

# Gestión de toma de estados digitales. Una solución para escenarios sub-urbanos

Aciti Claudio, Illescas Gustavo, Noceto Agustín

*Instituto de Investigación en Tecnología Informática Avanzada, Facultad de Ciencias Exactas,  
Universidad Nacional del Centro de la Provincia de Buenos Aires.  
Paraje Arroyo Seco, Tandil (7000), Argentina. +54 249 4385680.  
caciti@exa.unicen.edu.ar, illescas@exa.unicen.edu.ar, agusnoceto@gmail.com*

## RESUMEN.

En la actualidad diferentes organizaciones, como por ejemplo las cooperativas de servicios de luz y gas, cuentan con limitaciones para la recolección de datos, especialmente en escenarios sub-urbanos. Estos servicios requieren la visita al domicilio del usuario por parte de algún agente ya que el medidor está ubicado en la vivienda del consumidor. Hoy día siguen existiendo casos donde se observa a los agentes completar la información en papel o, en los mejores casos cuentan con colectores de datos especializados, aunque la carga sigue siendo manual. Cuando se procesa el dato surge un potencial inconveniente de que fuera mal registrado in situ lo que lleva a tener que volver al lugar a tomar una nueva medición o usar una estimación basada en historiales. Una cuestión adicional que surge a partir de la llegada de los servicios a zonas sub-urbanas es la necesidad de poder auditar a los agentes para conocer si la tarea se ha realizado in-situ, por ejemplo a partir del registro de la geolocalización. El objetivo del trabajo es proveer una solución adaptada a las necesidades de organizaciones que necesitan recolectar datos distribuidos espacialmente abarcando tanto zonas urbanas como sub-urbanas, contando con validaciones de posicionamiento como así también de procesamiento de datos. Consta de dos etapas bien definidas: a) el proceso de generación de recorridos para los agentes recolectores de datos y b) el proceso de registración y auditoría. La metodología a utilizar se fundamenta en la gestión de procesos y procedimientos como así también en lo concerniente a proyectos de software para la elección de la tecnología y desarrollo. Esta solución puede ser aplicada a numerosos casos como por ejemplo encuestas, entrega de encomiendas y otros servicios puerta a puerta. Los resultados previos se obtuvieron de un caso de estudio en una cooperativa de la costa atlántica.

**Palabras Claves:** Toma de estados. Optimización de recorridos. Geolocalización. Cooperativas.

## ABSTRACT

Nowadays, different organizations, such as gas and electric companies, have data collection limitations, especially on suburban scenarios. These services require an agent's visit to the user's domicile since the meter is located in his house. Today, there are still cases where the agents are seen completing the information on paper or in the best case scenario there is a specialized data collector, even though the loading of information is still manually. When the data is processed a potential inconvenience of being wrongly registered in situ arises, which leads to going back to the place to perform a new measuring or using an estimate based on previous records. An additional issue that appears with the arrival of services to suburban areas is the necessity of being able to audit the agents to know if the task has been carried out in situ, for example starting from the record of the geolocating. The objective of this work is to provide a solution adapted to the needs of the organizations that need to collect spatially distributed data covering urban areas as well as suburban ones, not only counting on positioning validation, but also on data processing validation. The work consists of two well defined stages: a) the production of the data collection agents' itinerary and b) the process of recording and auditing. The methodology to be used is based on the processes and procedures management as well as on everything related to software projects to the selection of technology and development. This solution may be applied to numerous cases such as surveys, package deliveries and other door to door services. The previous results were obtained from a study case in a company on the Atlantic coast.

**Key Words:** Taking states. Path optimization. Geolocation. Cooperatives.

## 1. INTRODUCCIÓN

En un gran número de organizaciones, el desarrollo de actividades requiere la visita al domicilio del usuario por parte de algún agente. En algunas de ellas es un requisito fundamental, como por ejemplo en los servicios que cuentan con un medidor ubicado en la vivienda del consumidor, especialmente en escenarios sub-urbanos, donde hay que recorrer muchos kilómetros para tomar la medición.

Estas tareas “a domicilio” requieren de la contratación de empleados que las realicen, cuyas remuneraciones representan en muchas ocasiones el presupuesto más importante [1]. Es por esto que es importante que dichas entidades puedan contar con una herramienta que permita minimizar la cantidad de tiempo que toma a sus empleados recorrer los distintos puntos geográficos o, lo que es lo mismo, maximizar la cantidad de lugares que puedan visitar en una jornada. Una cuestión adicional que surge a partir de la llegada de los servicios a zonas sub-urbanas es la necesidad de poder auditar a los agentes para conocer si la tarea se ha realizado in-situ, por ejemplo a partir del registro de la geolocalización.

Además actualmente, algunas entidades dependen fuertemente de la finalización del recorrido para poder continuar con la siguiente fase de sus procesos, como por ejemplo las cooperativas de luz o gas, que necesitan las lecturas de los medidores para poder emitir las facturas correspondientes.

Generalmente, estas lecturas son registradas en papel en una primera instancia y, una vez finalizado el recorrido, son ingresadas manualmente al sistema donde se validan las mediciones y se almacenan. Las desventajas de este método son varias. Entre ellas la más destacable es el costo que trae aparejada la tardía validación de los datos. Esto se ve reflejado especialmente en los casos en que la lectura ingresada no es válida, exigiendo un retorno al lugar a corroborar los datos obtenidos previamente.

A partir de la necesidad de adelantar determinadas tareas del proceso con el objetivo tanto de reducir costos como de optimizar resultados, es posible pensar en el enérgico avance de las tecnologías de hardware, software, y telecomunicaciones: éste ha promovido un nuevo subsector dentro de las TICs (tecnologías de la información y la comunicación), que está dando inicio a una nueva generación de sistemas o herramientas tecnológicas de gran impacto en el sector corporativo. Se trata de las soluciones basadas en tecnologías móviles e inalámbricas, confluencia que hoy se convierte en sinónimo de productividad.

Asistimos entonces al nacimiento de una nueva categoría de producto en el mercado de soluciones de automatización, que más allá de garantizar la portabilidad de los datos de múltiples procesos de campo, permite un tránsito bidireccional de la información entre el agente que se encuentra realizando una actividad remota (como procesos de venta, entrega de mercancías, encuestas, etc.) y su compañía, entregándole información relevante en tiempo real que anteriormente solo residía en sistemas centrales de información, y del mismo modo reportando en su empresa el balance pormenorizado de su gestión.

Si bien se podría utilizar un colector de datos para realizar la misma tarea, el beneficio de utilizar dispositivos móviles es que es una tecnología que hoy en día, prácticamente todos utilizan con facilidad. Además se tiene disponible para realizar cualquier tipo de llamada de emergencia. Pero la ventaja más clara que se observa es la de enviar los datos online para comprobar su correctitud en el instante mismo de la recolección del dato.

Teniendo en cuenta entonces las limitaciones que presenta el sistema actual y haciendo uso de dichos avances tecnológicos, se plantea la necesidad de desarrollar una aplicación en un dispositivo móvil que permita obtener datos y registrarlos online en una base de datos conectada a un servidor, usando tecnologías que permitan una colección de datos más rápida y eficiente. Por otro lado se propone el desarrollo de una herramienta que sea capaz de generar recorridos más eficientes para la toma de datos y su posterior procesamiento.

Objetivos parciales:

- Desarrollar una aplicación en un servidor capaz de:
  - optimizar los recorridos maximizando así la cantidad de destinos a visitar en una jornada.
  - Verificar, de forma online, los datos recibidos y almacenarlos.
- Desarrollar una aplicación en un dispositivo móvil que permita:
  - Conectarse al servidor referido para solicitar la información de los recorridos.
  - Recolectar datos y enviarlos a dicho servidor de forma online para su validación.
  - Optimizar la toma de datos para reducir costos y disminuir la tasa de error de registro de datos.
  - Almacenar los datos localmente como copia de seguridad.

## 2. DESARROLLO

### 2.1 Visión general del sistema

Para cumplir con los objetivos definidos precedentemente se diseñó una herramienta que colecta y almacena mediciones de servicios públicos, que utiliza dispositivos móviles como herramienta de colección, capaz de generar recorridos optimizados y establecer una comunicación bidireccional con dispositivos móviles utilizando conceptos emitidos en [2 - 3].

El sistema está dividido en dos módulos principales. El primero de ellos se denomina *Módulo Servidor* y es el encargado de:

- almacenar información de los medidores en una base de datos
- generar los recorridos
- definir una interfaz de comunicación con los dispositivos móviles

El segundo módulo se titula *Modulo Cliente* y es el módulo que se ejecuta en los dispositivos móviles. Está encargado de presentarle al usuario la información del recorrido solicitado al servidor, y permitir el ingreso y actualización on-line de las mediciones. Su interfaz gráfica, por medio de diferentes vistas, facilita la visualización de la información del recorrido y el ingreso de lecturas, mejorando el tiempo de obtención de datos.

La comunicación que se establece entre estos dos módulos es bidireccional. Así, el servidor primero envía tanto la información de los recorridos previamente generados como la información de cada uno de los medidores que componen dicho recorrido. Luego actualizará la base de datos con la información de las lecturas que reciba de los distintos módulos cliente.

Por otro lado, el módulo cliente, una vez recibida la información de cada medidor (dirección, titular, id, observaciones, posición, etc.), la presentará en el orden establecido por el servidor al momento de generar los recorridos.

Esta vista, junto con la de completitud del recorrido y la vista georreferenciada, conforman la interfaz de usuario necesaria para la recolección de datos.

### 2.2 Generación de Recorridos

Uno de los principales problemas del procedimiento actual de recolección de datos es el tiempo asociado a los recorridos. En consecuencia, uno de los requerimientos principales del sistema es la maximización de la cantidad de destinos a visitar durante una jornada laboral. Dicha maximización tendrá un importante impacto en el tiempo total del proceso de facturación de los servicios. Nace así la necesidad de optimizar los recorridos efectuados por cada uno de los agentes.

En el estudio de algoritmos de recorrido mínimo es frecuente resolver este tipo de problemáticas. Desde hace décadas son conocidas soluciones casi óptimas y no aparenta haber mucho lugar para innovar. No obstante, hay un impedimento a la hora de intentar solucionar la problemática utilizando cualquiera de estos algoritmos, impedimento que viene dado por la cantidad de actores involucrados en el recorrido.

En el caso de estudio escogido, la tarea de recolección de datos se divide en diversos agentes. Cada uno de ellos efectúa un recorrido distinto, siendo la suma de todos los recorridos el total de los destinos a visitar. Esta particularidad impide solucionar el problema aplicando directamente alguno de los algoritmos conocidos.

Se plantea entonces la necesidad de dividir el conjunto inicial creando distintas zonas de manera que los destinos que le correspondan a cada zona estén tan cerca entre sí como sea posible. De esta manera se minimizará el recorrido a efectuar para pasar por cada uno de ellos.

Por tratarse de un problema tipo NP-completo, la única manera de hallar la solución óptima es evaluar todas las posibles combinaciones de recorridos y quedarse con aquella cuyo trazado utiliza la menor distancia.

El inconveniente de esto reside en el número de posibles combinaciones, que viene dado por el factorial del número de destinos –  $N!$ , Siendo  $N$  la cantidad total de destinos -. Esto hace que la solución exhaustiva sea impracticable para valores de  $N$  incluso moderados, con los medios computacionales actualmente al alcance.

Por lo tanto se optó por crear una solución aproximada en la cual en una primera instancia se realiza la generación de zonas, es decir se divide el conjunto de destinos en grupos o zonas, agrupados por proximidad; luego se procede a optimizar el recorrido interno de cada una de esas zonas.

### 2.3. Generación de zonas

#### 2.3.1 Definiciones y Conceptos

Punto: se denomina con el término punto a una coordenada geográfica en la cual se localiza un sitio de interés, en este caso un medidor de servicios.

Zona: Se conoce como zona a un conjunto de puntos. El conjunto total de puntos se fracciona en una cantidad convenida de zonas de manera tal que cada punto pertenezca exactamente a una zona. Cada zona luego será analizada para generar un recorrido de los puntos de distancia mínima.

### 2.3.2 Elección de la estructura a utilizar

Para facilitar la generación de las zonas primero se almacena el conjunto total de puntos en una estructura especial de árbol de partición binaria del espacio (BSP por sus siglas en inglés), llamado Árbol kd.

Un Árbol kd es una estructura de datos de particionado del espacio que organiza los puntos en un Espacio euclídeo de k-dimensiones.

Estos árboles son un caso especial de los árboles BSP pero, a diferencia de éstos (donde los planos pueden ser arbitrarios), el árbol kd emplea sólo planos perpendiculares a uno de los ejes del sistema de coordenadas. Además, todos los nodos de un árbol kd, desde el nodo raíz hasta los nodos hoja, almacenan un punto. Como consecuencia, cada plano debe pasar a través de uno de los puntos del árbol kd.

En la Figura 1 podemos ver un Árbol kd-tree tridimensional. El primer corte (rojo) separa la celda raíz (en blanco) en 2 subceldas, cada una de las cuales es luego separada (verde) en 2 subceldas más. Finalmente, cada uno de esos 4 cortes (azul) es separado en 2 subceldas nuevamente. Como ya no hay más más cortes, las 8 celdas finales son llamadas hojas.

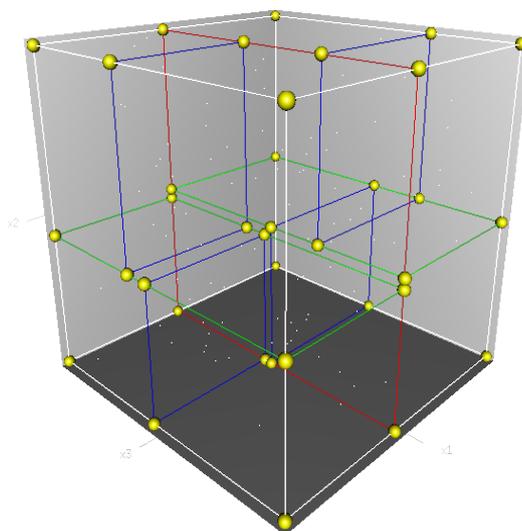


Figura 1. Árbol K-D tridimensional

### 2.3.3 Construcción del árbol KD

La idea principal del algoritmo para la construcción del árbol k-d a partir de un conjunto de puntos P es la siguiente: encontrar un hiperplano que divida el conjunto P en dos subconjuntos y proceder recursivamente con los subconjuntos. El principal aspecto a resolver es la elección del hiperplano y de la coordenada que va a servir para dirigir la búsqueda a un lado u otro del hiperplano, a saber: la coordenada discriminante.

Para intentar conseguir que cualquier punto tenga la misma probabilidad de estar a un lado o a otro del hiperplano, y por tanto que el árbol resulte lo más equilibrado posible, se suele elegir el hiperplano de forma que se sitúe en la mediana de los valores de la coordenada discriminante.

El árbol se construye de la siguiente forma: en cada nodo, que representa un conjunto de puntos (el nodo raíz representa a todo el conjunto inicial), se elige la coordenada discriminante y se obtiene la mediana de los valores de dicha coordenada en los puntos del conjunto; a continuación, se divide dicho conjunto en dos subconjuntos utilizando la mediana, situando en un subconjunto aquellos puntos para los que el valor de la coordenada discriminante sea menor o igual que el de la mediana, y en el otro subconjunto los prototipos cuya coordenada discriminante sea mayor que la mediana. Luego, se crean recursivamente los árboles asociados a cada uno de los subconjuntos. El proceso termina cuando el tamaño del conjunto de prototipos es menor o igual que el valor fijado como tamaño de 'cluster', y en este caso el nodo será una hoja.

En este caso en particular el árbol kd es un árbol *bidimensional*, donde cada plano se corresponde con uno de los ejes del sistema de coordenadas geográficas. La coordenada discriminante variará, entonces, entre *latitud* y *longitud* dependiendo del eje que se utilice para ordenar. El 'clúster' será del tamaño de *zona* elegido al momento de la generación de recorridos.

## 2.4 Optimización del recorrido interno de cada zona

La construcción de las zonas a partir del árbol Kd proporciona una buena distribución de los puntos en las distintas zonas. No obstante, por tratarse de un algoritmo de aproximación, dicha solución no es óptima. En especial para los casos de puntos que se encuentran en los límites entre dos zonas, con lo cual surge la necesidad de enriquecer la solución.

Para ello se opta por diseñar un algoritmo que analice los puntos de una determinada zona y compruebe si su pertenencia a dicha zona es adecuada o si este debiera pertenecer a otra por cuestiones de proximidad.

Una vez conformadas las distintas zonas están dadas las condiciones para aplicar el algoritmo tradicional del camino del viajante [4-5-6] a cada una de ellas para lo cual se elige la versión *greedy*.

Se considera como punto inicial de cada zona al punto con mayor distancia a la media geográfica de la zona. Una vez seleccionada la coordenada se procede a elegir el siguiente punto que es escogido en base a su distancia con respecto al punto de origen. El punto de menor distancia es el siguiente punto a visitar en el recorrido. Se procede de la manera análoga hasta completar todos los puntos de la zona.

## 2.5 Diseño detallado del sistema

Como se detalla en el inciso anterior, el sistema completo se divide en dos módulos principales. El primero, llamado Módulo Servidor, encargado de la generación de recorridos y almacenamiento de datos. El segundo, llamado Módulo Cliente, compuesto por una aplicación que reside y ejecuta en un dispositivo móvil con sistema operativo Android, conexión a internet y GPS, capaz de solicitar la información del recorrido al módulo servidor y presentarla al usuario a través de una interfaz gráfica intuitiva.

En la Figura 2 se observa la subdivisión en módulos de granularidad más fina, que pueden identificarse dentro de ambos módulos, y cómo estos interactúan entre sí.

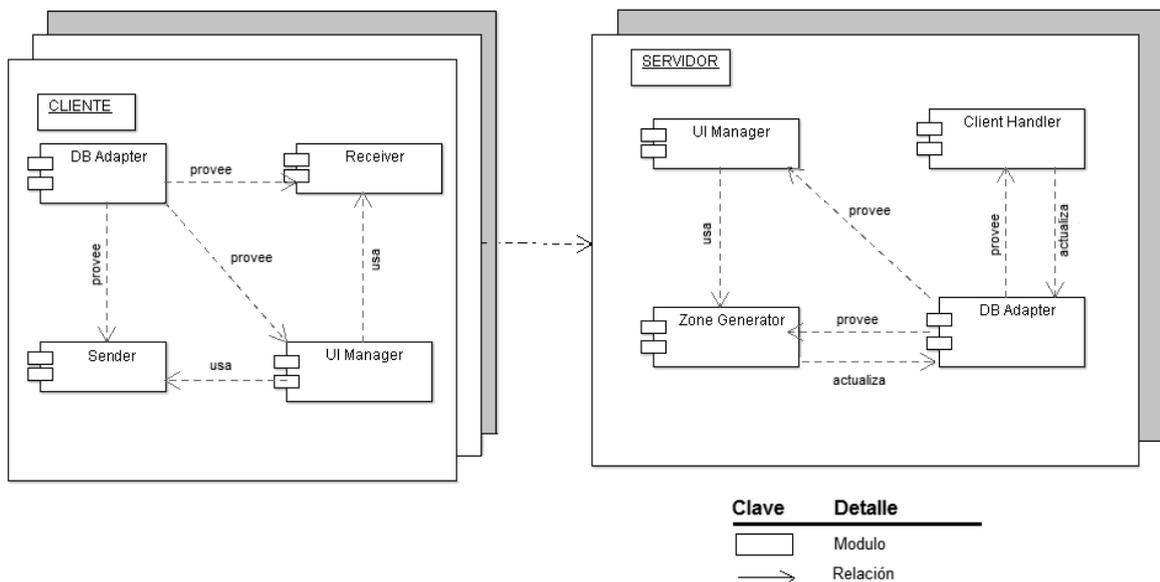


Figura 2: Vista Lógica del Sistema

Entre las responsabilidades que cada sub módulo posee se identifican las siguientes:

- **Módulo Cliente**
  - **Receiver:** Es el encargado de solicitar los datos de un recorrido al módulo Servidor. Este, luego de procesar los datos los pasará al sub módulo db Adapter.
  - **Db Adapter:** La información del recorrido requerida al servidor es almacenada temporalmente en una base de datos interna en el dispositivo móvil. Éste sub módulo es la interfaz entre la aplicación y la base de datos.
  - **Sender:** Se encarga de interactuar con el sub módulo db Adapter para solicitar las lecturas colectadas y establecer una conexión con el servidor para enviarlas.
  - **UI Manager:** este submódulo es el encargado de ejecutar correctamente las operaciones que el usuario ingresa, y presentar por pantalla los datos del recorrido, así como las notificaciones del sistema.
- **Módulo Servidor:**
  - **Zone Generator:** Es el módulo encargado de la generación de las zonas y los recorridos e interactuar con el módulo DbAdapter para almacenarlos en la base de datos principal.
  - **DbAdapter:** de manera análoga con el submódulo db Adapter del módulo cliente, éste se encarga de controlar la interacción directa con la base de datos principal del sistema.
  - **Client Handler:** Cuando un dispositivo móvil establece una conexión con el Servidor este submódulo se inicia y se encarga de establecer el diálogo con el Módulo Cliente.
  - **UIManager:** Ese submódulo se encarga de presentar la información de los medidores al usuario, mostrarle las opciones disponibles, y ejecutar el comando ingresado.
  - Para ilustrar los procesos involucrados entre los distintos módulos y submódulos se desarrollaron los diferentes diagrama de secuencia como el representado en la Figura 3:

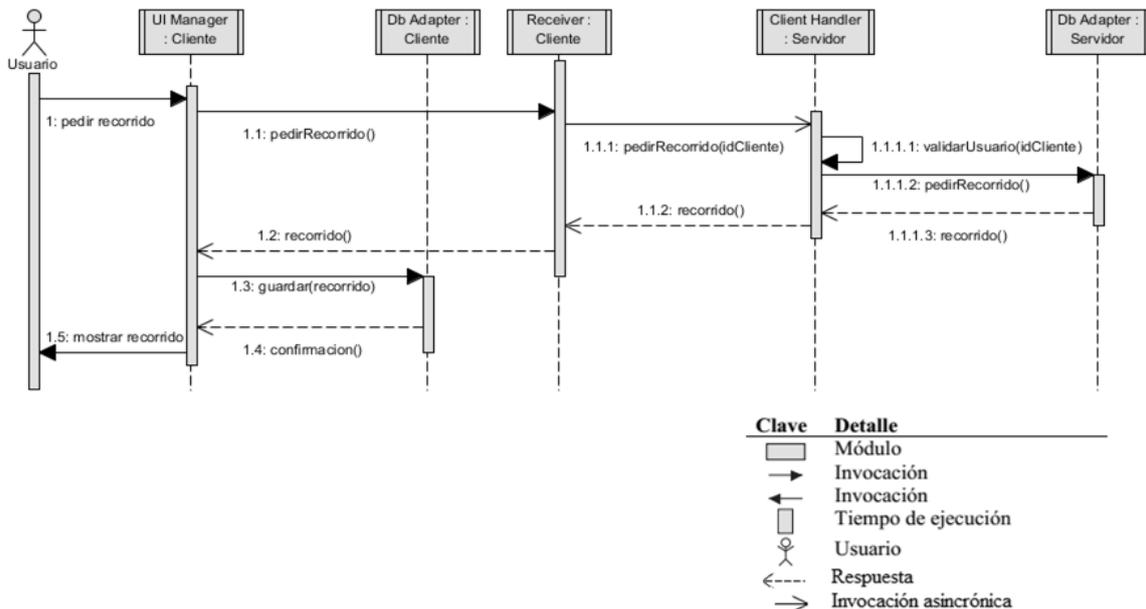


Figura 3. Diagrama de Secuencia cuando el usuario pide el recorrido

Para asegurar la performance del sistema al momento de generar los recorridos, se requiere como mínimo un procesador de 2.0 GHz, con 3 GB de memoria RAM.

Para una mejor usabilidad se diseñó como una aplicación web, de esta manera puede ser accedida desde cualquier computadora con acceso a internet, evitando el uso de la máquina donde ejecuta el Servidor para realizar las tareas de importación o exportación de datos y generación de recorridos.

## 2.6 Generación de recorridos

Como proceso de inicialización se deben importar los datos de localización de los medidores, una vez importados están dadas las condiciones para ejecutar la generación de recorridos.

### 2.6.1 Instanciación de la solución

Las clases implementadas en la solución son:

- **Ipath Generator:** Es la interfaz principal del patrón, la implementan todas las clases. Tiene el método generar Recorrido que deben extender las clases que la implementen.
- **Concrete Generator:** Es una clase abstracta de la cual heredan los dos componentes concretos:
  - **Cantidad De Zonas:** genera C cantidad de zonas con C como parámetro del sistema. Esto se realiza recorriendo una única vez la lista de puntos y superándola en N cantidad de subconjuntos. El tiempo total de ejecución, es por lo tanto  $O(N)$  con N la cantidad de puntos.
  - **Medidores Por Zona:** genera M cantidad de zonas con  $M = \text{cantidad total de puntos} / \text{medidores Por Zona}$ , donde medidores Por Zona es una variable pasada como parámetro. El costo computacional es de  $O(N)$  ya que la lista de puntos se recorre una única vez.
- El componente concreto a utilizar se elige en tiempo de ejecución a través de la clase Zone Generator Factory que implementa el patrón Factory Pattern para desligar al cliente (en este caso el Zone Controller) de la instanciación del objeto, como se puede apreciar en la figura 4.
- **Abstract Decorator:** Al igual que en el diagrama del patrón original, es la clase abstracta de la cual heredarán los otros decoradores. Esta tiene una instancia de IPath Generator, dando lugar a que se apile un decorador encima de otro.
- **KdTree Decorator:** Es el decorador que implementa la solución del árbol kd bidimensional explicado anteriormente. Su costo es  $O(N \cdot \log N)$ .
- **Point Relocator Decorator:** Es el decorador que reasigna los puntos a zonas que estén más cercanas a él. Este refinamiento recorre cada punto del conjunto de puntos y por cada uno de ellos se fija en las distintas zonas intentando descubrir una más adecuada a la zona a la que pertenece actualmente, en cuyo caso lo reasigna. Este cálculo de distancia se realiza comparando la latitud y longitud del punto contra la media de latitud y longitud de una zona (que se calcula contemplando todos los puntos de dicha zona). El costo computacional entonces es  $O(N \cdot M)$  con N cantidad de puntos y M cantidad de Zonas.
- **Zone Optimizator Decorator:** Es una clase abstracta que define la estructura de un algoritmo para modificar cada zona. Cuenta con un método abstracto optimizar Recorrido De Zona el cual deberán implementar las clases que hereden de ella. Este método es el "hook" de donde "cuelgan" las diferentes alternativas para la modificación del recorrido interno de una zona.
- **Greedy Zone Decorator:** Esta clase es un decorador que extiende de Zone Optimizator Decorator e implementa el gancho optimizar Recorrido De Zona en donde aplica un algoritmo greedy para efectuar el recorrido de todos los puntos de una zona comenzando por el punto más lejano al punto medio de la zona, y hasta recorrer todos los puntos.
- El costo para una determinada zona es de  $O(N^2)$  donde N es la cantidad de puntos por zona. Entonces el costo computacional total – para todas las zonas – será de  $O(M \cdot N^2)$  con M cantidad de Zonas.

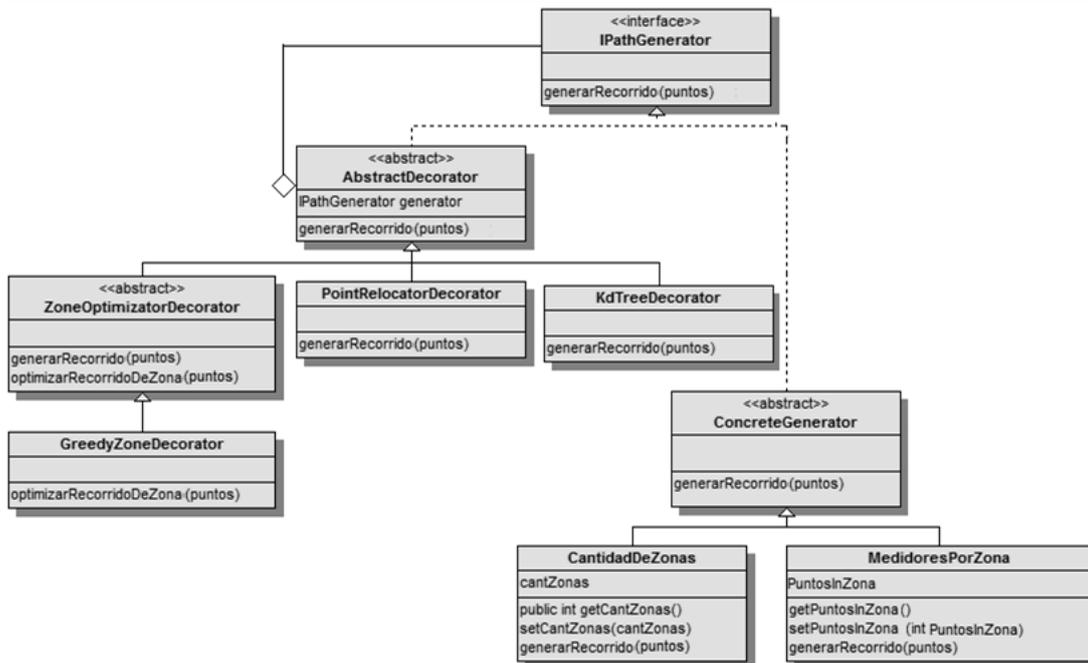


Figura 4. Diagrama de Clases, instanciación del patrón elegido.

El módulo UIManager presentará al usuario administrador todos los parámetros de entrada del algoritmo generador. Una vez configurada la combinación adecuada, el módulo ZoneGenerator se encargará de solicitarle la información de los medidores al módulo DbAdapter, para poder generar las zonas, optimizar los recorridos y actualizar la información otra vez invocando al DbAdapter.

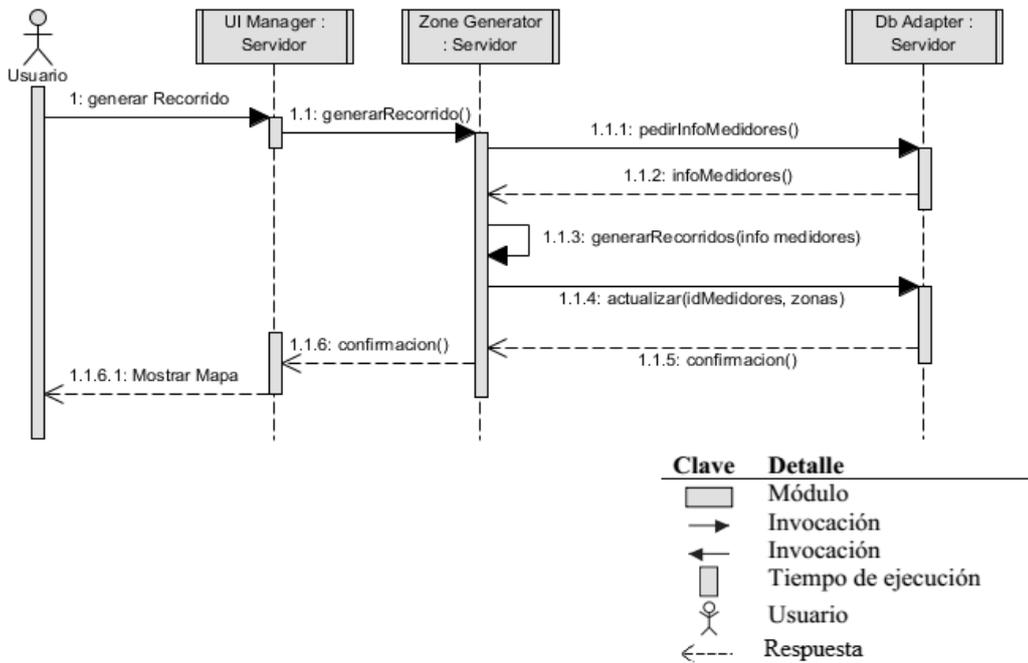


Figura 5. Diagrama de secuencia. El administrador genera los recorridos.

Así, en el caso de estudio se ejecutaría el decorador del kdtree, luego el reubicador de puntos, y finalmente la solución greedy al recorrido interno de cada una de las zonas.

La siguiente figura muestra vistas de la solución propuesta para el módulo servidor web:



Figura 6. Módulo servidor web.

En la figura 7 se pueden apreciar las zonas generadas de una de las pruebas realizadas para verificar su correctitud y/o la necesidad de realizar alguna modificación manual. En caso que el resultado no sea apropiado o se desee experimentar con los distintos filtros o tamaños/cantidad de zonas, se pueden modificar las opciones así hasta quedar conforme con un resultado.



Figura 7. Ensayo de zonificación

### 3. CONCLUSIONES

A partir del análisis realizado en el presente trabajo, se puede afirmar que resulta factible desarrollar un sistema de colecta online de datos por medio del uso de tecnologías móviles e inalámbricas, capaz de generar recorridos optimizados para la colecta de dichos datos utilizando herramientas al alcance de las entidades correspondientes.

Se considera que el sistema desarrollado puede mejorar considerablemente el tiempo total que supone el proceso de colecta de datos a partir tanto de una reducción en el tiempo de los recorridos – dado por la optimización de tales recorridos – como por la practicidad que brinda la validación de los datos de forma online.

Dentro de las propiedades del diseño del sistema se destacan la siguiente como más importante:

- La utilización de árboles KD para la generación de recorridos permitió el procesamiento de la información con un tiempo de ejecución bajo – del orden  $O(n \log n)$  – especialmente para grandes volúmenes de datos, satisfaciendo así los requerimientos de performance de las usinas de luz y gas.
- La aplicación de filtros por etapas para la modificación de la solución parcial concede una gran flexibilidad a la hora de generar recorridos que se adapten a las singularidades de los datos de entrada. Además de dejar campo para incorporación de nuevos filtros que enriquezcan la solución.
- La posibilidad de relocalizar un punto en particular de manera manual luego de haberse generado las zonas, permite personalizar las soluciones para adaptarse a los requerimientos específicos de una situación determinada.
- La puesta en funcionamiento del sistema desarrollado resulta accesible, debido principalmente a que la capacidad de procesamiento necesaria para la computadora que funcione como Módulo Servidor no es elevada para las velocidades de procesamiento media de las computadoras actualmente existentes en el mercado.
- Los costos de mantenimiento en principio son más elevados que el sistema anterior, pero estos se ven despreciados debido a la reducción del tiempo total de facturación. Este hecho es de suma importancia ya que la diferencia entre la fecha de captura de datos y la emisión de la factura significa una desviación importantísima en el costo total de la prestación del servicio. El hecho de que la interfaz de comunicación con el sistema de facturación sea a través de un archivo con un formato específico da la posibilidad de interactuar con cualquier tipo de sistema de facturación siempre y cuando se respete dicha interfaz de comunicación, o que se extienda brevemente la interfaz de tal sistema agregando un módulo adaptador.

Se considera por lo tanto que el sistema desarrollado implica mejoras significantes en el proceso de colecta y facturación.

#### 3.1 Trabajos Futuros

La arquitectura diseñada contempla la integración de diversas tecnologías, pero de todas formas se ha conservado sencilla de modo que se pueda utilizar como base para quienes deseen adicionar funcionalidades o servicios más complejos. Se ha logrado cumplir con requisitos de calidad que abarcan la performance del sistema y las posibilidades del mismo.

Respecto a la seguridad se han contemplado mecanismos para proteger el ingreso al sistema, asegurando el repositorio central de almacenamiento y los canales de comunicación entre los

componentes principales. De todas maneras, este es uno de los aspectos del sistema a mejorar, mediante la inclusión de un módulo encriptador que codifique los mensajes enviados entre el servidor y los distintos clientes.

El módulo de generación de recorridos se diseñó de manera tal que sea simple agregar nueva funcionalidad y que esta pueda ser usada como complemento de lo existente o independiente de este. Uno de los aspectos interesantes a agregar para la generación de los recorridos es tener en cuenta el sentido de las calles. En el caso de estudio planteado esto no se hace indispensable ya que los recorridos son realizados a pie, pero podría ser de gran utilidad para tareas que requieran de un vehículo como medio de transporte.

Una de las mejoras posibles al módulo cliente es la de tomar una fotografía al medidor utilizando la cámara del celular y, a partir de esa imagen, reconocer el valor de la lectura. Esto no solo ahorraría valioso tiempo sino que reduciría aún más la probabilidad de valores mal ingresados.

Adicionalmente se pueden usar imágenes como observaciones, funcionalidad que se torna de bastante utilidad para las zonas rurales o sitios en donde las direcciones no están señalizadas correctamente.

Es posible agregar un sistema de traqueo del recorrido para mantener un registro del tiempo involucrado y alertar a la central en el módulo servidor de cualquier tipo de desvío imprevisto. Esto puede ser de utilidad en caso de extravío o robo del dispositivo móvil así como también para evitar el fraude de los agentes.

#### 4. REFERENCIAS

- [1] Ena, Belén; Delgado, Susana; Ena, Teresa. (2008). *Gestión Administrativa de Personal*. Thomson Editores Paraninfo, S.A. Madrid.
- [2] Rodríguez Valencia, Joaquín. (2006). *Estudio de sistemas y procedimientos administrativos*. International Thomson Editores, S.A. de C.V., México D.F.
- [3] Lema Recalde, Willians Alexander. (2016). "Implementación de una aplicación móvil para la toma de lectura y entrega de planillas del consumo de energía de los medidores de luz para la empresa MARSED S .A.". Tesis de grado. UNIVERSIDAD REGIONAL AUTÓNOMA DE LOS ANDES. Santo Domingo - Ecuador.  
<http://dspace.uniandes.edu.ec/handle/123456789/3790>.
- [4] Taha, Hamdy. (1998). *Investigación de Operaciones: una introducción*. Sexta Ed. Alfaomega Grupo Editor S.A de C. V.
- [5] Stockdale María Lorena (2011). "El problema del viajante: un algoritmo heurístico y una aplicación". Tesis de Licenciatura. Universidad de Buenos Aires.  
[http://cms.dm.uba.ar/academico/carreras/licenciatura/tesis/2011/Stockdale\\_Lorena.pdf](http://cms.dm.uba.ar/academico/carreras/licenciatura/tesis/2011/Stockdale_Lorena.pdf)
- [6] Veiga Losada Francisco José (2013). "El problema del viajante". Universidad de Santiago de Compostela. Máster en Técnicas Estadísticas.  
[http://eio.usc.es/pub/mte/descargas/ProyectosFinMaster/Proyecto\\_774.pdf](http://eio.usc.es/pub/mte/descargas/ProyectosFinMaster/Proyecto_774.pdf)