Decanato de Ingeniería en Informática

Escuela de Informática

Análisis Sistema de Selección de Turno Móvil para instituciones con sistemas de turnos tradicionales en la República Dominicana.

Sustentantes:

Br. Jesús Gabriel Sánchez Estrella - 20121902

Br. Mabel Nicole Feliz Henríquez - 20140307

Asesor:

Prof. Santo Rafael Navarro

Monografía para optar por el título de

Ingeniería de Software

Distrito Nacional, República Dominicana

2019

Los datos expuestos en esta investigación es responsabilidad del autor

Análisis Sistema de Selección de Turno Móvil para instituciones con sistemas de turnos tradicionales en la República Dominicana.

TABLA DE CONTENIDO PRELIMINAR

I. Agradecimientos
II. Dedicatorias
III. Resumen Ejecutivo.
IV. Introducción.

CAPÍTULO I: ANTECEDENTES Y ASPECTOS GENERALES DE LOS SISTEMAS DE GESTIÓN DE TURNOS

1.1. Origen de la gestión de turnos en República Dominicana.
1.2. Sistemas de Gestión de Turnos actuales en el país.
1.3. Esquema del proceso actual de Gestión de Turnos.
1.4. Duración promedio para la adquisición del turno y el cumplimiento de éste.
1.5. Insatisfacciones y quejas de los usuarios con respecto a la gestión actual de los turnos en diversas instituciones dominicanas.
1.6. Evaluación de resultados obtenidos en las técnicas de investigación.
1.7. Resumen

CAPÍTULO II: COMPONENTES Y TECNOLOGÍAS ASOCIADAS AL SISTEMA DE GESTIÓN DE TURNOS MÓVIL.

2.1. Sistemas de Posicionamiento Global.
2.2. Sistemas de Recomendaciones.
2.3. Sistema de Información Geográfica (GIS).
2.4. Amazon Web Services y componentes.
2.5. Conceptos principales de Bases de Datos y tipos.
2.6. Lenguajes de programación.
2.7. Servicios Web y las Interfaces de Programación de Aplicaciones.

2.8. Resumen

CAPÍTULO III: PLANTEAMIENTO DE PROPUESTA Y ARQUITECTURA DE LA APLICACIÓN.

3.1. Visión y Alcance de la Propuesta
3.2. Impacto Financiero y Costos a recurrir
3.3. Requisitos y Requerimientos
3.3.1 Requerimientos Funcionales
3.3.2 Requerimientos No Funcionales
3.3.3 Casos de Uso y diagramas de secuencia
3.4. Arquitecturas
3.4.1 Arquitectura App Móvil
3.4.2 Arquitectura de la Base de Datos
3.5. Implementación de los Lenguajes de Programación
3.6. Aplicación del Sistema de Posicionamiento Global (GPS) y el Sistema de Información Geográfica (GIS) al Sistema de Gestión de Turnos Móvil
3.7. Turno móvil como Sistema de Recomendación
3.8. Material para toma de decisiones
3.8.1 Reportes e Informes
3.8.2 Dashboards
3.9 Resumen

1. CONCLUSIONES & RECOMENDACIONES

1. Conclusiones
2. Recomendaciones

1. REFERENCIAS BIBLIOGRÁFICAS

1. Índice de Tablas.

2. Índice Gráficas y Figuras.

ÍNDICE DE TABLAS

Tabla 1. Costos y presupuesto a recurrir.
Tabla 2. Ejemplo de Informe de turno tomado por parámetros.
Tabla 3. Ejemplo de informe de turnos rechazados.

ÍNDICE DE GRÁFICAS Y FIGURAS

Figura 1. Sistemas de Gestión de Turnos Tradicionales.
Gráfica 2. Resultado en Porcentaje de Tickets perdidos.
Gráfica 3. Resultado en Porcentajes de Frecuencia en toma de turnos.
Gráfica 4. Resultado en porcentaje de uso de tickets de dispensador de turnos.
Gráfica 5. Resultado en porcentaje de reacciones al perder un ticket.
Gráfica 6. Resultado en porcentajes de sucursales más abarrotadas al público que otras.
Gráfica 7. Resultado en Porcentaje de idea de desarrollo del Sistema de Gestión de Turnos móviles.
Gráfica 8. Resultado en Porcentajes de solución viable del sistema de gestión de turnos móviles.
Gráfica 9. Resultado en porcentaje de propuesta como ayuda al medio ambiente.
Figura 10. Visión general de navegación GPS.
Figura 11. Recomendación de turnos de sucursal bancaria.

Figura 12. Uso de Superposición en capas de un perímetro en el sistema de información geográfica (GIS).
Figura 13. Componentes de sistema de información geográfica (GIS).
Figura 14. Infraestructura general de Amazon Web Services.
Figura 15. Modelo de Base de Datos genérico.
Figura 16. Caso de uso general.
Figura 17. Caso de uso consulta institución vía móvil.
Figura 18. Caso de uso Listado de instituciones vía móvil.
Figura 19. Caso de uso Listado de instituciones vía administrador.
Figura 20. Caso de uso Ver detalle de institución.
Figura 21. Caso de uso Crear nuevo usuario.
Figura 22. Caso de uso consultar listado usuario vía administrador
Figura 23. Caso de uso consultar detalle usuario administrador
Figura 24. Caso de uso creación nueva institución
Figura 25. Caso de uso Habilitar/ Deshabilitar institución.
Figura 26. Caso de uso Seleccionar/ Rechazar turno.
Figura 27. Diagrama de secuencia para el flujo de consultar una institución desde el portal.
Figura 28. Diagrama de secuencia para el flujo de consultar todas las instituciones desde el portal.
Figura 29. Diagrama de secuencia para el flujo de crear una institución desde el portal.
Figura 30. Diagrama de secuencia para el flujo de habilitar o deshabilitar una institución desde el portal.
Figura. Diagrama de secuencia para el flujo de consultar un usuario de la aplicación móvil desde el portal.
Figura. Diagrama de secuencia para el flujo de consultar todos los usuarios de la aplicación móvil desde el portal.
Figura 33. Diagrama de secuencia para el flujo de registro de usuario desde el móvil.

Figura 34. Diagrama de secuencia para el flujo de eliminar usuario desde el móvil.
Figura 35. Diagrama de secuencia para el flujo de consultar todas las instituciones desde el móvil.
Figura 36. Diagrama de secuencia para el flujo de consultar una institución desde el móvil.
Figura 37. Diagrama de secuencia para el flujo de seleccionar o rechazar turnos de institución desde el móvil.
Figura 38. Arquitectura App móvil.
Figura 39. Arquitectura Base de Datos.
Figura 40. IDE Intellij IDEA.
Figura 41. Visual Studio Code.
Figura 42. Sistema de recomendación de turnos.
Figura 43. Ilustración de un turno desde una aplicación.
Figura 44. Proceso de Recomendación.
Figura 45. Dashboard App.
Figura 46. Esquema de proceso seguido en el capítulo tres sobre la construcción propuesta de la solución de la problemática.
Figura 47. Resumen conceptual de diferencia entre la base de datos que será utilizada para almacenar la data normal (Relacional, izquierda) y la base de datos para recolectar datos espaciales a través del Sistema de Información Geográfica para ser utilizados en el Sistema de Gestión de Turnos Móviles.

AGRADECIMIENTOS

En primer lugar, le doy gracias a Dios, por darme las fortalezas para seguir adelante a pesar de las dificultades que se presentaron en el camino. Gracias, porque me dio la oportunidad se seguir los pasos de quien fue mi inspiración y permitirme llegar hasta aquí.

Gracias infinitas a mi padre por siempre motivarme a dar lo mejor y por siempre escucharme emocionado cuando le contaba cómo iba con mis estudios de programación. Hoy no está con nosotros y aunque duele, sé que estaría orgulloso, y eso me hace sentir feliz.

Gracias a mi querida madre que siempre me ha apoyado y guiado para tomar las mejores decisiones. Gracias porque siempre ha estado ahí cuando más la necesito.

Gracias a mi abuela por siempre cuidarme y educarme cuando mis padres no estaban. Siempre con mucho cariño y dedicación.

Jesús G. Sánchez Estrella

AGRADECIMIENTOS

Agradezco al Dios todopoderoso por permitir que todo esto sea posible, que hoy en día seré una de esas profesionales que con mucho esmero ha logrado expandirse, profesionalizarse y ayudar a otros, gracias a todas las bendiciones que me ha traído a lo largo de mi vida.

Agradezco enteramente a mi familia, porque todo el amor, apoyo incondicional e incluso apoyo económico, fueron los peldaños que me sostuvieron hasta llegar a la máxima meta.

Agradezco felizmente a todos mis amigos y compañeros, que siempre estuvieron allí brindándome todo el apoyo que necesitaba, y que persigan sus sueños hasta llegar a ser lo que desean ser, bajo la voluntad del Señor Jesús.

Agradezco de corazón a todos mis compañeros de trabajo, los que al día de hoy han sido mis superiores y los que también han sido mis subordinados, porque tuve mucho apoyo a través de ellos, y aportaron felizmente cuando lo necesitaba todas las facilidades para que yo continuara mi plan de estudios sin interrupciones o dificultades laborales.

Mabel Nicole Feliz Henríquez

DEDICATORIAS

A mi padre Jesús Sánchez Ortiz, quien fue mi inspiración y guía en este camino. Me hubiera gustado poder compartir este logro con él, poder decirle gracias infinitas por tu apoyo, que esté logro hoy alcanzo no es solo mío, sino tuyo también.

Jesús G. Sánchez Estrella

DEDICATORIAS

A ti, nuestro Dios y Salvador, porque fuiste quién hizo posible todo esto, estoy orgullosa de tenerte en mi vida y quiero que sigas allí en cada paso que dé.

A mi padre Miguel Feliz y Miguelina Henríquez, porque gracias a su fuerza de voluntad, desde mi nacimiento hasta todos los días en la tierra que sigan ustedes acá conmigo, por darme todo lo que necesitaba, apoyo incondicional y ser los mejores padres del mundo, hoy todo lo que tengo se lo debo a ustedes, los amo con todo mi corazón.

A mi hermanita Vanessa Feliz, mi hermanita menor, eres y siempre serás el motivo por el cual siempre me quise superar, para apoyarte y ayudarte a lograr tus sueños. Eres la personita más especial de mi vida.

A ti Erwin Bastardo, porque tú fuiste más allá, cuando más te necesitaba siempre estabas, hoy siéntete orgulloso de que en tus manos tienes a la persona que más te ama en este planeta, mi amor hacia ti no tiene condiciones, este logro también lleva tu nombre. Tu princesa te adora.

A ti Juan Adón, qué te digo, te conocí en la mitad de la carrera, pero no puedo negar que hiciste el papel de como si te conociera desde el principio, en mi vida siempre tendrás un lugar especial, y nadie te lo quitará.

A Magdalena Tavárez y a mi abuelo Miguel Henríquez, por ser una de las personas que me brindó todo su amor y su aportación, mi carrera también fue posible gracias a ellos.

A Magda Henríquez y Miguel Henríquez, mis tíos, qué decirles… Desde que abrí los ojos por primera vez siempre me han agasajado con detalles y mucho amor, hoy a cambio les agradezco todo con mis éxitos. Se lo dedico a ustedes.

A Mis abuelos paternos Luis y Santa, por su apoyo incondicional para lograr mi objetivo.

A Maritza, José, Alexandra, Gabriela y José Gabriel (Los Aracena Guzmán), por ser aquella familia considerada como “Mi familia de crianza” donde

con los últimos tres, crecí y siempre me trataron como parte de su familia siempre que mis padres no estaban. Su casa siempre será mi casa y la mía será la suya. Este logro va para ustedes.

A todos mis tíos del lado de la familia Henríquez, hoy me convierto en un miembro exitoso más de esta gran familia que ha llegado muy lejos. Gracias.

A todos mis amigos que por alguna forma u otra también colaboraron con este logro y a todas aquellas personas incluyendo a Yaritza Miranda, que siempre me demostraron su afecto para conmigo. ¡Esto también va para ustedes! Son muy especiales para mí.

Mabel Nicole Feliz Henríquez

Resumen Ejecutivo.

El siguiente trabajo ofrecerá una excelente alternativa para mejorar el proceso que ejecutan los sistemas de turnos actuales en el país, estos que se usan de manera tradicional y no tienen un estándar de mejora para el síndrome de sobrepoblación actual que sufre la república dominicana.

Debido a los grandes avances tecnológicos producidos cada año y cada siglo, es necesario hacer usos de estos para abrir puertas a la brecha tecnológica que desde varios años nos sigue. El objetivo del presente trabajo se centraliza en el surgimiento de esta nueva alternativa aprovechar el surgimiento del Sistema de Posicionamiento Global (GPS) y Sistemas de Información Geográfica (GIS), tener a la par opciones para predecir y recomendar a los usuarios sobre qué lugar, qué turno y cuánto tiempo le tomará llegar a la ubicación recomendada y así ayudarles a optimizar sus tareas diarias, con un objetivo añadido de volver el proceso más tecnológico y evitar la emisión de material gastable para gestionar y controlar los turnos tomados.

Introducción.

En la República Dominicana, se cuenta con diversas instituciones públicas o privadas cuyo manejo de clientela se basa en el uso de turnos, con el fin de que, en una manera ordenada, de acuerdo muchas veces al orden de llegada o al tipo de solicitud, los turnos se asignan. Con esto, se pretende lograr que los clientes de estas instituciones se atiendan de manera organizada y como pasa muchas veces, evitar conflictos por personas a las que se le dan una preferencia no merecida.

Enfocando el tema de la organización, también suceden casos de que los turnos asignados por un sistema común, pueden extraviarse, como sucede en el caso de la ruleta de tickets de turnos o los counters donde se elige el servicio y este emite un papel, o en los peores de los casos, cuando la persona adquiere un turno escrito por una persona en un pedazo de papel con un lapicero, dando como resultado una serie de conflictos venideros entre las distintas partes o inconformidad de los clientes.

Con motivo de cumplir con nuestros objetivos, la siguiente monografía está compuesta por tres capítulos o secciones principales: CAPÍTULO I: ANTECEDENTES Y ASPECTOS GENERALES DE LOS SISTEMAS DE GESTIÓN DE TURNOS, CAPÍTULO II: PLANTEAMIENTO DE PROPUESTA Y ARQUITECTURA DE LA APLICACIÓN Y CAPÍTULO III: COMPONENTES Y TECNOLOGÍAS ASOCIADAS AL SISTEMA DE GESTIÓN DE TURNOS MÓVIL. En el primer capítulo, se realizará una evaluación de los antecedentes previos al sistema de gestión de turnos en la república dominicana, haciendo énfasis en cómo la población justifica la necesidad de aplicar mejoras a estos sistemas y cuál es el manejo de estos hoy en día. En el segundo capítulo, se definirá la arquitectura propuesta para la ejecución de la aplicación móvil y su estructura. En el tercer capítulo, se estará detallando los componentes y tecnologías

principales que tendrán participación en el sistema de turnos móvil como el mencionado Sistema de Información Geográfica y el Sistema de Posicionamiento global como parte fundamental para coordinar una recomendación exacta para los usuarios de la aplicación móvil.

CAPÍTULO I

ANTECEDENTES Y ASPECTOS GENERALES DE LOS SISTEMAS DE GESTIÓN DE TURNOS

CAPÍTULO I: ANTECEDENTES Y ASPECTOS GENERALES DE LOS SISTEMAS DE GESTIÓN DE TURNOS

Es de suma importancia conocer los antecedentes y ver más allá de los orígenes de donde proviene la gestión de turnos en la República Dominicana, puesto que actualmente existe una población determinada que hace uso de los métodos tradicionales y se torna conveniente la realización de una evaluación acerca de qué tan eficientes son estos métodos y si sería conveniente mejorarlos o sustituirlos parcial o totalmente.

1.1 Origen de la gestión de turnos en República Dominicana.

En la República Dominicana, contamos con una población de más de 10 millones de habitantes, cuya mayor concentración está ubicada en la zona metropolitana del Gran Santo Domingo. Sin importar que las instituciones públicas o privadas cuenten o no con sucursales en otras regiones del país, existen regiones puntuales donde la población se concentra más para solicitar a sus proveedores los artículos o servicios que necesitan consumir. En el mejor ejemplo de la zona metropolitana, muchas instituciones como los centros de atención a clientes, se reúnen largas colas con el fin de obtener el artículo o servicio que se requiere con el desconocimiento si en otra sucursal puede obtenerlo de manera más rápida y eficaz.

1.2 Sistemas de Gestión de Turnos actuales en el país.

La mayoría de las empresas proveedoras de artículos y servicios en la República Dominicana, cuentan con al menos tres formas de gestionar los turnos. La primera es en la obtención de una ruleta de tickets, donde el cliente consumidor saca un ticket y este le indica el número de turno que le toca. La segunda es la colocación de un Counter, que cuenta con un sistema software

que gestiona los diferentes servicios que se pueden ofrecer y al cliente elegirlo, emite un ticket de material gastable con el que el usuario puede seguir o esperar a que en una pantalla su turno sea llamado, muchas veces el usuario cuando es llamado está distraído y no escucha el turno, o lo pierde y tiene que generar otro. La tercera es la colocación de una persona que entregue los tickets en papel manualmente a los usuarios, contribuyendo a la duplicidad y a perder los tickets.

Un ejemplo de sistemas de turnos es el denominado e-flow Gestión de Filas, cuya finalidad consiste en asignar turnos introduciendo el número de identificación personal y servicio a elegir. Sin embargo, este sistema presenta una carencia sustancial en que el ticket no se puede volver a reimprimir, sino tomar uno nuevo, puede fácilmente perderse y no hay un aviso adicional que le permita al usuario saber en cuánto tiempo llegará su turno, por si necesita hacer otra diligencia antes de que este llegue, pudiendo fácilmente el turno pasarse.

1.3 Esquema del proceso actual de Gestión de Turnos.

A continuación, se presentan los dos esquemas principales de gestión de turnos en la República Dominicana:

A. Gestión de Turnos a través de Ruleta de Tickets: Consiste en que el usuario llega al establecimiento, toma un ticket de la ruleta y espera su turno hasta ser atendido.
B. Sistema en máquina de Gestión de Turnos digital: Este sistema cuenta con un counter en la entrada, donde el usuario al llegar al establecimiento coloca sus datos personales y/o elige el servicio que desea adquirir.

1.4 Duración promedio para la adquisición del turno y el cumplimiento de éste.

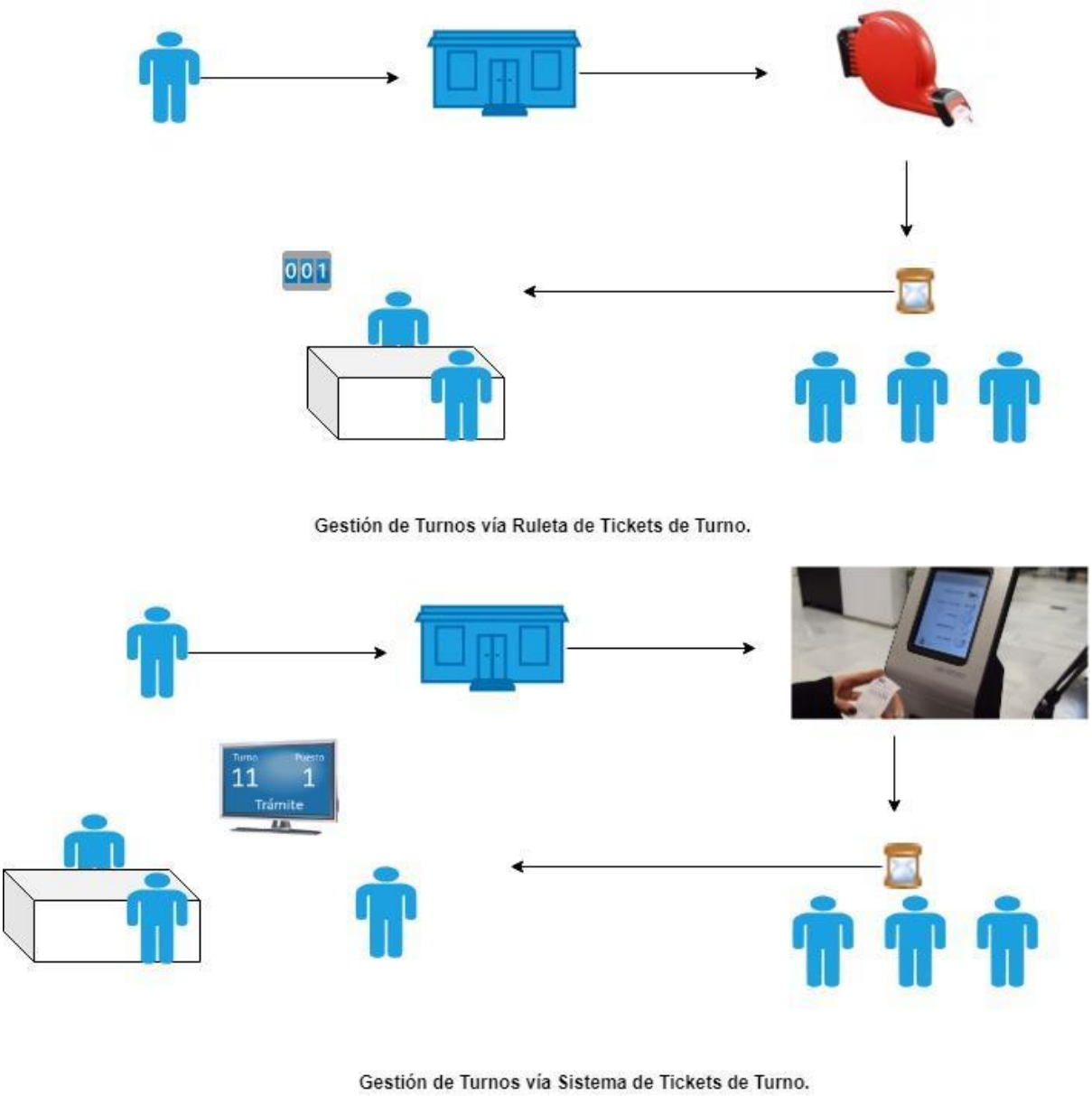


Figura 1. Sistemas de turnos tradicionales. Fuente: Propia

En los sistemas de turnos tradicionales, obtener un turno promedio alrededor de 30 segundos a dos minutos aproximadamente. El cálculo se hace

dependiendo del tipo de sistema tradicional al que se esté refiriendo:

A. Si es una ruleta de tickets, toma alrededor de 30 segundos a 1 minuto con proximidad a demorarse más tiempo si la ruleta queda distante del usuario, o si hay que recargar la ruleta de turnos nuevamente para seguir adquiriendo.

   En el caso de los sistemas de turnos tradicionales digitales, se demora alrededor de 30 segundos a dos minutos. Estos factores e intervalos de tiempo podrían variar si el usuario desconoce cómo obtener el turno, así como la proximidad de la ubicación de la máquina y donde el usuario se encuentra. Otros factores que pudieran reincidir en este hecho son, si hay que cargar la máquina de material gastable (como el papel) para poder imprimir el turno.

1.5 Insatisfacciones y quejas de los usuarios con respecto a la gestión actual de los turnos en diversas instituciones dominicanas.

Los usuarios que utilizan al menos los dos principales sistemas de turnos tradicionales existentes, presentan múltiples quejas que debemos prestar atención. Muchas de las quejas presentadas se resumen en las siguientes:

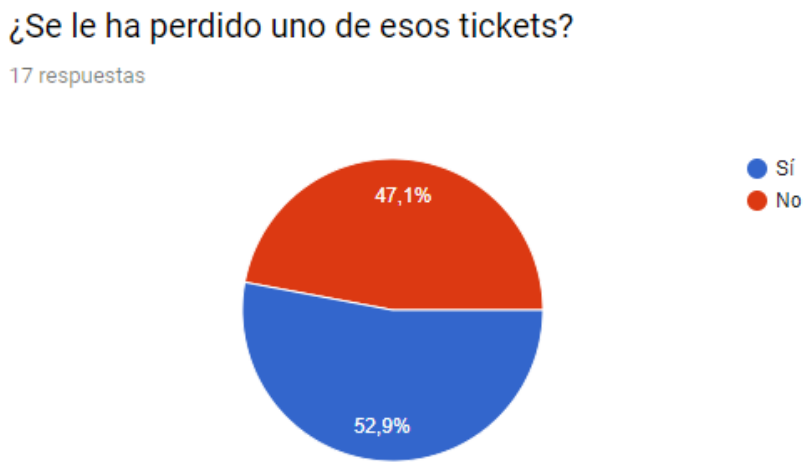
A. El sistema de turnos tradicional no cuenta con un manual de uso que pueda ayudar a los usuarios inexpertos a introducir sus datos y elegir el servicio que desean.
B. El sistema de tickets puede duplicar los tickets.
C. Personas pueden tomar los tickets de otras personas si este deja su ticket abandonado.
D. Los tickets se pueden perder fácilmente.

E. La persona que atenderá al usuario está atendiendo una numeración incorrecta, dando preferencia a otros usuarios que tienen menor tiempo en espera e ignorando los de mayor tiempo de espera.

1.6 Evaluación de resultados obtenidos en las técnicas de investigación.

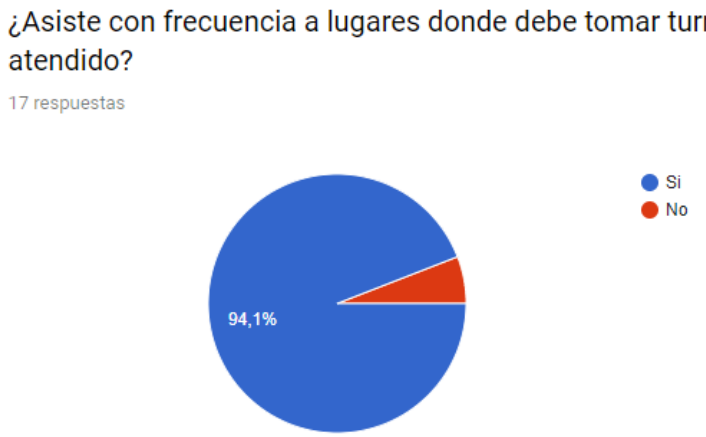
Según los datos recopilados en la encuesta “Satisfacción de la Gestión de turnos actuales en diversas instituciones de República Dominicana” se pudo comprobar:

a) Pérdida de tickets impresos una vez tomados (52,9% de las personas encuestadas) Dando un resultado positivo a la existencia de este hecho.



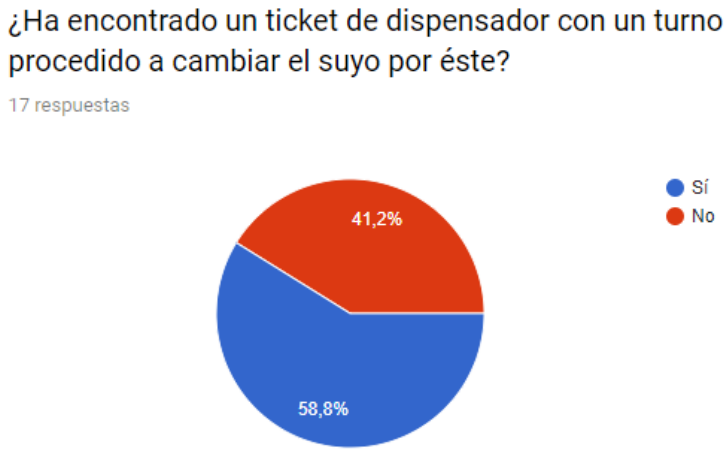
Gráfica 2. Resultado en porcentajes de tickets perdidos Fuente: Propia

b) Los usuarios normalmente frecuentan en lugares donde deben de hacer turnos para obtener sus servicios (94.1% de las personas encuestadas).



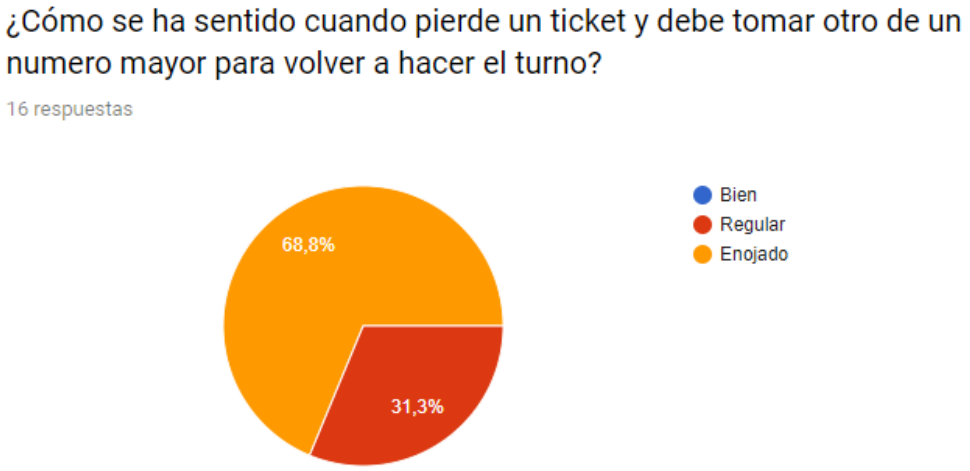
Gráfica 3. Resultados en porcentajes de frecuencia de toma de turnos. Fuente: Propia

c) Alteración de turnos producto de encontrar un ticket con un turno menor (58,8% de las personas encuestadas.)



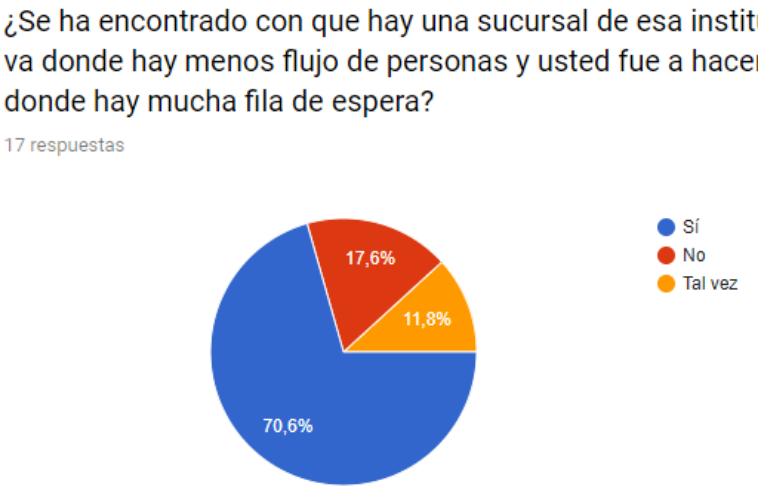
Gráfica 4. Porcentaje de uso de tickets de dispensador de turnos. Fuente: Propia

d) Cambio de humor producido al perder un ticket que se estaba reservando y el usuario está obligado a tomar uno nuevo (68,8% de las personas encuestadas demostraron cambios de humor al encontrarse en esta situación.)



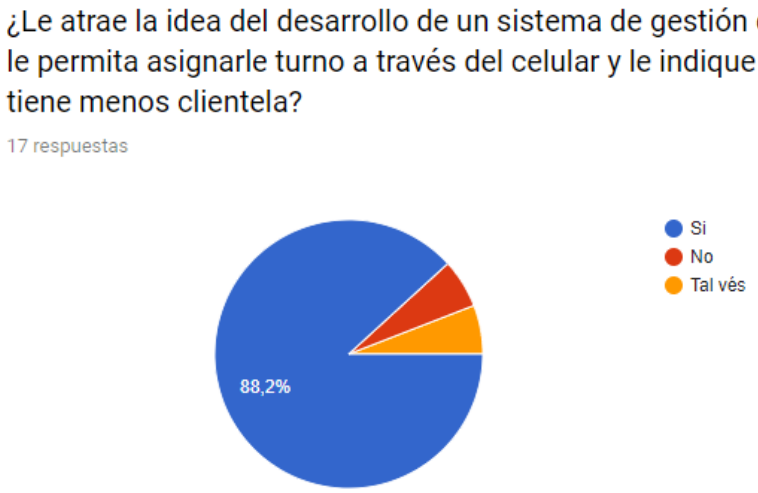
Gráfica 5. Resultado en porcentaje de emociones al perder ticket de dispensador de turnos. Fuente: Propia

e) Los usuarios se han percatado de que habían sucursales que contaban con menor flujo de personas y que podían ser atendidos más rápidamente (El 70% de los encuestados declaró haberse percatado de esa parte).

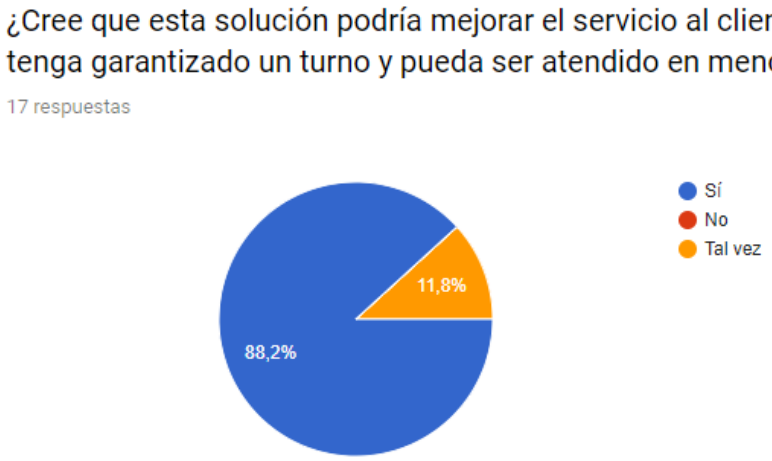


Gráfica 6. Resultado en Porcentajes de sucursales más abarrotadas de público que otras. Fuente: Propia

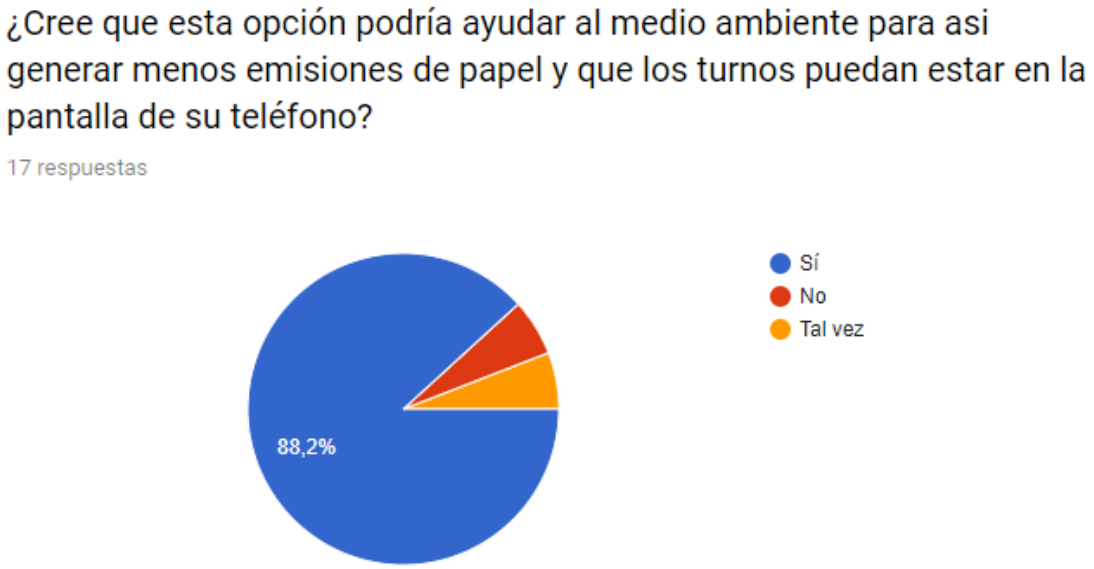
f) Calificación de usuarios en cuanto a la idea de proponer un nuevo Sistema de Gestión de Turnos vía móvil, apoyando la idea de que puede mejorar el servicio al cliente convirtiéndolo en más eficiente y ayudando así al medio ambiente para evitar las emisiones de papel innecesarias (el 88,2% de los encuestados está de acuerdo con la idea).



Gráfica 7. Resultado en porcentaje de idea de desarrollo del Sistema de Gestión de turnos móvil. Fuente: Propia



Gráfica 8. Resultado en porcentaje de solución viable del Sistema de Gestión de Turnos móviles Fuente: Propia



Gráfica 9. Resultado en porcentaje de propuesta como ayuda al medio ambiente. Fuente: Propia

1.7 Resumen

En el capítulo uno, se indagó acerca de los aspectos actuales que las instituciones de la República Dominicana están actualmente haciendo objeto de los sistemas de turnos tradicionales, se determinó que estos sistemas son ambiguos y necesitan de una innovación para eficientizar el servicio al cliente.

También se nombraron las diferentes formas existentes de gestionar turnos, entre las principales se encuentra el gestionar turnos a través de ruleta de tickets, o usando el sistema automatizado donde se eligen los servicios y este imprime un ticket, seguido de la pantalla.

La idea de proponer el Sistema de Gestión de Turnos Móviles tiene el fin de mejorar todas las hendiduras que ha dejado el sistema actual de gestión de turnos. Las encuestas realizadas a la muestra demostraron que un porcentaje muy alto está en desacuerdo a la hora de que se generan los tickets y éstos se encuentran en una gran vulnerabilidad. Muchos de los encuestados propusieron mejoras para el sistema de turnos móviles, creando ellos mismos la necesidad y pidiendo ellos mismos su pronta solución. El apoyo a la emisión de papel afecta al medio ambiente ya que el papel no se suele reciclar, permitiendo la extensión de muerte de árboles que contienen los componentes de papel.

CAPÍTULO II

COMPONENTES Y TECNOLOGÍAS ASOCIADAS AL SISTEMA DE GESTIÓN DE TURNOS MÓVIL.

CAPÍTULO II: COMPONENTES Y TECNOLOGÍAS ASOCIADAS AL SISTEMA DE GESTIÓN DE TURNOS MÓVIL.

Es inminente la necesidad que surge de incluir en la nueva propuesta del Sistema de Gestión de Turnos Móvil el uso de mapas electrónicos, sub seguidos de información alimentada del Sistema de Posicionamiento Global (Por sus siglas GPS) y el Sistema de Información Geográfica (Por sus siglas GIS), teniendo como papel fundamental la búsqueda por medio de coordenadas y superficies, de acuerdo al tiempo y lugar, información útil para brindar sugerencias y recomendaciones al usuario del Sistema de Gestión de Turnos Móvil a la hora de realizar su búsqueda de lugares cercanos con menos flujo de personas y turnos.

2.1 Sistemas de Posicionamiento Global

El sistema de Posicionamiento Global (GPS) es un mecanismo desarrollado por el Departamento de Defensa de los Estados Unidos con el fin de determinar la posición de un objeto o lugar en el espacio terrestre. Este cuenta con un enorme nivel de posición de hasta unos pocos centímetros de longitud, compuesto por alrededor de veinticuatro satélites comunicados sobre la tierra a unos 21,430 kilómetros de altitud. Cuenta con un sistema de coordenadas 8,000 con una cobertura que abarca todo el mundo y cada uno de los veinticuatro satélites que están posicionados tienen una vida útil de 7,5 años a una hora de 1ns.

Cabe destacar que muchos aparatos que actualmente se utilizan cuentan con sistemas GPS, tales como los automóviles, tabletas, teléfonos móviles, ordenadores, dispositivos contra robos de vehículos, entre otros. Estos pueden ser colocados de diversos motivos, ya sea para iniciar una navegación en un mapa, para vigilar el lugar a la que se

traslada una cierta persona con el dispositivo que contiene el GPS, para conocer la ubicación actual, entre otros.

Figura 10. Visión general de navegación con GPS. Fuente: www.Garmin.com

En el proyecto del Sistema de Gestión de Turnos Móvil, se utilizarán las coordenadas medidas a través del GPS con el fin de gestionar la ubicación actual de las diferentes sucursales y proponer diversas opciones al usuario para elegir la opción que más le convenga de acuerdo a su ubicación, espacio y disponibilidad de turnos.

En la imagen siguiente, se mostrará un ejemplo de la búsqueda de una institución bancaria dominicana, llamada Banco Popular Dominicano, donde un usuario aleatorio desea ir al departamento de Servicio al Cliente del banco, y el sistema le da a elegir dos sucursales que se encuentran en su ruta. El punto azul es la ubicación actual del usuario y los dos mensajes se despliegan, con posibilidades de seleccionar lo que más le

convenga.

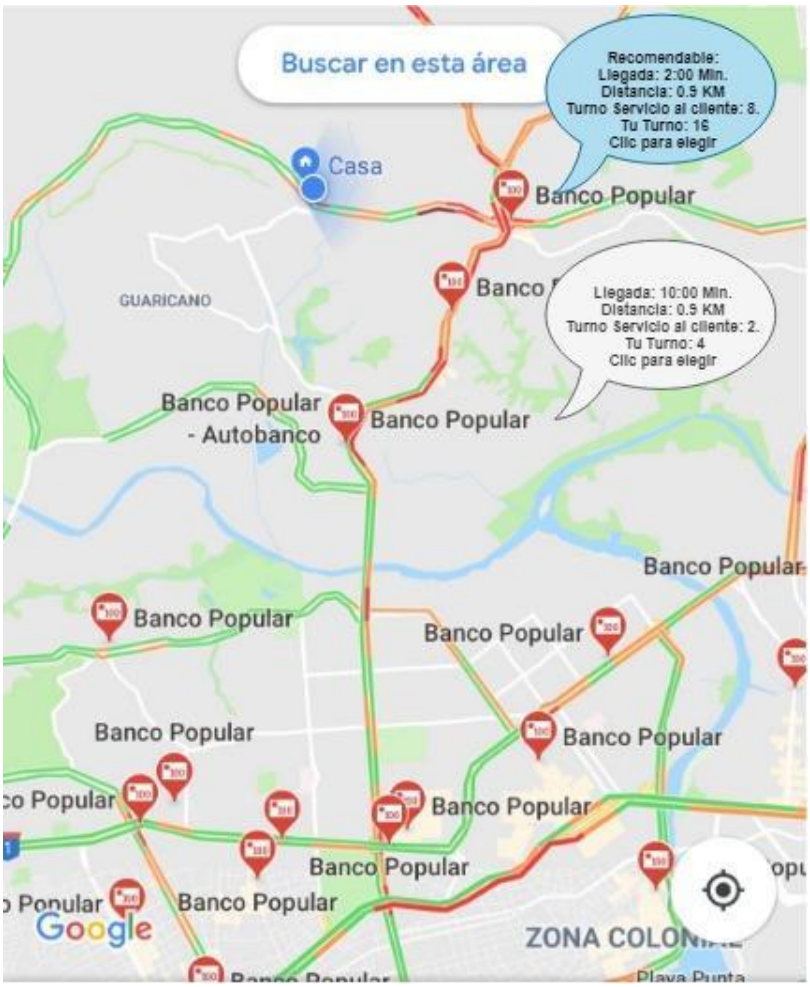


Figura 11. Recomendación de turnos de sucursal bancaria. Fuente: Propia

2.2 Sistemas de Recomendaciones

Un Sistema de recomendación está basado en una serie de algoritmos que, en base a un conjunto de información anteriormente provista, brinda las mejores opciones o alternativas para tomar una decisión.

Algunos ejemplos muy conocidos que podemos mencionar y que utilizamos con mucha frecuencia están:

- Netflix
- Spotify
- Uber
- Anuncios de Google

2.3 Sistema de Información Geográfica (GIS)

El sistema de Información Geográfica (Por sus siglas en Inglés GIS) Es un sistema que se encarga de recopilar información , permite manipular, almacenar y analizar información contenida dentro de un entorno espacial, relacionando así la gran variabilidad de componentes que la conforman, como usuarios, programas, dispositivos y procesos con el fin de tomar decisiones certeras a la hora de analizar los resultados que arroja este sistema.

Estas tecnologías son muy utilizadas en el ámbito científico, puesto que si se desea analizar, por ejemplo, una zona del suelo donde sea factible descubrir algún factor determinante para tomar una decisión seria, es posible hacerlo a través de este sistema.

En la siguiente imagen, se puede apreciar el uso de una superposición de capas en el Sistema de Información Geográfica (GIS). Se puede ver la zona verde la cual se estima está cubierta de forestación esta super posicionada a otra capa que se conforma por curvas de subnivelación y la hidrografía, pudiéndose apreciar los límites geográficos del entorno.

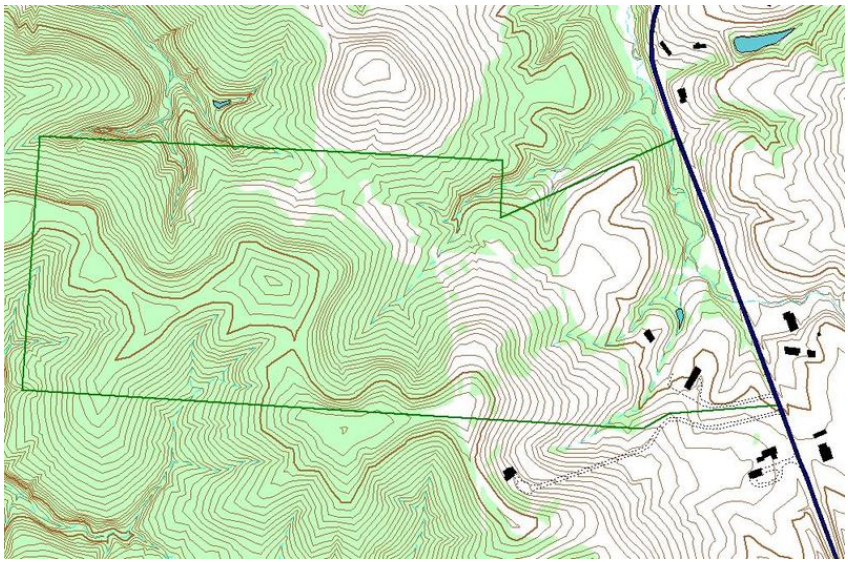


Figura 12. Uso de superposición en capas de un perímetro en el Sistema de Información geográfica (GIS). Fuente: www.gis.ethz.com

Figura 13. Componentes del Sistema de Información Geográfica (GIS) Fuente: http://img.webme.com/pic/2/29-04-91/sigg.jpg

2.4 Amazon Web Services y componentes

AWS (Por sus siglas en inglés de Amazon Web Services) Es un sistema de gestión de nube que proporciona la compañía Amazon. Esta plataforma está compuesta de servidores virtuales, instancias y almacenamiento de bases de datos, replicación, posibilidad de utilizar otros gestores de bases de datos utilizando instancias y proporcionando un sistema de seguridad y alta rentabilidad, proveyendo a sus clientes alta disponibilidad de la información y una capacidad muy alta de peticiones por segundo (alrededor de 20 billones).

Figura 14. Infraestructura general de Amazon Web Services (AWS) Fuente: www.Amazon.com

Las ventajas de utilizar Amazon Web Services para la Implementación del Sistema de Gestión de Turnos Móvil en diferencia con su gran competidor Microsoft Azure son las siguientes:

- Mientras que AWS ofrece un modelo de precios flexible a cambio Azure ofrece un Modelo de precios comparativamente menos flexible en comparación con AWS.
- AWS tiene un mercado de software con un extenso ecosistema de socios: Windows y Linux, a cambio con opciones limitadas de Linux, Azure todavía está construyendo su ecosistema de socios.
- El almacenamiento de EBS es super rápido para Big Data en AWS mientras que en azure el almacenamiento estándar tiene dificultades para big data y, por lo tanto, se requiere almacenamiento premium.
- En AWS el servicio Elastic Compute Cloud (EC2) se paga por hora mientras en azure es por minutos lo que lo hace más costoso.
- AWS ofrece un servicio para archivar y recuperar datos a largo plazo a través de amazon Glacier mientras que Azure no posee esa opción todavía.

En AWS la seguridad se proporciona a través de roles definidos por el usuario con controles de permisos excepcionales ósea una cuenta puede poseer varios usuarios con diferentes roles mientras que Azure proporciona seguridad al habilitar permisos en toda la cuenta.

2.5 Conceptos principales de Bases de Datos y tipos

Se define como datos aquella estructura o partícula más pequeña de la información, la cual no tiene sentido si se toma unitariamente y es necesario relacionar esta para para convertirla en un conjunto de información con sentido.

Se define como Base de Datos a un conjunto de elementos tipo dato que están guardados y que se pueden interrelacionar con otros

datos. Estas bases están compuestas de tablas que contienen esta información y a la vez subdividida en columnas, de tal modo que estas se puedan relacionar y sea posible la interactividad de los datos por parte de programas y usuarios.

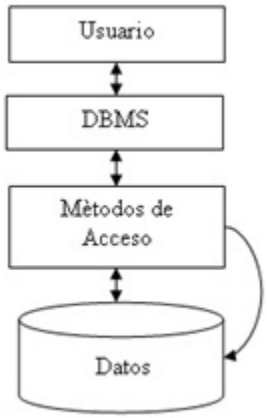


Figura 15. Modelo de base de datos genérico. Fuente: Propia

Existen dos tipos de bases de datos:

- Bases de Datos Relacionales: Son bases de datos que utilizan un modelo relacional y es recomendable que sean usadas cuando hay consistencia en los datos. Un ejemplo de ellas es Oracle, SQL Server, MYSQL.
- Bases de Datos No relacionales o No-SQL: Son bases de datos que cuentan con mayor flexibilidad con la consistencia de los datos y cuentan con algunas añadiduras a favor de las bases de datos relacionales. MongoDB y Dynamo es un ejemplo de ellas.

2.5.1 Base de Datos Oracle

Oracle es una base de datos relacional con manejo de objetos perteneciente a la compañía Oracle Corporation. Es una gran utilidad y cuenta con mucha potencia para el manejo de grandes volúmenes de datos y gestión y mantenimiento en ello. Convirtiéndolo en uno de los gestores más potentes del mercado.

2.5.2 Microsoft SQL Server

SQL Server es una base de datos relacional confeccionada por Microsoft. Esta base de datos se caracteriza por su fácil uso y también cuenta con gran estabilidad y disponibilidad semejantes a Oracle. Su versión web es la base de datos de Azure gestionada también por la misma Microsoft.

2.5.3 Base de Datos MYSQL

MYSQL es un sistema de base de datos también relacional cuyo licenciamiento al público es totalmente gratuito y para las corporaciones o empresarial es al pago con Oracle Corporation.

2.5.4 MongoDB

MongoDB es un sistema de bases de datos no relacional cuyos datos no son almacenados en tablas, sino en estructuras con mayor dinamismo como lo es la estructura BDSON (como un esquema binario donde se usa un esquema preliminar entre data y planos o mapas).

2.5.5 Dynamo DB

En cambio, la base de datos que será de uso en el proyecto actual, será la Dynamo DB, que es una base de datos no relacional (las

estructura son similares a la MongoDB) la cual proporciona un mayor rendimiento y escalabilidad y cuenta como una ventaja crucial el uso de que está albergada en la nube, manejando por segundo hasta 20 millones de peticiones con alta replicación síncrona, lo que facilita tener una alta rentabilidad, integridad y disponibilidad, logrando a su vez tener una capa de seguridad contra eventos no deseados ya incluida.

2.5.6 PostGIS

Un nuevo concepto entrante relacionado con el Sistema de Información Geográfica (GIS) es la base de datos PostGIS, es una base de datos que transforma datos ráster y con utilización de vectores con los cuales podemos conseguir una amplia variedad de datos en el entorno analizado, dando a lugar la oportunidad de usar herramientas para analizar estas informaciones y cada uno de los datos geográficos obtenidos, contando además con un visor geométrico incrustado. La arquitectura de esta base de datos está centrada en PostgreSQL, permitiendo la creación de una base de datos limitada espacialmente.

2.6 Lenguajes de programación.

Un lenguaje de programación es un conjunto de comandos, instrucciones y otros usos de sintaxis para crear un programa de software. Los lenguajes que los programadores utilizan para escribir código se denominan "lenguajes de alto nivel". Este código se puede compilar en un "lenguaje de bajo nivel", que es reconocido directamente por el hardware de la computadora.

Los lenguajes de alto nivel están diseñados para ser fáciles de leer y entender. Esto permite a los programadores escribir código fuente de forma natural, utilizando palabras y símbolos lógicos.

2.6.1 Java

Según el creador de Java, James Gosling: “Java no es mágico, es una herramienta para que la gente construya cosas. Es el martillo y el

clavo. Resulta fácil hacer cosas como AI y otros sistemas bastante inteligentes”

Java es un lenguaje de programación que produce software para múltiples plataformas. Cuando un programador escribe una aplicación Java, el código compilado (conocido como código de bytes) se ejecuta en la mayoría de los sistemas operativos (OS), incluidos los sistemas operativos Windows, Linux y Mac. Java deriva gran parte de su sintaxis de los lenguajes de programación C y C ++.

2.6.2 Angular

La página oficial de angular la define como una plataforma que facilita la creación de aplicaciones con la web. Angular combina plantillas declarativas, inyección de dependencia, herramientas de extremo a extremo y mejores prácticas integradas para resolver los desafíos de desarrollo. Angular permite a los desarrolladores crear aplicaciones para la web, el móvil o el escritorio.

Angular es un lenguaje que tiene similitudes con C# o Java, debido a esto los desarrolladores con cierta experiencia en esos lenguajes se les facilitara el proceso de aprendizaje. Angular maneja los conceptos de

clases, interfaces, tipado, entre otros conceptos que también comparte C# o Java.

2.6.3 TypeScript

Según Aaron Cornelius: “TypeScript es una opción inteligente al escribir una aplicación moderna basada en web o JavaScript. Las características y funciones del lenguaje cuidadosamente consideradas de TypeScript, y sus herramientas que mejoran constantemente, dan como resultado una experiencia de desarrollo tremendamente productiva”.

Es un lenguaje de programación de código abierto desarrollado por Microsoft. Es un superconjunto sintáctico estricto de JavaScript y agrega tipografía estática opcional al lenguaje.

TypeScript está diseñado para el desarrollo de grandes aplicaciones y compilaciones a JavaScript. Dado que TypeScript es un superconjunto de JavaScript, los programas existentes de JavaScript también son programas válidos de TypeScript. TypeScript puede usarse

para desarrollar aplicaciones JavaScript para la ejecución del lado del cliente y del servidor (Node.js).

2.7 Entornos de desarrollo integrado:

Los entornos de desarrollo integrado (IDE por sus siglas en inglés), son herramientas que nos facilitan el proceso desarrollo. Entre las bondades de estas herramientas están:

- Detección de errores de sintaxis mediante resaltados.
- Plugins
- Integración con servicios de terceros
- Generación de código
- Sugerencias de buenas prácticas

2.7.1 Eclipse IDE

Es un entorno de desarrollo de código abierto para Java, extensible mediante el uso de plugins. El proyecto de Eclipse fue originalmente creado por IBM (International Business Machines) en noviembre del 2001.

Este IDE era uno los favoritos para desarrollar en Java pero debido a su complejidad, inestabilidad y diseño un tanto anticuado, ha perdido popularidad en los últimos años. Algunos de los errores observados son

al momento de generar archivos de recursos, los cuales los crea en el root del proyecto en vez de la carpeta resources.

2.7.2 NetBeans IDE

NetBeans IDE le permite desarrollar rápido y fácilmente aplicaciones de escritorio, móviles y web de Java, así como aplicaciones HTML5 con HTML, JavaScript y CSS. El IDE también proporciona un gran conjunto de herramientas para desarrolladores de PHP y C / C ++. Es gratuito y de código abierto y tiene una gran comunidad de usuarios y desarrolladores en todo el mundo.

2.8 Servicios Web y las Interfaces de Programación de Aplicaciones

Un servicio web consiste en una serie de métodos para realizar peticiones HTTP. Su popularidad se debe a que los servicios

pueden ser consumidos por cualquier otro lenguaje de programación diferente al que fue desarrollado el servicio. Estos pueden ser de 5 tipos:

- **GET:** Este método es utilizado para consultar uno o más registros.
- **POST:** Este método es utilizado para guardar un registro. La información a ser guardada viaja en el cuerpo del mensaje.
- **PUT:** Este método es utilizado para actualizar un registro. La información a ser actualizada viaja en el cuerpo del mensaje.
- **DELETE**: Este método es utilizado para borrar un registro.
- **PATCH:** Este método es para actualizar parcialmente un registro

Las Interfaces de Programación de Aplicaciones (API por sus siglas en inglés) definen las especificaciones de cómo un componente de software se comunica con otro.

2.8 Resumen

En el capítulo número dos, se indagó acerca de los conceptos básicos que componen cada una de las herramientas que serán utilizadas para el análisis del Sistema de Gestión de Turnos Móvil. Los conceptos incluidos poseen todas sus posibles clasificaciones y todos se enmarcan dentro del contexto que está en el proceso de análisis. Las bases de datos, los lenguajes de programación, las interfaces, el modelo GIS, los Web Services y la estructura dinámica en la nube de Amazon Web Services (AWS) que engloba todo el esquema. También, la estructura de los servicios web y API, como los servicios Get, Post, Put, entre otros.

En este capítulo se encuentran las diferentes definiciones acerca de la metodología de trabajo utilizando el Sistema de Información Geográfica, lo que nos brinda junto a la conectividad del Sistema de Posicionamiento global, con el fin de identificar aquellas parametrizaciones necesarias que pueden utilizarse en la confección del Sistema de Gestión de Turnos Móvil.

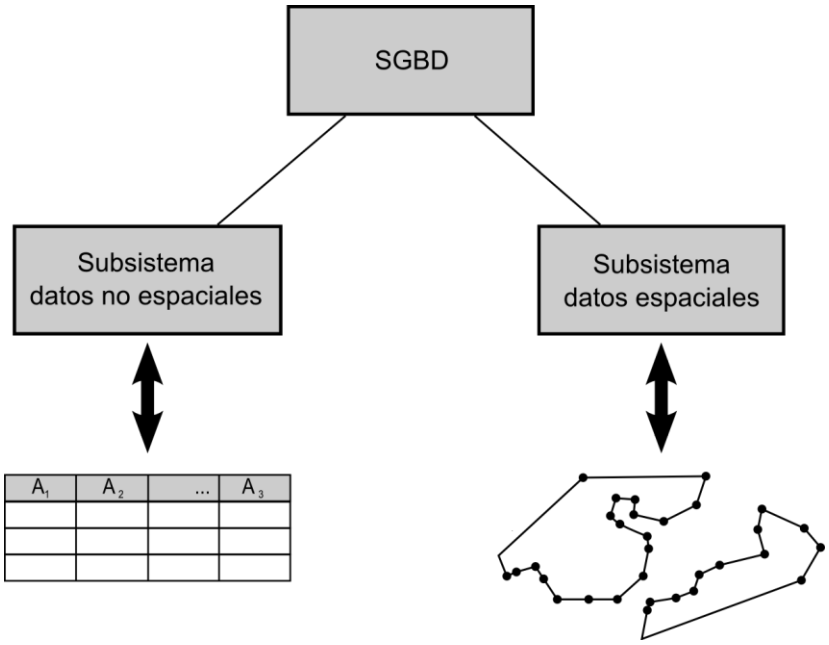


Figura 47. Resumen conceptual de diferencia entre la base de datos que será utilizada para almacenar la data normal (Relacional, izquierda) y la base de datos para recolectar datos espaciales a través del Sistema de Información Geográfica para ser utilizados en el Sistema de Gestión de Turnos Móviles. Fuente: google.com

CAPÍTULO III

PLANTEAMIENTO DE PROPUESTA Y ARQUITECTURA DE LA APLICACIÓN.

CAPÍTULO III: PLANTEAMIENTO DE PROPUESTA Y ARQUITECTURA DE LA APLICACIÓN

La necesidad de reconstruir y mejorar la perspectiva de un sistema que gestione turnos nace a través de la necesidad de hacerlo más eficiente, con menos dificultad de uso y facilitando al usuario de múltiples opciones de la cual elegir. En este capítulo se estarán proponiendo cual es el motivo principal del nuevo proyecto, cuáles son los costos en que se incurre, la descripción de los recursos como son los requisitos de hardware y software, los requerimientos funcionales y no funcionales, la estructura de la base de datos, entre otros aspectos.

3.1 Visión y Alcance de la Propuesta.

Visión:

Convertir la propuesta de la aplicación en una revolución tecnológica con capacidad de seleccionar los turnos de manera inteligente y eficiente, mejorando exponencialmente los tiempos de espera, logrando de esta manera mayor conformidad y una mejor experiencia de servicio para los clientes.

Alcance:

Proponer un sistema para la selección y recomendación de turnos vía móvil en la ciudad de Santo Domingo capaz de gestionar los turnos a través de la ubicación de las instituciones registradas, asegurando la disponibilidad del servicio y la eficiencia en el servicio al cliente.

3.2 Impacto Financiero y Costos a recurrir

El impacto financiero para el proyecto del sistema de gestión de turnos es un factor importante a la hora de medir la rentabilidad del proyecto. Este impacto, se estima, que irá en un crecimiento porcentual alto debido a la plataforma tecnológica que serán utilizadas, como los servicios en la nube, cuya garantía se estima considerada con menos riesgo de sufrir impactos de daños. Los planes de contingencia, como los servicios de respaldo, se encuentran como una de las prioridades con el fin de disminuir los riesgos de una caída de la plataforma y reducir el tiempo de respuesta antes esas eventualidades. Otro de los factores importantes son los respaldos en físico, por si se presenta la situación de dificultad de acceso a los respaldos de la nube. La disponibilidad e integridad de la aplicación estarán salvaguardadas en servidores potentes capaz de gestionar múltiples peticiones en el menor lapso de tiempo.

El personal recibido para la tarea de desarrollo se conforma de un equipo de tamaño pequeño considerados como recursos, cuyas nóminas oscilarán entre los RD$295.000,00 al mes. Están incluidas los gastos de arrendamientos y servicios comunes.

En la siguiente tabla se encontrarán de forma más detallada todos los recursos incluidos y los costos a recurrir.

Tipo	Descripción	Cantidad	Precio Unitario	Sub - Total	Tiempo (Por Mes)	Total
Hardware	Dell PowerEdge R230 Intel Xeon Silver 4110, S-3647, 2.10GHz, 8-Core, 11MB L3 Cache con 16GB DDR4 de RAM y 5 Tb Hard Drive	1	RD$65.660,00	RD$65.660,00	-	RD$65.660,00
Hardware	SQL Server 2017 Standard -Servidor + cal	1	RD$46.965,00	RD$46.965,00	-	RD$46.965,00
Software	Licencia de Linux	1	-	-	-	-
Software	Office Professional	1	RD$35.350,00	RD$35.350,00	12	RD$35.350,00
Hardware	Servicios de respaldo Amazon Glacier 10 TB al mes	1	RD$3.500,00	RD$3.500,00	12	RD$42.000,00
Servicio Nube	Servidor web Amazon AWS EC2 (7 Cores, 128 bits, 32 GB, 2 x 1900 NVMe SSD /EBS)	1	RD$1.353.682,00	RD$1.353.682,00	12	RD$1.353.682,00
Recurso	Arquitecto de Software	1	RD$65.000,00	RD$65.000,00	8	RD$520.000,00
Recurso	Desarrollador	2	RD$45.000,00	RD$90.000,00	12	RD$1.080.000,00
Recurso	Ingeniero de Base de Datos	1	RD$45.000,00	RD$45.000,00	4	RD$180.000,00
Recurso	Ingeniero Electrónico	1	RD$45.000,00	RD$45.000,00	12	RD$540.000,00
Recurso	Tester	2	RD$25.000,00	RD$50.000,00	12	RD$600.000,00
Gastos de Arrendamiento	Local Oficina	1	RD$10.000,00	RD$10.000,00	12	RD$120.000,00
Servicios	Servicios comunes, eléctricos, de red, limpieza.	1	RD$12.000,00	RD$12.000,00	12	RD$144.000,00
Total general del proyecto (Todos los montos expresados en Peso Dominicano (RD$) --->						**RD$4.727.657,00**

Tabla 1. Costos y presupuesto a recurrir. Fuente: Propia

3.3 Requisitos y Requerimientos

Los requisitos y requerimientos son aquellos atributos de calidad necesarios para el correcto funcionamiento del aplicativo. Es decir, todas las funcionalidades, cualidades o características y sus respectivas restricciones que el Sistema de Turno Móvil necesita para su correcto funcionamiento.

3.3.1 Requerimientos Funcionales

I. Consultar instituciones

A. Los usuarios podrán consultar desde el aplicativo móvil, las instituciones registradas en el sistema de turnos utilizando el criterio de nombre.
B. Los usuarios podrán digitar el criterio de nombre.
C. Los usuarios podrán consultar utilizando comandos de voz.

D. Personal administrativo podrá consultar desde el portal de administración, las instituciones registradas en el sistema de turnos de acuerdo a distintos criterios de búsqueda: nombre, ciudad, activo, inactivo.

II. Seleccionar una institución

A. Los usuarios podrán seleccionar una institución de las 3 recomendadas al momento de consultar.
B. Los usuarios podrán cancelar el proceso de selección.
C. Los usuarios podrán cambiar el criterio de búsqueda de ser requerido.
D. Los usuarios tendrán un lapso de 10 segundos para realizar una acción, de lo contrario se seleccionará la primera opción.

III. Asignar turno

A. El sistema le deberá asignar un turno al usuario en la institución que seleccionó.
B. El sistema le deberá asignar un turno al usuario en la institución de mayor recomendación en un lapso de 10 segundos sin el usuario no realiza una acción.
C. El usuario podrá cancelar su turno y el sistema lo reasignará a los usuarios subsiguientes.

IV. Gestionar Instituciones

A. Personal administrativo podrá habilitar / deshabilitar instituciones desde el portal administrativo.

3.3.2 Requerimientos No Funcionales

I. Requisitos de protección y seguridad:

A. La base de datos debe ser diseñada con un alto nivel de seguridad, confiabilidad y confidencialidad.
B. La base de datos debe mantener copias de seguridad secuenciales.
C. El diseño del sistema debe permitir que la base de datos se mantenga sin riesgo de modificaciones por parte de personas ajenas o no autorizadas. Por lo tanto, el sistema deberá estar diseñado con varios niveles de acceso de acuerdo a categorías de usuario como son:

    1. Administrador de base de datos: Pueden consultar, crear, editar y borrar contenidos.
    2. Personal administrativo: Pueden obtener informes estadísticos de consultas por distintos criterios. Pueden habilitar o inhabilitar instituciones.
    3. Usuario: Podrán ingresar al sistema con el propósito de tomar un turno.

II. Requisitos de recursos de cómputos:

A. Requisitos de Hardware:

Los requisitos de hardware se establecen en base al requerimiento de un software de tipo open source utilizando Ionic para la aplicación móvil como tecnología híbrida. El cual, tendrá un diseño basado en móvil/web y una arquitectura Cliente-Servidor.

B. Equipo servidor para respaldo:

1. El servidor de base de datos deberá tener capacidades de copias de seguridad automáticas en la nube de Amazon Glacier y copias automáticas en físico dentro del disco duro local. Amazon Glacier es un gestor de almacenamiento de AWS Service, que permite almacenar grandes volúmenes de datos por una cantidad de tiempo definida, con precio mensual.
2. Servidor Dell PowerEdge R230 Linux Debian con 16GB DDR4 de RAM y 10 TB de disco duro
3. Servidor de Amazon Web Service EC2.
4. Procesador Intel Xeon Silver 4110, S-3647, 2.10GHz, 8-Core, 11MB L3 Cache.
5. Servidor físico con acceso a internet.

C. Clientes:

1. Dispositivos móviles con sistemas operativos Android en su versión 6.0 o superior y Iphone en su versión 11 o superior
2. Cualquier PC con capacidad para ejecutar un explorador WEB.

D. **Requisitos de Software:**

1. El software requerido está orientado al desarrollo de un sistema de tipo código abierto.
2. La aplicación móvil estará desarrollada utilizando Ionic, el cual es un SDK de código abierto para desarrollo de aplicaciones móviles híbridas.
3. El portal web será un SPA (Single page application), utilizando Angular como framework de desarrollo web.
4. El backend estará desarrollado en Java, el cual estará publicando servicios de tipo REST, utilizando SQL Server como gestor de base de datos.

III. Requisitos de Apariencia o Interfaz Externa:

1. El sistema debe ser intuitivo y de fácil uso.

IV. Requisitos de Rendimiento:

1. El sistema deberá poder ser usado 24 horas al día los 365 días del año.
2. El sistema deberá poder ser actualizado a intervalos regulares dependiendo de si han ocurrido actualizaciones.
3. El sistema deberá poder responder en no menos de dos segundos desde la solicitud de usuario, no obstante, podrá tomar más tiempo al realizarse trabajos de procesamiento más grandes.
4. Las respuestas de solicitud de información no deberán tomar más de cinco segundos en aparecer en pantalla.

3.3.3 Casos de Uso y diagramas de secuencia

Los casos de uso son la representación visual de los requerimientos funcionales. Es decir, la representación de las diferentes funcionalidades y los actores que intervienen en cada de una de ellas.

Un caso de uso se encarga de describir detalladamente la acción que realiza una persona u objeto. Esta se basa en el uso de los esquemas de UML que contienen figuras que pueden ser utilizadas con el fin de describir cuál será la serie de funciones o procesos que el usuario tendrá en sus manos para realizar una actividad. Estos usuarios poseen el nombre de Actores, mientras que cada una de las funcionalidades y pasos serán denominada Casos de Uso.

Los diagramas de secuencia son la representación visual de los flujos que ocurren a nivel de código. Es que decir cuando un actor realizar una acción en el sistema, un conjunto de clases que soportan esta funcionalidad se ven afectados, y dicho proceso a nivel de clases es el flujo representado en los diagramas de secuencia.

Durante esta sección se mostraremos los diferentes diagramas de casos de uso y de secuencia que intervienen en el Sistema de Turnos Móvil.

3.3.3.1 Casos de Uso (UML):

En el diagrama UML principal, se encuentran dos actores, el usuario final y el administrador de sistemas, cuya funcionalidad en común es la de generar informes, pues tanto el usuario final como el administrador pueden ver un historial de los usos que le han dado a la

aplicación con respecto a la data guardada.

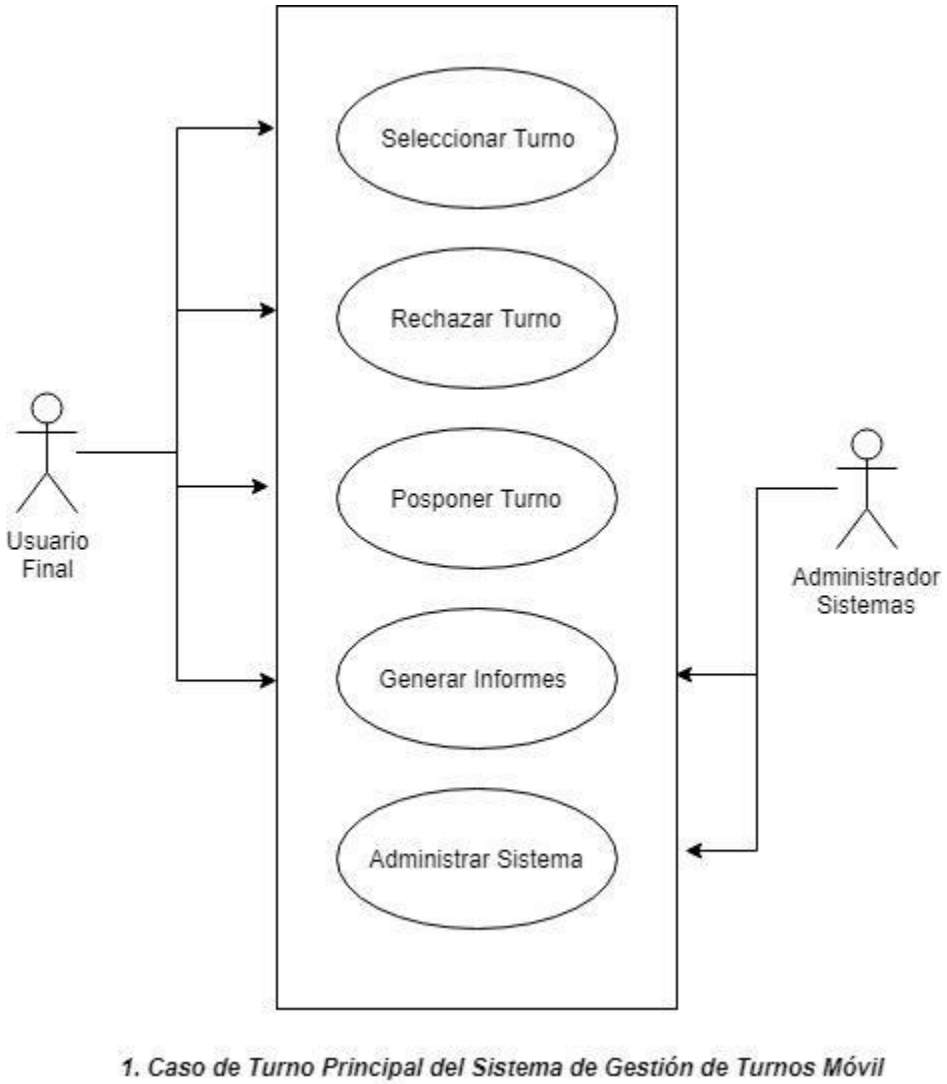


Figura 16. Caso de uso general. Fuente: Propia

En el segundo caso de uso, se encuentra como actor principal el usuario móvil, donde realiza el proceso de visualizar el detalle de una institución en la aplicación móvil.

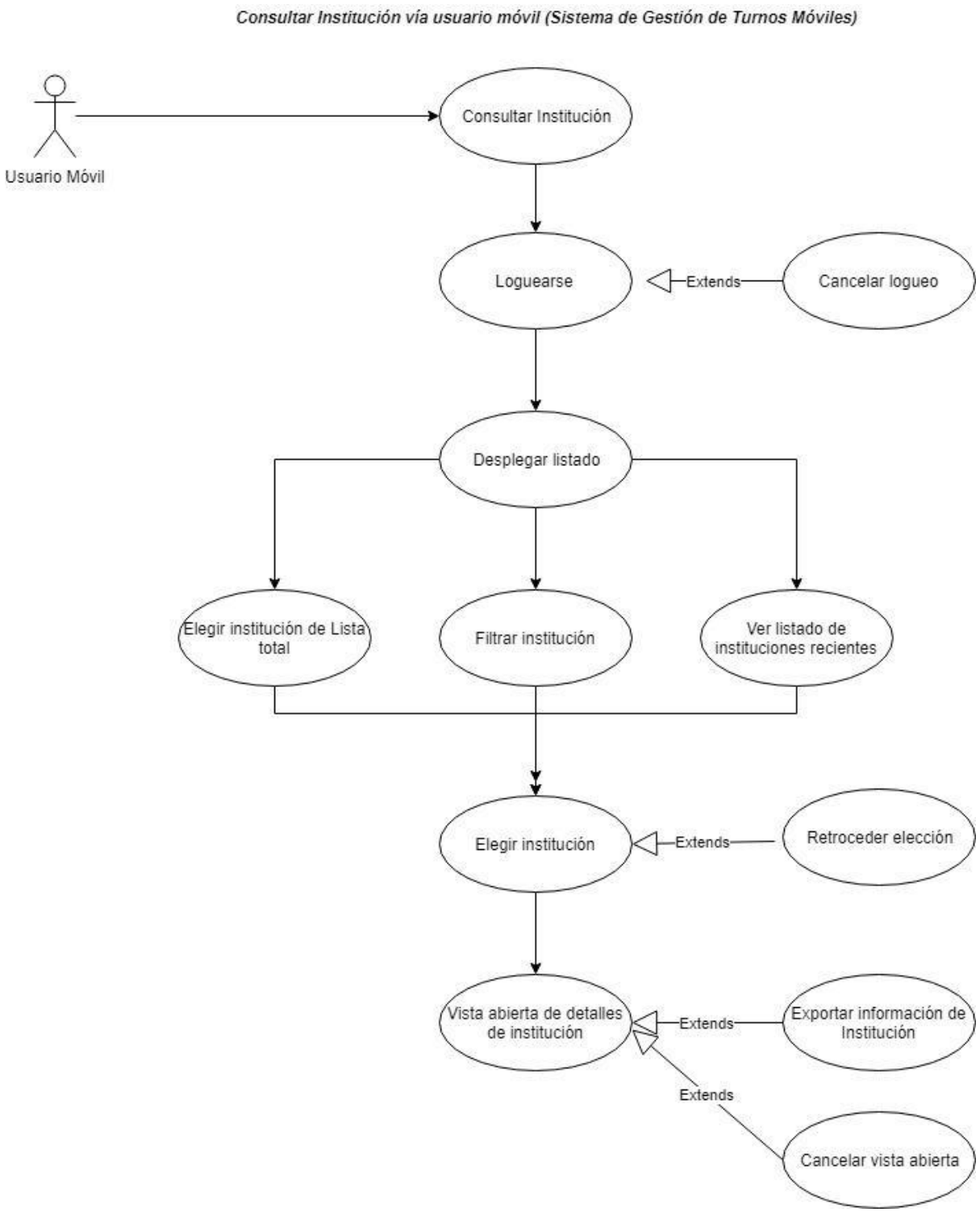


Figura 17. Caso de uso Consulta Institución vía móvil. Fuente: Propia

En el siguiente caso de uso se aprecia el proceso de ver el listado total de instituciones desde la aplicación móvil.

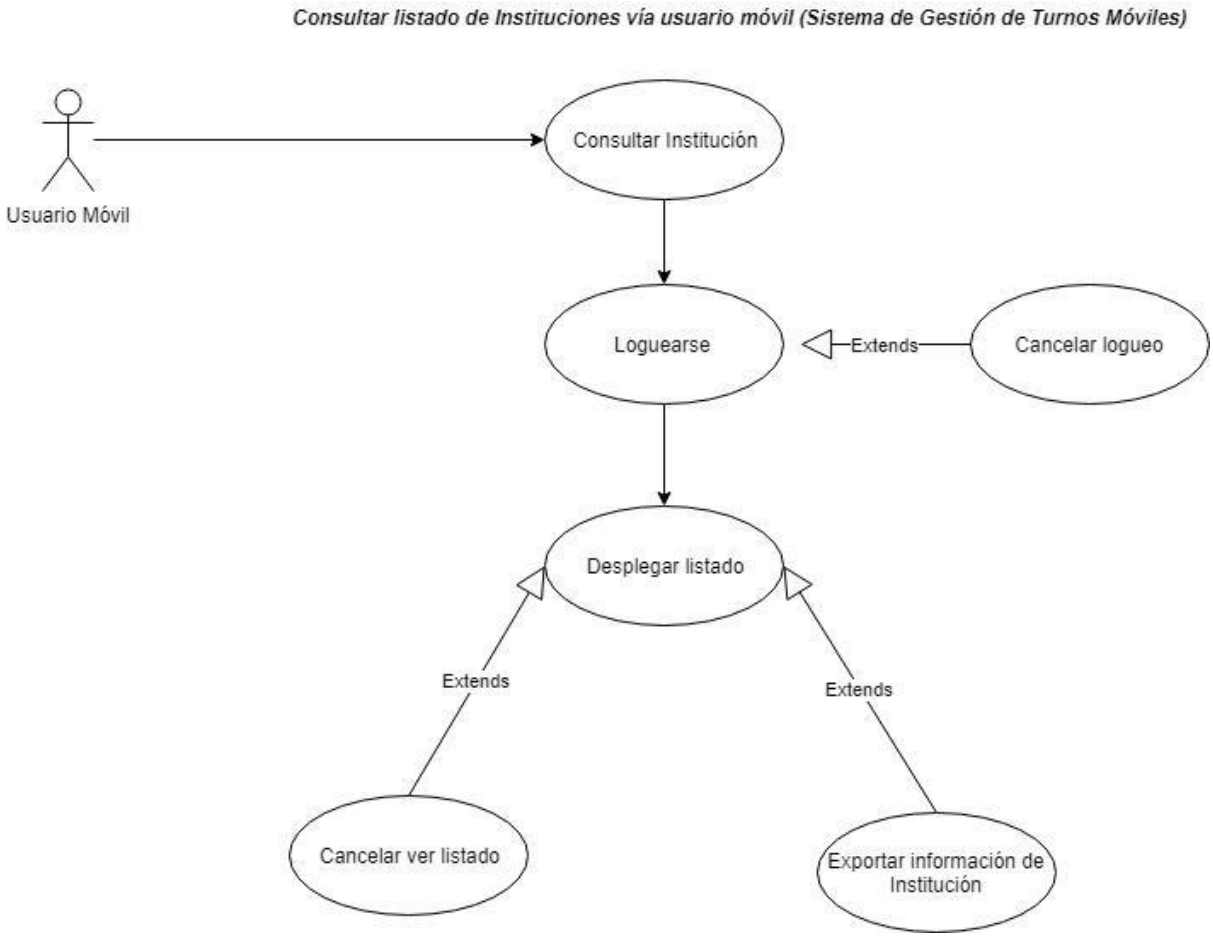


Figura 18. Caso de uso Listado de Usuarios vía móvil. Fuente: Propia

En el siguiente caso de uso, se muestra el proceso para ver el listado de todas las instituciones registradas desde el actor Administrador:

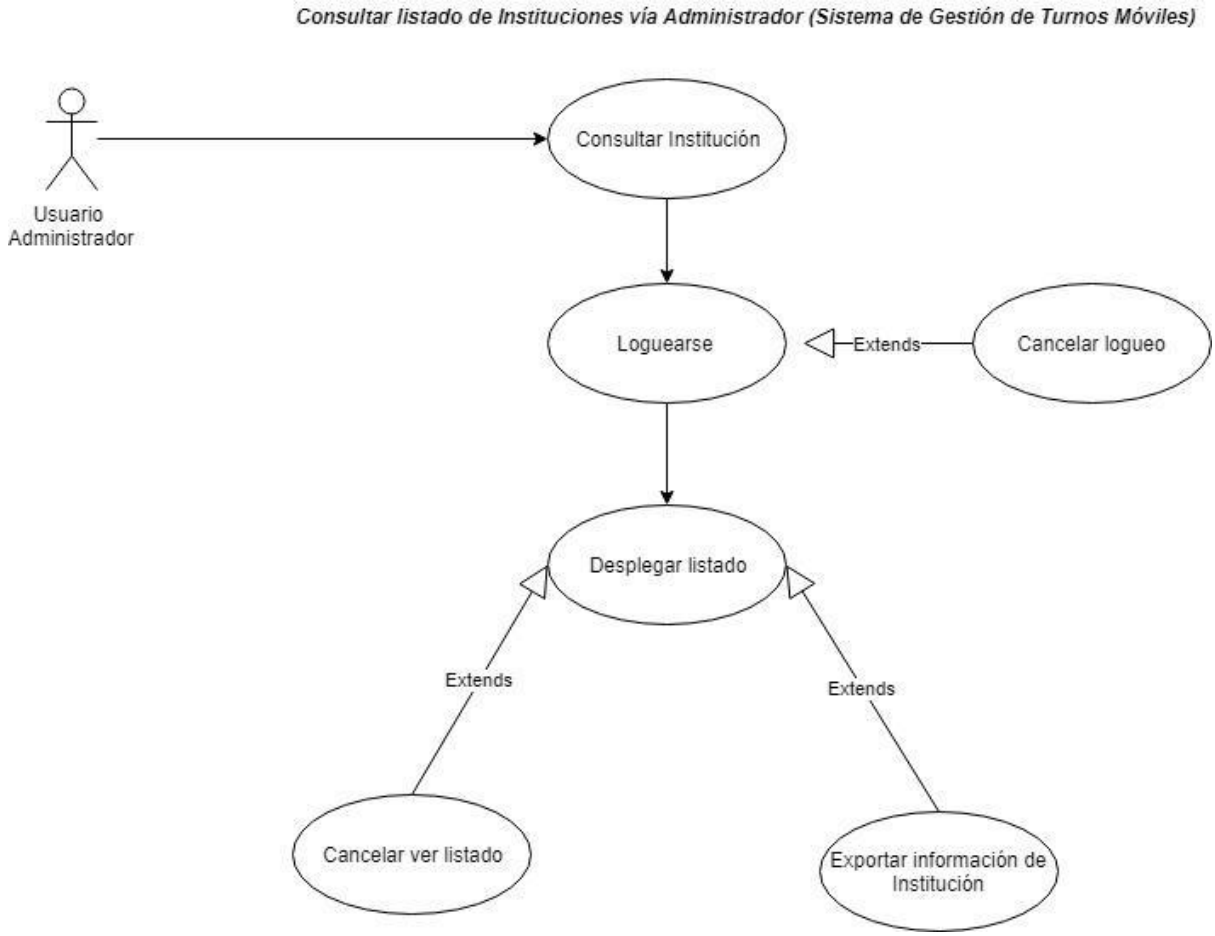


Figura 19. Caso de Uso Listado consultar instituciones vía administrador.
Fuente: Propia

En el siguiente caso de uso, se puede ver el proceso que se requiere para visualizar el detalle de ver una institución en particular y las diferentes opciones.

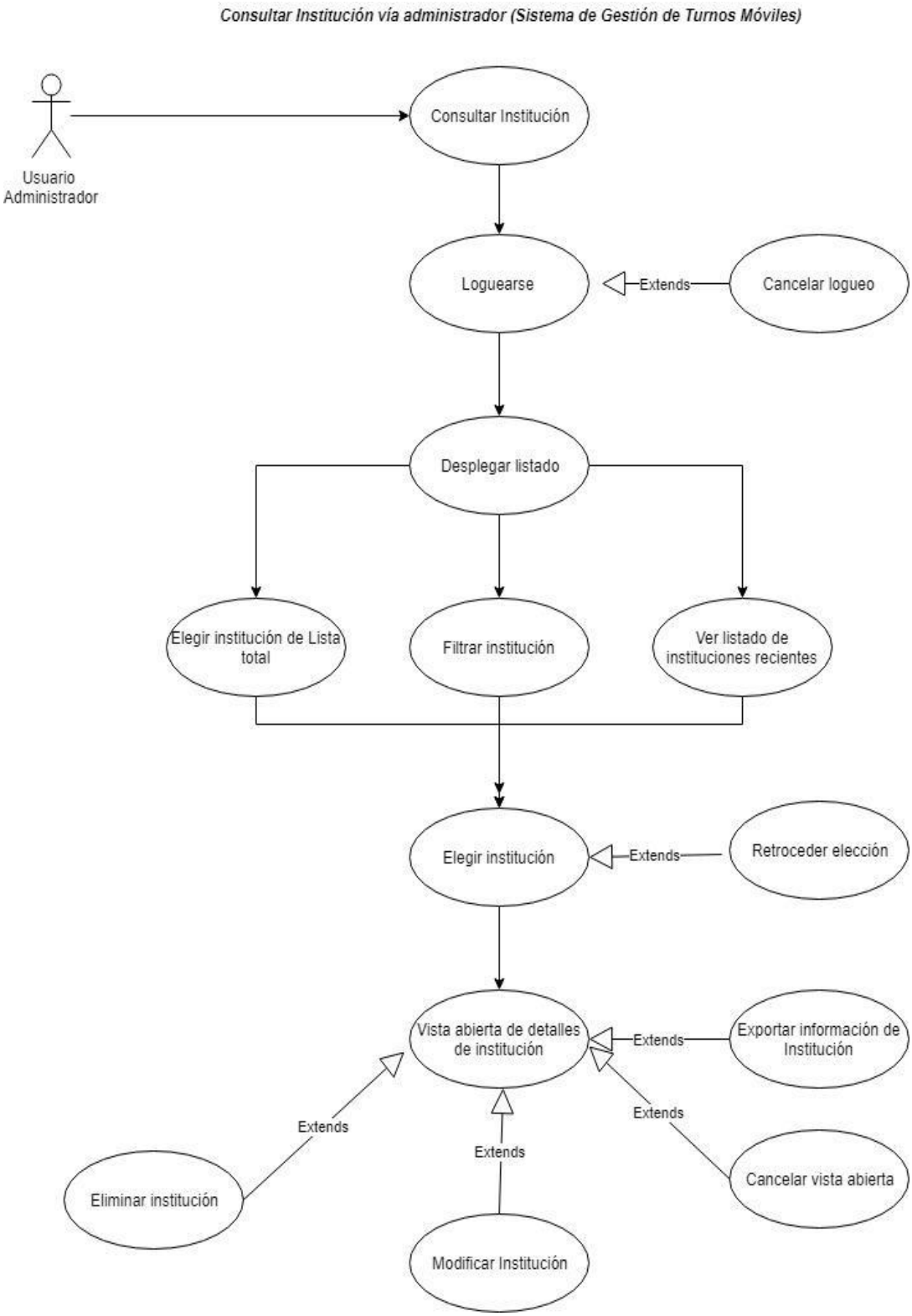


Figura 20. Caso de Uso Ver detalle de institución. Fuente: Propia

En el siguiente caso de uso, se muestran los pasos para crear un nuevo usuario desde los roles Administrador y Usuario Móvil.

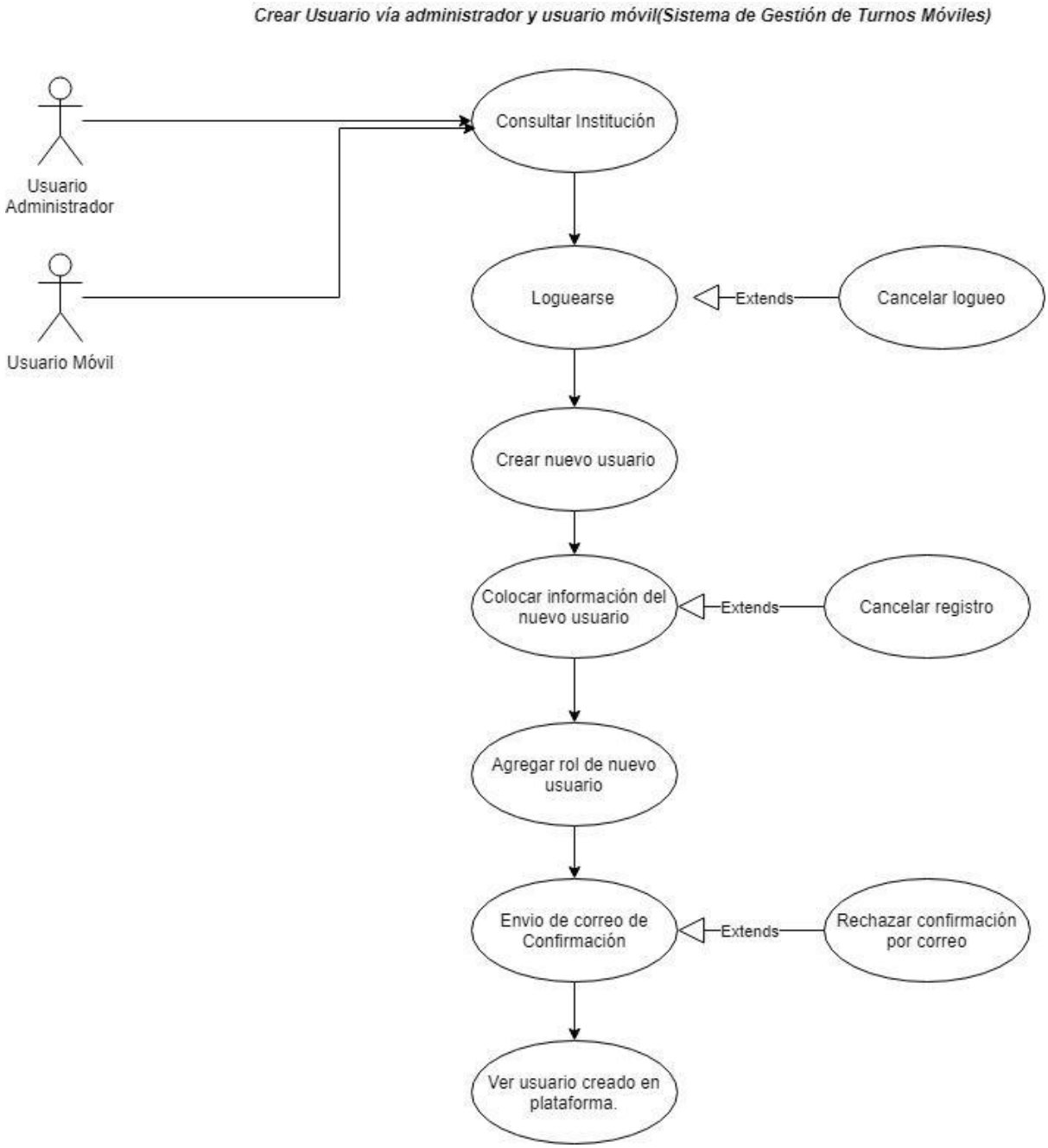


Figura 21. Caso de Uso crear nuevo usuario. Fuente: Propia

En el siguiente caso de uso se puede apreciar el proceso para el actor Administrador de consultar listado de usuarios.

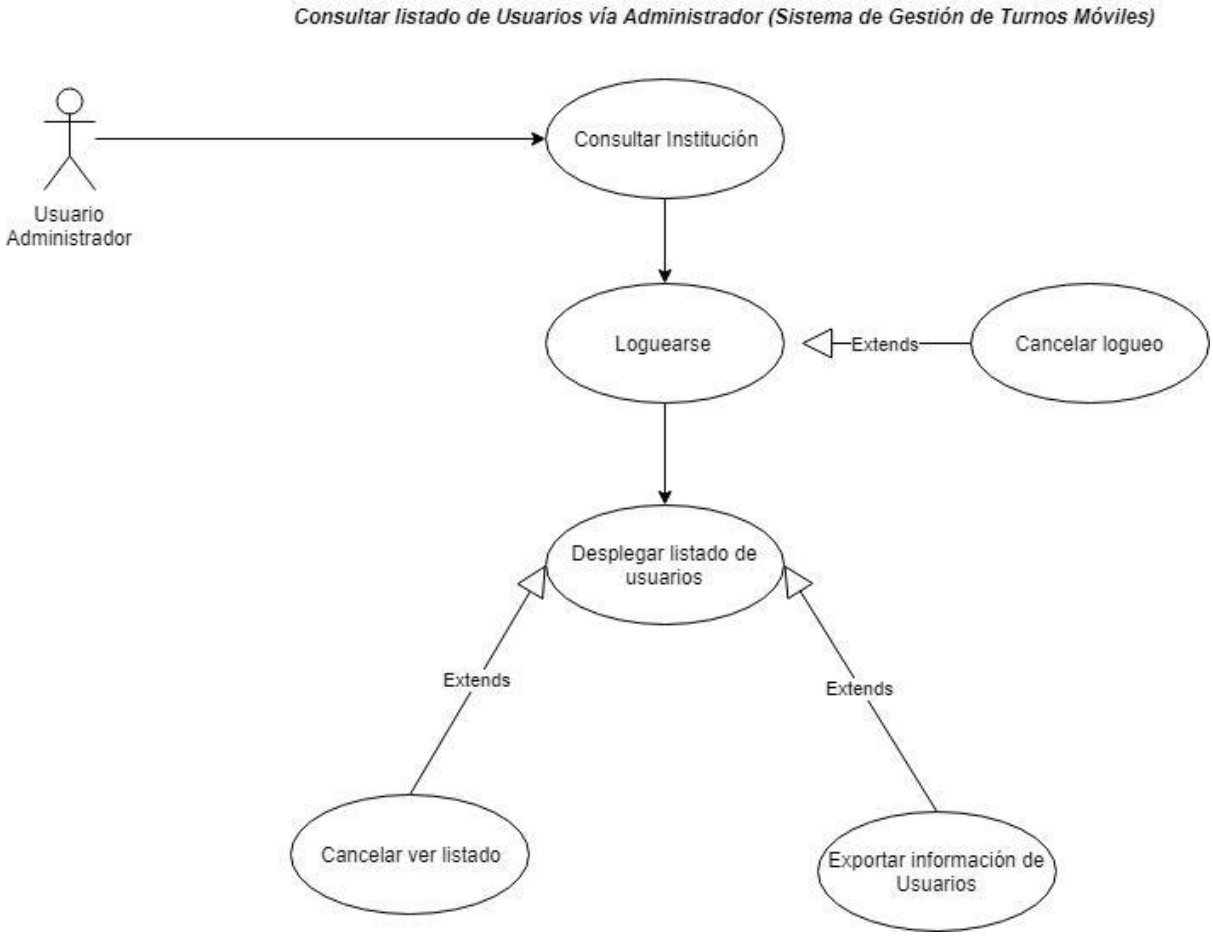


Figura 22. Caso de Uso Consultar listado usuarios vía administrador.
Fuente: Propia

En el siguiente UML se detalla el proceso de consultar un usuario en específico vía actor Administrador.

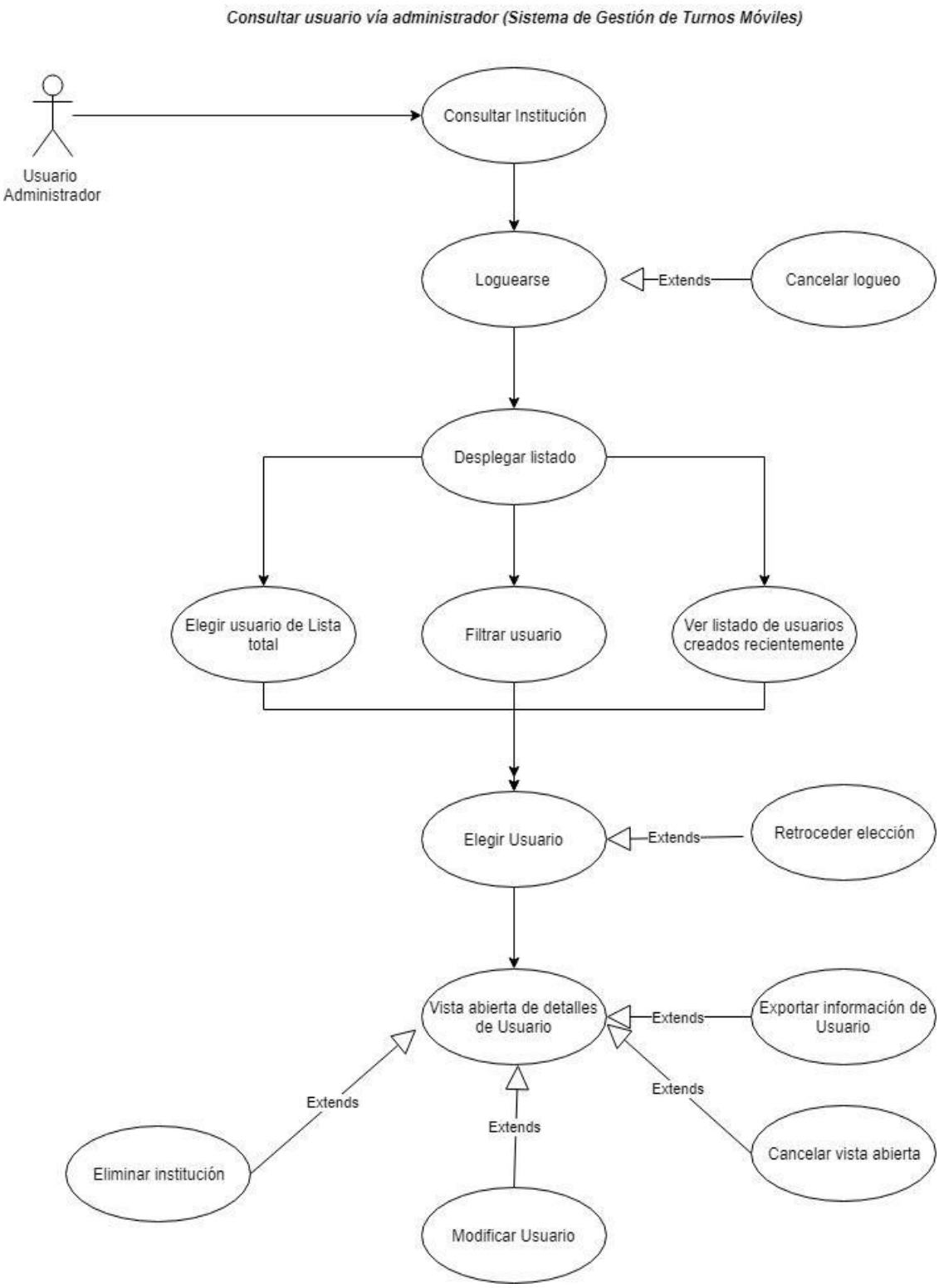


Figura 23. Caso de uso Consultar detalle usuario administrador. Fuente: Propia

En el siguiente UML se detalla la creación de una nueva institución.

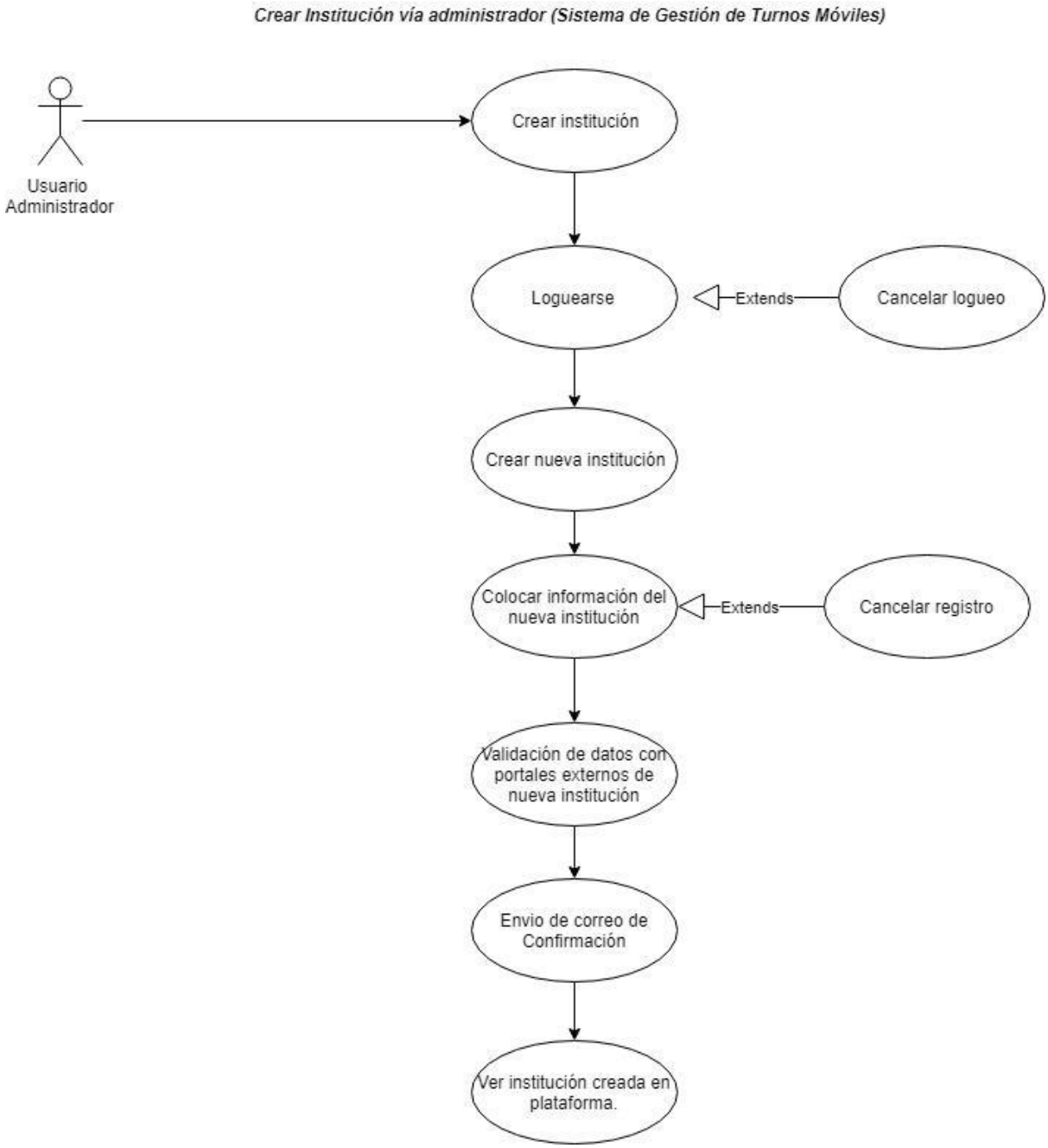


Figura 24. Caso de Uso Creación de nueva institución. Fuente: Propia.

En el siguiente UML, se aprecia el proceso de habilitar/deshabilitar institución a través de Rol Administrador:

Figura 25. Caso de Uso Habilitar/Deshabilitar institución. Fuente: Propia

En el siguiente UML, el proceso de seleccionar y rechazar turno a través del usuario móvil:

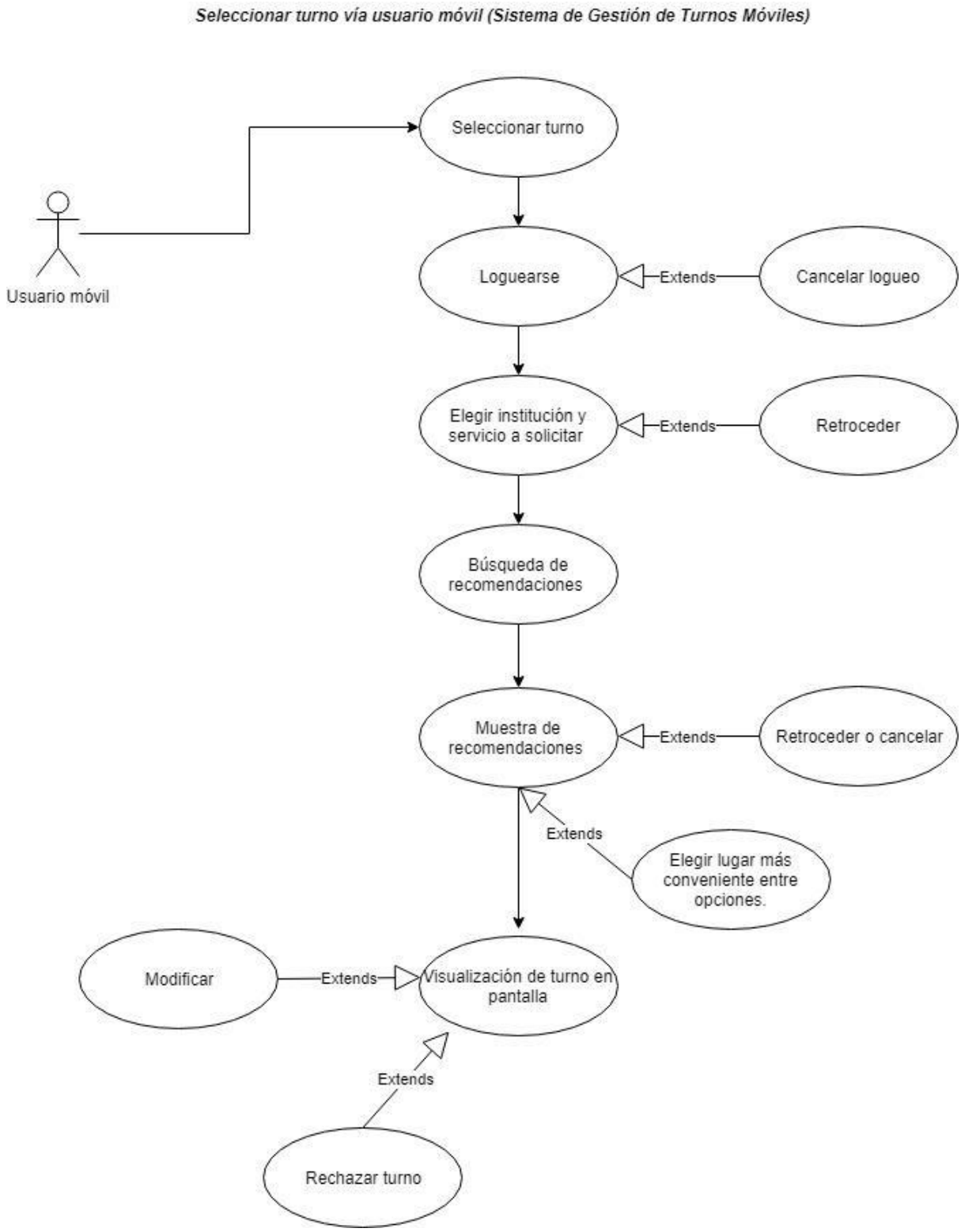


Figura 26. Caso de uso Seleccionar/ Rechazar turno. Fuente: Propia

3.3.3.2 Diagramas de Secuencia:

Los diagramas de secuencia representan el flujo a nivel de clases de los procesos o casos de usos. El flujo empieza cuando se realiza una petición o HTTP request al servicio REST publicado y finaliza con la respuesta del mismo.

Los diagramas de secuencia para soportar los requerimientos funcionales del Sistema de Turnos Móvil son los siguientes:

Consultar Institución:

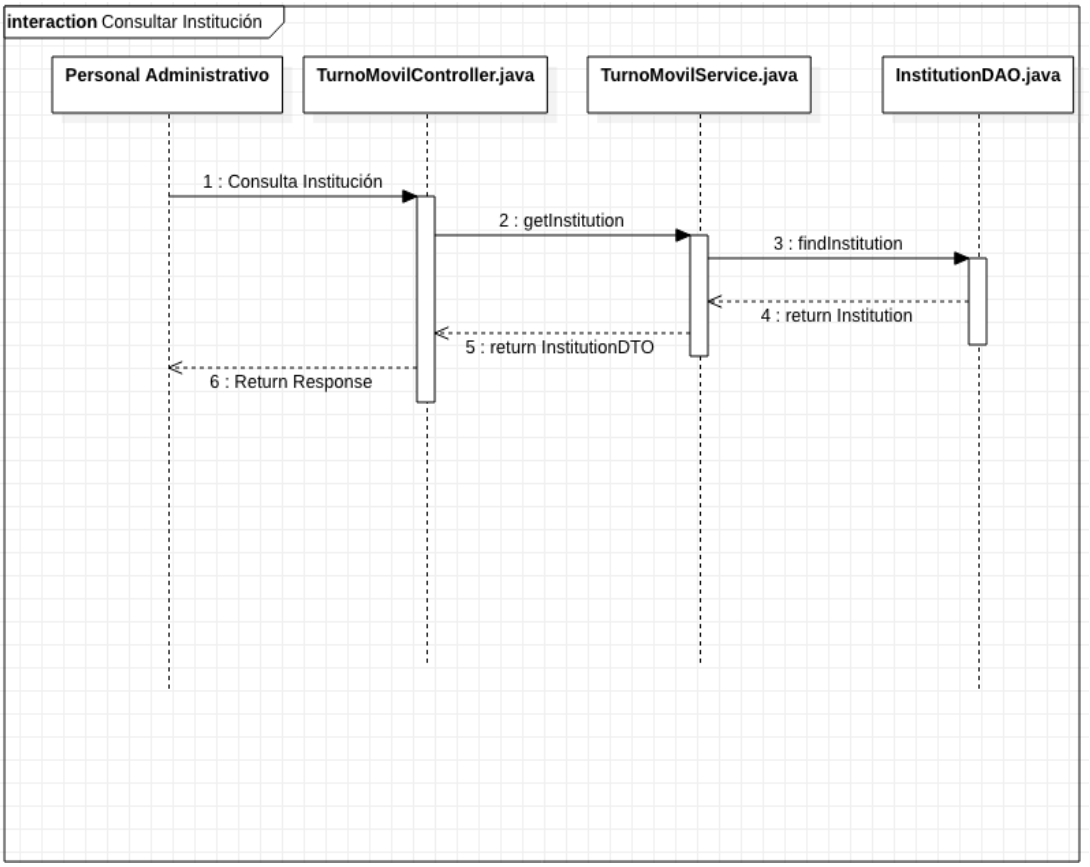


Figura 27. Diagrama de secuencia para el flujo de consultar una institución desde el portal. Fuente: Propia.

La figura muestra el flujo que tomara a nivel de back-end el proceso de consultar una institución desde el portal administrativo. El cual está descrito más abajo:

A. El flujo empiezo cuando un usuario administrativo desde el portal consulta una institución utilizando el criterio de nombre.
B. El servicio REST es llamado en la clase TurnoMovilController.java utilizando el método de tipo GET: getInstitution(Long institutionID). Esté publicado en el endpoint: https://${host}:{port}/turnomovil/api/rest/institutions/1/ , Siendo 1 el ID o identificador de la institución consultada.
C. Luego, TurnoMovilController.java llama el método getInstitution(Long institutionID) de la clase TurnoMovilService.java el cual implementa la lógica necesaria para llenar las propiedades de InstitutionDTO.java, llamando al método findInstitution(Long institutionID) de la clase InstitutionDAO.java.
D. Finalmente, la clase InstitutionDAO.java retorna un objeto Institution.java a TurnoMovilService.java y este a su vez, retorna un objeto InstitutionDTO.java a TurnoMovilController.java para retornar la respuesta con la información de la institución consultada.

Consultar todas las Instituciones:

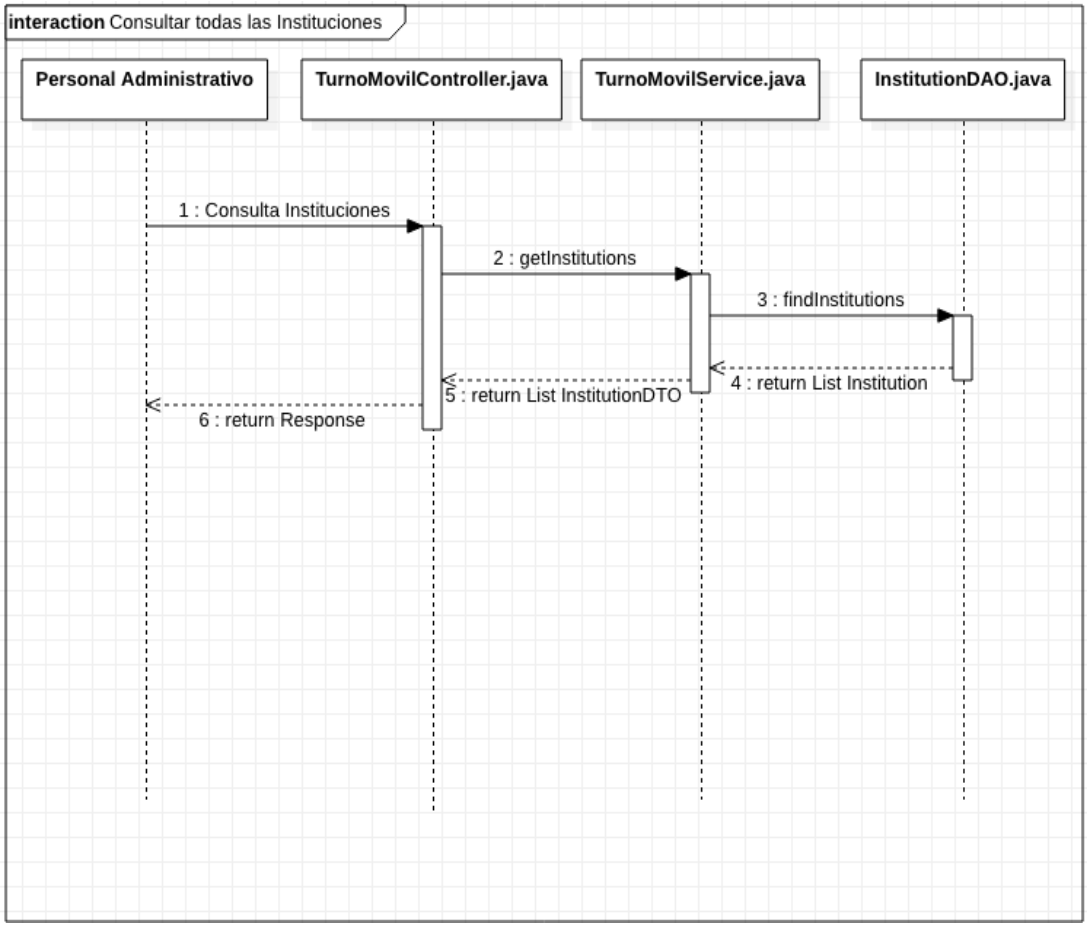


Figura 28. Diagrama de secuencia para el flujo de consultar todas las instituciones desde el portal. Fuente: Propia

La figura muestra el flujo que tomara a nivel de back-end el proceso de consultar todas las instituciones desde el portal administrativo. El cual está descrito más abajo:

A. El flujo empiezo cuando un usuario administrativo desde el portal consulta todas las instituciones.
B. El servicio REST es llamado en la clase TurnoMovilController.java utilizando el método de tipo GET: getInstitutions (). Esté publicado

en el endpoint: https://${host}:{port}/turnomovil/api/rest/institutions/.

C. Luego, TurnoMovilController.java llama el método getInstitutions () de la clase TurnoMovilService.java el cual implementa la lógica necesaria para llenar las propiedades de InstitutionDTO.java, llamando al método findInstitutions() de la clase InstitutionDAO.java.

D. Finalmente, la clase InstitutionDAO.java retorna un objeto Institution.java a TurnoMovilService.java y este a su vez, retorna un objeto InstitutionDTO.java a TurnoMovilController.java para retornar la respuesta con la información de la institución consultada.

Crear Institución

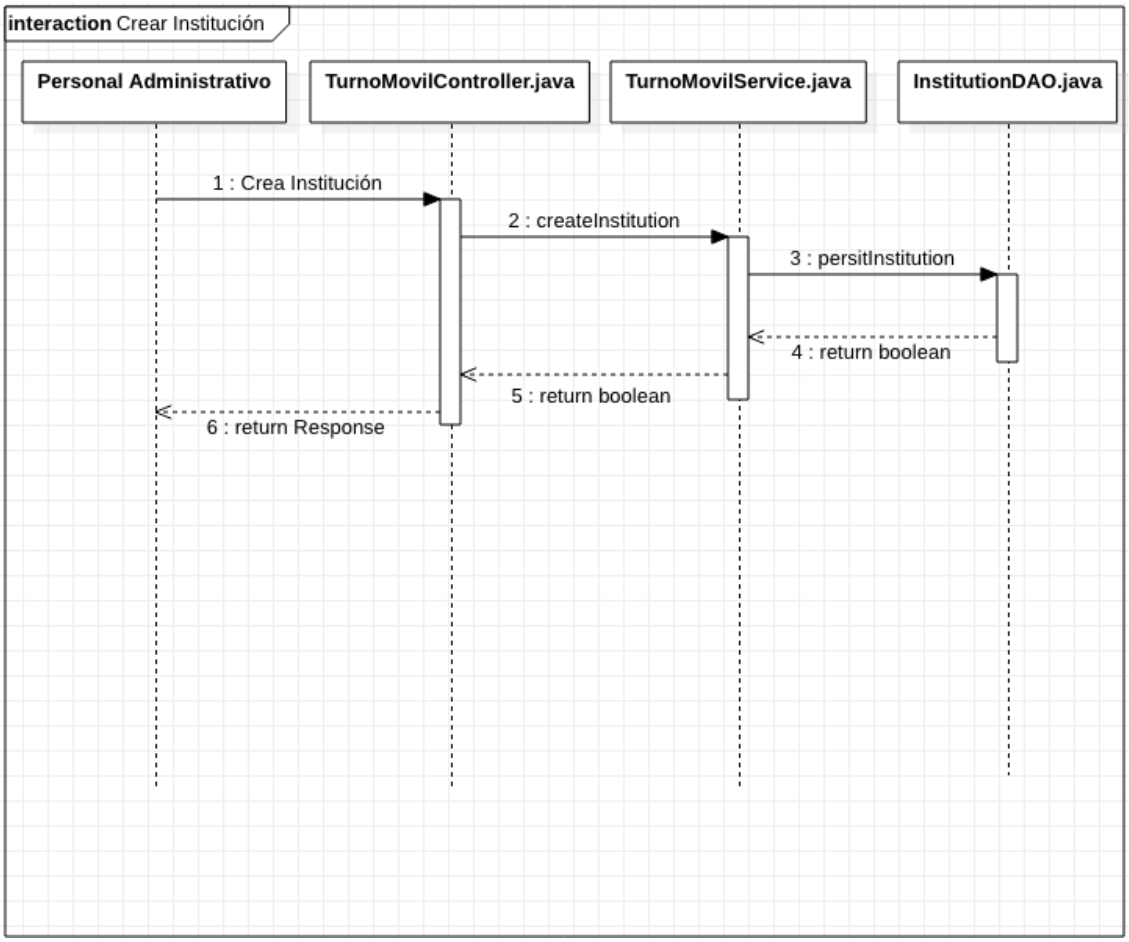


Figura 29. Diagrama de secuencia para el flujo de crear una institución desde el portal. Fuente: Propia

La figura muestra el flujo que tomara a nivel de back-end el proceso de crear una institución desde el portal administrativo. El cual está descrito más abajo:

A. El flujo empiezo cuando un usuario administrativo desde el portal digita la información de la institución y selecciona guardar.
B. El servicio REST es llamado en la clase TurnoMovilController.java utilizando el método de tipo POST: createInstitution (Institution institution), siendo Institution el objeto que recibe en el cuerpo del

mensaje. Esté publicado en el endpoint: https://${host}:{port}/turnomovil/api/rest/institutions/institution/.

C. Luego, TurnoMovilController.java llama el método createInstitution (Institution institution) de la clase TurnoMovilService.java el cual implementa la lógica necesaria para popular las propiedades de Institution.java, llamando al método persistInstitution(Institution institution) de la clase InstitutionDAO.java.

D. Finalmente, la clase InstitutionDAO.java retorna un boolean siendo TRUE si el registro es exitoso, de lo contrario FALSE. El boolean lo recibe TurnoMovilService.java y este a su vez, retorna el mismo boolean a TurnoMovilController.java para retornar la respuesta con la confirmación.

Habilitar/Deshabilitar Institución

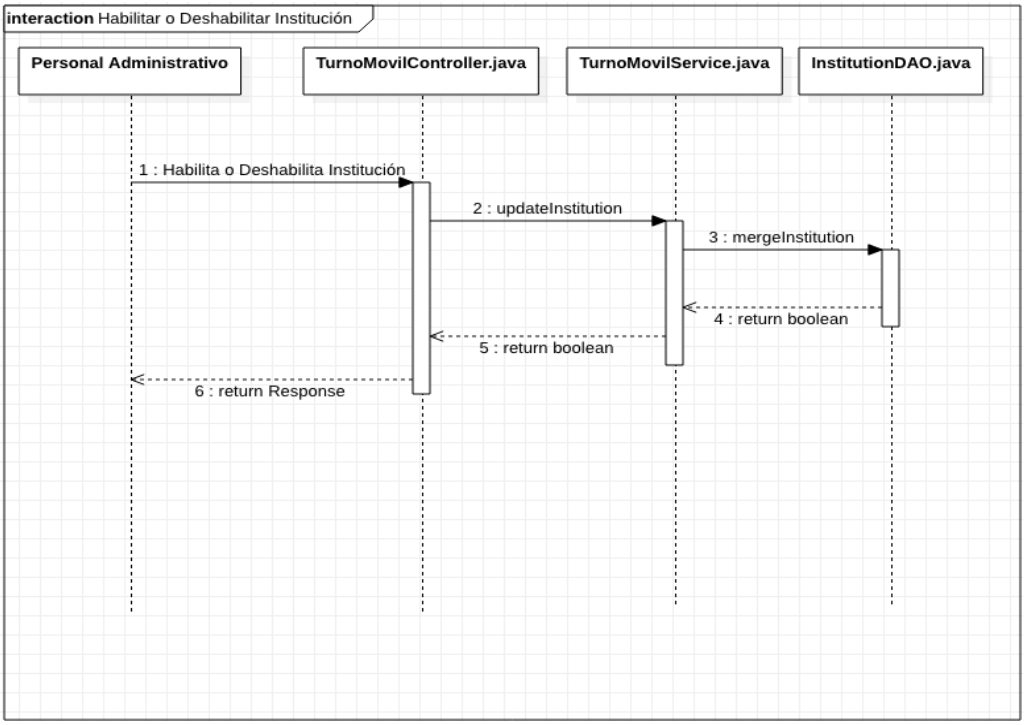


Figura 30. Diagrama de secuencia para el flujo de habilitar o deshabilitar una institución desde el portal. Fuente: Propia

La figura muestra el flujo que tomara a nivel de back-end el proceso de habilitar o deshabilitar una institución desde el portal administrativo. El cual está descrito más abajo:

A. El flujo empiezo cuando un usuario administrativo desde el portal selecciona la opción de habilitar o deshabilitar la institución y selecciona guardar.
B. El servicio REST es llamado en la clase TurnoMovilController.java utilizando el método de tipo PUT: updateInstitution(Institution institution), siendo Institution el objeto que recibe en el cuerpo del mensaje. Esté publicado en el endpoint: https://${host}:{port}/turnomovil/api/rest/institutions/institution/.

C. Luego, TurnoMovilController.java llama el método updateInstitution(Institution institution) de la clase TurnoMovilService.java el cual implementa la lógica necesaria para popular las propiedades de Institution.java, llamando al método mergeInstitution(Institution institution) de la clase InstitutionDAO.java.

D. Finalmente, la clase InstitutionDAO.java retorna un boolean siendo TRUE si la actualización es exitosa, de lo contrario FALSE. El boolean lo recibe TurnoMovilService.java y este a su vez, retorna el mismo boolean a TurnoMovilController.java para retornar la respuesta con la confirmación.

Consultar Usuario

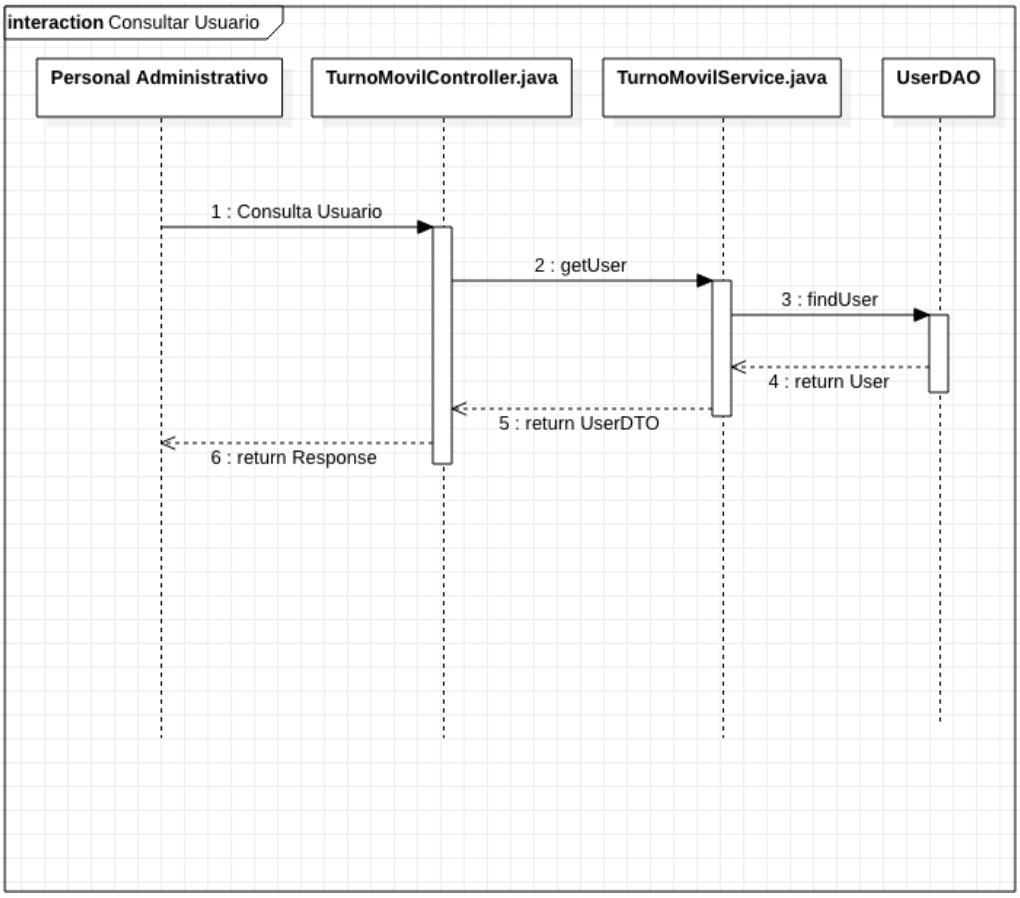


Figura 31. Diagrama de secuencia para el flujo de consultar un usuario de la aplicación móvil desde el portal. Fuente: Propia.

La figura muestra el flujo que tomara a nivel de back-end el proceso de consultar un usuario desde el portal administrativo. El cual está descrito más abajo:

A. El flujo empiezo cuando un usuario administrativo desde el portal consulta un usuario de la aplicación móvil utilizando el criterio de nombre o documento de identidad.
B. El servicio REST es llamado en la clase TurnoMovilController.java utilizando el método de tipo GET: getUser(Long userID). Esté publicado en el endpoint:

https://${host}:{port}/turnomovil/api/rest/users/1/ , Siendo 1 el ID o identificador del usuario consultado.

C. Luego, TurnoMovilController.java llama el método getUser(Long userID) de la clase TurnoMovilService.java el cual implementa la lógica necesaria para llenar las propiedades de UserDTO.java, llamando al método findUser(Long userID) de la clase UserDAO.java.

D. Finalmente, la clase UserDAO.java retorna un objeto User.java a TurnoMovilService.java y este a su vez, retorna un objeto UserDTO.java a TurnoMovilController.java para retornar la respuesta con la información del usuario consultado.

Consultar Usuarios

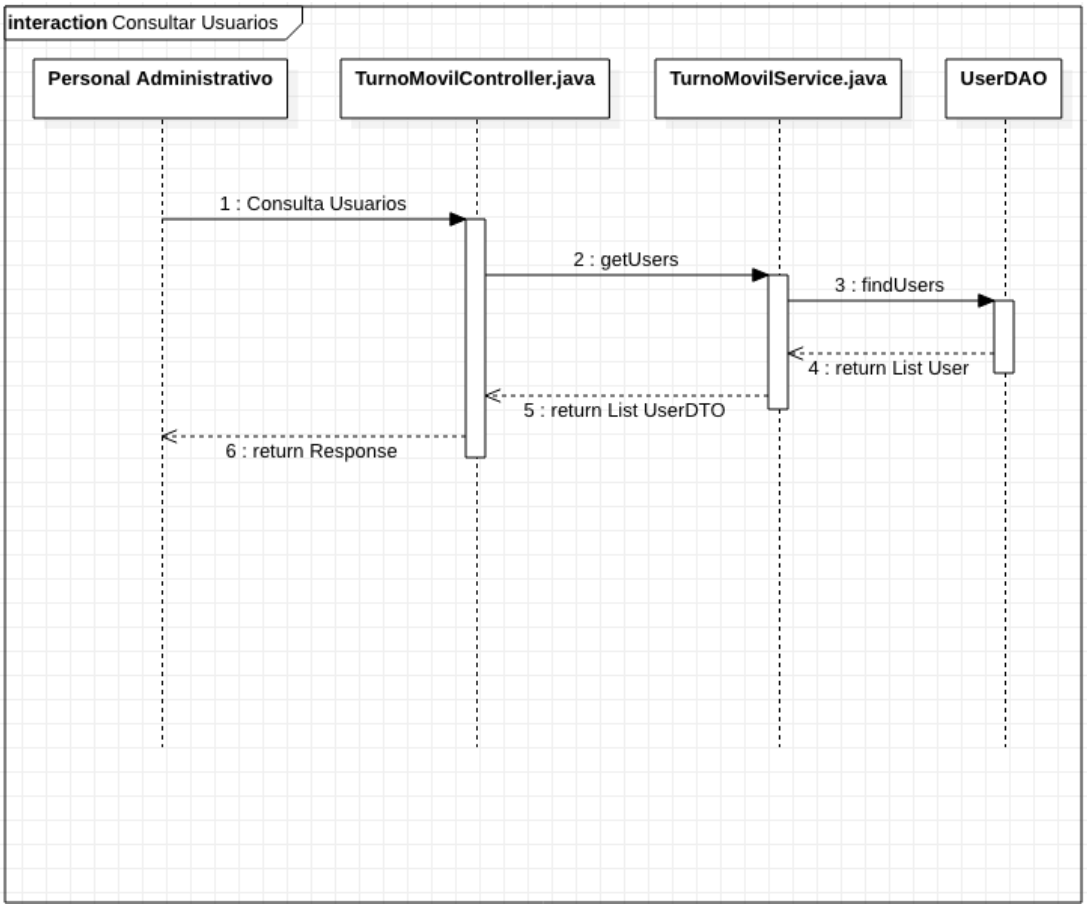


Figura 32. Diagrama de secuencia para el flujo de consultar todos los usuarios de la aplicación móvil desde el portal. Fuente: Propia.

La figura muestra el flujo que tomara a nivel de back-end el proceso de consultar todos los usuarios desde el portal administrativo. El cual está descrito más abajo:

A. El flujo empiezo cuando un usuario administrativo desde el portal consulta todos los usuarios de la aplicación móvil.
B. El servicio REST es llamado en la clase TurnoMovilController.java utilizando el método de tipo GET: getUsers(). Esté publicado en el endpoint: https://${host}:{port}/turnomovil/api/rest/users/.

C. Luego, TurnoMovilController.java llama el método getUsers() de la clase TurnoMovilService.java el cual implementa la lógica necesaria para llenar las propiedades de UserDTO.java, llamando al método findUsers() de la clase UserDAO.java.

D. Finalmente, la clase UserDAO.java retorna un objeto User.java a TurnoMovilService.java y este a su vez, retorna un objeto UserDTO.java a TurnoMovilController.java para retornar la respuesta con la información del usuario consultado.

Registrar Cuenta Usuario

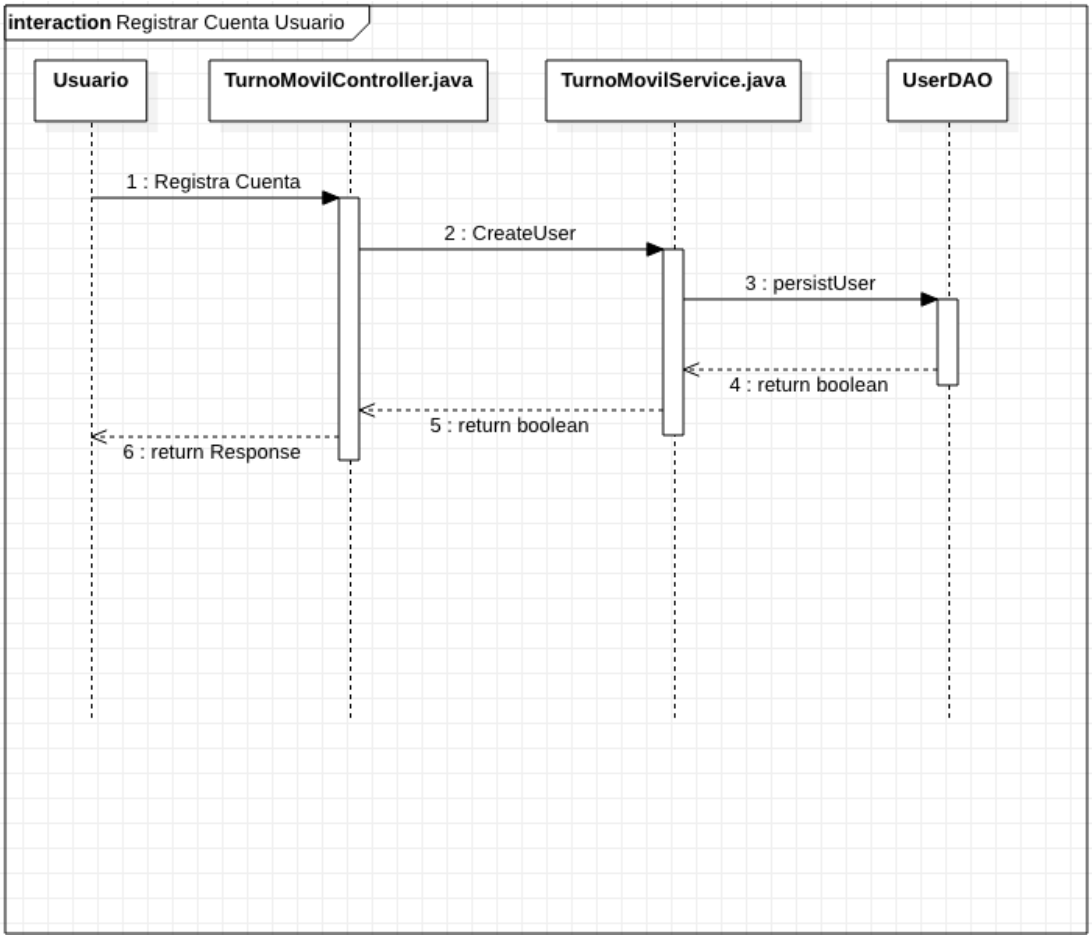


Figura 33. Diagrama de secuencia para el flujo de registro de usuario desde el móvil. Fuente: Propia

La figura muestra el flujo que tomara a nivel de back-end el proceso de registro de usuario desde el móvil. El cual está descrito más abajo:

A. El flujo empiezo cuando un usuario desde el móvil digita sus datos personales y selecciona guardar.
B. El servicio REST es llamado en la clase TurnoMovilController.java utilizando el método de tipo POST: createUser(User user), siendo User el objeto que recibe en el cuerpo del mensaje. Esté publicado en el endpoint: https://${host}:{port}/turnomovil/api/rest/users/user/.
C. Luego, TurnoMovilController.java llama el método createUser(User user) de la clase TurnoMovilService.java el cual implementa la lógica necesaria para popular las propiedades de User.java, llamando al método persistUser(User user) de la clase UserDAO.java.
D. Finalmente, la clase UserDAO.java retorna un boolean siendo TRUE si el registro es exitoso, de lo contrario FALSE. El boolean lo recibe TurnoMovilService.java y este a su vez, retorna el boolean a TurnoMovilController.java para retornar la respuesta con la confirmación.

Eliminar Cuenta Usuario

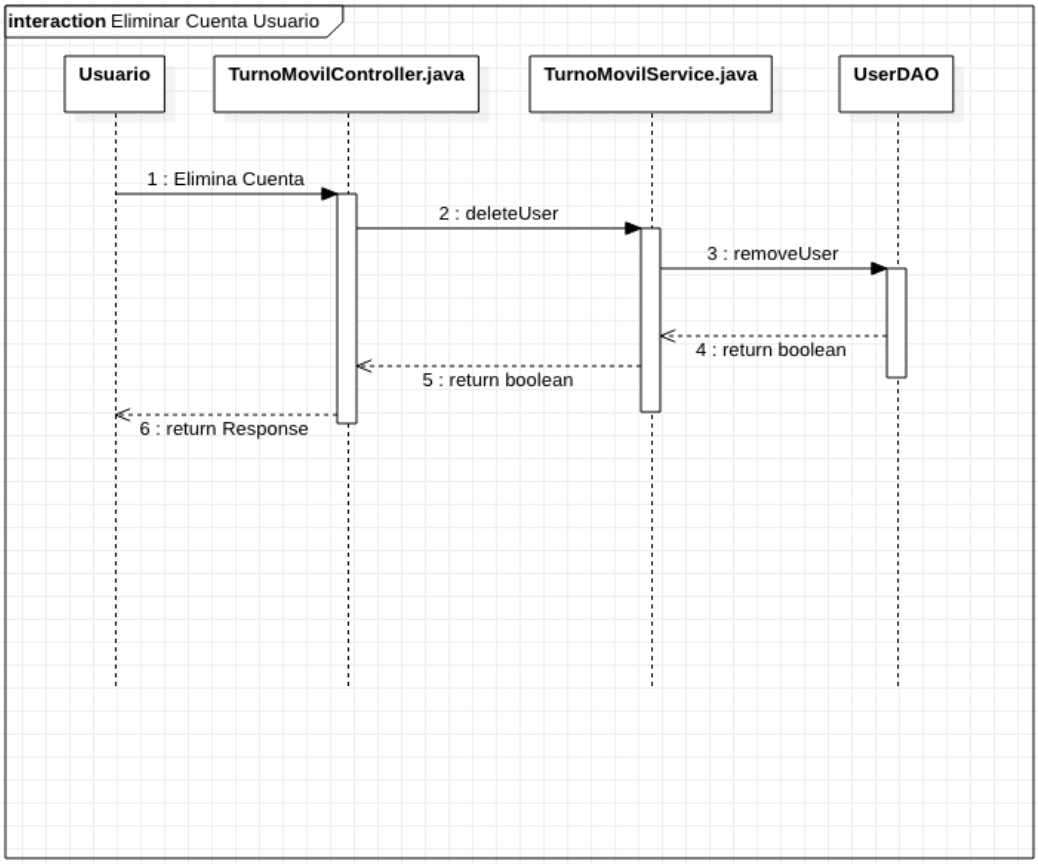


Figura 34. Diagrama de secuencia para el flujo de eliminar usuario desde el móvil. Fuente: Propia

La figura muestra el flujo que tomara a nivel de back-end el proceso de eliminar usuario desde el móvil. El cual está descrito más abajo:

A. El flujo empiezo cuando un usuario desde el móvil, selecciona eliminar cuenta.
B. El servicio REST es llamado en la clase TurnoMovilController.java utilizando el método de tipo DELETE: deleteUser(Long id). Esté publicado en el endpoint: https://${host}:{port}/turnomovil/api/rest/users/1/ , Siendo 1 el ID o identificador del usuario a eliminar.

C. Luego, TurnoMovilController.java llama el método deleteUser(Long id) de la clase TurnoMovilService.java el cual implementa la lógica necesaria para popular las propiedades de User.java, llamando al método removeUser(Long id) de la clase UserDAO.java.
D. Finalmente, la clase UserDAO.java retorna un boolean siendo TRUE si el registro es exitoso, de lo contrario FALSE. El boolean lo recibe TurnoMovilService.java y este a su vez, retorna el boolean a TurnoMovilController.java para retornar la respuesta con la confirmación.

Consultar Instituciones en la Aplicación Móvil

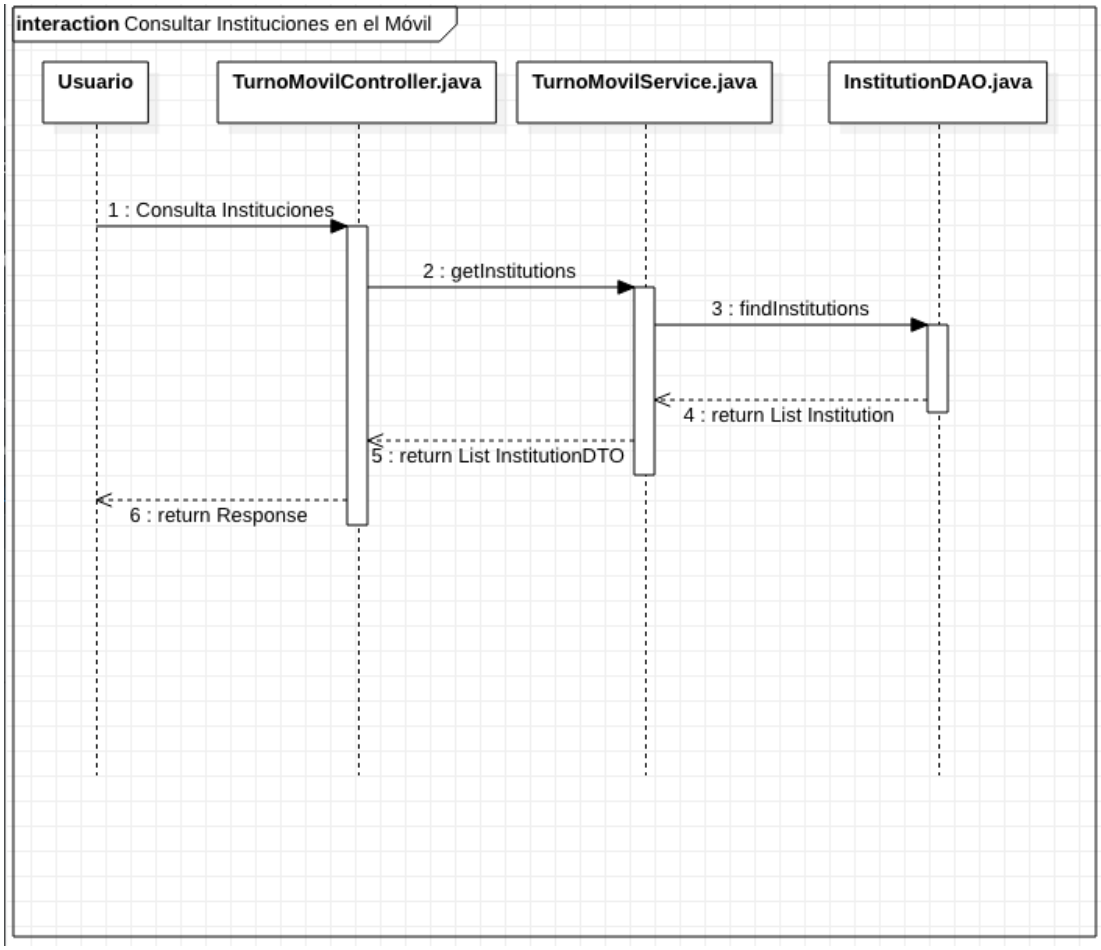


Figura 35. Diagrama de secuencia para el flujo de consultar todas las instituciones desde el móvil. Fuente: Propia

La figura muestra el flujo que tomara a nivel de back-end el proceso de consultar todas las instituciones desde el móvil. El cual está descrito más abajo:

A. El flujo empiezo cuando un usuario desde el móvil consulta todas las instituciones.
B. El servicio REST es llamado en la clase TurnoMovilController.java utilizando el método de tipo GET: getInstitutions(). Esté publicado

en el endpoint: https://${host}:{port}/turnomovil/api/rest/institutions/.

C. Luego, TurnoMovilController.java llama el método getInstitutions() de la clase TurnoMovilService.java el cual implementa la lógica necesaria para llenar las propiedades de InstitutionDTO.java, llamando al método findInstitutions() de la clase InstitutionDAO.java.

D. Finalmente, la clase InstitutionDAO.java retorna un objeto Institution.java a TurnoMovilService.java y este a su vez, retorna un objeto InstitutionDTO.java a TurnoMovilController.java para retornar la respuesta con la información de la institución consultada.

Consultar Institución en la Aplicación Móvil

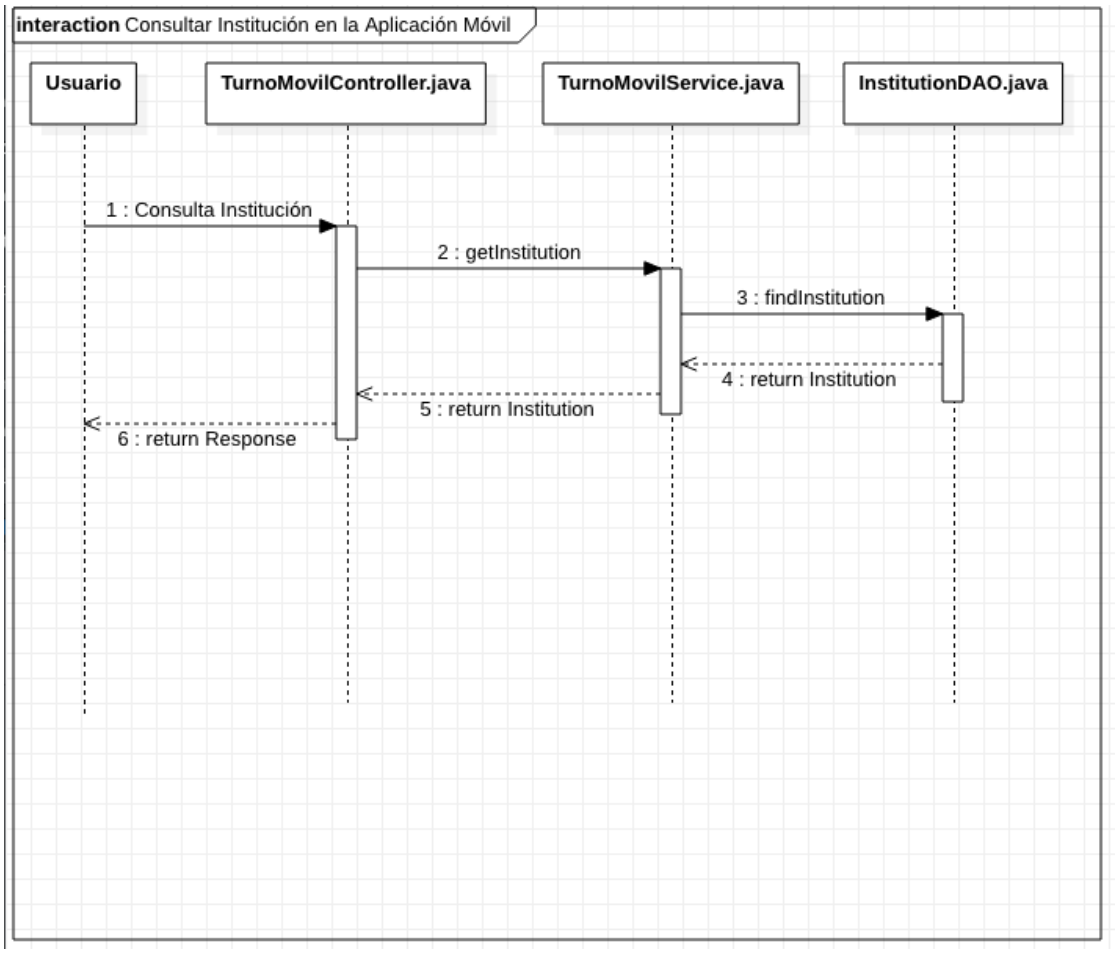


Figura 36. Diagrama de secuencia para el flujo de consultar una institución desde el móvil. Fuente: Propia.

La figura muestra el flujo que tomara a nivel de back-end el proceso de consultar una institución desde el móvil. El cual está descrito más abajo:

A. El flujo empiezo cuando un usuario desde el móvil consulta una institución utilizando el criterio de nombre.
B. El servicio REST es llamado en la clase TurnoMovilController.java utilizando el método de tipo GET: getInstitution(Long institutionID).

Esté publicado en el endpoint: https://${host}:{port}/turnomovil/api/rest/institutions/1/ , Siendo 1 el ID o identificador de la institución consultada.

C. Luego, TurnoMovilController.java llama el método getInstitution(Long institutionID) de la clase TurnoMovilService.java el cual implementa la lógica necesaria para llenar las propiedades de InstitutionDTO.java, llamando al método findInstitution(Long institutionID) de la clase InstitutionDAO.java.

D. Finalmente, la clase InstitutionDAO.java retorna un objeto Institution.java a TurnoMovilService.java y este a su vez, retorna un objeto InstitutionDTO.java a TurnoMovilController.java para retornar la respuesta con la información de la institución consultada.

Seleccionar/Rechazar Turno

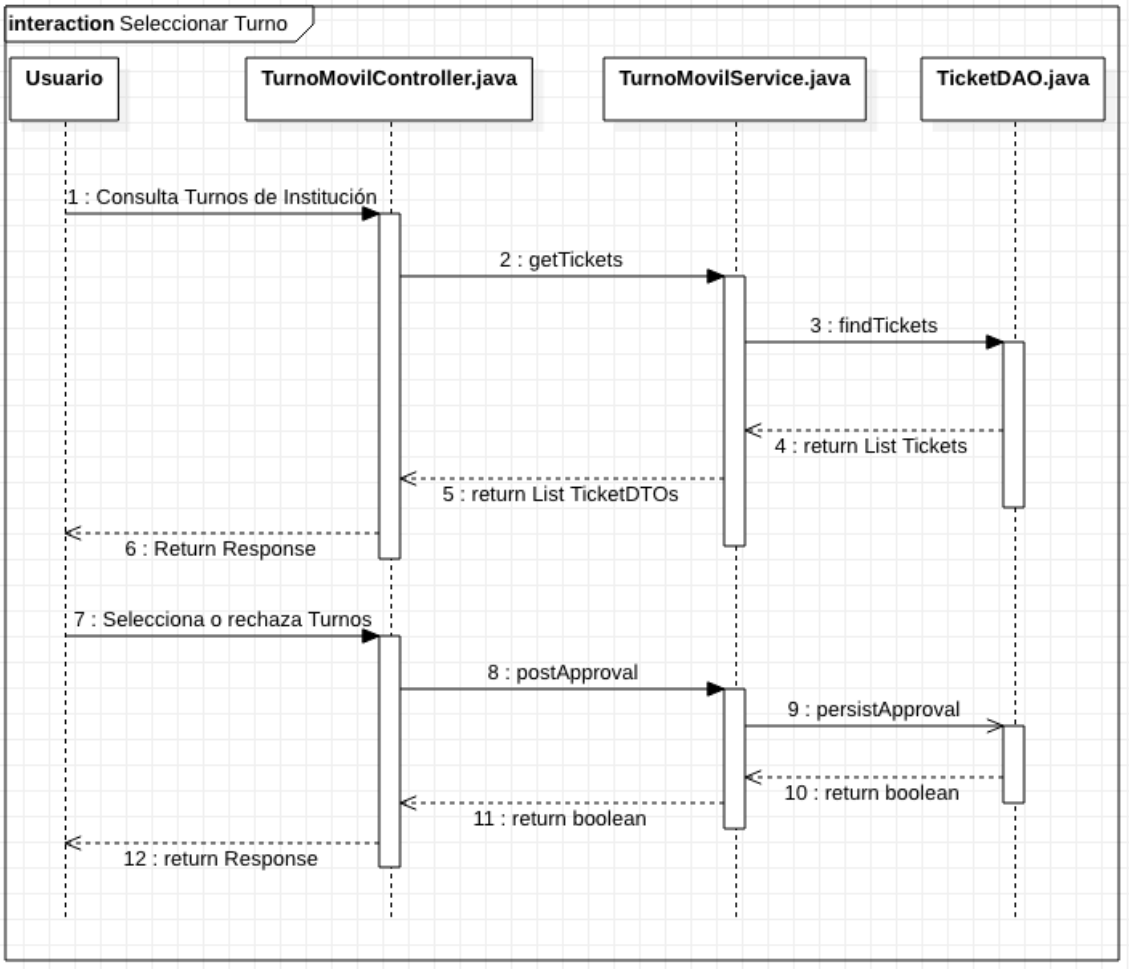


Figura 37. Diagrama de secuencia para el flujo de seleccionar o rechazar turnos de institución desde el móvil. Fuente: Propia.

La figura muestra el flujo que tomara a nivel de back-end el proceso de seleccionar o rechazar turnos de institución desde el móvil. El cual está descrito más abajo:

A. El flujo empiezo cuando un usuario desde el móvil consulta los turnos de una institución utilizando el criterio de nombre.
B. El servicio REST es llamado en la clase TurnoMovilController.java utilizando el método de tipo GET: getTickets(Long institutionID).

Esté publicado en el endpoint: https://${host}:{port}/turnomovil/api/rest/institutions/1/tickets/ , Siendo 1 el ID o identificador de la institución en la que se tomará el turno.

C. Luego, TurnoMovilController.java llama el método getInstitution(Long institutionID) de la clase TurnoMovilService.java el cual implementa la lógica necesaria para llenar las propiedades de TicketDTO.java, llamando al método findTickets(Long institutionID) de la clase TicketDAO.java.

D. La clase TicketDAO.java retorna un objeto Ticket.java a TurnoMovilService.java y este a su vez, retorna un objeto TicketDTO.java a TurnoMovilController.java para retornar la respuesta con la información del turno consultado.

E. Luego del usuario recibir del sistema 3 recomendaciones en base a disponibilidad y distancia. Esté al seleccionar o rechazar las opciones el servicio REST es llamado en la clase TurnoMovilController.java utilizando el método de tipo POST: postApproval(Long institutionID, boolean isApproval). Esté publicado en el endpoint: https://${host}:{port}/turnomovil/api/rest/institutions/1/tickets/true/ , Siendo 1 el ID o identificador de la institución en la que se tomará el turno y TRUE la confirmación de la selección.

F. Luego, de ser negativa la selección termina el flujo, de lo contrario, TurnoMovilController.java llama el método persistApproval(Long institutionID) de la clase TurnoMovilService.java el cual implementa la lógica necesaria para popular las propiedades de Ticket.java, llamando al método persistTicket(Ticket ticket) de la clase TicketDAO.java.

G. Finalmente, la clase TicketDAO.java retorna un boolean siendo TRUE si el registro es exitoso, de lo contrario FALSE. El boolean

lo recibe TurnoMovilService.java y este a su vez, retorna el boolean a TurnoMovilController.java para retornar la respuesta con la confirmación.

3.4 Arquitecturas

La arquitectura son el conjunto de patrones que definen como está diseñado un sistema y bajo qué está soportado. Este a su vez está conformado o definido por un conjunto de requisitos y restricciones. Siendo los requisitos los elementos necesarios para el funcionamiento del sistema, a su vez, las restricciones son aquellas limitantes tecnológicas del sistema, en este caso dado la naturaleza del Sistema de Turnos Móvil, estará limitado a una aplicación móvil y un portal web.

3.4.1 Arquitectura App Móvil

Existen otras plataformas existentes como lo son Google y Azure de la empresa de Microsoft. Ambas son plataformas con la que se pueden contar con servicios similares a Amazon Web Services, pero lo que la hace más rentable para el Sistema de Gestión de Turnos Móvil es su alta disponibilidad y rápido

rendimiento en las peticiones que maneja de forma simultánea.

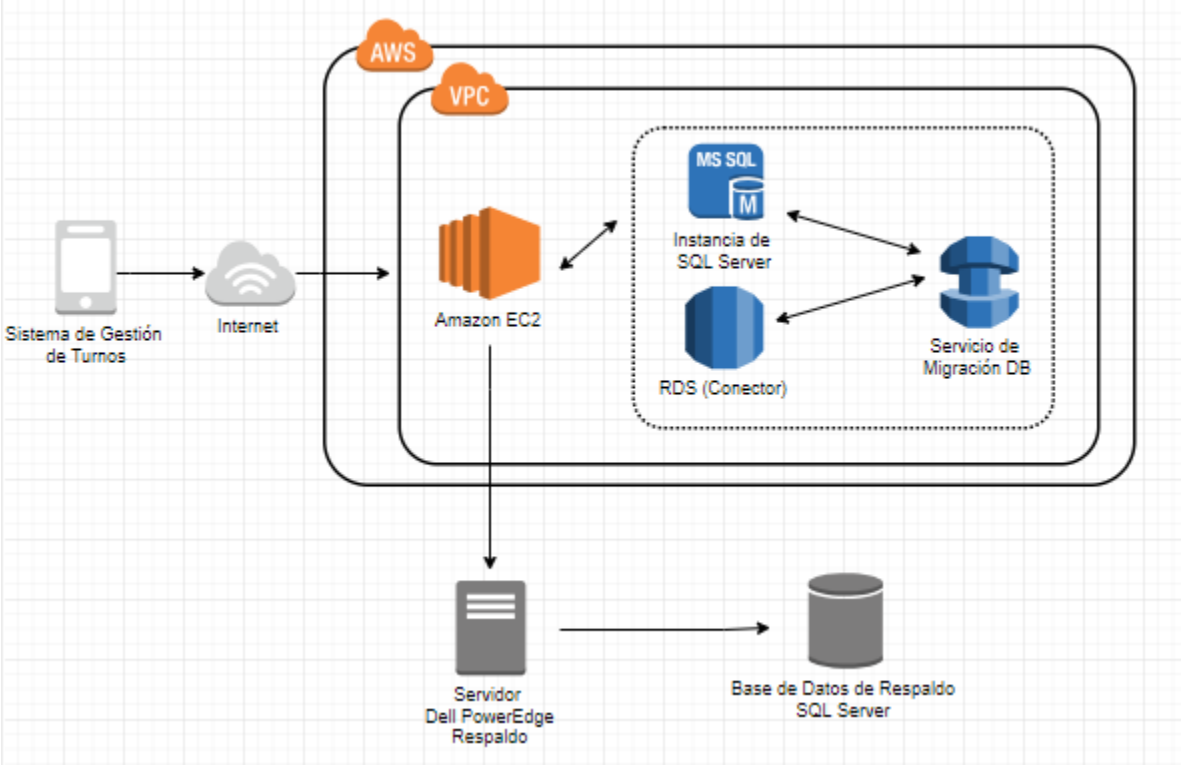


Figura 38. Arquitectura App Móvil. Fuente: Propia

3.4.2 Arquitectura de la Base de Datos

En el presente diagrama de arquitectura de la base del Sistema de Gestión de Turnos Móvil, se implementa la arquitectura de AWS (Amazon Web Services), donde se hace uso de la nube y donde se alojará la información que el cliente guarda en su dispositivo móvil (Microsoft SQL Server) y donde ésta mantiene sincronización constante.

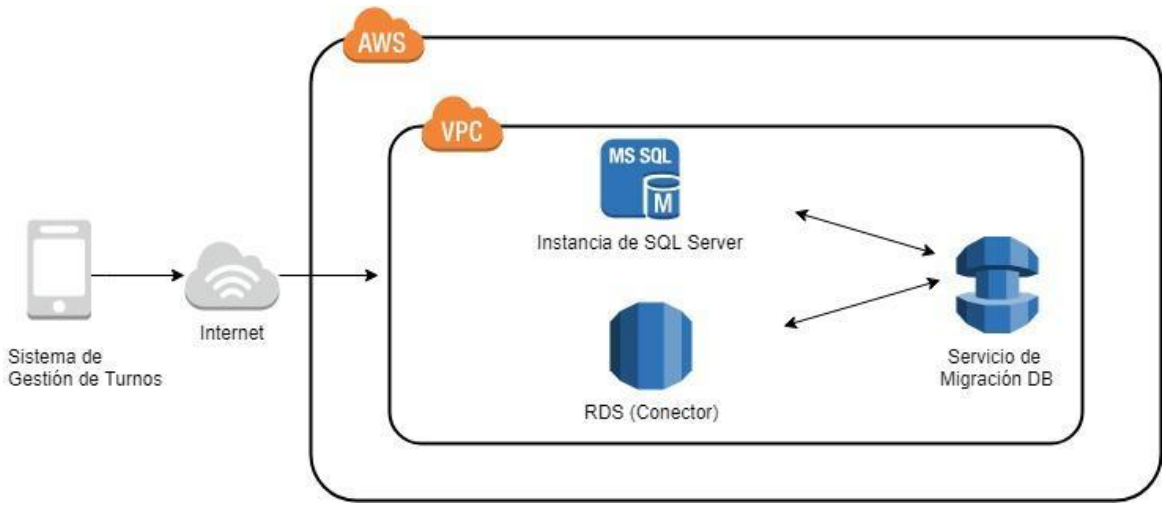


Figura 39. Arquitectura Base de Datos. Fuente: Propia

Estructura en código de la base de datos de SQL Server:

```
USE [master]
GO
/****** Object:  Database [DR_SGT]    Script Date: 04-Apr-19 8:44:31 PM ******/
CREATE DATABASE [DR_SGT]
 CONTAINMENT = NONE
 ON  PRIMARY
( NAME = N'DR_SGT', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL13.MSSQLSERVER\MSSQL\DATA\DR_SGT.mdf' , SIZE = 8192KB , MAXSIZE =
UNLIMITED, FILEGROWTH = 65536KB )
 LOG ON
( NAME = N'DR_SGT_log', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL13.MSSQLSERVER\MSSQL\DATA\DR_SGT_log.ldf' , SIZE = 8192KB , MAXSIZE =
2048GB , FILEGROWTH = 65536KB )
GO
ALTER DATABASE [DR_SGT] SET COMPATIBILITY_LEVEL = 130
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [DR_SGT].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [DR_SGT] SET ANSI_NULL_DEFAULT OFF
GO
```

```
ALTER DATABASE [DR_SGT] SET ANSI_NULLS OFF
GO
ALTER DATABASE [DR_SGT] SET ANSI_PADDING OFF
GO
ALTER DATABASE [DR_SGT] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [DR_SGT] SET ARITHABORT OFF
GO
ALTER DATABASE [DR_SGT] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [DR_SGT] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [DR_SGT] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [DR_SGT] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [DR_SGT] SET CURSOR_DEFAULT  GLOBAL
GO
ALTER DATABASE [DR_SGT] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [DR_SGT] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [DR_SGT] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [DR_SGT] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [DR_SGT] SET  DISABLE_BROKER
GO
ALTER DATABASE [DR_SGT] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [DR_SGT] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [DR_SGT] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [DR_SGT] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [DR_SGT] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [DR_SGT] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [DR_SGT] SET HONOR_BROKER_PRIORITY OFF
```

```
GO
ALTER DATABASE [DR_SGT] SET RECOVERY FULL
GO
ALTER DATABASE [DR_SGT] SET  MULTI_USER
GO
ALTER DATABASE [DR_SGT] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [DR_SGT] SET DB_CHAINING OFF
GO
ALTER DATABASE [DR_SGT] SET FILESTREAM( NON_TRANSACTED_ACCESS = OFF )
GO
ALTER DATABASE [DR_SGT] SET TARGET_RECOVERY_TIME = 60 SECONDS
GO
ALTER DATABASE [DR_SGT] SET DELAYED_DURABILITY = DISABLED
GO
ALTER DATABASE [DR_SGT] SET QUERY_STORE = OFF
GO
USE [DR_SGT]
GO
ALTER DATABASE SCOPED CONFIGURATION SET LEGACY_CARDINALITY_ESTIMATION = OFF;
GO
ALTER DATABASE SCOPED CONFIGURATION SET MAXDOP = 0;
GO
ALTER DATABASE SCOPED CONFIGURATION SET PARAMETER_SNIFFING = ON;
GO
ALTER DATABASE SCOPED CONFIGURATION SET QUERY_OPTIMIZER_HOTFIXES = OFF;
GO
USE [DR_SGT]
GO
/****** Object:  Table [dbo].[ESTABLISHMENT]    Script Date: 04-Apr-19 8:44:32 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[ESTABLISHMENT](
	[ID] [int] NULL,
	[NAME] [varchar](50) NULL,
	[SUCURSAL_NUMBER] [int] NULL,
	[ADDRESS] [varchar](100) NULL,
	[COORDINATES] [text] NULL,
	[PICTURE] [varchar](50) NULL,
```

```
	[RESPONSIVE_PEOPLE] [varchar](50) NULL,
	[PHONE] [varchar](50) NULL,
	[FAX] [varchar](50) NULL,
	[CREATE_DATE] [date] NULL,
	[ESTABLISHMENT_ID] [int] NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/****** Object:  Table [dbo].[TURN_TYPE]    Script Date: 04-Apr-19 8:44:32 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[TURN_TYPE](
	[ID] [int] NULL,
	[TURN_TYPE_NAME] [varchar](50) NULL,
	[TYPE_STATUS] [nchar](1) NULL,
	[CREATE_DATE] [date] NULL,
	[MODIFY_DATE] [date] NULL,
	[USERSIGNID] [int] NULL
) ON [PRIMARY]
GO
/****** Object:  Table [dbo].[TURNS]    Script Date: 04-Apr-19 8:44:32 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[TURNS](
	[ID] [int] NULL,
	[USERID] [int] NULL,
	[TURN_NUMBER] [int] NULL,
	[TURN_TYPE_ID] [int] NULL,
	[COORDINATES] [varchar](50) NULL,
	[TURN_STATUS] [nchar](1) NULL,
	[ADDRESS] [text] NULL,
	[START_POINT] [text] NULL,
	[COORDINATES_START] [text] NULL,
	[GPS_STATUS] [nchar](1) NULL,
	[ALERT_BATTERY] [nchar](1) NULL,
	[USERSIGN] [int] NULL,
	[CREATEDATE] [date] NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

```
GO
/****** Object:  Table [dbo].[Users]    Script Date: 04-Apr-19 8:44:32 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Users](
	[ID] [int] NULL,
	[USERNAME] [varchar](50) NULL,
	[NAME] [varchar](50) NULL,
	[PASSWORD] [varchar](50) NULL,
	[COUNTRY] [varchar](50) NULL,
	[CITY] [varchar](50) NULL,
	[PHONE] [varchar](50) NULL,
	[MOBILE] [varchar](50) NULL,
	[IDENTIFICATION] [int] NULL,
	[ADDRESS] [varchar](100) NULL,
	[CREATE_DATE] [date] NULL
) ON [PRIMARY]
GO
USE [master]
GO
ALTER DATABASE [DR_SGT] SET  READ_WRITE
GO
```

Estructura de la base de datos espacial GIS en PostGIS externa:

```
CREATE DATABASE rev WITH OWNER = postgres ENCODING = 'UTF8'
TABLESPACE=rev_sde_dict;
ALTER DATABASE rev SET search_path="$user", public, sde;
GRANT ALL ON DATABASE rev TO public;
GRANT ALL ON DATABASE rev TO postgres;
```

```
\connect rev CREATE EXTENSION postgis;
```

```
\connect rev
--'sde' schema.
CREATE SCHEMA sde AUTHORIZATION sde;
GRANT ALL ON SCHEMA sde TO sde;
GRANT ALL ON SCHEMA sde TO public;
--'rev' schema.
CREATE SCHEMA rev authorization rev;
GRANT USAGE ON SCHEMA rev to public;
--'pmeditor' schema.
CREATE SCHEMA pmeditor authorization pmeditor;
GRANT USAGE ON SCHEMA pmeditor to public;
GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA rev TO role_rev_rev_editor;
--'pmviewer' schema.
CREATE SCHEMA pmviewer autorization pmviewer;
GRANT USAGE ON SCHEMA pmviewer to public;
GRANT SELECT ON ALL TABLES IN SCHEMA rev TO role_rev_rev_viewer;
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON public.geometry_columns to rev;
GRANT SELECT ON public.spatial_ref_sys to rev;
```

```
select schemaname, tablespace, indexname, tablespace;
from pg_indexes
  where schemaname = 'sde'
  order by schemaname, tablename, indexname, tablespace;
```

3.5 Implementación de los Lenguajes de Programación.

Java será utilizado del lado de backend para exponer los servicios de tipo REST. Utilizando una arquitectura en capas o modelo MVC (modelo, vista, controlador). Lo módulos para crear serán los siguientes:

- turno-movil-api
  - Se crearán las interfaces de java que contendrá los métodos que se expondrán en el servicio REST.
- turno-movil-api-specification
  - Utilizando Swagger, el cual es una herramienta para documental APIs REST. Se publicará las especificaciones del servicio sobre Node.JS
- turno-movil-service

    - En este módulo, se crearán las clases de Java que implementan las interfaces del módulo turno-móvil-api y llamarán los DAOs creados en el módulo turno-movil-core.
- turno-movil-core
    - En este módulo, se crearán los DAOs para persistir la información utilizando Hibernate como ORM.

El portal Admin se creará utilizando Angular en su versión 7, el cual estar consumiendo el servicio expuesto por el Backend.

Se crearán los siguientes componentes:

- Login
    - login.component.ts: Maneja la lógica necesaria para el correcto funcionamiento del login.
    - login.component.html: contendrá la maquetación necesaria para mostrar la pantalla de login.
- Home
    - home.component.ts: Maneja la lógica necesaria desplegar información sobre el proyecto.
    - home.component.html: contendrá la maquetación necesaria para mostrar la pantalla de home.
- Navbar
    - navbar.component.ts: Maneja la lógica necesaria para el correcto funcionamiento de la navegación entre las páginas del portal.
    - navbar.component.html: contendrá la maquetación necesaria para mostrar la pantalla de login.

- Instituciones
    - institutions.component.ts: Maneja la lógica necesaria para consultar todas las instituciones consumiendo los servicios publicados en el back-end.
    - institutions.component.html: contendrá la maquetación necesaria para mostrar todas las instituciones registradas.
- Institución
    - institution.component.ts: Maneja la lógica necesaria para ver el detalle de una institución consumiendo los servicios publicados en el back-end.
    - institution.component.html: contendrá la maquetación necesaria para mostrar el detalle de una institución.
- Usuarios
    - users.component.ts: Maneja la lógica necesaria para consultar todos los usuarios consumiendo los servicios publicados del back-end.
    - users.component.html: contendrá la maquetación necesaria para mostrar todos los usuarios registrados.
- Usuario
    - user.component.ts: Maneja la lógica necesaria mostrar el detalle de un usuario consumiendo los servicios publicados del back-end.
    - user.component.html: contendrá la maquetación necesaria para mostrar el detalle de un usuario.

Intellij IDEA

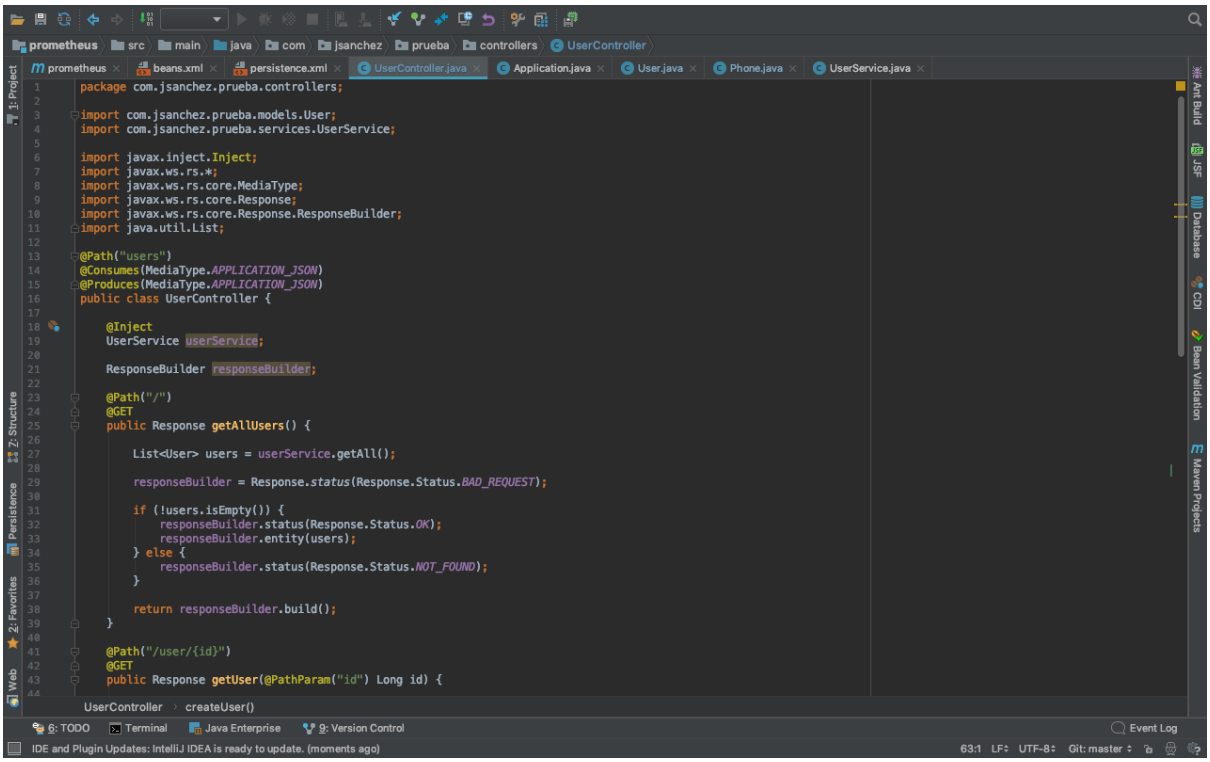


Figura 40. IDE Intellij IDEA. Fuente: Propia

Para el desarrollo del back-end utilizaremos Intellij IDEA. Es uno de los entornos de desarrollo más utilizado para el desarrollo en JAVA. Este posee un gran catálogo de plugins y características que mejoran la productividad de los desarrolladores.

Entre las principales características están:

- Excelente autocompletado de código
- Autoguardado por defecto
- Integración con gestores de base de datos
- Integración con sistemas de control de versiones

Editores de Código

- Sublime Text
- NotePad++
- Visual Studio Code

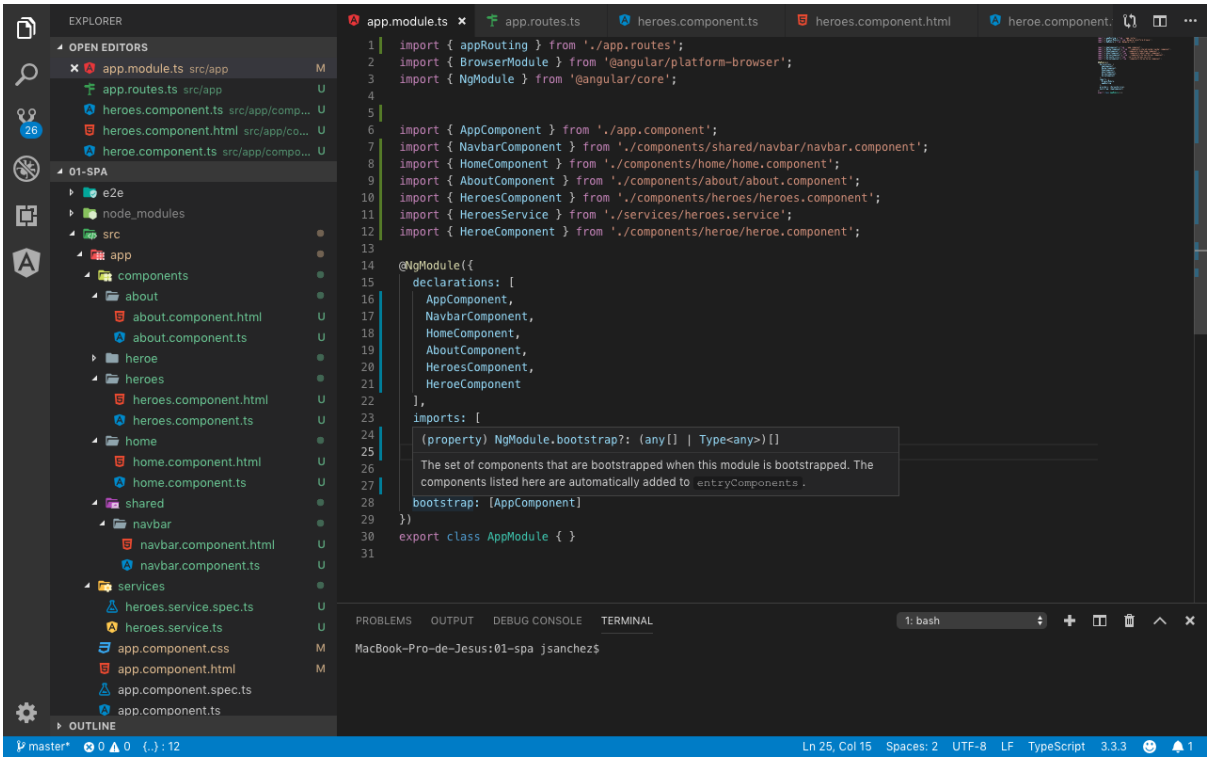


Figura 41. Visual Studio Code. Fuente: Propia

Para el desarrollo del front-end utilizaremos Visual Studio Code. Es un ligero pero poderoso editor de código excelente para el desarrollo de aplicaciones Web o móviles. Está disponible en sistemas operaciones como Windows, Linux y macOS. Viene por defecto con soporte para JavaScript, TypeScript y Node.js. Puede ser extensible mediante plugins a lenguajes como Java, C#, C++, Python y GO.

3.6 Aplicación del Sistema de Posicionamiento Global (GPS) y el Sistema de Información Geográfica (GIS) al Sistema de Gestión de Turnos Móvil.

En la imagen siguiente, se muestra un ejemplo de la búsqueda de una institución bancaria dominicana, llamada Banco Popular Dominicano, donde un usuario aleatorio desea ir al departamento de Servicio al Cliente del banco, y el sistema le da a elegir dos sucursales que se encuentran en su ruta. El punto azul es la ubicación actual del usuario y los dos mensajes se despliegan, con posibilidades de seleccionar lo que más le convenga.

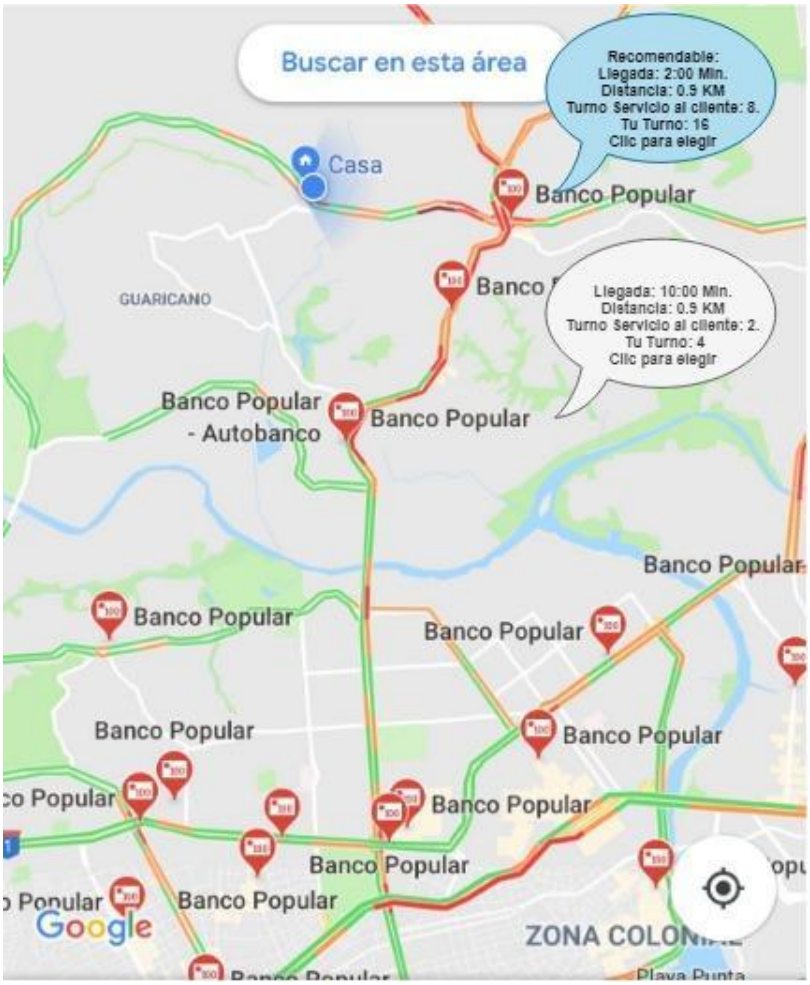


Figura 42. Sistema de recomendación de turnos. Fuente: Propia

En el Sistema de Gestión de Turnos Móvil, se pretende utilizar la tecnología para eficientizar según vectores y características espaciales, la rentabilidad del sistema propuesto con el fin de brindar estadísticas y recomendaciones a clientes y administradores de la plataforma, sobre la necesidad de instaurar/retirar instituciones en ciertos lugares y a la vez donde y como eficientizar las ya establecidas para un mejoramiento sustancial en el servicio al cliente con turnos. Con la ayuda de GIS, se medirán estructuras espaciales, coordenadas, lugar y tiempo como intermedio a este sistema.

3.7 Turno móvil como Sistema de Recomendación

El Sistema de Selección de Turno Móvil se alimenta de los siguientes datos para recomendar:

- Cantidad de personas
- Distancia
- Tiempo entre turnos
- Tránsito

Estas informaciones son tomadas de los demás usuarios que utilizan la aplicación como de las instituciones registradas. Debido a esto, a medida que más usuarios la utilizan, el nivel de precisión aumentará.

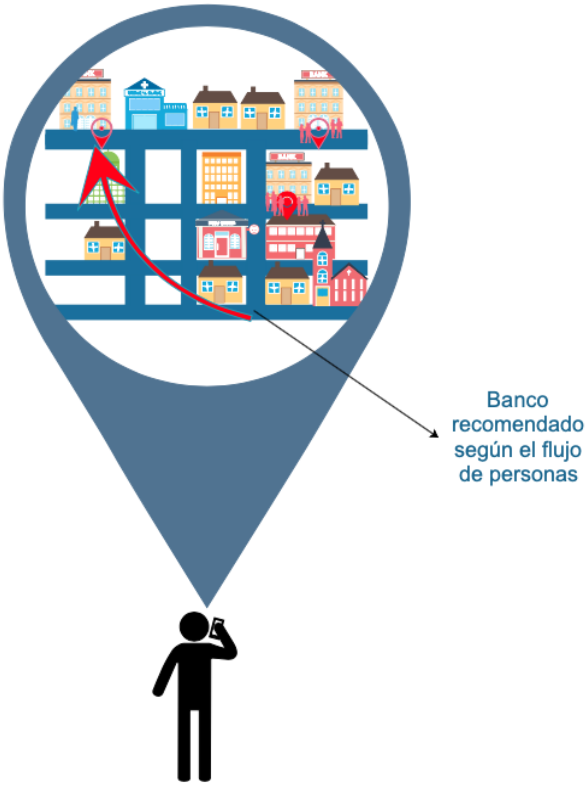


Figura 43. Ilustración del proceso de recomendación. Fuente: Propia.

La figura muestra cómo la aplicación le recomienda al usuario el banco que a simple vista parece el más distante, pero a diferencia de los otros dos, solo tiene una persona en turno, reduciendo el tiempo de espera considerablemente.

Figura 44. Ilustración de un turno desde una aplicación. Fuente: https://inaltoeirl.com/new/wp-content/uploads/Mobile-300x219.jpg

La figura muestra cómo se vería un turno desde un móvil con sistema operativo Android. En la mismo se puede observar la enumeración del turno y si acepta o rechaza el mismo.

3.8 Material para toma de decisiones.

Los sistemas al manejar mucha información y la cual es registrada en base de datos, facilita tomar decisiones en base de patrones de comportamientos. El sistema de Turnos Móvil dispondrá de los siguientes reportes:

3.8.1 Reportes e Informes

La aplicación pondrá a la disposición una serie de informes, los cuales servirán de referencia para evaluar el desempeño y la satisfacción de los clientes en el sistema de gestión de turnos, facilitando la toma de decisiones gerenciales en cuanto a beneficio de la aplicación. Por otro lado, la aplicación será capaz de ofrecer a sus usuarios detalles acerca de los turnos tomados, donde han sido tomados, estadísticas de turnos cancelados o transferidos. Entre otros. El listado de informes serán los siguientes:

- Informe de turnos tomados según fecha, establecimiento, hora.

Informe de Turnos Segun Fecha, Establecimiento, Hora						
Fecha	Hora	Institución/ Establecimiento	Dirección	País	Turno Tomado	Servicio Solicitado
05/01/2019	9:38	Ministerio de Educación Superior, Ciencia y Tecnología (Mescyt)	Av. Máximo Gómez 31, Santo Domingo	República Dominicana	6	Entrega de Documentos
10/03/2019	8:42	Banco BDH León	Independencia Km. 8 Zona: Santo Domingo	República Dominicana	9	Servicio al Cliente Negocios
12/03/2019	11:36	MAPFRE Seguros	Ave.Abraham Lincoln #952 Esq.Jose A. Soler, Piantini	República Dominicana	5	Reporte de Siniestro
18/03/2019	14:11	Centro del Automovilista	Av. John F, Kennedy, Calle Paseo de Los Periodistas No. 8, Santo Domingo 10413	República Dominicana	3	Reporte de Accidente
22/03/2019	8:57	Hospital General de la Plaza de la Salud (HGPS)	Av Ortega y Gasset, Santo Domingo	República Dominicana	10	Visita a Doctor
28/03/2019	17:32	Banco Popular Dominicano	Avenida John F. Kennedy, Santo Domingo 10203	República Dominicana	18	Caja

Tabla 2. Ejemplo de Informe de turno tomado por parámetros. Fuente: Propia

- Informe de turnos rechazados según fecha, establecimiento y hora.

Informe de Turnos Rechazados Segun Fecha, Establecimiento, Hora						
Fecha	Hora	Institución/ Establecimiento	Dirección	País	Turno Asignado	Servicio Solicitado
08/01/2019	10:22	Torre Banreservas	Av. Winston Churchill, Santo Domingo 10148	República Domir	19	Servicio al Cliente
04/03/2019	7:49	Laboratorios Referencia	Calle Max Henriquez Ureña & Calle Respaldo Rafael Augusto Sánchez, Santo Domingo	República Dominicana	5	Servicios Múltiples Asegurado
08/03/2019	15:39	Cresa Radiología Especializada	La Esperilla, Esperilla, Av. Pedro Henríquez Ureña, Santo Domingo	República Dominicana	6	Tomografía
10/03/2019	17:35	Banco Popular Dominicano	Avenida John F. Kennedy, Santo Domingo 10203	República Dominicana	18	Servicio al Cliente

Tabla 3. Ejemplo de informe de turnos rechazados. Fuente: Propia

3.8.2 Dashboards

Los dashboards son herramientas útiles para ver un resumen completo que cuenta con gráficos e información compactada para la toma de decisiones del departamento que lo amerite. Los dashboard son útiles para aquellas empresas o personas que deseen ver una vista general de todos sus procesos y les permita acceder a funciones especiales tales como las de una página de inicio.

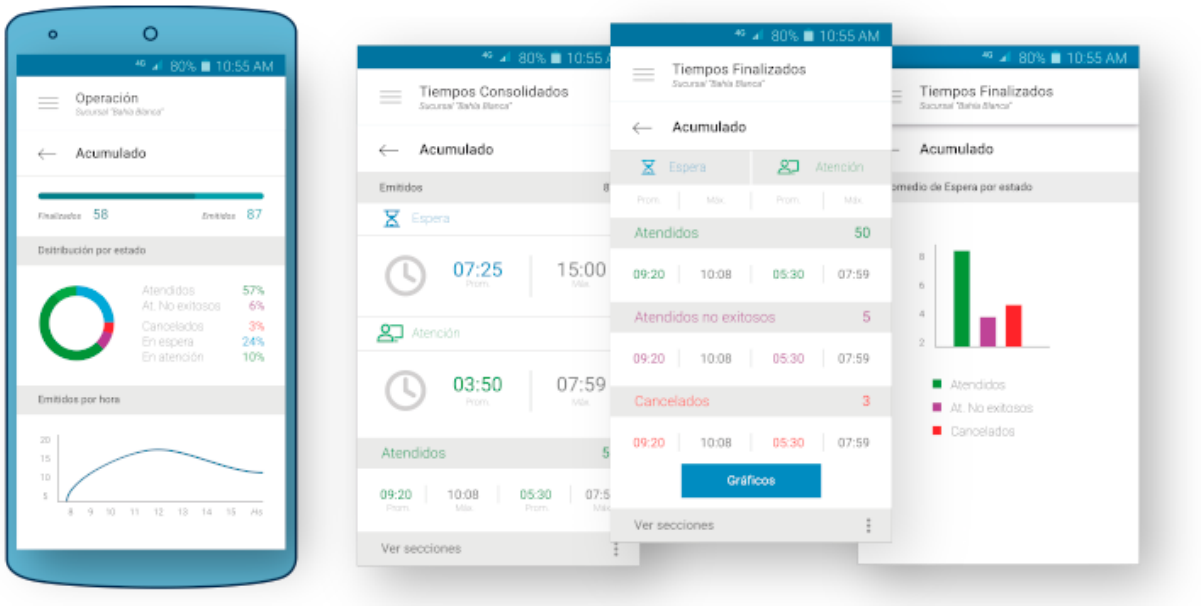


Figura 45. Dashboard de aplicación de Sistema de Gestión de turno móvil. Fuente: Google.com

3.9 Resumen

En el tercer y último capítulo, se puede constatar el esquema propuesto como tal, para el mejoramiento del Sistema de Gestión de Turnos Móviles. El fin es mostrar la arquitectura propuesta a nivel de backend y parte visual, y cuáles serían las demás utilidades a tomar en cuenta. En esta fase, la parte del presupuesto se lleva a cabo ya que se incluyen los recursos necesarios tanto en personal, como la arquitectura en la nube y física requerida y el plan de funcionamiento. Los reportes para tomas de decisiones juegan un papel fundamental ya que en ellos se desglosan un historial de informaciones necesarias para conocer la rentabilidad de la información y el uso de ésta, y con el fin de mejorar la auditoría de los procesos llevados con esta aplicación.

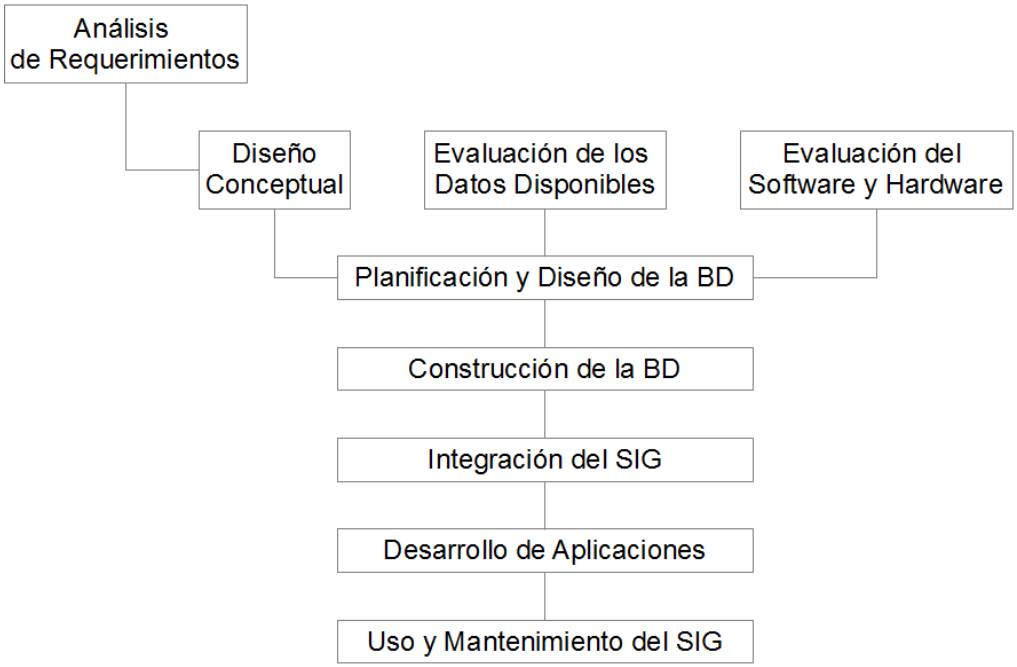


Figura 46. Esquema de proceso seguido en el capítulo tres sobre la construcción propuesta de la solución de la problemática. Fuente: http://volaya.github.io/libro-sig/chapters/Bases_datos.html

CONCLUSIONES Y RECOMENDACIONES

1. Conclusiones

A lo largo del presente trabajo, se demostró la necesidad existente de hacer una mejora sustancial en el manejo de los turnos en la República. Factores determinantes como la organización y la mejora del servicio al cliente, alimentan el menester que actualmente existe en los establecimientos e instituciones existentes en el País. Se observó que una gran población, admite que la viabilidad de la idea es totalmente efectiva, y de llevar a cabo el plan, mejoraría sustancialmente la atención al cliente y sumaría un adicional aporte al movimiento del desarrollo tecnológico del país, demostrando este con la tecnología de administración, procesamiento y almacenamiento propuesta, como la gestión del uso del aplicativo y el uso de mapas geográficos con información espacial, factor que en la República Dominicana sigue en su cuna de nacimiento.

Se determinaron que los costes de la implementación abordarán los recursos, las dos infraestructuras (local y nube) y los costes en licenciamiento, entre otros.

Entre los detalles de infraestructura de software, se hizo mención de la estructura de la base de datos, la cual almacenará una entidad caché para guardar datos cargados utilizando el Sistema de Información Geográfica, estudiando los espacios vectoriales y espaciales, para hacer los cálculos pertinentes para la mejora de la ubicación geoglobal del Sistema de Gestión de Turnos.

Por último, se definió la estructura del back-end (Código interno), los procesos, los pasos a dar en la aplicación y una previsualización de lo que se espera obtener en el momento del desarrollo. Los materiales de toma de decisión como Reportes y Dashboards están incluidos.

2. Recomendaciones

Dentro de esta propuesta tan ambiciosa, se requiere que se continúen haciendo mejoras progresivas del Sistema de Gestión de Turnos Móvil, de este modo, a las futuras personas que les haya parecido interesante la propuesta, es válida la complementación de agregar más factores con el fin de mejorar aún más los tiempos en espera del servicio al cliente con el uso del Sistema de Gestión de Turnos Móvil, además de la agregación de más metaheurísticas y comparar con los resultados finales.

Otra recomendación sería la mejora de la adaptación en el uso de la tecnología móvil para que el sistema abarque a la mayor cantidad de tipos de personas, puesto que, en la República Dominicana, gran cantidad de personas aún no cuentan con conocimiento o dicha económica de contar dispositivos móviles o saber manipularlos y la aplicación se quedaría siendo hábil solo para el grupo reducido de personas que tienen conocimiento de usar un dispositivo móvil y la capacidad monetaria para adquirirlo.

REFERENCIAS BIBLIOGRÁFICAS

- Ramos Rosales, E. C. R., Verduzco Ramírez, J. A., García Díaz, N. y Arceo Díaz, S. (2019). Diseño e implementación de un sistema integral de control de turnos. 3C Empresa. Investigación y pensamiento crítico, 8(1), pp. 92-111. doi: http://dx.doi.org/10.17993/3cemp.2019.080137.92-111

- Oblitas, José Fernando Pardo, and Jessica Janina Cabezas Quinto. Diseño de Una Aplicación Móvil Para La Gestión de Turnos En Un Centro de Atención Al Cliente. 2017.

- Mendieta, Zumba, and Iván José. Diseño y Desarrollo de Una Aplicación Móvil Para Gestión de Turnos y Visualización Del Catálogo de Diseños Para El Salón de Belleza Andrea. 2016.

- Saguay, Cristhian Eduardo Guachamin, and Diego Armando Hurtado Guapulema. Desarrollo de Una Aplicación Web Adaptiva Para La Gestión de Turnos e Historias Clínicas Utilizando El FRAMEWORK VAADIN Para La Clínica “Nova Clínica San Marcos”. 2017.

- Christensson, P. (2011, September 23). Programming Language Definition. Retrieved 2019, Mar 30, from https://techterms.com

- Rivera, D., & Lorena, J. (2014). Desarrollo de un sistema web para la reservación de turnos para licencias en la Agencia Nacional de Tránsito de la ciudad de Babahoyo. Retrieved from http://dspace.utb.edu.ec/handle/49000/1194

- Averos, B., Beatriz, M., Zúñiga, V., & Humberto, Á. (2014). La tecnología android y su incidencia en el desarrollo de una aplicación móvil para la geolocalización de los centros asistenciales y farmacias de turnos para la dirección provincial de salud los ríos ubicada en la ciudad de Babahoyo. Retrieved from http://dspace.utb.edu.ec/handle/49000/1206

- Ramos, M., Rivera, R. y Lorena, J. (2018). Desarrollo de un sistema web para la reservación de turnos para licencias en la Agencia Nacional de Tránsito de la ciudad de Babahoyo [Consulta: 06-Apr-2019]. Recuperado de: http://dspace.utb.edu.ec/handle/49000/1194

- Rahendra, Orissa Rivelino, and Arief Subroto. “MAKE A RESERVATION APPLICATION AND PURCHASE CONCERT TICKETS BY USING WIRELESS MOBILE TELEPHONE APPLICATION.” Faculty of Industrial Technology, 2011.

- Kharwade, Pranjali, et al. Smartphone Application for Railway Ticket Reservation and Validation Using Mobile Network. 2014.

- Anuar, J., Musa, M., & Khalid, K. (2014). Smartphone’s Application Adoption Benefits Using Mobile Hotel Reservation System (MHRS) among 3 to 5-star City Hotels in Malaysia☆. Procedia - Social and Behavioral Sciences, 130, 552–557.

- Fowler, M. (1999). UML gota a gota. Londres, Inglaterra: Pearson Education. González, Y., y Romero, Y. (2012). Patrón Modelo-Vista-Controlador. Revista Telemática, 11(1), pp. 47-57 [Consulta: 06-Apr-2019]. Recuperado de: http://revistatelematica.cujae.edu.cu/index.

- “Angular,” Angular. [Online]. Available: https://angular.io/. [Accessed: 05-Apr-2019].

- “IntelliJ IDEA: The Java IDE for Professional Developers by JetBrains,” JetBrains. [Online]. Available: https://www.jetbrains.com/idea/. [Accessed: 06-Apr-2019].

- “AWS | Cloud Computing - Servicios de informática en la nube,” Amazon. [Online]. Available: https://aws.amazon.com/es/. [Accessed: 06-Apr-2019].