

Universidad Nacional de Salta  
Facultad de Ciencias Exactas  
Seminario de Sistemas



Proyecto Maixon Buscador Inmobiliario

Informe Final

**Nro de actuación:** 1008/07  
**Alumno:** Dante Rubén Alfredo Cáceres - LU: 207.655  
**Tutora:** Lic. Loraine Gimson  
**Subcomisión evaluadora:** Lic Patricia Mac Gaul  
Lic Martín Diaz  
Lic Carina Reyes

## Índice

- [1. Objetivos](#)
- [2. El sistema Objeto](#)
  - [2.1 Objetivos y Metas](#)
  - [2.2 Equipo Actual](#)
  - [2.3 Estado Actual](#)
  - [2.4 Servicios](#)
  - [2.5 Descripción del Mercado](#)
  - [2.6 Tecnologías de desarrollo](#)
- [3. Justificación del tema](#)
  - [3.1 Alcance del Seminario](#)
- [4. Plan de trabajo](#)
- [5. Estudio de factibilidad](#)
  - [5.1 Factibilidad técnica](#)
  - [5.2 Factibilidad económica y financiera](#)
  - [5.3 Factibilidad operativa](#)
  - [5.4 Factibilidad legal](#)
  - [5.5 Conclusión del Estudio de Factibilidad](#)
- [6. Metodologías](#)
  - [6.1 Introducción](#)
  - [6.2 Modelos de procesos de software](#)
  - [6.3 Modelos prescriptivos](#)
    - [6.3.1 Modelo en Cascada o Ciclo de vida clásico](#)
    - [6.3.2 Modelo de Prototipos](#)
    - [6.3.3 Modelo en espiral](#)
    - [6.3.4 El proceso unificado](#)
  - [6.4 Modelos no prescriptivos](#)
    - [6.4.1 eXtreme Programming](#)
    - [6.4.2 SCRUM](#)
- [7 - Aplicación Práctica de SCRUM](#)
  - [7.1 Roles y Responsabilidades](#)
  - [7.2 Product Backlog Inicial](#)
  - [7.3 Determinación de Duración de Sprint y Velocidad](#)
  - [7.4 Estimación del funcionalidades del Product Backlog](#)
  - [7.5 Estado actual del sistema](#)
    - [7.5 Sprint 0](#)
      - [7.5.1 Planificación del Sprint 0](#)
      - [7.5.2 Seguimiento y Control del Sprint 0](#)
      - [7.5.3 Revisión del Sprint 0](#)
      - [7.5.4 Resultados del Sprint 0](#)
      - [7.5.4 Retrospectiva del Sprint 0](#)
      - [7.5.5 Conclusión preliminar](#)
    - [7.6 Sprint 1](#)
      - [7.6.1 Planificación de Sprint 1](#)
      - [7.6.2 Seguimiento y Control del Sprint 1](#)
      - [7.6.3 Revisión del Sprint 1](#)
      - [7.6.4 Resultados del Sprint 1](#)

[7.6.4 Retrospectiva del Sprint 1](#)

[7.6.5 Conclusión preliminar](#)

[7.7 Sprint 2](#)

[7.7.1 Planificación del Sprint 2](#)

[7.7.2 Seguimiento y Control del Sprint 2](#)

[7.7.3 Revisión del Sprint 2](#)

[7.7.4 Resultados del Sprint 2](#)

[7.7.4 Retrospectiva del Sprint 2](#)

[7.7.5 Conclusión preliminar](#)

[8. Conclusiones generales](#)

[9. Anexo](#)

[9.1 Historias de Usuario](#)

[10. Bibliografía](#)

## 1. Objetivos

El principal objetivo de este seminario es el estudio y aprendizaje de la metodología ágil Scrum para su implementación en tres iteraciones para una serie de requisitos definidos por el cliente y que serán desarrollados sobre un sistema existente denominado Maixon.com (<http://www.maixon.com/>).

## 2. El sistema Objeto

Maixon.com es un sistema web argentino lanzado en febrero de 2012 en el cual las Inmobiliarias se registran, publican sus propiedades para la venta y gestionan clientes. Cuenta con 3 subsistemas: FrontEnd, Manager y Backend. FrontEnd es de acceso público para todo el mundo y cuenta con un robusto buscador de propiedades y posicionamiento de propiedades en mapas. Manager es el sistema que utilizan los usuarios de las inmobiliarias para publicar propiedades, gestionar clientes, llevar registro de su calendario de actividades, etc. Backend es el sistema de los dueños de Maixon en donde configuran todo el portal, seguimiento de abonos, gestión de inmobiliarias, propiedades, etc.

### 2.1 Objetivos y Metas

El objetivo de Maixon es dominar el mercado inmobiliario en internet. Ser un referente a la hora de buscar propiedades para compra o alquiler. Para ello, las metas son desarrollar y brindar novedosos servicios que otros portales del rubro inmobiliario no ofrece a las inmobiliarias ni a los usuarios finales que buscan propiedades.

### 2.2 Equipo Actual

El equipo de Maixon esta conformado por cuatro personas:

- Leandro Borges, Gerente. Ituzaingó, Buenos Aires.
- Norberto Riccitelli, Director ejecutivo. Ituzaingó, Buenos Aires.
- Dante Cáceres, Líder del proyecto. San Salvador de Jujuy, Jujuy.
- Mauricio Andrada, Diseñador. Rosario, Santa Fé.

La modalidad de trabajo es el teletrabajo, se trata de un equipo descentralizado. La manera de comunicarnos es diaria y continuamente mediante llamadas, teleconferencias y sesiones de chats a través del software de comunicación Skype (<http://www.skype.com/es/>), seguimiento de tareas a través del software Asana (<http://asana.com/>).

### 2.3 Estado Actual

A un año de su lanzamiento Maixon se encuentra en etapa de investigación, crecimiento y desarrollo de nuevos servicios, cuenta con 120 inmobiliarias registradas, de las cuales 40 pagan su abono regularmente y las demás usan solamente servicios gratuitos. Si bien las inmobiliarias confían en los servicios, Maixon necesita crecer y acaparar más clientes para autoabastecer los gastos de emprender en nuevos y modernos servicios.

### 2.4 Servicios

Maixon ofrece a las inmobiliarias contratar servicios de acuerdo a 3 planes, Basic, Advanced y Premium. El primero, Basic es gratuito y limitado, ofrece publicar propiedades y que las mismas sean accedidas a través del buscador del Front End. Advanced permite además de Basic, la posibilidad de crear sucursales y agregar usuarios a la inmobiliaria, permite imprimir carteleras de propiedades y la creación de la web personalizada e imagen corporativa de la empresa. Premium es un tipo de abono que ofrece lo mismo que Advanced más privilegios de posicionamiento en resultados de búsqueda de sus propiedades en FronEnd e integración con redes sociales, especialmente Facebook (<https://www.facebook.com/>) y Google Plus (<https://plus.google.com/>).

## 2.5 Descripción del Mercado

El mercado inmobiliario describe a Demanda como gente que busca inmuebles (casas, departamentos, locales, oficinas, etc) para alquilar o comprar y a la Oferta como empresas inmobiliarias o simplemente particulares que ofrecen propiedades inmuebles para alquiler o venta. Si a este mercado le sumamos el factor tecnológico podemos decir que un portal inmobiliario es un sitio web que engloba usuarios oferentes y usuarios demandantes brindándoles a ambos los servicios necesarios para ayudarlos a que logren concretar su objetivo: alquilar, vender o comprar propiedades.

Para los usuarios demandantes, algunas ventajas de usar un sistema web son: buscador de propiedades, visualización de ubicaciones de las propiedades en un mapa, posibilidad de realizar preguntas a las inmobiliarias, visualizar fotos y videos de la propiedad.

Para los usuarios oferentes, algunas ventajas son: gestión y seguimiento de propiedades, bajo costo del abono comparado con un sistema hecho a medida, posicionamiento en motores de búsqueda como Google (<http://www.google.com>). Integración con otras inmobiliarias y acceso a un mercado de competencia perfecta.

## 2.6 Tecnologías de desarrollo

El sistema actual se encuentra desarrollado con las siguientes tecnologías:

- Servidor Web Apache (<http://www.apache.org/>)
- Servidor de bases de datos MySQL (<http://www.mysql.com/>)
- Lenguaje de programación PHP (<http://php.net/>)
- Framework MVC de programación CakePhp (<http://cakephp.org/>)
- Framework Ajax y Javascript jQuery (<http://jquery.com/>)
- API Google Maps (<https://developers.google.com/maps/?hl=es>) para el posicionamiento de propiedades en mapa.
- API Facebook (<http://developers.facebook.com/>) para el registro y login de usuarios.
- API Foursquare (<https://developer.foursquare.com/>) para las recomendaciones de puntos de interés cercanos a una propiedad.

### 3. Justificación del tema

En su primera versión Maixon fue lanzado en Febrero de 2012 y al obtener un buen impacto en el mercado, los dueños deciden junto conmigo continuar el desarrollo y crecimiento de Maixon pero eran necesarias nuevas ideas y servicios que brindar. Por su parte, los dueños proponen el módulo CRM (Customer Relationship Management) un módulo que permita a los usuarios de las inmobiliarias gestionar y fidelizar el contacto con los clientes, o sea realizar un seguimiento casi automático de acuerdo al perfil de búsqueda de propiedades de un cliente, el sistema almacena ese perfil de búsqueda y luego a medida que se publican nuevas propiedades que cumplen ese perfil el sistema debe notificar automáticamente a los clientes.

Por mi parte y en base a mi experiencia en el desarrollo de sistemas para el rubro inmobiliario que adquirí en Remax Argentina (<http://www.agentesremaxuno.com.ar/>), sabía que las principales características que las inmobiliarias desean pero que no encuentran en portales web son:

- El acceso a un sistema de Pagos de abonos e historial del mismo.
- Posibilidad de contar con su propia web personalizada e imagen corporativa.
- Disponibilidad de expansión del sistema hacia otros países, aprovechando los patrones de diseño Internacionalización y Localización.
- Importación y Exportación de propiedades.

Estas propuestas fueron analizadas en reuniones con los dueños de Maixon y aprobadas para su desarrollo y forman parte del alcance del presente seminario. Para ello se desarrollarán los módulos aplicando la metodología ágil SCRUM (<http://scrummethodology.com/>, <http://es.wikipedia.org/wiki/Scrum>). La elección de la metodología se debe a los siguientes motivos:

- Se cuenta con la total colaboración y disponibilidad por parte de los dueños de Maixon quienes son expertos en el rubro Inmobiliario, nos comunicamos a diario vía Skype y mantenemos reuniones sobre las tareas a realizar y el estado de las mismas.
- Los dueños de Maixon cuentan con nivel de experiencia baja o media en tecnologías, herramientas y métodos de desarrollo de sistemas, por lo que puede suceder que haya confusión a la hora de establecer requerimientos y luego haya que hacer modificaciones sobre la marcha.
- Al ser un trabajo a distancia, ellos en Buenos Aires y yo en San Salvador de Jujuy, los avances de desarrollo de tareas necesitan ser reportados a diario para justificar las horas de trabajo invertidas en el desarrollo de los requerimientos.
- La forma en que venimos trabajando actualmente es: realizamos una reunión en donde el cliente expone cuales son los nuevos requerimientos para el sistema, se asesora al cliente sobre la factibilidad de los mismos y si es necesario se proponen características adicionales, luego de eso se cuenta con un listado de requerimientos formales sobre los cuales se va a trabajar, se estiman los tiempos de cada uno y se los registran en el sistema de seguimiento de tareas Asana. Se comienza con la ejecución de las tareas, se actualizan en el sistema Asana cuáles de las tareas se encuentran en ejecución y se

realizan a diario 2 o 3 reuniones: una al principio de la jornada para comunicar con cuales tareas se va a trabajar durante el día, otra reunión en caso de ser necesario para consultar al cliente sobre dudas o dificultades encontradas durante la ejecución de una tarea y una reunión al finalizar la jornada informando los avances durante el día, se presenta una nueva versión del sistema para que el cliente pueda testear y verificar el avance y se planifica la jornada siguiente. También se actualizan las tareas en el sistema Asana con los estados correspondientes: continua en ejecución, terminada, etc. Considero que esta modalidad se asemeja en algunos puntos a Scrum, por ejemplo en la definición de requisitos con el cliente y las reuniones para informar los avances, actualización de estados de tareas y planificación de nuevas tareas.

### 3.1 Alcance del Seminario

En recientes reuniones realizadas, el cliente ha determinado cuatro nuevos módulos como requerimientos a desarrollarse e integrarse a Maixon. Con el presente seminario se desea aplicar la metodología Scrum en tres iteraciones de diez días laborales cada una para de cubrir en gran parte las tareas definidas por el cliente para estos módulos.

- **Módulo Payments:** hasta el lanzamiento de Maixon.com en 2012, las inmobiliarias pagaban manualmente su abono, hacían un deposito bancario en la cuenta de los dueños de Maixon. Entonces se me ocurrió en base a mi experiencia con Webservices, que habia que aprovechar la herramienta DineroMail (<http://www.dineromail.com.ar/>), que consiste en una pasarela de pago que permite realizar transferencias de dinero de una cuenta a otra de manera automática, amigable y sin acudir a una entidad bancaria o en su defecto generar una factura correspondiente al abono y realizar el pago en las entidades Rapipago. Será la única manera de pago de abonos de inmobiliarias en Maixon.com
- **Sistema Maixon Pages:** Cada inmobiliaria cuenta con la gestión de sus propiedades publicadas en la web general de Maixon.com en donde se publican todas las demas propiedades de todas las inmobiliarias, pero ¿porqué no permitirles a cada inmobiliaria contar con su propia web en donde se publiquen solo sus propiedades y un diseño personalizado? Nos reunimos con los dueños de Maixon y un diseñador externo que realizaría las tareas de generación de Templates los cuales serían configurables al gusto de cada inmobiliaria con colores e imágenes y luego el módulo Maixon Pages que será un único sistema que se comporta de manera diferente según la URL por la cual se accede, esto es, para cada URL de una inmobiliaria el sistema actúa de manera independiente recuperando y mostrando las propiedades de esa inmobiliaria, mapas, datos e imágenes de la inmobiliaria.
- **Módulo de Internacionalización y Localización:** la idea es básicamente expandir la aplicación que en un principio era disponible solo para Argentina, poder disponerla en otros países e idiomas. El Framework Cakephp (<http://cakephp.org/>) utilizado para desarrollar Maixon.com nos brindaba la posibilidad de aplicar Internacionalización, o sea, que los datos de la aplicación estuvieran disponibles para otros idiomas manteniendo la estructura de aplicación base, por ejemplo: cuando en argentina cargan una propiedad "Casa en Venta zona Ituzaingó" en otro país pueden cargar "Home for Sale in Detroit" también teniendo en cuenta que al tratarse de propiedades, las cotizaciones de las mismas varían en cuanto a



la moneda de cada país y en cuanto a Localización, brindar la posibilidad de que la aplicación tenga una interfaz adecuada a cada país, localización tiene que ver detectar el idioma del navegante y mostrar los textos de la aplicación según corresponda en cada país. Esta propuesta fue aceptada y actualmente se encuentra en ejecución, se desea que Maixon.com se encuentre disponible en sus versiones Argentina (pesos y dólares), España (euros) y México (peso mexicano y dolar) y con posibilidad de abarcar más países en un futuro.

- **Módulo de Importación y Exportación de Propiedades:** es la propuesta más importante, robusta y novedosa para Maixon.com, por un lado permitir exportar las propiedades desde Maixon hacia otros portales y por otro lado importar propiedades desde las bases de datos de las inmobiliarias hacia Maixon.
  - **Exportación:** el objetivo es que Maixon se promoció a través de otros portales y de esa manera lograr que las propiedades de las inmobiliarias tengan más posibilidades de concretar sus ventas, para ello había que utilizar los webservices de otros portales para realizar las publicaciones. Esta propuesta fue aprobada y se encuentra en ejecución, se desea desarrollar un proceso en Maixon que recupere las propiedades publicadas el día anterior y las exporte a los portales: Alamaula (<http://www.alamaula.com/>), Natyo (<http://natyo.com.ar/>), Olx (<http://www.olx.com.ar/>), Sumavisos (<http://www.sumavisos.com.ar/>) y Trovit (<http://www.trovitargentina.com.ar/>).
  - **Importación:** uno de los inconvenientes que nos presentaban las inmobiliarias a la hora de registrarse y formar parte de Maixon era “quiero formar parte de Maixon, pero necesito que migre las propiedades de mi página web hacia Maixon” ó “veníamos trabajando con Argenprop, queremos poder migrar nuestras propiedades hacia Maixon”. Claramente teníamos un inconveniente ya que desconocíamos las tecnologías que usaban en las páginas web de las inmobiliarias y también los portales como Argenprop (<http://www.argenprop.com/>) no permiten realizar exportación de propiedades, sino solamente publicación. Entonces se me ocurrió que si existen herramientas XML Parser para comunicación entre dos sistemas, podríamos desarrollar algo similar, un HTML Parser para poder leer las propiedades, sus atributos como Título, Dirección, Imágenes, Descripción, Ubicación en Mapa, etc de las inmobiliarias en Argenprop y de esa manera importarlas hacia Maixon.

Con la aplicación de la metodología se desea aportar al seminario la siguiente documentación, herramientas y roles:

- Product Owner: son los dueños del sistema Maixon, formados por el Gerente y Director Ejecutivo.
- Scrum Master: Líder del proyecto.
- Equipo de trabajo: Líder del proyecto y Diseñador.
- Product Backlog: en donde el Product Owner va a definir el listado de funcionalidades prioritarias necesarias para desarrollar los módulos, junto con los valores de negocio de cada funcionalidad y los valores de esfuerzo estimados por el Scrum Master. Se van a entregar versiones del Product Backlog inicial, intermedias y final, ya que la misma puede variar en tamaño y esfuerzo a lo largo de las iteraciones.

- Sprint Backlog: para cada Sprint se va a describir las reuniones entre el Product Owner y el Scrum Master, se va a documentar la planificación del Sprint en donde se van a revisar los elementos de alta prioridad del Product Backlog y cuáles de ellos desea el Product Owner se ejecuten en el siguiente Sprint. Luego se van a describir las reuniones entre el Scrum Master y el Diseñador para detallar cómo se van a implementar técnicamente las funcionalidades seleccionadas en el Sprint. Durante el desarrollo de las funcionalidades, si surgen dudas y reuniones con el Product Owner, serán documentadas sólo las que aporten experiencia al seminario. Se va a documentar las pruebas de unidad realizadas por el equipo y pruebas realizadas por el cliente. También serán documentadas las Revisiones de Sprint donde el equipo y el Product Owner analizan el incremento que hubo durante el Sprint.
- Scrum Diario: en algunas ocasiones relevantes se van a exponer reportes del Scrum diario de trabajos realizados por el equipo de trabajo, debido a que las iteraciones van a ser de 10 días, se considera documentar uno o dos Scrum diarios por iteración.
- Burndown Chart: para cada Sprint se va a presentar el gráfico estimado de esfuerzo y las actualizaciones reales que se llevaron a cabo.
- Documentación acerca del framework de trabajo CakePhp y como se ha implementado en el proyecto.
- Conclusiones: se van a documentar las conclusiones generales obtenidas luego de aplicar la metodología.

## 4. Plan de trabajo

Los módulos se desarrollarán bajo el siguiente cronograma de tareas en 3 iteraciones aplicando la metodología Scrum.

- Tarea 1: Estudio de metodología Scrum
- Tarea 2: Reunión con cliente, especificación de requerimientos
- Tarea 3: Recopilación de herramientas para el proyecto
- Tarea 4: Definición Product Backlog
- Tarea 5: Definición y Planificación de Sprint
- Tarea 6: Iteración 1 - Ejecución y seguimiento del Sprint
- Tarea 7: Revisión del Sprint y retroalimentación
- Tarea 8: Definición y Planificación de Sprint
- Tarea 9: Iteración 2 - Ejecución y seguimiento del Sprint
- Tarea 10: Revisión del Sprint y retroalimentación
- Tarea 11: Definición y Planificación de Sprint
- Tarea 12: Iteración 3 - Ejecución y seguimiento del Sprint
- Tarea 13: Revisión del Sprint y retroalimentación
- Tarea 14: Conclusiones

	!	Nombre	Duración	Inicio	Fin	Predecesoras	Recursos
1		Estudio de metodología Scrum	10d	13/05/2013	24/05/2013		Scrum Master / Programador
2		Reunion con Product Owner, especificación de requerimientos	3d	13/05/2013	15/05/2013		Stakeholder / Product Owner, Scrum Master / Programador, Diseñador
3		Recopilación de herramientas para el proyecto	4d	16/05/2013	21/05/2013	2	Scrum Master / Programador
4		Definición Product Backlog	1d	27/05/2013	27/05/2013	1,2	Stakeholder / Product Owner, Scrum Master / Programador
5		Definición y Planificación de Sprint	1d	28/05/2013	28/05/2013	4	Stakeholder / Product Owner, Scrum Master / Programador
6		Iteración 1 - Ejecución y seguimiento del Sprint	10d	29/05/2013	11/06/2013	5	Scrum Master / Programador, Diseñador
7		Revisión del Sprint y retroalimentación	1d	12/06/2013	12/06/2013	6	Stakeholder / Product Owner, Scrum Master / Programador
8		Definición y Planificación de Sprint	1d	13/06/2013	13/06/2013	7	Stakeholder / Product Owner, Scrum Master / Programador
9		Iteración 2 - Ejecución y seguimiento del Sprint	10d	14/06/2013	27/06/2013	8	Scrum Master / Programador, Diseñador
10		Revisión del Sprint y retroalimentación	1d	28/06/2013	28/06/2013	9	Stakeholder / Product Owner, Scrum Master / Programador
11		Definición y Planificación de Sprint	1d	01/07/2013	01/07/2013	10	Stakeholder / Product Owner, Scrum Master / Programador
12		Iteración 3 - Ejecución y seguimiento del Sprint	10d	02/07/2013	15/07/2013	11	Scrum Master / Programador, Diseñador
13		Revisión del Sprint y retroalimentación	1d	16/07/2013	16/07/2013	12	Stakeholder / Product Owner, Scrum Master / Programador
14		Conclusiones	5d	17/07/2013	23/07/2013	13	Scrum Master / Programador

Figura 4.1 Plan de trabajo, cronograma de tareas

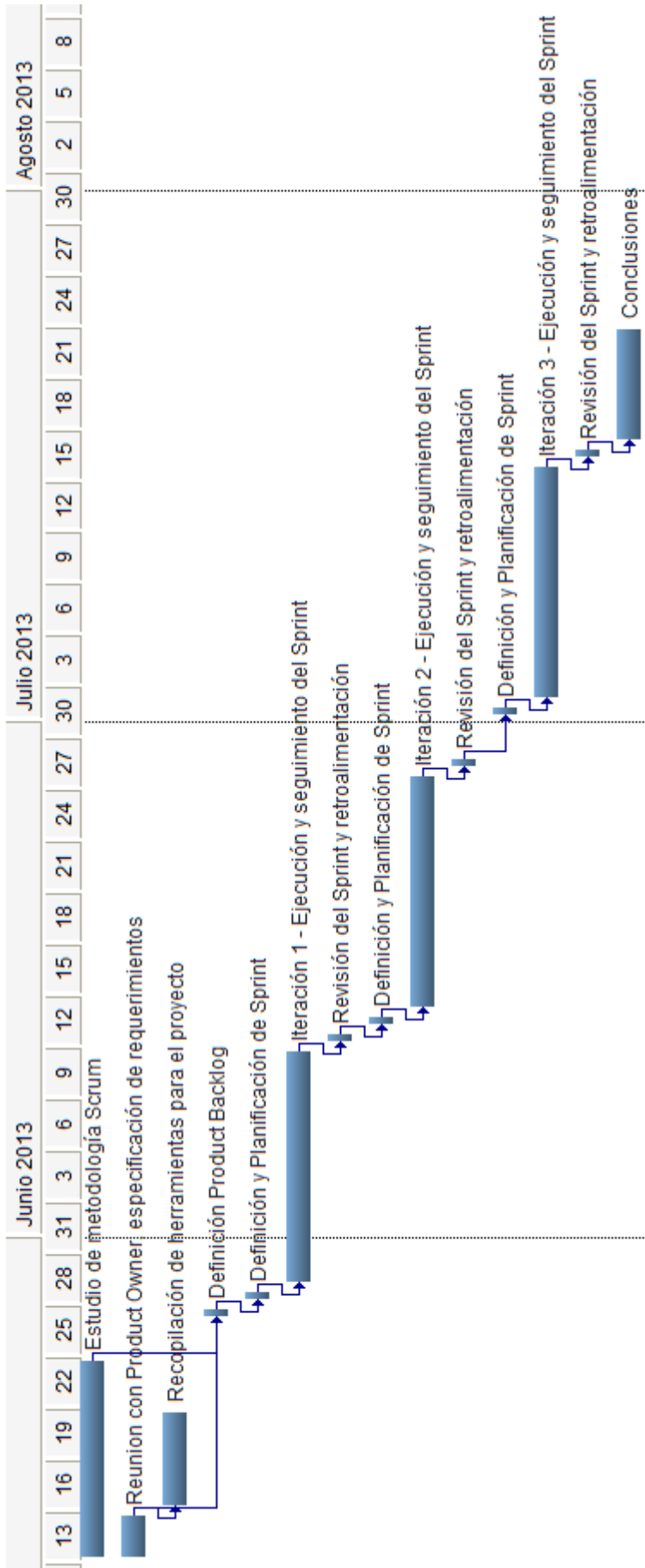


Figura 4.2 Diagrama de Gantt de cronograma de tareas

## 5. Estudio de factibilidad

El objetivo del siguiente estudio es determinar si se disponen de los recursos necesarios para desarrollar los requerimientos que solicita el cliente para el sistema web Maixon.com

### 5.1 Factibilidad técnica

El sistema actual se encuentra en ambiente de producción alojado en un servidor dedicado en el cual se pueden realizar las configuraciones necesarias para la creación de nuevos dominios y bases de datos. Para las nuevas funcionalidades a implementar no se requieren nueva tecnología hardware.

- Sistema Maixon Pages será una aplicación independiente de la aplicación Maixon pero que compartirá la misma base de datos de propiedades y será montado en dominio independiente dentro del mismo servidor.
- Para los módulos Payments e Importador/Exportador que requieren uso de APIs de pagos y exportadoras, será necesario contar con la biblioteca Curl de php (<http://php.net/manual/es/book.curl.php>) que comunmente se usa para interactuar con otros sistemas a través de APIs. Esta biblioteca se encuentra disponible en la configuración de PHP del servidor, solo debe ser habilitada para su uso.
- Para el módulo de Payments, la API de pago de DineroMail (<https://ar.dinero.com/biblioteca>) cuenta con una librería desarrollada en PHP, por lo que es viable su integración y uso en Maixon.
- Para el módulo de Internacionalización y Localización se cuenta con documentación libre propia del framework CakePhp (<http://book.cakephp.org/1.3/es/The-Manual/Common-Tasks-With-CakePHP/Internationalization-Localization.html>).

### 5.2 Factibilidad económica y financiera

Se tienen en cuenta los siguientes factores de análisis:

- Las librería de pago de DineroMail es de uso libre y gratuito. La entidad no cobra por su uso, el cargo viene por cada transacción realizada, como cualquier transacción que cobran las entidades dedicadas a este tipo de operaciones, por ejemplo en un pago, el impuesto que se cobra por la transacción está incluido dentro del detalle del pago que se realiza.
- El framework CakePhp con el cual esta desarrollado Maixon.com es libre y gratuito. Las nuevas funcionalidades serán desarrolladas usando el mismo framework.
- Los gastos que deben afrontar los dueños del sistema es por el desarrollo de los módulos, honorarios de programación y diseño.
- El beneficio que se espera obtener es significativo ya que se espera agilidad y facilidad en el cobro de abonos y apertura hacia nuevos mercados internacionales.

### 5.3 Factibilidad operativa

- El hecho de brindar la posibilidad a las inmobiliarias de realizar el pago de sus abonos a través de un servicio web, les aporta rapidez y evita pérdidas de tiempo en realizar pagos de manera personal. También a los dueños les permite llevar un registro exacto de ingresos por pagos de abonos, ya que luego de una operación de pago, la entidad DineroMail informa el resultado del mismo, si fue exitoso o no.
- El módulo Importador/Exportador facilitará el trabajo a las inmobiliarias para migrar los datos de sus propiedades desde sus sistemas actuales hacia Maixon como también generar migraciones hacia otros sistemas de manera automática, de esta manera los agentes de las inmobiliarias se acostumbrarán a mantener actualizados los datos de las propiedades sólo en el sistema Maixon.
- El módulo de Internacionalización y Localización permitirá la apertura de nuevos mercados internacionales y configurarlos desde un solo sistema, el sistema Admin, evitando tener que programar un portal por cada país.

### 5.4 Factibilidad legal

Luego de un análisis realizado en conjunto con los dueños de Maixon, no se detectan impedimentos legales para llevar a cabo el desarrollo de los módulos. Cabe mencionar que en el caso de los pagos, quien tienen acceso a las cuentas bancarias previa autorización de los representantes de las inmobiliarias son los servicios web de DineroMail, a Maixon sólo le llega el resultado de la transacción, pero nunca datos de tarjetas de créditos ni cuentas bancarias, para más información se puede consultar el siguiente enlace en donde se modela la transacción haciendo uso de la API de DineroMail <https://ar.dineromail.com/Ejemplos>.

### 5.5 Conclusión del Estudio de Factibilidad

En base a los análisis de factibilidad realizados, se concluye que no existen impedimentos técnicos, operativos, económico-financieros ni legales para los nuevos requerimientos del cliente, por lo tanto es viable el desarrollo de los módulos solicitados.

## 6. Metodologías

### 6.1 Introducción

Una metodología o proceso de desarrollo de software es el conjunto estructurado de actividades necesarias y resultados asociados que producen un producto software [1]. Estas actividades son:

- Especificación del software: en donde el cliente y los ingenieros definen el software a producir y las restricciones sobre su operación.
- Desarrollo del software: donde el software se diseña y se programa.
- Validación del software: donde el software se valida para asegurar que es lo que el cliente requiere.
- Evolución del software: donde el software se modifica para adaptarlo a los cambios requeridos por el cliente y el mercado.

### 6.2 Modelos de procesos de software

Un modelo de procesos es una representación del mundo real, que captura el estado de actual de las actividades para guiar, reforzar o automatizar partes de la producción de los procesos. [1]

*“El mejor proceso de software es el que está cerca de la gente que realizará el trabajo. Si un modelo de proceso de software ha sido desarrollado en un ámbito corporativo u organizacional, puede ser efectivo sólo si es en gran medida adaptable para satisfacer las necesidades del equipo del proyecto, que es el que en realidad lleva a cabo el trabajo de ingeniería de software.” [2]*

Existen dos tipos de modelos de procesos de software, los tradicionales o prescriptivos y los ágiles o no prescriptivos.

### 6.3 Modelos prescriptivos

También denominadas **Metodologías Tradicionales**, que surgieron antes de los años '90, pretendían ayudar a los profesionales indicando pautas a realizar y documentar cada una de las tareas del proceso de desarrollo de software.

Las principales características son:

- Requisitos fijados a lo largo de todo el proyecto.
- Basadas en procesos.
- Proyectos muy bien documentados.
- Gestión predictiva de los proyectos.

Entre ellas podemos mencionar:

### 6.3.1 Modelo en Cascada o Ciclo de vida clásico

Es un enfoque sistemático y secuencial hacia el desarrollo de software, que se inicia con la especificación de requerimientos del cliente, continúa con el diseño, le siguen la implementación y pruebas y por último la operación y mantenimiento.

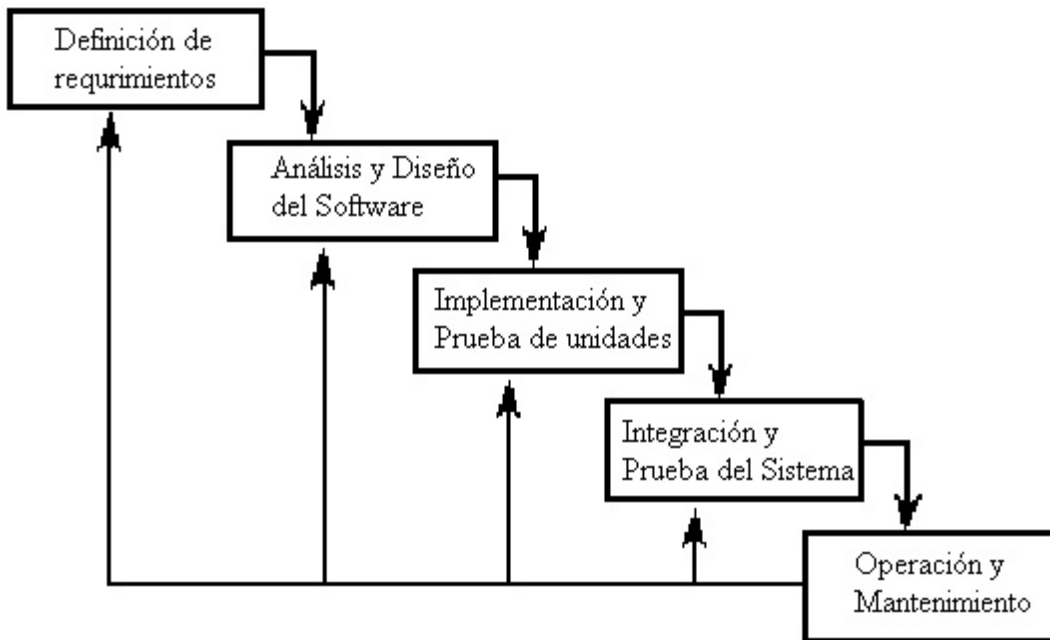


Figura 6.1 Modelo de ciclo de vida en Cascada [8]

Las desventajas de este modelo son:

- Es muy raro de que proyectos reales sigan un flujo secuencial. A pesar de que el modelo incluye iteraciones, lo hace de manera indirecta. Como resultado, los cambios confunden mientras el equipo de proyecto actúa.
- Con frecuencia es difícil para el cliente establecer todos los requisitos de manera explícita. El modelo en cascada lo requiere y enfrentan dificultades al incorporar la incertidumbre natural presente en el inicio de muchos proyectos.
- El cliente debe tener paciencia. Una versión que funcione de los programas estará disponible cuando el proyecto esté muy avanzado. Un error grave será desastroso si no se detecta antes de la reversión del programa.

En la actualidad, el trabajo de desarrollar software es acelerado y sujeto a una cadena infinita de cambios (de características, funciones y contenido de información). Con frecuencia, el modelo en cascada no es apropiado para dicho trabajo. Sin embargo, suele servir como modelo de proceso en situaciones donde los requerimientos son fijos y el trabajo se realiza en forma lineal.



### 6.3.2 Modelo de Prototipos

Suele ser útil en situaciones en donde el cliente define un conjunto de objetivos generales para el software, pero no define los requisitos detallados de entrada, procesamiento y salida; o en otros casos, el responsable de desarrollo del software no está seguro de la eficacia de un algoritmo, de la adaptabilidad de un sistema operativo o de la forma que debería tomar la interacción *hombre-máquina*. En estas situaciones, un paradigma de construcción de prototipos puede ofrecer un mejor enfoque que ayude al cliente y al ingeniero de sistemas a entender de mejor manera cuál será el resultado de la construcción cuando los requisitos estén satisfechos.

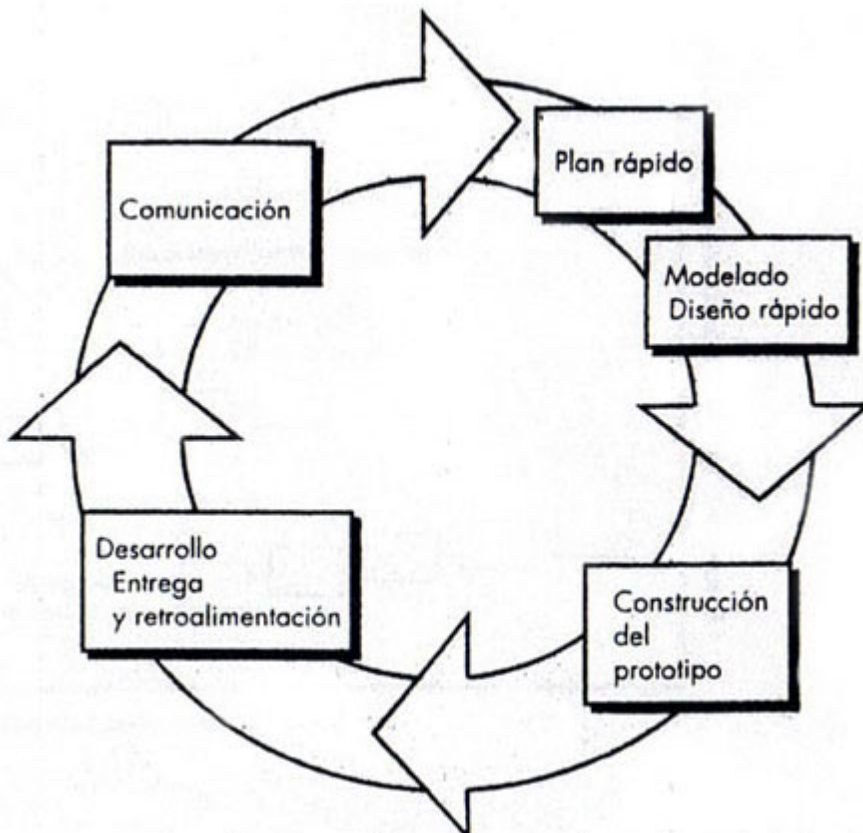


Figura 6.2 Modelo de Prototipos [2]

El paradigma de construcción de prototipos se inicia con la comunicación. El ingeniero de software y el cliente encuentran y definen los objetivos globales para el software. Se plantea con rapidez la iteración de construcción de prototipos y se presenta el modelado en la forma de un diseño rápido. El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles para el cliente o usuario final. El diseño rápido conduce a la construcción de un prototipo, luego al prototipo lo valida el usuario o cliente y con la retroalimentación se refinan los requisitos del software que se desarrollará.

El objetivo de un prototipo es servir como mecanismo para identificar requisitos.

Las desventajas de usar prototipos son:

- El cliente percibe al prototipo como una versión en funcionamiento del software

cuando en realidad en la construcción del prototipo no se ha tenido en cuenta la calidad del software global o factibilidad de mantenimiento a largo plazo.

- En ocasiones se crean prototipos en un sistema operativo o lenguaje de programación inadecuado.

### 6.3.3 Modelo en espiral

Es un modelo de proceso de software evolutivo que conjuga la naturaleza iterativa de la construcción de prototipos con los aspectos controlados y sistemáticos del modelo en cascada. Proporciona el material para el desarrollo rápido de versiones incrementales del software.

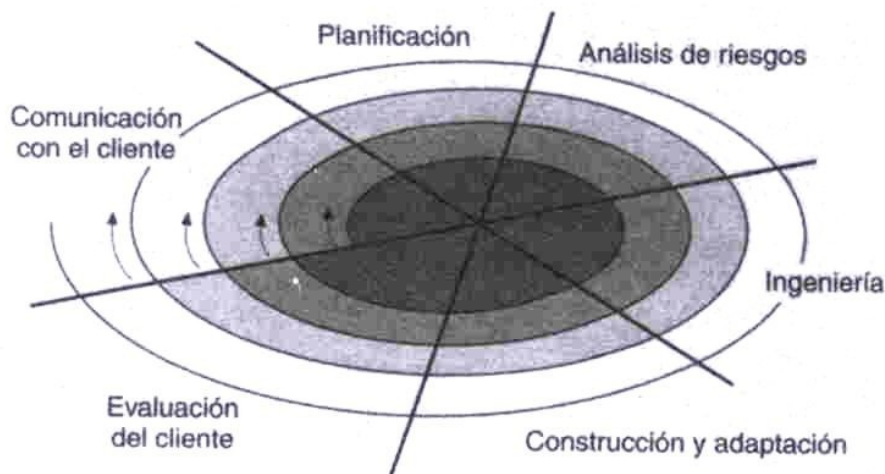


Figura 6.3 Modelo de desarrollo en espiral [2]

El primer circuito alrededor del espiral quizá genere el desarrollo de una especificación del producto; los pasos siguientes alrededor del espiral se pueden aprovechar para desarrollar un prototipo y después en forma progresiva, versiones más elaboradas del software. Cada paso a través de la región de planificación resulta en ajustes al plan de proyecto. Los costos e itinerario se ajustan en base a la retroalimentación derivada de la relación con el cliente después de la entrega. El administrador del proyecto ajusta el número de iteraciones planeado que se requiere para completar el software.

Desventajas del modelo en espiral:

- Es difícil convencer a clientes, en particular en situaciones bajo contrato, de que el enfoque evolutivo es controlable ya que se requiere una habilidad considerable para evaluar riesgos y se confía en dicha habilidad para obtener el éxito. Si un riesgo importante no se descubre y administra, sin duda surgirán problemas.

### 6.3.4 El proceso unificado

Proceso Unificado de Desarrollo es un proceso de desarrollo guiado por casos de uso, centrado en su arquitectura, iterativo e incremental. Reconoce la importancia de comunicación con el cliente, enfatiza el papel de la arquitectura del software enfocándose en metas correctas como el entendimiento, el ajustes a cambios a futuro y la reusabilidad. Sugiere un flujo de proceso iterativo e incremental y que proporciona el sentido evolutivo esencial del desarrollo de software moderno.

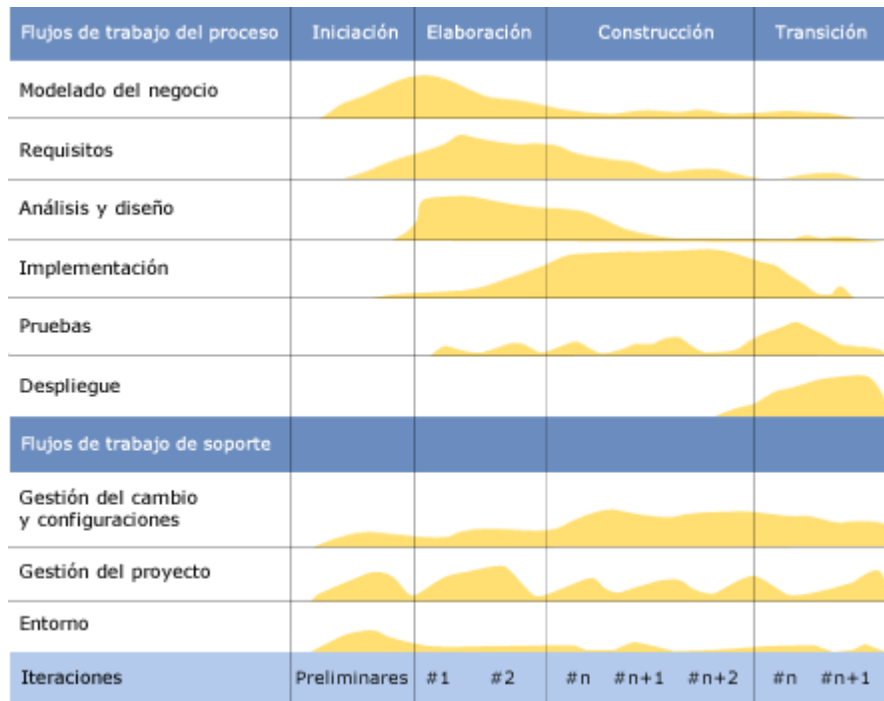


Figura 6.4 Fases de trabajo del Proceso Unificado [9]

Como cualquier modelo de proceso, el proceso unificado reconoce cinco actividades. También incorpora cuatro fases: inicio, elaboración, construcción y transición.

La fase de inicio abarca la comunicación con el cliente para determinar los requisitos y la planificación, elaboración de un plan de naturaleza iterativa e incremental. Los requisitos fundamentales se describen a través de casos de uso. Un caso de uso describe una secuencia de acciones que desarrolla un actor cuando interactúa con el software.

La fase de elaboración abarca las actividades de comunicación con el cliente y actividades de modelado genérico del proceso. Refinación y expansión de los casos de uso. La arquitectura ahora se compone de: modelo de casos de uso, modelo de análisis, modelo de diseño, modelo de implementación y modelo de despliegue.

La fase de construcción desarrolla o adquiere los componentes de software que harán que cada caso de uso sea operativo para usuarios finales, o sea, construir el software en código fuente.

En la fase de transición el software se entrega a los usuarios finales para realizar pruebas beta y para la retroalimentación que ellos reportan tanto por defectos como nuevos requisitos. Al final de esta fase se produce el incremento del software y se obtiene una nueva versión.

Es probable que mientras se realiza alguna fase ya se hayan realizado otros trabajos del siguiente incremento del software. Esto quiere decir que las fases del proceso unificado no son secuenciales sino que puede ocurrir concurrencia por etapas.

## 6.4 Modelos no prescriptivos

También conocidas como **Metodologías Ágiles** o Modernas. Son metodologías posteriores a los años '90, surgen como una alternativa, una reacción a las metodologías tradicionales y principalmente a su burocracia.

En 2001, Kent Beck y otros 16 notables desarrolladores, escritores y consultores se reunieron para definir los métodos que estaban surgiendo como alternativa a las metodologías formales, consideradas excesivamente “pesadas” y rígidas por su carácter normativo y su fuerte dependencia de planificaciones detalladas, previas al desarrollo. Formaron lo que hoy se conoce como la “**Agile Alliance**” (la Alianza Ágil) (<http://www.agilealliance.org/>) que es una organización sin fines de lucro con membresía global comprometida con el avance de los principios y prácticas de desarrollo ágil. Apoya a explorar y aplicar los principios y prácticas ágiles para que la industria del software más productivo, humano y sostenible [3]. Comparten su pasión por ofrecer software mejor día a día.

Los principios y prácticas propuestas por este grupo de personas forman el “**Manifiesto para el desarrollo ágil de software**” [4], que básicamente son los principios sobre los cuales se basan los métodos conocidos como ágiles, este manifiesto establece:

*Hemos descubierto mejores formas de desarrollar software al construirlo por nuestra cuenta y ayudar a otros a hacerlo. Por medio de este trabajo hemos llegado a valorar:*

*A los **individuos y sus interacciones** sobre los procesos y herramientas.*

*Al **software en funcionamiento** sobre la documentación extensa.*

*A la **colaboración del cliente** sobre la negociación del contrato.*

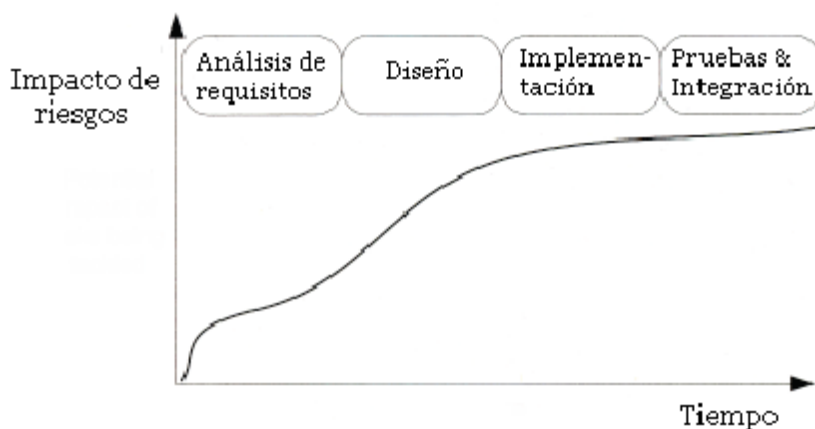
*A la **respuesta al cambio** sobre un seguimiento de un plan.*

Como se mencionó anteriormente, las metodologías ágiles surgen como una reacción a las metodologías tradicionales, los factores más importantes que promovieron esa reacción fueron:

- La pesadez y lentitud de reacción, exceso de documentación y falta de agilidad de los procesos tradicionales.
- Las metodologías existentes no cumplieron las expectativas planteadas inicialmente.
- Explosión de la red y las aplicaciones Web.
- Movimiento open source.

Las metodologías tradicionales exigen un esfuerzo extra por parte del equipo de desarrollo en fases poco productivas, como es la documentación y debido a su estructuración (típicamente

siguiendo el modelo en Cascada o más actualmente el Proceso Unificado) son poco flexibles a los cambios, de requisitos, de nuevas funcionalidades, etc. Otra dificultad añadida al uso de metodologías convencionales es la lentitud del proceso de desarrollo. En ocasiones, es difícil para los desarrolladores comprender un sistema complejo en su totalidad lo que provoca que las diferentes etapas del ciclo de vida convencional se ejecuten lentamente. Dividir el trabajo en módulos pequeños ayuda a minimizar los fallos y, por tanto, el costo de desarrollo. Además, permite liberar versiones del sistema progresivamente. Actualmente en el mercado del software, en donde los productos quedan obsoletos rápidamente, se pide básicamente rapidez, calidad y reducción de costos, pero para asumir estos retos, es necesario tener agilidad y flexibilidad.



**Figura 6.5** Distribución del riesgo en un desarrollo en Cascada.

Al mismo tiempo, las metodologías convencionales tienden a acumular los riesgos y dificultades que surgen en el desarrollo del producto al final del proyecto, como puede apreciarse en la figura 5.5, repercutiendo en retrasos en la entrega de productos o influyendo en la incorrecta ejecución de las últimas fases del ciclo de vida.

En cambio, las metodologías ágiles se caracterizan por incluir un desarrollo iterativo y que están centradas en la interacción, comunicación, y en la reducción de la creación de artefactos intermedios. Desarrollar mediante iteraciones permite al equipo de desarrollo adaptarse a los cambios de los requisitos. Trabajar con un alto grado de comunicación permite que el equipo pueda tomar decisiones y aplicarlas inmediatamente. La reducción de artefactos intermedios que no dan valor añadido al producto final, nos permite asignar más recursos al producto y posibilitar su pronta finalización. Las metodologías ágiles se adaptan al entorno, son apropiadas cuando los requisitos son emergentes y cambian rápidamente. Tienen la capacidad de respuesta a cambios a lo largo del proceso de desarrollo, mejoran el sistema e incrementa la satisfacción del cliente. La entrega continua en plazos breves del software funcional permite al cliente verificar *in situ* el desarrollo del proyecto. El desarrollo en ciclos cortos permite que se reduzcan los riesgos y dificultades de mayor a menor a lo largo del tiempo como muestra la figura 6.6.

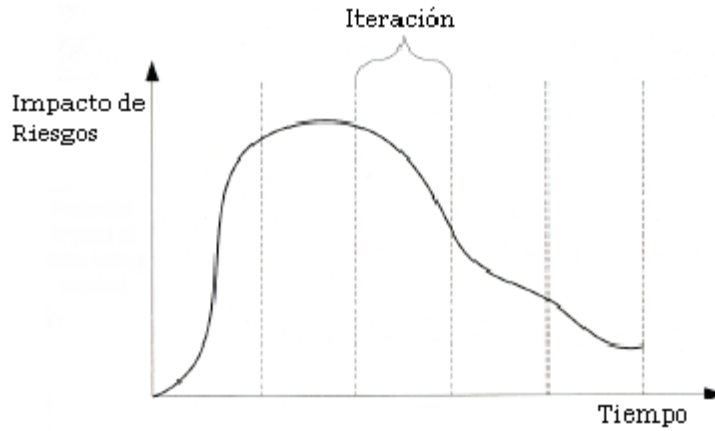


Figura 6.6 Distribución del riesgo en un desarrollo ágil

El trabajo en conjunto con el cliente y el equipo de desarrollo hace que se establezca una comunicación directa y se reduzcan los malos entendidos.

### 6.4.1 eXtreme Programming

eXtreme Programming o XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo software. Oficialmente fue creada por Kent Beck en 1999 con la publicación de su libro “Extreme Programming Explained” [7]. Se centra en la continua retroalimentación entre el cliente y el equipo de desarrollo, la comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Se define como una metodología especialmente adecuada para proyectos con requisitos muy cambiantes e imprecisos, donde existe un alto riesgo técnico.

El ciclo de vida de XP se representa en la figura 6.7.

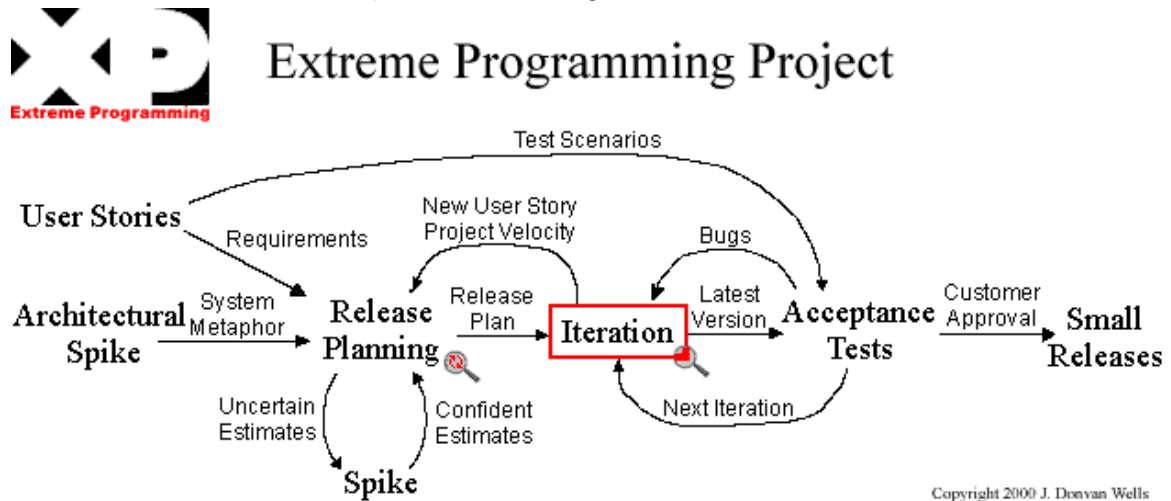


Figura 6.7 Ciclo de vida de XP [10]

**Fase de exploración:** En esta fase los clientes plantean las historias de usuario que requieren que sean incluidas en la primera versión del producto. Cada una de las historias de usuario describen una funcionalidad que será añadida al programa. En este tiempo el equipo del

proyecto se familiariza con las tecnologías y herramientas que se usarán a lo largo del proyecto.

**Fase de planificación:** En esta fase se establece la prioridad de las diferentes historias de usuario y se acuerda el contenido de la primera entrega del proyecto. La estimación temporal se basa en un cálculo estimado por parte de los desarrolladores de cada una de las historias, la primera entrega no suele tardar más de dos meses en realizarse. La duración de esta fase no suele exceder el plazo de unos pocos días.

**Fase de iteraciones:** Esta fase incluye la realización de diferentes fases antes de liberar la primera versión del producto. La planificación realizada en la etapa anterior se divide en diferentes iteraciones, de una duración variable entre una semana y cuatro. Los usuarios son los que deciden que historias se van a realizar en cada iteración, sabiendo que en la primera se suele realizar un sistema con la arquitectura de todo el sistema, seleccionando aquellas historias que ayuden a construirla. Las pruebas funcionales creadas por el cliente son ejecutadas al final de cada iteración, de tal manera que al final de esta fase obtenemos una versión lista para producción.

**Fase de producción:** En esta fase se llevan a cabo un conjunto de pruebas extras, de rendimiento y funcionamiento que son necesarias antes de poder entregar el producto al cliente. Si se encuentran cambios importantes que se deban realizar al producto, se debe decidir si incluirlos en esta versión o dejarlos para posteriores. Las iteraciones de esta fase no deben durar más de tres semanas.

**Fase de mantenimiento:** Una vez se ha liberada la primera versión a los usuarios, el proyecto se debe mantener en el entorno de producción siempre y cuando aún hayan iteraciones en fase de producción. Esto supone un esfuerzo considerable en la fase de mantenimiento e incluso se sugiere la contratación de nuevo personal para dar soporte a los clientes e incluso cambiar la estructura del equipo.

**Fase de cierre del proyecto:** Es la fase en que los clientes ya no tienen más historias que deban ser implementadas. En esta fase se debería considerar que se cubren todas las necesidades de los clientes y otros aspectos como fiabilidad, rendimiento, etc. La documentación del proyecto se realiza en esta fase, ya que ni la arquitectura, ni el diseño, ni el código sufrirán cambio alguno.

Roles y responsabilidades:

**Cliente:** Encargado de describir las historias de usuario y las pruebas funcionales, decide cuando un requisito está satisfecho. Establece las prioridades de las funcionalidades.

**Programador:** escribe el código fuente de la aplicación y realiza las pruebas unitarias.

**Encargado de pruebas o Tester:** Ayuda al cliente a escribir las pruebas funcionales para validar su implementación, transmite los resultados de las pruebas.

**Encargado de seguimiento o Tracker:** Proporciona retroalimentación al equipo, realiza un seguimiento de las estimaciones e informa de las desviaciones. Realiza el seguimiento del progreso de cada iteración.

**Entrenador o Coach:** Responsable del progreso global. Provee las guías a equipo de forma que se aplique el modelo XP de manera correcta.

**Consultor:** Miembro externo de equipo con conocimiento específico de algún tema necesario para el proyecto.

**Manager:** Es el intermediario entre clientes y programadores.

Herramientas adicionales:

**Historias de usuario:** Son los requisitos del sistema formulados en una o dos sentencias, en el lenguaje común del cliente.

**Definición de Hecho:** junto a cada Historia de Usuario, las definiciones de hecho determinan cuando las mismas han sido testeadas, integradas y documentadas.

**Juego de Planificación:** El equipo de desarrolladores estiman el esfuerzo necesario para implementar las historias y los clientes determinan los objetivos y tiempos de entrega.

**Metáforas:** El sistema se define utilizando un conjunto de metáforas acordadas entre el cliente y los programadores. Esta historia compartida guiará todo el proceso describiendo cómo funciona el sistema.

**Entregas pequeñas, cortas iteraciones:** Producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad del sistema. Esta versión ya constituye un resultado de valor para el negocio.

**Pruebas:** El desarrollo del software es orientado a pruebas (Test driven development) Las pruebas unitarias se escriben antes que el código y están funcionando continuamente. Las pruebas funcionales las escriben los clientes.

**Refactorizar:** Reestructurar el sistema eliminando duplicaciones, mejorando la comunicación, simplificando y añadiendo flexibilidad.

**Programación por pares:** Es la técnica que promulga que dos personas escriban código en el mismo ordenador.

**Propiedad colectiva:** Cualquiera puede compartir cualquier parte de código con cualquier otro componente del equipo.

**Integración continua:** Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día.

**40 horas por semana.** Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre, probablemente está ocurriendo un problema que debe ocurrir y corregirse. El trabajo extra desmotiva al equipo.

**Disponibilidad del cliente:** Los clientes deben estar disponibles y presentes cuando sean



requeridos por el equipo de desarrollo.

**Estándares de codificación.** Reglas de codificación son establecidas y seguidas por los programadores. Se enfatiza la comunicación a través del código.

### **6.4.2 SCRUM**

En 1986 Ikujiro Nonaka e Irotaka Takeuchi presentaron el artículo “The New Product Development Game” [5], en el cual presentaban un proceso adaptativo, rápido y auto-organizado de desarrollo de productos tecnológicos. El término Scrum hace referencia a la estrategia utilizada en el deporte Rugby en la que todos los integrantes del equipo actúan juntos para avanzar la pelota y ganar el partido. En 1996 Sutherland presentó junto con Ken Schwaber las prácticas que empleaba como procesos formal para la gestión del desarrollo de software y lo denominaron Scrum.

Scrum es un simple marco de trabajo iterativo e incremental para la colaboración en equipo eficaz en proyectos complejos. Proporciona un conjunto de reglas que crean suficiente estructura para que los equipos puedan centrar su innovación en la solución de lo que podría ser un desafío insuperable [6]. Define métodos de gestión y control para complementar la aplicación de otros métodos ágiles como XP, que centrados en prácticas de tipo técnico, carecen de ellas.

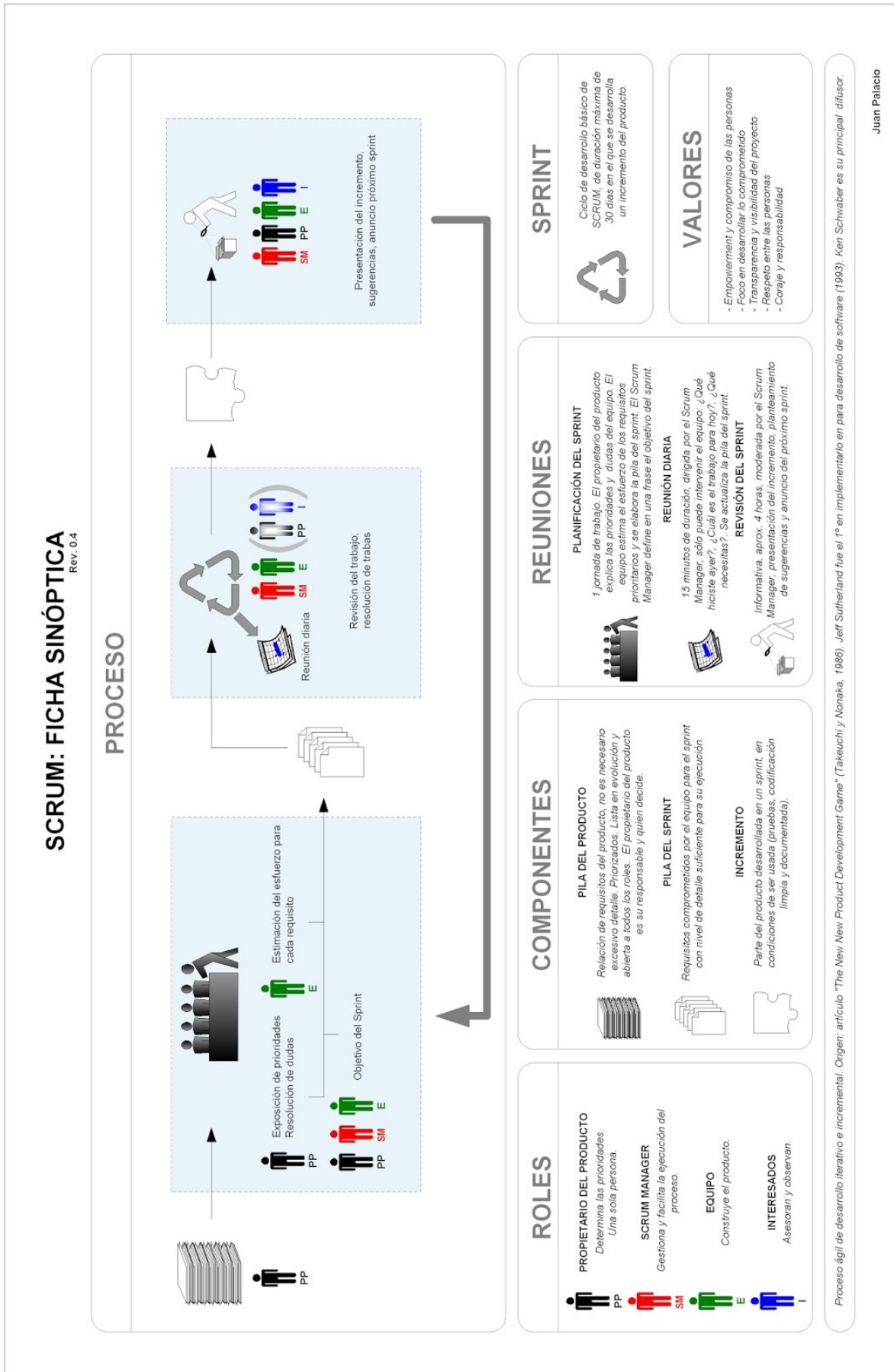


Figura 6.8 Scrum ficha sinóptica del proceso, herramientas y roles [11]

Al principio del proyecto se define el **Product Backlog** (Pila del Producto), que contiene todos los requerimientos funcionales y no funcionales que deberá satisfacer el sistema a construir. Los mismos estarán especificados de acuerdo a las convenciones del equipo de trabajo ya sea mediante: tareas, casos de uso, diagramas de flujo de datos, tareas, etc. El Product Backlog será definido durante reuniones de planeamiento con los stakeholders (cliente).

**Planificación del Sprint:** se definirán las iteraciones, conocidas como Sprint en las que se irá evolucionando el producto. Cada Sprint tendrá su propio Sprint Backlog (Pila del sprint) que será un subconjunto de tareas del Product Backlog con los requerimientos a ser construidos en el Sprint correspondiente. Las tareas están asignadas a personas y tienen estimados el tiempo y los recursos necesarios. La duración recomendada del Sprint es de una a cuatro semanas.

**Seguimiento del Sprint:** dentro de cada Sprint el Scrum Master (equivalente al Líder de Proyecto) llevará a cabo la gestión de la iteración, convocando diariamente al Scrum Daily Meeting que representa una reunión de avance diaria de no más de 15 minutos con el propósito de tener realimentación sobre las tareas de los recursos y los obstáculos que se presentan. Aquí se deben responder 3 preguntas:

- ¿Qué trabajo se ha realizado desde la reunión anterior ?
- ¿Qué trabajo que se va a hacer hasta la próxima reunión?
- ¿Existen impedimentos que deben solventarse para proseguir con el trabajo?

**Revisión del Sprint:** Al final de cada Sprint, se realizará un Sprint Review, se realiza un análisis y revisión del incremento generado, evalúan los artefactos construidos y comentar el planeamiento del próximo Sprint. Se recomienda siempre tener preparada una demo para mejorar la retroalimentación con los stakeholders, reconocimiento del trabajo.

Roles y responsabilidades:

**Product Owner:** Es la única persona del proyecto conocedora del entorno de negocio del cliente y de la visión del producto y es el responsable de obtener el resultado de mayor valor posible para el cliente.

**Scrum Master:** Es el encargado de garantizar el funcionamiento de los procesos y de la metodología. Es importante darse cuenta que Scrum master es más que un rol, es la responsabilidad de funcionamiento de modelo, por tanto muchas veces es aconsejable utilizar a personas y puestos más adecuados según la organización. Un Scrum Master debe interactuar tanto con el equipo como con el cliente y con los gestores.

**Equipo de desarrollo:** Es el equipo del proyecto y tiene la autoridad para decidir en las acciones necesarias y para auto-organizarse con la finalidad de alcanzar los objetivos del sprint. Está involucrado en la estimación del esfuerzo de las tareas del product backlog, en la creación del sprint backlog, etc.

**Stakeholders:** es el Cliente que participa en la creación del Product Backlog y en la revisión de los Sprints.

**Manager:** Está a cargo de la toma de decisiones finales, participa en la elección de

objetivos y requisitos. Es el encargado de seleccionar al Product Owner o en la reducción del Product Backlog junto con el Scrum Master.

Herramientas adicionales:

**Historias de Usuario:** Herramienta textual que describe clara y concisamente la funcionalidad detallada de cada elemento del Product Backlog.

**Estimación de esfuerzo:** Es un proceso iterativo en el cual las estimaciones de las tareas del Product Backlog son reajustadas acorde a la información obtenida en la última iteración. Este reajuste lo llevan a cabo el equipo de desarrollo y el Product Owner. El tamaño de las Historias de Usuario se miden por **Puntos Historia**, que son una estimación del monto de trabajo en equipo necesario para completar una tarea respecto de otra, o sea se toma una tarea como referencia para compararla con las demás.

**Burn-down Chart:** Es una herramienta para gestionar y seguir el trabajo de cada sprint y representa gráficamente el avance del sprint. En la figura 6.9 se puede observar en el eje horizontal una escala temporal que representa los días de duración del sprint y en el eje vertical otra escala que indica los puntos de historia o estimación de esfuerzo. De esta manera en las reuniones diarias el Scrum Master puede evaluar cuánto se ha avanzado de acuerdo a lo planificado los incrementos que se obtienen.

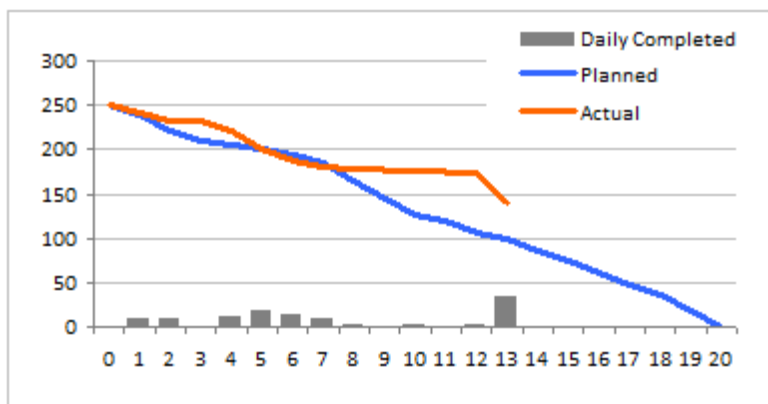


Figura 6.9 Burn-down Chart

**Burn-up Chart:** Herramienta de gestión y seguimiento que sirve al Product Owner para controlar las versiones de producto previstas, las funcionalidades de cada una, la velocidad estimada, fechas probables de cada versión, margen de error previsto en las estimaciones y avance real.

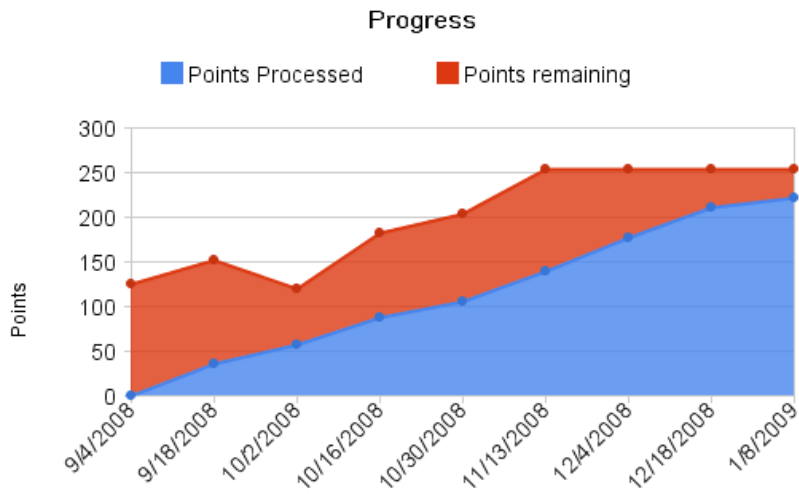


Figura 6.10 Burn-up Chart

**Planning Poker o estimación de Poker:** Juego que ayuda al equipo a establecer una estimación de las tareas en la reunión de inicio de sprint.

## 7 - Aplicación Práctica de SCRUM

Se propuso a los dueños del sistema Maixon aplicar la metodología de desarrollo SCRUM para las próximas tareas a desarrollarse.

Se interiorizo brevemente en qué consiste la metodología, las similitudes y diferencias con la manera con la cual veníamos trabajando hasta ahora y se establecen los Roles y Responsabilidades de cada persona.

### 7.1 Roles y Responsabilidades

Se definen los siguientes:

- Product Owner [PO]: Dueños del sistema Maixon, Leandro Borges y Norberto Riccitelli. Definirán el Product Backlog y realizarán tareas de Testing Pruebas de Unidad.
- Scrum Master [SM] : Dante Cáceres. Planificación de Sprints, Seguimiento, Revisión y Retrospectiva.
- Team:
  - Programador: Dante Cáceres.
  - Diseñador: Mauricio Andrada.

Realizarán las tareas relacionadas con los elementos del Sprint Backlog, registrarán un Scrum Daily sobre lo trabajado en el día.

- Stakeholders: Usuarios de inmobiliarias, clientes finales que utilizan el sistema y en oportunidades reportan errores.

### 7.2 Product Backlog Inicial

Existe una reunión con el Product Owner en donde se establecen las próximas funcionalidades que se requieren para el sistema y se pone en conocimiento la metodología experimental con la cual se va a trabajar en los próximos meses.

En esta etapa se ha introducido al Product Owner a la forma de trabajar que indica la metodología Scrum, debido a que en etapas anteriores de trabajo, solían producirse interrupciones en la ejecución de tareas y esto generaba entregas parciales no testeadas completamente. Lo que indica Scrum es que una vez iniciado un Sprint, no deben surgir interrupciones y el equipo debe trabajar según las estimaciones y realizar las entregas al Product Owner al finalizar el mismo. Se hizo conocer al Product Owner de la necesidad de cumplir con esta forma de trabajo.

También se asesoró en la manera de dividir los módulos en elementos para el Product Backlog ya que si bien el Product Owner tenía conocimiento de lo que quería para el sistema, era necesario dividir y organizar los módulos en tareas más pequeñas y tangentes a la hora de realizar testing o visualizar su funcionalidad. Por ejemplo para el módulo de pago automático, el Product Owner relataba su requerimiento:

*“Quiero que las Inmobiliarias puedan realizar el pago de manera automática usando una API de pago de DineroMail, que el mismo impacte en nuestro sistema y que se actualice al instante su abono en el sistema Manager de Maixon”*

Este requerimiento no puede incluirse como elemento del Product Backlog porque es muy general, contiene mucha funcionalidad implícita dentro. Lo que se realizó fue un asesoramiento sobre casos de éxito y excepciones. Se realizaron preguntas para ir desmenuzando el requerimiento en tareas o elementos que pudieran ser realizados y testeados para medir su grado de avance. Preguntas como:

- *¿Cuáles roles de usuarios de las Inmobiliaria estarán autorizados a realizar el pago de abono?*
- *¿Qué pasa si la API devuelve un error porque el pago no fue exitoso?*
- *Nos enfrentamos a un Riesgo (entendiendo por Riesgo el grado de incertidumbre de algo) de aprendizaje de la API de pago ¿Está dispuesto a invertir tiempo y dinero en el aprendizaje de la misma?*
- *Como usuario administrador del sistema, Usted debería poder contar con un panel en el cual se resume el estado de los pagos de abonos de las Inmobiliarias y de cuánto dinero ha obtenido mensualmente por estos pagos. ¿Le interesa que le desarrollemos esta funcionalidad?*
- *Según un simple análisis de la documentación de la API de DineroMail, existen dos posibilidades de pago: Tarjeta de Crédito y generar un Cupón de Pago con Código de Barras. ¿Entiende que es necesario el desarrollo de un proceso automático Cron que se ejecute y recupere de la API el estado de los pagos de Cupones y de acuerdo a ello actualice el estado de abono de las inmobiliarias?*

También se puede mencionar el asesoramiento con el requerimiento que luego dió vida al sistema Maixon Pages, el requerimiento del Product Owner era:

*“Debemos ofrecer a las inmobiliarias una web personalizada para cada una. Mediante un panel donde ingresen un dominio, por ejemplo <http://www.inmobiliariagalas.com>, que Maixon dé de alta el dominio y lo apunte a nuestros DNS. Luego que cuando se acceda a ese dominio, cada web debe recuperar mostrar las propiedades de esa Inmobiliaria según corresponda. Este servicio solo debe ser para Inmobiliarias del Plan Premium”*

Se orientó entre lo factible y no factible de su requerimiento y sobre la mejor manera de realizarlo:

- *No estoy seguro de que exista algún mecanismo automático de registro de dominio por lo que considero que esa tarea debe realizarse manualmente por alguna persona de la inmobiliaria o de Maixon e indicar al momento de registro de dominio a cuáles DNS debe apuntar el dominio.*
- *Recomiendo que en vez de desarrollar para cada inmobiliaria una web personal, mejor desarrollamos un sistema que se comporte de manera apropiada según el dominio por el cual se accede. O sea que cada inmobiliaria registre su dominio y nos indique cuál es ese dominio. Luego si ingresan por: <http://www.inmobiliariauno.com.ar> se muestren y recuperen las propiedades de Inmobiliaria Uno, si ingresan por <http://www.inmobiliariados.com.ar> se recuperen y muestren las propiedades de Inmobiliaria Dos. (De esta manera nace la idea del sistema Maixon Pages).*
- *Podemos ofrecer a las inmobiliarias que seleccione uno de un conjunto de templates que*

desea para su Page.

- ¿Que debería mostrar el sistema Maixon Pages para las inmobiliarias del plan Premium que no realizaron los pagos de abonos?

De esta manera se fué asesorando al Product Owner y elaborando en conjunto los elementos que son medibles de realizar. El Product Backlog quedó definido de la siguiente manera y se tienen en cuenta categorías para agrupar los elementos, o mejor dicho subsistemas:

- Maixon Pages
  1. Para todas las Inmobiliaria de plan Premium, permitir la configuración de su Page desde el sistema Manager
  2. Funcionamiento de acceso público al sistema Maixon Page de cada Inmobiliaria
  3. Recuperar y mostrar propiedades de Maixon en las Pages de cada Inmobiliaria
- Módulo Payments
  4. Sistema Manager permitir la posibilidad de contratar un abono de cualquier plan Premium o Advanced, realizar el pago a través de DineroMail y registrar el mismo
  5. Verificar los pagos realizados por las inmobiliarias y actualizar los periodos de abono de las mismas
- Módulo Internacionalización y Localización
  6. Desde Sistema Admin poder habilitar un nuevo país, seleccionado idioma.
  7. Para cada país habilitado, poder editar el contenido visual
- Módulo Importador Exportador.
  8. Desde sistema Admin poder importar propiedades de una Inmobiliaria desde Argenprop hacia Maixon
  9. Exportar propiedades a través de API de Zonaprop
  10. Exportar propiedades a través de feeds xmls para : Trovit
  11. Exportar propiedades a través de feeds xmls para : Olx
  12. Exportar propiedades a través de feeds xmls para : Reporte Inmobiliario

### 7.3 Determinación de Duración de Sprint y Velocidad

El equipo trabaja en Maixon cuatro horas diarias.

Anteriormente se dijo que un Punto Historia es una estimación del monto de trabajo en equipo necesario para completar una Historia de Usuario comparándola con otra.

Para determinar la velocidad se tomó como Punto Historia de referencia un ABML (alta, baja, modificación y listado) de una tabla relacionada Padre-Hijo, por ejemplo un ABML de País-Provincia que al equipo actual le lleva 4hs de trabajo, o sea **1 Punto Historia = 4 hs de trabajo**. Hay que aclarar que es solo el Seniority del equipo de desarrollo actual y en consideración framework Cakephp usado. Si no se usara un framework o el seniority fuese inferior, podría tomarse como 1 Punto Historia = 8hs de trabajo.



Entre el Product Owner y Scrum Master definen que la **duración del Sprint sea de 4 semanas o 20 días laborales.**

Tendiendo estos factores claros decimos que la **velocidad de nuestro equipo es de 20 puntos historia por Sprint.**

## 7.4 Estimación del funcionalidades del Product Backlog

Primero se solicita al Product Owner que determine las prioridades de los elementos del Product Backlog, asignando un valor numérico . A mayor valor, más prioridad tiene el elemento.

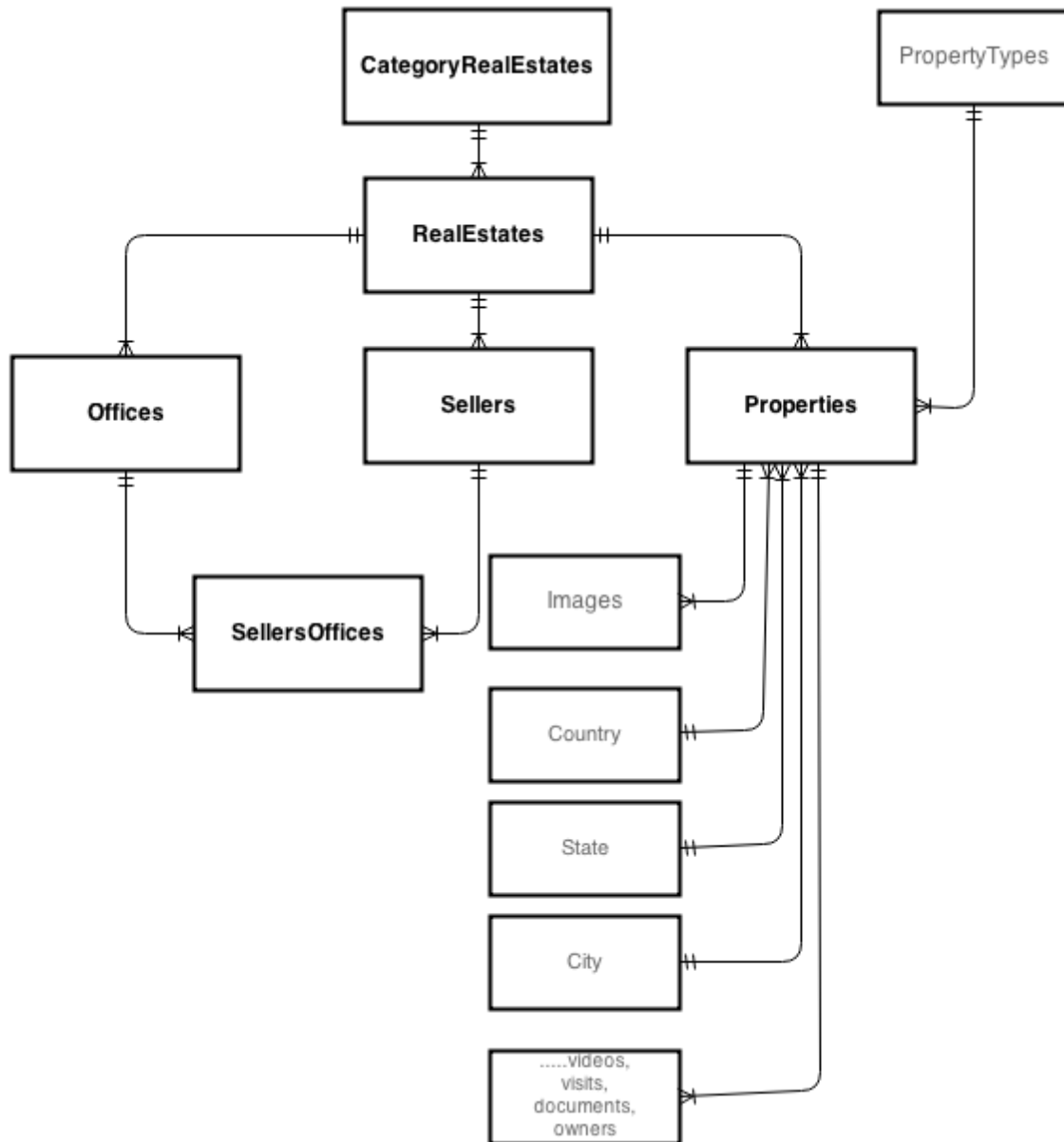
Luego de esto, el equipo: Programador y Diseñador se reúnen para realizar las estimaciones de las funcionalidades del Product Backlog y se asignan valores a los elementos.

ID	Categoría	Nombre	Prioridad	Estimación Inicial	Sprint	Como probarlo	Requisitos
1	Sistema Maixon Pages	Para todas las Inmobiliaria de plan Premium, permitir la configuración de su Page desde el sistema Manager	10	8	0	en sistema Manager, para todas las inmobiliarias plan Premium	dar de alta que es imp API)
2	Sistema Maixon Pages	Funcionamiento de acceso público al sistema Maixon Page de cada Inmobiliaria	10	6	0	visualizar a partir de URL configurada por cada inmobiliaria	
3	Sistema Maixon Pages	Recuperar y mostrar propiedades de Maixon en las Pages de cada Inmobiliaria	10	6	0	Verificar urls maixon con urls de pages	
3	Módulo Payments	Sistema Manager permitir la posibilidad de contratar un abono de cualquier plan Premium o Advanced, realizar el pago a través de DineroMail y registrar el mismo	5	5	2	en entorno DEV se debe realizar un pago...(falta definir el proceso)	Tiempo de DineroMail
4	Módulo Payments	Verificar los pagos realizados por las inmobiliarias y actualizar los periodos de abono de las mismas	5	5	2	consultar con API	Tiempo de DineroMail
5	Módulo Internacionalización y Localización	Desde Sistema Admin poder habilitar un nuevo país, seleccionado idioma.	3	6		Desde sistema Admin realizar el proceso	Creación a subdominio language P
6	Módulo Internacionalización y Localización	Para cada pais habilitado, poder editar el contenido visual	3	4			
7	Módulo Importador Exportador	Desde sistema Admin poder importar propiedades de una Inmobiliaria desde Argenprop hacia Maixon	8	4	1	Desde sistema Admin seleccionar Inmobiliaria y realizar proceso de importacion	Encontrar i Parser
8	Módulo Importador Exportador	Exportar propiedades a través de API de Zonaprop	9	15	1	realizando publicaciones a traves de API	Conocimien Zonaprop
9	Módulo Importador Exportador	Exportar propiedades a través de feeds xmls para : Trovit	8	2	1	Contactar con Sistemas Trovit para consultar el resultado de exportacion	
10	Módulo Importador Exportador	Exportar propiedades a través de feeds xmls para : Olx	8	1	1	Contactar con Sistemas Olx para consultar el resultado de exportacion	
11	Módulo Importador Exportador	Exportar propiedades a través de feeds xmls para : Reporte Inmobiliario	8	1	1	Contactar con Sistemas Reporte Inmobiliario para consultar el resultado de exportacion	
12	Módulo Importador Exportador	Exportar propiedades a través de API de Argenprop	8	15	2	realizando publicaciones a traves de API	Se usa VP

Figura 7.1 - Product Backlog inicial con estimaciones de prioridad y esfuerzo

## 7.5 Estado actual del sistema

Al momento de aplicar la metodología SCRUM, el sistema Maixon se encontraba con el siguiente modelo de tablas, si bien existen muchas más, se destacan las principales para entender conceptualmente su funcionamiento.



**Figura 7.2** - Estado del modelo de bases de datos del sistema Maixon antes de aplicar SCRUM

Básicamente, las Inmobiliarias (**RealEstates**) se registran en el sistema a través de un usuario (**Sellers**) administrador y por default se le asigna una Oficina principal (**Offices**). Luego de validar la cuenta, desde su panel del sistema Manager, el usuario administrador puede crear otros usuarios y asignarlos a oficinas. Todos los usuarios pueden publicar Propiedades (**Properties**) de diferentes tipos (**PropertyTypes**). A las propiedades se les pueden adjuntar imágenes (**Images**) y diferentes tipos de datos como Videos, Documentos, posicionarla en una ubicación para una fácil búsqueda (**Country**, **State**, **City**).

Se adjunta en el CD, en el directorio /codeProductBacklog, el código fuente controlador **Properties**, modelo **Properties** y una vista de carga de Propiedades que corresponden al estado en

el que se encontraba el código fuente de Maixon hasta la fecha de definición del Product Backlog.

The screenshot displays the 'Maixon' web application interface for property management. The top navigation bar includes 'Inicio', 'Propiedades', 'Sitio Web', 'Vendedores', 'Contactos', and 'Calendario'. The user is logged in as 'Administrador Galas'. The main content area is titled 'Departamento / Ubicación del inmueble' and contains several form fields:

- Zona Geográfica:** Bs.As. G.B.A. Zona Oeste
- Partido / Localidad:** Ituzaingó
- Barrio / Ciudad:** Parque Leoir
- Calle:** Alberni N° 200
- Ocultar dirección:**
- Pisos:**
- Entre:** CASTRO BARROS y LAMARCA
- Cerca de:** 16 Cuadras de estación

Below the form fields, there is a map of Buenos Aires with a red pin indicating the location. The map includes a legend and a 'Mapa Satellite' button. The coordinates are displayed as Latitud -34.65001297 and Longitud -58.6610527000003. The interface also features buttons for 'Guardar y cerrar', 'Cancelar', and 'Siguiente'.

Figura 7.3 - Vista de carga de Propiedad, sistema Manager.

## 7.5 Sprint 0

### 7.5.1 Planificación del Sprint 0

Para la planificación del primer Sprint se tuvo en cuenta las recomendaciones de la metodología en [12], recomienda realizar dos reuniones:

#### Reunión de Planificación Parte Uno

Mantuvimos una reunión entre Scrum Master y Product Owner en donde se solicita a éste último que seleccione del Product Backlog los elementos que desea que se desarrollen en la primer iteración y que en base a la estimación realizada por equipo sumen 20 Puntos Historia, valor que corresponde a la velocidad estimada por el equipo por sprint.

Se seleccionan los elementos y se asesora al Product Owner para describir las Historias de Usuario relacionadas y las Definiciones de Hecho correspondientes.

Las historias de usuario que corresponden al primer sprint quedan definidas:

- **ID 0 - Para todas las Inmobiliaria de plan Premium, permitir la configuración de su Page desde el sistema Manager**
- **ID 1 - Funcionamiento de acceso público al sistema Maixon Page de cada Inmobiliaria**
- **ID 2 - Recuperar y mostrar propiedades de Maixon en las Pages de cada Inmobiliaria**

De lo expuesto por el Product Owner se le hace algunas observaciones:

- Que no es factible la funcionalidad que requiere acerca del alta de Dominios, no se pueden dar de Alta a través de una API, ya que es un trámite que lleva tiempo y en algunos casos se requiere un pago internacional. No hay manera de realizarlo como un proceso integrado al sistema.
- La importancia de que los administradores de inmobiliarias no editen el diseño de su Maixon Page sino que seleccione uno de un grupo de Templates predefinidos.
- La necesidad de contar con un entorno para el sistema Maixon Pages independiente de Maixon pero que comparta la misma base de datos de propiedades.

#### Reunión de Planificación Parte Dos

Mantuvimos reunión entre el equipo diseñador, programador y también Product Owner porque es quien va a hacer las pruebas de unidad.

En base a las Historias de Usuario, definimos las tareas necesarias para cumplir con las definiciones de hecho de cada una. El cuanto al tiempo que cada integrante del equipo le va destinar al sprint, estaba preestablecido que siempre eran 4 horas diarias. El resultado fue la **estimación en horas** de dichas tareas.

También analizamos las posibilidad de paralelismo de ejecución de tareas, o sea, la

posibilidad de realizar tareas no dependientes entre programador y diseñador.

El resultado de la planificación del sprint 0 arrojó:

**Meta del Sprint:** lanzamiento del sistema Maixon Pages.

**Miembros:** 1 Programador, 1 Maquetador, Product Owner como tester.

**Fecha de Inicio:** 03/07/2013

**Fecha de finalización:** 28/06/2013

Sprint 0 - Sistema Maixon Page							
Backlog ID	Elemento del Backlog	Recurso	Tarea	Estado	Tiempo Estimado HS	Tiempo Usado HS	Sistema
0	Para todas las Inmobiliaria de plan Premium, permitir la configuración de su Page desde el sistema Manager	Diseñador	Diseño de interfaz de configuración de Maixon Pages	Done	10		Manager
		Programador	Programación de configuración de maixon Pages para inmobiliarias	Done	20		Manager
		Programador	Testear	Started	12		Manager
1	Funcionamiento de acceso público al sistema Maixon Page de cada Inmobiliaria	Diseñador	Diseño de Templates para sistema Maixon Pages	Not Started	20		Maixon Pages
		Programador	Implementación de sistema maixon Pages con templates seleccionados	Not Started	20		Maixon Pages
		Programador	Testing, pruebas de unidad	Not Started	4		Maixon Pages
2	Recuperar y mostrar propiedades de Maixon en las Pages de cada Inmobiliaria	Programador	home de Maixon Pages	Not Started	8		Maixon Pages
		Programador	Buscador de Maixon Pages	Not Started	8		Maixon Pages
		Programador	Ficha de Propiedad de Maixon Pages	Not Started	8		Maixon Pages

Figura 7.4 - Sprint Backlog 0

### 7.5.2 Seguimiento y Control del Sprint 0

Se establece entre los miembros del equipo una reunión diaria o **Scrum Daily Meeting**, en donde cada uno va a registrar en un documento:

- Fecha
- ¿Qué ha hecho desde la última reunión?
- ¿Qué tiene planificado hacer hasta la próxima reunión?
- ¿Algún problema con la tarea asignada?

Se analiza que las historias de usuario dependen unas de otras y que dentro de la primera historia de usuario si bien es cierto también hay dependencia de tareas, es posible realizar tareas en paralelo:

- Diseño de configuración de Page en sistema Manager (10 días)
- Programación de configuración de Page de Inmobiliaria (20 días), en bruto, sin diseño html.

y luego en el momento que se tenga el terminado el diseño (día 10), implementar lo diseñado a la tarea de programación.

Lo mismo para las tareas:

- Diseño de Templates para sistema Maixon Pages
- Implementación de sistema maixon Pages con templates seleccionados



Figura 7.5 - Configuración de Maixon Pages

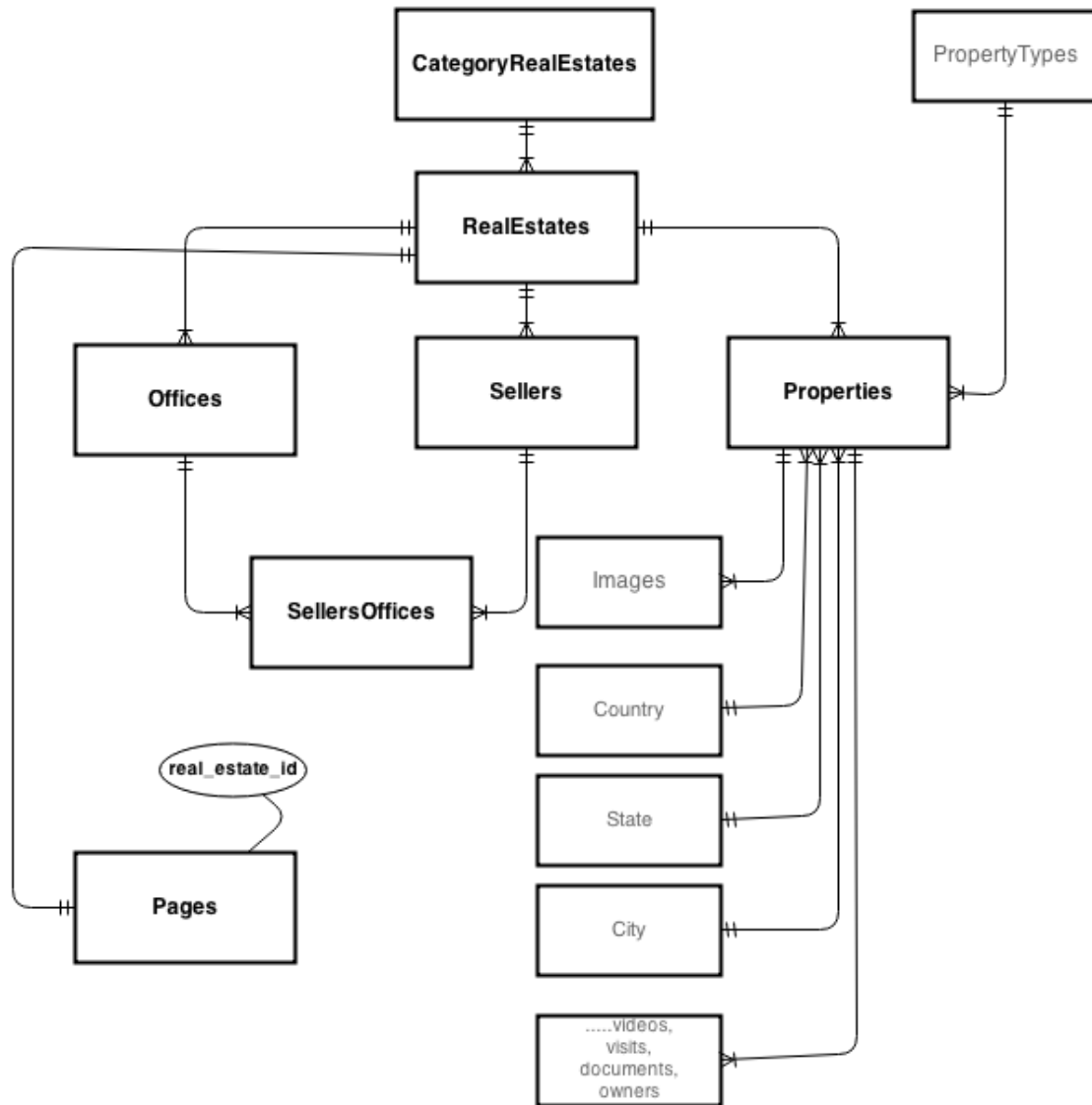
### 7.5.3 Revisión del Sprint 0

En reunión con el Product Owner se analizó como se había trabajado durante el sprint, se hizo notar que al principio se venía llevando un buen ritmo hasta la tercera semana en donde el trabajo fue más que el estimado: habían tareas dependientes en las cuales no se podía comenzar con al no tener finalizada la primera. También el Product Owner solicitaba que se resuelvan errores del sistema que no estaban registrados como tareas en el Product Backlog. Por este motivo se tuvo que interrumpir el Sprint.

### 7.5.4 Resultados del Sprint 0

Se desarrolló la estructura y edición del módulo de configuración Maixon Pages en el sistema Manager.

La Figura 7.6 muestra como evolucionó el modelo de datos.



**Figura 7.6** - Modelo de datos luego de Sprint 0

Se adjunta en el CD, dentro del directorio /codeSprint\_0 el código fuente correspondiente al controlador Pages, modelo Page y vista de la configuración del Page de Inmobiliarias en sistema Manager.

## 7.5.4 Retrospectiva del Sprint 0

Detectados los inconvenientes durante el desarrollo del Sprint se realizó una investigación con el objetivo de saber de qué manera encarar esta situación ya que en los libros consultados no especifica de qué manera actuar ante eventuales sucesos como peticiones de tareas extras por parte del Product Owner durante el desarrollo del Sprint, o qué hacer si una tarea demora más de lo estimado.

Se realizaron consultas a profesionales de empresa jujeña Softlogía SRL <[www.softlogia.com](http://www.softlogia.com)> quienes trabajan habitualmente desarrollando software siguiendo la metodología Scrum. De ellos se obtuvo la experiencia de que para ellos **Scrum en la vida real debe ser adaptativo, flexible a cambios y adaptarse a los mismos**, los nuevos requerimientos deben introducirse de manera razonable al Sprint Backlog e incrementarse la duración del actual Sprint solamente, aunque el realizar estas acciones es un indicador de que algo no está bien en la planificación o en los roles que debe ocupar cada miembro en el proyecto, en este caso el Product Owner debería respetar lo planificado.

## 7.5.5 Conclusión preliminar

Por ser la primera vez incursionando con la metodología, si bien se tuvo un comienzo organizado, no se tuvo la madurez para saber cómo solventar los cambios bruscos durante el desarrollo del primer sprint. Se deseaba mantener los principios de la metodología de:

- Mantener los límites de tiempo y duración fija de Sprint
- Respetar los Roles de los miembros del proyecto.
- No interrumpir constantemente a los equipos durante el Sprint.

En la realidad no se pudo evitar esto y se optó por suspender el Sprint para dar prioridad a la resolución de tareas pendientes demandadas por el Product Owner.

Se realizaron investigaciones, consultas a profesionales para saber de qué manera encarar los problemas mencionados. Se obtuvieron consejos para los próximos Sprints:

- En la manera razonable, permitir agregar funcionalidades durante el desarrollo del Sprint, aumentar la duración del mismo.
- Hacer hincapié en los roles de los miembros.
- Mejorar planificación, solicitar al Product Owner todas las funcionalidades necesarias en la planificación, tengan o no que ver con la meta del Sprint.

## 7.6 Sprint 1

### 7.6.1 Planificación de Sprint 1

Ahora con la experiencia del Sprint 0, se solicita una nueva reunión con el Product Owner.



## Reunión de Planificación Parte Uno

Mantuvimos una nueva reunión con el Product Owner en donde se le solicita nuevamente que seleccione del Product Backlog las funcionalidades que desea que se desarrollen en la siguiente iteración. Se le solicita también que considere la experiencia de la anterior iteración para, en lo posible, tratar de incluir todas las funcionalidades necesarias y que no ocurran interrupciones durante la próxima iteración. Nuevamente se le solicita que los elementos seleccionados deben sumar 20 puntos historias correspondiente a la velocidad del equipo.

Las Historias de Usuario definidas para el segundo Sprint son:

- **ID 8 - Exportar propiedades a través de API de Zonaprop**
- **ID 9 - Exportar propiedades a través de feeds xmls para : Trovit**
- **ID 10 - Exportar propiedades a través de feeds xmls para : Olx**
- **ID 15 - Dashboard de estadísticas de Inmobiliarias registradas, propiedades publicadas por cada una, historial de registro de propiedades**
- **ID 11 - Exportar propiedades a través de feeds xmls para : Reporte Inmobiliario**

De lo expuesto por el Product Owner se le hace algunas observaciones y asesoramiento:

- El requerimiento de exportar propiedades usando la API de Zonaprop tiene consigo un **riesgo**, o sea, un grado de incertidumbre sobre su realización, ya que lleva un tiempo de aprendizaje para su uso, aunque el Equipo cuenta con experiencia de desarrollo en la misma por haber formado parte en el desarrollo de otro proyecto en el cual también se realizaban publicaciones de propiedades usando API de Zonaprop. Se evalúa si hubo modificaciones o actualizaciones en la documentación de la API.  
El Product Owner se contacta personalmente con el personal de Zonaprop para asesorarse y obtener información sobre cómo realizar publicaciones automáticas en su portal, la respuesta es a través de una API que Zonaprop posee.  
Se requiere análisis de las funcionalidades que permite la API
- Se requiere importar las ubicaciones que posee el sistema Zonaprop para realizar un mapeo con las ubicaciones del sistema maixon para poder determinar qué datos enviar para cada propiedad.  
Se requiere determinar los datos a enviar según el tipo de propiedad : Casa, Departamento, Quinta, etc, ya que en cada caso el conjunto de datos a enviar es diferente por cada propiedad.
- Se hace saber al Product Owner la complejidad media en cuanto a recursos de sistemas para la funcionalidad de toma de Dashboard de estadísticas. Este tipo de procesos consume memoria y procesador, por lo que se recomienda usar un sistema de Cache con el cual ya se viene trabajando en Maixon para este tipo de consultas, con la mínima desventaja de no contar con datos actualizados constantemente sino cada cierto tiempo en que se actualiza la caché.

## Reunión de Planificación Parte Dos

Para estas funcionalidades es necesario un solo recurso: 1 programador. Ya que son tareas que no requieren diseño, simplemente procesos automáticos Cron que se ejecutan a

determinada hora y generan los xmls, lo mismo para la API de Zonaprop, la funcionalidad que publica la propiedad lo hará de manera transparente al usuario una vez que ha publicado una propiedad en Maixon, automáticamente se replica la propiedad en Zonaprop.

Se dividen las funcionalidades en tareas más pequeñas e independientes y se estiman las duraciones de las mismas.

El resultado de la planificación del sprint 1 arrojó:

**Meta del Sprint:** Desarrollo de Módulo Exportador. Exportación de propiedades desde maixon hacia otros sistemas.

**Miembros:** 1 Programador, Product Owner como tester.

**Fecha de Inicio:** 01/07/2013

**Fecha de finalización:** 26/07/2013

Sprint 1 - Modulo de exportación de propiedades					
Inicio:		1/07/2013			
Fin:		26/07/2013			
ID backlog	Elemento Backlog	Recurso	Tarea	Estado	Esfuerz Estima/ Real
	<b>Exportar propiedades a través de API de Zonaprop</b>	Programador	Analisis, aprendizaje y prueba de API de publicacion de propiedades de Zonaprop.com.ar	Done	4 4
		Programador	Recuperacion de Ubicaciones del sistema Zonaprop, guardarlo en una tabla local y armar mapeo de ubicaciones Maixon-Zonaprop	Done	12 13
		Programador	Determinar atributos necesarios segun Tipo de Propiedad a Publicar	Done	12 13
		Programador	Crear Proceso que se encarga de recuperar las propiedades de una Inmobiliaria, para cada propiedad: preparar datos segun tipo de propiedad, mapear ubicacion y publicar; guardar resultado del proceso para cada propiedad	Done	20 12
		Programador	Informar en sistema Manager el estado de publicacion de cada propiedad	Done	8 8
		Programador	Pasar a entorno PRODUCCION el proceso desarrollado y configurar el horario en que se ejecutará el mismo	Done	4 6
	<b>Exportar propiedades a través de feeds xmls para : Trovit</b>	Programador	Analisis de documentacion de procesos de feeds de exportaciones de propiedades para los sitios: Trovit	Done	4 4
		Programador	crear un proceso que se encargue de recuperar las propiedades que cumplan las condiciones para ser exportadas y que el mismo cree un archivo xml segun la documentacion correspondiente.	Done	4 4
	<b>Exportar propiedades a través de feeds xmls para : Olx</b>	Programador	Analisis de documentacion de procesos de feeds de exportaciones de propiedades para los sitios: Olx	Done	2 2
		Programador	crear un proceso que se encargue de recuperar las propiedades que cumplan las condiciones para ser exportadas y que el mismo cree un archivo xml segun la documentacion correspondiente.	Done	4 4
	<b>Dashboard de estadísticas de Inmobiliarias</b>	Programador	Tomar estadísticas de ultimas inmobiliarias registradas. Grafico de Pie	Done	4 2
		Programador	Tomar estadísticas de cantidad Propiedades Publicadas por cada Inmobiliaria. Grafico de Pie	Done	4 3
		Programador	Tomar estadísticas de Propiedades Publicadas en los ultimos 6 meses en todo Maixon. Grafico lineal	Done	4 3
	<b>Exportar propiedades a través de feeds xmls para : Reporte</b>	Programador	Analisis de documentacion de procesos de feeds de exportaciones de propiedades para los sitios: Reporte Inmobiliario	Done	2 2
		Programador	crear un proceso que se encargue de recuperar las propiedades que cumplan las condiciones para ser exportadas y que el mismo cree un archivo xml segun la documentacion correspondiente.	Done	4 2

Figura 7.7 - Sprint Backlog 1

### 7.6.2 Seguimiento y Control del Sprint 1

Al ser un solo recurso activo para este Sprint fue más sencillo el seguimiento del mismo.

Uno de los principios del desarrollo ágil es “ritmo sostenible” o sea trabajar horas delimitadas en el horario laboral establecido. En este Sprint hubo momentos en los que no se pudo cumplir con este principio, el equipo se vió forzado a cumplir con las fechas establecidas por la urgencia del Product Owner de brindar estos servicios a las inmobiliarias, para ello tuvo que trabajar horas extras en alguno días.

A continuación se muestran algunos Scrum Daily Meetings.

#### 02/07/2013 - **Daily Scrum**

*Qué es lo que se ha hecho desde la última reunión?*

Se tiene conocimiento de la funcionalidad de la API de Zonaprop, se solicitan credenciales de acceso a entorno de pruebas de Zonaprop.

*Qué es lo que se planea hacer hoy?*

Se desea recuperar las Ubicaciones disponibles en ZOnaprop.

Armar la Clase PHP Zonaprop con la cual se va a trabajar.

*Existe algún impedimento para continuar con las tareas?*

No se puede impactar con la API, credenciales no están habilitadas. Se solicitan nuevamente

#### 03/07/2013 - **Daily Scrum**

*Qué es lo que se ha hecho desde la última reunión?*

Se recuperaron las Ubicaciones y se guardaron en una tabla:

Table: **zonaprop\_locations**

Fields:

- id : (int autoincrement)
- zp\_id: (id de la ubicacion en zonaprop)
- type: suburb, city, county, private\_suburb
- name : nombre de la ubicacion
- parent\_id : va a indicar quien es la ubicación padre dentro de la misma tabla.
- full\_text: campo de texto para almacenar en formato JSON todo lo relacionado a esa ubicacion que devuelva la api.

*Qué es lo que se planea hacer hoy?*

Una vez recuperadas las ubicaciones se desea determinar dentro de la misma tabla cuales ubicaciones son Barrios, Barrios Privados y Ciudades. Para todos ellos , se debe determinar su ParentId dentro de la misma tabla.

Comenzar con el Mapeo de ubicaciones de Maixon con su correspondiente en Zonaprop.

Un mapeo es una correspondencia uno a uno de ubicaciones de Maixon con las ubicaciones de Zonaprop, se debe realizar debido a que la API de Zonaprop acepta la publicación de propiedades de acuerdo a sus IDs de ubicaciones. Para llevar a cabo esto se debe convertir las ubicaciones de las propiedades de Maixon en ubicaciones de Zonaprop.

Se debe crear una tabla: **zonaprop\_map**

*Existe algún impedimento para continuar con las tareas?*

--

#### 04/07/2013 - **Daily Scrum**

*Qué es lo que se ha hecho desde la última reunión?*

Se creó la tabla de mapeo: **zonaprop\_map**

fields:

- id . (autoinc) deben haber tantos registros como ciudades y barrios hay en Maixon
- suburb\_woeid , corresponde a un barrio de Maixon (puede ser nulo)

- suburb\_name, nombre del barrio
- city\_woeid, corresponde a la ciudad de Maixon (no nulo)
- city\_name, nombre de la ciudad
- county\_woeid, partido de maixon
- county\_name, nombre del partido
- state\_woeid, provincia de maixon
- state\_name, nombre de la provincia
- zonaprop\_location\_id , id de tabla zonaprop\_locations
- zonaprop\_location\_zp\_id, el ID de ubicacion de zonaprop que responderá la tabla cuando para las ubicaciones ya sean barrios o ciudades indicadas en suburb\_woeid, city\_woeid
- zonaprop\_location\_name, nombre de la ubicacion de zonaprop.
- zonaprop\_location\_full\_text

*Qué es lo que se planea hacer hoy?*

Realizar el proceso de Mapeo automático de ubicaciones. Zonaprop acepta ubicaciones por un solo código, diferente de como se maneja en Maixon que se carga Provincia, Partido, Localidad y si es necesario Barrio, entonces, para estos casos se debe hacer un Mapeo para que dada una Propiedad en Maixon, con su correspondiente Provincia, Partido, Localidad y Barrio, se determine cuál ID de Ubicación le corresponde en Zonaprop.

*Existe algún impedimento para continuar con las tareas?*

no

**05/07/2013 - Daily Scrum**

*Qué es lo que se ha hecho desde la última reunión?*

Se finalizó el Mapeo, se cuenta con el %90 de las ubicaciones Mapeadas, debido a que existen ciudades o barrios que sintácticamente no coinciden pero en la semántica se sobreentiende por ejemplo: en maixon es "Paternal" y en zonaprop "La Paternal", queda tarea para el PO realizar un matcheo manual para estas ubicaciones.

*Qué es lo que se planea hacer hoy?*

Determinar los atributos del XML a enviar según el Tipo de Propiedad en cada publicación. En base a la documentación de Zonaprop, comenzar con el proceso que recupera los datos de cada propiedad y arma el xml correspondiente.

*Existe algún impedimento para continuar con las tareas?*

no

**06/07/2013 - Daily Scrum**

(Dia Sabado, adelantando trabajo, programador no disponible Jueves 11 y Viernes 12)

*Qué es lo que se ha hecho desde la última reunión?*

Para los avisos a publicarse, existen datos comunes a todos los tipos de propiedades, estos son:

Título,

Descripción,

Precio,

Moneda,

Ubicación(calle, altura, coordenadas, id ubicación ),

Datos de contacto (email, nombre, apellido, teléfono),  
listado de urls de imágenes.

luego existen otros datos según cada tipo de propiedad, son muchos y se esta trabajando sobre ellos.

*Qué es lo que se planea hacer hoy?*

Continuar con la labor de armar las especificaciones de datos para cada tipo de propiedad.  
Servicios, Servicios Adicionales, Ambientes, Medidas de Ambientes.

*Existe algún impedimento para continuar con las tareas?*

no

**08/07/2013 - Daily Scrum**

*Qué es lo que se ha hecho desde la última reunión?*

Armado de atributos adicionales para cada propiedad.

*Qué es lo que se planea hacer hoy?*

Probar publicaciones, guardar resultado de la publicación, corregir errores que indique la API, optimizar la publicación.

En tabla Properties de Maixon se crean nuevos campos para esto:

Fields:

- zonaprop\_status: indica el estado de cada propiedad de maixon en zonaprop: pending, failed, published
- zonaprop\_id, en caso de haber sido publicada, guardar el ID de ZOnaprop, para que luego los vendedores puedan saber cual propiedad ha sido publicada
- zonaprop\_json\_historial, guardar cualquier información adicional que devuelva la API en la publicación de esta propiedad, se almacena en formato JSON.

*Existe algún impedimento para continuar con las tareas?*

Si, pero se ha solventado, este tipo de operaciones debe hacerse desde una IP fija, localmente se cuenta con un abono básico de internet por lo que la dirección IP es dinámica, cada actualización del proceso debe subirse al entorno DEV de Maixon y realizar desde allí las pruebas a través de comandos.

**09/07/2013 - Daily Scrum**

(feriado , adelantando trabajo, no voy a estar disponible Jueves 11 y viernes 12)

*Qué es lo que se ha hecho desde la última reunión?*

Se han publicado exitosamente algunas propiedades y otras se han almacenado exitosamente los resultados de porqué la api rechaza la publicación, esta información es útil para indicarle a los vendedores cuales campos faltan y deben ser corregidos.

*Qué es lo que se planea hacer hoy?*

Refinar el proceso de publicación. obtener los mínimos errores posibles en las publicaciones.

*Existe algún impedimento para continuar con las tareas?*

no

10/07/2013 - **Daily Scrum**

*Qué es lo que se ha hecho desde la última reunión?*

Se ha refinado el proceso de publicación de propiedades, el proceso se encuentra listo para pasar a entorno de Producción.

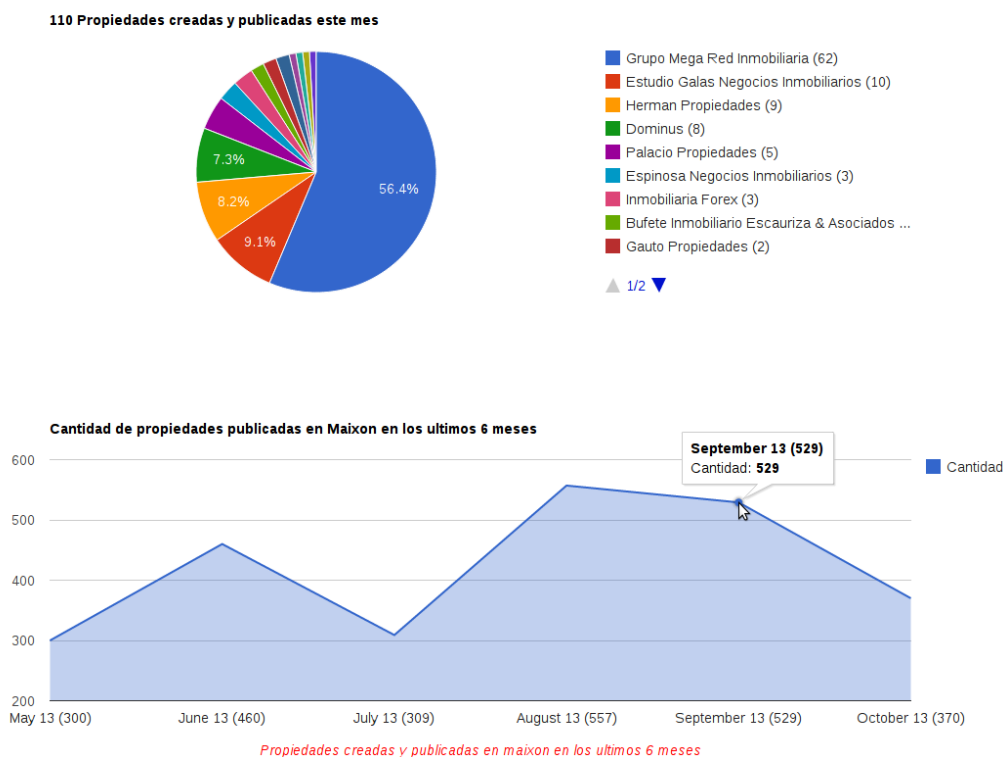
*Qué es lo que se planea hacer hoy?*

Último refinamiento del proceso de publicación de propiedades, se cuenta con todos los datos necesarios para cada tipo de propiedad. Los únicos errores reportados por la API son aquellos de Ubicación , que nuestro proceso no encuentra la correspondencia de ubicación maixon-zonaprop para una propiedad.

El proceso esta listo para instalarse en entorno de Producción, faltan las credenciales correspondientes.

*Existe algún impedimento para continuar con las tareas?*

no



**Figura 7.8 - Dashboard de estadísticas de sistema Admin**

**Gráficos de BurnDown Chart y Cumulative Flow Chart.**

Se lleva control del corriente Sprint con el diagrama de BurnDown Chart que representa las comparaciones de trabajo estimado y trabajo restante del equipo durante el Sprint.

También se llevó a cabo el Cumulative Flow Chart que representa las horas estimadas acumuladas de trabajo versus las horas reales acumuladas de trabajo, como se mencionó anteriormente, se trabajó horas extras en algunos días para poder cumplir con las fechas estimadas del Sprint, entonces el **Cumulative Flow Chart representa cuánta distancia existe**

entre las horas estimadas de trabajo versus las horas reales trabajadas y sirve para optimizar las futuras estimaciones de tareas.

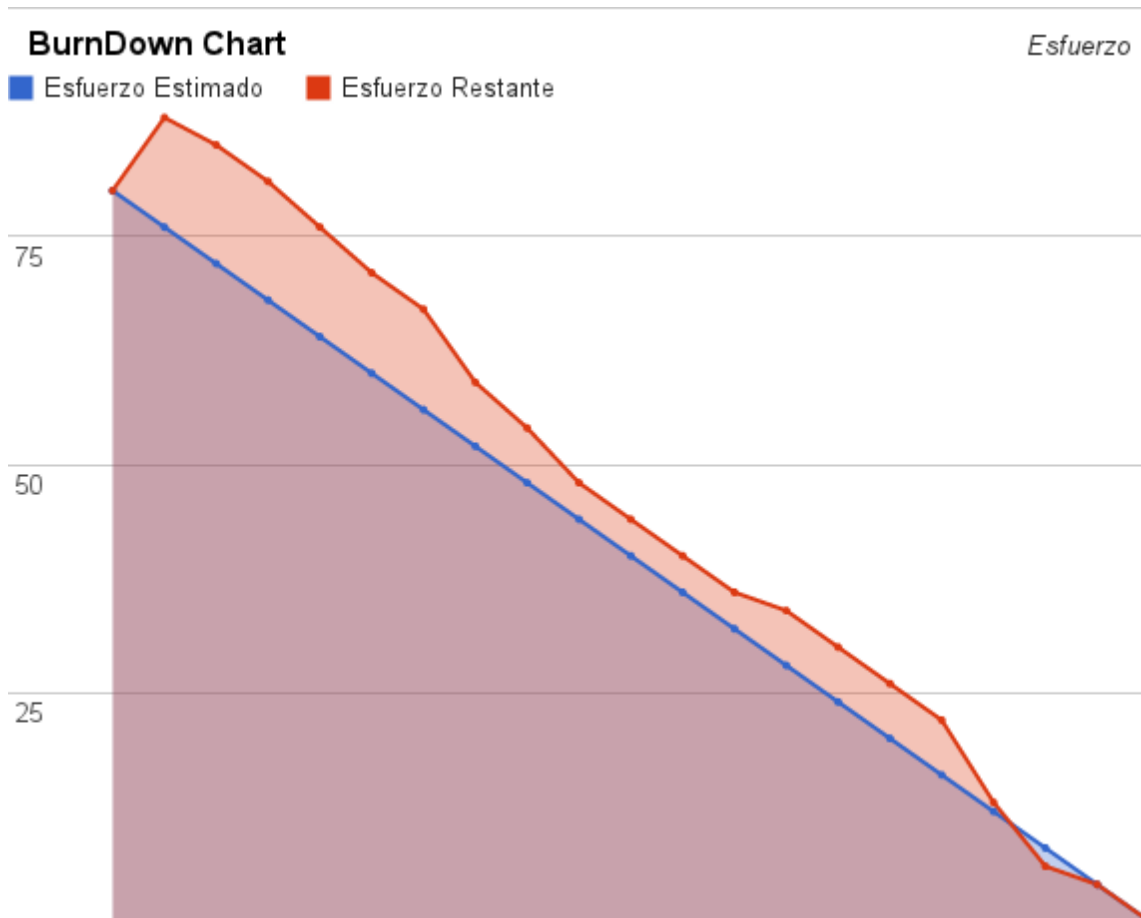


Figura 7.9 - Burndown Chart Sprint 1

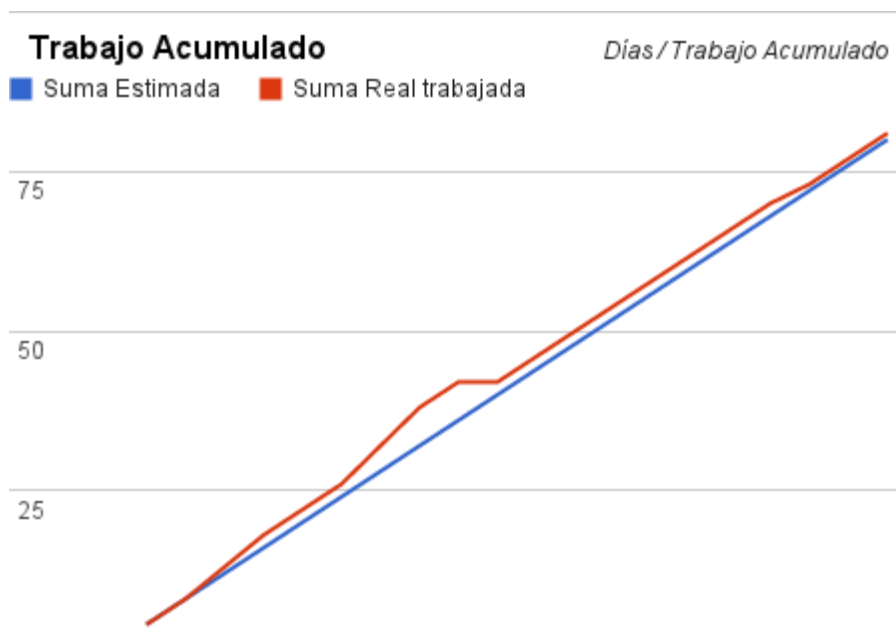


Figura 7.10 - Cumulative Flow Chart Sprint 1

### 7.6.3 Revisión del Sprint 1

Se cumplieron con el desarrollo de todos los elementos del Product Backlog que fueron incluidos en Sprint.

Por cada elemento cumplido del Product Backlog se hacía una Revisión y testing junto con el Product Owner y se aceptaron conforme.

Se mejoraron las estimaciones, se tuvo en cuenta en un tiempo de holgura para cada tarea, ese tiempo era considerado para posibles modificaciones que el Product Owner podría requerir en medio del Sprint. El Product Owner tuvo conocimiento de las tareas que se realizaban, esto quizás contradice en parte la metodología que dice que solo se ven Equipo y Product Owner al inicio y final del Sprint. Pero el hecho de que el Product Owner tenga conocimiento de las tareas le permite en este proyecto: tomar decisiones e informar a las Inmobiliarias sobre el estado de las nuevas funcionalidades.

Un inconveniente que se presentó durante el desarrollo de las tareas relacionadas con la API de Zonaprop, fue que no se tuvo en cuenta el tiempo que lleva estudiar la conexión con la API y que a veces se realizaron consultas a los encargados de Zonaprop y las mismas demoraron en ser contestadas por lo que esto en alguna manera retrasaba el tiempo planificado, pero se pudo solventar.

Se testeó junto con el cliente la ejecución del proceso de publicaciones en Zonaprop, se le informa cuales son los errores que devuelve la API en algunos casos. El Product Owner se encargará de comunicar a las Inmobiliarias las recomendaciones necesarias que deben considerarse a la hora de cargar propiedades para que cumplan las condiciones para publicarse en Zonaprop.

Se le enseña el sistema de Mapping y Matching de Ubicaciones de Maixon a Zonaprop, el Product Owner está conforme con el mismo.

Respecto de los archivos XMLs de exportación hacia OLX, Trovit y Reporte Inmobiliario, si bien parecen fáciles de crear, el factor crítico son los recursos del sistema para generar esos archivos, ya que en promedio son 5000 propiedades exportadas todos los días (propiedades creadas hace 3 meses hasta la fecha y en estado activo) y para cada una realizar las consultas a tablas relacionadas, esto genera un consumo excesivo de memoria y performance en el servidor MySQL, esta tarea de mejorar la performance es candidata al próximo sprint.

Los XMLs exportados pueden ser testeados por herramientas que provee cada portal que recibe los archivos.



### 7.6.4 Resultados del Sprint 1

El modelo de datos fue evolucionando de acuerdo a los nuevos requerimientos y a los objetivos cumplidos en el Sprint 1 como se ve a continuación.

Las propiedades se publican en Zonaprop, para ello cada Oficina cuenta con un *username* con el que se verifica la empresa que publica. Luego de la publicación la api devuelve datos como *zonaprop\_id*, *zonaprop\_status* y otros datos guardados en *zonaprop\_json\_data*.

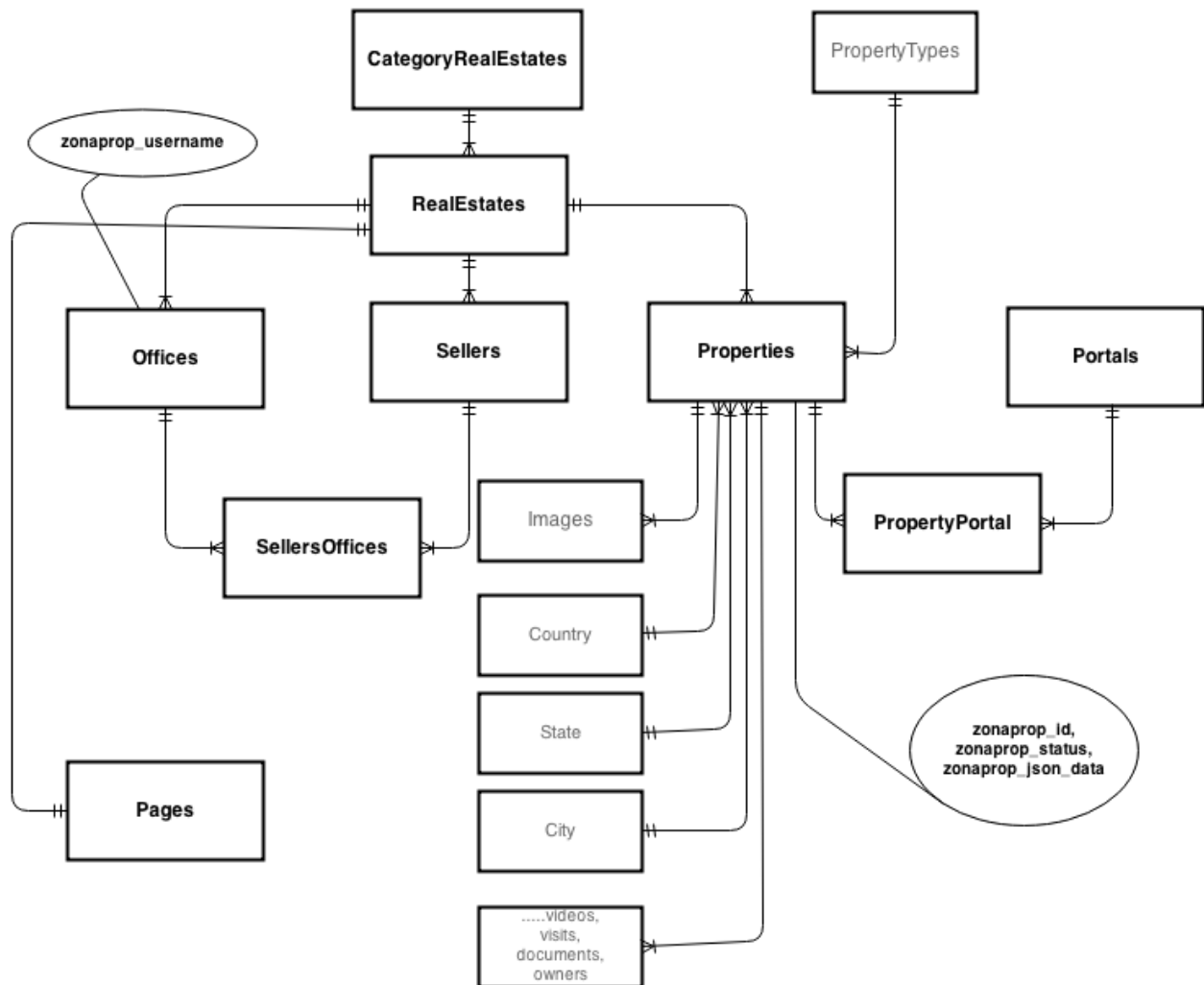


Figura 7.11 - Modelo de datos al finalizar el Sprint 1

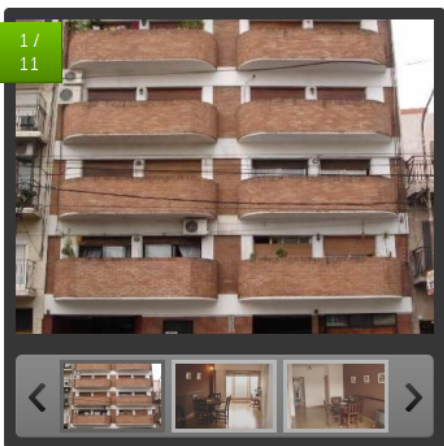
Se pueden observar la correspondencia de propiedad cargada en sistema Manager de Maixon y su correspondiente publicada en Zonaprop a través de la API. Como ejemplo se toma la propiedad código: **gal-1156** [13], que al momento de habilitarse en Maixon se publicó automáticamente en Zonaprop a través de la API [14]:

VENTA Código: GAL-1156

## Gaona 2200, Ramos Mejía, La Matanza, Bs.As. G.B.A. Zona Oeste

USD 90.000

Detalle Fotos (11) Mapa



**Tipo de propiedad:** Departamento  
**Edificación:** Departamento  
**Disposición:** Al frente  
**Dormitorios:** 2 Dorm  
**Baños:** 1 Baños  
**Sup. Cubierta:** 70 Mts2  
**Antigüedad:** 8años  
**Cerca de:** 3 cuadras de la estación de Ramos Mejía  
**Última modificación:** 2014-05-30 13:30:03  
**Visitas:** 29

Departamento en muy buena zona sobre Av. Gaona. Tiene 70m2 cubiertos al frente. Consta de amplio y luminoso living comedor, cocina totalmente equipada con muebles bajo mesada y alacena, dos cómodos dormitorios con placard, baño completo, lavadero y gran balcón. A SOLO 3 CUADRAS DE ESTACIÓN RAMOS MEJÍA!!!

Sé el primero de tus amigos al que le guste esto.

### ¿Qué hay cerca?

- Showcase Haedo**  
 Av. Dr. Luis Guemes 393
- El Fortín**  
 Av Alvarez Jonte 5299
- Valinor The Irish Pub**  
 Av. Rivadavia 14802
- Estadio José Amalfitani (Vélez Sársfield)**  
 Av. Juan B. Justo 9200

Regístrate, es gratis!

Ponte en contacto con nosotros

#### Contactar inmobiliaria



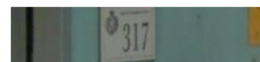
Estudio Galas Negocios Inmobiliarios

Nombre y apellido

Correo electrónico

Teléfono

**Mensaje**  
 Encontré en Maixon la siguiente propiedad y me gustaría tener más detalles GAL-1156 - Gaona 2200, Ramos Mejía, La Matanza, Bs.As. G.B.A. Zona Oeste



Ingrese texto de la imagen

Guardar esta propiedad en mis favoritas

Ud. se compromete a [Términos de Uso y Política de Privacidad](#) de Maixon.

Figura 7.12 - Propiedad código gal-1156, publicada en sistema Maixon Manager.

Venta, Departamentos Ramos Mejia, Gaona 2200 - zonaprop.com
Alquiler | Venta | Emprendimientos | Inmobiliarias | Mobile | Otras secciones
Ingresar | Búsquedas | Avisos | Regístrate | Publicar

Encuentra tu próximo Inmueble

---

**Gaona 2200**  
 Ramos Mejia - La Matanza - Bs.As. G.B.A. Zona Oeste

U\$S 90.000

Venta

DESCRIPCIÓN
UBICACIÓN

Click para ampliar

TIPO DE INMUEBLE

Venta de Departamentos Departamento

UD 90000 Departamento en Venta Gaona 2200

Superficie total	75,00 m <sup>2</sup>	Ambientes	3
Dormitorios	2	Baños	1
Antigüedad	8 años	Estado	Muy bueno

Contactar

Agendar visita

Quiero que me llamen

Descripción

Departamento en muy buena zona sobre Av. Gaona. Tiene 70m2 cubiertos al frente. Consta de amplio y luminoso living comedor, cocina totalmente equipada con muebles bajo mesada y alacena, dos cómodos dormitorios con placard, baño completo, lavadero y gran balcón. A SOLO 3 CUADRAS DE ESTACIÓN RAMOS MEJIA!!!

Solicitar más información »

Características Generales

Tipo de departamento:	Departamento
Superficie total (m2):	75
Superficie cubierta (m2):	70
Antigüedad:	8 años
Ambientes:	3
Cantidad dormitorios:	2
Cantidad baños:	1
Orientación:	N
Tipo edificio:	Entre medianeras
Estado del edificio:	Muy bueno
Estado del inmueble:	Muy bueno
Expensas (\$):	615
Apto profesional:	Si

Solicitar más información »

Servicios

- ✓ Agua corriente
- ✓ Desagüe cloacal
- ✓ Gas natural
- ✓ Pavimento
- ✓ Teléfono

Solicitar más información »

Ambientes

✓ Living comedor

Solicitar más información »

Medidas

Dormitorio con placa	3 x 3,5	Dormitorio con placa	3,5 x 3,5
Living comedor	7 x 4	Cocina	3 x 2

Solicitar más información »

Contacta ahora al anunciante

Estudio GALAS  
Otros avisos de este anunciante

Contacta ahora al anunciante

Estudio GALAS  
Otros avisos de este anunciante

Ingresar tu nombre

Ingresar tu e-mail

Ingresar tu teléfono

Comentario (200 caracteres)

Ingresar aquí tu comentario...

✓ Acepto que me envíen información de zonaprop.com

Ver Datos del Anunciante

Avisos Similares

**Maipu 500 - Ramos ...**  
Departamentos - Venta  
U\$S 85.000

**Magallanes 800 - R...**  
Departamentos - Venta  
U\$S 85.000

**Pueyrredon 100 - R...**  
Departamentos - Venta  
U\$S 87.000

**Bartolomé Mitre 60...**  
Departamentos - Venta  
U\$S 85.000

Figura 7.13 - Propiedad publicada en Zonaprop a través de API.

Por otro lado, en la carga de propiedades, para cada propiedad se selecciona el Portal hacia donde se exportará a través de los feeds XML.

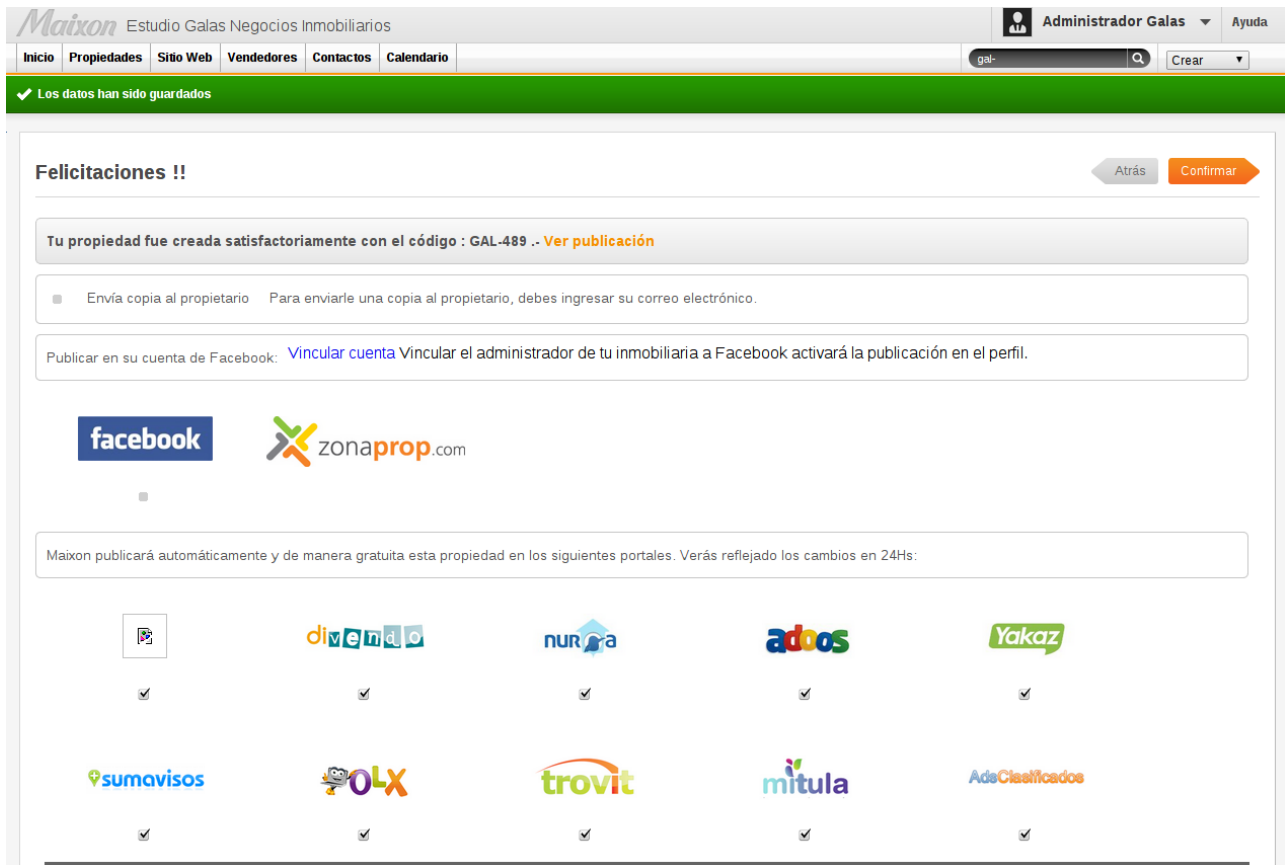


Figura 7.14 - Carga de Propiedad, selección de portales a exportar

Se adjunta en el CD, dentro del directorio /codeSprint\_1 el código fuente correspondiente al controlador Properties, modelo Properties y vista de la exportación de la propiedad hacia portales y publicación en Zonaprop.

## 7.6.4 Retrospectiva del Sprint 1

Para ser el primer Sprint llevado a cabo en forma completa, el equipo que ha trabajado, un solo programador, se puede decir que fue satisfactorio, ya que se han cumplido todos los objetivos en su totalidad, aunque se han trabajado en horas extras en algunas ocasiones o bien se ha adelantado trabajo porque en una fecha determinada no se iba a poder trabajar. Esto último no es recomendable por la metodología, pero al ser parte de la vida real, el equipo asume el compromiso y desea llegar a cumplir con las tareas en el tiempo estimado.

Existe una tarea relacionada con las publicaciones de propiedades en Zonaprop la cual fue estimada en 20 horas y gracias a la experiencia previa del equipo de haber trabajado anteriormente con la API, se cumplió en 12 horas. Reduciendo considerablemente el desarrollo del elemento del Product Backlog relacionado a Zonaprop, lo que permitió contar con tiempo extra para desarrollar las tareas relacionadas con las estadísticas.

El hecho de dividir a los elementos en tareas más pequeñas y estimarlas ha ayudado mucho a la hora de fijar metas diarias y de alguna manera a trabajar “contra-reloj”.

Debido a la experiencia de trabajar con el Product Owner, siempre se han estimado las tareas dando un poco de holgura para contemplar casos de modificaciones “sobre la marcha” y poder cumplir las tareas en los tiempos estimados. Aun así se ha trabajado horas extras en algunos casos.

### **¿Que fue bien?**

Se cumplieron todas las tareas establecidas y, si quedan modificaciones, son muy pequeñas o insignificantes.

### **¿Que se podría mejorar?**

Las estimaciones un poco más, para no trabajar horas extras.

Algunas interrupciones por parte del cliente solicitando una tarea cuando se encuentra en medio de otra. Pero considero que esto es algo de la realidad del trabajo en cualquier metodología de trabajo y aunque no sea correcto hay que atender las necesidades del cliente. En este caso se ha solicitado agregar las tareas al Product Backlog y al Sprint Backlog (en medio de ejecución del sprint).

## **7.6.5 Conclusión preliminar**

El desarrollo del Sprint fue satisfactorio, se ha logrado optimizar el uso de la metodología, se cumplieron los objetivos del Sprint, no hubo interrupciones significativas por parte del Product Owner.

Se obtuvieron buenos resultados aunque se trataba de un equipo reducido porque en las tareas en las cuales se ha trabajado solo era necesario un programador.

Cabe mencionar que a mediados del Sprint, luego de implementar las funcionalidades con Zonaprop, el Product Owner recibió noticias de dicha empresa relacionadas interés en Maixon de parte de Zonaprop, sobre todo en las funcionalidades de CRM de Contactos y envío de alertas.

## **7.7 Sprint 2**

### **7.7.1 Planificación del Sprint 2**

Nuevamente reunión con Product Owner con experiencia en estimaciones de tareas.

#### **Reunión de Planificación Parte Uno**

El Product Owner expone en reunión que existen tratativas de empresa Zonaprop para adquirir Maixon y formar parte de ellos, en realidad de empresa Dridco que es dueña de Zonaprop,

Zonajobs, Zonacitas y varios portales más por lo cual varias tareas con las cuales se debería trabajar en este Sprint están relacionadas con:

- Mejorar el sistema actual antes de pasar a formar parte de Dridco.
- Nuevas funcionalidades vinculadas a los objetivos solicitados por Dridco en Maixon.

Como en los anteriores Sprints, se solicita al Product Owner (ahora formado por personal de Dridco y anteriores dueños de Maixon) seleccione del Product Backlog los elementos que desea que sean desarrollados en la siguiente iteración y que sumen 20 Puntos Historia en total.

La empresa Dridco se mostró interesada en la metodología que se estaba implementando en Maixon ya que ellos también trabajan usando Scrum. Se solicitaron experiencia respecto de si tuvieron que enfrentarse a situaciones como las que se vivió en los Sprints 0 y 1 de Maixon, se recibieron consejos y aportes significativos que se exponen en la conclusión de este Sprint.



Figura 7.15 - Fotografía empresa Dridco y sus proyectos.

El Product Owner recomienda que la prioridad es comenzar con las funcionalidades para el Módulo de Payments, Sistema Manager y Admin.

El Sprint Backlog queda definido por las siguientes Historias de Usuario.

- **ID 14 - Dashboard para resumen de Nuevos Contactos, Contactos que Necesitan Acción y Últimos Contactados**
- **ID 16 - Estadísticas para administrador de inmobiliaria: Propiedades Publicadas, Contactos Agregados, Contactos por Vendedor, Precio de Mt2 promedio**
- **ID 3 - Sistema Manager permitir la posibilidad de contratar un abono de cualquier plan Premium o Advanced, realizar el pago a través de DineroMail y registrar el mismo**
- **ID 4 - Verificar los pagos realizados por las inmobiliarias y actualizar los periodos de abono de las mismas**
- **ID 17 - Grupo Mega, diseño de portal personalizado**

- **ID 18 - Resolución de bugs y mejoras de Maixon (requerimiento agregado durante el desarrollo del Sprint 2)**

De lo expuesto por el Product Owner se le hace algunas observaciones y asesoramiento:

- Nuevamente se requieren Dashboard con estadísticas tanto de Contactos en sistema Manager como de Inmobiliarias y Propiedades en sistema Admin, se le indica el consumo de recursos de sistema que esto implica y la recomendación de usar Cache.
- Actualmente los pagos de abonos no son automáticos sino que los administradores de inmobiliarias son notificados vía email del vencimiento de su abono y que deben realizar el pago del mismo y enviar comprobante de pago para que los administradores de Maixon activen manualmente el nuevo periodo del abono. Se desea automatizar este proceso mediante el uso de la API de DineroMail. Como toda nueva API se comunica al Product Owner que esto lleva consigo un riesgo, o sea una incertidumbre de cuánto tiempo puede llevar el aprendizaje de dicha API.

- Según el resumen analizado en la página web de DineroMail, las transacciones de las operaciones de pago ya sea por Tarjeta de Crédito o por Cupon de Pago no son inmediatas, tienen un tiempo en el cual las entidades financieras habilita la transferencia de dinero.

Se requiere un Proceso que se ejecute en un momento del día y que verifique, mediante API de DineroMail, todos los pagos realizados y el estado de los mismos y para cada uno de ellos realice la actividad que corresponda: si fue un pago exitoso, actualice el plan por el periodo abonado, en caso contrario, bajar el plan a la Inmobiliaria a Basic con servicios restringidos.

- Grupo Mega es una inmobiliaria en Maixon que agrupa a cuarenta inmobiliarias reales, actualmente es el cliente principal por lo cual tiene un cierto tipo de privilegios. Las necesidades de Grupo Mega se convierten usualmente en requerimientos a desarrollar en el sistema Maixon y que también resulta útil para las demás inmobiliarias. Para este requerimiento, Grupo Mega solicita su propia web personalizada. Se recomienda al Product Owner mantener el sistema Maixon Page como motor de webs para las inmobiliarias y agregar funcionalidades sobre el mismo motor para que responda a los requerimientos de Grupo Mega.

### **Reunión de Planificación Parte Dos**

Se reúnen el Equipo: Programador y Diseñador para estimar las tareas necesarias para cumplir cada requerimiento del Sprint 2 y los tiempos de cada una.

Se hace notar que existen nuevamente riesgos relacionados con el aprendizaje de la API de DineroMail.

Se analizan las posibles no dependencias de tareas que pueden ejecutarse en paralelo por un lado el diseñador y por otro el programador y determinan en qué tiempos entregar el diseñador el trabajo que requiere el programador para cumplir con la tarea.

El resultado de la planificación del sprint 2 arrojó:

**Meta del Sprint:** Desarrollo de Módulo Payments, Estadísticas y portal de Grupo Mega

**Miembros:** 1 Programador, 1 Diseñador, Product Owner como tester.

**Fecha de Inicio:** 12/08/2013

**Fecha de finalización:** 06/09/2013

		Sprint 2 - Módulo Payments y Estadísticas			
		Inicio:	12/08/2013		
		Fin:	6/09/2013		
ID backlog	Elemento Backlog	Recurso	Tarea	Estado	Esfuere Estima Real Sistema
14	Dashboard para resumen de Nuevos Contactos, Contactos que Necesitan Accion y Ultimos Contactados	Programador, Diseñador	Recuperar Nuevos Contactos y filtro Propio,Sucursal e Inmobiliaria	Done	4 2 Manage
		Programador, Diseñador	Recuperar Contactos que necesitan Accion y filtro Propio,Sucursal e Inmobiliaria	Done	4 6 Manage
16	Estadísticas para administrador de inmobiliaria: Propiedades Publicadas, Contactos Agregados, Contactos por Vendedor, Precio de Mt2 promedio	Programador, Diseñador	Recuperar Contactos Atendidos y filtro Propio,Sucursal e Inmobiliaria	Done	4 2 Manage
		Programador, Diseñador	Propiedades publicadas en toda la inmobiliaria y filtro para los ultimos 3,6 y 12 meses	Done	3 3 Manage
		Programador, Diseñador	Contactos agregados en toda la inmobiliaria y filtro para los ultimos 3,6 y 12 meses	Done	4 3 Manage
		Programador, Diseñador	Contactos agregados por cada vendedor y filtro para los ultimos 3,6 y 12 meses	Done	3 3 Manage
3	Sistema Manager permitir la posibilidad de contratar un abono de cualquier plan Premium o Advanced, realizar el pago a través de DineroMail	Programador, Diseñador	Calculo del precio del Metro2 promedio en base a las propiedades de la inmobiliaria y Filtro por ubicaciones	Done	3 1 Manage
		Programador, Diseñador	Proceso realizar pago Api DineroMail	Done	8 10 Manage
4	Verificar los pagos realizados por las inmobiliarias y actualizar los periodos de abono de las mismas	Programador	Historial de Pago	Done	4 3 Manage
		Programador	Estudio de API de DineroMail para capturar movimientos de cuenta	Done	6 7 Admin
17	Grupo Mega, diseño personalizado	Programador	Proceso que verifica los pagos efectuados y actualiza los periodos de abonos	Done	12 12 Admin
		Diseñador	Diseño de web personalizada para Inmobiliaria: Grupo Mega	Done	10 10 Pages
		Product Owner	Solicitar Alta de Dominio: http://www.gmega.com.ar/	Done	3 3 Pages
18	Resolución de bugs y mejoras de Maixon	Programador	Buscador de Propiedades	Done	4 4 Pages
		Programador	Aplicacion de Diseño	Done	8 10 Manage
		Programador	Front: en el buscador de propiedad si se carga la ubicacion, pero no el tipo de operacion da un error invalido el sistema debe asumir por default que busca departamento en venta	Done	1 1 Manager
		Programador	Plan Basic al "Enviar Ficha" y al imprimir debe abrir el modal que invita a pagar y explicar mediante una key por que no puede enviar la ficha. y otra key que no puede imprimir pues eso pertenece al servicio pago	Done	1 0 Manager
Programador	con Eric.	Exportacion de Xmls, optimizar porque consume muchos recursos y baja el servidor, optimizar la generacion de XMLs	Done	1 0 Manager	
Programador			Done	4 4 Manager	

Figura 7.16 - Sprint Backlog 2

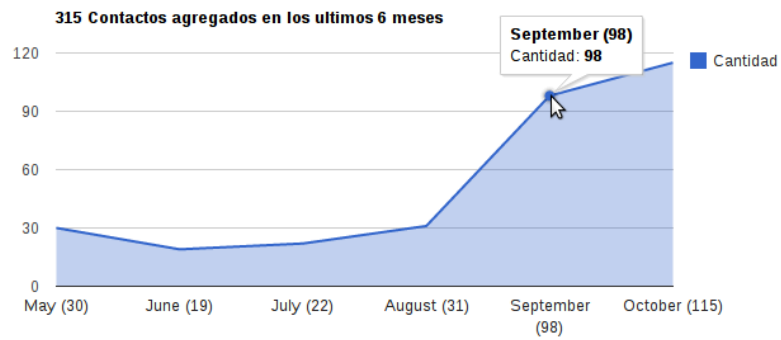
### 7.7.2 Seguimiento y Control del Sprint 2

El Sprint se desarrolló bajo un control minucioso, en base a las experiencias anteriores se buscaba:

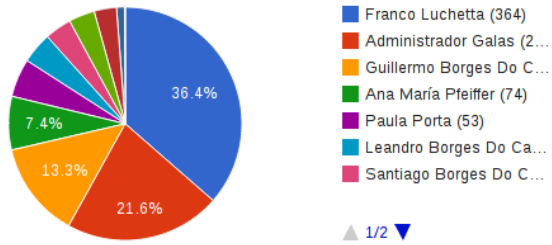
- Cumplir en tiempo y forma con las fechas estimadas de tareas, se pesumía otra vez horas extras por el riesgo de aprendizaje de la API de DineroMail
- El mejor rendimiento en equipo ya que en este Sprint contábamos con mejor experiencia.
- Adaptación y flexibilidad a eventuales cambios.



Filtrar para : 6 meses ▾



**999 contactos agregados en total**



**Figura 7.17 - Estadísticas para administrador de inmobiliaria**

Maixon Estudio Galas Negocios Inmobiliarios Administrador Galas

Inicio Propiedades Sitio Web Vendedores **Contactos** Calendario gal-

Propio Sucursal Inmobiliaria

**Nuevos Interesados** 0 Ver todos >>

Estos contactos tienen categoría "Interesados" y necesitan tu ayuda. Relaciónate ahora con ellos.

**Todavía no hay nuevos interesados**

**Necesitan Acción** 8 Ver todos >>

Contactate con tus contactos "Activos". Un buen seguimiento o marcarlo "Inactivo" hará que no te recordemos que lo contactes.

Nombre y apellido	Vendedor	Último contacto	Tipo de acción	Acción
Horacio Blanco	Administrador Galas	26/09/2013	Urgente	Contactar
Anabella Bouzo	Administrador Galas	26/09/2013	Urgente	Contactar
Daniel Franco	Administrador Galas	26/09/2013	Urgente	Contactar
Maria Florencia Miechkowski	Franco Luchetta	26/09/2013	Urgente	Contactar
Evangelina Quiroz	Administrador Galas	26/09/2013	Urgente	Contactar
Liliana Spione	Administrador Galas	26/09/2013	Urgente	Contactar
Nelly Tramaglia	Administrador Galas	26/09/2013	Urgente	Contactar
Panuel Manolo	Franco Luchetta	26/09/2013	Urgente	Contactar

**Últimos contactados** 6 Ver todos >>

Estos contactos fueron recientemente contactados.

Nombre y apellido	Último contacto	Acción
Alejandra Pondo	02/10/2013	Contactar
Daniel Addario	02/10/2013	Contactar
Victor Salinas	30/09/2013	Contactar

**Novedades**

- Se enviaron 8 alertas hoy [Ver todos >>](#)
- No tienes propiedades nuevas o modificadas [Ver novedades >>](#)
- 8 Contacto/s necesitan acción. [Ver todos >>](#)

**Agenda** [Crear tarea](#)

Octubre 2013

Lu	Ma	Mi	Ju	Vi	Sá
	1	2	3	4	5
7	8	9	10	11	12
14	15	16	17	18	19
21	22	23	24	25	26
28	29	30	31		

**Hoy**

Sin eventos hoy. AGREGAR

**Mañana**

Sin eventos mañana. AGREGAR

Figura 7.18 - Dashboard de contactos en sistema Manager

Gráficos de BurnDown Chart y Cumulative Flow Chart.

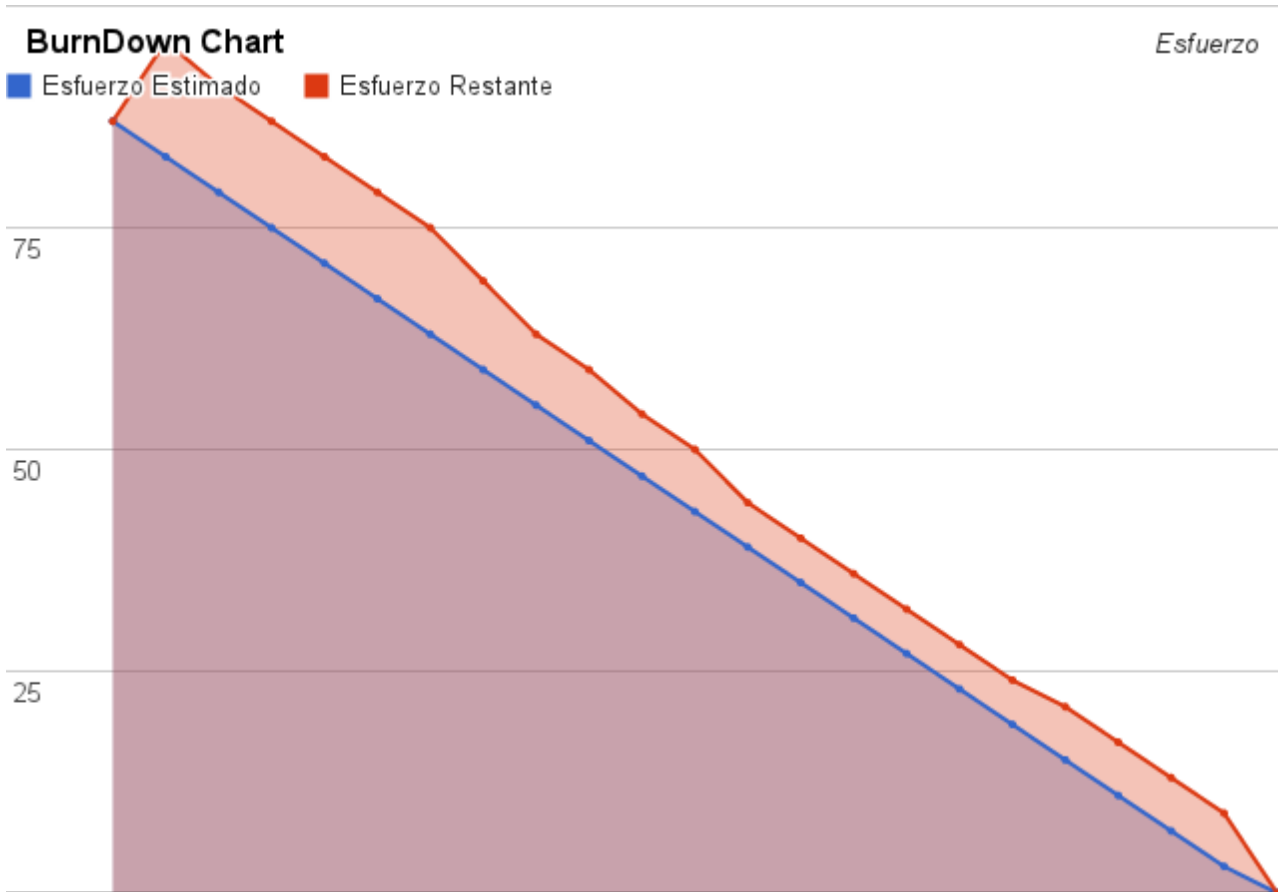


Figura 7.19 - BurnDown Chart Sprint 2

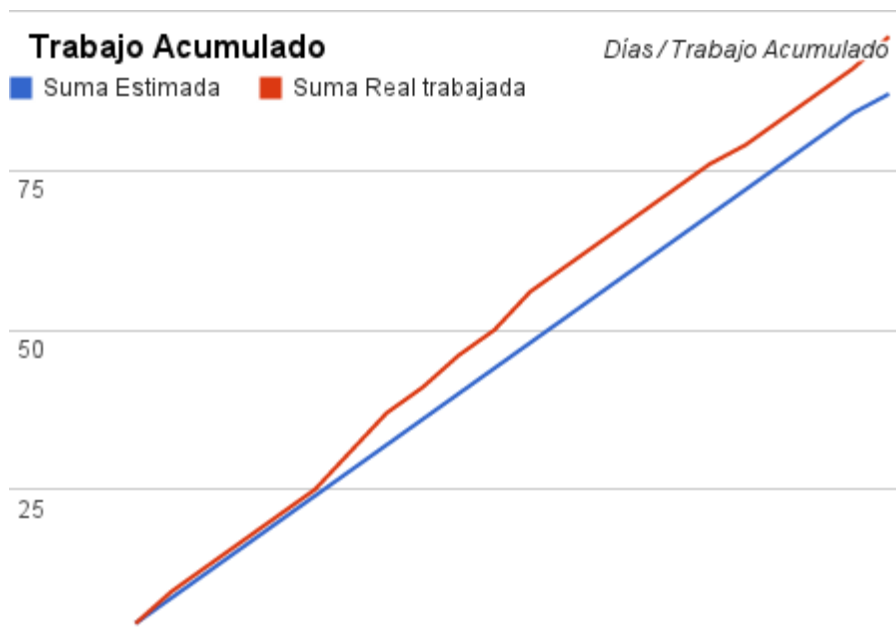


Figura 7.20 - Cumulative Flow Chart Sprint 2

### **7.7.3 Revisión del Sprint 2**

En base al trabajo y experiencia del Sprint 1 se pudo refinar las estimaciones para las nuevas tareas.

Se introdujo al Product Owner en la metodología y la importancia de no realizar cambios de bruscos de tareas sobre la marcha.

Se cumplieron en tiempo y forma con las tareas estimadas, aunque hubo un riesgo de aprendizaje de la API de DineroMail que hizo que el tiempo empleado aumentara para las tareas relacionadas con el uso de dicha API.

Se incorporó funcionalidades atractivas para los administradores de inmobiliarias: gráficos, charts que son útiles para la toma de decisiones y generan un resumen del estado actual de toda la inmobiliaria, usuarios y propiedades.

Las funcionalidades con relación al Dashboard de Contactos o posibles compradores tuvo la experiencia de ser más compleja de lo que se planteaba por tener que realizar muchas consultas a la base de datos para mostrar un resumen, nuevamente se dedicó mas tiempo de lo planificado. El resultado fue positivo ya que esto brinda a los vendedores un seguimiento óptimo de ventas de propiedades.

Grupo Mega es el cliente más importante de Maixon, es una grupo de 40 inmobiliarias de la ciudad de Buenos Aires unidas en una sola, por lo cual muchas funcionalidades nuevas se agregaron por requerimiento y consejo de ellos. Realizarles el Sitio Web personalizado era una tarea necesaria, tarea que no tuvo complejidades y en la cual se pudo reflejar el trabajo conjunto de Diseñador, Programador y Product Owner.

### **7.7.4 Resultados del Sprint 2**

Luego del Sprint 2 el modelo de datos continúa evolucionando, se le suman las tablas de Payments (Pagos) y de Contactos (People)

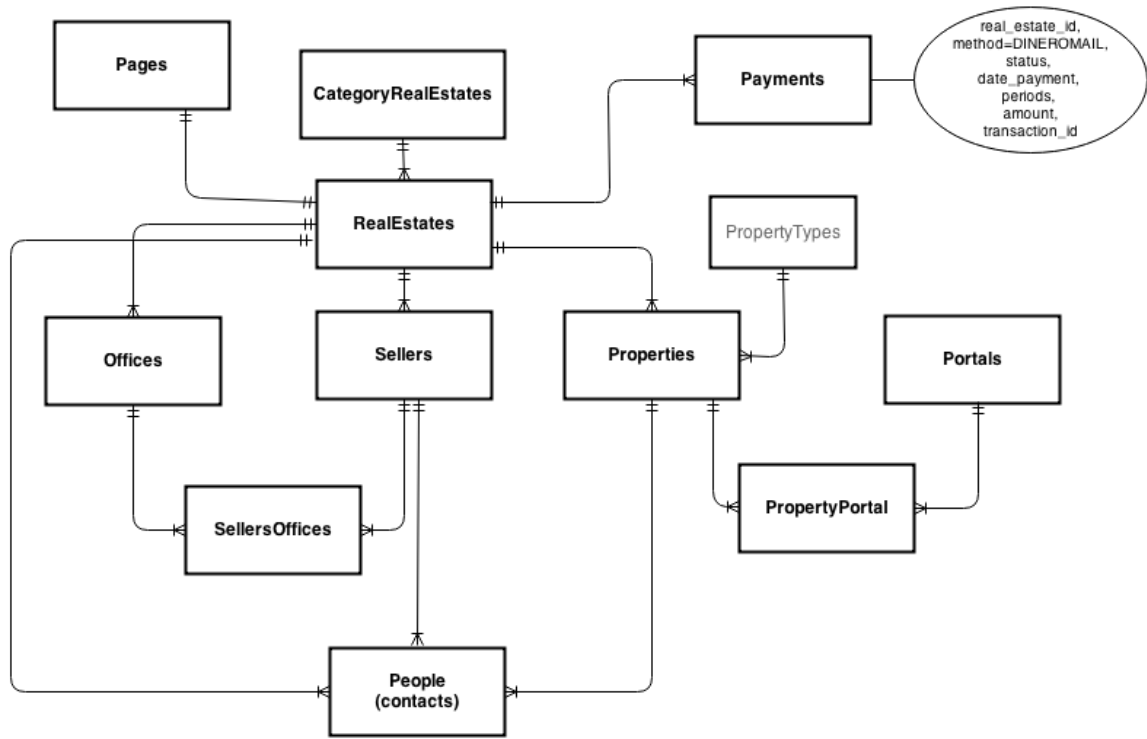


Figura 7.21 - Modelo de datos luego del Sprint 2

Proceso de Pagos, se selecciona un plan a contratar y mediante API de Dineromail se realiza el pago del mismo.

The screenshot shows the 'Servicio de pagos' (Payment Service) interface in Maixon Manager. It is titled 'Plazo a contratar y método de pago' (Contract term and payment method).

**Plazo de contrato (Contract Term):**

Producto	Valor Mensual	Meses
(Maixon Plan) PLATINUM	\$ 597,00	1 Mes
Sucursales contratadas	3	
<b>Total</b>	\$ 1.611,90 (IVA: \$ 338,50, Total: \$ 1.950,40, c / desc % 0,10 )	
Sitio Web	\$ 177,30 (IVA: \$ 37,23, Total: \$ 214,53)	
(Change Plan)	<b>Total: 2.165</b>	

**Método de pago (Payment Method):** Dineromail

Buttons: Cancelar, Siguiente

Footer: Maixon © 2014 - Todos los derechos reservados | Términos y condiciones | Soporte

Figura 7.22 - Generación de Pago, sistema Maixon Manager

Español (ES) ▼

---

**Maixon Real Estate Search** Sido Seguro

Resumen	Costo	Cantidad	Sub-Total [-]
#53869179ed722 Maixon PLATINUM - 1 Meses(0% {Off})	\$ 2.165,00	1	\$ 2.165,00
Costo de Financiamiento y Procesamiento	\$ 0,00	-	\$ 0,00
<b>Total</b>			<b>\$ 2.165,00</b>

▼ Paso 1: Elige el medio de pago [Cambiar medio de pago]

[\[Cambiar medio de pago\]](#)

Seleccione emisor de la tarjeta

- Hipotecario [?] ¡Sin interés!
- Provincred [?] ¡Sin interés!
- OTRO

► Paso 2: Completa los datos adicionales [Cambiar Datos Adicionales]

<b>Total</b>	<b>\$ 2.165,00</b>
--------------	--------------------

Powered by 

Cancelar
Pagar

**Figura 7.23** - Recepción de Pago a través de API DineroMail

En el CD se adjunta, dentro del directorio /codeSprint\_2, el código fuente del controlador Payments, modelo Payment y vistas de los procedimientos de Pagos de abonos.

## 7.7.4 Retrospectiva del Sprint 2

En este Sprint han trabajado en conjunto Diseñador, Programador y Product Owner.

El resultado ha sido satisfactorio, en base al Sprint 1, se logró mejorar las estimaciones de trabajo y cumplir con las mismas. Aunque fué inevitable realizar en pocos casos nuevamente horas extras para llegar a tiempo con las tareas.

Existen tareas que se realizaron en paralelo, como la construcción del portal de Grupo Mega, Dashboards de Administradores y Dashboard de Contactos: mientras el Programador se

encontraba realizando una tarea, el Diseñador avanzaba con otras, si no contaba todavía con el diseño, lo realizaba en crudo (sin css) y cuando contaba con el diseño lo aplicaba sobre la funcionalidad ya implementada, esto gracias al patrón Modelo-Vista-Controlador que provee el framework Cakephp. Las estimaciones de este tipo de tarea se hicieron tomando en cuenta los tiempos que los recursos podíamos trabajar en paralelo y los tiempos en que no podíamos hacerlo. De esta manera el tiempo total empleado es menor a la suma de los tiempos individuales de las tareas de los recursos.

Product Owner nuevamente encargado de las pruebas de unidad y reportando el resultado de las mismas.

### **¿Que fue bien?**

Se realizaron estimaciones más precisas de las tareas. Nos adaptamos a los cambios de nuevos requerimientos por parte del Product Owner. Logramos estimaciones de trabajo en equipo.

El Product Owner desde la perspectiva de cliente se adaptó a la metodología SCRUM.

### **¿Que se podría mejorar?**

Todavía hay que estimar mejor los riesgos de tiempos de investigación, por ejemplo de API de DineroMail o tomarlo como una tarea independiente.

Continuar integrando al Product Owner en la importancia de los principios de la metodología SCRUM

## **7.7.5 Conclusión preliminar**

El desarrollo del Sprint 2 se llevó a cabo satisfactoriamente.

A lo experimentado en los dos Sprints anteriores se suma la experiencia de trabajar en equipo y con tareas en paralelo.

La empresa Dridco interesada en adquirir Maixon se siente conforme de que el Equipo use Scrum como metodología ya que ellos la usan en el desarrollo de sus proyectos.

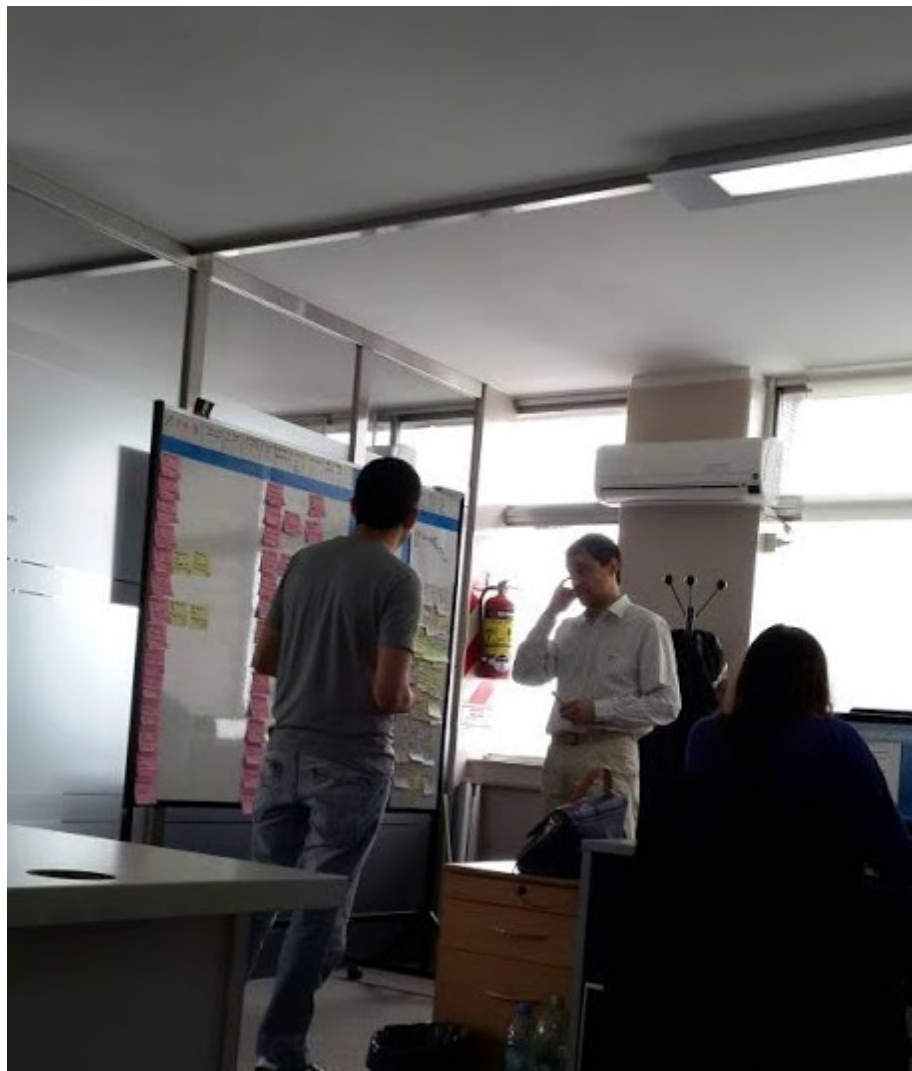
Se adquirió experiencia de la metodología en la vida diaria consultando a profesionales de empresa Dridco S.A. y Softlogía S.R.L. Se adquirieron aportes valiosos sobre que se debe aceptar introducción de requerimientos por parte del Product Owner sobre la marcha y de manera razonable como así también que es natural trabajar horas extras para cumplir con los objetivos pero esto es un indicador de que algo está fallando en la planificación.

## 8. Conclusiones generales

Son muchas las experiencias adquiridas a lo largo de estos 3 Sprints en el proyecto Maixon.

Si bien el Sprint 0 falló, ha dejado la experiencia de que hay que investigar más acerca de la vida diaria en proyectos reales con Scrum. Se ha consultado a profesionales de empresas desarrolladoras de software como Softlogia S.R.L. y Dridco S.A quienes aconsejaron que si bien la teoría de la metodología dice que no deben existir interrupciones por parte del Product Owner, en la vida real esto debe ser flexible en una manera razonable, permitir el ingreso de nuevos requerimientos, realizar nuevas estimaciones y prolongar la duración del Sprint.

Al inicio del Sprint 2, cuando la empresa Dridco S.A. formaliza el trabajo en conjunto con Maixon y luego de reuniones en las cuales se presenta la forma de trabajar en Maixon, los desarrolladores de Dridco muestran valoración por Scrum como metodología de trabajo ya que ellos la usan con énfasis en sus proyectos.



**Figura 8.1** - Desarrolladores de Dridco en una Scrum meeting



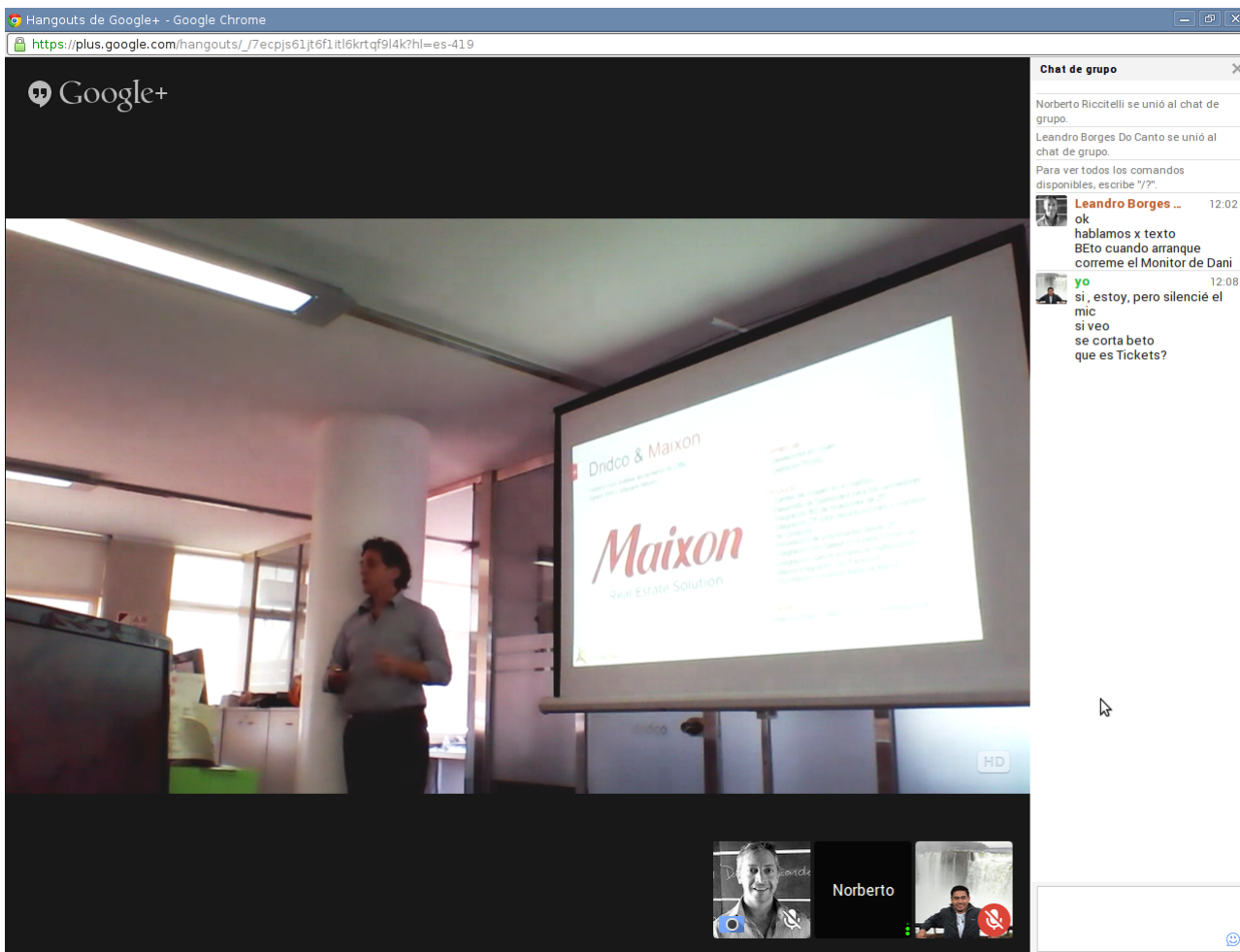


Figura 8.2 - Gerente de Dridco S.A. en presentación de Maixon como nuevo producto.

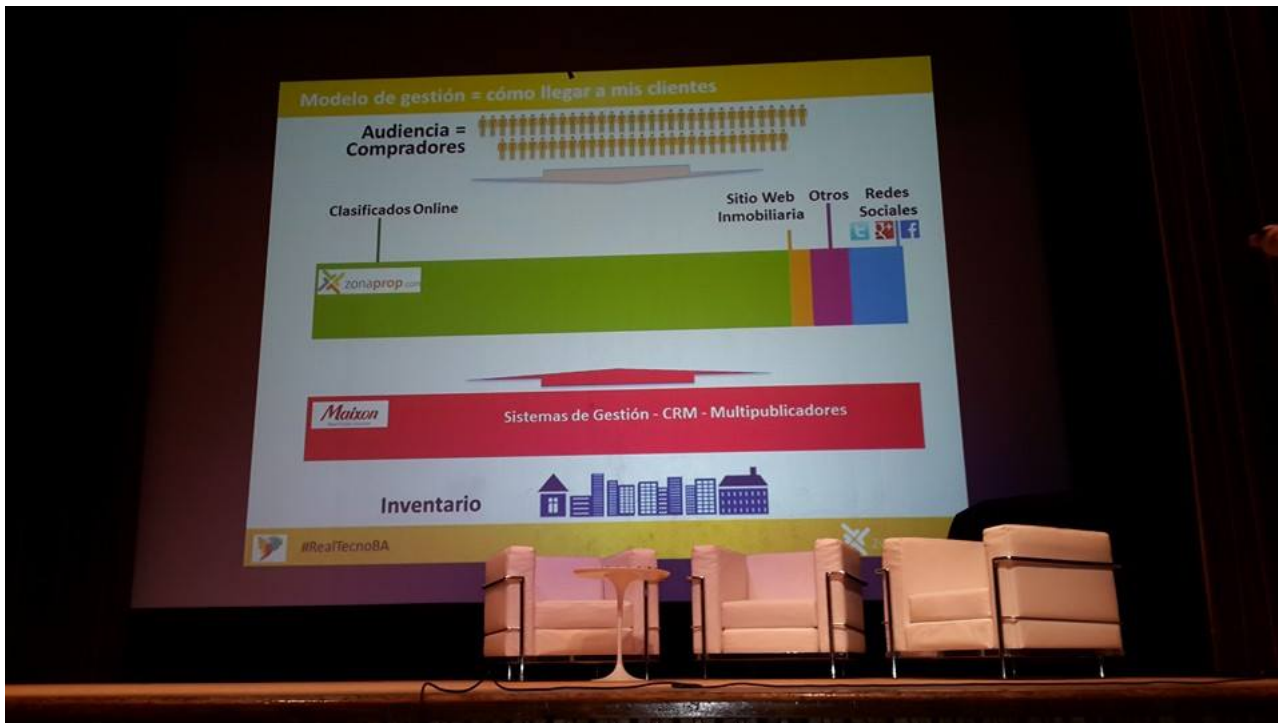
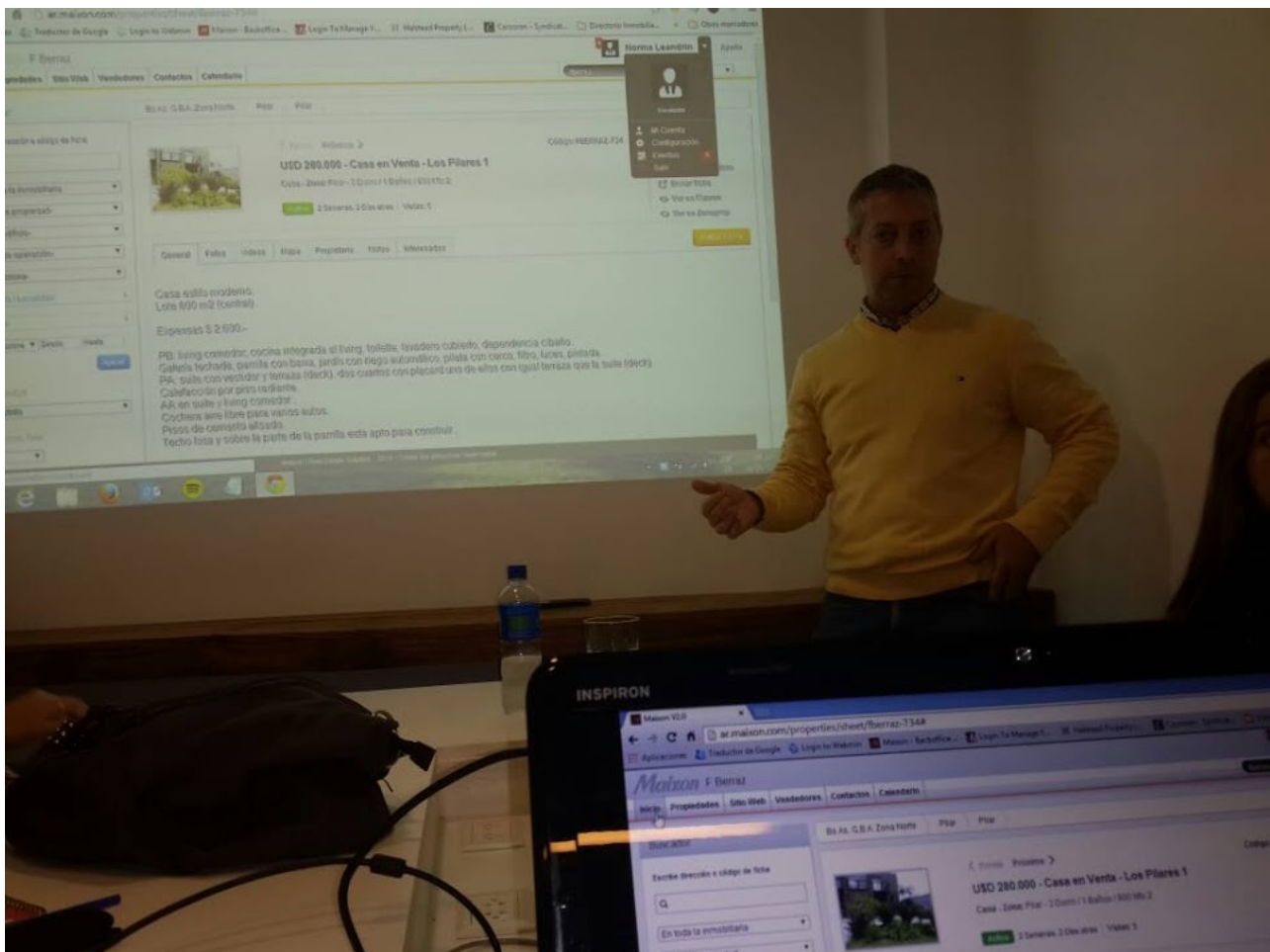


Figura 8.3 - En la actualidad, Maixon en conferencias de Dridco S.A. junto a Zonaprop



**Figura 8.4** - Personal de Marketing realizando exposición de ventas del producto Maixon

También se recibió consejos y experiencia de que es posible trabajar horas extras para cumplir con los objetivos del Sprint en los tiempos estimados, si bien la metodología dice que esto no es correcto, en la práctica es válido aplicar este recurso pero es un indicador de que algo está fallando en la planificación de los Sprints y también en las estimaciones que realizan los miembros del equipo respecto de las tareas.

Otra experiencia adquirida fue de estimar tareas y asignar un tiempo extra de holgura para contemplar posibles situaciones que hagan que el tiempo de desarrollo de la tarea sea superior a lo estimado.

Se logró evolucionar a través de los Sprints, del Sprint 0 que falló, al Sprint 1 en donde se logró completar satisfactoriamente pero con un equipo de un solo programador y luego pasar al Sprint 2 en donde se pudieron coordinar tareas en paralelo entre los miembros del equipo. Aunque sigue siendo la estimación de tareas un factor a mejorar por parte del Equipo.

Existen herramientas CASE para la gestión de tareas en Scrum como KanbanTools o lo que usamos una simple Planilla de Cálculo, pero también es muy común usar una pizarra de posts en la cual se ubican las: tareas por hacer, tareas en progreso y tareas hechas. Cada post es como una historia de usuario y lleva registro de qué miembro del equipo la está desarrollando.

Una experiencia que también se logró con esta metodología es acerca de los riesgos de

estimación de las tareas que implican investigación, por ejemplo el uso de una nueva API sobre la cual hay que estudiar e investigar sobre su funcionamiento, este tipo de tareas debe analizarse si esa investigación que conlleva un riesgo, conviene ponerlo en una tarea aparte del sprint. para que la subdivisión de tareas dependan cada una de si mismo.

Fue gratificante saber que Scrum es una metodología usada comúnmente en desarrollo de aplicaciones web. Dridco S.A y Softlogía S.R.L son empresas de renombre y usan la metodología. Es bueno saber que es adaptativa a cada proyecto de software. El hecho de que Dridco S.A. introdujera sus requerimientos no modificó la forma de trabajo sino que hubo que adaptar el Product Backlog y los Sprints a los nuevos requerimientos y se continuó trabajando de manera normal.

## **9. Anexo**

### **9.1 Historias de Usuario**

## 10. Bibliografía

- [1] SOMMERVILLE, Ian Ingeniería del Software, 7º edición, Madrid, Pearson Educación SA, 2005.
- [2] PRESSMAN, Roger Ingeniería de Software un enfoque práctico, 6º edición, New York, Mc Graw - Hill, 2005.
- [3] Agile Alliance <<http://www.agilealliance.org/>> [consulta: 20 de abril de 2013].
- [4] Manifiesto por el Desarrollo Ágil de Software <<http://agilemanifesto.org/iso/es/>> [consulta: 20 de abril de 2013]
- [5] TAKEUCHI Irotaka y NONAKA Ikujiro, *The New Product Development Game*. Harvard Business Review, Enero de 1986. [consulta: 22 de abril de 2013]. Disponible en: <<http://hbr.org/1986/01/the-new-new-product-development-game/>>
- [6] Scrum.org <<http://www.scrum.org/Resources/What-is-Scrum>> [consulta: 25 de abril de 2013]
- [7] BECK Kent, *Extreme Programming Explained*. 1º Edición, Addison-Wesley, Septiembre de 1999. [consulta: 22 de abril de 2013]. Disponible en: <[http://books.google.com.ar/books/about/Extreme\\_Programming\\_Explained.html?id=G8EL4H4vf7UC&redir\\_esc=y](http://books.google.com.ar/books/about/Extreme_Programming_Explained.html?id=G8EL4H4vf7UC&redir_esc=y)>
- [8] WIKIPEDIA, *Software* [en línea] [consulta: 5 de mayo de 2013] Disponible en <<https://commons.wikimedia.org/wiki/File:ModeloCascadaRealimentado.jpg>>
- [9] WIKIPEDIA, *Proceso Unificado de Rational* [en línea] [consulta: 5 de mayo de 2013] Disponible en <[http://commons.wikimedia.org/wiki/File:Rup\\_espanol.gif](http://commons.wikimedia.org/wiki/File:Rup_espanol.gif)>
- [10] EXTREME PROGRAMMING, *A gentle introduction* [en línea] [consulta: 5 de mayo de 2013] Disponible en <<http://www.extremeprogramming.org/map/project.html>>
- [11] WIKIPEDIA, *Scrum* [en línea] [consulta: 5 de mayo de 2013] Disponible en <[http://commons.wikimedia.org/wiki/File:Ficha\\_scrum.png](http://commons.wikimedia.org/wiki/File:Ficha_scrum.png)>
- [12] DEEMER Peter, BENEFIELD Gabrielle, LARMAN Craig y VODDE Bas, *The Scrum Primer*. Version 1.1 <<http://www.scrumprimer.com/>> <<http://scrumprimer.org/>>
- [13] Maixon, propiedad publicada por Inmobiliaria Galas, código de propiedad: gal-1156 [en línea] [Consulta: 31 de Mayo de 2014]. Disponible en: <http://ar.maixon.com/18532-departamento-venta-2dorm-1banos.html>
- [14] Zonaprop, propiedad publicada por Inmobiliaria Galas en sistema Zonaprop a través de API desde Maixon Manager [13], ID de Propiedad: 7019778 [en línea] [Consulta: 31 de Mayo de 2014]. Disponible en: <http://aviso.zonaprop.com.ar/departamentos-3-ambientes-ramos-mejia-la-matanza-bsas-gba-zona-oeste-7019778>