

FACULTAD DE INGENIERÍA

DIVISIÓN DE ESTUDIOS PROFESIONALES PARA EJECUTIVOS

CARRERA DE INGENIERÍA DE SISTEMAS

SISTEMA DE GESTIÓN PARA VETERINARIA

PROYECTO PROFESIONAL PRESENTADO POR GOSU’S:

|  |  |
| --- | --- |
| Elianni Cordova Marttini | U920863 |
| Renzo Huertas Navarro | U201216398 |
| Omar Carpio Carpio | U201200336 |
| Renzo Portocarrero Heredia | U201200337 |
| Ricardo Campos | U800052 |
| Oscar Chafloque Tampeck | U921300 |

**URL GITHUT:**

<https://github.com/rc800052/Veterinaria-MyFriend>

**PARA EL CURSO DE DESARROLLO DE SISTEMAS DISTRIBUIDOS**

**PROFESOR:**

**ING. JOSE DIAZ**

**Lima, 08 de Diciembre de 2012**

**ÍNDICE**

1. INTRODUCCIÓN 2

2. ANTECEDENTES 2

3. OBJETIVOS DEL PROYECTO 2

4. BENEFICIOS DEL PROYECTO 2

5. PROCESOS DEL NEGOCIO 4

6. HISTORIAS DE USUARIO 4

7. ARQUITECTURA DEL SISTEMA 0

8. CONCLUSIONES 0

# INTRODUCCIÓN

Los sistemas de información siempre han apoyado en la gestión del negocio. Antes era necesario adquirir un gran hardware para poder procesar toda la información relacionada con la empresa y personal para reserva de citas. Sin embargo, hoy es posible separar el negocio para hacerlo más eficiente y rápido. Los sistemas distribuidos nos permite segmentar el sistema en partes independientes, a su vez, estos se encuentran integrados para ofrecer un mejor servicio.

Actualmente, los doctores que laboran en la clínica solo pueden ver sus citas en la oficina ubicada en la veterinaria debido a que es personal contratado. Asimismo, por la buena atención que brinda la clínica se ha incrementado las citas médicas. Todo el proceso de atención se realiza en el sistema pero no tiene alcance desde el exterior.

Ello ha generado que no se cumpla con muchas citas programadas porque los doctores no tenían las citas registradas al día. Asimismo, la veterinaria cuenta con una gama de productos que podrían ser consumidos por otras clínicas atreves de un WebService.

Atendiendo a este requerimiento, el presente proyecto plantea una solución alternativa a este tipo de trabajo.

# ANTECEDENTES

La veterinaria GOSUS cuenta con un staff de médicos internos y externos que generan una gran demanda de citas a diario en Lima. Asimismo, los productos que vende la clínica son exclusivos y tienen demanda ante otras clínicas. Esta demanda sin embargo ha generado que las citas no se hayan atendido en las fechas y horarios reservados debido a la falta de aviso y revisión por parte del medico y de la oficina. Además, los productos son de exclusividad de la veterinaria.

Por otro lado, los dueños de los pacientes solo pueden realizar sus reservas de citas en las oficinas de la veterinaria.

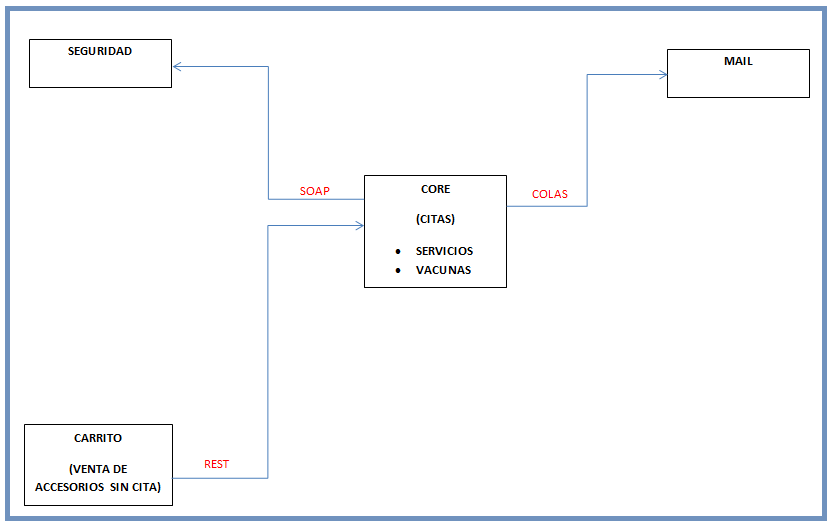
# OBJETIVOS DEL PROYECTO

El presente proyecto se enfocará en brindar una solución integral el cual consistirá en reservar citas médicas. Este proceso permitirá contar con la lista de veterinarios de los diferentes gremios. Además, el doctor podrá ingresar al sistema para revisar sus citas reservadas que le fueron confirmadas por mensajería. De igual manera, los clientes serán notificados por mensajería por cada cita registrada.

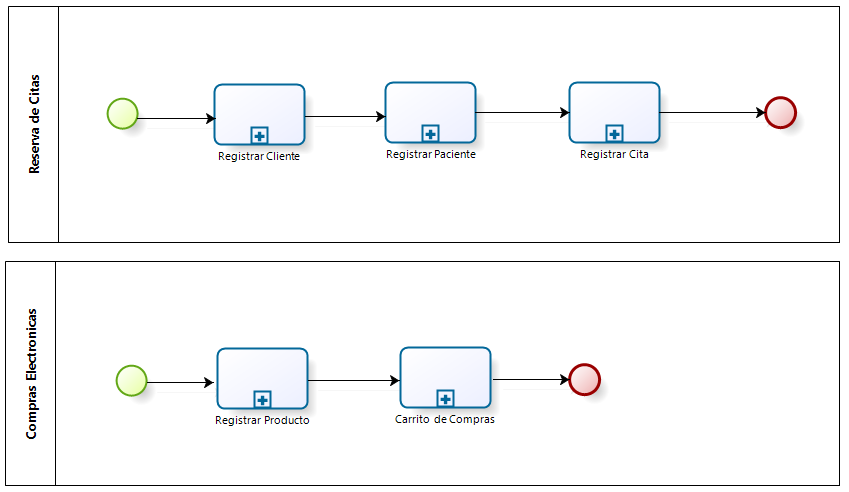
# BENEFICIOS DEL PROYECTO

Los principales beneficios son los siguientes:

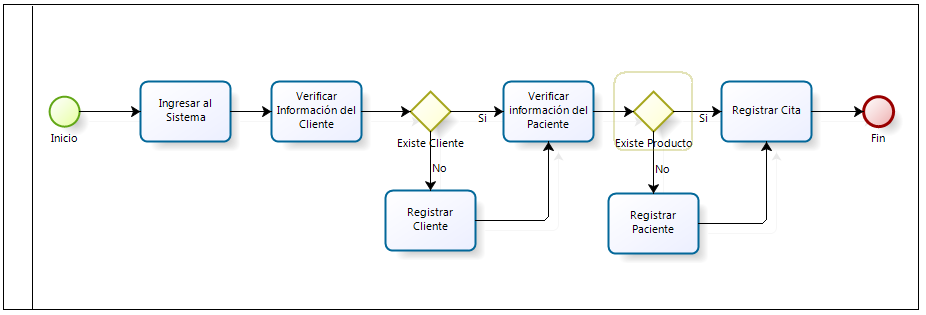
* **Bajo Acoplamiento.**- Cada servicio es independiente y es reutilizable.
* **Integración de sistemas**.- La integración de los diversos servicios proporcionará una manera más eficiente del negocio, al integrar con otros sistemas, algunos de tipo Legacía.
* **Economía -** Se realizará una gran reducción de costos ya que necesitará menos hardware y personal para funcionar igual o mejor que la actualidad.
* **Tiempo.-** El tiempo en desarrollar nuevas aplicaciones se reduce considerablemente ya que se pueden reutilizar los servicios.
* **Interoperabilidad.-** Permitirá comunicarse con múltiples plataformas a través de estándares como HTTP, SOAP y XML.
* **Escalabilidad.-** Dependiendo de la demanda del servicio, este puede crecer aumentando un clúster más, de esta manera, el sistema cambia sin tener mucho impacto sobre los demás servicios.
* **Simplificación tecnológica.-** SOA y RESTful permite usar los mismos servicios para una gran cantidad de procesos de la clínica, simplificando las operaciones.
* **Flexibilidad y adaptación**.- Esta plataforma nos permitirá integrar fácilmente cualquier otro aplicativo en el futuro, gracias a su alto poder de interoperabilidad.



# PROCESOS DEL NEGOCIO



**Detalle del Proceso Registrar Citas**



# HISTORIAS DE USUARIO

## ****HU01. Login del Sistema (Doctores)****

Como usuario del sistema luego de haber ingresado mi usuario y contraseña deseo realizar lo siguiente:

* Registrar doctores
* Listar los doctores
* Consultar citas

Los siguientes requerimientos aplican para el ingreso al sistema:

|  |  |
| --- | --- |
| Cuando:  El doctor recibe el correo (Queue), ingresará su usuario y contraseña para acceder al sistemas. (Servicio SOA).  El doctor visualizará sus citas.  El doctor con perfil administrador elimina registros | Resultado esperado de las acciones:   * Confirmación de la acción. * Mensajes de error. * Validar que el registro ya exista para no duplicar registros. * Impedir eliminar doctores que tengan registrado citas |

El servicio de AUTENTIFICACIÓN validando por una cuenta de usuario.

App Web debe grabar la información en la BD SQL.

Las aplicaciones se desarrollarán en .NET, usando servicios SOAP.

**HU02. Dueños (Clientes).-** Como cliente de la veterinaria, luego de haber ingresado con mi código de usuario y clave, deseo poder hacer lo siguiente:

* Registrar mis citas médicas
* Registrar y eliminar mis Pacientes
* Listar mis citas pendientes

Los siguientes requerimientos aplican para el registro del Dueño:

|  |  |
| --- | --- |
| Cuando el cliente accede al sistema se valida su usuario y contraseña (Servicio SOAP).  El cliente registrará su cita con los datos del paciente.  El cliente visualiza su cita registrada.  El cliente accesa al carrito de compras para adquirir productos. (Servicio REST) | Resultado esperado de las acciones:   * Confirmación de la acción. * Mensajes de error. * Validar que el registro ya exista para no duplicar registros. |

Cuando se muestra la información específica del dueño, se desea además de mostrar su información personal tener la siguiente información:

* Notas sobre el dueño
* La lista de pacientes que tiene registrado el Dueño

El servicio de AUTENTIFICACIÓN validando por una cuenta de usuario.

App Web debe grabar la información en la BD MySQL.

Las aplicaciones se desarrollarán en JAVA, usando servicios REST.

**HU03. Administrador de Pacientes.-** Como usuario del sistema, luego de haber ingresado con mi código de usuario y clave, deseo poder hacer lo siguiente:

* Listar pacientes registrados previamente
* Mostrar la información específica de un paciente
* Registrar información básica del paciente
* Modificar información básica del paciente
* Eliminar pacientes

La información básica del paciente es la siguiente:

|  |  |
| --- | --- |
| Datos obligatorios del paciente:   * Dueño al que pertenece * Especie * Raza * Genero * Tipo de Sangre * Esterilizado (Si / No) * Tamaño (Pequeño/Mediano/Grande) * Actividad(Baja/Media/Alta) * Peso(En Kg) * Fecha Nacimiento | Resultado esperado de las acciones:   * Confirmación de la acción. * Mensajes de error. * Validar que el registro ya exista para no duplicar registros. |

Las aplicaciones se desarrollarán en JAVA y se graban en MySQL.

**HU04. Búsqueda de Pacientes / Clientes.-** Como usuario del sistema, luego de haber ingresado con mi código de usuario y clave, deseo poder hacer lo siguiente:

|  |  |
| --- | --- |
| Listar los pacientes y clientes inscritos.  Listar los clientes que tienen citas pendientes de atención | Resultado esperado de las acciones:   * Confirmación de la acción. * Mensajes de error. * Validar existencia de paciente * Validar existencia de cliente |

Las aplicaciones se desarrollarán en JAVA y se graban en MySQL.

**HU05. Carrito de Compras.-** Como cliente, me permite realizar lo siguiente:

* Listar los productos de Gozus.
* Seleccionar productos del catalogo.
* Pagar mi pedido.

|  |  |
| --- | --- |
| Cuando un cliente desea realizar un pedido virtual, accede a la Web de Gozus para seleccionar los productos que desea. Luego, el sistema los direcciona a un servicio de pago.  Los productos son publicados por la veterinaria. (Servicio REST)  El servicio es consumido por la página comercial de Gozus. | Resultado esperado de las acciones:   * Envía el pedido a un servicio de terceros que gestionar el cobro y los datos de envío entre Gozus y el cliente. |

El servicio de CARRITO DE COMPRAS lista los productos publicados desde la intranet Gozus y envía el pedido a Paypal.

La página Web del Carrito de Compras se desarrollará en PHP, usando servicios REST en Java.

**HU06. Citas.-** Como usuario del sistema, luego de haber ingresado con mi código de usuario y clave, deseo poder hacer lo siguiente:

* Registrar citas
* Listar citas en estado pendiente
* Envía cita a cola

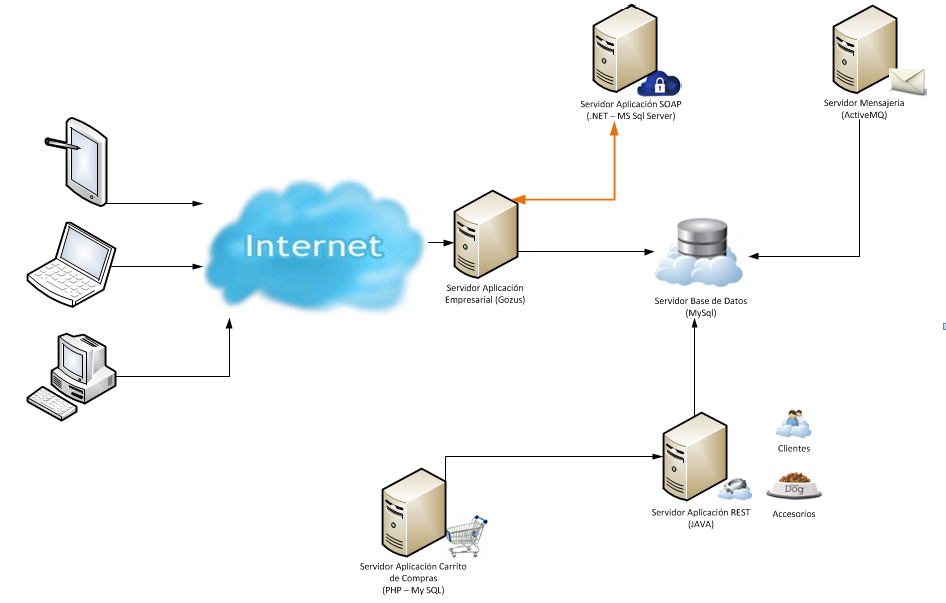
|  |  |
| --- | --- |
| Cuando las citas por servicio son registradas por el doctor están generan mensajería.  Los clientes reciben una notificación de mensajería por cada cita registrada. (Queue)  Los doctores reciben una notificación de mensajería por cada cita reservada del cliente. (Queue) | Resultado esperado de las acciones:   * Confirmación de la acción. * Mensajes de error. * Validar que la mensajería recepcionada. |

El servicio de REGISTRO DE CITAS validando por una cuenta de usuario.

App Web debe grabar la información en la BD MySQL.

Las aplicaciones se desarrollarán en JAVA, usando servicios QUEUE (Active MQ).

# ARQUITECTURA DEL SISTEMA



# CONCLUSIONES

* El curso nos ha permito evaluar y desarrollar un proyecto complejo y real permitiéndonos poner en práctica los objetivos del curso.
* Crear aplicaciones distribuidas usando web Services, son más fáciles, rápidas y poseen un bajo costo al momento de implementarse.
* El uso de la tecnología SOA nos permite interactuar con otros sistemas con el mínimo esfuerzo posible y optimizando los recursos económicos.
* Ss
* Dddd
* El trabajo y el esfuerzo de todo el equipo es indispensable para concluir de manera eficiente este sistema y desplegarlo a producción.
* El desarrollo ágil nos permite trabajar en conjunto y de manera colaborativa
* El uso de repositorio git común, en línea, nos permite obtener rápidamente los avances de todos los integrantes del grupo, de manera que todos apuntamos a un objetivo común.