZA_ID = 29
    GROUP BY
      lom.PLAZA_ID,
      lom.lane_id
  ) a1 ON l.plaza_id = a1.plaza_id AND l.lane_id = a1.lane_id
  LEFT JOIN (
    SELECT
      lom.PLAZA_ID AS PLAZA_ID,
      lom.lane_id AS LANE_ID,
      to_char( lom.date_d, 'HH24' ) AS HOURS,
      count( lom.transit_id ) AS TRANSITS
    FROM
      pt_transits lom
    WHERE
      lom.date_s > 1588334400
      AND lom.PLAZA_ID = 29
    GROUP BY
      lom.PLAZA_ID,
      lom.lane_id,
      to_char( lom.date_d, 'HH24' )
  ) a2 ON l.PLAZA_ID = a2.PLAZA_ID AND l.lane_id = a2.lane_id
WHERE
  l.lane_showable_flag = 1
  AND l.lane_id != 0
  AND l.PLAZA_ID = 29
ORDER BY
  l.PLAZA_ID,
  l.LANE_ID,
  a2.HOURS

```

OPERATION	OBJECT	COST	CARDINALITY
SELECT STATEMENT ( )		5534	1936
SORT (ORDER BY)		5534	1936
HASH JOIN (OUTER)		5533	1936
JOIN FILTER (CREATE)	:BF0000	1877	7
HASH JOIN (OUTER)		1877	7
TABLE ACCESS (BY INDEX ROWID BATCHED)	PT_LANES	4	7
INDEX (RANGE SCAN)	IDX_PT_LANES_1	1	7
VIEW ( )		1873	1
HASH (GROUP BY)		1873	1
NESTED LOOPS (SEMI)		1873	1
NESTED LOOPS ( )		1869	1
VIEW ( )	VW_GBF_18	1833	6
HASH (GROUP BY)		1833	6
VIEW ( )		1832	8
HASH (GROUP BY)		1832	8
TABLE ACCESS (BY INDEX ROWID BATCHED)	PT_LANE_BLOCKS	1830	5699
INDEX (RANGE SCAN)	PT_LANE_OM_1	25	5699
TABLE ACCESS (BY INDEX ROWID BATCHED)	PT_LANE_BLOCKS	6	1
INDEX (RANGE SCAN)	IDX_PT_LANE_OPERATIONAL_MODE_2	2	28
TABLE ACCESS (BY INDEX ROWID BATCHED)	PT_TRANSITS	4	3
INDEX (RANGE SCAN)	IDX_PT_TRANSITS_10	3	1
VIEW ( )		3657	5766
HASH (GROUP BY)		3657	5766
JOIN FILTER (USE)	:BF0000	3656	5766
TABLE ACCESS (BY INDEX ROWID BATCHED)	PT_TRANSITS	3656	5766
INDEX (SKIP SCAN)	PT_TRANSITS_CONDOR_1	10	5766

Tabla 3. Plan de Ejecución.

Para poder analizar el resultado del plan de ejecución de la consulta previa, es necesario conocer algunos conceptos previos como los métodos de acceso a los datos, el costo implicado para el procesamiento de la consulta y el orden de ejecución de cada operación.

Como se puede observar en la tabla anterior, en la primera columna se especifica la operación realizada también conocido como método de acceso, que tiene como finalidad obtener los datos requeridos, sin embargo, el orden de cada instrucción radica en la indentación, es decir, las operaciones que se ejecutan al inicio son las que poseen una indentación mayor, respecto al resto de operaciones.

Existen distintos métodos de acceso que están implementados en una base de datos Oracle, los cuales describiré a continuación. Cabe mencionar que esta sección tiene como objetivo mencionar los conceptos básicos que se aplican al momento de analizar la optimización de una consulta, por otra parte, quiero mencionar que la optimización de consultas y los planes de ejecución son un tema de estudio que necesita de un análisis más profundo.

La interpretación del plan de ejecución comienza con el método de acceso:

**Table Access By Index:** Recupera una fila de la tabla usando el ROWID, es decir hace uso del índice definido para el campo requerido en el predicado de la consulta para poder realizar el primer filtro de la consulta.

**Index Range Scan:** Realiza el recorrido del B-tree y y sigue la cadena de los nodos hoja para encontrar todos los registros que coinciden.

**Nested Loop:** Se trata de unir dos tablas yendo a buscar el resultado desde una tabla y seleccionando la otra tabla por cada fila desde la primera.

**Sort Group By:** Ordena el resultado configurado sobre las columnas del group by y combina el resultado ordenado en la segunda etapa. Esta operación necesita una cantidad importante de memoria para materializar el conjunto del resultado intermedio (sin pipeline)

Por otro lado, el costo es una unidad de tiempo que se determina a partir de la siguiente formula:

$$Costo = \frac{(\#Lsim \times TLsim) + (\#LMul \times TLMul) + \left(\frac{\#CProc}{VProc}\right)}{TLsim}$$

Donde:

- #Lsim: Número de lecturas de bloque.
- TLsim: Tiempo de lectura de bloque simple.
- #LMul: Número de lecturas de múltiples bloques.
- TLMul: Tiempo de lectura de múltiples bloques.
- #CProc: Número de ciclos del procesador.
- VProc: Velocidad de procesador.

Cabe mencionar que un bloque es la mínima unidad de lectura en una base de datos Oracle, es decir, Oracle no lee y escribe en bloques del sistema operativo, sino que lo hace en unidades lógicas que son los bloques de datos y que varían de una base de datos a otra en la misma máquina ya que es un valor que se debe indicar en la creación de cada base de datos Oracle.

Por lo tanto, el costo deberá tender a ser lo más optimo posible, mientras más alto sea este, más lenta será la consulta.

La siguiente tabla es un extracto del resultado de la consulta anterior

PLAZA	LANE	SENSE	HOURS	TRANSITS
29	2106	S	0	98
29	2106	S	1	98
29	2106	S	2	99
29	2106	S	3	84
29	2106	S	4	113
29	2106	S	5	96
29	2106	S	6	182
29	2106	S	7	193
29	2106	S	8	162

29	2106	S	9	142
29	2106	S	10	117
29	2106	S	11	133
29	2106	S	12	140
29	2106	S	13	123
29	2106	S	14	119
29	2106	S	15	124
29	2106	S	16	120
29	2106	S	17	145
29	2106	S	18	131
29	2106	S	19	112
29	2106	S	20	108
29	2106	S	21	108
29	2106	S	22	111
29	2106	S	23	87

La ilustración 16. Explica el comportamiento del aforo de vehículos por cada hora, para un único carril de una plaza de cobro, el punto más alto de aforo se observa a las 7 de la mañana con un total de 193 vehículos.



*Ilustración 18. Resultado de Aforo*

Por otra parte, además de analizar los distintos métodos de acceso al momento de la ejecución de la consulta. Existen otros principios que en la práctica ayudan a mejorar los tiempos de ejecución, este concepto es también conocido como la optimización heurística, la cual consiste en cambiar la consulta original por una equivalente con el objetivo de minimizar los resultados intermedios.

Teóricamente consiste en aplicar equivalencias de los operadores algebraicos de forma que las selecciones y proyecciones se apliquen lo antes posible con el objetivo de excluir datos que son innecesarios. Prácticamente este concepto radica en la aplicación de las reglas de la algebra lineal, donde haciendo usos de dichas reglas se puede llegar a la expresión mínima de una consulta de base de datos.

## Índices

Ya que se ha mencionado anteriormente el concepto de optimización de consultas existe una estructura definida como índice que trabaja de manera inherente al momento de ejecutar alguna consulta sobre la base de datos.

El objetivo principal de un índice es poder agilizar el acceso a los datos por medio de una representación ordenada de los datos de una tabla, esta representación es un orden lógico independientemente del orden físico donde estén almacenados los datos. Estrictamente un índice es una lista doblemente enlazada para poder conectar los nodos hojas, donde cada nodo esta almacenado dentro de un bloque de la base de datos. Además de la lista enlaza para que el concepto de índice pueda funcionar existe otra estructura de datos definida como un árbol de búsqueda. Prácticamente una búsqueda por un índice requiere tres etapas:

1. El recorrido del árbol.
2. Seguir la cadena de los nodos hojas.
3. Devolver los datos de la tabla.

## Bloqueos

El siguiente punto trata de un problema prácticamente común en los sistemas de bases de datos, debido un hecho originado por la concurrencia de transacciones.

Dicho de otra manera, me refiero a la ejecución de distintas operaciones en un instante de tiempo sobre un mismo recurso, provocando que el gestor de la base de datos deba de restringir, limitar o priorizar el acceso a de los recursos que se están utilizando.

Existen tres niveles de bloqueos:

- Bloqueos de tablas
- Bloqueos a nivel de fila
- Bloqueos a nivel de usuario

No obstante, para este informe me enfocaré en los bloqueos a nivel de tablas, debido a que es un tema común para el desarrollo del sistema.

Este tipo de bloqueos es originado por las operaciones DELETE, UPDATE, INSERT. Cabe mencionar que el principal propósito de los bloqueos es prevenir los conflictos que estas operaciones puedan provocar. Explicaré los distintos tipos de bloqueos.

- **X** (Exclusivo): Consiste en bloquear toda la tabla, impidiendo la ejecución de sentencias DML<sup>15</sup> sobre ella.
- **S** (Compartido): Se marca la tabla como “Sólo Lectura”. Varias transacciones pueden adquirir este bloqueo sobre una misma tabla.
- **RS** (Filas Compartidas): Este bloqueo se utiliza para indicar que se va a realizar en un futuro cercano alguna modificación a las filas de esta tabla.
- **RX** (Filas Exclusivas): Con este bloqueo se indica que ya se realizaron modificaciones INSERT, UPDATE o DELETE a las filas de esta tabla.
- **SRX** (Filas Compartidas Exclusivas): Sólo una transacción puede obtener este bloqueo, ya que permite prohibirles a todas las demás transacciones la modificación de la fila, pero se permite la lectura

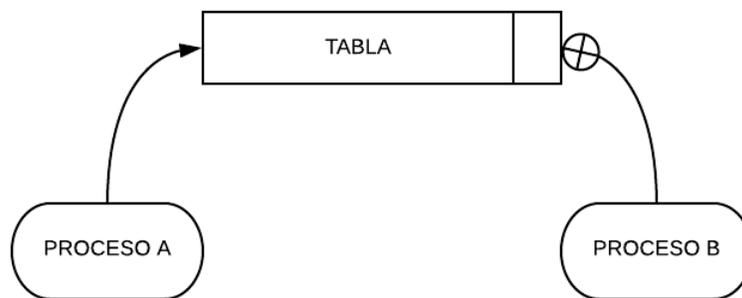
---

<sup>15</sup> DML. Por sus siglas en inglés Data Manipulation Language, permiten introducir y manipular los datos por medio de consultas, modificaciones o eliminaciones.

Por otra parte, a nivel de usuario final, la concurrencia que genera algún tipo de bloqueo cuando varios procesos implican los mismos datos, provoca que la sesión del usuario no pueda realizar la ejecución de las tareas que comúnmente se puede hacer en el sistema.

Este problema de sesiones es generado debido a que un primer proceso que provoco un bloqueo de tipo exclusivo, es decir de toda la tabla no se ha liberado hasta que se termine de realizar su cometido. Y al mismo tiempo, existe otra sesión, esperando a que termine el primer proceso para poder ejecutarse.

Esto generará una espera a nivel de usuario final para los procesos que se quieran ejecutar con los mismos recursos, es decir, desde otro punto de vista, lo que el usuario verá es que la aplicación demora más tiempo de lo estimado o en el peor de los casos no realiza ninguna funcionalidad.



*Ilustración 19. Bloqueo Exclusivo.*

Así que es conveniente identificar las sesiones que generan bloques exclusivos. La siguiente consulta, permite mostrar las sesiones que han generado un bloque sobre la base de datos.

```

SELECT SUBSTR(a.object,1,25) TABLENAME,
SUBSTR(s.username,1,15) USERNAME,
SUBSTR(p.pid,1,5) PID,
SUBSTR(p.spid,1,10) SYSTEM_ID,
DECODE(l.type,
  'RT','Redo Log Buffer',
  'TD','Dictionary',
  'TM','DML',
  'TS','Temp Segments',
  'TX','Transaction',
  'UL','User',
  'RW','Row Wait',
  l.type) LOCK_TYPE
FROM gv$access a, gv$process p, gv$session s, gv$lock l
WHERE s.sid = a.sid
AND s.paddr = p.addr
AND l.sid = p.pid
GROUP BY a.object, s.username, p.pid, l.type, p.spid
ORDER BY a.object, s.username;

```

Una solución para poder liberar el bloqueo es identificar la sesión que lo genera y posteriormente se debe de eliminar, así que, con la siguiente consulta se puede observar.

```

SELECT a.USERNAME, a.SID, a.SERIAL#, b.SPID
FROM v$session a, v$process b
WHERE a.PADDR= b.ADDR

```

El resultado de la consulta anterior devolverá el nombre del usuario, el número serial del proceso Oracle y el proceso Oracle en el sistema operativo.

USERNAME	SID	SERIAL#	SPID
USER 1	108	3916	56061
USER 2	155	75	55623
USER 3	177	4678	62199

Para poder eliminar la sesión que genera el bloqueo se puede hacer con la siguiente instrucción. Donde se debe de agregar el SID de la sesión a eliminar.

```

ALTER SYSTEM KILL SESSION 'SID, SERIAL#';

```

## Antifraude

Uno de los factores que se deben de considerar en el sistema de Telepeaje es la posibilidad que tiene los automovilistas para efectuar algún tipo de estafa con los dispositivos TAG, debido a que en la realidad existen algunos casos donde los automovilistas buscarán la oportunidad de evitar realizar el pago por cualquier método.

Para poder solucionar esta problemática el principal elemento que debe de estimar es el tiempo de resolución debido a que la demora de tiempo en encontrar el fraude provocará que se refleje en pérdidas económicas, por ello debe de existir un proceso que se encargue de analizar cada posible intento de fraude e identificarlo de manera automática.

Es por esto por lo que esta tarea está desarrollada a nivel de base de datos por medio de distintos triggers <sup>16</sup> que se encargan de validar la veracidad de los cruces y por lo tanto aplicar las reglas antifraude.

El uso de los triggers para la solución de esta problemática se adapta de manera muy eficaz debido a que una de las características de este tipo de operaciones es que se pueden realizar inmediatamente al momento de un registro de una transacción posiblemente fraudulenta, en otras palabras, como los triggers se ejecutan de manera automática cuando ocurre algún evento sobre la base de datos como la inserción, modificación o eliminación por lo tanto cuando se registra un nuevo evento se puede obtener las últimas transacciones de un TAG en específico y en base a las reglas de negocio o reglas antifraude poder determinar si el automovilista está realizando algún tipo de estafa.

Sirva de ejemplo el siguiente código donde se aplica una regla de antifraude que consiste en dar de baja temporal el TAG cuando se encuentran diez cruces en seis horas para una sola plaza de cobro.

La comprobación del posible fraude se realiza con el siguiente trigger el cual se encarga de verificar el número de cruces a partir de la fecha de la última transacción que se acaba de registrar en la base de datos, posteriormente si se cumple las restricciones definidas se procede a actualizar los campos correspondientes en la base de datos para el TAG para definido como fraudulento, por lo tanto, para la generación de la lista blanca este cliente será marcado como inhabilitado.

---

<sup>16</sup> Trigger o disparado es un objeto dentro de la base de datos que tiene como finalidad ejecutar alguna operación de manera automática cuando se cumple una condición previamente establecida.

```

DECLARE
    CONT_Y NUMBER;
    CVA NUMBER;      -- Clase vehicular actual
    IUT VARCHAR2(29); -- Identificador Único de Transacción actual
    SAC NUMBER;      -- Sentido actual
    EXIST_TAG NUMBER; -- Valor para comprobar si existe el TAG en Inter_TAG
    VAL_ST NUMBER;   -- Valor de status en INTER_TAG
    TAGS_VIP NUMBER; -- Valor de tag VIPS, si está el TAG no aplica regla
    OPERADOR NUMBER; -- Valor para obtener el OTI del TAG
    CONT_OPERS NUMBER; -- Valor para obtener cuántos OTI'a tiene el TAG

    IUT_EXIST NUMBER; -- Para almacenar count a inter_invalid_tag para
determinar si existe ya un registro
    IUT_AUX VARCHAR2(29);

BEGIN

    -- Obtiene cuántos OTI's tiene el TAG
    SELECT COUNT(ID_OPERADOR) INTO CONT_OPERS FROM INTER_TAG WHERE
REPLACE(NUMTAG, '.', '') = :NEW.TAG_BAND;

    -- Verifica si hay una longitud mayor a 11 y si empieza con 009 ara
recortar el TAG
    IF((SUBSTR(:NEW.TAG_BAND, 1, 3) = '009') AND (LENGTH(:NEW.TAG_BAND) =
19))THEN
        :NEW.TAG_BAND := substr(:NEW.TAG_BAND, 1, 11);
    END IF;

    -- Tabla de VIP's
    SELECT
        COUNT(*)
    INTO
        TAGS_VIP
    FROM INTER_TAG_VIPS
    WHERE
        INTER_TAG_VIPS.NUMTAG LIKE '%'||:NEW.TAG_BAND||'%'
        AND SUBSTR(INTER_TAG_VIPS.MASCARA_PLAZA, :NEW.PLAZA_ID, 1) = '1'
        AND (sysdate - to_date('01-01-1970', 'DD-MM-YYYY')) * (86400) >=
FECHA_INICIO_S
        AND (
            (sysdate - to_date('01-01-1970', 'DD-MM-YYYY')) *
(86400) <= FECHA_FIN_S
            OR FECHA_FIN_S IS NULL);

    IF NOT(:NEW.PLAZA_ID = 45 AND (:NEW.lane_id = 2108 OR :NEW.lane_id =
2309 ) ) THEN

        IF (TAGS_VIP = 1) THEN
            log_antifraude('ANTIFRAUD_RULE_05', :NEW.TAG_BAND, 'Tag EXISTENTE en
tabla VIPS');
        END IF;

```

```

        IF (:NEW.TAG_BAND IS NOT NULL AND :NEW.PAYMENT_FLAG_VOIDED = 0 AND
:NEW.PAyment_method_id = 5 AND TAGS_VIP = 0) THEN

        -- Obtiene las últimas transacciones del TAG en las últimas 6 horas en
esa plaza de cobro...
        SELECT
            COUNT(*)
        INTO
            CONT_Y
        FROM PT_TRANSITS T
        WHERE
            T.PLAZA_ID = :NEW.PLAZA_ID
            AND T.aux_field_y = :NEW.TAG_BAND
            AND T.DATE_D > :NEW.DATE_D - (6/24)
            AND T.DATE_D < :NEW.DATE_D
            AND T.PAYMENT_FLAG_VOIDED = 0
            AND T.aux_field_y IS NOT NULL
            AND NOT (T.PLAZA_ID = 45 AND (T.lane_id = 2108 OR T.lane_id =
2309 ))
            AND T.PAYMENT_METHOD_ID = 5;

        IF (CONT_Y > 9) THEN
            --VALIDACION PARA QUE NO SE DUPLIQUEN REGISTROS
            SELECT
                T.TRANSIT_ID_UNIQ INTO IUT_AUX
            FROM PT_TRANSITS T JOIN PT_LANES L ON T.PLAZA_ID =
L.PLAZA_ID AND T.LANE_ID = L.LANE_ID WHERE T.PLAZA_ID = :NEW.PLAZA_ID AND
T.LANE_ID = :NEW.LANE_ID AND T.TRANSIT_ID = :NEW.TRANSIT_ID;

            SELECT COUNT(*) INTO IUT_EXIST FROM INTER_INVALID_TAG
            WHERE NUM_UNIC_TRANS = IUT_AUX AND DESCRIPCION = 'Tag con
10 transacciones en un tiempo de 6h aprox.';

            IF (IUT_EXIST = 0) THEN
                -- Inserta en la tabla INTER_INVALID_TAG para
guardar el registro de fraude
                INSERT INTO
                "CCONTROL"."INTER_INVALID_TAG"
                (
                    "ID_PLAZA",
                    "ID_CARRIL",
                    "FECHA",
                    "NUM_TAG",
                    "DIR_CARRIL",
                    "CLASE_V",
                    "NUM_UNIC_TRANS",
                    "DESCRIPCION"
                )
                (
                    SELECT
                    T.PLAZA_ID,
                    T.LANE_ID,
                    T.DATE_D,
                    T.aux_field_y,

```

```

L.LANE_DIRECTION,
T.CLASS_ID_PAYMENT,
T.TRANSIT_ID_UNIQ,
'tag con 10 transacciones en un
tiempo de 6h aprox.'

FROM PT_TRANSITS T JOIN PT_LANES L ON
T.PLAZA_ID = L.PLAZA_ID AND T.LANE_ID = L.LANE_ID WHERE T.PLAZA_ID =
:NEW.PLAZA_ID AND T.LANE_ID = :NEW.LANE_ID AND T.TRANSIT_ID = :NEW.TRANSIT_ID
);

SELECT COUNT(*) INTO EXIST_TAG FROM INTER_TAG
WHERE REPLACE(NUMTAG, '.', '') = :NEW.TAG_BAND AND ID_OPERADOR =
:NEW.tag_issuer_code_id;

IF(EXIST_TAG > 0) THEN
SELECT STATUS INTO VAL_ST FROM
INTER_TAG WHERE REPLACE(NUMTAG, '.', '') = :NEW.TAG_BAND AND ID_OPERADOR =
:NEW.tag_issuer_code_id;

IF(VAL_ST = 1) THEN

-- Actualiza el estado del TAG, inhabilitándolo en la tabla INTER_TAG
(Centor de control)
UPDATE "CONTROL"."INTER_TAG" SET
"STATUS" = '0', "COMISION_ADQ" = '1' WHERE REPLACE(NUMTAG, '.', '') =
:NEW.TAG_BAND AND ID_OPERADOR = :NEW.tag_issuer_code_id ;
UPDATE PT_TRANSITS SET
PAYMENT_METHOD_ID_9 = 5 WHERE TRANSIT_ID_UNIQ = (SELECT T.TRANSIT_ID_UNIQ
FROM PT_TRANSITS T WHERE T.PLAZA_ID = :NEW.PLAZA_ID AND T.LANE_ID =
:NEW.LANE_ID AND T.TRANSIT_ID = :NEW.TRANSIT_ID);

IF(CONT_OPERS = 1) THEN

-- Obtiene el OTI del TAG
SELECT ID_OPERADOR INTO
OPERADOR FROM INTER_TAG WHERE REPLACE(NUMTAG, '.', '') = :NEW.TAG_BAND;

IF(OPERADOR = 1) THEN
--Actualiza
estado en PASE (Esquema PASE)

log_antifraude('ANTIFRAUD_RULE_05', :NEW.TAG_BAND, 'Invalidando TAG por
antifraude en CENTRAL y esquema PASE');

UPDATE INTER_TAG@DB_LINK_PASE SET "STATUS" = '0', "COMISION_ADQ" = '1' WHERE
REPLACE(NUMTAG, '.', '') = :NEW.TAG_BAND;

ELSE

```



## Zona horaria

Debido a que el sistema de cobro funciona de manera unificada en todo el país pueden existir diferencias al momento de realizar consultas, generar informes o analizar información a partir de cierta fecha y hora ya que ciertos estados se encuentran en una zona horaria diferente, es por esto que surge una problemática que necesitar ser tratada para homologar el uso de las zonas horarias.



Así que para poder tratar este tipo de datos se utilizó el concepto de *Timestamp*<sup>17</sup> ó también conocido como *Tiempo Unix* el cual es un sistema para la implementación del tiempo, el valor del timestamp es una definición que representa una fecha en un valor numérico expresado en segundos a partir del primero de enero de 1970 para sistemas UNIX en formato UTC, en otras palabras, a medida que transcurre el tiempo en cada segundo el valor de timestamp también aumenta de manera directa.

Por lo tanto, una fecha expresada en este tipo de sistema se puede interpretar de distintas formas dependiendo de las necesidades de la zona horaria, así que una vez que se conoce el este valor numérico puede ser tratado con distintos lenguajes de programación dependiendo de las necesidades requeridas.

---

<sup>17</sup> Timestamp es un sistema que se define como la cantidad de segundos transcurridos desde la media noche del 1de enero de 1970 en formato UTC el cual es un estándar del tiempo internacional.

## Conclusiones

Finalmente, al terminar el presente documento en el cual describo los retos y procesos implicados para el desarrollo del Telepeaje puedo concluir en dos principales puntos:

Los resultados obtenidos del desarrollo del sistema y el impacto personal al haber participado en un proyecto profesional.

Así que comenzaré con la perspectiva de los resultados del proyecto, en definitiva, el reto implicado de la mejora del sistema de cobro electrónico de peaje, se ha beneficiado con la optimización de tiempos para poder replicar los datos de cada cliente en los distintos tramos carreteros, garantizando la disponibilidad y actualización de la información de cada automovilista.

Es por esto que cada persona que hace uso de los carriles exclusivos de Telepeaje en el país obtiene la garantía que, al utilizar el sistema de cobro electrónico, podrá hacerlo con la confianza de que el depósito de su dinero está asociado y listo para usarse en su dispositivo TAG, independientemente de la empresa con la que ha contratado el dispositivo este servicio, es decir el concepto de interoperabilidad se vuelve completamente transparente.

Además, no solo cada automovilista se beneficia con la arquitectura del sistema, también las tareas administrativas, como la gestión de cuentas, atención al cliente, monitoreo de hardware, trazabilidad de transacciones, análisis de información y generación de estadísticas se convierten en tareas eficientes debido a la replicación de datos en tiempo real, este es el principal factor para poder crear una alta disponibilidad de los datos.

Por otra parte, como aprendizaje individual, comprendí que el curso de un proyecto no radica únicamente en temas técnicos, simultáneamente existen factores implicados en el éxito de la aspiración de un objetivo en común. Por ejemplo, conocer las limitaciones y el alcance de los recursos disponibles es esencial para la obtención de resultados, es decir saber que los recursos son finitos exige a realizar una mejor administración de estos para poder generar expectativas de los resultados a cumplir.

Además de estas particularidades mencionadas, existen otras áreas de conocimiento que no se relacionan directamente con aspectos de ingeniería, que se deben de dominar para tener una visión completa de los retos implicados, sirva de ejemplo, conocer la lógica de procesos administrativos, como la estimación costos y la definición de los tiempos necesarios para generar resultados.

A su vez, otro factor que es imprescindible en un proyecto es la comunicación que se debe de realizar con todas las personas involucradas, porque la falta de precisión durante algún proceso o incluso una falla, puede ser originada debido a la ausencia de la claridad los objetivos, lo que quiero decir es, que se debe ser claro al expresar las ideas y necesidades.

Es por esto por lo que el desarrollo como profesional va más allá de cuestiones técnicas o del dominio de un tema en específico, se deben de construir una mejora constantemente en habilidades como la administración, organización y comunicación a nivel personal y grupal.

Finalmente, me gustaría agregar un comentario desde el punto de vista como egresado de la facultad de ingeniería. Debido a que como universitario surge el dilema si en verdad lo que aprendes en la facultad te servirá en un ambiente profesional, en mi caso considero que los conocimientos adquiridos funcionaron como punto de partida para conocer temas más complejos de ingeniería y poder involucrarme en actividades que presentan riesgos reales.

Probablemente existen casos de otros universitarios en donde no aplica esta conclusión, porque tal vez se enfrentan a situaciones completamente diferentes, sin embargo, las asignaturas que se imparten en un plan de estudios, considero que cumplen con un objetivo únicamente académico, que es la evaluación de los conocimientos suficientes. A pesar de esto, quiero mencionar que un modelo de evaluación, en base a una calificación no funciona como un indicador del desempeño de cada persona.

No obstante, considero que otro factor que influye en aplicar los conocimientos adquiridos como universitario, es la ambición de cada persona. Con esto quiero decir, que la responsabilidad que tiene cada universitario de poder desarrollarse como profesional no depende plenamente de un plan de estudios o de una calificación, depende del nivel del ahínco de cada persona.

Definitivamente el progreso se compone de distintos aspectos y paradigmas que se deben de ir descubriendo y resolviendo, en cada reto al que se enfrenta una persona.

## Anexos

Tabla de las plazas de cobro.

	<b>Nombre</b>	<b>Tramo Carretero</b>	<b>Delegación</b>
1	Tlalpan	México - Cuernavaca	IV CUERNAVACA
2	Iguala	Puente de Ixtla - Iguala	IV CUERNAVACA
3	Tepotztlán	México - Querétaro	III QUERÉTARO
4	Palmillas	México - Querétaro	III QUERÉTARO
5	Querétaro	Querétaro - Irapuato	III QUERÉTARO
6	San Marcos	México - Puebla	V PUEBLA
7	San Martin	México - Puebla	V PUEBLA
8	Oacalco	La Pera - Cuautla	IV CUERNAVACA
9	Tepoztlán	La Pera - Cuautla	IV CUERNAVACA
10	Amozoc	Puebla - Acatzingo	V PUEBLA
11	Esperanza	Acatzingo - CD Mendoza	V PUEBLA
12	Playas de Tijuana	Tijuana - Ensenada	I TIJUANA
13	Rosarito	Tijuana - Ensenada	I TIJUANA
14	Ensenada	Tijuana - Ensenada	I TIJUANA
15	Salamanca	Querétaro - Irapuato	III QUERÉTARO
16	Fortín	Ciudad Mendoza - Córdoba	VII VERACRUZ
17	Zacapalco	Zacapalco - Rancho viejo	IV CUERNAVACA
18	Lib. Sur Poniente Querétaro	Querétaro - Irapuato	III QUERÉTARO
19	Jorobas	México - Querétaro	III QUERÉTARO
20	Polotitlan	México - Querétaro	III QUERÉTARO
21	Chalco	México - Puebla	V PUEBLA
22	Pte. Reynosa Pharr	Puente Reynosa - PHARR	VIII REYNOSA
23	Tehuacán	Cuacnopalan - Tehuacán Oaxaca	OAXACA
24	Miahuatlan	Cuacnopalan - Tehuacán Oaxaca	OAXACA
25	Suchixtlahuaca	Cuacnopalan - Tehuacán Oaxaca	OAXACA
26	Huitzo	Cuacnopalan - Tehuacán Oaxaca	OAXACA
27	El Hongo	Tecate - La Rumorosa	I TIJUANA
28	Tehuantepec	Salina Cruz - La ventosa	VI COATZACOALCOS
29	Ixtepec	Salina Cruz - La ventosa	VI COATZACOALCOS
30	Durango	Durango - Mazatlán	X MONTERREY

31	Llano Grande	Durango - Mazatlán	X MONTERREY
32	Coscomate	Durango - Mazatlán	X MONTERREY
33	Mesillas	Durango - Mazatlán	X MONTERREY
34	Alpuyeca Balnearios	Cuernavaca - Acapulco	IV CUERNAVACA
35	Paso Morelos	Cuernavaca - Acapulco	IV CUERNAVACA
36	Palo Blanco	Cuernavaca - Acapulco	IV CUERNAVACA
37	La Venta	Cuernavaca - Acapulco	IV CUERNAVACA
38	XOCHITEPEC	Cuernavaca - Acapulco	IV CUERNAVACA
39	Aeropuerto	Cuernavaca - Acapulco	IV CUERNAVACA
40	Central de Abastos	Cuernavaca - Acapulco	IV CUERNAVACA
41	Sabinas	Monterrey - Nuevo Laredo	X MONTERREY
42	Libramiento Tampico	Libramiento Poniente de Tampico	VII VERACRUZ
43	Cuitláhuac	Córdoba - Veracruz	VII VERACRUZ
44	Paso del Toro	Córdoba - Veracruz	VII VERACRUZ
45	Cosamaloapan	La Tinaja - Cosoleacaque	VI COATZACOALCOS
46	Acayucan	La Tinaja - Cosoleacaque	VI COATZACOALCOS
47	Chichimequillas	Libramiento Noreste de Querétaro	III QUERÉTARO
48	General Bravo	Cadereyta - Reynosa	X MONTERREY
49	Seyba Playa	Champotón - Campeche	VI COATZACOALCOS
50	Chamapa	Chamapa - Lechería	III QUERÉTARO
51	Nopala	Chamapa - Lechería	III QUERÉTARO
52	Cipreses	Chamapa - Lechería	III QUERÉTARO
53	Madin	Chamapa - Lechería	III QUERÉTARO
54	Atizapán	Chamapa - Lechería	III QUERÉTARO
55	Lago de Guadalupe	Chamapa - Lechería	III QUERÉTARO
56	Cuautitlán	Chamapa - Lechería	III QUERÉTARO
57	La Carbonera	Carbonera - Ojo caliente	X MONTERREY
58	Los Chorros	La Carbonera - Puerto México	X MONTERREY
59	Huachichil	La Carbonera - Puerto México	X MONTERREY
60	Estación Don	Estación Don - Nogales	II GERENCIA PACIFICO
61	Fundición	Estación Don - Nogales	II GERENCIA PACIFICO
62	Esperanza/2 (Sonora)	Estación Don - Nogales	II GERENCIA PACIFICO

63	Guaymas	Estación Don - Nogales	II GERENCIA PACIFICO
64	Hermosillo	Estación Don - Nogales	II GERENCIA PACIFICO
65	Magdalena	Estación Don - Nogales	II GERENCIA PACIFICO
66	Bermejillo	Gómez Palacios - Corralitos	X MONTERREY
67	Ceballos	Gómez Palacios - Corralitos	X MONTERREY
68	Plan de Ayala	Torreón - Saltillo	X MONTERREY
69	La Cuchilla	Torreón - Saltillo	X MONTERREY
70	Pte. Zacatal	Puente El Zacatal	VI COATZACOALCOS
71	Nuevo Progreso	Reynosa - Matamoros	VIII REYNOSA
72	Pte. Ignacio Zaragoza	Puente Ignacio Zaragoza	VIII REYNOSA
73	Sánchez Magallanes	Cárdenas - Entronque Agua dulce	VI COATZACOALCOS
74	Totomoxtle	Gutiérrez - Zamora Tihutlán	VII VERACRUZ
75	Las Choapas	Las Choapas - Ocozocoautla	GERENCIA CHIAPAS
76	Malpasito	Las Choapas - Ocozocoautla	GERENCIA CHIAPAS
77	Ocozocuatlá	Las Choapas - Ocozocoautla	GERENCIA CHIAPAS
78	Villagran	Querétaro - Irapuato	III QUERÉTARO
79	Ing. Fco. Velasco D	Cuernavaca - Acapulco	IV CUERNAVACA
80	Lomas Verdes	Chamapa - Lechería	III QUERÉTARO
81	Cerro Gordo	Querétaro - Irapuato	III QUERÉTARO
82	San Marcos Bis.	México - Puebla	V PUEBLA
83	Nodo Popotla	Tijuana - Ensenada	I TIJUANA
84	Lagos de Moreno	Lagos de Moreno - Villa de Arriaga	III QUERÉTARO
85	Ixtapaluca	México - Puebla	V PUEBLA
86	Ozumba	Libramiento Amecameca - Cuautla	IV CUERNAVACA
87	San Fernando	Libramiento Sur de Reynosa	VIII REYNOSA
88	Palmillas Bis	México - Querétaro	III QUERÉTARO
89	Lucio Blanco	Libramiento Sur de Reynosa	VIII REYNOSA
90	Tres Marías	México - Cuernavaca	IV CUERNAVACA
91	Taxco	Rancho Viejo - Taxco	IV CUERNAVACA
95	Ramones	Cadereyta - Reynosa	X MONTERREY
96	Aeropuerto-San Lucas	Libramiento Sur de Reynosa	VIII REYNOSA

97	Plaza Cabo San Lucas	Cadereyta - Reynosa	X MONTERREY
98	Plaza Mangle	Cadereyta - Reynosa	X MONTERREY
99	Plaza San José del Cabo	Cadereyta - Reynosa	X MONTERREY
110	Agualeguas	Monterrey - Nuevo Laredo	X MONTERREY
111	Paras	Monterrey - Nuevo Laredo	X MONTERREY
112	Herrerias	Cadereyta - Reynosa	X MONTERREY
113	DR. Cross	Cadereyta - Reynosa	X MONTERREY

Resultado de la consulta optimización de consultas.

PLAZA	LANE	SENSE	HOURS	TRANSITS
29	2105	S	0	0
29	2106	S	0	98
29	2106	S	1	98
29	2106	S	2	99
29	2106	S	3	84
29	2106	S	4	113
29	2106	S	5	96
29	2106	S	6	182
29	2106	S	7	193
29	2106	S	8	162
29	2106	S	9	142
29	2106	S	10	117
29	2106	S	11	133
29	2106	S	12	140
29	2106	S	13	123
29	2106	S	14	119
29	2106	S	15	124
29	2106	S	16	120
29	2106	S	17	145
29	2106	S	18	131
29	2106	S	19	112
29	2106	S	20	108
29	2106	S	21	108
29	2106	S	22	111
29	2106	S	23	87

29	2107	S	0	12
29	2107	S	1	6
29	2107	S	2	5
29	2107	S	3	5
29	2107	S	4	16
29	2107	S	5	20
29	2107	S	6	32
29	2107	S	7	37
29	2107	S	8	49
29	2107	S	9	50
29	2107	S	10	30
29	2107	S	11	40
29	2107	S	12	70
29	2107	S	13	99
29	2107	S	14	70
29	2107	S	15	61
29	2107	S	16	49
29	2107	S	17	74
29	2107	S	18	49
29	2107	S	19	26
29	2107	S	20	34
29	2107	S	21	38
29	2107	S	22	23
29	2107	S	23	10
29	2301	N	0	5
29	2301	N	1	2
29	2301	N	2	4
29	2301	N	4	2
29	2301	N	5	9
29	2301	N	6	10
29	2301	N	7	27
29	2301	N	8	19
29	2301	N	9	15
29	2301	N	10	29
29	2301	N	11	30
29	2301	N	12	36
29	2301	N	13	48
29	2301	N	14	50

29	2301	N	15	54
29	2301	N	16	43
29	2301	N	17	39
29	2301	N	18	38
29	2301	N	19	17
29	2301	N	20	21
29	2301	N	21	30
29	2301	N	22	19
29	2301	N	23	6
29	2302	N	0	7
29	2302	N	1	2
29	2302	N	2	2
29	2302	N	3	8
29	2302	N	4	36
29	2302	N	5	37
29	2302	N	6	81
29	2302	N	7	119
29	2302	N	8	129
29	2302	N	9	95
29	2302	N	10	62
29	2302	N	11	64
29	2302	N	12	53
29	2302	N	13	52
29	2302	N	14	60
29	2302	N	15	55
29	2302	N	16	62
29	2302	N	17	34
29	2302	N	18	23
29	2302	N	19	16
29	2302	N	20	14
29	2302	N	21	12
29	2302	N	22	6
29	2302	N	23	5
29	2303	N	0	83
29	2303	N	1	76
29	2303	N	2	64
29	2303	N	3	51
29	2303	N	4	64

29	2303	N	5	52
29	2303	N	6	83
29	2303	N	7	93
29	2303	N	8	87
29	2303	N	9	132
29	2303	N	10	199
29	2303	N	11	144
29	2303	N	12	125
29	2303	N	13	136
29	2303	N	14	135
29	2303	N	15	134
29	2303	N	16	143
29	2303	N	17	136
29	2303	N	18	153
29	2303	N	19	162
29	2303	N	20	135
29	2303	N	21	112
29	2303	N	22	116
29	2303	N	23	116
29	2304	N	0	0

**Caminos y Puentes Federales de Ingresos y Servicios Conexos (CAPUFE).** Es un organismo público descentralizado del Gobierno Federal de México, con personalidad jurídica y patrimonio propio, que opera y da mantenimiento mayor y menor a los caminos y puentes federales. También participa en proyectos de inversión y coinversión para la construcción y operación de vías generales de comunicación. Su cabeza de sector es la Secretaría de Comunicaciones y Transportes (SCT). El máximo órgano rector de CAPUFE lo constituye un consejo de administración presidido por el Secretario de Comunicaciones y Transportes

**La Red FNI (Fondo Nacional de Infraestructura).** que está integrada por las autopistas y puentes concesionados por el Gobierno Federal al Banco Nacional de Obras y Servicios Públicos (BANOBRAS), a través de la SCT, para su operación, conservación, mantenimiento y explotación, que fueron rescatadas por el FARAC (Fideicomiso de Apoyo al Rescate de Autopistas Concesionadas) al quebrar los concesionarios privados originales. CAPUFE está contratado para la operación y mantenimiento de esta red. A partir de 2008 (febrero) los fondos de FARAC se integraron al Fondo Nacional de Infraestructura (FONADIN ó FNI).

**Telepeaje.** Corresponde a un sistema de comunicación remota entre una antena situada en el pórtico de peaje y un dispositivo TAG que va adherido en el parabrisas del carro. Este mecanismo permite registrar el paso del vehículo de manera automática, sin que deba detenerse por completo en el pórtico de cobro de peajes. Actualmente en México, en las autopistas de cobro de peaje operan los siguientes sistemas de Telepeaje: IAVE y TAG TeleVía.

**TAG.** Es un dispositivo pasivo electrónico que se energiza por medio de un campo electrónico para poder realizar el pago de peaje.

**Lista Blanca:** Lista unificada que contiene todos los clientes validos de todos los operadores interoperables de telepeaje del país.

**Lista Negra:** Lista unificada que contiene la lista de clientes que no cumplen con las reglas de negocio.

**OTIS:** Operadores de Telepeaje Interoperables.

**Scrum master:** Persona que lidera al equipo guiándolo para que cumpla las reglas y procesos de la metodología. Gestiona la reducción de impedimentos del proyecto y trabaja con el Product Owner para maximizar el ROI.

**Product owner (PO):** Representante de los accionistas y clientes que usan el software. Se focaliza en la parte de negocio y él es responsable del ROI del proyecto (entregar un valor superior al dinero invertido). Traslada la visión del proyecto al equipo, formaliza las prestaciones en historias a incorporar en el Product Backlog y las reprioriza de forma regular.

**Team:** Grupo de profesionales con los conocimientos técnicos necesarios y que desarrollan el proyecto de manera conjunta llevando a cabo las historias a las que se comprometen al inicio de cada sprint

**UTC:** Sistema internacional para la medición del tiempo es un estándar de tiempo calculado a partir de una media ponderada de las señales de los relojes atómicos, localizados en cerca de 70 laboratorios nacionales de todo el mundo.

## Fuentes

*Nallely Ortigoza. Capufe ahorra 60% con nuevo proveedor de IAVE*

[www.elfinanciero.com.mx/empresas/capufe-ahorra-60-con-nuevo-proveedor](http://www.elfinanciero.com.mx/empresas/capufe-ahorra-60-con-nuevo-proveedor)

*Caminos y Puentes Federales. Capufe IAVE (Caminos y Puentes Federales de Ingresos y Servicios Conexos)*

[www.capufe-iave.com.mx/](http://www.capufe-iave.com.mx/)

*Banco Nacional de Obras y Servicios Públicos, SNC. Concluye proceso de licitación de Servicio de Gestión de Cobro de Telepeaje Sujeto a Estándares de Desempeño*

[www.gob.mx/banobras/prensa/concluye-proceso-de-licitacion-de-servicio-de-gestion-de-cobro-de-telepeaje-sujeto-a-estandares-de-desempeno?idiom=es](http://www.gob.mx/banobras/prensa/concluye-proceso-de-licitacion-de-servicio-de-gestion-de-cobro-de-telepeaje-sujeto-a-estandares-de-desempeno?idiom=es)

*Metodologías tradicionales y metodologías Ágiles*

[www.eumed.net/libros-gratis/2009c/584/Metodologias%20tradicionales%20y%20metodologias%20agiles.htm](http://www.eumed.net/libros-gratis/2009c/584/Metodologias%20tradicionales%20y%20metodologias%20agiles.htm)

*Michele Casey. Announcing the general availability of Unbreakable Enterprise Kernel Release 4*

[blogs.oracle.com/linux/announcing-the-general-availability-of-unbreakable-enterprise-kernel-release-4](http://blogs.oracle.com/linux/announcing-the-general-availability-of-unbreakable-enterprise-kernel-release-4)

*Instalación de PHP y Oracle Instant Client para Linux y Windows*

<https://www.oracle.com/technetwork/es/articles/dsl/instalacion-php-con-instant-client-2106535-esa.html>

*The DRBD User's Guide*

<https://www.linbit.com/drbd-user-guide/users-guide-drbd-8-4/>