

PROYECTO DEL CURSO - ITERACIÓN 1

VacuAndes

Semana 5 - 202101

202110_ISIS2304_04 - SISTEMAS TRANSACCIONALES

Por:

a.trianaa - Santiago Triana 201923265

j.ramirez - Juan Sebastián Ramírez 201923800

Tabla de contenido

Requerimientos funcionales	3
Requerimientos funcionales de consulta.....	10
Diagrama UML	13
Modelo de datos relacional.....	13
Normalización	14
1ra forma normal	15
2da forma normal	15
3ra forma normal	15
Requerimientos funcionales	16
Requerimientos no funcionales	18
Privacidad	18
Persistencia.....	18
Concurrencia.....	19
Distribución	19

Requerimientos funcionales

Nombre	Registrar las condiciones de priorización para las vacunas
Resumen	Usuario: Administrador del Plan de Vacunación Descripción: Se reciben las condiciones que se van a cumplir a la hora de distribuir las vacunas
Entradas	
Instrucciones de vacunas	
Resultados	
Instrucciones de vacunas registradas en la base de datos	
RNF asociados	
Atomicidad: No se quiere que las instrucciones por ningún caso pasen a medias, esto podría generar demasiados inconvenientes, es por esto que es importante que pasen todas las instrucciones o no pase ninguna.	
Coherencia: Debido a que se están manejando reglas que son reales, es importante que no se cometan errores, ya que en computador se puede generar cualquier regla que se quiera, pero eso no quiere decir que en la vida real van a ser válidas. Es por eso por lo que se debe seguir este comportamiento transaccional	
Durabilidad: Se quiere que estas reglas se mantengan a menos que alguien las vaya a cambiar, bajo ninguna situación se pueden perder estas reglas en ningún momento, podría ser terrible	

Nombre	Registrar la secuencia de estados válidos para el proceso de vacunación de una persona
Resumen	Usuario: Administrador del Plan de Vacunación Descripción: Se recibe una secuencia de estados que serán admitidos como validos en el proceso de vacunación
Entradas	
Estados del proceso de vacunación	
Resultados	
Estados válidos para el proceso de vacunación registrados en la base de datos	

PROYECTO DEL CURSO - ITERACIÓN 1
a trianaa y j.ramirez

RNF asociados
<p>Coherencia: Debido a que se están manejando estados que son reales, es importante que no se cometan errores, ya que en computador se puede generar cualquier estado que se quiera, pero eso no quiere decir que en la vida real va a ser válido.</p> <p>Durabilidad: Una vez que queden los estados registrados, no deben borrarse de la base de datos, solo se pueden borrar bajo la condición de que se solicitó que se borrarán para cambiarlos por nuevos estados. De cualquier otra manera no se pueden borrar, puede dañar toda la operación de la entrega de vacunas si esto sucede de manera imprevista.</p>

Nombre	Registrar una oficina de EPS regional
Resumen	<p>Usuario: Administrador del Plan de Vacunación</p> <p>Descripción: Se registra una nueva oficina de EPS que representará en el programa y en la base de datos la misma oficina que existe en la vida real y que hace parte del proceso de las entregas de las vacunas. Se genera un <i>ID</i> para esa oficina automáticamente.</p>
Entradas	
Región en la que está ubicada la oficina	
Resultados	
Se le asigna un <i>ID</i> a la oficina	
Se crea la nueva oficina con su respectivo ID, su región y sus puntos de vacunación	
RNF asociados	
<p>Coherencia: Las oficinas en la base de datos son representaciones de la vida real, es por esto por lo que no se puede permitir que se generen oficinas distintas a lo que sería una oficina en la vida real, por ejemplo, una oficina con una región que no existe</p> <p>Aislamiento: Se quiere que en ningún momento se cruce información de las oficinas cuando se están haciendo distintas operaciones a la vez, si por algún motivo esto fallara, se generaría un serio problema donde un proceso para una oficina podría terminar modificando otra oficina diferente.</p> <p>Atomicidad: Es importante que las oficinas se creen de manera correcta o no se creen. Ya que, por ejemplo, si se creara una oficina sin región alguna, sería una oficina completamente inútil y solo generaría errores en el programa y en la base de datos.</p>	

Nombre	Registrar un usuario de Vacu-Andes
Resumen	<p>Usuario: Administrador del Plan de Vacunación, nuevo usuario</p> <p>Descripción: Se registra un nuevo usuario en la base de datos, todos los usuarios son ciudadanos y tienen derecho a ser vacunados</p>

PROYECTO DEL CURSO - ITERACIÓN 1
a.trianaa y j.ramirez

Entradas
Username, nombre del usuario para registrarlo es el <i>ID</i>
Password
email
Resultados
Crea el usuario
Conecta el usuario con su ciudadano que ya fue registrado antes por el administrador
RNF asociados
Durabilidad: Lo más importante es que los usuarios se puedan mantener en la base de datos, en especial si aún no se han vacunado porque es muy importante que puedan buscar sus citas en el software
Atomicidad: El usuario solo se debe crear si todo está perfecto

Nombre	Registrar ciudadano colombiano
Resumen	Usuario: Administrador del Plan de Vacunación, ciudadano colombiano Descripción: Se registra un ciudadano nuevo en la base de datos de Vacu-Andes, se deben registrar todos los ciudadanos del país
Entradas	
Cedula del usuario (que servirá como su <i>ID</i> en la base de datos, User Assigned)	
Nombre completo (todos sus nombres, primer y segundo apellido)	
Condiciones especiales (incluye si trabaja en alguna de las condiciones, sirven para clasificarlo en su fase respectiva, las condiciones de edad se verifican automáticamente con su fecha de nacimiento tomada de la cedula)	
Resultados	
Creado el ciudadano en la base de datos	
El ciudadano se le asigna EPS regional	
El ciudadano se le asigna su fase en la que será vacunado	
RNF asociados	
Coherencia: Es extremadamente relevante que la base de datos sea exactamente igual a la vida real, esto se debe a que se están manejando personas reales e interpretándolas de manera virtual, es crucial que esté correcto todo, pues un error con un ciudadano puede dificultar su proceso de vacunación y puede generar información y real o ciudadanos que no existen.	
Aislamiento: Las transacciones de ciudadanos deben estar hechas de manera completamente independiente, no se quiere que se dañe la información de ningún ciudadano por un error en las transacciones	
Atomicidad: Los ciudadanos solo deben ser inscritos en la base de datos si toda su información está completa, de otra manera no se deben registrar en el sistema y toda la transacción se quiere cancelar. Cualquier caso en el que se tenga un ciudadano con información incompleta es bastante peligroso para la base de datos y para garantizar que el ciudadano se pueda vacunar.	

PROYECTO DEL CURSO - ITERACIÓN 1
a trianaa y j.ramirez

Nombre	Registrar punto de vacunación
Resumen	Usuario: Administrador de la EPS regional Descripción: Se registra un nuevo punto de vacunación en una oficina regional determinada
Entradas	
Oficina regional en la que se va a registrar el punto	
Localización	
Capacidad Atención Simultanea	
Capacidad Atención Total Diaria	
Infraestructura para dosis	
Cantidad de vacunas enviabiles	
Tipo de punto	
Resultados	
Se registró el punto de vacunación en la base de datos y el programa	
Basado en la cantidad de vacunas que se pueden recibir, se le asignan las vacunas respectivas	
RNF asociados	
Coherencia: El punto de vacunación es el que distribuye vacunas por medio de citas, cualquier error en la coherencia podría llevar a que se pierdan vacunas en el proceso o que no haya como guardarlas	
Durabilidad: Los puntos se deben guardar en la base de datos de manera indefinida y solo deben salir si se quiere borrar alguno	
Atomicidad: El punto debe ser registrado con la información completa o no debe ser registrado, debido a que, por ejemplo, si se registra un punto sin su infraestructura o cantidad de vacunas que puede guardar, entonces se tendría un error ya que no se sabría cuales vacunas se pueden enviar.	

Nombre	Asignar talento humano a un punto de vacunación
Resumen	Esta operación registra el personal de talento humano en salud y seguridad que es asignado a un punto de vacunación. Las personas asignadas ya deben ser conocidas por la plataforma para poder realizar dicha asignación. La persona que está en la capacidad de realizar dicha asignación es el administrador de oficina de EPS regional.
Entradas	
Username del usuario que desea realizar la operación. Debe verificarse que esta persona sea el Administrador de la oficina de EPS regional	
Cedula del trabajador a añadir; se debe verificar que esta persona exista en la base de datos y se trate de un trabajador THS (talento humano en salud)	
Resultados	
El trabajador a sido asignado correctamente al punto de vacunación y ahora es visible dentro de dicha entidad.	
RNF asociados	

PROYECTO DEL CURSO - ITERACIÓN 1
a.trianaa y j.ramirez

A pesar que solamente se hace un acceso de modificación a la BD, es importante respetar el principio de transaccionalidad a la hora de realizar todo el requerimiento funcional. Esto implica que se debe verificar que se restablezcan todo tipo de actualizaciones a los datos en caso que se tenga un estado final de no realizada.

Por el otro lado, se debe velar que la información brindada sea persistida de forma congruente (siguiendo las reglas de integridad preestablecidas), respetando las propiedades ACID (atomicidad, coherencia, aislamiento y durabilidad)

Nombre	Registrar la llegada de un lote de vacunas a una EPS regional
Resumen	Pone a disposición de una EPS un conjunto de dosis de vacunas. Con estas dosis disponibles, las EPS regionales pueden proceder a la realización del Plan de Vacunación.
Entradas	
Lista de idVacuna con cada una de las vacunas del lote que se van a asignar a la EPS regional.	
IdOficina de la oficina regional a la que se le van a asignar las vacunas disponibles.	
Resultados	
Se actualiza la cantidadVacunasActuales de la oficina regional de la EPS.	
Se añaden nuevas relaciones entre vacunas y oficinas regionales.	
RNF asociados	
Es importante respetar el principio de transaccionalidad a la hora de realizar todo el requerimiento funcional. La existencia de cada una de las vacunas tiene que ser verificada antes de ser asignadas. De igual forma es necesario verificar que haya una oficina con el idOficina dado por parámetro. En el caso que alguna de estas búsquedas previas sea fallida debe abortarse toda la transacción (Y se debe reestablecer todas las vacunas ya asignadas a la oficina anteriormente en el caso que se hayan añadido algunas con éxito).	
Se debe revisar que la información brindada sea persistida de forma congruente (siguiendo las reglas de integridad preestablecidas para VacuAndes), respetando las propiedades ACID.	

Nombre	Registrar la llegada de un lote de vacunas a un Punto de vacunación
Resumen	Pone a disposición de un punto de vacunación una lista de dosis de vacunas. Con estas dosis disponibles, los administradores de los puntos de atención pueden proceder a la aplicación de la vacuna a los ciudadanos asignados. Esta operación es realizada por los Administradores de los Puntos de vacunación.
Entradas	
Lista de idVacuna con cada una de las vacunas del lote que se van a asignar al punto de vacunación	
IdPuntoVacunacion del punto de vacunación al que se le van a asignar las vacunas disponibles.	
IdOficina de la oficina regional de dónde se van a asignar las vacunas disponibles.	
Username del usuario que desea realizar la operación. Debe verificarse que esta persona sea el Administrador del punto de vacunación	
Resultados	
Se actualiza la cantidadVacunasActuales del punto de vacunación.	

PROYECTO DEL CURSO - ITERACIÓN 1
a trianaa y j. ramirezb

Se añaden nuevas relaciones entre vacunas y puntos de vacunaciones.
Se actualiza la cantidadVacunasActuales de la oficina regional de la EPS.
RNF asociados
Es importante respetar el principio de transaccionalidad a la hora de realizar todo el requerimiento funcional. La existencia de cada una de las vacunas en la oficina tiene que ser verificada antes de ser asignadas. De igual forma es necesario verificar que haya un PuntoVacunacion con el id dado por parámetro. En el caso que alguna de estas búsquedas previas sea fallida debe abortarse toda la transacción (Y se debe reestablecer todas las vacunas ya asignadas al punto de vacunación anteriormente, en el caso que se hayan añadido algunas con éxito).
Se debe revisar que la información brindada sea persistida de forma congruente (siguiendo las reglas de integridad preestablecidas para VacuAndes), respetando las propiedades ACID.

Nombre	Asignar un ciudadano a un punto de vacunación.
Resumen	Se debe asegurar que un ciudadano esté asignado a un único punto de vacunación. Esta operación es realizada por el Administrador de oficina de EPS regional.
Entradas	
Cedula del ciudadano que se desea agregar al punto de vacunación	
IdPuntoVacunacion del punto de vacunación al que se le va a asignar el ciudadano	
IdOficina de la oficina regional del punto de vacunación al que se le va a asignar el ciudadano.	
Username del usuario que desea realizar la operación. Debe verificarse que esta persona sea el Administrador de oficina de EPS regional	
Resultados	
Se añade una relación ciudadano-PuntoVacunacion	
RNF asociados	
Es importante respetar el principio de transaccionalidad a la hora de realizar todo el requerimiento funcional. La existencia de cada ciudadano debe ser verificada antes de ser asignado el punto de vacunación. De igual forma es necesario verificar que haya un PuntoVacunacion con el id dado por parámetro. En el caso que alguna de estas búsquedas previas sea fallida debe abortarse toda la transacción.	

Nombre	Asignar una cita a un ciudadano.
Resumen	Esta operación se hace siguiendo la priorización definida en el plan y la capacidad de atención del punto de vacunación. Es realizada por los Administradores de los Puntos de vacuna.
Entradas	
Cedula del ciudadano al que se desea crear la cita	
IdPuntoVacunacion del punto de vacunación al que se le va a asignar el ciudadano	

PROYECTO DEL CURSO - ITERACIÓN 1
a trianaa y j. ramirez

IdOficina de la oficina regional del punto de vacunación al que se le va a signar el ciudadano.
Hora y minuto de la cita.
Día de la cita.
Mes de la cita.
Año de la cita.
Resultados
Se crea la cita en el punto de vacunación.
Se añade un ciudadano a la cita creada.
RNF asociados
Es importante respetar el principio de transaccionalidad a la hora de realizar todo el requerimiento funcional. La existencia de cada ciudadano debe ser verificada antes junto a que haya un PuntoVacunacion con el id dado por parámetro. En el caso que alguna de estas búsquedas previas sea fallida debe abortarse toda la transacción.
Se debe revisar que la información brindada sea persistida de forma congruente (siguiendo las reglas de integridad preestablecidas para VacuAndes), respetando las propiedades ACID.

Nombre	Registrar avance en el proceso de vacunación de una persona
Resumen	Se puede registrar cuando un paciente sea vacunado, de igual forma se incluye el caso en el que una persona no quiera recibir la vacuna y también la posibilidad de que cambie de opinión y ahora sí la quiera. Esta operación es realizada por los Operadores de los Puntos de vacunación.
Entradas	
	Cedula del ciudadano del que se le desea registrar el avance
	IdPuntoVacunacion del punto de vacunación al que pertenece el ciudadano
	IdOficina de la oficina regional del punto de vacunación al que pertenece el ciudadano
	Username del usuario que desea realizar la operación. Debe verificarse que esta persona sea el Administrador del punto de vacunación
	Información sobre el estado del paciente en cuanto a la vacunación vacunado/no vacunado/primera dosis
	Información sobre si el ciudadano desea vacunarse/no desea vacunarse
Resultados	
	Se modifica el estado del proceso de vacunación de un ciudadano
RNF asociados	
	Es importante respetar el principio de transaccionalidad a la hora de realizar todo el requerimiento funcional. La existencia de cada ciudadano debe ser verificada antes de intentar actualizar los datos, de igual forma, si fue vacunado debe verificarse si tiene una cita pasada en la que fue vacunado. En caso que no haya tenido una cita previa deben reintegrarse los datos a su estado inicial.
	Se debe revisar que la información brindada sea persistida de forma congruente (siguiendo las reglas de integridad preestablecidas para VacuAndes), respetando las propiedades ACID.

Requerimientos funcionales de consulta

Nombre	Mostrar todos los ciudadanos atendidos por un punto de vacunación
Resumen	<p>Usuario: Administrador del plan de vacunación</p> <p>Descripción: Retorna al administrador una lista con todos los ciudadanos que están registrados en un punto de vacunación</p>
Entradas	
Punto de vacunación del que se quieren encontrar los ciudadanos	
Resultados	
Se registró el punto de vacunación en la base de datos y el programa	
Basado en la cantidad de vacunas que se pueden recibir, se le asignan las vacunas respectivas	
RNF asociados	
Coherencia: La coherencia es importante por el manejo de los ciudadanos, se debe garantizar que en ningún momento se dañe la información de estos, pues generaría un resultado ajeno a la realidad y un ciudadano podría terminar desaparecido de la lista y por lo tanto no recibiría su vacuna.	
Atomicidad: Se requiere de la lista completa, si por algún motivo no se puede garantizar que se retorna la lista completa, entonces se debe cancelar la transacción. Una lista incompleta significaría que alguien se está quedando sin vacuna y es un riesgo que no se debe tomar.	

Nombre	Mostrar los 20 puntos de vacunación más efectivos
Resumen	<p>Usuario: Administradores de Vacu-Andes</p> <p>Descripción: Se puede determinar en un rango de tiempo específico, cuáles son los puntos de vacunación que más dosis están aplicando en ese tiempo. Muy útil para determinar qué cosas se están haciendo bien, ya que quienes logran vacunar a más personas están haciendo un buen proceso con su infraestructura y es crucial que esta información llegue a los demás puntos para que puedan adaptar estrategias similares</p>
Entradas	
Intervalo de tiempo en el que se quieren buscar los puntos más efectivos.	
Fecha en la que se quieren buscar los puntos más efectivos	
Resultados	
Una lista con los 20 puntos de vacunación más efectivos bajo los parámetros dados	
RNF asociados	
Coherencia: Los datos deben ser 100% coherentes porque serán datos que se utilicen para mejorar la efectividad en general del sistema, es por esto por lo que no se puede dejar un margen de error en donde se retornen puntos erróneos porque el análisis posterior de esa data saldría inutilizable e ilógico.	

PROYECTO DEL CURSO - ITERACIÓN 1
a.trianaa y j.ramirez

Aislamiento: Así se vayan a pedir muchos datos de muchos puntos, no se puede dar el caso en que por algún motivo se presente un error que los mezcle, la integridad de los datos debe mantenerse en las transacciones a pesar de que sean muchas al mismo tiempo.

Nombre	Mostrar el índice de vacunación para un grupo poblacional
Resumen	<p>Usuario: Administradores de plan y administradores regionales</p> <p>Descripción: Se retorna el porcentaje de ciudadanos vacunados en la región, es importante para saber el número de vacunas que faltan y también para saber cómo transcurre el proceso de vacunación</p>
Entradas	
Grupo poblacional: Una o varias regiones, una o varias EPS, grupo de priorización	
Etapa del proceso que se desea verificar (Ya agendados, una dosis, dosis completadas, etc.)	
Resultados	
Retorna un porcentaje de personas vacunadas con respecto a los ciudadanos totales de los segmentos seleccionados por parámetro	
RNF asociados	
Coherencia: Se debe mantener la coherencia en todo momento, esto se debe a que los números no deben variar y deben ser exactamente iguales a los de su contra parte en la vida real, de no ser de esta manera se corre el riesgo de que las vacunas sean enviadas a lugares que no las necesitan y se desperdicien	
Atomicidad: Es preferible que si se presenta un error se detenga la operación completa, si no se hace de esta manera se corre el riesgo de representar de manera errónea el porcentaje y esto podría traer como consecuencia que terminen sobrando o faltando vacunas para los ciudadanos.	

Nombre	Mostrar los puntos de vacunación con disponibilidad de dosis de vacunas
Resumen	Se deben poder desplegar la lista de las regiones, localización y cantidad de unidades de los puntos de vacunación que tiene disponibilidad de vacunas.
Entradas	
Username del usuario que desea realizar la operación. Debe verificarse si el trabajador es administrador de VacuAndes para poder realizar la operación	
Resultados	
Lista de las regiones y localización de los puntos de vacunación que tiene disponibilidad de vacunas.	
Cantidad de vacunas correspondientes a cada punto de vacunación	
RNF asociados	
Debido a que el requerimiento no requiere crear, modificar, actualizar o eliminar información de la base de datos, no es necesario añadir un carácter transaccional o persistente durante el proceso. Es necesario acceder a la información y lograr interpretarla para retornar el resultado deseado.	

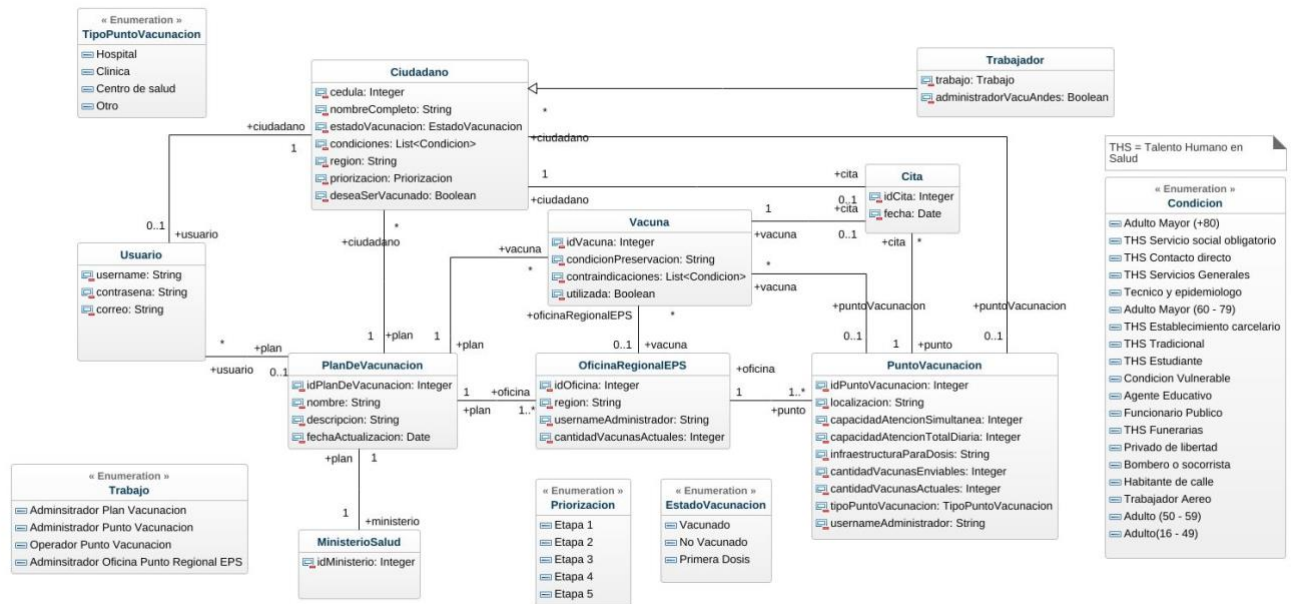
PROYECTO DEL CURSO - ITERACIÓN 1
a trianaa y j. ramirezb

Nombre	Mostrar el proceso de vacunación de un ciudadano.
Resumen	Se debe desplegar toda la información con respecto a ese ciudadano y su estado de vacunación. (Todas las operaciones, desde la asignación de punto de vacunación, hasta la aplicación de todas las dosis de vacuna requeridas).
Entradas	
Cedula del ciudadano del que se desea conocer la información.	
Username del usuario que desea realizar la operación. Debe ser el mismo el que accede a su información, o un trabajador que trabaje en su oficina/punto de vacunación.	
Resultados	
Se despliega el nombre completo del ciudadano buscado.	
Se despliega el estado de vacunación del ciudadano buscado. (vacunado/no vacunado/primera dosis)	
Se despliegan las condiciones del ciudadano buscado.	
Se despliega la priorización del ciudadano buscado.	
Se despliega la región, la oficina de la EPS y el punto de vacunación que le fue asignado al ciudadano buscado.	
Se despliega la cita del ciudadano buscado en caso de tenerla.	
RNF asociados	
Debido a que el requerimiento no requiere crear, modificar, actualizar o eliminar información de la base de datos, no es necesario añadir un carácter transaccional o persistente durante el proceso. Es necesario acceder a la información y lograr interpretarla para retornar el resultado deseado.	

Nombre	Mostrar el estado de las vacunas registradas en el plan de vacunación.
Resumen	Se deben poder desplegar una lista con todas las vacunas registradas en el sistema (así hayan sido usadas o no)
Entradas	
Username del usuario que desea realizar la operación. Debe verificarse si el trabajador es administrador de VacuAndes para poder realizar la operación	
Resultados	
Lista de las vacunas con información como; si han sido usadas, su condición de preservación y contraindicaciones.	
Cantidad de vacunas correspondientes a cada punto de vacunación	
RNF asociados	
Debido a que el requerimiento no requiere crear, modificar, actualizar o eliminar información de la base de datos, no es necesario añadir un carácter transaccional o persistente durante el proceso. Es necesario acceder a la información y lograr interpretarla para retornar el resultado deseado.	

PROYECTO DEL CURSO - ITERACIÓN 1
a.trianaa y j.ramirez

Diagrama UML



Modelo de datos relacional

Ciudadano								
cedula	nombreCompleto	estadoVacunacion	region	priorizacion	deseaSerVacunado	planVacunacion	puntoVacunacion	oficinaRegionalAsignada
PK, UA	NN	NN	NN	NN	NN	NN, FKPlanDeVacunacion.idPlanDeVacunacion	NN, FKpuntoVacunacion.idPuntoVacunacion	NN, FKoficinaRegionalEPS.idOficina

Usuario				
username	contrasena	consec	planVacunacion	ciudadano
PK, UA	NN	NN, ND	NN, FKPlanDeVacunacion.idPlanDeVacunacion	FK Ciudadano.cedula

PlanDeVacunacion			
idPlanDeVacunacion	nombre	descripcion	fechaActualizacion
PK, UA	NN, ND	NN	NN, CK

MinisterioSalud	
idMinisterio	planVacunacion
PK, UA	NN, FKPlanDeVacunacion.idPlanDeVacunacion

PuntoVacunacion								
idPuntoVacunacion	localizacion	capacidadAtencionSimultanea	capacidadAtencionTotalDaria	infraestructuraParaDoosis	cantidadVacunasEnviabiles	cantidadVacunasActuales	tipoPuntoVacunacion	usernameAdministrador
PK, UA	NN, ND	NN	NN	NN	NN	NN	NN	ND

Cita				
idCita	fecha	ciudadano	puntoVacunacion	vacuna
PK, UA	NN, CK	NN, FK Ciudadano.cedula	NN, FK PuntoVaunacion.idPuntoVacunacion	NN, FK vacuna.idVacuna

Trabajador		
cedula	trabajo	administradorVacuAndes
PK, FK ciudadano.cedula	NN, FK Trabajo.trabajo	NN

Vacuna					
idVacuna	condicionPreservacion	puntoVacunacion	planDeVacunacion	oficinaRegional	utilizada
PK, UA	NN	NN, FK PuntoVacunacion.idPuntoVacunacion	NN, FK PlanDeVacunacion.idPlanDeVacunacion	NN, FK oficinaRegionalEPS.idOficina	NN

OficinaRegionalEPS				
idOficina	region	usernameAdministrador	cantidadVacunasActuales	planDeVacunacion
PK, UA	NN	NN, ND	NN	NN, FKPlanDeVacunacion.idPlanDeVacunacion

THS = Talento Humano en Salud

condicion
PK, UA
Adulto Mayor (+80)
THS Servicio social obligatorio
THS Contacto directo
THS Servicios Generales
Tecnico y epidemiologo
Adulto Mayor (60 - 79)
THS Establecimiento carcelario
THS Tradicional
THS Estudiante
Condicion Vulnerable
Agente Educativo
Funcionario Publico
THS Funerarias
Privado de libertad
Bombero o socorrista
Habitante de Calle
Trabajador Aereo
Adulto (50 - 59)
Adulto (16 - 49)

TipoPuntoVacunacion
tipoPuntoVacunacion
PK, UA
Hospital
Clinica
Centro de Salud
Otro

EstadoVacunacion
estadoVacunacion
PK
Vacunado
No vacunado
Primera Dosis

listCondicionesCiudadano	
ciudadano	condicion
PK, FK Ciudadano.cedula	PK, FK Condiciones.condicion

Trabajo
trabajo
PK, UA
Administrador Plan de Vacunacion
Administrador Punto de Vacunacion
Operador Punto de Vacunacion
Administracion Oficina Punto Regional EPS

Priorizacion
etapa
PK, UA
Etapa 1
Etapa 2
Etapa 3
Etapa 4
Etapa 5

listContraindicacionesVacuna	
vacuna	condiciona
PK, FK Vacuna.idVacuna	PK, FK Condiciones.condicion

Normalización

Se modelará cada una de las clases y su dependencia funcional de forma muy rudimentaria para analizar la normalización del modelo de negocio propuesto. Para representar los atributos primos se utilizará el subrayado en verde, por el lado contrario, los atributos no primos no tendrán ningún tipo de subrayado.<

Ciudadano (cedula, nombreCompleto, estadoVacunacion, región, priorización, deseaSerVacunado, planVacunacion, puntoVacunacion, oficinaRegionalAsignada)
Llave = {cedula}

Usuario(username, contrasena, correo, planVacunacion, ciudadano)
Llave= {username}

PlanDeVacunacion (idPlanDeVacunacion, nombre, descripcion, fechaActualizacion)
Llave= {idPlanDeVacunacion}

MinisterioDeSalud(idMinisterio, planVacunacion)
Llave= {idMinisterio}

OficinaRegionalEPS (idOficina, region, usernameAdminsitrator, cantidadVacunasActuales, planVacunacion)
Llave= {idOficina}

PuntoVacunacion (idPuntoVacunacion, localizacion, capacidadAtencionSimultanea, capacidadAtencionTotalDiaria, infraestructuraParaDosis, cantidadVacunasEnviabiles, cantidadVacunasActuales, tipoPuntoVacunacion, usernameAdministrador)
Llave = {idPuntoVacunacion}

Cita (idCita, fecha, ciudadano, puntoVacunacion, vacuna)

Llave = {idCita}

Vacuna (idVacuna, condicionPreservacion, puntoVacunacion, planDeVacunacion, oficinaRegional, utilizada)

Llave = {idVacuna}

Trabajador (cedula, trabajo, administradorVacuAndes)

Llave = {cedula}

Enumeraciones

Trabajo (trabajo) **Llave** = {trabajo}

Condicion (condicion) **Llave** = {condicion}

TipoPuntoVacunacion (tipoPuntoVacunacion) **Llave** =
(tipoPuntoVacunacion)

EstadoVacunacion (estadoVacunacion) **Llave** = (estadoVacunacion)

1ra forma normal

Como se puede observar en el modelo relacional o en los conjuntos anteriormente mostrados, no se tiene ningún atributo que guarde listas. Esto significa que no existen atributos multivalor y los dominios son atómicos para todas las clases. Es posible decir que la solución se encuentra en 1FN.

2da forma normal

Debido a que dentro de la base de datos no existe redundancia entre los datos almacenados, obtenemos que los atributos primos de una clase dependen de forma completa de todas la llaves candidatas. Esto, sumado a que ya se encuentra en 1FN, podemos concluir que se encuentra en 2FN.

3ra forma normal

A pesar que tenemos la 2FN, es posible evidenciar dependencias transitivas entre atributos primos. En ejemplo claro, entre los tantos que pudimos identificar es si se desea obtener el nombre completo de un usuario. Es necesario acceder primero a ciudadano usando su llave almacenada en usuario.

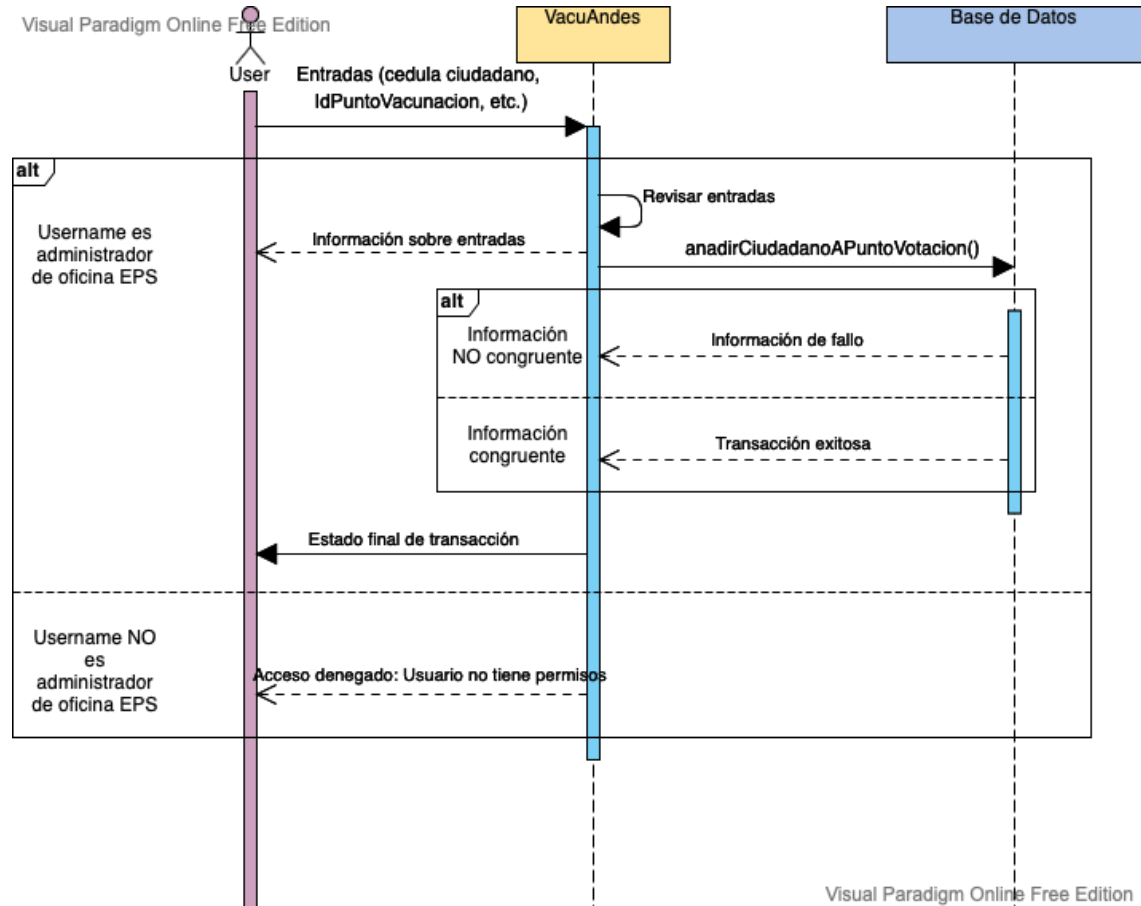
Usuario -> Ciudadano

Ciudadano -> nombreCompleto

Podemos entonces concluir que nuestro modelo de negocio propuesto para VacuAndes se encuentra en la 2da forma normal, pues a pesar que no tiene dependencias parciales, cuenta con dependencias transitivas entre atributos no primos.

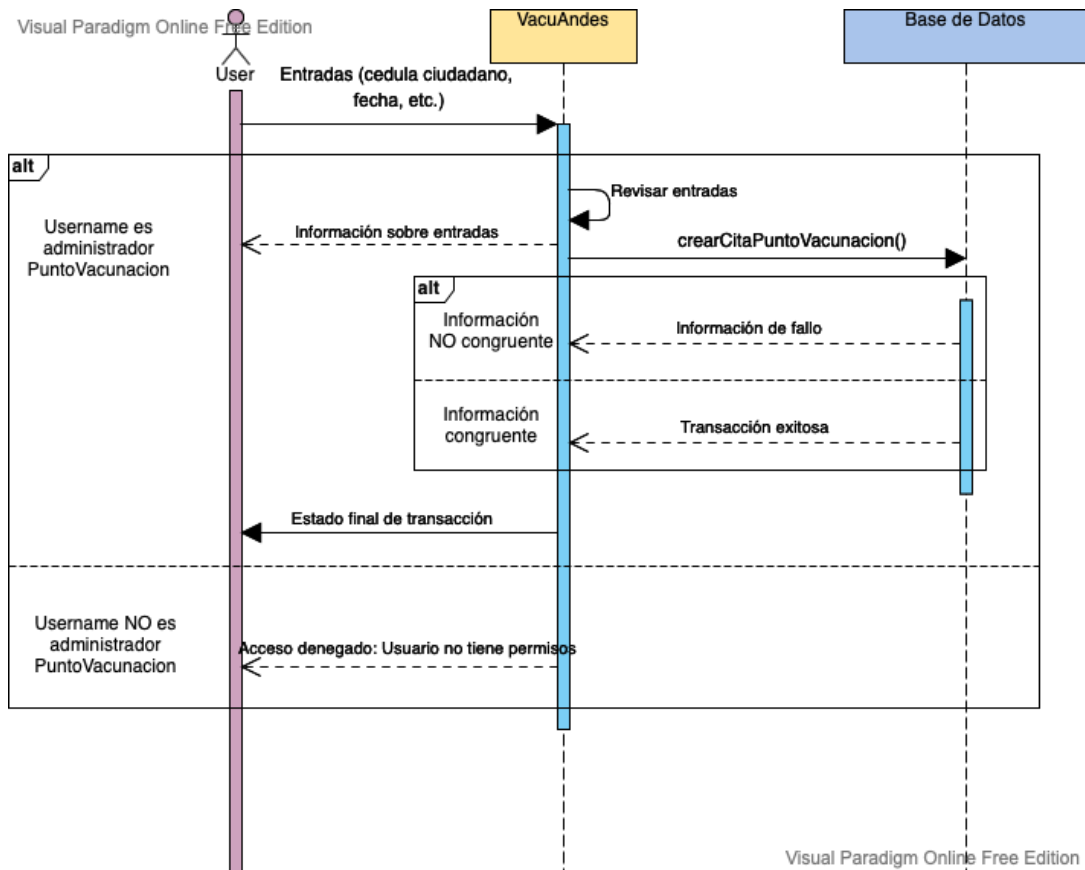
Requerimientos funcionales

RF10

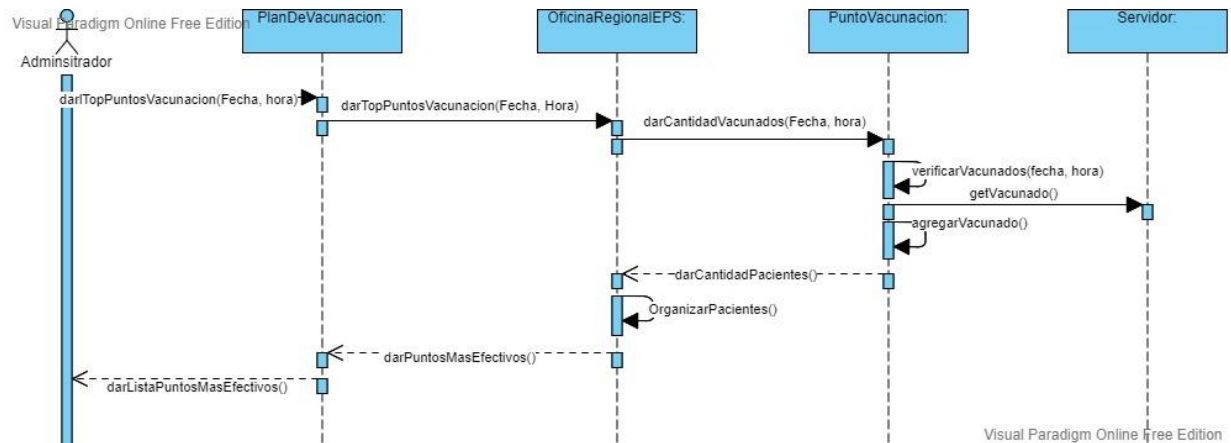


RF11

PROYECTO DEL CURSO - ITERACIÓN 1 a trianaa y j.ramirez

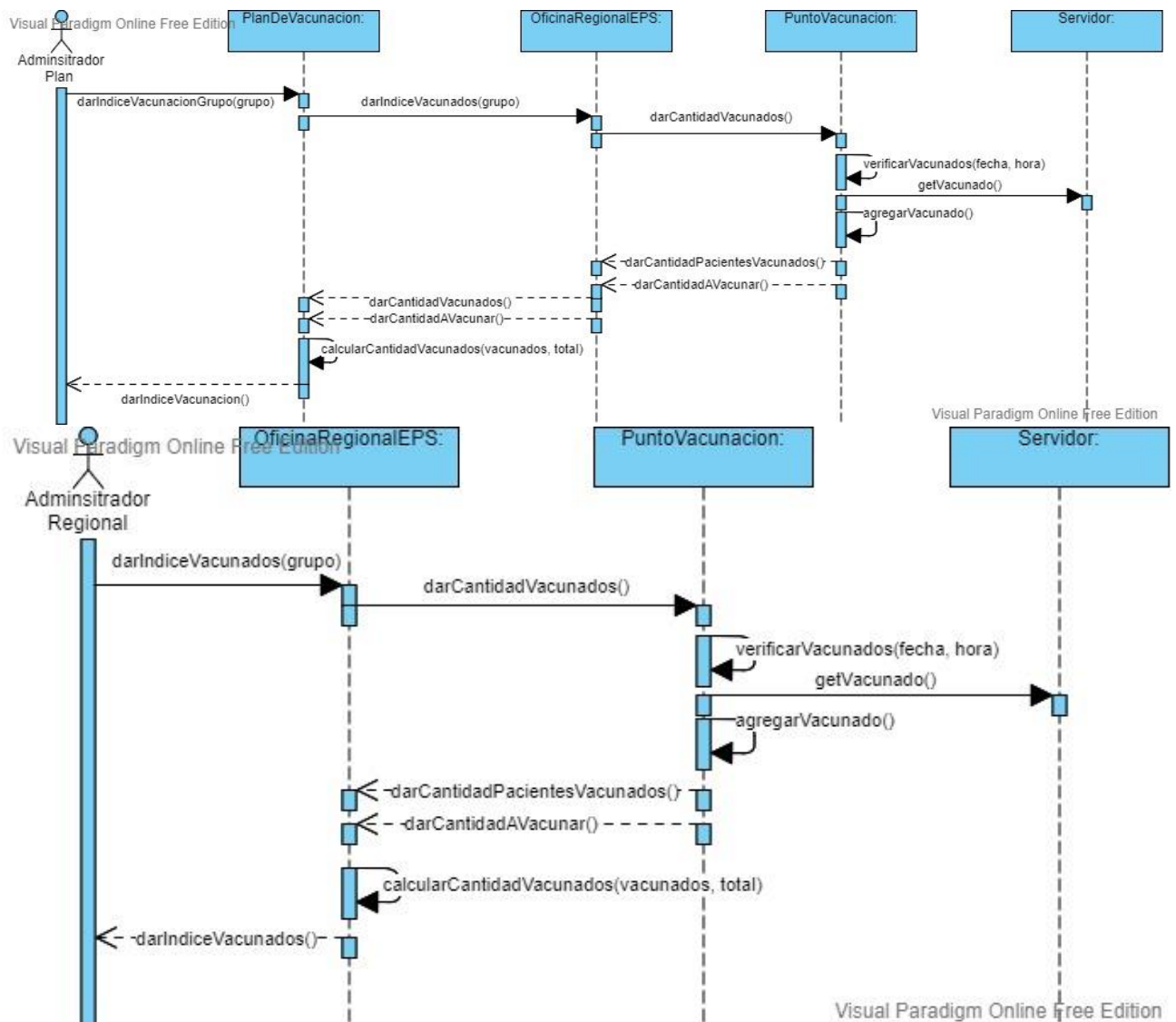


RFC2



RFC3

PROYECTO DEL CURSO - ITERACIÓN 1
a.trianaa y j.ramirez



Requerimientos no funcionales

Privacidad

El modelo de Vacu-Andes propuesto está diseñado para que el acceso a la información personal de cada usuario solo se pueda dar por los trabajadores que tienen los permisos necesarios. Esto significa que las personas que acceden como usuarios no trabajadores, no tienen ningún acceso a la información de los demás usuarios o las dependencias de vacunación, pues solo pueden acceder a su información y a su cita asignada previamente. Esto se logra mediante la diferenciación de roles no solo entre ciudadanos, sino también entre cargos de empleados del plan de vacunación actual.

Persistencia

Una de las necesidades claves del programa es que persista a través del tiempo. Esto no significa únicamente guardar la información en un lugar físico, sino asegurarse que dicha información sea congruente con las necesidades del negocio, y congruente con las leyes de la vida real. Para resolver esto se plantean atributos de vital importancia a los objetos identificados en el plan de vacunación nacional. De igual forma, se establecen relaciones que

se presentan en el plan de acción nacional y son viables a nivel administrativo para una solución de este tipo.

De igual forma, el manejo de la información y los requerimientos funcionales planeados dentro del diseño de la solución, se piensan para que puedan ser aplicados en hardware actual. Esto significa que se maneja y tiene en mente la cantidad de información que puede manejar cada una de las capas involucradas en el programa. (Esto con el fin de no encontrarse con problemas como que el cliente no pueda recibir los paquetes del servidor debido a falta de memoria principal).

Concurrencia

Utilizando los principios de la transaccionalidad y los protocolos para mostrar/asignar información, la solución puede atender a que los requerimientos sean solicitados de manera concurrente, manteniendo la congruencia de la información contenida. Los requerimientos funcionales que tienden a modificar la información tienen carácter transaccional y son manejados de manera que, se respeten las reglas de negocio en todo momento y no exista una afectación en la información persistida.

De igual forma, se buscó durante el diseño de la solución que los costos temporales y espaciales sean los mínimos posibles. De esta manera se vela por un funcionamiento fluido cuando se debe manejar concurrencia a altos niveles (como en este caso, donde se plantea una sola solución con implementación de carácter nacional).

Distribución

La arquitectura y modelaje de la aplicación se hizo para que todas las clases giren alrededor de la clase de PlanDeVacunación. La ramificación de una clase principal permite que esta tenga acceso a todas las demás y conecte principalmente con la base de datos para guardar, actualizar, eliminar, obtener y distribuir en general la información desde y hacia un servidor. En adición, permite cosas como la flexibilidad a cambios futuros (solicitud explícita dentro de las reglas de negocio) y la implementación localizada de los RF y RFC.