

**Universidad ORT Uruguay**  
**Facultad de Ingeniería**

**Servicio de búsqueda y gestión de propiedades para  
Inmobiliaria López & Cia.**

Entregado como requisito para la obtención del título de Analista Programador

**Juan González – 197767**

**Bruno Dutra – 179870**

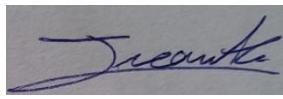
**Tutor: Fernando Martínez**

**2016**

## Declaración de Autoría

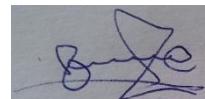
Nosotros, Juan González y Bruno Dutra, declaramos que el trabajo que se presenta en esta obra es de mi propia mano. Puedo asegurar que:

- La obra fue producida en su totalidad mientras realizábamos el proyecto final de carrera.
- Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad.
- Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la obra es enteramente nuestra.
- En la obra, hemos acusado recibo de las ayudas recibidas.
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue contribuido por otros, y qué fue contribuido por nosotros.
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.



Juan González

4.336.649-5



Bruno Dutra

5.137.870-9

## **Agradecimientos**

Agradecemos a nuestro tutor Fernando Martínez por el apoyo brindando en el transcurso del proyecto, su disponibilidad y enseñanzas.

Al cliente Alejandro González por la confianza en el equipo para el trabajo planteado y su tiempo para evacuar consultas y brindar *feedback* sobre el producto.

Por último, un agradecimiento especial a nuestras familias y amigos por su paciencia, motivación y apoyarnos en esta etapa.

## **Abstract**

El presente trabajo consiste en el desarrollo de servicios *web* para la búsqueda y gestión de propiedades, así como para el manejo de usuarios. El mismo, se desprende de la necesidad del cliente de ampliar las funcionalidades de su ya existente aplicación web y brindar un mejor servicio.

Para esto se implementó una API *RESTful* donde a partir de las propiedades que ingresa la inmobiliaria los usuarios pueden realizar distintos tipos de búsqueda con la novedad de incluir su perfil y contexto. A su vez, brinda un detalle de propiedad más profundo focalizándose en la zona del inmueble con los servicios que posee.

Por otro lado, los usuarios autenticados tienen la posibilidad de gestionar propiedades, hacer un seguimiento en el día a día mediante notificaciones.

Se utilizó el lenguaje de desarrollo C# con *framework* .Net y una base de datos *SQL Server* dejando la solución alojada en la nube *Azure*.

Para cumplir con los objetivos planteados se utilizaron las directrices y fundamentos de la metodología ágil *Scrum* con adaptaciones, y así llegar al producto solicitado por el cliente.

## **Palabras clave:**

- Inmobiliaria
- Perfil
- Buscador
- Mapa
- Servicios
- Notificaciones
- Servicio Web
- API
- Venta
- Alquiler
- C#

# Índice

1	Anteproyecto.....	10
1.1	Introducción.....	10
1.2	Presentación del Cliente.....	10
1.3	Presentación del problema .....	10
1.4	Lista de necesidades .....	11
1.5	Análisis estratégico .....	12
1.6	Actores involucrados .....	13
1.7	Objetivos.....	13
1.8	Lista de requerimientos.....	13
1.8.1	Requerimientos Funcionales .....	13
1.8.2	Requerimientos No Funcionales .....	17
1.9	Alcance y limitaciones .....	18
1.10	Estudio de alternativas .....	19
1.10.1	Arquitectura .....	20
1.10.2	Análisis de factibilidad.....	20
1.10.3	Selección de la alternativa .....	21
1.11	Descripción de riesgos .....	21
1.12	Plan de proyecto .....	23
1.12.1	Definición del proceso.....	23
1.13	Compromiso de trabajo .....	31
1.14	Glosario .....	31
2	Desarrollo .....	33
2.1	<i>Sprint 3</i> .....	33
2.1.1	Resumen.....	33
2.1.2	Tareas .....	33
2.1.3	Cronograma de Trabajo con horas reales .....	35
2.1.4	Gráficas.....	36
2.1.5	Seguimiento de riesgos .....	37
2.1.6	Re planificación de horas .....	40
2.1.7	Conclusiones del <i>Sprint</i> .....	43
2.2	<i>Sprint 4</i> .....	43
2.2.1	Resumen.....	43

2.2.2	Tareas .....	44
2.2.3	Cronograma de Trabajo con horas reales .....	45
2.2.4	Gráficas.....	46
2.2.5	Seguimiento de riesgos .....	49
2.2.6	Conclusiones del <i>Sprint</i> .....	51
2.3	<i>Sprint 5</i> .....	52
2.3.1	Resumen.....	52
2.3.2	Tareas .....	52
2.3.3	Cronograma de Trabajo con horas reales .....	54
2.3.4	Gráficas.....	55
2.3.5	Seguimiento de riesgos .....	56
2.3.6	Conclusiones del <i>Sprint</i> .....	59
2.4	<i>Sprint 6</i> .....	59
2.4.1	Resumen.....	59
2.4.2	Tareas .....	60
2.4.3	Cronograma de Trabajo con horas reales .....	61
2.4.4	Gráficas.....	62
2.4.5	Re planificación de horas .....	63
2.4.6	Seguimiento de riesgos .....	64
2.4.7	Conclusiones del <i>Sprint</i> .....	67
2.5	<i>Sprint 7</i> .....	67
2.5.1	Resumen.....	67
2.5.2	Tareas .....	68
2.5.3	Cronograma de Trabajo con horas reales .....	69
2.5.4	Gráficas.....	70
2.5.5	Re planificación de horas .....	70
2.5.6	Seguimiento de riesgos .....	71
2.5.7	Conclusiones del <i>Sprint</i> .....	73
2.6	<i>Sprint 8</i> .....	74
2.6.1	Resumen.....	74
2.6.2	Tareas .....	74
2.6.3	Cronograma de Trabajo con horas reales .....	75
2.6.4	Gráficas.....	76
2.6.5	Seguimiento de riesgos .....	77
2.6.6	Conclusiones del <i>Sprint</i> .....	79
3	Solución implementada.....	80

3.1	<i>API</i> .....	80
3.2	Sitio Web.....	85
4	Cierre Proyecto.....	90
4.1	Gráficas .....	90
4.1.1	Horas reales vs horas planificadas .....	90
4.1.2	Horas <i>testing</i> vs horas desarrollo.....	91
4.1.3	Tareas realizadas vs tareas planificadas vs tareas nuevas.....	91
4.2	Satisfacción del cliente .....	92
4.3	Trabajos a futuro .....	92
4.4	Errores conocidos .....	93
4.5	Lecciones aprendidas .....	93
4.6	Conclusiones de <i>Testing</i> .....	93
4.7	Conclusiones de SQA y SCM .....	94
4.8	Conclusiones del proyecto.....	94
4.9	Diagrama de arquitectura de la solución: .....	95
4.10	Tecnologías utilizadas: .....	96
4.11	Conclusiones de riesgos.....	96
4.11.1	Tabla de riesgos final.....	97
5	Referencias bibliográficas .....	100
6	Bibliografía .....	101
7	Anexos .....	102
7.1	Anexo A: Investigación .....	102
7.1.1	Mercado local.....	102
7.1.2	Mercado internacional.....	103
7.2	Anexo B: Justificación del producto .....	104
7.2.1	Respuesta del equipo luego de reunión con cliente.....	104
7.2.2	Respuesta del cliente .....	104
7.3	Anexo C: Explicación de notificaciones .....	105
7.4	Anexo D: Explicación de búsqueda por perfil.....	106
7.5	Anexo E: Ingreso de propiedades de prueba .....	107
7.6	Anexo F: Explicación de parámetros adicionales en la búsqueda por perfil.....	108
7.7	Anexo G: Casos de uso.....	109
7.8	Anexo H: Casos de prueba.....	120
7.9	Anexo I: Actas de reuniones .....	129
7.9.1	Reuniones con cliente .....	129
7.9.2	Reuniones con tutor.....	138

7.10	Anexo J: Explicación algoritmo de sugerencia de propiedades.....	168
7.11	Anexo K: Control de versionado .....	169
7.11.1	Ejemplo de <i>Checkin</i> .....	169
7.11.2	Comparación de versiones .....	170
7.12	Anexo L: <i>Azure WebJob</i> .....	171
7.13	Anexo M: Servicios de aplicaciones en <i>Azure</i> .....	172
7.14	Anexo N: Pruebas unitarias.....	174
7.14.1	Proyecto de pruebas .....	174
7.14.2	Ejemplo de prueba .....	175
7.14.3	Ejecución de casos de prueba .....	175
7.15	Anexo O: Manual de <i>Deployment</i> .....	176
7.15.1	Configuración de servicio de aplicaciones .....	176
7.15.2	Publicación de las aplicaciones .....	176
7.15.3	Base de Datos.....	177
7.16	Anexo P: Utilización del Id de sistema externo.....	178
7.17	Anexo Q: Investigacion e implementación de <i>APIs</i> de <i>Google</i> .....	179
7.18	Anexo R: Modelo base de datos .....	180
7.19	Anexo S: Diagrama de clases .....	181
7.20	Anexo T: Manual de usuario .....	182
7.20.1	Registro de usuario.....	182
7.20.2	Inicio de sesión .....	182
7.20.3	Búsqueda de propiedades.....	183
7.20.4	Comparación de propiedades .....	184
7.20.5	Detalle de propiedad.....	185
7.20.6	Notificaciones.....	188
7.21	Anexo U: Credenciales de prueba para corrector .....	189
7.22	Anexo V: <i>Google Analytics</i> .....	190
7.23	Anexo W: Evidencias de <i>testing</i> .....	194

# **1 Anteproyecto**

## **1.1 Introducción**

El cliente desea integrar a su aplicación *web* un buscador de inmuebles más sofisticado que los buscadores tradicionales que se encuentran comúnmente.

Se presenta la idea del cliente de realizar un buscador innovador basándose en el perfil del interesado además de darle importancia a la zona de los inmuebles, destacando los servicios que se encuentran allí. Es el objetivo del cliente que los visitantes se puedan autenticar como usuarios para así crear una red de participantes asiduos a la aplicación donde podrán buscar inmuebles como también recibir notificaciones e intercambiar opiniones con otros usuarios.

Con esto se busca ofrecerles a los clientes una solución al momento de buscar una propiedad, ahorrando tiempo, brindando más información, afinando sus búsquedas, lo que hará que se llegue a un mejor resultado. Dentro del mercado local no encontramos soluciones similares por lo que se llegaría a un producto “distinto” dentro de lo que hay actualmente en el país.

Completando este punto se adjunta el anexo B (Justificación del producto) donde se detalla la justificación por parte del cliente para la realización de estos servicios.

## **1.2 Presentación del Cliente**

El cliente será la Inmobiliaria López & Cia ubicada en la zona de Pocitos, Montevideo. La misma se encuentra en el rubro inmobiliario desde el año 1982 y ofrece a los clientes diferentes servicios como Alquiler y Venta de Propiedades, Administración de Gastos Comunes, Administración de propiedades, tasaciones y servicios profesionales.

Actualmente cuenta con la administración de 64 edificios, más de 30 propiedades a la venta y para alquiler, y varios profesionales al servicio de la empresa. A su vez, posee la confianza de distintos proveedores del rubro sanitario, servicio de ascensores, electricistas y otros.

El personal de la empresa es de 12 personas, de las cuales hay 3 encargadas específicamente a la venta y alquiler de propiedades y las demás abocados a la administración de edificios.

## **1.3 Presentación del problema**

Se identifica como principal necesidad de la empresa la integración de un servicio de búsqueda a su Aplicación Web donde sus clientes puedan interactuar.

En una primera instancia se diseñará e implementará un prototipo del sistema que tendrá como principal funcionalidad ofrecer los inmuebles que posee en alquiler y a la venta. La misma se focalizará en la búsqueda según el perfil del interesado. Existirán diversos perfiles a seleccionar aunque también se podrán distinguir otros parámetros enfocados en el contexto del interesado para mejorar la búsqueda. Luego se le brindarán sugerencias en base al perfil ingresado, tomando en cuenta los servicios de la zona y las características de la

propiedad. También se dispondrá de la búsqueda tradicional dando la posibilidad de ingresar las características del inmueble deseado.

Los visitantes se podrán autenticar mediante un *email* o red social. El visitante autenticado quedará registrado como usuario y podrá utilizar las funcionalidades del sistema. Dicho usuario tendrá la posibilidad de realizar consultas, agregar en su cuenta las propiedades de su interés, donde podrá agendar una visita y realizar un seguimiento del estado de la propiedad, por ejemplo recibiendo *mailing* en caso de alguna actualización. A su vez, el sistema brindará un servicio de *newsletter* general pero también el usuario podrá, según su criterio de búsqueda, recibir la información de su interés.

Una vez realizada la búsqueda, al encontrar un inmueble, deberá aparecer la ubicación del mismo junto a los servicios disponibles en sus alrededores. Como se detalla en el anexo 2.1 (Investigación) no se encontraron aplicaciones que brinden servicios similares en el mercado local.

Por otro lado, se agrega el rol de copropietario de edificio el cual tendrá además de los permisos del visitante autenticado, acceso a todo lo referido a la administración de su edificio.

Allí, cada edificio, tendrá su información; llámeselo, reglamento de copropiedad, servicios y otros. También el usuario dispondrá de un área donde visualizará el detalle de los gastos comunes y tendrá la opción de realizar el pago online, mediante la aplicación web o un sistema externo de pagos.

#### 1.4 Lista de necesidades

Actualmente el cliente cuenta con una aplicación *web* donde los visitantes tienen la posibilidad de buscar inmuebles mediante un buscador tradicional. En él puede seleccionar parámetros como barrio, cantidad de dormitorios, cantidad de baños, con o sin garaje, tipo de inmueble y la operación a realizar (alquiler o venta). Una vez realizada la búsqueda se listan las propiedades que cuentan con esas características teniendo la opción de indicar más filtros.

Por otro lado, se tiene la posibilidad de ingresar a la sección “Inmuebles” y seleccionar “alquiler” o “venta” para así, obtener el listado todas las propiedades que existen en la opción seleccionada.

Eligiendo cualquiera de estas dos opciones, al ingresar a cada inmueble se tiene el detalle del mismo con sus fotos.

Ante esto y luego de las reuniones con el cliente surgen las siguientes necesidades:

- Brindar un servicio más personalizado
- Ayudar a los futuros compradores o interesados en alquilar a tener una búsqueda más precisa según su contexto y sus necesidades.
- Darle mayor especificación al cliente y así también poder captar más propietarios que quieran trabajar con nosotros.
- Fidelizar al cliente con la empresa.

- Brindarle más información al usuario acerca de la zona de los inmuebles.

## 1.5 Análisis estratégico

A continuación se detalla la visión y misión presentada por el cliente:

### **Visión:**

Ser una empresa “multiservicio” ofreciendo además de Administración de propiedades, asesoramiento en distintas áreas como ventas, arrendamientos, seguros, refacción, jurídico y notarial, buscando informatizar y adaptarse a las distintas necesidades de los clientes.

### **Misión:**

Somos una empresa con más de 30 años brindando servicio de administración de Propiedad Horizontal (PH) con un equipo multidisciplinario en el área de Montevideo. Brindando a su vez asesoramiento en distintas áreas que refieren a la PH así como asesorando a clientes en la compra, venta o arrendamiento de propiedades.

Por otro lado, por parte del equipo se detalla el siguiente análisis para el proyecto:

### **Análisis FODA: Fortalezas, Oportunidades, Debilidades y Amenazas**

#### Fortalezas:

- El equipo tiene experiencia y conocimiento sobre el negocio.
- El cliente se encuentra a total disposición y compenetrado en el proyecto.
- El cliente es cercano y se lo puede contactar en cualquier momento.
- El equipo cuenta con lugar de trabajo fijo y cómodo.
- El equipo posee las herramientas necesarias para desarrollar el producto así como el apoyo que le brinda la Universidad ORT.
- El cliente busca estar actualizado e innovando en todo lo que refiere a tecnologías y servicios inmobiliarios.

#### Oportunidades:

- Obtener un producto escalable e innovador para el mercado local que pueda interesar a otras inmobiliarias.

#### Debilidades:

- El equipo posee poca experiencia en el desarrollo de proyectos de esta magnitud.
- El equipo es reducido en función del tiempo con el que se cuenta para el desarrollo del producto.

#### Amenazas:

- Por más que se trate de un producto diferente existe mucha competencia en el mercado.

## 1.6 Actores involucrados

A continuación se detallan los actores que estarán involucrados en el sistema:

**Administrador:** es el usuario de la empresa, podrá haber más de uno. Es el encargado de configurar y administrar información de los servicios *web*.

**Visitante:** es el usuario no autenticado, podrá tener acceso solamente a las funcionalidades básicas del servicio, solo podrá realizar búsquedas de inmuebles.

**Usuario Autenticado:** es el usuario que podrá interactuar con el sistema, pudiendo acceder al servicio y las funcionalidades que se integran (llámese: realizar consultas, agregar como favoritas, agendar visitas, insertar comentarios, otras).

## 1.7 Objetivos

De la entrevista con el cliente y la investigación previa se definen los siguientes objetivos para el proyecto:

- Brindar a los usuarios un servicio que permita tener resultados más precisos de propiedades.
- Aumentar el número de usuarios que vuelvan a visitar el sitio web. Ofrecer un servicio personalizado en caso de que el usuario no encuentre lo que está buscando.
- Brindar información que es clave para el cliente al elegir una propiedad. Focalizarse no solo en el inmueble sino en todo su entorno.
- Alcanzar a un mayor número de clientes así como también un público diferente del que cuenta actualmente. Se proyecta tener entre 200 y 500 visitas por semana.
- Generar vínculo entre cliente-empresa y así afianzar la confianza.
- Aumentar ventas y alquileres en al menos un 50%.

Por otro lado, contamos con nuestros objetivos personales de cara a este proyecto. Los mismos son:

- Fortalecer y mejorar nuestros conocimientos en las tecnologías utilizadas.
- Obtener la experiencia de cómo llevar a cabo un proyecto de principio a fin.
- Obtener nuevos conocimientos a partir de las eventualidades que puedan surgir.
- Generar experiencia en el negocio y en el contacto con los clientes.

## 1.8 Lista de requerimientos

En este punto se detallan los requerimientos funcionales y no funcionales que el sistema debe cumplir. Los mismos son los requerimientos deseables para el prototipo a desarrollar. Se definirá en el alcance cuáles serán los requerimientos a ser desarrollados en una primera instancia.

### 1.8.1 Requerimientos Funcionales

- **RF01 Registro de usuario**

Este requerimiento hace referencia a la posibilidad que tendrán los visitantes de poder registrarse en el sistema. Para registrarse como usuario se deberán proporcionar al sistema los siguientes datos: nombre, apellido, correo electrónico, documento de identidad, contraseña y teléfono. El sistema deberá validar que los datos sean correctos y que el usuario no se encuentre registrado con el mismo correo electrónico y/o documento de identidad. Una vez registrado éste podrá acceder de forma completa a todas las funcionalidades del sistema.

- **RF02 Inicio de sesión**

Este requerimiento hace referencia a la posibilidad que tienen los clientes de ingresar al sistema, se solicitará el *email* y contraseña, en caso de no ser correctos se informará el error y se solicitará que se ingresen nuevamente los datos. El *login* de usuario también podrá ser realizado mediante la red social Facebook, en caso de ingresar por primera vez con este medio, se guardarán los datos proporcionados por la red social.

- **RF03 Recuperar contraseña**

Este requerimiento hace referencia a la posibilidad que se tiene de poder recuperar contraseña de un usuario. Se le deberá proporcionar al sistema el correo electrónico del usuario, luego de confirmar si el usuario existe se enviará un e-mail con un link que le dará la posibilidad de ingresar su nueva contraseña.

- **RF04 búsqueda de propiedades**

- **RF04.01 Búsqueda tradicional**

El sistema deberá permitir realizar búsqueda de propiedades mediante los datos de la propiedad deseada, se deberá proporcionar al sistema la siguiente información: tipo de inmueble, tipo de propiedad, tipo de operación, barrio, cantidad de dormitorios, cantidad de baños y si posee garaje o no. Luego el sistema deberá retornar una lista de las propiedades que poseen las características ingresadas.

- **RF04.02 Búsqueda por perfil y parámetros**

El sistema deberá permitir realizar búsqueda de propiedades mediante un perfil determinado. A su vez, para cada perfil, se brindará una lista de parámetros donde el usuario podrá seleccionar, o no, para así realizar una búsqueda más específica. En un principio se definen los siguientes perfiles:

- Empresa.
- Pareja Joven.
- Soltero/a.
- Familia.

En ambos casos se tendrán en cuenta las zonas de los inmuebles con los servicios que se encuentren en las proximidades de los mismos.

También en la búsqueda se dará la posibilidad de ingresar la opción “buscar en zonas aledañas”, con esto se incluirán las propiedades que cuenten con las mismas características en los barrios cercanos al seleccionado.

- **RF05 Detalle de propiedad**

- **RF05.01 Listado de fotos**

Dada una propiedad, el sistema deberá proporcionar una lista de imágenes, los datos de las imágenes serán los siguientes: descripción, orden, url de imagen, y si es principal o no.

- **RF05.02 Detalle de comodidades**

Dada una propiedad, el sistema deberá proporcionar los detalles de las comodidades de la propiedad, los cuales serán: cantidad de baños, tipo de cocina, tipo de living, si contiene portero eléctrico, cierre perimetral, patio, parrillero, metrajes de la superficie, año de construcción y estado general de la propiedad.

- **RF05.03 Información general**

Dada una propiedad, el sistema deberá proporcionar la información general de la propiedad, estos serán: dirección, barrio, ciudad, precio, descripción, si posee gastos comunes y el valor, tipo de propiedad y tipo de operación.

- **RF05.04 Mapa con datos geográficos**

En el detalle de propiedad el sistema deberá proporcionar un mapa con los datos geográficos de la propiedad, estos son: dirección y ubicación geográfica, una lista de servicios de la zona con tipo de servicio y descripción, los servicios se agruparán por tipo mostrando la cantidad disponible. También se podrá filtrar por tipo y solamente se mostrarán en el mapa los servicios del tipo seleccionado.

- **RF06 ABM de propiedades**

Este requerimiento hace referencia a la posibilidad que tendrán los administradores del sistema de poder ingresar, eliminar y modificar las propiedades que se disponen para la venta y/o alquiler. El sistema deberá poder almacenar los datos de la propiedad los cuales serán: id de propiedad, título, descripción, precio, tipo, imágenes (eligiendo una principal), dirección, propiedad destacada, cantidad de baños, tipo de cocina, tipo de living, si contiene portero eléctrico, cierre perimetral, patio, parrillero, metrajes de la superficie, año de construcción y estado general de la propiedad. El sistema deberá validar que el usuario sea un administrador registrado, en caso contrario se retornará un error con el mensaje correspondiente.

- **RF07 Listado de propiedades**

El sistema deberá poder desplegar las propiedades disponibles para alquiler y/o venta, según el criterio de búsqueda establecido, se podrá filtrar por los siguientes criterios: precio, zona, cantidad de dormitorios, tipo de propiedad y por tipo de operación (alquiler

o venta). Para cada propiedad se detallará la información básica: foto principal, título, dirección y precio. A su vez, deberá brindar la posibilidad de ampliar la información de la misma así como también ingresarla como “favorita”.

- **RF08 Solicitud de alta de inmueble**

El sistema deberá poder permitirle a los usuarios enviar una solicitud para ingresar un inmueble en la cartera de la Inmobiliaria. Para esto, deberá ingresar la dirección, la ciudad y el barrio de la propiedad, el precio en que desea vender la misma y regularización BPS. Una vez hecha la solicitud, el usuario queda a la espera de su aprobación.

- **RF09 Listado de solicitudes**

El sistema deberá poder desplegar las solicitudes recibidas. Para cada solicitud se detallará quién la solicita, dirección, ciudad y barrio del inmueble, precio en que se estima la venta y por último si está regularizado en BPS. También se deberá brindar la posibilidad de ampliar la información de la solicitud (texto ingresado por el interesado) donde allí el Administrador aprueba o no la solicitud.

- **RF10 Aprobar solicitud**

Este requerimiento hace referencia a la posibilidad que tendrá el usuario Administrador para aprobar o no una solicitud de ingreso.

Se deberá marcar la opción para aprobarla o rechazarla, y en ambas, se enviará un correo automático al interesado informándole. En caso de aceptarla, luego la empresa se volverá a contactar.

- **RF11 Agregar propiedad como favorita y seguimiento de propiedad**

El sistema brindará la posibilidad a los usuarios autenticados de poder marcar una propiedad como favorita. Es decir, agregar a determinada propiedad como una de su interés. Esto le permitirá hacer un seguimiento día a día de la misma con las novedades que puedan surgir. Por ejemplo si cambió el tipo de operación (alquiler o venta), si cambió el precio, si se vendió/alquiló, etc. Este seguimiento se hará mediante notificaciones que recibirá el usuario.

- **RF12 Notificaciones**

El sistema deberá poder enviar notificaciones a los usuarios según sus intereses marcados (propiedades favoritas, comentarios, sugerencias, pedido de propiedades). Las mismas se anunciarán mediante un envío de mail.

- **RF13 Listado notificaciones**

El sistema deberá brindar la posibilidad de poder listar las notificaciones de cada usuario. Los mismos tendrán la posibilidad de ver en un listado, todas las notificaciones que le

fueron enviadas.

- **RF14 ABM comentarios**

El sistema deberá poder permitirle al usuario el ingreso, modificación y eliminación de comentarios en las distintas propiedades. Los mismos contendrán el nombre del usuario que lo envío, la fecha y el contenido.

- **RF15 Recomendaciones de propiedades (por búsquedas realizadas y coincidencias con otros usuarios)**

El sistema deberá poder realizar recomendaciones de propiedades a los usuarios. Las mismas serán en base a las búsquedas que el usuario realizó y también a las coincidencias en las búsquedas con otros usuarios. Las recomendaciones serán enviadas como notificaciones vía mail.

- **RF16 Realizar pedido de propiedad**

Este requerimiento hace referencia a la posibilidad que tendrán los usuarios autenticados de realizar el pedido de una propiedad. Se deberá enviar las características de la misma para luego realizar el pedido. Una vez realizado, la empresa a través del sistema le brindará al usuario información acerca de propiedades que coincidan con las características buscadas. Podrán ser propiedades que estén en cartera o también de colegas. Los pedidos de propiedades se verán en un listado que siempre estará visible para el usuario que realizó el pedido. Los mismos se podrán cancelar y así no le llegará más información acerca de ese pedido.

- **RF17 Agendar visita**

Este requerimiento hace referencia a la posibilidad que tienen los usuarios autenticados de poder agendar una visita a una propiedad. Dada una propiedad se podrá agendar una visita proporcionando la fecha y hora disponible quedando a la espera de la confirmación del usuario Administrador.

- **RF18 Cerrar sesión**

El sistema deberá brindar al usuario autenticado la posibilidad de cerrar la sesión del mismo.

## 1.8.2 Requerimientos No Funcionales

- Será un servicio que se pueda integrar con la página web de López y Cia.
- Para el futuro se pretende que la aplicación se pueda acceder desde dispositivos móviles.
- Para el manejo de base de datos se utilizará el motor SQL Server 2014.
- La aplicación deberá funcionar a través de un navegador web en cualquier sistema operativo.

- La aplicación restringirá el acceso a las distintas funcionalidades dependiendo si el usuario es visitante o usuario autenticado.
- La comunicación entre los distintos equipos involucrados en la aplicación se realizará mediante protocolos estándares de Internet como son TCP/IP, HTTP.
- Escalabilidad: se debe poder escalar a futuros requerimientos, como la integración con otros sistemas.
- Mantenibilidad: la programación será modular, permitiendo agregar nuevos requerimientos sin afectar las funcionalidades realizadas en las primeras versiones.

## 1.9 Alcance y limitaciones

A continuación se detalla el alcance definido para el desarrollo del producto solicitado:

En el prototipo inicial se desarrollarán todos los requerimientos definidos relativos a la búsqueda y gestión de propiedades. A continuación se detalla la pila de producto definida para el proyecto:

- RF04 búsqueda de propiedades.
- RF04.01 Búsqueda tradicional.
- RF04.02 Búsqueda por perfil y parámetros.
- RF07 Listado de propiedades.
- RF05 Detalle de propiedad.
- RF05.01 Listado de fotos.
- RF05.02 Detalle de comodidades.
- RF05.03 Información general.
- RF05.04 Mapa con datos geográficos.
- RF01 registro de usuario.
- RF02 inicio de sesión.
- RF03 Recuperar contraseña.
- RF18 Cerrar sesión.
- RF06 ABM de propiedades.
- RF14 ABM comentarios.

- RF11 Agregar propiedad como favorita y seguimiento de propiedad.
- RF12 Notificaciones.
- RF13 Listado notificaciones.
- RF15 Recomendaciones de propiedades (por búsquedas realizadas y coincidencias con otros usuarios).
- RF16 Realizar pedido de propiedad.
- RF08 Solicitud de alta de inmueble.
- RF09 Listado de solicitudes.
- RF10 Aprobar solicitud.
- RF17 Agendar visita.

Con respecto a las necesidades del cliente en cuanto al rol de copropietario para la sección de administración de edificios, las mismas no serán implementadas para esta etapa del proyecto. Esto se debe a que el tiempo disponible no es suficiente.

Respecto a las limitaciones, la aplicación web ya está definida por un tercero por lo que está fuera de los objetivos del producto a desarrollar por el equipo.

## 1.10 Estudio de alternativas

Para el desarrollo del proyecto se tomaron en cuenta las siguientes tecnologías:

### Lenguajes de programación:

#### *JavaEE*

Es una plataforma de programación para desarrollar y ejecutar software en lenguaje Java, se utiliza para implementar software con enfoque empresarial. Posee librerías necesarias para la implementación de *web services*.

#### *Microsoft .NET*

Es un *framework* de desarrollo que permite un rápido desarrollo de aplicaciones. Posee las herramientas necesarias para el desarrollo de aplicaciones Web. En las librerías está incluida la tecnología Web API que es un *framework* que aprovecha las capacidades *RESTful* para poder hacer servicios HTTP, a su vez esta se basa en el patrón MVC lo que facilita el desarrollo de un servicio Web.

### Base de Datos

#### *MySql*

Sistema de gestión de base de datos relacional desarrollado por *Oracle*. Puede ser conectado con cualquier lenguaje de programación, pero la integración con .Net no es óptima ya que el conector limita ciertas operaciones. A su vez el equipo no posee experiencia en esta herramienta.

#### *SQL Server*

Sistema de manejo de base de datos de Microsoft. Herramienta que permite una conexión óptima con sistemas .Net. El equipo de desarrollo posee el conocimiento necesario para implementar con esta herramienta.

##### **1.10.1 Arquitectura**

Para el desarrollo del servicio web se utilizará el *framework Web API* que se basa en la arquitectura MVC, esto nos permite un manejo más fácil de la estructura de nuestro servicio.

A su vez, el acceso a los datos se realizará mediante *Entity Framework*. Es un ORM (mapeo de objetos relacional). Esto nos permite tener la aplicación conectada con la base de datos teniendo las mismas estructuras.

Por otro lado los servidores de aplicación y base de datos estarán alojados en la nube de la plataforma *Azure*.

##### **1.10.2 Análisis de factibilidad**

Con respecto a la factibilidad del proyecto, según un estudio del cliente, una investigación por parte del equipo y reuniones en conjunto se determina que el mismo es viable.

Creemos que el proyecto final abarcará las necesidades actuales del cliente, siendo escalable e insertándose en el mercado inmobiliario tanto local como internacional.

###### **1.10.2.1 Operativa**

Consideramos que la factibilidad operativa del proyecto es alta ya que el servicio se puede integrar con cualquier tipo aplicación independiente del lenguaje.

###### **1.10.2.2 Técnica**

No se han encontrado inconvenientes con respecto a los aspectos técnicos ya que el equipo posee las herramientas aptas para llevar a cabo un proyecto de esta magnitud. A su vez se cuenta con una plataforma en la nube que será suficiente para el desarrollo del producto.

###### **1.10.2.3 Legal**

La publicación de todas las propiedades, con fotos y su información, en el servicio están autorizadas previamente por los propietarios de las mismas. A su vez cualquier tipo de información brindada por colegas será también con el aval de ellos y mencionando su nombre.

Respecto al producto final, se acordó que el producto será compartido entre el cliente y el equipo de desarrollo.

#### 1.10.2.4 Económica

Con respecto a la factibilidad económica, al ser el proyecto de final de carrera, no existirán costos que el cliente deba asumir para el desarrollo del producto.

En lo que refiere a las herramientas utilizadas, así como también licencias que se precisan se utilizarán los beneficios que brinda la Universidad ORT para estudiantes.

En ambos casos, una vez culminado el proyecto en caso de que el cliente desee seguir vinculado al equipo se deberán replantear los términos de la vinculación entre ambos para cubrir los costos necesarios.

#### 1.10.3 Selección de la alternativa

De las herramientas ya mencionadas en el punto 1.11 (Estudio de Alternativas), se decidió por parte del equipo de desarrollo la siguiente selección:

Se utilizará el *IDE Visual Studio 2015 Community*. Utilizando el *framework .Net 4.5*, lenguaje C#, ya que el equipo posee experiencia en esta tecnología, y a su vez por el soporte y facilidad que brinda la misma debido a la alta gama de librerías incluidas.

Para el manejo de la base de datos se utilizará el motor *SQL Server 2014*. La cantidad de datos que se manejará será de tamaño medio y éste nos permite un rápido acceso a la información soportando correctamente la misma.

La integración de *Visual Studio 2015* con *SQL server 2014* también fue tenido en cuenta para la elección de éstos.

### 1.11 Descripción de riesgos

En esta etapa se detallan los riesgos que pueden llegar a haber durante el desarrollo del proyecto. En el mismo, se observan los riesgos que pueden ocurrir, el tipo de éste y su probabilidad. Una vez identificado el riesgo se elabora un plan de contingencia previniendo por si el riesgo llega a ocurrir.

Riesgo	Tipo de Riesgo	Probabilidad	Descripción	Plan de contingencia
Rotación del personal	Proyecto	Baja	Alguno de los integrantes del equipo abandone el proyecto.	Se evaluará si se puede recortar el alcance para que alguno de los integrantes pueda continuar.
Cambio de requerimientos	Proyecto y producto	Alta	Habrá más cambios en los requerimientos que lo anticipado.	Prever los cambios de requerimientos en una etapa temprana del proyecto. En caso de surgir cambios durante el mismo se renegociará con el

				cliente.
Cambio de tecnología	Negocio	Baja	El cliente pide un cambio de tecnología para el desarrollo.	Renegociar el contrato.
Reducción del presupuesto	Proyecto	Baja	Problemas financieros en la organización fuerzan a reducir el presupuesto del proyecto.	Que el alcance cuente con lo mínimo indispensable.
Problemas legales	Producto	Baja	Problemas legales en la viabilidad del producto que afectan al mismo o a alguna funcionalidad.	Cambiar la funcionalidad tratando de evitar el problema y a su vez el que cumpla su propósito, en la medida de lo posible.
Nuevas tecnologías	Proyecto y producto	Media	Aparición de nuevas tecnologías más eficientes que podrían hacer un mejor producto.	Evaluar el proyecto junto con el cliente para saber si esas nuevas tecnologías mejorarían sustancialmente el mismo.
Incumplimiento de plazos	Proyecto	Medio	Incumplimiento de los plazos previstos antes de comenzar el proyecto.	Renegociar calendario con el cliente.
Incumplimiento de integración	Proyecto	Medio	Incumplimiento en la integración con el sistema desde la página web diseñada por terceros.	Proporcionar las firmas de métodos y la documentación técnica lo antes posible. En caso de no cumplirse la integración el equipo tendrá que tener ejemplos de integración para entregar en el proyecto.
Integridad de datos del sistema.	Producto	Medio	Pérdida de datos una vez puesto en producción.	En lo que refiere a datos cuando el sistema se suba a un ambiente de producción, se deberá hacer una política de respaldo.
Integración del servicio con la página web.	Producto	Medio	En caso de darse el riesgo “Incumplimiento de integración” una vez finalizado el proyecto el	El cliente podrá contratar al equipo u otro para que integre su sitio web actual con

			cliente quedará con una solución web y por otro lado la Api.	la Api. Renegociar con el cliente.
--	--	--	--	------------------------------------

Tabla 1 Descripción de riesgos.

## 1.12 Plan de proyecto

En esta sección se definen las técnicas dirigidas al proceso de desarrollo, se hace foco en las metodologías a utilizar, las iteraciones y estimaciones de tiempos, el control de calidad y la administración de la configuración.

### 1.12.1 Definición del proceso

#### 1.12.1.1 Metodología

Se seguirán directrices de la metodología ágil *Scrum*, es un *framework* de desarrollo ágil de software. El trabajo es estructurado en ciclos de trabajos llamados iteraciones o *sprints*.

Se realizará una adaptación a la metodología original ya que el equipo de desarrollo es reducido. Se tomarán en cuenta las características que se ajusten al tipo de proyecto. Los *sprints* o iteraciones fueron definidos por la Universidad ORT, se dividirán los requerimientos a desarrollar y se especificarán las tareas para poder cumplir dichos requerimientos en cada *sprint*.

#### 1.12.1.2 – Ciclo de vida elegido

Según la metodología elegida se define un ciclo de vida que a partir de una pila de producto (lista de requerimientos), se debe planear cuáles serán desarrollados en cada iteración. El proceso se puede ver claramente en la imagen a continuación.

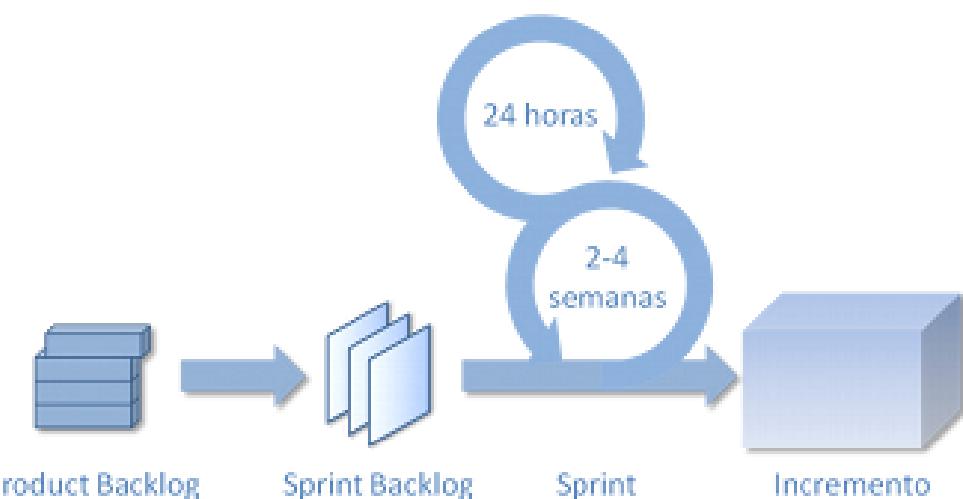


Ilustración 1 Ciclo de vida.

Las fechas de entrega de cada *sprint* fueron definidas por la Universidad ORT, cada iteración durará aproximadamente 15 días.

#### 1.12.1.3 – Incrementos o iteraciones definidas

El proyecto será dividido en 8 iteraciones de 15 días cada una aproximadamente. Los *sprints* 1 y 2 serán dedicados a la investigación y elaboración del Anteproyecto. Luego se determinan 5 *sprints* los cuales serán destinados al desarrollo de los requerimientos, estas serán entregadas al tutor y al cliente como avance. Por último se realizará el *sprint* N° 8 donde se elaborará la documentación final y correcciones.

A continuación se detallan las fechas establecidas para las iteraciones:

*Sprint 1 y 2* – Finaliza el 05 de mayo de 2016.

*Sprint 3* - Comienzo 06 de mayo y finalización 30 de mayo de 2016.

*Sprint 4* – Comienzo 31 de mayo y finalización 16 de junio de 2016.

*Sprint 5* - Comienzo 17 de junio y finalización 7 de julio de 2016.

*Sprint 6* - Comienzo 8 de julio y finalización 28 de julio de 2016.

*Sprint 7* - Comienzo 29 de julio y finalización 20 de agosto de 2016.

*Sprint 8* - Comienzo 21 de agosto y finalización 28 de setiembre de 2016.

Se establece en el – Cronograma de trabajo las actividades a realizar en cada *sprint*.

#### 1.12.1.4 – Integrantes y roles

El grupo de proyecto se compone por Juan Andrés González y Bruno Dutra. Al ser un grupo reducido, los mismos serán los encargados de desempeñar los roles de implementación y análisis definidos por la metodología *Scrum*. Estos son:

- Mediador con el cliente.
- Analista.
- Programador.
- *Tester*.
- *Scrum Master*.

A su vez el *Product Owner* será el Sr. Alejandro González, socio propietario de Inmobiliaria López y Cia. Es el que conoce el negocio y el que proporcionará las necesidades.

#### 1.12.1.5 – Descripción y selección de herramientas

Se utilizará el IDE Visual Studio 2015 con lenguaje C# .Net, ya que el equipo posee experiencia en esta tecnología, y a su vez por el soporte y facilidad que brinda la misma debido a la alta gama de librerías incluidas.

Por otro lado, permite incluir otras tecnologías como *jQuery*, *bootstrap* y otros para el manejo de las vistas.

Para el manejo de la base de datos se utilizará el motor *SQL Server* 2014. La cantidad de datos que se manejará será de tamaño medio y éste nos permite un rápido acceso a la información soportando correctamente la misma.

La integración de *Visual Studio 2015* con *SQL server 2014* también fue tenido en cuenta para la elección de éstos.

Utilizaremos el portal *Azure* como servidor de aplicaciones y de base de datos.

Por último, se utilizó la herramienta *Google Drive* para almacenar la documentación y archivos para el seguimiento del proyecto.

#### 1.12.1.6 Plan de SQA

En esta sección se define el marco de estándares y procesos de organización que conducirán a software de mejor calidad. La gestión de la calidad se encarga de garantizar que los resultados de proyecto estén en conformidad con los estándares aplicables.

##### 1.12.1.6.1 Estándares definidos y convenciones

###### **Requerimientos:**

Para la especificación de los requerimientos nos basamos en las directrices dadas por el estándar “*IEEE Recomended Practice for Software Requirement Especifications ANSI/IEEE 830 1998*”. [1]

###### **Programación:**

Se elige seguir como referencia de programación las convenciones de código para C# definidos por la MSDN (*Microsoft Developer Network*). Estas definen reglas para la nomenclatura, diseño, comentarios, lenguaje, variables y tipos de datos, manejo de excepciones y seguridad. [2]

###### **Documentación:**

Para la documentación de proyecto se utilizarán las normas del documento 302 para presentación de trabajos de final de carrera de universidad ORT. [3]

##### 1.12.1.6.2 – Plan de *testing*

Dado a que el sistema será desarrollado en .Net, se realizarán pruebas unitarias con *NUnit* para probar las principales funcionalidades del servicio.

A su vez se realizará *testing* funcional, se realizarán casos de prueba los cuales serán ejecutados por el grupo de trabajo. Una vez finalizado el desarrollo de las funcionalidades, los integrantes deberán ejecutar los casos de prueba de las funcionalidades que no haya desarrollado, ya que muchas veces la percepción no es la misma.

También se implementará *Shake Out testing* a las funcionalidades principales del sistema una vez que una o más funcionalidades sean liberadas para verificar la integridad del sistema y así poder continuar con el *testing* correspondiente.

### 1.12.1.7 – Plan de SCM

El equipo será responsable de la gestión de los cambios utilizando su juicio y experiencia, por lo tanto, si se quiere modificar un elemento estable de la línea base del proyecto, se debe llegar a un acuerdo en común, analizando el impacto de dicho cambio y definir la prioridad del mismo.

#### 1.12.1.7.1 – Control de versionado

Para el versionado de código y organización de software se utilizará el servidor *Team Foundation Service* de Microsoft. Esta herramienta permite que cada integrante del equipo posea una copia completa del repositorio de manera local, también permite hacer un seguimiento del historial de cambios.

Para la base de datos se utilizará una instancia en común alojada en el portal *Azure*, también se podrá mantener una instancia local de la misma para el manejo independiente evitando conflictos y colisiones de datos. Para los cambios de estructuras o ingreso de datos, cada integrante deberá generar un *script* con las modificaciones. Se elige alojar dichos *scripts* en *Google Drive* y ejecutarlos cuando sea necesario.

#### 1.12.1.8 – Plan de capacitación

Para la planificación del proyecto, el equipo asistió al Taller de proyecto dictado en Universidad ORT como parte de la carrera Analista Programador, este curso nos brinda las herramientas para los procesos de gestión de proyectos y la utilización de metodologías ágiles en el desarrollo de software.

A su vez, se concurrió al taller de *Azure* que se desarrolló en la Universidad ORT donde a base de ejercicios prácticos y demostraciones se obtuvo una introducción en el tema.

Luego de terminado el proyecto se realizará un manual para el usuario Administrador y a su vez se generará una guía técnica del servicio desarrollado.

#### 1.12.1.9 – Cronograma de trabajo

	Tarea	Estimación (horas)	Inicio	Fin
1	Desarrollo	<b>630</b>	06/05/2016	28/09/2016
1.1	<b>Sprint 3</b>	<b>140</b>	06/05/2016	30/05/2016
1.1.1	<b>Estudio de herramientas y tecnologías</b>	<b>10</b>		
1.1.2	<b>Configuración de ambiente de desarrollo</b>	<b>9</b>		
1.1.2.1	Configuración servidor de base de datos	4		
1.1.2.2	Configuración repositorio de código	3		
1.1.2.3	Configuración de servidor de la aplicación	2		

1.1.3	<b>Primera versión diagrama de casos de uso</b>	<b>3</b>		
1.1.4	<b>Primera versión diagrama de clases</b>	<b>4</b>		
1.1.5	<b>Primera versión MER</b>	<b>4</b>		
1.1.6	<b>Implementación controladoras y métodos a desarrollar de la API</b>	<b>8</b>		
1.1.7	<b>RF04: Búsqueda de propiedades</b>	<b>54</b>	13/05/2016	23/05/2016
1.1.7.1	Base de datos	10		
1.1.7.1.1	Creación de tablas	6		
1.1.7.1.2	Ingreso de datos	4		
1.1.7.2	Desarrollo de lógica de búsqueda	26		
1.1.7.2.1	Desarrollo de lógica de búsqueda tradicional	10		
1.1.7.2.2	Desarrollo de lógica de búsqueda por perfil	16		
1.1.7.3	<i>Web service</i>	10		
1.1.7.4	<i>Testing</i>	8		
1.1.8	<b>RF07: Listado de propiedades</b>	<b>30</b>	24/05/2016	29/05/2016
1.1.8.1	Desarrollo de lógica de Listado de propiedades	12		
1.1.8.2	<i>Web service</i>	10		
1.1.8.3	<i>Testing</i>	8		
1.1.9	<b>Reuniones grupo</b>	<b>6</b>		
1.1.10	<b>Reuniones con tutor</b>	<b>2</b>		
1.1.11	<b>Documentación Sprint</b>	<b>10</b>	29/05/2016	30/05/2016
1.2	<b>Sprint 4</b>	<b>90</b>	31/05/2016	16/06/2016
1.2.1	<b>RF05: Detalle de Propiedades</b>	<b>76</b>	31/05/2016	12/06/2016
1.2.1.1	RF05.01 Listado de fotos	12		
1.2.1.1.1	Desarrollo lógica de listado de fotos	6		
1.2.1.1.2	<i>Web service</i>	3		
1.2.1.1.3	<i>Testing</i>	3		
1.2.1.2	RF05.02 Detalle de comodidades	12		
1.2.1.2.1	Desarrollo lógica de detalle de comodidades	6		
1.2.1.2.2	<i>Web service</i>	3		
1.2.1.2.3	<i>Testing</i>	3		
1.2.1.3	RF05.03 Información general	12		
1.2.1.3.1	Desarrollo lógica de detalle de información general	6		
1.2.1.3.2	<i>Web service</i>	3		
1.2.1.3.3	<i>Testing</i>	3		
1.2.1.4	RF05.04 Mapa con datos	40		

	geográficos			
1.2.1.4.1	Desarrollo lógica de detalle de información general	22		
1.2.1.4.2	Desarrollo de conexión con API de Google Places	8		
1.2.1.4.3	<i>Web service</i>	3		
1.2.1.4.4	<i>Testing</i>	7		
1.2.2	<b>Reuniones grupo</b>	<b>2</b>		
1.2.3	<b>Reuniones con tutor</b>	<b>2</b>		
1.2.4	<b>Documentación Sprint</b>	<b>10</b>	12/06/2016	16/06/2016
1.3	<b>Sprint 5</b>	120	17/06/2016	07/07/2016
1.3.1	<b>RF01 Registro de usuario</b>	<b>23</b>	17/06/2016	20/06/2016
1.3.1.1	Creación de tablas	4		
1.3.1.2	Ingreso de datos de prueba	2		
1.3.1.3	Desarrollo de lógica de registro de usuario	10		
1.3.1.4	<i>Web Service</i>	4		
1.3.1.5	<i>Testing</i>	3		
1.3.2	<b>RF02 Inicio de sesión</b>	<b>15</b>	21/06/2016	25/06/2016
1.3.2.1	Desarrollo de lógica de inicio de sesión	8		
1.3.2.2	<i>Web Service</i>	4		
1.3.2.3	<i>Testing</i>	3		
1.3.3	<b>RF03 Recuperar contraseña</b>	<b>15</b>	25/06/2016	27/06/2016
1.3.3.1	Desarrollo de lógica de recuperar contraseña	8		
1.3.3.2	<i>Web service</i>	4		
1.3.3.3	<i>Testing</i>	3		
1.3.4	<b>RF18 Cerrar sesión</b>	<b>15</b>	27/06/2016	30/06/2016
1.3.4.1	Desarrollo de lógica de cerrar sesión	8		
1.3.4.2	<i>Web Service</i>	4		
1.3.4.3	<i>Testing</i>	3		
1.3.5	<b>RF06 ABM de propiedades</b>	<b>20</b>	01/07/2016	03/07/2016
1.3.5.1	Desarrollo lógica de ABM propiedades	13		
1.3.5.2	<i>Web service</i>	4		
1.3.5.3	<i>Testing</i>	3		
1.3.6	<b>RF14 ABM comentarios</b>	<b>20</b>	03/07/2016	06/07/2016
1.3.6.1	Desarrollo lógica de ABM comentarios	13		
1.3.6.2	<i>Web service</i>	4		
1.3.6.3	<i>Testing</i>	3		

1.3.7	<b>Reuniones grupo</b>	<b>1</b>		
1.3.8	<b>Reuniones con tutor</b>	<b>2</b>		
1.3.9	<b>Documentación Sprint</b>	<b>9</b>	06/07/2016	07/07/2016
1.4	<b>Sprint 6</b>	<b>120</b>	08/07/2016	28/07/2016
1.4.1	<b>RF11 Agregar propiedad como favorita y seguimiento de propiedad</b>	<b>24</b>	08/07/2016	10/07/2016
1.4.1.1	Creación de tablas	5		
1.4.1.2	Ingreso de Datos	3		
1.4.1.3	Desarrollo lógica de propiedad favorita	9		
1.4.1.4	<i>Web service</i>	4		
1.4.1.5	<i>Testing</i>	3		
1.4.2	<b>RF12 Notificaciones</b>	<b>24</b>	11/07/2016	16/07/2016
1.4.2.1	Creación de tablas	5		
1.4.2.2	Ingreso de datos	3		
1.4.2.3	Desarrollo lógica notificaciones	9		
1.4.2.4	<i>Web service</i>	4		
1.4.2.5	<i>Testing</i>	3		
1.4.3	<b>RF13 Listado notificaciones</b>	<b>16</b>	16/07/2016	18/07/2016
1.4.3.1	Desarrollo lógica listado notificaciones	9		
1.4.3.2	<i>Web service</i>	4		
1.4.3.3	<i>Testing</i>	3		
1.4.4	<b>RF15 Recomendaciones de propiedades</b>	<b>24</b>	18/07/2016	23/07/2016
1.4.4.1	Desarrollo lógica recomendaciones	14		
1.4.4.2	<i>Web service</i>	5		
1.4.4.3	<i>Testing</i>	5		
1.4.5	<b>RF16 Realizar pedido de propiedad</b>	<b>16</b>	24/07/2016	26/07/2016
1.4.5.1	Desarrollo lógica pedido de propiedad	9		
1.4.5.2	<i>Web service</i>	4		
1.4.5.3	<i>Testing</i>	3		
1.4.6	<b>Reuniones grupo</b>	<b>4</b>		
1.4.7	<b>Reuniones con tutor</b>	<b>2</b>		
1.4.8	<b>Documentación Sprint</b>	<b>10</b>	26/07/2016	28/07/2016
1.5	<b>Sprint 7</b>	<b>160</b>	29/07/2016	28/08/2016
1.5.1	<b>RF08 Solicitud alta inmueble</b>	<b>15</b>	29/07/2016	31/07/2016
1.5.1.1	Desarrollo solicitud alta inmueble	8		
1.5.1.2	<i>Web service</i>	4		
1.5.1.3	<i>Testing</i>	3		

1.5.2	<b>RF09 Listado solicitudes</b>	<b>18</b>	31/07/2016	04/08/2016
1.5.2.1	Desarrollo lógica listado solicitudes	11		
1.5.2.2	<i>Web service</i>	4		
1.5.2.3	<i>Testing</i>	3		
1.5.3	<b>RF10 Aprobar solicitud</b>	<b>12</b>	05/08/2016	07/08/2016
1.5.3.1	Desarrollo lógica aprobar solicitud	6		
1.5.3.2	<i>Web service</i>	4		
1.5.3.3	<i>Testing</i>	2		
1.5.4	<b>RF17 Agendar visita</b>	<b>18</b>	07/08/2016	11/08/2016
1.5.4.1	Desarrollo lógica agendar visita	11		
1.5.4.2	<i>Web service</i>	4		
1.5.4.3	<i>Testing</i>	3		
1.5.5	<b>Reuniones grupo</b>	<b>4</b>		
1.5.6	<b>Reuniones con tutor</b>	<b>3</b>		
1.5.7	<b>Documentación Sprint</b>	<b>10</b>	12/08/2016	14/08/2016
1.5.8	<b>Faltantes y arreglos de todo el producto</b>	<b>80</b>	15/08/2016	28/08/2016
2	<b>Sprint 8</b>	<b>140</b>	29/08/2016	28/09/2016
2.1	<b>Preparación entrega final</b>	<b>40</b>		
2.2	<b>Documentación proyecto</b>	<b>60</b>		
2.3	<b>Reuniones grupo</b>	<b>10</b>		
2.4	<b>Reuniones tutor</b>	<b>3</b>		
2.5	<b>Arreglos finales - Entrega</b>	<b>27</b>		

Tabla 2 Cronograma de trabajo.

## Diagrama de GANTT



Ilustración 2 Diagrama de GANTT.

### 1.13 Compromiso de trabajo

El equipo de desarrollo se compromete a realizar lo establecido en el alcance. Es decir, cumplir con todos los requerimientos marcados.

A su vez, queda determinado la cantidad de horas de trabajo por semana que se dedicará al proyecto, las cuales se detallan a continuación:

Lunes a viernes: 2 horas por día por integrante.

Sábado y domingo: 5 horas por día por integrante.

Esta estimación hace que entre los dos integrantes del equipo se obtengan 40 horas semanales. La misma, podrá variar ante cualquier eventualidad siendo lo mínimo establecido para el desarrollo del producto.

### 1.14 Glosario

ABM: alta baja y modificación de datos.

API: *Application Programming Interface*, es un conjunto de funciones y procedimientos que cumplen una o muchas funciones con el fin de ser utilizadas por otro software.

Framework: es un entorno o ambiente de trabajo para desarrollo.

**IDE:** *Integrated Development Environment* (entorno de desarrollo integrado), consiste de un editor de código fuente, herramientas de construcción automáticas y un depurador.

**Scrum:** es un marco de desarrollo ágil que se destaca por adoptar una estrategia de desarrollo incremental e iterativo.

**Sprint:** es un período de tiempo en el cual se desarrolla el trabajo en sí, el mismo consta entre 2 y 3 semanas. Es la unidad básica de desarrollo en *Scrum*.

**SQA:** *Software Quality Assurance*., son un conjunto de actividades sistemáticas que aseguran que los procesos y productos del software cumplan con los requerimientos, estándares, y procedimientos.

**SCM:** *Software Configuration Management*, es una especialización de la gestión de configuración a todas las actividades en el sector del desarrollo de software.

**Servicio Web:** tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

## 2 Desarrollo

### 2.1 Sprint 3

#### 2.1.1 Resumen

Este *sprint* fue dedicado a la investigación de herramientas para el proyecto, configuraciones y a la implementación de los requerimientos relacionados a la búsqueda de propiedades.

En el comienzo del mismo surgieron algunos contratiempos los cuales retrasaron lo planteado en una primera instancia. A continuación el detalle de los mismos:

- Al comienzo del Sprint tuvimos inconvenientes con una de las máquinas personales del equipo lo que atrasó unos días el desarrollo del mismo.
- Al ser un proyecto que utiliza herramientas que en un principio se pensaba no resultarían ser tan complejas, como es Web Api para implementar el servicio REST, también la integración con la API de Google y Entity Framework, el tiempo de estudio y pruebas de las mismas resultó ser mayor al estimado.
- Durante el desarrollo, la integración entre la base de datos y el sistema llevó más tiempo de lo esperado debido a los errores que surgieron durante los cambios en la misma.
- Por más que se sabe el enfoque de la aplicación, al realizar el análisis de la misma y entrar en algunos detalles, estos hicieron que surgieran interrogantes que en un principio no se habían tenido en cuenta. Para esto, hubo que reunirse con el cliente y así, evacuar dudas respecto a qué parámetros se iban a tomar en cuenta para cada perfil en la búsqueda y como se haría la separación entre la búsqueda tradicional y la búsqueda por perfil.

Luego de estos contratiempos, los cuales fueron solucionados, se siguió con lo planteado en el cronograma aunque no se pudo terminar con la totalidad de las tareas establecidas. Es por esta razón, que las tareas pendientes de este *sprint* pasarán al próximo.

#### 2.1.2 Tareas

A continuación se muestra el detalle de las tareas planteadas para este *sprint* y el estado de cada una.

##### 2.1.2.1 Tareas generales

- Estudio de herramientas: *Web Api*, *Entity Framework*, API de *Google Places*, Base de Datos. –culminada.
- Configuración cuenta Portal *Azure*. – culminada.
- Creación y configuración de servidor de base de datos. – culminada.
- Creación y configuración del servidor de aplicaciones en la nube. – culminada.
- Primera versión de casos de uso. – culminada.
- Primera versión de diagrama de clases. – culminada.
- Primera versión de MER. – culminada.

### 2.1.2.2 Tareas para inicio de desarrollo

- Instalación de IDE *Visual Studio*. – culminada.
- Instalación de *Sql Management Studio*. – culminada.
- Configuración de repositorio de código en *Team Foundation Service*. – culminada.
- Creación de tablas en la base de datos. – culminada.
- Configuración de modelo de datos de la aplicación. – culminada.

### 2.1.2.3 Tareas de requerimientos

#### **RF04** Búsqueda de propiedades.

- Implementación de controladora y métodos para obtener datos para realizar la búsqueda. – culminada.
- Ingreso de datos – culminada.
- Implementación método *GetPropertyTypes*. – culminada.
- Implementación método *GetCities*. – culminada.
- Implementación método *GetNeighborhoods*. – culminada.
- Implementación método *GetProfileTypes*. – culminada.

Estado: culminado.

#### **RF04.1** Búsqueda tradicional

- Implementación de controladora de búsqueda y lógica de búsqueda. – culminada.
- Filtros de búsqueda:
  - *FilterOperation*. – culminada.
  - *FilterZone*. – culminada.
  - *FilterDetails*. – culminada.
- *Testing* de métodos implementados. – culminada.

Estado: culminado.

#### **RF04.2** Búsqueda por perfil

- Ingreso de datos. – culminada.
- Implementación método *GetProfileTypes*. – culminada.
- Ingreso de parámetros de servicios para la búsqueda por perfil. –pendiente.
- Desarrollo de lógica de búsqueda por perfil. –pendiente.

Estado: Pendiente.

#### **RF07** Listado de propiedades:

- Implementación de listado y objetos de propiedades. – culminada.

### 2.1.3 Cronograma de Trabajo con horas reales

	Tarea	Horas estimadas (horas)	Horas reales	Estado
Nº	Desarrollo	<u>140</u>	<u>118</u>	
	<b>Sprint 3</b>	<b>140</b>	<b>118</b>	
1	<b>Estudio de herramientas y tecnologías</b>	<b>10</b>	<b>15</b>	Culminada
	<b>Configuración de ambiente de desarrollo</b>	<b>9</b>	<b>12</b>	Culminada
2	Configuración servidor de base de datos	4	9	Culminada
3	Configuración repositorio de código	3	3	Culminada
4	Configuración de servidor de la aplicación	2	2	Culminada
5	<b>Primera versión diagrama de casos de uso</b>	<b>3</b>	<b>8</b>	Culminada
6	<b>Primera versión diagrama de clases</b>	<b>4</b>	<b>10</b>	Culminada
7	<b>Primera versión MER</b>	<b>4</b>	<b>4</b>	Culminada
8	<b>Implementación controladoras y métodos a desarrollar de la API</b>	<b>8</b>	<b>6</b>	Culminada
	<b>RF04: Búsqueda de propiedades</b>	<b>54</b>	<b>35</b>	Culminada
9	Base de datos	10	6	Culminada
10	Creación de tablas	6	4	Culminada
11	Ingreso de datos	4	3	Culminada
12	Desarrollo de lógica de búsqueda	26	29	Culminada
13	Desarrollo de lógica de búsqueda tradicional	10	12	Culminada
14	Desarrollo de lógica de búsqueda por perfil	16	8	Pendiente
15	<i>Web service</i>	10	5	Pendiente
16	<i>Testing</i>	8	4	Pendiente
	<b>RF07: Listado de propiedades</b>	<b>30</b>	<b>8</b>	Culminada
17	Desarrollo de lógica de Listado de propiedades	12	4	Culminada
18	<i>Web service</i>	10	2	Culminada
19	<i>Testing</i>	8	2	Culminada
20	<b>Reuniones grupo</b>	<b>6</b>	<b>6</b>	Culminada
21	<b>Reuniones con tutor</b>	<b>2</b>	<b>2</b>	Culminada
22	<b>Documentación Sprint</b>	<b>10</b>	<b>12</b>	Culminada

Tabla 3 Cronograma de trabajo con horas reales *Sprint 3*.

## 2.1.4 Gráficas

### 2.1.4.1 Horas estimadas y horas reales

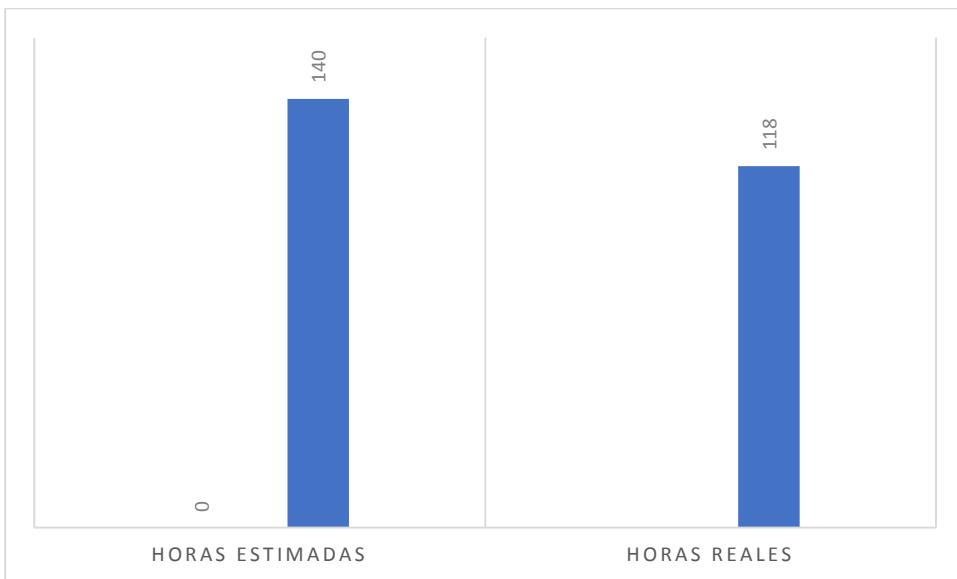


Ilustración 3 Horas estimadas vs horas reales.

### 2.1.4.2 Tareas planificadas, realizadas y pendientes

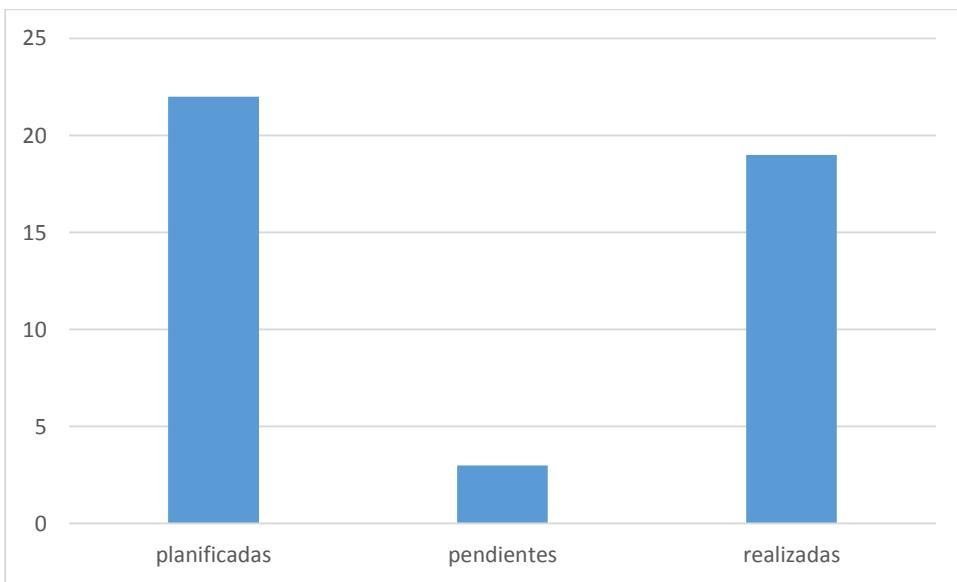


Ilustración 4 tareas planificadas, realizadas y pendientes.

### 2.1.5 Seguimiento de riesgos

Riesgo	Tipo de Riesgo	Probabilidad	Impacto	Plan de contingencia
Rotación del personal	Proyecto	Baja	No se presenta	Se evaluará si se puede recortar el alcance para que alguno de los integrantes pueda continuar
Cambio de Requerimientos	Proyecto y producto	Alta	No se presenta	Prever los cambios de requerimientos en una etapa temprana del proyecto. En caso de surgir cambios durante el mismo se renegociará con el cliente.
Cambio de tecnología	Negocio	Baja	No se presenta	Renegociar el contrato
Reducción del presupuesto	Proyecto	Baja	No se presenta	Que el alcance cuente con lo mínimo indispensable
Problemas legales	Producto	Baja	No se presenta	Cambiar la funcionalidad tratando de evitar el problema y a su vez el que cumpla su propósito, en la medida de lo posible
Nuevas tecnologías	Proyecto y producto	Media	No se presenta	Evaluar el proyecto junto con el cliente para saber si esas nuevas tecnologías mejorarían

				sustancialmente el mismo.
Incumplimiento de plazos	Proyecto	Medio	Se presenta para este <i>sprint</i> . Para el desarrollo de la búsqueda por perfil. Se asignan estas tareas para el siguiente <i>sprint</i>	Renegociar calendario con el cliente
Incumplimiento de integración	Proyecto	Medio	No se presenta	Proporcionar las firmas de métodos y la documentación técnica lo antes posible. En caso de no cumplirse la integración el equipo tendrá que tener ejemplos de integración para entregar en el proyecto.
Integridad de datos del sistema.	Producto	Medio	No se presenta.	En lo que refiere a datos cuando el sistema se suba a un ambiente de producción, se deberá hacer una política de respaldo.

Integración del servicio con la página web.	Producto	Medio	No se presenta.	El cliente podrá contratar al equipo u otro para que integre su sitio web actual con la Api. Renegociar con el cliente.
---	----------	-------	-----------------	--

Tabla 4 Control de riesgos Sprint 3.

#### 2.1.5.1 Nuevos riesgos

Se detectaron nuevos riesgos durante el desarrollo de este *Sprint*. Los cuales se analizaron y se tomaron medidas para evitarlos a futuro. Estos se agregan al Plan de Riesgos.

Riesgo	Tipo de Riesgo	Probabilidad	Impacto	Plan de contingencia
Horas reales de trabajo menores a las horas estimadas	Proyecto	Alta	Se presenta para este <i>Sprint</i>	Al presentarse este riesgo se debe replantear las horas estimadas para los siguientes <i>sprints</i> . A su vez buscar de re organizar los horarios de trabajo para que no se vuelva a presentar.
Errores en la estimación de tareas	Proyecto	Alta	Se presenta para este <i>Sprint</i> . Muchas tareas llevaron más horas reales de las estimadas.	Se debe replantear las siguientes tareas para poder reducir este riesgo.

Contratiempos con equipos personales	Proyecto	Media	Se presentó en este <i>Sprint</i> . Uno de los equipos personales presentó problemas técnicos lo que llevó a un arreglo del mismo. Esto hizo que por unos días se trabajara sin una de las máquinas personales	Se debe buscar remplazar el equipo lo antes posible, también dar solución al desperfecto. También se cuenta con los equipos de laboratorio de la universidad. En este caso se asistió 2 veces a este recurso.
--------------------------------------	----------	-------	--	---

Tabla 5 Nuevos riesgos *sprint* 3.

## 2.1.6 Re planificación de horas

Producto de las tareas pendientes del *sprint* 3 se tuvo que re planificar el siguiente *sprint*. Debido a la experiencia generada en el *sprint* mencionado sabemos que hay tareas que llevarán menos tiempo del estipulado por lo que se podrá usar ese margen para las tareas pendientes así como para otras tareas de este *sprint* que pueden llevar más tiempo.

1.2	<b>Sprint 4</b>	<b>95</b>
0.1	Re planificación de <i>sprint</i>	<b>2</b>
0.1.1	Desarrollo de lógica de búsqueda por perfil ( <i>sprint</i> 3)	10
0.1.2	<i>Web service</i> ( <i>sprint</i> 3)	7
0.1.3	<i>Testing</i> ( <i>sprint</i> 3)	5
1.2.1	<b>RF05: Detalle de Propiedades</b>	<b>57</b>
1.2.1.1	RF05.01 Listado de fotos	5
1.2.1.1.1	Desarrollo lógica de listado de fotos	2
1.2.1.1.2	<i>Web service</i>	2
1.2.1.1.3	<i>Testing</i>	1
1.2.1.2	RF05.02 Detalle de comodidades	7
1.2.1.2.1	Desarrollo lógica de detalle de comodidades	3
1.2.1.2.2	<i>Web service</i>	2
1.2.1.2.3	<i>Testing</i>	2
1.2.1.3	RF05.03 Información general	5
1.2.1.3.1	Desarrollo lógica de detalle de información general	2
1.2.1.3.2	<i>Web service</i>	2
1.2.1.3.3	<i>Testing</i>	1
1.2.1.4	RF05.04 Mapa con datos geográficos	40
1.2.1.4.1	Desarrollo de integración de los servicios con mapa	22
1.2.1.4.2	Desarrollo de conexión con API de Google Places	8

1.2.1.4.3	<i>Web service</i>	3
1.2.1.4.4	<i>Testing</i>	7
1.2.2	<b>Reuniones grupo</b>	<b>2</b>
1.2.3	<b>Reuniones con tutor</b>	<b>2</b>
1.2.4	<b>Documentación Sprint</b>	<b>10</b>

Tabla 6 Re planificación de horas *sprint* 3.

Para las siguientes iteraciones no fue necesario re planificar las tareas por lo cual no se modificará el cronograma. A continuación se detallan las tareas de las siguientes *sprints*.

1.3	<b>Sprint 5</b>	120
1.3.1	<b>RF01 Registro de usuario</b>	<b>23</b>
1.3.1.1	Creación de tablas	4
1.3.1.2	Ingreso de datos de prueba	2
1.3.1.3	Desarrollo de lógica de registro de usuario	10
1.3.1.4	<i>Web Service</i>	4
1.3.1.5	<i>Testing</i>	3
1.3.2	<b>RF02 Inicio de sesión</b>	<b>15</b>
1.3.2.1	Desarrollo de lógica de inicio de sesión	8
1.3.2.2	<i>Web Service</i>	4
1.3.2.3	<i>Testing</i>	3
1.3.3	<b>RF03 Recuperar contraseña</b>	<b>15</b>
1.3.3.1	Desarrollo de lógica de recuperar contraseña	8
1.3.3.2	<i>Web service</i>	4
1.3.3.3	<i>Testing</i>	3
1.3.4	<b>RF18 Cerrar sesión</b>	<b>15</b>
1.3.4.1	Desarrollo de lógica de cerrar sesión	8
1.3.4.2	<i>Web Service</i>	4
1.3.4.3	<i>Testing</i>	3
1.3.5	<b>RF06 ABM de propiedades</b>	<b>20</b>
1.3.5.1	Desarrollo lógica de ABM propiedades	13
1.3.5.2	<i>Web service</i>	4
1.3.5.3	<i>Testing</i>	3
1.3.6	<b>RF14 ABM comentarios</b>	<b>20</b>
1.3.6.1	Desarrollo lógica de ABM comentarios	13
1.3.6.2	<i>Web service</i>	4
1.3.6.3	<i>Testing</i>	3
1.3.7	<b>Reuniones grupo</b>	<b>1</b>
1.3.8	<b>Reuniones con tutor</b>	<b>2</b>
1.3.9	<b>Documentación Sprint</b>	<b>9</b>
1.4	<b>Sprint 6</b>	120
1.4.1	<b>RF11 Agregar propiedad como favorita y seguimiento de propiedad</b>	<b>24</b>

1.4.1.1	Creación de tablas	5
1.4.1.2	Ingreso de Datos	3
1.4.1.3	Desarrollo lógica de propiedad favorita	9
1.4.1.4	<i>Web service</i>	4
1.4.1.5	<i>Testing</i>	3
<b>1.4.2</b>	<b>RF12 Notificaciones</b>	<b>24</b>
1.4.2.1	Creación de tablas	5
1.4.2.2	Ingreso de datos	3
1.4.2.3	Desarrollo lógica notificaciones	9
1.4.2.4	<i>Web service</i>	4
1.4.2.5	<i>Testing</i>	3
<b>1.4.3</b>	<b>RF13 Listado notificaciones</b>	<b>16</b>
1.4.3.1	Desarrollo lógica listado notificaciones	9
1.4.3.2	<i>Web service</i>	4
1.4.3.3	<i>Testing</i>	3
<b>1.4.4</b>	<b>RF15 Recomendaciones de propiedades</b>	<b>24</b>
1.4.4.1	Desarrollo lógica recomendaciones	14
1.4.4.2	<i>Web service</i>	5
1.4.4.3	<i>Testing</i>	5
<b>1.4.5</b>	<b>RF16 Realizar pedido de propiedad</b>	<b>16</b>
1.4.5.1	Desarrollo lógica pedido de propiedad	9
1.4.5.2	<i>Web service</i>	4
1.4.5.3	<i>Testing</i>	3
<b>1.4.6</b>	<b>Reuniones grupo</b>	<b>4</b>
<b>1.4.7</b>	<b>Reuniones con tutor</b>	<b>2</b>
<b>1.4.8</b>	<b>Documentación Sprint</b>	<b>10</b>
<b>1.5</b>	<b>Sprint 7</b>	<b>160</b>
<b>1.5.1</b>	<b>RF08 Solicitud alta inmueble</b>	<b>15</b>
1.5.1.1	Desarrollo solicitud alta inmueble	8
1.5.1.2	<i>Web service</i>	4
1.5.1.3	<i>Testing</i>	3
<b>1.5.2</b>	<b>RF09 Listado solicitudes</b>	<b>18</b>
1.5.2.1	Desarrollo lógica listado solicitudes	11
1.5.2.2	<i>Web service</i>	4
1.5.2.3	<i>Testing</i>	3
<b>1.5.3</b>	<b>RF10 Aprobar solicitud</b>	<b>12</b>
1.5.3.1	Desarrollo lógica aprobar solicitud	6
1.5.3.2	<i>Web service</i>	4
1.5.3.3	<i>Testing</i>	2
<b>1.5.4</b>	<b>RF17 Agendar visita</b>	<b>18</b>
1.5.4.1	Desarrollo lógica agendar visita	11

1.5.4.2	<i>Web service</i>	4
1.5.4.3	<i>Testing</i>	3
1.5.5	<b>Reuniones grupo</b>	<b>4</b>
1.5.6	<b>Reuniones con tutor</b>	<b>3</b>
1.5.7	<b>Documentación Sprint</b>	<b>10</b>
1.5.8	<b>Faltantes y arreglos de todo el producto</b>	<b>80</b>
2	<b>Sprint 8</b>	<b>140</b>
2.1	<b>Preparación entrega final</b>	<b>40</b>
2.2	<b>Documentación proyecto</b>	<b>60</b>
2.3	<b>Reuniones grupo</b>	<b>10</b>
2.4	<b>Reuniones tutor</b>	<b>3</b>
2.5	<b>Arreglos finales - Entrega</b>	<b>27</b>

Tabla 7 Siguientes iteraciones, sprint 3.

### 2.1.7 Conclusiones del *Sprint*

Terminado el tiempo estipulado para el *sprint 3*, concluimos que debido a la falta de experiencia e imprevistos ya mencionados, no se pudo culminar el mismo. Esto hizo que se debiera hacer una re planificación del *sprint 4*, el cual tendrá tareas del *sprint 3* lo que llevará a aumentar la dedicación de horas. Esto nos sirvió para aprender del error y detectar nuevos riesgos para poder evitarlos en el transcurso del proyecto.

Por otro lado, por más que algunas tareas nos llevaron más tiempo del estimado en un principio, como por ejemplo el estudio de herramientas y tecnologías, esto ayudó a definir mejor el enfoque del trabajo y el producto. Fueron horas que si bien no teníamos previstas a la larga facilitaron en otras tareas del desarrollo, así como conocimientos nuevos para los integrantes del equipo.

Se realizaron reuniones con el tutor todas las semanas desde el comienzo del *sprint*, así como comunicación vía *email*. Con el cliente, tuvimos reuniones más esporádicas pero siempre ha estado a nuestra disposición por cualquier duda que ha surgido.

## 2.2 *Sprint 4*

### 2.2.1 Resumen

Este *sprint* fue dedicado a la re planificación e implementación de las tareas pendientes del *sprint 3*.

A su vez, se comenzó con lo planificado en el *sprint 4* con unos días de retraso producto lo mencionado anteriormente. Este *sprint* se basó en el desarrollo del requerimiento “RF05 Detalle de Propiedad”. El mismo se dividió en cuatro partes: listado de fotos, detalle de comodidades, información general y mapa con datos geográficos.

Durante el desarrollo, la mayoría de las tareas se realizaron sin mayores inconvenientes. Donde sí hubo que aumentar la investigación y replantear el desarrollo fue en la conexión con la *API* de *Google places* para poder visualizar correctamente lo que se desea del mapa

con los datos geográficos. Esto se debió, a que necesitábamos un mayor conocimiento del que teníamos respecto a este tema.

Una vez culminadas las nuevas investigaciones se procedió a terminar con las tareas planificadas y así finalizar con el *sprint 4*.

## 2.2.2 Tareas

A continuación se muestra el detalle de las tareas planteadas para este *sprint* y el estado de cada una.

### 2.2.2.1 Tareas generales

- Estudio de herramientas: *Google places API*. – culminada.
- Publicación del sistema al ambiente *Azure*. – culminada.

### 2.2.2.2 Tareas para el desarrollo

- Creación de nuevas tablas en la base de datos. – culminada.
- Actualización de modelo de datos de la aplicación. – culminada.
- Ingreso de nuevos datos de prueba. – culminada.

### 2.2.2.3 Tareas de requerimientos

#### **RF04.2** Búsqueda por perfil

Terminación de requerimiento correspondiente al *sprint 3*.

- Ingreso de parámetros de servicios para la búsqueda por perfil. – culminada.
- Desarrollo de lógica de búsqueda por perfil. – culminada.
- *Testing*. – culminada.

Estado: Terminado.

#### **RF05.1** Listado de fotos

- Implementación de controladora y métodos para obtener los datos de las fotos de la propiedad. – culminada.
- Implementación método *MapPropertyPhotos*. – culminada.

Estado: Terminado.

#### **RF05.2** Detalle de comodidades

- Implementación de controladora y métodos para obtener las comodidades de la propiedad. – culminada.
- Ingreso de datos – culminada.
- Implementación método *GetProperty*. – culminada.
- Implementación método *MapDetail*. – culminada.
- *Testing*. – culminada.

Estado: Terminado.

#### RF05.3 Información General

- Implementación de controladora y método para obtener la información de la propiedad. – culminada.
- Ingreso de datos – culminada.
- Implementación método *MapPropertyInformation*. – culminada.
- *Testing*. – culminada.

Estado: Terminado.

#### RF05.4 Mapa con datos geográficos:

- Implementación de controladora y métodos para obtener el mapa con datos geográficos de la propiedad. – culminada.
- Implementación método *GetPlacesNearby*. – culminada.
- Implementación método *GetPlacesRadarSearch*. – culminada.
- Integración con Java Script. – culminada.
- *Testing*. – culminada.

### 2.2.3 Cronograma de Trabajo con horas reales

Nº	Tarea	Horas estimadas	Horas reales	Estado
		(horas)		
		<u>95</u>	<u>88</u>	-
1	Re planificación de <i>sprint</i>	2	2	culminada
	Desarrollo de lógica de búsqueda por perfil ( <i>sprint 3</i> )	10	12	culminada
2	Método en <i>Web service</i> y creación de objetos ( <i>sprint 3</i> )	7	5	culminada
3	<i>Testing</i> ( <i>sprint 3</i> )	5	3	culminada
	<b>RF05: Detalle de Propiedades</b>	<b>57</b>	<b>54</b>	culminada
	RF05.01 Listado de fotos	5	4	culminada
4	Desarrollo lógica de listado de fotos	2	2	culminada
5	Método en <i>Web service</i> y creación de objetos	2	1	culminada
6	<i>Testing</i>	1	1	culminada
	RF05.02 Detalle de comodidades	7	5	culminada
7	Desarrollo lógica de detalle de comodidades	3	2	culminada
8	Método en <i>Web service</i> y creación de objetos	2	1	culminada
9	<i>Testing</i>	2	2	culminada
	RF05.03 Información general	5	6	culminada

10	Desarrollo lógica de detalle de información general	2	2	culminada
11	Método en <i>Web service</i> y creación de objetos	2	2	culminada
12	<i>Testing</i>	1	2	culminada
	<b>RF05.04 Mapa con datos geográficos</b>	40	39	culminada
13	Desarrollo de integración de los servicios con mapa	22	20	culminada
14	Desarrollo de conexión con API de Google Places	8	10	culminada
15	Método en <i>Web service</i> y creación de objetos	3	4	culminada
16	<i>Testing</i>	7	5	culminada
17	<b>Reuniones grupo</b>	<b>2</b>	2	culminada
18	<b>Reuniones con tutor</b>	<b>2</b>	2	culminada
19	<b>Documentación Sprint</b>	<b>10</b>	8	culminada

Tabla 8 Cronograma de trabajo con horas reales *sprint 4*.

## 2.2.4 Gráficas

### 2.2.4.1 Horas estimadas y horas reales

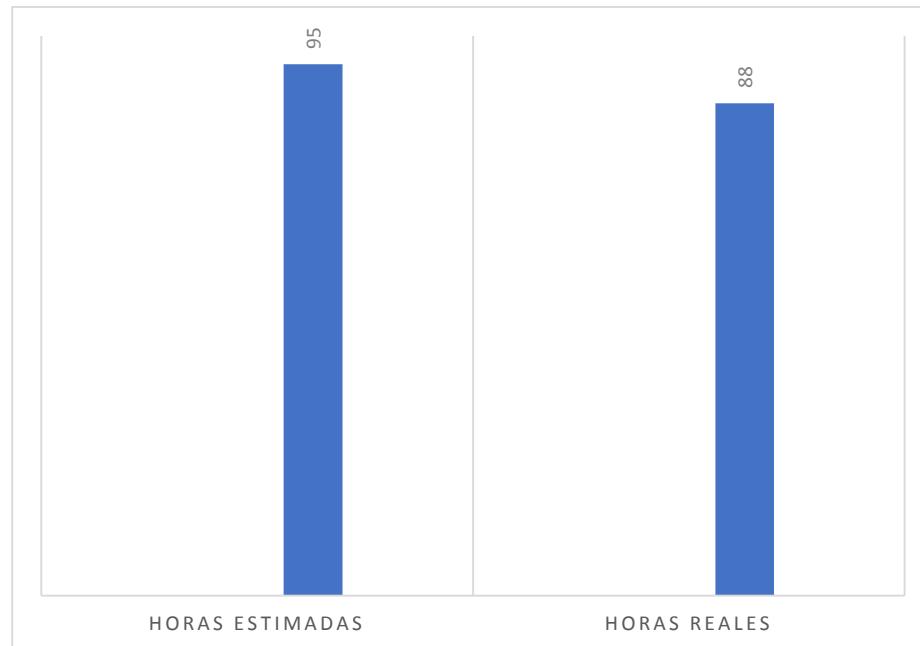


Ilustración 5 Horas estimadas vs horas reales.

#### 2.2.4.2 Tareas planificadas, realizadas y pendientes

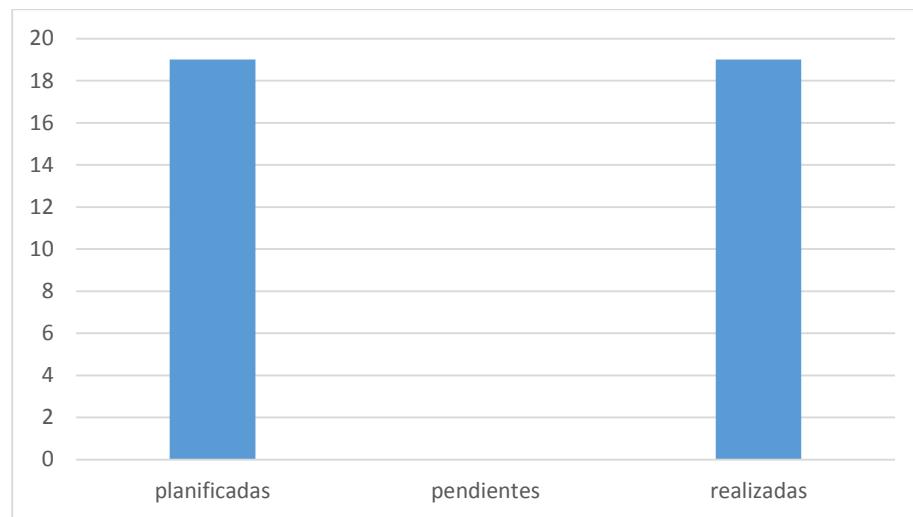


Ilustración 6 tareas planificadas, realizadas y pendientes *sprint 4*.

Para las siguientes iteraciones no fue necesario re planificar las tareas por lo cual no se modificará el cronograma. A continuación se detallan las tareas de los siguientes *sprints*.

1.3	<b>Sprint 5</b>	120
1.3.1	<b>RF01 Registro de usuario</b>	<b>23</b>
1.3.1.1	Creación de tablas	4
1.3.1.2	Ingreso de datos de prueba	2
1.3.1.3	Desarrollo de lógica de registro de usuario	10
1.3.1.4	<i>Web Service</i>	4
1.3.1.5	<i>Testing</i>	3
1.3.2	<b>RF02 Inicio de sesión</b>	<b>15</b>
1.3.2.1	Desarrollo de lógica de inicio de sesión	8
1.3.2.2	<i>Web Service</i>	4
1.3.2.3	<i>Testing</i>	3
1.3.3	<b>RF03 Recuperar contraseña</b>	<b>15</b>
1.3.3.1	Desarrollo de lógica de recuperar contraseña	8
1.3.3.2	Método en <i>Web service</i> y creación de objetos	4
1.3.3.3	<i>Testing</i>	3
1.3.4	<b>RF18 Cerrar sesión</b>	<b>15</b>
1.3.4.1	Desarrollo de lógica de cerrar sesión	8
1.3.4.2	Método en <i>Web service</i> y creación de objetos	4
1.3.4.3	<i>Testing</i>	3
1.3.5	<b>RF06 ABM de propiedades</b>	<b>20</b>
1.3.5.1	Desarrollo lógica de ABM propiedades	13
1.3.5.2	Método en <i>Web service</i> y creación de objetos	4
1.3.5.3	<i>Testing</i>	3
1.3.6	<b>RF14 ABM comentarios</b>	<b>20</b>

1.3.6.1	Desarrollo lógica de ABM comentarios	13
1.3.6.2	Método en <i>Web service</i> y creación de objetos	4
1.3.6.3	<i>Testing</i>	3
1.3.7	<b>Reuniones grupo</b>	1
1.3.8	<b>Reuniones con tutor</b>	2
1.3.9	<b>Documentación Sprint</b>	9
1.4	<b>Sprint 6</b>	120
1.4.1	<b>RF11 Agregar propiedad como favorita y seguimiento de propiedad</b>	24
1.4.1.1	Creación de tablas	5
1.4.1.2	Ingreso de Datos	3
1.4.1.3	Desarrollo lógica de propiedad favorita	9
1.4.1.4	Método en <i>Web service</i> y creación de objetos	4
1.4.1.5	<i>Testing</i>	3
1.4.2	<b>RF12 Notificaciones</b>	24
1.4.2.1	Creación de tablas	5
1.4.2.2	Ingreso de datos	3
1.4.2.3	Desarrollo lógica notificaciones	9
1.4.2.4	Método en <i>Web service</i> y creación de objetos	4
1.4.2.5	<i>Testing</i>	3
1.4.3	<b>RF13 Listado notificaciones</b>	16
1.4.3.1	Desarrollo lógica listado notificaciones	9
1.4.3.2	Método en <i>Web service</i> y creación de objetos	4
1.4.3.3	<i>Testing</i>	3
1.4.4	<b>RF15 Recomendaciones de propiedades</b>	24
1.4.4.1	Desarrollo lógica recomendaciones	14
1.4.4.2	Método en <i>Web service</i> y creación de objetos	5
1.4.4.3	<i>Testing</i>	5
1.4.5	<b>RF16 Realizar pedido de propiedad</b>	16
1.4.5.1	Desarrollo lógica pedido de propiedad	9
1.4.5.2	Método en <i>Web service</i> y creación de objetos	4
1.4.5.3	<i>Testing</i>	3
1.4.6	<b>Reuniones grupo</b>	4
1.4.7	<b>Reuniones con tutor</b>	2
1.4.8	<b>Documentación Sprint</b>	10
1.5	<b>Sprint 7</b>	160
1.5.1	<b>RF08 Solicitud alta inmueble</b>	15
1.5.1.1	Desarrollo solicitud alta inmueble	8
1.5.1.2	Método en <i>Web service</i> y creación de objetos	4
1.5.1.3	<i>Testing</i>	3
1.5.2	<b>RF09 Listado solicitudes</b>	18

1.5.2.1	Desarrollo lógica listado solicitudes	11
1.5.2.2	Método en <i>Web service</i> y creación de objetos	4
1.5.2.3	<i>Testing</i>	3
1.5.3	<b>RF10 Aprobar solicitud</b>	<b>12</b>
1.5.3.1	Desarrollo lógica aprobar solicitud	6
1.5.3.2	Método en <i>Web service</i> y creación de objetos	4
1.5.3.3	<i>Testing</i>	2
1.5.4	<b>RF17 Agendar visita</b>	<b>18</b>
1.5.4.1	Desarrollo lógica agendar visita	11
1.5.4.2	Método en <i>Web service</i> y creación de objetos	4
1.5.4.3	<i>Testing</i>	3
1.5.5	<b>Reuniones grupo</b>	<b>4</b>
1.5.6	<b>Reuniones con tutor</b>	<b>3</b>
1.5.7	<b>Documentación Sprint</b>	<b>10</b>
1.5.8	<b>Faltantes y arreglos de todo el producto</b>	<b>80</b>
2	<b>Sprint 8</b>	<b>140</b>
2.1	<b>Preparación entrega final</b>	<b>40</b>
2.2	<b>Documentación proyecto</b>	<b>60</b>
2.3	<b>Reuniones grupo</b>	<b>10</b>
2.4	<b>Reuniones tutor</b>	<b>3</b>
2.5	<b>Arreglos finales – Entrega</b>	<b>27</b>

Tabla 9 siguientes iteraciones sprint 4.

## 2.2.5 Seguimiento de riesgos

Riesgo	Tipo de Riesgo	Probabilidad	Impacto	Plan de contingencia
Rotación del personal	Proyecto	Baja	No se presenta.	Se evaluará si se puede recortar el alcance para que alguno de los integrantes pueda continuar.
Cambio de Requerimiento	Proyecto y producto	Alta	No se presenta.	Prever los cambios de requerimientos en una etapa temprana del proyecto. En caso de surgir cambios durante el mismo se renegociará con el cliente.
Cambio de tecnología	Negocio	Baja	No se presenta.	Renegociar el contrato.
Reducción del presupuesto	Proyecto	Baja	No se presenta.	Que el alcance cuente con lo mínimo indispensable.

Problemas legales	Producto	Baja	No se presenta.	Cambiar la funcionalidad tratando de evitar el problema y a su vez el que cumpla su propósito, en la medida de lo posible.
Nuevas tecnologías	Proyecto y producto	Media	No se presenta.	Evaluar el proyecto junto con el cliente para saber si esas nuevas tecnologías mejorarían sustancialmente el mismo.
Incumplimiento de plazos	Proyecto	Medio	No se presenta.	Renegociar calendario con el cliente.
Incumplimiento de integración	Proyecto	Medio	No se presenta.	Proporcionar las firmas de métodos y la documentación técnica lo antes posible. En caso de no cumplirse la integración el equipo tendrá que tener ejemplos de integración para entregar en el proyecto.
Horas reales de trabajo menores a las horas estimadas	Proyecto	Alta	No se presenta.	Al presentarse este riesgo se debe replantear las horas estimadas para los siguientes <i>sprints</i> . A su vez buscar de reorganizar los horarios de trabajo para que no se vuelva a presentar.
Errores en la estimación de tareas	Proyecto	Alta	Se presenta, se pudo terminar el <i>sprint 4</i> en una menor cantidad de horas de las estimadas.	Se debe replantear las siguientes tareas para poder reducir este riesgo.

Contriatiempos con equipos personales	Proyecto	Media	No se presenta.	Se debe buscar remplazar el equipo lo antes posible, también dar solución al desperfecto. También se cuenta con los equipos de laboratorio de la universidad. En este caso se asistió 2 veces a este recurso.
Integridad de datos del sistema.	Producto	Medio	No se presenta.	En lo que refiere a datos cuando el sistema se suba a un ambiente de producción, se deberá hacer una política de respaldo.
Integración del servicio con la página web.	Producto	Medio	No se presenta.	El cliente podrá contratar al equipo u otro para que integre su sitio web actual con la Api. Renegociar con el cliente.

Tabla 10 Control de riesgos *sprint 4*.

## 2.2.6 Conclusiones del *Sprint*

Terminado el tiempo estipulado para el *sprint 4*, este se pudo culminar en tiempo y forma. Luego de la experiencia obtenida en el *sprint* anterior, donde no se pudo culminar con la totalidad de las tareas, hubo una mayor dedicación y aprovechamiento del tiempo lo que resultó clave para poder finalizar las tareas planteadas, así como lo pendiente del *sprint 3*.

Al realizar el requerimiento que abarca el mapa con los datos geográficos hubo que realizar una investigación más exhaustiva respecto a la API de *Google places* ya que posee muchas variantes las cuales no conocíamos. Esto hizo que se tuviera que modificar el enfoque de desarrollo de este requerimiento. En un principio se optó por realizarlo desde el lado del servidor y luego por un tema de performance y tiempo, se tomó la decisión de realizarlo desde el lado del cliente.

Se realizaron reuniones con el tutor todas las semanas desde el comienzo del *sprint*, así como comunicación vía *email*. Con el cliente, tuvimos reuniones más esporádicas pero siempre ha estado a nuestra disposición por cualquier duda que ha surgido.

## 2.3 Sprint 5

### 2.3.1 Resumen

Este *sprint* fue dedicado al alta, baja y modificación de usuarios, propiedades y comentarios. También se desarrollaron las funcionalidades para que el usuario sea autenticado o administrador, así como también agregar los roles a los métodos que requieren autenticación ya hechos en los anteriores *sprints*.

En el comienzo del *sprint* hubo que realizar un estudio de las tecnologías ya brindadas por el entorno de desarrollo, respecto a la administración de usuarios, para poder integrarlas con el sistema. Para esto se decidió utilizar *Identity* e integrarlo a la base de usuarios del sistema con la creación de nuevas tablas.

Luego de realizar estos estudios se prosiguió al desarrollo de las tareas planificadas sin mayores contratiempos.

### 2.3.2 Tareas

A continuación se muestra el detalle de las tareas planteadas para este *sprint* y el estado de cada una.

#### 2.3.2.1 Tareas generales

- Estudio de herramientas para el manejo de usuario brindadas por el *framework* de desarrollo.
- Pruebas de *Identity* para el manejo de usuarios.

#### 2.3.2.2 Tareas para el desarrollo

- Creación de nuevas tablas en la base de datos. – culminada.
- Actualización de modelo de datos de la aplicación. – culminada.
- Ingreso de nuevos datos de prueba. – culminada.
- Configuración de *Identity* y acceso a datos. – culminada.

#### 2.3.2.3 Tareas de requerimientos

##### RF01 Registro de Usuario

- Método *RegisterUser* para el registro de usuario. – culminada.
- Creación de modelos. – culminada.
- *Testing*. – culminada.

Estado: Terminado.

##### RF02 Inicio de Sesión

- Integración de *Identity* con nuestro sistema. – culminada.
- *Testing*. – culminada.

Estado: Terminado.

#### **RF03** Recuperar contraseña

- Creación de método *ForgotPassword*. – culminada.
- *Testing*. – culminada.

Estado: Terminado.

#### **RF18** Cerrar Sesión

- Integración de *Identity* con nuestro sistema. – culminada.
- *Testing*. – culminada.

Estado: Terminado.

#### **RF06** ABM de propiedades:

- Método *AddModifyProperty* – culminado.
- Validaciones. – culminada.
- Método *DeleteProperty*. – culminada.
- *Testing*. – culminada.

Estado: culminado.

#### **RF14** ABM de comentarios:

- Método *AddModifyComment*. –culminada.
- Validaciones. – culminada.
- Método *DeleteProperty*. – culminada.
- *Testing*. – culminada.

Estado: Terminado.

#### 2.3.2.4 Tareas de autenticación

- Guardar búsqueda de un usuario. – culminada.
- *Testing* de búsquedas por usuario. – culminada.

### 2.3.3 Cronograma de Trabajo con horas reales

	Tarea	Horas estimadas (horas)	Horas reales	Estado
Nº		120	85	
	Estudio de herramienta <i>Identity</i> - Manejo de Usuarios	0	7	culminada
	Configuración de <i>Identity</i>	0	4	culminada
	Creación de tablas de <i>Identity</i>	0	2	culminada
	Integración de <i>Identity</i> con el sistema	0	1	culminada
	<b>RF01 Registro de usuario</b>	<b>23</b>	<b>15</b>	culminada
1	Creación de tablas	4	2	culminada
2	Ingreso de datos de prueba	2	1	culminada
3	Desarrollo de lógica de registro de usuario	10	8	culminada
4	Método en <i>Web service</i> y creación de objetos	4	3	culminada
5	<i>Testing</i>	3	1	culminada
	<b>RF02 Inicio de sesión</b>	<b>15</b>	<b>4</b>	culminada
6	Desarrollo de lógica de inicio de sesión	8	1	culminada
7	Método en <i>Web service</i> y creación de objetos	4	1	culminada
8	<i>Testing</i>	3	2	culminada
	<b>RF03 Recuperar contraseña</b>	<b>15</b>	<b>8</b>	culminada
9	Desarrollo de lógica de recuperar contraseña	8	6	culminada
10	Método en <i>Web service</i> y creación de objetos	4	1	culminada
11	<i>Testing</i>	3	1	culminada
	<b>RF18 Cerrar sesión</b>	<b>15</b>	<b>2</b>	culminada
12	Desarrollo de lógica de cerrar sesión	8	0	culminada
13	Método en <i>Web service</i> y creación de objetos	4	0	culminada
14	<i>Testing</i>	3	2	culminada
	<b>RF06 ABM de propiedades</b>	<b>20</b>	<b>16</b>	culminada
15	Desarrollo lógica de ABM propiedades	13	10	culminada
16	Método en <i>Web service</i> y creación de objetos	4	2	culminada
17	<i>Testing</i>	3	4	culminada
	<b>RF14 ABM comentarios</b>	<b>20</b>	<b>18</b>	culminada
18	Desarrollo lógica de ABM comentarios	13	12	culminada
19	Método en <i>Web service</i> y creación de objetos	4	3	culminada
20	<i>Testing</i>	3	3	culminada
21	<b>Reuniones grupo</b>	<b>1</b>	<b>1</b>	culminada
22	<b>Reuniones con tutor</b>	<b>2</b>	<b>1</b>	culminada
23	<b>Documentación Sprint</b>	<b>9</b>	<b>8</b>	culminada

Tabla 11 cronograma de trabajo con horas reales *sprint 5*.

## 2.3.4 Gráficas

### 2.3.4.1 Horas estimadas y horas reales

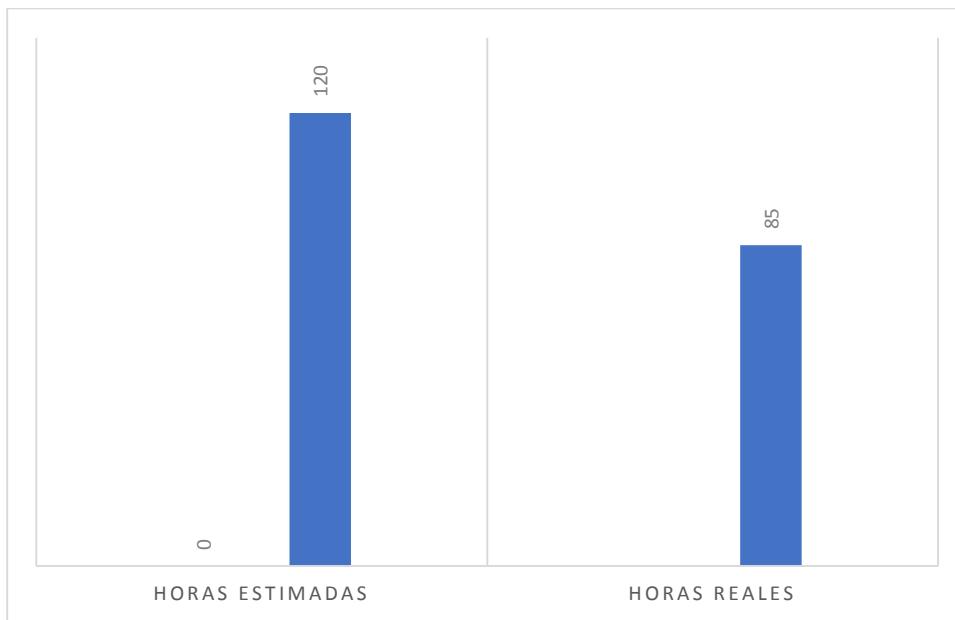


Ilustración 7 horas estimadas vs horas reales *sprint 5*.

### 2.3.4.2 Tareas planificadas, realizadas, pendientes y nuevas

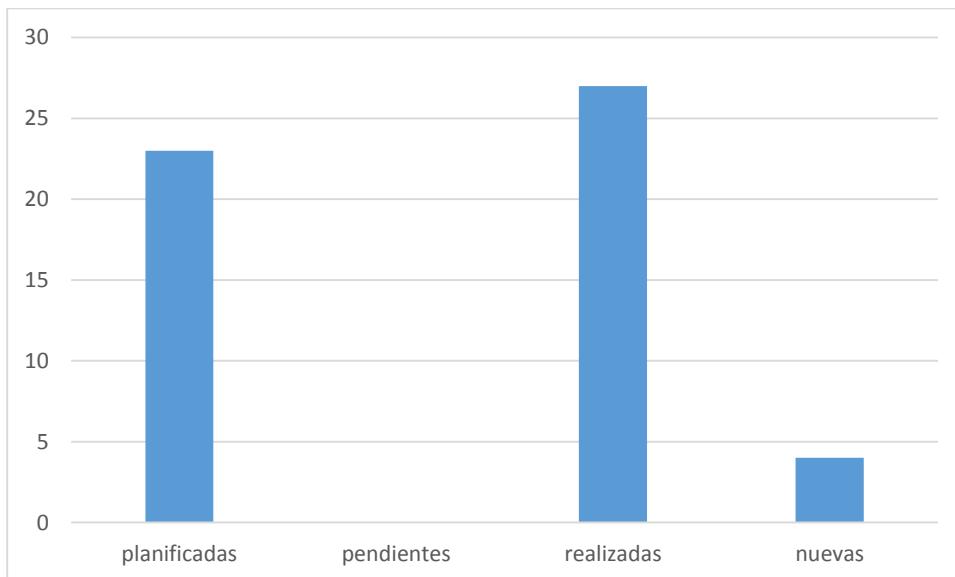


Ilustración 8 tareas planificadas, realizadas, pendientes y nuevas *sprint 5*.

### 2.3.5 Seguimiento de riesgos

Riesgo	Tipo de Riesgo	Probabilidad	Impacto	Plan de contingencia
Rotación del personal	Proyecto	Baja	No se presenta.	Se evaluará si se puede recortar el alcance para que alguno de los integrantes pueda continuar.
Cambio de Requerimiento	Proyecto y producto	Alta	No se presenta.	Prever los cambios de requerimientos en una etapa temprana del proyecto. En caso de surgir cambios durante el mismo se renegociará con el cliente.
Cambio de tecnología	Negocio	Baja	No se presenta.	Renegociar el contrato.
Reducción del presupuesto	Proyecto	Baja	No se presenta.	Que el alcance cuente con lo mínimo indispensable.
Problemas legales	Producto	Baja	No se presenta.	Cambiar la funcionalidad tratando de evitar el problema y a su vez el que cumpla su propósito, en la medida de lo posible.
Nuevas tecnologías	Proyecto y producto	Media	No se presenta.	Evaluar el proyecto junto con el cliente para saber si esas nuevas tecnologías mejorarían sustancialmente el mismo.
Incumplimiento de plazos	Proyecto	Medio	No se presenta.	Renegociar calendario con el cliente.

Incumplimiento de integración	Proyecto	Medio	No se presenta.	Proporcionar las firmas de métodos y la documentación técnica lo antes posible. En caso de no cumplirse la integración el equipo tendrá que tener ejemplos de integración para entregar en el proyecto.
Horas reales de trabajo menores a las horas estimadas	Proyecto	Alta	No se presenta.	Al presentarse este riesgo se debe replantear las horas estimadas para los siguientes <i>sprints</i> . A su vez buscar de re organizar los horarios de trabajo para que no se vuelva a presentar.
Errores en la estimación de tareas	Proyecto	Alta	Se presenta, se puede terminar el <i>sprint 5</i> en una menor cantidad de horas de las estimadas.	Se deben replantear las siguientes tareas para poder reducir este riesgo.
Contriatiempos con equipos personales	Proyecto	Media	No se presenta.	Se debe buscar remplazar el equipo lo antes posible, también dar solución al desperfecto. También se cuenta con los equipos de laboratorio de la universidad. En este caso se asistió 2 veces a este recurso.

Tabla 12 control de riesgos *sprint 5*.

Para las siguientes iteraciones no fue necesario re planificar las tareas por lo cual no se modificará el cronograma. A continuación se detallan las tareas de los siguientes *sprints*.

1.4	<b>Sprint 6</b>	120
1.4.1	<b>RF11 Agregar propiedad como favorita y seguimiento de propiedad</b>	<b>24</b>
1.4.1.1	Creación de tablas	5
1.4.1.2	Ingreso de Datos	3
1.4.1.3	Desarrollo lógica de propiedad favorita	9
1.4.1.4	Método en <i>Web service</i> y creación de objetos	4
1.4.1.5	<i>Testing</i>	3
1.4.2	<b>RF12 Notificaciones</b>	<b>24</b>
1.4.2.1	Creación de tablas	5
1.4.2.2	Ingreso de datos	3
1.4.2.3	Desarrollo lógica notificaciones	9
1.4.2.4	Método en <i>Web service</i> y creación de objetos	4
1.4.2.5	<i>Testing</i>	3
1.4.3	<b>RF13 Listado notificaciones</b>	<b>16</b>
1.4.3.1	Desarrollo lógica listado notificaciones	9
1.4.3.2	Método en <i>Web service</i> y creación de objetos	4
1.4.3.3	<i>Testing</i>	3
1.4.4	<b>RF15 Recomendaciones de propiedades</b>	<b>24</b>
1.4.4.1	Desarrollo lógica recomendaciones	14
1.4.4.2	Método en <i>Web service</i> y creación de objetos	5
1.4.4.3	<i>Testing</i>	5
1.4.5	<b>RF16 Realizar pedido de propiedad</b>	<b>16</b>
1.4.5.1	Desarrollo lógica pedido de propiedad	9
1.4.5.2	Método en <i>Web service</i> y creación de objetos	4
1.4.5.3	<i>Testing</i>	3
1.4.6	<b>Reuniones grupo</b>	<b>4</b>
1.4.7	<b>Reuniones con tutor</b>	<b>2</b>
1.4.8	<b>Documentación Sprint</b>	<b>10</b>
1.5	<b>Sprint 7</b>	160
1.5.1	<b>RF08 Solicitud alta inmueble</b>	<b>15</b>
1.5.1.1	Desarrollo solicitud alta inmueble	8
1.5.1.2	Método en <i>Web service</i> y creación de objetos	4
1.5.1.3	<i>Testing</i>	3
1.5.2	<b>RF09 Listado solicitudes</b>	<b>18</b>
1.5.2.1	Desarrollo lógica listado solicitudes	11
1.5.2.2	Método en <i>Web service</i> y creación de objetos	4
1.5.2.3	<i>Testing</i>	3
1.5.3	<b>RF10 Aprobar solicitud</b>	<b>12</b>

1.5.3.1	Desarrollo lógica aprobar solicitud	6
1.5.3.2	Método en <i>Web service</i> y creación de objetos	4
1.5.3.3	<i>Testing</i>	2
1.5.4	<b>RF17 Agendar visita</b>	<b>18</b>
1.5.4.1	Desarrollo lógica agendar visita	11
1.5.4.2	Método en <i>Web service</i> y creación de objetos	4
1.5.4.3	<i>Testing</i>	3
1.5.5	<b>Reuniones grupo</b>	<b>4</b>
1.5.6	<b>Reuniones con tutor</b>	<b>3</b>
1.5.7	<b>Documentación Sprint</b>	<b>10</b>
1.5.8	<b>Faltantes y arreglos de todo el producto</b>	<b>80</b>
2	<b>Sprint 8</b>	<b>140</b>
2.1	<b>Preparación entrega final</b>	<b>40</b>
2.2	<b>Documentación proyecto</b>	<b>60</b>
2.3	<b>Reuniones grupo</b>	<b>10</b>
2.4	<b>Reuniones tutor</b>	<b>3</b>
2.5	<b>Arreglos finales - Entrega</b>	<b>27</b>

Tabla 13 siguientes iteraciones *sprint* 5.

### 2.3.6 Conclusiones del *Sprint*

Terminado el tiempo estipulado para el *sprint* 5, al igual que en el *sprint* anterior, se pudo culminar en tiempo y forma.

Previo al comienzo del desarrollo de los requerimientos se realizó un estudio para integrar el manejo de usuario que brinda el *Framework* a nuestro sistema. Por más que en un principio no se habían planificado horas para dicho estudio, una vez comprendido, utilizado e integrado al sistema nos facilitó el desarrollo de los requerimientos ya que muchas tareas ya están resueltas con la herramienta.

Se realizaron reuniones con el tutor cada dos semanas desde el comienzo del *sprint*, así como comunicación vía *email*. Con el cliente, tuvimos reuniones más esporádicas pero siempre ha estado a nuestra disposición por cualquier duda que ha surgido.

## 2.4 Sprint 6

### 2.4.1 Resumen

Este *sprint* fue dedicado a la creación y envío de las notificaciones. También se realizaron funcionalidades relacionadas al pedido de una propiedad y el seguimiento de una propiedad como favorita, en las cuales se utiliza el manejo de notificaciones. El envío de notificaciones se realizó vía *email*, y en algunos casos el envío se hará cada determinada cantidad de días.

En el comienzo del *sprint* hubo que realizar un estudio acerca del sistema a utilizar para el envío de *emails*. Finalmente se decidió hacerlo con el servicio que posee *Google*, lo que llevó también a realizar la configuración de SMTP para los envíos. Luego en el caso de una puesta en producción se deberá cambiar el servidor de correo por otro que permita el envío masivo

de correos. A su vez, se investigó acerca de *WebJobs* de *Azure*, ya que en algunos casos las notificaciones (*emails*) se deberán enviar semanalmente.

Luego de realizar estos estudios y configuraciones se prosiguió al desarrollo de las tareas planificadas sin mayores contratiempos. Por último se realizó la creación de una consola para el envío de los *emails*.

## 2.4.2 Tareas

A continuación se muestra el detalle de las tareas planteadas para este *sprint* y el estado de cada una.

### 2.4.2.1 Tareas generales

- Estudio de herramientas para el envío de notificaciones vía *emails*.
- Estudio de *WebJobs* de *Azure*

### 2.4.2.2 Tareas para el desarrollo

- Creación de nuevas tablas en la base de datos. – culminada.
- Actualización de modelo de datos de la aplicación. – culminada.
- Ingreso de nuevos datos de prueba. – culminada.
- Configuración de *SMTP* para el envío de *emails*. – culminada.
- Configuración de *WebJob* para en Servidor *Azure* – culminada.
- Creación de Consola para el envío de *emails* – culminada.

### 2.4.2.3 Tareas de requerimientos

#### RF11 Agregar propiedad como favorita y seguimiento de propiedad

- Método *AddFavoriteProperty* para agregar una propiedad como favorita. – culminada.
- Método *DeleteFavoriteProperty* para eliminar una propiedad de favoritas. – culminada.
- Método *GetFavoriteProperties* para listar las propiedades favoritas de un usuario. – culminada.
- Validaciones. – culminada.
- *Testing*. – culminada.

Estado: Terminado.

#### RF12 Notificaciones

- Método *SendNotification* para enviar una notificación. – culminada.
- Método *SendAllNotification* para enviar todas las notificaciones. – culminada.
- Método *GetFavoriteNotification* para la creación de una notificación de propiedades favoritas. – culminada.
- Método *NotifyPropertyChanges*, notificación de cambios en propiedad favorita. – culminada.

- Método *CreateSuggestionNotification* para la creación de notificaciones de sugerencias de propiedad. – culminada.
- *Testing.* – culminada.

Estado: Terminado.

#### **RF13** Listado de notificaciones

- Método *GetNotificationsUser* para listar las notificaciones de un usuario – culminada.
- *Testing.* – culminada.

Estado: Terminado.

#### **RF15** Recomendaciones de propiedades

- Métodos *CreateSuggestionsForAllUsers* y *ExecuteSimilarSearches* para la búsqueda de propiedades similares y recomendaciones. – culminada.
- *Testing.* – culminada.

Estado: Terminado.

#### **RF16** Realizar pedido de propiedad:

- Método *PropertyRequest* para realizar el pedido de una propiedad – culminado.
- *Testing.* – culminada.

Estado: culminado.

#### 2.4.3 Cronograma de Trabajo con horas reales

	Tarea	Horas estimadas (horas)	Horas reales	Estado
Nº		120	123	culminada
	Investigación de envío de <i>emails</i>	0	4	culminada
	Investigación de <i>Webjobs</i> en <i>Azure</i>	0	5	culminada
	Creación de consola para envío de notificaciones	0	3	culminada
	Configuración de <i>smtp</i> par envío de <i>emails</i>	0	2	culminada
	Configuración de <i>WebJob</i> en Servidor <i>Azure</i>	0	5	culminada
	<b>RF11 Agregar propiedad como favorita y seguimiento de propiedad</b>	<b>24</b>	<b>21</b>	culminada
1	Creación de tablas	5	3	culminada
2	Ingreso de Datos	3	2	culminada
3	Desarrollo lógica de propiedad favorita	9	10	culminada
4	Método en <i>Web service</i> y creación de objetos	4	3	culminada
5	<i>Testing</i>	3	3	culminada

	<b>RF12 Notificaciones</b>	<b>24</b>	<b>20</b>	culminada
6	Creación de tablas	5	4	culminada
7	Ingreso de datos	3	2	culminada
8	Desarrollo lógica notificaciones	9	8	culminada
9	Método en <i>Web service</i> y creación de objetos	4	3	culminada
10	<i>Testing</i>	3	3	culminada
	<b>RF13 Listado notificaciones</b>	<b>16</b>	<b>10</b>	culminada
11	Desarrollo lógica listado notificaciones	9	5	culminada
12	Método en <i>Web service</i> y creación de objetos	4	2	culminada
13	<i>Testing</i>	3	3	culminada
	<b>RF15 Recomendaciones de propiedades</b>	<b>24</b>	<b>22</b>	culminada
14	Desarrollo lógica recomendaciones	14	12	culminada
15	Método en <i>Web service</i> y creación de objetos	5	5	culminada
16	<i>Testing</i>	5	5	culminada
	<b>RF16 Realizar pedido de propiedad</b>	<b>16</b>	<b>15</b>	culminada
17	Desarrollo lógica pedido de propiedad	9	8	culminada
18	Método en <i>Web service</i> y creación de objetos	4	4	culminada
19	<i>Testing</i>	3	3	culminada
20	<b>Reuniones grupo</b>	<b>4</b>	<b>4</b>	culminada
21	<b>Reuniones con tutor</b>	<b>2</b>	<b>2</b>	culminada
22	<b>Documentación Sprint</b>	<b>10</b>	<b>10</b>	culminada

Tabla 14 cronograma con horas reales *sprint* 6.

## 2.4.4 Gráficas

### 2.4.4.1 Horas estimadas y horas reales

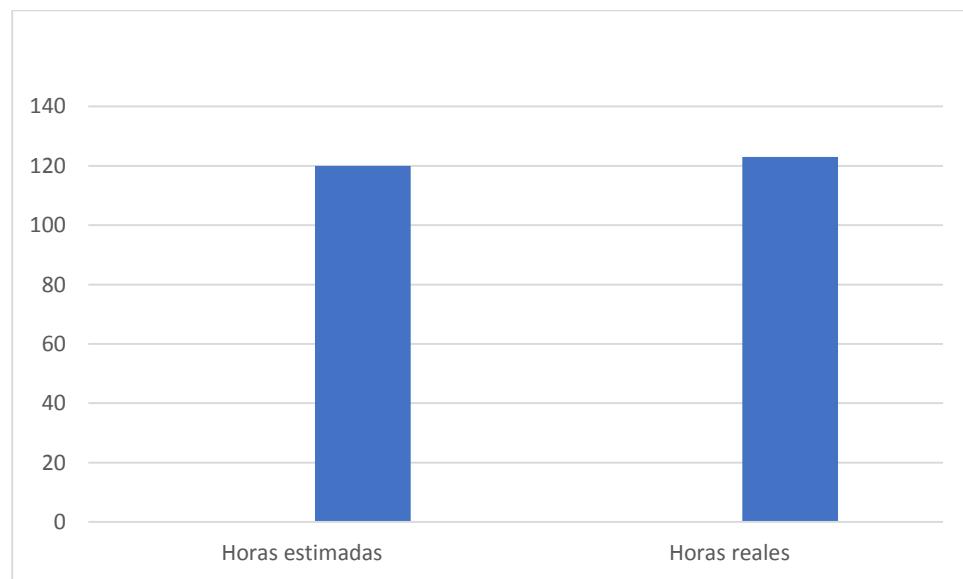


Ilustración 9 Horas estimadas vs horas reales *sprint* 6.

#### 2.4.4.2 Tareas planificadas, pendientes, realizadas y nuevas

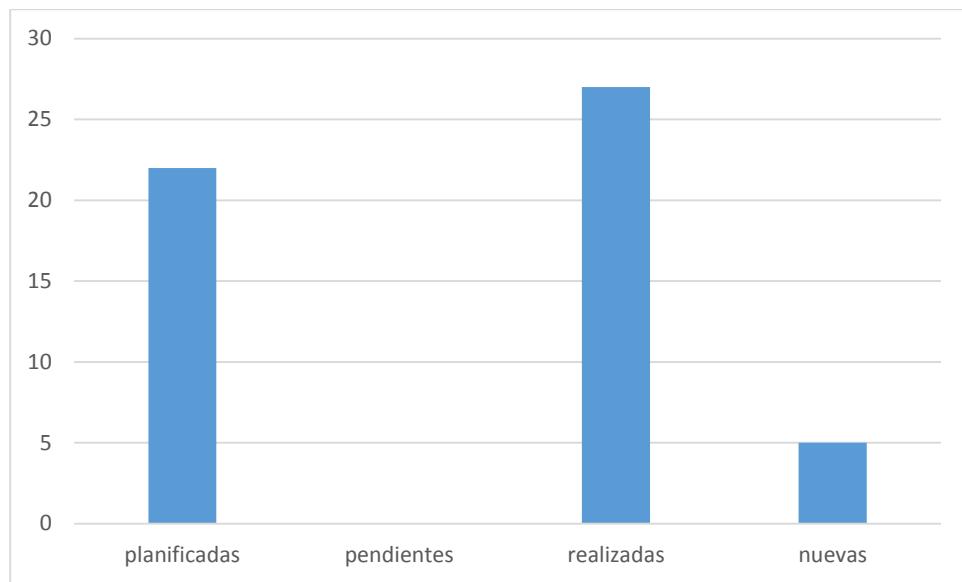


Ilustración 10 Tareas planificadas, pendientes, realizadas y nuevas.

#### 2.4.5 Re planificación de horas

Dado que se presenta el riesgo Incumplimiento de integración el cual se puede observar en el [Seguimiento de riesgos](#), en el próximo *sprint* se llevarán a cabo las tareas de interfaz gráfica e integración con el servicio web. En este *sprint* ya se había planificado un margen de horas para tareas pendientes o arreglos de desarrollo, por lo que se tomarán parte de estas para lo mencionado anteriormente.

1.5	<b>Sprint 7</b>	
1.5.0	<b>Re planificación</b>	<b>45</b>
1.5.0.1	Integración de funcionalidades	25
1.5.0.2	Interfaz gráfica	15
1.5.0.3	<i>Testing</i>	5
1.5.1	<b>RF08 Solicitud alta inmueble</b>	<b>15</b>
1.5.1.1	Desarrollo solicitud alta inmueble	8
1.5.1.2	Método en <i>Web service</i> y creación de objetos	4
1.5.1.3	<i>Testing</i>	3
1.5.2	<b>RF09 Listado solicitudes</b>	<b>18</b>
1.5.2.1	Desarrollo lógica listado solicitudes	11
1.5.2.2	Método en <i>Web service</i> y creación de objetos	4
1.5.2.3	<i>Testing</i>	3
1.5.3	<b>RF10 Aprobar solicitud</b>	<b>12</b>
1.5.3.1	Desarrollo lógica aprobar solicitud	6
1.5.3.2	Método en <i>Web service</i> y creación de objetos	4
1.5.3.3	<i>Testing</i>	2
1.5.4	<b>RF17 Agendar visita</b>	<b>18</b>
1.5.4.1	Desarrollo lógica agendar visita	11

1.5.4.2	Método en <i>Web service</i> y creación de objetos	4
1.5.4.3	<i>Testing</i>	3
1.5.5	<b>Reuniones grupo</b>	<b>4</b>
1.5.6	<b>Reuniones con tutor</b>	<b>3</b>
1.5.7	<b>Documentación Sprint</b>	<b>10</b>
1.5.8	<b>Faltantes y arreglos de todo el producto</b>	<b>35</b>

Tabla 15 Siguientes iteraciones *sprint* 6.

Para el siguiente y último *sprint* no fue necesario re planificar las tareas por lo cual se mantiene el cronograma.

2	<b>Sprint 8</b>	140
2.1	<b>Preparación entrega final</b>	<b>40</b>
2.2	<b>Documentación proyecto</b>	<b>60</b>
2.3	<b>Reuniones grupo</b>	<b>10</b>
2.4	<b>Reuniones tutor</b>	<b>3</b>
2.5	<b>Arreglos finales - Entrega</b>	<b>27</b>

Tabla 16 Cronograma último sprint.

#### 2.4.6 Seguimiento de riesgos

Riesgo	Tipo de Riesgo	Probabilidad	Impacto	Plan de contingencia
Rotación del personal	Proyecto	Baja	No se presenta.	Se evaluará si se puede recortar el alcance para que alguno de los integrantes pueda continuar.
Cambio de Requerimiento	Proyecto y producto	Alta	No se presenta.	Prever los cambios de requerimientos en una etapa temprana del proyecto. En caso de surgir cambios durante el mismo se renegociará con el cliente.
Cambio de tecnología	Negocio	Baja	No se presenta.	Renegociar el contrato.
Reducción del presupuesto	Proyecto	Baja	No se presenta.	Que el alcance cuente con lo mínimo indispensable.

Problemas legales	Producto	Baja	No se presenta.	Cambiar la funcionalidad tratando de evitar el problema y a su vez el que cumpla su propósito, en la medida de lo posible.
Nuevas tecnologías	Proyecto y producto	Media	No se presenta.	Evaluar el proyecto junto con el cliente para saber si esas nuevas tecnologías mejorarían sustancialmente el mismo.
Incumplimiento de plazos	Proyecto	Medio	No se presenta.	Renegociar calendario con el cliente.
Incumplimiento de integración	Proyecto	Medio	Se presenta en este <i>sprint</i> . El cliente informa que no podrá cumplir en tiempo y forma con la integración. El equipo re planifica las horas para proporcionar ejemplos de la integración.	Proporcionar las firmas de métodos y la documentación técnica lo antes posible. En caso de no cumplirse la integración el equipo tendrá que tener ejemplos de integración para entregar en el proyecto.
Horas reales de trabajo menores a las horas estimadas	Proyecto	Alta	No se presenta.	Al presentarse este riesgo se debe replantear las horas estimadas para los siguientes <i>sprints</i> . A su vez buscar de re organizar los horarios de

				trabajo para que no se vuelva a presentar.
Errores en la estimación de tareas	Proyecto	Alta	No se presenta.	Se debe replantear las siguientes tareas para poder reducir este riesgo.
Contratiempos con equipos personales	Proyecto	Media	No se presenta.	Se debe buscar remplazar el equipo lo antes posible, también dar solución al desperfecto. También se cuenta con los equipos de laboratorio de la universidad. En este caso se asistió 2 veces a este recurso.
Integridad de datos del sistema.	Producto	Medio	No se presenta.	En lo que refiere a datos cuando el sistema se suba a un ambiente de producción, se deberá hacer una política de respaldo.

Integración del servicio con la página web.	Producto	Medio	No se presenta.	El cliente podrá contratar al equipo u otro para que integre su sitio web actual con la API. Renegociar con el cliente.
---	----------	-------	-----------------	---

Tabla 17 Control de riesgos *sprint* 6.

#### 2.4.7 Conclusiones del *Sprint*

Terminado el tiempo estipulado para el *sprint* 6, al igual que en el *sprint* anterior, se pudo culminar en tiempo y forma.

Previo al comienzo del desarrollo de los requerimientos se realizó un estudio para el envío de notificaciones vía correo electrónico, así como también la utilización de herramientas automáticas. En un principio estas tareas no estaban previstas en el cronograma de trabajo, pero igualmente se decidió llevarlas a cabo dado a que un mejor conocimiento nos lleva a ahorrar tiempo y realizar las tareas de forma más efectiva.

Finalmente, se presentó el riesgo de Incumplimiento de integración. El mismo consistía en que la inmobiliaria sería la responsable de integrar los servicios desarrollados por el equipo a su aplicación *web*. Esto no se va a cumplir, por lo que el equipo se encargará de integrar las principales funcionalidades del servicio. Debido a esta razón se re planificó el *sprint* 7, el cual llevará como tareas a realizar, la interfaz de usuario y la integración con la *API*.

Se realizaron reuniones con el tutor cada dos semanas desde el comienzo del *sprint*, así como comunicación vía *email*. Con el cliente, tuvimos reuniones más esporádicas pero siempre ha estado a nuestra disposición por cualquier duda que ha surgido.

### 2.5 *Sprint* 7

#### 2.5.1 Resumen

Este *sprint* fue dedicado a la integración de la *API* y al diseño de interfaces, así como también el desarrollo de los requerimientos planteados en este *sprint* relacionados con la solicitud de alta inmueble y la agenda de visitas.

Dado que se presentó el riesgo de la integración de la *API* al sistema ya existente por parte del cliente, en el comienzo del *sprint* hubo que realizar un análisis acerca de qué funcionalidades integraríamos para que estas tuvieran su interfaz a modo de ejemplo. Una vez realizado se decidió integrar las siguientes funcionalidades: búsqueda de propiedades, listado de propiedades, detalle de propiedad, agregar propiedad favorita, comentarios, listado de notificaciones, agendar visita, inicio de sesión y cerrar sesión. Esta decisión se basó en lo que el equipo y el cliente consideraron las más importantes, y que son las principales funcionalidades del sistema.

Luego de realizar la integración mencionada se pasó a desarrollar los requerimientos previamente planificados en el *sprint*.

Por último se ingresaron datos de prueba. Para ello tuvimos que contactarnos con el desarrollador del sistema existente del cliente el cual nos facilitó la importación de todas las propiedades desde *Magento* (plataforma utilizada por el desarrollador) [ver Anexo E: Ingreso de propiedades de prueba.](#)

## 2.5.2 Tareas

A continuación se muestra el detalle de las tareas planteadas para este *sprint* y el estado de cada una.

### 2.5.2.1 Tareas para el desarrollo

- Ingreso de nuevos datos de prueba. – culminada.
- Creación del proyecto *web* en *Visual Studio*. – culminada.
- Creación de servidor de aplicaciones en *Azure*. – culminada.

### 2.5.2.2 Tareas de requerimientos

#### **RF08** Solicitud de alta inmueble

- Método *EnterRequest* para solicitar el ingreso de una propiedad. – culminada.
- *Testing.* – culminada.

Estado: Terminado.

#### **RF09** Listado de solicitudes

- Método *GetRequest* lista todas solicitudes para ingresar un inmueble. – culminada.
- *Testing.* – culminada.

Estado: Terminado.

#### **RF10** Aprobar solicitud

- Método *ApproveRequest* donde el administrador aprueba o no el pedido de ingresar un inmueble. – culminada.
- *Testing* – culminada.

Estado: Terminado.

#### **RF17** Agendar visita

- Método *ScheduleVisit* donde el usuario agenda una visita – culminada.
- *Testing.* – culminada.

Estado: Terminado.

### Tareas de integración y diseño

- Inicio de sesión.
- Cerrar sesión.

- Búsqueda de propiedades.
- Detalle de propiedad.
- Listado de propiedades.
- Agregar propiedad favorita.
- Comentarios.
- Listado de notificaciones.
- Agendar visita.

### 2.5.3 Cronograma de Trabajo con horas reales

	Tarea	Horas estimadas (horas)	Horas reales	Estado
		160	143	culminada
1	Importación de propiedades	0	16	culminada
	<b>Re planificación</b>	<b>45</b>	<b>50</b>	culminada
2	Integración de funcionalidades	25	30	culminada
3	Interfaz gráfica	15	16	culminada
4	<i>Testing</i>	5	4	culminada
	<b>RF08 Solicitud alta inmueble</b>	<b>15</b>	<b>12</b>	culminada
5	Desarrollo solicitud alta inmueble	8	7	culminada
6	Método en <i>Web service</i> y creación de objetos	4	3	culminada
7	<i>Testing</i>	3	2	culminada
	<b>RF09 Listado solicitudes</b>	<b>18</b>	<b>12</b>	culminada
8	Desarrollo lógica listado solicitudes	11	8	culminada
9	Método en <i>Web service</i> y creación de objetos	4	2	culminada
10	<i>Testing</i>	3	2	culminada
	<b>RF10 Aprobar solicitud</b>	<b>12</b>	<b>9</b>	culminada
11	Desarrollo lógica aprobar solicitud	6	5	culminada
12	Método en <i>Web service</i> y creación de objetos	4	2	culminada
13	<i>Testing</i>	2	2	culminada
	<b>RF17 Agendar visita</b>	<b>18</b>	<b>13</b>	culminada
14	Desarrollo lógica agendar visita	11	10	culminada
15	Método en <i>Web service</i> y creación de objetos	4	1	culminada
16	<i>Testing</i>	3	2	culminada
17	<b>Reuniones grupo</b>	<b>4</b>	<b>4</b>	culminada
18	<b>Reuniones con tutor</b>	<b>3</b>	<b>3</b>	culminada
19	<b>Documentación Sprint</b>	<b>10</b>	<b>10</b>	culminada
20	<b>Faltantes y arreglos de todo el producto</b>	<b>35</b>	<b>30</b>	culminada

Tabla 18 cronograma de trabajo con horas reales *sprint 7*.

## 2.5.4 Gráficas

### 2.5.4.1 Horas estimadas y horas reales

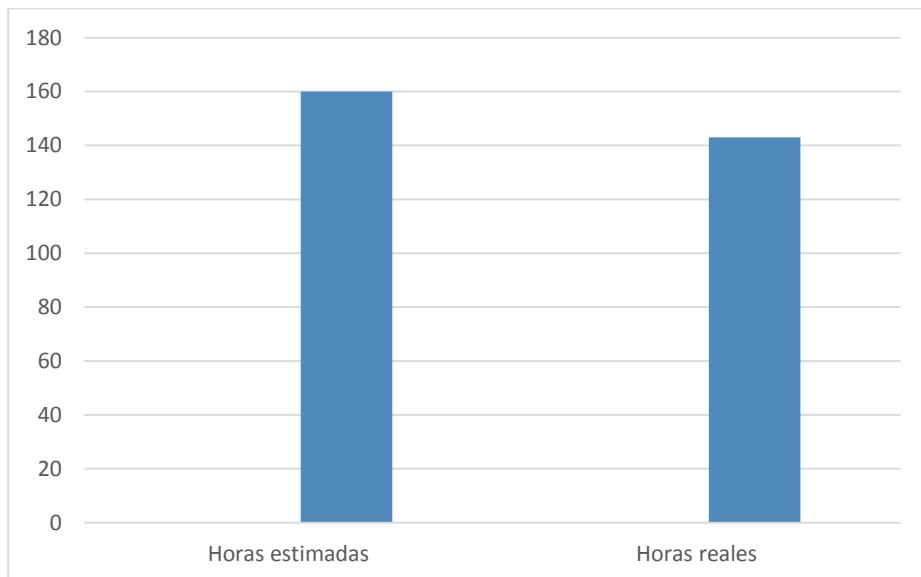


Ilustración 11 Horas estimadas vs horas reales *sprint 7*.

### 2.5.4.2 Tareas planificadas, pendientes, realizadas y nuevas

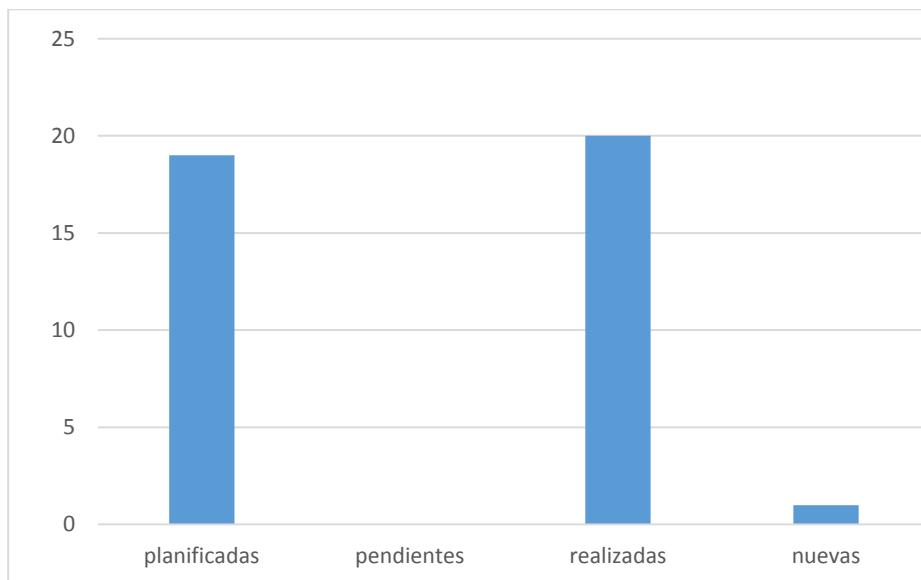


Ilustración 12 Tareas planificadas, pendientes, realizadas y nuevas.

## 2.5.5 Re planificación de horas

Luego de reuniones con el cliente y con el tutor han surgido nuevas ideas las cuales entendemos le dan un valor adicional al producto. Además se intenta realizar un diferencial con sistemas ya existentes en este rubro. Es por esta razón que se decide realizar funcionalidades adicionales y esto hace que se deba realizar una re planificación de horas para llevarlo a cabo.

<b>2</b>	<b>Sprint 8</b>	<b>140</b>
2.0	<b>Re planificación</b>	<b>30</b>
2.0.1	Mapa de propiedades en portada	6
2.0.2	Mapa de propiedades en listado	6
2.0.3	<i>Google Analytics</i>	4
2.0.4	Comparador de propiedades	14
2.1	<b>Preparación entrega final</b>	<b>30</b>
2.2	<b>Documentación proyecto</b>	<b>40</b>
2.3	<b>Reuniones grupo</b>	<b>10</b>
2.4	<b>Reuniones tutor</b>	<b>3</b>
2.5	<b>Arreglos finales - Entrega</b>	<b>27</b>

Tabla 19 Re planificación de horas sprint 7.

## 2.5.6 Seguimiento de riesgos

Riesgo	Tipo de Riesgo	Probabilidad	Impacto	Plan de contingencia
Rotación del personal	Proyecto	Baja	No se presenta.	Se evaluará si se puede recortar el alcance para que alguno de los integrantes pueda continuar.
Cambio de Requerimiento	Proyecto y producto	Alta	No se presenta.	Prever los cambios de requerimientos en una etapa temprana del proyecto. En caso de surgir cambios durante el mismo se renegociará con el cliente.
Cambio de tecnología	Negocio	Baja	No se presenta.	Renegociar el contrato.

Reducción del presupuesto	Proyecto	Baja	No se presenta.	Que el alcance cuente con lo mínimo indispensable.
Problemas legales	Producto	Baja	No se presenta.	Cambiar la funcionalidad tratando de evitar el problema y a su vez el que cumpla su propósito, en la medida de lo posible.
Nuevas tecnologías	Proyecto y producto	Media	No se presenta.	Evaluuar el proyecto junto con el cliente para saber si esas nuevas tecnologías mejorarían sustancialmente el mismo.
Incumplimiento de plazos	Proyecto	Medio	No se presenta.	Renegociar calendario con el cliente.
Horas reales de trabajo menores a las horas estimadas	Proyecto	Alta	No se presenta.	Al presentarse este riesgo se debe replantear las horas estimadas para los siguientes <i>sprints</i> . A su vez buscar de re organizar los horarios de trabajo para que no se vuelva a presentar.
Errores en la estimación de tareas	Proyecto	Alta	No se presenta.	Se debe replantear las siguientes tareas para poder reducir este riesgo.
Contratiempos con equipos personales	Proyecto	Media	No se presenta.	Se debe buscar remplazar el equipo lo antes posible, también dar solución al desperfecto. También se cuenta con los equipos de laboratorio de la universidad. En este caso se asistió 2 veces a este recurso.

Integridad de datos del sistema.	Producto	Medio	No se presenta.	En lo que refiere a datos cuando el sistema se suba a un ambiente de producción, se deberá hacer una política de respaldo.
Integración del servicio con la página web.	Producto	Medio	No se presenta.	El cliente podrá contratar al equipo u otro para que integre su sitio web actual con la Api. Renegociar con el cliente.

Tabla 20 Control de riesgos *sprint* 7.

### 2.5.7 Conclusiones del *Sprint*

Terminado el tiempo estipulado para el *sprint* 7, al igual que en el *sprint* anterior, se pudo culminar en tiempo y forma.

Previo al comienzo del desarrollo de los requerimientos se realizó un análisis para la integración de las principales funcionalidades del servicio *web*, lo cual surge de la presentación del riesgo Incumplimiento de integración en el *sprint* 6. Luego de reuniones con el cliente y tutor se decide crear la interfaz para las principales funcionalidades, las cuales se detallan en las tareas de integración y diseño. Esto hizo que se utilizaran nuevas tecnologías como *HTML* y *JavaScript* que no se habían utilizado hasta el momento en el proyecto. Más allá de que en los riesgos identificamos a uno de ellos como “Nuevas Tecnologías”, este no es el caso ya que son tecnologías conocidas por el equipo y estaba previsto utilizarlas en caso de que tuviéramos que realizar la integración como finalmente sucedió.

Además de los requerimientos planificados para este *sprint*, los cuales se realizaron con éxito, a partir de la necesidad de ingresar datos de prueba en cantidad, se procedió a realizar un método de importación de propiedades para evitar el trabajo manual. Gracias a esto se pudo ingresar más de 50 propiedades reales, las cuales existen en el sistema actual del cliente.

Más allá de las nuevas ideas que surgieron de las reuniones con el cliente y tutor, culminado este *sprint* hemos llegado a obtener un prototipo del sistema culminado, cumpliendo con los requerimientos planificados en un principio. No obstante lo cual, se seguirán dedicando

horas al desarrollo para realizar las ideas mencionadas así como también mejoras del producto.

Se realizaron reuniones con el tutor cada dos semanas desde el comienzo del *sprint*, así como comunicación vía *email*. Con el cliente, tuvimos reuniones más esporádicas pero siempre ha estado a nuestra disposición por cualquier duda que ha surgido.

## 2.6 *Sprint 8*

### 2.6.1 Resumen

Este *sprint* fue dedicado a realizar tareas nuevas las cuales no estaban previstas pero que durante el desarrollo del producto surgieron como nuevas ideas. Se tomó la decisión de realizarlas y así brindarle un valor agregado al producto, ya que por otro lado no afectaba los tiempos que teníamos programados. A su vez, luego de lo ya mencionado se siguió con lo planificado en este *sprint* que era culminar la documentación realizando el cierre del proyecto.

Dentro de las nuevas tareas se ingresó el mapa marcando la ubicación de cada propiedad. Esto se realizó tanto en la portada como en el listado de propiedades. A su vez, se agregó la funcionalidad que permite comparar dos propiedades cualesquiera.

A pedido del cliente, también se agregó la herramienta de *Google Analytics* la cual ayudará al mismo para poder realizar distintas medidas de su sistema *web*.

Por último se siguió con todo lo que refiere a la documentación. Se realizaron correcciones, se agregaron anexos, actas con tutor y cliente, terminando con el cierre del proyecto.

### 2.6.2 Tareas

A continuación se muestra el detalle de las tareas planteadas para este *sprint* y el estado de cada una.

#### 2.6.2.1 Tareas de nuevos requerimientos

##### **RF** Mapa de propiedades en portada

- Métodos *createMap* y *addMarkers* para visualizar el mapa con las propiedades – culminada.
- *Testing.* – culminada.

Estado: Terminado.

##### **RF** Mapa de propiedades en listado de propiedades

- Métodos *createMap* y *addMarkers* para visualizar el mapa con las propiedades- culminada.
- *Testing.* – culminada.

Estado: Terminado.

#### RF Comparador de propiedades

- Método *compare* y *listCompareProperties* para realizar la comparación entre dos propiedades. – culminada.
- *Testing* – culminada.

Estado: Terminado.

#### RF Ingreso de *Google Analytics*

- Ingreso de la herramienta – culminada.
- *Testing*. – culminada.

Estado: Terminado.

#### 2.6.2.2 Tareas de documentación

- Cierre de proyecto.
- Solución implementada.
- Anexos.
- Arreglos y correcciones.

#### 2.6.3 Cronograma de Trabajo con horas reales

	Tarea	Horas estimadas	Horas reales	Estado
	<b>Sprint 8</b>	140	151	
	<b>Re planificación</b>	<b>30</b>	<b>30</b>	culminada
1	Mapa de propiedades en portada	6	10	culminada
2	Mapa de propiedades en listado	6	3	culminada
3	<i>Google Analytics</i>	4	5	culminada
4	Comparador de propiedades	14	12	culminada
5	<b>Preparación entrega final</b>	<b>30</b>	<b>30</b>	culminada
6	<b>Documentación proyecto</b>	<b>40</b>	<b>43</b>	culminada
7	<b>Reuniones grupo</b>	<b>10</b>	<b>10</b>	culminada
8	<b>Reuniones tutor</b>	<b>3</b>	<b>3</b>	culminada
9	<b>Arreglos finales - Entrega</b>	<b>27</b>	<b>35</b>	culminada

Tabla 21 Cronograma de trabajo con horas reales *sprint 8*.

## 2.6.4 Gráficas

### 2.6.4.1 Horas estimadas y horas reales

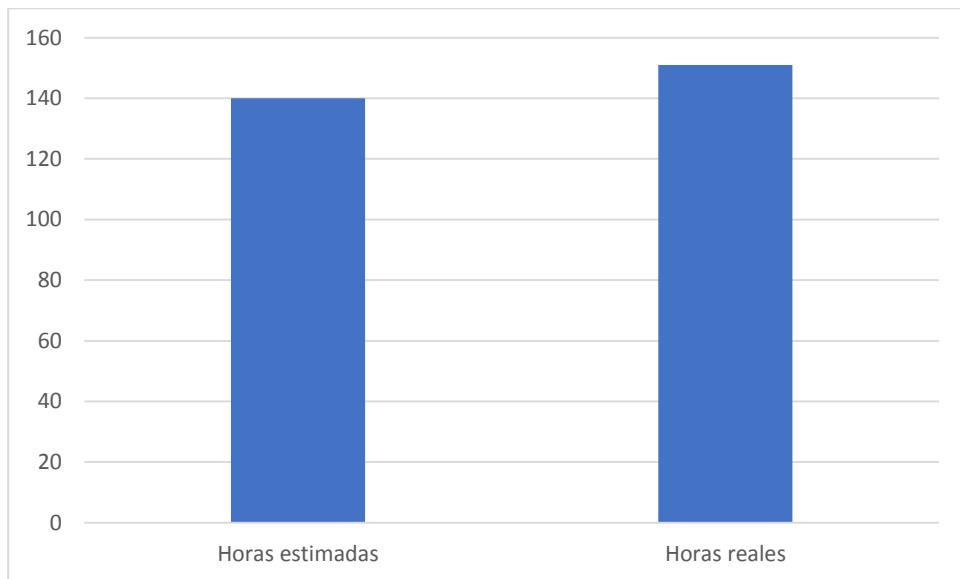


Ilustración 13 Horas estimadas vs horas reales *sprint 8*.

### 2.6.4.2 Tareas planificadas, pendientes, realizadas y nuevas

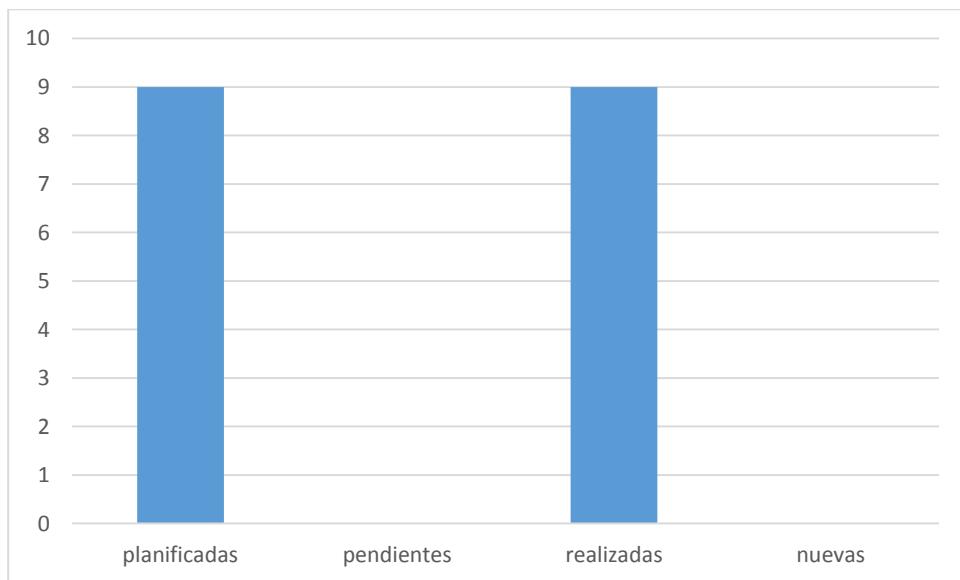


Ilustración 14 Tareas planificadas, pendientes, realizadas y nuevas *sprint 8*.

## 2.6.5 Seguimiento de riesgos

Riesgo	Tipo de Riesgo	Probabilidad	Impacto	Plan de contingencia
Rotación del personal	Proyecto	Baja	No se presenta.	Se evaluará si se puede recortar el alcance para que alguno de los integrantes pueda continuar.
Cambio de Requerimiento	Proyecto y producto	Alta	No se presenta.	Prever los cambios de requerimientos en una etapa temprana del proyecto. En caso de surgir cambios durante el mismo se renegociará con el cliente.
Cambio de tecnología	Negocio	Baja	No se presenta.	Renegociar el contrato.
Reducción del presupuesto	Proyecto	Baja	No se presenta.	Que el alcance cuente con lo mínimo indispensable.
Problemas legales	Producto	Baja	No se presenta.	Cambiar la funcionalidad tratando de evitar el problema y a su vez el que cumpla su propósito, en la medida de lo posible.
Nuevas tecnologías	Proyecto y producto	Media	No se presenta.	Evaluar el proyecto junto con el cliente para saber si esas nuevas tecnologías mejorarían sustancialmente el mismo.
Incumplimiento de plazos	Proyecto	Medio	No se presenta.	Renegociar calendario con el cliente.

Horas reales de trabajo menores a las horas estimadas	Proyecto	Alta	No se presenta.	Al presentarse este riesgo se debe replantear las horas estimadas para los siguientes <i>sprints</i> . A su vez buscar de re organizar los horarios de trabajo para que no se vuelva a presentar.
Errores en la estimación de tareas	Proyecto	Alta	No se presenta.	Se debe replantear las siguientes tareas para poder reducir este riesgo.
Contratiempos con equipos personales	Proyecto	Media	No se presenta.	Se debe buscar remplazar el equipo lo antes posible, también dar solución al desperfecto. También se cuenta con los equipos de laboratorio de la universidad. En este caso se asistió 2 veces a este recurso.
Integridad de datos del sistema.	Producto	Medio	No se presenta.	En lo que refiere a datos cuando el sistema se suba a un ambiente de producción, se deberá hacer una política de respaldo.
Integración del servicio con la página web.	Producto	Medio	No se presenta.	El cliente podrá contratar al equipo u otro para que integre su sitio web actual con la Api. Renegociar con el cliente.

Tabla 22 Control de riesgos *sprint* 8.

## 2.6.6 Conclusiones del *Sprint*

Terminado el tiempo estipulado para el *sprint 8*, al igual que en el *sprint* anterior, se pudo culminar en tiempo y forma, aunque se le dedicaron más horas de lo planificado en un principio.

Al ser el último *sprint* para este proyecto concluimos que se logra obtener el producto deseado por el cliente cumpliendo las necesidades planteadas al inicio del proyecto. Dada la experiencia que obtuvimos con *sprints* anteriores se pudo agregar nuevas tareas para darle valor agregado al proyecto y también cumplir con lo planteado.

Más allá de las tareas realizadas de desarrollo, este *sprint* fue dedicado en su mayoría a la documentación e informes en donde se hizo un balance final de lo que resultó ser el producto pasando por todas las etapas. Desde su comienzo hasta su culminación.

Se realizaron reuniones con el tutor cada dos semanas desde el comienzo del *sprint*, así como comunicación vía mail. Con el cliente, tuvimos reuniones más esporádicas pero siempre ha estado a nuestra disposición por cualquier duda que ha surgido. También se realizó una reunión final con el cliente en la cual se le presentó el prototipo final quedando conforme con el resultado obtenido.

## 3 Solución implementada

### 3.1 API

Se desarrolló un Servicio *Web REST* que cuenta con los métodos para llegar a la solución planteada por el cliente. A continuación detallamos los mismos agrupados por las diferentes controladoras que surgieron de la implementación. Las mismas están detalladas en la documentación de la API la cual se encuentra en la sección de ayuda del sitio *web* de la API, accediendo a cada uno de los links de métodos se podrá ver en detalle los ejemplos de entrada y salida de cada uno, así como también los tipos de datos y los campos obligatorios.

Link de documentación: <http://homesearchapp.azurewebsites.net/Help>

#### Método de notificación:

### Notification

---

#### API

---

[GET api/Notification/Notifications?type={type}](#)

---

Ilustración 15 Método de notificación.

#### Método de pedido de propiedad:

### OrderProperty

---

#### API

---

[POST api/OrderProperty/OrderProperty](#)

---

Ilustración 16 Método de pedido de propiedad.

**Métodos de propiedad:**

## Property

---

API

---

POST api/Property/Search

---

POST api/Property/AddModify

---

POST api/Property/Delete

---

POST api/Property/AddFavoriteProperty?externalId={externalId}

---

GET api/Property/GetFavoriteProperties

---

POST api/Property/DeleteFavoriteProperty?externalId={externalId}

---

GET api/Property/ImportProperties

---

POST api/Property/ScheduleVisit

---

GET api/Property?externalId={externalId}

---

POST api/Property

---

Ilustración 17 Métodos de propiedad.

**Métodos de Usuario:**

## Account

---

### API

---

GET api/Account/UserInfo

---

POST api/Account/Logout

---

GET api/Account/ManageInfo?returnUrl={returnUrl}&generateState={generateState}

---

POST api/Account/ChangePassword

---

POST api/Account/SetPassword

---

POST api/Account/AddExternalLogin

---

POST api/Account/RemoveLogin

---

GET api/Account/ExternalLogin?provider={provider}&error={error}

---

GET api/Account/ExternalLogins?returnUrl={returnUrl}&generateState={generateState}

---

POST api/Account/Register

---

POST api/Account/RegisterExternal

---

POST api/Account/ForgotPassword

---

Ilustración 18 Métodos de usuario.

**Método de ciudad:**

City

---

API

---

[GET api/City/Cities](#)

---

Ilustración 19 Método de ciudad.

**Método de barrios:**

Neighborhood

---

API

---

[GET api/Neighborhood/Neighborhoods?idCity={idCity}](#)

---

Ilustración 20 Método de barrios.

**Método de tipo de perfil:**

ProfileType

---

API

---

[GET api/ProfileType/Types](#)

---

Ilustración 21 Método de tipo de perfil.

**Métodos de ingreso de propiedad:**

## EnterPropertyRequest

---

API

---

POST api/PublishYourProperty/EnterRequest

---

GET api/PublishYourProperty/GetRequests

---

POST api/PublishYourProperty/ApproveRequest?id={id}

---

Ilustración 22 Método de ingreso de propiedad.

**Métodos de comentarios:**

## Comments

---

API

---

POST api/Comment/AddModify

---

POST api/Comment/Delete?id={id}

---

GET api/Comment/GetComments?externalId={externalId}

---

POST api/Comments

---

Ilustración 23 Métodos de comentarios.

**Método de tipo de propiedad:**

## .PropertyType

---

API

---

[GET api/PropertyType/Types](#)

---

Ilustración 24 Método de tipo de propiedad.

**Método de fotos:**

## Photos

---

API

---

[POST api/Photos](#)

---

Ilustración 25 Método de fotos.

### 3.2 Sitio Web

Dado a que no se llegó a realizar la integración por parte del cliente, se implementó una aplicación *web* que consume los principales métodos de la API. Por esta razón en dicha aplicación se podrá visualizar de mejor manera las funcionalidades implementadas en el servicio *web*. A continuación se detallan las distintas secciones que aparecen en dicha aplicación.

Url: <http://homesearchweb.azurewebsites.net>

#### Portada

Cuenta con un carrousel de propiedades en oferta. También se muestra un mapa con las ubicaciones de todas las propiedades del sistema y dos botones para ver propiedades en alquiler o venta.

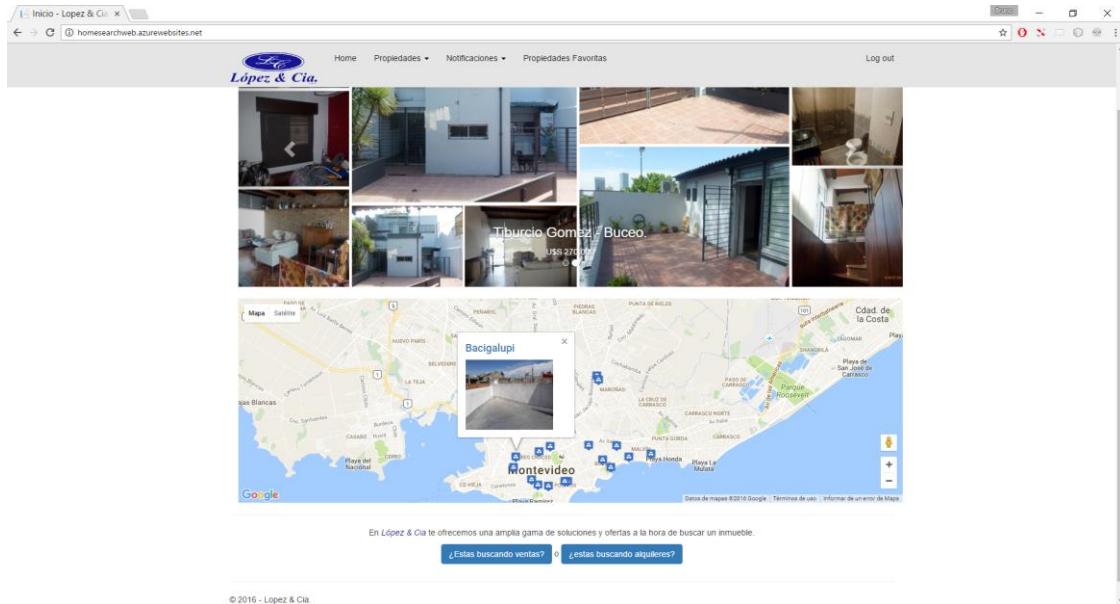


Ilustración 26 Portada.

## Listado propiedades – búsqueda

Se podrá listar propiedades para la venta y/o alquiler, una vez ingresado al listado se podrá visualizar un mapa con la ubicación de las propiedades que resultaron de la búsqueda. También se podrá realizar búsqueda tradicional y por perfil.

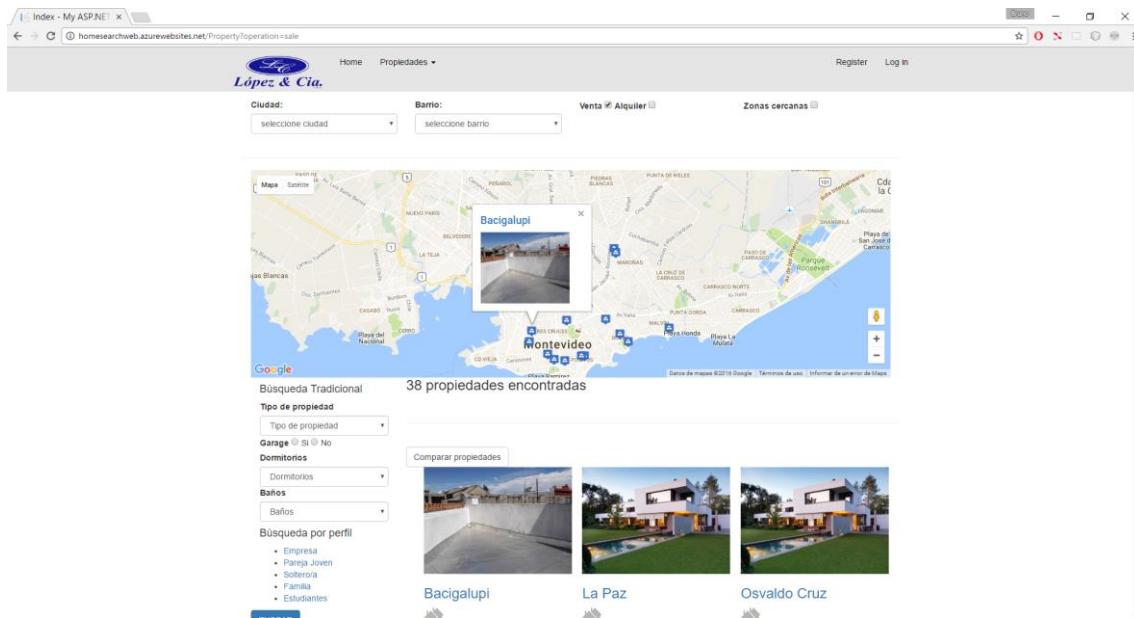


Ilustración 27 Listado de propiedades.

## Comparador de propiedades

En el listado de propiedades, cada una de ellas tendrá un ícono para seleccionar como propiedad a comparar. La comparación deberá ser únicamente entre dos propiedades. Una vez seleccionadas y elegida la opción comparar propiedades, se desplegará un modal con los

datos de ambas propiedades.

Ilustración 28 Comparador de propiedades.

## Notificaciones

Para los usuarios autenticados, se le mostrará la opción para visualizar las distintas notificaciones. Estas podrán ser comentarios, propiedades favoritas, sugerencias y pedidos de propiedad.

Ilustración 29 Notificaciones.

## Listado de propiedades favoritas

Para los usuarios autenticados, se le mostrará la opción para visualizar sus propiedades favoritas.

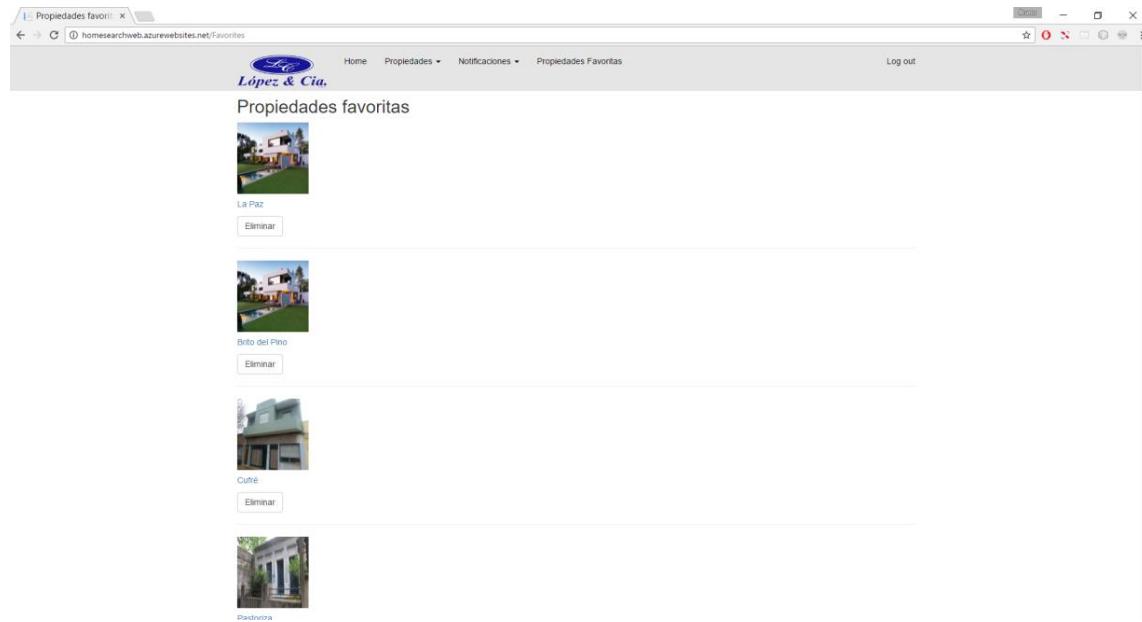


Ilustración 30 Listado de propiedades favoritas.

## Detalle de propiedad

Cada propiedad tendrá su página de detalle, en la cual se mostrarán las fotos, descripción de la misma, precio y una sección donde se podrán observar comodidades, medidas y estado general. A su vez el detalle de propiedad contiene las siguientes secciones:

### Comentarios:

Se mostrarán los comentarios realizados sobre la propiedad y otorga la posibilidad de ingresar nuevos comentarios a los usuarios autenticados.

A screenshot of a comment section. The title 'Comentarios' is at the top. Below it, there are two comments from 'juagon88': one from '17/09/2016' saying 'Muy buena propiedad, muchos servicios cerca.' and another from '17/09/2016' asking 'Acepta banco?'. Each comment has a 'Eliminar' (Delete) button. At the bottom, there is a large text input field with placeholder text 'Escriba su comentario...' and a 'Agregar' (Add) button.

Ilustración 31 Comentarios.

## Agenda de visita:

Los usuarios autenticados tendrán la posibilidad de solicitar una visita a la propiedad indicando fecha y hora de disponibilidad.

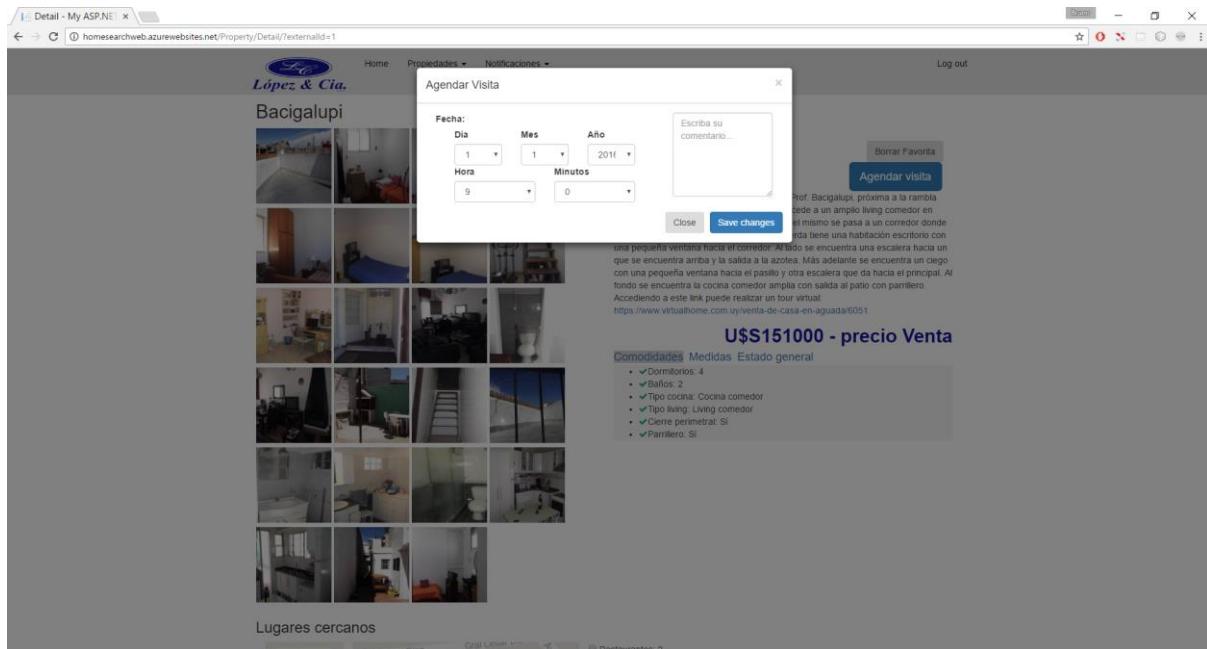


Ilustración 32 Agenda de visita.

## Lugares cercanos:

En esta sección se mostrará la ubicación de la propiedad con los servicios cercanos a la misma. También brinda la posibilidad de filtrar los servicios por tipo (restaurante, supermercado, cajero, farmacia, gimnasio, banco, escuela, universidad, hospital o todos). Se mostrarán solamente las opciones disponibles.

### Lugares cercanos

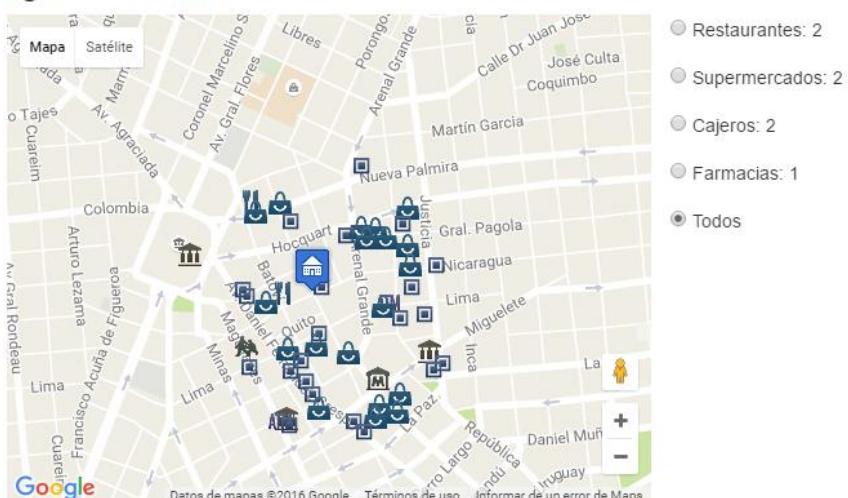


Ilustración 33 Lugares cercanos.

## 4 Cierre Proyecto

### 4.1 Gráficas

#### 4.1.1 Horas reales vs horas planificadas

A continuación se detallan las horas estimadas y las horas dedicadas reales. Durante el desarrollo del proyecto hubo errores en estimaciones ya que no conocíamos la velocidad del equipo, sobre todo al comienzo, luego pudimos ir mejorando a partir de nuevas estimaciones y sobre todo la experiencia que se fue generando. Al comienzo, en el *sprint 3* faltaron horas de trabajo para poder terminar las tareas pero en cambio en otros *sprints* por más que no se dedicaron las horas estimadas se pudo cumplir con la totalidad de lo estipulado.

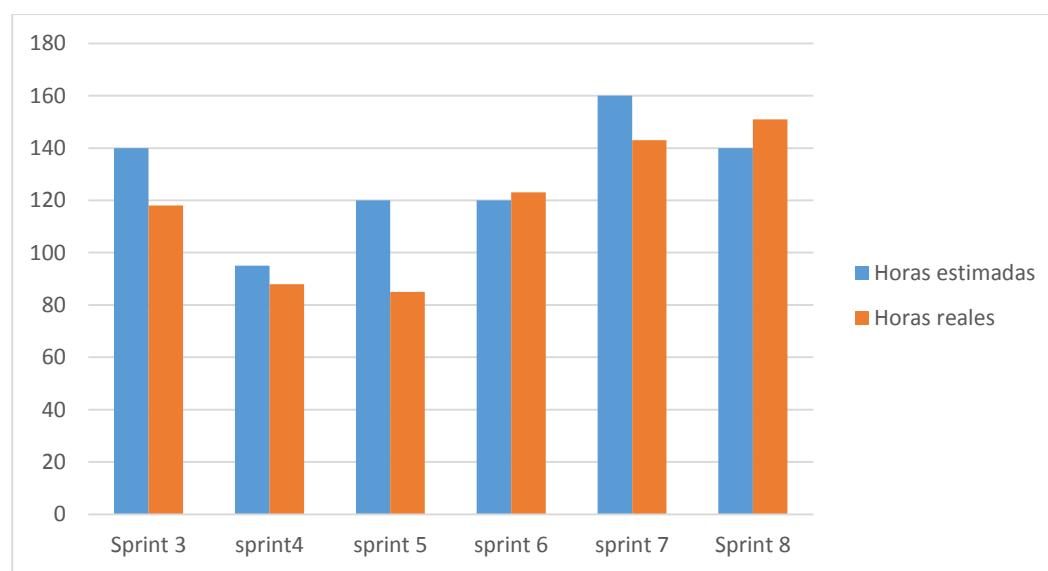


Ilustración 34 Horas reales vs horas planificadas por sprint.

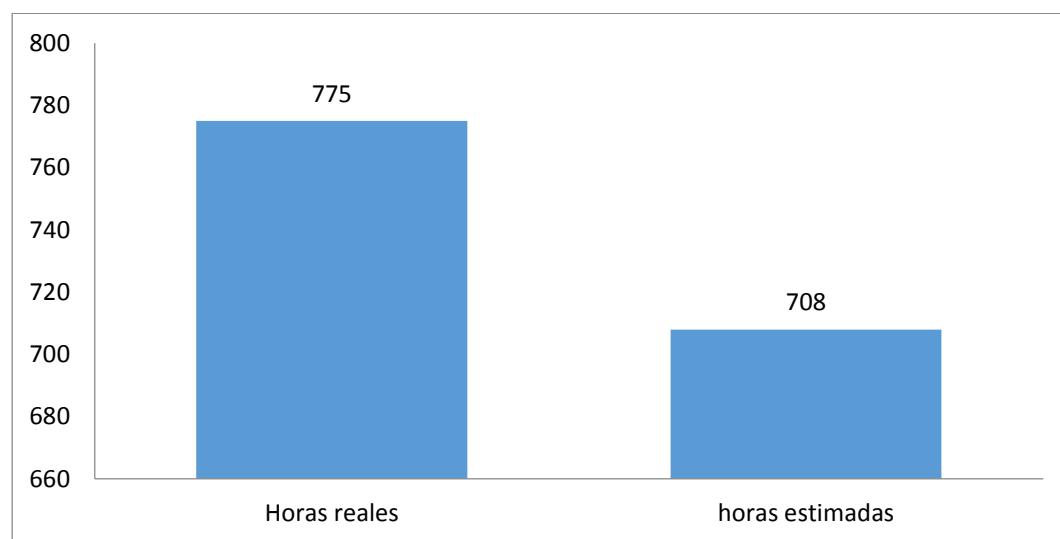


Ilustración 35 Horas reales vs horas planificadas totales.

#### 4.1.2 Horas *testing* vs horas desarrollo

A continuación se detallan las horas de *testing* contra horas de desarrollo. En el proyecto se puso esfuerzo en *testing* ya que es esencial durante el desarrollo, al comienzo se planteó realizar un mínimo de 10 % sobre las horas de desarrollo, finalmente se terminó dedicando un 15,34 %.

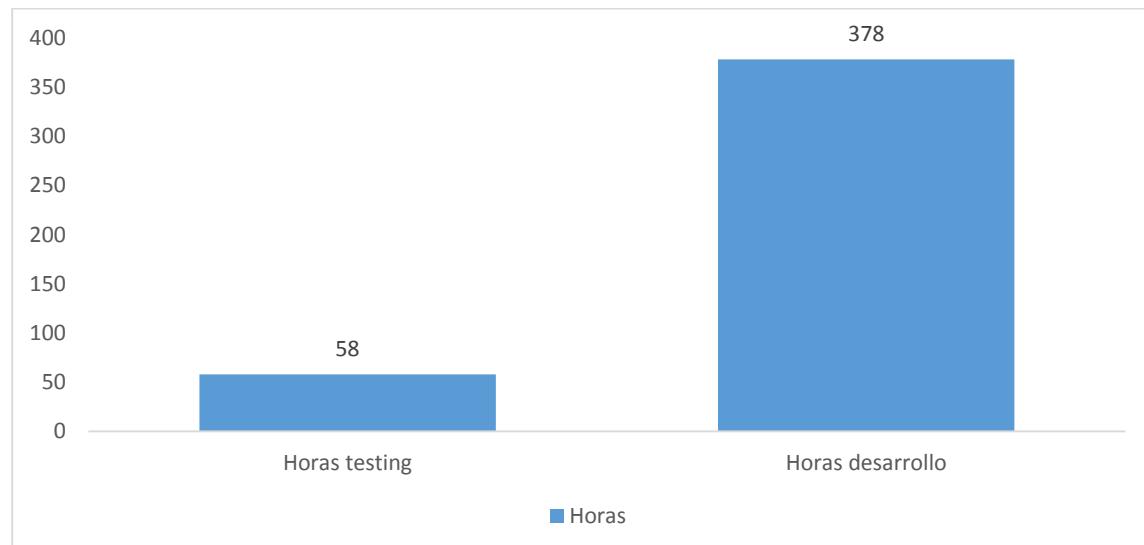


Ilustración 36 Horas de *testing* vs horas de desarrollo.

#### 4.1.3 Tareas realizadas vs tareas planificadas vs tareas nuevas

A continuación se detallan los datos de las tareas, se tomaron en cuenta las tareas planificadas, nuevas y realizadas. En general se cumplieron con las tareas establecidas. Solamente en el primer *sprint* no se logró realizar la totalidad de las tareas. Luego a partir de las re planificaciones se pudo llegar con lo planteado.

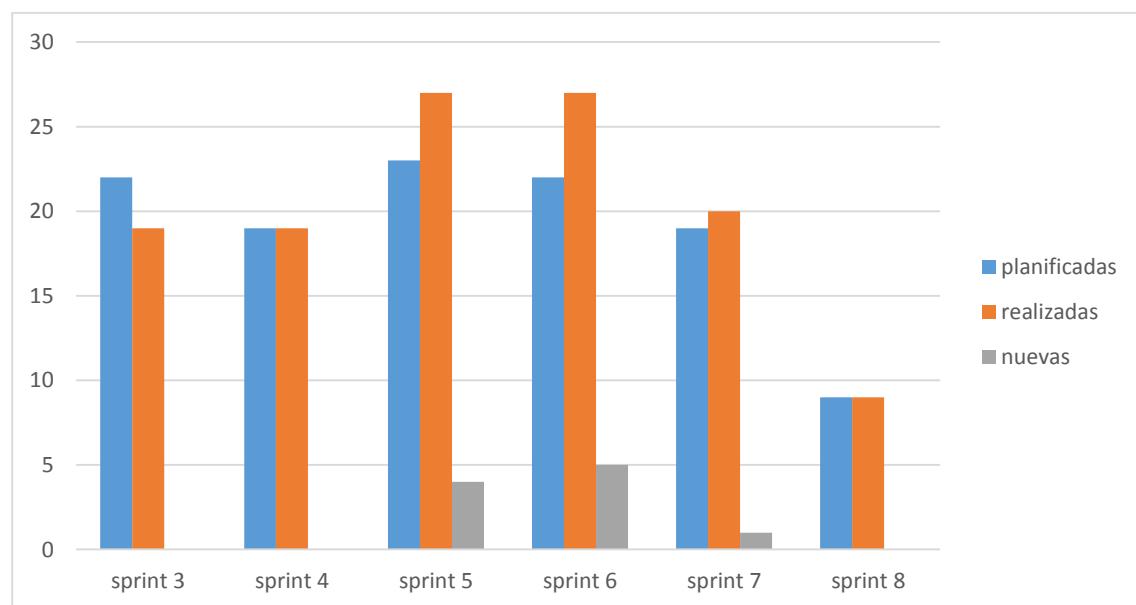


Ilustración 37 Tareas realizadas vs tareas planificadas vs tareas nuevas proyecto.

## 4.2 Satisfacción del cliente

Se le solicitó al cliente Alejandro Gonzalez una devolución de lo que fue el desarrollo del proyecto y acerca del producto obtenido. A continuación se incluye dicha devolución.

“El proyecto realizado por Juan González y Bruno Dutra cumple el objetivo planteado desde un inicio en ser un prototipo funcional para nuestra empresa abarcando los puntos principales de nuestra solicitud y mejorando en algunos aspectos lo solicitado.

La calidad del producto es buena, pero creemos que se puede mejorar, en este punto es que hemos solicitado una revisión en la visualización de los servicios en el mapa.

Asimismo, cabe destacar que se captó muy bien la idea reflejando claramente lo que se deseaba obtener y asimilando en forma satisfactoria la idea de negocio en la que está basada permitiéndonos seguir trabajando sobre ello en nuevas funcionalidades en un futuro.

La comunicación entre el equipo y nuestra empresa fue más que satisfactoria consultándonos entre si habitualmente y con una muy buena fluidez.

Se destaca también la predisposición a escuchar nuevas ideas y la pro actividad de ofrecer nuevas ideas que mejoren el producto. Esto hizo que dicha comunicación mejorará aún más. Para un futuro, definiremos la total integración de los servicios implementados con nuestro sistema para así llevar a cabo la idea que se tenía en un principio.”

## 4.3 Trabajos a futuro

- En el mapa (general o detalle de propiedad) que aparezcan paradas con los ómnibus que pasan.
- Que el usuario tenga la posibilidad de calcular distancias respecto a las propiedades, integrar a la vista parcial del mapa de servicios.
- Desarrollo de casos de prueba unitarios.
- Agregar más parámetros a perfiles.
- Agregar *Google StreetView* a la vista parcial del mapa de servicios.
- Agregar buscador general del sistema.
- Para la funcionalidad de Agenda de visita integrarla con *Google Calendar* para que la misma quede registrada en el momento y se pueda validar que no coincidan las visitas con las de otros usuarios.
- Generar estadísticas de búsqueda para el administrador en función de las búsquedas realizadas por los usuarios en el sistema. Para esto se deberá analizar con el cliente para ver cuáles son sus necesidades.
- Comparación entre más propiedades. Así como también dada una propiedad se muestre para qué tipo de perfil es más adecuada.
- Integrar registro de usuarios con redes sociales.
- Agregar rol de copropietario de edificio.

#### 4.4 Errores conocidos

- En el detalle de propiedad, la carga de los servicios tarda más de lo esperado. Deberían cargarse todos al mismo tiempo.
- Error en Base de datos de *Azure*: Por un tema de balanceo de cargas, el servidor de *Azure* demora no responde la primera vez que se le hace una llamada.
- Mapa de propiedades en portada en algunas ocasiones no toma los marcadores de propiedades en el mapa, luego al refrescar se muestran correctamente.
- Al realizar un comentario sobre una propiedad. Si esta ya tiene varios comentarios de otros usuarios, dicho comentario demorará en ingresarse ya que previamente notifica a los usuarios mencionados que se ha comentado esta propiedad.

#### 4.5 Lecciones aprendidas

Estimaciones de horas: de los aprendizajes que obtuvimos fue estimar las horas con mayor precisión. Al comienzo del proyecto estimamos una cantidad de horas mayor a las que realmente se trabajaron. Luego, durante el desarrollo del producto el equipo fue ajustando las estimaciones, las cuales terminaron siendo más acorde con las horas reales que finalmente les dedicamos a las tareas.

Investigaciones: durante el proyecto, en varios momentos, nos enfrentamos a situaciones que requerían una investigación profunda de lo que finalmente terminaríamos haciendo. A diferencia de los obligatorios hechos durante la carrera, donde la mayoría de las tareas estaban predefinidas o si surgían dudas se consultaba con el profesor de turno, aquí las tareas las definía el equipo y en alguna de ellas, antes de realizarlas se tuvo que hacer una investigación exhaustiva para no terminar en errores. Esto hizo que al terminar el proyecto el equipo obtuviera una experiencia en torno a investigaciones que antes no poseía, y la cual es muy valorada por el mismo.

Metodología ágil: desde las clases teóricas en el comienzo del proyecto el equipo se enfrentó a una nueva metodología de trabajo, *Scrum*. En un principio, además de dichas clases, se hizo un estudio de la misma y una vez comenzado el desarrollo se puso en práctica. El equipo nunca había programado con esta metodología pero podemos afirmar, una vez terminado el proyecto, que fue muy útil y de nuestro agrado haber desarrollado de esta forma.

#### 4.6 Conclusiones de *Testing*

Para asegurar la calidad del *software* se diseñaron una serie de casos de prueba. A cada requerimiento le corresponde un caso de prueba. Los casos de prueba fueron ejecutados al finalizar el desarrollo de cada requerimiento y al culminar cada *sprint*. Las funcionalidades del servicio *web* fueron probadas mediante la herramienta *PostMan* la cual nos permitió evaluar que la salida de cada método fuera la correcta. A su vez, una vez realizada la integración se ejecutaron los casos de prueba de las funcionalidades integradas. [Ver Anexo H: Casos de prueba](#). A su vez se podrá visualizar evidencia de algunos casos base de juego de pruebas como son búsqueda tradicional, búsqueda tradicional con zonas cercanas, búsqueda por perfil y búsqueda por perfil con parámetros. [Ver Anexo W: Evidencias de testing](#).

También se utilizaron casos de prueba unitarios con la herramienta *Nunit*. En este caso se diseñaron casos solamente para las principales funcionalidades del sistema y no se cubrieron todos los casos ya que se priorizó utilizar más tiempo al desarrollo de funcionalidades y dejar como trabajo a futuro el análisis y desarrollo de casos de prueba unitarios. En una posible puesta a producción se deberán desarrollar los casos correspondientes cubriendo la totalidad de las funcionalidades. [Ver Anexo N: Pruebas unitarias.](#)

## 4.7 Conclusiones de SQA y SCM

En lo que refiere a los requerimientos, se siguieron las directrices de los estándares de la *IEEE* para el diseño de los mismos, se toman en cuenta las sugerencias para la elaboración de buenas especificaciones de requerimientos. Esto nos sirvió para entender mejor el sistema y reflejar las necesidades del cliente haciendo que este los valide al comienzo del proyecto.

Para el desarrollo, al guiarnos por los estándares de programación definidos, lo que ayudó al equipo a un mejor entendimiento del código y sus funcionalidades. A su vez estos nos facilitó los cambios y mantenibilidad del código.

Respecto a la documentación nos basamos en las normas del documento 302 de la universidad ORT para la presentación de trabajos de final de carrera. Esto se hizo desde un principio sirviendo como guía lo que nos facilitó las tareas de documentación.

Por último, para el plan de SCM el equipo realizó lo planteado en un principio. No hubo mayores cambios en lo que respecta a funcionalidades. Para el control de versionado se utilizó el repositorio de código *Team Foundation Service de Microsoft* desde el comienzo lo cual facilitó la inclusión de cambios por parte de los integrantes del equipo. [Ver Anexo K Control de versionado.](#)

## 4.8 Conclusiones del proyecto

Como primer punto queremos destacar el aprendizaje y experiencia que obtuvimos al trabajar con un cliente real. Más allá de la experiencia que el equipo pudiese tener en tareas de desarrollo, en este proyecto por primera vez se tuvo un vínculo tan directo con el cliente. Esto hizo que además de la responsabilidad que el equipo tenía con el proyecto en sí, al tratar directamente con un cliente real, la responsabilidad fue aún mayor.

Otro punto a destacar es la diferencia que notamos entre este proyecto y los obligatorios que cursamos en la carrera. Por un lado, los obligatorios realizados ya tenían las tareas predefinidas, en cambio, en dicho proyecto la toma de decisiones durante el recorrido del mismo correspondieron al equipo de desarrollo debiendo evaluar los intereses, objetivos y prioridades del cliente. Para esto, se contó con la ayuda del tutor, el cual nos brindó un soporte y una guía en las decisiones a tomar.

Por otro lado, el proyecto significó una experiencia importante de investigación por el hecho de comenzar un trabajo de este tipo desde el inicio. Para esto, al comienzo se realizó un estudio acerca de los sistemas y aplicaciones que hay en el mercado (nacional e internacional) de este rubro [ver Anexo A: Investigación](#). Por más que se encontraron muchas soluciones similares, el interés del cliente era tener su propio sistema intentando marcar un diferencial en el mercado. También se realizó una investigación exhaustiva respecto a la API

de *Google Places* ya que posee muchas variantes las cuales no se conocían. Esto se hizo para el requerimiento que abarca el mapa con los datos geográficos, el cual en un principio se había optado por realizar desde el lado del servidor pero por un tema de performance y tiempo se tomó la decisión de realizarlo desde el lado del cliente. Otra de las investigaciones importantes que se hizo fue sobre la plataforma *Azure*. Se utilizó para alojar las aplicaciones y base de datos, y también para la automatización de notificaciones mediante *CronJobs*. Por último se realizó un estudio para integrar el manejo de usuario que brinda el *framework (Asp .Net Identity 2.0)* utilizado a nuestro sistema. Una vez integrado, esto nos facilitó el desarrollo de algunos requerimientos ya que muchas tareas ya están resueltas con esta herramienta.

En lo que refiere al desarrollo del proyecto se nos presentó el riesgo de la integración de nuestros servicios con el sistema ya existente del cliente. Por esta razón el equipo se encargó de integrar las principales funcionalidades del servicio. Esto hizo que se debiera realizar una re planificación de tareas para así llevar a cabo la interfaz del usuario y la integración de la *Api*. Por esta razón se utilizaron nuevas tecnologías como *HTML* y *JavaScript* las cuales no se habían usado hasta el momento. Si no hubiéramos tenido que realizar dicha integración se podrían haber implementado nuevas funcionalidades y enriquecer los servicios..

Respecto a los tiempos dedicados en el proyecto a medida que pasaban los *sprints* se fueron mejorando las estimaciones y entregas de tareas. Debido a la poca experiencia en implementación de proyectos y del equipo utilizando metodología *Scrum* en el primer *sprint* no se llegó a cumplir con la totalidad de las tareas. Esto hizo que para los siguientes *sprints* el equipo tomara recaudos para cumplir con las tareas planteadas y comprendiendo las directrices de la metodología *Scrum*.

Una de las principales problemáticas del proyecto fue la conexión de datos del sistema ya existente con el nuestro. Para esto se procedió a realizar un método de importación de propiedades Ver Anexo lo cual nos facilitó para el ingreso de datos de prueba en cantidad. En una eventual puesta en producción se deberá implementar la conexión en tiempo real con los métodos ya expuestos en la *Api*.

Si bien los integrantes del equipo ya habían trabajado juntos a lo largo del curso, esta fue una experiencia que el equipo de desarrollo considera muy positiva, se logró un buen equipo de trabajo en conjunto con el tutor, el cual nos aconsejó y enseñó en este proceso.

En la finalización del proyecto, podemos afirmar que se alcanzó a cumplir con el objetivo más ambicioso planteado en el anteproyecto, de poder apreciar una primera versión de prototipo y obtener el producto deseado por el cliente. A su vez se dedicó un esfuerzo extra para poder brindar un valor agregado al producto desarrollando nuevas tareas que fueron surgiendo en el transcurso del proyecto por ejemplo: comparador de propiedades, mapa de propiedades en portada y listado e inclusión de *Google Analytics* [Ver Anexo V Google Analytics](#).

#### 4.9 Diagrama de arquitectura de la solución:

La solución se encuentra alojada en la nube *Azure*. Se compone de una Base de datos, una *Api Rest Ful* y un cliente web que consume dicha *Api*.

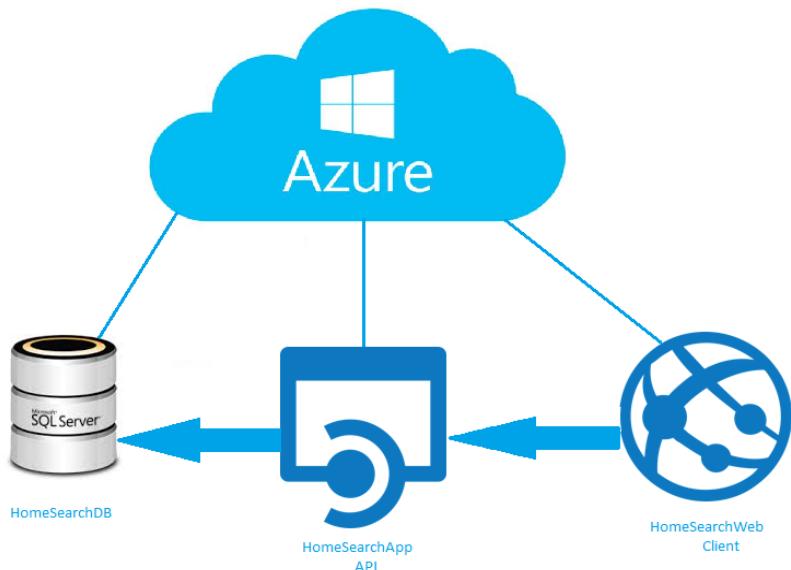


Ilustración 38 Diagrama de arquitectura de la solución.

#### 4.10 Tecnologías utilizadas:

Base de Datos:

Se utiliza el motor de base de datos Microsoft SQL Server 2014. Por otro lado para el modelo de base de datos se utilizó Entity Framework Model First [Ver Anexo R Modelo de base de datos.](#)

Api:

Se utiliza el framework Microsoft .Net 4.5, lenguaje C#. Entity Framework y Web Api.

Cliente Web:

Se utiliza HTML, JavaScript, jQuery, Bootstrap y CSS.

#### 4.11 Conclusiones de riesgos

Una vez concluido el proyecto podemos concluir que durante el mismo no se presentaron mayores dificultades respecto a los riesgos identificados. Por más que surgieron algunos riesgos, los cuales se detallan en el resumen de los *sprints*, se aplicó el plan de contingencia para así poder continuar con el proyecto con normalidad.

A su vez, a lo largo del proyecto surgieron nuevos riesgos. Estos fueron analizados y se tomaron medidas para que no vuelvan a surgir. Los mismos se agregaron a la descripción de riesgos.

Por último, destacar que una vez finalizado el proyecto se presenta el riesgo más importante que es la integración del servicio con la página web. Planteada esta situación, el cliente queda con su página web actual y nuestro proyecto con los servicios por separado. Por lo tanto, en caso de querer realizar la integración con el equipo deberá renegociar las pautas y condiciones para desarrollar esta nueva tarea.

#### 4.11.1 Tabla de riesgos final

Riesgo	Tipo de Riesgo	Probabilidad	Impacto	Plan de contingencia
Rotación del personal	Proyecto	Baja	No se presenta.	Se evaluará si se puede recortar el alcance para que alguno de los integrantes pueda continuar.
Cambio de Requerimiento	Proyecto y producto	Alta	No se presenta.	Prever los cambios de requerimientos en una etapa temprana del proyecto. En caso de surgir cambios durante el mismo se renegociará con el cliente.
Cambio de tecnología	Negocio	Baja	No se presenta.	Renegociar el contrato.
Reducción del presupuesto	Proyecto	Baja	No se presenta.	Que el alcance cuente con lo mínimo indispensable.
Problemas legales	Producto	Baja	No se presenta.	Cambiar la funcionalidad tratando de evitar el problema y a su vez el que cumpla su propósito, en la medida de lo posible.
Nuevas tecnologías	Proyecto y producto	Media	No se presenta.	Evaluar el proyecto junto con el cliente para saber si esas nuevas tecnologías mejorarían sustancialmente el mismo.
Incumplimiento de plazos	Proyecto	Medio	Se presenta en el tercer <i>sprint</i> . Se asignaron las tareas para el siguiente <i>sprint</i> .	Renegociar calendario con el cliente.

Incumplimiento de integración	Proyecto	Medio	Se presenta en el sexto <i>sprint</i> . El cliente informa que no podrá cumplir en tiempo y forma con la integración. El equipo re planifica las horas para proporcionar ejemplos de la integración.	Proporcionar las firmas de métodos y la documentación técnica lo antes posible. En caso de no cumplirse la integración el equipo tendrá que tener ejemplos de integración para entregar en el proyecto.
Horas reales de trabajo menores a las horas estimadas	Proyecto	Alta	Se Presenta, se re planificaron las horas para los siguientes <i>sprints</i> .	Al presentarse este riesgo se debe replantear las horas estimadas para los siguientes <i>sprints</i> . A su vez buscar de re organizar los horarios de trabajo para que no se vuelva a presentar.
Errores en la estimación de tareas	Proyecto	Alta	Se presenta. En el sprint 3. Se replantean las tareas.	Se debe replantear las siguientes tareas para poder reducir este riesgo.
Contriatiempos con equipos personales	Proyecto	Media	No se presenta.	Se debe buscar remplazar el equipo lo antes posible, también dar solución al desperfecto. También se cuenta con los equipos de laboratorio de la universidad. En este caso se asistió 2 veces a este recurso.

Integridad de datos del sistema.	Producto	Medio	No se presenta.	En lo que refiere a datos cuando el sistema se suba a un ambiente de producción, se deberá hacer una política de respaldo.
Integración del servicio con la página web.	Producto	Medio	Se presenta una vez finalizado el proyecto.	El cliente podrá contratar al equipo u otro para que integre su sitio web actual con la Api. Renegociar con el cliente.

Tabla 23 Tabla de riesgos final.

## 5 Referencias bibliográficas

- [1] *IEEE Recomended Practice for Software Requirement Especifications* ANSI/IEEE 830 1998. [Online] Disponible: <http://www.cse.msu.edu/~chengb/RE-491/Papers/IEEE-SRS-practice.pdf>
- [2] Convenciones de código de C# (Guía de programación de C#). [Online] Disponible: <https://msdn.microsoft.com/es-es/library/ff926074.aspx>
- [3] Documento 302 Normas específicas para la presentación de trabajos finales de carrera (TFDC). [Online] Disponible: [https://aulas.ort.edu.uy/pluginfile.php/237122/mod\\_resource/content/1/Documento%20302%20ET.pdf](https://aulas.ort.edu.uy/pluginfile.php/237122/mod_resource/content/1/Documento%20302%20ET.pdf)
- [4] *ADO.NET Entity Framework Documentation*. [Online] Disponible: [https://msdn.microsoft.com/en-us/library/bb399567\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/bb399567(v=vs.110).aspx)

## 6 Bibliografía

A. Freedman, *Expert ASP.NET Web API 2 for MVC Developers*. 1a ed. New York, NY, USA: Apress, 2014.

BOOTSTRAP. S.f. [Online]. Disponible: <http://getbootstrap.com/>

Google. *Google Maps JavaScript API*, documentación sobre *Google Maps* [Online]. Disponible: <https://developers.google.com/maps/documentation>

Google. *Google Places API*, documentación sobre *Google Places* [Online]. Disponible: <https://developers.google.com/places>

jQury – Referencia Ajax requests, y múltiples consultas a la documentación [Online] Disponible: <https://jquery.com/>

MSDN “Microsoft Developer Network” [Online]. Disponible: <https://msdn.microsoft.com>

Refsnes Data. Documentación y referencias de HTML5, CSS y jQuery [Online]. Disponible: <http://www.w3schools.com>

Stack Exchange Inc. Foro y tutoriales [Online]. Disponible: <http://stackoverflow.com/>

Sommerville, *Ingeniería del Software*, 9a ed. Pearson 2011.

Student2Student, *Primer acercamiento a Windows Azure*, Abril 2012. [Online]. Disponible: <https://studentsexperience.wordpress.com/2012/04/16/primer-acercamiento-a-windows-azure>

## 7 Anexos

### 7.1 Anexo A: Investigación

En el mercado *web* inmobiliario existen varios sitios que proveen servicio de búsqueda de inmuebles.

La gran mayoría ofrece un filtrado por los parámetros de búsqueda tradicionales de una propiedad, los sitios no muestran información sobre la zona (servicios y otros datos).

#### 7.1.1 Mercado local

##### **[www.infocasas.com.uy](http://www.infocasas.com.uy)**

Portal *web* que anuncia propiedades tanto de inmobiliarias como de particulares. Ofrece buscar propiedades por tipo de operación (venta, alquiler o alquiler temporal), tipo de propiedad, ubicación, rango de precios, dormitorios y baños. En el detalle de la propiedad despliega un mapa con la ubicación. No dispone información sobre los servicios que posee a su alrededor.

##### **[tucasaaqui.com.uy](http://tucasaaqui.com.uy)**

Portal inmobiliario del diario EL PAÍS, ofrece la búsqueda tradicional donde se pide completar campos como el tipo de inmueble, operación, departamento, país, barrio y otro tipo de búsqueda por perfil, algo parecido a lo que nosotros queremos implementar. En esta, distingue los perfiles en 4 categorías: pareja joven, solteros/as, familias y profesionales. Una vez seleccionado el perfil, destaca los aspectos a tener en cuenta y el porqué.

En la búsqueda “común”, se pide ingresar los campos mencionados y a partir de allí realiza una búsqueda general. Luego de mostrado el resultado, se podrá acotar la búsqueda a algo más específico.

En la búsqueda por perfil, se deja constancia del criterio que se usará para la búsqueda y luego muestra el resultado. En este caso los filtros no son configurables. La búsqueda será general, y luego el interesado podrá quitar y/o agregar distintos parámetros. A su vez, a pesar de indicar la zona en la que deseas realizar la búsqueda, al ser un resultado general, se deberá mediante filtros llegar a la zona deseada.

En el detalle de la propiedad sugerida mediante la búsqueda realizada muestra información y fotos, y para la zona solamente dispone de un link a un mapa con la ubicación. No ofrece ningún tipo de información sobre la zona y servicios.

Además posee una búsqueda por tipo (apto., casa, oficina, etc), por operación, por departamento.

Por último, mencionar que tiene una sección de propiedades destacadas y lanzamientos.

## 7.1.2 Mercado internacional

### **www.immobilienscout24.de**

Portal inmobiliario de Alemania. Cuenta con un buscador de propiedades por zona o ciudad, cantidad de dormitorios, rango de precios, etc.

En el detalle de propiedad se despliega un mapa con la ubicación de la propiedad, a su vez muestra los lugares de interés que existen alrededor, pudiendo filtrarlos por tipo (supermercado, comidas, escuelas, médicos, transporte, etc.).

### **www.trulia.com**

Portal inmobiliario de Estados Unidos. Busca por zona o ciudad. En el detalle de propiedad, además de fotos y toda la información se muestra un mapa con los comercios y lugares en los alrededores pudiendo filtrarlos por tipo. Dispone de una opción para mostrar los crímenes reportados en la zona. También se pueden mostrar los datos demográficos. Posee una opción para calcular el tiempo hacia un determinado lugar como por ejemplo el trabajo.

### **Scorpios.com.ar**

Para la búsqueda de propiedades tiene un link y desde allí varios links para llegar a lo que se quiere. No hay búsqueda sino que son varios links.

Por ejemplo: buscar propiedades → ventas → locales → “muestra locales que hay a la venta”

### **Pabellon.cl**

Posee un buscador de propiedades donde se define si es alquiler o venta.

En el mismo se puede definir uno o varios tipos de lo que desea (aptos., casa, oficina, terreno, hotel), seleccionar la zona y el rango de precios.

Como novedad, tiene otro tipo de búsqueda “otras búsquedas” donde allí se puede seleccionar alquiler de estudiantes, propiedades amuebladas, remates, tasaciones y servicios, propiedades en el extranjero, piezas con pensión arriendan, etc donde al seleccionar algún tipo de esta búsqueda muestra los distintos inmuebles disponibles.

En caso de no encontrar lo que el cliente desea existe la opción de solicitar que en caso de que se publique una propiedad con esas características se avise.

Además se puede seleccionar solo las propiedades que tengan foto.

## 7.2 Anexo B: Justificación del producto

### 7.2.1 Respuesta del equipo luego de reunión con cliente

El proyecto se realizará en función del interés del cliente. El mismo, desea integrar a su aplicación *web* ya existente un servicio de búsqueda para los inmuebles que la inmobiliaria tiene en cartera.

La búsqueda, además de tener los filtros tradicionales (llámese: barrio, cantidad dormitorios, baños, etc.), contará con la posibilidad de buscar entorno al perfil del interesado especificando sus características y/o contexto. Los mismos pueden ser: cuántas personas habitarán el inmueble, estado civil, con hijos/sin hijos, trabajo, intereses personales y otros.

En comparación con el mercado, este no es un servicio que se ofrezca comúnmente y por lo que es otra de las razones por la cual el cliente quiere realizar su propio proyecto.

También agregar que la ubicación es un factor clave a la hora de buscar una propiedad, por esto se desea que la funcionalidad resalte los servicios que se encuentran en las proximidades del inmueble sugerido.

Muchos inversores buscan propiedades en base al tipo de inquilinos que desean tener. Por lo tanto el sistema de búsqueda permitirá al inversor tener las mejores opciones.

Por otro lado, la página permitirá autenticar usuarios. Estos podrán hacer un seguimiento de las propiedades, puedan realizar consultas, ofertas y agendar visitas. Con esto, lo que nuestro cliente busca es "fidelizar" a los usuarios y que el manejo de inmuebles se pueda ver como una "red social". Es decir, una vez interesado en una propiedad si él lo solicita le podrán llegar las novedades de la misma así como también sugerencias de inmuebles similares.

### 7.2.2 Respuesta del cliente

La solicitud del proyecto en referencia es debido a la continua búsqueda de la empresa de brindar un servicio más personalizado. Ayudar a los futuros compradores o interesado en alquilar a afinar su búsqueda mediante parámetros generales de acuerdo a su situación actual.

Dentro de esos parámetros se puede establecer su estado civil, la locación de su empleo, sus lugares de mayor interés, gustos hasta de sus ingresos aproximados para evaluar cantidad de habitaciones "necesarias", servicios de locomoción disponibles, servicios cerca, etc. En caso de que en nuestra cartera no haya nada que se asemeje a sus intereses poder tener un detalle lo más aproximado de sus gustos para poder informarle mediante un correo cualquier novedad acerca de su búsqueda.

Actualmente el mercado ofrece diversas opciones de búsquedas en general pero que no fidelizan al cliente y muchas veces la búsqueda se vuelve frustrante al aparecer varias opciones repetidas, sin ser lo que uno busca y con poca información. Nosotros buscamos darle mayor especificación al cliente y así también poder captar más propietarios que quieran trabajar con nosotros.

### 7.3 Anexo C: Explicación de notificaciones

- Comentarios:  
Una vez que un Usuario Autenticado realiza un comentario en una propiedad, luego cada vez que otros usuarios comenten en la misma, a dicho Usuario Autenticado le llegarán notificaciones de los nuevos comentarios ingresados.
- Sugerencia de propiedad  
A cada Usuario Autenticado se le brindará sugerencia de propiedades. Para ello, la sugerencia se realizará de la siguiente manera:  
Búsquedas realizadas por dicho usuario en el sistema: dado el historial de búsquedas del mismo, al tener una propiedad que coincida con los parámetros búsqueda de dichas búsquedas se le sugerirá la mencionada propiedad.
- Pedido de propiedad  
Una vez que un Usuario Autenticado realice un pedido de propiedad, al mismo le llegarán notificaciones acerca de ese pedido que realizó. En las mismas, le llegarán propiedades que tengan las características realizadas en el pedido. Dichas propiedades serán de la cartera de López & Cia.
- Modificaciones en propiedad favorita  
Una vez que un Usuario Autenticado agrega una propiedad como favorita, luego todo cambio que se produzca en la misma será notificado al usuario.

## 7.4 Anexo D: Explicación de búsqueda por perfil

El sistema permitirá realizar búsquedas de propiedades en base a perfiles de usuarios. Cada perfil tendrá configurado una serie de parámetros lo que ayudará a realizar una búsqueda más precisa. En caso de que el usuario desee especificar algún parámetro (por ejemplo: cantidad de dormitorios) deberá incluir los filtros de la búsqueda tradicional.

En lo que refiere a los parámetros incluidos en los perfiles, estos se tomarán en cuenta para que la búsqueda se realice en función de servicios que puedan ser de su interés. Los mismos no excluirán propiedades en el resultado sino que servirán para ordenar el mismo mostrando primero aquellas propiedades que más servicios cercanos y de su interés se encuentren en las proximidades de las mismas.

A continuación se detallan los perfiles ingresados en el sistema.

- Pareja: se buscarán propiedades de uno y dos dormitorios y a su vez el tipo de propiedad será casa o apartamento.
- Solteros: se buscarán propiedades de un dormitorio, y las mismas serán del tipo casa o apartamento.
- Empresas: se buscarán propiedades grandes, apartamentos o casas de dos y tres dormitorios así como también locales comerciales.
- Estudiantes: se buscarán propiedades grandes, de dos, tres y cuatro dormitorios, con uno o dos baños y el tipo de propiedad será casa o apartamento.
- Familia: se buscarán propiedades grandes, de dos, tres y cuatro dormitorios, con uno o dos baños y el tipo de propiedad será casa o apartamento.

## 7.5 Anexo E: Ingreso de propiedades de prueba

Para el ingreso de propiedades de prueba entendíamos que los datos deberían ser en cantidad, mínimo contar con 50 propiedades, lo que nos ayudaría para el *testing*. Para ello, el equipo realizó un método para la lectura de un archivo csv con los atributos de cada propiedad. Luego se contactó al desarrollador del sistema *web* de López & Cia. Por más que finalmente no pudo cumplir con la integración de la API en dicho sistema, él quedó a disposición para cualquier dificultad que se nos presentara con lo que ya tenía hecho. En ese caso, nuestra dificultad era realizar la importación de todas las propiedades desde *Magento* (plataforma utilizada por el desarrollador) a un archivo csv lo cual con su ayuda se pudo realizar. Una vez creado el archivo y luego de ejecutar el método mencionado anteriormente, quedan ingresadas la totalidad de las propiedades con las que cuenta la inmobiliaria.

## 7.6 Anexo F: Explicación de parámetros adicionales en la búsqueda por perfil

Al elegir realizar una búsqueda por perfil, se desplegarán determinados parámetros que el usuario podrá seleccionar o no para efectuar la búsqueda. Dichos parámetros ayudarán a realizar una búsqueda con mayor exactitud a lo que el usuario desea. Para ello, a partir de los mismos se tendrán en cuenta los servicios que más puedan serle útil en los alrededores de la propiedad buscada. Estos parámetros no excluirán ninguna propiedad del resultado de la búsqueda sino que el resultado se ordenará mediante la propiedad que más servicios de utilidad tenga en sus proximidades. Cada parámetro tendrá varios servicios a buscar, para esto se crea una tabla de configuración en la base de datos.

## 7.7 Anexo G: Casos de uso

Cada caso de uso está descrito de la siguiente manera:

<b>Nombre</b>	<i>Approve Request</i>	
<b>Actores</b>	Administrador, Usuario Autenticado	
<b>Descripción</b>	El caso de uso comienza cuando el Usuario Autenticado envía una solicitud para ingresar un inmueble al Sistema. La misma, llega al Administrador el cual en función de los datos enviados por el Usuario Autenticado decide si aprueba la solicitud o no. La respuesta se envía vía mail.	
<b>Pre-Condición</b>	El Administrador debe estar autenticado como tal al igual que el Usuario Autenticado.	
<b>Post-Condición</b>	La propiedad queda aprobada o no, lo cual se informa al usuario vía mail.	
<b>Especificación</b>		
<i>Actores</i>		<i>Sistema</i>
1 – El Usuario Autenticado ingresa al sistema seleccionando la opción para el ingreso de propiedades.		
		2 – Despliega la opción para el ingreso de propiedades
3 – El Usuario Autenticado ingresa la propiedad completando los datos necesarios al sistema: dirección, ciudad, barrio, precio en que se estima la venta y si esta regularizado en BPS. A su vez, tendrá la posibilidad de ingresar un texto para completar la información.		
		4 – Se verifican los campos ingresados
		5 – Llega al Sistema la solicitud que será notificada al Administrador para evaluar.
6 – El Administrador acepta la solicitud.		
		7 – El Sistema informa al Usuario Autenticado vía mail que su solicitud fue aprobada con el mensaje: “Su solicitud de ingreso de inmueble ha sido aceptada. Nos contactaremos a la brevedad”.

Tabla 24 Caso de uso *ApproveRequest*.

Cursos Alternativos:

4.1: Si algún dato es vacío o incorrecto el sistema deberá devolver un error y se vuelve al paso 3.

6.1: El administrador no acepta la solicitud

7.1: El Sistema informa al Usuario Autenticado vía mail que su solicitud no fue aprobada con el mensaje: “Su solicitud de ingreso de inmueble no ha sido aceptada. Por consultas contáctenos a [administracion@lopezycia.com.uy](mailto:administracion@lopezycia.com.uy).”

Nombre	<i>Add Property</i>	
Actores	Administrador	
Descripción	El caso de uso comienza cuando el Administrador quiere agregar una propiedad al Sistema. Para ello, el mismo ingresa la información del inmueble así como las fotos para luego darla de alta. La información a ingresar es:	
Pre-Condición	El Administrador debe estar autenticado como tal.	
Post-Condición	La propiedad queda ingresada al sistema	
<b>Especificación</b>		
<i>Actor</i>		<i>Sistema</i>
1 - El Administrador ingresa al sistema seleccionando la opción para el alta de una propiedad.		
		2 – Despliega la opción para el alta de propiedades
3 – El Administrador ingresa la propiedad completando los datos necesarios al sistema: título, descripción, precio, tipo, imágenes (eligiendo una principal), dirección, propiedad destacada, cantidad de baños, tipo de cocina, tipo de living, si contiene portero eléctrico, cierre perimetral, patio, parrillero, metrajes de la superficie, año de construcción y estado general de la propiedad.		
		4 – Se verifican los campos ingresados.
		5 – El sistema informa mediante el mensaje “La propiedad fue dada de alta” que el la propiedad fue ingresada al sistema.

Tabla 25 Caso de uso *Add Property*.

#### Cursos Alternativos:

4.1: Si algún dato es vacío o incorrecto el sistema deberá devolver un error y se vuelve al paso 3.

Nombre	<i>Delete Property</i>
Actores	Administrador
Descripción	El caso de uso comienza cuando el Administrador desea eliminar una propiedad del Sistema. Para ello, el mismo selecciona el inmueble para luego darlo de baja.
Pre-Condición	El Administrador debe estar autenticado como tal.
Post-Condición	La propiedad queda eliminada del sistema.
<b>Especificación</b>	
<i>Actor</i>	<i>Sistema</i>
1 - El Administrador ingresa al sistema seleccionando la opción para la baja de una propiedad.	
	2 – Despliega la opción para la baja de propiedades
3 – El administrador indica el “externalId” de la propiedad a eliminar.	
	4 – Se verifica que el “externalId” esté ingresado en el sistema.
	5 – Se elimina la propiedad del sistema.

Tabla 26 Caso de uso *Delete Property*.

#### Cursos Alternativos:

4.1: Si el dato ingresado no existe en el sistema deberá devolver un error y se vuelve al paso.

Nombre	<i>Modify Property</i>
Actores	Administrador
Descripción	El caso de uso comienza cuando el Administrador quiere modificar una propiedad del Sistema. Para ello, el mismo el mismo selecciona el inmueble para luego modificar los datos y/o fotos que se deseen.
Pre-Condición	El Administrador debe estar autenticado como tal.
Post-Condición	La propiedad queda modificada en el sistema.
<b>Especificación</b>	
<i>Actor</i>	<i>Sistema</i>
1 - El Administrador ingresa al sistema seleccionando la opción para la modificación de una propiedad.	
	2 – Despliega la opción para la modificación de propiedades
3 – El administrador cambia el campo deseado.	
	4 – Se verifica que todos los datos sean

	validos
	5 – La propiedad queda con la modificación hecha.

Tabla 27 Caso de uso *Modify Property*.

#### Cursos Alternativos:

4.1: Si los datos no son válidos el sistema deberá devolver un error y se vuelve al paso 3.

Nombre	<i>Logout</i>	
Actores	Usuario Autenticado y Administrador	
Descripción	El caso de uso comienza cuando el Usuario Autenticado o el Administrador desean cerrar sesión en el Sistema. Para ello, se les brinda la posibilidad de una vez ingresado al sistema poder salir del mismo.	
Pre-Condición	El Administrador/Usuario Autenticado están ingresados al sistema como tales.	
Post-Condición	El usuario queda deslogueado del sistema.	
<b>Especificación</b>		
<i>Actor</i>		<i>Sistema</i>
1 - El Usuario Autenticado/Administrador selecciona la opción para desloguearse.		
		2 – Despliega la opción para salir del sistema.
3 – Confirma la salida del sistema.		

Tabla 28 Caso de uso *Logout*.

Nombre	<i>Reset Password</i>	
Actores	Usuario Autenticado y Administrador	
Descripción	El caso de uso comienza cuando el Usuario Autenticado o el Administrador desean restablecer la contraseña. Para ello, el Sistema brinda la posibilidad de poder cambiar la contraseña. El usuario marca dicha opción y luego el Sistema solicita ingresar la contraseña a cambiar y la nueva contraseña. Una vez ingresados, la contraseña queda reestablecida.	
Pre-Condición	El Administrador/Usuario Autenticado están ingresados al sistema como tales.	
Post-Condición	Queda reestablecida la contraseña	
<b>Especificación</b>		
<i>Actor</i>		<i>Sistema</i>
1 - El Usuario Autenticado/Administrador selecciona la opción para cambiar la contraseña.		
		2 – Despliega la opción para el cambio de contraseña.
3 – El Administrador/Usuario Autenticado		

ingresa su contraseña actual, la nueva contraseña y confirma su nueva contraseña.	
	4 – Se verifica que los datos ingresados sean válidos.
	5 – Se confirma el cambio de contraseña.

Tabla 29 Caso de uso *Reset password*.

#### Cursos Alternativos:

4.1: Si los datos no son válidos el sistema deberá devolver un error y se vuelve al paso 3.

Nombre	<i>Recover Password</i>
Actores	Usuario autenticado y Administrador
Descripción	El caso de uso comienza cuando el Usuario Autenticado o el Administrador desean recuperar la contraseña. Para ello, el Sistema brinda la posibilidad de poder recuperarla indicando la opción “Recuperar Contraseña”. Una vez indicada, se envía un mail al ingresado por el Usuario al momento del registro, detallando cual es la contraseña.
Pre-Condición	No hay.
Post-Condición	El Administrador/Usuario Autenticado queda registrado con una nueva contraseña.
<b>Especificación</b>	
<i>Actor</i>	<i>Sistema</i>
1 – Ingresa a la opción para recuperar la contraseña.	
	2 – El sistema despliega la opción para recuperar contraseña solicitándole al Administrador/Usuario Autenticado su correo electrónico.
3 – El Administrador/Usuario Autenticado ingresa su correo electrónico solicitando recuperar su contraseña.	
	4 – Se verifica que el correo electrónico ingresado exista en el sistema.
	5 – Se envía un mail al correo electrónico ingresado con la nueva contraseña.
	6 – El Administrador/Usuario Autenticado queda registrado en el sistema con una nueva contraseña.

Tabla 30 Caso de uso *Recover password*.

#### Cursos Alternativos:

4.1: Si el correo electrónico no existe en el sistema se deberá devolver un error y se vuelve al paso 3.

Nombre	<i>Login</i>
Actores	Usuario autenticado y Administrador
Descripción	El caso de uso comienza cuando el Usuario Autenticado o el Administrador desean ingresar al Sistema. Para ello, los mismos deben ingresar su mail (nombre de usuario) y su contraseña. Una vez ingresados, el Sistema valida si los mismos son válidos. En caso de que lo sean, se ingresa al sistema. Caso contrario, se envía una alerta informando que los datos no son correctos dando la posibilidad de ingresar los datos nuevamente.
Pre-Condición	No hay
Post-Condición	Queda ingresado en el sistema un nuevo Administrador/Usuario Autenticado
<b>Especificación</b>	
<i>Actor</i>	<i>Sistema</i>
1 – Ingresa a la opción para registrarse en el sistema.	
	2 – Despliega la opción para registrarse en el sistema.
3 – Se ingresan los siguientes datos para registrarse: nombre, apellido, correo electrónico, documento de identidad, contraseña y teléfono.	
	4 – Se valida que los datos sean correctos y que el usuario no se encuentre registrado con el mismo correo electrónico y/o documento de identidad.
	5 – El usuario queda registrado en el sistema.

Tabla 31 Caso de uso *Login*.

#### Cursos Alternativos:

4.1: Si el correo electrónico no existe en el sistema se deberá devolver un error y se vuelve al paso 3.

Nombre	<i>Delete Favorite Property</i>
Actores	Usuario autenticado
Descripción	El caso de uso comienza cuando un Usuario Autenticado desea eliminar de su lista de propiedades favoritas una de ellas. Para ello, el Sistema le otorga la posibilidad de “desmarcar” una propiedad marcada como favorita. Una vez hecho esto último, al Usuario Autenticado no le llegarán más novedades acerca de la propiedad mencionada.
Pre-Condición	
Post-Condición	
<b>Especificación</b>	
<i>Actor</i>	<i>Sistema</i>

1. El usuario, dada una propiedad, selecciona la opción eliminar propiedad favorita.	2. El sistema verifica que el usuario se encuentre registrado al sistema.
	3. El sistema verifica que la propiedad exista en el sistema.
	4. El sistema elimina la propiedad de la lista de favoritas de ese usuario y notifica.
5. El usuario recibe la confirmación.	

Tabla 32 Caso de uso *Delete favorite property*.

#### Cursos Alternativos:

2.1: Si el usuario no se encuentra registrado el sistema deberá retornar el error correspondiente y se vuelve al paso 1.

3.1: Si la propiedad no existe el sistema deberá retornar el error correspondiente y se vuelve al paso 1.

Nombre	<i>Add Favorite Property</i>	
Actores	Usuario autenticado	
Descripción	El caso de uso comienza cuando un Usuario Autenticado desea agregar una propiedad de su interés a su lista de favoritos. Para ello, al momento de listar las propiedades el Sistema brinda la posibilidad de poder seleccionar una propiedad como favorita. Una vez marcada como tal, al usuario autenticado le llegarán todas las novedades que haya de la propiedad mencionada.	
Pre-Condición		
Post-Condición		
<b>Especificación</b>		
<i>Actor</i>		<i>Sistema</i>
		1. El sistema listará las propiedades.
2. El usuario autenticado selecciona una propiedad como favorita.		
		3. El sistema agrega la propiedad seleccionada a la lista de favoritas de ese usuario y envía un mensaje de confirmación.
4. El usuario ve el mensaje de confirmación y la propiedad marcada como favorita.		
		5. El sistema enviará notificaciones vía correo electrónico cuando las propiedades ingresadas como favoritas sufran modificaciones de

	precio.
--	---------

Tabla 33 Caso de uso *Add Favorite property*.

Cursos Alternativos:

- 3.1: Si el usuario no se encuentra registrado se deberá retornar el error correspondiente y se regresa al paso 1.
- 3.2: Si la propiedad no existe el sistema deberá retornar el error correspondiente y se regresa al paso 1

Nombre	<i>Order Property</i>
Actores	Usuario autenticado
Descripción	El caso de uso comienza cuando un Usuario Autenticado desea agregar hacer un pedido de una propiedad con ciertas características. Para ello se le solicitará que ingrese datos del pedido. Luego estas solicitudes deberán ser listadas al usuario administrador para que pueda enviarle información en base al pedido. A su vez El sistema deberá enviar información en caso de que se ingrese una propiedad con las características del pedido ingresado.
Pre-Condición	
Post-Condición	
<b>Especificación</b>	
Actor	<i>Sistema</i>
	<ol style="list-style-type: none"> <li>1. El sistema solicita las características del inmueble deseado. Cantidad de dormitorios, cantidad de baños, tipo de inmueble, tipo de operación, ciudad y barrio.</li> </ol>
2. El usuario ingresa los campos solicitados.	
	<ol style="list-style-type: none"> <li value="3">3. El sistema verifica los datos ingresados, y notifica al cliente.</li> </ol>
	<ol style="list-style-type: none"> <li value="4">4. El sistema ingresa la orden de propiedad y se la agrega al administrador de sistemas.</li> </ol>

Tabla 34 Caso de uso *Order property*.

Curso Alternativo:

- 3.1: en caso de que los campos ingresados no sean correctos, el sistema deberá retornar el error correspondiente y se regresa al paso 1.

Nombre	<i>Enter property request</i>
Actores	Usuario autenticado
Descripción	El caso de uso comienza cuando un Usuario Autenticado desea ingresar una solicitud de alta de propiedad. Para esto se solicitarán los datos de la propiedad que desea ingresar. Luego estas solicitudes serán asignadas al usuario administrador para que evalúe la solicitud y pueda aprobar la solicitud.
Pre-Condición	
Post-Condición	
<b>Especificación</b>	
Actor	Sistema
	<ol style="list-style-type: none"> <li>1. EL sistema solicita los datos de la propiedad: dirección, la ciudad y el barrio de la propiedad, el precio en que desea vender la misma y regularización BPS.</li> </ol>
2. El usuario ingresa los datos de su propiedad.	<ol style="list-style-type: none"> <li value="2">3. El sistema verifica los datos e ingresa la solicitud, asignándola al usuario administrador.</li> </ol>
4. El usuario recibe la confirmación.	

Tabla 35 Caso de uso *Enter property request*.

#### Cursos Alternativos:

3.1: Todos los campos serán mandatorios, en caso de que no haya ingresado alguno de los mismos el sistema deberá retornar el error correspondiente.

Nombre	<i>Add comment</i>
Actores	Usuario autenticado
Descripción	El caso de uso comienza cuando un Usuario Autenticado desea ingresar un comentario a una propiedad. Para esto se solicita que el usuario ingrese un mensaje, luego este mensaje será agregado a la propiedad.
Pre-Condición	
Post-Condición	
<b>Especificación</b>	
Actor	Sistema
	<ol style="list-style-type: none"> <li>1. EL sistema solicita que se ingrese el mensaje.</li> </ol>
2. El usuario ingresa el mensaje-	

	3. EL sistema asigna el mensaje a la propiedad.
4. El usuario recibe la confirmación.	

Tabla 36 Caso de uso *Add comment*.

Nombre	<i>Modify comment</i>
Actores	Usuario autenticado
Descripción	El caso de uso comienza cuando un Usuario Autenticado desea modificar un comentario.
Pre-Condición	
Post-Condición	
<b>Especificación</b>	
<i>Actor</i>	<i>Sistema</i>
	1. EL sistema solicita que se ingrese el nuevo mensaje.
2. El usuario ingresa el mensaje.	
	3. EL sistema asigna el mensaje a la propiedad.
4. El usuario recibe la confirmación.	

Tabla 37 Caso de uso *Modify comment*.

Nombre	<i>Delete comment</i>
Actores	Usuario autenticado
Descripción	El caso de uso comienza cuando un Usuario Autenticado desea eliminar un mensaje.
Pre-Condición	
Post-Condición	
<b>Especificación</b>	
<i>Actor</i>	<i>Sistema</i>
1. Dado un mensaje de una propiedad el usuario selecciona eliminar mensaje.	
	El sistema elimina el mensaje.
1. El usuario recibe la confirmación.	

Tabla 38 Caso de uso *Delete comment*.

Nombre	<i>Schedule visit</i>
Actores	Usuario autenticado
Descripción	
Pre-Condición	
Post-Condición	
<b>Especificación</b>	
<i>Actor</i>	<i>Sistema</i>
1. Dada una propieda el usuario selecciona la fecha y hora que desea realizar la visita.	
	2. El sistema guarda la fecha y hora de visita y envía un mail al usuario administrador.
3. El usuario recibe la confirmación y queda a la espera que el usuario administrador se comunique para concretar la visita.	

Tabla 39 Caso de uso *Schedule visit*.

## 7.8 Anexo H: Casos de prueba

### Búsqueda tradicional

objetivo	esperado	pasos	resultado
Realizar una búsqueda de propiedades mediante parámetros de propiedad	el sistema debe devolver una lista de propiedades en base al criterio seleccionado	1. Ingresar todos los datos de la búsqueda: cantidad de dormitorios, cantidad de baños, ciudad, barrio, si posee garage, tipo de propiedad, tipo de operación y tipo de propiedad.	El sistema debe retornar las propiedades que cumplan con las características seleccionadas.
		2. Ingresar solamente los datos de zona de la búsqueda: ciudad y barrio	El sistema debe retornar las propiedades que se encuentren en la ciudad y barrio ingresados
		3. Ingresar solamente los datos de comodidades de propiedad: cantidad de dormitorios, cantidad de baños y garage.	El sistema debe retornar todas las propiedades que posean las comodidades seleccionadas
		4. Ingresar solamente los parámetros de tipo de operación.	El sistema debe retornar solamente las propiedades que cumplan con el tipo de operación seleccionado
		5. Ingresar solamente los parámetros de tipo de propiedad.	El sistema deberá retornar solamente las propiedades que sean del tipo de propiedad ingresado.
		6. No ingresar parámetros de búsqueda.	El sistema deberá retornar una lista vacía.

Tabla 40 Caso de prueba búsqueda tradicional.

## Búsqueda por perfil

objetivo	esperado	pasos	resultado
Realizar una búsqueda de propiedades mediante parámetros de perfil.	El sistema debe devolver una lista de propiedades en base al criterio seleccionado.	1. Ingresar perfil con los parámetros correspondientes.	El sistema debe retornar las propiedades que cumplan con las características seleccionadas. Deberán estar ordenadas en base a coincidencias de servicios. Ver campo <i>ServicesCount</i> .
		2. Ingresar perfil, y a su vez ciudad y barrio.	El sistema debe retornar las propiedades que cumplan con las características, ubicadas en la ciudad y barrio seleccionados. Deberán estar ordenadas en base a coincidencias de servicios. Ver campo <i>ServicesCount</i> .
		3. repetir pasos 1 y 2 para todos los perfiles.	

Tabla 41 Caso de prueba búsqueda por perfil.

## Registro de usuario

objetivo	esperado	pasos	resultado
Registrar un usuario al sistema.	El sistema debe poder registrar a un usuario dado con <i>email</i> contraseña.	1. ingresar los campos solicitados, mail, nombre, apellido, documento, contraseña y repetir contraseña.	El sistema debe registrar el usuario y devolver el mensaje correspondiente.
		2. Ingresar un <i>email</i> no válido.	El sistema debe informar que el <i>email</i> no es válido y solicitar nuevamente los valores.
		3. ingresar un usuario ya registrado.	El sistema debe informar el mensaje correspondiente y solicitar nuevamente los valores.

Tabla 42 Caso de prueba registro de usuario.

### Detalle de propiedad

objetivo	esperado	pasos	resultado
Dada una propiedad el sistema debe retornar los datos correspondientes.	El sistema deberá retornar la información de una propiedad, los datos correspondientes son: Información, Detalle, Fotos, Operaciones.	1. Dada una propiedad existente y activa, solicitar el detalle de propiedad.	El sistema deberá retornar la propiedad con los datos correspondientes, ver documentación de API.
		2. Solicitar el detalle de una propiedad existente pero no activa en el sistema	El sistema deberá retornar un objeto vacío.
		3. Dada una propiedad no existente en el sistema, solicitar detalle de propiedad.	El sistema deberá retornar un objeto vacío.

Tabla 43 Caso de prueba Detalle de propiedad.

### ABM de propiedades

objetivo	esperado	pasos	Resultado
El sistema deberá permitir dar alta, modificar y eliminar una propiedad.	El sistema deberá guardar la información ingresada de una propiedad, así como también eliminar una propiedad dada.	1. Ingresar los datos de una propiedad nueva.(ver documentación API)	El sistema deberá guardar los datos ingresados en la BD y retornar un mensaje de confirmación.
		2. Ingresar los datos de una propiedad ya existente: en este caso se deberá ingresar el id externo correspondiente.	El sistema deberá guardar los datos ingresados en la BD y retornar un mensaje de confirmación.
		3. dada una propiedad, eliminar dicha propiedad.	El sistema deberá eliminar la propiedad y devolver el mensaje correspondiente. EN caso de que no exista la propiedad devolver un mensaje de error.

Tabla 44 Caso de prueba ABM de propiedad.

## Solicitud de alta de propiedad

objetivo	esperado	pasos	Resultado
El sistema deberá poder permitirle a los usuarios enviar una solicitud para ingresar un inmueble en la cartera de la Inmobiliaria.	El sistema deberá guardar la información ingresada de una propiedad, e informar a los usuarios administradores.	<p>1. Ingresar los datos correspondientes de la solicitud ( ver documentación API)</p> <p>2. Ingresar los datos de una propiedad ya existente: en este caso se deberá ingresar el id externo correspondiente.</p>	<p>El sistema deberá guardar los datos ingresados en la BD y retornar un mensaje de confirmación. A su vez se deberá enviar un <i>email</i> a los usuarios administradores con dicha solicitud.</p>
			El sistema deberá guardar los datos ingresados en la BD y retornar un mensaje de confirmación.

Tabla 45 Caso de prueba solicitud de alta de propiedad.

## Listado de solicitudes

objetivo	esperado	pasos	resultado
El sistema deberá poder desplegar las solicitudes recibidas.	El sistema deberá proporcionarle al usuario administrador un listado con la información de las solicitudes activas.	<p>1. Ingresar como usuario administrador y llamar al listado de solicitudes de alta</p> <p>2. cambiar el estado de una solicitud, y llamar al listado nuevamente</p>	<p>El sistema deberá retornar el listado de solicitudes con estado no aprobado.</p>
			El sistema deberá retornar el listado de solicitudes con estado no aprobado.

Tabla 46 Caso de prueba listado de solicitudes.

## Listado de propiedades favoritas

objetivo	esperado	pasos	resultado
El sistema brindará la posibilidad a los usuarios autenticados de poder marcar una propiedad como favorita.	El sistema deberá agregar dicha propiedad a la lista de propiedades favoritas del usuario, para luego notificarlo mediante <i>email</i> sobre los cambios de sus propiedades favoritas.	1. Ingresar como usuario, y agregar una propiedad como favorita.	El sistema deberá guardar la propiedad a la lista de favoritas.
		2. Modificar el precio, operación, gastos comunes de dicha propiedad.	El sistema deberá enviar una notificación a la casilla del usuario notificando los cambios de su propiedad favorita
		3. Eliminar la propiedad de la lista de favoritas.	El sistema deberá eliminar dicha propiedad de la lista de favoritas y dejar de notificar al usuario sobre esa propiedad.

Tabla 47 Caso de prueba Listado de propiedades favoritas.

## Agregar propiedad favorita

objetivo	esperado	pasos	resultado
El sistema brindará la posibilidad a los usuarios autenticados de poder marcar una propiedad como favorita.	El sistema deberá agregar dicha propiedad a la lista de propiedades favoritas del usuario, para luego notificarlo mediante <i>email</i> sobre los cambios de sus propiedades favoritas.	1. Iniciar sesión como usuario, y agregar una propiedad como favorita.	El sistema deberá guardar la propiedad a la lista de favoritas.
		2. Modificar el precio, operación, gastos comunes de dicha propiedad.	El sistema deberá enviar una notificación a la casilla del usuario notificando los cambios de su propiedad favorita
		3. Eliminar la propiedad de la lista de favoritas.	El sistema deberá eliminar dicha propiedad de la lista de favoritas y dejar de notificar al usuario sobre esa propiedad.

Tabla 48 Caso de prueba agregar propiedad favorita.

## Notificaciones

objetivo	esperado	pasos	resultado
El sistema deberá poder enviar notificaciones a los usuarios según sus intereses marcados (propiedades favoritas, comentarios, sugerencias, pedido de propiedades).	El sistema deberá notificar al usuario vía correo electrónico cuando se genere una notificación.	1. Generar una notificación de propiedad favorita. Realizar cambios de precios y operaciones.	El sistema deberá enviar una notificación a todos los usuarios que tengan agregada la propiedad como favorita, se deberá notificar el cambio de precio y operación.
		2. Generar una notificación de pedido de propiedad. Agregar una propiedad con características iguales a las de un pedido de propiedad.	El sistema deberá enviar una notificación a la casilla del usuario notificando el ingreso de una nueva propiedad con características similares a las de su pedido.
		3. Generar una notificación de comentario. Comentar sobre una propiedad.	El sistema deberá notificar a aquellos usuarios que hayan realizado comentarios sobre esa propiedad.
		4. Generar una notificación de sugerencia de propiedad. Ingresar nuevas búsquedas de un usuario. Generar búsquedas de otros usuarios con similitudes.	El sistema deberá notificar al usuario con sugerencias en base a las búsquedas de otros usuarios.

Tabla 49 Caso de prueba Notificaciones.

## Listado de notificaciones

objetivo	esperado	pasos	resultado
El sistema deberá brindar la posibilidad de poder listar las notificaciones de cada usuario.	El sistema deberá listar las notificaciones de un usuario dado un tipo de notificación: comentario, propiedad favorita, pedido de propiedad, sugerencias.	1. Iniciar sesión y solicitar el listado de notificaciones para el tipo comentario.  2. Iniciar sesión y solicitar el listado de notificaciones para el tipo propiedad favorita.  3. Iniciar sesión y solicitar el listado de notificaciones para el tipo pedido de propiedad.  4. Iniciar sesión y solicitar el listado de notificaciones para el tipo sugerencias.	El sistema deberá devolver una lista de notificaciones solamente del tipo comentario.  El sistema deberá devolver una lista de notificaciones solamente del tipo propiedad favorita.  El sistema deberá devolver una lista de notificaciones solamente del tipo pedido de propiedad.  El sistema deberá devolver una lista de notificaciones solamente del tipo sugerencias.

Tabla 50 Caso de prueba Listado de notificaciones.

## Comentarios

objetivo	esperado	pasos	resultado
El sistema deberá poder permitirle al usuario el ingreso, modificación y eliminación de comentarios en las distintas propiedades.	El sistema deberá permitir la administración de comentarios dada una propiedad y un usuario.	1. Iniciar sesión y realizar un comentario sobre una propiedad.  2. Iniciar sesión y modificar un comentario.  3. Iniciar sesión y eliminar un comentario.  4. Iniciar sesión y solicitar el listado de comentarios para una propiedad.	El sistema deberá ingresar el comentario.  El sistema deberá guardar los cambios en el comentario.  El sistema deberá eliminar el comentario.  El sistema deberá devolver una lista de comentarios para esa propiedad.

Tabla 51 Caso de prueba Comentarios.

## Recomendaciones de propiedades

objetivo	esperado	pasos	resultado
El sistema deberá poder realizar recomendaciones de propiedades a los usuarios.	El sistema recomendar propiedades en base a las búsquedas de otros usuarios.	1. Ingresar búsquedas para un usuario.	El sistema deberá guardar las búsquedas para ese usuario.
		2. Ingresar búsquedas similares con otros usuarios.	El sistema deberá guardar las búsquedas para ese usuario.
		3. Ejecutar aplicación para generar recomendaciones.	El sistema deberá generar las recomendaciones y enviar por correo electrónico una notificación.

Tabla 52 Caso de prueba Recomendaciones de propiedades.

## Pedido de propiedad

objetivo	esperado	pasos	resultado
El sistema deberá brindar la posibilidad a los usuarios autenticados de realizar el pedido de una propiedad.	Se deberá poder ingresar pedidos de propiedad concretos, los cuales serán guardados en el sistema, luego al ingresar una propiedad si coincide con las características se deberá notificar al usuario. También se deberá enviar un correo electrónico al administrador con el pedido.	1. Iniciar sesión e ingresar un pedido de propiedad.	El sistema ingresar el pedido y enviar un <i>email</i> al usuario administrador con el pedido.
		2. Ingresar una propiedad al sistema con las características del pedido.	El sistema deberá generar una notificación con la información de la propiedad ingresada y enviarla al usuario que realizó el pedido
		3. Iniciar sesión y solicitar el listado de notificaciones para el tipo pedido de propiedad.	El sistema deberá devolver una lista de notificaciones solamente del tipo pedido de propiedad.

Tabla 53 Caso de prueba Pedido de propiedad.

## Agenda visita

objetivo	esperado	pasos	resultado
El sistema deberá brindar la posibilidad de agendar una visita a una propiedad.	Dada una propiedad se deberá poder agendar una visita ingresando fecha y hora deseada y un mensaje.	1. Iniciar sesión e ingresar fecha, hora y mensaje para agendar visita a una propiedad.	El sistema deberá ingresar el pedido de agenda y notificar al usuario administrador con los datos de la agenda y del usuario.
		2. No ingresar fecha y hora y solicitar la agenda.	El sistema deberá informar que fecha y hora son requeridos.

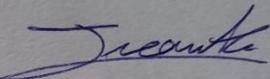
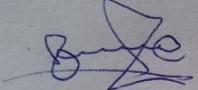
Tabla 54 Caso de prueba Agenda de visita.

## 7.9 Anexo I: Actas de reuniones

### 7.9.1 Reuniones con cliente

<b>ACTA N° 1</b>	Monografía	
	Proyecto	x
	Tesis	
	Artículo técnico	
	Pasantía	

<b>Carrera</b>	<b>Analista Programador / Analista en Tecnologías de la Información</b>
----------------	---

<b>Nº de Estudiante</b>	<b>Nombre</b>	
<b>197767</b>	<b>Juan González</b>	
<b>179870</b>	<b>Bruno Dutra</b>	

**Cliente: Alejandro González**

<b>Reunión</b>	<b>Fecha</b>	<b>Firma</b>
	05/04/2016	

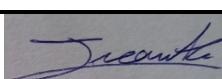
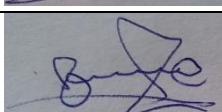
OBSERVACIONES

Primera reunión con el cliente. Comentarios acerca del proyecto de lo que él pretende para su empresa.

Intercambio de ideas y queda agendada una próxima reunión para luego del primer encuentro con el tutor.

<b>ACTA N° 2</b>	Monografía	
	Proyecto	x
	Tesis	
	Artículo técnico	
	Pasantía	

<b>Carrera</b>	<b>Analista Programador / Analista en Tecnologías de la Información</b>
----------------	---

<b>Nº de Estudiante</b>	<b>Nombre</b>	
<b>197767</b>	<b>Juan González</b>	
<b>179870</b>	<b>Bruno Dutra</b>	

**Cliente: Alejandro González**

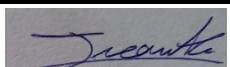
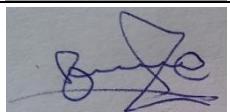
<b>Reunión</b>	<b>Fecha</b>	<b>Firma</b>
	15/04/2016	

<b>OBSERVACIONES</b>
Comentarios acerca de las recomendaciones del tutor e intercambio de ideas sobre lo que terminará siendo el producto.

	Monografía	
	Proyecto	x
	Tesis	
	Artículo técnico	
	Pasantía	

## ACTA N° 3

<b>Carrera</b>	<b>Analista Programador / Analista en Tecnologías de la Información</b>
----------------	---

<b>Nº de Estudiante</b>	<b>Nombre</b>	
<b>197767</b>	<b>Juan González</b>	
<b>179870</b>	<b>Bruno Dutra</b>	

**Cliente: Alejandro González**

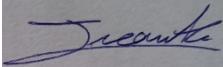
<b>Reunión</b>	<b>Fecha</b>	<b>Firma</b>
	20/05/2016	

OBSERVACIONES

Comentarios acerca de lo que viene siendo el desarrollo. Consultas de parte del equipo para definir lo que serán las búsquedas por perfiles donde se incluirán parámetros en cada uno de ellos.

<b>ACTA N° 4</b>	Monografía	
	Proyecto	x
	Tesis	
	Artículo técnico	
	Pasantía	

<b>Carrera</b>	<b>Analista Programador / Analista en Tecnologías de la Información</b>
----------------	---

<b>Nº de Estudiante</b>	<b>Nombre</b>	
<b>197767</b>	<b>Juan González</b>	
<b>179870</b>	<b>Bruno Dutra</b>	

**Cliente: Alejandro González**

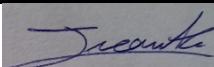
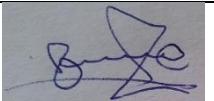
<b>Reunión</b>	<b>Fecha</b>	<b>Firma</b>
	20/06/2016	

<b>OBSERVACIONES</b>
Comentarios generales del proyecto y una muestra sobre cómo se verán las propiedades en el mapa con los servicios cercanos. Intercambio de ideas sobre mejoras que pueden realizarse.

	Monografía	
	Proyecto	x
	Tesis	
	Artículo técnico	
	Pasantía	

## ACTA N° 5

<b>Carrera</b>	<b>Analista Programador / Analista en Tecnologías de la Información</b>
----------------	---

<b>Nº de Estudiante</b>	<b>Nombre</b>	
<b>197767</b>	<b>Juan González</b>	
<b>179870</b>	<b>Bruno Dutra</b>	

**Cliente: Alejandro González**

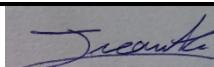
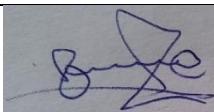
<b>Reunión</b>	<b>Fecha</b>	<b>Firma</b>
	03/08/2016	

OBSERVACIONES
Muestra de lo que es el producto al momento. Comentarios generales y nuevas ideas que se podrían llegar a realizar en caso de que dé el tiempo.

	Monografía		
	Proyecto		X
	Tesis		
	Artículo técnico		
	Pasantía		

## ACTA N° 6

Carrera	Analista Programador / Analista en Tecnologías de la Información
---------	--

Nº de Estudiante	Nombre	
197767	Juan González	
179870	Bruno Dutra	

**Cliente: Alejandro González**

Reunión	Fecha	Firma
	20/09/2016	

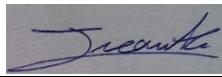
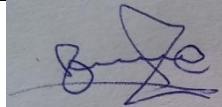
OBSERVACIONES

Muestra del producto final y comentarios al respecto.

7.9.2 Reuniones con tutor

<b>ACTA N° 1</b>	Monografía	
	Proyecto	x
	Tesis	
	Artículo técnico	
	Pasantía	

<b>Carrera</b>	<b>Analista Programador / Analista en Tecnologías de la Información</b>
----------------	---

<b>Nº de Estudiante</b>	<b>Nombre</b>	
<b>197767</b>	<b>Juan González</b>	
<b>179870</b>	<b>Bruno Dutra</b>	

**Tutor: Fernando Martínez**

<b>Reunión</b>	<b>Fecha</b>	<b>Firma</b>
	11/04/2016	

OBSERVACIONES

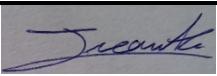
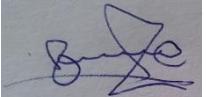
1- Presentación del grupo y del tutor.

2- Aspectos generales del proyecto.

3- Modalidad de trabajo.

<b>ACTA N° 2</b>	Monografía	
	Proyecto	x
	Tesis	
	Artículo técnico	
	Pasantía	

<b>Carrera</b>	<b>Analista Programador / Analista en Tecnologías de la Información</b>
----------------	---

<b>Nº de Estudiante</b>	<b>Nombre</b>	
<b>197767</b>	<b>Juan González</b>	
<b>179870</b>	<b>Bruno Dutra</b>	

**Tutor: Fernando Martínez**

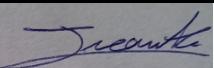
<b>Reunión</b>	<b>Fecha</b>	<b>Firma</b>
	18/04/2016	

OBSERVACIONES

- 1- Comentarios primera semana de trabajo.
- 2- Avance de justificación de proyecto.
- 3- Tarea siguiente: generar primera versión de anteproyecto.

<b>ACTA N° 3</b>	Monografía	
	Proyecto	x
	Tesis	
	Artículo técnico	
	Pasantía	

<b>Carrera</b>	<b>Analista Programador / Analista en Tecnologías de la Información</b>
----------------	---

<b>Nº de Estudiante</b>	<b>Nombre</b>	
<b>197767</b>	<b>Juan González</b>	
<b>179870</b>	<b>Bruno Dutra</b>	

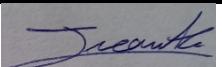
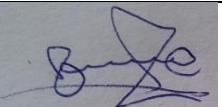
**Tutor: Fernando Martínez**

<b>Reunión</b>	<b>Fecha</b>	<b>Firma</b>
	25/04/2016	

<b>OBSERVACIONES</b>
1- Temas generales de anteproyecto

<b>ACTA N° 4</b>	Monografía	
	Proyecto	x
	Tesis	
	Artículo técnico	
	Pasantía	

<b>Carrera</b>	<b>Analista Programador / Analista en Tecnologías de la Información</b>
----------------	---

<b>Nº de Estudiante</b>	<b>Nombre</b>	
<b>197767</b>	<b>Juan González</b>	
<b>179870</b>	<b>Bruno Dutra</b>	

**Tutor: Fernando Martínez**

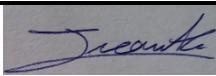
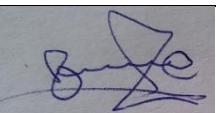
<b>Reunión</b>	<b>Fecha</b>	<b>Firma</b>
	02/05/2016	

<b>OBSERVACIONES</b>
----------------------

1- Temas generales de anteproyecto y seguimiento del mismo.

<b>ACTA N° 5</b>	Monografía	
	Proyecto	x
	Tesis	
	Artículo técnico	
	Pasantía	

<b>Carrera</b>	<b>Analista Programador / Analista en Tecnologías de la Información</b>
----------------	---

<b>Nº de Estudiante</b>	<b>Nombre</b>	
<b>197767</b>	<b>Juan González</b>	
<b>179870</b>	<b>Bruno Dutra</b>	

**Tutor: Fernando Martínez**

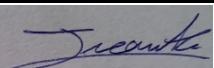
<b>Reunión</b>	<b>Fecha</b>	<b>Firma</b>
	23/05/2016	

<b>OBSERVACIONES</b>
----------------------

- 1- Comentarios de lo hecho desde la última reunión.
- 2- Muestra de los servicios invocados.
- 3- Comentarios de cómo seguir, el tutor nos aconseja acelerar en lo que se viene realizando.

<b>ACTA N° 6</b>	Monografía	
	Proyecto	x
	Tesis	
	Artículo técnico	
	Pasantía	

<b>Carrera</b>	<b>Analista Programador / Analista en Tecnologías de la Información</b>
----------------	---

<b>Nº de Estudiante</b>	<b>Nombre</b>	
<b>197767</b>	<b>Juan González</b>	
<b>179870</b>	<b>Bruno Dutra</b>	

**Tutor: Fernando Martínez**

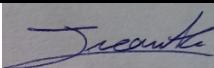
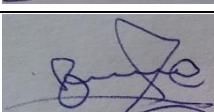
<b>Reunión</b>	<b>Fecha</b>	<b>Firma</b>
	30/05/2016	

<b>OBSERVACIONES</b>
----------------------

- 1- Muestra de avance hasta el momento.
- 2- Comentarios sobre los casos de uso.
- 3- Entrega de primera versión de los mismos.
- 4- Comentarios acerca del desarrollo del *sprint* 3 que se entregará la semana siguiente.

<b>ACTA N° 7</b>	Monografía	
	Proyecto	x
	Tesis	
	Artículo técnico	
	Pasantía	

<b>Carrera</b>	<b>Analista Programador / Analista en Tecnologías de la Información</b>
----------------	---

<b>Nº de Estudiante</b>	<b>Nombre</b>	
<b>197767</b>	<b>Juan González</b>	
<b>179870</b>	<b>Bruno Dutra</b>	

**Tutor: Fernando Martínez**

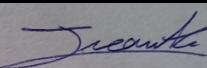
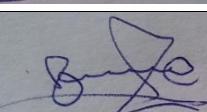
<b>Reunión</b>	<b>Fecha</b>	<b>Firma</b>
	02/06/2016	

<b>OBSERVACIONES</b>
----------------------

- 1 – Corrección de caso de uso. Comentarios para realizar mejoras.
- 2 - Ingreso del cumplimiento de horas como riesgo.
- 3 – Comentarios acerca de los riesgos en los informes de cada *sprint*. Detalle de los mismos, indicando si aparecieron o no, y en caso de que aparezcan como lo contrarrestamos.
- 4 – Ingreso de las reuniones con el cliente.
- 5 – Comentarios acerca de cómo quedaría el siguiente *sprint* producto de los tiempos que nos llevó este.

<b>ACTA N° 8</b>	Monografía	
	Proyecto	x
	Tesis	
	Artículo técnico	
	Pasantía	

<b>Carrera</b>	<b>Analista Programador / Analista en Tecnologías de la Información</b>
----------------	---

<b>Nº de Estudiante</b>	<b>Nombre</b>	
<b>197767</b>	<b>Juan González</b>	
<b>179870</b>	<b>Bruno Dutra</b>	

**Tutor: Fernando Martínez**

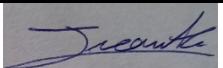
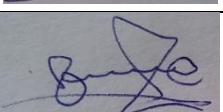
<b>Reunión</b>	<b>Fecha</b>	<b>Firma</b>
	09/06/2016	

OBSERVACIONES

- 1 – Comentarios acerca de la documentación, unir la misma e incluir los anexo (con letras).
- 2 – Detalles para la entrega del 16/06/2016. Subir a aulas el *sprint* 3 y 4 y mandar al tutor.
- 3 – Comentarios acerca del *testing*. Documentar los que hicimos así como también los estándares que mencionamos en el anteproyecto.
- 4 – Ingresar en los informes las buenas prácticas de C# que utilizamos, indicando cuáles son.
- 5 – Agregar MER al informe.
- 6 – Ingreso de nuevas ideas. Quedan como trabajos a futuro o se agregan al proyecto si da el tiempo.

<b>ACTA N° 9</b>	Monografía	
	Proyecto	x
	Tesis	
	Artículo técnico	
	Pasantía	

<b>Carrera</b>	<b>Analista Programador / Analista en Tecnologías de la Información</b>
----------------	---

<b>Nº de Estudiante</b>	<b>Nombre</b>	
<b>197767</b>	<b>Juan González</b>	
<b>179870</b>	<b>Bruno Dutra</b>	

**Tutor: Fernando Martínez**

<b>Reunión</b>	<b>Fecha</b>	<b>Firma</b>
	23/06/2016	

<b>OBSERVACIONES</b>
----------------------

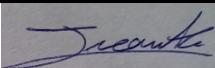
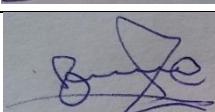
1 – Consulta del equipo acerca de las mejoras a realizar en el mapa, intercambio de ideas para elegir el momento indicando ya que el *sprint* terminó. Se definió que las mejoras se realizarán en el transcurso del proyecto, pudiendo hacer un proyecto aparte las mismas. Dejar documentadas las mejoras.

2 – Comentarios del proyecto en general acerca de en qué funcionalidades profundizar.

3 – Ideas acerca de filtrar búsquedas por los valores paramétricos (por ejemplo, el radio que se le asigna a los servicios a mostrar)

<b>ACTA N° 10</b>	Monografía	
	Proyecto	x
	Tesis	
	Artículo técnico	
	Pasantía	

<b>Carrera</b>	<b>Analista Programador / Analista en Tecnologías de la Información</b>
----------------	---

<b>Nº de Estudiante</b>	<b>Nombre</b>	
<b>197767</b>	<b>Juan González</b>	
<b>179870</b>	<b>Bruno Dutra</b>	

**Tutor: Fernando Martínez**

<b>Reunión</b>	<b>Fecha</b>	<b>Firma</b>
	07/07/2016	

<b>OBSERVACIONES</b>
----------------------

- 1 – Comentarios acerca de las entregas, donde se coordina entregar la documentación del *sprint* 5 junto con la del 6.
- 2 – Comentarios respecto a acortar los tiempos del desarrollo del *sprint* 7 para poder profundizar en mejoras de los otros.
- 3 – Comentarios acerca de la integración del servicio con la página *web*. Se habló acerca de la vista a realizar que podría ser invocada por AJAX, así como también validar la vista.
- 4 – Comentarios acerca de los mails a enviar como notificación. Se vieron diferentes opciones como *mailing* de Amazon o Google.
- 5 – Recomendación del tutor de tener un buen juego de datos de prueba para la parte de recomendaciones de propiedades.

<b>ACTA N° 11</b>	Monografía	
	Proyecto	x
	Tesis	
	Artículo técnico	
	Pasantía	

<b>Carrera</b>	<b>Analista Programador / Analista en Tecnologías de la Información</b>
----------------	---

<b>Nº de Estudiante</b>	<b>Nombre</b>
<b>197767</b>	<b>Juan González</b>
<b>179870</b>	<b>Bruno Dutra</b>

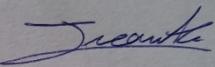
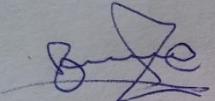
**Tutor: Fernando Martínez**

<b>Reunión</b>	<b>Fecha</b>	<b>Firma</b>
	21/07/2016	

OBSERVACIONES
<p>1 – Consultas al tutor sobre Azure y su WebJobs.</p> <p>2 – Comentarios acerca de realizar un anexo para WebJobs.</p> <p>3 – Realizamos pruebas de mandar notificaciones vía mail.</p> <p>4 – Comentarios en general del proyecto y de la entrega a realizar el 28/07/2016.</p>

<b>ACTA N° 12</b>	Monografía	
	Proyecto	x
	Tesis	
	Artículo técnico	
	Pasantía	

<b>Carrera</b>	<b>Analista Programador / Analista en Tecnologías de la Información</b>
----------------	---

<b>Nº de Estudiante</b>	<b>Nombre</b>	
<b>197767</b>	<b>Juan González</b>	
<b>179870</b>	<b>Bruno Dutra</b>	

**Tutor: Fernando Martínez**

<b>Reunión</b>	<b>Fecha</b>	<b>Firma</b>
	04/08/2016	

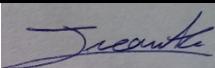
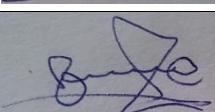
<b>OBSERVACIONES</b>
----------------------

Comentarios generales del proyecto así como también, acerca del cierre y correcciones de *Sprints*.

Pedido del tutor de ir unificando toda la documentación para así lograr una entrega preliminar a lo que será la definitiva.

<b>ACTA N° 13</b>	Monografía	
	Proyecto	x
	Tesis	
	Artículo técnico	
	Pasantía	

<b>Carrera</b>	<b>Analista Programador / Analista en Tecnologías de la Información</b>
----------------	---

<b>Nº de Estudiante</b>	<b>Nombre</b>	
<b>197767</b>	<b>Juan González</b>	
<b>179870</b>	<b>Bruno Dutra</b>	

**Tutor: Fernando Martínez**

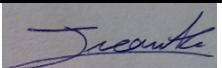
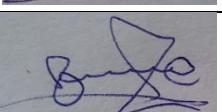
<b>Reunión</b>	<b>Fecha</b>	<b>Firma</b>
	23/08/2016	

<b>OBSERVACIONES</b>
----------------------

- 1- Nuevas ideas, por ejemplo mapa con propiedades en la home, comparación de propiedades, en los resultados de búsqueda que aparezca el mapa con las propiedades e inclusión de *Google Analytics*.
- 2- Realizar explicación sobre búsquedas por perfil.
- 3- Idea de para el olvido de contraseña, enviar contraseña nueva al usuario.
- 4- Agenda de visita al administrador, de *Google calendar*.
- 5- Ir dejando encaminado el armado de documentación para entregar una versión preliminar.
- 6- Comentarios acerca de cierre de proyecto.

<b>ACTA N° 14</b>	Monografía	
	Proyecto	x
	Tesis	
	Artículo técnico	
	Pasantía	

<b>Carrera</b>	<b>Analista Programador / Analista en Tecnologías de la Información</b>
----------------	---

<b>Nº de Estudiante</b>	<b>Nombre</b>	
<b>197767</b>	<b>Juan González</b>	
<b>179870</b>	<b>Bruno Dutra</b>	

**Tutor: Fernando Martínez**

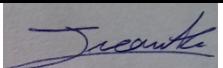
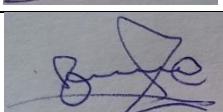
<b>Reunión</b>	<b>Fecha</b>	<b>Firma</b>
	01/09/2016	

<b>OBSERVACIONES</b>
----------------------

- 1- Comentarios acerca del fin de proyecto: conclusiones, trabajos a futuro, lecciones aprendidas, grafica general del proyecto (uniendo las de los *sprints*), horas de *testing* vs horas de implementación.
- 2- Creación de manual de usuario y *deploy*.
- 3- Tener evidencias de pruebas.
- 4- Anexos, los mismos van uno por hoja.
- 5- Agregar bibliografía, documentarla antes de los anexos.
- 6- Evidencia de *Google Analytics*.
- 7- Carta cliente

<b>ACTA N° 15</b>	Monografía	
	Proyecto	x
	Tesis	
	Artículo técnico	
	Pasantía	

<b>Carrera</b>	<b>Analista Programador / Analista en Tecnologías de la Información</b>
----------------	---

<b>Nº de Estudiante</b>	<b>Nombre</b>	
<b>197767</b>	<b>Juan González</b>	
<b>179870</b>	<b>Bruno Dutra</b>	

**Tutor: Fernando Martínez**

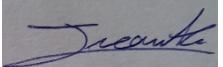
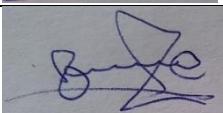
<b>Reunión</b>	<b>Fecha</b>	<b>Firma</b>
	15/09/2016	

<b>OBSERVACIONES</b>
----------------------

- 1- Sugerencia del tutor de agregar la funcionalidad de que cada usuario pueda obtener el listado de sus propiedades favoritas.
- 2- Sugerencia del tutor para que el administrador pueda obtener un listado de las visitas agendadas. A su vez, agregar validaciones en el ingreso de los datos para la visita. Por ejemplo que no se puedan escoger ni sábados ni domingos.
- 3- Para la defensa, se puede realizar una ppt básica donde nosotros desarrollemos todas las ideas.
- 4- Cerrar una primera versión de proyecto.
- 5- Los *scripts* de base de datos incluirlas en un archivo.

<b>ACTA N° 16</b>	Monografía	
	Proyecto	x
	Tesis	
	Artículo técnico	
	Pasantía	

<b>Carrera</b>	<b>Analista Programador / Analista en Tecnologías de la Información</b>
----------------	---

<b>Nº de Estudiante</b>	<b>Nombre</b>	
<b>197767</b>	<b>Juan González</b>	
<b>179870</b>	<b>Bruno Dutra</b>	

**Tutor: Fernando Martínez**

<b>Reunión</b>	<b>Fecha</b>	<b>Firma</b>
	22/09/2016	

**OBSERVACIONES**

- 1- Consulta del equipo y explicación del tutor sobre las diferencias que existen entre solución implementada y el manual de usuario.
- 2- Sugerencia del tutor de incluir un archivo con datos de prueba para el corrector.
- 3- Comentarios generales acerca de la entrega final.

## 7.10 Anexo J: Explicación algoritmo de sugerencia de propiedades

“Aprende” cuáles son los parámetros de búsqueda de ese usuario según el historial de búsqueda. Para esto, se queda con aquellos parámetros que haya utilizado en la mayoría de las búsquedas.

Para los parámetros de propiedad: dormitorios, baños, ciudad, barrio y tipo de propiedad, se cuenta cuántas veces se incluyó el mismo valor y se queda con el mayor, luego evalúa si este estuvo en más del 50 % de las búsquedas.

Para los parámetros de tipo de operación: venta o alquiler cuenta las veces que se incluyó cada una y se queda con el mayor, en caso de ser iguales incluirá a ambas.

Luego de obtener los parámetros, obtiene las búsquedas similares de los demás usuarios del sistema y para cada una de las búsquedas coincidentes ejecuta la misma. Del resultado selecciona las propiedades distintas. Se crea una notificación para ese usuario con el resultado la cual será enviada vía *email*.

Las sugerencias de propiedades serán ejecutadas automáticamente una vez por semana. Esto se debe a que si el usuario no utiliza el sistema por cierta cantidad de tiempo pueden haber nuevas propiedades que le puedan interesar. A su vez las sugerencias permiten mostrarle al cliente otros resultados que quizás no haya obtenido.

## 7.11 Anexo K: Control de versionado

Se elige versionar el proyecto con la herramienta *TeamService* de Microsoft, se crea un servidor en el cual se versionan los proyectos actuales.

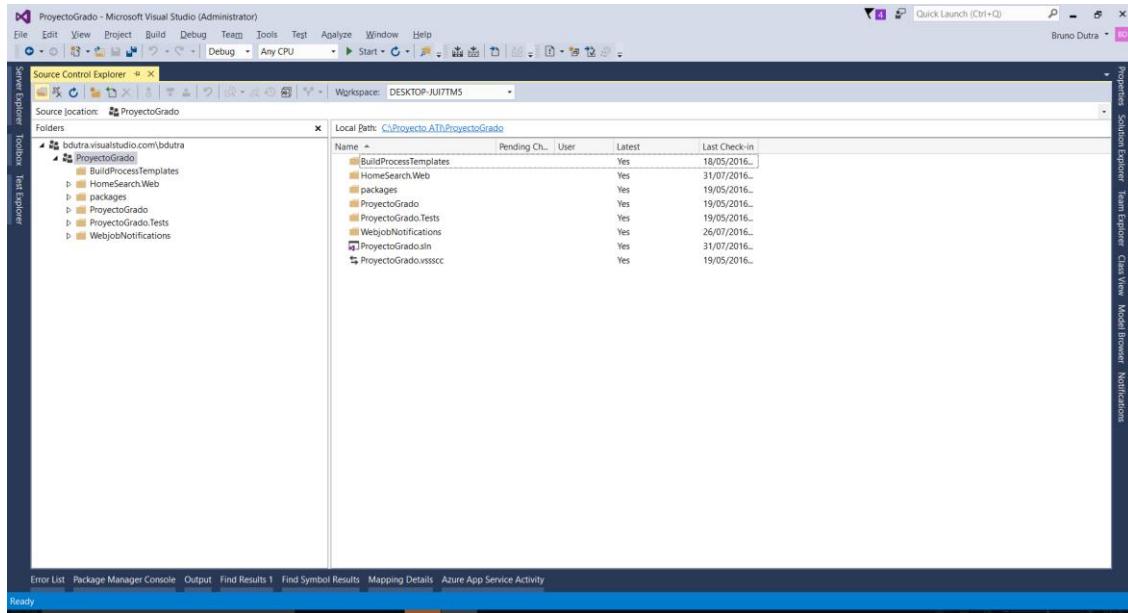


Ilustración 39 *Source control*.

### 7.11.1 Ejemplo de *Checkin*

Una vez que se realizan modificaciones de código, los cambios aparecerán en el menú de *Team Explorer*. Esto permite subir las diferencias y nuevos archivos para mantener una consistencia del código.

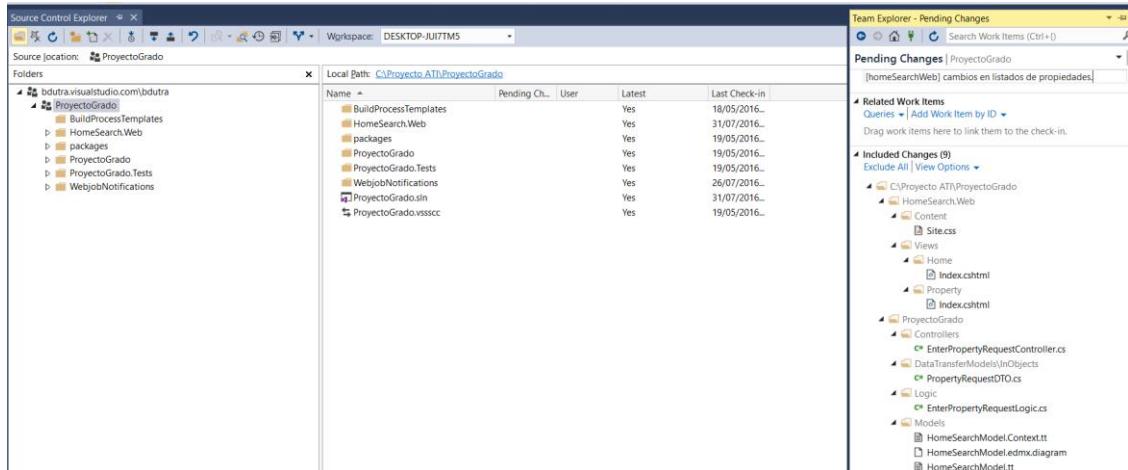


Ilustración 40 Ejemplo de *checkin*.

### 7.11.2 Comparación de versiones

Nos permite hacer un seguimiento del trabajo, pudiendo hacer comparaciones con versiones anteriores.

```
Server Explorer / Toolbox / Test Explorer / Properties / Solution Explorer / Team Explorer / Class View / Model Browser / Notifications
```

```
History - PropertyController.cs History - PropertyController.cs51
Server: $/ProyectoGrado/ProyectoGrado/Controllers/PropertyController.cs49 [ ] Misce... [ ] PostSearch(SearchDTO query)
130     return logic.DeleteFavoriteProperty(externalId, userEmail);
131 }
132 [Route("ImportProperties")]
133 public IHttpActionResult GetImportProperties()
134 {
135     try
136     {
137         using (var context = new HomeSearchDBEntities())
138         {
139             ImportPropertiesLogic.ImportProperties(context);
140         }
141         return Ok("ok");
142     }
143     catch (Exception ex)
144     {
145         return BadRequest("error");
146     }
147 }
148 }
149 }

150 [Authorize(Roles = "Authenticated")]
151 [Route("ScheduleVisit")]
152 public IHttpActionResult PostVisitSchedule(ScheduleVisitDTO visitModel)
153 {
154     try
155     {
156         var userEmail = User.Identity.Name;
157         using (var context = new HomeSearchDBEntities())
158         {
159             var logic = new PropertyLogic(context);
160             logic.CreateScheduleProperty(visitModel.ExternalId, userEmail, visitModel.Date, visitModel.Comer-
161         }
162     }
163     catch (Exception ex)
164     {
165         return BadRequest("error");
166     }
167 }
168 }

169 
```

#### Ilustración 41 Comparación de versiones.

## 7.12 Anexo L: Azure WebJob

Se utilizó la herramienta *Webjob* del portal *Azure* para ejecutar el algoritmo de sugerencia de propiedades y enviar notificaciones pendientes. Para esto se creó una aplicación de consola la cual se ejecutará una vez por semana según la configuración del trabajo. El tiempo de ejecución se puede cambiar dependiendo de las necesidades del cliente, se debe crear una expresión CRON la cual indica a cada cuánto se ejecutará.

La expresión CRON: 0 0 20 \*/7 \* 4 hará que se ejecute a las 20 horas, a cada 7 días solo los jueves.

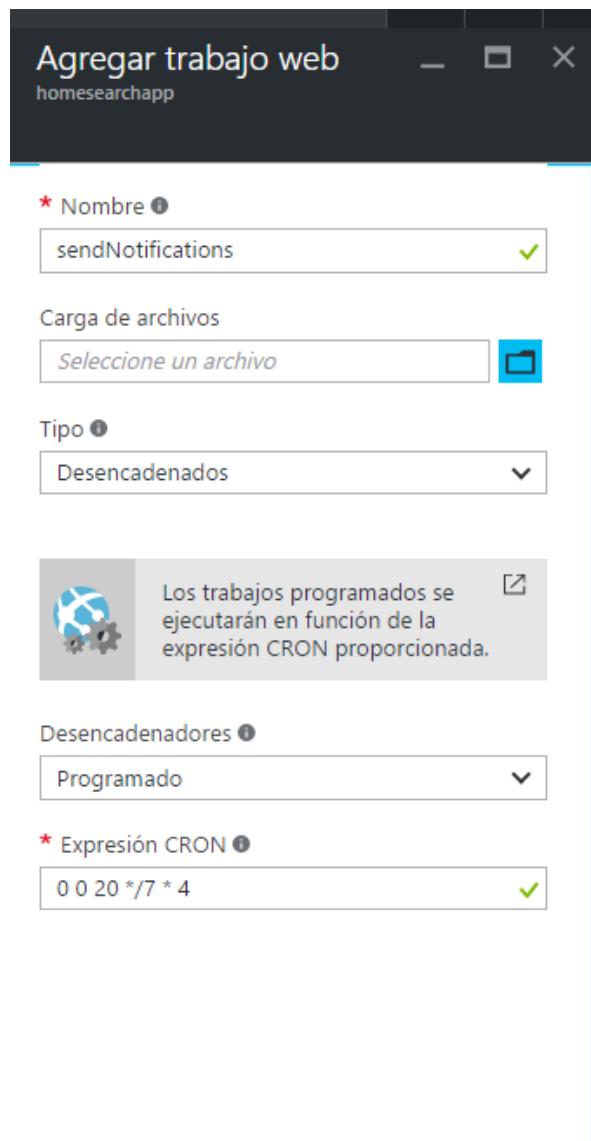


Ilustración 42 Agregar trabajo web.

## 7.13 Anexo M: Servicios de aplicaciones en Azure

Se crean 2 servicios de aplicaciones en Azure, el primero llamado *HomeSearchApp* aloja la *Api*, el segundo, llamado *HomeSearchWeb* aloja al sitio *web* que consume la *Api*.

*Api:* <http://homeseachapp.azurewebsites.net>  
*Sitio web:* <http://homeseachweb.azurewebsites.net>

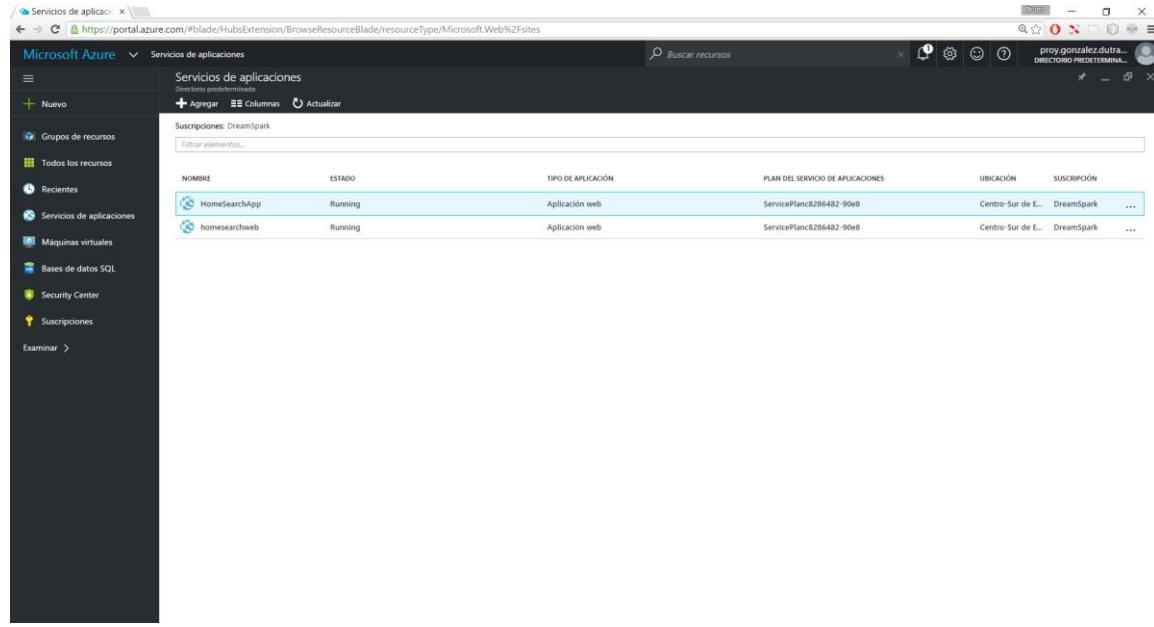


Ilustración 43 Servicios de aplicaciones.

### *HomeSearchApp*

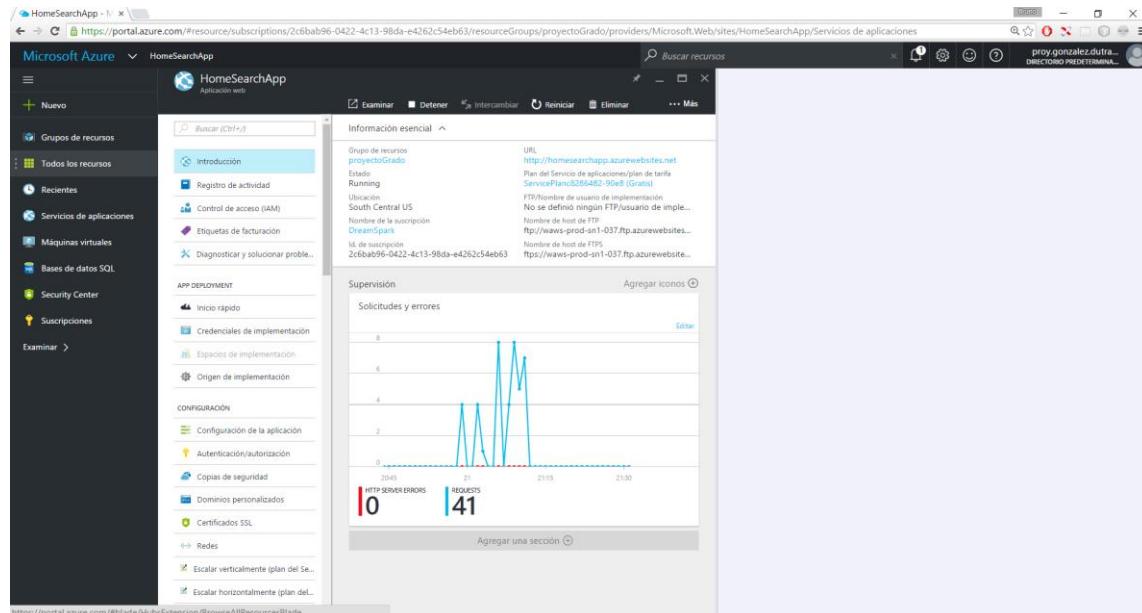


Ilustración 44 Servicio de aplicaciones *HomeSearchApp*.

## HomeSearchWeb

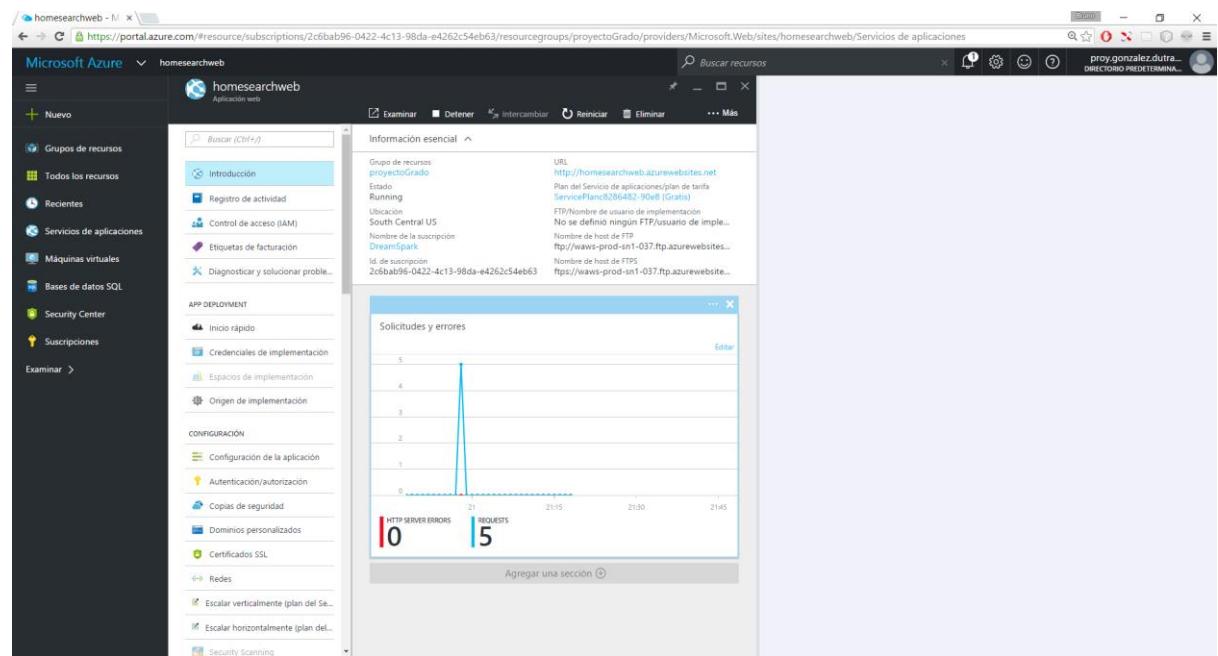


Ilustración 45 Servicio de aplicaciones *HomeSearchWeb*.

## 7.14 Anexo N: Pruebas unitarias

Se utilizó la herramienta *Nunit* de Microsoft para realizar casos de prueba unitarios. Se creó un proyecto de pruebas en el cual generamos para cada Lógica del sistema una clase de prueba. Durante el desarrollo se fueron creando casos para las principales funcionalidades. Finalmente no se pudo cubrir toda la lógica con casos de prueba. En una eventual puesta a producción es deseable tener todos los métodos cubiertos.

### 7.14.1 Proyecto de pruebas

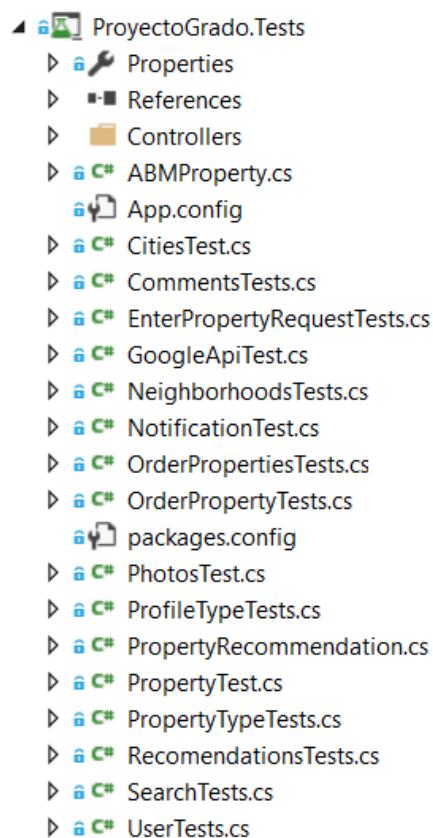
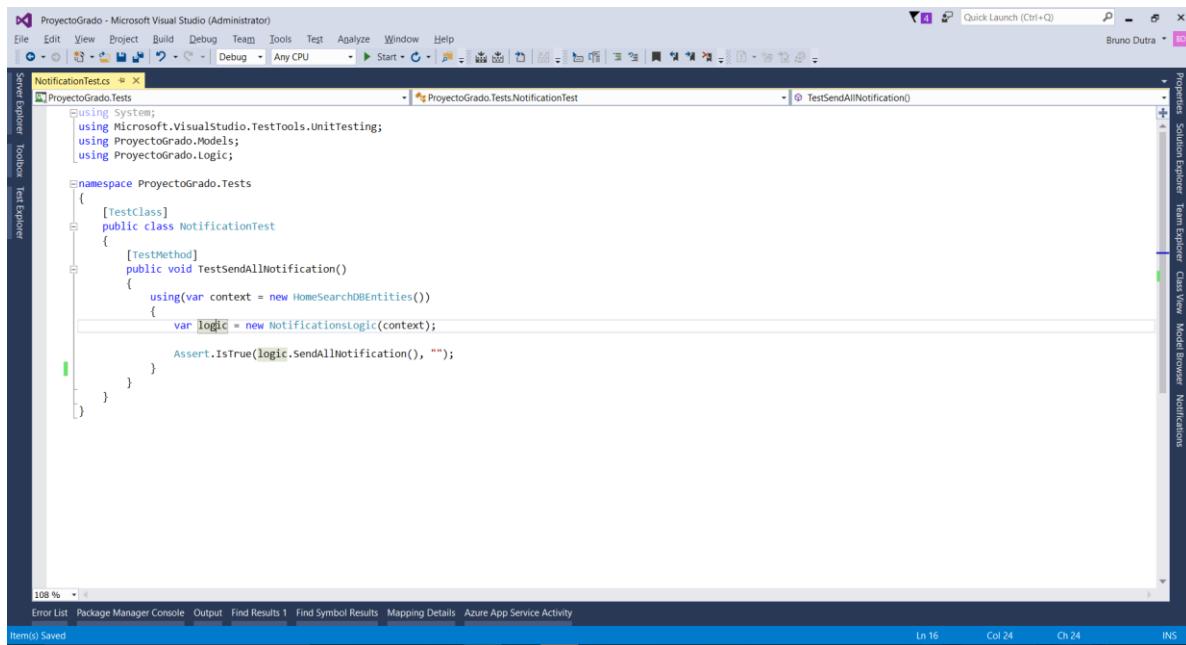


Ilustración 46 Proyectos de prueba.

## 7.14.2 Ejemplo de prueba



The screenshot shows the Microsoft Visual Studio interface with the title bar "ProyectoGrado - Microsoft Visual Studio (Administrator)". The main window displays a C# code file named "NotificationTest.cs" under the namespace "ProyectoGrado.Tests". The code contains a single test method, "TestSendAllNotification()", which uses a context and logic to assert the result of the "SendAllNotification()" method. The Visual Studio interface includes toolbars, a status bar at the bottom, and a vertical sidebar on the right.

```
ProjectoGrado - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Server Explorer Toolbox Test Explorer
NotificationTest.cs  x
ProyectoGrado.Tests
using System;
using Microsoft.VisualStudio.TestTools.UnitTesting;
using ProyectoGrado.Models;
using ProyectoGrado.Logic;

namespace ProyectoGrado.Tests
{
    [TestClass]
    public class NotificationTest
    {
        [TestMethod]
        public void TestSendAllNotification()
        {
            using(var context = new HomeSearchDBEntities())
            {
                var logic = new NotificationsLogic(context);
                Assert.IsTrue(logic.SendAllNotification(), "");
            }
        }
    }
}

108 %
Error List Package Manager Console Output Find Results 1 Find Symbol Results Mapping Details Azure App Service Activity
Item(s) Saved
Ln 16 Col 24 Ch 24 INS
```

Ilustración 47 Ejemplo de prueba.

## 7.14.3 Ejecución de casos de prueba

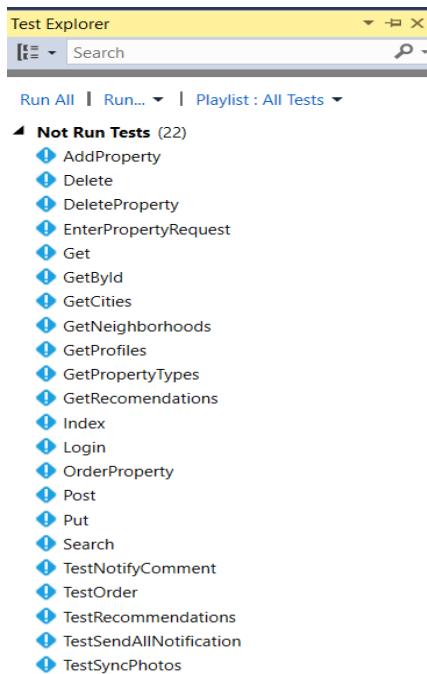


Ilustración 48 Ejecución de casos de prueba.

## 7.15 Anexo O: Manual de *Deployment*

A continuación se detalla el proceso que se lleva a cabo para publicar las aplicaciones en los servidores de *Azure*.

### 7.15.1 Configuración de servicio de aplicaciones

Un servicio de aplicaciones es el espacio que brinda la plataforma para alojar una aplicación. Para esto se debe configurar el tipo de aplicación, qué versión de *framework* utilizará. Las aplicaciones utilizarán el *framework* .Net versión 4.6. Los servicios a ser utilizados se detallan en el Anexo M [Servicios de aplicaciones en Azure](#).

### 7.15.2 Publicación de las aplicaciones

Desde *Visual Studio* se deberá configurar la conexión con los servidores, una vez configurado se pasa a publicar indicando a qué servicio se publicará.

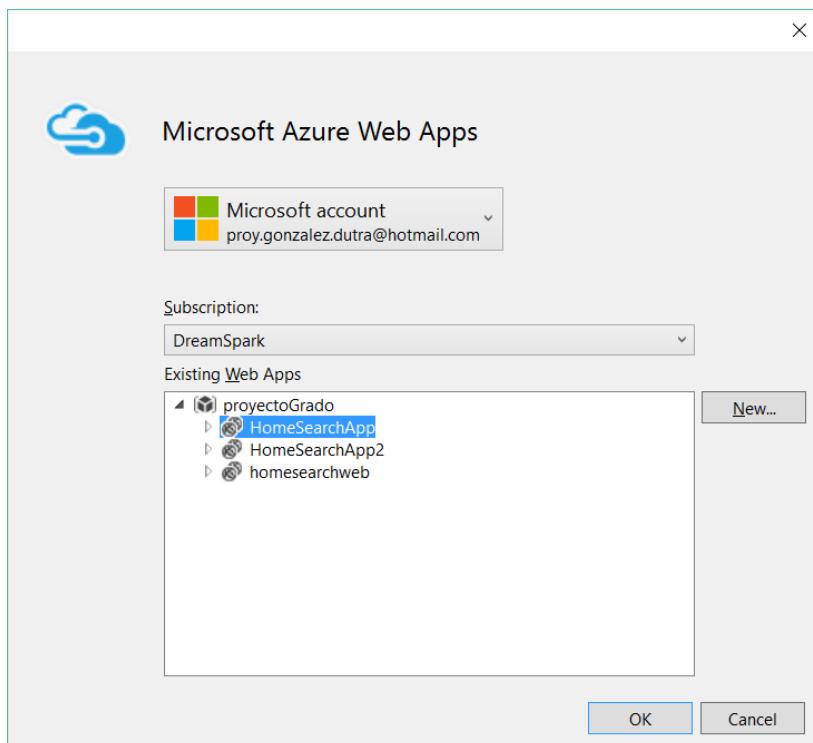


Ilustración 49 Publicación de aplicaciones.

### 7.15.3 Base de Datos

Para la base de datos se debe crear una instancia en el portal Azure para luego obtener las cadenas de conexión a ser utilizadas por las aplicaciones.

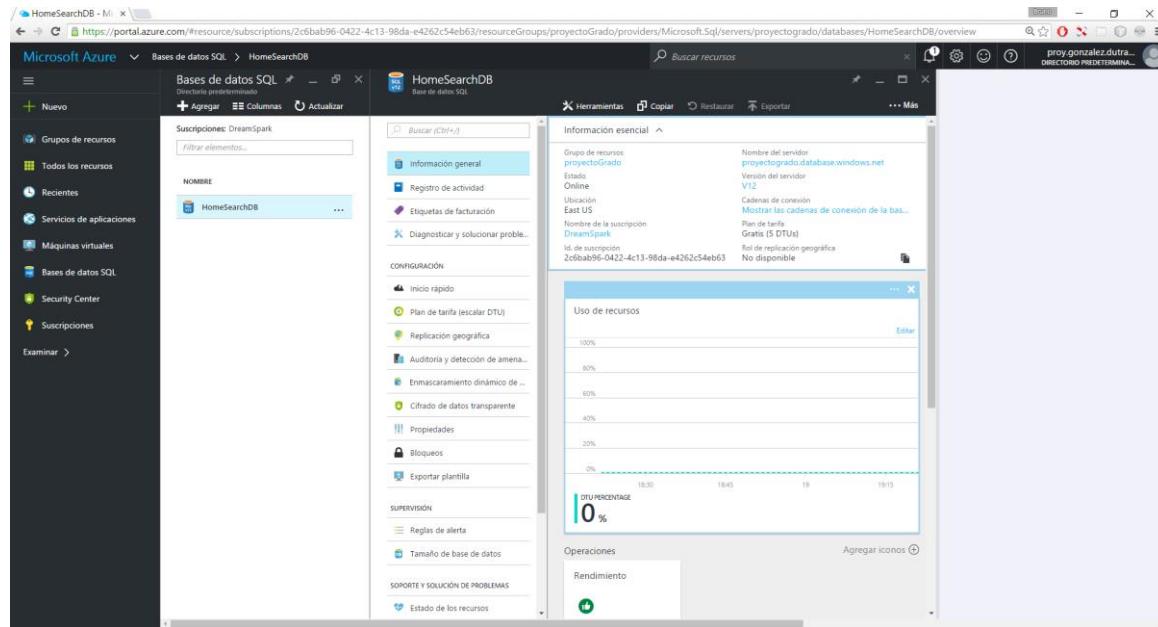


Ilustración 50 Base de datos Azure.

## 7.16 Anexo P: Utilización del Id de sistema externo

En el desarrollo del sistema se utilizó el Id que las propiedades poseen en el sistema ya existente de López & Cia. Esta decisión se adoptó con el fin de facilitar el acceso a las propiedades desde nuestro sistema. Para esto, se definió un atributo de las propiedades que se lo nombró "*ExternalId*".

Al ya existir un sistema en el cual se encontraba toda la información de las propiedades y desde el cual el equipo de desarrollo importó los datos, mediante prácticas desarrolladas (Anexo E) y la utilización del "*ExternalId*" se pudo realizar dicha importación con mayor facilidad.

## 7.17 Anexo Q: Investigación e implementación de APIs de Google

*Google* nos brinda una serie de servicios para acceder a datos de lugares cercanos a una localidad. A partir de una ubicación (latitud y longitud) y tipos de servicios nos devuelve una lista de lugares que están ingresados en *Google Maps*. Optamos por hacer la llamada desde el lado del servidor ya que se puede utilizar tanto para el mapa de servicios como para la búsqueda de propiedades por perfil. Para esto optamos por utilizar los métodos de *Places API Web Service*.

*API Web Service:* <https://developers.google.com/places/web-service/?hl=es>

Para la implementación de los lugares cercanos en la vista se utilizaron los métodos de *Places JavaScript Library*. A su vez para poder marcar los puntos en el mapa se desarrolló sobre los métodos de *Maps JavaScript API*.

*Maps JavaScript API:*

<https://developers.google.com/maps/documentation/javascript/markers?hl=es>

*Places JavaScript Library:* <https://developers.google.com/places/javascript/?hl=es>

## 7.18 Anexo R: Modelo base de datos

A continuación se detalla el modelo de base de datos final. Se utilizó Entity Framework Model First, esto hace que el modelo esté normalizado según la documentación de Entity Framework. [4]

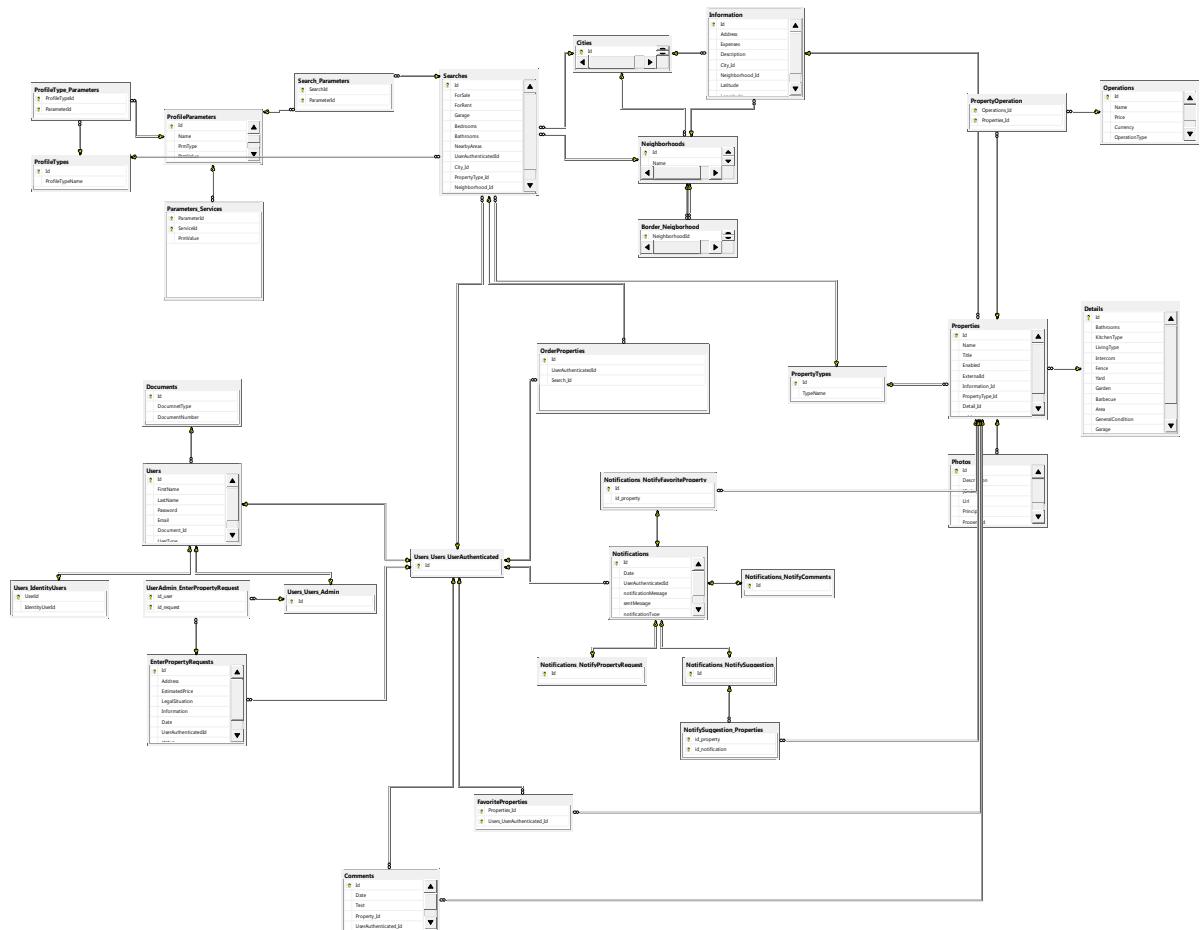


Ilustración 51 Modelo de Base de Datos.

## 7.19 Anexo S: Diagrama de clases

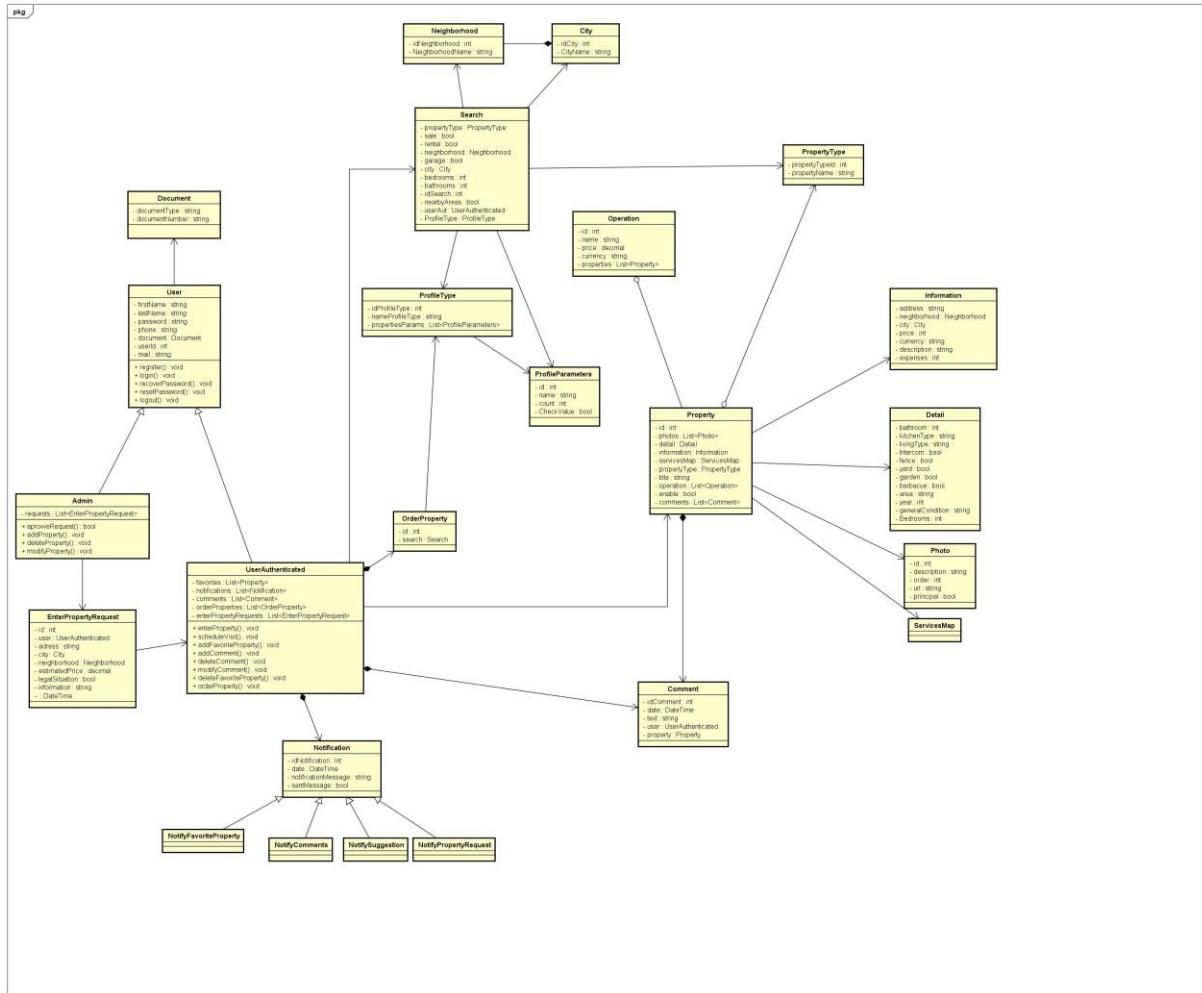


Ilustración 52 Diagrama de clases.

## 7.20 Anexo T: Manual de usuario

### 7.20.1 Registro de usuario

Para el registro de un nuevo usuario se deberán ingresar los datos del mismo, estos son nombre, apellido, *email* y contraseña. Todos los campos son obligatorios. Al hacer *click* en registrarse, si los datos son válidos el usuario queda autenticado en el sistema y se lo redirige al listado de propiedades.

The screenshot shows a web browser window titled 'Registro - My ASP.NET'. The URL in the address bar is 'homeseachweb.azurewebsites.net/Account/Register'. The page header includes a logo for 'López & Cia.', a 'Home' link, a 'Propiedades' dropdown menu, and 'Register' and 'Log In' buttons. The main content area is titled 'Registro.' and contains a form with the following fields:

Nombre	Pepe
Apellido	Perez
Tipo de documento	CI
Documento	51378709
Email	pepe@mail.com
Password	.....
Confirm password	.....

At the bottom of the form is a 'Registrarse' button.

Ilustración 53 Registro de usuario.

### 7.20.2 Inicio de sesión

Para iniciar sesión se solicita el correo electrónico y contraseña, una vez validadas las credenciales el usuario queda autenticado en el sistema y se lo redirige al listado de propiedades.

The screenshot shows a web browser window titled 'Log in - My ASP.NET'. The URL in the address bar is 'homeseachweb.azurewebsites.net/Account/Login'. The page header includes a logo for 'López & Cia.', a 'Home' link, a 'Propiedades' dropdown menu, and 'Register' and 'Log In' buttons. The main content area is titled 'Log in.' and contains a form with the following fields:

Email	pepe@mail.com
Password	.....

Below the fields are 'Remember me?' checkboxes and 'Log In' and 'Register as a new user' buttons.

Ilustración 54 Inicio de sesión.

### 7.20.3 Búsqueda de propiedades

El sistema permite realizar búsqueda de propiedades en base a distintos parámetros. Estos son parámetros de localidad (ciudad, barrio y zonas cercanas), tipo de operación (venta o alquiler), parámetros de propiedad (tipo de propiedad, garaje, dormitorios y baños). A su vez permite realizar búsquedas por perfil de usuario. Al seleccionar un perfil se buscarán propiedades en base al criterio establecido para cada uno, también se brinda la posibilidad de ingresar parámetros de perfil los cuales serán tomados en cuenta para destacar los servicios cercanos en base a los parámetros ingresados. El resultado de propiedades será organizado en base a la cantidad de coincidencias de servicios.

#### Parámetros de localidad:



Ilustración 55 Parámetros de localidad. Un formulario que incluye campos para Ciudad (seleccione ciudad), Barrio (seleccione barrio), tipos de operación (Venta y Alquiler) y zonas cercanas.

Ciudad: seleccione ciudad	Barrio: seleccione barrio	Venta <input type="checkbox"/> Alquiler <input checked="" type="checkbox"/>	Zonas cercanas <input type="checkbox"/>
------------------------------	------------------------------	---	---

Ilustración 55 Parámetros de localidad.

#### Parámetros de propiedad:



Ilustración 56 Parámetros de propiedad. Un formulario que incluye campos para Tipo de propiedad (seleccione tipo de propiedad), Garage (radio botones Si/No), Dormitorios (seleccione número de dormitorios), Baños (seleccione número de baños).

Tipo de propiedad seleccione tipo de propiedad	
Garage <input checked="" type="radio"/> Si <input type="radio"/> No	
Dormitorios seleccione número de dormitorios	
Baños seleccione número de baños	

Ilustración 56 Parámetros de propiedad.

## Parámetros de perfil:

Búsqueda por perfil

- Empresa
- Pareja Joven
- Soltero/a
- Familia
  - **mascota**
  - **deportista**
  - **auto**
  - **Hijos**
  - **Hijos Niños**
  - **Hijos Adolescentes**
  - **Hijos Adultos**
  - **Entretenimiento**
- Estudiantes

**BUSCAR**

Ilustración 57 Parámetros de perfil.

### 7.20.4 Comparación de propiedades

En el listado de propiedades, se podrán comparar propiedades. Para esto se deberá seleccionar las propiedades a comparar haciendo *click* en el ícono de comparación y luego hacer *click* en el botón de comparar. Solamente se podrán comparar dos propiedades.

#### Propiedades seleccionadas:

Comparar propiedades	
	
<b>Bacigalupi</b>  <ul style="list-style-type: none"><li>• Aguada - Montevideo</li><li>• 4 dormitorios</li><li>• Bacigalupi y Estrella del Norte</li><li>• Casa</li></ul> <b>U\$S151000 - precio</b> <b>Venta</b>	<b>La Paz</b>  <ul style="list-style-type: none"><li>• Cordon - Montevideo</li><li>• 3 dormitorios</li><li>• La Paz y Gaboto</li><li>• Apartamento</li></ul> <b>U\$S120000 - precio</b> <b>Venta</b>

Ilustración 58 seleccionar propiedades a comparar.

## Comparación:

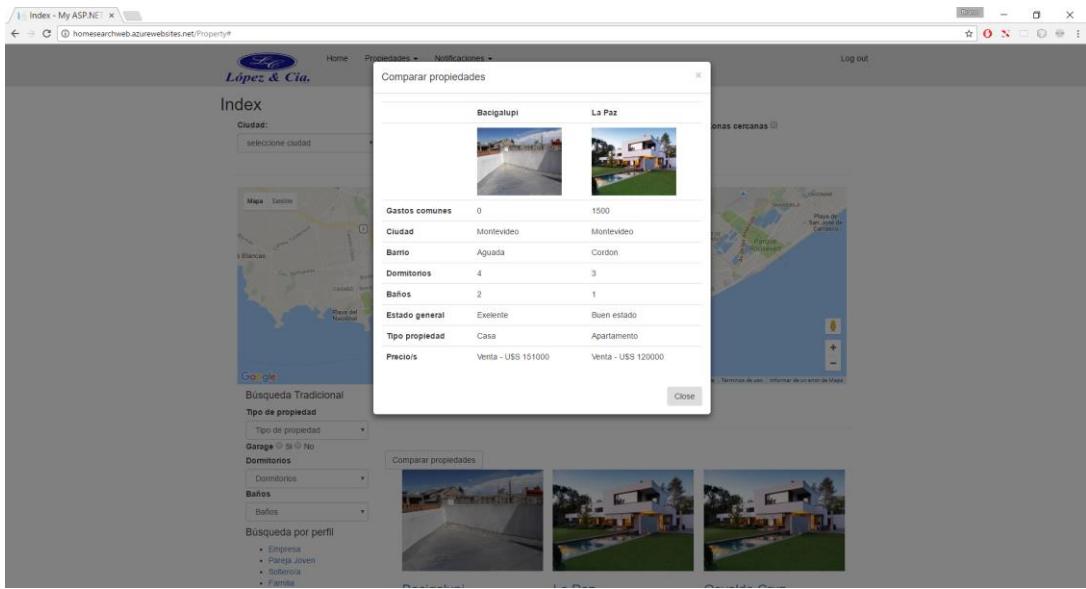


Ilustración 59 Comparar propiedades.

### 7.20.5 Detalle de propiedad

#### 7.20.5.1 Agenda de visita

Para solicitar una agenda una visita a una propiedad hacer *click* en la opción “Agendar visita”, se abrirá un modal con las opciones a ingresar para la agenda, estas son, fecha y hora de disponibilidad (obligatorios) y un mensaje (opcional). Se deberá ingresar una fecha mayor a la fecha actual.

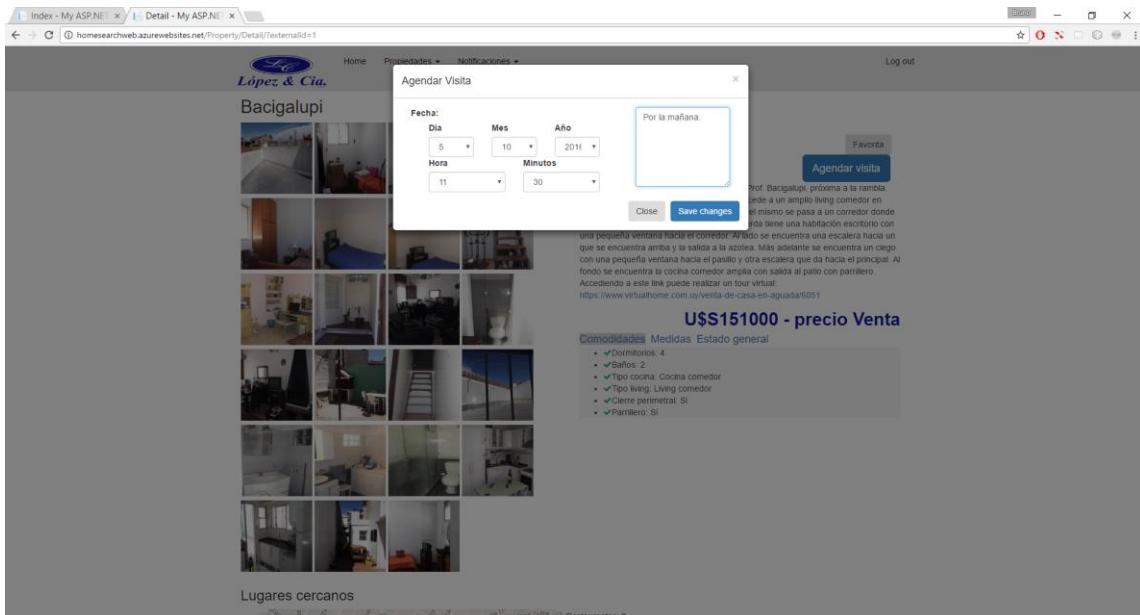


Ilustración 60 Agenda de visita.

### 7.20.5.2 Agregar propiedad favorita

Los usuarios autenticados tendrán la opción de agregar una propiedad como favorita. Para esto tendrá habilitado un botón para agregarla a la lista de favoritas. En caso de que la propiedad ya sea favorita tendrá la opción de eliminarla de la lista.

#### Marcar como favorita:

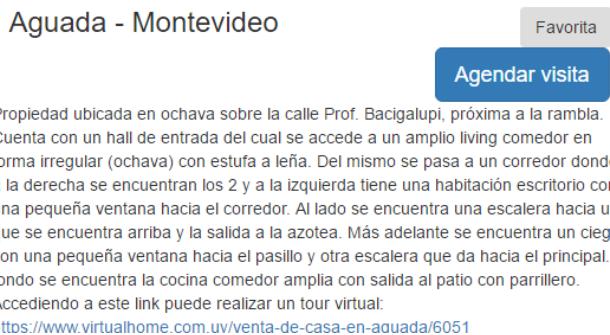


Ilustración 61 Marcar como favorita.

#### Propiedad agregada como favorita:

Propiedad agregada a lista de favoritas.

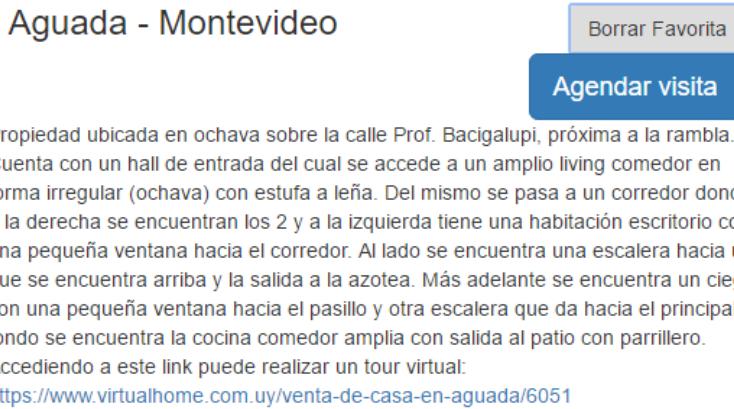


Ilustración 62 Propiedad agregada como favorita.

Luego de agregada como favorita se podrá ver en la lista de propiedades favoritas donde se podrá quitar haciendo *click* en el botón eliminar.

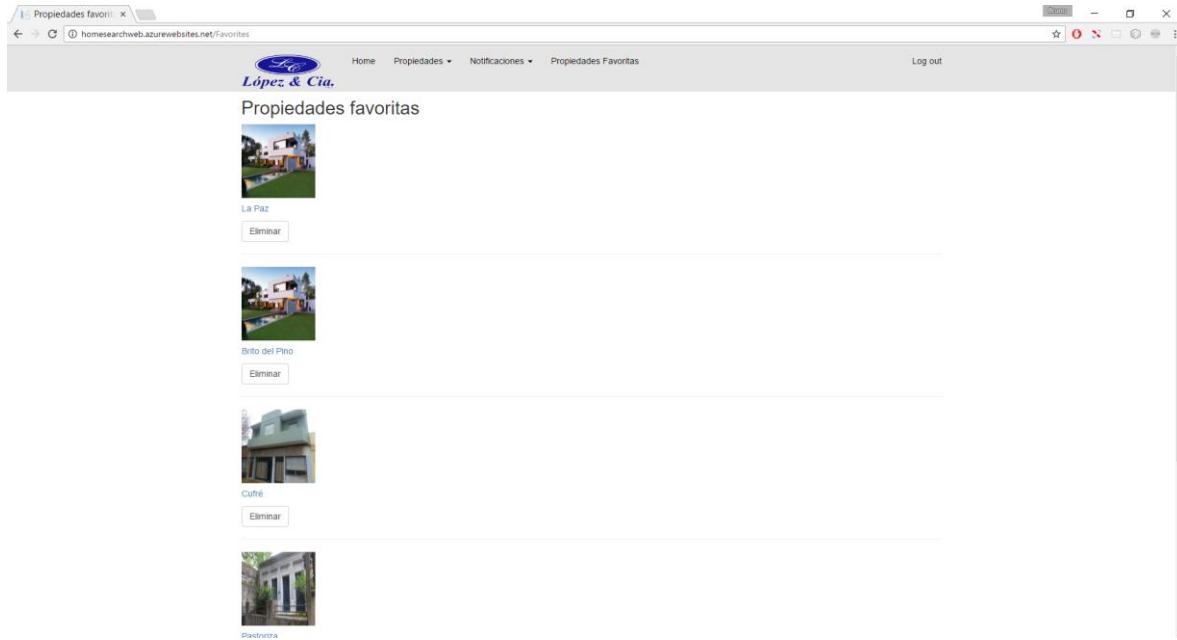


Ilustración 63 Listado de propiedades favoritas.

#### 7.20.5.3 Comentarios

Para ingresar un comentario se deberá estar autenticado en el sistema, luego de ingresado el comentario se mostrará en el listado con la opción de eliminarlo.

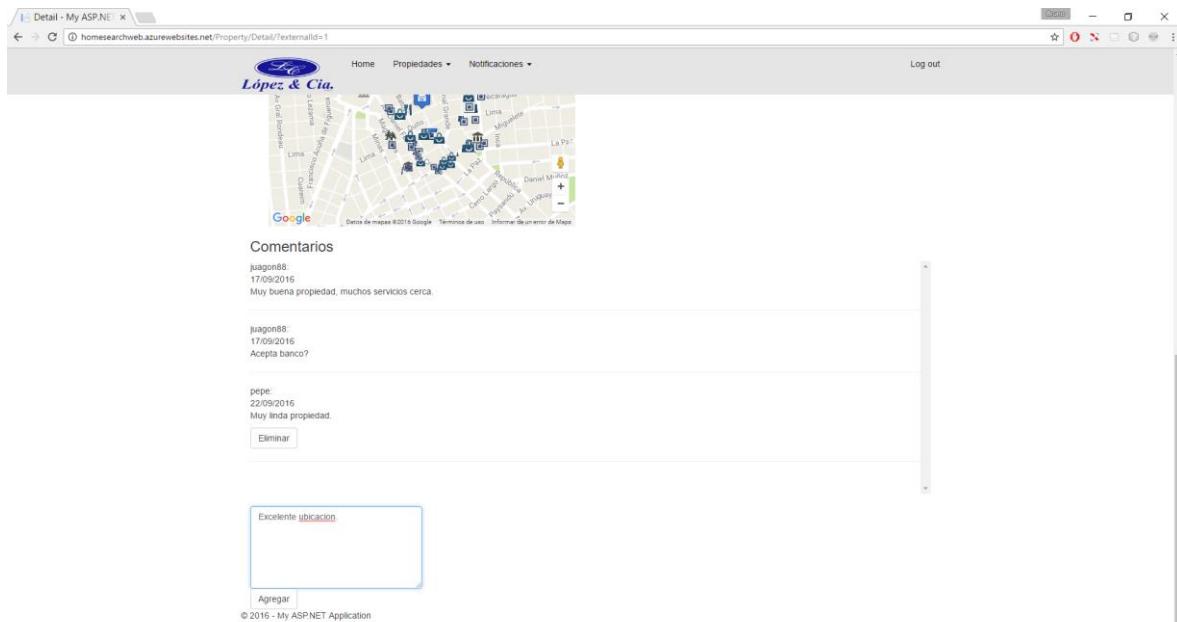
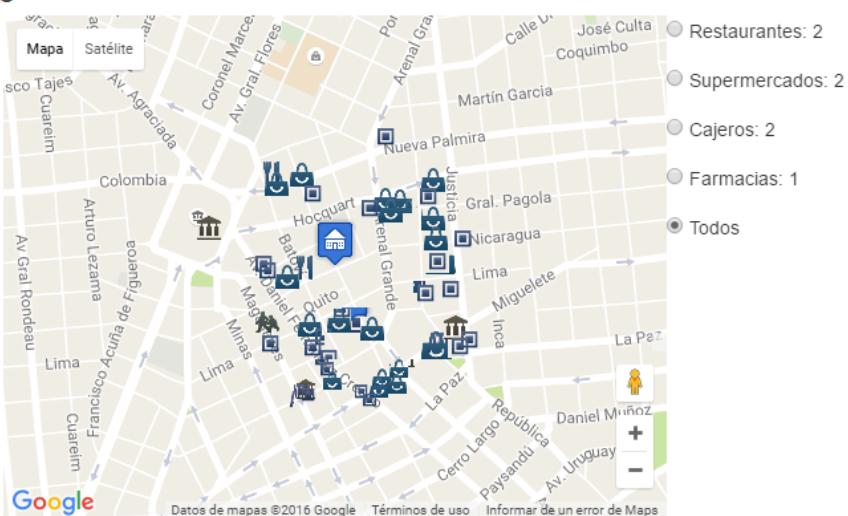


Ilustración 64 Comentarios.

#### 7.20.5.4 Mapa con servicios

Se mostrará un mapa con la localidad de la propiedad y los servicios que tiene a su alrededor. Para filtrar por tipo se deberá seleccionar la opción que desee y se mostrarán solamente los servicios del tipo seleccionado.

## Lugares cercanos



## Ilustración 65 Lugares cercanos.

## 7.20.6 Notificaciones

Para ver el listado de notificaciones se deberá estar autenticado en el sistema. Se mostrarán los tipos de notificaciones. Al seleccionar se mostrarán las notificaciones del tipo seleccionado.

Notificaciones ▾

- Comentarios
- Propiedades favoritas
- Sugerencias
- Pedidos de propiedad

## Notificaciones

22/09/2016:  
Nuevo comentario para propiedad: Bacigalupi  
Test Test ha realizado un comentario:  
**excelente ubicacion.**  
[Ver propiedad](#)

---

22/09/2016:  
Nuevo comentario para propiedad: Bacigalupi

## Ilustración 66 Notificaciones.

## 7.21 Anexo U: Credenciales de prueba para corrector

Se adjunta el archivo de texto “*readme.txt*” donde se encuentran las *urls* de la *API* y del sitio *web*. Así como también las credenciales.

Con estos datos se podrá ingresar al sitio como usuario autenticado permitiéndole utilizar las funcionalidades para los mismos.

También podrá probar los métodos implementados del servicio *web*, para esto le recomendamos la utilización de la herramienta *PostMan* ayudándose con la documentación de la *API* que la podrá encontrar en la *url* proporcionada.

Para los métodos que requieren autenticación deberá obtener el *Token* y proporcionarlo en el *header* de cada *request*.

Obtención de *Token*:

Se deberá realizar una llamada al método “*/Token*” con los datos de acceso com se puede ver en la siguiente imagen:

```
1 > [ { "access_token": "ItzIQlOvok3ea3dmE2-NGcIDCYjHxiiz2a16sXeakZMDKvEfdfSoB4pDQHZfp5VJxhF21CRw02k6a7oNcoh7xV80sfys-vFvjkChGan75n-j_jprs36Vt8wLtvjD7ob3Qo2t-mJ5Afz1Z1fTThMyY5kr- d8YKUfn1018pdz2pkhndwnuo593d80KHKn4nmABF2zfg7cpEqM0btbnyPFOAK9wrsj6c1jTLhyGQoGGVhD0sn6sb55Zgr7d16nLdC77NAst5TAHmKEX05wODIBay4rvlqQn121Ta420R1oER2Epv71wACGenT1ft93taakTlgEixFI7kLU0hr1KL27xxKs8 Udy1foSqdEg17L76J09hE_EZTAAQngY8rcRVJX_Iryau0oQudepq9w1B1jzuUz060XqvuAljyjuvNzpr0338s3rvLqTqK8pZCFHbLxSPXF4-WF4L_h1g-35k5mnjWQjsFdH1jndUKg6X9A8-PASyLH_kej431B6FmFyP3tv -Zed_urOSTp9U_kt5DCByCoyZdnxuV830", "token_type": "bearer", "expires_in": 1209599, "username": "Corrector_proyecto.2016@gmail.com", "issued": "Mon, 26 Sep 2016 23:51:09 GMT", "expires": "Mon, 18 Oct 2016 23:51:09 GMT" } ]
```

Ilustración 67 Obtención de *Token*.

Luego se deberá incluir el token en el *header* de cada *request* en el parámetro *Authorization* antecedido de la palabra *Bearer* como se muestra en la siguiente imagen:

```
Authorization: Bearer fg5ThK0WmHsYCnfKyp0jQvUX6r6T_EUKhNrXkomloR_fNIhdd6XhmhkCUGXcPGAtC0
```

Ilustración 68 Incluir *token* en *request*.

## 7.22 Anexo V: Google Analytics

En reunión con el cliente, el mismo nos manifiesta la necesidad que tiene la empresa de poder contar en el sistema con la herramienta *Google Analytics*. Mediante esta, se obtienen datos de suma importancia para la empresa los que aprovecharan para su beneficio.

A continuación, algunos de los puntos destacados por el cliente para obtener mediante *Google Analytics*, junto con su gráfica correspondiente:

- Tráfico, cantidad de visitantes que ingresan al sistema: es importante para el cliente saber el tráfico que posee el sistema para así, tener conocimiento de si el mismo es visitado asiduamente o no. De darse esto último, deberá plantearse cómo poder hacer llegar el sistema a más usuarios.



Ilustración 69 Gráfica de tráfico.

- Horarios y días en que ingresan más visitantes al sistema: este dato le es útil por ejemplo para el caso de que si en algún momento desea promocionar alguno de sus productos, sabrá cuál es el mejor día y horario (estadísticamente) para hacerlo.

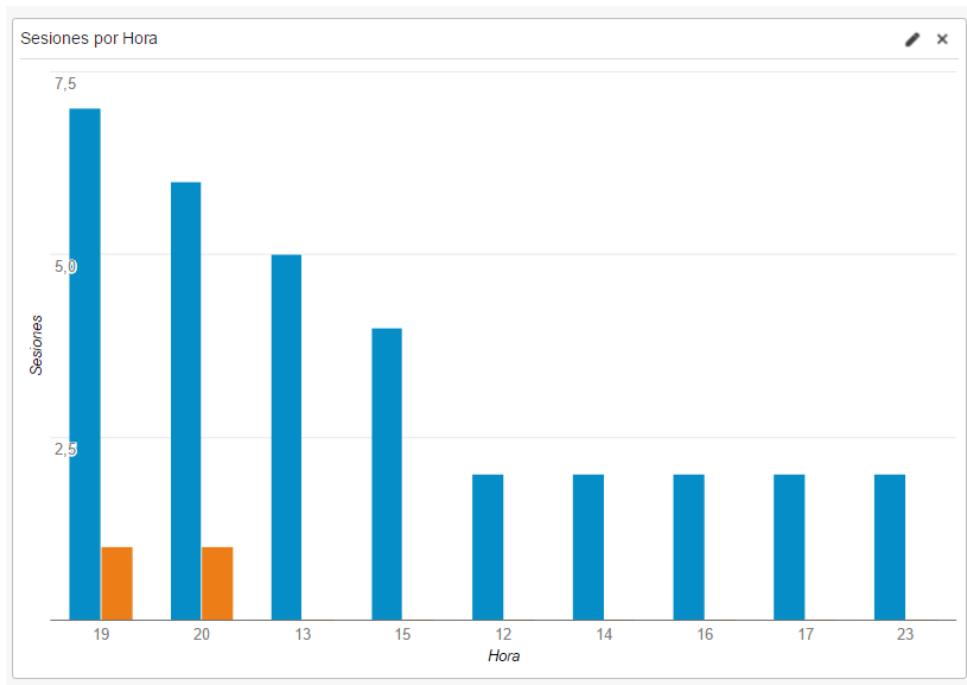


Ilustración 70 Gráfica de sesiones por hora.

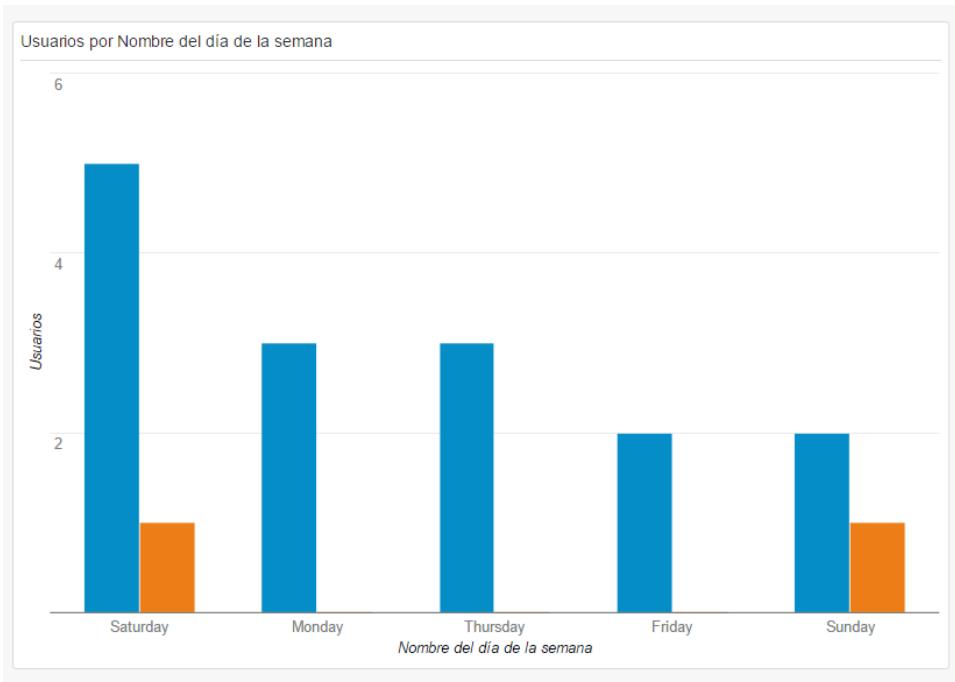


Ilustración 71 Gráfica de Usuarios por día de la semana.

- De qué países ingresan los visitantes: este dato es útil para saber la cantidad de visitas que se dan desde nuestro país lo cual es importante ya que es donde se desarrolla la actividad de la empresa. Por otro lado, en caso de tener visitas desde el exterior puede aprovechar estos datos para captar inversores extranjeros.

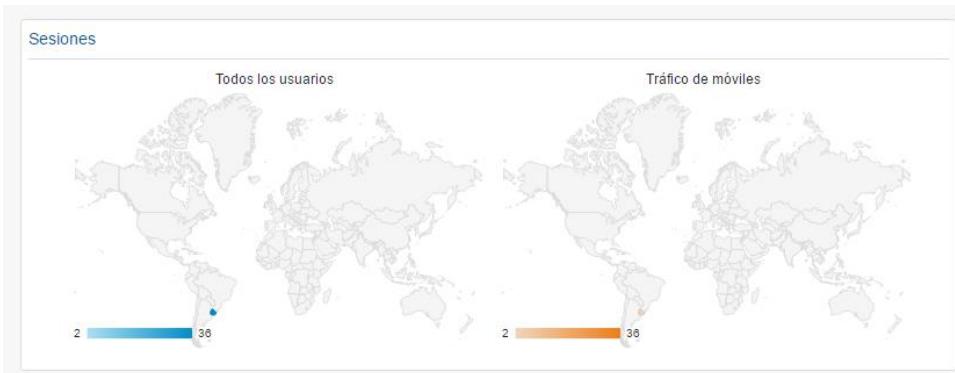


Ilustración 72 Gráfica de sesiones por país.

- Páginas y propiedades más visitadas y menos visitadas: es importante saber cuáles son las páginas y propiedades que más y menos se visitan. Para el caso de las propiedades, el cliente podrá tener una idea de qué propiedad tiene más demanda y cual no, y así en cada caso tomar las decisiones que crea conveniente para fortalecer lo bueno y promocionar lo que menos se visita.

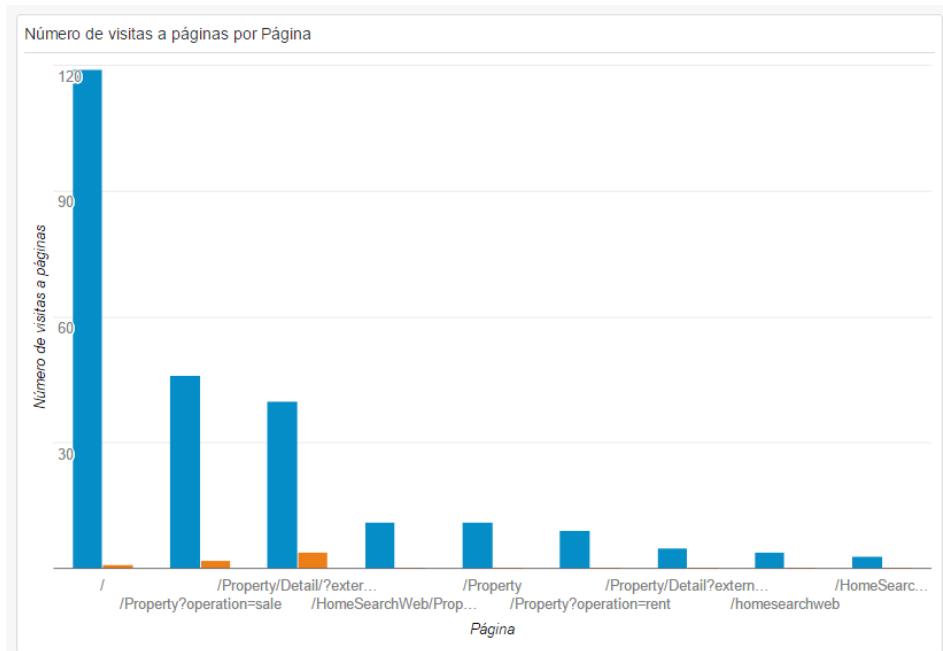


Ilustración 73 Gráfica de páginas más visitadas.

- Comportamiento de los visitantes: con este dato el cliente tendrá un panorama de cómo se comportan los visitantes. Es decir, luego de ingresar al sistema saber a dónde se re dirigen, si es a la sección de ventas, alquileres o alguna propiedad en particular.

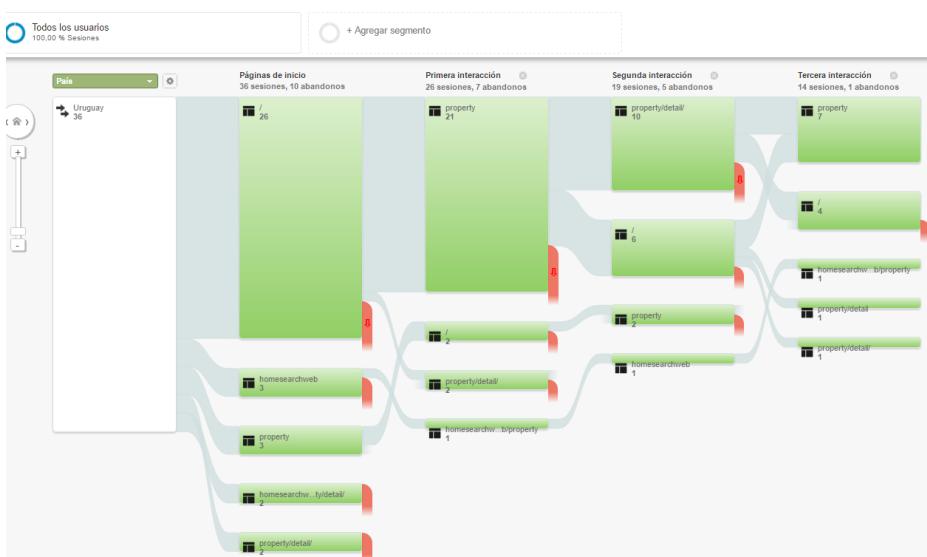


Ilustración 74 Datos de comportamiento de usuarios.

- Desde donde ingresan: es importante para el cliente saber desde donde se ingresa al sistema, si fue por *mailing*, buscador de Google, redes sociales, etc. Estos datos le servirán por ejemplo para saber dónde fortalecer la publicidad de su empresa.

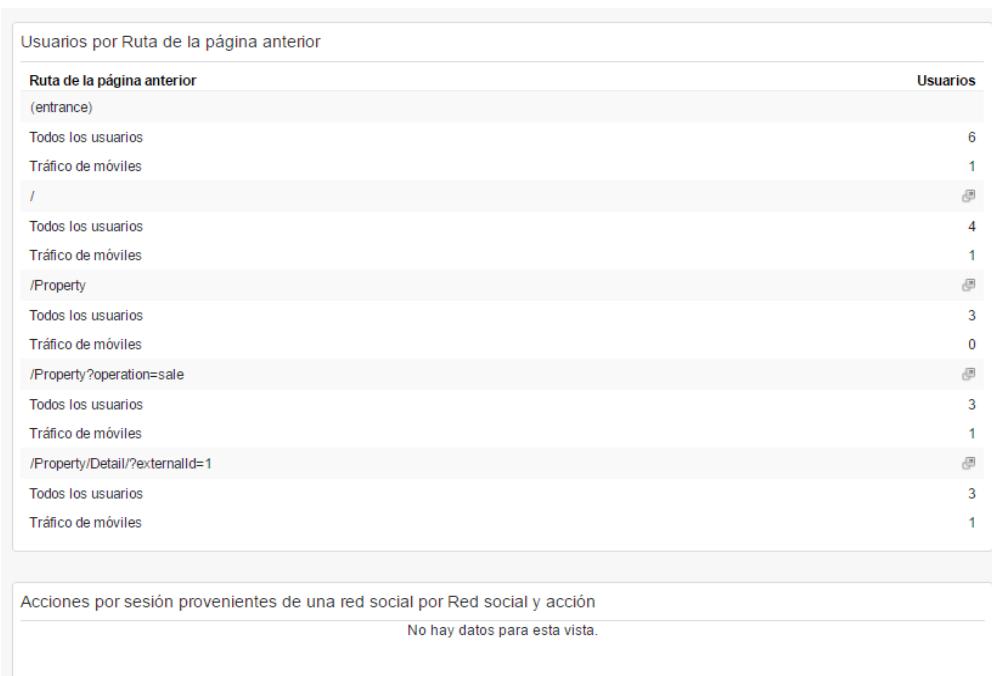


Ilustración 75 Datos de páginas desde donde ingresan los usuarios.

- Desde qué dispositivos ingresan al sistema: este dato es útil por ejemplo para el caso de que el sistema tenga mucho tráfico desde dispositivos móviles. Esto hará plantearse al cliente la importancia que puede tener realizar una aplicación móvil y/o mejorar la vista para estos dispositivos. En las tablas de datos que se muestran anteriormente se observa en color naranja los ingresos desde dispositivos móviles.

## 7.23 Anexo W: Evidencias de testing

### Búsqueda tradicional:

The screenshot shows a web-based real estate search platform. At the top, there's a navigation bar with links for Home, Propiedades, Notificaciones, Propiedades Favoritas, and Log out. Below the navigation is a search form with fields for Ciudad (Montevideo) and Barrio (Buceo), and dropdowns for Venta (checkmark) and Alquiler. A 'Zonas cercanas' button is also present. The main area features a map of Montevideo with various neighborhoods labeled. A specific property listing for '1 propiedad encontrada' is highlighted in the center. The listing includes a thumbnail image of a modern house with a pool, the address 'Buceo - Montevideo', and details about it being a 2 dormitorios property. Below the listing, there's a section titled 'Asamblea' with a list of properties and their details.

Ilustración 76 Búsqueda tradicional.

### Búsqueda tradicional con zonas cercanas:

This screenshot is from the same real estate platform, but with the 'Zonas cercanas' button selected in the search form. The results page now displays '3 propiedades encontradas'. The first two properties listed are 'Brito del Pino' and 'Masini | VENDIDO', each with a thumbnail image and a brief description. The third property listed is 'Asamblea', which is identical to the one shown in the previous screenshot. The interface remains largely the same, with the map and search filters visible.

Ilustración 77 Búsqueda tradicional con zonas cercanas.

## Búsqueda por perfil:

Búsqueda Tradicional 19 propiedades encontradas

Tipo de propiedad

Garaje  Si  No

Dormitorios

Baños

Baños

Búsqueda por perfil

- Empresa
- Pareja Joven
  - + mascota
  - + deportista
  - + auto
  - + Entretenimiento
- Solteros/a
- Familia
- Estudiantes

**BUSCAR**

**Brito del Pino** **Magariño Cervantes** **Pereira - Apto**

**US\$150000 - precio Venta** **US\$167000 - precio Venta** **US\$165000 - precio Venta**

Ilustración 78 Búsqueda por perfil.

## Búsqueda por perfil con parámetros (resultado ordenado respecto al parámetro):

Búsqueda Tradicional 19 propiedades encontradas

Tipo de propiedad

Garaje  Si  No

Dormitorios

Baños

Baños

Búsqueda por perfil

- Empresa
- Pareja Joven
  - + mascota
  - + deportista
  - + auto
  - + Entretenimiento
- Solteros/a
- Familia
- Estudiantes

**BUSCAR**

**Pagola | VENDIDO** **Magariño Cervantes** **Masini | VENDIDO**

**US\$125000 - precio Venta** **US\$167000 - precio Venta** **US\$295000 - precio Venta**

Ilustración 79 Búsqueda por perfil con parámetros.