

[Hospital Regional Cajamarca]

Block de Notas de la Arquitectura

[Sistema de Gestión de Información]

Cristopher Ramón Quiroz Sánchez
[Fecha]

[Sistema de Gestión de Información]	
Block de Notas de la Arquitectura	Fecha: <dd/mm/yy>

Contenido

Historia de revisiones	2
1. Propósito.....	3
2. Objetivos y principios de la Arquitectura	3
3. Supuestos y dependencias	3
4. Requisitos importantes para la arquitectura	4
5. Decisiones, limitaciones, y justificaciones	5
6. Componentes de la Arquitectura.....	7
6.1 Componente de Arquitectura 1	¡Error! Marcador no definido.
6.2 Componente de Arquitectura 2	¡Error! Marcador no definido.
7. Abstracciones Claves	7
8. Capas o marco arquitectónico	9
9. Vistas de la Arquitectura.....	10

[Sistema de Gestión de Información]	
Block de Notas de la Arquitectura	Fecha: <dd/mm/yy>

Historia de revisiones

Fecha	Versión	Descripción	Autor
[04/10/2014]	[0.1]	Elaboración del documento	Cristopher Ramón Quiroz Sánchez

[Sistema de Gestión de Información]	
Block de Notas de la Arquitectura	Fecha: <dd/mm/yy>

<Nombre del Proyecto> Block de Notas de la Arquitectura

1. Propósito

Este documento proporciona una descripción de la arquitectura del sistema, haciendo uso de diversas visiones arquitectónicas para representar diversos aspectos del sistema y explicando el diseño lógico y físico del mismo.

Se presenta un sistema de gestión de información en el área de CRED e inmunizaciones, que concierne al registro de paciente, registro de aplicación de vacunas, resultados de descarte de anemia, resultados de descarte de parásitos, CRED y evaluación psicomotriz.

2. Objetivos y principios de la Arquitectura

Este documento apoya el desarrollo de una visión de la arquitectura del sistema explorando y evaluando las opciones de alto nivel, propicia que se llegue a un acuerdo con el jefe del proyecto y otros involucrados en el proyecto.

En relación a la estructura del sistema refleja las decisiones y suposiciones de trabajo en relación a la parte física y lógica a los requerimientos no funcionales del sistema.

La arquitectura está representada usando Star UML, esto permite visualizar, entender, y razonar sobre los elementos significativos e identificar los riesgos, en este documento veremos la forma de comunicar el modelo UML en contexto, presentando la información.

Principio/Política	La arquitectura de software es de especial importancia ya que la manera en que se estructura un sistema tiene un impacto directo sobre la capacidad de este para satisfacer lo que se conoce como los atributos de calidad del sistema
Descripción	Los atributos de calidad son el desempeño, que tiene que ver con el tiempo de respuesta del sistema a las peticiones que se le hacen, la usabilidad, que tiene que ver con qué tan sencillo les resulta a los usuarios realizar operaciones con el sistema. Los atributos de calidad son parte de los requerimientos (no funcionales) del sistema y son características que deben expresarse de forma cuantitativa.
Motivación	La manera en que se estructura un sistema permitirá que se satisfagan los atributos de calidad
Implicaciones	Los diseños arquitectónicos que se crean en un proyecto pueden ser reutilizados para crear sistemas distintos.

3. Supuestos y dependencias

El desarrollo del sistema tiene como objetivos proporcionar los siguientes.

- ✓ El sistema será desplegado en los servidores del Hospital Regional Cajamarca.
- ✓ Las postas de salud tendrán los recursos necesarios para acceder al sistema.
- ✓ El sistema está en Web.

[Sistema de Gestión de Información]	
Block de Notas de la Arquitectura	Fecha: <dd/mm/yy>

- ✓ El sistema está disponible para el grupo de usuarios que lo utilice.
- ✓ El sistema puede ser reparado en un tiempo razonable.
- ✓ El sistema puede ser levantado después de una falla.
- ✓ El sistema tiene un manejo práctico, entendible, amigable y rápido.
- ✓ El sistema es de gran practicidad y sencillez.
- ✓ El sistema tiene una facilidad de uso en relación a la personas que lo van a utilizar.

4. **Requisitos importantes para la arquitectura**

Comentado [SMA1]: Hacer referencia al documento de requisitos de soporte.

A continuación se muestran los requerimientos no funcionales con su descripción:

Requisitos de Seguridad:

La seguridad de un sistema no solo tiene en cuenta la seguridad del sistema sino, además, el ambiente en el que se usará el sistema. Por lo que se tiene que contemplar la seguridad física del lugar donde se usa la aplicación, los controles administrativos que se establecen de acceso al sistema y las regulaciones legales que afecta o determina el uso del sistema y que serán tenidas en cuenta si se incumple. La seguridad puede ser tratada en tres aspectos diferentes:

Confidencialidad: La información manejada por el sistema está protegida de acceso no autorizado y divulgación.

Integridad: La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes, de la misma forma será considerada igual a la fuente o autoridad de los datos. Pueden incluir también mecanismos de chequeo de integridad y realización de auditorías.

Disponibilidad: Significa que los usuarios autorizados se les garantizará el acceso a la información y que los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.

Requisitos de Rendimiento y Escalabilidad:

Se debe de tener en cuenta el requisito de Rendimiento y Escalabilidad porque convendría que los usuarios examinaran con detenimiento hasta qué punto el proyecto se ajusta a sus expectativas en cuanto a los tiempos de respuesta y es capaz de prestar servicio adecuadamente de acuerdo al tipo y tamaño para el que ha sido concebido.

Requisito de Disponibilidad:

las instituciones o centros de trabajos utilizan los sistemas informáticos, conectados a la red. Un cambio fundamental lo constituirá el aumento espectacular de la dependencia de los usuarios con respecto a la red, de modo que si el software dejara de estar disponible, a los usuarios les será imposible continuar su trabajo. Por consiguiente, convendría que los usuarios que se disponen a contratar un sistema identifiquen las necesidades de sus usuarios en cuanto a disponibilidad y las detallen en el momento de la contratación.

Requisito de Fiabilidad:

Uno de los más importantes por parte del usuario es la Fiabilidad debido a que debe de tener en cuenta la recuperación frente a fallos de conexión: asegurar que no se pierdan los datos del perfil definido por el usuario. Esto incluye no perderlos en el envío al servidor o en el envío a otras máquinas, como no perderlos si hay un fallo de conexión entre el receptor del usuario y el servidor.

[Sistema de Gestión de Información]	
Block de Notas de la Arquitectura	Fecha: <dd/mm/yy>

La recuperación frente a fallos del sistema: posibilidad de reiniciar el sistema. Verificar la Fiabilidad en la autenticación de los usuarios y la posibilidad de dar marcha atrás en la definición del perfil de cada usuario.

Orientados al desarrollador:

Es semejante a la clasificación Orientado al usuario, surge por las mismas razones, para unificar aquellas clasificaciones con las que trabaja directamente el desarrollador como: Disponibilidad, Portabilidad, Adaptabilidad, Testabilidad, Comprensibilidad.

Requisito de Facilidad de Uso:

La definición de este requisito es una cuestión de especial importancia, pues un proyecto puede fracasar porque sus usuarios no encuentren sencillo su uso, porque no exista una interfaz sencilla y atractiva, o un manual que describa el funcionamiento y el uso del sistema al usuario final.

Requisito de Soporte:

Le brinda al usuario la facilidad de instalación, facilidad de mantenimiento, lo que requiere código y diseño documentado y facilidad de actualización hacia versiones más moderna.

Requisitos de Software:

Debe mencionarse el software del que se debe disponer, después de implementado el sistema.

Requisitos de Hardware:

Se deben enunciar los elementos de hardware que se necesitan para que el software cumpla sus funcionalidades.

Restricciones en el diseño y la implementación:

Especifica o restringe la codificación o construcción de un sistema, son restricciones que han sido ordenadas y deben ser cumplidas estrictamente.

Requisitos de apariencia o interfaz externa:

Este tipo de requisito describe la apariencia del producto. Es importante destacar que no se trata del diseño de la interfaz en detalle sino que especifican cómo se pretende que sea la interfaz externa del producto. También pueden ser necesidades de cumplir con normas estándares, o con los estándares de la empresa para la cual se esté desarrollando el software.

Requisitos de Usabilidad:

Describen los niveles apropiados de usabilidad, dados los usuarios finales del producto, para ello deben revisarse las especificaciones de los perfiles de usuarios y las clasificaciones de sus niveles de experiencia.

5. Decisiones, limitaciones, y justificaciones

- ✓ Restricciones de Tiempo.
- ✓ Restricciones de Seguridad y Privacidad
- ✓ Restricciones de Tecnología.

[Sistema de Gestión de Información]	
Block de Notas de la Arquitectura	Fecha: <dd/mm/yy>

Limitación Arquitectónica	Tiempo
Código y Nombre	01- Restricción de tiempo
Declaración del Problema	Existe una fecha de entrega a la cual el proyecto debe estar terminada
Supuestos	Terminar a tiempo para la entrega final
Motivación	Aprender nuevas herramientas para el desarrollo de software
Alternativas	Aprovechar el tiempo para la ejecución del proyecto
Decisión	Realizar los entregables en el tiempo establecido
Estatus	En proceso
Justificación	El manejo de mucha información en poco tiempo genera demoras
Implicaciones	Entregables a tiempo

Limitación Arquitectónica	Seguridad y Privacidad
Código y Nombre	02 – Restricciones de Seguridad y Privacidad
Declaración del Problema	Prestar toda la seguridad a los usuarios
Supuestos	Poder acceder al sistema.
Motivación	Aprender nuevas herramientas para el desarrollo de software
Alternativas	Aprovechar de la fortaleza de la seguridad para los datos del usuario
Decisión	Realizar los entregables en el tiempo establecido
Estatus	En proceso
Justificación	La protección de la seguridad de la información de los datos
Implicaciones	Entregables a tiempo

Limitación Arquitectónica	Tecnología
Código y Nombre	03- Restricción de tecnología
Declaración del Problema	Usar la tecnología que está a nuestro alcance
Supuestos	A través de la tecnología mejorar los procesos
Motivación	Aprender nuevas herramientas para el desarrollo de software

[Sistema de Gestión de Información]	
Block de Notas de la Arquitectura	Fecha: <dd/mm/yy>

Alternativas	Aprovechar la tecnología para la ejecución del proyecto
Decisión	Realizar los entregables en el tiempo establecido
Estatus	En proceso
Justificación	Mejorar el proceso con la tecnología que tenemos
Implicaciones	Entregables a tiempo

6. Componentes de la Arquitectura

La arquitectura está representada siguiendo las recomendaciones de RUP, Las vistas necesarias se enumeran a continuación.

Vista	Elemento de Modelado	Descripción
Casos de Uso	Diagrama	Describe el proceso de negocio más significativo y el modelo del dominio. Presenta los actores y los casos de uso para el sistema.
Vista de Restricciones		Describe restricciones tecnológicas, normativas, uso de estándares, entre otros, las cuales deben ser respetadas tanto por el proceso de desarrollo como por el producto desarrollado
Lógica	Diagrama de Clases	Describe la arquitectura del sistema presentando varios niveles de refinamiento. Indica los módulos lógicos principales, sus responsabilidades y dependencias.
Vista de Procesos		Describe los procesos concurrentes del sistema
Implementación	Diagrama	También llamada vista de desarrollo, describe la organización estática del software.
Modelo de Datos		Presenta los modelos de datos, los servicios de persistencia.
Implantación	Diagrama	También conocida como vista física y se encarga de modelar la disposición física del software.

7. Abstracciones Claves

- **Arquitectura de software:**

Vista abstracta de alto nivel del sistema, incluyen los componentes principales con conducta que interactúan para alcanzar la misión del sistema, es un artefacto primario, conceptualizar, gerencia y construir la evaluación del sistema en desarrollo.

- **Modelo vista controlar (MVC):**

Se forma en un modelo, varias vistas, varios controladores entonces las vistas y los controladores están muy relacionados, los controladores tratan los

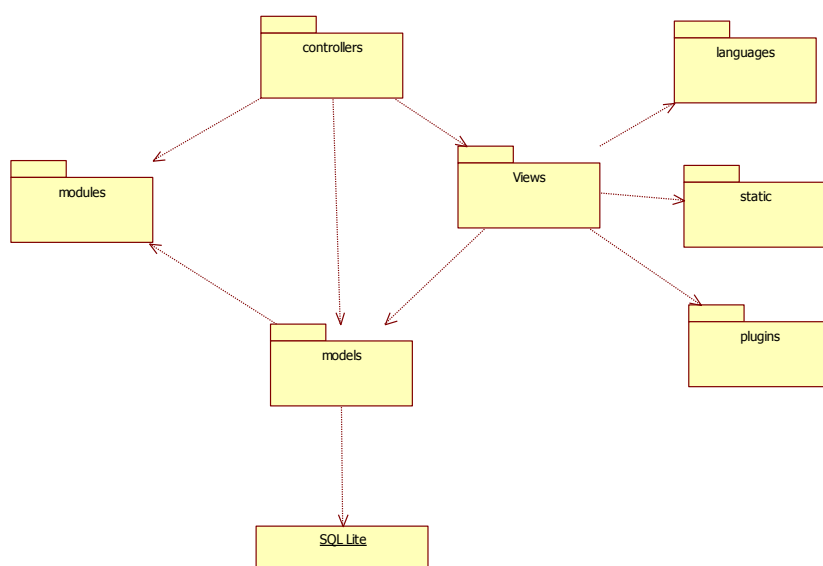
[Sistema de Gestión de Información]	
Block de Notas de la Arquitectura	Fecha: <dd/mm/yy>

eventos que se producen en la interface gráfica.

- **Web2py:**
El framework Web2py es una herramienta que está relacionada directamente a Python, web2py lo que significa que contiene todos los componentes necesarios para construir aplicaciones web totalmente funcionales
- **SqlDesigner:**
Esta herramienta es muy útil en la elaboración y diagramado de base de datos para posterior poder usar los scripts.
- **Sqllite:**
Es un sistema de gestión de base de datos, que se enlaza con el programa pasando a ser parte integral.
- **Casos de uso:**
Describe el proceso de negocio más significativo y el modelo del dominio. Presenta los actores y los casos de uso para el sistema
- **Vista de restricciones:**
Describe restricciones tecnológicas, normativas, uso de estándares, entre otros, las cuales deben ser respetadas tanto por el proceso de desarrollo como por el producto desarrollado
- **Clase:**
Describe la arquitectura del sistema presentando varios niveles de refinamiento. Indica los módulos lógicos principales, sus responsabilidades y dependencias
- **Vista de procesos:**
Describe los procesos concurrentes del sistema
- **Implementación:**
Actividad donde las especificaciones son llevadas a código de computadora.
- **Modelo de datos:**
Presenta los modelos de datos, los servicios de persistencia
- **Implantación:**
También conocida como vista física y se encarga de modelar la disposición física del software.

[Sistema de Gestión de Información]	
Block de Notas de la Arquitectura	Fecha: <dd/mm/yy>

8. Capas o marco arquitectónico



Este documento muestra de manera general, las decisiones arquitectónicas del desarrollo del sistema, utilizando modelos y estándares para la representación del mismo, se muestran y detallan las vistas a desarrollar, incluyendo sus diagramas y modelos necesarios para una mejor y mayor comprensión de la arquitectura, se incluye vista de casos de uso, con una descripción completa de cada uno de ellos y su respectivos diagramas., luego la vista lógica donde se muestran las principales clases que modela al sistema y las relaciones que las unen.

Cada aplicación de web2py consiste en un grupo de archivos, que caen en las siguientes categorías:

- ✓ **Modelos:** describen la representación de los datos(BD).
- ✓ **controladores:** describen el workflow o flujo de trabajo de la aplicación.
- ✓ **Vistas:** describen la presentación de los datos.
- ✓ **Idiomas:** describen como se debe traducir la presentación de la aplicación a otros idiomas.
- ✓ **Módulos:** módulos de Python que pertenecen a la aplicación.
- ✓ **Archivos estáticos:** imágenes estáticas, archivos CSS, archivos JavaScript, etc.

[Sistema de Gestión de Información]	
Block de Notas de la Arquitectura	Fecha: <dd/mm/yy>

- ✓ **Plugin:** grupos de archivos diseñados para trabajar en conjunto.

Todo está prolijamente organizado según el patrón de diseño Modelo-Vista-Controlador. Cada sección en la página *editar* corresponde a una subcarpeta en la carpeta de la aplicación.

Nótese que haciendo clic en los encabezados de sección las secciones se contraen o expanden. Los nombres de las carpetas bajo archivos estáticos también tienen esta propiedad.

9. Vistas de la Arquitectura

A continuación se muestran las vistas en total en la representación arquitectónica del sistema.

- ✓ **Vistas de Casos de Uso.**

Esta vista presenta la percepción que tiene el usuario de las funcionalidades del sistema. Se presenta el proceso de negocio más importante del sistema y los casos de uso que se derivan de éste.

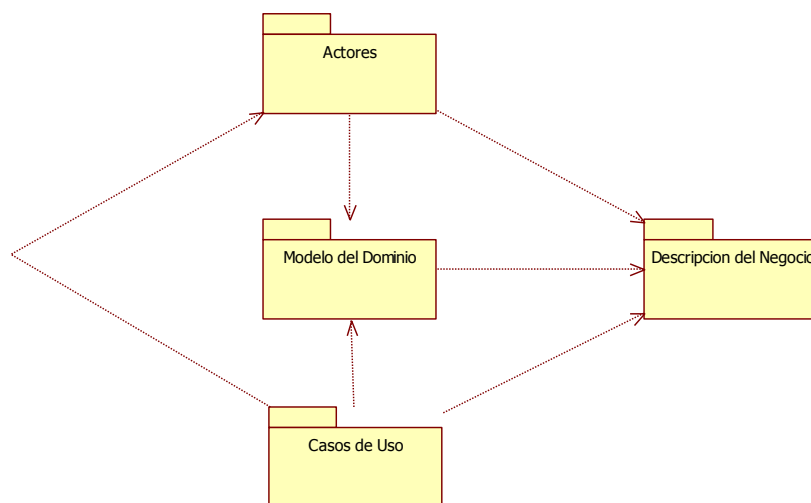


Diagrama Vista General

DESCRIPCION DEL NEGOCIO:

A continuación se describe el sistema de gestión de información – SIG Vacunas, procesos del negocio y subprocesos que se relaciona.

SISTEMA DE GESTION DE INFORAMCION – SIG Vacunas

[Sistema de Gestión de Información]	
Block de Notas de la Arquitectura	Fecha: <dd/mm/yy>

Dentro de la institución Hospital Regional Cajamarca especialmente en el área de CRED e INMUNIZACIONES, esta área que no es independiente a los procesos de la institución, no presenta un sistema que ayude su día a día, generando muchos contratiempos y conflictos en la labor que desempeñan las encargadas del área.

El sistema SIG-Vacunas será capaz de ayudar en las labores que se desarrollaran en relación a la atención del paciente, más aun tratándose de los menores de edad, este sistema contara con las funciones necesarias e importantes que se muestran en el carnet integral de salud.

Manejara el tema de las vacunas por paciente, se verá los registros de los resultados de descartes de anemia y parásitos, el registro del desarrollo de crecimiento del menor, la evaluación psicomotriz del paciente y ver la cartilla general del paciente.

SIG-Vacunas trabaja con Python y el framework web2py, ambos crean el proyecto que se está desarrollando y logrando las funcionalidades del carnet integral de salud.

PROCESO DEL NEGOCIO:

A continuación se muestran los procesos y subprocesos del negocio.

Atención General:

Este proceso es el inicio del proceso general, parte desde la llegada del paciente al establecimiento y la actividad que va a realizar.

Atención CRED:

Este proceso abarca desde que el paciente tiene la actividad de atención CRED según la edad del paciente, este recibe control de desarrollo y crecimiento, evaluación psicomotriz, análisis de algunos descartes como anemia o parásitos.

Aplicación de Vacunas:

Este proceso se da cuando el paciente tiene una fecha programada en relación a las fechas de vacunas que ya vienen establecidos, dependiendo de la capacidad de atención y la disponibilidad del turno el paciente puede ir a realizar esta actividad.

Proceso Vacunación:

Este proceso parte cuando el paciente tiene una fecha programada para una nueva vacunación (aplicación de vacunas), dependiendo del estado del paciente también puede haber cambios en este proceso y de este modo también cambiar la fecha de vacunación.

Atención por el pediatra:

En este proceso parte por la atención mensual que tiene el paciente hasta cumplir una cierta edad, la fecha es programada desde el nacimiento, las fechas de atención por el pediatra se dan a partir de la fecha de nacimiento.

SUBPROCESOS DEL NEGOCIO

Busca Historia Clínica:

[Sistema de Gestión de Información]	
Block de Notas de la Arquitectura	Fecha: <dd/mm/yy>

Este subproceso abarca la búsqueda de la historia clínica del paciente para ver comprar datos reales, ver fecha de próxima cita en relación al CRED o vacunación, para acceder a la historia clínica el paciente tiene que brindar el DNI a la enfermera y luego poder buscar en los archivos.

Carnet Integral de Salud:

En este proceso se verá el carnet integral de salud en relación a un paciente que va a ser atendido en relación a la fecha actual de atención y de acuerdo al tipo de atención que necesitara CRED o vacunación.

Revisa Datos:

En este proceso el paciente entrego el DNI a la enfermera y luego obtener la historia clínica, revisa los datos del paciente para ver que todos correspondan.

Esperar Atención:

En este proceso la paciente espera atención dependiendo de si tiene el carnet integral de salud a la mano, de no ser así a fecha que tenía programada varia en relación a otras fechas.

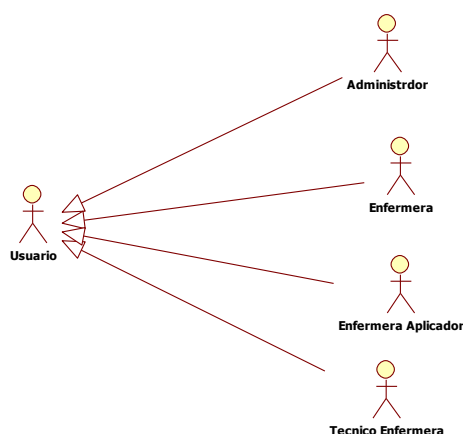
Transcripción de datos:

En este proceso la enfermera recibe las hojas de control después de las atenciones del paciente en relación a las diferentes actividades que se puedan dar (Aplicación de vacuna, resultado de destacarte de anemia, resultado de descarte de parásitos, control y desarrollo, evaluación psicomotriz)

MODELO DEL DOMINIO:

ACTORES:

Los actores del negocio en relación a como se relacionan en el negocio son:



CASOS DE USO:

[Sistema de Gestión de Información]	
Block de Notas de la Arquitectura	Fecha: <dd/mm/yy>

Contemplan el enfoque de agrupación por paquetes de StarUML, se muestra el diagrama del modelo de casos de uso.

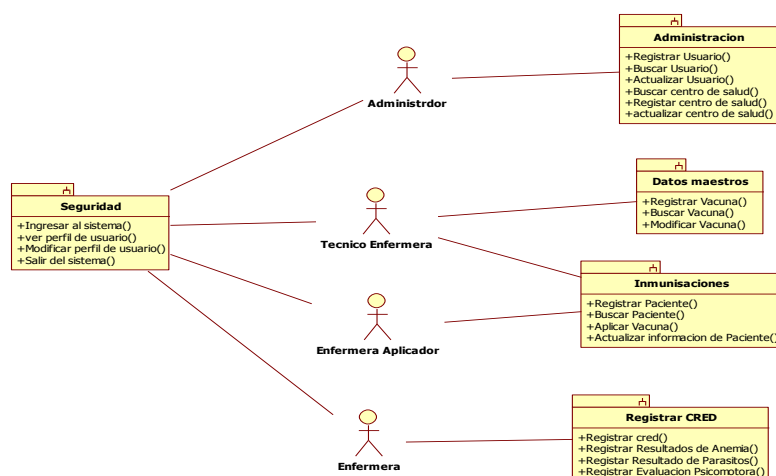


Diagrama de Casos de USO

Modelo de casos de uso:

SUB SISTEMAS

ADMINISTRACION:

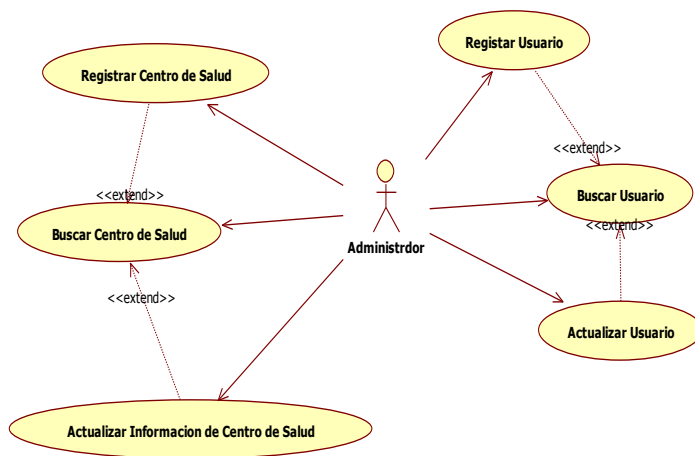


Diagrama de Administración

[Sistema de Gestión de Información]	
Block de Notas de la Arquitectura	Fecha: <dd/mm/yy>

INMUNIZACIONES:

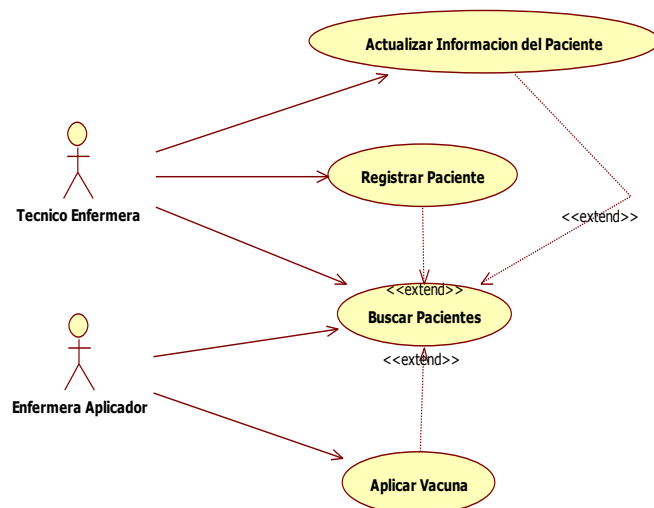


Diagrama de Inmunizaciones

CRED:

[Sistema de Gestión de Información]	
Block de Notas de la Arquitectura	Fecha: <dd/mm/yy>

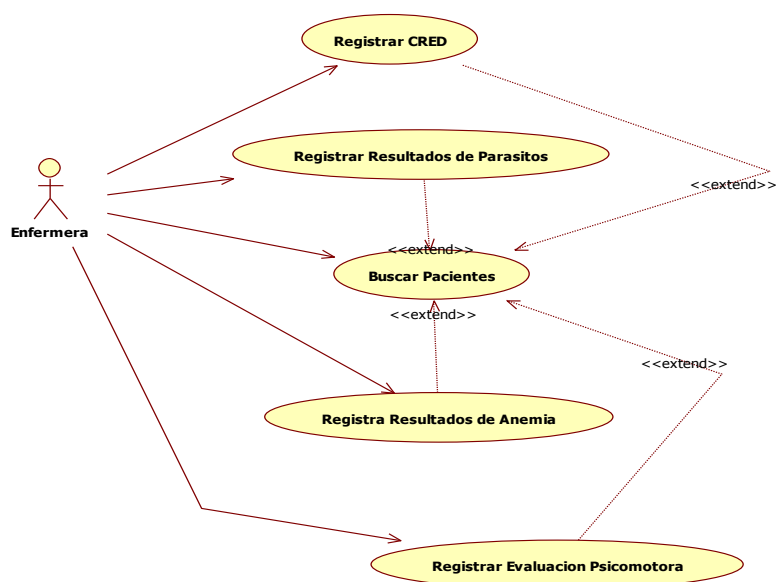


Diagrama de CRED

DATOS MAESTROS:

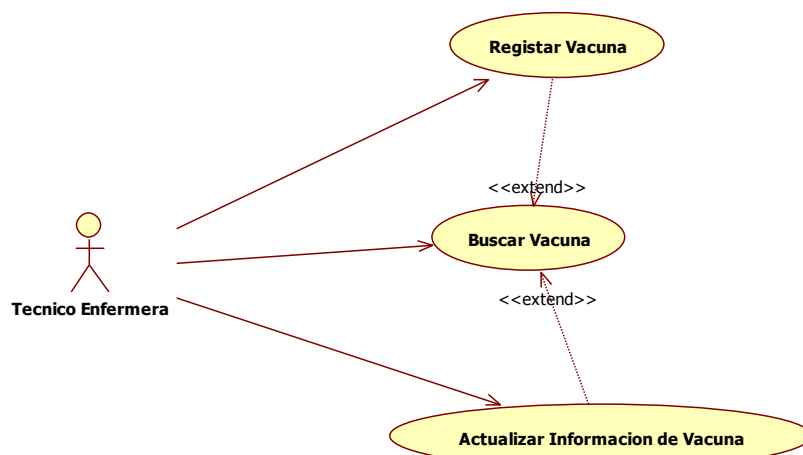


Diagrama de datos Maestros

SEGURIDAD:

[Sistema de Gestión de Información]	
Block de Notas de la Arquitectura	Fecha: <dd/mm/yy>

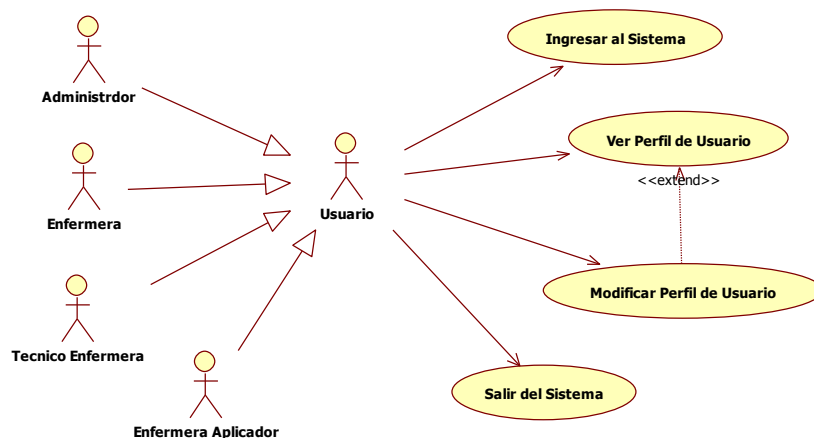


Diagrama de Seguridad

Especificación de casos de uso:

A continuación se muestran los casos de uso y una breve sinapsis del mismo:

Actualizar información del paciente:

Sinapsis: El caso de uso permite actualizar la información de un paciente en general, dependiendo de cuales sean las actualizaciones y con el consentimiento de la madre del paciente siendo estos casos (Domicilio)

Registrar Paciente:

Sinapsis: El caso de uso permite dar de alta a nuevo paciente, sea el caso de ser nuevo en el establecimiento "Hospital Regional Cajamarca" o que perteneció a otro centro de salud y a partir de la fecha se atenderá en el HRC.

Buscar paciente:

Sinapsis: El caso de uso permite buscar un paciente en general que este registrado en el sistema, se realiza la búsqueda a partir de la necesidad de aplicar una nueva vacuna, descarte de anemia, descarte de parasito, CRED, evaluación psicomotriz, o ver la cartilla.

Aplicar Vacuna:

Sinapsis: El caso de uso permite aplicar vacuna al paciente que lo solicite y este en fecha que indica su programación, esto parte de tener registrado y haber buscado al paciente a atender.

Registrar resultados de anemia:

Sinapsis: El caso de uso permite registrar el resultado de descarte de anemia, este análisis se realiza en la fecha indicada y para este caso el paciente tiene que estar registrado y hacerlo buscado para agregar registro.

[Sistema de Gestión de Información]	
Block de Notas de la Arquitectura	Fecha: <dd/mm/yy>

Registrar resultados de parásitos:

Sinapsis: El caso de uso permite registrar el resultado de descarte de parásitos, este análisis se realiza en la fecha indicada y para este caso el paciente tiene que estar registrado y hacerlo buscado para agregar registro.

Registrar CRED:

Sinapsis: El caso de uso permite registrar el resultado de crecimiento y desarrollo, este análisis se realiza en la fecha indicada y para este caso el paciente tiene que estar registrado y hacerlo buscado para agregar registro

Registrar evaluación psicomotora:

Sinapsis: El caso de uso permite registrar el resultado de evaluación psicomotriz, este análisis se realiza en la fecha indicada y para este caso el paciente tiene que estar registrado y hacerlo buscado para agregar registro

Buscar vacuna:

Sinapsis: El caso de uso permite buscar una vacuna que necesiten ver la información o si está registrada en el sistema.

Registrar Vacuna:

Sinapsis: El caso de uso permite registrar una vacuna para luego usarla de alguna forma con los pacientes.

Actualizar información de vacuna:

Sinapsis: El caso de uso actualiza la información de una vacuna, en el caso que tenga autorización de actualizar la información.

Registrar centro de salud:

Sinapsis: El caso de uso permite registrar un centro de salud, debido a que el usuario viene a ser a tendido de otro al que no pertenece.

Buscar centro de salud:

Sinapsis: El caso de usos permite buscar un centro de salud para luego asignar al paciente que necesite ser registrado u otro acción a realizar.

Actualizar información de centro de salud:

Sinapsis: El caso de uso permite actualizar la información de un centro de salud, siendo el caso que sea necesario y bajo la autorización de la encargada.

Registro de Usuario:

Sinapsis: El caso de uso permite dar de alta a un nuevo usuario siendo el caso que se integró un nuevo trabajador al área o sea el caso que lo autorice la encargada.

Buscar usuario:

Sinapsis: El caso de uso permite buscar a un usuario que este registrado en el sistema con anterioridad.

Actualizar usuario:

Sinapsis: El caso de uso permite actualizar los datos o información de un usuario registrado en el sistema., con la autorización de propio usuario o la encargada del área siendo el caso específico.

[Sistema de Gestión de Información]	
Block de Notas de la Arquitectura	Fecha: <dd/mm/yy>

Ingresar sistema:

Sinapsis: El caso de uso permite ingresar al sistema mediante un login de usuario, siendo el caso que el usuario este registrado con anterioridad y una vez en el sistema puedo acceder a diferentes prioridades

Salir sistema:

Sinapsis: El caso de uso permite salir del sistema en cualquier momento.

Modificar perfil de usuario:

Sinapsis: El caso de usos permite modificar el perfil de un usuario, autorizado por el mismo usuario.

Ver perfil de usuario:

Sinapsis: El caso de uso permite visualizar el perfil del usuario para mayor conformidad del mismo.

✓ **Vista de restricciones**

En esta vista podremos ver las restricciones, normativas, licenciamiento, estándares y otros a las cuales está sujeto el proceso de desarrollo como el producto desarrollado:

1.- Normativas:

Existen restricciones, normativas que determinan algunas decisiones del producto desarrollado.

Licenciamiento:

web2py se distribuye según los términos de la licencia LGPL versión 3.

En conformidad con LGPL (**LICENCIA PÚBLICA GENERAL MENOR**) puedes:

- ✓ Redistribuir web2py con tus aplicaciones, distribuir tus aplicaciones usando las librerías oficiales de web2py según la licencia que elijas.

Pero tienes la obligación de:

- ✓ Disponer de las aclaraciones necesarias para dar noticia del uso de web2py en la documentación.

No hay garantías para el programa, en la medida permitida por la ley aplicable. Excepto cuando se indique lo contrario por escrito, los titulares del copyright y/u otras partes proporcionan el programa "tal cual" sin garantías de ningún tipo, ya sean expresas o implicadas, incluyendo, pero no limitado a, las garantías implícitas de comercialización y aptitud para un propósito particular. El riesgo en cuanto a la calidad y rendimiento del programa queda bajo su responsabilidad. si el programa es defectuoso, usted asume el costo de todo servicio, reparación o corrección.

En ningún caso, a menos que lo exija la ley aplicable o que sea acordado por escrito, un titular de derecho de autor o cualquier otra parte que modifique y/o

[Sistema de Gestión de Información]	
Block de Notas de la Arquitectura	Fecha: <dd/mm/yy>

transmita el programa como se permite arriba, será responsable ante usted por daños, incluyendo cualquier daño general, especial, incidental o daños derivados del uso o imposibilidad de usar el programa, incluyendo pero no limitado a, la pérdida o daño de datos, las pérdidas sufridas por usted o terceros o un fallo del programa para operar con cualquier otro programa, incluso si el propietario o parte de otro tipo fue informada sobre la posibilidad de tales daños.

2.- Estándares:

UML:

El proyecto es un elemento de nivel superior y se almacena en un único archivo.

El modelado de un sistema de software se describe varios modelos, porque no es suficiente para describir el sistema con un solo punto de vista, por lo que normalmente hacemos varios modelos como el Modelo de Casos de Uso, Diseño Modelo, Modelo de Componentes, Modelo de implementación, u otros.

UML (Unified Modeling Language) es tan de uso general lenguaje de modelado que se podría utilizar para expresar cualquier tipo de sistemas intensivos en software. En esta razón, el uso de UML para un dominio o plataforma específica no es suficiente, por lo que puede que tenga que definir UML Profile. StarUML proporciona perfiles UML que se pueden utilizar para ampliar UML. Por ejemplo, los perfiles de UML se pueden utilizar para los siguientes fines:

- Perfiles para los lenguajes de programación específicos (C / C ++, Java, C #, Python, etc)
- Perfiles para metodologías de desarrollo específicos (RUP, Catálisis, UML Componentes, etc)
- Perfiles para dominios específicos (EAI, CRM, SCM, ERP, etc)

Interfaz Web:

La interface de usuario debe estar orientada a la web. Debe ser posible visualizar el contenido utilizando cualquiera de los navegadores más usados y conocidos.

Web Services:

Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen

Comentado [SMA2]: Tu sistema no usa web services, entonces solo quitálo...

3.- Tecnología:

El desarrollo completo del Sistema debe estar realizado con lenguaje de programación Python, utilizando el framework Web2py basándose fuertemente en las tecnologías involucradas del framework

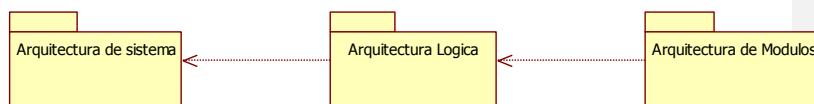
4.- Soporte:

El Sistema de Gestión de Información SIG-Vacunas tendrá mantenimiento evolutivo permanente orientado principalmente al desarrollo de nuevos módulos para cubrir nuevos servicios brindados en el área.

[Sistema de Gestión de Información]	
Block de Notas de la Arquitectura	Fecha: <dd/mm/yy>

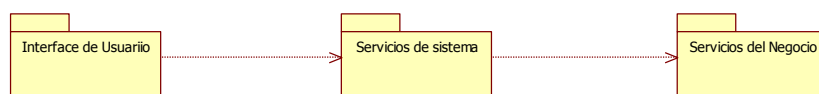
✓ Vista Lógica

Esta vista presenta tres niveles de arquitectura; a continuación se grafican los niveles a tratar.



Arquitectura del sistema:

En el primer nivel se especifica el estilo de arquitectura para el sistema de gestión de información. El siguiente diagrama presenta la arquitectura del sistema.



Cada capa determina un rol para los módulos que residen en ella. La Interfaz de Usuario tiene como objetivo el manejo de la lógica del usuario. Está conformada por el conjunto de páginas web dinámicas que interactúan entre si y por módulos que rodean la lógica de los casos de uso.

Los Servicios del Sistema representan los servicios básicos que debe proveer el sistema; estos servicios son directamente utilizados por los módulos de la capa superior.

Los Servicios de Negocio son servicios de manejo de información del negocio; son servicios aún más básicos.

Arquitectura Lógica:

La Arquitectura Lógica presenta un refinamiento de la Arquitectura del Sistema. La dimensión Requerimientos, principalmente la Vista de Casos de uso, va a verse realizada por los módulos aquí presentados. Se analizará los módulos presentes en cada capa de la Arquitectura del Sistema, presentando finalmente la Arquitectura Lógica.

Interface de usuario:

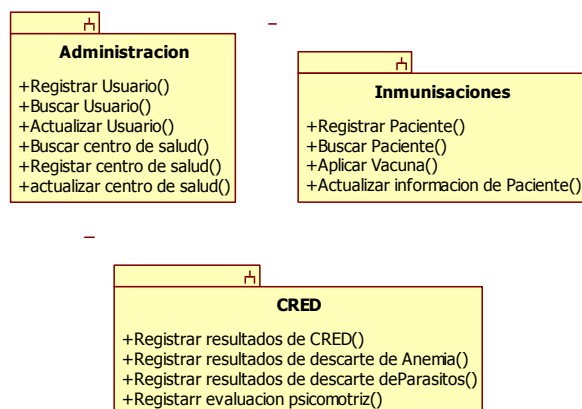
Esta capa consiste de un módulo por cada caso de uso identificado. Cada módulo contiene la lógica que lleva adelante el caso de uso y un conjunto de páginas dinámicas utilizadas por dicha lógica.

- Ver anexo de prototipos

Servicios del Sistema:

Se cuenta con una interfaz por caso de uso; esta ofrece los servicios que el módulo que maneja la lógica del caso de uso requiere. Este módulo realizará todas las interfaces requeridas por los casos de uso asociados a los subsistemas. El siguiente diagrama presenta los módulos identificados:

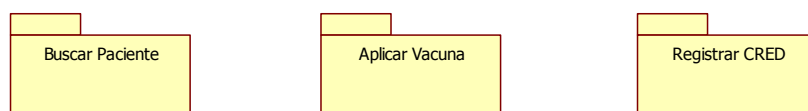
[Sistema de Gestión de Información]	
Block de Notas de la Arquitectura	Fecha: <dd/mm/yy>



Servicios de Negocio:

Los módulos localizados en esta capa son de interés para el Sistema de Gestión de Información. En cambio, los servicios provistos por estos módulos son los necesarios para poder llevar adelante las operaciones de los módulos de la capa superior.

El siguiente diagrama presenta los módulos detectados.



Arquitectura de Módulos:

La Arquitectura de los Módulos presenta un refinamiento de la Arquitectura Lógica. Esta incluye, para cada módulo, una vista que mejor define su diseño. Para cada tipo de módulo, para los módulos de cada capa, se utilizará una vista diferente.

✓ Vista de Procesos

La Vista de Procesos describe los módulos activos de los subsistemas del proyecto; estos son módulos que estarán en ejecución en forma simultánea. Esta vista describe, además, el soporte multi-usuario de la aplicación.

[Sistema de Gestión de Información]	
Block de Notas de la Arquitectura	Fecha: <dd/mm/yy>

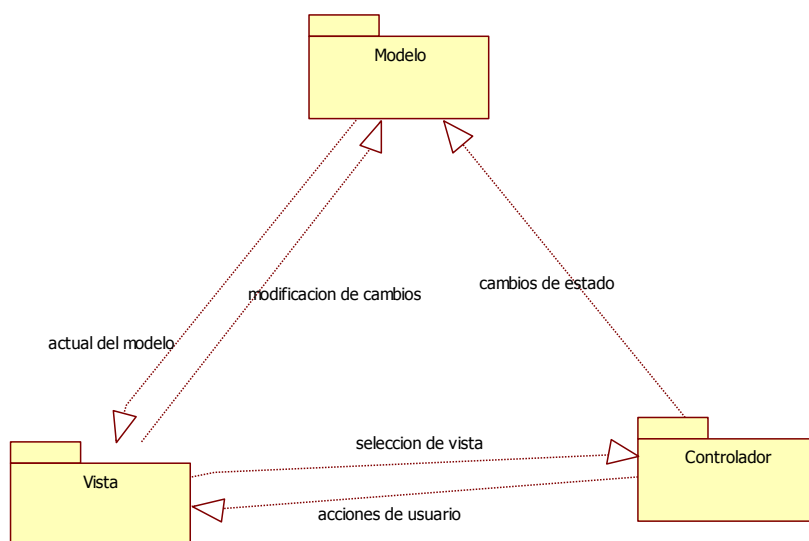


Diagrama de procesos

Procesos Distribuidos

Interfaz de Usuario

El sistema de gestión de información es una aplicación basada en el web. Por esta razón cuenta con un grado de distribución a nivel de interfaz de usuario. A nivel de usuario final, corre en su estación de trabajo una aplicación llamada navegador, como Google Chrome y otros. Esta aplicación es la encargada de presentar al usuario la interfaz de la aplicación y de enviar al servidor las acciones que el cliente realiza.

Distribución de las Capas

Los módulos presentes en las diferentes capas que componen a la aplicación no se encuentran distribuidos. Las razones de no utilizar distribución a este nivel son por las operaciones marcadas como críticas y para disminuir el riesgo del proyecto en función de las tecnologías.

Servicios de Infraestructura

Los servicios de infraestructura son de dos tipos, aplicaciones legadas que deben ser utilizadas y el propio motor de base de datos. El motor de base de datos corre en forma independiente..

✓ Vista de Implementación

La vista de implementación presenta los archivos ejecutables construidos para el sistema de gestión de información. Bajo la tecnología Python, utilizando el framework web2py, ayudando las funcionalidades que esta interpone y presenta apoyado de Python para hacer un proyecto óptimo.

[Sistema de Gestión de Información]	
Block de Notas de la Arquitectura	Fecha: <dd/mm/yy>

Estructura de la Aplicación.

Arquitectura de Implementación

Contemplan el enfoque de agrupación por paquetes de StarUML, se muestra el diagrama de componentes.

✓ Vista de Datos.

En esta vista se presenta el modelo de datos utilizado, su distribución, y una descripción de los servicios de persistencia utilizados.

Modelo de Datos

Contemplan el enfoque de agrupación por paquetes de StarUML, se muestra el diagrama de clases.

Comentado [SMA3]: Colocar también el modelo de datos del sqldesigner

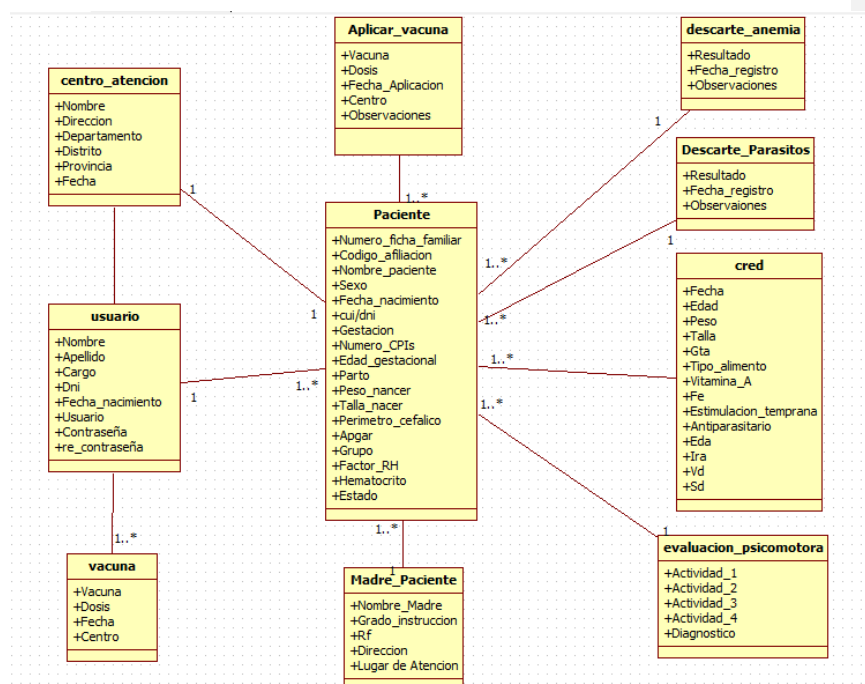


Diagrama de Clases

Distribución

No hay distribución a nivel de datos; todas las tablas radican en la misma base de datos.

Comentado [SMA4]: Quita esta parte

[Sistema de Gestión de Información]	
Block de Notas de la Arquitectura	Fecha: <dd/mm/yy>

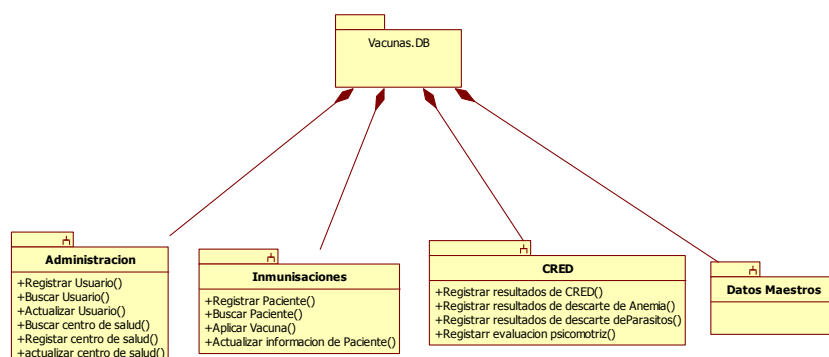


Diagrama de distribución

Servicios de Persistencia

En el nivel inferior se encuentra el motor de base de datos relacional, que para el sistema actual se utiliza el producto.

Por encima de éste se ubica la tecnología Python provista por el framework web2py. Esta tecnología brinda servicios de alto nivel, e independientes del producto, para acceder a fuentes de datos.

La tecnología web2py es utilizada directamente por el módulo db.py. Éste utiliza un sqlite con la conexión adecuado para el motor de web2py. El módulo, ubicado en la capa modelos, provee los servicios.

Servicio de Transaccionalidad

El poder transaccional utilizado es el brindado por web2py.. Cada ejecución de un caso de uso completo, dirigida por uno de estos módulos, está enmarcada en una transacción. Al iniciar un caso de uso da comienzo una transacción, y al terminar el mismo.

Cuando el caso de uso falla se realiza un ticket, generando una lista de ticket en función a los posibles errores que se muestren en la ejecución del módulo.

✓ Vista de Implantación.

La vista de implantación presenta la infraestructura necesaria para instalar el sistema SIG-Vacunas

Se presenta aquí la arquitectura técnica de la aplicación indicando los nodos presentes en la infraestructura tecnológica esperada, y la localización de los componentes en dichos nodos.

Arquitectura técnica:

Considerando la distribución de la aplicación desde el punto de vista de los procesos, es posible identificar tres tipos de nodos, a saber, usuario, SIG-Vacunas, Server, Framework.

- El primer nodo representa a las estaciones de trabajo de los usuarios finales.

[Sistema de Gestión de Información]	
Block de Notas de la Arquitectura	Fecha: <dd/mm/yy>

- El nodo SIG-Vacunas corresponde a la estación de trabajo en SIG-Vacunas. Estos nodos son distinguidos dado que es necesario conocer su dirección IP, ya que en base a esta se identifica la forma en que los casos de uso deben ser llevados adelante.
- El nodo Server representa a aquel equipo en donde correrá el servidor web y la aplicación del SIG-Vacunas
- El último nodo, Framework, representa a aquella infraestructura informática necesaria para correr los sistemas legados.

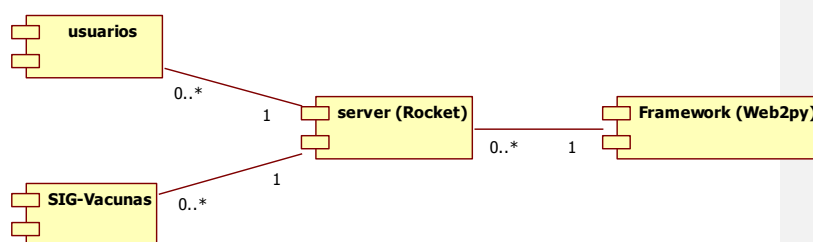


Diagrama de arquitectura técnica

Tecnología requerida

Las estaciones de trabajo de los usuarios (usuarios) deben contar con un browser, ya sea Google Chrome o Microsoft Internet Explorer, con la habilitación de ejecución. Este tipo de nodos comprende también a las terminales que están a disposición.

Las estaciones de trabajo de las SIG-Vacunas debe correr Google Chrome. Estas estaciones de trabajo deben tener un número IP conocido y fijo ya que de eso depende el correcto funcionamiento de la aplicación.

Hay únicamente una instancia del nodo Server, la cual centraliza todos los requerimientos de los clientes. Este nodo debe correr Internet Information Server con soporte del framework. Web2py.

Los nodos framework representan a aquellos equipos que corren los sistemas. El requerimiento tecnológico exigido es el soporte de WebServices.

Implantación

Contemplan el enfoque de agrupación por paquetes de StarUML, se muestra el diagrama de despliegue.