



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**“ESPECIFICACIÓN DE ESQUEMA INFORMÁTICO PARA MÉTODOS Y PRÁCTICAS
DE KUALI-BEH PARA FACILITAR LA VERIFICACIÓN SEMIAUTOMATIZADA DE
SUS CARACTERÍSTICAS Y PROPIEDADES”**

**TRABAJO DE TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA DE LA COMPUTACIÓN**

PRESENTA:

ROBERTO AZRAEL MEDINA DÍAZ

TUTORA:

HANNA JADWIGA OKTABA
FACULTAD DE CIENCIAS, UNAM

MÉXICO, D.F. SEPTIEMBRE 2015



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

DEDICATORIA

A mis padres: Roberto y Lilia

AGRADECIMIENTOS A

Dra. Hanna Jadwiga Oktaba

Dr. Miguel Ehécatl Morales Trujillo

UNAM, PCIC y CONACYT

Contenido

| | |
|--|----|
| Introducción | 7 |
| Problema..... | 7 |
| Objetivo..... | 8 |
| Estrategia | 9 |
| Resultados esperados | 9 |
| Estructura del documento | 10 |
| Capítulo 1 – KUALI-BEH | 11 |
| ¿Qué es KUALI-BEH? | 11 |
| El origen..... | 11 |
| Los conceptos..... | 12 |
| Ejemplo de aplicación | 15 |
| Capítulo 2 – Situación actual de esquemas informáticos | 21 |
| Perspectiva..... | 21 |
| Revisión Sistemática | 22 |
| Justificación | 22 |
| Preguntas de investigación | 22 |
| Estrategias de selección de fuentes de información | 22 |
| Criterios de selección | 23 |
| Estrategia de extracción de la información | 23 |
| Estrategia de síntesis de información | 23 |
| Resultados | 24 |
| Conclusiones de la revisión sistemática..... | 25 |
| Otras revisiones | 25 |

| | |
|---|----|
| Conclusiones del capítulo 2 | 28 |
| Capítulo 3 – Creación del Esquema XML para KUALI-BEH | 29 |
| KBL, un nuevo lenguaje para KUALI-BEH..... | 29 |
| Conceptos KUALI-BEH, para base del nuevo esquema | 29 |
| Decisiones relevantes de diseño para esquema KUALI-BEH.xsd..... | 30 |
| Aspectos XML en consideración para KBL | 32 |
| Alternativa de gramática de KBL..... | 32 |
| Patrón de diseño de la gramática de KBL | 33 |
| Esquema XSD para KUALI-BEH..... | 36 |
| Código fuente del esquema XSD..... | 37 |
| Guía de creación de archivos *.KBL..... | 40 |
| Ejemplo de archivo *.KBL | 43 |
| Conclusiones del capítulo 3 | 49 |
| Capítulo 4 – Experimento de comprobación del nuevo esquema para KUALI-BEH | 50 |
| Planteamiento del experimento de validación..... | 50 |
| Kualibs 1.0: herramienta de software para validar el lenguaje propuesto. | 51 |
| Especificaciones generales de la herramienta de software de comprobación | 53 |
| Características de manejo de conceptos KUALI-BEH dentro de Kualibs 1.0 | 53 |
| Características de uso de la herramienta Kualibs 1.0 | 54 |
| Características Técnicas de la Herramienta Kualibs 1.0: | 54 |
| Verificación de Propiedades de Métodos con Kualibs 1.0 | 57 |
| Caso de prueba reducido..... | 58 |
| Representación gráfica del caso de prueba reducido..... | 59 |
| Estructura del caso de prueba reducido | 59 |

| | |
|--|----|
| Resultado de Verificación | 60 |
| Conclusiones de capítulo 4 | 66 |
| Capítulo 5 – Resultados | 67 |
| Conclusiones | 67 |
| Trabajo futuro | 68 |
| Referencias | 70 |
| Índice de Figuras..... | 72 |
| Índice de Tablas | 72 |
| Apéndice A, Guía Rápida para uso de Kualibs 1.0 | 73 |
| Ejemplo con Scrum | 73 |
| Proceso de Descripción de Práctica de Ejemplo..... | 74 |
| Ingresar y conocer las opciones del menú del sistema Kualibs | 75 |
| Definir Actividades | 77 |
| Definir Conocimientos y Habilidades | 79 |
| Definir Criterios de Verificación | 79 |
| Definir Métricas..... | 81 |
| Definir Características | 81 |
| Definir Productos de Trabajo | 83 |
| Definir Actividades (Complemento)..... | 86 |
| Definir Práctica | 88 |
| Definir Guía | 89 |
| Imprimir Guía de la Práctica..... | 91 |
| Definir Método..... | 93 |

Introducción

La ingeniería de software se dedica a la creación y mantenimiento de software, disciplina que evoluciona día con día, pues sus aplicaciones cada vez son más sofisticadas y requeridas en diferentes ámbitos. El software no solo incluye la codificación de instrucciones de computadora en algún lenguaje de programación, también abarca documentación, capacitación y más; es por eso que la ingeniería de software es practicada por grupos de personas y equipos de trabajo en conjunto, lo que complica aún más su alcance, porque esta disciplina también abarca la creación y adaptación de los métodos utilizados por las personas involucradas, y la forma en que se organizan para obtener los resultados requeridos.

Problema

Es evidente que los equipos que crean o dan mantenimiento al software, logran el resultado a través de sus conocimientos y habilidades en conjunto; el “método” existe, aunque sea de manera empírica, pero no ha podido ser organizado de forma estándar, y tampoco ha sido repetido de manera sostenida, dada la dinámica de los grupos de personas; es por eso que es una de las áreas de la ingeniería de software con mayor oportunidad de mejora, pues es complicado implementar un proceso o un método a esos grupos de personas.

Dada la situación anterior, en la industria del software han surgido varias propuestas para la organización del trabajo y de las personas que lo realizan; algunas propuestas han tenido mejor aceptación que otras, pero como ya se mencionó con anterioridad, el software y su industria están en constante evolución. Es por eso que, al existir una persistente necesidad en la industria, e incluso, un llamado de la misma industria a cubrir esta necesidad, pueden surgir propuestas nuevas.

KUALI-BEH es una de esas propuestas, surgida desde la academia, en respuesta a un llamado de la industria de software.

La teoría desarrollada en KUALI-BEH, es factible de ser puesta en práctica por diferentes herramientas de software, y de hecho, el mismo grupo de investigación que la propuso ya ha desarrollado software que apoya su uso; sin embargo, una de las intenciones originales en KUALI-BEH, es que se puedan compartir los métodos, por una comunidad de equipos de desarrollo, y

aunque todavía no existen herramientas para apoyar el uso de KUALI-BEH, sería deseable que así sea.

Las potenciales herramientas de soporte al uso de KUALI-BEH pueden ser de diferentes tipos, como software libre, o de pago; de una o varias plataformas; grandes, o pequeñas; especializadas en algún aspecto, o generales; desarrollado por algún fabricante externo o por equipos internos de cada empresa. Asimismo, al igual que mucho del software que existe hoy en día, para estas herramientas sería deseable que la información que manejen la pudieran compartir, y ser compatibles en la estructura de datos que manejan. Esto permitiría compartir funciones de herramientas que trabajen con KUALI-BEH, funciones en las que algunos quizá sean especialistas; por ejemplo, una función podría ser la verificación de propiedades de los métodos. Tal vez en el futuro, algunos sistemas puedan usar técnicas de reconocimiento de lenguaje natural o algunos podrían usar lenguajes formales; en cualquier caso, para contribuir a que las herramientas puedan apoyar cada vez más el uso de la teoría de KUALI-BEH, y por lo tanto, contribuir a la organización de equipos de trabajo en la industria de software, es necesario definir un medio que pueda adoptarse por más de uno de esos posibles sistemas, con la intención de ampliar la distribución de métodos, definidos con KUALI-BEH, entre la comunidad.

Objetivo

La intención de este trabajo de tesis es definir un esquema que permita empacar la información concretada mediante KUALI-BEH, y que pueda ser considerado como un estándar por las herramientas que utilicen dicha información. De esta forma, las herramientas de apoyo podrán desarrollar funciones independientes de la información; por ejemplo, la verificación de propiedades de métodos, y esas herramientas podrán usar dicho esquema, al compartir información entre ellas, con otras herramientas que no tengan esa función, o que tengan otras funciones distintas, o que las implementen de manera diferente, pero todas usando el mismo esquema compatible, una esquema de información de métodos, que pueda usarse de manera amplia por diferentes tipos de sistemas.

Para aclarar un poco más el objetivo, puede servir una analogía; por ejemplo, en Internet existen diferentes tipos de aplicaciones que trabajan sobre un mismo tipo estructurado de

información, algunas aplicaciones usan esa estructura solo para interpretarla y presentarla al usuario, y aun solo presentándola, diferentes aplicaciones la presenten de diferente manera. Otras aplicaciones utilizan la estructura para generar la información, otras la utilizan para permitir a los usuarios modificar esa información; esas aplicaciones son navegadores, buscadores, servidores, editores web, y la información que manejan son documentos web, que tienen una estructura en común: encabezado, cuerpo, párrafos, títulos, listas, imágenes, enlaces, y demás. Y el esquema utilizado, sin duda muy difundido, es el HTML. De manera análoga, la información que pretendemos que se maneje de KUALI-BEH, son métodos y prácticas, entre otros, y el esquema deseado es lo que sería HTML a los páginas web.

Estrategia

Los pasos a seguir para alcanzar el objetivo, son:

1. Revisión de KUALI-BEH para comprender los elementos que lo componen y que serían los manejados mediante el nuevo esquema propuesto.
2. Seleccionar la tecnología a utilizar para definir el esquema como el esperado; se tienen que escoger los elementos a los que se les dará importancia y que deberían cubrir el nuevo esquema.
3. Definición del esquema, de acuerdo con la tecnología seleccionada y los elementos de KUALI-BEH.
4. Para comprobar la factibilidad de uso del nuevo esquema, se debe desarrollar una herramienta mínima de captura de información KUALI-BEH, que maneje la información de métodos y prácticas en un repositorio interno, que facilite la verificación de la propiedad de consistencia de los métodos, y que además pueda exportar e importar la información de un método completo, utilizando el esquema propuesto.

Resultados esperados

Se espera el reporte de pruebas del uso de herramienta, constatando los dos requisitos primordiales: exportar e importar información KUALI-BEH y el apoyo para la verificación de la propiedad consistencia de los métodos.

Una vez comprobada la función correcta de la herramienta, se tendrá una definición válida del esquema, que podrá ser usado por cualquier interesado en desarrollar herramientas de apoyo para KUALI-BEH.

Estructura del documento

Para delinear el alcance que aporta este trabajo de tesis es necesario describir, aunque de manera resumida, lo que es KUALI-BEH, ya que el producto obtenido al final de este trabajo es una pieza que apoya la puesta en práctica de la teoría desarrollada en ese proyecto; es por eso que en el primer capítulo se plantea, con un enfoque práctico, los conceptos de KUALI-BEH que se usarán a lo largo de esta obra.

Las bases con que se selecciona la tecnología a utilizar en la propuesta de esta tesis, se exponen en el capítulo dos, de manera especial, el reporte de la revisión sistemática llevada a cabo.

Para el capítulo tres se presenta el desarrollo del nuevo esquema, dejando constancia de las decisiones que se tomaron para llegar a ese diseño, ya que serían muy útiles en caso de que KUALI-BEH mismo o el esquema requieran una evolución.

En el capítulo cuatro se presenta la construcción de la herramienta a utilizar, como experimento de verificación de factibilidad de uso del nuevo esquema.

En el último capítulo, se presentan las conclusiones finales de este trabajo de tesis.

Capítulo 1 – KUALI-BEH

¿Qué es KUALI-BEH?

El origen

Un buen inicio para comprender lo que es KUALI-BEH¹, es un enfoque sobre su origen y los objetivos a cubrir por parte de sus autores, ya que, si bien el proyecto nace como respuesta a una convocatoria, por una organización internacional en la industria de software, la teoría involucrada contiene elementos de otras contribuciones a la ingeniería de software, que se hicieron durante años de trabajo, por parte del grupo de investigación que lo desarrolla en México, reconocido hoy en día por sus aportes en estándares nacionales e internacionales.

En el año 2011, la OMG² hizo un llamado a la comunidad a través de la publicación de su documento FACESEM RFP (*A Foundation for the Agile Creation and Enactment of Software Engineering Methods, Request for proposal*) (OMG, 2011), con el objetivo de crear una base estándar en la definición y ejecución de métodos de ingeniería de software, y que a diferencia de tratamientos anteriores en esta disciplina, la nueva base debería estar dirigida, de manera primordial, a las personas que ejecutan de manera activa la ingeniería de software, más que a los ingenieros de proceso, es decir, dirigida más a los que practican que a los teóricos.

Una de las propuestas que atendieron el FASECM RFP, fue KUALI-BEH, desarrollada en la UNAM, por el grupo de investigación de procesos de software, ahora llamado KUALI-KAANS³ y liderado por la Dra. Hanna Oktaba. Por supuesto, KUALI-BEH no fue la única propuesta; sin embargo, después de un largo proceso de evaluación y adecuación, en noviembre del 2014, la OMG publicó la versión 1.0 del nuevo estándar conocido como ESSENCE **Fuente especificada no válida.**, del cual, KUALI-BEH forma parte, como una extensión al núcleo de ESSENCE, y descrito en su anexo normativo B.

En el documento FACESEM RFP, donde la OMG hacía el llamado a la comunidad, se planteaba un problema que lleva décadas en la ingeniería de software, y es el siguiente: Por un

¹ Kuali es una palabra náhuatl que significa: bueno o apropiado. Beh es una palabra maya que significa: camino.

² OMG (Object Management Group) es un consorcio que agrupa tanto a fabricantes como a usuarios, para establecer estándares de tecnologías orientadas a objetos, tales como UML, XMI, CORBA, entre los más conocidos.

³ Kaans es una palabra maya que significa: educación.

lado, el desarrollo de software a través de métodos y procesos bien definidos, tiene claros beneficios y conviene utilizarlos, pero por otro lado, esos mismos métodos y procesos llegan a ser considerados tediosos, rígidos y hasta perjudiciales, cuando se aplican sin ciertas consideraciones. Y lo que sucede es que, por lo general, los que definen los métodos no son las mismas personas que los ejecutan día a día, y esto provoca descontento en los equipos de trabajo, porque cuando hay problemas, suele suceder que los planificadores culpan a los practicantes, y los practicantes a los planificadores de métodos y procesos.

Así que el requerimiento de la OMG, va más allá de generar un nuevo marco de trabajo para los equipos de desarrollo, ya que en lugar de definir un método estándar con sus herramientas de apoyo correspondientes, se busca más bien darles a los desarrolladores una manera para que ellos mismos definan y adapten sus propios marcos de trabajo, sus propios métodos y herramientas de apoyo, pero con un lenguaje común que les evite aislarse al ser incompatibles con otros marcos de trabajo.

KUALI-BEH 1.1 (Kuali-kaans, 2012), fue la versión incluida en ESSENCE 1.0, y cubría a cabalidad, los puntos del FACESEM RFP, incluyendo un conjunto de conceptos descritos más adelante; sin embargo, KUALI-BEH es un proyecto que continúa creciendo.

Los conceptos

Tomando como base el documento KUALI-BEH 1.1, en el cual se definen de manera formal estos términos, se puede considerar la siguiente idea general:

KUALI-BEH reúne los conceptos comunes que tienen que ver con los proyectos de software, e identifica cómo se relacionan entre sí, y los divide en dos conjuntos; los que componen la llamada Vista Estática, la cual permite definir prácticas y métodos que forman una infraestructura, y los que componen la Vista Operacional, que describen al método durante la ejecución del proyecto de software, sin importar el tamaño.

De acuerdo a KUALI-BEH, un método es: *Un conjunto coherente, consistente y suficiente de prácticas enlazadas, con un propósito específico, que satisface los requerimientos de los interesados, en determinadas condiciones* (Oktaba & Morales Trujillo, 2012).

Que el conjunto de prácticas sea *Cohérente* se refiere a que los objetivos particulares de cada una de las prácticas contribuyen para lograr el propósito general del método, es decir, que de entrada, no haya auto sabotaje en el método, haciendo actividades que vayan en contra del propósito general o desperdiciando tiempo en actividades improductivas.

Que el conjunto sea *Consistente* se refiere a dos cosas:

Primero, a que al menos una de las prácticas tome como entrada un elemento similar a la entrada del método, y también que al menos una de las prácticas tenga como resultado un elemento similar al resultado del método, por lo que se trata de asegurarse de que la entrada original del método será útil, y que siempre habrá un resultado del método, ya que debe existir una práctica que lo genere.

Y segundo, que las entradas de todas las prácticas (excepto por la entrada de la práctica del punto anterior) sean similares al resultado de alguna práctica, así como también, que los resultados de todas las prácticas (excepto por el resultado de la práctica del primer punto) sean similares a la entrada de alguna práctica, por lo que todas las entradas de las prácticas del método son generadas internamente, y que todos los resultados de las prácticas son útiles para el método o el resultado del mismo.

Que el conjunto sea *Suficiente*⁴ se refiere a que, con alcanzar el objetivo de cada una de las prácticas, sea suficiente para lograr el propósito del método, y así, asegurarse de que no falte ningún requerimiento por cubrir en el resultado final.

Una Práctica es: *Una guía de trabajo, con un objetivo específico, que recomienda cómo producir un resultado a partir de una entrada. La guía proporciona un conjunto de actividades sistemáticas y repetibles, enfocadas en alcanzar el objetivo y el resultado de la práctica. Los criterios de verificación asociados a los resultados son usados para determinar si se logró el objetivo. Se requieren conocimientos y habilidades particulares para seguir la guía de la práctica, que se ejecuta usando o no, herramientas. Para evaluar el desempeño y los objetivos alcanzados*

⁴ En versiones anteriores de KUALI-BEH, se utilizaba el término Completo, en lugar de Suficiente, pero por razones de formalismo del lenguaje, se hizo la nueva adopción.

por la práctica, se le pueden asociar ciertas métricas, que son estimadas y recogidas durante la ejecución de la misma (Oktaba & Morales Trujillo, 2012).

Hasta este punto se han mencionado los conceptos de mayor relevancia en la vista estática de KUALI-BEH, es decir, los que integran la también llamada “infraestructura de Métodos y Prácticas”; sin embargo, la manera en que se enlazan estos elementos, las prácticas y los métodos, es a través de sus entradas y resultados, por lo cual, los métodos requieren entradas, y producen resultados, y lo logran a través de prácticas, que de igual manera, requieren entradas y arrojan resultados; de hecho, el resultado del método, es el producido por una o más prácticas que lo integran. Las prácticas se vinculan unas con otras, también gracias a sus entradas y resultados, ya que las entradas de algunas prácticas son los resultados de otras prácticas dentro del mismo método. Las definiciones originales de KUALI-BEH 1.1 (Oktaba & Morales Trujillo, 2012), son:

Entrada: Características esperadas de un producto de trabajo, o condiciones, necesarias para empezar la ejecución de una práctica.

Resultado: Características esperadas de un producto de trabajo, o condiciones, requeridas como salida, después de la ejecución de una práctica.

Producto de trabajo: Artefacto utilizado o generado por una práctica.

Condición: Situación, circunstancia o estado específico, de algo o alguien, con respecto a su apariencia, capacidad, u orden de trabajo, y relacionado con el proyecto de software.

Para tener una referencia detallada, sería necesario y correcto, referirse al trabajo original de KUALI-BEH 1.1; no obstante, para propósitos de este trabajo de tesis, es conveniente tener una visión reducida de los conceptos de KUALI-BEH que se van a tratar, lo que se puede conseguir revisando el mapa conceptual de la Figura 1.

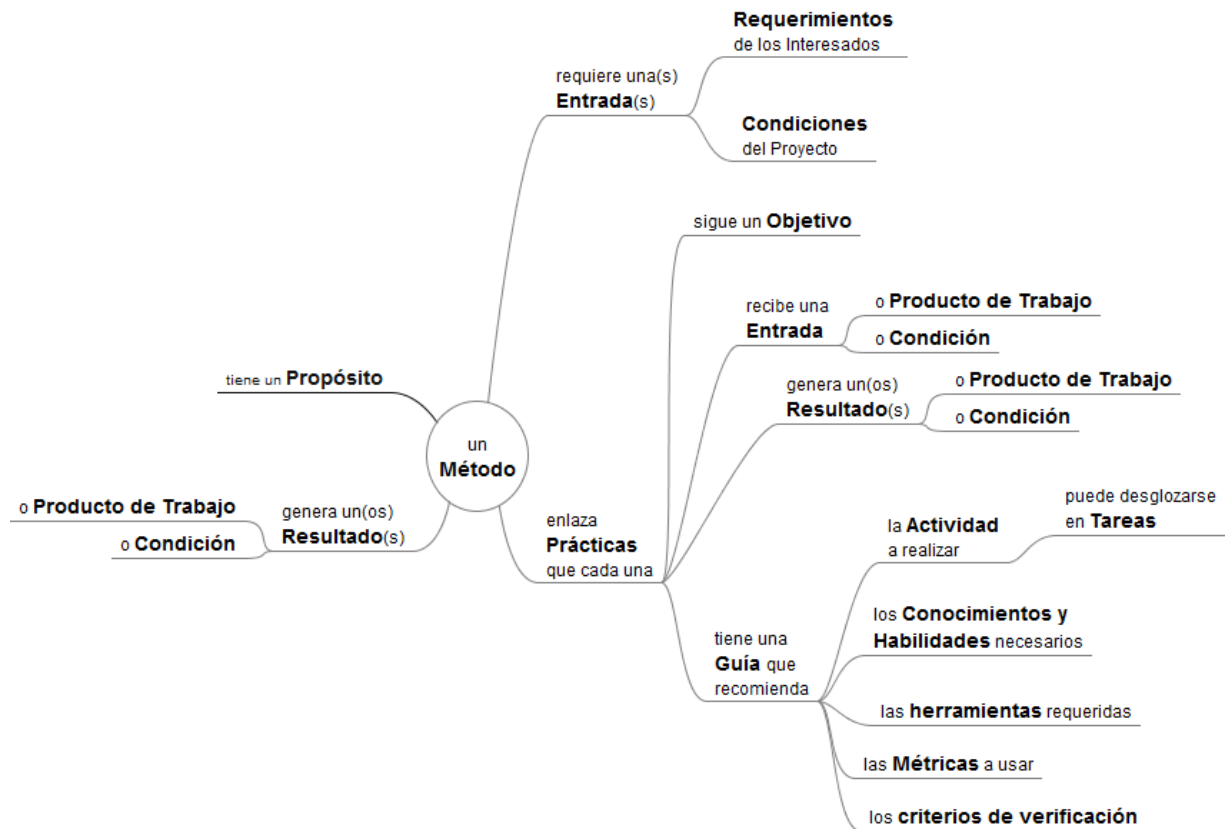


Figura 1 - Mapa Conceptual de Vista Estática de KUALI-BEH

La infraestructura de métodos y prácticas es un depósito que almacena un conjunto de métodos como los descritos en la Figura 1, y pretende servir como parte del marco de trabajo, para los equipos de personas dedicadas al software, y así poder definir, reutilizar, difundir y compartir su manera de ejecutar, porque en ellos reside una verdadera experiencia.

Para la propuesta de alcance de esta tesis, basta con la exposición de la Vista Estática de KUALI-BEH; la Vista Operacional debe ser consultada en el documento original KUALI-BEH 1.1 (Oktaba & Morales Trujillo, 2012), o la última versión disponible, ya que es un proyecto activo y que puede crecer más adelante.

Ejemplo de aplicación

Para dejar un poco más clara la terminología de KUALI-BEH, ayudará presentar un ejemplo que muestre cómo se relacionan todos los conceptos. KUALI-BEH debería de permitir expresar cualquier método utilizado por parte de las mismas personas que lo ejecutan, aunque ese método sea en apariencia deficiente o nulo; justo en ese aspecto es que KUALI-BEH hace una aportación

valiosa, ya que permitiría tener un panorama de lo que las personas hacen en realidad, y cambiarlo de acuerdo con sus necesidades, además de compartirlo con sus equipos de trabajo.

La suposición será el método utilizado por un equipo para crear un sitio web, de una pequeña empresa que quiera iniciar por tener presencia en Internet.

Las entradas del método son:

- a) Un “formato de requerimientos de secciones y estilos”.
- b) Un “paquete base de información digital”.

El resultado del método es:

- i. Un “sitio web funcional y en línea”.

Las prácticas que conforman el método son:

- A. Integración de contenido; con esta práctica se organiza la información a presentar, y se coloca en la base de datos o depósito de archivos donde residirá el sitio.
- B. Integración de funcionalidad y diseño; con esta práctica se define la apariencia visual, acorde con el contenido, y se integran elementos funcionales, menús, galerías de imágenes, formas de contacto.

De manera detallada, para la primera práctica, se tiene:

| Práctica | A | Integración de Contenido |
|---|--|--|
| Objetivo | Organizar la información a presentar y colocarla en la base de datos o depósito de archivos donde residirá el sitio. | |
| Entradas | | Salidas |
| <ul style="list-style-type: none"> • Formato de requerimientos de secciones y estilos: Documento predefinido con algunas secciones de selección múltiple y otras en formato libre, para que el cliente que hace la solicitud, especifique las características que espera de su sitio, secciones, menús, galerías, blog, catálogos u otras opciones. • Paquete base de información digital: Conjunto de archivos digitales de texto, imágenes, videos, o cualquier elemento que el cliente haya decidido | | <ul style="list-style-type: none"> • Sitio con contenido base, estructurado, sin formato. |

| | | |
|--|---|--|
| incluir en la versión inicial de su sitio web. | | |
| Guía | | |
| Actividades | Actividad | Tareas |
| | Verificar información. | |
| | Seleccionar contenido. | |
| | Definir estructura de secciones. | |
| | Instalación del sistema gestor de contenido (CMS, por sus siglas en inglés). | <ul style="list-style-type: none"> • Configuración ambiente de desarrollo con servidores web y de base de datos. • Descarga e instalación de última versión estable de Sistema Gestor de Contenido. • Instalación y configuración de Plug-in⁵ al sistema, de acuerdo con la solicitud. |
| | Carga de información seleccionada al sistema gestor de contenido. | |
| Habilidades y Conocimientos. | Atención al cliente. Administración de información multimedia. Manejo de sistemas gestores de contenidos. | |
| Herramientas | Correo electrónico. Servidor para ambiente de desarrollo. Sistema gestor de contenido. | |
| Métricas | Cantidad de horas por analista. | |
| Criterios de Verificación | Revisión de secciones de contenido contra formato de requerimientos de secciones y estilos. | |

Tabla 1 Ejemplo "A" de Práctica en KUALI-BEH

Y para la segunda práctica, se tiene:

| | | |
|----------|--|---------------------------------------|
| Práctica | B | Integración de funcionalidad y diseño |
| Objetivo | Definir apariencia visual, acorde con el contenido, integrar elementos funcionales, menús, galerías, formas de contacto y lo que se haya definido. | |
| Entradas | Salidas | |

⁵ Plug-in s una pieza de software que se instala sobre un sistema para darle una funcionalidad extra, en un sistema gestor de contenido. Un plug-in puede ser: una galería de videos, un calendario, una traducción a un idioma distinto del que viene por defecto, entre otros.

| | | |
|--|--|---|
| <ul style="list-style-type: none"> • Sitio con contenido base, estructurado, sin formato. | | <ul style="list-style-type: none"> • Sitio web funcional y en línea, con la información que el interesado proporcionó. |
| Guía | | |
| Actividades | Actividad | Tareas |
| | Definir una maqueta web de acuerdo con la solicitud. | |
| | Definición de paleta de colores. | |
| | Creación de multimedia. | <ul style="list-style-type: none"> • Edición de imágenes. • Edición de videos. • Edición de audios. |
| | Creación de plantilla para el sistema gestor de contenido. | |
| | Instalación de plantilla. | |
| | Liberación de sitio web final. | <ul style="list-style-type: none"> • Preparación de ambiente de producción. • Migración de desarrollo a producción. |
| Habilidades y conocimientos | Diseño gráfico. Diseño Web. Post producción multimedia. Diseño UX ⁶ . Optimización SEO. Manejo de marcos de trabajo o bibliotecas para desarrollo de plantillas para sistemas gestores de contenido. | |
| Herramientas | Servidor para ambiente de desarrollo. Servidor para ambiente de producción. Sistema gestor de contenido. Software de manipulación de imágenes, video, audio. Software de creación de plantillas para sistemas gestores de contenido. | |
| Métricas | Horas por diseñador web y horas por administrador de servidores. | |
| Criterios de verificación | Revisión de estilos contra formato de requerimientos de secciones y estilos. | |

Tabla 2 Ejemplo "B" de Práctica en KUALI-BEH

Este método de ejemplo puede ser representado de manera completa, en la Figura 2

⁶ UX es un término usado en la actualidad, y se refiere al diseño del uso del sistema por parte del usuario (en inglés: User eXperience).

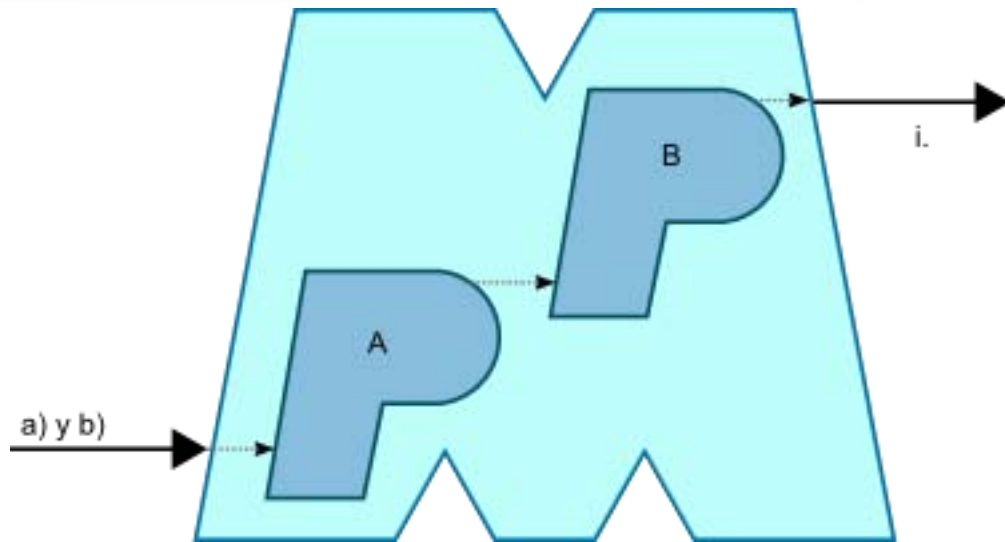


Figura 2 Representación gráfica de un método de ejemplo

Al tener un panorama de lo que es KUALI-BEH, y al retomar uno de sus propósitos originales, se tiene una oportunidad para contribuir en su camino para apoyar a la comunidad de ingeniería de software, ya que para facilitar la creación y adaptación de métodos, a las personas involucradas en proyectos de esta industria, además de utilizar los conceptos de KUALI-BEH, un paso lógico siguiente es tener el apoyo de herramientas de software, que puedan procesar, compartir o verificar la información creada por los desarrolladores o ingenieros de proceso.

Durante marzo del 2013, en una visita a la UNAM de Richard Mark Soley, CEO⁷ del OMG, se le hizo la pregunta: ¿cuál consideras sería la mayor diferencia entre los estándares generados por la ISO⁸ y la OMG? Ante lo cual dejó una idea clara; la diferencia es el impulso que le da la OMG a los estándares publicados; cuando un estándar se publica, casi de manera inmediata salen cursos, libros y herramientas para su uso, mientras que la ISO solo genera el estándar.

Así que, una vez definido el estándar, es decir, el núcleo de conceptos comunes a los proyectos de creación o actualización de software, los diferentes equipos que decidan utilizarlo, podrán construir sistemas que les permita manejar la información de sus métodos; por ejemplo, una base de datos, o plantillas de documentos de texto que permitan capturar los conceptos de KUALI-BEH; sin embargo, irremediablemente surgen nuevas preguntas y retos. ¿Si utilizamos un

⁷ Chief Executive Officer (Director Ejecutivo).

⁸ International Organization for Standardization (Organización Internacional de Normalización).

documento de texto para cada práctica de un método, cómo podríamos verificar su propiedad de suficiencia? Por supuesto, si son tan solo dos o tres prácticas, quizá sería suficiente revisarlas a simple vista, pero, ¿y si son decenas o cientos de prácticas o métodos?

Otra opción sería crear un sistema específico, con su propio modelo de base de datos para KUALI-BEH, entonces de esta manera, sería factible manejar cientos de métodos y prácticas; se haría realidad el concepto de tener una infraestructura de métodos y prácticas, pero en ese escenario, si se tuviera la necesidad de transferir los métodos y prácticas a otro sistema, con un modelo de base de datos diferente, ¿se debería crear una interfaz para ese caso particular? ¿Y cuando exista otro sistema diferente? Queda claro que, entre más herramientas de software existan para manejar conceptos de KUALI-BEH, más complejo se vuelve el proceso de compartir información entre ellas, esto no representaría problema alguno si los equipos de ingeniería de software que utilicen esas herramientas, sólo utilicen una y no la cambien nunca, sin embargo eso es imposible en la práctica, ¿qué sucede si un equipo de trabajo quiere compartir su experiencia con otros equipos por medio de los métodos que ya tiene definidos? Sobre todo si manejan, sus métodos y prácticas, en sistemas con esquemas diferentes para almacenar su información de tipo KUALI-BEH.

Estas son las preguntas que se pretenden resaltar, y que desde el punto de vista de un practicante de la ingeniería de software, motivan la propuesta para este trabajo de tesis.

Capítulo 2 – Situación actual de esquemas informáticos

Perspectiva

Para desarrollar un esquema informático que pueda contener información de métodos y prácticas bajo los conceptos de KUALI-BEH, existen varias opciones tecnológicas en la industria, y se debe seleccionar una que ayude a conseguir el objetivo de poder compartir el conocimiento práctico de los equipos de desarrollo de software, a través de diferentes herramientas computacionales.

Se puede intuir que los equipos de desarrollo que decidan utilizar KUALI-BEH, y utilicen una herramienta para ese fin, requerirán que la información manejada por esa herramienta se pueda compartir de manera independiente de la plataforma, pues algunos equipos usarán equipos personales, ya sea con sistemas Windows de Microsoft, Apple OS, o Linux, pero otros podrán usar computadoras con algún sistema UNIX, o incluso, como es la tendencia actual, podrían usar algún dispositivo móvil, donde existen otros sistemas como Android de Google.

Los elementos de KUALI-BEH están definidos de manera precisa, y por lo tanto se requiere una tecnología que permita definir de esta misma forma el contenido, que sea estricta en su aplicación; por ejemplo, una base de datos, ya que tiene reglas muy bien definidas para almacenar información, aunque claro, los sistemas gestores de base de datos no comparten sus esquemas de manera regular, y en no pocas ocasiones, es un trabajo considerable transferir información de uno a otro de estos sistemas.

Otro aspecto a considerar, es que KUALI-BEH ha sido un proyecto en evolución, y entonces sería deseable que la tecnología adoptada tenga esa característica de flexibilidad, de adaptación.

Al final, el esquema adoptado no tiene que ser el único; como por ejemplo, en el área de imagen digital existen diferentes esquemas para almacenar y transferir archivos de imágenes; los hay de tipo vectorial, y de tipo mapa de bits, y dentro de estos últimos, los hay con compresión o sin compresión, o con otras características. Son varios los esquemas utilizados en el área de imágenes digitales, así se tiene: GIF, JPG, BMP o PNG, y todos han funcionado bien al cubrir los propósitos de cada uno, por lo que se puede decir que no hay un formato único para todas las imágenes digitales; una imagen, en apariencia, podría ser idéntica a otra, pero la información que

la describe puede estar contenida en diferente formato, ya que algunos formatos serán mejor para ser transferidos, otros para ser impresos, otros para formar parte de una secuencia de imágenes (video). De manera análoga, el nuevo esquema buscado para almacenar información KUALI-BEH, no tendría que ser único en su formato.

Para dar validez a los requerimientos planteados en este apartado, se llevó a cabo una revisión sistemática de problemáticas similares, y aunque también los requerimientos pueden cambiar en el tiempo, esta revisión proporciona un punto de inicio para la selección del esquema a utilizar.

Revisión Sistemática

Para llevar a cabo la revisión sistemática, se definió el siguiente protocolo, seguido de sus resultados:

Justificación

Dar fundamento a la selección de la tecnología, para definir el esquema.

Preguntas de investigación

1. ¿Cuáles son las características de los proyectos que requieren un esquema informático para manejar su información?
2. ¿Cuáles son las características del manejo de información en esquemas informáticos?
3. ¿Cuáles son los perfiles de usuarios generales de un esquema informático para KUALI-BEH?
4. ¿Qué opciones se han usado como esquemas informáticos para otros proyectos?
5. ¿Cuáles ventajas y desventajas se pueden observar en el uso de otros esquemas en diferentes proyectos?
6. ¿Cuáles criterios se han utilizado para seleccionar otros esquemas en otros proyectos?
7. ¿Existen evaluaciones de esquemas ya utilizados en otros proyectos?

Estrategias de selección de fuentes de información

Se realizarán búsquedas en los siguientes repositorios:

- IEEE Xplore.
- ACM Digital Library.

La búsqueda se realizará con las siguientes palabras clave para tratar de responder cada pregunta de investigación:

- Pregunta 1: [web OR data] AND [analyzing OR formatting OR sharing OR interchanging] AND [issues OR techniques OR software model OR approaches].
- Pregunta 2: [web OR data] AND [analyzing OR formatting OR sharing OR interchanging] AND [types OR characteristics OR classes].
- Pregunta 3 y 4: [web OR data] AND [analyzing OR formatting OR sharing OR interchanging] AND [applications OR uses OR systems OR software].
- Pregunta 5, 6 y 7: [web OR data] AND [analyzing OR formatting OR sharing OR interchanging] AND [evaluations OR options OR choices OR analysis].

Criterios de selección

- Cumplimiento de requerimientos de la comunidad de software.
- Características con más ventajas detectadas para el uso de esquemas informáticos.
- Coincidencias en los criterios anteriores.

Estrategia de extracción de la información

- Lectura de títulos acordes a las frases/palabras empleadas en las búsquedas.
- Lectura de resúmenes (abstracts).
- Lectura de información (introducción, resultados/conclusiones).
- Lectura del artículo.

Estrategia de síntesis de información

Se utiliza una tabla de base de datos para guardar las referencias encontradas, considerando id, título, resumen, autores, fecha, revista, palabras clave. En la figura 3 se muestra un ejemplo de la tabla utilizada en MySQL.

| articuloId | fuente | title | authors | abstract | anio | fecha | otro |
|------------|--------|---|---|------------------------------------|------|------------|-------------------|
| 1 | IEEE | A distributed spatial data integrated management prototype for geological appl | Xingguang Wang; Xiaojuan Wu; Yi Zhang; Y | With the development of netw | 2013 | 0000-00-00 | |
| 2 | IEEE | Based on grammar analysis's expressiveness among the different XML-Schema | Wei Gong; Peng Yao | The XML (eXtensible Markup L | 2013 | 0000-00-00 | 4673-4997-0 |
| 3 | IEEE | Efficient Online Sharing of Geospatial Big Data Using NoSQL XML Databases | Amirian, P.; Basiri, A.; Winstanley, A. | Summary form only given; Tod | 2013 | 0000-00-00 | |
| 4 | IEEE | NodeXL: Simple network analysis for social media | Smith, M.A. | NodeXL (http://nodexl.codeple | 2013 | 0000-00-00 | 978-1-4673-6403-4 |
| 5 | IEEE | Runtime generation of data processors on local user computers | Patel, J.; Dascalu, S.M.; Harris, F.C. | Data interoperability in scientifi | 2013 | 0000-00-00 | 978-1-4673-6403-4 |
| 6 | IEEE | A common conceptual model for clone detection results | Kapser, C.J.; Harder, J.; Baxter, Ira | As the field of code clone resea | 2012 | 0000-00-00 | 978-1-4673-1794-8 |
| 7 | IEEE | A consumer e-loyalty assessment model: B2C service management by fuzzy MC | Yuching Chen; Gwo-Hshiung Tzeng | Customer loyalty is one of the c | 2012 | 0000-00-00 | 978-1-4673-0894-6 |
| 8 | IEEE | A Watershed-Based Debris Flow Early Warning System Using Sensor Web Enabli | Jen-Cheng Chiu; Chyi-Ren Dow; Cheng-Mir | Debris flow disasters have incre | 2012 | 0000-00-00 | |
| 9 | IEEE | Annotation based personal knowledge management of research scholars | Shukla, A.; Chaudhary, B.D. | Students enrolled in postgradu | 2012 | 0000-00-00 | 978-1-4673-2267-6 |
| 10 | IEEE | Application-Level CPU Consumption Estimation: Towards Performance Isolation | Wei Wang; Xiang Huang; Xiulei Qin; Wenbo | Performance isolation is a key r | 2012 | 0000-00-00 | 978-1-4673-2892-0 |
| 11 | IEEE | Ensuring XML Integrity Using Watermarking Techniques | Tchokpon, R.; Cimato, S.; Bennani, N. | Today, XML is the most used d | 2012 | 0000-00-00 | 978-1-4673-5152-2 |
| 12 | IEEE | Gaussian Mixture Model and Particle Filter | Kaji, K.; Kawaguchi, N. | Up to now Scene Analysis has b | 2012 | 0000-00-00 | 978-1-4673-1955-3 |
| 13 | IEEE | Maestro: A cloud computing framework with automated locking | Demirbas, M.; Tasci, S.; Kulkarni, S. | Concurrent execution is a big c | 2012 | 0000-00-00 | 978-1-4673-2712-1 |
| 14 | IEEE | Material flow data representation and integration based on STEP | Shariatzadeh, N.; Sivard, G.; Hedlind, M.; Wi | A fundamental requirement for | 2012 | 0000-00-00 | 978-1-4673-1022-2 |
| 15 | IEEE | MyPHRMachines: Lifelong Personal Health Records in the cloud | Van Gorp, P.; Comuzzi, M. | Personal Health Records (PHRs | 2012 | 0000-00-00 | 978-1-4673-2049-8 |
| 16 | IEEE | Regulator visions on duplexing schemes & multiple access schemes | Rabee, S.A.; Al-Khwildi, A.N.; Abd, H.H. | This paper advises on the duple | 2012 | 0000-00-00 | 978-1-4673-4829-4 |
| 17 | IEEE | Research on web data integration framework based on cloud computing | Yanxia Wang | According to the sharing of infi | 2012 | 0000-00-00 | 978-1-4577-1414-6 |
| 18 | IEEE | Scrapy-Based Crawling and User-Behavior Characteristics Analysis on Taobao | Jing Wang; Yuchun Guo | The widespread use of Internet | 2012 | 0000-00-00 | 978-1-4673-2624-7 |
| 19 | IEEE | Taking the Pulse of Political Emotions in Latin America Based on Social Web Str | Diaz-Aviles, E.; Orellana-Rodriguez, C.; Nejd | Social media services have becc | 2012 | 0000-00-00 | 978-1-4673-4473-9 |

Figura 3 Captura de pantalla de ejemplo de la base de datos utilizada en la revisión sistemática

También se creará una tabla de valoraciones y ponderaciones que se obtengan durante la revisión.

Finalmente se crea una matriz en la que se encuentren coincidencias entre esquemas encontrados y características, para visualizar cuál cumple con más requisitos de acuerdo con el proyecto de KUALI-BEH.

Resultados

Para que una computadora pueda ayudar a una persona en el manejo de los conceptos de KUALI-BEH, es necesario que esa información la pueda interpretar y transferir, cumpliendo los siguientes requisitos:

- Manejarse en diferentes plataformas de hardware y software.
- Validarse con reglas específicas sobre los datos.
- Usar un formato estándar, flexible y abierto, para que pueda extenderse su uso entre la comunidad.

Ejecutando las consultas se obtuvieron:

- Total de artículos: 128
 - Por revista: ACM=125, IEEE=3
 - Por año: 2013=5, 2012=18, 2011 y 2006=9, 2010=22, 2009=12, 2007=10, entre 1990 y 2005=40
- Resúmenes (abstracts) leídos: 72
- Artículos leídos: 6

A continuación se describen los casos seleccionados que exponen problemáticas que abordaron y cómo o porqué decidieron la solución presentada.

- En un proyecto para intercambio de datos de pacientes se usaron esquemas: XML, DAT (video) y EDF, (European Data Format) (formato estándar para intercambio de datos médicos). El esquema XML fue para registro de propiedades de pacientes, es decir, salvo que la información sea muy específica, como el video, para lo más genérico en este caso, les funcionó el XML. (Penhaker, Krawiec, Krejcar, Novak, & Bridzik, 2010).
- En otro proyecto de información quirúrgica, notas de expertos, el intercambio de datos se dio sobre la web, aunque también podría darse fuera de línea, con USB o discos. En este caso se usó además un esquema de datos relacional (mencionan que usaron MySQL), pero el XML resultó ser el mejor formato para intercambio, y otra ventaja que les dio era que a

partir del XML podían derivar la información en otros formatos que usaron, como el PDF o el HTML. (Paoin & Boonchai-Apisit, 2009)

- En otro proyecto para compartir imágenes en Internet se usaba DXF (acrónimo del inglés Drawing Exchange Format, es un formato de archivo para dibujos de diseño asistido por computadora), pero requerían compartir gráficos de manera efectiva en un ambiente distribuido, y XML les ayudó además con la pérdida de información durante la transmisión, y a la hora de convertir los gráficos a SVG (acrónimo del inglés Scalar Vector Graphics, formato estándar para imágenes vectoriales muy usado por los navegadores web) (Wang & Yan, 2008).
- En otro caso, se creó una herramienta XMI2SQL para convertir archivos XMI (estándar de gráficos UML) a lenguaje SQL, ya que no es tan obvio crear modelos directamente en modelo relacional. Este caso expone que XMI es adecuado para manejar las gráficas UML, pero considerando que XMI está basado en XML, esto permite tener una aplicación que obtenga datos de ese esquema, no solo a la que se refiere al tema de representación gráfica (Chung & Hartford, 2006).

Conclusiones de la revisión sistemática

- Existen varios formatos de archivos para lograr objetivos similares al proyecto de KUALI-BEH, y en la mayoría de los casos se utilizó XML.
- En algunos proyectos se utilizaron formatos específicos de gráficos, datos médicos, o geolocalización (SVG, DAT, EDF) pero aun así, se complementó con XML para datos menos especializados.
- Las necesidades que más se repiten en los proyectos, son: interoperabilidad, independiente de lenguaje, demanda baja de poder de proceso, flexibilidad de datos, y manejo de modelos.
- Los sistemas que más utilizaron XML fueron en áreas de: Web, minería de datos, medicina, gráficas, toma de decisiones con base de datos sociales, información geográfica, negocios, semántica.
- XMI es más específico que XML para gráficas; representar diagramas de UML con XML se vuelve complejo rápidamente si son grandes.

Otras revisiones

Hasta este punto, la tecnología XML parece ser la indicada para desarrollar el esquema para KUALI-BEH, pero para ampliar un poco el panorama, se revisaron otras tecnologías conocidas ampliamente hoy en día, como son RDF, JSON o YAML.

En el caso de RDF (acrónimo del inglés *Resource Description Framework*), a sugerencia de la maestra Guadalupe Ibarguengoitia, se revisó un caso de aplicación concreto y cercano, la tesis

de maestría de un ex alumno de la profesora: “USO DE LA TECNOLOGÍA RDF PARA REPRESENTAR Y MANEJAR LOS PROCESOS MOPROSOFT Y SU APLICACIÓN EN HIM” (Hernández Uribe, 2005). Con este caso se pueden reafirmar las fortalezas para las que fue diseñada esta tecnología, como intercambiar datos en la web, a través de modelos que facilitan la definición de tripletas, que se componen de dos recursos en Internet y una relación entre ellos; la idea principal es facilitar una web semántica, una web a la que se agrega información extra a los datos, para poder clasificarlos. RDF es un estándar del W3C⁹ y se puede obtener más información en su sitio oficial (World Wide Web Consortium, 2014). Para el caso de KUALI-BEH, una de sus características es que el conjunto de conceptos son comunes, y ya fueron seleccionados y relacionados entre sí, entonces se puede prescindir de esa ventaja de RDF por ahora en el alcance de esta tesis; además, RDF fue diseñada para usarse en la web, y las herramientas que usen el esquema propuesto de KUALI-BEH podrían existir en otros ambientes; y por último, RDF está basado en XML, es una especialización de XML, y es otra razón para confiar en definir el nuevo esquema con XML.

En el caso de JSON (acrónimo del inglés de JavaScript Object Notation), es un formato ligero de intercambio de datos, independiente del lenguaje, que permite definir estructuras como objetos y árboles, útiles para manejar información de KUALI-BEH, y además es un estándar de la Ecma International¹⁰ (Ecma International, 2013). Así, JSON parece una opción interesante para el nuevo esquema para KUALI-BEH; no obstante, está orientado a los programadores, es decir, para intercambiar datos entre lenguajes de programación, pero no así entre aplicaciones como lo es XML, aplicaciones que podrían ser de presentación. A diferencia de JSON, XML se apoya de otras tecnologías estándares como XSLT¹¹ para generar otros formatos de archivo a partir de XML, y puede servir para dar formato a documentos KUALI-BEH, documentos impresos o en pantalla, ya que la información almacenada de KUALI-BEH, al final de cuentas, también se trata de métodos y prácticas que las personas leen.

⁹ W3C es como se le conoce al World Wide Web Consortium, Consorcio que hace recomendaciones y define estándares para la Web mundial, entre ellas, el mismo lenguaje de la red, el HTML.

¹⁰ Ecma International es el nombre al que cambió la European Computer Manufacturers Association y que se dedican a generar estándares, entre los que se encuentra C++/CLI, y el lenguaje C#.

¹¹ XSLT, acrónimo del inglés Extensible Stylesheet Language Transformations, otro estándar del World Wide Web Consortium (W3C), y es un lenguaje de hojas de estilo para documentos XML.

En el caso de YAML (acrónimo del inglés YAML Ain't Markup Language), es un estándar para todos los lenguajes de programación, de fácil visualización para humanos, para convertir datos dentro de una estructura de datos u objeto, a un formato en que pueden ser almacenados (Evans, 2009). Es un caso parecido a JSON, porque aquí, la facilidad que ofrece a las personas, es para visualizar estructuras de datos que utilizan lenguajes de programación, no para una vista impresa o de pantalla de los conceptos que representan los datos.

Para continuar las revisiones para seleccionar la tecnología a utilizar para crear el nuevo esquema para KUALI-BEH, se tomó otro caso específico y cercano, esta vez, de otra tesis de maestría del mismo posgrado: “ARQUITECTURA GENÉRICA PARA ALMACENAR DOCUMENTOS XML EN UNA BASE DE DATOS RELACIONAL” (Trillo Lares, 2005), en la que, en uno de sus capítulos, se analiza la tecnología XML desde sus orígenes, los objetivos que tenía el equipo que lo produjo, y las ventajas que ofrecía XML en ese contexto, pero que tienen sentido para lo que se necesita para el esquema buscado de KUALI-BEH, objetivos como ser formal en la definición de datos, ser fácil de implantar en distintos sistemas, ser utilizable en Internet, y ventajas como dar facilidades para la representación de datos en visualizadores, o que los generadores y consumidores pueden producir sus propios tipos de documentos.

Y por último, quizá el caso más reconocido a nivel internacional, de los presentados en este trabajo de tesis, es el caso del formato para archivos de suites ofimáticas Open Office XML (Microsoft, s.f.), creado por Microsoft, que a pesar de tener previamente, un formato propio muy difundido en la industria, optó por los formatos abiertos y basado en XML, entre otras razones, para lograr una mayor difusión e integración de su tecnología en la industria, con lo cual creo nuevos formatos, por ejemplo: *.docx y *.xlsx, para sus famosos programas MS-WORD y MS-EXCEL respectivamente, este formato de Microsoft incluso se convirtió en un estándar publicado por la ISO¹², bajo la denominación: “ISO/IEC 29500:2008, Information technology - Document description and processing languages - Office Open XML file formats” (International Organization for Standardization, 2008).

¹² International Organization for Standardization, Institución dedicada a desarrollar y publicar estándares internacionales, reconocida de manera amplia a nivel mundial.

Conclusiones del capítulo 2

Después de la definición de requerimientos para el nuevo esquema, y la revisión de diferentes alternativas, se concluye que sea XML la tecnología para especificar el esquema informático para métodos y prácticas de KUALI-BEH, sin perder de vista que este esquema no es excluyente de otros esquemas posibles, o el único, sobre todo en una industria donde la tecnología cambia de manera constante, y el reto de crear estándares ha estado presente desde el principio.

Capítulo 3 – Creación del Esquema XML para KUALI-BEH

KBL, un nuevo lenguaje para KUALI-BEH

En el capítulo anterior quedaron establecidos los motivos de selección de la tecnología XML para el nuevo esquema de KUALI-BEH. XML puede ser considerado un meta lenguaje, ya que sirve para definir otros lenguajes; por ejemplo, con XML se definió XMI, un lenguaje para describir gráficos UML. Otro ejemplo es SVG, un lenguaje para describir gráficos vectoriales.

Ahora es momento de diseñar un nuevo lenguaje, cuyo propósito será describir métodos KUALI-BEH, en concreto, un lenguaje para describir procesos de software, acorde a los conceptos definidos en la especificación de KUALI-BEH. A partir de este punto, se le llamará KBL (KUALI-BEH Language) a este nuevo lenguaje, y los archivos que contengan información descrita con KBL, serán archivos en formato XML, con la extensión *.kbl.

“Aplicación XML” es otra manera, bastante común, de llamar a los lenguajes definidos con XML, por lo que KBL puede considerarse una aplicación XML.

Conceptos KUALI-BEH, para base del nuevo esquema

Los conceptos seleccionados de KUALI-BEH, para ser parte del lenguaje KBL, incluyen los conceptos de la vista estática KUALI-BEH y comienzan por “Método” (Method), ya que el “Método” agrupa las “Prácticas” (“practices”)¹³, y éstas a su vez a las “Actividades” (“activities”), junto con los demás elementos que la conforman. Para tener una visión más clara, en la siguiente figura se presenta la gráfica del árbol de KLB:

¹³ Nombre en inglés utilizado en la especificación KUALI-BEH 1.1.

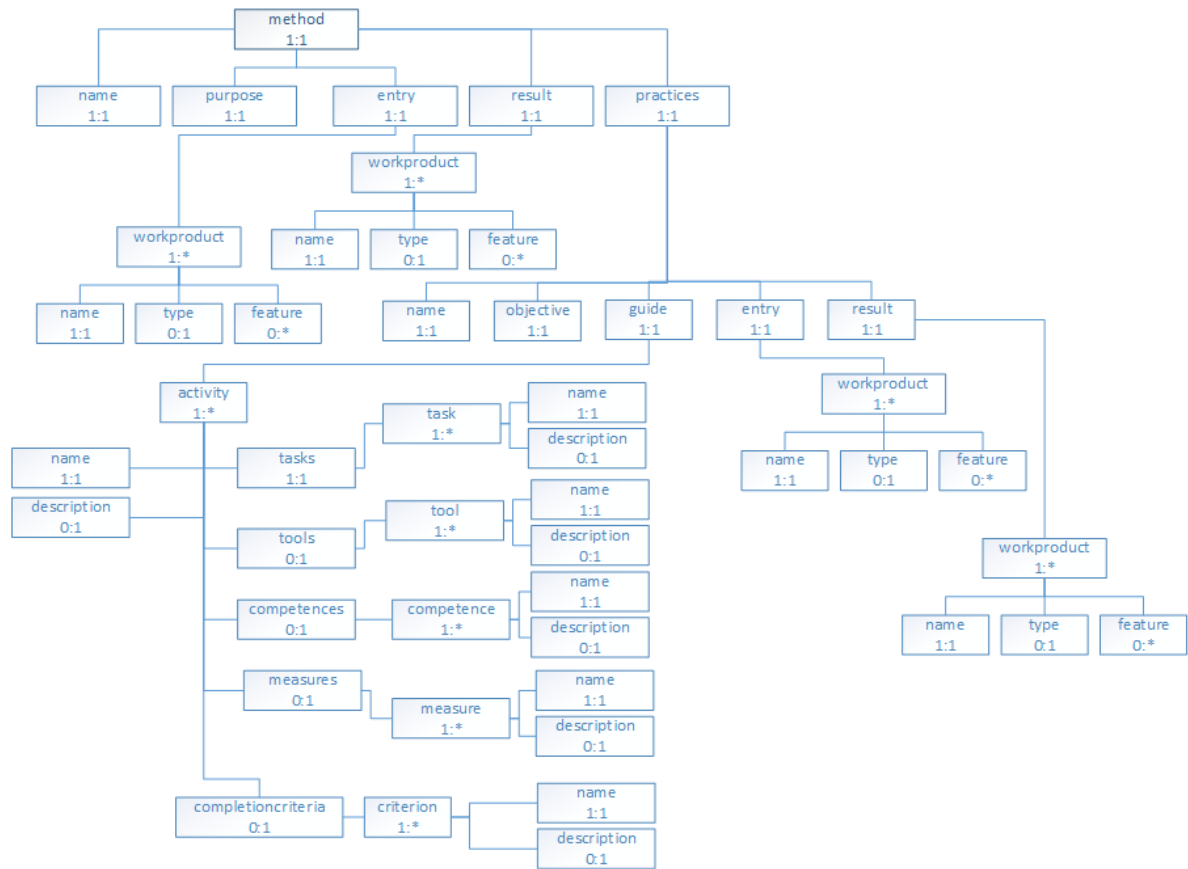


Figura 4. Árbol KLB

En el figura 4, debajo del nombre del concepto KUALI-BEH, se indica el número mínimo y máximo de apariciones que puede tener dicho elemento, lo cual forma parte de las decisiones de diseño, descritas más adelante. Los conceptos KUALI-BEH se corresponden con los elementos del nuevo esquema, lo que significa que son etiquetas del lenguaje KBL.

Decisiones relevantes de diseño para esquema KUALI-BEH.xsd

Por definición, el esquema XML (XSD) debe contener una estructura de árbol, con el elemento raíz estándar <schema>. Los elementos declarados como “hijos” de la etiqueta <schema> son considerados elementos globales, y todos los elementos declarados en un nivel más profundo son considerados locales. En la estructura arbórea para el nuevo lenguaje, se definió al elemento “method” como el único elemento global, y será la raíz del árbol en los documentos *.KBL.

El elemento global <method>, que representa el concepto “Método”, se definió como único, ya que es el elemento con el que se pueden trabajar las propiedades del método descritas en KUALI-BEH; no se podrían verificar esas propiedades con solo una “Práctica” o un conjunto de ellas, porque es necesario considerar las “entradas” y “salidas” de las prácticas, pero también las del método.

Tampoco se definió un conjunto de métodos para ser incluidos en un solo documento, debido a que eso supone definir un concepto similar a “Metodología” (conjunto de métodos) que hoy no existe en KUALI-BEH. Por otro lado, un archivo con más de un método, además de ser más complejo y más grande, también supondría la repetición de elementos como prácticas, actividades, tareas, y herramientas, entre otros, ya que se puede presentar la situación en que sean métodos utilizados por un mismo equipo de trabajo y compartan los mismos recursos.

Si alguna vez se vuelve absolutamente necesario definir varios métodos juntos, se tomó la decisión de no solo declarar el elemento, sino declarar una definición de tipo, el tipo <methodtype>, el cual se podría referenciar para crear un esquema nuevo que los agrupe.

De cualquier manera, al tener un esquema XSD, puede existir un software que facilite el intercambio de uno o más métodos entre equipos de trabajo, sin necesidad de ocuparse en la decisión de intercambiarse en uno o más archivos.

El elemento <method> debe tener un elemento <entry> y un elemento <result>, lo que significa que son obligatorios, ya que son los que contienen a los elementos <workproduct>, correspondientes al concepto “Producto de Trabajo” (Work Product) de KUALI-BEH, siendo estos elementos los que afectan las propiedades de suficiencia, coherencia y consistencia.

Al igual que en el elemento <method>, para los elementos <practice>, que corresponden al concepto “Práctica” de KUALI-BEH, los elementos <workproduct> son obligatorios, con sus correspondientes etiquetas <entry> y <result>.

Ya que los elementos <method> y <practice> utilizan los elementos <workproduct>, se creó un tipo de elemento <workproducttype> para permitir su reutilización en las definiciones de <method> y <practice>, esto con la intención de verificar las propiedades del método.

Los elementos <activity> corresponden al concepto “Actividad” de KUALI-BEH” y pueden formar una lista con mínimo un elemento, que se agrupan bajo el elemento <guide>, que corresponde al concepto “Guía” de KUALI-BEH.

Todos los elementos que pueden formar una lista dentro de los elementos <activity>, como son los que corresponden a los conceptos: Tareas, Herramientas, Competencias, Métricas y Criterios de Verificación, tienen la misma estructura, es decir, un elemento que representa su nombre y otro su descripción, que son: <name> y <description> respectivamente.

Por una parte, el elemento <name>, para este nuevo lenguaje KBL, queda definido como obligatorio, ya que es el atributo con el que se identifica, y por otro lado, el elemento <description> se define como opcional; a pesar de compartir la misma estructura, se definieron tipos para cada elemento por separado, ya que si alguna vez es necesario definir de manera diferente alguno de ellos, sería independiente y no afectaría a la definición de los nuevos elementos.

Los elementos a los que se hace referencia en el párrafo anterior son: <task>, <tool>, <competence>, <measure> y <criterion>.

Aspectos XML en consideración para KBL

Hoy en día, XML es una tecnología bastante difundida que cuenta con documentación adicional a la especificación original, tanto documentación detallada como resumida, y distribuida en diferentes medios, como son libros, revistas, guías, en Internet, videos tutoriales y demás. El objetivo de esta sección será resaltar de manera puntual los aspectos de la tecnología XML que sirvieron en la definición de KBL, en lugar de un tratado completo sobre XML.

Alternativa de gramática de KBL

Los lenguajes definidos con XML tienen una sintaxis y una gramática bien definida; a los documentos que contienen información descrita en ese lenguaje y cumplen con su sintaxis se les conocen como documentos “bien formados”; si además, cumplen con su gramática, son conocidos como documentos “válidos”.

La sintaxis para el nuevo lenguaje KBL es la misma de XML, por lo cual, si un archivo KBL cumple con la sintaxis XML, será un documento “bien formado”.

En cuanto a la gramática, se puede especificar con tecnologías como: DTD (Document Type Definition)¹⁴, Relax NG¹⁵ y las definiciones de esquemas XML (XML Schema Definition o XSD, por sus siglas en inglés)¹⁶.

Otra tecnología es DTD, pero a pesar de ser conocida y estar ampliamente difundida, tiene ciertas limitaciones, la principal es causada porque utiliza una sintaxis diferente al propio XML.

Otra alternativa es Relax NG ofrece mejoras al DTD, pero no es tan conocida como éste; y por último, XSD ofrece tanto difusión como mejoras a las dos anteriores, por lo que fue seleccionada para definir la gramática de KBL.

Patrón de diseño de la gramática de KBL

Para definir un nuevo lenguaje utilizando la tecnología XSD, existen patrones de diseño que facilitan encontrar soluciones estándar y probadas a necesidades frecuentes; por supuesto, no son reglas, pero sí buenas sugerencias. Los nombres utilizados para los patrones pueden variar, dependiendo de la fuente consultada, por lo cual conviene mejor identificar sus características.

Dependiendo de la fuente consultada, en algunos casos, la ventaja de utilizar un patrón de diseño específico, es desventaja en otros; por ejemplo, en un reporte¹⁷ preparado por la SAIC (*Science Applications International Corporation*)¹⁸, una empresa proveedora de servicios de TI para el gobierno de Estados Unidos, el patrón “Russian Dolls” tiene la ventaja de utilizar un solo archivo para la definición del esquema, mientras que en otro reporte, preparado por Oracle Corporation, una de las empresas más importantes de software a nivel mundial, se indica que tener un solo archivo de esquema es una desventaja. Por supuesto, las ventajas y desventajas dependen del contexto de aplicación de cada patrón.

En la siguiente tabla se presentan los nombres más comunes de patrones de diseño, sus características y las ventajas y desventajas identificadas para la creación del nuevo lenguaje KBL.

¹⁴ Especificación del w3c: <http://www.w3.org/TR/2004/REC-xml11-20040204/#NT-doctypepedcl>

¹⁵ Especificación de organización de estándares abiertos OASIS (<https://www.oasis-open.org/>): <http://www.relaxng.org/spec-20011203.html>

¹⁶ Especificación por el World Wide Consortium: <http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/>

¹⁷ Obtenido de http://www.exchangenetwork.net/dev_schema/xml_schema_architecture.pdf

¹⁸ <http://www.saic.com/about/about-saic/>

| Patrón de Diseño | Características | Ventajas | Desventajas |
|-----------------------|---|---|--|
| Russian Dolls | <ul style="list-style-type: none"> • Un solo elemento global. • Sin declaración de tipos. | <p>Tamaño compacto. Fácil lectura. Sin afectaciones si existen cambios. Un solo archivo de esquema.</p> | <p>No reutilizable.</p> |
| Salami Slice | <ul style="list-style-type: none"> • Varios elementos globales. • Sin declaración de tipos. | | <p>Mayor uso de etiquetas. Con afectaciones si existen cambios. Múltiples archivos de esquema. Difícil determinar la raíz.</p> |
| Venetian Blind | <ul style="list-style-type: none"> • Solo un elemento global. • Con declaración de tipos. | <p>Un solo archivo de esquema. Reutilizable tanto para tipos como para el elemento raíz.</p> | <p>Mayor uso de etiquetas.</p> |
| Garden of Eden | <ul style="list-style-type: none"> • Varios elementos globales. • Con declaración de tipos. | <p>Reutilizable, tanto de elementos como de tipos. Permite varios archivos de esquema.</p> | <p>Mayor uso de etiquetas. Con afectaciones, si existen cambios. Múltiples archivos de esquema. Contiene muchos elementos potenciales como raíz.</p> |

| | | | |
|--|--|--|--|
| | | | <p>Limita el encapsulamiento poder definir tipos.</p> <p>Difícil de leer y entender.</p> |
|--|--|--|--|

Tabla 3 Patrones de diseño de esquemas XML

El patrón de diseño seleccionado para KBL fue “Venetian Blind”, así que se definió un solo elemento global (<method>), y se usaron declaraciones de tipo para elementos comunes, como los <workproducttype>, o la definición de tipo <methodtype> para definir el elemento raíz.

La ventaja de que se maneje un solo archivo para el esquema, es que facilita la verificación de archivos del nuevo lenguaje KBL, porque solo se carga un archivo, sin necesidad de cargar varios archivos y mantener referencias. El esquema almacenado en un solo archivo se debe a que únicamente tiene un elemento global.

La ventaja de reutilizar los tipos es que el esquema podría ser ampliado en caso de ser necesario, aunque en ese caso, el esquema quedaría definido en más de un archivo y se volvería complejo.

La ventaja de que se solo se pueda reutilizar el elemento raíz <method> (porque solamente se declaró ese elemento global), es permitir, en caso de ser necesaria, una ampliación del esquema, y seguir manejado el concepto mínimo para poder llevar a cabo la verificación de propiedades: suficiencia, consistencia y coherencia. Una “Práctica” no podría ser factible de verificación, pero tampoco sería siquiera posible reutilizar el elemento, dado que es un elemento local.

La desventaja del patrón “Venetian Blind”, de usar muchas etiquetas, se debe a que se declaran tipos, y luego se declaran los elementos de esos tipos; sin embargo, existen herramientas que permiten manejar esto de manera automática.

Por ejemplo, la herramienta de software NetBeans, permite navegar de manera gráfica entre los elementos del esquema, y no hace falta ver el gran número de etiquetas que fueron necesarias para definirlo. Ver la figura 5:

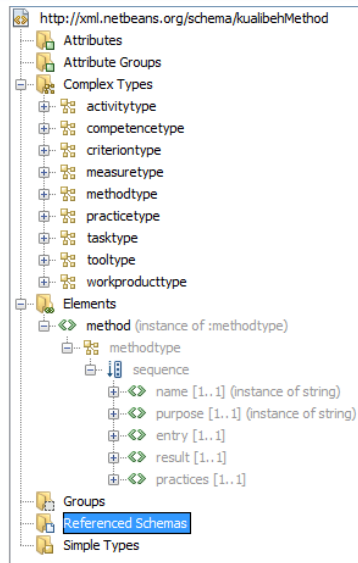


Figura 5 Captura de pantalla de ejemplo de Netbeans para manejo de esquemas XML

Esquema XSD para KUALI-BEH

El nuevo esquema XSD para KUALI-BEH, se nombró `kualibehMethod.xsd`, debido a que contiene la gramática para un solo método.

La URI para el esquema fue bajo el nombre de dominio <http://kualibeh.mx/>, que aunque no es necesario que exista, ya se cuenta con el registro del mismo.

Los elementos del esquema para KBL, son los que se muestran en la siguiente figura:

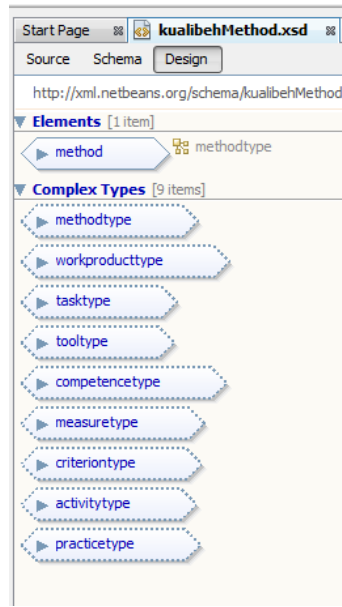


Figura 6 Elementos del esquema KBL

Un solo elemento global y 9 declaraciones de tipo que podrían ser utilizadas para crear nuevos esquemas.

Código fuente del esquema XSD

```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            targetNamespace="http://xml.netbeans.org/schema/kualibehMethod"
            xmlns:tns="http://kualibeh.mx/schema/kualibehMethod"
            elementFormDefault="qualified"
xmlns:ns="http://xml.netbeans.org/schema/kualibehMethod">
  <xsd:complexType name="methodtype">
    <xsd:sequence>
      <xsd:element name="name" type="xsd:string"/>
      <xsd:element name="purpose" type="xsd:string"/>
      <xsd:element name="entry">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element
type="ns:workproducttype" maxOccurs="unbounded"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="result">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element
type="ns:workproducttype" maxOccurs="unbounded"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="practices">
        <xsd:complexType>
          <xsd:sequence>
```

```

                                <xsd:element                                name="practice"
type="ns:practicetype"></xsd:element>
                                </xsd:sequence>
                                </xsd:complexType>
                                </xsd:element>
                                </xsd:sequence>
                                </xsd:complexType>
                                <xsd:complexType name="workproducttype">
                                <xsd:sequence>
                                <xsd:element name="name" type="xsd:string"></xsd:element>
                                <xsd:element                                name="type"                                type="xsd:string"
minOccurs="0"></xsd:element>
                                <xsd:element name="feature" type="xsd:string" maxOccurs="unbounded"
minOccurs="0"></xsd:element>
                                </xsd:sequence>
                                </xsd:complexType>
                                <xsd:complexType name="tasktype">
                                <xsd:sequence>
                                <xsd:element name="name" type="xsd:string"></xsd:element>
                                <xsd:element                                name="description"                                type="xsd:string"
minOccurs="0"></xsd:element>
                                </xsd:sequence>
                                </xsd:complexType>
                                <xsd:complexType name="tooltype">
                                <xsd:sequence>
                                <xsd:element name="name" type="xsd:string"></xsd:element>
                                <xsd:element                                name="description"                                type="xsd:string"
minOccurs="0"></xsd:element>
                                </xsd:sequence>
                                </xsd:complexType>
                                <xsd:complexType name="competencetype">
                                <xsd:sequence>
                                <xsd:element name="name" type="xsd:string"></xsd:element>
                                <xsd:element                                name="description"                                type="xsd:string"
minOccurs="0"></xsd:element>
                                </xsd:sequence>
                                </xsd:complexType>
                                <xsd:complexType name="measuretype">
                                <xsd:sequence>
                                <xsd:element name="name" type="xsd:string"></xsd:element>
                                <xsd:element                                name="description"                                type="xsd:string"
minOccurs="0"></xsd:element>
                                </xsd:sequence>
                                </xsd:complexType>
                                <xsd:complexType name="criteriontype">
                                <xsd:sequence>
                                <xsd:element name="name" type="xsd:string"></xsd:element>
                                <xsd:element                                name="description"                                type="xsd:string"
minOccurs="0"></xsd:element>
                                </xsd:sequence>
                                </xsd:complexType>
                                <xsd:complexType name="activitytype">
                                <xsd:sequence>
                                <xsd:element name="name" type="xsd:string"></xsd:element>
                                <xsd:element                                name="description"                                type="xsd:string"
minOccurs="0"></xsd:element>
                                <xsd:element name="tasks">
                                <xsd:complexType>
                                <xsd:sequence>
                                <xsd:element                                name="task"                                type="ns:tasktype"
maxOccurs="unbounded"></xsd:element>
                                </xsd:sequence>
                                </xsd:complexType>

```

```

</xsd:element>
<xsd:element name="tools" maxOccurs="1" minOccurs="0">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="tool" type="ns:tooltype"
maxOccurs="unbounded"></xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="competences" maxOccurs="1" minOccurs="0">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="competence" type="ns:competencetype"
maxOccurs="unbounded"></xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="measures" maxOccurs="1" minOccurs="0">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="measure" type="ns:measuretype"
maxOccurs="unbounded" minOccurs="1"></xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="completioncriteria" maxOccurs="1" minOccurs="0">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="criterion" type="ns:criteriontype"
maxOccurs="unbounded" minOccurs="1"></xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="practicetype">
  <xsd:sequence>
    <xsd:element name="name" type="xsd:string"></xsd:element>
    <xsd:element name="objective" type="xsd:string"></xsd:element>
    <xsd:element name="entry">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="workproduct"
type="ns:workproducttype" maxOccurs="unbounded"></xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="result">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="workproduct"
type="ns:workproducttype" maxOccurs="unbounded"></xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="guide">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="activity" type="ns:activitytype"
maxOccurs="unbounded"></xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
</xsd:element>

```

```

        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="method" type="ns:methodtype"></xsd:element>
</xsd:schema>

```

Guía de creación de archivos *.KBL

Para crear un documento KBL se debe cumplir con la sintaxis XML, como son las declaraciones iniciales, etiquetas balanceadas, y demás.

Para cumplir con la gramática, el documento debe cumplir con las reglas descritas en el esquema (XSD) con la definición de kaulibehMethod.xsd, y que podrá ser utilizado por cualquier herramienta de validación XML automática, en conjunto con el archivo *.KBL.

La creación de un archivo *.KBL, al igual que cualquier archivo XML, se puede crear de manera manual por el usuario, debido a que es un archivo en formato de texto plano, y puede hacerse en un editor de texto cualquiera; no obstante, lo más común es que estos archivos se generen y se procesen de manera automática por programas de cómputo, en cuyo caso, el software en cuestión determinará la secuencia de creación, solo con la restricción de generar un archivo válido bajo el esquema kualibehMethod.XSD.

La siguiente secuencia de pasos sirve para crear un documento *.KBL, pero el orden no tiene que ser exactamente así; si un usuario lo hace de manera manual, puede elegir escribir primero algunos elementos y después integrarlos, o si lo hace un software de manera automática, la secuencia quedará definida por el mismo.

| Paso | Descripción |
|---|---|
| Declarar las cabeceras de archivo XML | Se deben declarar las cabeceras de un documento XML, incluyendo la referencia al esquema kualibehMethod.xsd. |
| Declarar el elemento <method> | El documento debe contener solo un elemento de tipo <method>, y es el elemento raíz de todo el documento. |
| Declarar los elementos de <method> | El elemento <method> tiene 5 elementos y todos son obligatorios, todos deben aparecer una sola vez, y además todos deben aparecer en el siguiente orden: <name>, <purpose>, <entry>, <result>, <practices>. |

| | |
|--|---|
| | <p>Solo los elementos <name> y <purpose> contienen texto libre; en cambio, los elementos <entry>, <result> y <practices>, contienen otros elementos que deben ser definidos.</p> |
| <p>Declarar los elementos <workproduct> pertenecientes al elemento <entry> y al elemento <result> del elemento <method></p> | <p>Los elementos <workproduct> son del mismo tipo, tanto para el elemento <entry> como para el elemento <result>, y se deben declarar al menos uno, o en todo caso, dejar vacío el contenido de la etiqueta, para cada uno de ellos; y si es necesario más de uno, ya sea en <entry> o en <result>, basta con declararlo con un par de etiquetas igual al primero, y así, los que sean necesarios.</p> <p>Los elementos <workproduct> son elementos complejos; en sí mismos agrupan 3 elementos más, son de texto libre, y solo el primero es obligatorio, los dos últimos son opcionales, pero si se declaran, siempre deben tener el siguiente orden: <name>, <type>, y <feature>.</p> <p>La etiqueta <type> permite definir un tipo de producto de trabajo, como puede ser una “condición determinada” y no necesariamente un producto físico.</p> |
| <p>Declarar los elementos <practice> pertenecientes al elemento <practices></p> | <p>El elemento <practices> de <method> debe contener al menos un elemento <practice>, y si fuera necesario, se pueden agregar más elementos <practice> siguiendo la misma mecánica, y justo después del primero.</p> <p>Los elementos <practice> son elementos complejos, que a su vez, contienen 5 elementos más, los cuales son obligatorios; los 2 primeros: <name> y <objective>, son de texto libre.</p> <p>Los siguientes 2 elementos son: <entry> y <result> y funcionan de la misma manera que los elementos de mismo nombre que pertenecen al elemento <method>, es decir, contienen elementos de tipo <workproduct>.</p> <p>Y por último, el quinto elemento de <practice> es el elemento <guide>; se debe recordar que este orden de definición es estricto.</p> <p>El elemento <guide> es otro elemento completo que va a agrupar los elementos <activity>.</p> |

| | |
|---|--|
| <p>Declarar los elementos <activity> pertenecientes al elemento <guide> de cada <practice></p> | <p>Para cada elemento <activity> se tienen otros 7 elementos anidados; el primero es: <name>, es de texto libre y es obligatorio; el segundo es <description>, también de texto libre, pero es opcional.</p> <p>Los siguientes 5 elementos funcionan de manera casi idéntica entre ellos, porque son elementos complejos que a su vez agrupan colecciones de elementos muy similares.</p> <p>Tomando como ejemplo al primero de los 5 elementos, la declaración se hace con las etiquetas: <tasks></tasks> (en plural) y agrupa (de forma anidada) elementos de tipo <task></task> (en singular). El elemento anidado, el que está expresado en singular, solo contiene 2 elementos anidados más: <name> y <description>; los dos son de texto libre, y solo <name> es obligatorio, <description> es opcional. Cabe mencionar que si un elemento opcional se declara, entonces se debe hacer en el orden descrito.</p> <p>Se tiene que los 5 elementos anidados al final de un elemento <activity> son: <tasks>, <tools>, <competences>, <measures> y <completioncriteria>, los cuales agrupan a los elementos: <task>, <tool>, <competence>, <measure> y <criterion> respectivamente. Y como ya se mencionó, cada uno de estos elementos “en singular” tiene dos elementos: <name> y <description>.</p> <p>De los 5 elementos anteriormente descritos, podrían no estar definidos, y en ese caso, el contenido entre las etiquetas puede quedar vacío, lo que significa que no son obligatorios, por lo tanto, para un elemento <activity>, la mínima expresión es la conformada solo por las etiquetas de apertura y cierre sin contenido entre ellas.</p> |
|---|--|

Tabla 4 Guía de creación de documento KBL

Ejemplo de archivo *.KBL

El siguiente es un ejemplo de un archivo *.KBL, generado de manera automática con la herramienta NetBeans; no tiene contenido real, solo las etiquetas; sin embargo, precisamente esto permite comprender más la estructura definida, y a partir de este punto, un usuario podría ir colocando su contenido, entre las etiquetas correspondientes.

```
<?xml version="1.0" encoding="UTF-8"?>

<!--
  Document      : MetodoCualquiera.kbl
-->

<ns0:method xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:ns0='http://xml.netbeans.org/schema/kualibehMethod'
  xsi:schemaLocation='http://xml.netbeans.org/schema/kualibehMethod
kualibehMethod.xsd'>
  <ns0:name></ns0:name>
  <ns0:purpose></ns0:purpose>
  <ns0:entry>
    <ns0:workproduct>
      <ns0:name></ns0:name>
      <ns0:type></ns0:type>
      <ns0:feature></ns0:feature>
      <ns0:feature></ns0:feature>
      <ns0:feature></ns0:feature>
      <ns0:feature></ns0:feature>
      <ns0:feature></ns0:feature>
      <ns0:feature></ns0:feature>
    </ns0:workproduct>
    <ns0:workproduct>
      <ns0:name></ns0:name>
      <ns0:type></ns0:type>
    </ns0:workproduct>
    <ns0:workproduct>
      <ns0:name></ns0:name>
      <ns0:type></ns0:type>
    </ns0:workproduct>
    <ns0:workproduct>
      <ns0:name></ns0:name>
      <ns0:type></ns0:type>
    </ns0:workproduct>
    <ns0:workproduct>
      <ns0:name></ns0:name>
      <ns0:type></ns0:type>
    </ns0:workproduct>
    <ns0:workproduct>
      <ns0:name></ns0:name>
      <ns0:type></ns0:type>
    </ns0:workproduct>
  </ns0:entry>
  <ns0:result>
    <ns0:workproduct>
      <ns0:name></ns0:name>
      <ns0:type></ns0:type>
    </ns0:workproduct>
  </ns0:result>
  <ns0:practices>
    <ns0:practice>
```

```

<ns0:name></ns0:name>
<ns0:objective></ns0:objective>
<ns0:entry>
  <ns0:workproduct>
    <ns0:name></ns0:name>
    <ns0:type></ns0:type>
  </ns0:workproduct>
</ns0:entry>
<ns0:result>
  <ns0:workproduct>
    <ns0:name></ns0:name>
    <ns0:type></ns0:type>
  </ns0:workproduct>
</ns0:result>
<ns0:guide>
  <ns0:activity>
    <ns0:name></ns0:name>
    <ns0:description></ns0:description>
    <ns0:tasks>
      <ns0:task>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
      </ns0:task>
      <ns0:task>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
      </ns0:task>
      <ns0:task>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
      </ns0:task>
      <ns0:task>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
      </ns0:task>
      <ns0:task>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
      </ns0:task>
      <ns0:task>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
      </ns0:task>
      <ns0:task>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
      </ns0:task>
      <ns0:task>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
      </ns0:task>
      <ns0:task>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
      </ns0:task>
      <ns0:task>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
      </ns0:task>
    </ns0:tasks>
    <ns0:tools>
      <ns0:tool>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
      </ns0:tool>
      <ns0:tool>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
      </ns0:tool>
      <ns0:tool>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
      </ns0:tool>
      <ns0:tool>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
      </ns0:tool>
      <ns0:tool>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
      </ns0:tool>
    </ns0:tools>
  </ns0:activity>
</ns0:guide>

```

```

        <ns0:description></ns0:description>
    </ns0:tool>
    <ns0:tool>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
    </ns0:tool>
</ns0:tools>
<ns0:competences>
    <ns0:competence>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
    </ns0:competence>
    <ns0:competence>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
    </ns0:competence>
    <ns0:competence>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
    </ns0:competence>
    <ns0:competence>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
    </ns0:competence>
    <ns0:competence>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
    </ns0:competence>
    <ns0:competence>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
    </ns0:competence>
    <ns0:competence>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
    </ns0:competence>
</ns0:competences>
<ns0:measures>
    <ns0:measure>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
    </ns0:measure>
    <ns0:measure>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
    </ns0:measure>
    <ns0:measure>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
    </ns0:measure>
    <ns0:measure>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
    </ns0:measure>
    <ns0:measure>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
    </ns0:measure>
    <ns0:measure>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
    </ns0:measure>
    <ns0:measure>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
    </ns0:measure>
    <ns0:measure>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
    </ns0:measure>
</ns0:measures>
<ns0:completioncriteria>
    <ns0:criterion>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
    </ns0:criterion>
</ns0:completioncriteria>

```

```

        </ns0:criterion>
        <ns0:criterion>
            <ns0:name></ns0:name>
            <ns0:description></ns0:description>
        </ns0:criterion>
        <ns0:criterion>
            <ns0:name></ns0:name>
            <ns0:description></ns0:description>
        </ns0:criterion>
        <ns0:criterion>
            <ns0:name></ns0:name>
            <ns0:description></ns0:description>
        </ns0:criterion>
        <ns0:criterion>
            <ns0:name></ns0:name>
            <ns0:description></ns0:description>
        </ns0:criterion>
        <ns0:criterion>
            <ns0:name></ns0:name>
            <ns0:description></ns0:description>
        </ns0:criterion>
        </ns0:completioncriteria>
    </ns0:activity>
    <ns0:activity>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
        <ns0:tasks>
            <ns0:task>
                <ns0:name></ns0:name>
                <ns0:description></ns0:description>
            </ns0:task>
        </ns0:tasks>
        <ns0:tools>
            <ns0:tool>
                <ns0:name></ns0:name>
                <ns0:description></ns0:description>
            </ns0:tool>
        </ns0:tools>
        <ns0:competences>
            <ns0:competence>
                <ns0:name></ns0:name>
                <ns0:description></ns0:description>
            </ns0:competence>
        </ns0:competences>
        <ns0:measures>
            <ns0:measure>
                <ns0:name></ns0:name>
                <ns0:description></ns0:description>
            </ns0:measure>
        </ns0:measures>
        <ns0:completioncriteria>
            <ns0:criterion>
                <ns0:name></ns0:name>
                <ns0:description></ns0:description>
            </ns0:criterion>
        </ns0:completioncriteria>
    </ns0:activity>
    <ns0:activity>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
        <ns0:tasks>
            <ns0:task>
                <ns0:name></ns0:name>
    
```

```

        <ns0:description></ns0:description>
    </ns0:task>
</ns0:tasks>
<ns0:tools>
    <ns0:tool>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
    </ns0:tool>
</ns0:tools>
<ns0:competences>
    <ns0:competence>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
    </ns0:competence>
</ns0:competences>
<ns0:measures>
    <ns0:measure>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
    </ns0:measure>
</ns0:measures>
<ns0:completioncriteria>
    <ns0:criterion>
        <ns0:name></ns0:name>
        <ns0:description></ns0:description>
    </ns0:criterion>
</ns0:completioncriteria>
</ns0:activity>
<ns0:activity>
    <ns0:name></ns0:name>
    <ns0:description></ns0:description>
    <ns0:tasks>
        <ns0:task>
            <ns0:name></ns0:name>
            <ns0:description></ns0:description>
        </ns0:task>
    </ns0:tasks>
    <ns0:tools>
        <ns0:tool>
            <ns0:name></ns0:name>
            <ns0:description></ns0:description>
        </ns0:tool>
    </ns0:tools>
    <ns0:competences>
        <ns0:competence>
            <ns0:name></ns0:name>
            <ns0:description></ns0:description>
        </ns0:competence>
    </ns0:competences>
    <ns0:measures>
        <ns0:measure>
            <ns0:name></ns0:name>
            <ns0:description></ns0:description>
        </ns0:measure>
    </ns0:measures>
    <ns0:completioncriteria>
        <ns0:criterion>
            <ns0:name></ns0:name>
            <ns0:description></ns0:description>
        </ns0:criterion>
    </ns0:completioncriteria>
</ns0:activity>
<ns0:activity>

```

```

<ns0:name></ns0:name>
<ns0:description></ns0:description>
<ns0:tasks>
  <ns0:task>
    <ns0:name></ns0:name>
    <ns0:description></ns0:description>
  </ns0:task>
</ns0:tasks>
<ns0:tools>
  <ns0:tool>
    <ns0:name></ns0:name>
    <ns0:description></ns0:description>
  </ns0:tool>
</ns0:tools>
<ns0:competences>
  <ns0:competence>
    <ns0:name></ns0:name>
    <ns0:description></ns0:description>
  </ns0:competence>
</ns0:competences>
<ns0:measures>
  <ns0:measure>
    <ns0:name></ns0:name>
    <ns0:description></ns0:description>
  </ns0:measure>
</ns0:measures>
<ns0:completioncriteria>
  <ns0:criterion>
    <ns0:name></ns0:name>
    <ns0:description></ns0:description>
  </ns0:criterion>
</ns0:completioncriteria>
</ns0:activity>
<ns0:activity>
  <ns0:name></ns0:name>
  <ns0:description></ns0:description>
  <ns0:tasks>
    <ns0:task>
      <ns0:name></ns0:name>
      <ns0:description></ns0:description>
    </ns0:task>
  </ns0:tasks>
  <ns0:tools>
    <ns0:tool>
      <ns0:name></ns0:name>
      <ns0:description></ns0:description>
    </ns0:tool>
  </ns0:tools>
  <ns0:competences>
    <ns0:competence>
      <ns0:name></ns0:name>
      <ns0:description></ns0:description>
    </ns0:competence>
  </ns0:competences>
  <ns0:measures>
    <ns0:measure>
      <ns0:name></ns0:name>
      <ns0:description></ns0:description>
    </ns0:measure>
  </ns0:measures>
  <ns0:completioncriteria>
    <ns0:criterion>
      <ns0:name></ns0:name>

```

```
        <ns0:description></ns0:description>
      </ns0:criterion>
    </ns0:completioncriteria>
  </ns0:activity>
</ns0:guide>
</ns0:practice>
</ns0:practices>

</ns0:method>
```

Conclusiones del capítulo 3

Después de revisar las alternativas disponibles para la definición del nuevo lenguaje KBL, se concluye que a través de la tecnología XML Schema Definition se pueden definir los elementos de KUALI-BEH, como son “Métodos”, “Prácticas”, “Productos de Trabajo” entre otros, y que son necesarias esas definiciones para poder verificar propiedades de suficiencia, consistencia y coherencia de los métodos, otros conceptos también descritos en KUALI-BEH, para verificar estas propiedades de manera automática, podrían existir varias tecnologías de información, y esas tecnologías podrían tener un largo camino antes de llegar a convertirse en herramientas de software que lo hagan de manera automática, sin embargo, ese camino, sin lugar a dudas, pasa por la definición hecha a la medida, como se hizo en este nuevo esquema, y que pretender ser usado para contener, transferir y procesar ese tipo de información, tratando de cubrir los requerimientos mínimos necesarios para tal propósito, y dejar abierta la posibilidad de evolución.

Otro punto importante es que a pesar de la existencia de herramientas de apoyo para el diseño de esquemas XSD, que incluyen generadores automáticos de código, el diseño manual permite tomar decisiones acordes a las necesidades específicas del contexto único que ahora se presenta para KUALI-BEH.

Capítulo 4 – Experimento de comprobación del nuevo esquema para KUALI-BEH

Este capítulo describe el procedimiento de la validación del esquema propuesto, desde su planteamiento hasta el desarrollo.

Planteamiento del experimento de validación

Una vez definido el nuevo lenguaje KBL, se pueden crear archivos que, en términos de KUALI-BEH, contienen un método de software. Estos archivos pueden ser compartidos en medios electrónicos, por ejemplo, mediante herramientas de software como correo electrónico, servidores de archivos, bases de datos, entre otras. Y en cuanto a funcionalidad, estos archivos se pueden almacenar, imprimir, editar, o incluso más específico, para archivos que contienen un método de software, pueden ser procesados para verificar las propiedades de los métodos contenidos y definidos en términos de KUALI-BEH.

Las herramientas de software que procesen información de acuerdo a KUALI-BEH, y que pretendan utilizar el nuevo lenguaje KBL para compartir su información, requieren “exportar” sus datos al formato KBL definido. Por otra parte, en el caso de aceptar información en el mismo formato, requieren “importar” los archivos KBL y ayudarse del esquema `kualibehMethod.xsd` para validar su gramática, y de esta manera asegurarse de trabajar con archivos “válidos”.

Para comprobar el funcionamiento correcto del nuevo lenguaje, es necesario crear un ejemplo de KBL, es decir, un archivo de texto con sintaxis XML que contenga información típica de un método de software, y que siga la gramática definida en el esquema `kualibehMethod.xsd`. Posteriormente, se verificará su sintaxis y gramática con alguna herramienta XSD, escogiendo cualquiera disponible en el mercado o de uso libre en Internet. Sin embargo, una herramienta creada ex profeso con el objetivo de manejar información de KUALI-BEH, y que específicamente verifique de forma semiautomática las propiedades de sus métodos, permitirá cubrir un rango más amplio de posibles situaciones que pueden surgir a la hora de crear o adecuar software.

Dado que KUALI-BEH es un meta-modelo de creación reciente, aún no existe software disponible en el mercado que permita su uso. Por lo tanto, se propone la creación de una herramienta de software que tenga como objetivo el manejo de métodos y prácticas creadas con KUALI-BEH, en particular que provea un mecanismo de verificación semiautomática de sus

propiedades. Dicha herramienta pretende comprobar la factibilidad del nuevo lenguaje KBL mediante la ejecución de un caso de prueba reducido.

Kualibs 1.0: herramienta de software para validar el lenguaje propuesto.

Para comprobar la factibilidad de KBL, se creó Kualibs 1.0, la herramienta de software que responde al planteamiento descrito. Dicha herramienta utiliza una base de datos relacional para manejar los conceptos KUALI-BEH como información interna. Asimismo, su diseño permite manejar métodos y prácticas, específicamente, editar el contenido de cada elemento y asignar prácticas a métodos. Para realizar esta última función, Kualibs 1.0 utiliza una tabla de métodos y otra de prácticas, relacionadas entre sí con llaves correspondientes. Esta alternativa tecnológica de bases de datos relacionales fue seleccionada con el propósito de contrastarla con el uso de la tecnología del nuevo formato (Esquemas de XML o XSD); ésta última también es parte de Kualibs 1.0, pero es utilizada para manejar la información externa del sistema que proviene o se trasladará a otros sistemas, mediante las funciones de exportación e importación de métodos.

Por un lado, la información KUALI-BEH puede ser manejada a través de diferentes tecnologías; por otro lado el nuevo lenguaje propuesto es suficiente para compartir la información a través de medios digitales. Cabe señalar que también sería posible compartir la misma información por otros medios, por ejemplo, conectando de manera directa las bases de datos, sin embargo representaría un proceso un tanto difícil de repetir ya que cada sistema supone un diseño específico y único.

La Figura 7 exhibe el diagrama de la base de datos relacional, específico para Kualibs 1.0.

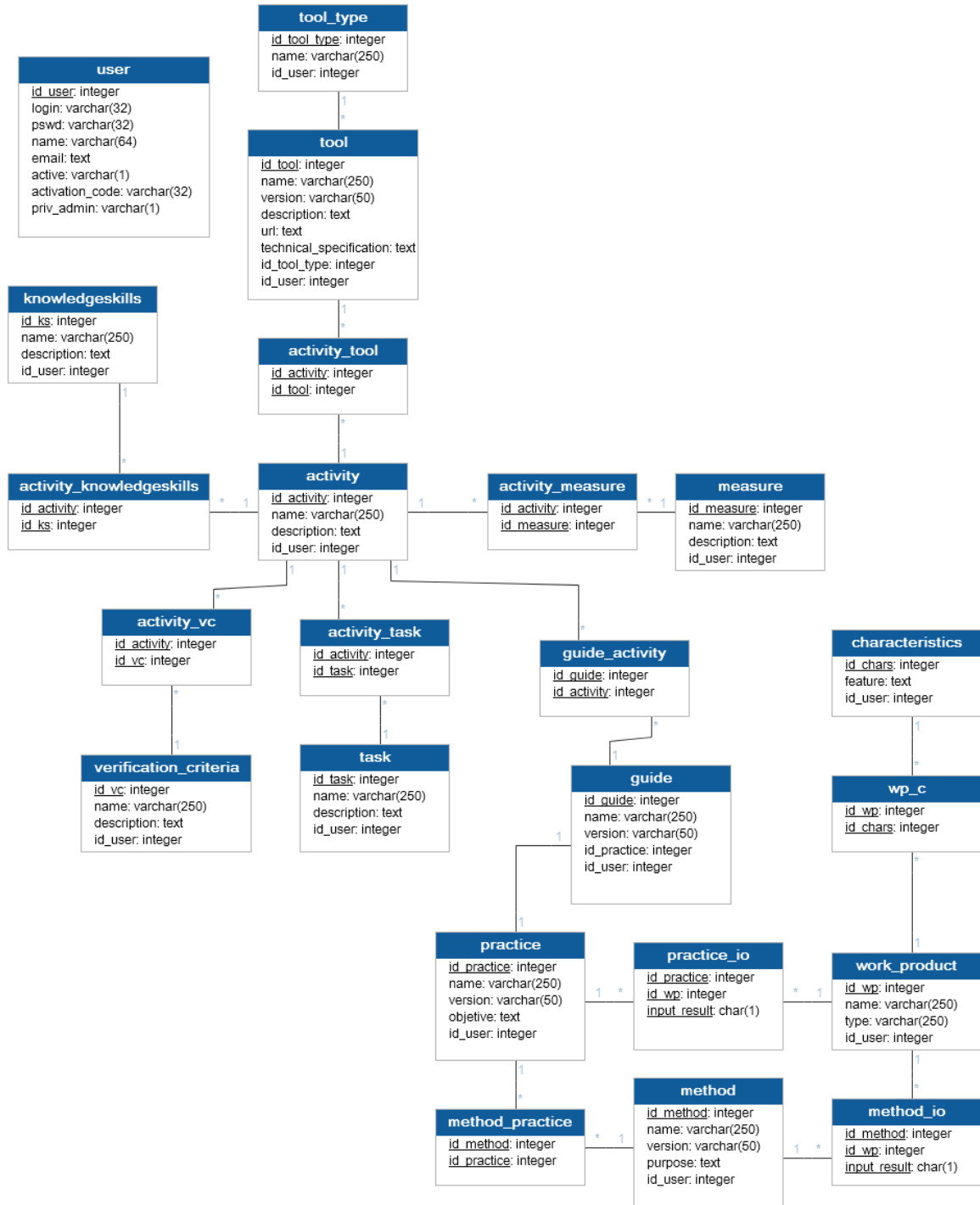


Figura 7 Diagrama de Base de Datos Relacional para Información KUALI-BEH

En este diagrama de la base de datos de Kualibs 1.0 se puede notar que, además de la información referente a KUALI-BEH, se maneja información propia del diseño único del sistema, como es el caso de la información de usuarios. Cada usuario puede manejar su propia

información, ya que el diseño de la herramienta permite explotarla simultáneamente por varios usuarios. Esta característica facilita la comprobación de uso del nuevo esquema KBL, debido a que la información de un usuario puede exportarse y luego ser importada por un segundo usuario, y al mismo tiempo continuar siendo visible para ambos dentro de la herramienta.

La lista de las características generales de la herramienta Kualibs 1.0 se presenta a más detalle en la siguiente sección.

Especificaciones generales de la herramienta de software de comprobación

Las características principales de Kualibs 1.0 han sido diseñadas, en primer lugar, con la intención de validar el uso del nuevo esquema; sin embargo, también se tomó en cuenta la intención de promover el uso y la diseminación de KUALI-BEH. A continuación se definen las características de KUALI-BEH, y posteriormente las de la herramienta Kualibs 1.0

Características de manejo de conceptos KUALI-BEH dentro de Kualibs 1.0

- Se manejan catálogos para:
 - Tipos de herramientas (Tool Types): clasificación de tipos de herramientas utilizadas en las actividades.
 - Herramientas (Tools): herramientas utilizadas para llevar a cabo las actividades.
 - Competencias (Competences): habilidades y conocimientos necesarios para realizar una actividad.
 - Criterios de finalización (Completion Criteria): determinan si una actividad está completa.
 - Métricas (Measures): medidas utilizadas en una actividad.
 - Características (Features): características de los productos de trabajo.
 - Productos de Trabajo (Work Products): abarcan productos de trabajo, entradas y resultados de las prácticas y métodos.
- La Infraestructura (Infrastructure): es el depósito que acumula métodos y prácticas de una organización y los maneja a través de catálogos:
 - Tareas (Tasks): son los pasos a seguir dentro de una actividad.
 - Actividades (Activities): son los elementos que constituyen una práctica.
 - Guías (Guides): son los documentos que especifican y detallan la manera en que ha de realizarse una práctica. Incluyen actividades, herramientas, tareas, métricas y criterios de finalización requeridos. Es el documento imprimible en formato PDF, que sirve a los usuarios para llevar a cabo una práctica.

- Prácticas (Practices): Permite definir las entradas (Inputs) y resultados (Results), seleccionables del catálogo de Productos de Trabajo (Work Products).
- Métodos (Methods): Permite manejar los elementos más generales de KUALI-BEH, agrupar Prácticas y Productos de Trabajo, formando un Método. Permite verificar las propiedades del Método e importar y exportar un archivo KBL válido.

Características de uso de la herramienta Kualibs 1.0

- Multi-usuario: el sistema permite múltiples usuarios, concediendo su uso a través del registro de un nombre de usuario (log-in) y contraseña.
- Multi-idioma: el sistema dispone de versiones en español y en inglés.
- Editor de texto compacto para HTML: que permite al usuario formatear de manera básica el contenido de un texto largo; por ejemplo, incluir listas, alinear de párrafos, cambiar la fuente, resaltar texto, entre otras. Eso debido a que la definición de métodos y prácticas, en general, se da en forma de “texto libre” redactado por los usuarios.
- Formato de Impresión para Guías y Métodos, basado en las plantillas que se proporcionan en la especificación KUALI-BEH 1.1.

Características Técnicas de la Herramienta Kualibs 1.0:

- Gestor de Base de Datos: MySQL 5.
- Lenguaje de programación: PHP 5.
- Servidor Web: Apache.
- Herramienta de desarrollo: Scriptcase.
- Manipulación de documentos XML con bibliotecas de código PHP.¹⁹

Una de las ventajas de haber seleccionado XSD, como tecnología para definir el nuevo lenguaje, es que al construir herramientas de software, como en este caso, y que se utilizó PHP, se pueden utilizar bibliotecas de código, que ya otorgan funcionalidad sobre documentos XML, lo que permite ahorrar tiempo de desarrollo.

A continuación se presentan capturas de pantalla de la herramienta Kualibs 1.0 para ejemplificar su uso.

¹⁹ El sitio de documentación “oficial” de PHP (<http://php.net/manual/es/refs.xml.php>), contiene referencias a estas bibliotecas del lenguaje, que sirven para trabajar con XML y XSD, entre otras: DOM (Modelo de Objetos de Documento), libxml, SDO (Objetos de Datos de Servicio), SimpleXML, XMLReader, XMLWriter.

La figura 8 muestra el acceso al catálogo de Productos de Trabajo, dando un ejemplo del comportamiento de las pantallas de catálogos y de la infraestructura. Comportamiento similar para el resto de catálogos de todo el sistema Kualibs 1.0.

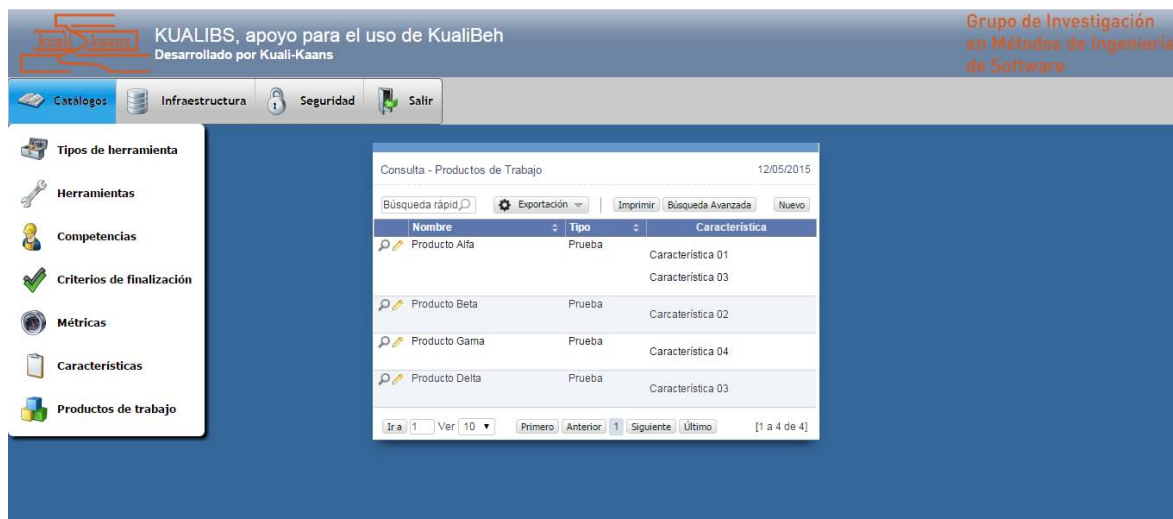


Figura 8 Pantalla de Catálogo de Productos de Trabajo, sistema Kualibs 1.0

En esta figura se aprecia la pantalla presentada por el sistema, en la que fue seleccionada la opción “Catálogos -> Productos de Trabajo” dentro de las opciones del menú principal. Idéntica a las demás opciones dentro de la lista de los Catálogos, tiene alternativas de navegación, exportación, impresión, búsqueda avanzada y generación de un nuevo elemento.

La figura 9 muestra el procedimiento para imprimir una guía en formato PDF, ingresando a la opción “Infraestructura -> Guías”.

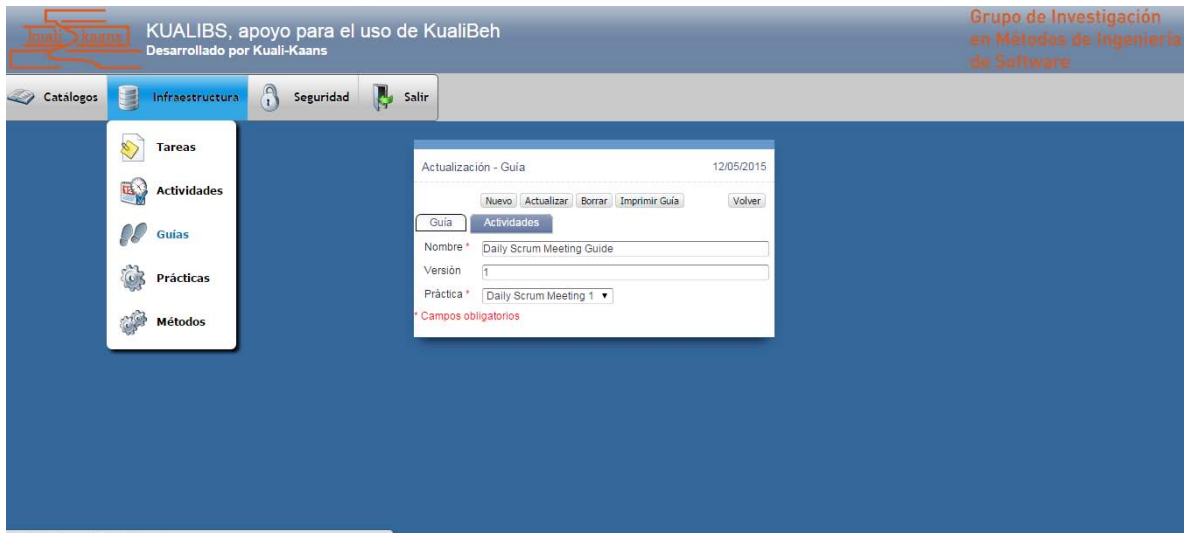


Figura 9 Pantalla de Guías, ingreso a opción de impresión en PDF

Al presionar el botón “Imprimir Guía” se obtiene el resultado presentado en la figura 10.

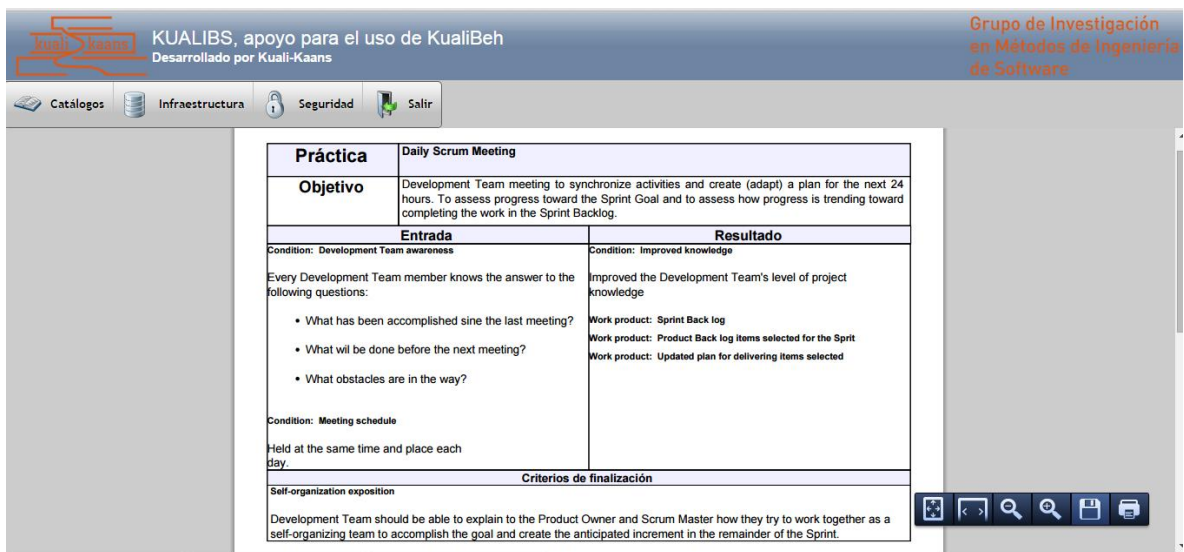


Figura 10 Captura de pantalla al oprimir el botón “Imprimir Guía”

En la figura anterior se presenta la pantalla, que resulta de imprimir la guía, de una “Práctica”, en este caso como ejemplo, se definió la Práctica “Reunión Diaria de SCRUM” (Daily Scrum Meeting), una de las prácticas pertenecientes al Método “SCRUM”; este mismo ejemplo se presentó en la especificación KUALI-BEH 1.1 (Oktaba & Morales Trujillo, 2012), con la intención de enseñar a los equipos de trabajo cómo puede aplicarse KUALI-BEH a una metodología ya

conocida. El formato de los documentos imprimibles “Guía” y “Método” se generó basándose en las plantillas propuestas en KUALI-BEH 1.1.

Las funciones aplicables a los métodos, tales como verificación de propiedades, importación y exportación de un método completo en un archivo KBL, se acceden desde la pestaña del menú principal “Infraestructura”, desplazándose a la opción “Método”.

Si bien en esta sección sólo se presentan algunas características generales y de uso de la herramienta en cuestión, en el anexo A se puede consultar una guía completa para su uso, utilizando como ejemplo el Método SCRUM y la Práctica “Reunión Diaria de SCRUM”.

Verificación de Propiedades de Métodos con Kualibs 1.0

En esta sección se discutirá el caso de la verificación de propiedades del Método con la herramienta Kualibs 1.0.

Las propiedades del Método incluyen consistencia, coherencia y suficiencia, de las cuales la propiedad de consistencia puede ser establecida de manera automática. Eso se debe a que la herramienta comprueba que las entradas y los resultados de los productos de trabajo coincidan entre todas las prácticas del mismo método. La herramienta de software lo indica con colores, al haber corroborado las siguientes cuatro condiciones:

- ¿Existe(n) alguna(s) Práctica(s) que tienen como Entrada el(los) mismo(s) Producto(s) de Trabajo que la Entrada del Método?
Si la respuesta es afirmativa, la pantalla de verificación despliega el nombre de la(s) Práctica(s) remarcado en color verde. Si la respuesta es negativa, el sistema despliega un recuadro vacío en color rojo, indicando la inconsistencia.
- ¿Existe(n) Resultado(s) de alguna(s) Práctica(s) que **no** está(n) siendo utilizado(s) como entrada de alguna(s) Práctica(s) o como Resultado del Método?
En caso de una respuesta afirmativa, el sistema despliega el nombre de la(s) Práctica(s) remarcado(s) en rojo. En caso de una respuesta negativa, el sistema despliega un recuadro vacío en color verde, ya que es lo esperado.
- ¿Existe una Práctica con productos de trabajo requeridos como su Entrada, pero que ninguna otra Práctica los está generando, o no son Entrada del Método? En caso afirmativo, el sistema muestra el nombre de dicha Práctica en color rojo, ya que esto resulta en una definición incorrecta; en caso negativo, se despliega un recuadro vacío en color verde, ya que es lo esperado.

- ¿Existe alguna(s) Práctica(s) que tiene(n) como Resultado el(los) mismo(s) Producto(s) de Trabajo que el Resultado del Método?
Si la respuesta es afirmativa, la pantalla de verificación presenta el nombre de la(s) Práctica(s) en color verde, ya que es lo esperado; si la respuesta que se da es negativa, el sistema muestra un recuadro vacío en color rojo.

En cuanto a las propiedades de coherencia y suficiencia, el sistema se limita a desplegar un listado de los objetivos de las Prácticas a un lado del propósito general del Método, lo cual facilita al usuario la lectura y análisis de los mismos. En este caso el usuario es quien tomará la decisión sobre las propiedades, al leer el texto libre y determinar si los Objetivos son suficientes para alcanzar el Propósito y/o si son coherentes con éste.

En la siguiente figura (Fig.11) se muestra un ejemplo de la pantalla resultado de la verificación de propiedades del Método, donde se pueden distinguir los colores utilizados por Kualibs 1.0.

KUALIBS, apoyo para el uso de KualiBeh
Desarrollado por Kuali-Kaans

Grupo de Investigación en Métodos de Ingeniería de Software

Catálogos Infraestructura Seguridad Salir

| Apoyo a la verificación de Propiedades | | | |
|---|--------------------------------------|--|---|
| Método: Scrum | | | |
| Consistencia | | | |
| Practica(s) con Entrada similar a Entrada de Método | Practica(s) con Resultados no usados | Practica(s) con Entradas no proporcionadas | Practica(s) con Resultado similar a Resultado de Método |
| • Daily Scrum Meeting | • Daily Scrum Meeting | • Daily Scrum Meeting | • Daily Scrum Meeting |

| Coherencia & Suficiencia | |
|-----------------------------|---|
| Propósito del Método | Objetivos de Prácticas |
| To agility develop software | Práctica: Daily Scrum Meeting Objetivo: Development Team meeting to synchronize activities and create (adapt) a plan for the next 24 hours. To assess progress toward the Sprint Goal and to assess how progress is trending toward completing the work in the Sprint Backlog. |

Figura 11 Resultado de verificación de propiedades del Método

Caso de prueba reducido

Dado el vasto alcance de la información que se puede manejar con KUALI-BEH, y el hecho de que las Prácticas y Métodos son descritos en formato de texto libre por sus autores, se usará un caso de prueba reducido. Este caso de prueba reducido tendrá como objetivo comprobar que la herramienta Kualibs 1.0, por un lado, ejecuta la verificación tal y como se describió en las especificaciones generales, y por otro lado, exporta e importa un archivo que satisface el nuevo

lenguaje KBL y de esta forma puede compartir la información entre varios usuarios del sistema. El cumplimiento de esta última funcionalidad es de gran importancia, ya que de otra manera no se lograría la distribución de la información y conocimiento, dado que el sistema restringe a cada usuario sólo ver y manejar su propia información.

Representación gráfica del caso de prueba reducido

A través de la nomenclatura gráfica propuesta en la especificación de KUALI-BEH 1.1, se presenta un esquema para el caso de prueba reducido (ver Fig.12).

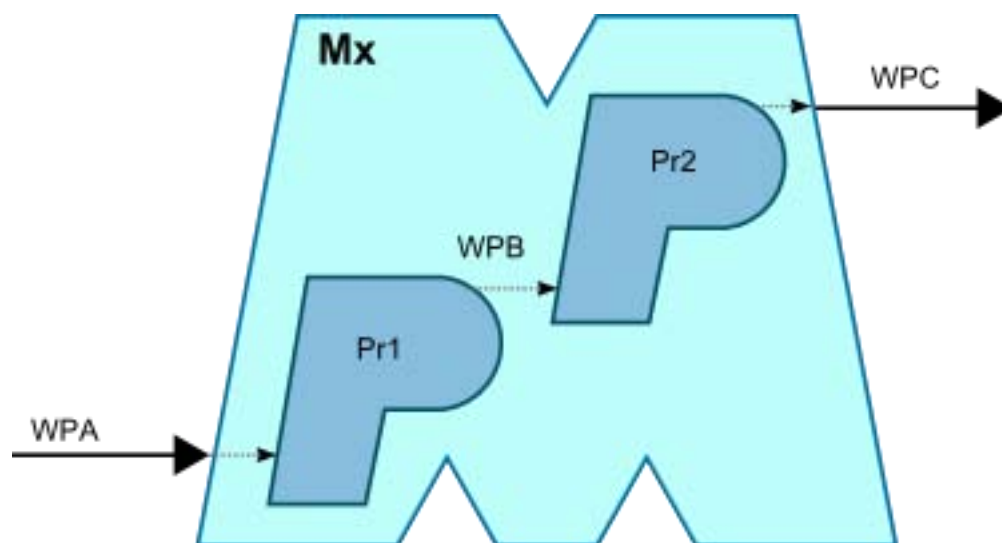


Figura 12 Diagrama de caso de prueba reducido

En esta figura se aprecian las siguientes características del caso de prueba reducido:

- Un sólo Método nombrado Mx.
- Dos Prácticas: Pr1 y Pr2.
- Tres Productos de trabajo: WPA, WPB y WPC
- El producto de trabajo WPA es la Entrada, tanto del Método Mx, como de la Práctica Pr1.
- El producto de trabajo WPB es Resultado de Pr1, y al mismo tiempo, funciona como Entrada de Pr2.
- El producto de trabajo WPC es Resultado, tanto del Método Mx, como de la Práctica Pr2.

Estructura del caso de prueba reducido

En la representación gráfica (Fig.12), no es posible distinguir los detalles necesarios del caso de prueba reducido, específicamente lo referente a la comprobación de la generación

correcta del archivo con KBL. Sin embargo, la estructura de dichos detalles se presenta en la Tabla 5.

| Tipo de elemento | Nombre y Descripción | Relación |
|--------------------------|-----------------------------|-------------------------------------|
| Método | Mx | |
| Práctica | PrA | Práctica de Mx |
| Práctica | PrB | Práctica de Mx |
| Producto de Trabajo | WPA | Entrada de Mx y Entrada de PrA |
| Característica | WPAf1 | Característica de WPA |
| Producto de Trabajo | WPB | Resultado de PrA y Entrada de PrB |
| Característica | WPBf1 | Característica de WPB |
| Producto de Trabajo | WPC | Resultado de PrB y Resultado de Mx |
| Característica | WPCf1 | Característica de WPC |
| Guía | PrAG1 | Guía de PrA |
| Actividad | PrAG1A1 | Actividad de PrAG1 |
| Conocimiento y Habilidad | PrAG1A1KS1 | Conocimiento y Habilidad de PrAG1A1 |
| Criterio de Verificación | PrAG1A1VC1 | Criterio de verificación de PrAG1A1 |
| Guía | PrBG1 | Guía de PrB |
| Actividad | PrBG1A1 | Actividad de PrBG1 |
| Conocimiento y Habilidad | PrBG1A1KS1 | Conocimiento y Habilidad de PrBG1A1 |
| Criterio de Verificación | PrBG1A1VC1 | Criterio de verificación de PrBG1A1 |

Tabla 5 Relación entre elementos de Método de prueba MX

Resultado de Verificación

Una vez registrado el caso de prueba reducido dentro de la herramienta Kualibs 1.0, se obtuvieron las siguientes pantallas de ejemplo para Productos de Trabajo y para Prácticas, ver Fig. 13 y 14.

Consulta - Productos de Trabajo 13/05/2015

Búsqueda rápida | |

| Nombre | Tipo | Característica |
|--------|------|----------------|
| WPA | | WPAf1 |
| WPB | | WPBf1 |
| WPC | | WPCf1 |

Ira 1 Ver 10 1 [1 a 3 de 3]

Figura 13 Pantalla de Catálogo de Productos de Trabajo

Consulta - Prácticas 13/05/2015

Búsqueda rápida | |

| Nombre | Versión | Objetivo |
|--------|---------|-----------------|
| PrA | | Objetivo de PrA |
| PrB | | Objetivo de PrB |

Ira 1 Ver 10 1 [1 a 2 de 2]

Figura 14 Pantalla de Consulta de Prácticas

Sin embargo, el contenido más completo se aprecia en la impresión de las Guías, ya que éstas concentran las actividades relacionadas a la Práctica, su objetivo, los productos de trabajo asociados, sus características y criterios de verificación, ver Fig.15 y 16.

| | | | |
|----------------------------------|---------------------|---------------------|-----------------|
| Práctica | PrA | | |
| Objetivo | Objetivo de PrA | | |
| Entrada | | Resultado | |
| WPA | | WPB | |
| WPAf1 | | WPBf1 | |
| Criterios de finalización | | | |
| PrAG1A1VC1 | | | |
| PrAG1A1VC1 | | | |
| Guía | | | |
| Actividad | PrAG1A1 | | |
| | PrAG1A1 | | |
| Tareas | Herramientas | Competencias | Métricas |
| | | PrAG1A1KS1 | |
| | | PrAG1A1KS1 | |

Figura 15 Guía PrAG1 de Caso de Prueba Reducido

| | | | |
|----------------------------------|---------------------|---------------------|-----------------|
| Práctica | PrB | | |
| Objetivo | Objetivo de PrB | | |
| Entrada | | Resultado | |
| WPB | | WPC | |
| WPBf1 | | WPCf1 | |
| Criterios de finalización | | | |
| PrBG1A1VC1 | | | |
| PrBG1A1VC1 | | | |
| Guía | | | |
| Actividad | PrBG1A1 | | |
| | PrBG1A1 | | |
| Tareas | Herramientas | Competencias | Métricas |
| | | PrBG1A1KS1 | |
| | | PrBG1A1KS1 | |

Figura 16 Guía PrBG1 de Caso de Prueba Reducido

La figura 17 presenta la impresión del Método Mx generada con el sistema Kualibs 1.0.

| | | |
|------------------------|-----------------|------------------|
| Método | Mx | |
| Propósito | Propósito de MX | |
| Entrada | | Resultado |
| : WPA WPAf1 | | : WPC WPCf1 |
| Prácticas | | |
| PrA Objetivo de PrA | | |
| PrB Objetivo de PrB | | |

Figura 17 Impresión de Método Mx de Caso de Prueba Reducido

La figura que se presenta a continuación es una impresión de la verificación de propiedades del Método MX, en la que es posible constatar que la propiedad de consistencia queda cumplida en este caso de prueba, puesto que las 4 condiciones aparecen en color verde, representando lo correcto. Asimismo, se puede confirmar que los objetivos de las Prácticas aparecen en forma de listado junto al Propósito del Método, todo con la finalidad de simplificar la verificación manual de las propiedades de coherencia y suficiencia por el usuario, ver Fig. 18.

| Apoyo a la verificación de Propiedades | | | |
|---|--------------------------------------|--|---|
| Método | Mx | | |
| Consistencia | | | |
| Practica(s) con Entrada similar a Entrada de Método | Practica(s) con Resultados no usados | Practica(s) con Entradas no proporcionadas | Practica(s) con Resultado similar a Resultado de Método |
| • PrA | Ninguno | | • PrB |
| Coherencia & Suficiencia | | | |
| Propósito del Método | Objetivos de Prácticas | | |
| Propósito de MX | Práctica: PrA Objetivo: | | |
| | Objetivo de PrA | | |
| | Práctica: PrB Objetivo: | | |
| | Objetivo de PrB | | |

Figura 18 Verificación de Propiedades de Método Mx

Por otro lado, se generó un archivo KBL que define el método MX y cuyo contenido se representa de la siguiente manera:

```
<?xml version="1.0" encoding="UTF-8"?>

<ns0:method xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
             xmlns:ns0='http://kualibeh.mx/schema/kualibehMethod'
             xsi:schemaLocation='http://kualibeh.mx/schema/kualibehMethod
kualibehMethod.xsd'>
  <ns0:name>Mx</ns0:name>
  <ns0:purpose>Propósito de Mx</ns0:purpose>
  <ns0:entry>
    <ns0:workproduct>
      <ns0:name>WPA</ns0:name>
      <ns0:feature>WPAf1</ns0:feature>
    </ns0:workproduct>
  </ns0:entry>
  <ns0:result>
    <ns0:workproduct>
      <ns0:name>WPC</ns0:name>
      <ns0:feature>WPCf1</ns0:feature>
    </ns0:workproduct>
  </ns0:result>
  <ns0:practices>
    <ns0:practice>
      <ns0:name>PrA</ns0:name>
      <ns0:objective>Objetivo de PrA</ns0:objective>
      <ns0:entry>
        <ns0:workproduct>
          <ns0:name>WPA</ns0:name>
          <ns0:feature>WPAf1</ns0:feature>
        </ns0:workproduct>
      </ns0:entry>
    </ns0:practice>
  </ns0:practices>
  <ns0:result>
    <ns0:workproduct>
      <ns0:name>WPB</ns0:name>
      <ns0:feature>WPBf1</ns0:feature>
    </ns0:workproduct>
  </ns0:result>
  <ns0:guide>
    <ns0:activity>
      <ns0:name>PrAG1A1</ns0:name>
      <ns0:tasks>
        <ns0:task>
          <ns0:name></ns0:name>
        </ns0:task>
      </ns0:tasks>
      <ns0:competences>
        <ns0:competence>
          <ns0:name>
            PrAG1A1KS1
          </ns0:name>
          <ns0:description>
            PrAG1A1KS1
          </ns0:description>
        </ns0:competence>
      </ns0:competences>
      <ns0:completioncriteria>
        <ns0:criterion>
          <ns0:name>
            PrAG1A1VC1
          </ns0:name>
        </ns0:criterion>
      </ns0:completioncriteria>
    </ns0:activity>
  </ns0:guide>
</ns0:method>
```

```

        </ns0:name>
        <ns0:description>
            PrAG1A1VC1
        </ns0:description>
    </ns0:criterion>
</ns0:completioncriteria>
</ns0:activity>
</ns0:guide>
</ns0:practice>
<ns0:practice>
    <ns0:name>PrB</ns0:name>
    <ns0:objective>Objetivo de PrB</ns0:objective>
    <ns0:entry>
        <ns0:workproduct>
            <ns0:name>WPB</ns0:name>
            <ns0:feature>WPBf1</ns0:feature>
        </ns0:workproduct>
    </ns0:entry>
    <ns0:result>
        <ns0:workproduct>
            <ns0:name>WPC</ns0:name>
            <ns0:feature>WPCf1</ns0:feature>
        </ns0:workproduct>
    </ns0:result>
    <ns0:guide>
        <ns0:activity>
            <ns0:name>PrBG1A1</ns0:name>
            <ns0:tasks>
                <ns0:task>
                    <ns0:name></ns0:name>
                </ns0:task>
            </ns0:tasks>
            <ns0:competences>
                <ns0:competence>
                    <ns0:name>
                        PrBG1A1KS1
                    </ns0:name>
                    <ns0:description>
                        PrBG1A1KS1
                    </ns0:description>
                </ns0:competence>
            </ns0:competences>
            <ns0:completioncriteria>
                <ns0:criterion>
                    <ns0:name>
                        PrBG1A1VC1
                    </ns0:name>
                    <ns0:description>
                        PrBG1A1VC1
                    </ns0:description>
                </ns0:criterion>
            </ns0:completioncriteria>
        </ns0:activity>
    </ns0:guide>
</ns0:practice>
</ns0:practices>
</ns0:method>

```

Al analizar este listado, se comprueba que cumple con la sintaxis y la gramática establecida, por lo que se considera como un archivo bien formado y válido.

Conclusiones de capítulo 4

Con los resultados obtenidos en el caso de prueba reducido se pudo comprobar el funcionamiento puntual del sistema Kualibs 1.0. La verificación satisfactoria de las propiedades y la correcta generación del archivo KBL, los objetivos propuestos para este trabajo desde el principio han sido alcanzados. Aún más, se implementaron funcionalidades que no habían sido previstas para esta versión de Kualibs 1.0, como son: creación de una versión web y el uso de diferentes idiomas. Pero se implementaron ahora, como un valor agregado, para aumentar la posibilidad de una mayor difusión de esta herramienta, y por lo tanto, del conocimiento y la tecnología desarrollada, apoyando no sólo el cumplimiento de los objetivos de este lenguaje XML, sino también el del desarrollo original de la propuesta KUALI-BEH. El alcanzar los objetivos de este trabajo de investigación da continuidad al propósito con el cual fue creado KUALI-BEH – estructurar e intercambiar el conocimiento existente.

El hecho de que, al desarrollar una herramienta, que, aunque no deja de ser experimental, cumpliera con requerimientos típicos de software comercial, trajo como consecuencia otro resultado no previsto: probar que la tecnología seleccionada, XML, corresponde a un nivel aceptable de adaptación. Aunque de antemano se podía intuir la madurez y lo recomendable de la tecnología XML para desarrollar este tipo de sistemas, dicha tecnología demostró su eficacia a la hora de formatear con etiquetas HTML básicas un texto largo, como es el caso de la información manejada con KUALI-BEH que debe ser impresa en una plantilla para facilitar su seguimiento y aplicación.

Capítulo 5 – Resultados

Durante la realización del presente trabajo, han habido varios cambios en la definición del esquema para KUALI-BEH, determinados, por un lado, por la insuficiencia del conocimiento inicial, o bien, por cambios en los conceptos mismos. Debido a que la construcción de este proyecto se inició con la especificación KUALI-BEH 1.0 y se finalizó con la versión 1.1. Ha sido grata la experiencia de observar y afirmar que la flexibilidad de la tecnología XML y XSD utilizada en este trabajo, permitió realizar dichos cambios sin mayores complicaciones. La misma experiencia se espera para el desarrollo de versiones futuras de la herramienta, dado que KUALI-BEH no es una propuesta final, sino que se espera que esté constantemente actualizada. La flexibilidad de la tecnología antes mencionada posibilitará que la herramienta evolucione de acuerdo a las necesidades de la industria, y que el nuevo lenguaje KBL crezca en conjunto con KUALI-BEH.

Además, el recurrir a la tecnología XML se favoreció un desarrollo rápido de la herramienta, en particular en algunos módulos. El desarrollo de la herramienta utilizó código libre y público, como es la biblioteca de PHP para manejar DOM²⁰ (Document Object Model), lo que permite crear documentos de XML, o validarlos con un esquema XSD, y ahorrar tiempo, al no tener que escribir funciones propias. Asimismo, se experimentó un desarrollo más ágil gracias a tales herramientas como NetBeans que ya tienen incorporadas de antemano utilerías que ocultan la parte repetitiva y propensa a error y son de gran ayuda en el manejo de esquemas y patrones de diseño XSD.

Conclusiones

Aún en la propuesta inicial para la creación del nuevo esquema para KUALI-BEH se consideró que el nuevo lenguaje KBL no llegaría a una versión final y definitiva. En general, los lenguajes y/o aplicaciones de XML sufren adecuaciones consecuencia de nuevos requerimientos, tomando como ejemplo el mismo HTML que se ha desarrollado desde la versión 1 hasta la versión 5 de la actualidad, por lo que se asume que KBL se desempeñe/funcione de forma similar. Sin embargo, no cabe duda que la evolución de una tecnología es impulsada por la comunidad, por

²⁰ DOM es un modelo, definido como estándar por el World Wide Web Consortium (W3C), que maneja clases que representan documentos HTML y XML.

ejemplo, a través de desarrollo de aplicaciones a su alrededor; o, en el caso contrario, es olvidada por la comunidad por dejar de ser práctica o útil y fácilmente cae en desuso.

En el caso de KBL y para cumplir con el objetivo de comprobar su verdadero potencial, aún son necesarias muchas herramientas a su alrededor. Como se mencionó en el capítulo del caso de prueba para el nuevo esquema, se tenía la posibilidad de realizarse con herramientas ya existentes que sólo verificaran sintaxis y gramática, sin poner a prueba la factibilidad de su uso en contextos reales. Por lo tanto surgió la necesidad de crear desde cero todo un sistema alrededor del nuevo esquema KBL, y aun así no se alcanzó una visión completamente clara del potencial de este nuevo lenguaje. En consecuencia, se debe lograr una mayor conciencia del problema y del alcance, no sólo del nuevo esquema, de los objetivos de KUALI-BEH, sino visualizando más allá, de mejorar la manera de trabajar en la industria de software en general. Es menester actuar en consecuencia, de ahora en adelante, para que se logre y se aproveche la difusión del conocimiento y de una mejor ingeniería de software, y con aún más prioridad, la difusión de la imperiosa necesidad de hacerlo.

Trabajo futuro

El trabajo de investigación en cuestión intenta colaborar con los objetivos originales sobre los cuales se basa la creación de KUALI-BEH: ayudar los practicantes de la industria de software a través de una guía para definir de manera más eficiente su conocimiento tácito, y posteriormente poder compartir las mejores prácticas de trabajo para generar conocimiento con bases científicas.

Aunque se haya propuesto y validado un esquema de intercambio digital de información bajo el modelo KUALI-BEH, el aspecto crucial para este trabajo, queda pendiente la incorporación de la vista dinámica de KUALI-BEH a la herramienta, así como también un periodo de maduración del lenguaje KBL, promoviendo su aplicación en contextos académicos e industriales con la finalidad de mayor retroalimentación, mejora constante y uso más amplio.

Hay mucho camino por delante del nuevo esquema KBL; actualmente es sólo una base del intercambio de información, no obstante, esa información requiere ser procesada, impresa, formateada y verificada ante otras operaciones. El esquema propuesto previene cierto nivel de adaptación, a pesar de que la dicha adaptación ya pertenecerá al software que posiblemente se

desarrollará alrededor de los conceptos de KUALI-BEH. Por lo tanto, quizá el mayor trabajo a futuro es seguir contribuyendo a la difusión de KUALI-BEH, no sólo por ser un trabajo sólido y útil, o por ser orgullosamente mexicano, o por ser una propuesta reconocida a nivel internacional, sino porque el motivo original de su creación es altamente importante, siendo en primer lugar, la omnipresente mejora en la industria del software.

Referencias

- Chung, S., & Hartford, E. (2006). Bridging the Gap between Data Models and Implementations: XMI2SQL. *IEEE: Telecommunications, 2006. AICT-ICIW '06. International Conference on Internet and Web Applications and Services/Advanced International Conference on*, 201.
- Ecma International. (2013, Octubre). *Standard ECMA-404*. Retrieved from The JSON Data Interchange Format: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>
- Evans, C. C. (2009, 10 1). *YAML*. Retrieved from <http://www.yaml.org/>: <http://www.yaml.org/>
- Hernández Uribe, E. (2005). *Uso de la tecnología rdf para representar y manejar los procesos moprosoft y su aplicación en him*. Universidad Nacional Autónoma de México. Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas.
- International Organization for Standardization. (2008, 11 15). *International Organization for Standardization*. Retrieved from ISO/IEC 29500-1:2008: http://www.iso.org/iso/catalogue_detail?csnumber=51463
- Kuali-kaans. (Mayo de 2012). *Kuali-kaans*. Obtenido de recursos: <http://kualikaans.fciencias.unam.mx/images/kuali/documentos/KUALI-BEH%20v1.1.pdf>
- Microsoft. (n.d.). *msdn.microsoft.com*. Retrieved from Office Open XML, ECMA-376, and ISO/IEC 29500: [https://msdn.microsoft.com/en-us/library/office/gg607163\(v=office.14\).aspx](https://msdn.microsoft.com/en-us/library/office/gg607163(v=office.14).aspx)
- Oktaba, H. J., & Morales Trujillo, M. E. (2012). OMG Document Number: ad/2012-02-06. *KUALI-BEH: Software Project Common Concepts*. Retrieved from Kuali-Beh 1.1: <http://kualikaans.fciencias.unam.mx/images/kuali/documentos/KUALI-BEH%20v1.1.pdf>
- OMG. (Junio de 2011). *A Foundation for the Agile Creation and Enactment of Software Engineering Methods RFP*. Obtenido de <http://www.omg.org/cgi-bin/doc?ad/2011-6-2616/05/11>
- Paoin, W., & Boonchai-Apisit, P. (2009). Development of surgical operation data interchange model using XML and relational database. *IEEE 978-1-4244-4138-9*.
- Penhaker, M., Krawiec, J., Krejcar, O., Novak, V., & Bridzik, R. (2010). Web System for Electrophysiological Data Management. *IEEE, 978-1-4244-6079-3*.
- Trillo Lares, K. R. (2005). *Arquitectura Genérica para Almacenar Documentos XML en una Base de Datos Relacional*. México, DF.: Universidad Nacional Autónoma de México, Instituto de Investigaciones en Matemáticas Aplicadas y Computación.

- Wang, J., & Yan, L. (2008). Graphic Sharing Based on XML Technology: Analysis and Web Expression of DXF Graphic. *IEEE 978-0-7695-3336-0*.
- World Wide Web Consortium. (2014, 02 25). *RDF*. Retrieved from Resource Description Framework: <http://www.w3.org/RDF/>

Índice de Figuras

| | |
|---|----|
| Figura 1 - Mapa Conceptual de Vista Estática de KUALI-BEH..... | 15 |
| Figura 2 Representación gráfica de un método de ejemplo | 19 |
| Figura 3 Captura de pantalla de ejemplo de la base de datos utilizada en la revisión sistemática..... | 23 |
| Figura 4 Árbol KLB..... | 30 |
| Figura 5 Captura de pantalla de ejemplo de Netbeans para manejo de esquemas XML.. | 36 |
| Figura 6 Elementos del esquema KBL..... | 37 |
| Figura 7 Diagrama de Base de Datos Relacional para Información KUALI-BEH | 52 |
| Figura 8 Pantalla de Catálogo de Productos de Trabajo, sistema Kualibs 1.0 | 55 |
| Figura 9 Pantalla de Guías, ingreso a opción de impresión en PDF | 56 |
| Figura 10 Captura de pantalla al oprimir el botón “Imprimir Guía” | 56 |
| Figura 11 Resultado de verificación de propiedades del Método | 58 |
| Figura 12 Diagrama de caso de prueba reducido..... | 59 |
| Figura 13 Pantalla de Catálogo de Productos de Trabajo | 61 |
| Figura 14 Pantalla de Consulta de Prácticas..... | 61 |
| Figura 15 Guía PrAG1 de Caso de Prueba Reducido | 62 |
| Figura 16 Guía PrBG1 de Caso de Prueba Reducido | 62 |
| Figura 17 Impresión de Método Mx de Caso de Prueba Reducido..... | 63 |
| Figura 18 Verificación de Propiedades de Método Mx | 63 |

Índice de Tablas

| | |
|---|----|
| Tabla 1 Ejemplo "A" de Práctica en KUALI-BEH..... | 17 |
| Tabla 2 Ejemplo "B" de Práctica en KUALI-BEH..... | 18 |
| Tabla 3 Patrones de diseño de esquemas XML | 35 |
| Tabla 4 Guía de creación de documento KBL..... | 42 |
| Tabla 5 Relación entre elementos de Método de prueba MX | 60 |

Apéndice A, Guía Rápida para uso de Kualibs 1.0

Ejemplo con Scrum

Scrum es uno de los conocidos métodos ágiles de software, que incluye la práctica de reunión diaria del equipo de trabajo para definir el rumbo del proyecto, versus los métodos tradicionales que tienen una planeación compleja al inicio, que rara vez se ve cumplida.

De la documentación de KUALI-BEH se extrae la siguiente imagen de la definición de la práctica: Daily Scrum Meeting practice, la cual se usará para ilustrar el uso de Kualibs en esta guía.

| DailyScrum | Practice | |
|--|----------|---|
| <i>Daily Scrum Meeting</i> | | |
| | | |
| Objective | | |
| <i>Development Team meeting to synchronize activities and create (adapt) a plan for the next 24 hours. To assess progress toward the Sprint Goal and to assess how progress is trending toward completing the work in the Sprint Backlog.</i> | | |
| Input | | Result |
| Conditions <ul style="list-style-type: none"> Every Development Team member knows the answer to the following questions: <ul style="list-style-type: none"> What has been accomplished since the last meeting? What will be done before the next meeting? What obstacles are in the way? Held at the same time and place each day. | | Work products <ul style="list-style-type: none"> Sprint Backlog Product Backlog items selected for this Sprint Updated Plan for delivering them Conditions <ul style="list-style-type: none"> Improved the Development Team's level of project knowledge. |
| Guide | | |
| Activities | | |
| <ol style="list-style-type: none"> During the meeting, each Development Team member explains: <ul style="list-style-type: none"> What has been accomplished since the last meeting? What will be done before the next meeting? What obstacles are in the way? The Development Team often meets immediately after the Daily Scrum to re-plan the rest of the Sprint's work. | | |
| Knowledge and Skills | | |
| <i>Development Team consists of professionals who do the work of delivering a potentially releasable Increment of "Done" product at the end of each Sprint.</i> | | |
| Verification Criteria | | |
| <i>Development Team should be able to explain to the Product Owner and Scrum Master how they try to work together as a self-organizing team to accomplish the goal and create the anticipated increment in the remainder of the Sprint.</i> | | |
| Measures | | |
| <i>Meeting duration [suggested time-box 15 minutes]</i> | | |

Proceso de Descripción de Práctica de Ejemplo

Para generar la práctica de ejemplo, se debe seguir la siguiente lista de acciones, y no necesariamente en ese preciso orden, aquí se presenta en forma de lista para tener una visión general, y luego en las siguientes secciones se describe cada acción:

- [Ingresar y conocer las opciones del menú del sistema Kualibs](#)
- [Definir Actividades](#)
- [Definir Conocimientos y Habilidades](#)
- [Definir Criterios de Verificación](#)
- [Definir Métricas](#)
- [Definir Características](#)
- [Definir Productos de Trabajo](#)
- [Definir Actividades \(Complemento\)](#)
- [Definir Práctica](#)
- [Definir Guía](#)
- [Imprimir Guía de la Práctica](#)
- [Definir Método](#)

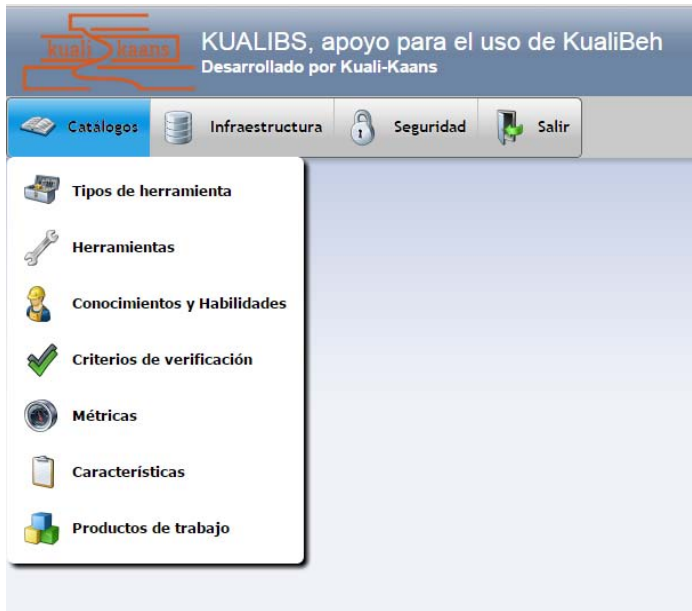
Ingresar y conocer las opciones del menú del sistema Kualibs

Al inicio del sistema será necesario identificarse y seleccionar el idioma en el que se desea trabajar



The screenshot shows the login page for 'KUALIBS para Kuali-Beh'. At the top left is the logo 'kuali kaans' in orange. To the right of the logo is the text 'KUALIBS para Kuali-Beh'. Below the logo, there are three input fields: 'Usuario *' with the value 'kualibeh', 'Contraseña *' with masked characters '*****', and 'Idioma' with a dropdown menu showing 'Español (México)'. A red asterisk and the text '* Campos obligatorios' are located below the input fields. A 'Login' button is positioned at the bottom right of the form area.

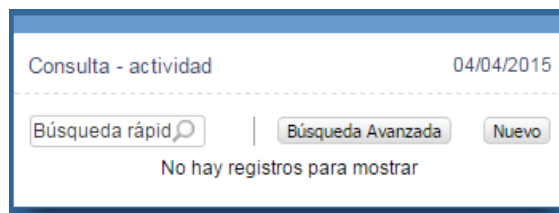
El menú presenta una opción de **Catálogos** para manejar los detalles de los métodos y prácticas: herramientas, habilidades y conocimientos, criterios de verificación y demás. Y también el menú **Infraestructura**, donde se almacenan los métodos, las prácticas, guías, actividades y tareas



Definir Actividades

De acuerdo al ejemplo, se tiene dos actividades en la Práctica, así que se puede empezar por definir las, aunque podría iniciarse definiendo la práctica, el método o cualquier elemento de los catálogos.

Al seleccionar la opción **Actividades**, del menú **Infraestructura**, aparece la pantalla de consulta, al igual que aparece en todos las opciones de Kualibs, ya que, por supuesto, se pueden tener varias prácticas, aunque la primera vez, aparece la pantalla vacía con el botón **Nuevo** para poder agregar un elemento:



Al dar clic en el botón nuevo, se puede agregar una nueva actividad. Aparecen varias pestañas para definir los elementos que están asociados a la actividad que se define, por el momento se puede dejar así, así como también en cualquier momento se puede regresar a esta pantalla a cambiar la definición:

Nuevo Registro - actividad 04/04/2015

Guardar Cancelar

Actividad Tareas Conocimiento y Habilidades Criterios de Verificación Métricas Herramientas

Nombre * Team members's explanation

Descripción

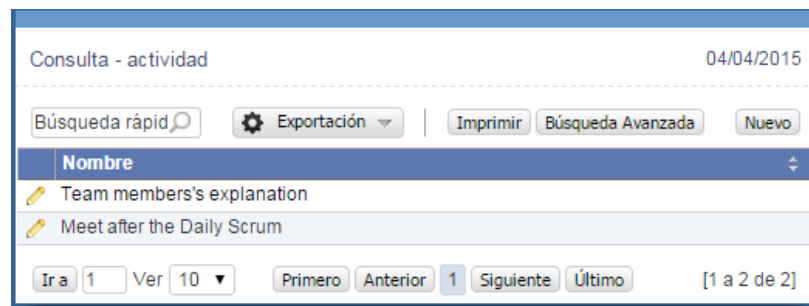
Durind the meeting, each Development Team member explains:

- What has been accomplished since the last meeting?
- What will be done before the next meeting?
- What obstacles are in the way?

* Campos obligatorios

En la pantalla de actividad, para el campo descripción, al igual que para todos los campos de descripción en el sistema, aparece un editor sencillo de HTML, para poder dar formato básico al texto que se escribe.

Una vez definidas las dos actividades, en la pantalla de búsqueda de actividades (la primera que aparece de la opción **Actividades** del menú **Infraestructura**), aparecen definidas como se muestra en la siguiente imagen, aunque en este punto falta definir los elementos de cada una y se deberá regresar a ellas para completar la definición, y para ello se utiliza la misma pantalla, presionando el **ícono de Lápiz (edición)** a lado de cada actividad



Definir Conocimientos y Habilidades

Antes de completar la definición de actividades (sección anterior) se puede definir (no es obligatorio seguir este orden) los conocimientos y habilidades necesarias para esta actividad, en la opción del menú **Catálogos->Conocimientos y Habilidades** como se muestra en la siguiente imagen:

Actualización - Conocimiento y habilidades 04/04/2015

Nuevo Actualizar Borrar Volver

Nombre * Team Knowledge

Descripción

B *I* U ABC [Listas] [Enlaces] [Párrafo] [Color] [HTML]

Development team consists of professionals who do the work of delivering a potentially releasable increment of "Done" product at the end of each sprint.

* Campos obligatorios

Definir Criterios de Verificación

Para definir los criterios de verificación de una práctica, se usa la opción del menú **Catálogos->Criterios de Verificación**, como se muestra en la siguiente imagen para el ejemplo de esta guía:

Nuevo Registro - Criterios de Verificación 04/04/2015

Nombre *

Descripción

B *I* U ABC Párrafo

Development Team should be able to explain to the Product Owner and Scrum Master how they try to work together as a self-organizing team to accomplish the goal and create the anticipated increment in the remainder of the Sprint.

* Campos obligatorios

Definir Métricas

Para definir las métricas de una práctica, se usa la opción del menú **Catálogos->Métricas**, como se muestra en la siguiente imagen para el ejemplo de esta guía:

Nuevo Registro - Métricas 04/04/2015

Guardar Cancelar

Nombre * Meeting duration

Descripción

B *I* U ABC Párrafo

suggested time-box 15 minutes


* Campos obligatorios

Definir Características


Las **Características** son las propiedades específicas de los **Productos de Trabajo**, o de las **Condiciones** que a su vez son las **Entradas** o **Resultados** de las Prácticas, y para el ejemplo de esta guía, existen 3 **Características** que pertenecen a las **Condiciones** tanto de **Entrada** como de **Resultado** de la práctica, y se definen en la opción **Catálogos->Características**.


Consulta - Características 04/04/2015

Característica

 Every Development Team member knows the answer to the following questions:

- What has been accomplished sine the last meeting?
- What wil be done before the next meeting?
- What obstacles are in the way?

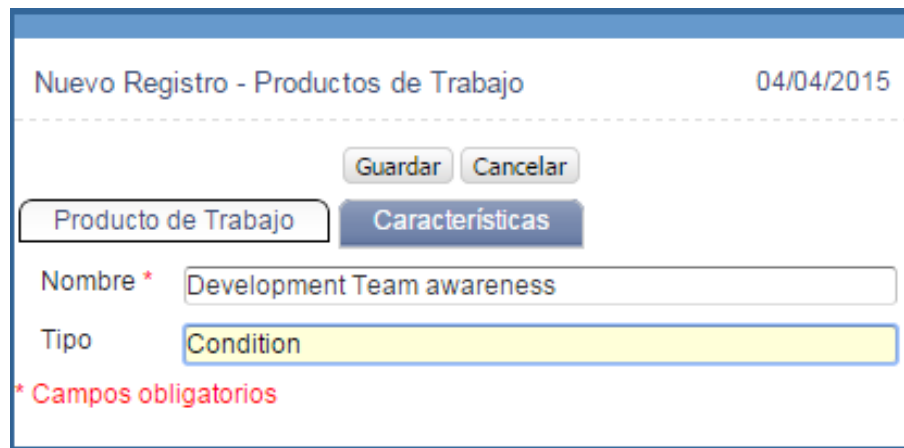
 Held at the same time and place each day.

 Improved the Development Team's level of project knowledge

[1 a 3 de 3]

Definir Productos de Trabajo

Los Productos de Trabajo son lo que necesita una Práctica como insumo, y/o lo que produce como Resultado después de ejecutarse, pueden ser productos de trabajo o condiciones, y pueden o no tener características más específicas. En esta guía, de acuerdo al ejemplo planteado, se tienen 2 condiciones que sirven de Entrada a la Práctica, y se tienen 3 Productos de Trabajo y una Condición como Resultado de la práctica, así que en se definirán estos 6 elementos en la pantalla a través de la opción del menú **Catálogos->Productos de Trabajo**, por ahora no se especifica si son de Entrada o de Salida, ya que eso se define en la Práctica, pues para una Práctica un mismo Producto de Trabajo que es su Resultado, a la vez podría ser Entrada de otra Práctica. También en este ejemplo, los productos de trabajo no tiene más características, pero las 3 condiciones sí las tienen, y es en esta pantalla que se pueden asociar de las que se definieron en la sección de Características, como puede verse en las siguientes pantallas:



Nuevo Registro - Productos de Trabajo 04/04/2015

Guardar Cancelar

Producto de Trabajo Características

Nombre * Development Team awareness

Tipo Condition

* Campos obligatorios

Dentro de esta pantalla, se presentan dos pestañas, la primera para definir el nombre y tipo de producto de trabajo, y en la segunda pestaña, se puede seleccionar la(s) característica(s) que tendrá, de las disponibles (definidas en la sección de Características):

Nuevo Registro - Productos de Trabajo 04/04/2015

Producto de Trabajo

Características

Held at the same time and place each day.

Improved the Development Team's level of project knowledge

Every Development Team member knows the answer to the following questions:

- What has been accomplished sine the last meeting?
- What will be done before the next meeting?
- What obstacles are in the way?

* Campos obligatorios

Después de definir los 6 elementos, en la pantalla inicial de Productos de Trabajo se tiene el siguiente aspecto:

Consulta - Productos de Trabajo 04/04/2015

Búsqueda rápida |

| Nombre | Tipo | Característica |
|---|--------------|--|
| Development Team awareness | Condition | Every Development Team member knows the answer to the following questions: <ul style="list-style-type: none"> • What has been accomplished sine the last meeting? • What will be done before the next meeting? • What obstacles are in the way? |
| Meeting schedule | Condition | Held at the same time and place each day. |
| Sprint Back log | Work product | |
| Product Back log items selected for the Sprit | Work product | |
| Updated plan for delivering items selected | Work product | |
| Improved knowledge | Condition | Improved the Development Team's level of project knowledge |

Ver [1 a 6 de 6]

En esta pantalla, al igual que en todas las pantallas del sistema que presentan listas de elementos, existen botones de navegación, para poder navegar entre ellos cuando son más de los que caben en una pantalla.

Definir Actividades (Complemento)

En esta sección, se muestra que es posible modificar en cualquier momento, cualquier elemento de las Prácticas o Métodos, o cualquiera de sus elementos que los conforman. En este ejemplo, en la sección previa de definición de actividades se definieron 2 actividades, pero no sus Conocimientos y Habilidades, Criterios de Verificación, o Métricas que requieren. Así que una vez definidas, cada una en su pantalla correspondiente, y de acuerdo el ejemplo, se puede ingresar a la pantalla de Actividades, y en cada una de las pestañas se puede asociar, de acuerdo a las necesidades, los elementos disponibles.

Actualización - actividad 04/04/2015

Nuevo Actualizar Borrar Volver

Actividad Tareas Conocimiento y Habilidades Criterios de Verificación Métricas Herramientas

Nombre *

Descripción

B I U ABC [List Icons] [Undo/Redo] Párrafo [Color/Background Color] [HTML]

Durind the meeting, each Development Team member explains:

- What has been accomplished since the last meeting?
- What will be done before the next meeting?
- What obstacles are in the way?

* Campos obligatorios

Actualización - actividad 04/04/2015

Nuevo Actualizar Borrar Volver

Actividad Tareas Conocimiento y Habilidades Criterios de Verificación Métricas Herramientas

Conocimientos y Habilidades Team Knowledge

* Campos obligatorios

Actualización - actividad
04/04/2015

Nuevo
Actualizar
Borrar

Volver

Actividad
Tareas
Conocimiento y Habilidades
Criterios de Verificación
Métricas
Herramientas

Métricas Meeting duration

* Campos obligatorios

Sólo se muestran las pantallas a la actividad: Team members's explanation, pero de manera equivalente se definen las otras del ejemplo, seleccionado los elementos que les corresponda.

Definir Práctica

Para definir una práctica, ya que se han definido los elementos que la van a conformar, sólo se requiere la pantalla de la opción del menú **Infraestructura->Prácticas** y completar la información adecuada, como se muestra en la siguiente imagen para el ejemplo planteado. Es aquí donde se definen si los productos de trabajo o condiciones son una **Entrada** o un **Resultado** esperado de la **Práctica**.

Nuevo Registro - Prácticas 04/04/2015

Guardar Cancelar

Práctica Entrada Resultado

Nombre * Daily Scrum Meeting

Versión 1

Objetivo *

Development Team meeting to synchronize activities and create (adapt) a plan for the next 24 hours. To assess progress toward the Sprint Goal and to assess how progress is trending toward completing the work in the Sprint Backlog.

* Campos obligatorios

Nuevo Registro - Prácticas 04/04/2015

Guardar Cancelar

Práctica Entrada Resultado

Entradas

- Development Team awareness
- Improved knowledge
- Meeting schedule
- Product Back log items selected for the Sprit
- Sprint Back log
- Updated plan for delivering items selected

* Campos obligatorios

Definir Guía

Una Guía describe una y sólo una Práctica, aunque una Práctica, en principio, podría tener diferentes guías, por diferentes equipos de trabajo o diferentes versiones. Para este ejemplo sólo se crea una, y se le asocia la práctica descrita en la sección de Práctica, así como las actividades también descritas previamente.

Actualización - Guía 04/04/2015

Nuevo Actualizar Borrar Imprimir Guía Volver

Guía Actividades

Actividades Meet after the Daily Scrum
 Team members's explanation

* Campos obligatorios

Imprimir Guía de la Práctica

Para imprimir una Guía de una Práctica, se requiere ingresar a la opción del menú **Infraestructura->Guías**, seleccionar la Guía deseada y seleccionar el ícono de edición, para poder ingresar a la pantalla de edición de la Guía, en la que se puede actualizar, borrar, con sus botones respectivos o usar el botón **Imprimir Guía**, y al realizar esta secuencia, el sistema genera un archivo PDF descargable, con el formato de la plantilla propuesta en la documentación de KUALI-BEH, con los datos relacionados a la Práctica, como se muestra en la siguiente imagen.

| | | |
|---|--|---|
| Práctica | Daily Scrum Meeting | |
| Objetivo | Development Team meeting to synchronize activities and create (adapt) a plan for the next 24 hours. To assess progress toward the Sprint Goal and to assess how progress is trending toward completing the work in the Sprint Backlog. | |
| Entradas | | Resultados |
| <p>Condition: Development Team awareness</p> <p>Every Development Team member knows the answer to the following questions:</p> <ul style="list-style-type: none"> • What has been accomplished since the last meeting? • What will be done before the next meeting? • What obstacles are in the way? <p>Condition: Meeting schedule</p> <p>Held at the same time and place each day.</p> | | <p>Condition: Improved knowledge</p> <p>Improved the Development Team's level of project knowledge</p> <p>Work product: Sprint Back log</p> <p>Work product: Product Back log items selected for the Sprint</p> <p>Work product: Updated plan for delivering items selected</p> |
| Guía | | |
| Actividades | | |
| <p>Team members's explanation</p> <p>During the meeting, each Development Team member explains:</p> <ul style="list-style-type: none"> • What has been accomplished since the last meeting? • What will be done before the next meeting? • What obstacles are in the way? <p>Meet after the Daily Scrum</p> <p>The Development Team often meets immediately after the Daily Scrum to re-plan the rest of the Sprint's work</p> | | |
| Tareas | | |
| Herramientas | | |
| Conocimientos y Habilidades | | |
| <p>Team Knowledge</p> <p>Development team consists of professionals who do the work of delivering a potentially releasable increment of "Done" product at the end of each sprint.</p> | | |
| Criterios de Verificación | | |
| <p>Self-organization exposition</p> <p>Development Team should be able to explain to the Product Owner and Scrum Master how they try to work together as a self-organizing team to accomplish the goal and create the anticipated increment in the remainder of the Sprint.</p> | | |
| Métricas | | |
| <p>Meeting duration</p> <p>suggested time-box 15 minutes</p> | | |

Definir Método

Para definir un Método, es necesario ingresar a la opción **Infraestructura->Métodos** y definir su nombre, propósito, y siguiendo la misma mecánica utilizada en todo el sistema, se pueden seleccionar los elementos capturados previamente, para definir las entradas, los resultados, y las prácticas del Método.

Para definir los datos iniciales, se tiene:

The screenshot shows a web form titled "Nuevo Registro - Métodos" with a date of "04/04/2015". At the top right are "Guardar" and "Cancelar" buttons. Below are four tabs: "Método", "Entrada", "Resultado", and "Prácticas". The "Método" tab is active. It contains three input fields: "Nombre *" with the value "Scrum", "Versión" with the value "1", and "Propósito *" with the text "To agility develop software". A rich text editor toolbar is visible above the text area. At the bottom left, there is a red asterisk and the text "* Campos obligatorios".

Para definir los productos de trabajo o condiciones, se utilizan las pestañas: "Entrada" y "Resultado", de acuerdo a su función dentro del Método.

The screenshot shows the same "Nuevo Registro - Métodos" form, but with the "Entrada" tab selected. The "Método" tab is now disabled. Under the "Entradas" section, there is a list of six items, each with an unchecked checkbox: "Development Team awareness", "Improved knowledge", "Meeting schedule", "Product Back log items selected for the Sprit", "Sprint Back log", and "Updated plan for delivering items selected". The red asterisk and text "* Campos obligatorios" are still present at the bottom left.

Y por último, para definir las “Prácticas” que conforman al “Método”, se seleccionan las deseadas en la lista presentada en la pestaña “Prácticas”, En este ejemplo, sólo se tiene una, “Daily Scrum Meeting”, pero en esta pantalla aparecen todas las “Prácticas” definidas con anterioridad.



Nuevo Registro - Métodos 04/04/2015

Guardar Cancelar

Método Entrada Resultado Prácticas

Prácticas Daily Scrum Meeting

* Campos obligatorios