



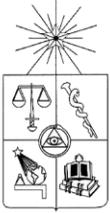
**UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**

**UNA HERRAMIENTA EXPRESIVA PARA IMPLEMENTACIÓN DE TABLEROS
KANBAN VIRTUALES**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

ROBERTO ANDRÉS RIQUELME TORRES

2011



**UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**

**UNA HERRAMIENTA EXPRESIVA PARA IMPLEMENTACIÓN DE TABLEROS
KANBAN VIRTUALES**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

ROBERTO ANDRÉS RIQUELME TORRES

PROFESOR GUÍA:
AGUSTÍN VILLENA MOYA

MIEMBROS DE LA COMISION:
ALEXANDRE BERGEL
LUIS MATEU BRULE

SANTIAGO DE CHILE
ABRIL 2011

RESUMEN

El Tablero Kanban es un sistema de información físico asociado a la metodología *lean*. Es un tablero que refleja los estados de las actividades que se deben realizar en un proceso determinado. Con tarjetas-señal las personas asumen sus responsabilidades. Al poder visualizar el flujo de trabajo, se muestran rápidamente los logros y problemas del proceso. Con esta herramienta los equipos logran enfocarse en terminar las tareas que tienen asignadas y no en acumular tareas iniciadas.

El objetivo general del presente trabajo es evaluar y desarrollar una herramienta tecnológica para la implementación de Tableros Kanban Virtuales, que sea expresiva, con el objeto de ser lo más parecido posible a un Tablero Kanban Físico, donde no se impone una sintaxis y no se define una semántica, sino que la define el grupo que lo ocupa. Con las herramientas existentes, la sintaxis y la semántica están contextualizadas y acopladas, por lo que los que las usen, deben adaptarse a esas estructuras, siendo que los principios de los Tableros Kanban definen que ellos se deben adaptar a la realidad, y que además se puedan ocupar en los más variados contextos.

El trabajo fue realizado evaluando las alternativas existentes de tableros físicos, virtuales y teóricos, analizando sus características, sintaxis, semánticas y agrupando los gestos ocupados para poder extraer la información general relevante que sea capaz de cubrir la mayor cantidad de contextos posibles para que el tablero Kanban Virtual sea capaz de adaptarse a la realidad de cualquier grupo que lo use.

El resultado final fue la implementación en ambiente Web de un grupo de gestos genéricos que se pueden aplicar a un grupo de objetos sintácticos, que fueron priorizados e implementados por importancia de su uso, resultando en un Tablero Kanban Virtual expresivo, adaptable a la mayoría de los contextos revisados, desarrollado de forma modular con estándares y publicado como código abierto.

Se concluye que es factible construir una herramienta que no imponga una semántica, que defina un mínimo de sintaxis, que soporte los gestos necesarios y que sea lo suficientemente expresiva como para reemplazar o complementar de manera correcta a un Tablero Kanban Físico, con las ventajas de un Tablero Kanban Virtual. Se llama *eKanban*.

A mi esposa Erika y a nuestro hijo Rafael, por su amor y su apoyo incondicional, absoluto y necesario en el cumplimiento de mis metas personales y profesionales;

A mis padres, por su total soporte en mi desarrollo personal y profesional;

A mis suegros y cuñados, por su invaluable ayuda y compañía en los momentos más necesarios.

Índice

1. Introducción	7
1.1. Acerca de la Agilidad.....	7
1.1.1. <i>Lean Thinking</i>	7
1.1.2. <i>Lean Software Development</i>	8
1.1.3. <i>Tableros Kanban</i>	12
1.1.4. <i>Tableros Kanban y Principios Lean</i>	15
1.2. Objetivos del Trabajo.....	16
2. Situación Actual.....	18
2.1. Metodología Kanban - Tableros Kanban.....	18
2.2. Revisión Tableros Kanban.....	20
2.2.1. <i>Tableros Kanban Físicos</i>	20
Tablero Físico 1: Área de Arquitectura.....	20
Tablero Físico 2: Ingeniería de Sistemas.....	23
Tablero Físico 3: Desarrollo Web.....	24
Ventajas y Desventajas de los Tableros Kanban Físicos.....	26
2.2.2. <i>Tableros Kanban Virtuales</i>	27
Tablero Virtual 1: RadTrack.....	27
Tablero Virtual 2: LeanKitKanban.....	28
Tablero Virtual 3: AgileZen.....	29
Tablero Virtual 4: Kanbanery.....	30
Tablero Virtual 5: Scrumy.....	31
Ventajas y Desventajas de los Tableros Kanban Virtuales.....	32
2.2.3. <i>Tableros Kanban Teóricos</i>	33
Kanban in Software Development.....	33
10 Kanban Boards and their context.....	34
2.3. Comparación: Ventajas y Desventajas (Problemas Actuales).....	36
3. Propuesta Nueva	38
3.1. Objetivo General.....	38
3.2. Objetivos Específicos.....	38
3.3. Plan de Trabajo.....	39
4. Diseño de la Herramienta	41
4.1. Sintaxis, Semántica y Gestos.....	41
4.3. Modelo de Datos.....	46
4.4. Observaciones.....	46
5. Implementación.....	47
5.1. Requerimientos.....	47
5.2. Plataformas de Desarrollo.....	47
5.3. Desarrollo.....	49
6. Pruebas	55
6.1. Seguimiento de Trabajo de Tesis.....	55
6.2. Kanban Arquitectura.....	55
6.3. Otras Observaciones.....	55
7. Conclusiones.....	56
8. Trabajo Futuro	57
9. Bibliografía	58

10. Anexos.....	60
10.1. Lean Thinking.....	60
10.2. W. Edwards Deming.....	60
10.3. The Toyota Way	61
10.4. Lean Software Development.....	62

Índice de Figuras

Figura 1: Tablero Kanban Básico.....	8
Figura 2: Principios Ágiles.....	9
Figura 3: Principios de Lean Software Development	10
Figura 4: Waterfall v/s Agile	11
Figura 5: Uso de Tableros Kanban	12
Figura 6: El cambio de herramienta	13
Figura 7: Push Scheduling v/s Pull Scheduling	14
Figura 8: Stand-up Meetings	15
Figura 9: Tablero Kanban.....	17
Figura 10: Tablero Kanban Arquitectura	22
Figura 11: Plantilla Kanban Arquitectura.....	22
Figura 12: Tablero Kanban IDS.....	24
Figura 13: Tablero Kanban Desarrollo Web.....	25
Figura 14: Plantilla Kanban Desarrollo.....	26
Figura 15: RadTrack.....	28
Figura 16: LeanKitKanban.....	29
Figura 17: AgileZen	30
Figura 18: Kanbanery.....	31
Figura 19: Scrumy	32
Figura 20: Kanban in Software Development (1)	33
Figura 21: Kanban in Software Development (2)	34
Figura 22: 10 Kanban Boards and Their Context.....	36
Figura 23: Semántica de Tableros Físicos.....	37
Figura 24: Semántica de Tableros Virtuales	37
Figura 25: Listado de Gestos Ordenados por Objeto.....	42
Figura 26: Relación de Total de Gestos por Tableros.....	43
Figura 27: Listado de Gestos Ordenados por Prioridad.....	44
Figura 28: Modelo de Datos	46
Figura 29: Kanban-Board (1).....	48
Figura 30: Kanban-Board (2).....	48
Figura 31: Cumplimiento de Gestos por Prioridad	51
Figura 32: eKanban - Lanes Activos	51
Figura 33: eKanban - Tablero Kanban (Lanes Seleccionados)	52
Figura 34: eKanban - Configuración	52
Figura 35: eKanban - Edición de una Tarjeta.....	53
Figura 36: eKanban - iPhone	53
Figura 37: eKanban - iPad	54

1. Introducción

1.1. Acerca de la Agilidad

1.1.1. *Lean Thinking*

En la actualidad existen muchas y variadas formas de manejar la **Gestión de Proyectos**. Todo depende del tipo de trabajo que se esté realizando: repetir una receta conocida (como en la *manufactura*), o crear un nuevo producto.

Dentro de la gestión de proyectos, cualquiera sea la que se elija, una de las realidades que menos se considera es: **¿Cómo lograr que un equipo haga un buen trabajo y a la vez “lo pase bien”?**

Existe una propuesta llamada *Lean Thinking* (“Pensamiento Esbelto”) [Lean Thinking] (Womack & Jones, 2003), inspirada en el “Conocimiento Profundo” y en los “14 Puntos” de W. Edwards Deming (1900-1993) [W. Edwards Deming], adoptada por Taichi Ohno en Toyota [The Toyota Way] (Liker, 2004) en los dos tipos de trabajo que se pueden realizar:

- Manufactura: *Toyota Production System*
- Desarrollo de Nuevos Productos: *Toyota Product Development System*

Estos puntos se oponen a la rigidez de los procesos, y a que los procesos son más importantes que las personas. Al automatizar un proceso, se olvida que hay gente que los maneja, los critica y los mejora. Si en algún momento se requiere mejorar los procesos, lo harán personas que no entienden cual es el real problema de negocio. Si existe alguien que puede hacer un trabajo, es la persona más capaz de realizarlo, considerando todos los derechos y responsabilidades que ello significa.

Para la organización de las responsabilidades de los obreros de la compañía Toyota, no se ocupan complejos sistemas como los ERP, sino que se realiza en simples tableros con “**Tarjetas-Señal**” (Kanban en japonés) como el que se muestra en la Figura 1: Tablero Kanban Básico), que gatilla las acciones que se deben realizar. Estos Tableros Kanban logran una efectiva auto-organización del personal, asignando las responsabilidades respectivas.



Figura 1: Tablero Kanban Básico

¿Qué tiene que ver esto con Ingeniería de Software?: *Kanban for Software Engineering*

1.1.2. Lean Software Development

Mary y Tom Poppendieck, en su libro “*Lean Software Development*” de 2003 (Poppendieck & Poppendieck, 2003) definen las bases teóricas de Metodologías Ágiles de Desarrollo de Software a partir del Lean Thinking [Lean Software Development], inspirada en el “*Toyota Product Development System*”.

La hipótesis del *Lean Software Development* dice que: **existe un conjunto simple de principios que, al ser seguidos y compartidos por un equipo de desarrollo de un nuevo producto, genera un comportamiento auto-organizado.** Esta es la base para equipos de alto rendimiento, como los obreros de Toyota.

Actualmente se utilizan en desarrollo de software varias **Prácticas Ágiles**, como por ejemplo: *Test Driven Development (TDD)*, *Planning Game*, *Pair Programming*, etc., que se generan a partir de **Principios** sobre los cuales están basados [Figura 2]:

Se Prefiere	Por sobre	Principio
Individuos e interacciones		Procesos y herramientas
Software funcional		Documentación exhaustiva
Colaboración con el cliente		Negociación de contratos
Responder al cambio		Seguir un plan

Figura 2: Principios Ágiles

La práctica ágil *Lean Software Development* se basa en 7 Principios, que forman un ciclo [Figura 3]:

- Eliminar Desperdicios
 - o Funcionalidades extra
 - o Sobrecarga de requerimientos o arreglos luego del testing
 - o Cruce de fronteras organizacionales
- Crear Conocimiento:
 - o Planificar es útil. Aprender es esencial
 - o Estándares existen para ser desafiados y mejorados
 - o Rendimiento predecible es guiado por Retroalimentación
- Calidad Intrínseca:
 - o Código a Prueba de Errores con Desarrollo Guiado por Tests (TDD)
 - o Impedir el desarrollo de código legado (sin tests)
 - o “Big Bang” está obsoleto
- Diferir el Compromiso (hasta el último momento responsable)
 - o Abolir la idea de partir con una especificación completa
 - o Romper dependencias (Arquitectura debe permitir evolucionar)
 - o Mantener opciones abiertas
 - o Planificar decisiones irreversibles hasta el último momento responsable
- Entregar Rápido:
 - o Listas y colas son amortiguadores entre organizaciones que enlentecen
 - o Entregas rápidas, con alta calidad y bajo costo son compatibles
 - o Teoría de Colas aplica al desarrollo, no sólo a los servidores
 - o Limitar el trabajo a la capacidad del equipo
- Respetar a las personas:
 - o Personas motivadas y pensantes proveen la ventaja competitiva más sostenible

- Equipos crecen con Orgullo, Compromiso, Confianza y Reconocimiento
- Proveer liderazgo efectivo
- Respetar a los socios
- Optimizar el todo (Mejorar el sistema):
 - Productos brillantes emergen de combinar oportunidad con tecnología
 - Enfocarse en el flujo de valor completo
 - Entregar un Producto Completo
 - Medir hacia el valor de negocio generado y satisfacción de los clientes

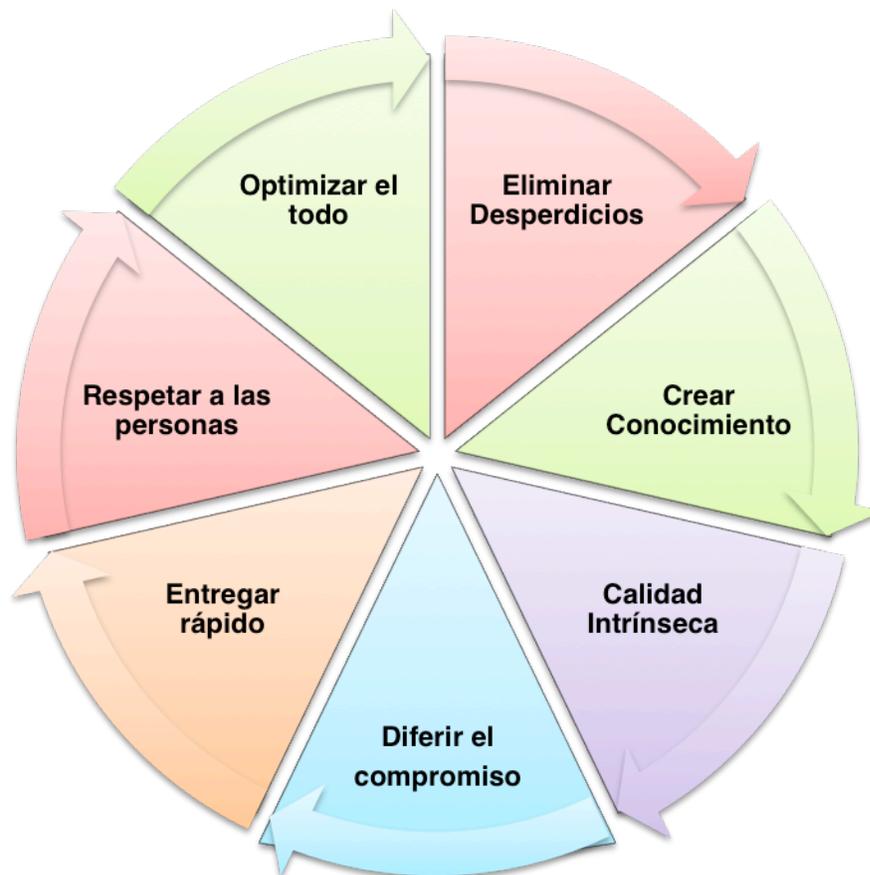


Figura 3: Principios de Lean Software Development

Con estos principios se produce un gran cambio en el paradigma clásico: ¡hacer entregas de forma más rápida y con más calidad!
 Esto es posible **acortando los tiempos de ciclo de entrega, maximizando la utilización de recursos.**

La siguiente pregunta que debemos responder es: **¿Cómo se puede calcular la carga óptima de trabajo de un equipo?**

Se debe analizar el problema de negocio para priorizar. Al tener entregas rápidas, los ciclos son cortos. Cada entrega incremental va generando valor apreciable para el cliente. En vez de hacer una sola gran entrega, se entrega cada “*feature*” independiente y desacopladamente. De ésta forma, se hace más fácil la adaptación a los cambios.

El número mínimo de *features* que el cliente negocia para que el sistema sea aceptado se conoce como “**Minimun Marketable Feature**” (MMF).

Esto se opone a la división por actividades del modelo “*waterfall*”. Una actividad puede tener uno o más (muchos más) *features* [Figura 4].

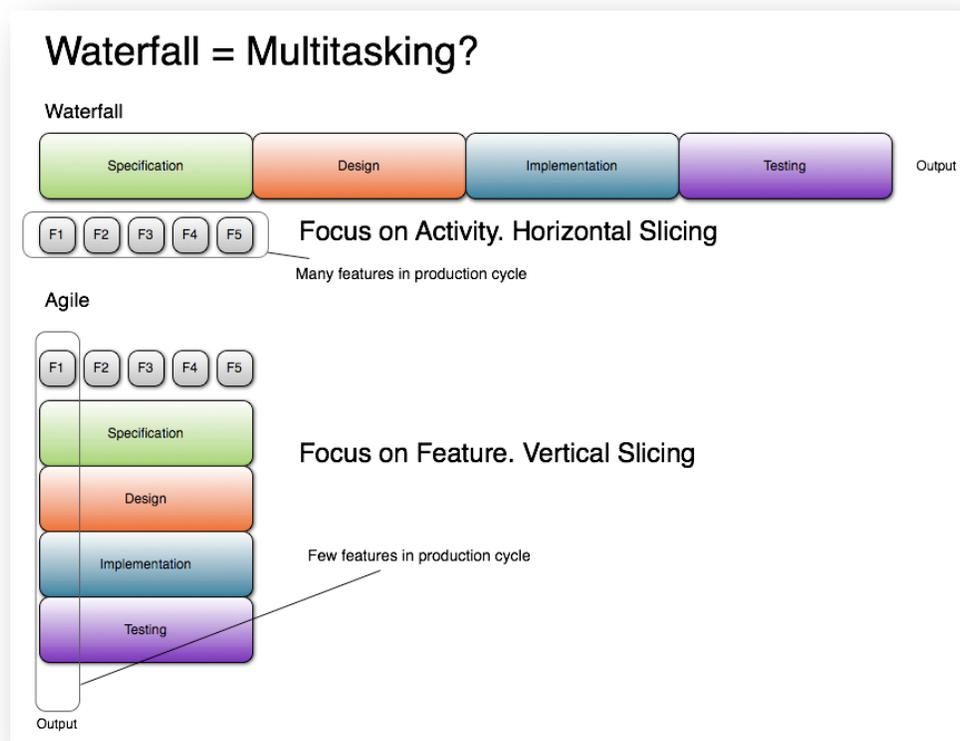


Figura 4: Waterfall v/s Agile

Y estos principios, ¿Cómo pueden ayudar a aumentar la productividad? Limitando el “Trabajo en Curso”, o “**Work In Progress**” (WIP). Al ser tareas cortas, se puede estar concentrado en una (o a lo más dos) tareas simultáneamente, y así dedicar tiempo a agregar valor a la solución. Estudios indican que la productividad decrece cuando se asignan más de dos tareas simultáneamente a alguna persona. **El cambio de contexto es carísimo**. Esto es lo que se conoce como “**Agile 2.0**”.

Y, ¿qué pasa con la calidad? Debe construirse junto con el sistema, no se puede agregar artificialmente al final. Estos criterios de calidad deben ser definidos por el mismo cliente. La finalidad es **eliminar los problemas lo antes posible**.

Toda esta descripción de Prácticas Ágiles y Principios parecen ser ideales. ¿Cómo se puede llevar a la práctica sin perder el control de las tareas asignadas y las responsabilidades?

1.1.3. Tableros Kanban

Acá es donde aparece la herramienta Lean aplicada al desarrollo de software, los **Tableros Kanban** [5.1 Metodología Kanban - Tableros Kanban y Figura 5].

Consisten en:

- Un tablero que refleja el estado de cada actividad de trabajo (con una Tarjeta Visual), por ejemplo:
 - o En Cola – En Ejecución – Finalizado.
 - o Por Hacer – Haciendo – Listo
 - o *To Do – Doing – Done*
- Se sincroniza el problema de negocio y la capacidad del equipo de desarrollo usando Kanban
- Cada persona escoge el trabajo a realizar: Responsabilidad asumida, no asignada (basado en eXtreme Programming)



Figura 5: Uso de Tableros Kanban

Esta herramienta es ligera y valiosa. Entrega mucha información:

- Visibilidad del flujo de trabajo
- Responde fácilmente: ¿Dónde estamos?
- Enfoca al equipo
- Presenta rápidamente los cuellos de botella
- Es sencilla y barata (se cambian complejos sistemas de planificación, como cartas Gantt, por lápices y papeles adhesivos) [Figura 6]



Figura 6: El cambio de herramienta

Con esta herramienta, el enfoque es en terminar las tareas, no en seguir acumulando tareas iniciadas. Se visualiza fácilmente el momento justo donde se tiene la capacidad para realizar una tarea (*Pull Scheduling*). No se sobrecarga al equipo como *Push Scheduling* [Figura 7], y se puede limitar conscientemente el Work In Progress (WIP).



Figura 8: Stand-up Meetings

El equipo, con Kanban, constantemente va revisando el flujo de trabajo, apoyando la mejora continua. Al ser una herramienta flexible y personalizable, cada equipo puede ir adaptando el tablero a sus necesidades, agregando o quitando etapas, ocupando la simbología que considere necesaria para indicar avances, problemas, excepciones, etc.

Kanban es una herramienta que va construyendo su **significado de manera emergente**, no hay una forma “correcta” de implementarla. Va adquiriendo significado con el equipo que lo implementa.

1.1.4. Tableros Kanban y Principios Lean

Y ¿Cómo Kanban adopta los Principios Lean?

- Da al equipo una visión clara de su avance: Respeta a las personas
- Revisión constante y mejoramiento del flujo de trabajo
 - o Entregas rápidas
 - o Optimiza el todo
 - o Crea conocimiento
 - o Calidad intrínseca en el producto final
- Limita el trabajo en curso (WIP): Entregas rápidas

Dentro de las metodologías ágiles, Lean y Kanban es un modelo más genérico que Scrum o eXtreme Programming, es decir, puede ser ocupado en contextos más diversos.

En Chile, existe una comunidad de Desarrolladores Ágiles de Software llamada ChileAgil (<http://www.chileagil.cl>) que promueve el uso de Principios y Prácticas ágiles.

1.2. Objetivos del Trabajo

La principal ventaja de los tableros Kanban físicos es que tienen una **sintaxis** (estructura) y una **semántica** (significado que se le da a cada parte) que depende de cada equipo que lo implementa. Si cambia la sintaxis, cambia el significado del tablero.

Los tableros físicos, tienen muchas ventajas como: facilidad de uso, flexibilidad, visibilidad, pero también tienen varias desventajas como: portabilidad, reportes, persistencia de la información, etc., que intentan ser resueltas con versiones virtuales (principalmente Web) de tableros Kanban.

Actualmente, en el mercado existen herramientas que implementan Tableros Kanban Virtuales, que resuelven algunas desventajas de los tableros físicos, pero que también limitan o ignoran las ventajas que debe tener el tablero, es decir, no cumplen con los objetivos básicos de la utilización de esta metodología.

Principalmente, las herramientas disponibles **imponen** una semántica y **definen** una sintaxis, evitando que se puedan realizar algunas “movidas” o personalizaciones que si se pueden hacer en un tablero físico.

Básicamente, las actividades realizadas (proyectos, tareas, etc.) deben adaptarse a estos tableros, y no al revés, como realmente está pensada la herramienta de Tableros Kanban. Uno de los principios Lean es que las herramientas deben apoyar la toma de decisiones de las personas, y no tomar decisiones por ellas.

El **objetivo** de éste trabajo de Tesis es desarrollar una herramienta colaborativa que permita tener un Tablero Virtual Kanban que soporte la mayor cantidad de cambios de sintaxis, y que defina algunas herramientas básicas de semántica (con la menor cantidad de supuestos posible, como: etiquetas, observaciones, fechas, etc.), de manera que sea lo más parecida a un tablero físico en su expresividad y personalización, de modo que sea el equipo el que le dé el significado a sus partes, y que no sean impuestas por el sistema utilizado.

Además, se busca tener una bitácora de la evolución de los cambios en la sintaxis y la semántica en el tiempo, para poder reconstruir la historia del equipo de trabajo, para obtener estadísticas y métricas de logros.

En resumen, desarrollar una herramienta colaborativa para la implementación de Tableros Kanban Virtuales con una **sintaxis expresiva**, donde la **semántica se la den los usuarios**, y de donde se puedan obtener estadísticas detalladas. Una herramienta flexible para la implementación de tableros Kanban virtuales, lo más parecido posible a los tableros físicos [Figura 9]



Figura 9: Tablero Kanban

2. Situación Actual

2.1. Metodología Kanban - Tableros Kanban

El trabajo con Tableros Kanban se basa en un marco teórico, convenciones que han sido acordadas por los grupos que han propuesto este enfoque de cambio de administración y evoluciones propias que se han observado de las organizaciones que lo han implementado.

Kanban no es un método de Administración de Proyectos, ni tampoco un Software de manejo de Ciclo de Vida de Desarrollo. Es un enfoque para cambiar la administración, un framework para catalizar el cambio en una organización.

Kanban usa la limitación del “Trabajo en Curso” o “*Work In Progress*” (WIP) como mecanismo de control para incentivar las discusiones de cambio. Si no se respetan los límites del WIP y no se conversan los problemas, el trabajo se estancará y no se podrá mejorar. Las discusiones sobre mejoras son objetivas, gracias a que la visualización, mediciones y claridad de las políticas y los modelos de Lean, *Theory of Constraints* (Teoría de Restricciones), y las enseñanzas de W. Edwards Deming, le permite al equipo analizar científicamente sus problemas y proponer soluciones.

Kanban tiene un enfoque más evolutivo que revolucionario de lo que estamos acostumbrados para asumir el cambio. Con Kanban, se comienza con un proceso que ya existe. Se “Kanbaniza” mapeando el flujo, visualizándolo, y limitando el WIP para crear un sistema Pull. Se mantienen intactos los roles existentes, responsabilidades, cargos, y prácticas. Los únicos cambios se realizan en la interfaz con quienes nos relacionamos en el trabajo, por ejemplo jefaturas y operaciones. Cualquier cambio que se haga, será específicamente elegido para evitar agitar la jerarquía social, o invocar una respuesta emocional defensiva de la gente afectada.

Kanban provoca un cambio evolutivo. Inicialmente, esto significa Optimización de Procesos (Kaizen), pero gradualmente, mientras la organización madura en sus capacidades, se transforma en grandes cambios administrados (Kaikaku). Se ha observado que eventos Kaizen progresivos llevan a una madurez organizacional mejorada, y a provocar niveles de cambio de Kaikaku más dramáticos.

Kanban está diseñado como un enfoque que permitirá personalizar y evolucionar un proceso existente, sin importar donde se encuentre el proceso. Por lo tanto, es imposible definir un test de conformidad para Kanban. Kanban usa 5 propiedades principales para catalizar el comportamiento emergente de la evolución. Estas 5 propiedades son:

- Visualizar el *Workflow*
- Limitar el *Work-In-Progress*
- Medir el Flujo

- Explicitar las Políticas de los Procesos
- Usar Modelos para Evaluar Oportunidades de Mejoras

Estas propiedades representan las 5 prácticas que deben estar presentes para que el enfoque Kanban funcione. El proceso del equipo para el desarrollo de software y la administración de proyectos, siempre será única, y con el tiempo será adaptada a medida, optimizada para darle valor al flujo de trabajo, medir el riesgo, capacidades y habilidades del equipo, demanda del cliente, cuellos de botella y variaciones de distintos índices que pueden afectar al equipo. No puede haber un test de conformidad. Cualquier medida que se pueda aplicar, debe servir para comprobar si alguna mejora afectó algún resultado económico o sociológico con la introducción de la propuesta del Kanban.

Kanban impulsa mecanismos para simplificar la coordinación de los elementos de todo un sistema. Por ejemplo, la combinación de la visualización y de la limitación del WIP (sistema Pull) permiten una interfaz simplificada con las jefaturas. Como resultado, la mayoría de las organizaciones que adoptan Kanban no necesitan el concepto de “*Product Owner*” (Dueño del Producto), y pueden fácilmente enfrentar múltiples canales de entrada, encolando requerimientos.

Las “*Stand-up Meetings*” (Reuniones de Pie) diarias se ha demostrado que son efectivas incluso con 50 o más personas. La razón es porque el equipo confía implícitamente en que el trabajo que se muestra en la visualización se está realizando. No es necesario usar estas reuniones para reforzar el compromiso personal, por lo que las reuniones se enfocan en el trabajo, y no en la gente. Los equipos iterarán sobre los tickets de trabajo, en vez de los miembros del equipo. Las tres preguntas se obvian. Equipos Kanban más maduros reducen la discusión sólo al trabajo que está bloqueado o defectuoso, enfocándose sólo en las excepciones en vez del trabajo que está avanzando con normalidad.

Se ha observado que las organizaciones que usan Kanban juntan grupos pequeños para aprovechar la ventaja de la reducción de costos de coordinación, y por el mejor resultado de su trabajo en conjunto. Se ha hecho común ver equipos de 20, 30, e incluso 50 personas creados de equipos más pequeños de otras iniciativas. La visualización del *workflow* generalmente implica múltiples filas (o *Swimlanes*) para representar los distintos flujos de desarrollo. Algunos miembros de los equipos pueden ser asignados a un *swimlane* como miembros permanentes de un equipo, mientras que otros pueden saltar entre distintos *swimlanes* para apoyar a otros equipos con habilidades especiales. El resultado es un uso más efectivo y eficiente de los recursos disponibles, dando como resultado entregas más rápidas, y menor tiempo de respuesta.

2.2. Revisión Tableros Kanban

El uso de Tableros Kanban es un tema que se ha investigado bastante y a medida que avanza, van apareciendo más inquietudes sobre los posibles usos que se les puede dar y las áreas en las cuales se puede implementar.

Existen casos de uso de tableros físicos, virtuales e incluso casos de estudio teóricos (de posibles implementaciones y usos de estos tableros).

Para la confección de éste informe, se realizó una profunda investigación del uso e implementación de variados tipos de tableros (físicos, virtuales y artículos con teoría sobre estos tableros)

A continuación, se hará una revisión en detalle de los tableros revisados: los físicos ocupados para la investigación, las alternativas virtuales más populares, y de los documentos describiendo teóricamente el uso de tableros Kanban, mostrando además la forma de uso, describiendo su **sintaxis** (forma), su **semántica** (significado) y los **gestos** soportados, para luego realizar una comparación de sus ventajas y desventajas.

2.2.1. Tableros Kanban Físicos

Los tableros Kanban son herramientas esencialmente físicas. Una pizarra con Tarjetas-Señal (Kanban) pegadas, con un significado compartido por los miembros de un proyecto. Cada parte del tablero (sintaxis) tiene su propio significado (semántica), las cuales se manejan con ciertos gestos que son soportados por el tablero.

Como parte de la investigación, se revisaron 3 tableros físicos que actualmente están siendo utilizados para la gestión de proyectos y desarrollo de software en el área de Informática de la empresa El Mercurio.

Tablero Físico 1: Área de Arquitectura

Este tablero Kanban, que se muestra en la Figura 10: Tablero Kanban Arquitectura), es utilizado para la gestión de proyectos y tareas del área de Arquitectura

Se observan las siguiente características:

- Sintaxis:
 - 6 columnas
 - 11 filas (número variable)
 - Hojas de distintos colores
 - Texto en las hojas pegadas
 - Fila destacada
- Semántica:
 - La primera columna representa el Título de la fila
 - Cada columna representa un estado: *Backlog, ToDo, Doing, Check, Done*

- Cada fila representa una tarea o proyecto en específico. Depende del tamaño de la tarea, es si se “merece” una fila propia. También hay filas genéricas para tareas repetitivas más pequeñas.
- Se ven distintos colores de papeles:
 - Azul: representa una fila
 - Amarillo claro: Responsable 1
 - Amarillo oscuro: Responsable 2
 - Amarillo oscuro en parte superior: título de las columnas
- Se ve una fila destacada en la parte inferior, donde se mueven las tareas urgentes, independiente del proyecto.
- Las hojas tienen plantillas acordada por el grupo, como se ve en la Figura 11: Plantilla Kanban Arquitectura. Hay una para las filas, y una para las actividades:
 - Azul (Fila):
 - Título
 - Dueño (Responsable)
 - Prioridad
 - Jefe de Proyecto
 - Fecha de Solicitud
 - Fecha de Compromiso
 - Amarillo (actividades):
 - Título
 - Contacto
 - Prioridad
 - Responsable
 - Peso
 - Fechas de: Solicitud, compromiso, inicio, término.
- Cabe notar que la ubicación donde se escriben los textos también es acordada por el grupo, al igual que cuales son los campos obligatorios y cuales son opcionales.
- Gestos:
 - Agregar, eliminar y modificar kanbans, filas y columnas
 - Se pueden mover kanbans entre columnas y filas
 - Se puede agregar, eliminar y modificar la información en cualquiera de los campos existentes
 - Se pueden marcar kanbans (agregar notas) para información relevante (poniendo un papel pequeño sobre el kanban original)



Figura 10: Tablero Kanban Arquitectura



Figura 11: Plantilla Kanban Arquitectura

Tablero Físico 2: Ingeniería de Sistemas

Este tablero Kanban, que se muestra en la Figura 12: Tablero Kanban IDS) es utilizado para la gestión de proyectos y tareas del área de Ingeniería de Sistemas.

Se observan las siguientes características:

- Sintaxis:
 - 6 columnas
 - 9 filas (número variable)
 - Hojas de distintos colores
 - Texto en las hojas pegadas
 - Filas destacada
- Semántica:
 - La primera columna contiene el Título de la fila
 - Cada columna representa un estado: *Backlog, ToDo, Doing, Check, Done*
 - Las 7 primeras filas, representan proyectos en desarrollo
 - Las 2 últimas filas, representan tareas repetitivas del área (RFCs e Incidentes)
 - Se ven distintos colores de papeles:
 - Rosado: Fila de Proyecto tipo 1
 - Rojo: Fila de Proyecto tipo 2
 - Amarillo: Tareas con Prioridad Normal
 - Azul: Tareas con alta prioridad
 - Rosado en la parte superior: título de las columnas
 - Colores de texto:
 - Rojo: Responsable 1
 - Verde: Responsable 2
 - Azul: Responsable 3
 - En este caso, la organización de la información en los papeles es la siguiente:
 - Rojo/Rosado (Fila):
 - Título
 - Amarillo/Azul (actividades):
 - Título
 - Iniciales del Responsable (a pesar de ser representado por el color del texto)
 - Número de RFC o Incidente
 - La ubicación donde se escriben los textos también es acordada por el grupo, al igual que cuáles son los campos obligatorios y cuáles son opcionales.
- Gestos:
 - Agregar, eliminar y modificar kanbans, filas y columnas
 - Se pueden mover kanbans entre columnas y filas
 - Se puede agregar, eliminar y modificar la información en cualquiera de los campos existentes

- Para cambiar la prioridad de un kanban, se debe eliminar el actual, y crear uno nuevo.



Figura 12: Tablero Kanban IDS

Tablero Físico 3: Desarrollo Web

Este tablero Kanban, que se muestra en la Figura 13: Tablero Kanban Desarrollo Web), es utilizado para la gestión de proyectos y tareas del área de Desarrollo Web.

Se observan las siguiente características:

- Sintaxis:
 - 6 columnas
 - 8 filas (número variable)
 - Hojas de distintos colores (aunque deberían ser todas blancas)
 - Texto en las hojas pegadas
 - Filas destacadas
- Semántica:
 - La primera columna representa el título de la fila
 - Cada columna representa un estado: *Backlog*, *Ready*, *Work In Progress*, *Check*, *Done*
 - Las 5 primeras filas representan proyectos del área
 - Las 3 últimas filas, representan tareas repetitivas del área (RFCs por área)
 - Los papeles son todos blancos
 - Los títulos de las filas y columnas son etiquetas con un tipo de letra específico.

- Las hojas de las actividades tienen plantillas acordadas por el grupo, como se ve en la Figura 14: Plantilla Kanban Desarrollo).
 - Proyecto/Incidente
 - Descripción de la tarea (Título)
 - Ingeniero (Responsable)
 - Orden o código / Pre-requisito
 - Puntos (Dificultad)
- La ubicación donde se escriben los textos también es acordada por el grupo, al igual que cuales son los campos obligatorios y cuales son opcionales.
- Gestos:
 - Agregar, eliminar y modificar kanbans, filas y columnas
 - Se pueden mover kanbans entre columnas, no entre filas. Cada actividad parte con una fila asignada
 - Se puede agregar, eliminar y modificar la información en cualquiera de los campos existentes



Figura 13: Tablero Kanban Desarrollo Web

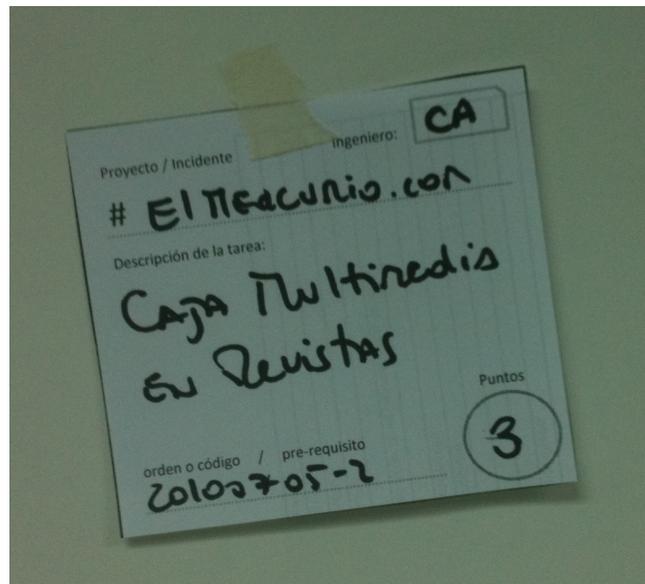


Figura 14: Plantilla Kanban Desarrollo

Ventajas y Desventajas de los Tableros Kanban Físicos

Como se mencionaba anteriormente, los tableros Kanban son esencialmente físicos por lo que cumplen con todos los requerimientos básicos como:

- Facilidad de uso
- Flexibilidad
- Visibilidad
- Concurrencia de múltiples usuarios (incluso un equipo entero)

Pero también, al ser básicamente un tablero con papeles, y dados los requerimientos más modernos, tiene las siguientes desventajas:

- Portabilidad
- Dificultad para generar reportes (en general)
- Incapacidad de generar reportes de forma digital
- Persistencia de la información
- Historia
- Cálculo de métricas relevantes
 - o Tiempo promedio de una tarea en comenzar
 - o Tiempo de ciclo, etc.

Con esto en mente, se realizó la revisión de Tableros Kanban Virtuales, para ver como se comportan versus los tableros físicos.

2.2.2. Tableros Kanban Virtuales

Los Tableros Kanban Virtuales nacen de la necesidad de modernizar un simple sistema de administración para, por ejemplo, coordinar equipos que no trabajan en conjunto, poder hacer mediciones sobre la información contenida, poder acceder a la información desde cualquier lugar, obtener la historia de un proyecto, etc.

A continuación, se muestran 5 sistemas de tableros virtuales que fueron analizados en esta investigación, para luego detallar sus ventajas y desventajas.

Un tema muy importante a tener en cuenta es que, debido a cómo están contruidos estos sistemas, se hará una descripción de sus características más que una descripción de su sintaxis y semántica (como en el punto 2.2.1 de Tableros Físicos), ya que a este nivel, ninguno de los sistemas hace diferencia entre ambas capas, estando amarrados sólo al tipo de proyectos y actividades que ellos soportan.

Tablero Virtual 1: RadTrack

La Figura 15: RadTrack muestra una imagen del tablero kanban virtual RadTrack: <http://radtrack.com>

Las características principales de este sistema son:

- Gratuita
- Es muy poco intuitiva, difícil de usar
- Está en versión beta, incompleta y con varios errores
- Soporta sólo 5 tipos de tarjetas:
 - o *MMF*
 - o *User Story*
 - o *Defect*
 - o *Technical Debt*
 - o *Other*
- No se puede cambiar el tipo de tarjeta
- Soporta *Backlogs*
- No se puede limitar el WIP
- No se pueden bloquear actividades
- Se pueden modificar etapas (columnas) en el tablero
- No soporta *swimlanes* (filas). Trabaja con una fila única
- Las tarjetas se pueden mover desde y hacia cualquier etapa
- Permite el manejo de prioridades
- Se puede asignar actividades a usuarios
- No se pueden agregar observaciones
- No tiene archivo histórico
- No se pueden agregar adjunto
- No es posible agregar etiquetas a las tarjetas
- No existen notificaciones de las tareas

- No tiene implementadas estadísticas

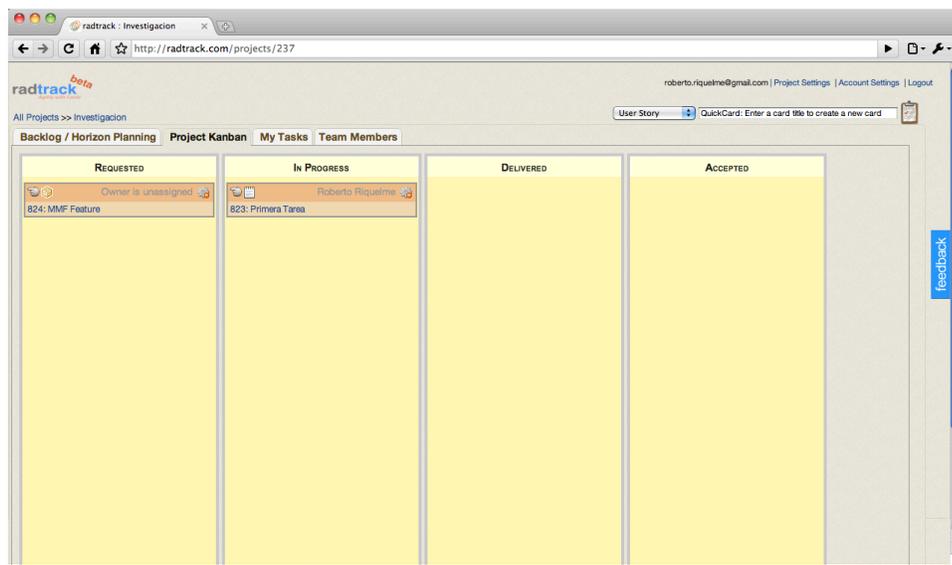


Figura 15: RadTrack

Tablero Virtual 2: LeanKitKanban

La Figura 16: LeanKitKanban muestra una imagen del tablero kanban virtual LeanKitKanban: <http://leankitkanban.com>

Las características principales de este sistema son:

- Tiene versiones gratuita (limitada) y de pago (USD \$20 mensuales por usuario)
- Muy intuitiva, muy parecida a un tablero real
- Actualmente es una versión final estable
- Soporta "Note Cards" genéricas
- Se puede cambiar el tipo de tarjeta
- Soporta Backlogs
- Se puede limitar el WIP
- No se pueden bloquear actividades
- Se pueden crear, modificar y eliminar columnas
- No soporta swimlanes (filas). Trabaja con una fila única
- Las tarjetas se pueden mover desde y hacia cualquier etapa
- Permite el manejo de prioridades a "tarefas".
- Se puede asignar actividades a usuarios
- Se pueden agregar comentarios y descripciones
- Tiene una bitácora de actividades terminadas
- No se pueden agregar adjunto
- Se pueden agregar etiquetas a las tarjetas
- Tiene notificaciones por RSS y por mail
- Tiene 4 tipos de estadísticas:

- *Cumulative Flow*
- *Cycle Time*
- *Card Distribution*
- *Efficiency*
- Actualmente es usado para la administración de actividades de ChileAyuda (con el apoyo de ChileAgil)

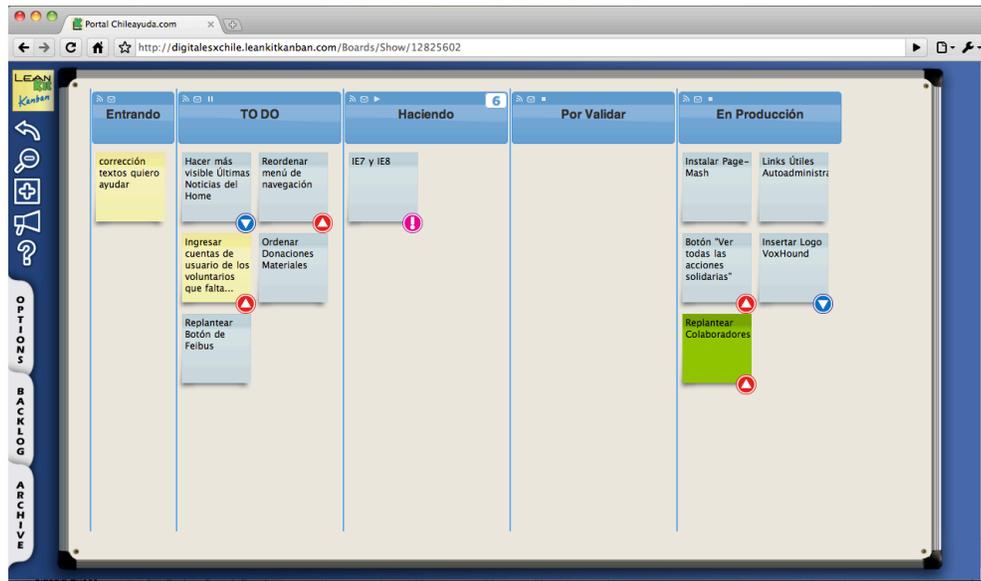


Figura 16: LeanKitKanban

Tablero Virtual 3: AgileZen

La Figura 17: AgileZen muestra una imagen del tablero kanban virtual AgileZen: <http://agilezen.com>

Las características principales de este sistema son:

- Tiene versiones gratuita (limitada) y de pago (desde USD \$9 mensuales por usuario)
- Dificultad de uso media, poco intuitiva
- Actualmente es una versión final estable
- Soporta "Historias" genéricas
- Se pueden editar las historias
- Soporta *Backlogs*
- Se puede limitar el WIP
- Se pueden bloquear actividades con observaciones
- Se pueden crear, modificar y eliminar columnas
- No soporta *swimlanes* (filas). Trabaja con una fila única
- Las tarjetas se pueden mover desde y hacia cualquier etapa
- Permite el manejo de etiquetas para asignar prioridades a "historias".
- Se puede asignar actividades a usuarios

- Se pueden agregar comentarios y descripciones
- Tiene una zona de archivo histórico
- Se pueden agregar adjuntos a las historias
- Se pueden agregar etiquetas a las tarjetas
- Tiene notificaciones por mail, Google Talk, MSN, AIM, ICQ, etc.
- Tiene gráficos y métricas de progreso

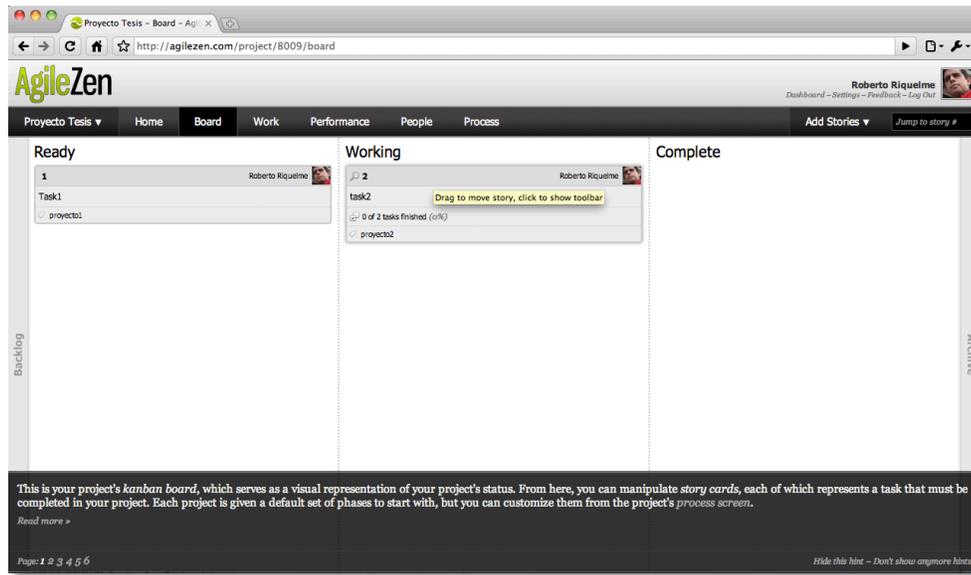


Figura 17: AgileZen

Tablero Virtual 4: Kanbanery

La Figura 18: Kanbanery muestra una imagen del tablero kanban virtual Kanbanery: <http://kanbanery.com>

Las características principales de este sistema son:

- Tiene versiones gratuita (limitada) y de pago (desde USD \$9 mensuales por usuario)
- Fácil de usar, muy intuitiva
- Actualmente es una versión final estable
- Soporta 3 tipos de "Tasks" genéricas
 - o Bug
 - o Story
 - o Chore
- Se pueden editar las historias
- Soporta Backlogs
- Se puede limitar el WIP
- Se pueden bloquear actividades con observaciones
- Se pueden crear, modificar y eliminar columnas
- No soporta swimlanes (filas). Trabaja con una fila única

- Las tarjetas se pueden mover desde y hacia cualquier etapa
- Permite el manejo de prioridades en *Tasks*.
- Se puede asignar actividades a usuarios
- Se pueden agregar comentarios y descripciones
- Tiene una bitácora de cambios
- Se pueden agregar adjuntos a las historias
- Se pueden agregar etiquetas a las tarjetas
- Tiene notificaciones por RSS y mail.
- Tiene gráficos y métricas de progreso

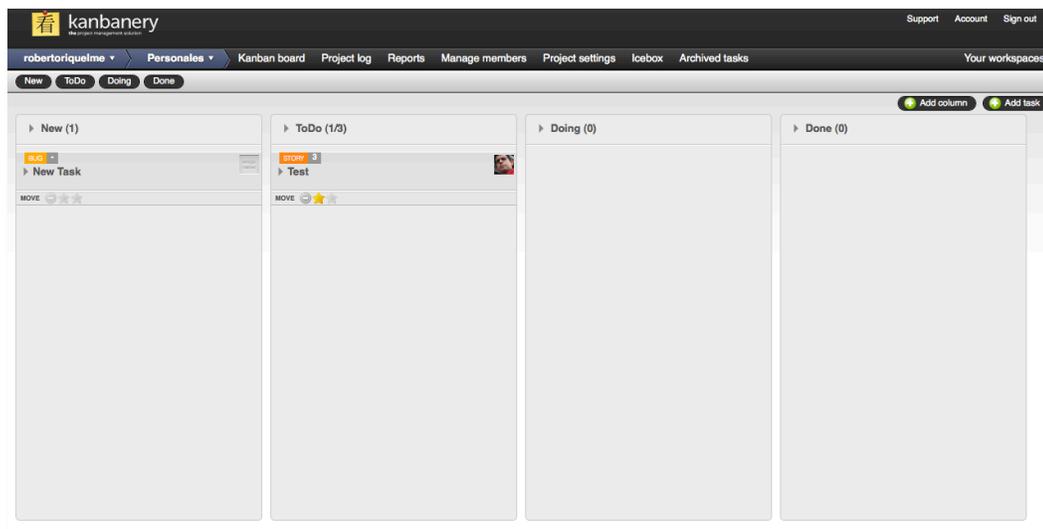


Figura 18: Kanbanery

Tablero Virtual 5: Scrumy

La Figura 19: Scrumy muestra una imagen del tablero kanban virtual Scrumy: <http://scrumy.com>

Las características principales de este sistema son:

- Tiene versiones gratuita (limitada) y de pago (desde USD \$7 mensuales por usuario)
- Muy fácil de usar, muy parecido a un tablero físico
- Actualmente es una versión final estable
- Soporta Kanbans genéricos
- Se pueden editar las historias
- Soporta *Backlogs* (en versión pagada)
- No se puede limitar el WIP
- No se pueden bloquear actividades con observaciones
- No se pueden crear, modificar y eliminar columnas
- Soporta *swimlanes* (filas).
- Las tarjetas se pueden mover desde y hacia cualquier etapa y a cualquier fila

- No permite el manejo de prioridades
- No se pueden asignar actividades a usuarios
- No se pueden agregar comentarios y descripciones
- No tiene una zona de archivo histórico
- No se pueden agregar adjuntos a las historias
- No se pueden agregar etiquetas a las tarjetas
- No soporta notificaciones
- No tiene estadísticas

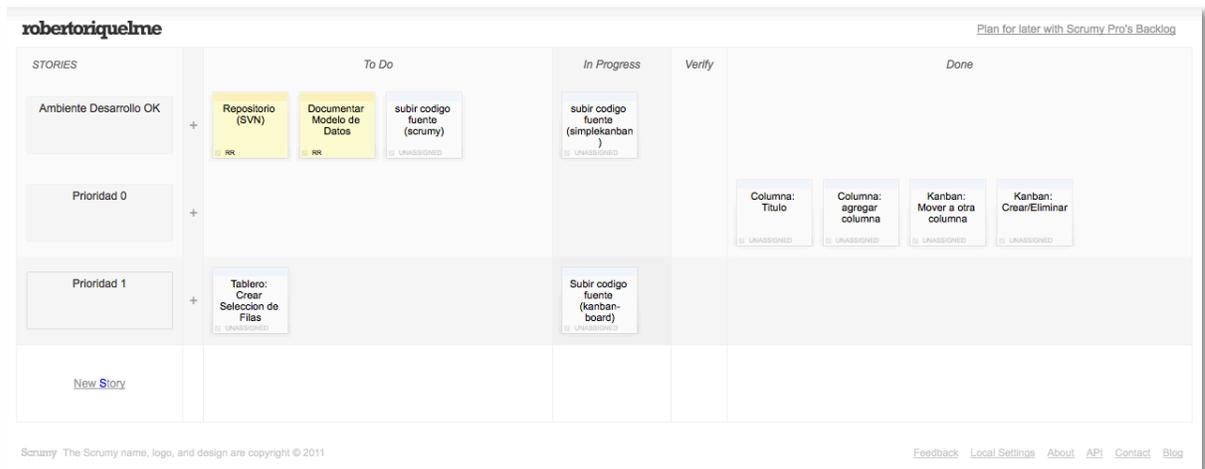


Figura 19: Scrumy

Ventajas y Desventajas de los Tableros Kanban Virtuales

Como se mencionaba anteriormente, los Tableros Kanban Virtuales vienen a solucionar muchas de las desventajas de los tableros físicos, pero dependiendo de la solución que se elija, tienen sus propias limitaciones en comparación con los mismos tableros físicos.

La mayor ventaja de los tableros físicos, es que su sintaxis (estructura) es independiente de su semántica (significado). Los tableros son lo suficientemente expresivos para que se acomoden a los proyectos, y no los proyectos acomodarse a los tableros. Si cambia la sintaxis, cambia el significado del tablero

Las herramientas disponibles **imponen una semántica** y **definen una sintaxis**, evitando que se puedan realizar algunas “jugadas” o personalizaciones que si se pueden hacer en un tablero físico. Ninguno de los sistemas utilizados soporta todas las posibles movidas más utilizadas en los tableros físicos, como por ejemplo:

- *Swimlanes* (filas)
- Atributos en las tarjetas
- Ubicación personalizada de partes y componentes del tablero, etc.

2.2.3. Tableros Kanban Teóricos

Aparte de las aplicaciones físicas y virtuales de los tableros que se han revisado, existen estudios teóricos sobre los contextos en los cuales se podrían ocupar los tableros kanban, donde se analizan las situaciones, sintaxis y semánticas de las posibles implementaciones.

A continuación se mostrarán dos investigaciones donde, evolutivamente se va armando un tablero, dependiendo de los cambios de requerimientos que los equipos van teniendo.

Kanban in Software Development

El sitio “LosTechies.com” en los artículos (Kanban in Software Development. Part 1: Introducing Kanban Boards and Pipelines), (Kanban in Software Development. Part 2: Completing the Kanban Board with Queues, Order Points and Limits), (Kanban in Software Development. Part 2.5: A Variation on Queues - Pipelines for WIP and Done) y (Kanban in Software Development. Part 3: Andon and Jidoka - Handling Bugs and Emergency Fixes in Kanban), analizan la evolución de un tablero de desarrollo de software, desde lo más simple como se muestra en la Figura 20: Kanban in Software Development (1), mostrando el flujo inicial de entrega de *features*, hasta la evolución final, que se muestra en la Figura 21: Kanban in Software Development (2).

En esta evolución se hicieron las siguientes modificaciones:

- Se agregaron más etapas
- Se agregaron múltiples pipelines (swimlanes o filas) para trabajos de equipos en paralelo
- Se agregaron límites de WIP para evitar cuellos de botella
- Algunas etapas son comunes para todos los equipos (Backlog, Analysis, Cust. Accept y Delivery)
- Se definieron colas de entrada en etapas clave (Analysis y Backlog)
- Se agregó una fila de emergencias, para actividades de mayor prioridad.

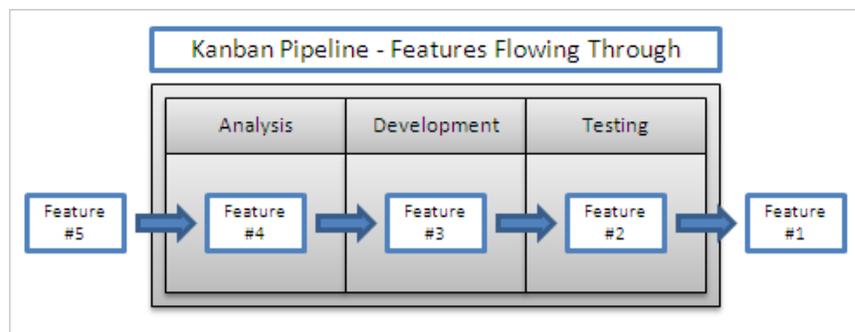


Figura 20: Kanban in Software Development (1)

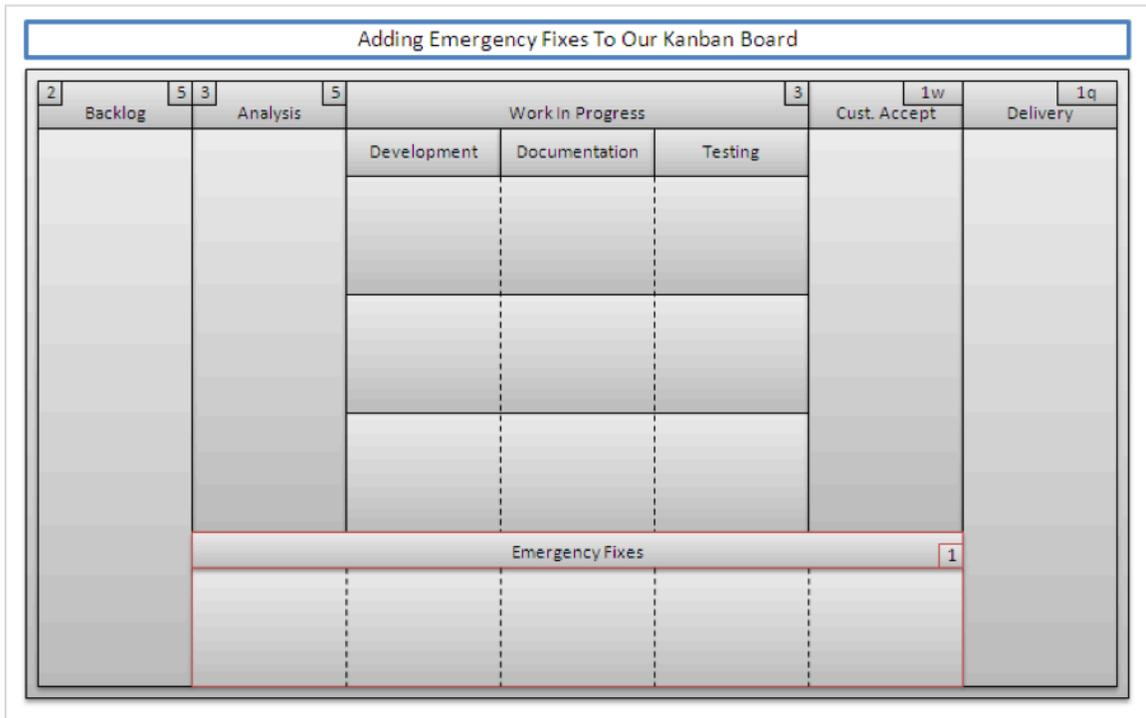


Figura 21: Kanban in Software Development (2)

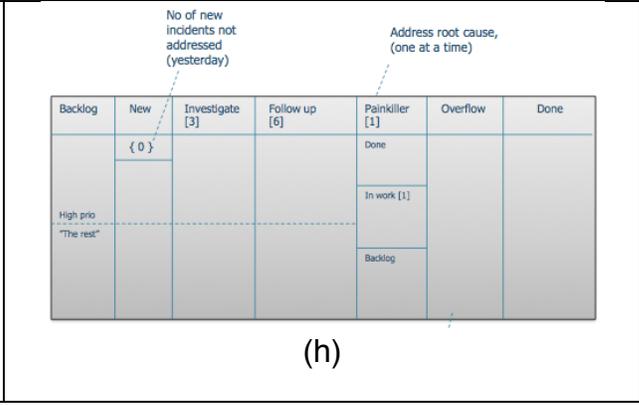
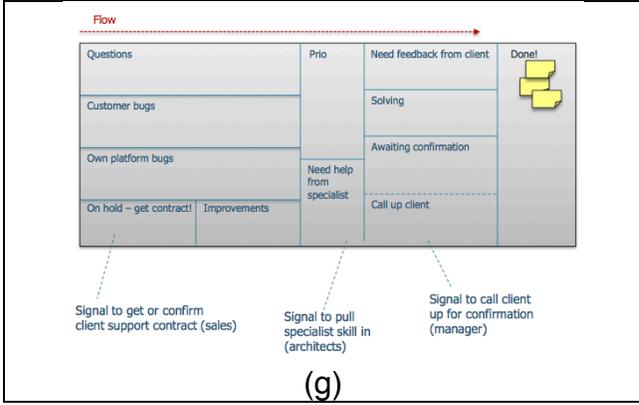
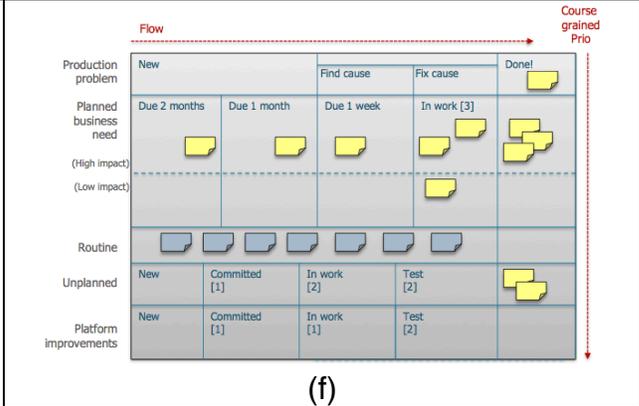
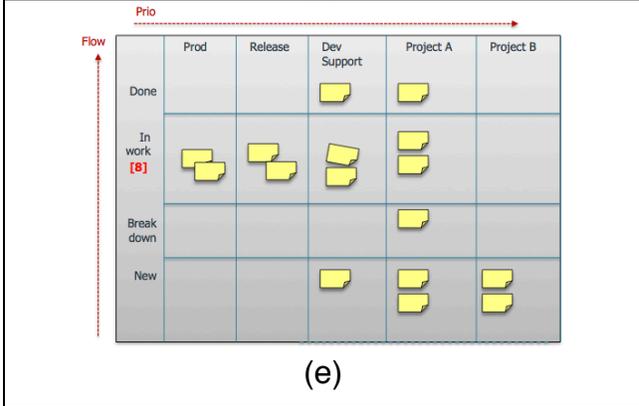
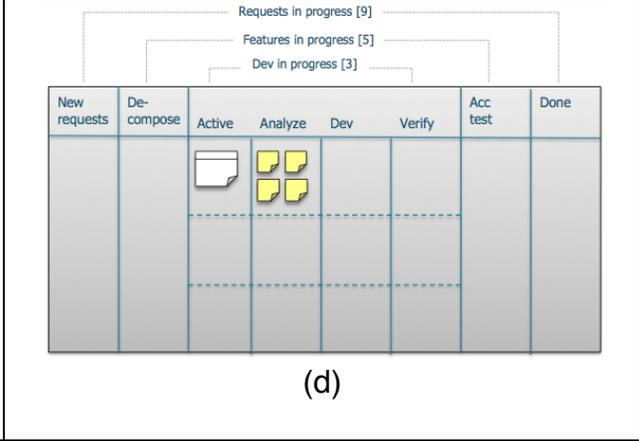
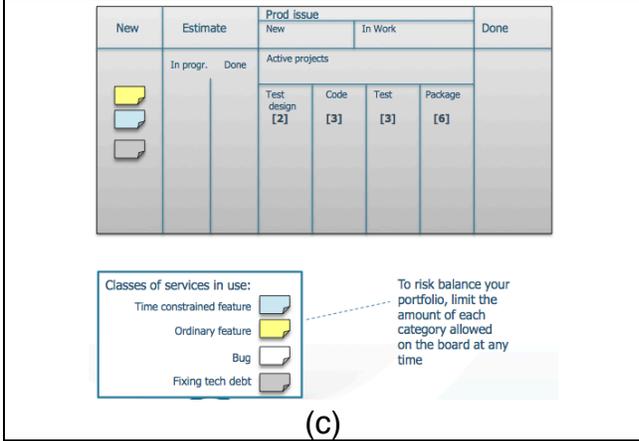
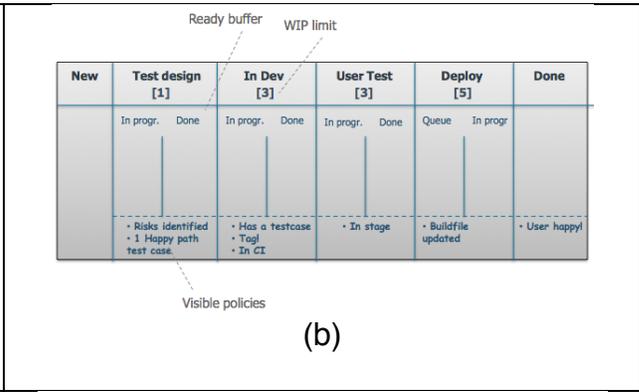
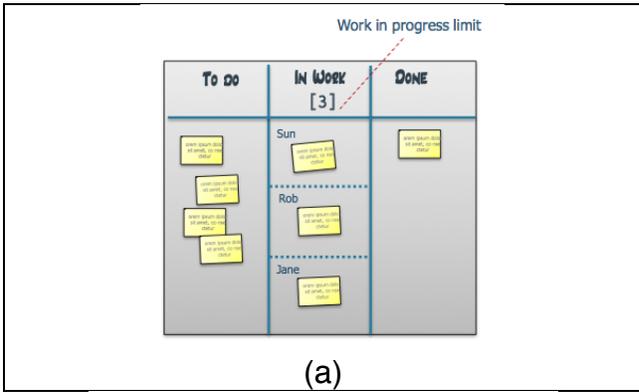
Lo interesante de la evolución que muestran estos artículos es poder ver el nivel de “complejidad” que puede llegar a tomar un “simple” tablero Kanban.

Además, vale notar que esta evolución sólo es posible en un tablero físico, ya que, de las alternativas revisadas (existentes) de tableros virtuales, ninguno es capaz de soportar esta evolución.

10 Kanban Boards and their context

En el artículo (10 Kanban Boards And Their Context), se analizan 10 situaciones en las cuales se pueden ocupar tableros Kanban, y sus contextos, mostrados en la Figura 22: 10 Kanban Boards and Their Context. Estas situaciones son:

- a) Equipo Scrum, aplicando límites de WIP
- b) Equipo de Desarrollo, usando procesos definidos
- c) Equipo de Desarrollo con múltiples clientes
- d) Tablero Kanban de múltiples capas con filas (*swimlanes*)
- e) Administración de Sistemas
- f) Operaciones, Mantenimiento de procesos de negocios
- g) Soporte de primer nivel
- h) Soporte de segundo nivel
- i) Equipo de ventas
- j) Equipo de marketing



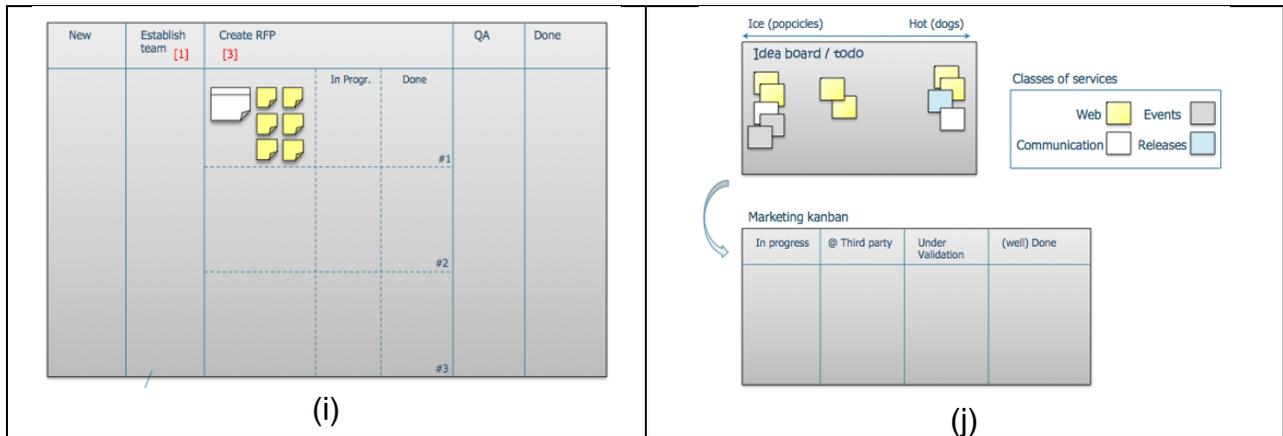


Figura 22: 10 Kanban Boards and Their Context

Nuevamente, lo interesante de éste artículo son los variados contextos en los cuales se pueden aplicar los tableros kanban, y debido a que las soluciones virtuales tienen una semántica exclusiva para desarrollo de software y manejo de proyectos, gran parte de estos contextos no se pueden implementar con los sistemas existentes.

2.3. Comparación: Ventajas y Desventajas (Problemas Actuales)

Luego de haber revisado una buena cantidad de tableros Kanban: 3 físicos, 5 virtuales y más de 10 teóricos, podemos hacer una comparación para ver en que situación nos encontramos actualmente:

- Los tableros físicos son muy expresivos y flexibles, muy fáciles de usar, con mucha visibilidad, soportan la concurrencia de múltiples usuarios, pero se pueden complejizar tanto que ningún sistema virtual los soporta.
- Los tableros virtuales ayudan a solucionar los problemas de los tableros físicos como: portabilidad, generación de reportes, persistencia de la información, historia y métricas, pero fallan en soportar la simpleza original de los tableros kanban.
- Uno de los principios de los tableros kanban es que éstos se deben adaptar a los proyectos, y no los proyectos a los tableros. Esto no sucede con las implementaciones actuales, debido a que los tableros virtuales tienen acopladas la sintaxis y la semántica, que deberían ser acordadas por el grupo que los ocupa.

En la Figura 23: Semántica de Tableros Físicos y en la Figura 24: Semántica de Tableros Virtuales se muestra un detalle de la semántica usada en los tableros físicos y de la semántica impuesta por los tableros virtuales.

Objeto Sintáctico	Semántica Tableros Físicos (Acordada)		
	Arquitectura	IDS	Desarrollo
Tablero	Tablero	Tablero	Tablero
Fila	Proyecto Actividad Tarea Repetitiva Urgencia	Proyectos RFC Incidentes	Proyecto Incidentes Requerimientos
Columna	Estado	Estado	Estado
Kanban	Actividad o Tarea	Actividad RFC Incidente	Actividad Incidente Requerimiento
Atributos	Color Fondo: Responsable Color Texto: Información Titulo Responsable Contacto Fecha Inicio/Fin Dificultad Estimada Dificultad Medida	Titulo Numero de Incidente Numero de RFC Responsable	Nombre Proyecto Titulo Código Prerequisito Puntos de Dificultad Responsable

Figura 23: Semántica de Tableros Físicos

Objeto Sintáctico	Semántica Tableros Virtuales (Impuesta)				
	RadTrack	LeanKitKanban	AgileZen	Kanbanery	Scrumy
Tablero	Proyecto	Proyecto	Proyecto	Workspace	Tablero
Fila	Proyecto	Lane/Fila	Proyecto	Proyecto	Historia
Columna	4 estados (fijos)	Etapas	3 etapas fijas	Etapas	4 etapas fijas
Kanban	Feature Historia Defecto Técnico Otro	Tarjeta (tipo genérico)	Tarjetas de colores	Tareas Bug Historia	Tarjeta única
Atributos	Titulo Notas Bloqueo	Titulo Tipo Tamaño Fecha Fin Descripción Comentarios	Titulo Tamaño Prioridad Fecha Fin Etiquetas	Titulo Prioridad Estimación Descripción	Titulo Responsable

Figura 24: Semántica de Tableros Virtuales

Vemos que, ninguna de las semánticas de los tableros Kanban virtuales existentes calza con la semántica acordada en los tableros físicos ocupados.

Lo que buscamos entonces es generar una herramienta lo suficientemente flexible para que sea posible implementar la mayoría de los tableros antes descritos.

Este proyecto pretende llenar el espacio que hay entre la expresividad de los tableros físicos y la funcionalidad de los tableros virtuales.

3. Propuesta Nueva

3.1. Objetivo General

Los tableros Kanban son herramientas físicas muy expresivas, con una sintaxis flexible y una semántica que se la da el equipo de trabajo que lo usa. Al intentar modernizar y virtualizar estos tableros, las soluciones existentes imponen una sintaxis y definen una semántica a la cual los equipos y actividades deben adaptarse.

Los tableros Kanban deben permitir a los equipos darse cuenta de cómo es su flujo de trabajo, y deben ir adaptándose constantemente, apoyando la toma de decisiones y no tomar decisiones por ellos, y fundamentalmente, enfocar al equipo en su trabajo

Como se vio en el punto 2.2 (Revisión de Tableros Kanban), hay algunas acciones y personalizaciones que no se pueden realizar sobre los tableros virtuales, pero que si son soportadas por los tableros físicos.

El **objetivo** del presente trabajo es diseñar e implementar una herramienta virtual más expresiva que las existentes para soportar la mayor cantidad posible de tableros Kanban, independiente del contexto en el cual se desee ocupar. Una herramienta multiusuario que permita tener un Tablero Kanban Virtual que soporte la mayor cantidad de cambios de sintaxis y que defina algunas herramientas básicas de semántica (con la menor cantidad de supuestos posible), de manera que sea lo más parecida a un tablero físico en su expresividad, personalización y significado.

3.2. Objetivos Específicos

Para crear una herramienta virtual para implementar tableros Kanban virtuales,

Específicamente, la herramienta expresiva a implementar debe soportar lo siguiente:

- Debe soportar gran cantidad de cambios de sintaxis, definiendo de forma general las: tarjetas (kanbans), columnas, atributos, filas (lanes) etc.
- Debe definir una semántica básica general de tableros Kanban como: etiquetas, observaciones, fechas, etc.
- Debe ser colaborativa, accesible y actualizable desde múltiples lugares
 - o Debe ser accesible y manipulable desde dispositivos móviles
- Los integrantes del equipo y el cliente sean capaces de agregar y modificar Kanbans (tarjetas)
- Debe ser moldeable a la realidad del proyecto que la esté utilizando, pudiendo crear líneas de trabajo, crear etapas, tarjetas, atributos genéricos, etc., que serán interpretados por cada equipo
- Debe permitir documentar de alguna forma la historia del proyecto, tener una bitácora de sus cambios. Con esta información, se pueden obtener importantes reportes de las actividades.

En resumen, la herramienta propuesta no debe tener ninguna limitación como las herramientas que existen actualmente.

Para lograr estos objetivos, se acordó el siguiente plan de trabajo:

3.3. Plan de Trabajo

A continuación se detallará el Plan de Trabajo seguido para intentar lograr los objetivos propuestos anteriormente.

- 1) Investigación sobre agilidad y estudiar el uso de tableros Kanban.
- 2) Lo primero que hay que hacer para entender la agilidad y el uso de tableros Kanban es usar uno, por lo que en la primera etapa se implementó un tablero Kanban personal, para comprender la sintaxis y definir la semántica. Luego, se invitó a más personas a participar de éste tablero Kanban, para administrar sus actividades de esta forma
- 3) Además se implementaron dos tableros Kanban adicionales en otras áreas, los que se ven en el punto 2.2.1.
- 4) En paralelo, se investigó alternativas virtuales para ver si era posible traspasar estos tableros físicos a alguna de las alternativas existentes en el mercado, las que se ven en el punto 2.2.2. Al usar algunas alternativas virtuales, se comenzaron a descubrir falencias en su diseño y en su uso.
- 5) Con esta información en la mesa, lo que se resolvió a continuación fue: ¿Cómo se puede generar un Tablero Kanban virtual que sea lo suficientemente expresivo y genérico para que soporte (casi) todos los contextos en los cuales se puede usar uno? ¿Y además no tenga las limitaciones de los que existen actualmente?
 - Al analizar los tableros físicos, virtuales y teóricos del punto 2.2, se puede ver que tienen elementos únicos, pero también, muchos elementos en común.
 - Se hizo un listado con todas las “jugadas” o “movidas” soportadas por cada uno de los tableros. Se listaron las **sintaxis** y las **semánticas** de los tableros.
 - Se agruparon todas las “movidas” por tablero, se definieron las que son comunes a todos los tableros y las que son únicas. La idea es tener un conjunto de “movidas” que representen la mayor cantidad de tableros posible.
 - Al tener este listado de “movidas”, se priorizaron por importancia de uso en cada tablero.
 - Este listado de “movidas” por importancia es el que se debe implementar de forma genérica.
- 6) Se evaluaron una serie de plataformas en las cuales se puede desarrollar este trabajo, considerando los requerimientos originales:
 - HTML5 + Javascript + CSS3
 - Adobe Flash
 - Microsoft .NET (Silverlight)

- Java
- 7) En este punto, habían dos alternativas: comenzar con un proyecto desde cero, o buscar extender alguna alternativa existente, de manera que cumpla con los requisitos.
 - 8) Por ser un estándar abierto y por tener la posibilidad de cumplir con todos los requerimientos, el proyecto fue desarrollado en HTML5 (PHP) + Javascript (jQuery) + CSS3.
 - Existen algunas alternativas (muy) básicas de tableros Kanban en código abierto (Google Code, Sourceforge, etc.)
 - El proyecto fue desarrollado como una extensión de una alternativa OpenSource existente en Google Code: Kanban-Board (Ahmad, 2010)
 - El modelo de datos es muy simple, por lo que cumple con parte de la base de generalidad necesaria para el proyecto.
 - 9) El proyecto fue renombrado a eKanban (*Expressive Kanban*) y extendido según las necesidades del proyecto.
 - 10) Para realmente probar que esta extensión cumple con los requerimientos originales, eKanban se utilizó como un complemento/reemplazo del tablero físico de Arquitectura (punto 2.2.1) y además como administración de éste trabajo de Tesis.
 - 11) Finalmente, se completó la actual documentación

4. Diseño de la Herramienta

A continuación, se describen las consideraciones que se tuvieron al momento de diseñar la herramienta Kanban virtual para que sea expresiva y cumpla con los objetivos.

4.1. Sintaxis, Semántica y Gestos

Uno de los primeros pasos para lograr nuestros objetivos es poder diferenciar Sintaxis (forma) de Semántica (significado) al representar los tableros Kanban.

Los tableros virtuales existentes tienen acopladas sintaxis y semántica. Hablan de Estado, Proyecto, Tarea para referirse a las partes del tablero. Nosotros los trataremos de forma genérica:

- Tablero, Board
- Tarjeta, Card, Kanban
- Lanes, Filas
- Columnas
- Atributos (Genéricos)

Sobre estos objetos **sintácticos**, se aplican ciertos **gestos** y se les aplica una **semántica**.

Esta división entre sintaxis y semántica es la que hace que un sistema sea **expresivo**, ya que no está asociado a ningún significado predefinido.

Por ejemplo, mover una tarjeta de una fila a otra puede tener varios significados, a pesar de que el **gesto** es el mismo:

- Moverla a un nuevo proyecto
- Modificar su prioridad
- Modificar su responsable, etc.

Teniendo la sintaxis genérica, y dejando la semántica “libre” (con pocos supuestos), nos queda definir los gestos que estos tableros soportan.

De todos los tableros que fueron revisados, se llegó a un conjunto de gestos total, que representa a todos los tableros, independiente de su semántica.

Estos son los que se aprecian en la Figura 25: Listado de Gestos Ordenados por Objeto.

N°	Objetos	Gestos
1	Columna	Dibujar Columna
2	Columna	Agregar/Quitar/Editar Texto en encabezado
3	Columna	Agregar número en encabezado
4	Columna	Agregar/Quitar/Editar Atributos estándares en layout predefinido
5	Columna	Subdividir columna
6	Columna	Consolidar columnas
7	Columna	Consolidar filas en una columna
8	Fila	Asociar encabezado de fila con atributo que heredarán Kanbans contenidos
9	Fila	Agregar/Quitar/Editar Atributos estándares en layout predefinido
10	Fila	Seleccionar Color de texto
11	Fila	Seleccionar color de identificador de fila
12	Kanban	Crear/Eliminar Tarjeta
13	Kanban	Mover a otra columna
14	Kanban	Agregar/Quitar/Editar Atributos estándares en layout predefinido
15	Kanban	Asignar Fecha de Creación
16	Kanban	Seleccionar Color de kanban
17	Kanban	Seleccionar Color de texto
18	Kanban	Mover a otra fila y heredar atributo por defecto de fila
19	Kanban	Agregar/Quitar/Editar Atributos personalizados en posiciones personalizadas
20	Kanban	Definir layout
21	Kanban	Poner papel encima (!)
22	Kanban	Eliminar Kanban, y reemplazarlo por 2 o mas nuevos kanban
23	Kanban(s)	Modificación grupal de propiedades
24	Kanban(s)	Agrupar según atributo
25	Tablero	Crear Tablero
26	Tablero	Crear/Eliminar columna
27	Tablero	Crear/Eliminar fila

Figura 25: Listado de Gestos Ordenados por Objeto

Si uno le agrega semántica a estos gestos, se observa un comportamiento como el que se muestra en la Figura 26: Relación de Total de Gestos por Tableros, donde hay un conjunto de gestos que significan exactamente lo mismo, o que son aplicables en todos los tableros.

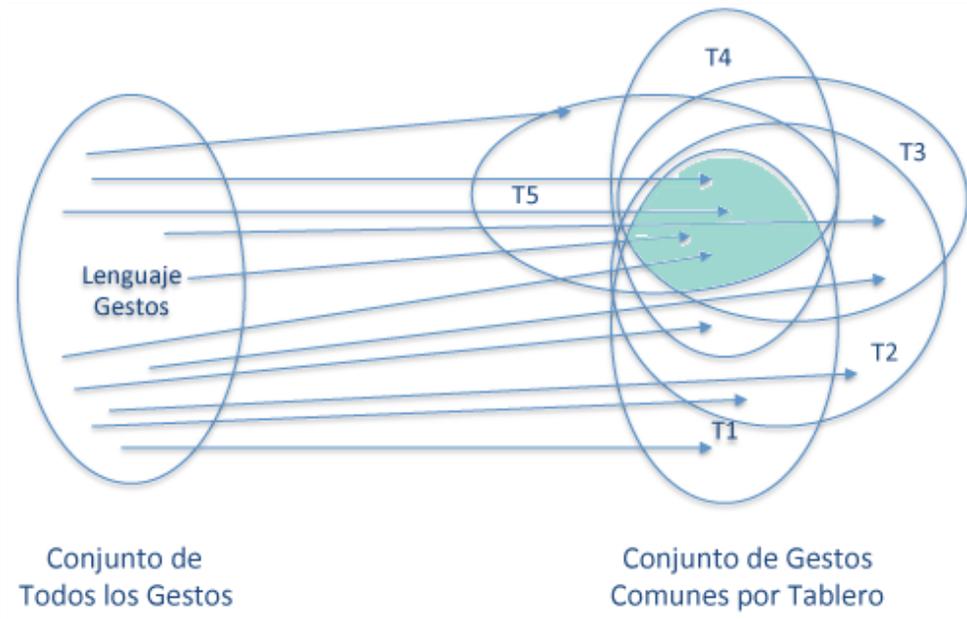


Figura 26: Relación de Total de Gestos por Tableros

Por lo tanto, lo que se busca es implementar este conjunto de objetos y gestos, asignando un mínimo de semántica para cubrir la mayor cantidad de tableros posible.

Este listado de gestos fue priorizado por importancia y relevancia en el conjunto total de tableros Kanban, como se muestra en la Figura 27: Listado de Gestos Ordenados por Prioridad. Las prioridades son:

- 0: Mínimo Funcional
- 1: Fundamentales
- 2: Importantes
- 3: Deseables

Prioridad	Objetos	Gestos
0	Columna	Dibujar Columna
0	Columna	Agregar/Quitar/Editar Titulo
0	Kanban	Crear/Eliminar
0	Kanban	Mover a otra columna
1	Fila	Asociar titulo de fila con atributo que heredarán Kanbans contenidos
1	Kanban	Agregar/Quitar/Editar Atributos estándares en layout predefinido
1	Kanban	Asignar Fecha de Creación
1	Kanban	Seleccionar Color de Kanban
1	Kanban	Seleccionar Color de texto
1	Tablero	Crear Tablero (Selección de Filas)
1	Tablero	Crear/Eliminar columna
1	Tablero	Crear/Eliminar fila
2	Columna	Agregar número en encabezado
2	Columna	Agregar/Quitar/Editar Atributos estándares en layout predefinido
2	Fila	Seleccionar color de identificador de fila
2	Kanban	Mover a otra fila y heredar atributo por defecto de fila
2	Kanban	Agregar/Quitar/Editar Atributos personalizados en posiciones personalizadas
2	Kanban	Definir layout
2	Kanban	Poner papel encima (!)
2	Kanban	Eliminar Kanban, y reemplazarlo por 2 o mas nuevos kanban
2	Kanban(s)	Agrupar según atributo
3	Columna	Subdividir columna (historia)
3	Columna	Consolidar columnas (historia)
3	Columna	Consolidar filas en una columna

Figura 27: Listado de Gestos Ordenados por Prioridad

Debido a que, para la implementación se necesita un mínimo de semántica, principalmente de los atributos, se ha acordado definir los siguientes, que son comunes para la mayoría de los tableros:

- Tablero:
 - o Título
- Fila:
 - o Título
 - o Prioridad
 - o Estado
- Columna:
 - o Título
 - o Número de encabezado
- Tarjeta:
 - o Título
 - o Tipo
 - o Responsable
 - o Prioridad

- Fecha Solicitud
- Fecha Compromiso
- Contacto
- Dificultad Estimada
- Dificultad Medida
- Color de fondo
- Color de texto

Si bien los atributos tienen nombre, el significado se los sigue dando el equipo de trabajo. Por ejemplo, hay un atributo “Tipo” en la tarjeta, pero no es “Tarea”, “Proyecto”, “Entrega”, etc. como en las otras soluciones disponibles. Para nosotros sigue siendo un atributo genérico.

El resto de gestos son sobre los objetos mismos, y no requieren de una semántica explícita (crear tarjeta, eliminar columna, etc.), aunque si tienen un esquema visual predefinido, pero ocupado por gran parte de los tableros anteriormente descritos.

4.3. Modelo de Datos

Dada la simplicidad de esta solución, el modelo de datos es igual de simple para guardar toda la información necesaria, como se puede ver en la Figura 28: Modelo de Datos.

Tabla	Campos
Column	<ul style="list-style-type: none">- id- name- serial- header_number
Lane	<ul style="list-style-type: none">- id- name- status
Card	<ul style="list-style-type: none">- id- body- card_type_id- create_date- lane_id- column_id- responsable- priority- start_date- end_date- notes- difficulty_estimated- difficulty_measured
Card_Type	<ul style="list-style-type: none">- id- name- back_color- front_color
Log	<ul style="list-style-type: none">- action- date- id

Figura 28: Modelo de Datos

4.4. Observaciones

Con el diseño anterior, podemos implementar prácticamente todos los tableros analizados anteriormente. Tenemos una herramienta mucho más expresiva que cualquiera de las disponibles actualmente en el mercado.

5. Implementación

Definido el diseño de la herramienta a implementar, nos queda definir la plataforma de desarrollo a ocupar, que cumpla con los requerimientos detallados a continuación.

5.1. Requerimientos

La herramienta a desarrollar debe cumplir con requerimientos mínimos, que satisfagan las ventajas y permitan suplir las desventajas de los tableros físicos. Estos requerimientos son:

- Facilidad de uso
- Flexibilidad
- Visibilidad
- Concurrencia de múltiples usuarios (incluso un equipo entero)
- Accesibilidad desde múltiples fuentes
- Portabilidad
- Posibilidad de generar reportes (en general)
- Persistencia de la información
- Historia
- Cálculo de métricas relevantes
 - o Tiempo promedio de una tarea en comenzar
 - o Tiempo de ciclo, etc.

Además, como es una herramienta realizada en un ambiente de investigación, lo ideal es que esté implementada con estándares abiertos, normas de codificación, etc.

Un requisito deseable es que, ya que debe ser accesible desde cualquier lugar, la herramienta debe ser accesible desde dispositivos móviles (pantallas táctiles: iOS)

5.2. Plataformas de Desarrollo

La plataforma de desarrollo elegida que cumple con todos los requerimientos detallados anteriormente es la plataforma Web (HTML), en particular, se eligieron los siguientes ambientes:

- HTML5
- Javascript (jQuery)
- CSS3

La implementación de la herramienta fue realizada como una extensión a un proyecto de código abierto ya existente, llamado Kanban-Board en Google Code: (Ahmad, 2010), con el consentimiento del autor. El proyecto original se puede apreciar en la Figura 29: Kanban-Board (1) y en la Figura 30: Kanban-Board (2).

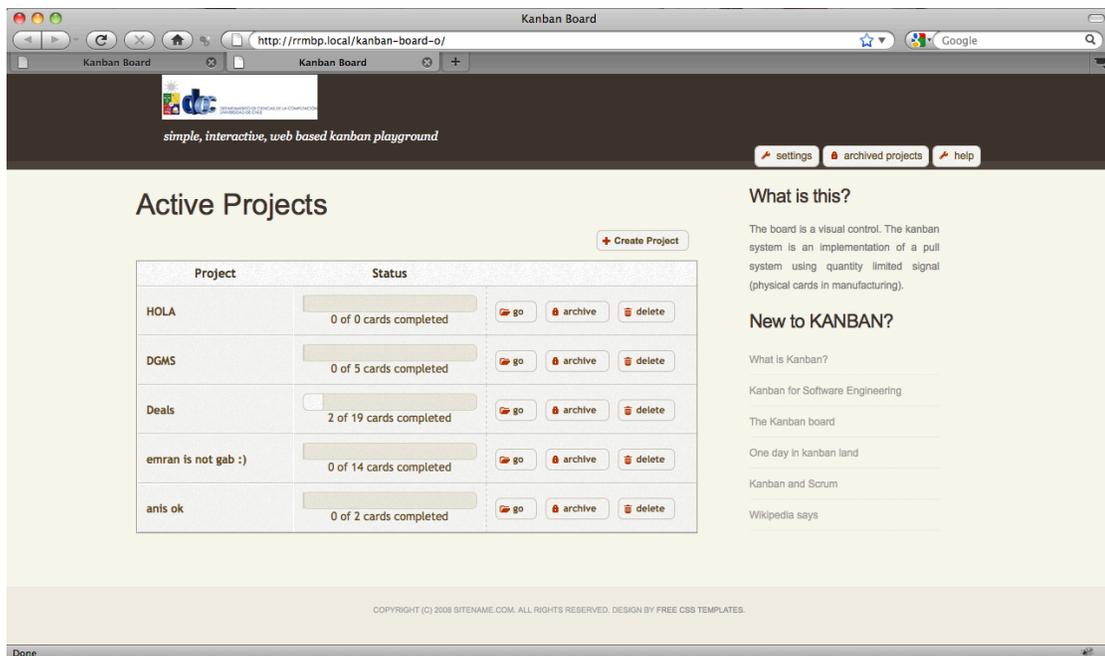


Figura 29: Kanban-Board (1)

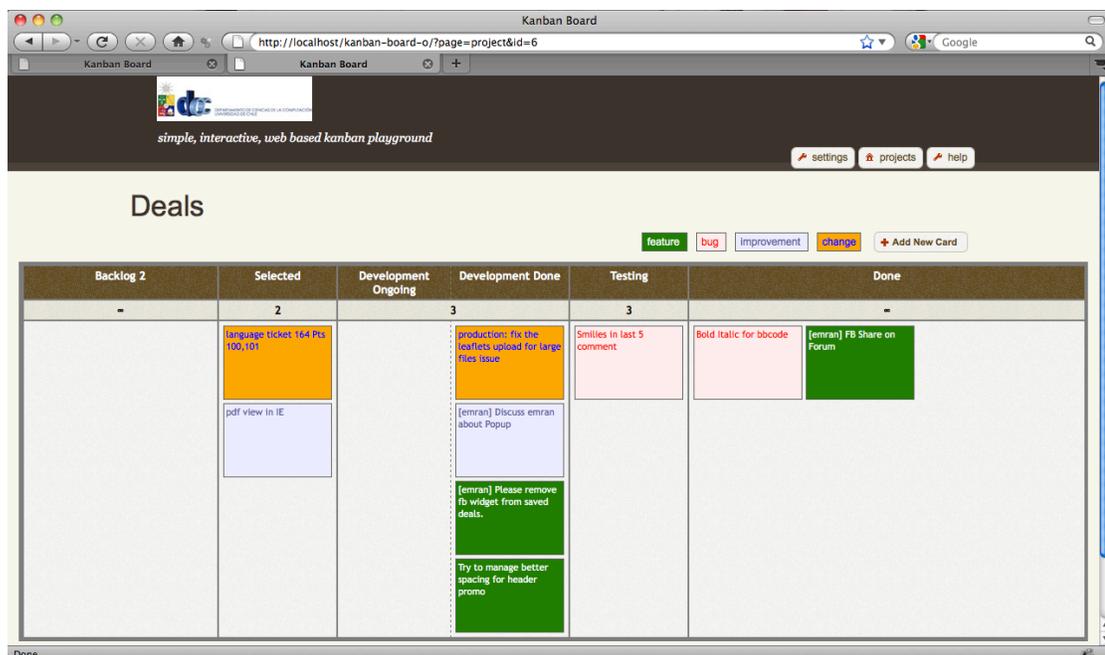


Figura 30: Kanban-Board (2)

La razón de esta decisión es que Kanban-Board es un tablero Kanban muy simple, pero que implementa los gestos básicos con una sintaxis y una semántica mezcladas, y además no tiene las funcionalidades requeridas por el presente proyecto.

Este proyecto está armado en el siguiente ambiente:

- HTML5
- PHP

- Javascript (jQuery)
- CSS3
- SQLite

Y fue desarrollado en:

- Mac OS X 10.6 (Snow Leopard)
- Apache 2.2.15
- PHP 5.3.3

5.3. Desarrollo

Como se mencionaba anteriormente, la herramienta Kanban-Board es un tablero Kanban virtual muy simple, pero que ya viene con una sintaxis y una semántica predefinidas para manejo de proyectos de software:

- Las filas y todas sus referencias se llaman “Proyectos”
- Sólo se pueden manejar proyectos individuales (1 fila)
- El espacio está muy mal aprovechado (para un tablero Kanban)
- No tiene bitácora
- No es soportado por dispositivos móviles

Este proyecto soporta los siguiente gestos, detallados en la Figura 25: Listado de Gestos Ordenados por Objeto:

- (1) Crear columna
- (2) Texto en encabezado
- (3) Número en encabezado
- (4) Atributos estándares de columna
- (5) Subdividir columna
- (10) Selección del color de texto de fila
- (12) Crear tarjeta
- (13) Mover a otra columna (dentro del a misma fila)
- (14) Atributos estándares de tarjeta (solo título)
- (15) Fecha de creación
- (16) Color de Kanban (por tipo de tarjeta)
- (17) Color de texto de Kanban
- (25) Crear Tablero

Para éste desarrollo, primero que todo, se le cambió el nombre al proyecto a *eKanban* (expressive Kanban, Kanban Expresivo) y está publicado en SubVersion (Riquelme, 2011)

Las modificaciones que se le realizaron fueron:

- Optimización del espacio ocupado
- Generalización de sintaxis y de semántica
- Cambio de texto de identificación de filas “Proyecto”, por el nombre genérico “Lane” (fila)

- Soporte para mostrar múltiples filas
- Posibilidad de elegir que filas mostrar
- Soporte para mover tarjetas entre filas y columnas

Y de los gestos ya soportados, se agregaron:

- (3) Soporte mejorado de advertencia cuando se sobrepase el número del encabezado
- (8) Asociar encabezados de fila con atributos que heredarán kanbans contenidos
- (13) Soporte mejorado de mover a otra columna, para mover también a otra fila
- (18) Mover a otra fila y heredar atributos por defecto de ella
- (20) Soporte para definir layout de tarjetas
- (21) Soporte para agregar notas

Además, se crearon los siguientes atributos semánticos optativos, compartidos entre la mayoría los tableros:

- Prioridad
- Fecha de inicio
- Fecha de término
- Dificultad estimada
- Dificultad medida
- Responsable

Y finalmente, se agregó un soporte inicial para bitácora, que guarda en base de datos la identificación de cada una de las acciones efectuadas sobre el tablero. Con la información obtenida, estamos a un paso de generar reportes.

La tabla de gestos para soportar, ordenados por prioridad quedó como se muestra en la Figura 31: Cumplimiento de Gestos por Prioridad.

Prioridad	Objetos	Gestos	Cumple
0	Columna	Dibujar Columna	Si
0	Columna	Agregar/Quitar/Editar Titulo	Si
0	Kanban	Crear/Eliminar	Si
0	Kanban	Mover a otra columna	Si
1	Fila	Asociar titulo de fila con atributo que heredarán Kanbans contenidos	Si
1	Kanban	Agregar/Quitar/Editar Atributos estándares en layout predefinido	Si
1	Kanban	Asignar Fecha de Creación	Si
1	Kanban	Seleccionar Color de Kanban	Si
1	Kanban	Seleccionar Color de texto	Si
1	Tablero	Crear Tablero (Selección de Filas)	Si
1	Tablero	Crear/Eliminar columna	Si
1	Tablero	Crear/Eliminar fila	Si
2	Columna	Agregar número en encabezado	Si
2	Columna	Agregar/Quitar/Editar Atributos estándares en layout predefinido	Si

2	Fila	Seleccionar color de identificador de fila	No
2	Kanban	Mover a otra fila y heredar atributo por defecto de fila	Si
2	Kanban	Agregar/Quitar/Editar Atributos personalizados en posiciones personalizadas	Si
2	Kanban	Definir layout	Si
2	Kanban	Poner papel encima (!)	Si
2	Kanban	Eliminar Kanban, y reemplazarlo por 2 o mas nuevos kanban	No
2	Kanban(s)	Agrupar según atributo	No
3	Columna	Subdividir columna (historia)	No
3	Columna	Consolidar columnas (historia)	No
3	Columna	Consolidar filas en una columna	No

Figura 31: Cumplimiento de Gestos por Prioridad

La herramienta expresiva para tableros Kanban **eKanban** se ve como se muestra en las Figura 32, Figura 33, Figura 34 y Figura 35.

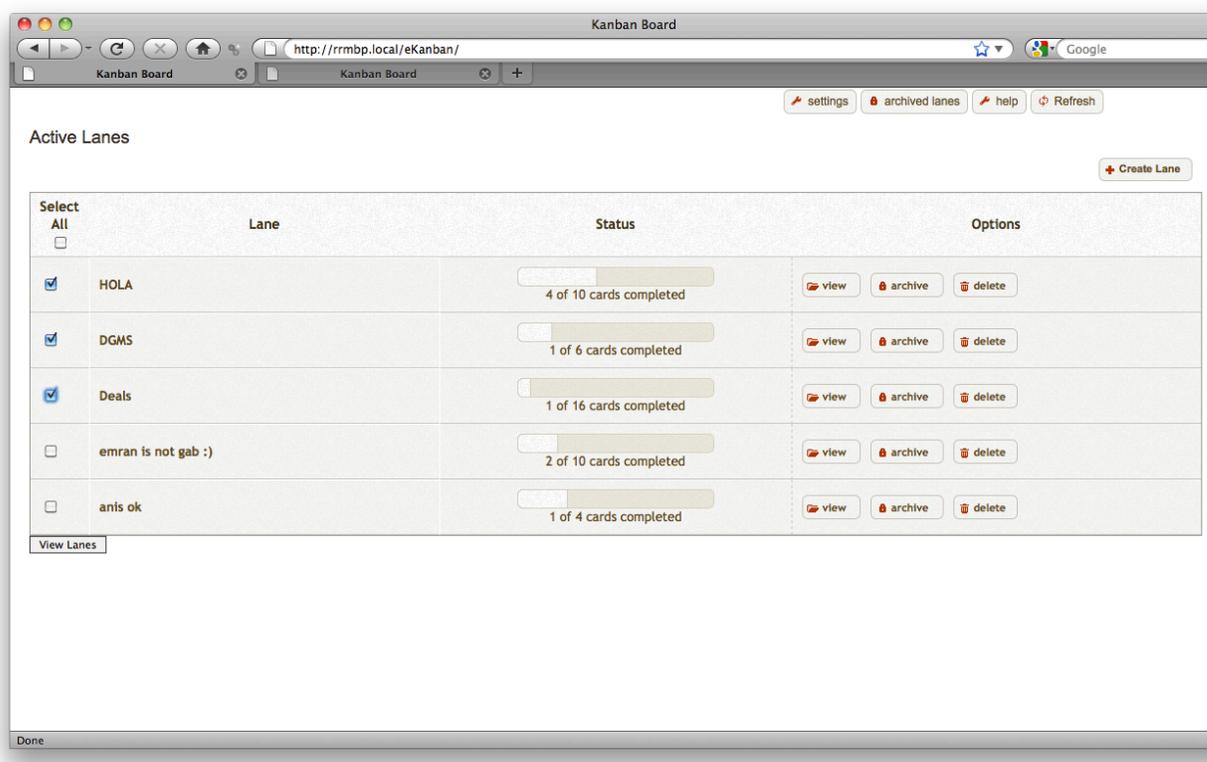


Figura 32: eKanban - Lanes Activos

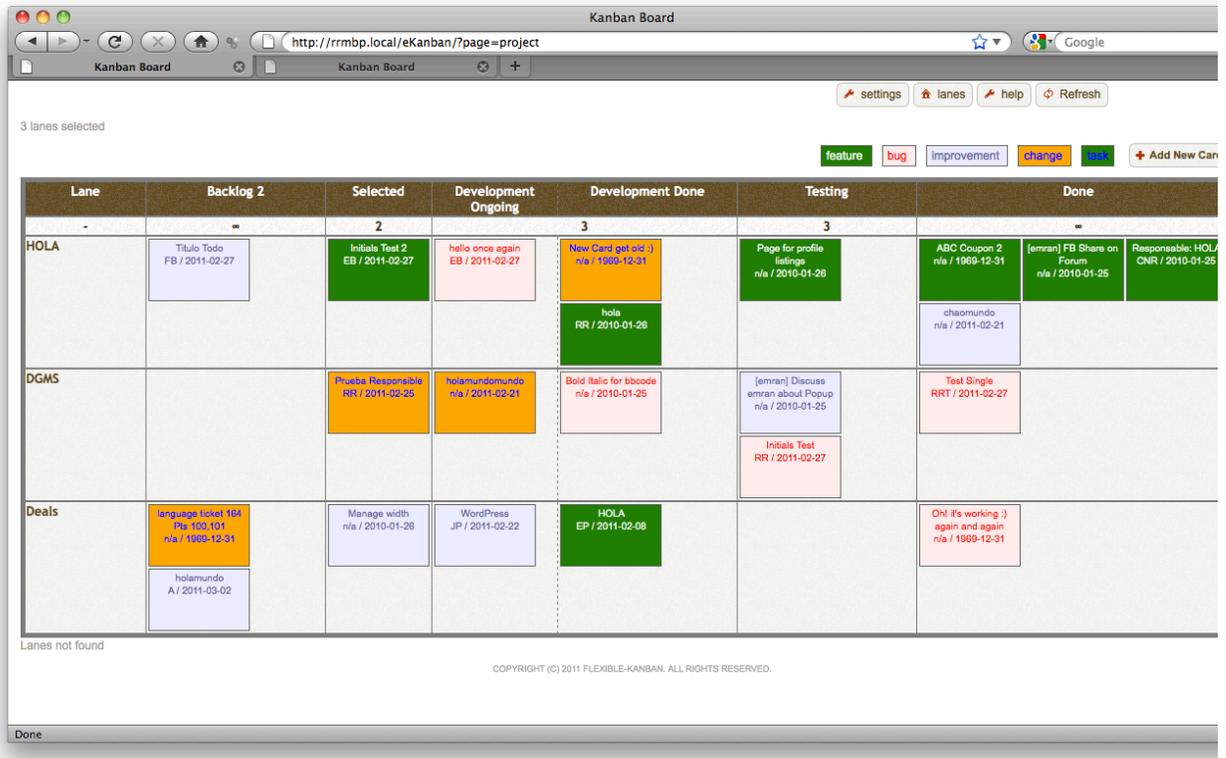


Figura 33: eKanban - Tablero Kanban (Lanes Seleccionados)

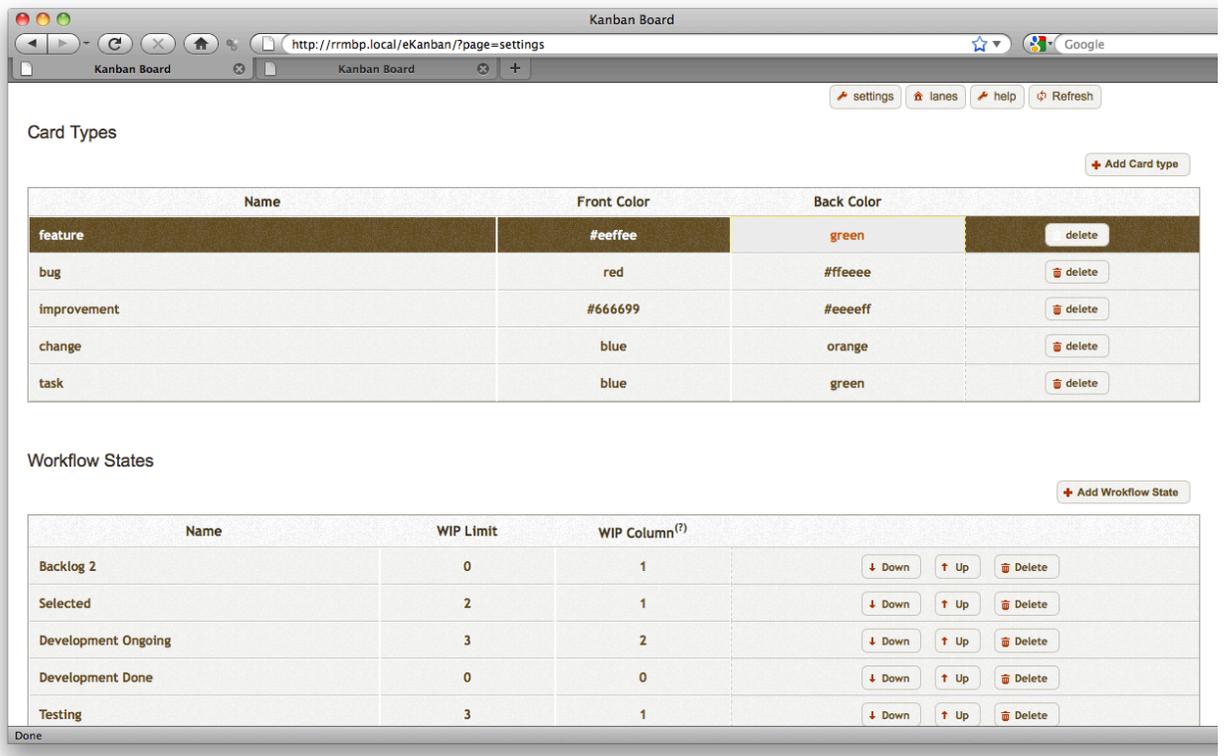


Figura 34: eKanban - Configuración

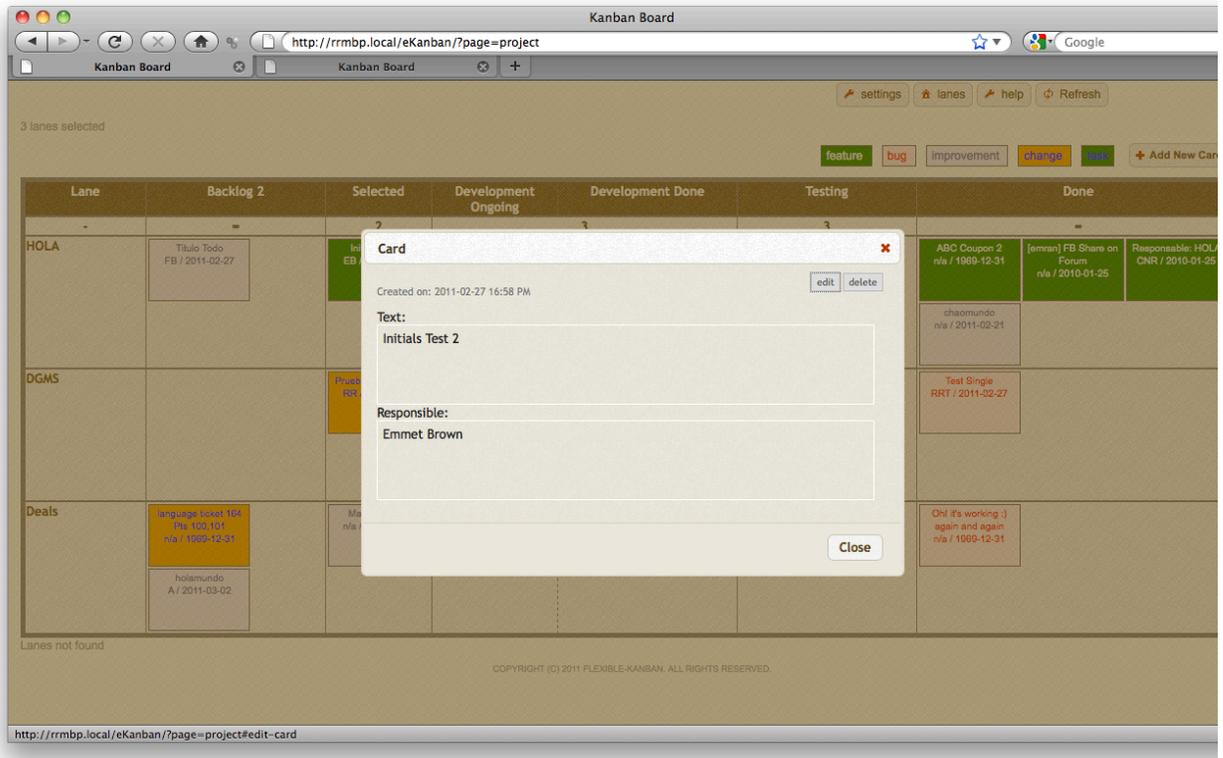


Figura 35: eKanban - Edición de una Tarjeta

Como una característica deseable, además eKanban es manipulable desde dispositivos móviles con pantallas táctiles (iPhone, iPad), como se muestra en la Figura 36: eKanban - iPhone y en la Figura 37: eKanban - iPad.

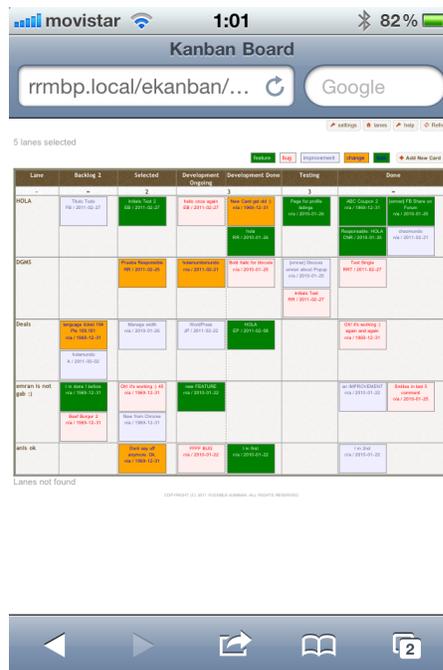


Figura 36: eKanban - iPhone

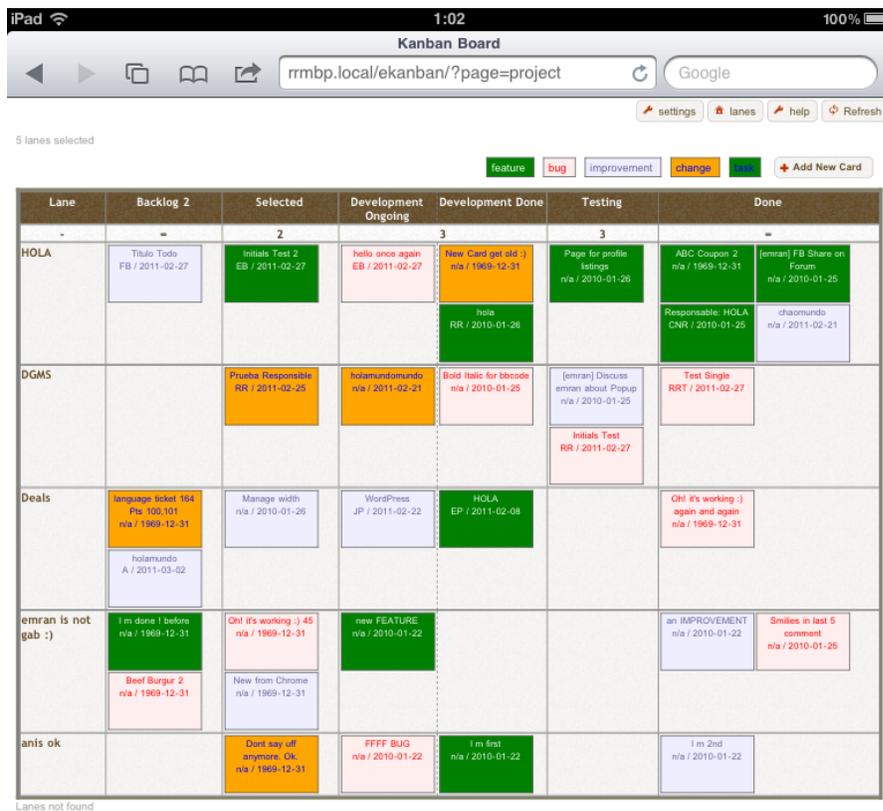


Figura 37: eKanban - iPad

El proyecto final está autodocumentado. Toda la información necesaria sobre su funcionamiento está en el mismo repositorio SubVersion del proyecto (Riquelme, 2011).

6. Pruebas

Para probar que realmente la herramienta implementada cumplía con los requerimientos básicos, fue utilizada en 2 situaciones del mundo real: el seguimiento de éste trabajo de tesis y como complemento al tablero Kanban del área de Arquitectura de la empresa El Mercurio.

Las experiencias se detallan a continuación.

6.1. Seguimiento de Trabajo de Tesis

eKanban con su diseño e implementación, no tuvo ningún problema ni limitación para soportar el seguimiento de un trabajo de tesis, sus etapas de desarrollo y la documentación.

Las extensiones desarrolladas ayudaron a simplificar la administración. El tablero se adaptó perfectamente al proyecto.

6.2. Kanban Arquitectura

eKanban tuvo un excelente comportamiento para complementar e incluso reemplazar el tablero Kanban de Arquitectura, especialmente con la extensión del manejo de filas y el hecho de poder elegir cuales filas mostrar.

En éste Kanban, las filas pueden ser desde tareas repetitivas, hasta grandes proyectos y actividades. Todas esas características son soportadas por eKanban.

Las filas de tareas repetitivas siempre existen, pero no siempre están activas, por lo que esa característica ayuda mucho a la visibilidad que incluso el tablero físico no tiene

6.3. Otras Observaciones

Dentro de las movidas permitidas por los tableros Kanban, solo las movidas más complejas vistas en los tableros teóricos no se pueden realizar.

De los tableros físicos y virtuales revisados, la mayoría de las movidas son posibles en eKanban.

7. Conclusiones

El pensamiento esbelto (Lean Thinking) busca agilizar los procesos. Para agilizar procesos, los equipos siguen ciertos principios y se auto-organizan. Esta organización maximiza recursos regulando la carga de trabajo.

Todas estas prácticas ágiles se pueden ordenar con Tableros Kanban, que reflejan de manera visible y clara el estado de cada actividad con Tarjetas.

Como los tableros son herramientas físicas, tienen muchas ventajas como: flexibilidad, visibilidad, concurrencia y facilidad de uso. Principalmente, porque tienen su sintaxis (forma) desacoplada de su semántica(significado). Pero también, tienen desventajas, ya que no son portables, no se tiene persistencia de la información, no hay historia, etc.

Para solucionar estas desventajas, existen variadas alternativas virtuales en internet para implementación de tableros Kanban, las que por la forma en que están diseñadas, acoplan sintaxis y semántica, obligando a que al ser usados, las actividades se acomoden a esas definiciones.

Algunas alternativas intentan desacoplar por el lado del código, pero su diseño viene acoplado.

La solución propuesta por el presente trabajo, resuelve en gran parte estos problemas. No es una solución tan flexible, pero si es muy expresiva, ya que al analizar gran cantidad de tableros físicos, virtuales y teóricos, se llegó a un conjunto de gestos, objetos y significados que son comunes para la gran mayoría de los tableros, especialmente minimizando la semántica y estandarizando los objetos sintácticos y definiendo operaciones sobre esos objetos, para que el tablero implementado pueda ser interpretado por el equipo que lo ocupa (como debe ser en la metodología de tableros Kanban).

El código asociado a la herramienta eKanban, al ser desarrollado como una extensión a un proyecto de código abierto ya existente, no está basado en el modelo de desacoplamiento propuesto, pero si está adaptado e inspirado en estos principios.

También cabe notar que la implementación fue realizada bajo los estándares abiertos de HTML5 (formato, estilos y funcionalidad desacoplados) y de forma modular con PHP.

Además, se implementó un sistema básico de bitácoras, donde por cada movimiento realizado en el tablero se guarda una bitácora. Al tener esta información, la generación de reportes es directa.

En resumen, el trabajo realizado cumple con la mayor parte de los objetivos propuestos.

8. Trabajo Futuro

A pesar de haber cumplido con gran parte de los objetivos propuestos, aún se puede seguir trabajando en la herramienta desarrollada en este proyecto. Las extensiones posibles recomendadas son las siguientes:

- Completar el desacoplamiento de las capas sintácticas y semánticas
- Generar un sistema de reportes que ocupe la información obtenida con el uso
- Implementar los gestos faltantes para que la herramienta sea capaz de soportar todos los tableros Kanban estudiados (Figura 31: Cumplimiento de Gestos por Prioridad)

Al ser un proyecto de código abierto, estar disponible en SVN, y estar construido de manera estándar y muy ordenada, hace que continuar con su desarrollo sea una tarea muy factible y no tan complicada.

9. Bibliografía

- Womack, J. P., & Jones, D. T. (2003). *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*. Free Press.
- Villena, A. (Diciembre de 2009). *Lean y Kanban: Agile Lean Day*. Retrieved Marzo de 2011 from <http://www.chileagil.cl/eventos/agileleanday-chile-2009/lean-y-kanban/>
- Villena, A. (Abril de 2010). *Nexon Podcast #4 - Agilidad*. Retrieved Marzo de 2011 from <http://www.podcaster.cl/2010/04/nexon-podcast-4/>
- Ahmad, A. (Septiembre de 2010). *kanban-board: A simple web based kanban board*. Retrieved Enero de 2011 from <http://code.google.com/p/kanban-board/>
- Anderson, D. (10 de Junio de 2010). *Thoughts On How Kanban Differs From Scrum*. Retrieved Marzo de 2011 from http://agilemanagement.net/index.php/Blog/thoughts_on_how_kanban_differs_from_scrum/
- Bailey, D. (8 de Diciembre de 2008). *Kanban in Software Development. Part 1: Introducing Kanban Boards and Pipelines*. Retrieved Marzo de 2011 from <http://www.lostechies.com/blogs/derickbailey/archive/2008/12/08/kanban-in-software-development-part-1-introducing-kanban-boards-and-pipelines.aspx>
- Bailey, D. (15 de Diciembre de 2008). *Kanban in Software Development. Part 2.5: A Variation on Queues - Pipelines for WIP and Done*. Retrieved Marzo de 2011 from <http://www.lostechies.com/blogs/derickbailey/archive/2008/12/15/kanban-in-software-development-part-2-5-a-variation-on-queues-pipelines-for-wip-and-done.aspx>
- Bailey, D. (8 de Diciembre de 2008). *Kanban in Software Development. Part 2: Completing the Kanban Board with Queues, Order Points and Limits*. Retrieved Marzo de 2011 from <http://www.lostechies.com/blogs/derickbailey/archive/2008/12/08/kanban-in-software-development-part-2-completing-the-kanban-board-with-queues-order-points-and-limits.aspx>
- Bailey, D. (19 de Diciembre de 2008). *Kanban in Software Development. Part 3: Andon and Jidoka - Handling Bugs and Emergency Fixes in Kanban*. Retrieved Marzo de 2011 from

<http://www.lostechies.com/blogs/derickbailey/archive/2008/12/19/kanban-in-software-development-part-3-andon-and-jidoka-handling-bugs-and-emergency-fixes-in-kanban.aspx>

Beck, K., Beedle, M., & Fowler, M. (2001). *Manifiesto por Agile Software*. Retrieved Marzo de 2011 from <http://www.agilemanifesto.org>

Kanbandev Yahoo! Group. (Marzo de 2011). Retrieved Marzo de 2011 from <http://finance.groups.yahoo.com/group/kanbandev>

Liker, J. (2004). *The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer*. McGraw-Hill Professional.

Limited WIP Society. (Marzo de 2011). Retrieved Marzo de 2011 from <http://www.limitedwipsociety.com>

Poppendieck, M., & Poppendieck, T. (2003). *Lean Software Development: An Agile Toolkit*. Addison Wesley.

Scotland, K. (11 de Junio de 2009). *Lean & Kanban 101*. Retrieved Marzo de 2011 from <http://availagility.wordpress.com/2009/06/11/zurich-lean-agile-scrum-slides/>

Skarin, M. (3 de Diciembre de 2010). *10 Kanban Boards And Their Context*. Retrieved Marzo de 2011 from <http://blog.crisp.se/mattiasskarin/2010/12/03/1291361993216.html>

So, Now You're An Agilist, What's Next? (2010). Retrieved Marzo de 2011 from <http://www.slideshare.net/jurgenappelo/so-now-youre-an-agilist-whats-next-10>

Riquelme, R. (Marzo de 2011). *eKanban - Assembla*. Retrieved Marzo de 2011 from SubVersion: <https://subversion.assembla.com/svn/ekanban/>

10. Anexos

10.1. Lean Thinking

Womack, James P.; Daniel T. Jones (1996). Lean Thinking

5 pillars:

1. Specify Value
2. Identify the Value Stream
3. Flow
4. Customer Pull
5. Pursue Perfection

10.2. W. Edwards Deming

W. Edwards Deming (1900 - 1993)

By adopting appropriate principles of management, organizations can increase quality and simultaneously reduce costs (by reducing waste, rework, staff attrition and litigation while increasing customer loyalty). The key is to practice continual improvement and think of manufacturing as a system, not as bits and pieces

System of Profound Knowledge, consisting of four parts:

1. Appreciation of a system
2. Knowledge of variation
3. Theory of knowledge
4. Knowledge of psychology

Fourteen key principles for management for transforming business effectiveness. First presented in his book "Out of the Crisis" (1986).

1. Create constancy of purpose toward improvement of product and service, with the aim to become competitive and stay in business, and to provide jobs.
2. Adopt the new philosophy. We are in a new economic age. Western management must awaken to the challenge, must learn their responsibilities, and take on leadership for change.
3. Cease dependence on inspection to achieve quality. Eliminate the need for inspection on a mass basis by building quality into the product in the first place.
4. End the practice of awarding business on the basis of price tag. Instead, minimize total cost. Move towards a single supplier for any one item, on a long-term relationship of loyalty and trust.
5. Improve constantly and forever the system of production and service, to improve quality and productivity, and thus constantly decrease cost.
6. Institute training on the job.
7. Institute leadership. The aim of supervision should be to help people and machines and gadgets to do a better job. Supervision of management is in need of overhaul, as well as supervision of production workers.
8. Drive out fear, so that everyone may work effectively for the company.

9. Break down barriers between departments. People in research, design, sales, and production must work as a team, to foresee problems of production and in use that may be encountered with the product or service.
10. Eliminate slogans, exhortations, and targets for the work force asking for zero defects and new levels of productivity. Such exhortations only create adversarial relationships, as the bulk of the causes of low quality and low productivity belong to the system and thus lie beyond the power of the work force.
- 11 a. Eliminate work standards (quotas) on the factory floor. Substitute leadership.
 - b. Eliminate management by objective. Eliminate management by numbers, numerical goals. Substitute workmanship.
- 12 a. Remove barriers that rob the hourly worker of his right to pride of workmanship. The responsibility of supervisors must be changed from sheer numbers to quality.
 - b. Remove barriers that rob people in management and in engineering of their right to pride of workmanship. This means, inter alia, abolishment of the annual or merit rating and of management by objective
13. Institute a vigorous program of education and self-improvement.
14. Put everyone in the company to work to accomplish the transformation. The transformation is everyone's work. "Massive training is required to instill the courage to break with tradition. Every activity and every job is a part of the process."

10.3. The Toyota Way

Liker, Jeffrey (2004). Toyota Way – 14 Management Principles from the World's Greatest Manufacturer

Section I — Long-Term Philosophy

1. Base your management decisions on a long-term philosophy, even at the expense of short-term financial goals.

Section II — The Right Process Will Produce the Right Results

2. Create a continuous process flow to bring problems to the surface.
3. Use "pull" systems to avoid overproduction.
4. Level out the workload (heijunka). (Work like the tortoise, not the hare).
5. Build a culture of stopping to fix problems, to get quality right the first time.
6. Standardized tasks and processes are the foundation for continuous improvement and employee empowerment.
7. Use visual control so no problems are hidden.
8. Use only reliable, thoroughly tested technology that serves your people and processes.

Section III — Add Value to the Organization by Developing Your People

9. Grow leaders who thoroughly understand the work, live the philosophy, and teach it to others.
10. Develop exceptional people and teams who follow your company's philosophy.
11. Respect your extended network of partners and suppliers by challenging them and helping them improve.

Section IV: Continuously Solving Root Problems Drives Organizational Learning

12. Go and see for yourself to thoroughly understand the situation (Genchi Genbutsu).
13. Make decisions slowly by consensus, thoroughly considering all options; implement decisions rapidly (nemawashi).
14. Become a learning organization through relentless reflection (hansei) and continuous improvement (kaizen).

10.4. Lean Software Development

Poppendieck's - 7 Principles

1. Eliminate Waste
2. Build Quality In
3. Create Knowledge
4. Defer Commitment
5. Deliver Fast
6. Respect People
7. Optimise the Whole