

Registro Universal de Prestaciones de Salud (RUP). Un modelo recursivo y dinámico utilizando tecnología NoSQL y Angular

Lic. Marcelo D. Carrascal, Lic. Silvina Roa, A.S. Manuel Urbano

Abstract—Debido a la gran cantidad de usos que se hace de los registros médicos, se han hecho innumerables intentos para estandarizarlos y esos intentos causaron que los mismos se vuelvan rígidos, llevándolos a un concepto antagónico y a una enemistad casi irremontable frente al dinamismo que naturalmente tienen los procesos del cuidado de la salud y, por transitividad, los registros que los modelan. La propuesta de este trabajo es aprovechar las características cíclicas de todo proceso asistencial, y comprendiendo el dinamismo de los mismos, proponer un modelo recursivo y dinámico de representación de todas las prestaciones de un sistema de salud.

IndexTerms— Angular, NoSQL, dinámico, modelo, recursión, SNOMED CT.

1 INTRODUCCIÓN

Este trabajo surge a partir de a) la comparación de procesos de atención de la salud que en apariencia no tienen puntos en común e incluso parecieran tener lógicas diferentes, y b) el análisis de multiplicidad de intentos de estandarizar los registros de salud.

Luego de años de experiencia en el desarrollo de sistemas informáticos para registrar información de salud surgen dos grandes hipótesis: a) Todos los procesos de atención de la salud se basan en la misma lógica cíclica; b) La información de salud es inherentemente compleja ya que consiste de miles de ideas modeladas en una multitud de formas determinadas por los episodios de la vida de la gente, las familias y las comunidades, la percepción de los miembros del equipo de salud, los pacientes, incluso la política, las interconexiones anatómicas, fisiológicas, psicológicas y culturales y las historias de lo que la gente dice acerca de sus vidas y su salud. Por lo tanto, el **dinamismo** y **no la estandarización** es lo que prevalece en los registros. En base a estas dos premisas desarrollamos un modelo recursivo y dinámico de registros de salud que permite representar cualquier proceso de atención de la salud. Desde la prescripción de un medicamento hasta la incorporación de un paciente a un programa de trasplante. Dado el nivel de abstracción de este modelo, lo hemos denominado Registro Universal de Prestaciones (RUP).

2 DESCRIPCIÓN DEL MODELO

2.1 Recursión

A partir de las dos grandes hipótesis ya planteadas realizamos un análisis de los cuatro grandes procesos de atención básicos: a) atención ambulatoria, b) internación, c) atención de emergencia, d) prestaciones Intermedias. Si se piensan desde el punto de

vista de cada uno de los procesos como una entidad aislada, es posible encontrar diferencias entre los mismos. Ejemplos de estas diferencias hay muchos: a) en cuanto a su duración, un episodio de atención ambulatoria pareciera ser mucho más corto que un episodio de internación b) en cuanto a su complejidad un proceso quirúrgico es mucho más complejo que una prestación intermedia c) a pesar de que el proceso de emergencia y el de internación parecieran ser similares, las prácticas efectuadas y su nivel de urgencia difieren notablemente. Tradicionalmente se ha pensado a los procesos médicos desde sus diferencias, por lo que a partir de nuestra primera hipótesis nos planteamos ponderar las similitudes.

Propusimos la siguiente definición: *Un proceso de atención de la salud es un conjunto de uno o más episodios en los que al menos un miembro del equipo de salud realiza una prestación a un paciente*

Esta simple definición, nacida a partir de la visión de los procesos de atención de la salud desde un nivel de abstracción más alto, hace desaparecer las aparentes diferencias. De hecho, tanto los procesos de atención ambulatoria, internación, emergencias, prestaciones intermedias, así como cualquier otro, se ven reflejados perfectamente en ella. Sin embargo, podríamos decir que no quedan evidenciadas las “diferentes complejidades” de los procesos en la nueva definición. Cualquier persona puede argumentar con razón, por ejemplo, que el proceso de atención ambulatoria parecería ser “más sencillo” o que “consta de episodios aislados” frente a un proceso de internación. Sin embargo, podemos decir que en todo proceso de atención de la salud la secuencia para ejecutarlo es la siguiente:

- Existe un requerimiento para realizar una prestación (alguien la solicita).
- Si el paciente accede a la prestación se lo valora, es decir, se toma una fotografía parcial y normalmente sesgada de la situación de su salud en el primer contacto con el equipo de salud (valoración).
- Se realizan pruebas diagnósticas (interrogatorios, toma de signos vitales, imágenes, laboratorios, etc.) para generar un conjunto de hipótesis (lista de signos, síntomas, diagnósticos, etc.).
- Se realizan indicaciones diagnósticas, terapéuticas,

Lic. Marcelo D. Carrascal, Autor. Departamento de Tecnologías de la Información del Hospital Provincial Neuquén. Neuquén, Argentina E-mail: mcarrascal@hospitalneuquen.org.ar

Lic. Silvina Roa, Autor Jr. Dirección General de TICS. Subsecretaría de Salud de la Provincia de Neuquén. Neuquén, Argentina E-mail: silviroa@gmail.com
A.S. Manuel Urbano Stordeur, Autor Jr. Servicio de Desarrollo de Software, Hospital Provincial Neuquén, Neuquén, Argentina E-mail: murbano@hospitalneuquen.org.ar

- educativas, etc. para tratar las hipótesis.
- Se evalúan los resultados de dichas indicaciones.
- Se evoluciona la situación de salud del paciente.

Si probamos este pequeño y sencillo pseudo algoritmo veremos que modela a *todos los procesos* de atención de la salud casi a la perfección, sin importar su complejidad. En consecuencia, podemos ensayar una nueva definición utilizando el concepto de **recursión**.

Previo a esto haremos unos pequeños ajustes semánticos a nuestra definición anterior. Diremos entonces que: *Una prestación de salud es cualquier episodio en el que al menos un miembro del equipo de salud realiza alguna acción relacionada con la salud de un paciente.*

La palabra prestación reemplaza al concepto de proceso de atención de la salud.

Llamaremos también **entidad observable** a todo campo de un registro de salud que pueda ser utilizado para codificar elementos de una lista o cualquier elemento al que pueda asignarse un valor.

Si refinamos a nuestro algoritmo y lo convertimos en una definición recursiva de prestación obtenemos que:

Una prestación de salud consta de dos grandes partes:

1. Una solicitud
2. Una ejecución

a su vez la ejecución puede ser:

- a) Un conjunto de entidades observables, o
- b) Un conjunto de entidades observables más un **conjunto de prestaciones** (que se realizan en ese mismo momento o se planifican a futuro)

Si vemos a cualquier proceso de la salud a través de esta definición entonces, un episodio de atención ambulatorio es una prestación, un episodio de internación es una prestación, un episodio de emergencias es una prestación, una tomografía es una prestación, la toma de signos vitales es una prestación, la prescripción de un medicamento es una prestación, etc.

En esta primera parte hemos reducido todos los procesos de atención de la salud a una definición recursiva única.

2.1.1 Tipos de prestaciones y granularidad

De la definición recursiva y por una cuestión de tratamiento y complejidad, diferenciamos dos grandes tipos de prestaciones.

a) **Prestaciones atómicas:** aquellas que constan de un conjunto de entidades observables cuyos subconjuntos aislados dejan de tener sentido. Una prestación atómica es considerada indivisible y tiene significado propio. Ej.: podemos hablar del átomo de tensión arterial sistólica, del átomo de tensión arterial diastólica, del átomo de temperatura corporal, etc. b) **Prestaciones moleculares:** aquellas que son agregados de entidades observables propias y de otras prestaciones. Ej.: Podemos hablar de la molécula de tensión arterial que consta de un átomo de tensión arterial diastólica y un átomo de tensión arterial sistólica y podemos hablar de la molécula de signos vitales que consta de una molécula de tensión arterial, un átomo de temperatura corporal, un átomo de saturación de O₂, etc.

Es en el conjunto de prestaciones moleculares en que se ve claramente su naturaleza recursiva.

Un beneficio adicional de estas definiciones, es el alto grado

de reusabilidad de los elementos del modelo. Cada átomo se define una sola vez y puede ser reusado en sinnúmero de moléculas en las que sea necesario registrar su valor asociado. Lo mismo pasa con algunas moléculas.

2.2 Dinamismo

El proceso de salud de un paciente, representado aquí por un conjunto de prestaciones, no es un asunto sencillo. Es algo completamente dinámico que depende de una multiplicidad bastante extensa de factores que requieren adaptabilidad al paciente y sus problemas, a los miembros del equipo de salud intervinientes, etc. Los buenos registros deben facilitar la continuidad del cuidado, deben estar bien organizados sin perder al paciente como eje, los datos deben ser pertinentes a los problemas de salud tratados y a los procedimientos y valores observados durante la ejecución de cada prestación. Este proceso dinámico requiere de registros. Esos registros, relacionados y organizados de alguna manera, se convierten en lo que conocemos como la **Historia de Salud** del paciente. A partir de esa Historia de Salud, es posible obtener información sobre la salud de la persona, la evolución de sus problemas y como resultados colaterales obtenemos información estadística, epidemiológica, de investigación y de gestión.

Dada la gran cantidad de usos y la importancia de estos registros, históricamente, se han hecho infinidad de intentos por estandarizarlos, ya que deberían poder reflejar los distintos tipos de información que los diferentes actores requieren. La *estandarización es sinónimo de rigidez*. Y la rigidez de los registros puede convertirlos en elementos antagónicos, incluso llegando a una enemistad casi irremontable, del dinamismo que requiere la recolección de datos en los procesos asistenciales. Cuando hablamos de rigidez nos referimos a su tratamiento en forma de "formulario" o "planilla" heredada de los registros en papel. Esto implica que los registros están compuestos por un conjunto estático de campos que deben llenarse, tratando así de uniformar las prestaciones. Esto tiene como resultado que, de alguna manera, se intente uniformar el proceso de atención. Los registros no diferencian las particularidades de los pacientes. Ante esto los miembros del equipo de salud recurren a los campos de texto libre y los campos estandarizados no son utilizados. Como demostración de esta hipótesis hicimos un estudio de los registros de atención ambulatoria de pacientes de las especialidades de clínica médica y pediatría del Hospital Provincial Neuquén y llegamos a los siguientes resultados:

El registro de consulta de clínica médica cuenta con 10 variables consensuadas con los médicos, para ser completadas. Se evaluaron 7324 registros y se observa que en promedio el 75,2% de las variables de cada registro están vacías. En las consultas de pediatría, sobre 10 variables consensuadas para ser registradas, se evaluaron 11825 registros y el promedio de variables vacías asciende a un 94,2%.

Estos números confirman nuestra hipótesis, es decir, que el registro electrónico tradicional no sólo no refleja las necesidades "particulares" de recolección de la información de acuerdo a las características del paciente, ni a las necesidades de cada miembro del equipo de salud.

Es por eso que la propuesta fue desarrollar registros de carga dinámica en ejecución. Esto significa que, dependiendo de los problemas y las características del paciente, los tipos de prestaciones que se realicen sobre ese paciente y las necesidades de cada miembro del equipo de salud, se podrán seleccionar en el momento de la ejecución las variables que se desea registrar.

3 DESAFÍOS

Dado que los procesos relacionados con la salud están normalmente muy alejados de la linealidad a los que se los pretende forzar, nuestro objetivo fue, en base a la definición recursiva de prestación y a la hipótesis del dinamismo y la no linealidad de los procesos asistenciales *desarrollar un modelo recursivo y dinámico para la generación de registros electrónicos de salud*.

Fue necesario seleccionar metodologías y tecnologías apropiadas para poder realizar este desarrollo. Tanto el front-end como el back-end debían poder implementar de una manera sencilla, ordenada y mantenible la plasticidad propia de los registros.

3.1 NoSQL

Particular atención se puso sobre el motor de base de datos, eligiéndose el paradigma NoSQL por su diferencia más destacada frente a SQL, que presentaba la tradicional "rigidez" del diseño de sus registros. En el paradigma SQL, no es posible almacenar en una misma tabla información de formato diferente o insertar un tipo de dato cuando se espera otro.

Dado que este proyecto tiene como características primordiales:

- Requerimientos y evolución de datos indeterminados y sin relación aparente.
- El modelo abstracto es simple y de rápida codificación.
- Almacenar estructuras dinámicas que puedan ser escalables y procesadas con gran velocidad.

NoSQL se presentó como el paradigma perfecto para el modelado ya que:

- Documentos similares pueden ser almacenados en una misma colección, pero a diferencia de SQL es posible almacenar cualquier tipo de dato en cualquier documento.
- Es posible agregar datos en cualquier lugar y en cualquier momento. No hay necesidad de diseñar de antemano el documento, ni tampoco la colección.
- La denormalización que permite, permite accesos de gran performance para lectura de documentos conteniendo estructuras o datos disímiles.

En particular, este proyecto fue desarrollado utilizando MongoDB y la herramienta de modelado Mongoose que corre en el ambiente de ejecución de Node.JS.

3.2 Web Components

Angular es un framework de código abierto para desarrollar aplicaciones para computadoras y dispositivos móviles basados en el lenguaje Typescript. En la versión más reciente de Angular (2.x y 4.x) los bloques fundamentales para construir aplicaciones son los componentes, que encapsulan la funcionalidad para visualizar y manipular conjuntos de datos. Como veremos luego, Angular permite la creación de componentes dinámicos en tiempo de ejecución por lo que permite modelar rápidamente las estructuras requeridas para los registros médicos.

4 IMPLEMENTACIÓN

4.1 Recursión

El modelo se implementa primariamente a través de una co-

lección maestra de tipos de prestación, las cuales se hayan mapeadas a conceptos de SNOMED CT. Esta colección **es maestra** ya que los elementos que la conforman rara vez se modifican y son únicos. Esto implica que si un tipo de prestación está definida como un elemento de la colección maestra, ese elemento se reusará sin modificación en cualquier lugar en que sea requerido.

Se planteó un esquema recursivo para modelar los tipos de prestación. Cada tipo de prestación requiere la ejecución de uno o más tipos de prestaciones. La figura 1 muestra un esquema básico de Mongoose que representa a cualquier tipo de prestación.

Mediante este esquema se plantea la reutilización desde el nivel más elemental de registro, pudiendo ser este un *átomo*, una *molécula*, una *fórmula de cálculo*, etc. Los átomos se utilizan para registrar valores simples (entidades observables). En caso de que un tipo de prestación requiera agrupar conjuntos de tipos de prestaciones, la misma se registrará como molécula y su definición será recursiva. Para lograr esto se mantiene un arreglo de ejecución que contiene las referencias a los tipos de prestaciones asociados. De esta manera se simplifica el registro de datos y se maximiza el reuso.

```

tipoPrestacionSchema = {
  key:String,
  nombre:String,
  descripcion:String,
  codigo:[codificadorSchema],
  autonoma:Boolean,
  solicitud:[{
    type:mongoose.Schema.Types.ObjectId,
    ref:'tipoPrestacion'
  }],
  ejecucion:[{
    type:mongoose.Schema.Types.ObjectId,
    ref:'tipoPrestacion'
  }],
  tipoProblemas:[{
    type:mongoose.Schema.Types.ObjectId,
    ref:'tipoProblema'
  }],
  activo:Boolean,
  componente: {
    ruta:String,
    nombre:String
  },
  granularidad:String,
  tipo: {
    type:String,
    enum:['atomo','molecula','formula']
  }
}

```

Figura 1

Las fórmulas, al igual que las moléculas, almacenan las mismas referencias a otros tipos de prestaciones. La diferencia radica en que las fórmulas además realizan un monitoreo constante de la disponibilidad de los parámetros referenciados y cuando todos los valores de sus parámetros de entrada están disponibles para resolver la fórmula, ésta es calculada y registrada.

Podemos ver gráficamente el esquema en la figura 2 y un ejemplo en la figura 3

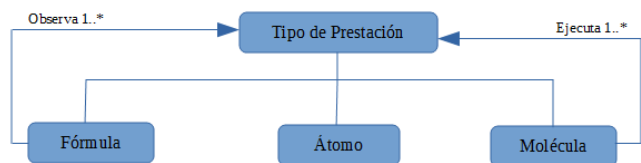


Figura 2

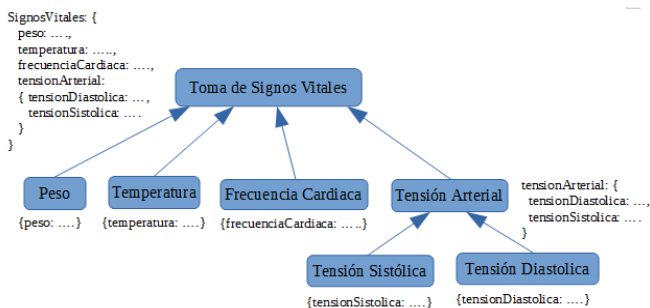


Figura 3

Utilizar una base de datos de tipo NoSQL nos permite almacenar los registros de los tipos de prestaciones con una estructura *clave: valor*. Una clave identifica unívocamente a un tipo de prestación; de esta manera se simplifican las búsquedas ya sea que los registros se almacenen como átomo o formando parte de una estructura más compleja como son las fórmulas o las moléculas.

4.2 CARGA DINÁMICA

La utilización de Angular 2.x/4.x para la implementación de este proyecto permitió dividir todos los tipos de prestaciones en pequeñas componentes autónomas y reusables. Cada una de estas componentes tiene su propia lógica, pero es capaz de interactuar con otras componentes para conformar una estructura más compleja. Este tipo de arquitectura aporta la flexibilidad para generar los registros de prestaciones utilizando la recursión y la carga dinámica tal como lo exigían las premisas iniciales del proyecto.

Para el front-end utilizamos el servicio *ComponentFactory Resolver* provisto por Angular que permite inyectar componentes en tiempo de ejecución. Se implementó una clase base abstracta, que mantiene una estructura con todos los componentes disponibles e inicializa en tiempo de ejecución aquellos que sean requeridos por el usuario (lazy loading). De esta clase base se extienden las nuevas clases que modelan átomos, moléculas y fórmulas e implementan particularidades funcionales de estos tipos de prestaciones. Al momento de modelar un átomo, se extienden estas clases intermedias y en sus componentes se ubica únicamente la lógica de validación, mensajes de retorno, y en la interface visual se resuelve qué y cómo mostrar los elementos (elementos HTML, gráficos, mensajes informativos, etc)

CONCLUSIONES

Hemos modelado ya gran cantidad de prestaciones siguiendo este modelo. Podemos nombrar algunas de ellas: Consulta Ambulatoria de Medicina General, Consulta de Control de Niño Sano, Internación, Atención de Emergencias, Tomografías, Mamografías, Consultorio de Piel, Plan de Indicaciones, Control de Ulceras por Presión, Control de Caídas, Score de NEWS, Escala del Dolor, etc. Todas ellas, sin importar su complejidad han podido ser modeladas sin inconvenientes. Pueden enumerarse algunas de las ventajas que hemos encontrado hasta ahora en la utilización de este modelo:

- Una drástica simplificación en el análisis de los requerimientos:** debido a que todo lo que se hace durante el proceso de atención de la salud es ahora considerado un tipo de prestación y su implementación es la misma. Cualquier proceso (tipo de prestación), aun los más complejos, puede descomponerse recursivamente en tipos de prestaciones más simples.
- Aumento del grado de participación de los usuarios finales en el proceso de análisis y modelado:** dado que el modelo es sencillo de comprender, todos los usuarios finales (aquellos que usarán los sistemas) que han participado del análisis de los procesos, se han visto más involucrados y han podido reflejar con mucha mayor claridad sus necesidades.
- Notable aumento de la reutilización de componentes:** dado que el análisis de los datos se hace a nivel atómico (de acuerdo a la definición en este trabajo), aquellas componentes que ya están modeladas, no se vuelven a analizar. Con el correr del tiempo y a medida que la colección maestra de tipos de prestación comienza a popularse, es sinónimo de que cada elemento de la colección podrá ser reutilizado innumerable cantidad de veces en el lugar en el que sea necesario.
- Disminución de los tiempos de implementación:** Debido a que se ha invertido mucho tiempo en la construcción de las estructuras subyacentes y en el análisis del modelo, esto se ve reflejado en que la construcción de cualquier tipo de prestación nuevo puede hacerse muy rápidamente. Las dos razones fundamentales son: 1) En el momento del análisis, todos los tipos de prestación se “deconstruyen” en porciones más pequeñas siguiendo el paradigma recursivo 2) La mayoría de los nuevos tipos de prestación se pueden construir a partir de otros tipos de prestación más simples que ya están implementadas.
- Aumento de los niveles de estandarización de los datos:** a partir del reuso de los componentes podemos hablar no solamente de que se ha logrado una estandarización en los tipos de datos sino también una estandarización semántica. Cada tipo de prestación, no importa dónde sea utilizada contiene los

mismos datos, el mismo formato y el mismo significado. Debemos notar aquí la diferencia entre estandarizar el registro de una consulta y estandarizar la semántica de los datos. En este segundo caso la estandarización es una herramienta muy útil a diferencia del primero. La estandarización de los datos hace que su significado no se modifique, independientemente de dónde sean utilizados.

- f) **Facilita la reconstrucción de la historia de cualquier variable de salud, sin importar el proceso en el que se haya registrado:** dado el grado de reuso y los niveles de estandarización semántica logrados, la reconstrucción de cualquier variable simplemente implica recorrer las colecciones para ubicarla y mostrarla de la manera que al usuario le sea más conveniente. Es posible, además, dado que sabemos el episodio que dio origen a cada valor atómico, filtrar partes de la historia de esa variable de acuerdo a las necesidades. Ej.: Si queremos mostrar la historia del peso de una paciente podemos hacerlo en forma muy simple buscando todos los átomos de peso que se hayan registrado en cualquier tipo de prestación (atención ambulatoria, internación, control, etc. y es posible tabularlo o graficarlo según las necesidades de quien deba mirarlo. Supongamos que se quiere obviar el peso de la paciente durante su embarazo es posible filtrar y no visualizar esas mediciones cuyo origen tengan que ver con las prestaciones asociadas al control de embarazo.
- g) **Gracias a la carga dinámica,** el volumen de datos nulos o vacíos tiende a cero, ya que en cada episodio de la ejecución de un tipo de prestación se registran los datos seleccionados por cada miembro del equipo de salud de acuerdo a la lista de problemas del paciente, a lo que se trate en ese episodio y a lo que se planifique para el futuro.
- h) **Cada miembro del equipo de salud tiene la capacidad de elegir qué registrar sin afectar las necesidades de otros miembros del equipo.** Es decir que los registros electrónicos pueden construirse personalmente para cada paciente y para cada miembro del equipo de salud sin la necesidad de desarrollar un módulo o una interface especial.
- i) **Personalización y Estandarización:** Es posible construir los registros en forma personalizada seleccionando en forma dinámica los átomos, moléculas o fórmulas. Sin embargo, debido a que los datos se eligen de la colección maestra de prestaciones, pueden considerarse estándares por lo que su tratamiento para obtener información estadística, epidemiológica, de gestión o de investigación puede ser realizada sin problemas de ningún tipo.
- j) **Desde el punto de vista del almacenamiento,** este se optimiza ya que dejan de existir las matrices ralas

del modelo SQL y son reemplazadas por colecciones completamente pobladas en el modelo NoSQL.

A través de este trabajo hemos logrado modelar registros electrónicos de salud que se alejan de la forma rígida y sean capaces de reflejar el concepto que planteamos al principio y que volvemos a transcribir aquí: *“La información de salud es inherentemente compleja ya que consiste de miles de ideas modeladas en una multitud de forma determinadas por los episodios de la vida de la gente, las familias y las comunidades, la percepción de los miembros del equipo de salud, los pacientes, incluso la política, las interconexiones anatómicas, fisiológicas, psicológicas y culturales y las historias de lo que la gente dice acerca de sus vidas y su salud”.*

AGRADECIMIENTOS

Lic. Marcelo D. Carrascal, Autor agradece al Lic. Juan F. Gabriel, a la Dra. Laura Monteverde y al Dr. Matias Neira por su participación y aportes, al Hospital Provincial Neuquén, por brindar el tiempo y la libertad profesional para trabajar en este proyecto; al Lic. Julio Santarelli, Ing. Leandro Lambertucci, Sr. Andrés Velazquez, Tec. Fernando Sastre, Ing. Edgardo Kristensen, Lic. Natalia Huenchuman por su inestimable colaboración.

REFERENCIAS

- [1] David Markwell, IHITDSO, Why Clinical Terminology Matters, SNOMED CT Foundation Course. 2014
- [2] Craig Buckler, SQL vs. NoSQL: The Differences, Sept 18 2015, Sitepoint.com
- [3] Patrik Szpisják, What are master tables in a database, Feb 2017, Quora.com
- [4] Margaret Rouse, Markup, Whatis.com
- [5] Thomas Robey, MD, PHD, The art of writing patient records notes, Julio 2011, Virtual Mentor, Volume 13, Number 7
- [6] Sandeep Kumar Pate, Learning Web Component Development; Mayo 2015, O'Reilly
- [7] Thomas G. Gutheil, MD, Fundamentals of Medical Records Documentation, Nov 2004, Psychiatry.