

**SOCRATES-EC: SISTEMA DE INFORMACIÓN POBLACIONAL BASADO  
EN PROCEDIMIENTOS DE CENSADO**

**DANKO JIMÉNEZ LONDOÑO  
DANIEL ANDRES VALENCIA VALENCIA**



**UNIVERSIDAD DE MANIZALES  
FACULTAD DE INGENIERÍA  
PROGRAMA INGENIERÍA DE SISTEMAS Y TELECOMUNICACIONES  
MANIZALES  
2008**

**SOCRATES-EC: SISTEMA DE INFORMACIÓN POBLACIONAL BASADO EN  
PROCEDIMIENTOS DE CENSADO**

**DANKO JIMENEZ LONDOÑO  
DANIEL ANDRÉS VALENCIA VALENCIA**

Trabajo de Grado presentado como opción parcial para optar  
al título de Ingeniero de Sistemas y Telecomunicaciones

Presidente  
**JAIME UL DARICO CARDONA**  
Ingeniero Electricista

**UNIVERSIDAD DE MANIZALES  
FACULTAD DE INGENIERÍA  
PROGRAMA INGENIERÍA DE SISTEMAS Y TELECOMUNICACIONES  
MANIZALES  
2008**

**PÁGINA DE ACEPTACIÓN**

---

---

---

---

---

---

---

---

**LUIS MARCIAL BERTEL  
JURADO**

---

**JAIME ARIEL RIOS  
JURADO**

---

**ALCIBIADES VALLEJO  
JURADO**

Manizales, 27 de octubre de 2008

## CONTENIDO

	<b>Pág.</b>
INTRODUCCIÓN.....	1
1. DESCRIPCIÓN DEL ÁREA PROBLEMÁTICA.....	2
1.1 DELIMITACIÓN ESPACIO-TEMPORAL.....	3
1.2 DELIMITACIÓN CONCEPTUAL.....	3
2. OBJETIVOS.....	4
2.1 OBJETIVO GENERAL.....	4
2.2 OBJETIVOS ESPECÍFICOS.....	4
3. JUSTIFICACIÓN.....	5
4. MARCO TEÓRICO.....	6
4.1 TIPOS DE DESARROLLOS DE SOFTWARE.....	6
4.2 LOS SISTEMAS DE INFORMACIÓN.....	6
4.3 CRM: MANEJO DE RELACIONES CON EL CLIENTE.....	7
4.4 WEB SERVICES.....	8
4.4.1 Estándares empleados.....	9
4.4.2 Ventajas de los servicios Web.....	9
4.4.3 Inconvenientes de los servicios Web.....	10
4.4.4 Razones para crear servicios Web.....	10
4.5 ANTECEDENTES.....	11
4.5.1. Cómo Funciona un Censo.....	11
4.5.2. Densidad y población de las etnias de Colombia.....	11
4.5.3. Densidad poblacional en áreas asociadas a ecosistemas.....	11
4.5.4. Censo poblacional México 2000.....	12
4.5.5. Así se actualizó el censo electoral.....	12
4.5.6. Estimaciones de población colombiana en el 2005.....	12
4.5.7. Sistemas de información en Bogotá.....	12
4.5.8. Sistema de información regional de Chile en Línea.....	12
4.5.9. Censo Colombia 2005 para discapacitados.....	13
4.5.10. Sistema de información Aquastat.....	13
5. METODOLOGIA.....	14
5.1 PROCEDIMIENTO.....	14
5.2 FASES Y ACTIVIDADES DE DESARROLLO DEL PROYECTO.....	14
5.2.1 Fase de Análisis.....	14
5.2.2 Fase de Diseño.....	14
5.2.3 Fase de implementación y validación.....	17
6. RESULTADOS.....	19
6.1 DESCRIPCIÓN DE RESULTADOS.....	19
6.2 DISCUSIÓN DE RESULTADOS.....	19
7. CONCLUSIONES.....	21
8. RECOMENDACIONES.....	22
BIBLIOGRAFÍA.....	23
ANEXOS.....	26

## LISTA DE ANEXOS

	<b>Pág.</b>
ANEXO A: CONCEPTOS TÉCNICOS RELACIONADOS	26
ANEXO B: DOCUMENTACIÓN TÉCNICA	37
ANEXO C: MANUALES TÉCNICOS Y DE USUARIO	147

## GLOSARIO

**Censo:** es un recuento de población que se realiza cada 10 años con el propósito de conocer las actividades económicas de los habitantes antes, el conocimiento, desplazamiento, nivel de estudios, infraestructura, poder adquisitivo, entre otros, con el fin de hacer finalmente un conteo a nivel nacional que de un resumen del estado actual de ese país o nación.

**CRM:** *Customer relationship management*, se refiere a una estrategia que se enfoca en el cliente, el propósito de esta estrategia es precisamente, la de recopilar la mayor cantidad de información posible sobre los clientes, para poder dar valor a la oferta. La empresa debe trabajar para conocer las necesidades de los mismos y así poder adelantar una oferta y mejorar la calidad en la atención.

## **RESUMEN**

SÓCRATES-EC es un sistema de información desarrollado bajo los modelos Web y de Cliente-Servidor, que permite la obtención, planeación y gestión de información de servicios. Entre ellos tenemos el sector salud, educación, servicios públicos y planeación aplicados sobre la población urbana del municipio de Manizales. Este Proyecto se convierte en una herramienta que ayuda a la toma de decisiones y que se presenta como un modelo que pondrá a disposición información a diferentes organizaciones prestando ésta con exactitud, para permitir desarrollar proyectos en los diferentes campos con la población.

SÓCRATES-EC constituye un propuesta práctica que tiene como principal motivación la posibilidad de administrar la información de todos los miembros de una comunidad específica, sobre la cual se aborda y explora la viabilidad técnica, legal y financiera de crear el Sistema de Información Poblacional basado en una serie de procedimientos y técnicas de censo sistematizado. Este proyecto presenta la posibilidad de brindar un sistema de información de población para todas las entidades públicas o privadas que proveen servicios a la comunidad y así otorgar la oportunidad de generar soluciones adecuadas y proveer mejores servicios tanto a la comunidad en general como a una determinada población que requiere de una necesidad específica.

**PALABRAS CLAVES:** Sistema, Censo, Web, Cliente, Servidor

## **ABSTRACT**

SÓCRATES-EC is a system of information Developed upon the web models and the Client Server, which allows to get, to plan and to promote the services' information. Among them there are the healthy, education, public services and planning sector, which are applied in the urban population of the municipality of Manizales. This project becomes a tool that helps to take decisions, being presented as a model that will give information to different organizations with exactitude that finally will permit to development of projects in different fields with the population selected.

SÓCRATES-EC integrates a practical proposal that has as principal motivation the possibility of administering the information of all the members of the specific community, in terms of exploring the technical legal and financial viability of creating The Population System of Information based on a series of techniques and procedures of systematized census. This project presents the possibility to offer a system of information of population for all the public or private entities that provides services to the community. In this way, it gives the opportunity to generate appropriate solutions and to provide better services not only for the entire community but also for a precise population that requires a specific need.

**KEY WORDS:** System, Census, Web, Client, Server



## RESUMEN ANALÍTICO

Título del Proyecto	SOCRATES-EC: SISTEMA DE INFORMACIÓN POBLACIONAL BASADO EN PROCEDIMIENTOS DE CENSADO
Autor(es)	JIMENEZ, Danko dankojimenez@hotmail.com VALENCIA, Daniel dvalencia@athenea.umanizales.edu.co
Presidente	CARDONA, Jaime Uldarico jucardona@umanizales.edu.co Ingeniero electricista
Tipo de documento	trabajo de grado
Referencia documento	Valencia, Daniel Andres y Jiménez, Danko. Sócrates-EC: Sistema de información poblacional basado en procedimientos de censo. Manizales, 2008, 222 p. Trabajo de grado (Ingeniero de Sistemas y Telecomunicaciones). Universidad de Manizales. Facultad de Ingeniería, Programa Ingeniería de Sistemas y Telecomunicaciones.
Institución	Ingeniería de Sistemas y Telecomunicaciones, facultad de Ingeniería, Universidad de Manizales
Palabras claves	Censo, Web, Cliente, Servidor
Descripción	Sistema de información basado en modelos de Web y cliente servidor que facilita la recolección de datos poblacionales y ayuda a la toma de decisiones a las organizaciones que ofrecen servicios a la comunidad
Fuentes	Formatos de formularios de recolección censal DANE, sistemas de información censales similares en otros países como MIDEPLAN en Chile y Aquastat
Contenido	Introducción y descripción general del problema de falta de información poblacional, metodología de desarrollo de la solución y evaluación final con los resultados correspondientes.

Metodología	<p>Tipo de desarrollo tecnológico en las áreas de informática y telecomunicaciones, se desarrollo en 3 fases correspondientes al análisis, diseño e implementación de la solución. En la primera fase se recogió la información necesaria relacionada con el censo y se analizó para encontrar la información y las técnicas más viables para el nuevo sistema. La segunda fase corresponde al diseño de la solución de acuerdo al análisis realizado, lo que dio como resultado un diseño modular con varios lenguajes involucrados. Y la fase de implementación en donde se crea la solución como tal y se prueba para comprobar su viabilidad y utilidad.</p>
Conclusiones	<p>Sócrates-EC es capaz de satisfacer la necesidad de las organizaciones que proveen servicios a la comunidad de información del tipo censal, pero es necesario que las mismas apliquen estrategias para atraer a los ciudadanos (proveedores primarios de la información) a que alimenten y actualicen la información en el sistema.</p>
Anexos	<p>Se compone de tres anexos esenciales:</p> <ul style="list-style-type: none"> <li>- Anexo A: contiene los conceptos técnicos y teóricos relacionados con el proyecto como lo hecho durante el censo de población 2005.</li> <li>- Anexo B: contiene toda la documentación técnica de análisis y diseño del sistema utilizando UML/UMT.</li> <li>- Anexo C: contiene todos los manuales técnicos y de usuario y documentación adicional para el mantenimiento y mejoras del sistema.</li> </ul>

## **INTRODUCCIÓN**

El presente trabajo tiene como objetivo desarrollar un sistema de información bajo los modelos de Web en su recolección de datos y actualización y de Cliente-Servidor en su administración de información que permita la obtención, planeación y gestión de información de servicios públicos, demográfica, salud y educación sobre la población urbana del municipio de Manizales.

Este trabajo constituye un desarrollo que tiene como principal motivación la posibilidad de administrar la información de todos los miembros de una comunidad específica, sobre la cual se aborda y explora la viabilidad técnica, legal y financiera de crear el Sistema de Información Poblacional basado en una serie de procedimientos de censado sistematizado, la posibilidad de brindar un sistema de información de población para todas las entidades públicas o privadas que proveen servicios a la comunidad, y otorga la capacidad de generar información para la toma de decisiones hacia la prestación de mejores servicios tanto a la comunidad en general como a una determinada población que requiere de una necesidad específica.

## 1. DESCRIPCIÓN DEL ÁREA PROBLEMÁTICA

Durante años el censo poblacional ha sido siempre el mejor método de recolección de información sobre una población específica, ya que muestra una aproximación bastante realista de la situación poblacional, y ayuda a la toma de decisiones en las áreas de planeación, administración pública, socioeconomía, y demás que ayuden a solucionar los problemas de la población.

Observando el anterior sistema de censado poblacional se descubre que por algunos factores carece de exactitud pues este se ejecuta manualmente cada 10 ó 12 años y la información poblacional solo se actualiza en estos mismos intervalos de tiempo. En estos momentos la información está almacenada en medios digitales para proporcionar información válida a las organizaciones públicas y privadas que ofrecen servicios a la comunidad ya que los avances tecnológicos son subestimados y subutilizados cuando éstos tienen que ser verdaderamente útiles y necesarios para el desarrollo de la sociedad.

Actualmente los mecanismos de obtención de información para censado no son prácticos para las organizaciones que prestan servicios a la comunidad ya que cuando dichas organizaciones pueden tener acceso a la información censal, esta ya no muestra la verdadera realidad social ni económica del sector, además, existe una falta de información poblacional y sectorial ocasionando desperdicios en los recursos con los que puede contar el municipio.

Las organizaciones que prestan servicios a la comunidad sufren de una falta de información del tipo poblacional cuando de toma de decisiones se trata, ya que para ejecutar algún proyecto concerniente al municipio deberían tener información actualizada del sector a tratar. Pero las organizaciones solo puede contar con información censal que se actualiza en intervalos de 10 ó 12 años (cuando es proporcionada por el DANE), y que ya pasado el tiempo pierde confiabilidad por lo que los proyectos ejecutados basándose en ese tipo de información no garantizan una verdadera utilidad. Situación que obliga a las organizaciones a hacer costosos estudios independientes para recoger la información que necesitan solo para ese proyecto específico; lo que significa volver a repetir los mismos pasos para cada nuevo proyecto o decisión que se vaya a tomar relacionada con la comunidad.

Como ya se mencionó antes, la información censal del tipo poblacional es la más completa porque muestra una aproximación bastante realista de la situación de la comunidad. Pero como en el país esta información no se actualiza con frecuencia pierde confiabilidad. Para las organizaciones que prestan servicios a la comunidad esto es crítico, ya que no pueden contar con la información que les podría ayudar a una mejor toma de decisiones sobre la comunidad en general.

Viendo este problema que poseen las organizaciones se propone un sistema completamente nuevo que basado en información censal de tipo poblacional, que provea a las organizaciones de información completa y siempre actualizada sobre la comunidad.

Se propone como solución a este problema la creación de un sistema de información que permita acceder a la información de censo poblacional de manera ágil, así como la permanente actualización de dicha información de manera continua. Esta solución está delimitada de la siguiente manera:

### **1.1 DELIMITACIÓN ESPACIO-TEMPORAL**

El proyecto se desarrolló en la ciudad de Manizales y tuvo una duración de aproximadamente 2 años.

### **1.2 DELIMITACIÓN CONCEPTUAL**

El proyecto propuso como solución la implementación de un sistema de información para mejorar los procesos de recolección y administración de la información poblacional en el cual se va a manejar desde los datos básicos de cada persona hasta la información de tipo censal relacionada con vivienda, estrato social, condiciones higiénicas y de salud, servicios públicos, educación y diversidad cultural. Siendo capaz de adaptarse a las necesidades de los diferentes organismos públicos; todo esto teniendo en cuenta como disciplina básica la Informática, en particular lo referente a sistemas de información para toma de decisiones.

## 2. OBJETIVOS

### 2.1 OBJETIVO GENERAL

Desarrollar un sistema de información bajo los modelos de Web y de Cliente-Servidor que permita la obtención, planeación y gestión de información de servicios públicos, demográfica, salud y educación sobre la población urbana del municipio de Manizales.

### 2.2 OBJETIVOS ESPECÍFICOS

- Plantear un modelo de recolección de datos adecuado a nuevas realidades de la gestión de información.
- Realizar un dimensionamiento de *software* y *hardware* requerido para la implantación del sistema.
- Diseñar la base de datos que cumpla con las necesidades de confiabilidad y robustez que requiere el sistema.
- Diseñar los módulos Web y Cliente-Servidor bajo la metodología de diseño orientado a objetos.
- Desarrollar el sistema de gestión de información.

### **3. JUSTIFICACIÓN**

Se expone y justifica la selección del municipio de Manizales como objeto de la presente investigación dadas las condiciones demográficas y político-administrativas que posee, así como las virtudes de un sistema de información radicado en centros comunitarios, que en primera instancia mejore la gestión municipal y que, a la vez, se constituya en un espacio propio y de participación, que sirva como:

- Indicador de gestión municipal.
- Instrumento de planificación municipal.
- Herramienta de cohesión social y real participación ciudadana.

Surge este proyecto de las diferentes inquietudes ciudadanas y académicas sobre las herramientas con las que cuentan actualmente las organizaciones que proveen servicios a la comunidad para consultar de manera real la información relacionada con la ciudadanía, de tal manera que se puedan desarrollar estrategias para determinar dónde, cuándo, cómo y a quiénes beneficia la gestión de la administración y si esta atiende o consulta prioritariamente las necesidades de las comunidades de dicha población.

## **4. MARCO TEÓRICO**

### **4.1 TIPOS DE DESARROLLOS DE SOFTWARE**

En informática se encuentran momentos donde es tanta la información y tecnología que se presenta al dar una solución, que se debe tomar gran parte del tiempo para investigar qué opciones podría ser óptimas, al final no resulta tan bueno, pues cuando se ha tomado la decisión, puede haber una nueva solución en el mercado.

El anterior antecedente podría ilustrarse con el siguiente ejemplo: ¿desarrollar una aplicación únicamente cliente? o ¿desarrollar una aplicación cliente-servidor? ¿Software libre o software propietario?, ¿qué pasará si se elige software libre?, ¿se ganará en recursos económicos o se ganará tiempo si se invierte en software propietario?, así que se entraría en un debate de gran interés y de actualidad y que es necesario abordarlo por variables de tiempo, costo-beneficio, soporte, etc.

En la actualidad hay varias aplicaciones cliente en su mayoría con software propietario, con una muy buena estructura de desarrollo y manejo, que da una buena solución en el tiempo en que fue desarrollado (más de 8 años), pero ¿servirá hoy donde la exigencia es conectividad? ¿Acceso global? ¿o la que se pueda desarrollar ahora servirá a futuro?

Es necesario crear soluciones de software que no se centren solo en resolver el problema actual si no en una aplicación duradera y que se atreva a indagar las posibles situaciones y necesidades que puedan surgir en el futuro, soluciones que sea cual sea el desarrollo se prospecten.

### **4.2 LOS SISTEMAS DE INFORMACIÓN**

Un Sistema de Información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa, es decir, un conjunto formal de procesos que, operando sobre una colección de datos estructurada según las necesidades de la empresa, recopilan, elaboran y distribuyen la información o parte de ella necesaria para las operaciones de dicha empresa y para las actividades de dirección y control correspondientes para desempeñar sus actividades de acuerdo a su estrategia de negocio.



### 4.3 CRM: MANEJO DE RELACIONES CON EL CLIENTE

CRM (del acrónimo inglés "*Customer Relationship Management*") es "una estrategia que permite a las empresas identificar, atraer y retener a sus clientes, además de ayudarles a incrementar la satisfacción de éstos y a optimizar así la rentabilidad de sus negocios. Hablamos, por tanto, de CRM como estrategia, lo que implica no sólo disponer del software adecuado que te permita gestionar las relaciones con los clientes, sino que además, supone un cambio en los procesos de la empresa y la inclusión de todos los empleados de la misma para que esta estrategia tenga éxito"<sup>1</sup>

CRM es parte de una estrategia de negocio centrada en el cliente. Una parte fundamental de su idea es, precisamente, la de recopilar la mayor cantidad de información posible sobre los clientes, para poder dar valor a la oferta. La empresa debe trabajar para conocer las necesidades de los mismos y así poder adelantar una oferta y mejorar la calidad en la atención. Por lo tanto, el nombre CRM hace referencia a una estrategia de negocio basada en los clientes, pero también a los sistemas informáticos que dan soporte a esta estrategia.

Desde un punto de vista metodológico, los pasos a seguir son básicamente cuatro:

En primer lugar, identificar a los clientes. Es preciso que podamos saber siempre quienes son, entren a través del canal que entren, para que sean siempre vistos como el mismo cliente a lo largo de todas las transacciones e interacciones que realicen a lo largo del tiempo. Debemos ser capaces de almacenar toda la información de un cliente sin que se pierda nada si queremos hacer posible lo que viene después. Segundo, diferenciarlos. Ser capaz de adscribir a esos consumidores que tenemos previamente identificados a grupos definidos y caracterizados por pautas comunes, por el tipo de necesidades que plantean a la compañía, por el valor que tienen para nosotros.

Tanto si me acerco a la compañía a través de su página web como si viene a mi casa un operario de servicio técnico, aunque sea de una empresa subcontratada, yo, como cliente, debo ver que se me conoce, y si soy o me considero un cliente muy bueno, que se sepa y se me trate como tal.

En tercer lugar, interactuar con ellos. Implica mantener contactos con esos consumidores basados en la información que tenemos de ellos y de sus necesidades, registrar esos contactos como fuentes adicionales de información. Estos contactos deben hacerse únicamente cuando con ello proporcionemos valor al cliente, bien con ofertas que podamos suponer que le interesan o con peticiones de información que le

---

<sup>1</sup> MICROSOFT ESPAÑA. Guía CRM [En línea]. Madrid. Fecha de consulta 18/05/2008. disponible en: [http://www.microsoft.com/spain/empresas/soluciones/guia\\_crm.msp](http://www.microsoft.com/spain/empresas/soluciones/guia_crm.msp)

parezcan relevantes. Se acabó la era del llamado database marketing, los envíos masivos a todos los clientes con la misma oferta. Con esto, si se hace bien, deberíamos no sólo reducir costes de envío y demás, sino además obtener una eficiencia muy superior. Por visualizar un ejemplo, piense en lo que ocurre cuando entra en una librería online y la página no sólo le saluda por su nombre, sino que además le ofrece recomendaciones que tienden a ajustarse muy bien a lo que usted busca. Esa prestación, está basada en mis compras anteriores, y en lo que la librería deduce a base de compararme con otros clientes parecidos a mi, mediante técnicas conocidas como collaborative filtering. Para ello, las dos fases anteriores, identificar al cliente y diferenciarlo, tienen que estar perfectamente desarrolladas. Finalmente, en cuarto lugar, debemos adaptar nuestro producto o servicio a esos clientes, para cubrir sus necesidades de una manera aún más eficiente. Este cuarto paso, la adaptación o, en palabra proveniente del término inglés, customización, es el paso más difícil en toda estrategia CRM, y requiere una gran integración y buen funcionamiento de las tres etapas anteriores. Se trata de, una vez que conocemos al cliente y que le hemos oído, hacer que efectivamente nuestro producto o servicio se ajuste a lo que ese cliente nos estaba pidiendo, dentro de los límites razonables<sup>2</sup>.

Uno de los mayores problemas para que las empresas exploten un producto orientado a CRM es el alto costo de estos comerciales, las licencias adicionales como un la de un sistema operativo y más aun el alto costo de la su implantación y puesta en marcha.

#### **4.4 WEB SERVICES**

Los *Web Services* (Servicios Web) son aplicaciones basadas en Web que utilizan una serie de estándares y protocolos abiertos basados en XML para intercambiar información con clientes sin importar la plataforma operativa en la que funcionen las distintas aplicaciones<sup>3</sup>.

Para ser mas exactos un *Web Service* es un conjunto de protocolos y estándares que sirven para intercambiar datos entre distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y que ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la

---

<sup>2</sup> DANS, Enrique. CRM, Customer Relationship Management [En Línea]. Madrid (España). International Excellence. 28/01/2001. Fecha de consulta: 20/07/2008. disponible en: [http://profesores.ie.edu/enrique\\_dans/download/crm.pdf](http://profesores.ie.edu/enrique_dans/download/crm.pdf)

<sup>3</sup> World Wide Web Consortium. Web Services [en línea]. Fecha de consulta. 25/08/2008. disponible en: <http://www.w3.org/2002/ws/>

adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares.

#### **4.4.1 Estándares empleados.** Los estándares son:

- Web Services Protocol Stack: Así se denomina al conjunto de servicios y protocolos de los servicios Web.
- XML (Extensible Markup Language): Es el formato estándar para los datos que se vayan a intercambiar.
- SOAP (Simple Object Access Protocol) o XML-RPC (XML Remote Procedure Call): Protocolos sobre los que se establece el intercambio.
- Otros protocolos: los datos en XML también pueden enviarse de una aplicación a otra mediante protocolos normales como HTTP (Hypertext Transfer Protocol), FTP (File Transfer Protocol), o SMTP (Simple Mail Transfer Protocol).
- WSDL (Web Services Description Languages): Es el lenguaje de la interfaz pública para los servicios Web. Es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios Web.
- UDDI (Universal Description, Discovery and Integration): Protocolo para publicar la información de los servicios Web. Permite comprobar qué servicios web están disponibles.
- WS-Security (Web Service Security): Protocolo de seguridad aceptado como estándar por OASIS (Organization for the Advancement of Structured Information Standards). Garantiza la autenticación de los actores y la confidencialidad de los mensajes enviados...

#### **4.4.2 Ventajas de los servicios Web.** Estas son las ventajas de los servicios Web:

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.

- Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar y abiertos. Las especificaciones son gestionadas por una organización abierta, la W3C, por tanto no hay secretismos por intereses particulares de fabricantes concretos y se garantiza la plena interoperabilidad entre aplicaciones.

**4.4.3 Inconvenientes de los servicios Web.** los servicios Web poseen ciertos inconvenientes y desventajas, las cuales son:

- Para realizar transacciones no pueden compararse en su grado de desarrollo con los estándares abiertos de computación distribuida como CORBA (Common Object Request Broker Architecture).
- Su rendimiento es bajo si se compara con otros modelos de computación distribuida, tales como RMI (Remote Method Invocation), CORBA, o DCOM (Distributed Component Object Model). Es uno de los inconvenientes derivados de adoptar un formato basado en texto. Y es que entre los objetivos de XML no se encuentra la concisión ni la eficacia de procesamiento.
- Al apoyarse en HTTP, pueden esquivar medidas de seguridad basadas en firewall cuyas reglas tratan de bloquear o auditar la comunicación entre programas a ambos lados de la barrera.

**4.4.4 Razones para crear servicios Web.** La tecnología de servicios Web es revolucionaria en si por su capacidad de funcionar independiente del lenguaje de programación en el que este construido y de la plataforma de funcionamiento, .as razones para crear servicios usando esta tecnología son:

- La principal razón para usar servicios Web es que se basan en HTTP sobre TCP (*Transmission Control Protocol*) en el puerto 80. Dado que las organizaciones protegen sus redes mediante *firewalls* -que filtran y bloquean gran parte del tráfico de Internet-, cierran casi todos los puertos TCP salvo el 80, que es, precisamente, el que usan los navegadores. Los servicios Web utilizan este puerto, por la simple razón de que no resultan bloqueados.
- Otra razón es que, antes de que existiera SOAP, no había buenas interfaces para acceder a las funcionalidades de otros ordenadores en red. Las que había eran *ad hoc* y poco conocidas, tales como EDI (Electronic Data *Interchange*), RPC, u otras APIs (*Application Programming Interface*).
- Una tercera razón por la que los servicios Web son muy prácticos es que pueden aportar gran independencia entre la aplicación que usa el servicio Web y el propio servicio. De esta forma, los cambios a lo largo del tiempo en uno no deben afectar al otro. Esta flexibilidad será cada vez más

importante, dado que la tendencia a construir grandes aplicaciones a partir de componentes distribuidos más pequeños es cada día más utilizada.

Los Servicios Web permitirán a los responsables de los sitios Web departamentales y/o de los centros, e incluso a los profesores con páginas personales, acceder a datos corporativos para mostrarlos en sus páginas Web con su estilo propio y lo que es más importante, sin tener que duplicar información. Se espera que para los próximos años mejoren la calidad y cantidad de servicios ofrecidos basados en los nuevos estándares.

## 4.5 ANTECEDENTES

**4.5.1. Cómo Funciona un Censo<sup>4</sup>.** Explicación de cómo funciona un censo en los Estados Unidos, reuniendo información valiosa en cuanto a la operatividad de este procedimiento, además de explicaciones básicas de lo que es en sí un censo poblacional, algo de historia del censo en Estados Unidos y explicaciones básicas de cómo recoger la información de las diferentes clases sociales, etnias y otros grupos especiales de la población estadounidense.

**4.5.2. Densidad y población de las etnias de Colombia<sup>5</sup>.** Artículo sobre las etnias de Colombia referentes a información poblacional, en el cual se explica la conformación de la población colombiana (mayoritariamente mestiza), de acuerdo a los datos recogidos en el censo poblacional hecho en el año 1996, y muestra datos generales sobre nuestro país entre los que se incluyen zona geográfica, población, diversidad ambiental, sistema político, entre otros.

**4.5.3. Densidad poblacional en áreas asociadas a ecosistemas<sup>6</sup>.** En esta dirección se encuentra un documento que explica algunos procedimientos estadísticos utilizados para el cálculo de resultados de censo poblacional teniendo en cuenta la región geográfica relacionada con la población muestreada, este trabajo lo realizó el Instituto Humboldt para un Sistema de Información Geográfica (SIG) sobre la población y los diferentes ecosistemas.

---

<sup>4</sup> HOWSTUFFWORKS, INC. How The Census Works. [En Línea]. Atlanta (USA). Fecha de Consulta. Abril 17 de 2007. Disponible en: <http://people.howstuffworks.com/census.htm/printable>

<sup>5</sup> Etnias de Colombia. Densidad y población de las etnias de Colombia [en línea] fecha de consulta. Abril 18 de 2007. disponible en: <http://www.etniasdecolombia.org/colombia.asp>

<sup>6</sup> Instituto Humboldt. Densidad poblacional en áreas asociadas a ecosistemas [En Línea]. Fecha de Consulta. Abril 18 de 2007. Disponible en: [http://www.humboldt.org.co/humboldt/homeFiles/sig/AnexoIII/Hojametodologicapoblacion\\_JCB%20\(2\).pdf](http://www.humboldt.org.co/humboldt/homeFiles/sig/AnexoIII/Hojametodologicapoblacion_JCB%20(2).pdf)

**4.5.4. Censo poblacional México 2000<sup>7</sup>.** Página que muestra en resumen la información recogida del censo poblacional hecho en México en el año 2000, adicionalmente hace comparaciones históricas con datos de los años anteriores, muestra las variaciones en los diferentes campos como política, zonas habitables, entidades, etc., también muestra datos sobre las tendencias de la comunidad sobre el empleo y algunas proyecciones a futuro.

**4.5.5. Así se actualizó el censo electoral<sup>8</sup>.** Explicación de cómo se actualizó el censo electoral de 2003, tomando en cuenta como base la votación para el referendo; este documento explica brevemente la depuración de cédulas registradas respecto a personas fallecidas, la exclusión de miembros de la fuerza pública de la votación y cédulas no reclamadas. Esta referencia es útil para mostrar el estado de la población habilitada para votar y de la cual depende este sistema.

**4.5.6. Estimaciones de población colombiana en el 2005<sup>9</sup>.** Artículo de prensa en el cual se muestran las estimaciones sobre la población colombiana en 2005, de la cual actualmente se calcula que hay 43,5 millones de habitantes (Julio 13 de 2004), entre los datos mostrados aparece información sobre la población juvenil y de territorios, además resalta la necesidad de tener el censo poblacional de 2005.

**4.5.7. Sistemas de información en Bogotá<sup>10</sup>.** Artículo sobre sistemas de información para servicios en Bogotá que resalta el hecho de que para proveer mejores servicios en la ciudad, se debe tener más información referente a la población y a su situación actual, señala el hecho de tener una buena comunicación con la gente entre otros esquemas.

**4.5.8. Sistema de información regional de Chile en Línea<sup>11</sup>.** Página oficial del sistema de información regional de Chile; aquí se encuentra información

---

<sup>7</sup> RIVEROLL, Pablo. Censo poblacional México 2000 [En Línea]. Fecha de Consulta. 18/04/2007. Disponible en: <http://jherrerapena.tripod.com/politica/rive.html>

<sup>8</sup> Registraduría nacional del estado civil [En Línea]. Fecha de Consulta. 18/04/2007. Disponible en: [www.registraduria.gov.co/docs/com\\_prensa\\_04\\_2003.pdf](http://www.registraduria.gov.co/docs/com_prensa_04_2003.pdf)

<sup>9</sup> Periodico El Pais [En Línea]. Cali. Fecha de Consulta. 18/04/2007. Disponible en: <http://elpais-cali.terra.com.co/paionline/notas/Julio132004/BREVE7.html>

<sup>10</sup> Alcaldía de Bogotá. Sistemas de Información en Bogotá [En Línea]. Bogotá. Fecha de Consulta. 18/04/2007. Disponible en: <http://www.bogota.gov.co/bogota/decide.php?patron=1.0309>

<sup>11</sup> Infopais-Mideplan. Sistema de información regional de Chile [En Línea]. Santiago (Chile). Fecha de Consulta. 18/04/2007. Disponible en: <http://infopais.mideplan.cl/>

estadística detallada sobre Chile y su población en relación con su economía, agricultura, entre otros aspectos importantes.

**4.5.9. Censo Colombia 2005 para discapacitados<sup>12</sup>.** Recopilación de información del Censo Colombia 2005 referente a discapacitados; aquí se explica la importancia de la población discapacitada en la recolección de la información censal y las formas de vigilancia de dichos procesos, también se muestran artículos que reseñan encuestas hechas sobre la población discapacitada y enlaces relacionados con este tipo de población.

**4.5.10. Sistema de información Aquastat<sup>13</sup>.** Sistema de información en línea sobre el uso del agua en la agricultura y el medio rural, desarrollado por la División de Fomento de Tierras y Aguas de la FAO. Esta página provee a los usuarios información sobre la gestión de los recursos acuíferos en relación con la agricultura en las zonas rurales, centrado principalmente en los países en vía de desarrollo.

---

<sup>12</sup> Disabled Peoples' Internacional. Censo Colombia 2005 para discapacitados [En Línea]. Mount Pearl (Canada). Fecha de Consulta. 18/04/2007. Disponible en: <http://www.dpi.org/sp/resources/documents/disnnetmar31.doc>

<sup>13</sup> FAO. Sistema de información Aquastat [En Línea]. Roma (Italia). Fecha de Consulta. 18/04/2007. Disponible en: <http://www.fao.org/ag/agl/aglw/aquastat/main/indexesp.stm>

## 5. METODOLOGÍA

### 5.1 TIPO DE TRABAJO

el presente proyecto corresponde a un Desarrollo tecnológico, correspondiente al Campo de conocimiento: Electrónica, Telecomunicaciones e Informática, y específicamente al Área de conocimiento: Soluciones Empresariales.

### 5.2 PROCEDIMIENTO

**5.2.1 Fase de Análisis.** Esta fase se compuso de los siguientes eventos:

- Consulta de formas de recolección y censado.

Durante el proceso de recolección de información se recurrió al Departamento Administrativo Nacional de Estadística (DANE) para obtener información base sobre el manejo del proceso censal y se participó activamente en el proceso de censo nacional de población y vivienda del año 2005, durante esta temporada se recogió todo tipo de información relacionada con formatos de cuestionarios, técnicas de recolección y procesamiento de datos tomando como base el trabajo hecho en las áreas urbana y rural de los municipios de Chinchiná y Supía (Caldas).

- Análisis de efectividad de dichos métodos y resultados.

El análisis de la información recogida tomo más tiempo del calculado ya que la cantidad de información que debía obtenerse durante el censo del año 2005 aún no se había terminado de recolectar, pero se logró definir las técnicas y procedimientos de recolección de la información, las herramientas utilizadas y la metodología de procesamiento de la información.

Se descubrió que la mayoría de las personas fueron censadas con un formulario corto de encuesta donde se recogieron los datos básicos y que solo una décima parte de la población fue censada con un formulario extendido, el cual fue ajustado para el resto de la población general utilizando herramientas estadísticas complementarias.

**5.2.2 Fase de Diseño.** Esta fase se compuso de los siguientes eventos:

- Determinar el motor de base de datos adecuado para el nuevo sistema.



Como se puede notar, se determinó el motor de la base de datos como PostgreSQL siguiendo estos cinco parámetros:

- a) *Disponibilidad en plataformas múltiples*
  - b) *Herramientas profesionales del desarrollo y de la administración*
  - c) *Ayuda para lenguajes de programación múltiples*
  - d) *Ayuda técnica*
  - e) *Apoyo a los requisitos de la empresa*
- Diseñar la base de datos y el modelo entidad-relación de acuerdo con los datos recogidos en la fase de análisis.

El diseño de la base de datos ha sido cuidadosamente hecho de tal manera que se permita almacenar información histórica; se descartaron datos que no aportaran información significativa a las organizaciones que proporcionan servicios a la comunidad; también se descartó información que el usuario prefiere, por razones naturales, mantener privada.

El tipo de información censal que fue incluido en la base de datos permite una comprensión realista de la situación del momento ya que su diseño permite realizar un seguimiento sobre los cambios en la población.

Un diseño eficaz de la base de datos es crucial para garantizar un tiempo de respuesta rápido por parte del cliente. Para diseñar una base de datos adecuada, es necesario distribuir apropiadamente los datos entre la base de datos y el sistema de archivos, se normalizó de que las tablas no requieran combinaciones excesivas para responder a las consultas, se crearon índices suficientes para optimizar las consultas más frecuentes sin el uso de una cantidad de espacio de disco excesivo y así poder usar procedimientos almacenados para llevar a cabo operaciones de base de datos frecuentes o complejas.

- Construcción de la base de datos utilizando el gestor de base de datos seleccionado.

Durante el diseño de la base de datos se tomó muy en cuenta la cantidad de información histórica a manejar (muchas de esta redundante), por lo que la escogencia del gestor de base de datos fue fundamental para el manejo de este tipo de información. Inicialmente se construyó en la herramienta *Microsoft Access* para poder visualizar y probar el modelo de la base de datos durante las primeras etapas del diseño; una vez corregidos los problemas grandes del primer modelo relacional de la base de datos, se comenzó la construcción de la

base de datos en PostgreSQL y se refinaron los últimos detalles de diseño en este proceso.

- Diseñar y construir el módulo central cliente-servidor que consiste en funciones básicas de administración de base de datos, generador de consultas, estadísticas, etc.

Durante el proceso de diseño del módulo cliente-servidor, se tuvieron en cuenta las siguientes características:

- Seguridad del sistema.
- Accesibilidad desde múltiples plataformas.
- Facilidad de uso.

Partiendo de esta premisa se escogió el lenguaje de programación JAVA para el desarrollo de este módulo.

- Diseñar y construir el módulo de recolección de información y actualización vía Web.

Durante este paso se diseñó el sitio Web como formulario dinámico con la ayuda de la herramienta *Macromedia Dreamweaver* y el lenguaje de programación utilizado fue PHP.

Este tiempo fue complementado en la realización de los formularios y la base de datos. Este módulo contiene las especificaciones de la base de datos necesarias para traducir las solicitudes enviadas desde el cliente, a un formato que sea reconocido por dicha base de datos, además contiene toda la información, estructura, variables y llamadas a funciones, necesarias para comunicarse con la base de datos.

Estas son algunas de las directrices que se tuvo en cuenta para el diseño de Interfaces Web:

- Planteamiento general.
- Organización de la información.
- Facilidad de navegación.
- Agilidad en las tareas.
- Comodidad en la lectura.
- Consistencia.
- Accesibilidad.

El diseño de la interfaz de usuario debe encontrar un justo equilibrio entre los tipos de exploradores admitidos, la estética de la aplicación y su facilidad de uso. Al diseñar una interfaz para usar en Internet, es necesario tener en cuenta la cantidad de tiempo que una página tardará en descargarse a través de la conexión telefónica que es más lenta que de una conexión de alta velocidad.

Se elaboró un plan de acceso a la base de datos en el que se especificó qué páginas HTML requerían contenido de base de datos, proporcionando ayuda para poder determinar la información que se utilizaría en las páginas que son totalmente dinámicas.

**5.2.3 Fase de implementación y validación.** Esta fase se compuso de los siguientes eventos:

- Implementar el sistema en las herramientas adecuadas según el análisis y diseño.

La implementación del sistema presento muchos problemas técnicos que retrasaron su progreso, la interacción de las dos partes del sistema es clave para su correcto funcionamiento, la parte compuesta por el modulo de recolección se conecta directamente con la base de datos, pero el modulo de administración de consultas basado en JAVA tuvo que requerir de la tecnología de *web services* para contar con esa posibilidad.

Para poder manejar la completa funcionalidad del sistema de administración y consultas se creo un servidor *Web Service* que pudiera conectar la base de datos con el modulo de administración y consultas, este servidor se desarrollo en lenguaje PHP con la librería nuSOAP versión 0.74, esta librería esta basada en el estándar de *Web Services* SOAP 1.1 el cual permite ciertas funcionalidades automáticas que otros paquetes de servicios web para PHP no ofrecen, como la generación automática del archivo de descripción de servicios (*Web Service Description Language*, WSDL), que es clave para la utilización del servicio Web con tras aplicaciones sin importar el lenguaje.

Del lado del modulo java se recurrió al *Java Web Services Development Pack* (JWSDP). Que permite tanto crear servicios Web como clientes para los mismos bajo los estándares SOAP 1.1 y 1.2.

Una vez solucionados los problemas de compatibilidad encontrados en dichos paquetes de desarrollo se procedió a hacer pruebas preliminares del sistema mientras se termino de desarrollar los 2 módulos.

- Validar el sistema.

La validación del sistema se ejecuto en una fase de pruebas en a cual se introdujeron datos de prueba a la base de datos mediante el sistema de

recolección de información vía Web y se probó el sistema de consultas con la aplicación JAVA, probando ser todo un éxito en todas las características y funciones del sistema.

## **6. RESULTADOS**

### **6.1 DESCRIPCIÓN DE RESULTADOS**

Sócrates-EC es un software diseñado en tres partes. La parte principal se compone de la base de datos hecha en PostgreSQL, la cual contiene toda la información poblacional más los datos de acceso a dicha información por las organizaciones que proveen servicios a la comunidad. La parte de recolección y actualización de información que esta construida en PHP y que permite como su nombre lo dice la recolección de la información poblacional por parte de las personas cabeza de hogar o miembros autorizados. Y la parte administración de consultas hecha como una aplicación JAVA que permite realizar casi cualquier tipo de consulta sobre la información poblacional almacenada en la base de datos.

### **6.2 DISCUSIÓN DE RESULTADOS**

El sistema de información poblacional basado en procedimientos de censo Sócrates-EC ha logrado garantizar la coherencia, la continuidad y la homogeneidad de la información censal en los aspectos relacionados con facilidades de recolección de información y administración de la misma, y adicionalmente logra de manera intuitiva facilitar el buen uso de dicha información por las organizaciones que ofrecen servicios a la comunidad, como identidades gubernamentales y de servicios públicos.

Se ha logrado desarrollar un sistema que integra la recolección de información censal en una manera que permite rapidez, sencillez y eficiencia de manejo de la información integrando también la posibilidad de utilizar la información recogida de forma histórica ya que permite actualizaciones constantes que pueden ser aplicadas mediante estrategias de recolección de información como CRM y demás.

La administración de la información almacenada es uno de los puntos clave de este sistema ya que el sistema de administración de consultas fue diseñado específicamente para servir como apoyo a la toma de decisiones y la información almacenada puede ser cruzada fácilmente con otros sistemas de información basados en tecnologías de bases de datos, módulos CRM y grandes sistemas de toma de decisiones sobre información de población.

Desde el punto de vista tecnológico, se ha logrado construir un sistema a partir de tecnologías de punta como servicios Web y XML, que permiten la intercomunicación entre los 2 módulos principales del sistema los cuales fueron

construidos en lenguajes completamente diferentes (PHP y JAVA) y aun así interactuar con efectividad y seguridad para lograr los objetivos propuestos con este proyecto y facilitar el trabajo de recolección, actualización y administración de información poblacional del tipo censal para su uso por parte de las organizaciones que proveen servicios a la comunidad en general.

## 7. CONCLUSIONES

- Las formas tradicionales de censo poblacional aunque se han actualizado gracias a la tecnología, aun no suplen las necesidades de las organizaciones que proveen servicios a la comunidad precisamente por el acceso restringido a esa información y por la desactualización rápida de la misma por el intervalo prolongado (en años) en el que se hace la recolección de la información. Este sistema (Sócrates-EC) es capaz de satisfacer la necesidad de las organizaciones que proveen servicios a la comunidad de dicha información, pero es necesario que las mismas apliquen estrategias para atraer a los ciudadanos (proveedores primarios de la información) a que alimenten y actualicen la información en el sistema.
- Es necesario que tecnologías como los servicios Web estén más estandarizadas sobre todo del lado de lenguajes diseñados para Web como PHP ya que ciertas características típicas de los servicios Web como la transferencia segura aun no están completamente soportadas a diferencia de los servicios Web bajo JAVA. Es decir, el *kit* de creación de *Web services* trabaja con el estándar SOAP 1.1 el cual posee todas las características necesarias para un servicio Web estable y bien implementado pero es solo parcialmente compatible con el estándar SOAP 1.2 el cual es soportado por JAVA.
- La actualización de las tablas de parámetros en la base de datos, generan las opciones de recolección del módulo Web, pero *no* las de consulta del módulo Java. Esto se debe a que el modelo utilizado para el desarrollo del sistema es un modelo estático basado en el esquema de información y manejo del censo durante el periodo 2005-2006.
- El sistema de información poblacional basado en procedimientos de censo Sócrates-EC puede ser considerado viable desde el punto de vista económico, siempre y cuando se tenga en cuenta que tiene que estar soportado por el marco legal existente en el país y que la información que pueda contener el sistema una vez puesta en servicio, sea utilizada únicamente como información estadística que ayude a la toma de decisiones que beneficien a la comunidad.

## 8. RECOMENDACIONES

- Proyectos de grado que hagan parte de procesos de desarrollo que beneficien a la comunidad en general y que impulsen nuevas tecnologías como es el caso de Sócrates-EC y otros tantos, no deben quedar se en la versión en la cual fueron entregados, sino ser parte de procesos permanentes de actualización, siendo abordados por otros estudiantes o ingenieros que apoyen estos procesos de cambio a nivel de Software, y de esta manera lograr que el producto sea competitivo y que realmente cumpla el objetivo y servicio por el cual fue creado.
- Aunque Sócrates-EC fue pensado desde un principio para proporcionar sencillez de uso tanto a los miembros de la población que alimentan el sistema como a las organizaciones que se pueden beneficiar de este, el ambiente de interacción con el usuario aun puede mejorarse, como referencia se tomó el libro de Alan Cooper *PRESOS DE LA TECNOLOGIA: Por qué los productos tecnológicos nos vuelven locos y como recuperar la cordura* el cual no solo muestra algunas técnicas básicas de diseño de interacción con el usuario si no que además muestra los beneficios de estas técnicas desde el punto de vista de los negocios.
- Se debe tener en cuenta para futuras experiencias en el desarrollo de temas que conciernan a la planeación y desarrollo de un sistema de información que se base en el censo poblacional, que la actualización de los parámetros se deba hacer de forma dinámica ya que para este caso específico se hizo de forma estática, ya que el desarrollo en el censo anterior no permite cambios en la estructura de parametrización de las tablas. Esta ya estaba diseñada de una forma específica para la recolección de la información, por esto se recomienda que la aplicación en futuro se desarrolle en forma de modelo dinámico de datos para permitir que los cambios en la estructura de las tablas se haga mas fácil para el administrador de la aplicación.
- El modo de consultas avanzadas del modulo de administración de consultas de este sistema requiere de un operador con conocimientos avanzados de lenguaje SQL y de la estructura interna de la base de datos, por lo cual NO es recomendable usar este modo excepto por personal entrenado para la tarea.



## BIBLIOGRAFÍA

COOPER, Alan. Presos de la Tecnología: porque los productos tecnológicos nos vuelven locos y como recuperar la cordura. México : Prentice Hall, 2001. 280 p.

GIRALDO, Juan Pablo. Modelo Orientado a objetos OMT + UML, 31 de marzo 2005 Facultad de Ingeniería Universidad de Manizales

PRESSMAN, Roger. Ingeniería del Software un Enfoque Práctico, 5 ed, Madrid : McGraw-Hill, 2002. 601 p

SENN, James. Análisis y Diseño de Sistemas de Información, 2 ed, México : McGraw-Hill, 1992. 942 p.

WHITTEN L, Jeffrey and BENTLEY, Lonnie and Ba. Análisis y Diseño de Sistemas Informáticos, México : McGraw-Hill - Hill, 2000. 907 p

## REFERENCIAS ELECTRONICAS

2bits.com, Inc. Benchmarking PostgreSQL vs MySQL performance using Drupal 5.x. [En Línea]. Ontario (Canada). 2Bits.com, Inc. Fecha de consulta: 20/07/2008. disponible en: <http://2bits.com/articles/benchmarking-postgresql-vs-mysql-performance-using-drupal-5x.html>

DANE. Censo 2005 [en línea]. Bogotá (Colombia). Departamento Administrativo Nacional de Estadística. Fecha de consulta: 10/12/2007. Disponible en: <http://www.dane.gov.co/censo/>

DANS, Enrique. CRM, Customer Relationship Management [En Línea]. Madrid (España). International Excellence. 28/01/2001. Fecha de consulta: 20/07/2008. disponible en: [http://profesores.ie.edu/enrique\\_dans/download/crm.pdf](http://profesores.ie.edu/enrique_dans/download/crm.pdf)

EnterpriseDB: PostgreSQL vs. MySQL. [En línea]. Edison, New Jersey (USA). EnterpriseDB Corp. Fecha de consulta: 20/07/2008. disponible en: [http://downloads.enterprisedb.com/whitepapers/PGvsMySQL\\_LowRes.pdf](http://downloads.enterprisedb.com/whitepapers/PGvsMySQL_LowRes.pdf)

HOWSTUFFWORKS, INC. How The Census Works. [En Línea]. Atlanta (USA). Fecha de Consulta. 17/04/2007. Disponible en: <http://people.howstuffworks.com/census.htm/printable>

PECOS, Daniel. PostgreSQL vs. MySQL [En Línea]. Madrid. Net Pecos. Fecha de consulta 25/10/2007. disponible en: [http://www.netpecos.org/docs/mysql\\_postgres/index.html](http://www.netpecos.org/docs/mysql_postgres/index.html)

SUN Microsystems. Metro Web Services Overview [En Línea]. Fecha de consulta. 12/02/2007. Disponible en: <http://java.sun.com/webservices/>

W3 Schools. Web Services Tutorial [en línea]. Noruega. Refsnes Data Inc. Fecha de consulta: 02/11/2007. Disponible en: <http://www.w3schools.com/webservices/default.asp>

World Wide Web Consortium [en línea]. Fecha de consulta. 25/08/2008. disponible en: <http://www.w3.org/2002/ws/>

## **ANEXOS**

## ANEXO A CONCEPTOS TÉCNICOS RELACIONADOS

CONTENIDO	PAG
1. SOBRE EL CENSO DE POBLACIÓN Y VIVIENDA 2005	26
2. DEFINICIÓN DE CONCEPTOS CENSALES	28
3. COMPARACIÓN MYSQL Y POSTGRESQL	31

### 1. SOBRE EL CENSO DE POBLACIÓN Y VIVIENDA 2005<sup>14</sup>

En materia censal el país presenta un notorio estancamiento; el último Censo Nacional de Población y Vivienda data de 1993; en lo económico de 1990 y en lo concerniente al sector agropecuario de 1970. Este estancamiento ha sido suplido parcialmente por un conjunto de encuestas muestrales cuyos marcos estadísticos se encuentran afectados por una reconocida obsolescencia, con detrimento de su representatividad sectorial y territorial. Los requerimientos de información actualizada, suficiente y confiable en los campos demográfico, económico, social, agropecuario y territorial son cada vez más apremiantes para la planeación económica y social en todos los niveles político administrativos. La oportunidad de realizar el Censo Nacional de Población y Vivienda en el 2005, aprobado por el CONPES a finales del año 2004, condujo al gobierno nacional a aprovechar esta oportunidad excepcional para darle un renovado impulso al Sistema de Información Básica en cabeza del DANE, ampliando el alcance del proyecto original en el sentido de un Censo General que incluyera, además de los temas de población y vivienda, los relativos al sector económico y agropecuario, sin incrementar los costos presupuestados.

El Censo General 2005 sobrepasa los objetivos y alcances del censo tradicional de población y vivienda con la inclusión de importantes ajustes metodológicos, relacionados con el período de recolección, el uso de cuestionario ampliado en una muestra censal, el aprovechamiento de tecnología de punta para la captura electrónica de datos *in situ*, el empleo de cartografía de precisión, la publicación

---

<sup>14</sup> DANE. Censo 2005 [en línea]. Bogota (Colombia). Departamento Administrativo Nacional de Estadística. Fecha de consulta: Diciembre 10 de 2007. Disponible en: <http://www.dane.gov.co/censo/>

de resultados oportunamente y la construcción de nuevos marcos estadísticos, georreferenciados para hacer investigaciones estadísticas de profundidad por censos, registros administrativos o a través de muestras probabilísticas como la muestra continua en hogares, establecimientos de industria, comercio, servicios y en actividades agropecuarias.

## 2. DEFINICIÓN DE CONCEPTOS CENSALES

Para la adecuada realización de los formularios en general se debe manejar una serie de conceptos con el fin de garantizar la calidad de la información y su adecuado uso por parte de los diferentes usuarios.

Como el Censo de Población y Vivienda tiene tres unidades de observación que permiten la estructuración adecuada de la información censal: Unidad de vivienda, hogar particular y las personas residentes, todas ellas referidas a una unidad geográfica, el manejo adecuado de cada una de ellas, en cada uno de los procesos del proyecto censal, son la garantía para obtener la información estratégica para cada uno de los entes territoriales.

Para su identificación se tienen los siguientes conceptos fundamentales, de cuya adecuada aplicación depende la calidad de la información censal:

• **Edificación:** Es una construcción, independiente y separada, compuesta por una o más unidades.

- *Independiente:* Edificación que tiene acceso directo desde la vía pública, caminos, senderos o espacios de circulación común.
- *Separada:* Edificación que generalmente tiene paredes que la delimitan y diferencian de otras construcciones.

• **Unidad:** Es un espacio independiente y separado que constituye parte, o la totalidad de una edificación cuyos usos pueden ser económico, vivienda o mixto.

- *Económico:* Cuando está destinado o está siendo utilizado para la industria, el comercio o los servicios.
- *Industria:* Es la transformación física y/o química de materiales y componentes en productos nuevos, ya sea que el trabajo se efectúe con máquinas o a mano, en una fábrica o a domicilio, que los productos se vendan al por mayor o al por menor.
- *Comercio:* Compra y venta al por mayor y al por menor de mercancías (bienes que van al mercado), nuevas y usadas y que no son sometidas a transformaciones en su naturaleza intrínseca durante las actividades inherentes a la comercialización, transporte, almacenaje, empaque, reempaque, embalaje, etc., o en aquellas a adecuar, exhibir, presentar o promocionar las mercancías objeto de la venta.

- *Servicios*: Son productos heterogéneos generados cada vez que son solicitados, sobre los cuales no recaen derechos de propiedad por parte del usuario y no pueden ser negociados por separado de su producción ni pueden ser transportados ni almacenados.

• **Unidad de Vivienda**: Es un espacio independiente y separado con áreas de uso exclusivo, habitado o destinado a ser habitado por una o más personas.

Con el fin de facilitar la cuantificación del *stock* de unidades de vivienda y el análisis de la información. Para efectos censales se tiene la siguiente tipología:

- *Casa*: Es la edificación constituida por una sola unidad cuyo uso es el de vivienda, con acceso directo desde la vía pública o desde el exterior de la edificación. El servicio sanitario y la cocina pueden estar o no dentro de ella. Para efectos censales, también se consideran casas aquellas en donde el garaje, la sala o alguna habitación se destina para uso económico.
- *Casa Indígena*: Es la edificación constituida por una sola unidad cuyo uso es el de vivienda, construida según la costumbre de cada grupo étnico conservando la estructura tradicional.
- *Apartamento*: Es una unidad de vivienda, que hace parte de una edificación, en la cual hay otra(s) unidad(es) que generalmente es (son) de vivienda. Tiene acceso directo desde el exterior o por pasillos, patios, corredores, escaleras o ascensores. Dispone de servicio sanitario y cocina en su interior.
- *Cuarto*: Es una unidad de vivienda, que hace parte de una edificación y que dispone de uno o más espacios. Tiene acceso directo desde el exterior o por pasillos, patios, zaguanes, corredores u otros espacios de circulación común. En general carece de servicio sanitario y cocina en su interior, o sólo dispone de uno de estos dos servicios.
- *Otro tipo de Vivienda*: Es un espacio adaptado para vivienda, donde en el momento de la entrevista habitan personas. Generalmente carece de servicio sanitario y cocina, incluye lugares tales como: los vagones de trenes, los contenedores, las embarcaciones, las tiendas de gitanos, las cuevas, los puentes, las casetas.

- **Residente Habitual:** Es la persona que vive permanentemente o la mayor parte del tiempo en una vivienda aunque en el momento de la entrevista se encuentre ausente.
- **Hogar:** Es la persona o grupo de personas, parientes o no, que ocupan la totalidad o parte de una vivienda; atienden necesidades básicas, con cargo a un presupuesto común y generalmente comparten las comidas.
- **Jefe o Jefa de Hogar:** Es el residente habitual reconocido como tal por los demás miembros de su hogar.
- **Informante en el Hogar:** Es cada una de las personas residentes habituales mayores de 16 años. En el caso de no estar presente y no poderse conseguir en las visitas, la información la debe dar el (la) Jefe(a) del Hogar, o su cónyuge o una persona residente mayor de 18 años que conozca la información de la persona.  
Para los menores de 18 años la información la debe dar el (la) Jefe(a) del Hogar o su cónyuge.
- **Lugar Especial de Alojamiento (L.E.A.):** Es una institución en la cual viven (es decir duermen) colectivamente un grupo de personas generalmente no parientes.



### 3. COMPARACIÓN MYSQL Y POSTGRESQL<sup>15</sup>

#### *Introducción*

Son muchos las pruebas de rendimiento que se han publicado sobre estos gestores de bases de datos, aunque muchos de ellos tienen una clara tendencia hacia uno de los dos bandos. Es por esto que hay que saber extraer bien las conclusiones a partir de una prueba. Por ejemplo, es importante que las comparaciones entre los dos gestores se realicen en igualdad de condiciones, cosa que por otra parte, muchas veces resulta complicado debido a las distintas implementaciones que poseen estos gestores.

La empresa EnterpriseDB decidió desarrollar una serie de pruebas de rendimiento y eficiencia con los gestores de bases de datos de código abierto PostgreSQL y MySQL teniendo en cuenta las fortalezas y debilidades de cada uno para aplicaciones de tipo empresarial de alta demanda, llegando a las siguientes conclusiones:

#### - Rendimiento:

MySQL es considerado por muchos como un gestor de bases de datos de alto rendimiento, y esto puede ser cierto para aplicaciones Web del tipo “solo lectura” (ejemplo: foros de discusión).

Sin embargo para una aplicación empresarial de alto nivel MySQL tiene un rendimiento apenas aceptable, ya que su motor interno (MyISAM) es una implementación de un sistema de recuperación rápida de información creado por IBM para acceso eficiente a la información que a pesar de su gran velocidad provoca ciertos problemas de rutinas que son críticos para el ambiente de alto trabajo empresarial. Como por ejemplo corrupción de datos, falta de bloqueo a nivel de filas, y mantenimiento fuera de línea.

En contraste PostgreSQL no presenta ninguno de estos problemas ya que está diseñado para ser escalable y puede hacerse mantenimiento en línea (sin bajar la base de datos), todo esto en un proceso transparente para el usuario y sin afectar las aplicaciones que dependen de la base de datos.

#### - Motores múltiples de almacenamiento y optimización de consultas:

---

<sup>15</sup> EnterpriseDB: PostgreSQL vs. MySQL. [En línea]. Edison, New Jersey (USA). EnterpriseDB Corp. Fecha de consulta: julio 20 de 2008. disponible en: [http://downloads.enterprisedb.com/whitepapers/PGvsMySQL\\_LowRes.pdf](http://downloads.enterprisedb.com/whitepapers/PGvsMySQL_LowRes.pdf)

No fue sino recientemente que MySQL implementó soporte para múltiples motores de almacenamiento el cual aun no es suficientemente sólido para lograr una precisa optimización de consultas y puede hacer que el mantenimiento y optimización de la base de datos sea difícil.

PostgreSQL ha tenido soporte para motores de almacenamiento múltiples desde finales de los 80's por lo que a lo largo de los años se ha vuelto uno de los sistemas más robustos en lo que a este campo se refiere.

- Soporte transaccional:

Las bases de datos de clase empresarial requieren por obligación soporte transaccional ya que para aplicaciones de alta demanda y alto rendimiento si un error transaccional ocurre y la aplicación dependiente de la base de datos no es capaz de solucionar el problema, la base de datos debe ser la encargada de deshacer la transacción incompleta garantizando la consistencia de la información. A esto se le llama atomicidad, una base de datos con soporte transaccional completo se le considera compatible con ACID (siglas en inglés de Atomicidad, Consistencia, Aislamiento, Durabilidad).

En este caso solo PostgreSQL es completamente compatible con ACID, mientras que MySQL (MyISAM) no lo es.

- Integridad referencial:

En lo que se refiere a integridad referencial solo unas cuantas versiones de MySQL tienen soporte completo, ya que por lo general MySQL acepta la sintaxis de las reglas de integridad referencial pero la misma naturaleza de su motor de almacenamiento, los administradores que utilizan MySQL por lo general tienen que hacer chequeo doble para la sintaxis de sus reglas de integridad referencial. Cosa que no se necesita hacer con un gestor de bases de datos como PostgreSQL.

- soporte de lenguaje procedimental:

PostgreSQL tiene soporte para lenguaje procedimental dentro de la misma base de datos desde 1998, el soporte para lenguaje procedimental fue apenas recientemente añadido a MySQL por lo que aun no está completamente "madurado". Además PostgreSQL ofrece la ventaja de ser gestor de bases de datos extensible, lo que le da el poder a los desarrolladores de implementar los procedimientos almacenados en lenguajes de programación comunes como php o perl.

- Soporte para disparadores:

El soporte para disparadores solamente fue añadido recientemente en MySQL, pero dicha implementación no es del tipo de disparadores a nivel de fila (se ejecuta por cada fila). El soporte para disparadores a nivel de fila existe en PostgreSQL desde la versión 6.3.

- Soporte para métodos de autenticación:

Para uso empresarial solo PostgreSQL provee metodos de autenticación basados en estándares como SSPI, Kerberos, Ident, LDAP y otros. MySQL solo provee métodos de autenticación propios no estándares que no son muy apropiados para aplicaciones empresariales de alto calibre.

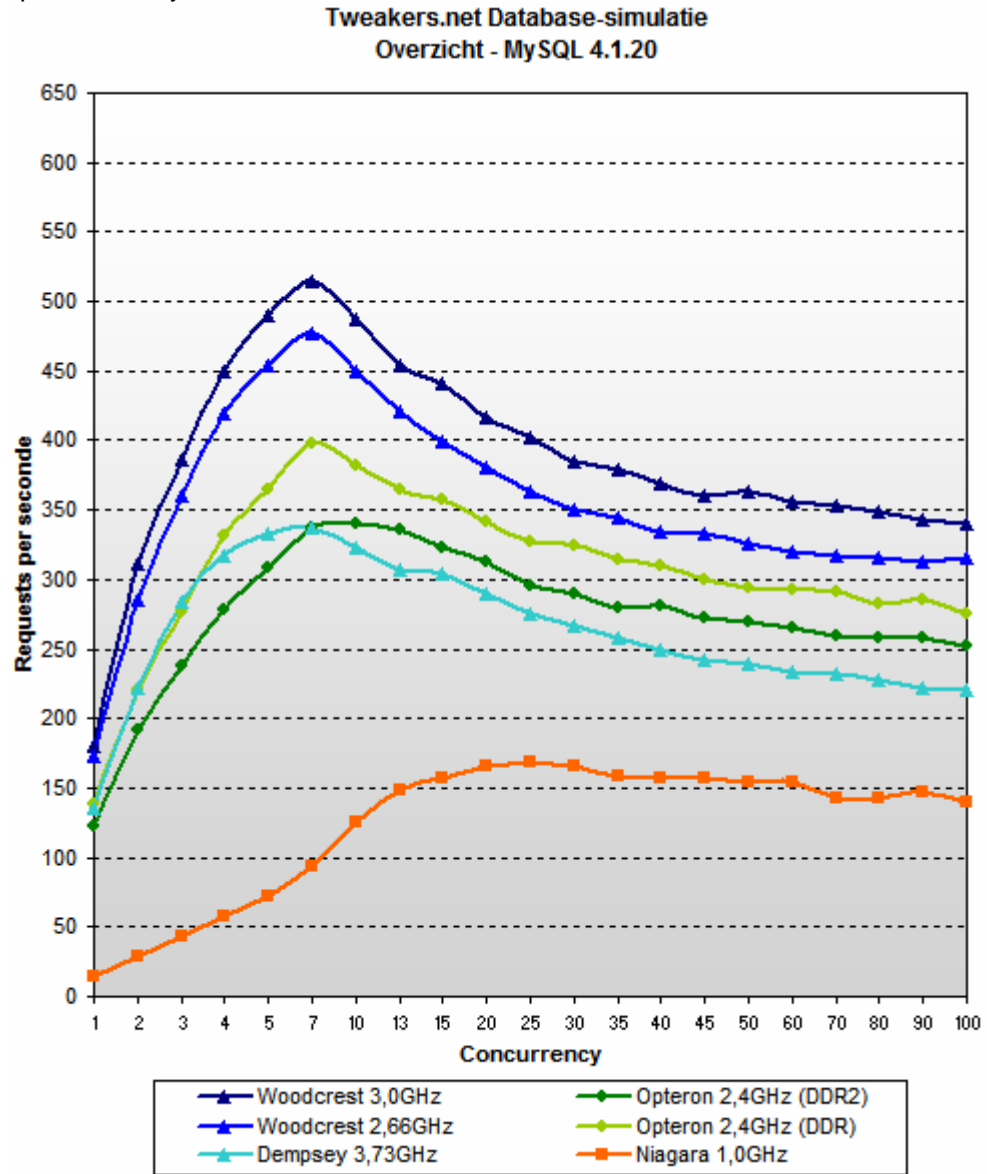
### Conclusiones

Basados en las características y funcionalidades que poseen ambos gestores de bases de datos, se concluye que PostgreSQL esta mejor adaptado para el ambiente empresarial donde la carga de datos es mucho mayor y se necesita un alto nivel de confiabilidad que MySQL aun no ofrece.

Para mayor comprensión de este tipo de evaluaciones se ha agregado una serie de pruebas que se hicieron a los 2 gestores de bases de datos en el año 2006 sobre varios tipos de servidores para mostrar su rendimiento.

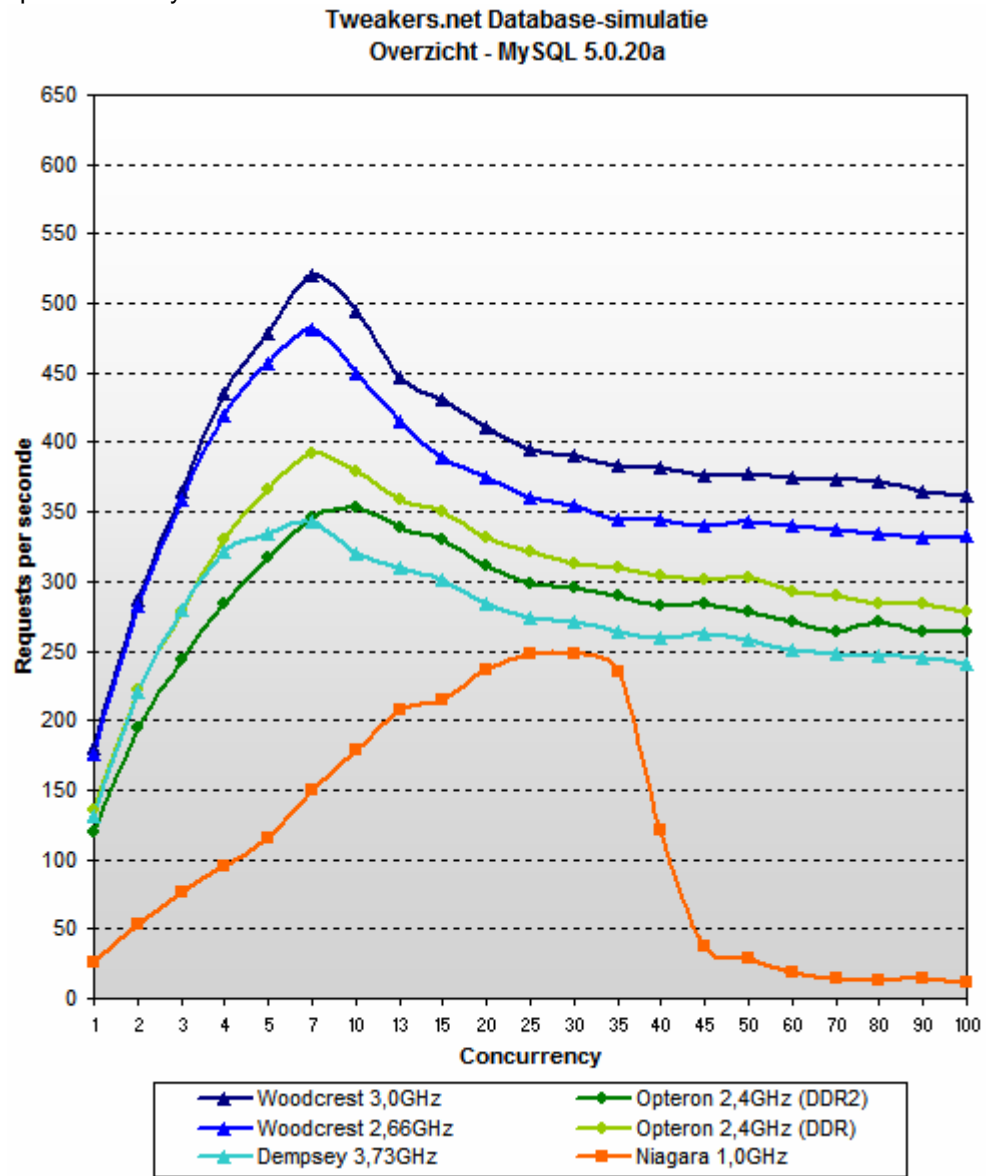
A continuación se muestran los gráficos de prueba de PostgreSQL y MySQL sobre un dual Intel Xeon y la comparativa con otras maquinas:

Figura 1: prueba de MySQL 4.1.20



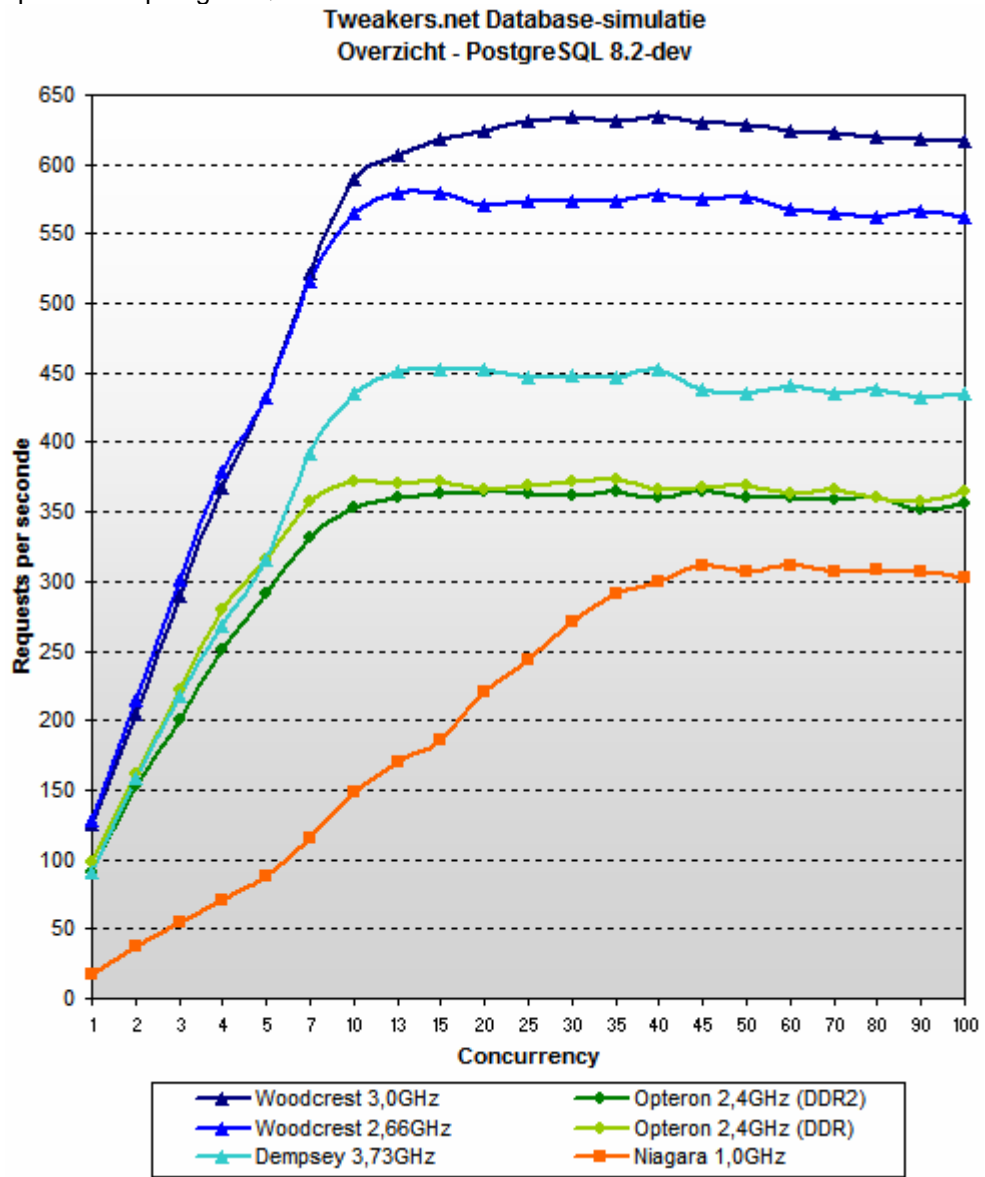
Fuente: TWEAKERS.NET [en linea]. Fecha de consulta: 13/11/2006. Disponible en: <http://tweakers.net/reviews/657/6>

Figura 2: prueba de MySQL 5.0.20a



Fuente: TWEAKERS.NET [en linea]. Fecha de consulta: 13/11/2006. Disponible en: <http://tweakers.net/reviews/657/6>

Figura 3: prueba de postgresQL 8.2-dev



Fuente: TWEAKERS.NET [en linea]. Fecha de consulta: 13/11/2006. Disponible en: <http://tweakers.net/reviews/657/6>

## **ANEXO B DOCUMENTACIÓN TÉCNICA**

### **CONTENIDO**

	<b>PAG</b>
<b>1. BREVE ESPECIFICACIÓN DEL PROBLEMA: MODELO DE CASOS DE USO</b>	<b>38</b>
<b>1.1 ESCENARIO NORMAL DE OPERACIÓN DE CENSO POBLACIONAL</b>	<b>38</b>
<b>1.2 MODELO DE SITUACIÓN DEL PROBLEMA (CASOS DE USO)</b>	<b>39</b>
<b>2. MODELO DE OBJETOS Y DICCIONARIO DE CLASES</b>	<b>40</b>
<b>2.1 MODELO DE OBJETOS: SISTEMA DE RECOLECCIÓN DE INFORMACIÓN VÍA PHP</b>	<b>41</b>
<b>2.2 MODELO DE OBJETOS: SISTEMA DE ADMINISTRACIÓN DE CONSULTAS JAVA</b>	<b>42</b>
<b>2.3 DICCIONARIO DE CLASES</b>	<b>43</b>
<b>2.4 ANALISIS Y DISEÑO: MODELO DINAMICO MAQUINA DE ESTADOS</b>	<b>54</b>
<b>2.4.1 Recolección de datos vía Web</b>	<b>55</b>
<b>2.4.2 Solicitud de consultas a la base de datos vía Web Service</b>	<b>56</b>
<b>2.5 ANALISIS Y DISEÑO: DIAGRAMAS DE SECUENCIA</b>	<b>57</b>
<b>2.6 MODELO FUNCIONAL: DIAGRAMA DE CASOS DE USO</b>	<b>120</b>
<b>2.6.1 Diagrama de casos de uso Sócrates-EC</b>	<b>121</b>
<b>2.7 MODELO FUNCIONAL: DIAGRAMA DE DESPLIEGUE, MODELO ENTIDAD RELACIÓN</b>	<b>134</b>
<b>2.7.1 Diagrama de despliegue</b>	<b>135</b>
<b>2.7.2 Modelo entidad – relación</b>	<b>136</b>
<b>2.7.3 Diccionario de datos del modelo entidad-relación</b>	<b>137</b>

## 1. BREVE ESPECIFICACIÓN DEL PROBLEMA: MODELO DE CASOS DE USO

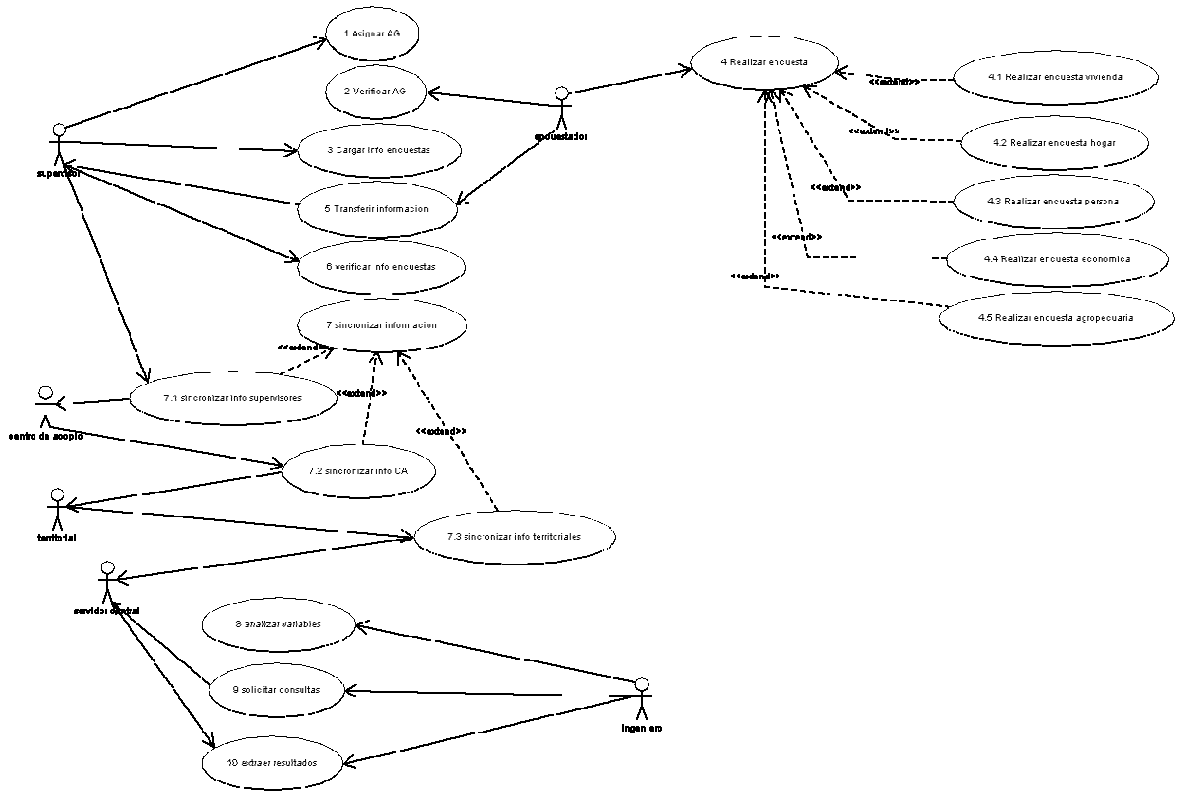
Durante el periodo 2005-2006 se ejecuto el proceso de censo poblacional, aunque se mejoraron las tecnologías, el proceso de recolección y procesamiento no cambio mucho respecto a años anteriores y las organizaciones que proveen servicios a la comunidad no tienen total acceso a la información recogida, eso sin tener en cuenta que el proceso de censado solo se ejecuta en intervalos de 10-12 años.

### 1.1 ESCENARIO NORMAL DE OPERACIÓN DE CENSO POBLACIONAL:

- El supervisor asigna el área geográfica en la terminal (dispositivo móvil de captura o DMC).
- El supervisor carga la información de encuestas de esa área geográfica en la DMC.
- El supervisor entrega la DMC configurada al encuestador.
- El encuestador va hacia la unidad censal (vivienda).
- El encuestador observa las características exteriores de la vivienda.
- El encuestador introduce dicha información en el DMC/formulario.
- El encuestador solicita entrar a la vivienda.
- El jefe de familia le permite entrar al encuestador.
- El encuestador solicita la información censal deseada.
- El jefe de familia le da la información al encuestador.
- El encuestador anota la información en el DMC/formulario.
- El encuestador guarda la información.
- El encuestador busca al supervisor.
- El encuestador le entrega la información al supervisor.
- El supervisor transfiere la información recogida de los encuestadores al centro de acopio.
- El centro de acopio recoge la información de todos los supervisores.
- El centro de acopio envía toda la información al centro territorial.
- El centro territorial reúne la información de todos los centros de acopio.
- El centro territorial envía la información al servidor central.
- El servidor central recibe toda la información de los centros territoriales.
- En el servidor central se almacena toda la información en la base de datos.
- El ingeniero analiza las variables de medición en la información almacenada.
- Se generan datos estadísticos de toda la información relacionados con las variables analizadas.



## 1.2 MODELO DE SITUACIÓN DEL PROBLEMA (CASOS DE USO):

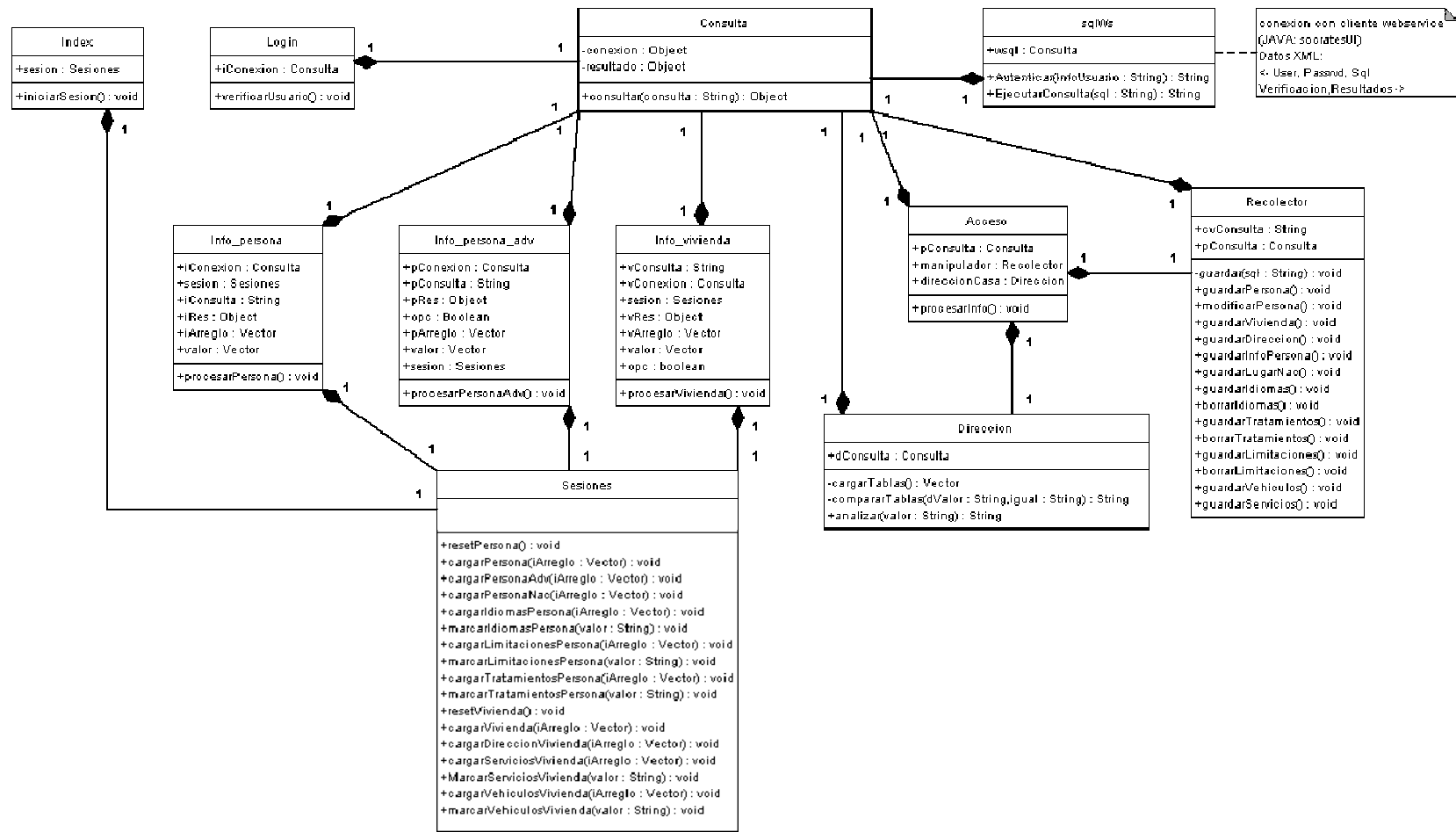


## 2. MODELO DE OBJETOS Y DICCIONARIO DE CLASES

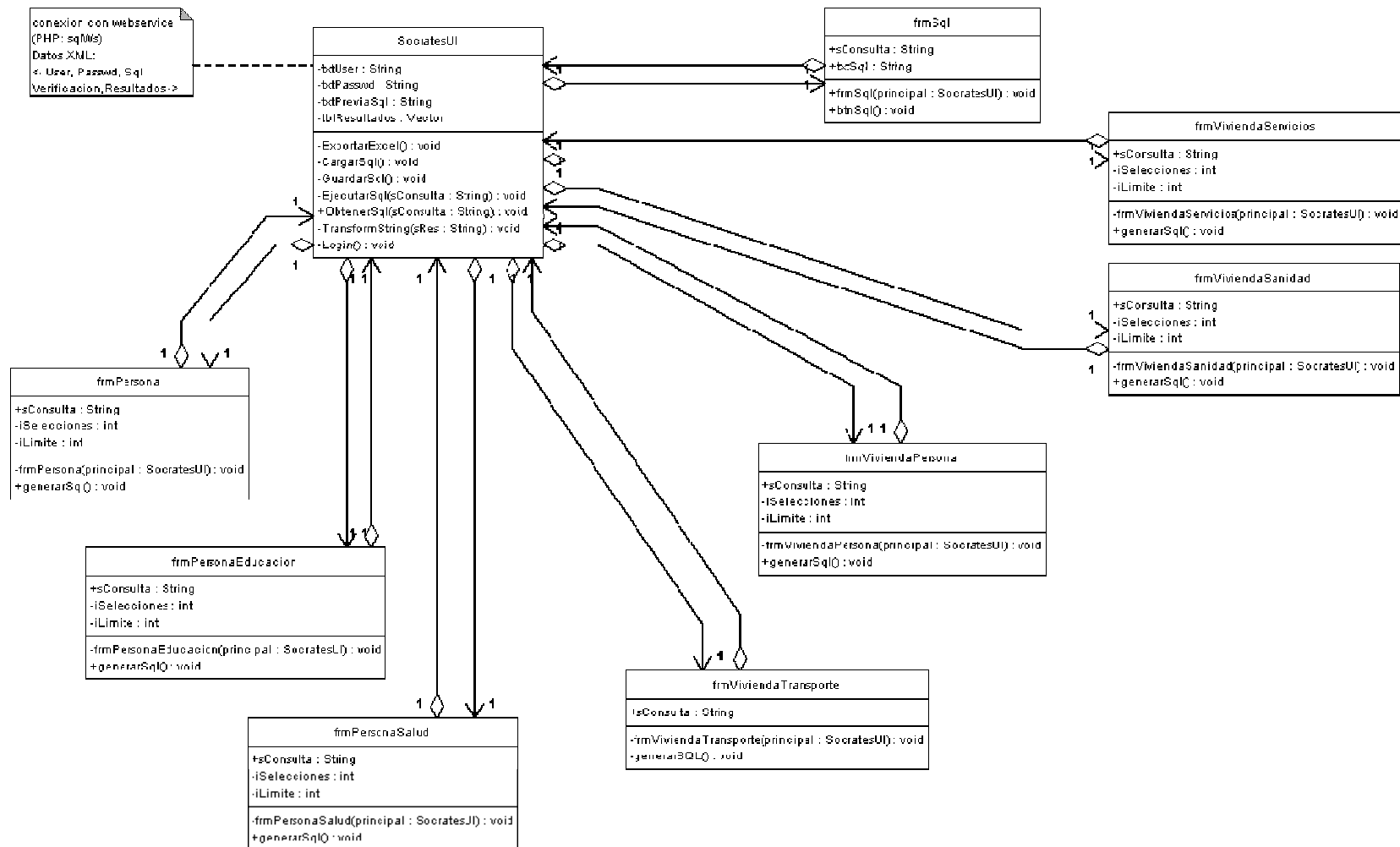
SOCRATES-EC: SISTEMA DE INFORMACIÓN POBLACIONAL BASADO EN  
PROCEDIMIENTOS DE CENSADO

Por:  
Danko Jiménez Londoño  
Daniel Andrés Valencia V.

## 2.1 MODELO DE OBJETOS: SISTEMA DE RECOLECCIÓN DE INFORMACIÓN VÍA PHP



## 2.2 MODELO DE OBJETOS: SISTEMA DE ADMINISTRACIÓN DE CONSULTAS JAVA



## 2.3 DICCIONARIO DE CLASES

Clase	<b>SocratesUI</b>	
<b>Descripción</b>	Clase principal java, parte del sistema de consulta para las organizaciones que proveen servicios a la comunidad. Permite el acceso a la generación y administración de consultas, la conexión con el webservice para traer los resultados y la visualización y exportación de los mismos.	
<b>Atributos</b>	txtUser String (textfield) txtPasswd String (textfield) txtPreviaSql String (textArea) tblResultados Vector (table)	Nombre de usuario Contraseña de acceso Vista previa SQL generado  Tabla de resultados
<b>Métodos</b>	ExportarExcel() CargarSql() GuardarSql() EjecutarSQL(sConsulta: String) ObtenerSql(sConsulta: String) TransformString(sRes: String) Login()	

Clase	<b>frmSQL</b>	
<b>Descripción</b>	Formulario avanzado de consulta, se puede insertar directamente aquí código SQL para hacer las consultas (validado solo para consultas de selección)	
<b>Atributos</b>	sConsulta String txtSql String	Cadena de SQL Campo de texto para ingresar consultas
<b>Métodos</b>	frmSql(principal: SocratesUI) btnSql()	

<b>Clase</b>	<b>frmPersona</b>	
<b>Descripción</b>	Generador de consultas SQL para los datos relacionados con información básica personal.	
<b>Atributos</b>	sConsulta String iSelecciones int iLimite int	Cadena de selecciones SQL Contador de selecciones SQL Limitador/validador de selecciones SQL
<b>Métodos</b>	frmPersona(principal: SocratesUI) generarSql()	

<b>Clase</b>	<b>frmPersonaEducacion</b>	
<b>Descripción</b>	Generador de consultas SQL para los datos relacionados con información sobre educación y cultura de la persona (población)	
<b>Atributos</b>	sConsulta String iSelecciones int iLimite int	Cadena de selecciones SQL Contador de selecciones SQL Limitador/validador de selecciones SQL
<b>Métodos</b>	frmPersonaEducacion(principal: SocratesUI) generarSql()	

<b>Clase</b>	<b>frmPersonaSalud</b>	
<b>Descripción</b>	Generador de consultas SQL para los datos relacionados con información de salud de la persona (población).	
<b>Atributos</b>	sConsulta String iSelecciones int iLimite int	Cadena de selecciones SQL Contador de selecciones SQL Limitador/validador de selecciones SQL
<b>Métodos</b>	frmPersonaSalud(principal: SocratesUI) generarSql()	

<b>Clase</b>	<b>frmViviendaTransporte</b>	
<b>Descripción</b>	Generador de consultas SQL para los datos relacionados con información de vivienda y los vehículos de los habitantes de la misma.	
<b>Atributos</b>	sConsulta String iSelecciones int iLimite int	Cadena de selecciones SQL Contador de selecciones SQL Limitador/validador de selecciones SQL
<b>Métodos</b>	frmViviendaTransporte(principal: SocratesUI) generarSql()	

<b>Clase</b>	<b>frmViviendaPersona</b>	
<b>Descripción</b>	Generador de consultas SQL para los datos relacionados con información de vivienda y algunos datos de las personas que habitan en ella.	
<b>Atributos</b>	sConsulta String iSelecciones int iLimite int	Cadena de selecciones SQL Contador de selecciones SQL Limitador/validador de selecciones SQL
<b>Métodos</b>	frmViviendaPersona(principal: SocratesUI) generarSql()	

<b>Clase</b>	<b>frmViviendaSanidad</b>	
<b>Descripción</b>	Generador de consultas SQL para los datos relacionados con información de vivienda en lo que respecta a servicios sanitarios y manipulación de desechos.	
<b>Atributos</b>	sConsulta String iSelecciones int iLimite int	Cadena de selecciones SQL Contador de selecciones SQL Limitador/validador de selecciones SQL
<b>Métodos</b>	frmViviendaSanidad(principal: SocratesUI) generarSql()	



<b>Clase</b>	<b>frmViviendaServicios</b>	
<b>Descripción</b>	Generador de consultas SQL para los datos relacionados con información de vivienda y los servicios publicos que posee	
<b>Atributos</b>	sConsulta String iSelecciones int iLimite int	Cadena de selecciones SQL Contador de selecciones SQL Limitador/validador de selecciones SQL
<b>Métodos</b>	frmViviendaServicios(principal: SocratesUI) generarSql()	

<b>Clase</b>	<b>Index</b>	
<b>Descripción</b>	Página PHP de inicio del sistema de recolección de información censal/poblacional vía Web.	
<b>Atributos</b>	sesion Sesiones	Objeto manipulador de datos de sesion
<b>Métodos</b>	iniciarSesion()	

<b>Clase</b>	<b>Login</b>	
<b>Descripción</b>	Página PHP de acceso del sistema de recolección de información censal/poblacional vía Web. Permite el acceso al área de modificación de datos de la familia/vivienda después de haberse previamente almacenado la información.	
<b>Atributos</b>	iConexion Consulta	Objeto que permite el acceso a la base de datos
<b>Métodos</b>	verificarUsuario()	

<b>Clase</b>	<b>Acceso</b>	
<b>Descripción</b>	Página PHP de administración de información del sistema de recolección. Aquí se verifican los nuevos ingresos de información y se presentan las opciones de modificación para información de familia/vivienda	
<b>Atributos</b>	pConsulta Consultador manipulador Recolector	Objeto que permite el acceso a la base de datos Objeto que guarda los datos importantes de sesión en base de datos
<b>Métodos</b>	ProcesarInfo()	

<b>Clase</b>	<b>Consulta</b>	
<b>Descripción</b>	Clase principal, permite el acceso a la base de datos para ingresar/extraer información y consultas de la base de datos. Casi todas las páginas PHP dependen de esta clase para su funcionamiento (incluido el webservice para darle acceso a la aplicación java)	
<b>Atributos</b>	Conexión Object Resultado Object	Objeto PostgreSQL que permite la conexión con la base de datos Objeto PostgreSQL que almacena los resultados de una consulta a la base de datos
<b>Métodos</b>	consultar(consulta: String): Object	

<b>Clase</b>	<b>Recolector</b>	
<b>Descripción</b>	Clase que permite la manipulación de la información de la base de datos, es decir, almacenar los nuevos datos y las actualizaciones por parte del usuario de la página Web (la persona cabeza de hogar)	
<b>Atributos</b>	cvConsulta Consulta	Objeto que permite el acceso a la base de datos
<b>Métodos</b>	guardar(sql: string) guardarPersona() modificarPersona() guardarVivienda() guardarDireccion() guardarInfoPersona() guardarLugarNac() guardarIdiomas() borrarIdiomas() guardarTratamientos() guardarLimitaciones() borrarLimitaciones() guardarVehiculos() guardarServicios()	

<b>Clase</b>	<b>Sesiones</b>	
<b>Descripción</b>	Clase que permite la manipulación de los datos de sesión, estos almacenan la información poblacional que se esta ingresando/modificando antes de ser guardada en la base de datos y cargan la información ya almacenada para su uso por parte del usuario de la página Web (persona cabeza de hogar)	
<b>Atributos</b>		
<b>Métodos</b>	resetPersona() cargarPersona(iArreglo: vector) cargarPersonaAdv(iArreglo: vector) cargarPersonaNac(iArreglo: vector) cargarIdiomasPersona(iArreglo: vector) marcarIdiomasPersona(valor: String) cargarLimitacionesPersona(iArreglo: vector) marcarLimitacionesPersona(valor: String) cargarTratamientosPersona(iArreglo: vector) marcarTratamientosPersona(valor: String) resetVivienda() cargarVivienda(iArreglo: vector) cargarDireccionVivienda(iArreglo: vector) cargarServiciosVivienda(iArreglo: vector) marcarServiciosVivienda(valor: String) cargarVehiculosVivienda(iArreglo: vector) marcarVehiculosVivienda(valor: String)	

<b>Clase</b>	<b>Info_persona</b>	
<b>Descripción</b>	Formulario básico de ingreso/ modificación de datos personales, solo contiene la información básica	
<b>Atributos</b>	iConexion Consulta sesión Sesiones iConsulta String iRes Object iArreglo vector valor vector	Objeto que permite el acceso a la base de datos Objeto manipulador de datos de sesión Cadena sql Objeto PostgreSQL para manipular resultados Tabla temporal de resultados Tabla de resultados para uso directo con los formularios
<b>Métodos</b>	procesarPersona()	

<b>Clase</b>	<b>Info_persona_adv</b>	
<b>Descripción</b>	Formulario avanzado de ingreso/ modificación de datos personales, contiene la información relacionada con salud, cultura y educación	
<b>Atributos</b>	pConexion Consulta sesión Sesiones pConsulta String pRes Object pArreglo vector valor vector opc Boolean	Objeto que permite el acceso a la base de datos Objeto manipulador de datos de sesión Cadena sql Objeto PostgreSQL para manipular resultados Tabla temporal de resultados Tabla de resultados para uso directo con los formularios Validador de selección de campo en formulario
<b>Métodos</b>	procesarPersonaAdv()	

<b>Clase</b>	<b>Info_Vivienda</b>	
<b>Descripción</b>	Formulario avanzado de ingreso/ modificación de datos de vivienda	
<b>Atributos</b>	vConexion Consulta sesión Sesiones vConsulta String vRes Object vArreglo vector valor vector opc Boolean	Objeto que permite el acceso a la base de datos Objeto manipulador de datos de sesión Cadena sql Objeto PostgreSQL para manipular resultados Tabla temporal de resultados Tabla de resultados para uso directo con los formularios Validador de selección de campo en formulario
<b>Métodos</b>	procesarVivienda()	

<b>Clase</b>	<b>Dirección</b>	
<b>Descripción</b>	Clase especial que permite analizar y separar una dirección de unidad de vivienda	
<b>Atributos</b>	dConsulta Consulta	Objeto que permite el acceso a la base de datos
<b>Métodos</b>	cargarTablas() compararTablas(dValor: String, igual: String) analizar(valor: String)	

<b>Clase</b>	<b>sqlWs</b>	
<b>Descripción</b>	Clase especial PHP que contiene el sistema de conexión WebService que permite la comunicación entre la base de datos y la aplicación de administración y consultas java vía mensajes XML	
<b>Atributos</b>	wSql Consulta	Objeto que permite el acceso a la base de datos
<b>Métodos</b>	Autenticar(infoUsuario: String): String EjecutarConsulta(sql: String): String	

2.4 ANALISIS Y DISEÑO  
MODELO DINAMICO  
MAQUINA DE ESTADOS

SOCRATES-EC: SISTEMA DE INFORMACIÓN POBLACIONAL BASADO EN  
PROCEDIMIENTOS DE CENSADO

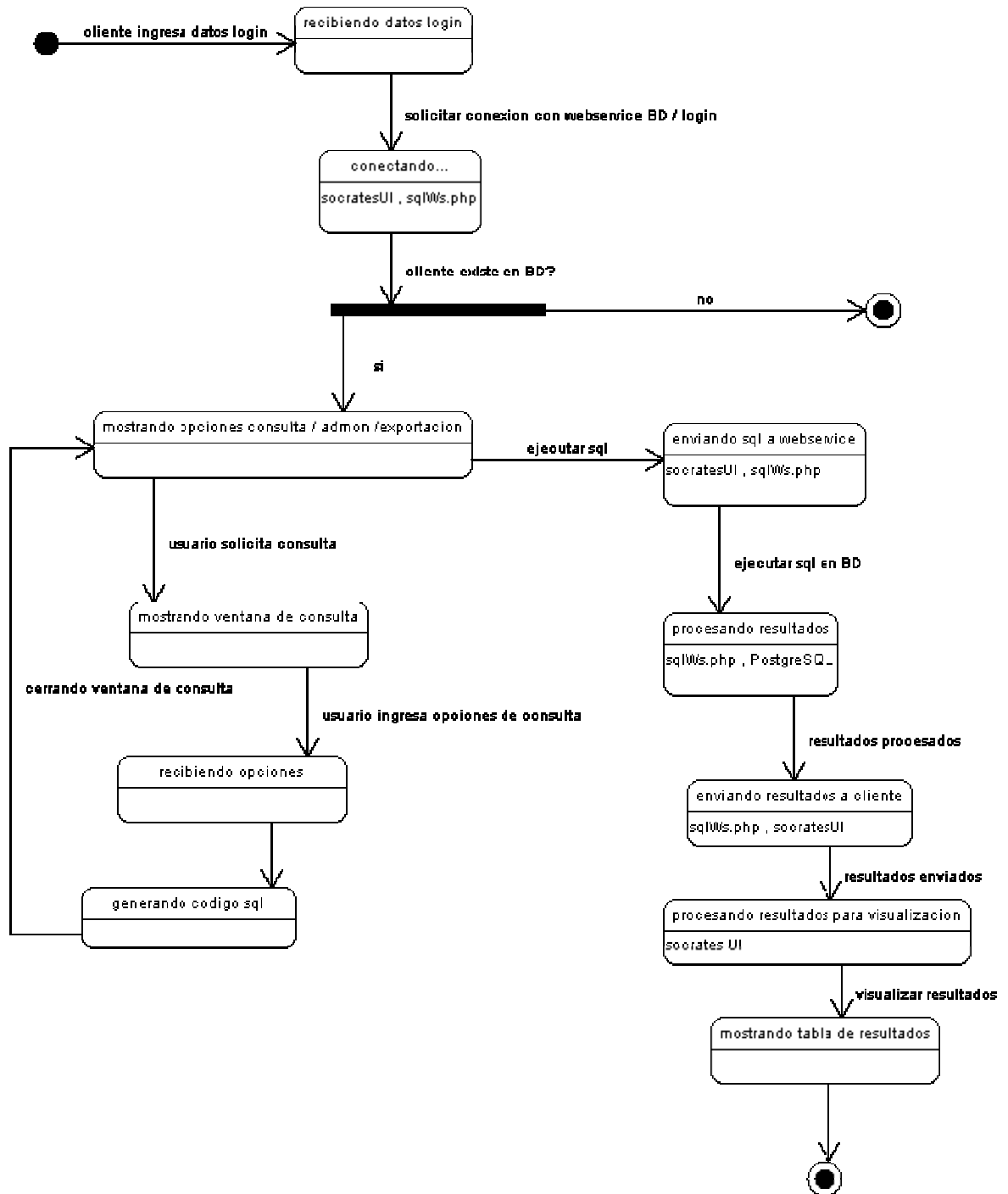
Por:

Danko Jiménez Londoño  
Daniel Andrés Valencia V.





## 2.4.2 Solicitud de consultas a la base de datos vía *Web Service*

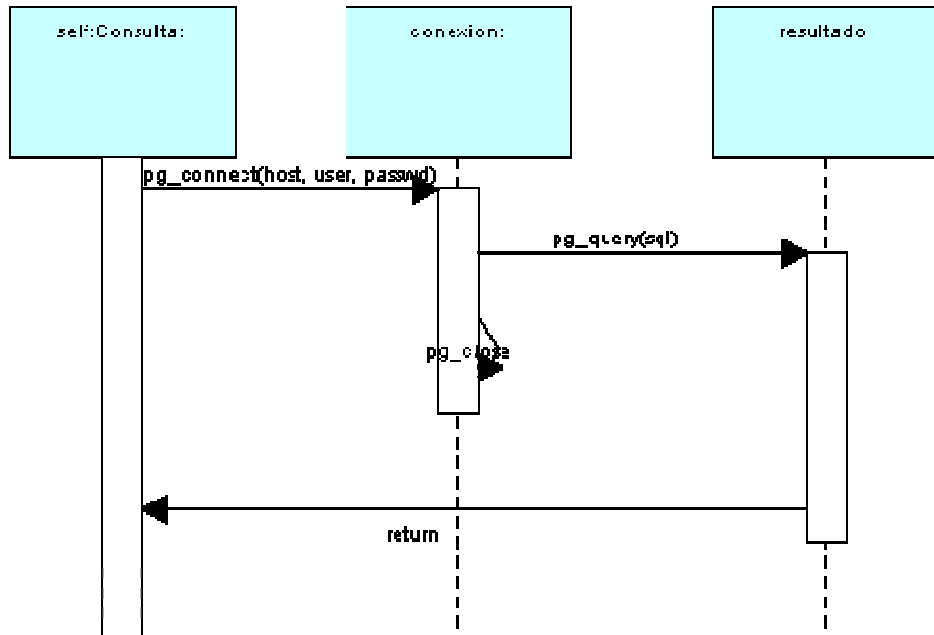


## 2.5 ANALISIS Y DISEÑO DIAGRAMAS DE SECUENCIA

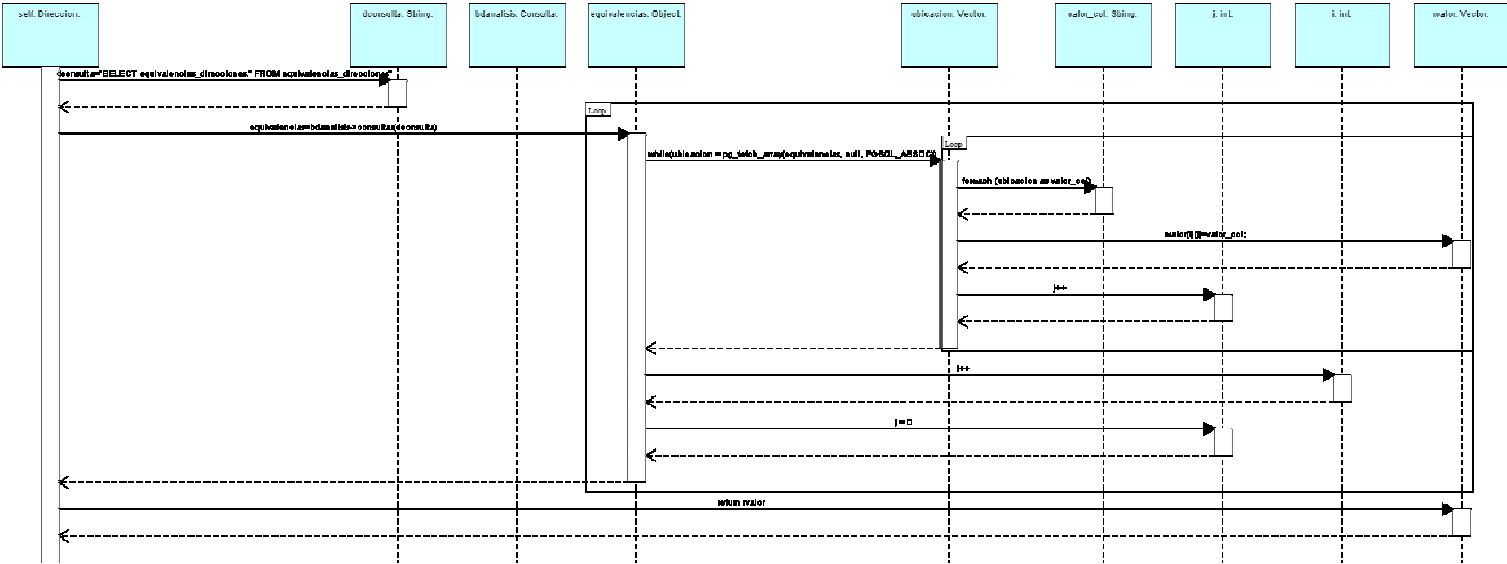
SOCRATES-EC: SISTEMA DE INFORMACIÓN POBLACIONAL BASADO EN  
PROCEDIMIENTOS DE CENSADO

Por:  
Danko Jiménez Londoño  
Daniel Andrés Valencia V.

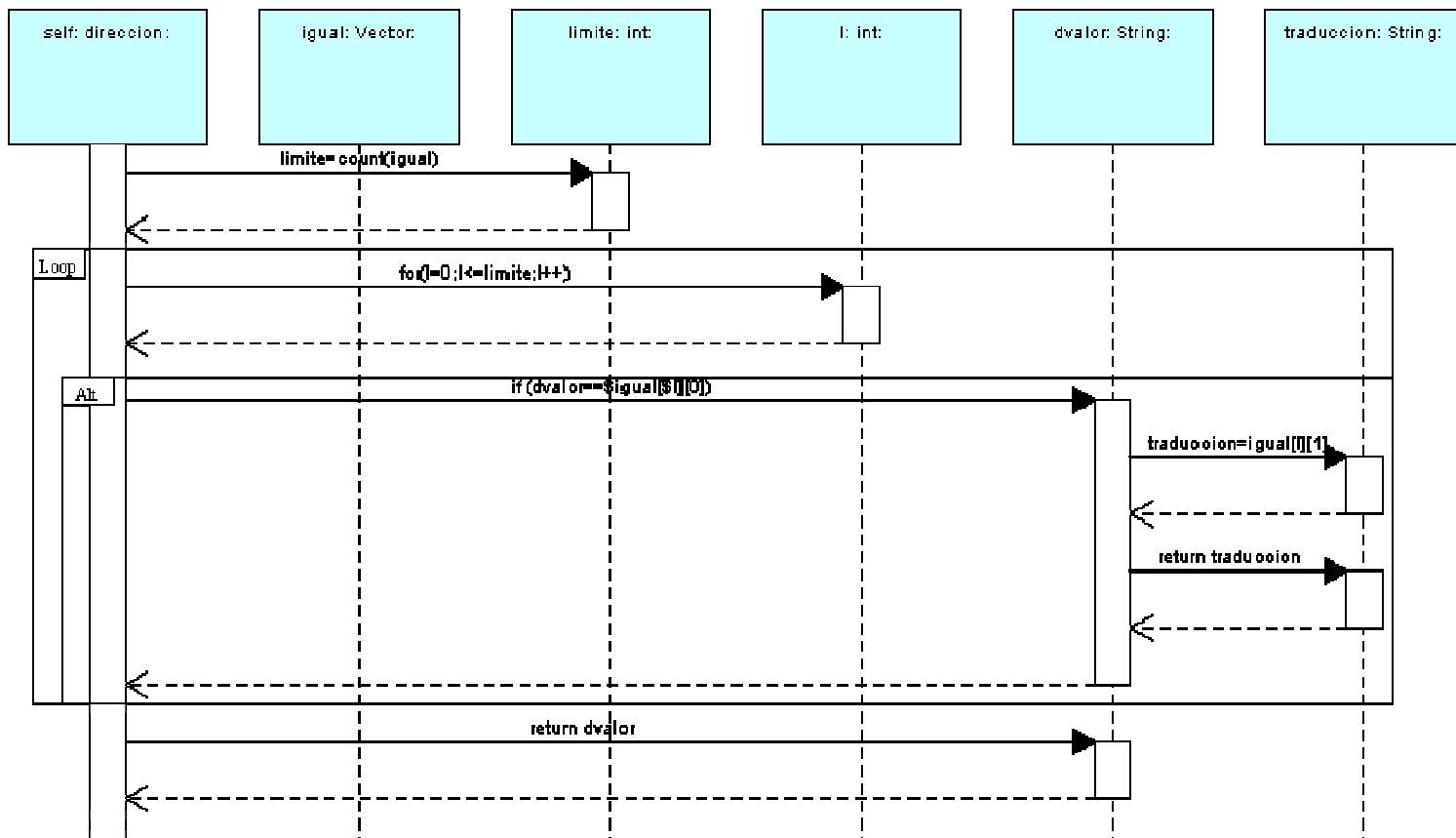
Consultar/insertar/modificar información en base de datos  
(php: clase consulta)



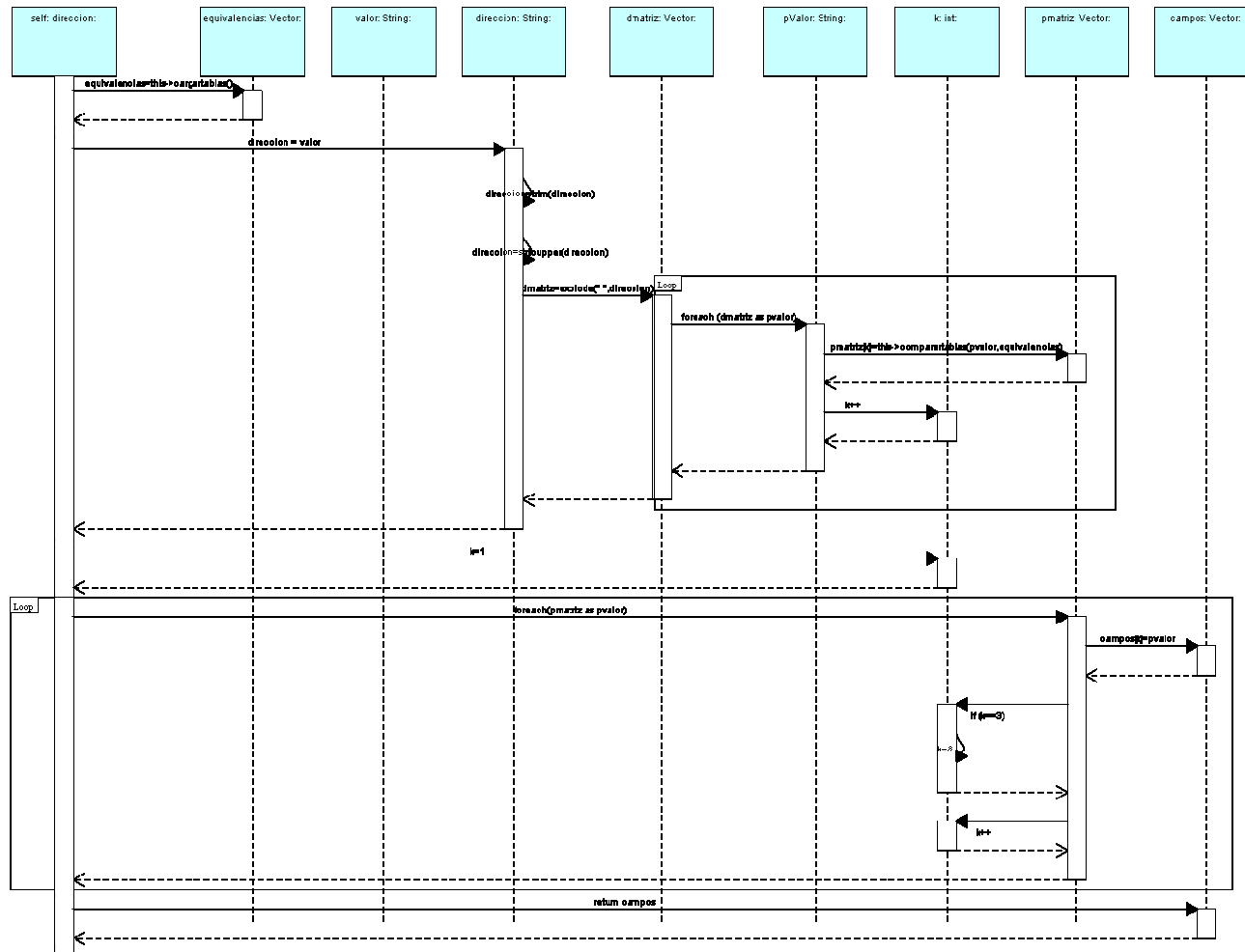
### Cargar tablas de equivalencias para dirección de vivienda (php: clase dirección)



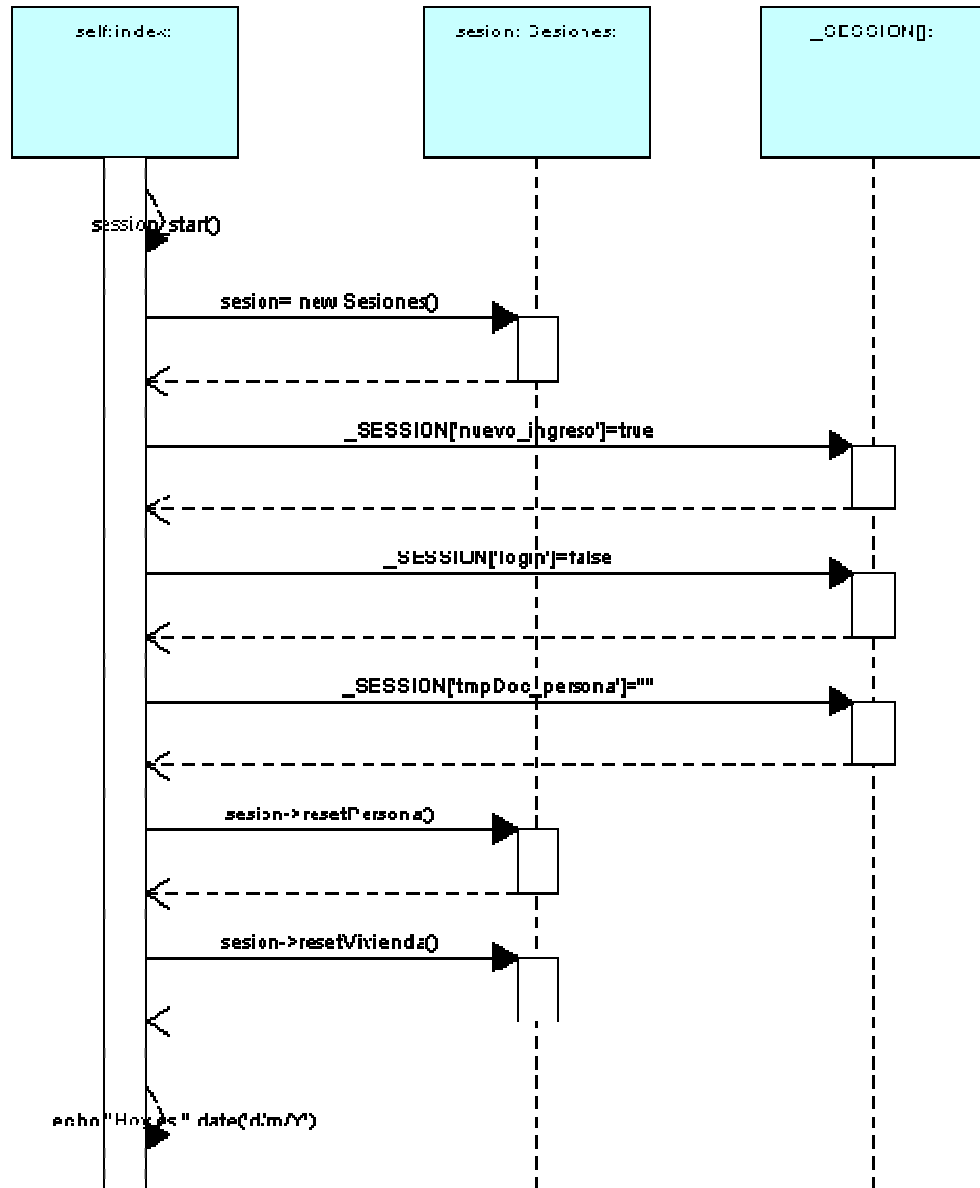
Comparar tablas de dirección con valores entregados por usuario  
(php: clase dirección)



## Analizar/traducir/parametrizar dirección de vivienda dada por el usuario (php: clase dirección)

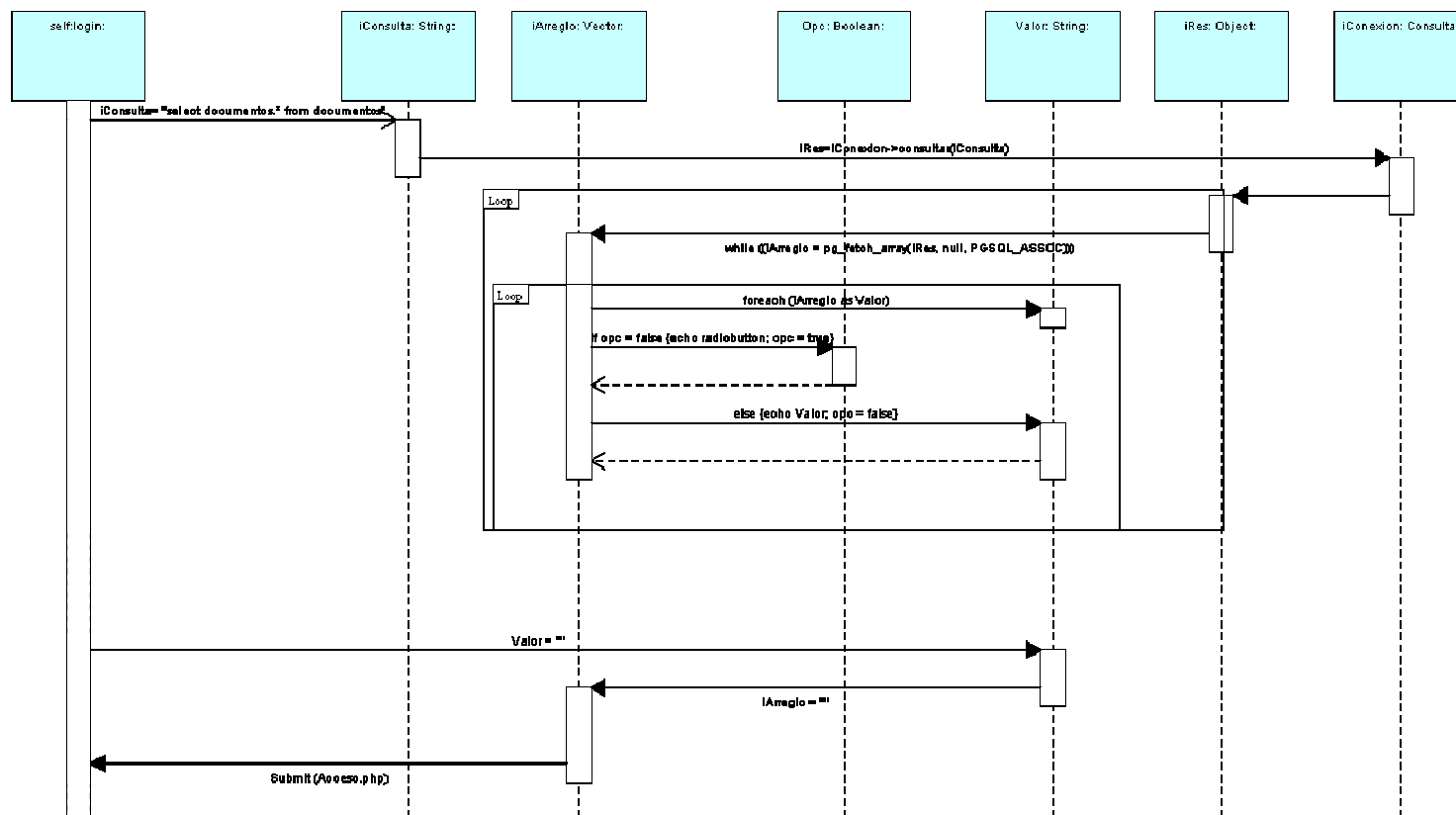


## Iniciar sesión en la página Web (php: index)

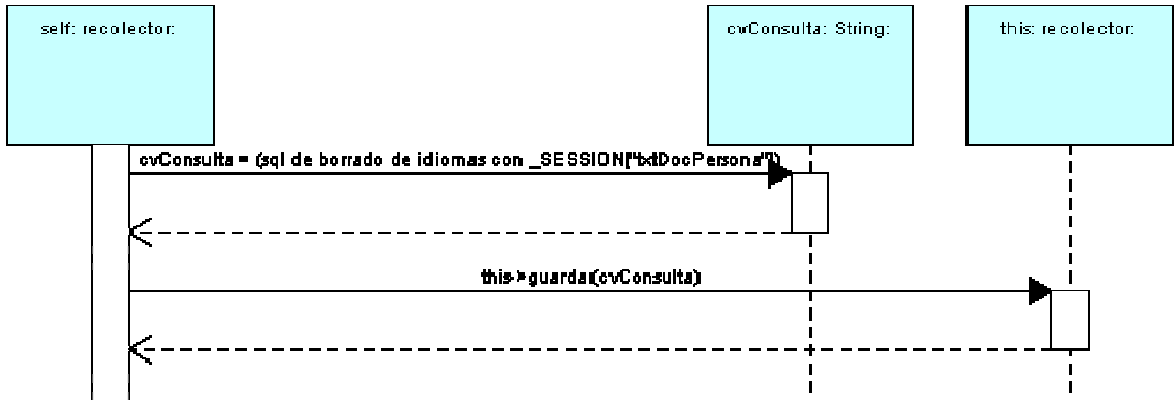




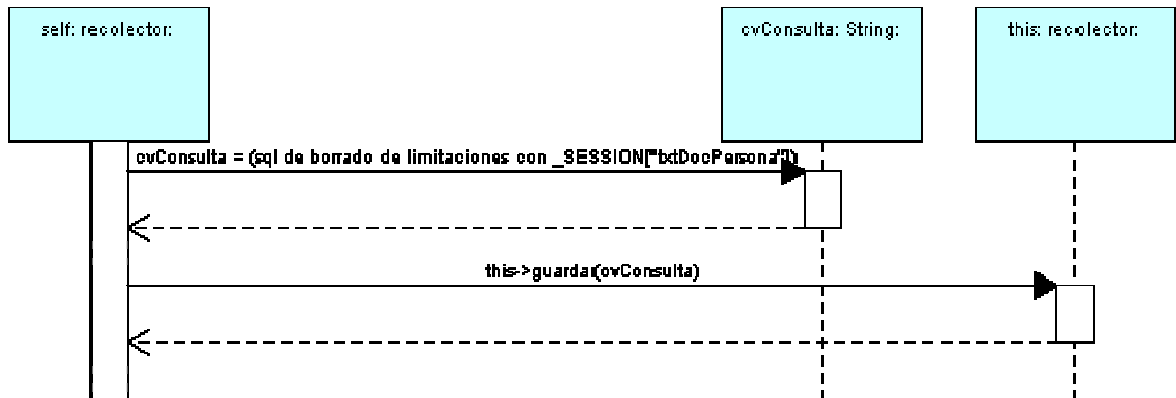
## Verificar que el usuario ya existe en base de datos (php: login)



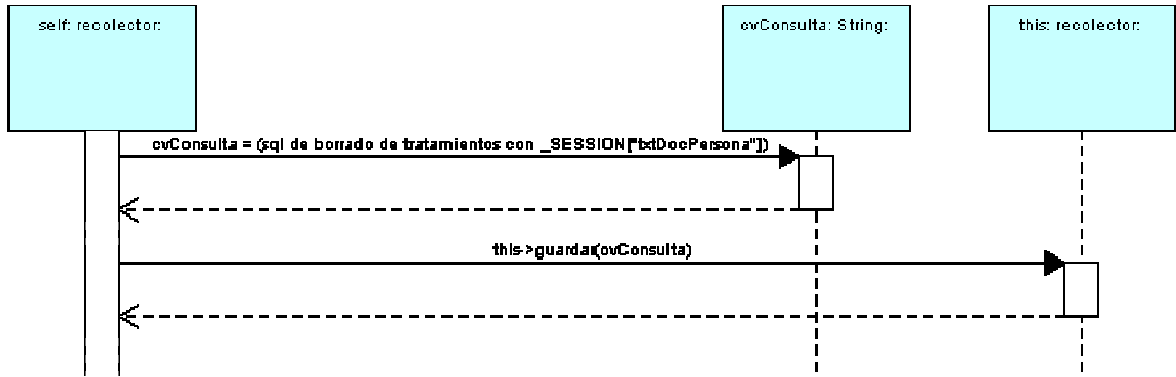
Borrar idiomas de la persona de la base de datos  
(php: clase recolector)



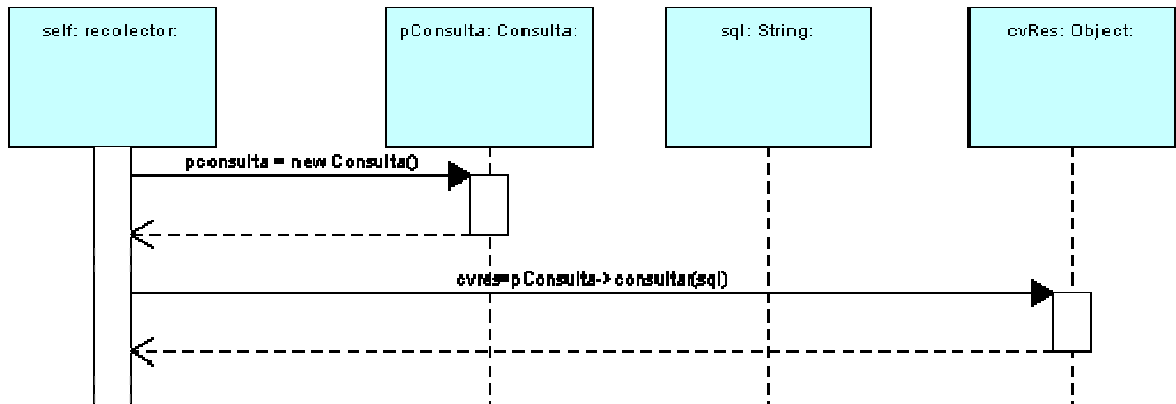
Borrar limitaciones físicas de la persona de la base de datos  
(php: clase recolector)



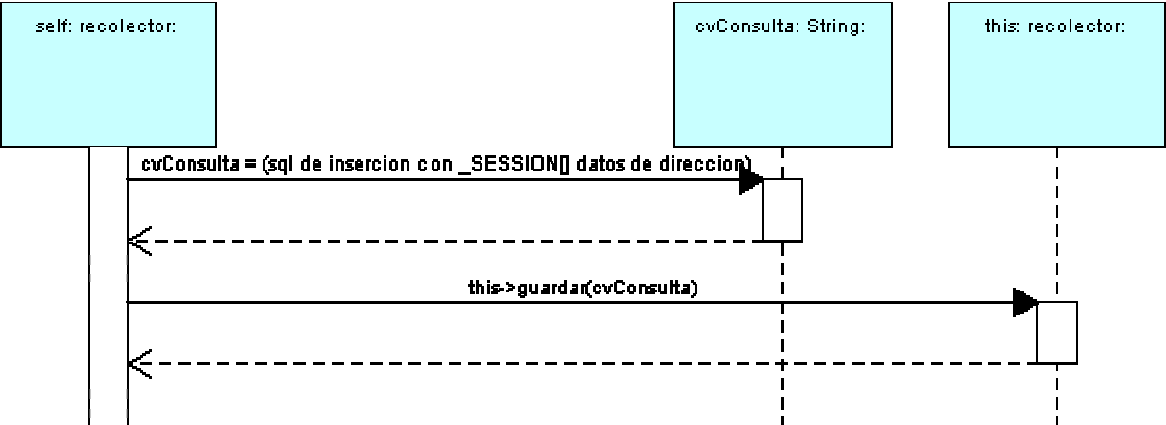
Borrar tratamientos médicos especiales de la persona de la base de datos  
(php: clase recolector)



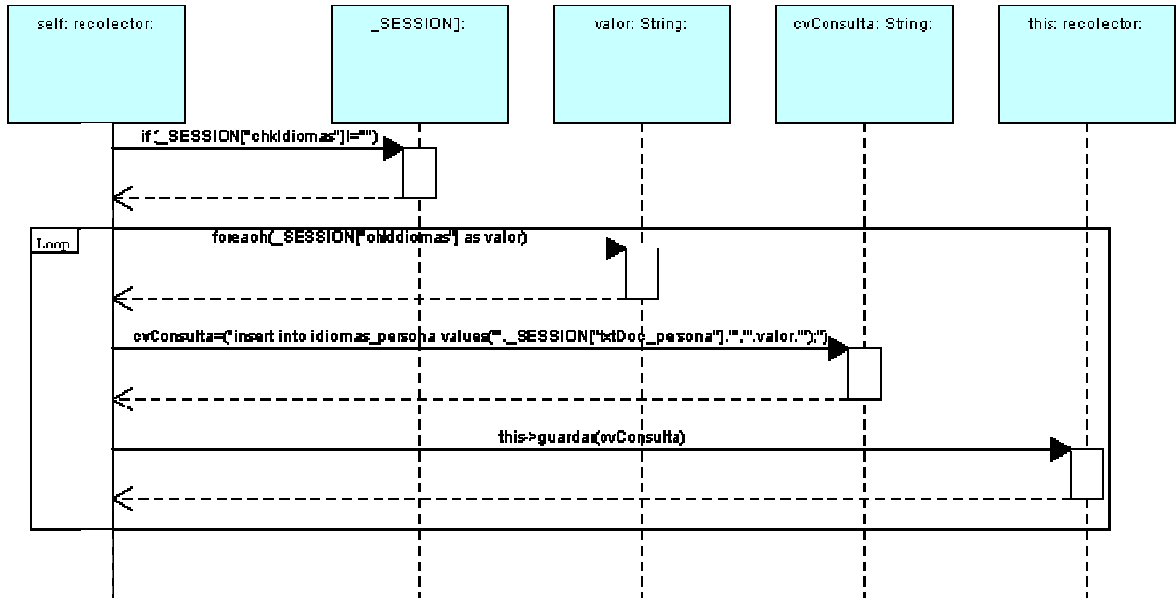
## Guardar paquete de datos en base de datos (php: clase recolector)



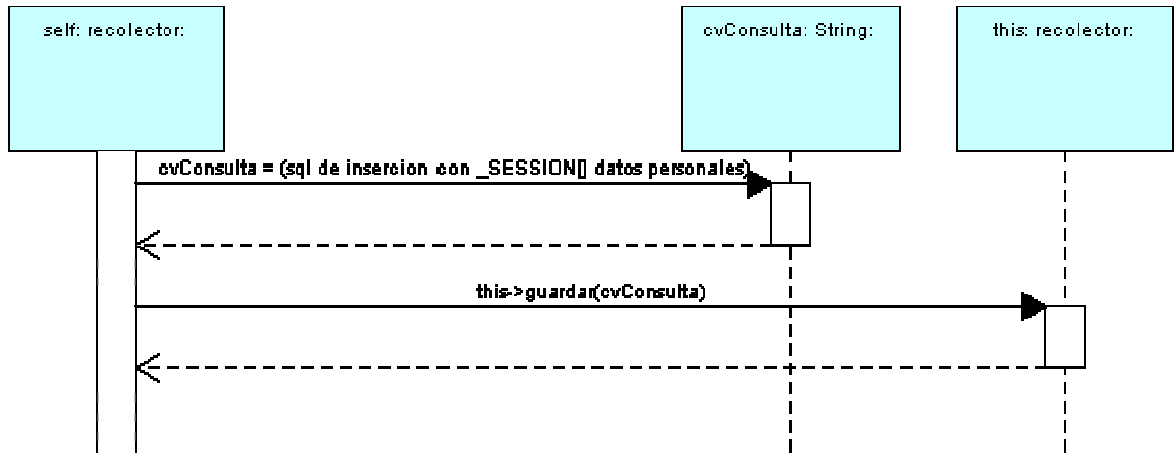
Guardar paquete "dirección" en base de datos  
(php: clase recolector)



### Guardar paquete "idiomas" en base de datos (php: clase recolector)

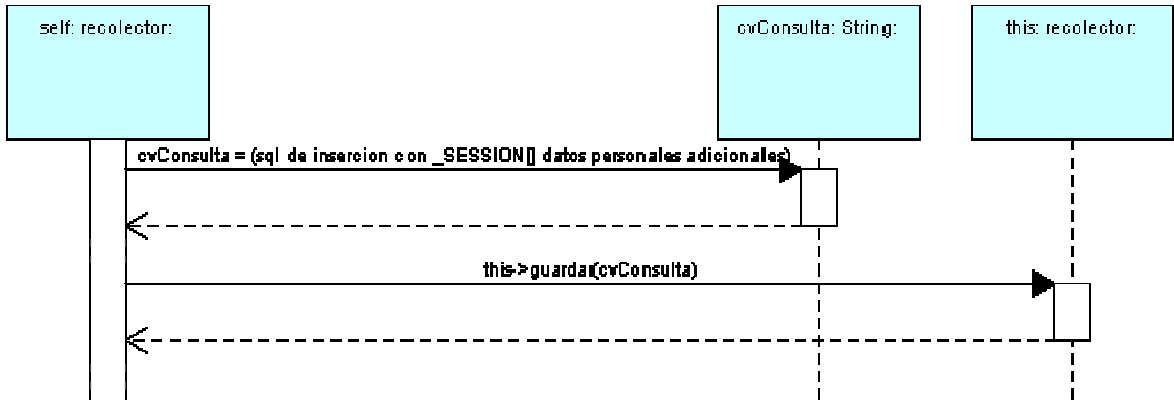


Guardar paquete "persona" en base de datos  
(php: clase recolector)

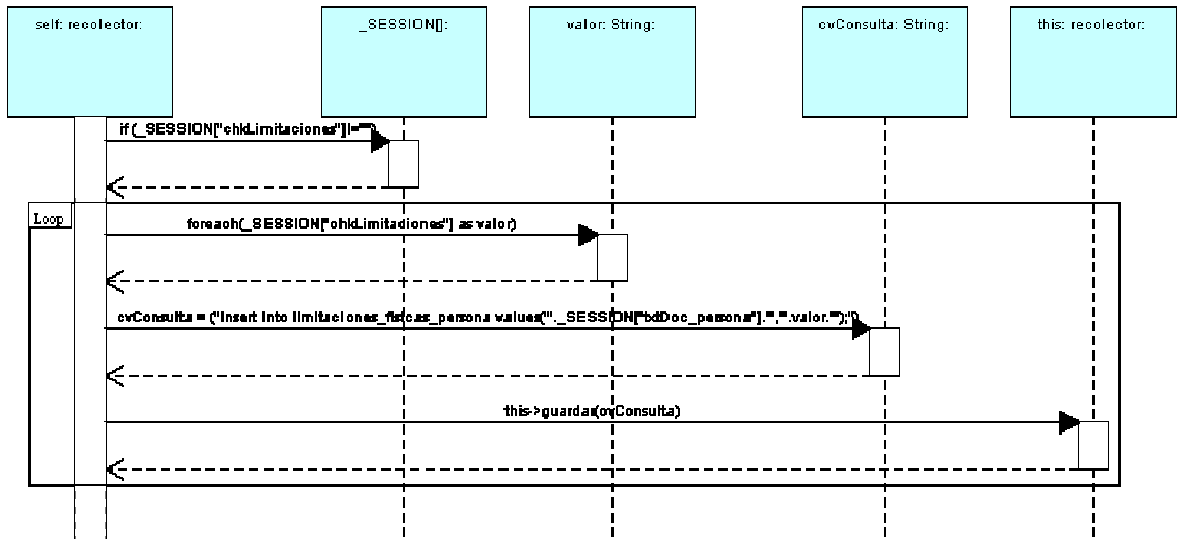




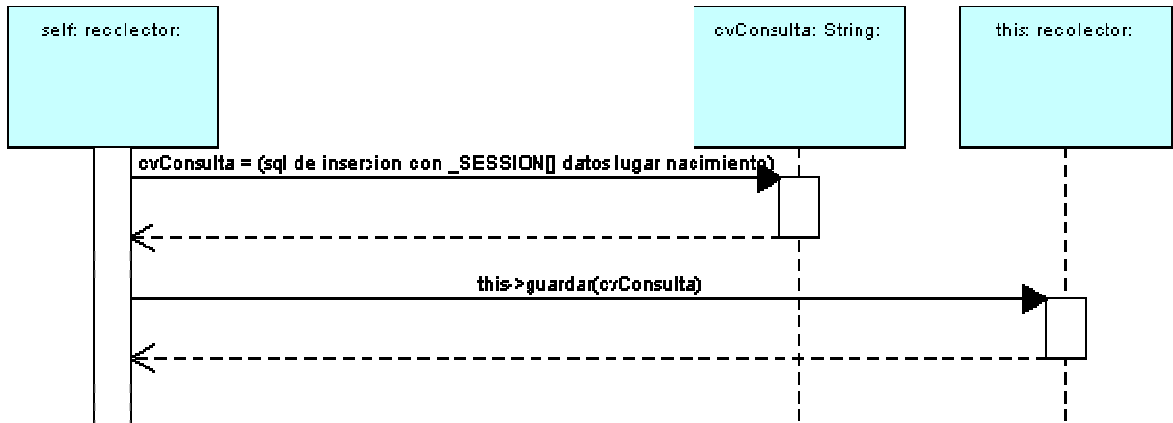
Guardar paquete "info persona" en base de datos  
(php: clase recolector)



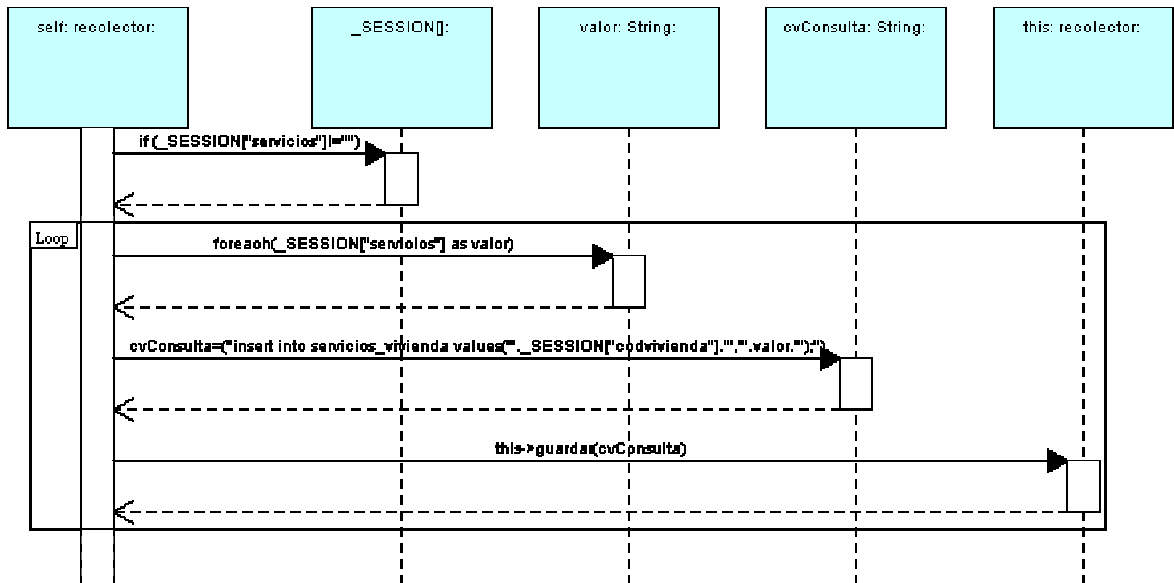
### Guardar paquete "limitaciones" en base de datos (php: clase recolector)



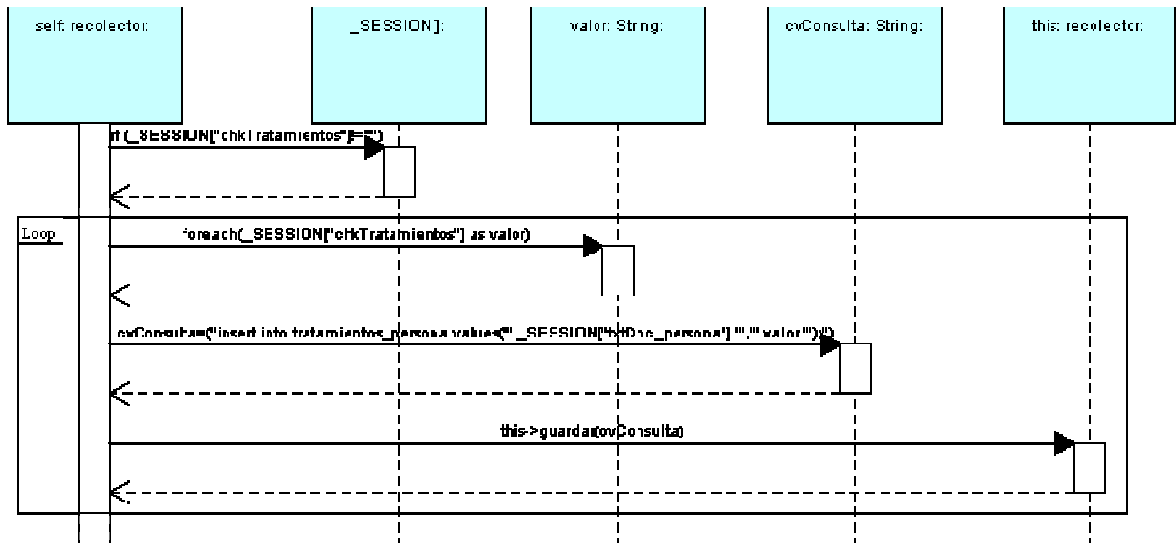
Guardar paquete "lugar nacimiento" en base de datos  
(php: clase recolector)



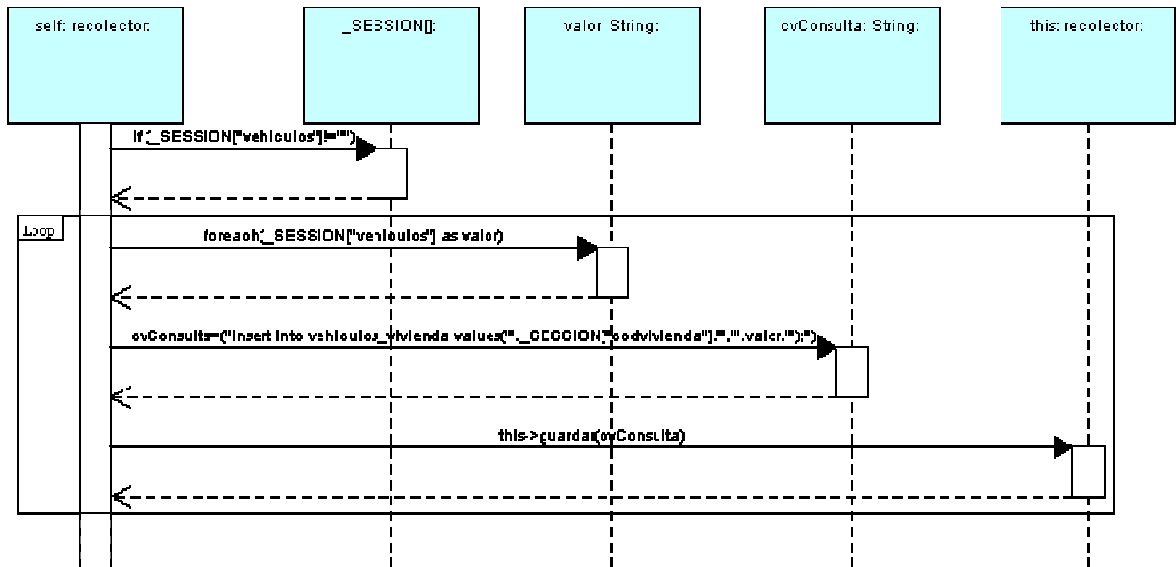
### Guardar paquete "servicios publicos" en base de datos (php: clase recolector)



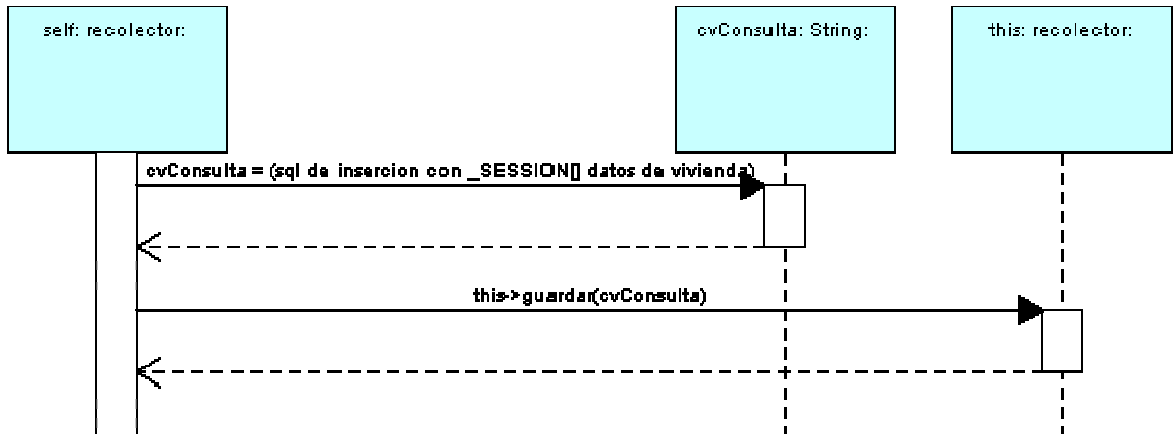
## Guardar paquete "tratamientos" en base de datos (php: clase recolector)



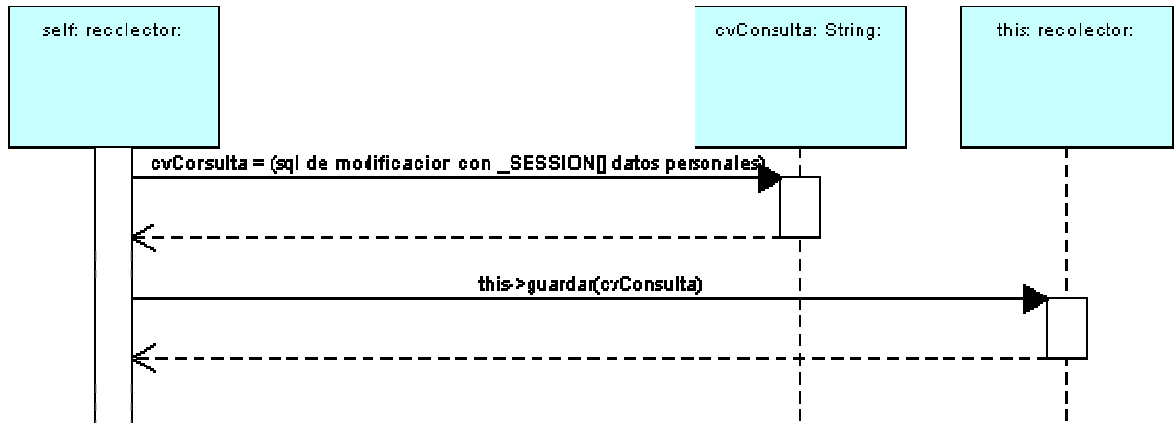
### Guardar paquete "vehiculos" en base de datos (php: clase recolector)



Guardar paquete "vivienda" en base de datos  
(php: clase recolector)

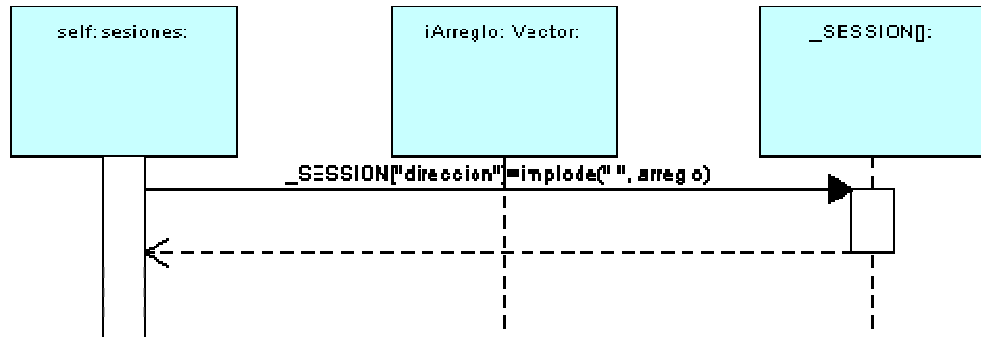


Modificar información de persona en base de datos  
(php: clase recolector)

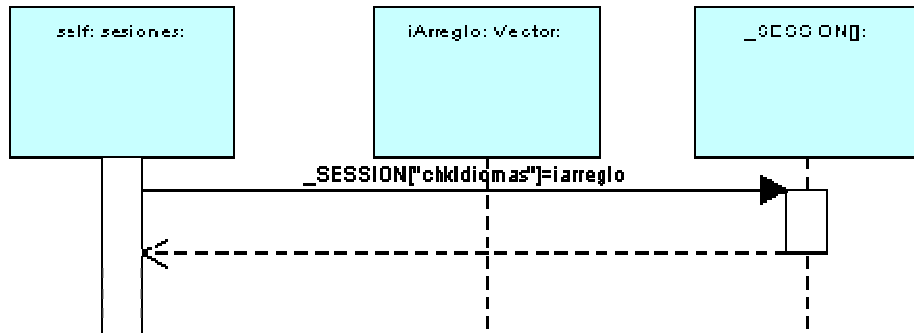




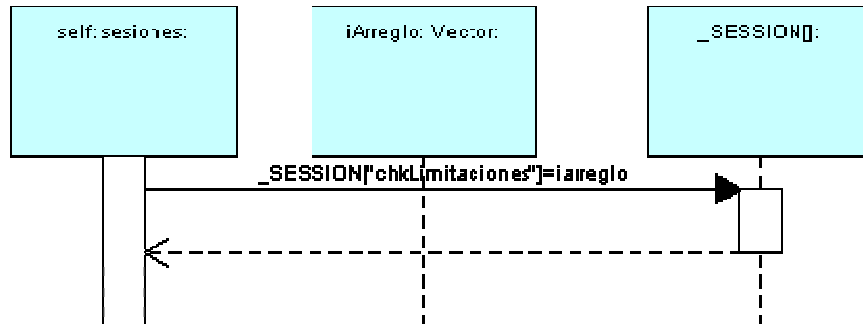
Cargar dirección de la vivienda en datos se sesión  
(php: clase sesiones)



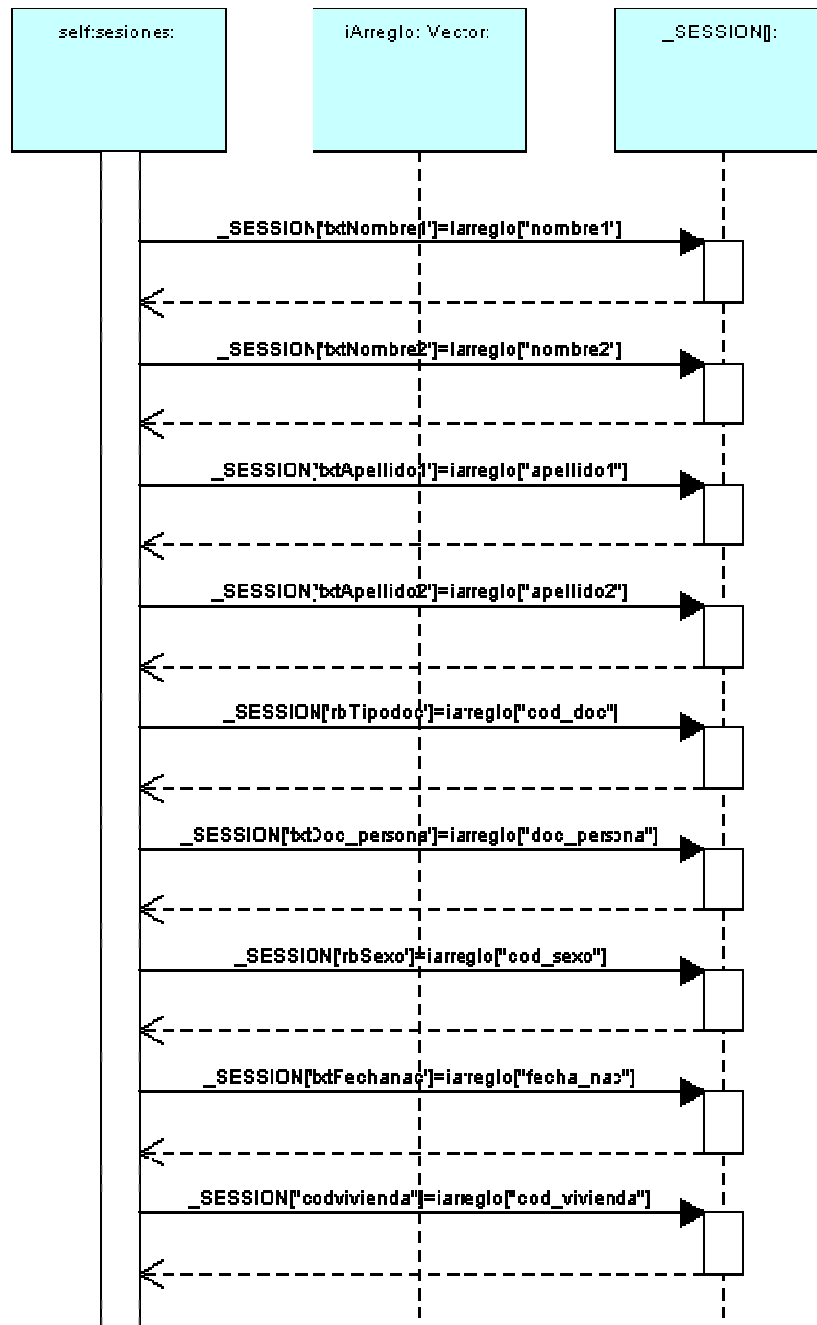
Cargar idiomas de la persona en datos de sesión  
(php: clase sesiones)



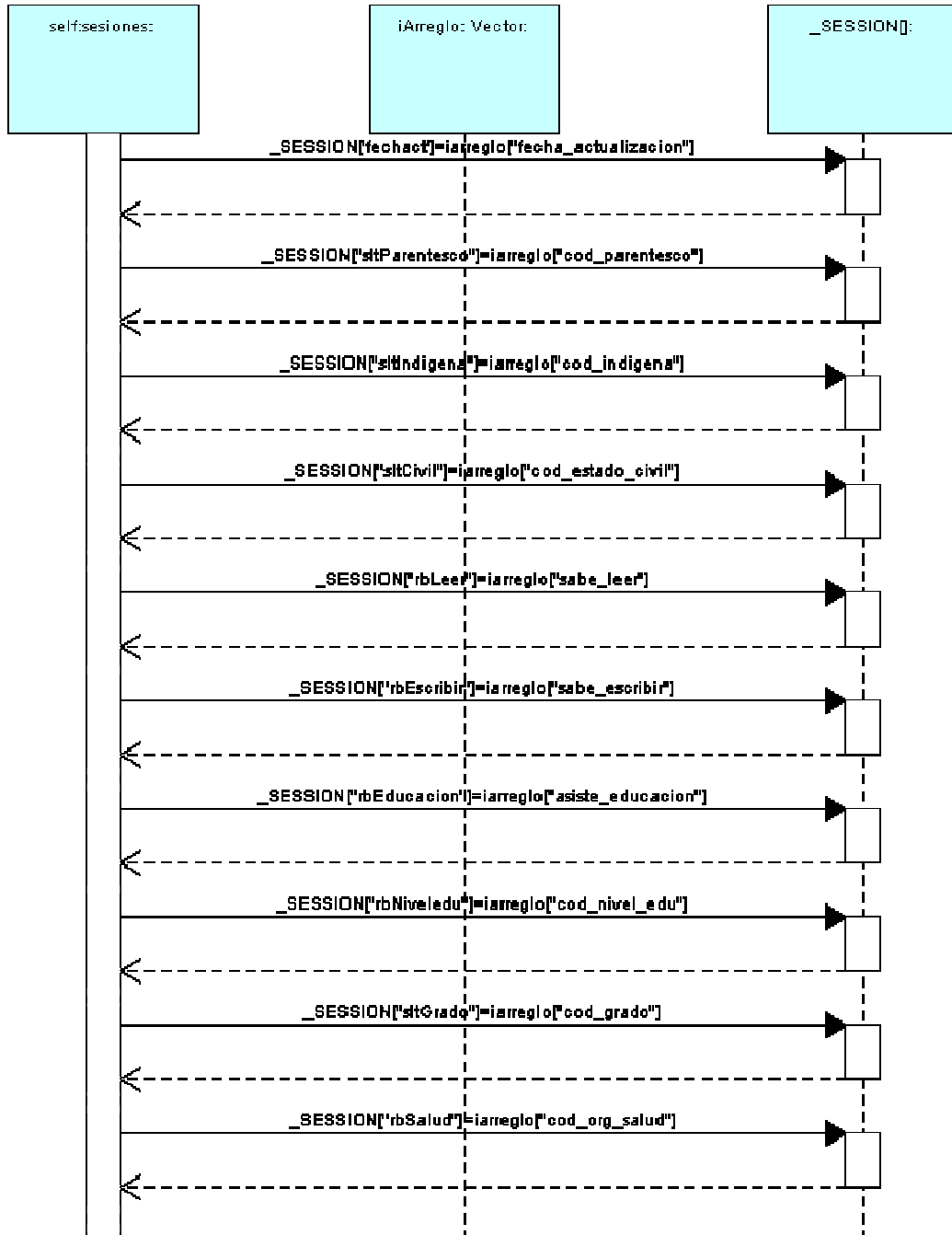
Cargar limitaciones físicas de la persona en datos de sesión  
(php: clase sesiones)



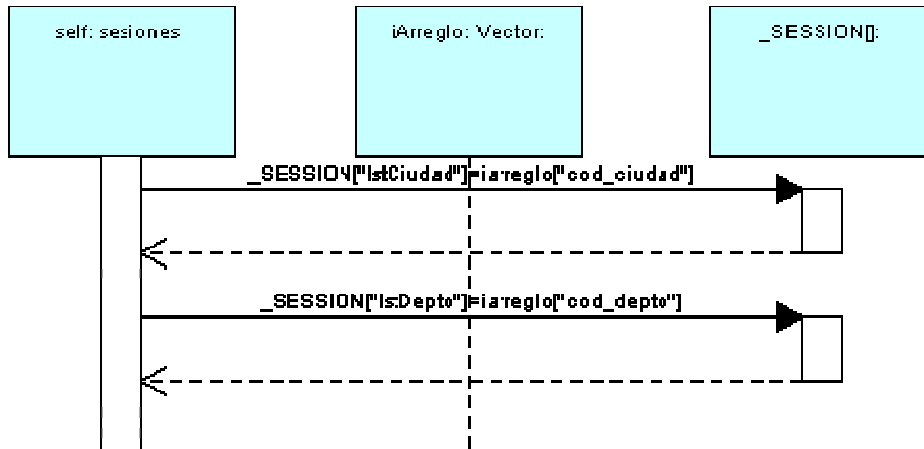
Cargar datos básicos de persona en datos de sesión  
(php: clase sesiones)



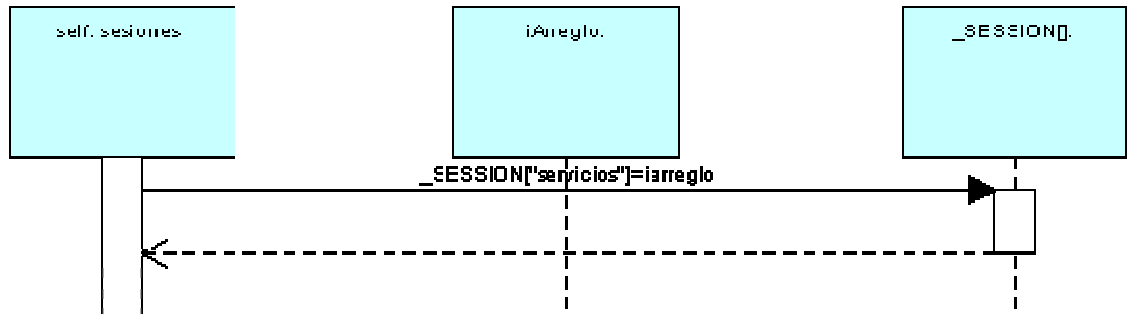
Cargar datos avanzados (adicionales) de persona en datos de sesión  
(php: clase sesiones)



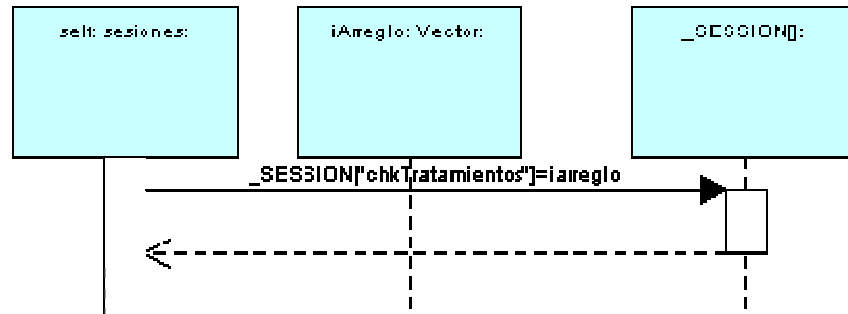
Cargar lugar de nacimiento de la persona en datos de sesión  
(php: clase sesiones)



Cargar servicios públicos de vivienda en datos se sesión  
(php: clase sesiones)

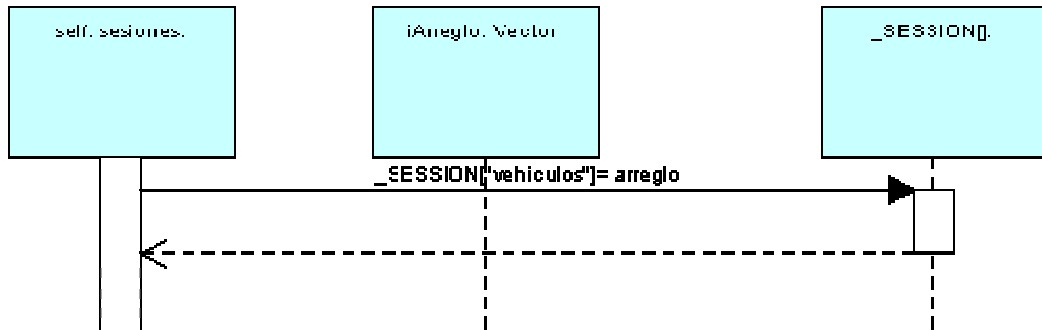


Cargar tratamientos médicos especiales de la persona en datos de sesión  
(php: clase sesiones)

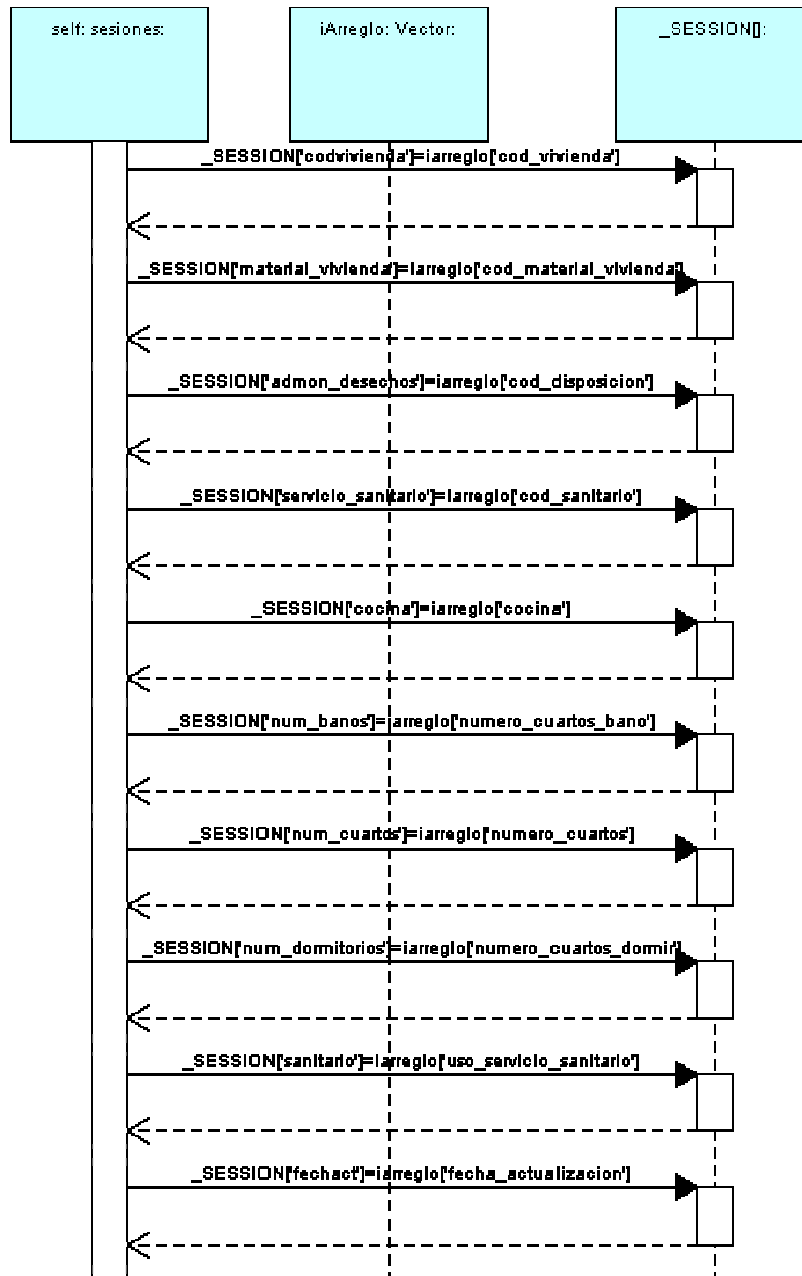




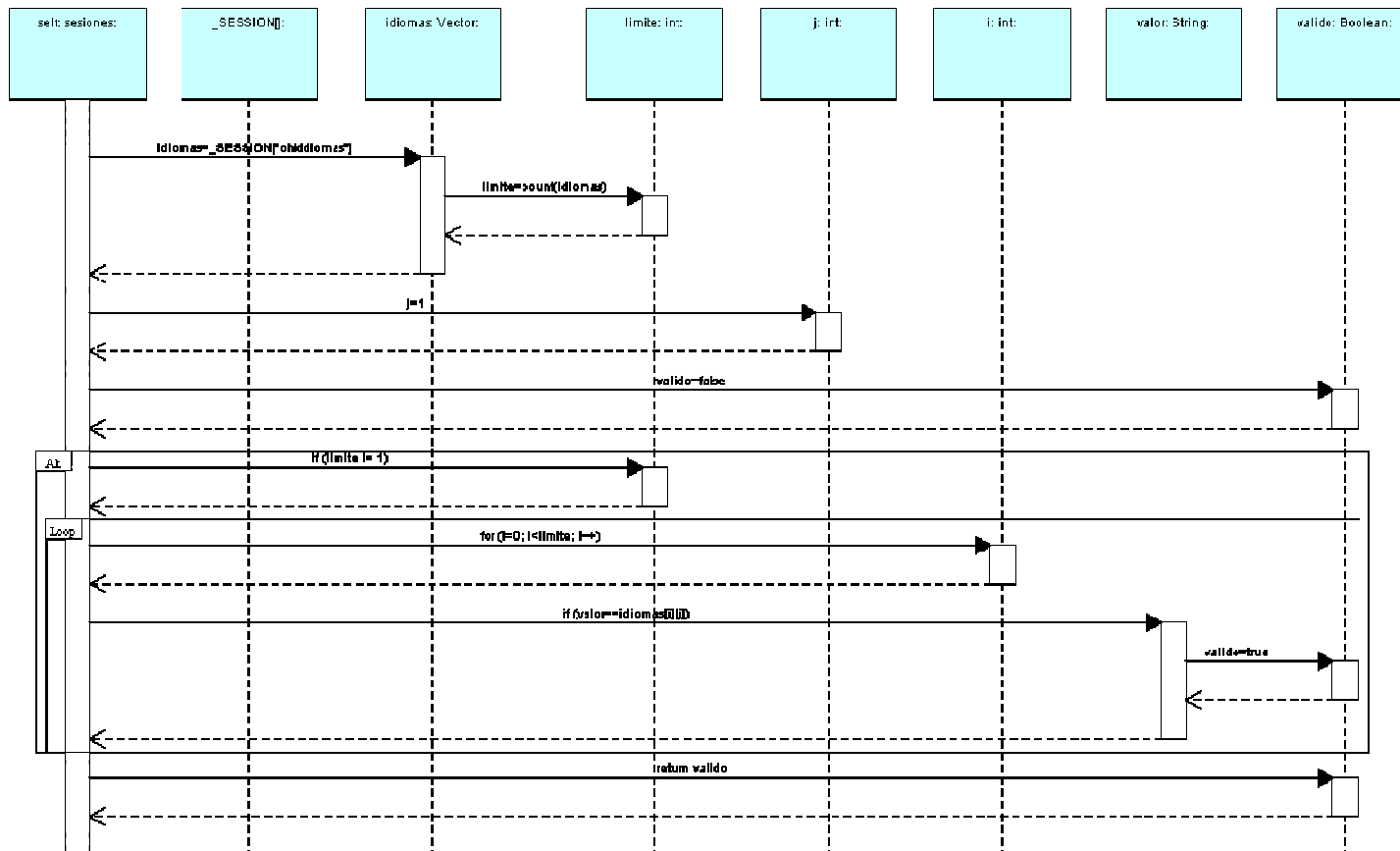
Cargar vehículos en datos de sesión  
(php: clase sesiones)



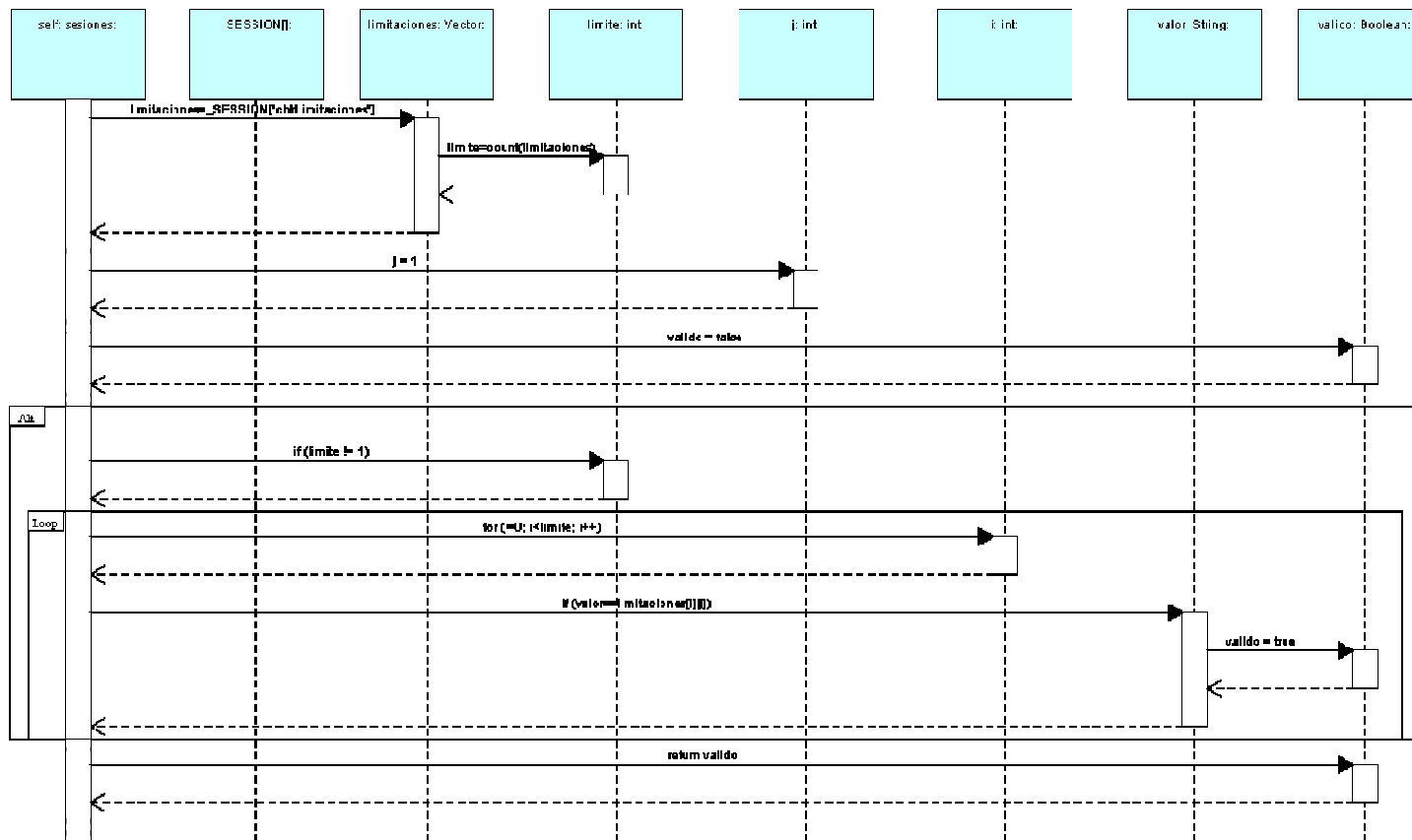
### Cargar vivienda en datos de sesión (php: clase sesiones)



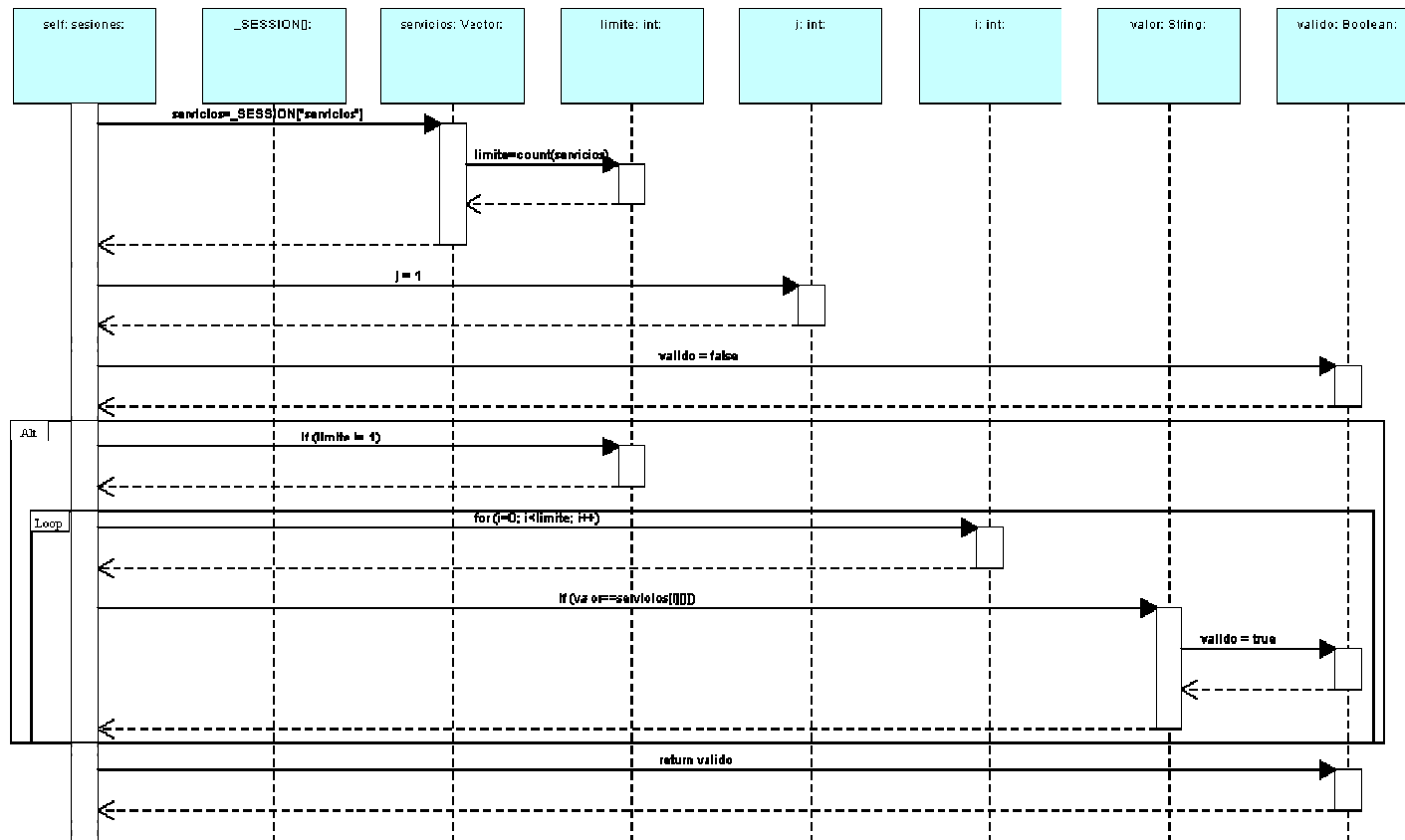
### Marcar para la página Web los idiomas de la persona (php: clase sesiones)



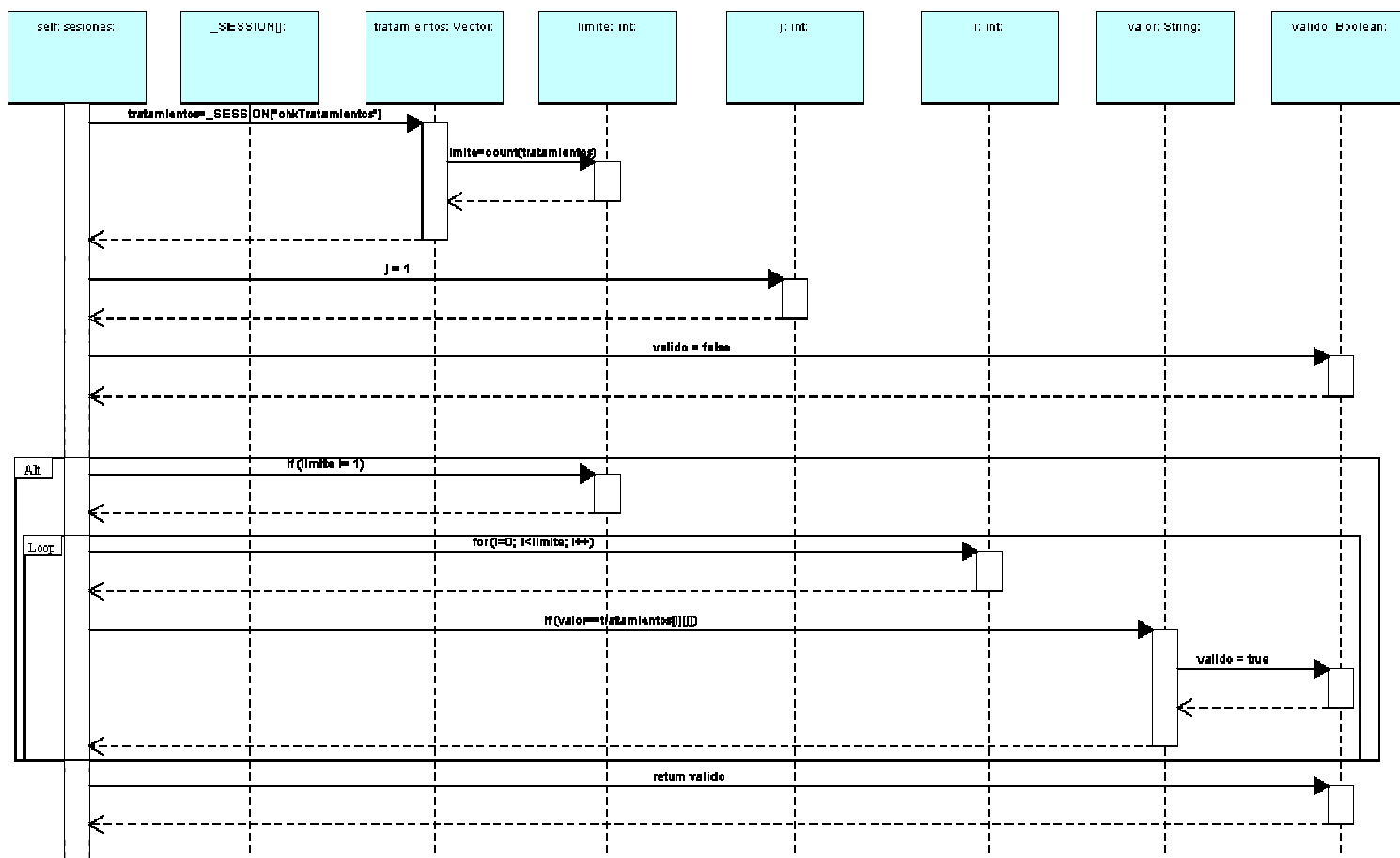
Marcar para la página web las limitaciones físicas de la persona  
(php: clase sesiones)



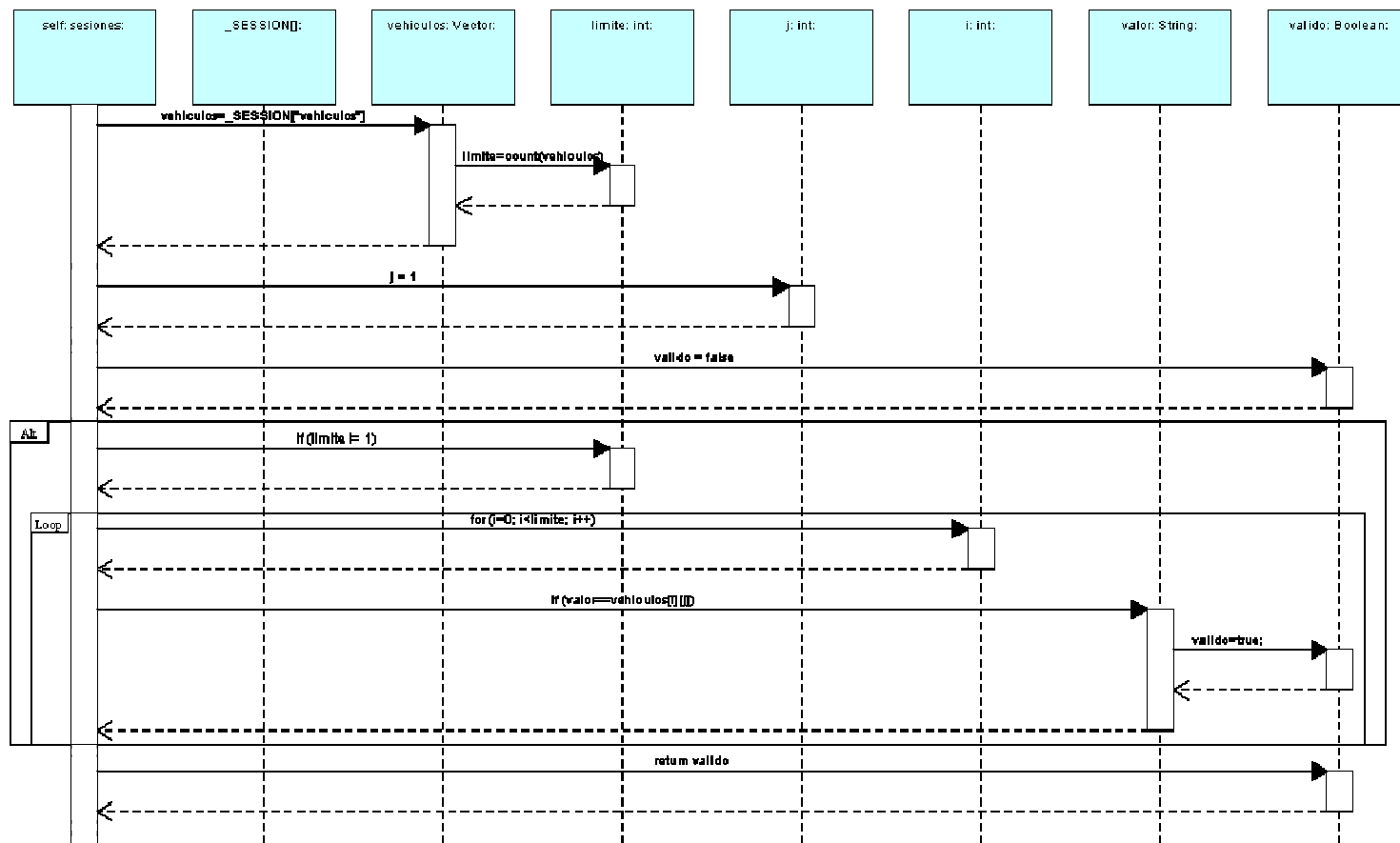
## Marcar para la página web los servicios públicos de la vivienda (php: clase sesiones)



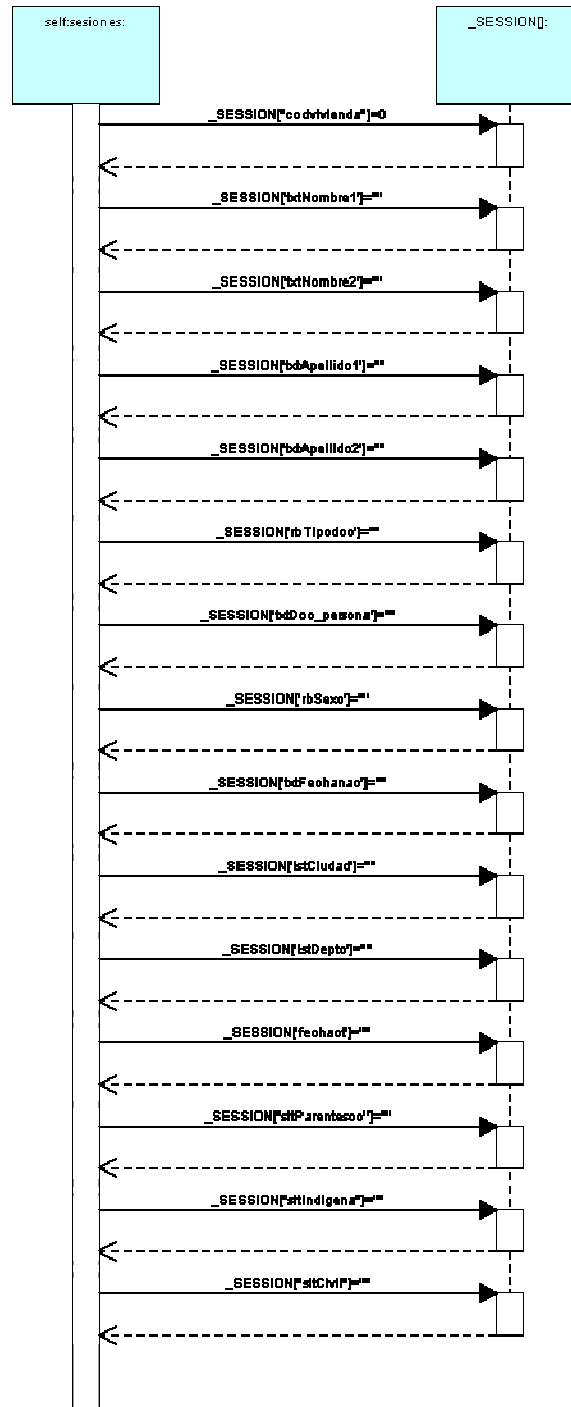
Marcar para la página web los tratamientos médicos de la persona  
(php: clase sesiones)



### Marcar sobre la página Web los vehículos de la vivienda (php: clase sesiones)

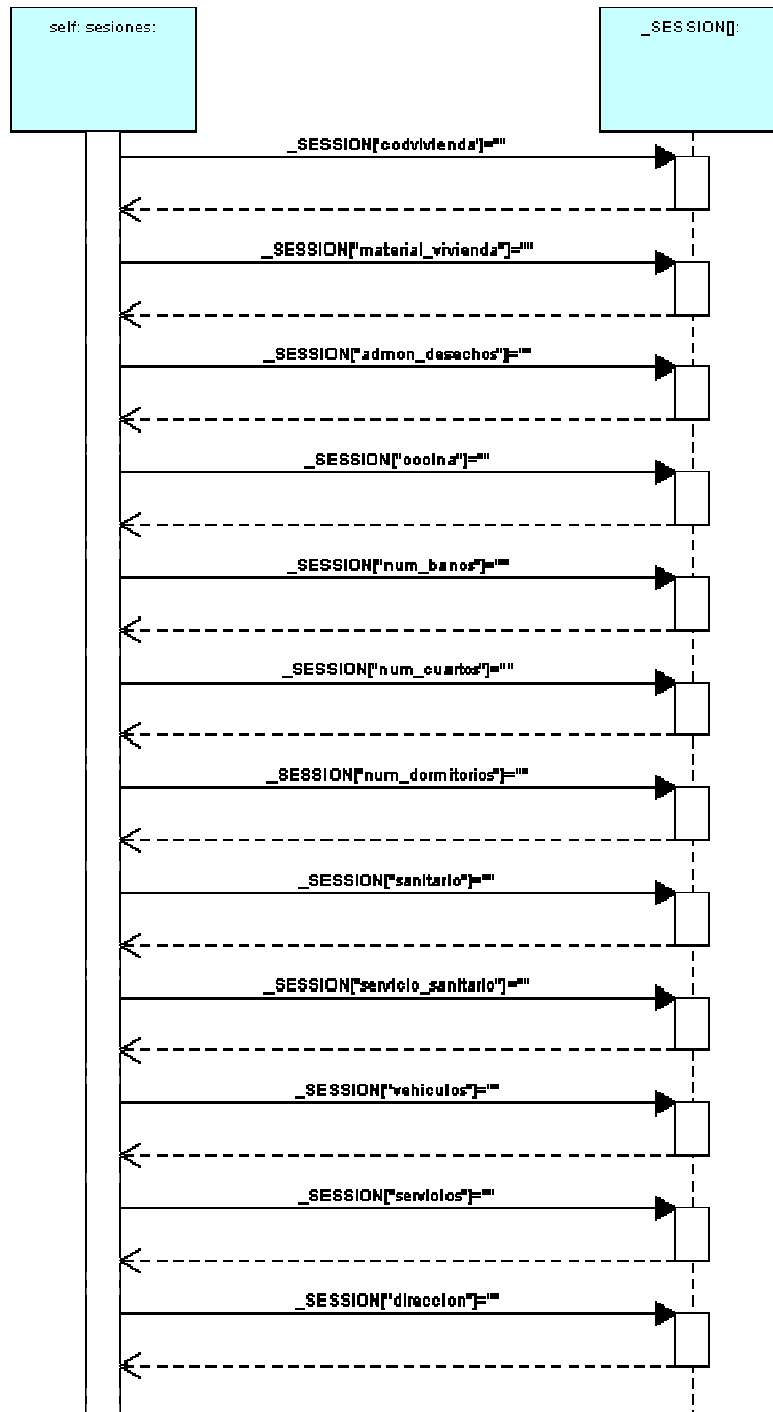


Borrar datos de persona de los datos de sesión  
(php: clase sesiones)



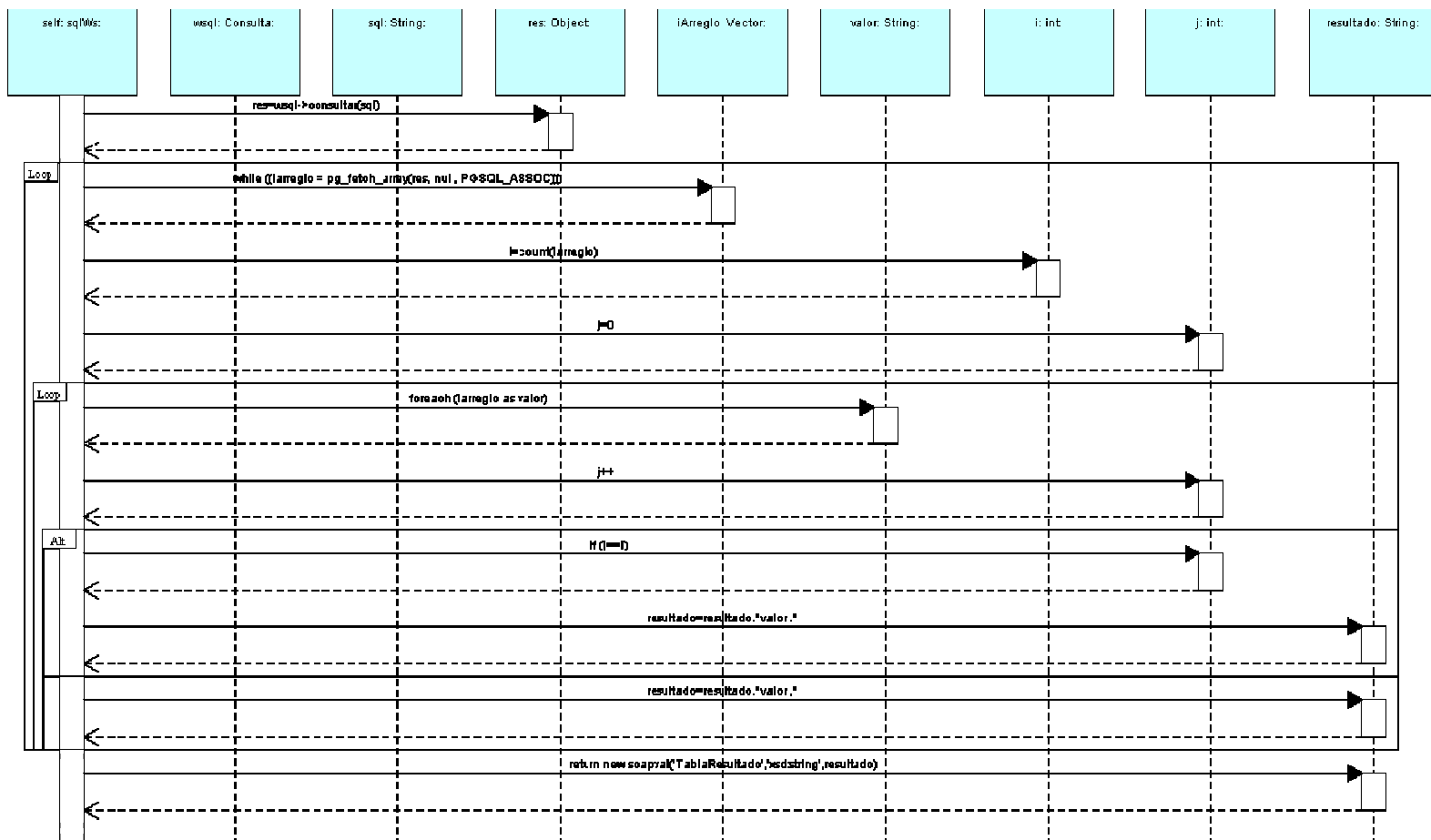


Borrar datos de vivienda de los datos de sesión  
(php: clase sesiones)

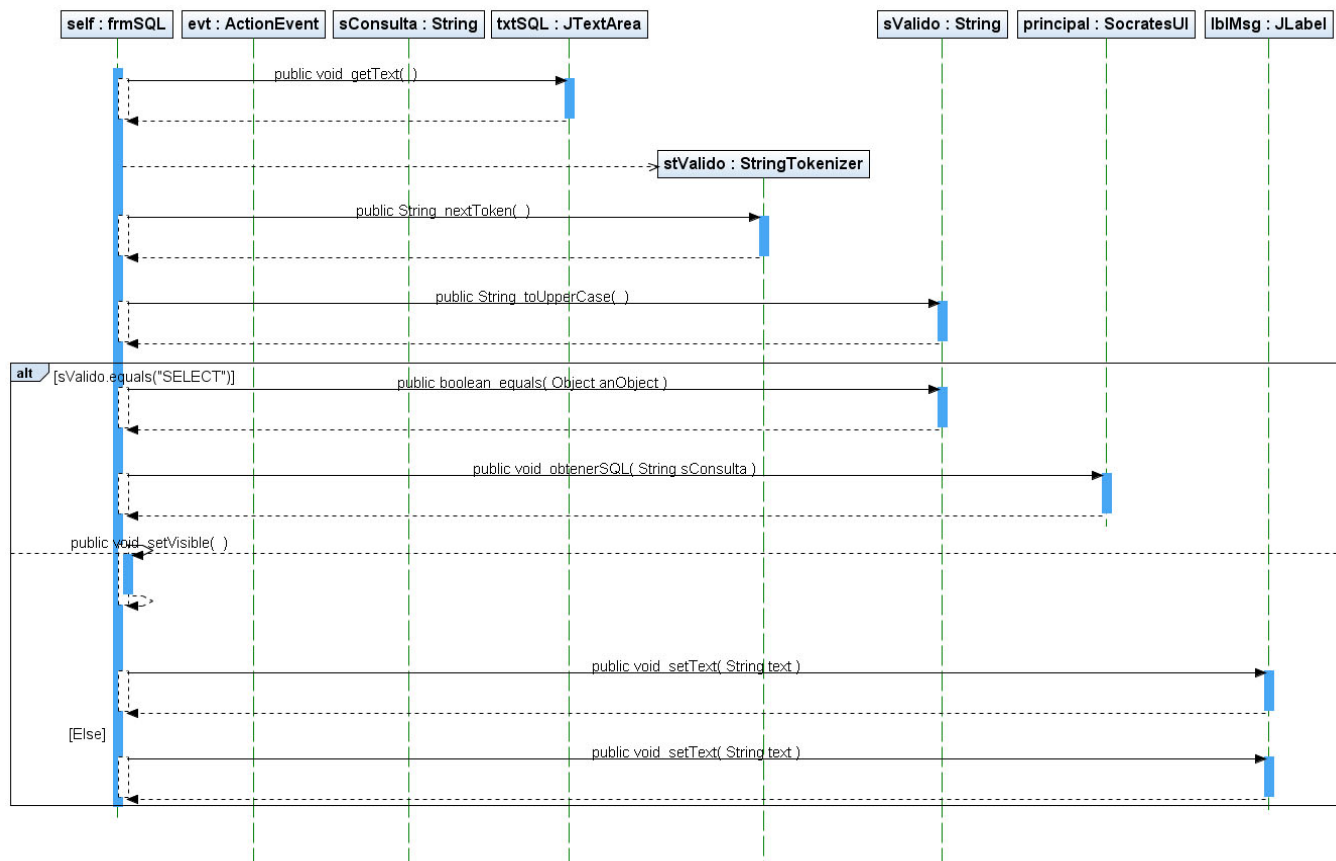




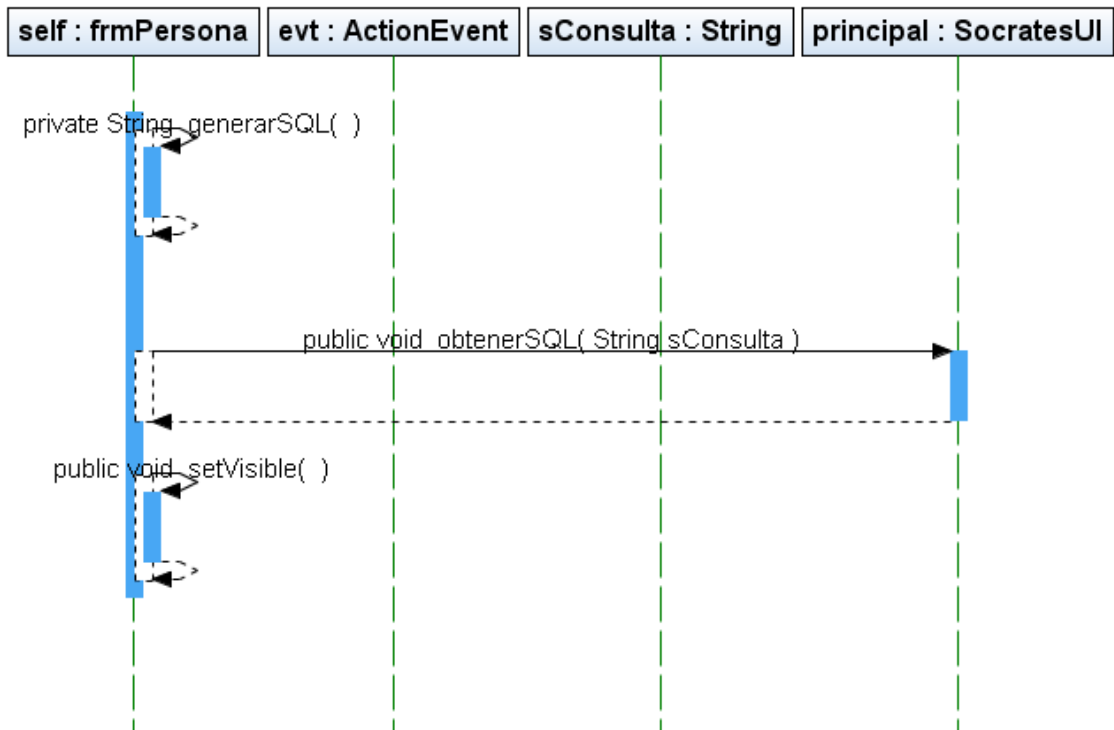
## Ejecución de consulta y entrega de resultados al cliente mediante webservice (php: clase sqlWs)



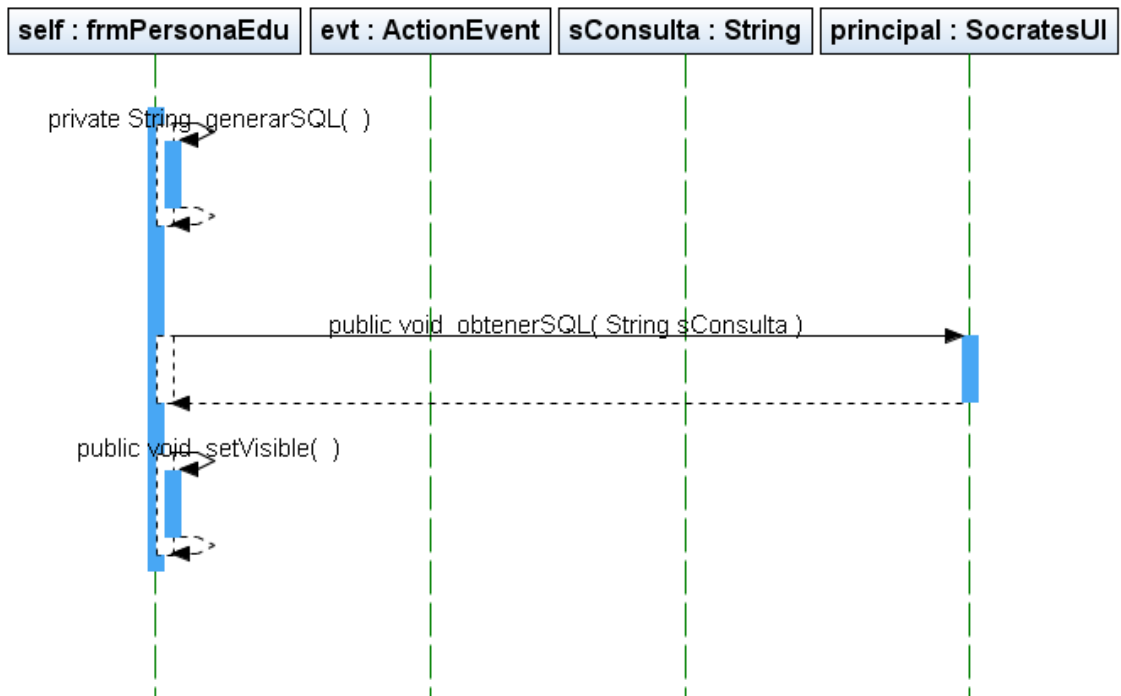
## Envío de código sql modo avanzado a la ventana principal del cliente (java: clase frmSql)



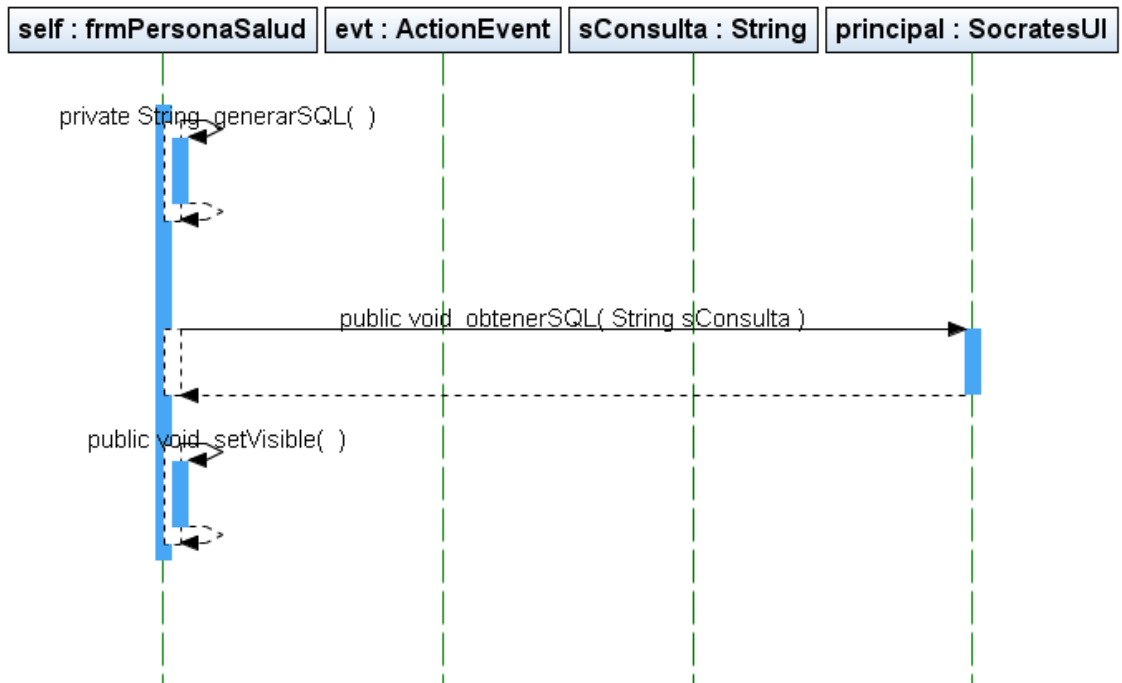
Envío de código generado de consulta tipo persona-básico a la ventana principal del cliente  
(java: clase frmPersona)



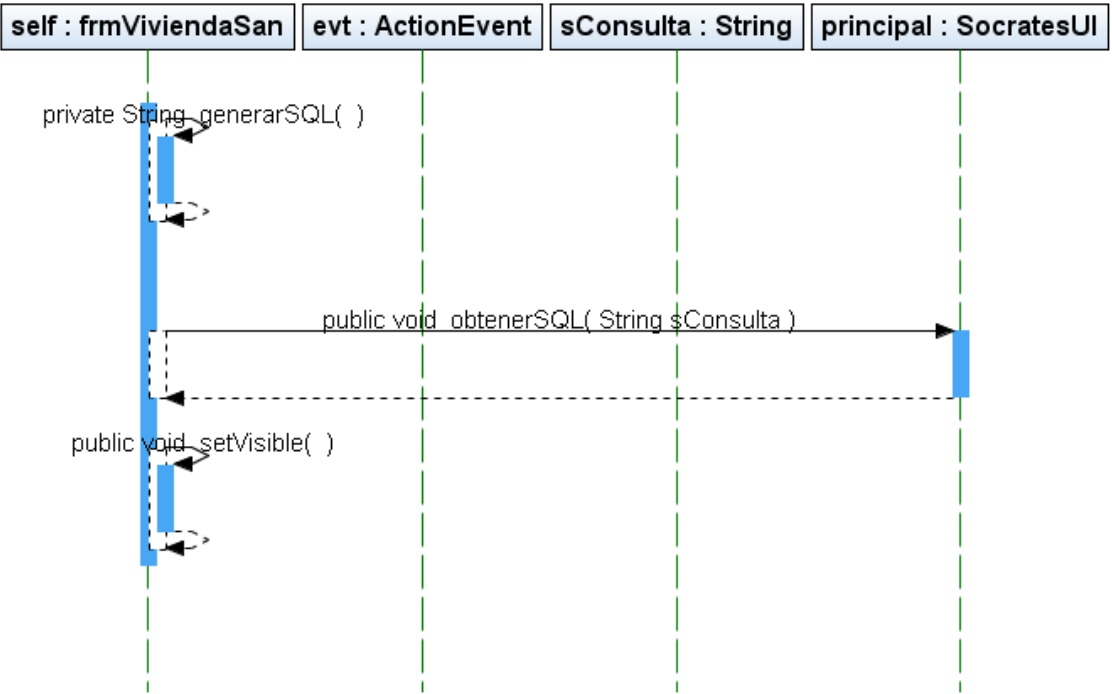
Envío de código generado de consulta tipo persona-educación a la ventana principal del cliente  
(java: clase frmPersonaEdu)



Envío de código generado de consulta tipo persona-salud a la ventana principal del cliente  
(java: clase frmPersonaSalud)

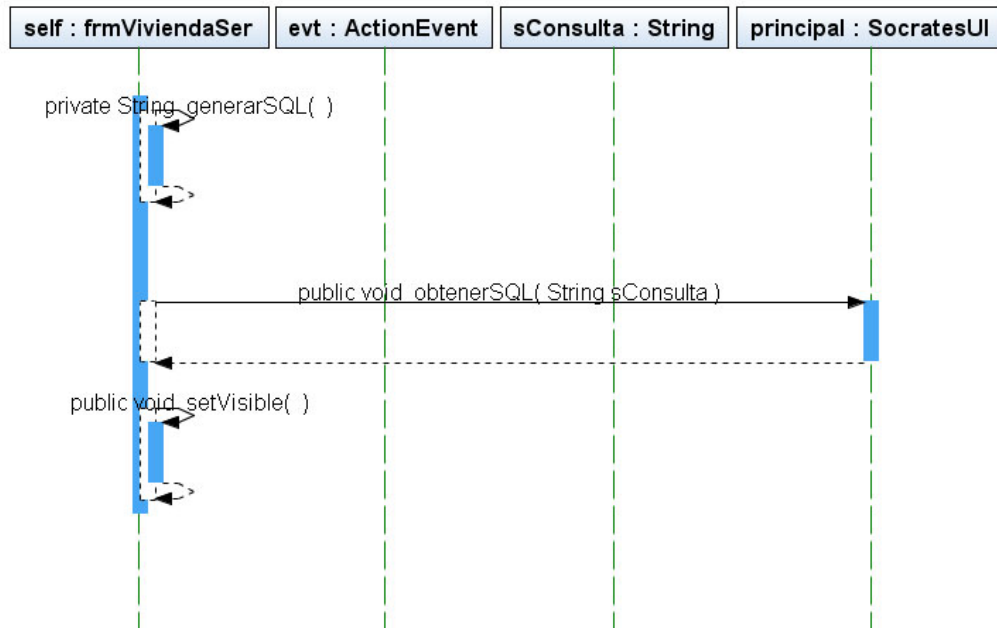


Envío de código generado de consulta tipo vivienda-sanidad a la ventana principal del cliente  
(java: clase frmViviendaSan)

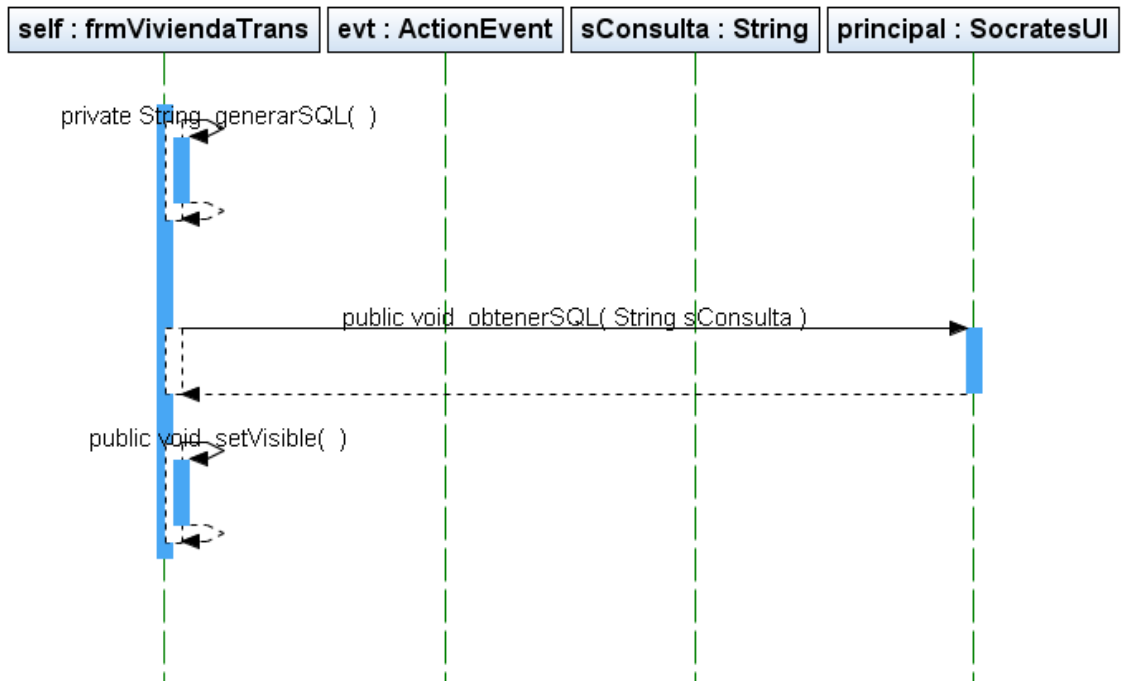




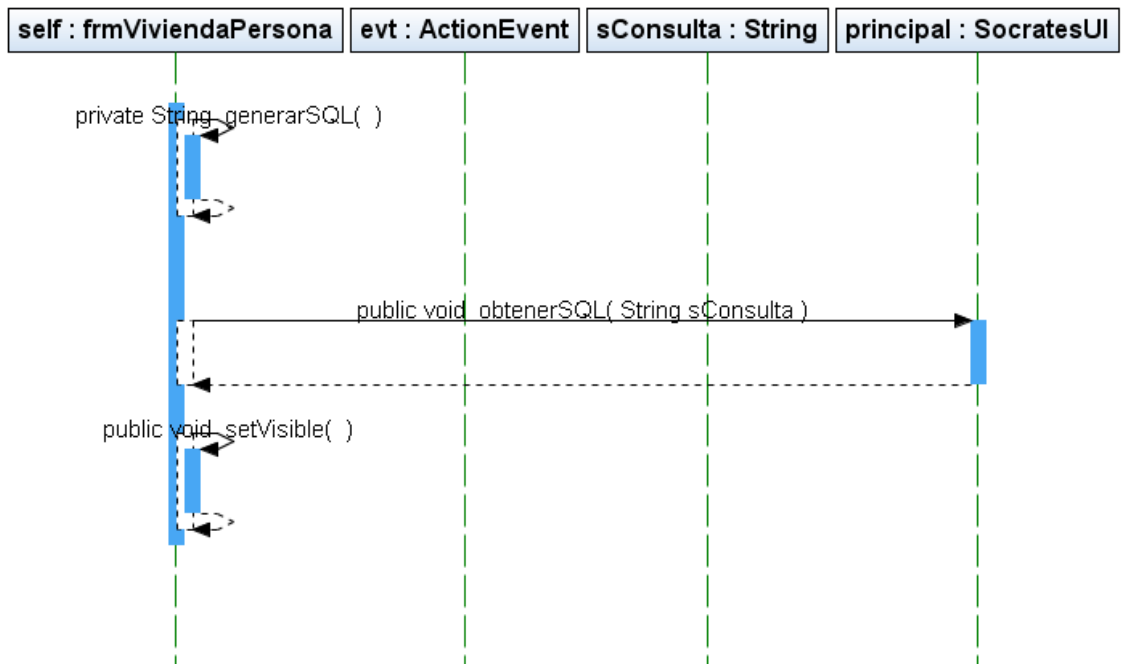
Envío de código generado de consulta tipo vivienda-servicios a la ventana principal del cliente  
(java: clase frmViviendaSer)



Envío de código generado de consulta tipo vivienda-transporte (vehículos) a la ventana principal del cliente (java: clase frmViviendaTrans)



Envío de código generado de consulta tipo vivienda-persona a la ventana principal del cliente  
(java: clase frmViviendaPersona)





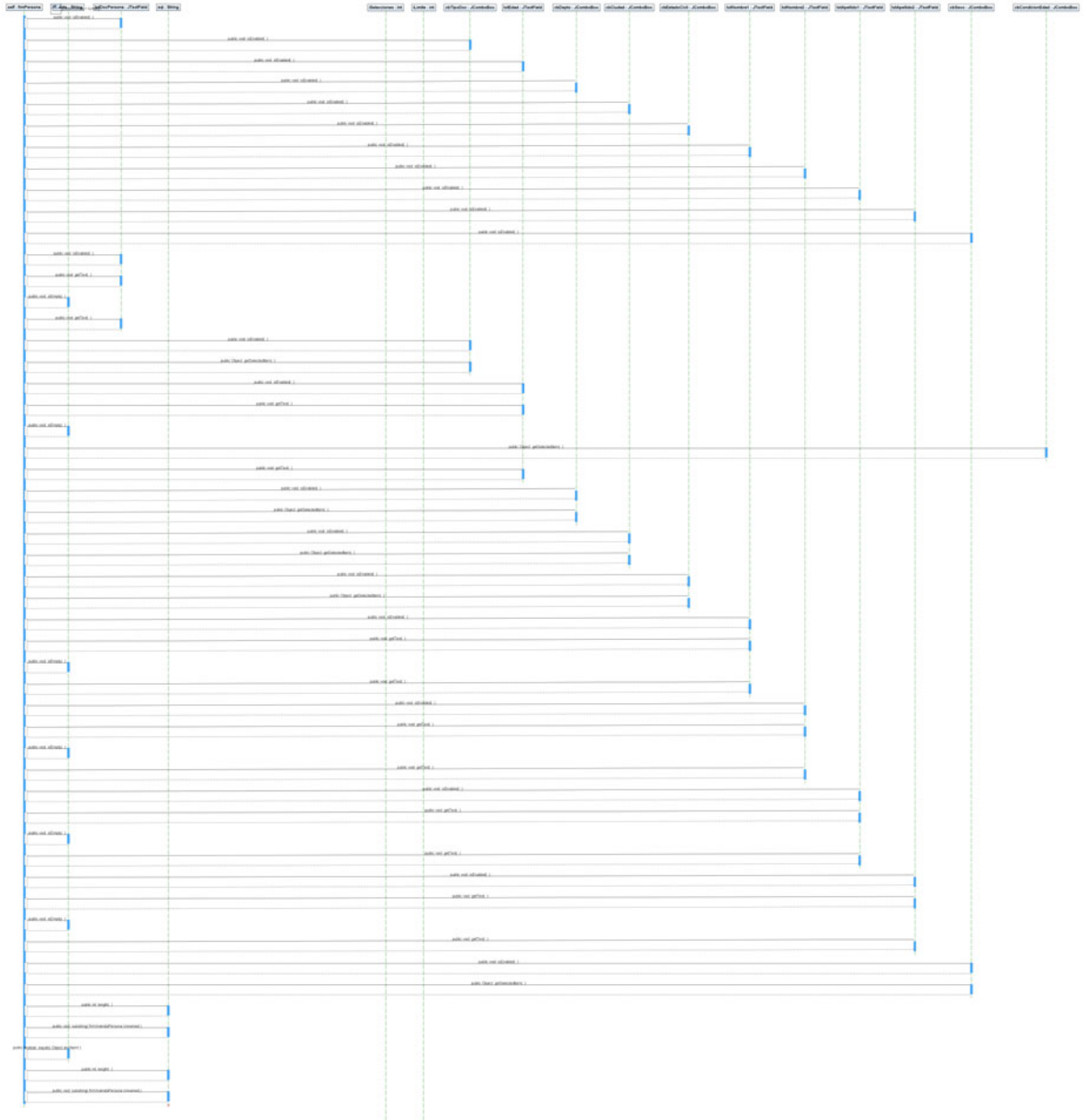


Ejecutar código sql (generado o modo avanzado) en la ventana del cliente hacia el webservice y recibir los resultados  
(java: clase SócratesUI)





# Generar código sql desde formulario de datos básicos de persona (java: clase frmPersona)

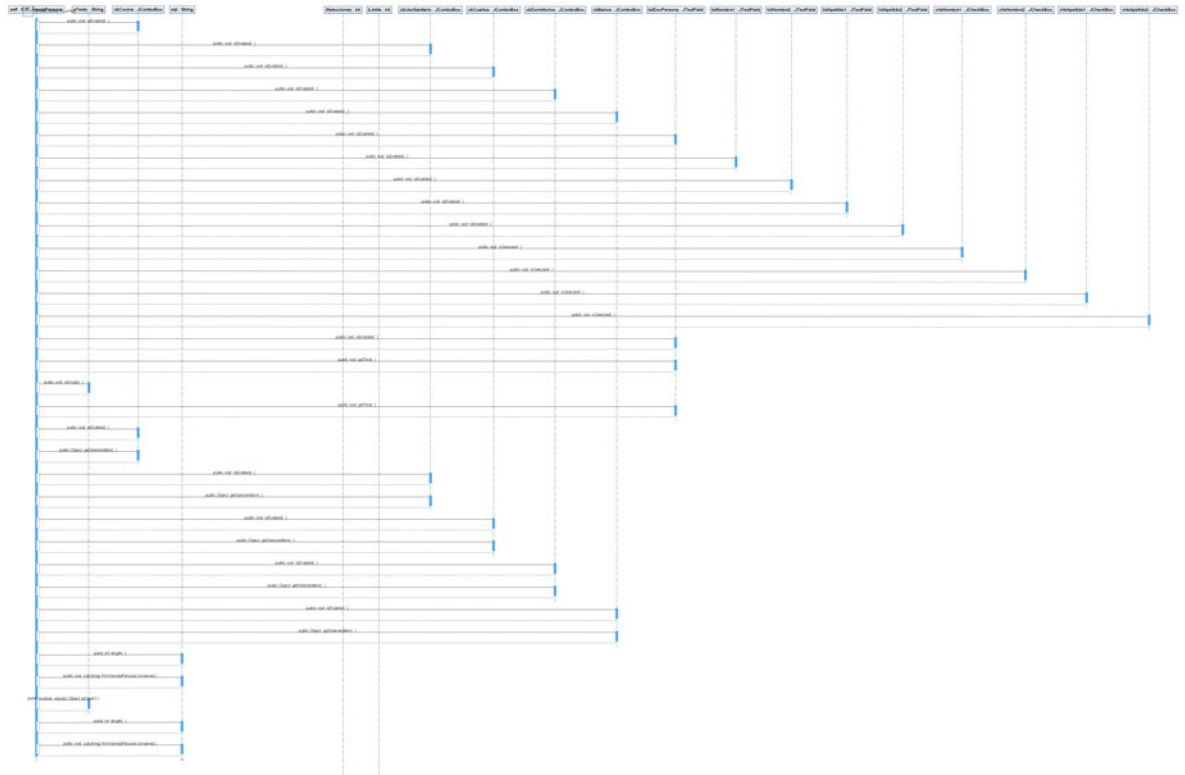




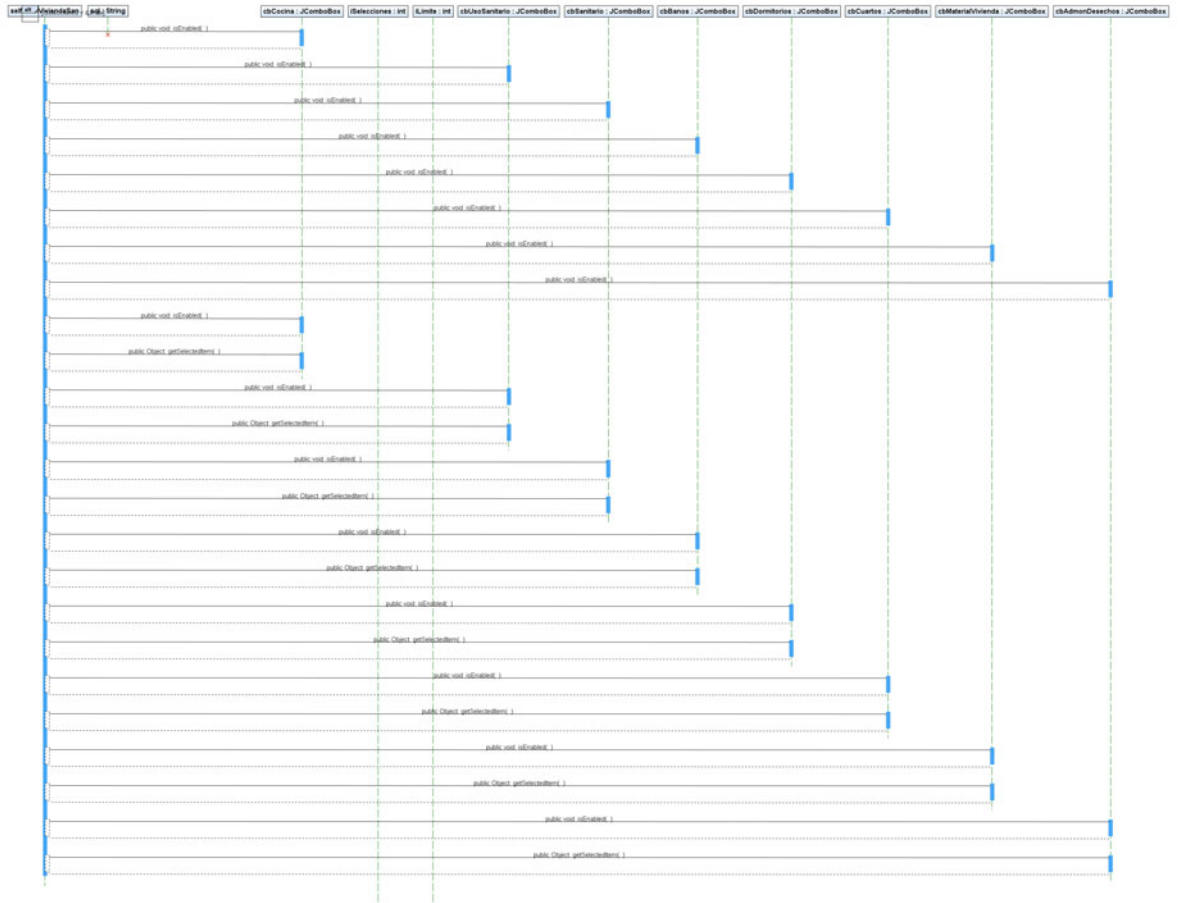




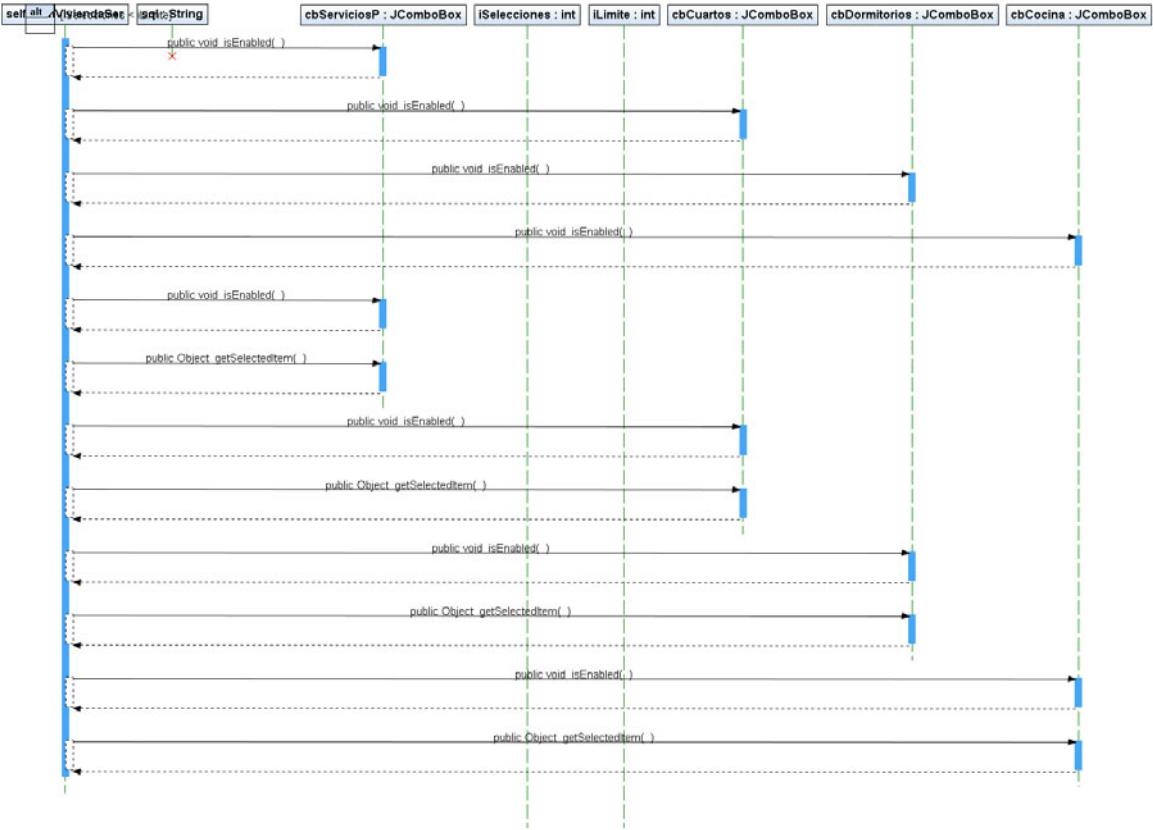
## Generar código sql desde formulario de datos de vivienda-persona (java: clase frmViviendaPersona)



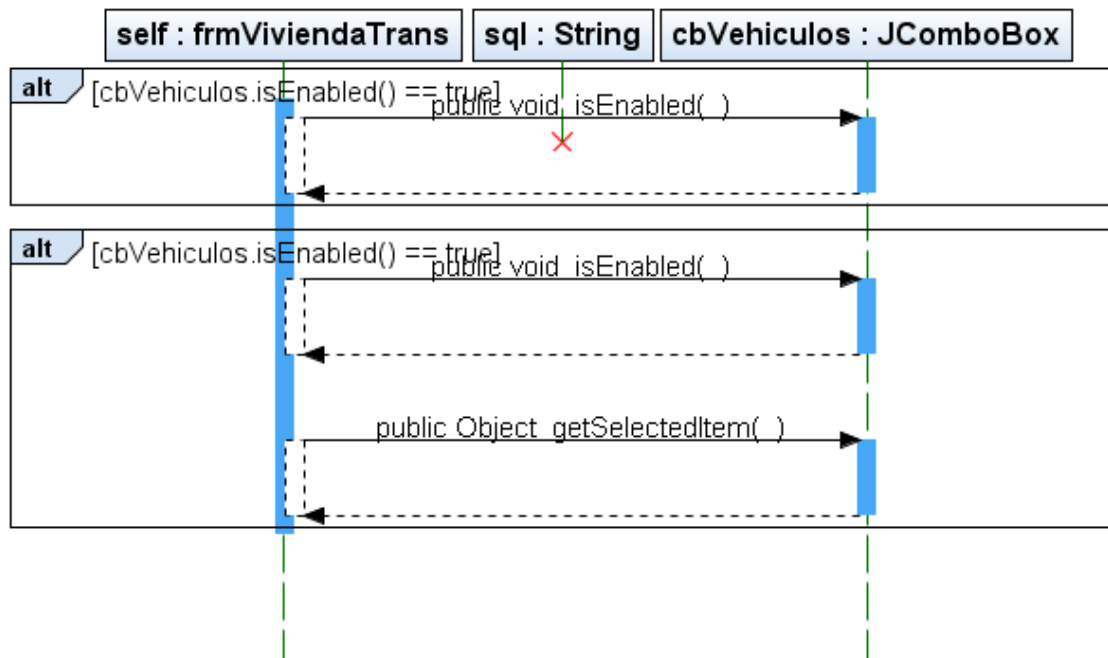
## Generar código sql desde formulario de datos de vivienda-Sanidad (java: clase frmViviendaSan)



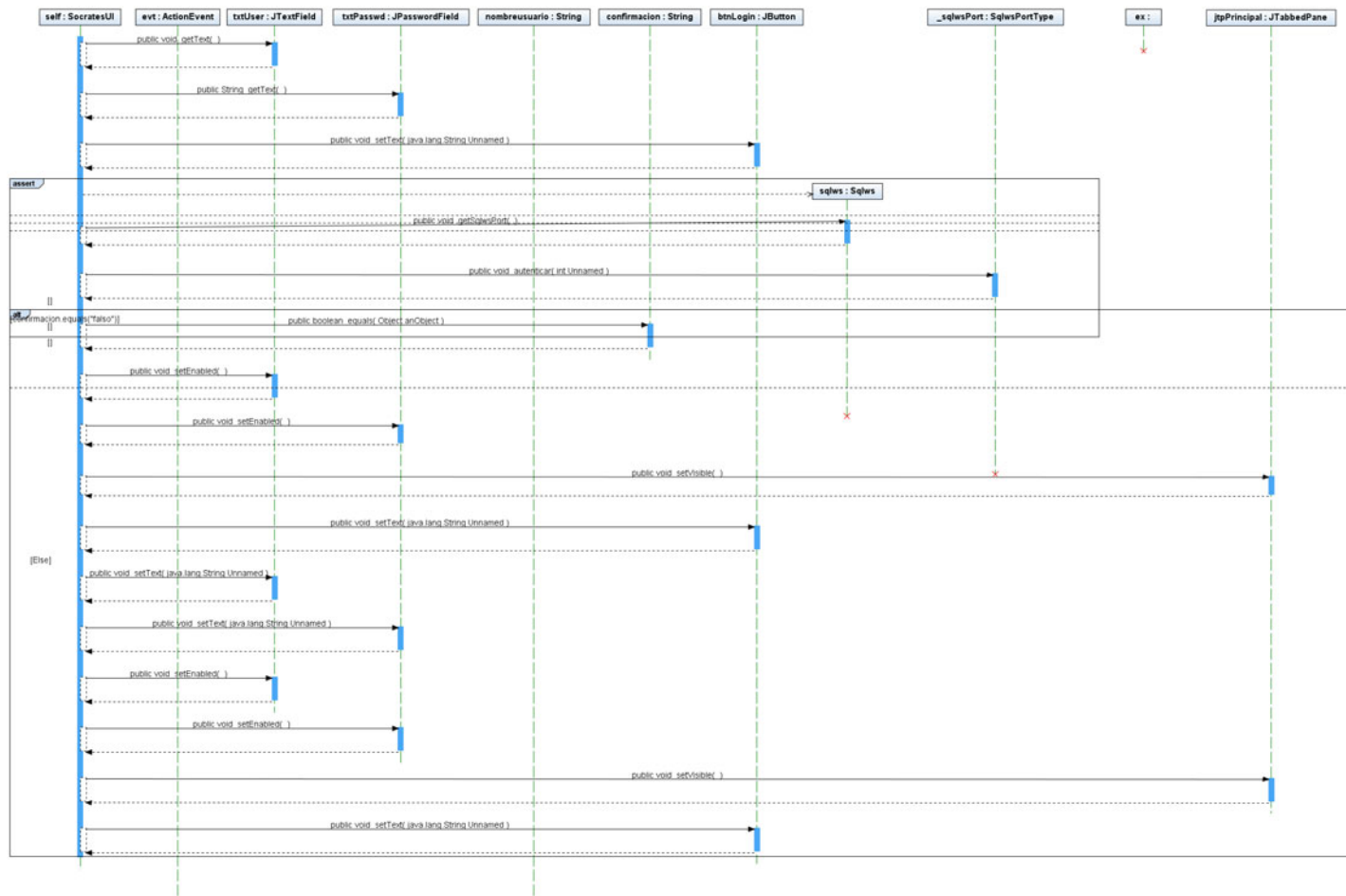
Generar código sql desde formulario de datos de vivienda-servicios públicos  
(java: clase frmViviendaSer)



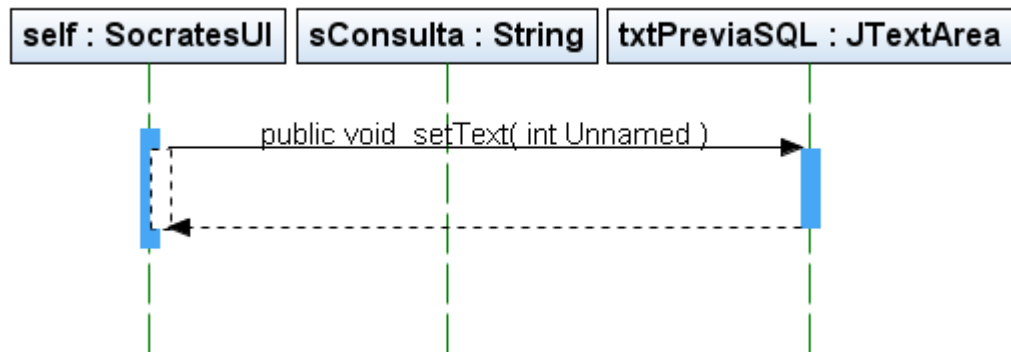
Generar código sql desde formulario de datos de vivienda-Transporte  
(java: clase frmViviendaTrans)



Conectar el cliente con el webservice / verificar nombre de usuario y contraseña de acceso  
(java: clase socratesUI)

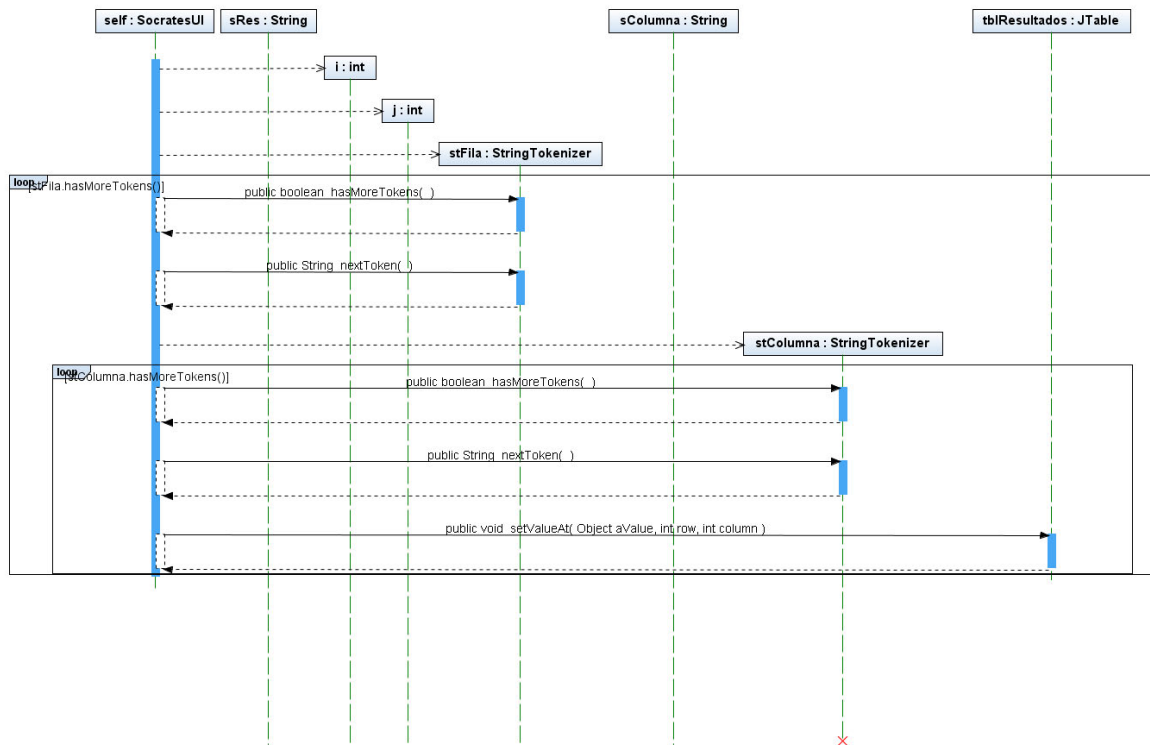


Obtener código sql generado desde cualquiera de los formularios y mostrarlo en  
vista previa  
(java: clase socratesUI)





Transformar cadena de resultados que viene del *Web service* en tabla de resultados en el cliente (java: clase *socratesUI*)



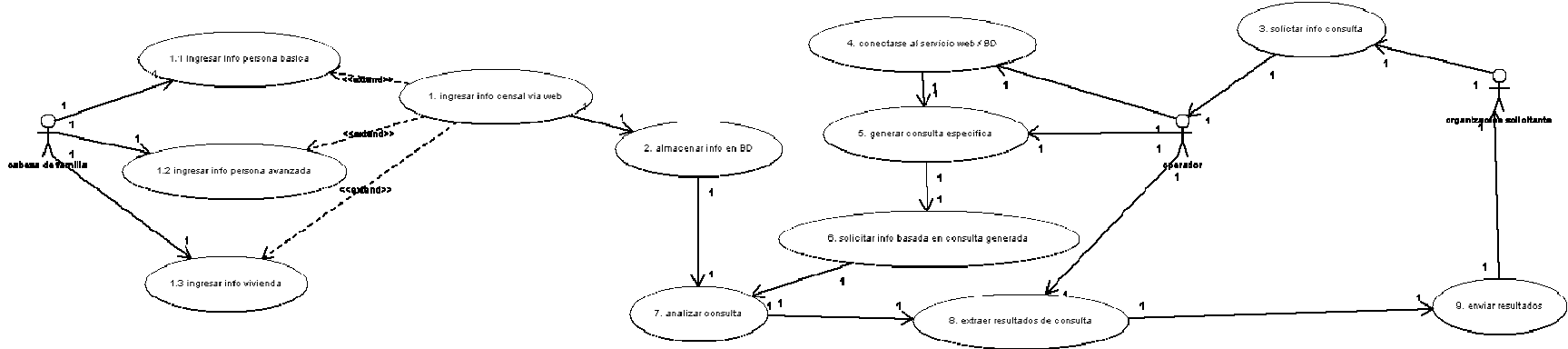
2.6 MODELO FUNCIONAL  
DIAGRAMA DE CASOS DE USO

SOCRATES-EC: SISTEMA DE INFORMACIÓN POBLACIONAL BASADO EN  
PROCEDIMIENTOS DE CENSADO

Por:

Danko Jiménez Londoño  
Daniel Andrés Valencia V.

### 2.6.1 Diagrama de casos de uso Sócrates-EC



## Tarjeta Descriptiva Caso de Uso 1

ID Caso de Uso:	1
Nombre del Caso de Uso:	Ingresar información censal vía Web.
Creado Por:	Daniel Andres Valencia V. & Danko Jiménez L.
Fecha de Creación:	15/10/2007
Actores:	Cabeza de familia
Descripción:	La persona encargada del hogar ingresa la información personal, de sus miembros de familia, y de la vivienda en el sistema
Precondiciones:	<ol style="list-style-type: none"> <li>1. debe haber acceso a Internet disponible para acceder al sistema.</li> <li>2. la persona cabeza de familia puede ingresar / modificar la información con ayuda de un asistente.</li> </ol>
Poscondiciones:	la información debe almacenarse inmediatamente en la Base de datos
Flujo Normal:	La persona que es cabeza de familia (encargado del hogar) ingresa a la página Web y selecciona primer ingreso, aquí es donde ingresa por primera vez sus datos personales y los de la vivienda en la que habita.
Flujos Alternativos:	<ol style="list-style-type: none"> <li>1. una vez ingresados los datos de la primera persona y de la vivienda, se agrega la opción de agregar mas personas al sistema y se repite el ciclo por cada persona que se ingrese.</li> <li>2. en vez de seleccionar primer ingreso la persona puede seleccionar usuario existente, ingresar unos datos personales básicos para comprobar que existe en base de datos e inmediatamente hacer modificaciones.</li> </ol>
Excepciones:	El usuario no puede ingresar al sistema ya sea porque esta tratando de ingresar como nuevo cuando sus datos ya existen en base de datos o que haya ingresado mal los mismos al tratar de hacer modificaciones.
Incluidos:	
Prioridad:	ALTA
Frecuencia de Uso:	Una vez por familia (primer ingreso), N veces por familia (actualización de datos)
Requerimientos Especiales:	Conexión al servidor Web y a la base de datos

## Tarjeta Descriptiva Caso de Uso 1.1

ID Caso de Uso:	1.1
Nombre del Caso de Uso:	Ingresar datos personales básicos vía Web.
Creado Por:	Daniel Andres Valencia V. & Danko Jiménez L.
Fecha de Creación:	15/10/2007
Actores:	Cabeza de familia
Descripción:	La persona encargada del hogar ingresa la información personal
Precondiciones:	<ol style="list-style-type: none"> <li>1. debe haber acceso a Internet disponible para acceder al sistema.</li> <li>2. la persona cabeza de familia puede ingresar / modificar la información con ayuda de un asistente.</li> </ol>
Poscondiciones:	Se almacenan los datos temporalmente en sesión para pasar a recoger información personal adicional
Flujo Normal:	La persona que es cabeza de familia (encargado del hogar) ingresa a la página Web y selecciona primer ingreso, aquí es donde ingresa por primera vez sus datos personales.
Flujos Alternativos:	<ol style="list-style-type: none"> <li>1. en vez de seleccionar primer ingreso la persona puede seleccionar usuario existente, ingresar unos datos personales básicos para comprobar que existe en base de datos e inmediatamente hacer modificaciones.</li> </ol>
Excepciones:	El usuario no puede ingresar al sistema ya sea porque esta tratando de ingresar como nuevo cuando sus datos ya existen en base de datos o que haya ingresado mal los mismos al tratar de hacer modificaciones.
Incluidos:	
Prioridad:	ALTA
Frecuencia de Uso:	Una vez por familia (primer ingreso), N veces por familia (actualización de datos)
Requerimientos Especiales:	En caso de que sea primer ingreso se valida que no exista ya en base de datos

## Tarjeta Descriptiva Caso de Uso 1.2

ID Caso de Uso:	1.2
Nombre del Caso de Uso:	Ingresar datos personales adicionales (avanzados) vía Web.
Creado Por:	Daniel Andres Valencia V. & Danko Jiménez L.
Fecha de Creación:	15/10/2007
Actores:	Cabeza de familia
Descripción:	La persona encargada del hogar ingresa la información personal adicional relacionada con salud y educación.
Precondiciones:	1. la persona cabeza de familia ya debió haber ingresado la información personal básica.
Poscondiciones:	1. Se almacenan los datos temporalmente en sesión para pasar a recoger información de vivienda.
Flujo Normal:	Inmediatamente terminado el ingreso de datos personales básicos, se procede a ingresar los datos personales más detallados relacionados con salud, educación y cultura de la persona.
Flujos Alternativos:	
Excepciones:	No se pueden adicionar/modificar datos personales avanzados sin antes haber pasado por los datos básicos.
Incluidos:	
Prioridad:	ALTA
Frecuencia de Uso:	Una vez por familia (primer ingreso), N veces por familia (actualización de datos)
Requerimientos Especiales:	En caso de que sea primer ingreso se valida que no exista ya en base de datos

### Tarjeta Descriptiva Caso de Uso 1.3

ID Caso de Uso:	1.3
Nombre del Caso de Uso:	Ingresar datos de vivienda vía Web.
Creado Por:	Daniel Andres Valencia V. & Danko Jiménez L.
Fecha de Creación:	15/10/2007
Actores:	Cabeza de familia
Descripción:	La persona encargada del hogar ingresa la información relacionada con la vivienda en la que habita.
Precondiciones:	<ol style="list-style-type: none"> <li>1. la persona cabeza de familia ya debió haber ingresado la información personal avanzada.</li> <li>2. en caso de ser modificación de datos la persona ya debió haber comprobado que existe en el sistema.</li> </ol>
Poscondiciones:	<ol style="list-style-type: none"> <li>1. Se almacenan los datos en la base de datos.</li> </ol>
Flujo Normal:	Inmediatamente terminado el ingreso de datos personales básicos y avanzados, se procede a ingresar los datos relacionados con vivienda.
Flujos Alternativos:	
Excepciones:	
Incluidos:	
Prioridad:	ALTA
Frecuencia de Uso:	Una vez por familia (primer ingreso), N veces por familia (actualización de datos)
Requerimientos Especiales:	

## Tarjeta Descriptiva Caso de Uso 2

ID Caso de Uso:	2
Nombre del Caso de Uso:	Almacenar la información recogida por la página Web en la base de datos.
Creado Por:	Daniel Andres Valencia V. & Danko Jiménez L.
Fecha de Creación:	15/10/2007
Actores:	
Descripción:	La información recogida se almacena de forma organizada en la base de datos
Precondiciones:	<ol style="list-style-type: none"> <li>1. la persona ya debió haber ingresado / modificado todos los datos.</li> </ol>
Poscondiciones:	<ol style="list-style-type: none"> <li>1. es posible que después de almacenados pueden ser llamados de nuevo para su modificación.</li> <li>2. los datos almacenados se pueden utilizar para el sistema de administración de consultas.</li> </ol>
Flujo Normal:	Inmediatamente terminado el ingreso de datos personales básicos, avanzados, y de Vivienda, se almacena todo de forma organizada en la base de datos.
Flujos Alternativos:	
Excepciones:	
Incluidos:	
Prioridad:	ALTA
Frecuencia de Uso:	N veces por ingreso al sistema
Requerimientos Especiales:	



### Tarjeta Descriptiva Caso de Uso 3

ID Caso de Uso:	3
Nombre del Caso de Uso:	Solicitar información de consulta al sistema
Creado Por:	Daniel Andres Valencia V. & Danko Jiménez L.
Fecha de Creación:	15/10/2007
Actores:	Organización solicitante, operador
Descripción:	Un miembro de una organización que provee servicios a la comunidad necesita información de la base de datos y la solicita mediante un operador (que puede ser de la misma organización)
Precondiciones:	1. se debe saber suficiente información de lo que la organización necesita para crear la solicitud
Poscondiciones:	1. el operador debe conectarse al sistema para hacer la solicitud de consulta.
Flujo Normal:	Un miembro de la organización solicita una consulta al sistema sobre unos datos específicos, este le entrega la solicitud al operador el modulo de administración de consultas para que haga la consulta y extraiga los resultados
Flujos Alternativos:	El operador mismo puede ser de la organización y solicitar la información directamente al sistema.
Excepciones:	No se pueden hacer solicitudes sobre tipos de información que no se encuentren en la base de datos (ej. Información de sueldos)
Incluidos:	
Prioridad:	ALTA
Frecuencia de Uso:	Una vez por cada solicitud
Requerimientos Especiales:	

### Tarjeta Descriptiva Caso de Uso 4

ID Caso de Uso:	4
Nombre del Caso de Uso:	Conectar el sistema al servicio Web y la base de datos.
Creado Por:	Daniel Andres Valencia V. & Danko Jiménez L.
Fecha de Creación:	15/10/2007
Actores:	Operador
Descripción:	El operador del sistema de administración de consultas se conecta con el servicio Web que provee acceso a la base de datos.
Precondiciones:	<ol style="list-style-type: none"> <li>1. el sistema de administración de consultas debe tener la maquina virtual instalada y conexión a Internet.</li> <li>2. el operador debió haber recibido previamente la solicitud de consulta de la organización solicitante.</li> <li>3. la organización solicitante ya debió ser registrada previamente en la base de datos</li> </ol>
Poscondiciones:	<ol style="list-style-type: none"> <li>1. el operador debe utilizar la solicitud de consulta para generar una consulta para ejecutar en la base de datos.</li> </ol>
Flujo Normal:	El operador (se asume que pertenece a la misma organización solicitante) ingresa su nombre de usuario y contraseña en el sistema y este se conecta al servicio Web para verificar dichos datos, si estos están correctos, el sistema habilita el área de trabajo para hacer la solicitud.
Flujos Alternativos:	
Excepciones:	Si el nombre de usuario y contraseña no se escribieron correctamente o nunca fueron registrados en las base de datos en el momento de comprar el acceso al servicio, el sistema no permitirá el acceso a la base de datos.
Incluidos:	
Prioridad:	ALTA
Frecuencia de Uso:	Una vez por cada solicitud
Requerimientos Especiales:	Maquina virtual instalada en la maquina donde el sistema de administración este instalado, conexión a Internet.

### Tarjeta Descriptiva Caso de Uso 5

ID Caso de Uso:	5
Nombre del Caso de Uso:	Generar consulta especifica para la base de datos de acuerdo a la solicitud de consulta
Creado Por:	Daniel Andres Valencia V. & Danko Jiménez L.
Fecha de Creación:	15/10/2007
Actores:	Operador
Descripción:	El operador, una vez conectado a la base de datos vía servicio Web, abre el generador de consultas que este mas acorde con la solicitud de consulta de la organización y construye la consulta que se necesita.
Precondiciones:	<ol style="list-style-type: none"> <li>1. debe haber se verificado que el nombre de usuario y contraseña de la organización fueran los correctos.</li> <li>2. el sistema de administración debe mostrar las opciones de generación de consultas.</li> </ol>
Poscondiciones:	<ol style="list-style-type: none"> <li>1. el operador ejecuta la consulta generada.</li> </ol>
Flujo Normal:	El operador, una vez conectado a la base de datos vía servicio Web, ejecuta en el sistema el generador de consultas que este mas acorde con la solicitud de consulta de la organización y construye la consulta que se necesita, este generador de consultas tiene las opciones necesarias para construir una consulta correspondiente a la solicitud de información necesitada.
Flujos Alternativos:	Si la consulta necesita opciones que no provee los generadores de consultas, entonces el operador puede escoger utilizar el modo avanzado, pero se requieren conocimientos de lenguaje SQL y de la estructura interna de la base de datos para poder utilizarlo.
Excepciones:	No se pueden generar consultas sobre tipos de información que no se encuentren en la base de datos (ej. Información de sueldos), ni mucho menos consultas que modifiquen el contenido y/o la estructura de la base de datos
Incluidos:	
Prioridad:	ALTA
Frecuencia de Uso:	Una vez por cada solicitud
Requerimientos Especiales:	Conexión a Internet.

### Tarjeta Descriptiva Caso de Uso 6

ID Caso de Uso:	6
Nombre del Caso de Uso:	Enviar la consulta generada al servicio Web para que este la ejecute en la base de datos
Creado Por:	Daniel Andres Valencia V. & Danko Jiménez L.
Fecha de Creación:	15/10/2007
Actores:	
Descripción:	Una vez generada la consulta se envía al servicio Web para que este la ejecute en la base de datos
Precondiciones:	1. el operador debe haber construido la consulta a enviar
Poscondiciones:	1. la consulta es analizada por el gestor de la base de datos. En ese momento el sistema entra en modo de espera para recibir la respuesta.
Flujo Normal:	El sistema toma la consulta generada por el operador y la envía como un mensaje XML al servicio Web, este una vez recibe el mensaje lo entrega a la base de datos para su ejecución.
Flujos Alternativos:	
Excepciones:	
Incluidos:	
Prioridad:	ALTA
Frecuencia de Uso:	Una vez por cada solicitud
Requerimientos Especiales:	Conexión a Internet.

### Tarjeta Descriptiva Caso de Uso 7

ID Caso de Uso:	7
Nombre del Caso de Uso:	Analizar y ejecutar la consulta en la base de datos
Creado Por:	Daniel Andres Valencia V. & Danko Jiménez L.
Fecha de Creación:	15/10/2007
Actores:	
Descripción:	Una vez el servicio Web recibe la consulta, la manda al gestor de la base de datos y este analiza y ejecuta la consulta
Precondiciones:	1. el servicio Web debe convertir el mensaje XML de nuevo en una consulta SQL valida para su uso con la base de datos.
Poscondiciones:	1. el servicio Web debe convertir los resultados de la consulta en un mensaje xml para enviar de regreso al sistema de administración de consultas
Flujo Normal:	La base de datos toma la consulta SQL y la analiza, si es una consulta valida la ejecuta y devuelve los resultados al servicio Web para su envío de respuesta al sistema de administración de consultas ubicado en el computador de la organización
Flujos Alternativos:	La consulta valida puede no producir resultados por lo que se regresa una respuesta vacía al servicio Web.
Excepciones:	Si la consulta no es valida y produce un error, el gestor de bases de datos devuelve resultado vacío al servicio Web.
Incluidos:	
Prioridad:	ALTA
Frecuencia de Uso:	Una vez por cada solicitud
Requerimientos Especiales:	

### Tarjeta Descriptiva Caso de Uso 8

ID Caso de Uso:	8
Nombre del Caso de Uso:	Extraer los resultados devueltos por la consulta generada por el generador de consultas y enviada al servicio Web.
Creado Por:	Daniel Andres Valencia V. & Danko Jiménez L.
Fecha de Creación:	15/10/2007
Actores:	Operador
Descripción:	El sistema de administración de consultas recibe los resultados del servicio Web y los convierte en una tabla lista para su uso
Precondiciones:	1. la base de datos debe procesar la consulta generada
Poscondiciones:	1. los resultados deben estar en un formato listo para sus uso por la organización solicitante
Flujo Normal:	El sistema de administración recibe los resultados del servicio Web como un mensaje XML, este se transforma en un formato de tabla de datos listo para su uso por el operador, este extrae la tabla de datos en un formato listo para usar por la organización solicitante.
Flujos Alternativos:	
Excepciones:	Cuando no existen resultados a la consulta solicitada o la consulta genero error en la base de datos el servicio Web envía un mensaje vacío al sistema de administración, por lo este no muestra resultados de ningún tipo.
Incluidos:	
Prioridad:	ALTA
Frecuencia de Uso:	Una vez por cada solicitud
Requerimientos Especiales:	Conexión a Internet.

### Tarjeta Descriptiva Caso de Uso 9

ID Caso de Uso:	9
Nombre del Caso de Uso:	Entregar resultados a la organización solicitante
Creado Por:	Daniel Andres Valencia V. & Danko Jiménez L.
Fecha de Creación:	15/10/2007
Actores:	Operador, organización solicitante
Descripción:	El operador del sistema de administración entrega los resultados de la consulta a la organización para su apropiado uso
Precondiciones:	1. el sistema debe haber procesado los resultados devueltos por el servicio Web.
Poscondiciones:	
Flujo Normal:	Una vez extraídos los resultados en un formato compatible para su uso (CSV), el operador los entrega a la organización solicitante.
Flujos Alternativos:	
Excepciones:	Cuando no hay resultados a la consulta solicitada, el operador entrega el respectivo informe a la organización.
Incluidos:	
Prioridad:	ALTA
Frecuencia de Uso:	Una vez por cada solicitud
Requerimientos Especiales:	

2.7 MODELO FUNCIONAL  
DIAGRAMA DE DESPLIEGUE  
MODELO ENTIDAD RELACIÓN

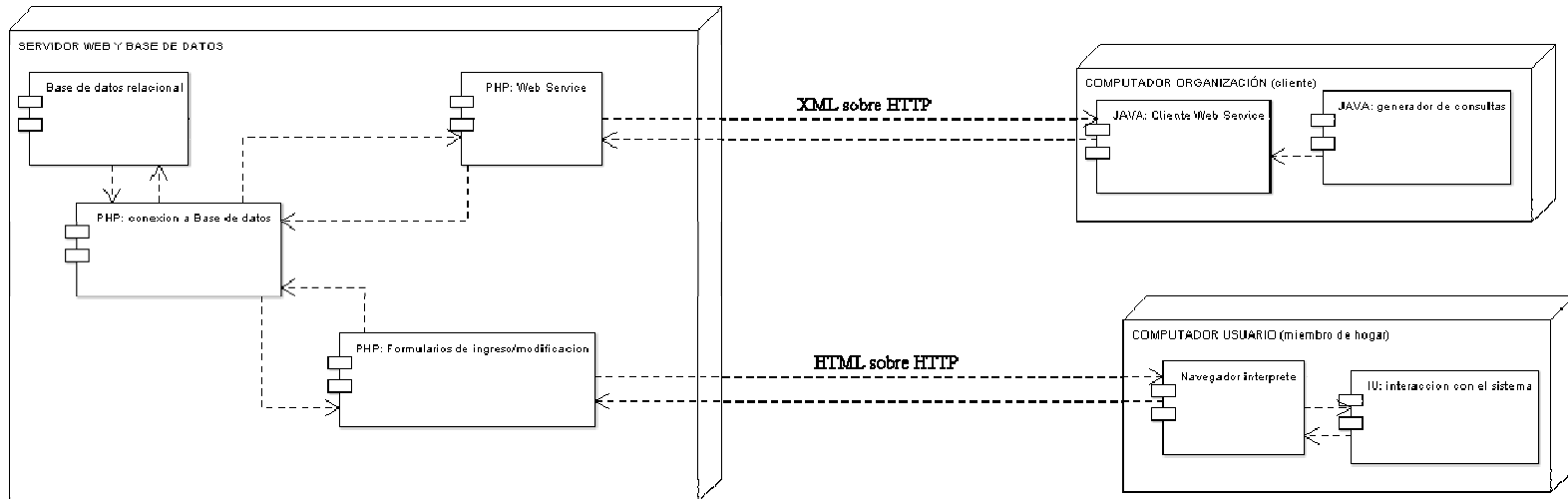
SOCRATES-EC: SISTEMA DE INFORMACIÓN POBLACIONAL BASADO EN  
PROCEDIMIENTOS DE CENSADO

Por:

Danko Jiménez Londoño  
Daniel Andres Valencia V.



## 2.7.1 Diagrama de despliegue





### 2.7.3 Diccionario de datos del modelo entidad-relación

<b>Vivienda:</b> tabla que almacena todos los datos relacionados con la vivienda, desde su estructura básica hasta los servicios con los que cuenta		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo de dato</b>
Cod_vivienda	Código único que representa cada vivienda dentro de la base de datos	Varchar (no nulo)
Cod_disposicion	Código que identifica el tipo de disposición de desechos dentro de la vivienda	varchar
Cod_sanitario	Código que representa el tipo de servicio sanitario disponible en la vivienda	varchar
cocina	Indicador de existencia de una cocina dentro de la vivienda	Boolean
Numero_cuartos_bano	Almacena el numero total de cuartos de baño (incluyendo ducha) que existen en la vivienda	varchar
Numero_cuartos	Numero total de habitaciones dentro de la vivienda (incluyendo baños cocina, sala, etc)	varchar
Numero_cuartos_dormir	Numero total de dormitorios dentro de la vivienda	varchar
Uso_servicio_sanitario	Indicador de exclusividad para los miembros de la vivienda del servicio sanitario	boolean
Cod_material_vivienda	Código que representa el tipo de material principal del cual esta construida la vivienda	varchar
Fecha_actualizacion	Fecha en la que se adicionaron/modificaron por ultima vez los datos anteriormente mencionados	Date

<b>Persona:</b> tabla que almacena los datos básicos relacionados con la persona, es decir, los datos esenciales como nombres y apellidos, documento de identidad, fecha de nacimiento, etc.		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo de dato</b>
Doc_persona	Documento de identificación de la persona	Varchar (no nulo)
Cod_doc	Código que representa el tipo de documento de identidad de la persona	Varchar
Nombre1	Primer nombre de la persona	varchar
Nombre2	Segundo nombre de la persona	Varchar
Apellido1	Primero apellido de la persona	Varchar
Apellido2	Segundo apellido de la persona	Varchar
Cod_sexo	código del género de la persona (masculino o femenino)	Varchar
Cod_vivienda	Código que relaciona la persona con una vivienda específica	Varchar
Fecha_nac	Fecha de nacimiento de la persona	Date

<b>Info_persona:</b> tabla que almacena todos los datos adicionales relacionados con la persona, estos incluyen información de salud, educación y relaciones familiares		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo de dato</b>
Doc_persona	Documento de identidad de la persona	Varchar (no nulo)
Cod_parentesco	Código que represente el tipo de parentesco que tiene la persona respecto a la cabeza de hogar (o si la persona es la cabeza de hogar)	Varchar
Cod_indigena	Código que representa la pertenencia de una persona a una etnia indígena o raza en particular	Varchar
Cod_estado_civil	Código que representa el estado civil de la persona	Varchar
Sabe_leer	Indicador que muestra si la persona sabe leer	boolean
Sabe_escribir	Indicador que muestra si la persona sabe escribir	Boolean
Asiste_educacion	Indicador que muestra si la persona actualmente asiste a un centro educativo	Boolean
Cod_nivel_edu	Código que representa el nivel educativo en el que se encuentra actualmente la persona	Varchar
Cod_grado	Código que representa el grado	Varchar
Cod_org_salud	Código que representa la afiliación de la persona a una organización promotora de salud	Varchar
Fecha_actualizacion	Fecha en la que se añadió/modificó por última vez la información anterior	Date

<b>Parentesco:</b> tabla que maneja las relaciones de parentesco de la persona con la cabeza del hogar (o si la persona es la cabeza del hogar)		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo de dato</b>
Cod_parentesco	Código que representa el tipo de parentesco de la persona con la cabeza de hogar	Varchar (no nulo)
Tipo_parentesco	Nombre del parentesco de la persona con la cabeza de hogar	Varchar

<b>Indigenas:</b> tabla que maneja la posible raza/etnia indígena a la cual pertenece la persona		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo de dato</b>
Cod_indigena	Código que representa la pertenencia de la persona a una etnia indígena en particular	Varchar (no nulo)
Tipo_indigena	Nombre de la etnia indígena	Varchar

<b>Estados_civiles:</b> tabla que maneja los posibles estados civiles de la persona		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo de dato</b>
Cod_estado_civil	Código que represente el estado civil que posee la persona actualmente	Varchar (no nulo)
Tipo_estado_civil	Tipo de estado civil de la persona	Varchar

<b>Nivel_educacion:</b> tabla que maneja el nivel educativo actual de la persona		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo de dato</b>
Cod_nivel_edu	Código que representa el nivel educativo de la persona	Varchar (no nulo)
Nivel_edu	Nombre del nivel educativo	Varchar

<b>Grado_educacion:</b> tabla que maneja la lista de últimos grados cursados de la persona		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo de dato</b>
Cod_grado	Código correspondiente al último grado cursado por la persona	Varchar (no nulo)
Nombre_grado	Nombre del último grado cursado por la persona	Varchar

<b>Pertenencia_salud:</b> tabla que maneja la lista de posibles entidades promotoras de salud a las cuales puede pertenecer la persona		
Atributo	Descripción	Tipo de dato
Cod_org_salud	Código correspondiente a la entidad promotora de salud	Varchar (no nulo)
Tipo_org_salu	Tipo de entidad promotora de salud	Varchar

<b>Lugar_nacimiento:</b> tabla que almacena el lugar de nacimiento de la persona		
Atributo	Descripción	Tipo de dato
Doc_persona	Numero de identificación de la persona	Varchar
Cod_ciudad	Código correspondiente a la ciudad o municipio donde nacio la persona	Varchar
Cod_depto	Código correspondiente al departamento donde nacio la persona	Varchar

<b>ciudades:</b> tabla que maneja la lista de ciudades / municipios que tiene el país, se usa para ubicar el lugar de nacimiento de la persona		
Atributo	Descripción	Tipo de dato
Cod_ciudad	Código correspondiente a la ciudad o municipio	Varchar (no nulo)
Nombre_ciudad	Nombre de la ciudad o municipio	Varchar
Cod_depto	Código correspondiente al departamento al que pertenece la ciudad	Varchar

<b>departamentos:</b> tabla que maneja la lista de departamentos que tiene el país, se usa para ubicar el lugar de nacimiento de la persona		
Atributo	Descripción	Tipo de dato
Cod_depto	Código correspondiente al departamento	Varchar (no nulo)
Nombre_depto	Nombre del departamento	Varchar

<b>Limitaciones_fisicas_persona:</b> tabla que almacena las posibles limitaciones físicas que posee la persona		
Atributo	Descripción	Tipo de dato
Doc_persona	Numero de documento de identidad de la persona	Varchar
Cod_limitacion	Código correspondiente al tipo de limitación física que posee la persona	Varchar

<b>Limitaciones_fisicas:</b> tabla que maneja todas las posibles limitaciones físicas que puede poseer una persona		
Atributo	Descripción	Tipo de dato
Cod_limitacion	Código correspondiente al tipo de limitación física	Varchar (no nulo)
Tipo_limitacion	Tipo de limitación física	Varchar

<b>Tratamientos_persona:</b> tabla que almacena los tratamientos médicos especiales que recibió la persona por alguna causa en particular		
Atributo	Descripción	Tipo de dato
Doc_persona	Numero de documento de identidad de la persona	Varchar
Cod_tratamiento	Código correspondiente al tipo de tratamiento medico especial que recibió la persona	Varchar

<b>tratamientos:</b> tabla que maneja todos los posibles tratamientos médicos especiales que puede recibir una persona		
Atributo	Descripción	Tipo de dato
Cod_tratamiento	Código correspondiente al tipo de tratamiento medico especial	Varchar (no nulo)
Tipo_tratamiento	Tipo de tratamiento medico especial	Varchar

<b>Idiomas_persona:</b> tabla que almacena todos los idiomas que conoce la persona		
Atributo	Descripción	Tipo de dato
Doc_persona	Numero de documento de identidad de la persona	Varchar
Cod_idioma	Código correspondiente al idioma que conoce (lee/escribe/habla) la persona	varchar



<b>idiomas:</b> tabla que maneja todos los posibles idiomas que puede conocer a persona		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo de dato</b>
Cod_idioma	Código correspondiente al idioma	Varchar (no nulo)
Nombre_idioma	Nombre del idioma	Varchar

<b>documentos:</b> tabla que maneja todos los posibles tipo de documentos de identidad que tiene la persona		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo de dato</b>
Cod_doc	Código del tipo de documento de identidad	Varchar (no nulo)
Tipo_doc	Tipo de documento de identidad	Varchar

<b>sexo:</b> tabla que maneja todos los posibles géneros de la persona		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo de dato</b>
Codsexo	Código del sexo (genero) al que pertenece la persona	Varchar (no nulo)
Tiposexo	Tipo de sexo (genero)	varchar

<b>Servicios_vivienda:</b> tabla que almacena los servicios públicos con los que cuenta la vivienda		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo de dato</b>
Cod_vivienda	Código único que representa la vivienda dentro de la base de datos	Varchar
Cod_servicio	Código que representa el servicio que posee la vivienda	Varchar

<b>Servicios_publicos:</b> tabla que maneja todos los posibles servicios públicos que puede tener la vivienda		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo de dato</b>
Cod_servicio	Código que representa el servicio publico	Varchar (no nulo)
Nombre_servicio	Nombre del servicio publico	Varchar

<b>Vehículos_vivienda:</b> tabla que almacena todos los vehículos que están en la vivienda		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo de dato</b>
Cod_vivienda	Código único que representa la vivienda dentro de la base de datos	Varchar
Cod_vehiculo	Código del tipo de vehiculo	Varchar

<b>Vehiculos:</b> tabla que maneja todos los posibles vehículos que pueden poseer los habitantes de la vivienda		
Atributo	Descripción	Tipo de dato
Cod_vehiculo	Código del tipo de vehículo	Varchar (no nulo)
Tipo_vehiculo	Tipo de vehiculo	Varchar

<b>Admon_desechos:</b> tabla que maneja todas las posibles manera de disposición de desechos dentro de la vivienda		
Atributo	Descripción	Tipo de dato
Cod_disposicion	Código que representa el tipo de disposición de desechos	Varchar (no nulo)
Nombre_disposicion	Nombre del tipo de disposición de desechos	Varchar

<b>Servicios_sanitarios:</b> tabla que maneja todos los tipos posibles de servicio sanitario dentro de la vivienda		
Atributo	Descripción	Tipo de dato
Cod_sanitario	Código que representa el tipo de servicio sanitario	Varchar (no nulo)
Nombre_sanitario	Nombre del tipo de servicio sanitario	Varchar

<b>Material_vivienda:</b> tabla que maneja todos los posibles materiales de que esta principalmente hecha la vivienda		
Atributo	Descripción	Tipo de dato
Cod_material_vivienda	Código que representa el material principal del que esta hecha la vivienda	Varchar (no nulo)
Nombre_material_vivienda	Nombre del material del que esta hecha la vivienda	Varchar

<b>Direcciones_Vivienda:</b> tabla que almacena la dirección de cada vivienda (ubicación geográfica)		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo de dato</b>
Cod_vivienda	Código único que representa cada vivienda dentro de la base de datos	Varchar
Vía_ppal	Vía principal sobre la cual esta ubicada la vivienda	Varchar
Numero_via	Numero de vía principal	Varchar
Sufijo_via	Sufijo de vía principal	Varchar
Bis1	Indicador de bis (bifurcación de via)	Varchar
cruce	Vía secundaria, con la cual se cruza la vía donde esta ubicada la vivienda	Varchar
Numero_cruce	Numero de vía secundaria	Varchar
Sufijo_cruce	Sufijo de vía secundaria	Varchar
Bis2	Indicador de bis secundario	Varchar
Numero_porton	Numero que indica la ubicación de la vivienda en el sector	varchar
Tipo_complemento1	Tipo de complemento adicional para precisar mejor la ubicación de la vivienda (por ejemplo "bloque")	varchar
Numero_complemento1	Numero de complemento (ejemplo bloque "3ª")	varchar
Tipo_complemento2	Tipo de complemento de segundo nivel (ejemplo "apto")	varchar
Numero_complemento2	Numero de complemento de segundo nivel (ejemplo apto "325")	Varchar
Tipo_complemento3	Tipo de complemento adicional de tercer nivel	varchar
Numero_complemento3	Numero de complemento de tercer nivel	varchar

<b>Equivalencias_direcciones:</b> tabla que almacena todas las posibles formas de escribir direcciones y sus equivalentes estandarizados para traducción de dirección		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo de dato</b>
Valor_hallado	Valor reconocido por el analizador de direcciones de la escritura de la dirección en el formulario de vivienda	Varchar
Valor_asignado	Valor de traducción estandarizado de direcciones	Varchar

<b>clientes:</b> tabla que almacena la lista de clientes que tienen acceso a la información de la base de datos vía generador de consultas		
<b>Atributo</b>	<b>Descripción</b>	<b>Tipo de dato</b>
Nombre_cliente	Nombre del cliente (organización que provee servicios a la comunidad)	Varchar
Password	Contraseña de acceso a la base de datos para uso del generador de consultas	varchar

**ANEXO C  
MANUALES TÉCNICOS Y DE USUARIO**

<b>CONTENIDO</b>	<b>PAG</b>
<b>1. MANUAL DE INSTALACIÓN Y ADMINISTRACIÓN DE BASE DE DATOS POSTGRESQL BAJO WINDOWS</b>	<b>148</b>
<b>2. MANUALES DE USUARIO, CATALOGO DE FUENTES</b>	<b>163</b>
<b>2.1 MANUAL DE USUARIO PARA EL LLENADO DE LA INFORMACIÓN EN LOS FORMULARIOS WEB</b>	<b>164</b>
<b>2.2 MANUAL DE USUARIO: SÓCRATES-EC ADMINISTRACIÓN Y CONSULTAS</b>	<b>175</b>
<b>3. FORMATO DE FORMULARIOS Y PREGUNTAS DE TIPO CENSAL</b>	<b>202</b>
<b>4. CATALOGO DE ARCHIVOS FUENTE</b>	<b>205</b>
<b>4.1 ESQUEMA BÁSICO DE CONSTRUCCION DEL SISTEMA</b>	<b>206</b>
<b>4.2 ARCHIVOS QUE COMPONEN EL SISTEMA DE RECOLECCIÓN/ACTUALIZACIÓN DE DATOS VÍA WEB</b>	<b>209</b>
<b>4.3 ARCHIVOS QUE COMPONEN EL SISTEMA DE ADMINISTRACIÓN DE CONSULTAS (CLIENTE - SERVIDOR)</b>	<b>211</b>

1. MANUAL DE INSTALACIÓN Y ADMINISTRACIÓN DE BASE DE DATOS  
POSTGRESQL BAJO WINDOWS

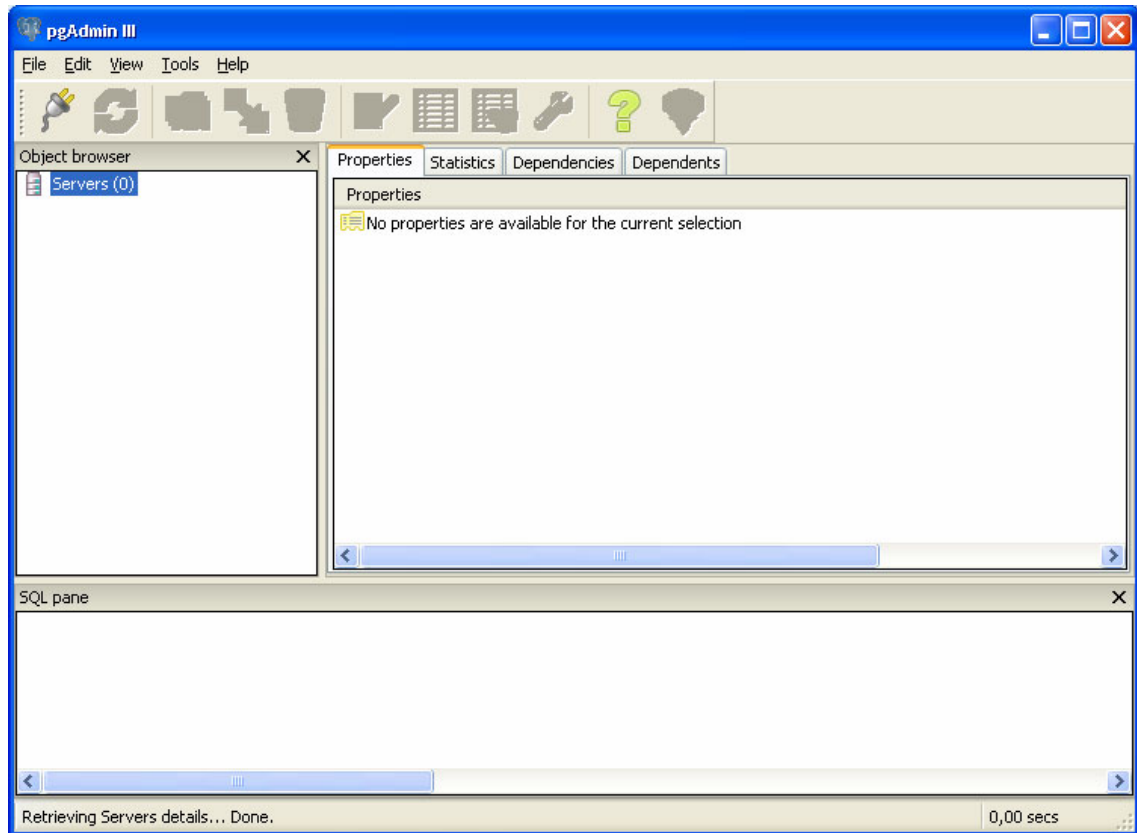
SOCRATES-EC: SISTEMA DE INFORMACIÓN POBLACIONAL BASADO EN  
PROCEDIMIENTOS DE CENSADO

Por:

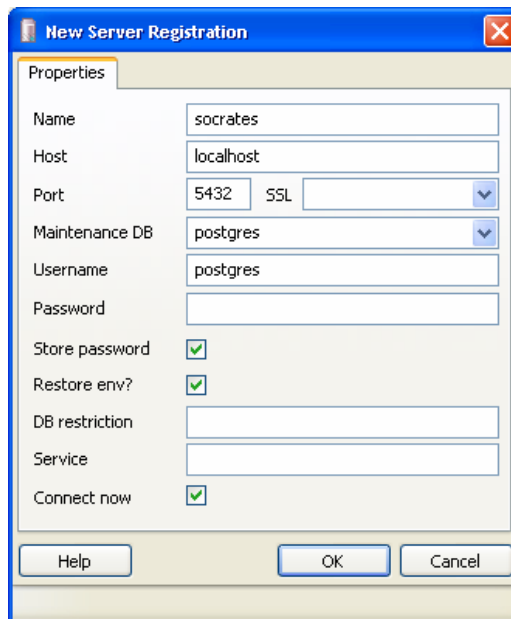
Danko Jiménez Londoño  
Daniel Andrés Valencia V.

Para empezar deberemos configurar el PostgreSQL después de su instalación de la siguiente manera:

Conectamos el servidor para poder crear la base de datos



Registramos el servidor colocando primero el nombre que para este caso será socrates y el host será localhost



**1:** Asumiendo que el PostgreSQL ya se encuentre instalado procederemos a generar la base de datos de la siguiente manera:

En la línea de comandos de PostgreSQL se digitará lo siguiente:

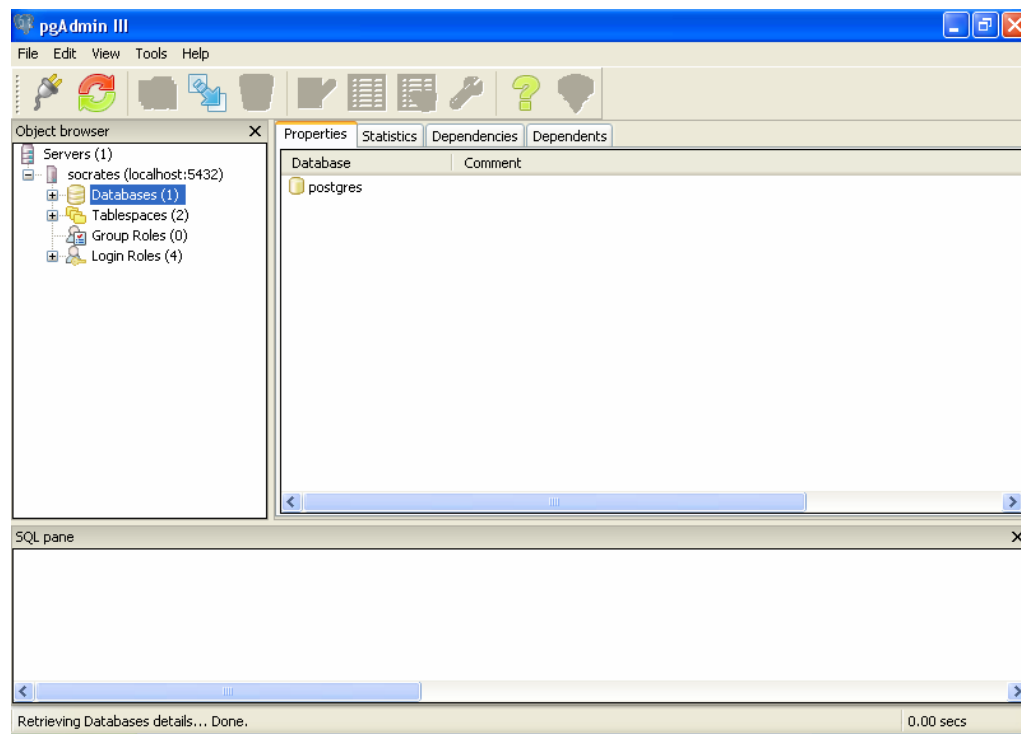
```
create user socrates createdb;  
create database socrates;  
alter user socrates nocreatedb;  
alter user socrates with encrypted password 'socrates';
```

Observar el gestor de la base de datos para saber si es necesario reiniciar.

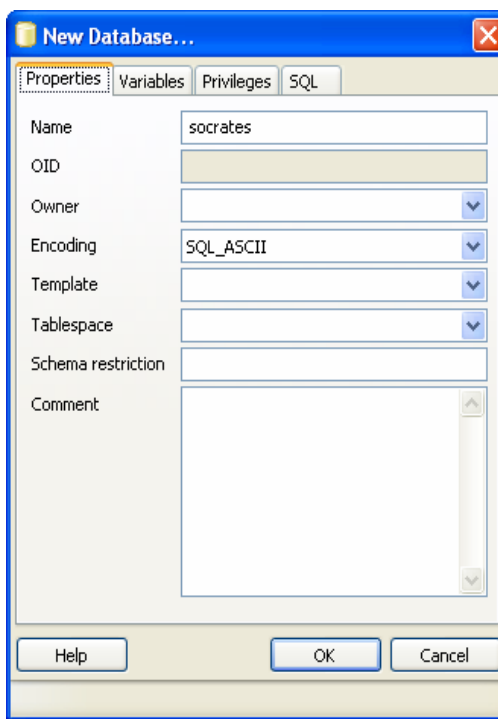
**2:** Si para el proceso de instalación usted tiene pgadmin III se procederá de la siguiente manera :

**a:** Parados en el Object Browser del pgadmin III se procederá a crear la base de datos poniendo el puntero sobre Databases y con clic secundario se seleccionará la opción "New Database".

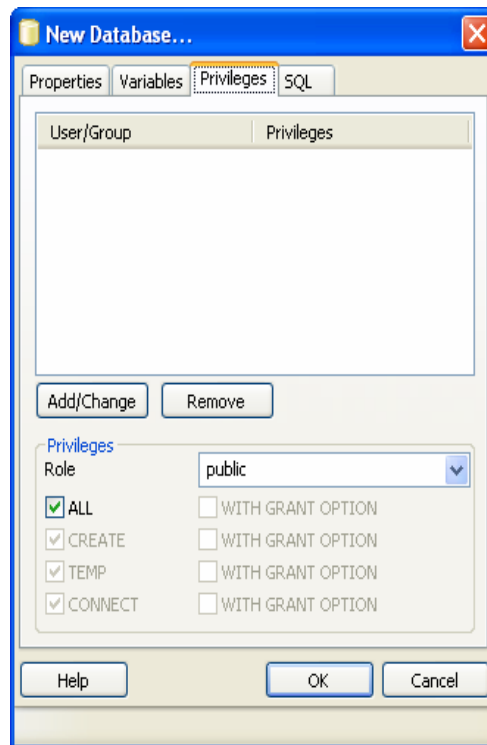




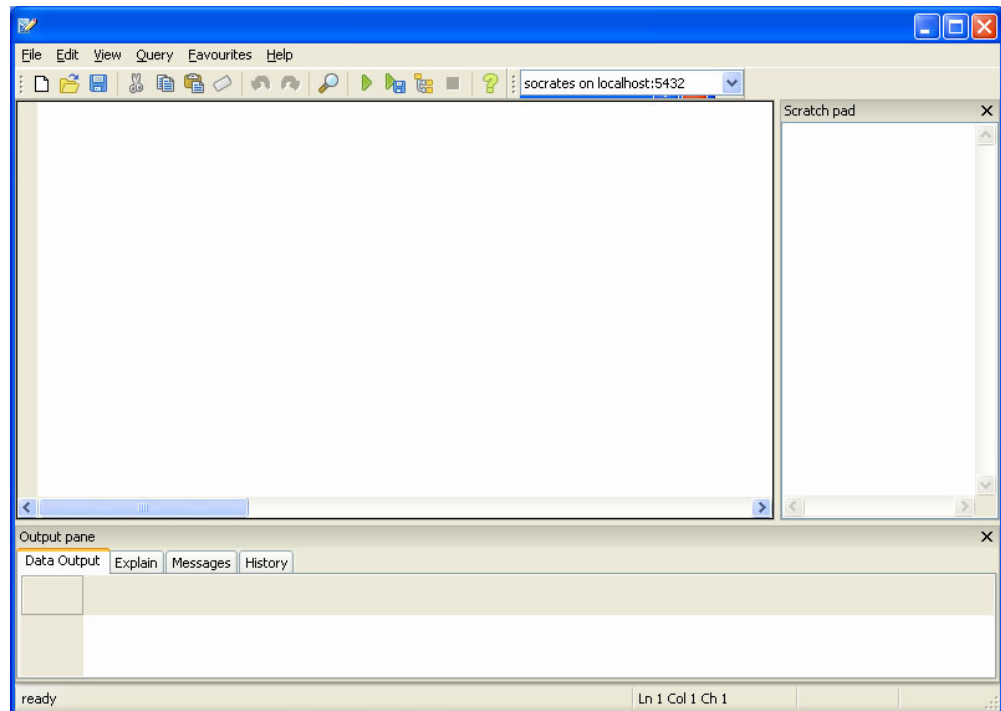
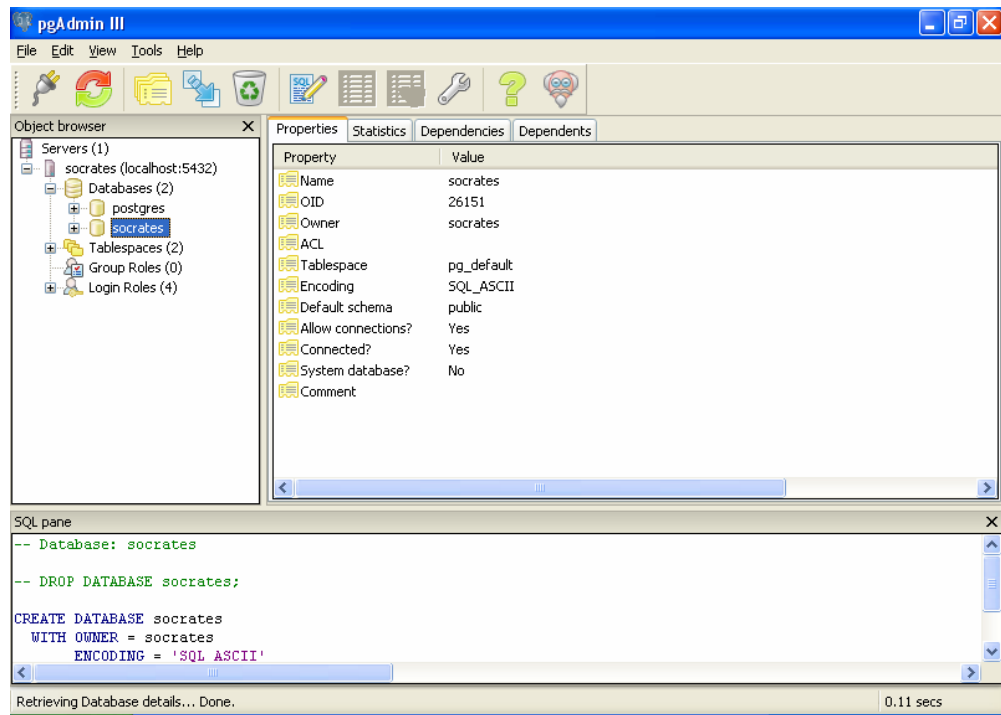
**b:** En el campo Name procederemos a colocar el nombre de la base de datos "Socrates".



**c:** En el campo **Privileges** se procederá a activar el campo All. Y se dará clic en el botón OK.



**d:** Luego de crear la base de datos se procederá a correr los scripts donde se encuentra la información de las tablas y parametrización. Para ello se procederá a seleccionar con el puntero la base de datos y se dará clic en el botón SQL, esto abre la ventana de comandos de SQL.



e: Se procederá a abrir el script “socratescompleto.SQL” y se da click sobre el botón ejecutar.

```

[H:\danko\scriptsococratesbd\socratescompleto.sql]
File Edit View Query Favourites Help
socrates on localhost:5432

CREATE TABLE admon_desechos
(
  cod_disposicion character varying(6) NOT NULL,
  nombre_disposicion character varying(35) NOT NULL,
  CONSTRAINT admon_desechos_pkey PRIMARY KEY (cod_disposicion)
);

CREATE TABLE ciudades
(
  cod_ciudad numeric(6) NOT NULL,
  nombre_ciudad character varying(30) NOT NULL,
  cod_depto numeric(9),
  CONSTRAINT ciudades_pkey PRIMARY KEY (cod_ciudad)
);

CREATE TABLE departamentos
(
  cod_depto numeric(6) NOT NULL,
  nombre_depto character varying(30),
  CONSTRAINT departamentos_pkey PRIMARY KEY (cod_depto)
);

Output pane
Data Output Explain Messages History
Query returned successfully with no result in 1402 ms.

OK. Ln 1 Col 1 Ch 1 1402 ms

```

f: La operación anterior se repite con el segundo script llamado “script\_parametrizacion\_tablas.SQL”.

```

[H:\danko\scriptsococratesbd\script_parametrizacion_tablas.sql] *
File Edit View Query Favourites Help
socrates on localhost:5432

insert into admon_desechos values('adm01','otra forma');
insert into admon_desechos values('adm02','la entierran');
insert into admon_desechos values('adm03','la queman');
insert into admon_desechos values('adm04','la tiran al patio');
insert into admon_desechos values('adm05','la tiran al rio');
insert into admon_desechos values('adm06','Utilizan los servicios de aseo');

insert into documentos values('cc','cedula');
insert into documentos values('psp','pasaporte');
insert into documentos values('ti','T. identidad');

insert into equivalencias_direcciones values('#','#');
insert into equivalencias_direcciones values('AERO','AEROPUERTO');
insert into equivalencias_direcciones values('AEROP','AEROPUERTO');
insert into equivalencias_direcciones values('AEROPUERT','AEROPUERTO');
insert into equivalencias_direcciones values('AEROPUERTO','AEROPUERTO');
insert into equivalencias_direcciones values('AEROPURTO','AEROPUERTO');
insert into equivalencias_direcciones values('AP','AP');
insert into equivalencias_direcciones values('APARTAM','AP');
insert into equivalencias_direcciones values('APARTAMENTO','AP');
insert into equivalencias_direcciones values('APT','AP');

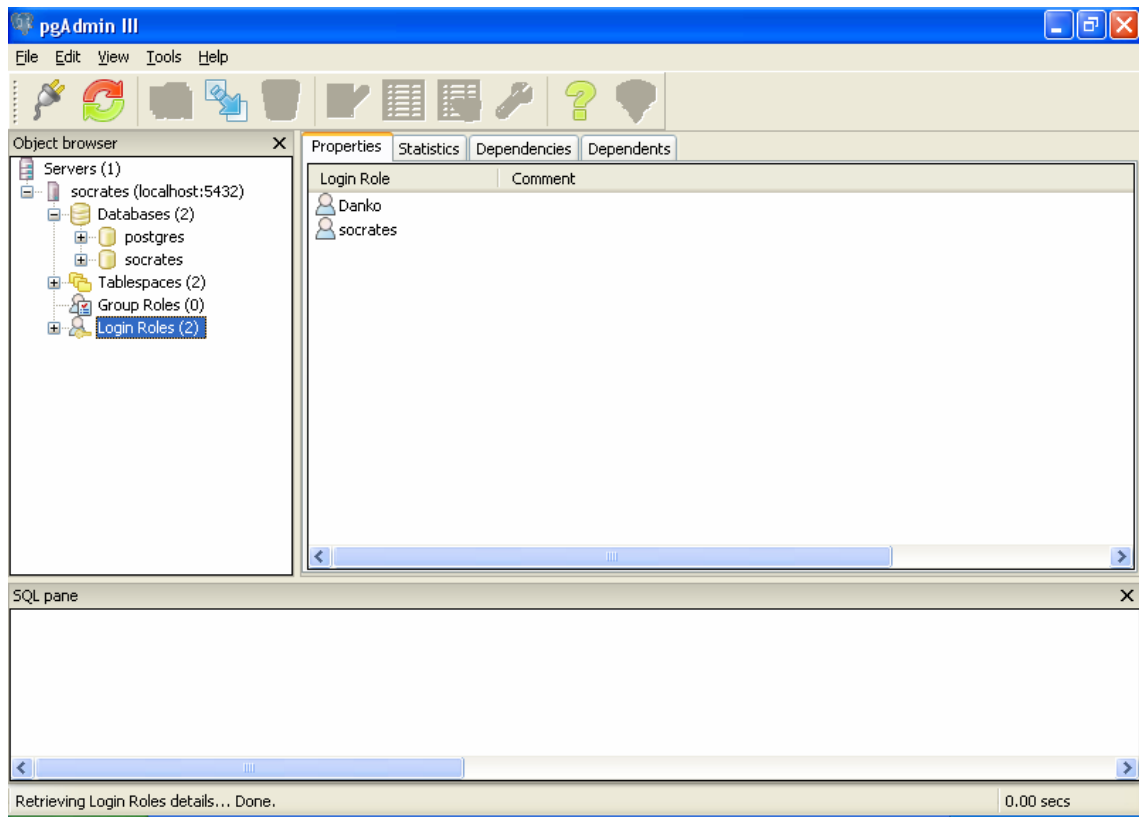
Output pane
Data Output Explain Messages History
Query returned successfully: 1 rows affected, 600 ms execution time.

1 rows affected. Ln 1 Col 1 Ch 1 600 ms

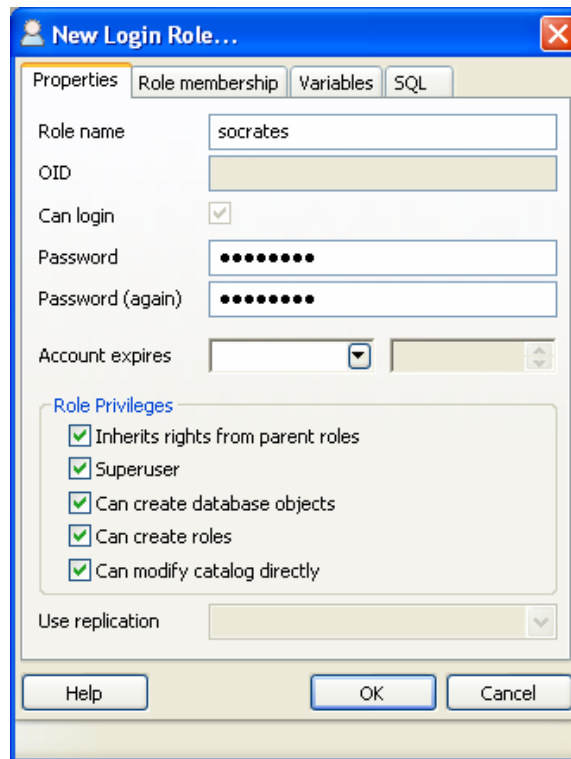
```

**g:** Inmediatamente creada la base de datos se procede a crear el usuario de acceso desde el cual la aplicación Web y el Webservice tendrán acceso.

**h:** En el Object browser del pgAdminIII se selecciona “login roles” y con clic secundario se selecciona “new login role...”.



**i:** En la ventana que aparece se escribe el nombre de usuario “socrates” y la contraseña “socrates” en sus respectivos campos, además de los privilegios apropiados (todos por defecto, más tarde se modificarán) y se da clic en el botón OK.



j: Una vez creado el usuario se procede a abrir la ventana de comandos SQL y se ingresa la siguiente línea y se da clic en ejecutar:

```
alter user socrates nocreatedb;
```

Para el correcto funcionamiento del sistema se debe hacer mantenimiento a su componente mas importante: la base de datos, sin esta no habría forma de que el sistema funcionara, por lo tanto un correcto mantenimiento es necesario para mantener todo el sistema en funcionamiento sin ningún tipo de inconvenientes.

Estas son las características básicas de la base de datos:

- Nombre de la base de datos: socrates
- Gestor de base de datos: PostgreSQL ver 8.2
- Usuario de acceso (módulos de recolección y admón. De consultas): socrates
- Contraseña de acceso: afrodita
- Numero total de tablas: 30
  - o Tablas de parametrización: 21
  - o Tablas de almacenado de datos: 10
  - o Tablas de acceso a clientes (admón. De consultas): 1
- Archivos de scripts necesarios para crear la base de datos: 2

- Plataforma operativa: multiplataforma (Windows server o Linux)

Para hacer expansiones al sistema como agregar nuevos parámetros para recolección de información se deben tener en cuenta las tablas de parametrización, estas permiten la expansión de las opciones de recolección de información del sistema Web.

Las tablas a tener en cuenta son:

- servicios\_publicos
- vehículos
- admon\_desechos
- servicios\_sanitarios
- material\_vivienda
- ciudades
- departamentos
- limitaciones\_fisicas
- tratamientos
- idiomas
- documentos
- sexo
- parentesco
- indígenas
- estados\_civiles
- nivel\_educacion
- grado\_educacion
- pertenencia\_salud

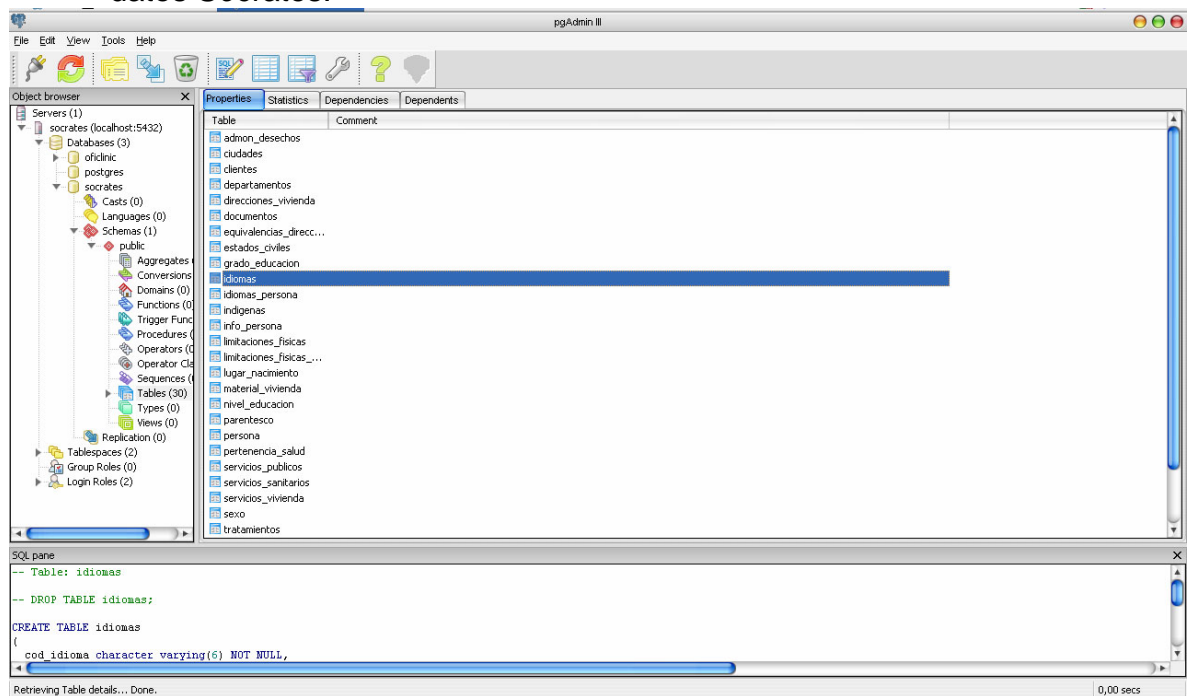
## CONSIDERACIÓN DE MANTENIMIENTO DEL NUCLEO DE LA SOLUCION

La actualización de las tablas de parámetros en la base de datos, generan las opciones de recolección del modulo Web, pero NO las de consulta del modulo java. Por lo que se debe modificar directamente el código fuente de la aplicación para su respectiva ejecución. El diseño estático de la aplicación java de consulta, se basa en los formatos del periodo censal 2005-2006 ejecutado por el DANE; por tal motivo la actualización de parámetros de la base de datos, en relación al sistema se tomo estrictamente de los lineamientos que se utilizaron en el censo 2005-2006. ya que dicha estructura no cambia y por tal razón el diseño del modelo se hizo estático.

Se recomienda para actualizaciones futuras de mayor importancia y para ofrecer compatibilidad con los formatos de los siguientes periodos censales, que se cambie el código de la aplicación java por uno basado en un modelo dinámico que pueda obtener todos los parámetros necesarios directamente desde la base de datos.

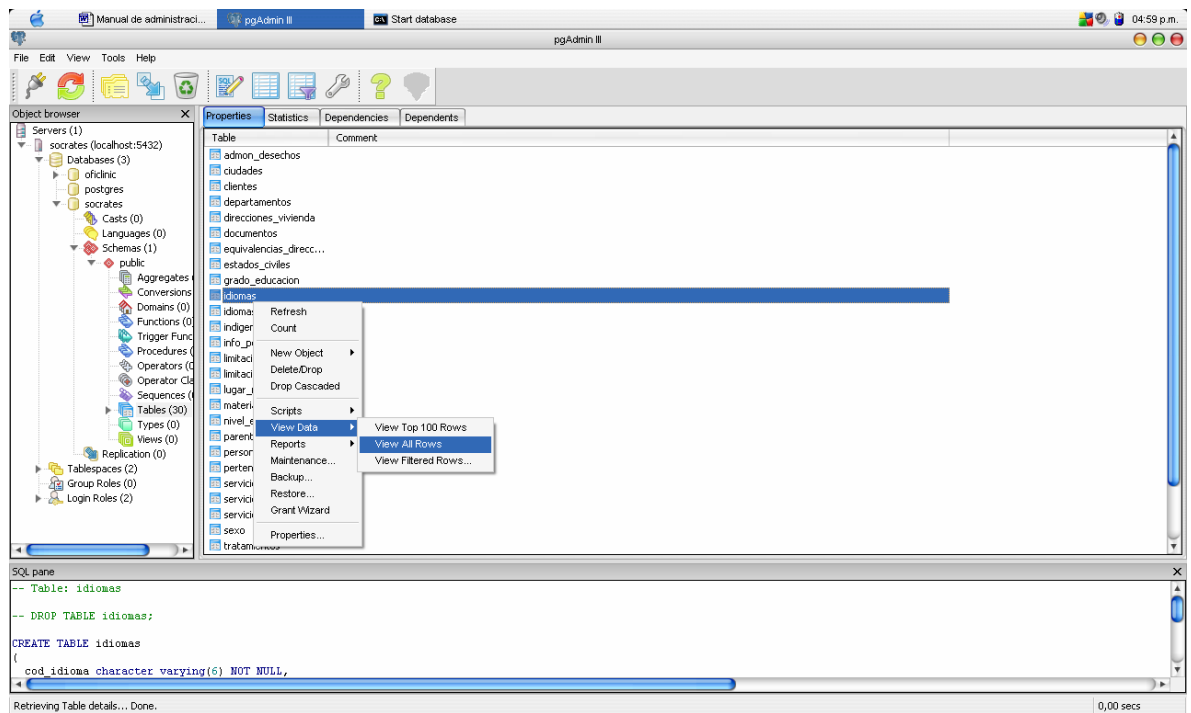
Para insertar parámetros a la base de datos se deben seguir lo siguientes pasos:

- a- En el software pgAdmin (incluido con PostgreSQL) se ingresa a la base de datos Sócrates.



- b- se selecciona la tabla de parámetros a la que se la va a adicionar datos y se abre con el comando “view all rows” (ejemplo: tabla idiomas).





- c- Se abre una nueva ventana mostrando todos los datos existentes, solo es cuestión de ir al ultimo registro y añadir el nuevo parámetro.

	cod_idioma [PK] caracte	nombre_idior character var
1	idio01	Español
2	idio02	Ingles
3	idio03	Frances
4	idio04	Italiano
5	idio05	Aleman
6	idio06	Portugues
*		

Scratch pad

6 rows.

Para adicionar los parámetros necesarios del modulo de consultas hay que modificar las listas contenidas en las siguientes partes de la aplicación java:

- servicios\_publicos
  - o Formulario frmViviendaSer.java: cbServiciosP (combobox)
- vehículos
  - o Formulario frmViviendaTrans.java: cbVehiculos (combobox)
- admon\_desechos
  - o Formulario frmViviendaSan.java: cbAdmonDesechos (combobox)
- servicios\_sanitarios
  - o Formulario frmViviendaSan.java: cbSanitario (combobox)
- material\_vivienda
  - o Formulario frmViviendaSan.java: cbMaterialVivienda (combobox)
- ciudades
  - o Formulario frmPersona.java: cbDeptoActionPerformed (metodo)
- departamentos
  - o Formulario frmPersona.java: cbDeptoActionPerformed (metodo)

- limitaciones\_fisicas
  - o Formulario frmPersonaSalud.java: cbLimitaciones (combobox)
- tratamientos
  - o Formulario frmPersonaSalud.java: cbTratamientos (combobox)
- idiomas
  - o Formulario frmPersonaEdu.java: cbIdiomas (combobox)
- documentos
  - o Formulario frmPersona.java: cbTipoDoc (combobox)
- sexo
  - o Formulario frmPersona.java: cbSexo (combobox)
  - o Formulario frmPersonaSalud.java: cbSexo (combobox)
- parentesco
  - o Formulario frmPersonaSalud.java: cbParentesco (combobox)
- indígenas
  - o Formulario frmPersonaEdu.java: cbIndigena (combobox)
- estados\_civiles
  - o Formulario frmPersona.java: cbEstadoCivil (combobox)
- nivel\_educacion
  - o Formulario frmPersonaEdu.java: cbNivelEdu (combobox)
- grado\_educacion
  - o Formulario frmPersonaEdu.java: cbGradoEdu (combobox)
- pertenencia\_salud
  - o Formulario frmPersonaSalud.java: cbPertenenciaSalud (combobox)

Para el acceso de los clientes de la aplicación java se deben crear usuarios, esto se hace en la tabla clientes, la cual permite el acceso de las organizaciones de servicios a la comunidad tener acceso a la base de datos mediante la aplicación java.

La tabla clientes se compone de 2 campos:

- nombre\_cliente: nombre asignado a la persona / organización a la cual se le entrega la aplicación java.
- Passwd: contraseña de acceso.

Esta es la única tabla que se necesita manejar para el acceso de clientes referente a la aplicación java, para el modulo de recolección no se necesita ya que la función de acceso la maneja el mismo gestor de bases de datos.

Para el mantenimiento de la base de datos se debe recurrir a la ventana de comandos SQL e ingresar el siguiente comando:

*\$VACUUM --all --full --verbose --analyze*

Esto permite que la base de datos mantenga su eficiencia en todo momento, pero este comando solo debe ejecutarse una vez a la semana y en horarios en los que no haya uso de la base de datos, porque durante el proceso hay bloqueo de tablas y la base de datos no puede utilizarse.

Los backups de la base de datos deben hacerse al menos una vez al mes ya que la gran cantidad de información que puede almacenar la base de datos puede crecer de forma casi exponencial al no borrar ningún registro por ser diseñada para consulta de datos en el tiempo.

Para hacer los backups de la base de datos se ejecuta el siguiente comando (hay que tener los respectivos privilegios para ello):

```
pg_dump socrates > socrates.sql (archivo plano sql)  
pg_dump socrates > socrates.dump.out (archivo binario)  
pg_dump socrates | gzip -c > socrates.dump.out.gz (archivo comprimido)
```

Para restaurar el backup de la base de datos se utiliza el siguiente comando:

```
psql -d socrates -f socrates.dump.out
```

## 2. MANUALES DE USUARIO CATALOGO DE FUENTES

SOCRATES-EC: SISTEMA DE INFORMACIÓN POBLACIONAL BASADO EN  
PROCEDIMIENTOS DE CENSADO

Por:

Danko Jiménez Londoño  
Daniel Andrés Valencia V.

## 2.1 MANUAL DE USUARIO PARA EL LLENADO DE LA INFORMACIÓN EN LOS FORMULARIOS WEB

SOCRATES-EC: SISTEMA DE INFORMACIÓN POBLACIONAL BASADO EN  
PROCEDIMIENTOS DE CENSADO

Por:

Danko Jiménez Londoño  
Daniel Andrés Valencia V.

## Funcionamiento De La Aplicación:

La aplicación “Formulario para la entrada y actualización de datos poblacional”. Permite la creación, control de formularios obtenidos para su presentación en las Dependencias de la entidad solicitada, bien físicamente o vía Web. Una vez instalada la aplicación, podremos introducir todos los datos necesarios para la presentación de dicho Formulario a cualquier entidad que este interesada en ello. Tras ejecutar la aplicación.

*Visualizaremos La Pantalla De Presentación.*



*Este Botón Permitirá ingresar si es nuevo usuario.*



*Este Botón Permitirá actualizar o modificar datos si es un usuario que ya este registrado en el sistema.*

OPCION 1: CUANDO SE ES NUEVO USUARIO EN EL SISTEMA...

*A Continuación Veremos El Formulario De Llenado De Datos Personales*



Numero del documento de identidad:

Tipo de documento:  Cedula  Pasaporte  T. Identidad

Primer Apellido  Segundo Apellido

Primer Nombre  Segundo Nombre

Sexo:  Femenino  Masculino

¿Cual es su fecha de Nacimiento?  

¿Cual fue la ciudad donde nacio?

Departamento

Ciudad

Para verificar los datos ingresados y continuar con el siguiente paso, por favor haga click en el siguiente boton:

Se ingresan los datos correspondientes a cada campo y después del llenado de los campos se dará clic en el botón **Confirmar Datos/ Siguiete paso**



*A Continuación Veremos El Formulario De Llenado De Datos avanzados de Persona*



Parentesco con la cabeza de hogar

A que raza o comunidad indigena pertenece?

Estado civil



¿Sabe leer?

SI  NO

¿Sabe escribir?

SI  NO

¿Asiste actualmente a algun instituto de educacion formal?

SI  NO

- Oír aun con aparatos especiales
- Hablar
- Entender o Aprender
- Relacionarse por problemas mentales
- Bañarse, Vestirse, Alimentarse
- Otra Limitación permanente

Actualmente padece o ha padecido de alguna enfermedad que haya requerido... (Marque mas de una si es necesario. Si no, salte al boton de confirmación)

- Cirujia cardiaca
- Transplante riñon
- Transplante medula osea
- Trasplante cornea
- Transplante higado
- Neurocirujia del sistema nervioso central
- Tratamiento medico y quirurgico de trauma mayor
- Tratamiento quirurgico por enfermedad congenita
- Reemplazos articulares
- Dialisis por insuficiencia cronica
- Manejo por grandes quemaduras
- Tratamiento por vih-sida
- Quimioterapia o radioterapia para el cancer
- Unidad de cuidados intensivos

Se procederá al llenado de los campos y se dará clic en el botón **Confirmar Datos/ Siguiete paso**

*A Continuación Visualizaremos El Formulario vivienda.*

Donde se le dará click en ingresar



Siguiente Paso: ingresar informacion de la vivienda donde vive

\* Ingrese información sobre la vivienda en la que vive

A Continuación Veremos El Formulario De Llenado De Datos de vivienda



Información de Hogar/Vivienda #856525055

Dirección:

En que material esta construida principalmente la vivienda?

Bloque

Que servicios publicos posee esta vivienda?

- Agua
- Electricidad
- Telefono
- Gas
- Internet
- TV suscripcion

Como eliminan principalmente la basura en esta vivienda?

- otra forma
- entierranla
- quemanla
- tiranla
- al patio
- tiranla
- rio
- Utilizan lo
- servicios de aseo

El tipo de servicio sanitario que posee la vivienda es...

- inodoro conectado al alcantarillado
- inodoro conectado a un pozo
- inodoro sin conexion
- no tiene servicio

El servicio sanitario de la vivienda es de uso exclusivo de los miembros de la familia?

- SI
- NO

Posee la vivienda un lugar exclusivo para preparar alimentos (una cocina)?

- SI
- NO

Cuantos cuartos de baño con regadera o ducha tiene la vivienda

Cuantos cuartos en total (incluidos sala y comedor) tiene la vivienda

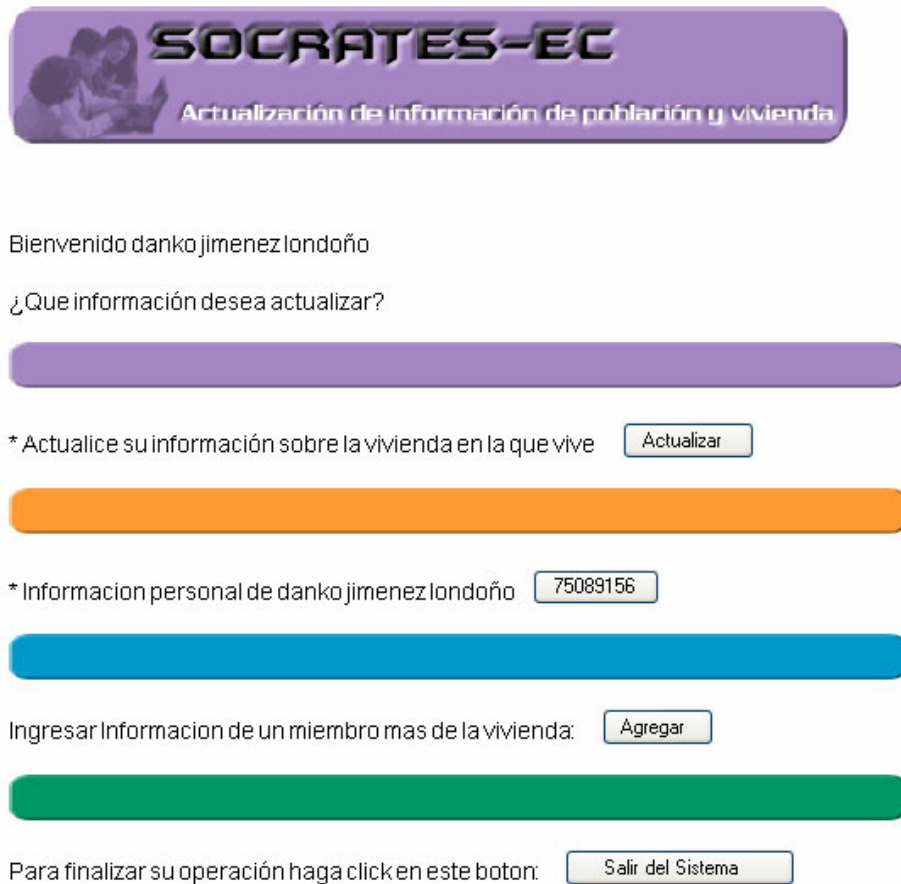
Cuantos cuartos de dormir posee la vivienda

¿que vehiculos poseen los miembros de esta vivienda?

- bicicleta
- moto
- bote
- carro
- avión

Se procederá al llenado de los campos y se dará clic en el botón **Confirmar Datos/ Siguiente paso**

Después de hacer el llenado de los datos y confirmar el sistema nos muestra en una pantalla la opción de actualizar la información de vivienda en la que vive, la información personal o ingresar la información de un miembro más de la vivienda. En caso de no ingresar mas datos se finaliza la operación dando click en el botón finalizar.



**SOCRATES-EC**  
Actualización de información de población y vivienda

Bienvenido danko jimenez londoño

¿Que información desea actualizar?

\* Actualice su información sobre la vivienda en la que vive

\* Información personal de danko jimenez londoño

Ingresar Información de un miembro mas de la vivienda:

Para finalizar su operación haga click en este boton:

## OPCION 2: CUANDO YA SE ESTA REGISTRADO Y SE VAN A MODIFICAR DATOS EXISTENTES...

Este es el formulario de ingreso al sistema.



Favor ingresar los siguientes datos para ingresar al sistema

Numero del documento de identidad		<input type="text"/>
Tipo de documento		<input type="radio"/> Cedula <input type="radio"/> Pasaporte <input type="radio"/> T. Identidad
Primer Apellido	Segundo Apellido	
<input type="text"/>	<input type="text"/>	
Primer Nombre	Segundo Nombre	
<input type="text"/>	<input type="text"/>	

Ingrese los datos que se le solicitan para comprobar que existe en el sistema:

- Numero de documento de identidad.
- Tipo de documento de identidad.
- Apellidos y nombres.

Después de estar registrados en el sistema nos dará la opción de modificación de datos como lo podemos observar en el siguiente formulario.



Bienvenido danko jimenez londoño

¿Que información desea actualizar?

\* Actualice su información sobre la vivienda en la que vive

Actualizar

\* Información personal de danko jimenez londoño

75089156

Ingresar Información de un miembro mas de la vivienda:

Agregar

Para finalizar su operación haga click en este boton:

Salir del Sistema

Se puede escoger entre modificar los datos de la vivienda en la que vive o los datos personales de los miembros de la vivienda. Para mostrar un ejemplo se da clic en el botón para actualizar información de vivienda.

Este es el formulario para modificación de datos existentes de vivienda.



Informacion de Hogar / Vivienda # 2074143109

Direccion:

En que material esta construida principalmente la vivienda?



Que servicios publicos posee esta vivienda?

- Agua
- Electricidad
- Telefono
- Gas
- Internet
- TV suscripcion

...

SI  NO

Posee la vivienda un lugar exclusivo para preparar alimentos (una cocina)?

SI  NO

Cuantos cuartos de baño con regadera o ducha tiene la vivienda?

Cuantos cuartos en total (incluidos sala y comedor) tiene la vivienda?

Cuantos cuartos de dormir posee la vivienda?

¿que vehiculos poseen los miembros de esta vivienda?

- bicicleta
- moto
- bote
- carro
- avión

Se procederá al llenado de los campos o a su modificación y se dará clic en el botón **Confirmar Datos/ Siguiente paso**.

*Esto nos llevara de nuevo formulario inicial de modificación o actualización de datos.*



## 2.2 MANUAL DE USUARIO: SÓCRATES-EC ADMINISTRACIÓN Y CONSULTAS

### SOCRATES-EC: SISTEMA DE INFORMACIÓN POBLACIONAL BASADO EN PROCEDIMIENTOS DE CENSADO

Por:

Danko Jiménez Londoño  
Daniel Andrés Valencia V.

## Funcionamiento De La Aplicación

Permite la obtención de información relacionada con la población, esta información ya debe haberse recolectado previamente usando el sistema de formularios Web. Una vez instalada la aplicación, podremos extraer todos los datos necesarios para cumplir con las solicitudes de cualquier entidad que ofrece servicios a la comunidad y que este interesada en dicha información.

NOTA: para el correcto funcionamiento de la aplicación se requiere acceso a Internet.

Tras ejecutar la aplicación veremos por primera vez esta ventana:



The image shows a screenshot of a web application window titled "Socrates-EC Consultas y Administración". The window has a standard operating system title bar with minimize, maximize, and close buttons. On the left side, the text "SOCRATES-EC" is displayed in a large, bold, black font, with "Sistema de Información Poblacional" written in a smaller font below it. On the right side, there is a login form titled "Acceso al sistema". This form contains two input fields: "Nombre de usuario" and "Contraseña". To the right of these fields is a blue button labeled "Acceso".

A continuación ingresaremos nuestro nombre de usuario y contraseña (previamente suministradas por los administradores de la base de datos) y daremos clic en el botón "Acceso".

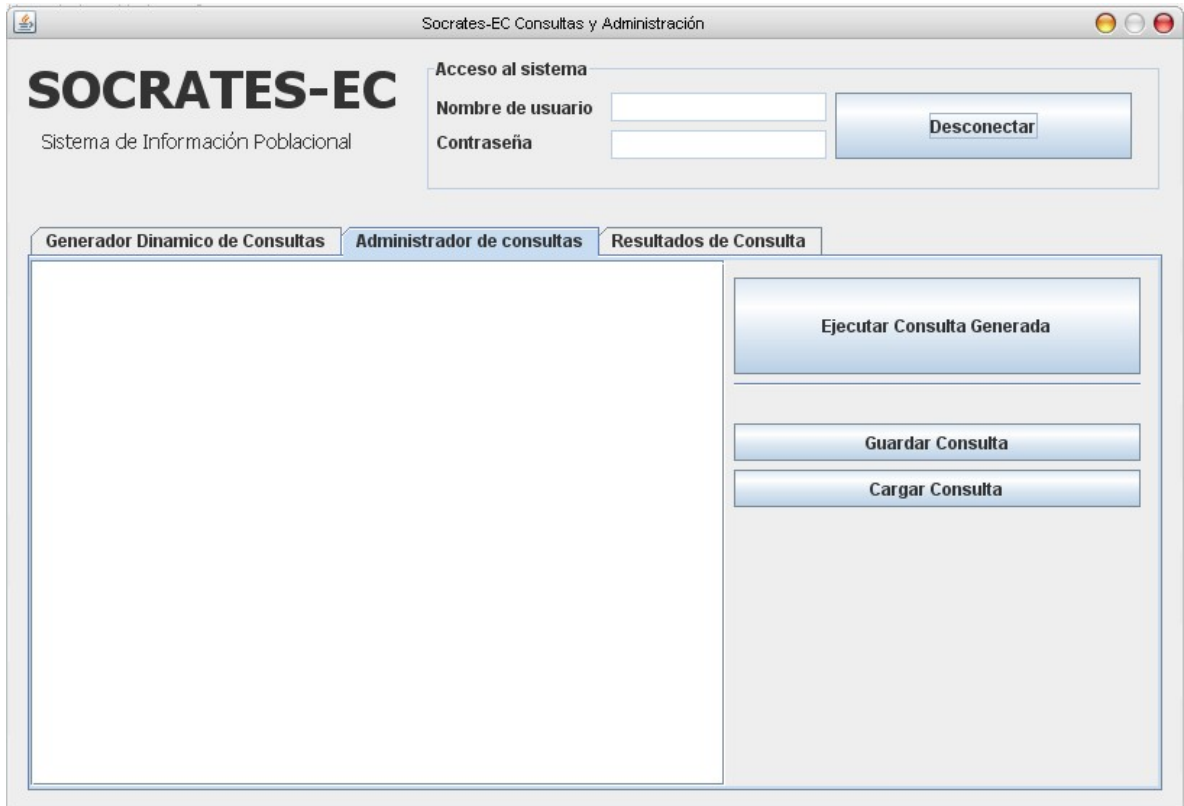
Si el nombre de usuario y la contraseña ingresados son correctos la ventana debe verse ahora así:



La ventana mostrara un área con tres pestañas con diferentes opciones, siendo la primera de ellas la pestaña llamada “Generador dinámico de consultas”, aquí es donde están las todas las herramientas necesarias para hacer solicitudes consulta de información a la base de datos en línea de Sócrates-EC.

NOTA: para desplazarse entre pestañas solo tiene que dar clic sobre cada una de ellas.

La segunda pestaña muestra el administrador de consultas:



Esta pestaña es donde se muestran y ejecutan las solicitudes de consulta generadas por el sistema de generador de consultas, aquí se pueden adicionalmente guardar las consultas en disco y cargar consultas almacenadas.

La tercera pestaña muestra el área de respuesta llamada “Resultados de Consulta”:

The screenshot shows a web browser window titled "Socrates-EC Consultas y Administración". The main header displays "SOCRATES-EC" and "Sistema de Información Poblacional". On the right, there is a login section titled "Acceso al sistema" with fields for "Nombre de usuario" and "Contraseña", and a "Desconectar" button. Below the header, there are three tabs: "Generador Dinamico de Consultas", "Administrador de consultas", and "Resultados de Consulta", with the third tab being active. Under the active tab, there is a button labeled "Exportar resultados a un archivo CSV (compatible con Excel)". Below this button is a large table with 10 columns labeled "Title 1" through "Title 10" and multiple empty rows. The table has a scrollbar on the right side.

Esta área proporciona una tabla que por defecto permanece vacía, pero una vez ejecutada la consulta generada, muestra los resultados solicitados. Estos datos pueden ser exportados a un archivo CSV compatible con muchas otras aplicaciones de manejo de información.

¿Cómo crear una solicitud de consulta?

NOTA: para esta parte se asume que se esta en esta pantalla:

The screenshot shows the 'SOCRATES-EC' web application interface. At the top, there is a title bar 'Socrates-EC Consultas y Administración'. Below it, the main header displays 'SOCRATES-EC' and 'Sistema de Información Poblacional'. To the right, there is a login section titled 'Acceso al sistema' with input fields for 'Nombre de usuario' and 'Contraseña', and a 'Desconectar' button. Below the header, there are three tabs: 'Generador Dinamico de Consultas' (selected), 'Administrador de consultas', and 'Resultados de Consulta'. The main content area is titled 'CONSULTAS DISPONIBLES' and lists three categories of queries: 1. 'Consultas de tipo PERSONA: se definen como las consultas que se van a realizar sobre la población. Los tipos de consultas disponibles de esta categoria son: Datos Básicos, Información de Salud, e Información de Educación y Cultura.' with buttons for 'Persona: Datos Básicos', 'Persona: Información de Salud', and 'Persona: Educación y Cultura'. 2. 'Consultas de tipo VIVIENDA: se definen como las consultas que se van a realizar sobre las viviendas/hogares. Los tipos de consultas disponibles son: Información de Sanidad, Servicios Públicos, y Medios de Transporte.' with buttons for 'Vivienda: Información de Sanidad', 'Vivienda: Servicios Publicos', and 'Vivienda: Transporte'. 3. 'Consulta de Relación VIVIENDA-PERSONA: permite "conectar" una población especifica con un área habitable en particular.' with a button for 'Relacion Vivienda - Persona'. At the bottom, there is a section for 'Modo Avanzado: solamente para expertos, requiere Lenguaje SQL' with a button for 'Consulta Avanzada'.

Las consultas que se pueden hacer son de 3 tipos:

- Consultas de tipo Persona.
- Consultas de tipo Vivienda.
- Consultas de relación Vivienda-Persona.

Hay adicionalmente un modo de consulta llamado "Consulta Avanzada" que nos permite hacer solicitudes especiales de consulta, pero se requieren conocimientos en lenguaje SQL por lo que solo los usuarios expertos pueden manejar este modo de consulta.

¿Cómo solicitar una consulta?

Ejemplo:

Para aprender el mecanismo básico de consulta, haremos una consulta de persona; damos clic en el botón “Persona: Datos Básicos” y enseguida nos debe aparecer una ventana como esta:

Persona: Datos Basicos

Documento de Identidad

Tipo de documento

Cedula

Edad =

Lugar de nacimiento

Departamento:

Municipio

Estado Civil

soltero

Primer Nombre

Segundo Nombre

Primer Apellido

Segundo Apellido

Sexo Femenino

Operación de resumen

Total

Sobre el campo...

Documento de identidad

Generar Consulta

Los tipos de datos que podemos solicitar están organizados utilizando tres tipos de entrada:

- Entrada por campo de texto: en el cual podemos digitar directamente la información que queremos o solamente seleccionar el tipo de dato (modo dual).
- Entrada por campo de selección: aquí se selecciona la información conocida y se valida como condición de la consulta (ej. Personas nacidas en el departamento de caldas en el municipio de Manizales).
- Entrada hibrida: aquí no solo encontramos un campo de texto si no también uno de selección que nos permite manejar una condición adicional.

Las entradas por campo de texto requieren especial atención ya que manejan una característica llamada modo dual, es decir, cuando seleccionamos un campo de

texto pero lo dejamos vacío, le damos a entender al sistema que solo nos muestre todos registros que tengan ese campo de texto

Ej. Si seleccionamos el campo de texto de documento de identidad pero lo dejamos vacío le indicamos al sistema que nos muestre todos los números de documento de identidad.

Ya siendo el caso de que ingresemos alguna información en el campo de texto no solo le estamos pidiendo al sistema que nos muestre los registros que tengan ese campo de texto si no que adicionalmente esos registros TIENEN que ser iguales al contenido del campo de texto.

Ej. Si teniendo seleccionado el campo de texto de documento de identidad y le ingresamos un valor como 75096157 le estamos diciendo al sistema que busque todos los registros que coincidan con ese campo.

Para ponerlo mas claro vamos a hacer una consulta de ejemplo:

Para seleccionar los campos que queremos solo tenemos que marcar los cuadros adyacentes (o encima) de los campos que queremos utilizar y estos automáticamente se activaran para su uso:



En este ejemplo hemos seleccionado Documento de identidad, Tipo de documento, Edad, Primer nombre, Segundo nombre, Primer apellido, y Segundo apellido.

Nótese los siguientes detalles:

1. hemos seleccionado documento de identidad y le hemos ingresado el número 75096157 para decirle al sistema que nos muestre los registros de documento de identidad que contengan ese numero.
2. hemos seleccionado el tipo de documento como cedula dándole la orden al sistema de que adicionalmente el tipo de documento de identidad cumpla exactamente con ese criterio.
3. los campos de edad, nombres y apellidos de han seleccionado pero se han dejado en blanco con el propósito de mostrar estos datos adicionales, pero sin ninguna condición en particular, es decir: que me muestre todas las edades, nombres y apellidos.

Una vez hecho esto solo damos clic en el botón “generar consulta”, esto cierra esta ventana y nos regresa a la aplicación principal. En un principio puede parecer que no hizo absolutamente nada así que para comprobarlo le damos clic a la pestaña de “Administrador de consultas”:

Socrates-EC Consultas y Administración

# SOCRATES-EC

Sistema de Información Poblacional

**Acceso al sistema**

Nombre de usuario

Contraseña

**Generador Dinamico de Consultas** | **Administrador de consultas** | **Resultados de Consulta**

```

select persona.doc_persona, documentos.tipo_doc, age(current_date, fecha_nac), per
sona.nombre1, persona.nombre2, persona.apellido1, persona.apellido2 FROM sexo I
INNER JOIN (((documentos INNER JOIN persona ON documentos.cod_doc = persona.
cod_doc) INNER JOIN (parentesco INNER JOIN (estados_civiles INNER JOIN info_per
sona ON estados_civiles.cod_estado_civil = info_persona.cod_estado_civil) ON parent
esco.cod_parentesco = info_persona.cod_parentesco) ON persona.doc_persona = inf
o_persona.doc_persona) INNER JOIN (departamentos INNER JOIN (ciudades INNER
JOIN lugar_nacimiento ON ciudades.cod_ciudad = lugar_nacimiento.cod_ciudad) ON (
departamentos.cod_depto = lugar_nacimiento.cod_depto) AND (departamentos.cod_d
epto = ciudades.cod_depto)) ON persona.doc_persona = lugar_nacimiento.doc_perso
na) ON sexo.cod_sexo = persona.cod_sexo where(((persona.doc_persona) = '750961
57') and ((documentos.tipo_doc) = 'Cedula'))

```

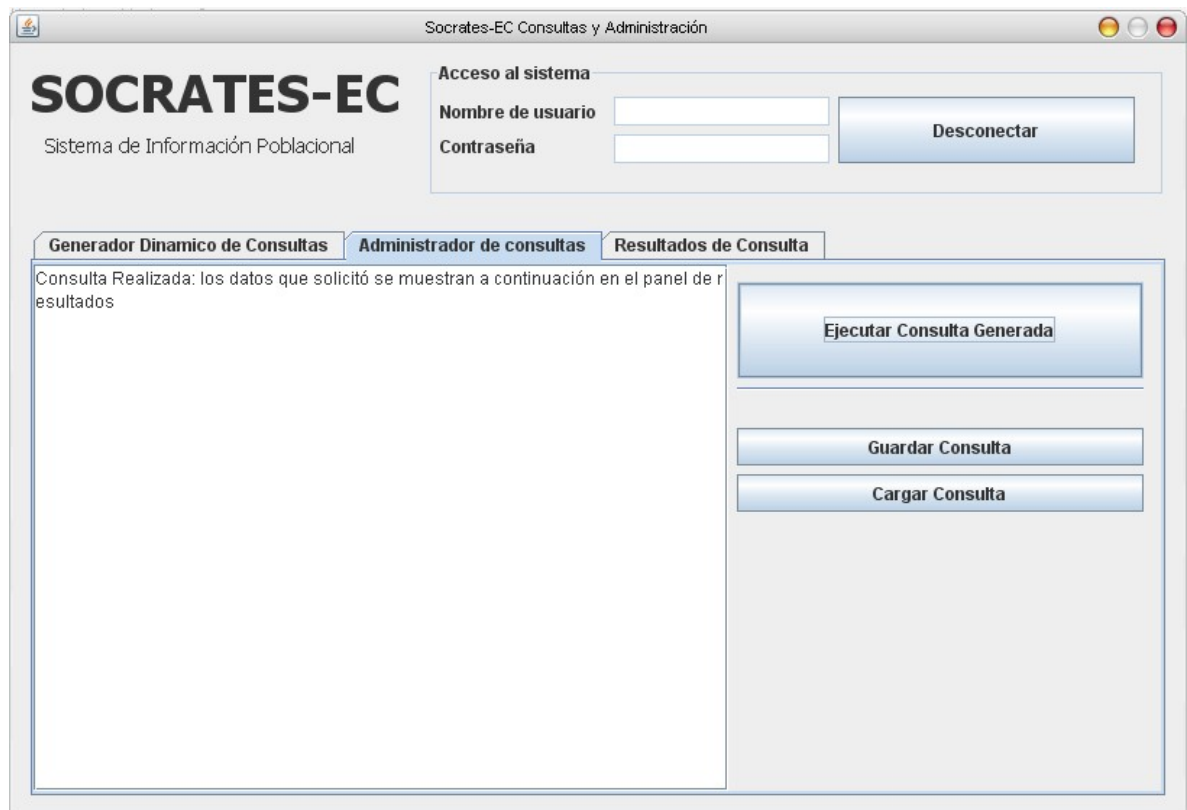
---

Ese texto es el resultado del sistema de generación de consultas de la aplicación y que nos permitirá hacer la solicitud de los datos que deseamos a la base de datos en línea.

Para ejecutar esta consulta damos clic en el botón “Ejecutar consulta generada”.

En el momento en el que se ejecuta dicho comando, la aplicación envía el texto generado hacia la base de datos diciéndole que es el tipo de consulta que deseamos (en este caso los datos básicos de la persona que tiene la cedula numero 75096157).

En el momento en que se recibe los resultados el área de texto del administrador de consultas cambiara a un indicador que nos dirá que la consulta fue realizada con éxito:



A continuación damos clic en la pestaña “Resultados de Consulta”:

Acceso al sistema

Nombre de usuario

Contraseña

Desconectar

Generador Dinamico de Consultas    Administrador de consultas    **Resultados de Consulta**

Exportar resultados a un archivo CSV (compatible con Excel)

Title 1	Title 2	Title 3	Title 4	Title 5	Title 6	Title 7	Title 8	Title 9
75096157	Cedula	27 years 7 mons 19 days	DANIEL	ANDRES	VALENCIA	VALENCIA		
75096157	Cedula	27 years 7 mons 19 days	DANIEL	ANDRES	VALENCIA	VALENCIA		
75096157	Cedula	27 years 7 mons 19 days	DANIEL	ANDRES	VALENCIA	VALENCIA		
75096157	Cedula	27 years 7 mons 19 days	DANIEL	ANDRES	VALENCIA	VALENCIA		
75096157	Cedula	27 years 7 mons 19 days	DANIEL	ANDRES	VALENCIA	VALENCIA		

NOTA: como la base de datos es capaz de guardar información histórica, los datos aquí mostrados corresponden a diferentes actualizaciones de información en la base de datos.

Hemos visto como se puede generar una consulta a partir de la ventana de datos básicos de persona, para las demás ventanas de consulta el procedimiento es el mismo, solo seleccionar las opciones que queremos y en algunos casos ingresar unos pocos datos que refinen la información que necesitamos.

A continuación se muestran las ventanas de consulta y se hace una breve descripción de lo que se puede hacer con ellas, también tendremos en cuenta la ventana de consulta avanzada.

## Tipos de ventanas de consulta

### Consulta de persona: Datos Básicos

The screenshot shows a window titled "Persona: Datos Basicos" with a standard Mac OS window header. The window is divided into two main sections. The left section contains several input fields, each preceded by a checkbox: "Documento de Identidad" (text input), "Tipo de documento" (dropdown menu with "Cedula" selected), "Edad" (text input with a small calculator icon), "Lugar de nacimiento" (with sub-sections for "Departamento:" and "Municipio", both dropdown menus), and "Estado Civil" (dropdown menu with "soltero" selected). The right section also has checkboxes: "Primer Nombre", "Segundo Nombre", "Primer Apellido", and "Segundo Apellido" (all text inputs); "Sexo" (dropdown menu with "Femenino" selected); and "Operación de resumen" (dropdown menu with "Total" selected). Below this is a "Sobre el campo..." dropdown menu with "Documento de identidad" selected. At the bottom right of the window is a large button labeled "Generar Consulta".

Los datos que se pueden extraer con esta ventana son los siguientes:

- Documento de identidad y tipo de documento (cedula, tarjeta de identidad o pasaporte).
- Edad de la persona.
- Lugar de nacimiento.
- Estado civil.
- Nombres y apellidos.
- Sexo.

## Consulta de persona: Educación y cultura

Persona: Educación y Cultura

Documento de Identidad

Primer Nombre

Segundo Nombre

Primer Apellido

Segundo Apellido

Edad

¿Sabe Leer? Si

¿Sabe Escribir? Si

¿Asiste a algun instituto educativo? Si

Nivel Educativo Actual Preescolar

Ultimo Grado Cursado 1° primaria

Conocimiento del Idioma Español

Raza / Pertenecia Indigena Ninguno

Operación de resumen

Total

Sobre el campo...

Documento de identidad

Generar Consulta

Los datos que se pueden extraer con esta ventana son los siguientes:

- Documento de identidad (numero).
- Nombres y apellidos.
- Edad.
- Conocimientos básicos de lectura/escritura.
- Asistencia actual a algún centro educativo.
- Nivel de educación y último grado cursado.
- Conocimiento de algún idioma.
- Raza y/o pertenencia indígena.

## Consulta de persona: Salud

Persona: Salud

Tipo de Consulta: Limitaciones Físicas Y Tratamientos Especiales

Documento de Identidad

Parentesco con la cabeza de hogar: cabeza o jefe

Operación de resumen: Total

Primer Nombre

Segundo Nombre

Primer Apellido

Segundo Apellido

Pertenencia a Sistema de Salud: Instituto de seguros sociales

Sexo: Femenino

Edad: =

Limitaciones Físicas / Mentales: Moverse o Caminar

Tratamientos Especiales: Cirujía cardíaca

Sobre el campo...: Documento de identidad

Generar Consulta

La ventana de salud posee tres modalidades: una para búsqueda con limitaciones físicas, otra para tratamientos médicos especiales y la última es una búsqueda combinada en caso de que los datos a extraer tengan que ver con personas que tengan las 2 situaciones.

Los datos que se pueden extraer con esta ventana son:

- Documento de identidad.
- Nombres y apellidos.
- Sexo.
- Edad.
- Parentesco con la cabeza del hogar.
- Pertenencia a algún sistema de salud.
- Limitaciones físicas y/o mentales.
- Tratamientos médicos especiales.

## Consulta de vivienda: Sanidad

Windows title: Vivienda: Sanidad

Rango de vias:

Ubicación inicial...    Número

Via Principal    CL    [ ]

Via de Cruce    CL    [ ]

Ubicación final...    Número

Via Principal    CL    [ ]

Via de Cruce    CL    [ ]

Una vía específica

CL    [ ]

Lugar exclusivo para Cocina    Si    [ ]

Uso exclusivo del servicio sanitario    Si    [ ]

Tipo de Servicio Sanitario    Inodoro conectado al alcantarillado    [ ]

Número de baños    1    [ ]

Número de dormitorios    1    [ ]

Número total de cuartos    1    [ ]

Material base de vivienda    Bloque    [ ]

Administración de basuras    otra forma    [ ]

Operación de resumen

Total    [ ]

Sobre la tabla...

Lugar para la cocina    [ ]

Generar Consulta

NOTA: todas las ventanas de vivienda tienen un área de búsqueda de direcciones que permite buscar datos en un área específica (un área creada por dos ubicaciones específicas) o a lo largo de una vía en especial.

Los datos que se pueden extraer con esta ventana son:

- Dirección de la vivienda.
- Lugar exclusivo para una cocina.
- Uso exclusivo (no compartido) del servicio sanitario para los habitantes de la vivienda.
- Número de baños.
- Número de habitaciones para dormir.
- Total de cuartos (incluidos cocina, baños, dormitorios, etc).
- Material principal de la vivienda.
- Manejo de desechos y basuras.



## Consulta de vivienda: Servicios públicos

Rango de vías:

Ubicacion inicial... Numero

Via Principal CL ▾

Via de Cruce CL ▾

Ubicacion final... Numero

Via Principal CL ▾

Via de Cruce CL ▾

Una vía específica:

CL ▾

Servicios Publicos Instalados

Agua ▾

Numero total de cuartos 1 ▾

Numero de dormitorios 1 ▾

Lugar exclusivo para Cocina Si ▾

Operación de resumen

Total ▾

Sobre la tabla...

Lugar para la cocina ▾

Generar Consulta

Los datos que pueden extraerse con esta ventana son los siguientes:

- Dirección de la vivienda.
- Servicios públicos instalados en la vivienda (agua, energía, teléfono, etc.).
- Numero total de cuartos.
- Numero de dormitorios.
- Ubicación exclusiva de la cocina.

## Consulta de vivienda: Transporte

**Vivienda: Transporte**

**Rango de vías:**

**Ubicacion inicial...**      **Numero**

Vía Principal    CL   

Vía de Cruce    CL   

**Ubicacion final...**      **Numero**

Vía Principal    CL   

Vía de Cruce    CL   

**Una vía específica**

CL   

**Vehiculos disponibles**

bicicleta   

**Operación de resumen**

Total   

**Sobre la tabla...**

Lugar para la cocina   

**Generar Consulta**

Esta ventana solo permite consultar información sobre los tipos de vehículos que poseen los habitantes de la vivienda.

Los datos que pueden extraerse con esta ventana son:

- Dirección de la vivienda.
- Tipo de vehículo que poseen los habitantes de la vivienda.

## Consulta de Vivienda: Persona

Rango de vías:

Ubicacion inicial... Numero

Via Principal CL

Via de Cruce CL

Ubicacion final... Numero

Via Principal CL

Via de Cruce CL

Una via especifica

CL

Rango de actualizacion

desde  Meses hasta hoy

Lugar exclusivo para Cocina Si

Uso exclusivo del servicio sanitario Si

Numero total de cuartos 1

Numero de dormitorios 1

Numero de baños 1

Documento de Identidad

Primer Nombre

Segundo Nombre

Primer Apellido

Segundo Apellido

Operación de resumen

Total

Sobre el campo...

Documento de identidad

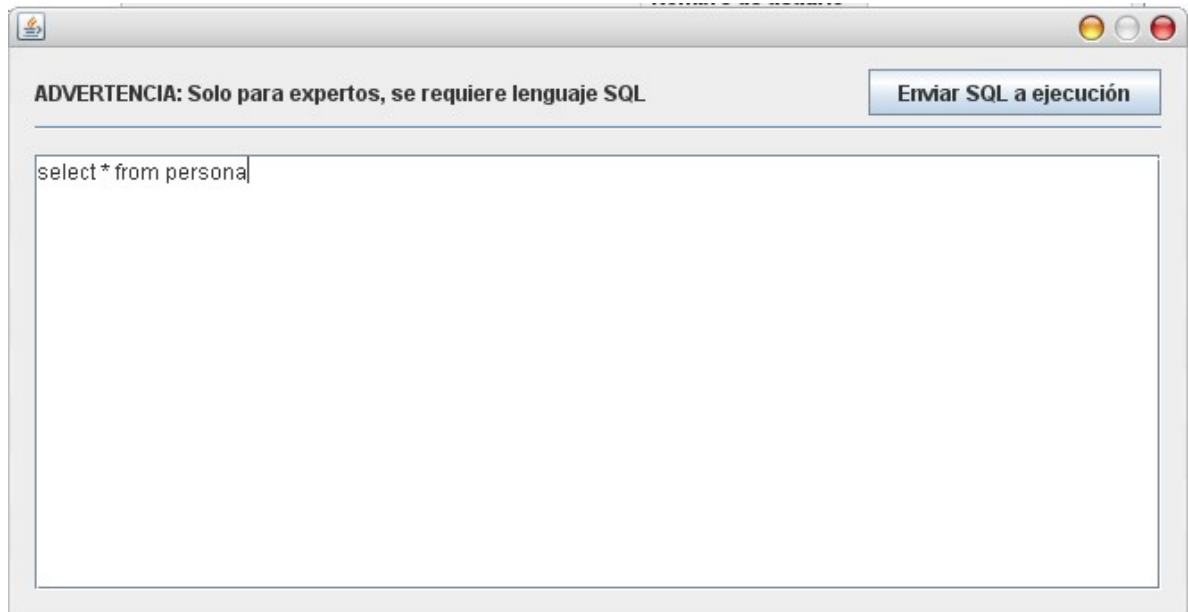
**Generar Consulta**

Esta ventana enlaza información de la vivienda con información básica de los habitantes de la misma.

Los datos que pueden extraerse con esta ventana son:

- Dirección de la vivienda.
- Lugar exclusivo para una cocina.
- Numero total de cuartos.
- Numero de habitaciones para dormir.
- Numero de baños.
- Documento de identidad de la persona.
- Nombres y apellidos.

## Consulta Avanzada:



Esta ventana puede parecer muy sencilla ya que no tiene datos que seleccionar pero de hecho esta ventana es la que menos se debe utilizar si no se conoce a fondo la base de datos en línea.

Para utilizar esta ventana se requieren conocimientos de lenguaje estructurado de consulta (llamado también SQL) y saber la estructura interna de la base de datos, por lo que solo los expertos como ingenieros entrenados pueden utilizarla.

NOTA: por razones de confiabilidad esta ventana NO permite consultas de inserción y/o modificación a la base de datos.

¿Cómo guardamos y cargamos las consultas?

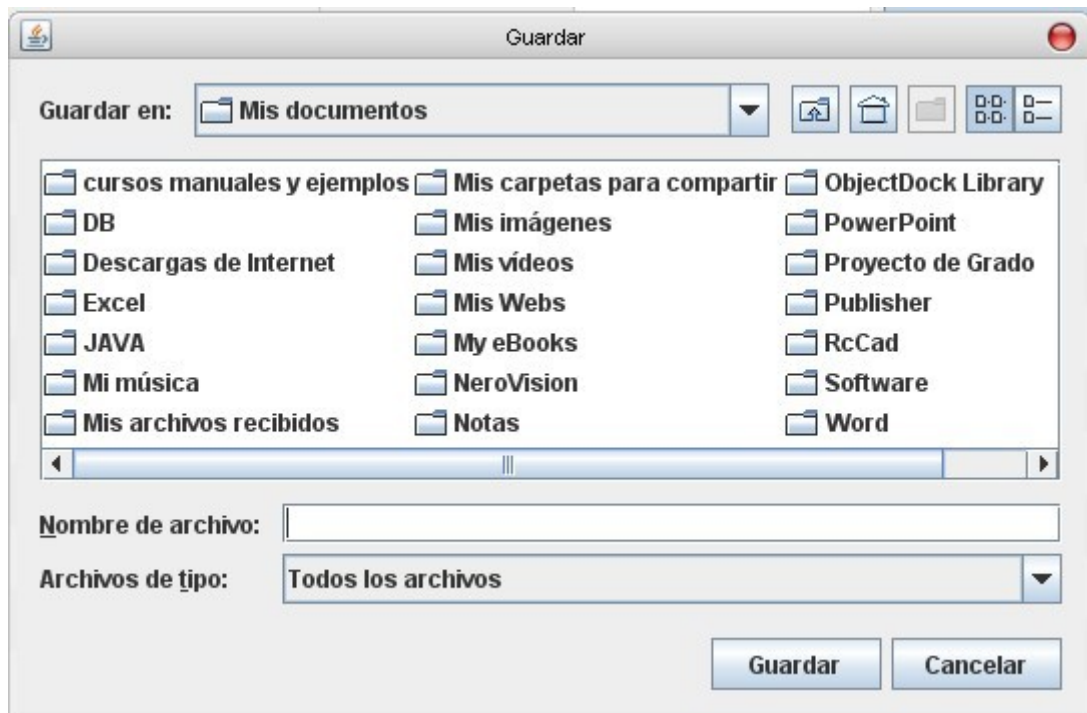
Cuando creamos una consulta y deseamos utilizarla para mas tarde podemos guardarla en disco, el procedimiento es el siguiente:

Primero que todo no aseguramos de ver la consulta en el área de administración de consultas

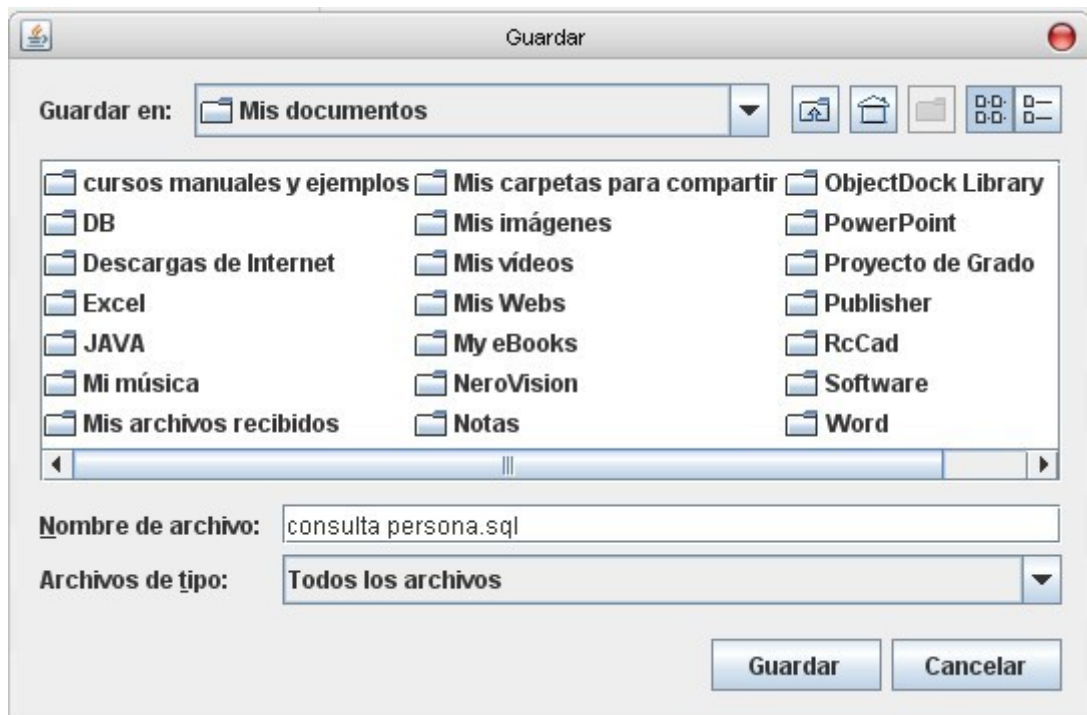


Podemos encontrar 3 botones, el más grande es para ejecutar la consulta y los otros 2 son para guardar la consulta en disco y para cargar la consulta previamente guardada.

Damos clic en el botón “guardar consulta” y nos debe aparecer una ventana como esta:



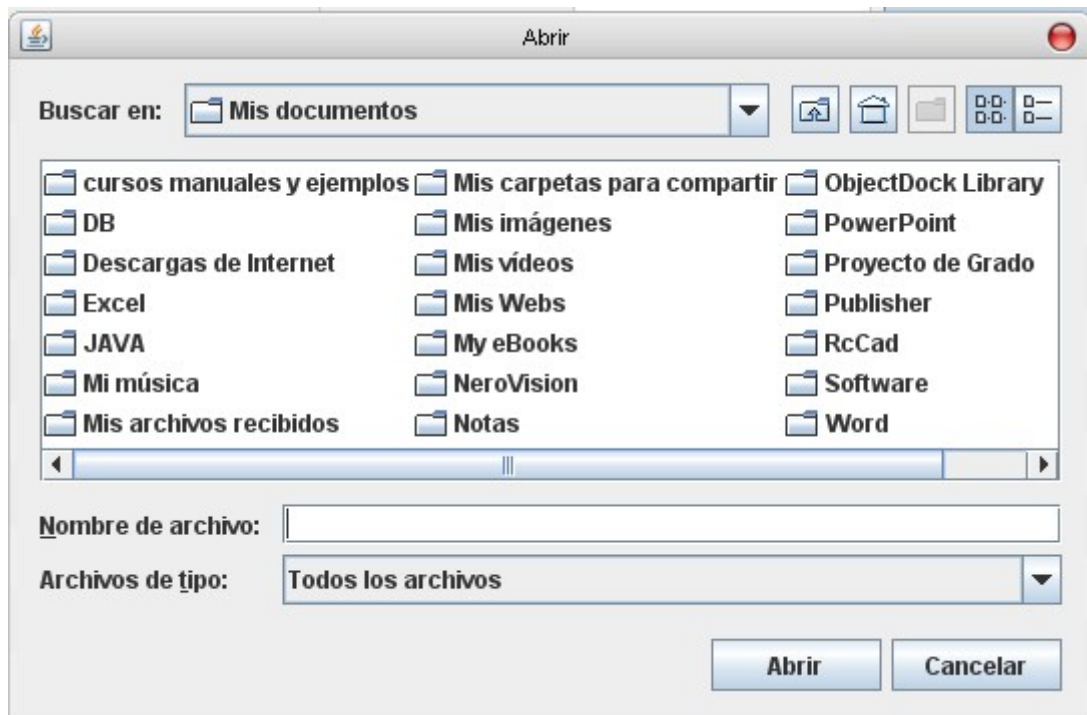
Solo seleccionamos la ubicación en la que queremos guardar nuestro archivo de consulta y le damos un nombre (Ej. “consulta persona.sql”)



NOTA: siempre se debe poner después del nombre del archivo la extensión “.sql”

Después solo tenemos que dar clic en el botón guardar.

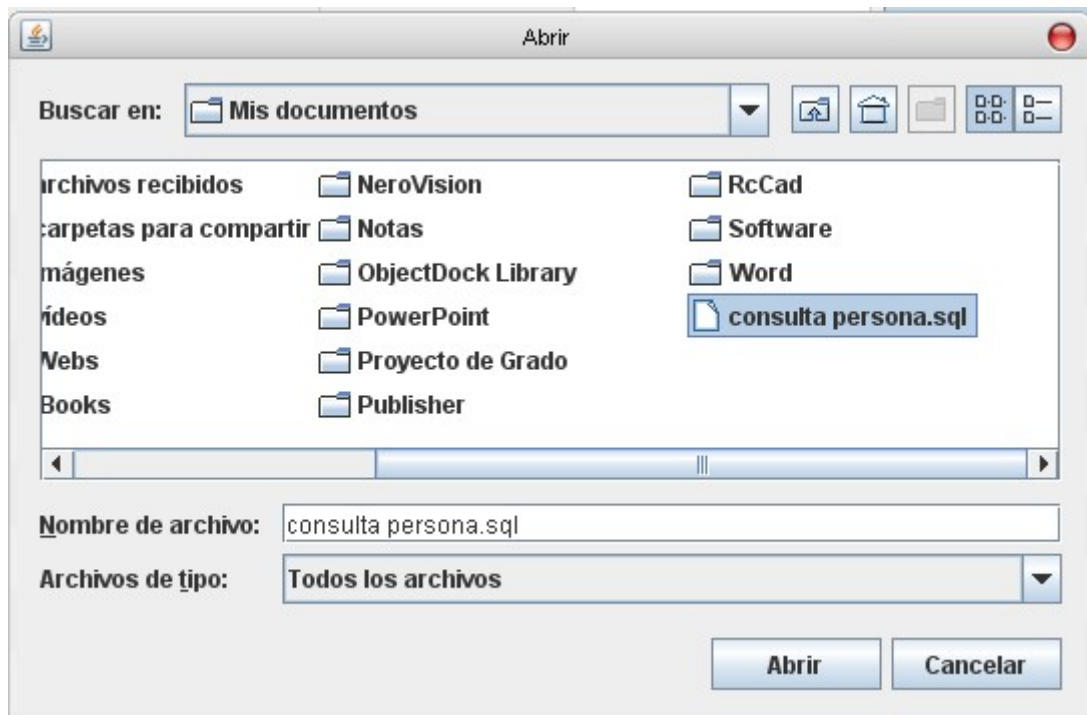
Para cargar una consulta guardada solo tenemos que darle clic al botón “cargar consulta” en el área de administración de consultas, esto nos abrirá una ventana muy similar a la de guardar.



Buscamos el archivo de consulta que queremos cargar, este archivo tiene que tener la extensión “.sql” al final del nombre (Ej. “consulta persona.sql”)



Seleccionamos el archivo que queremos cargar, la ventana debe entonces verse así:



Después de eso solo tenemos que dar clic en el botón "abrir" y la aplicaron cargara el archivo con la consulta previamente guardada que necesitamos.

¿Cómo exportar los resultados de una consulta?

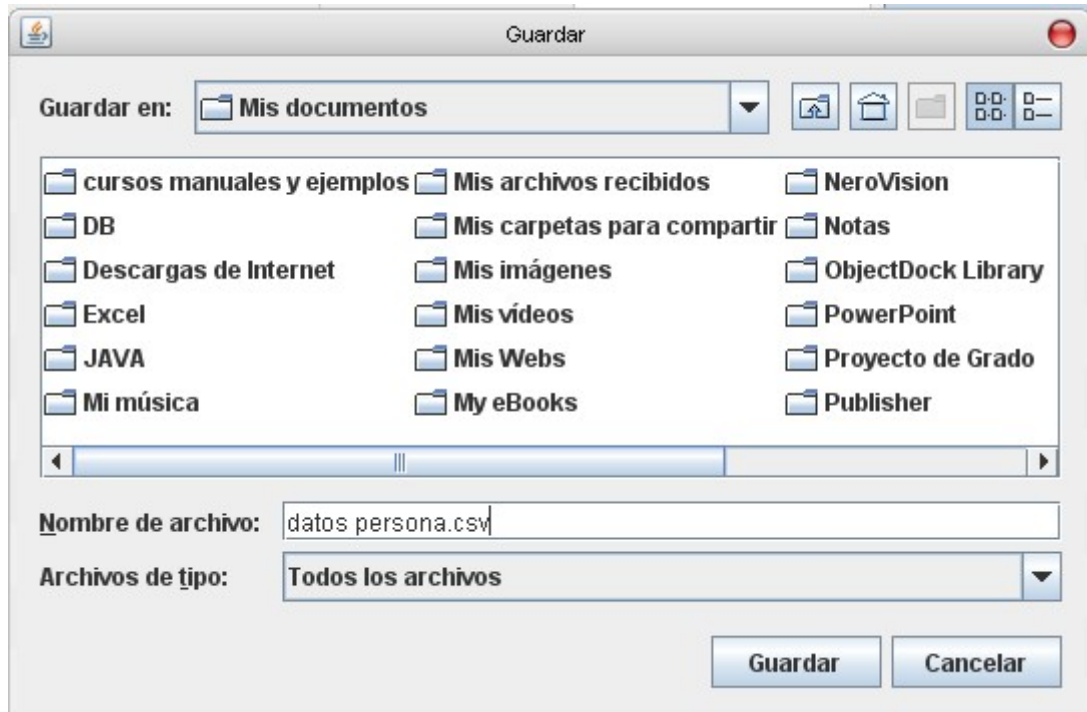
Cuando tenemos los resultados de la consulta lo mejor es exportar dicha información a un archivo CSV que es compatible con la mayoría de las aplicaciones de manejo de información como *Microsoft Excel*.

Vamos a tomar como ejemplo una información de persona ya solicitada y que se muestra en el área de resultados de consulta

The screenshot shows the 'SOCRATES-EC' web application interface. At the top, there is a login section titled 'Acceso al sistema' with fields for 'Nombre de usuario' and 'Contraseña', and a 'Desconectar' button. Below this, there are three tabs: 'Generador Dinamico de Consultas', 'Administrador de consultas', and 'Resultados de Consulta'. The 'Resultados de Consulta' tab is active, displaying a table with 10 columns labeled 'Title 1' through 'Title 10'. Above the table is a button labeled 'Exportar resultados a un archivo CSV (compatible con Excel)'. The table contains four rows of data, each representing a person's record.

Title 1	Title 2	Title 3	Title 4	Title 5	Title 6	Title 7	Title 8	Title 9	Title 10
75096157	Cedula	27 years 8 ...	Caldas	Manizales	soltero	DANIEL	ANDRES	VALENCIA	VALENCIA
75096157	Cedula	27 years 8 ...	Caldas	Manizales	soltero	DANIEL	ANDRES	VALENCIA	VALENCIA
75096157	Cedula	27 years 8 ...	Caldas	Manizales	soltero	DANIEL	ANDRES	VALENCIA	VALENCIA
75096157	Cedula	27 years 8 ...	Caldas	Manizales	soltero	DANIEL	ANDRES	VALENCIA	VALENCIA

Solo damos clic en el botón “exportar resultados a un archivo CSV (compatible con Excel)” y nos debe aparecer una ventana similar a la de guardar consultas sql, solo que en esta caso guardaremos el archivo con extensión “.csv” (Ej. “datos persona.csv”)



El procedimiento es básicamente el mismo que el de guardar consultas sql.

### 3. FORMATO DE FORMULARIOS Y PREGUNTAS DE TIPO CENSAL

SOCRATES-EC: SISTEMA DE INFORMACIÓN POBLACIONAL BASADO EN  
PROCEDIMIENTOS DE CENSADO

Por:

Danko Jiménez Londoño  
Daniel Andrés Valencia V.

A continuación se muestran las preguntas de tipo censal que se formulan en el sistema de recolección/actualización de información vía Web y que son necesarias para recoger la información necesaria para la operación del sistema.

Información básica de persona:

- Numero del documento de identidad: (numero documento)
- Tipo de documento: (Cedula, Pasaporte, T. Identidad)
- Primer Apellido
- Segundo Apellido
- Primer Nombre
- Segundo Nombre
- Sexo: (Femenino, Masculino)
- ¿Cual es su fecha de Nacimiento? (dd/mm/aaaa)
- ¿Cual fue la ciudad donde nacio? (Departamento, Municipio)

Información avanzada de persona:

- Parentesco con la cabeza de hogar: (cabeza de hogar, familiar, otro).
- ¿A que raza o comunidad indígena pertenece? (raza o comunidad indígena).
- Estado civil: (soltero, casado, viudo, divorciado).
- ¿Sabe leer? (si, no)
- ¿Sabe escribir? (si, no)
- ¿Asiste actualmente a algún instituto de educación formal? (si, no)
- ¿En que nivel educativo se encuentra actualmente? (de preescolar a universitario postgrado).
- ¿Cual fue el ultimo grado que cursó? (ultimo grado cursado)
- ¿Cuales de los siguientes idiomas habla? (español, ingles, frances, italiano, alemán, portugués).
- Es aportante, cotizante o Beneficiario de... (pertenencia a sistema de salud).
- Tiene limitaciones físicas permanentes para... (limitaciones físicas)
- Actualmente padece o ha padecido de alguna enfermedad que haya requerido... (tratamientos médicos especiales).

Información de vivienda:

- Dirección: (dirección de la vivienda).
- ¿En que material esta construida principalmente la vivienda? (material base).
- ¿Que servicios públicos posee esta vivienda? (agua, teléfono, gas, electricidad).
- ¿Como eliminan principalmente la basura en esta vivienda? (administración de desechos).
- El tipo de servicio sanitario que posee la vivienda es... (tipo de baño).
- ¿El servicio sanitario de la vivienda es de uso exclusivo de los miembros de la familia? (si, no).

- ¿Posee la vivienda un lugar exclusivo para preparar alimentos (una cocina)? (si, no).
- ¿Cuántos cuartos de baño con regadera o ducha tiene la vivienda? (numero de baños).
- ¿Cuántos cuartos en total (incluidos sala y comedor) tiene la vivienda? (numero total de cuartos)
- ¿Cuántos cuartos de dormir posee la vivienda? (numero de dormitorios)
- ¿que vehículos poseen los miembros de esta vivienda? (auto, moto, bicicleta)

#### 4. CATALOGO DE ARCHIVOS FUENTE

SOCRATES-EC: SISTEMA DE INFORMACIÓN POBLACIONAL BASADO EN  
PROCEDIMIENTOS DE CENSADO

Por:

Danko Jiménez Londoño  
Daniel Andrés Valencia V.

#### 4.1 ESQUEMA BÁSICO DE CONSTRUCCION DEL SISTEMA

El sistema de información poblacional basado en procedimientos de censo Sócrates-EC esta construido en 2 módulos especialmente diseñados para que cumplan un puposito especifico y bajo 2 modelos de aplicaron de software completamente diferentes pero que interactúan entre si de una manera eficiente y segura para poder cumplir con el propósito por el que fue creado este sistema.

Se presenta en su modelo web bajo el concepto de construcción dinámica lo que quiere decir que sus acciones, su interfaz, sus modos de navegación y operación son construidos directamente desde la base de datos.

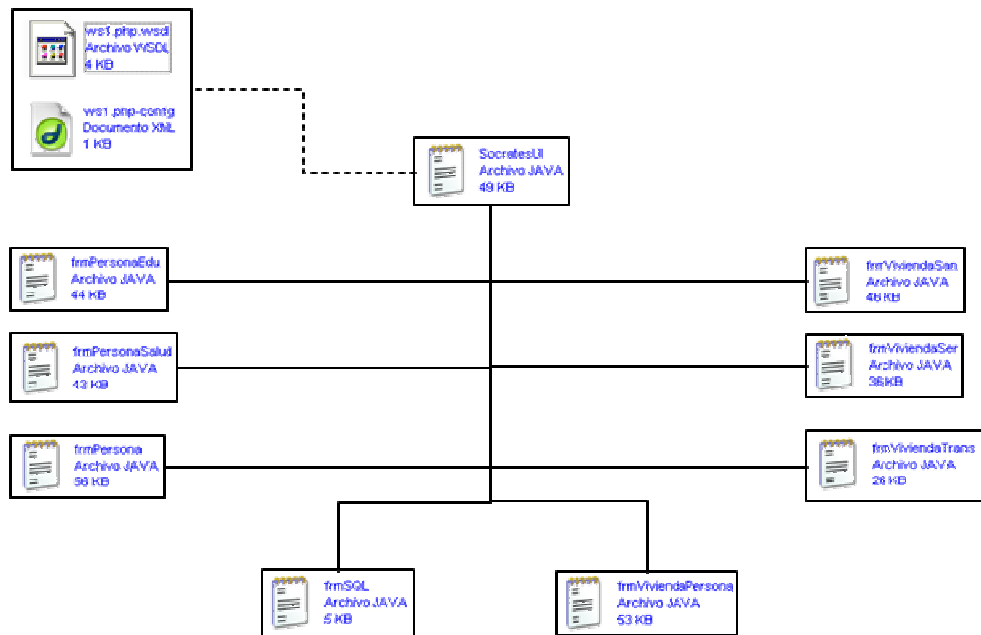
Mientras que su modelo cliente – servidor esta construido como una aplicación independiente que se conecta a la base de datos a través de la tecnología de servicios Web, permitiendo portabilidad de plataformas y facilidad de uso.



## Sistema de recolección/actualización vía Web + servidor servicio Web:

Sitio - SocratesWeb (C:\Documents and Setti...		Carpeta
[-]	acc_calendar	Carpeta
	acc_calendar.css	4KB Hoja de estilos en cascada
	acc_calendar.gif	1KB Imagen GIF
	acc_calendar.js	33KB JScript Script File
	acc_calendar.png	1KB Imagen PNG
	loading.gif	6KB Imagen GIF
	logo.png	23KB Imagen PNG
	riquelao.png	2KB Imagen PNG
[-]	Imagenes	Carpeta
	actualizar.gif	3KB Imagen GIF
	banner1.gif	18KB Imagen GIF
	banner4.gif	14KB Imagen GIF
	loading.gif	3KB Imagen GIF
	modpersona.gif	12KB Imagen GIF
	modvivienda.gif	12KB Imagen GIF
	nuevapersona.gif	14KB Imagen GIF
	nuevavivienda.gif	11KB Imagen GIF
	nuevo.gif	3KB Imagen GIF
	separador1.gif	1KB Imagen GIF
	separador2.gif	1KB Imagen GIF
	separador3.gif	1KB Imagen GIF
	separador4.gif	1KB Imagen GIF
[-]	js	Carpeta
	prototype.js	57KB JScript Script File
[-]	lib	Carpeta
	changelog	35KB Archivo
	class.nusoap_base.php	30KB Archivo PHP
	class.soap_fault.php	3KB Archivo PHP
	class.soap_parser.php	25KB Archivo PHP
	class.soap_server.php	38KB Archivo PHP
	class.soap_transport_http.php	45KB Archivo PHP
	class.soap_val.php	3KB Archivo PHP
	class.soapclient.php	33KB Archivo PHP
	class.wsdl.php	77KB Archivo PHP
	class.wsdlcache.php	6KB Archivo PHP
	class.xmlschema.php	34KB Archivo PHP
	nusoap.php	283KB Archivo PHP
	nusoapmime.php	16KB Archivo PHP
	acceso.php	14KB Archivo PHP
	analisis.php	3KB Archivo PHP
	consulta.php	1KB Archivo PHP
	index.php	3KB Archivo PHP
	info.php	1KB Archivo PHP
	info_persona.php	10KB Archivo PHP
	info_persona_adv.php	11KB Archivo PHP
	info_vivienda.php	10KB Archivo PHP
	login.php	3KB Archivo PHP
	lstMunicipios.php	1KB Archivo PHP
	recolector.php	6KB Archivo PHP
	sesiones.php	7KB Archivo PHP
	ws1.php	4KB Archivo PHP

## Sistema de administración de consultas (cliente – servidor):



## 4.2 ARCHIVOS QUE COMPONEN EL SISTEMA DE RECOLECCIÓN/ACTUALIZACIÓN DE DATOS VÍA WEB

### Index.php

Web de presentación del sistema, inicia la sesión de seguridad y ofrece las opciones de ingresar nueva información a la base de datos o modificar información existente

### Login.php

Web de ingreso al sistema para modificar datos de la vivienda y de los habitantes que viven en ella, para incrementar el nivel de seguridad, se preguntan unos datos básicos de la persona para permitir el acceso; estos datos son: Nombres y apellidos, documento de identidad y tipo de documento. Si estos datos ya están en base de datos se permitirá el ingreso a la persona para que haga las modificaciones respectivas.

### Acceso.php

Web que presenta las opciones de adición/modificación de datos y controla el orden de presentación, validación, y almacenamiento de la información en base de datos.

### Info\_persona.php

Web que permite la adición/modificación de los datos básicos de la persona.

### LstMunicipios.php

Web que complementa a info\_persona.php en el área del lugar de nacimiento, mas específicamente la ciudad donde nacio, para que esta página funcione, requiere el código javascript contenido en la carpeta "js".

### Acc\_calendar y js:

Carpetas que contienen código javascript necesario para el funcionamiento correcto de la Web info\_persona.php; acc\_calendar contiene un calendario que facilita la escogencia de la fecha de nacimiento de la persona, mientras que js da soporte para la selección dinamica del lugar de nacimiento de la persona (departamento – municipio).

### Info\_persona\_adv.php

Web que permite la adición/modificación de datos adicionales de la persona, tales como información de salud, educación y cultura. Para poder utilizar esta Web es necesario pasar primero por info\_persona.php ya que la consistencia de los datos a manipular lo requiere.

#### Info\_vivienda.php

Web que permite la adición/modificación de datos de vivienda, tales como material principal de la construcción, servicios públicos con los que cuenta, entre otros.

#### analisis.php

Librería Web que es requerida por info\_vivienda.php e incluye la clase dirección que permite parametrizar la dirección de la vivienda y traducirla a valores codificados y separados para permitirle al sistema de administración de consultas hacer solicitudes por búsqueda de direcciones y a futuro geo-referenciar la información almacenada en base de datos.

#### consulta.php

Librería Web que contiene la clase consulta (bdconsulta), esta permite la conexión y transferencia de información con la base de datos, esta librería es esencial para todo el sistema, incluido el servicio Web, ya que es a través de ella que se permite el acceso a la base de datos. Adicionalmente contiene la información de conexión a la base de datos.

#### Sesiones.php

Librería Web que administra la información de sesión del sistema de recolección/actualización de datos vía Web, gracias a esta librería puede mantenerse constancia de la información modificada o a añadir a la base de datos.

#### Recolector.php

Librería Web que permite la carga de la información de persona/vivienda de la base de datos para su visualización y modificación. Adicionalmente permite el almacenamiento ordenado de la información nueva/modificada en la base de datos.

#### Ws1.php

Servicio Web. Permite la conexión y la transferencia de datos entre el sistema de administración de consultas java y la base de datos. Para su correcto funcionamiento es requerido el kit de *Web services* nuSOAP contenido en la carpeta "lib" y la librería Web consulta.php.

### 4.3 ARCHIVOS QUE COMPONEN EL SISTEMA DE ADMINISTRACIÓN DE CONSULTAS (CLIENTE - SERVIDOR)

Ws1.php.wsdl y ws1.php-config

Archivos de descripción de funciones del servicio Web. Le permiten al administrador de consultas saber como se transfiere la información desde y hasta el servicio Web.

SocratesUI.java

Clase principal del sistema. Se encarga de llamar los generadores para su uso, hacer la conexión y transferencia de datos con el servicio Web, cargar y guardar consultas sql y exportar los resultados de la consulta en un formato compatible con otros sistemas (CSV).

frmPersona.java

Generador de consultas sql para datos básicos de persona, la consulta generada es entregada a socratesUI para su utilización.

frmPersonaEdu.java

Generador de consultas sql para datos de persona relacionados con educación y cultura, la consulta generada es entregada a socratesUI para su utilización.

frmPersonaSalud.java

Generador de consultas sql para datos de persona relacionados con salud, la consulta generada es entregada a socratesUI para su utilización.

frmViviendaSan.java

Generador de consultas sql para datos de vivienda relacionados con sanidad, la consulta generada es entregada a socratesUI para su utilización.

frmViviendaSer.java

Generador de consultas sql para datos de vivienda relacionados con servicios públicos, la consulta generada es entregada a socratesUI para su utilización.

frmViviendaTrans.java

Generador de consultas sql para datos de vivienda relacionados con transporte (vehículos de los habitantes de la vivienda), la consulta generada es entregada a socratesUI para su utilización.

frmViviendaPersona.java

Generador especial de consultas sql para datos de vivienda relacionados con los habitantes que habitan en ella, la consulta generada es entregada a socratesUI para su utilización.

frmSQL.java

Generador avanzado de consultas, requiere conocimientos de lenguaje SQL y de la estructura interna de la base de datos para su efectivo uso.