



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

**HEURÍSTICA DE VALIDACIÓN DE INFORMACIÓN GEORREFERENCIADA,
BASADA EN CROWDSOURCING Y COMPUTACIÓN SOCIAL**

**TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN CIENCIAS MENCIÓN
COMPUTACIÓN**

CHRISTIAN EDUARDO PALOMARES PERALTA

**PROFESOR GUÍA:
SERGIO OCHOA DE LORENZI**

**MIEMBROS DE LA COMISIÓN:
JOSÉ A. PINO URTUBIA
BARBARA POBLETE LABRA
JAIME NAVÓN COHEN**

**SANTIAGO DE CHILE
2015**

Resumen

Uno de los problemas más significativos a resolver en sistemas que generan información en base a crowdsourcing, ha sido el controlar la calidad de la misma. A pesar de la existencia de casos conocidos que han resuelto este problema en forma exitosa (por ejemplo, Wikipedia o ReCaptcha), las soluciones encontradas son ad hoc al problema abordado, por lo tanto carecen de generalidad y no se pueden aplicar de la misma forma a otros escenarios.

Por otra parte, la masificación de la computación móvil ha llevado a que el problema de verificar la calidad de la información que se ingresa a través de crowdsourcing, se haga presente en diversos escenarios, por ejemplo en la validación de la información georreferenciada que ingresan los usuarios de aplicaciones móviles acerca de diversos puntos de interés como: bares, restaurantes, colegios, hospitales, servicios públicos, etc.

Las soluciones disponibles para validar información georreferenciada usualmente no involucran mecanismos crowdsourcing, y las que lo hacen, plantean mecanismos de validación un tanto limitados. Debido a esto, este trabajo de tesis busca explorar ese camino como una nueva alternativa de solución al problema planteado. Particularmente este trabajo desarrolló una heurística que permite la validación de la información que ingresan los usuarios, mientras éstos se desplazan por la ciudad (por ejemplo, tags). La heurística utiliza conceptos de computación social y crowdsourcing para reducir la incertidumbre acerca de la validez de dicha información.

Para evaluar la solución se desarrolló un sistema móvil y una API (Application Programming Interface). La heurística propuesta se implementó y se dejó disponible a través de esta API para que otros desarrolladores puedan hacer uso de ésta. Por una cuestión de factibilidad del proceso de evaluación, dicho proceso fue realizado en Lima, Perú, y contó con la participación de 30 usuarios. Estos usuarios contaban con experiencia en el uso de aplicaciones similares a la presentada en esta tesis.

A pesar de que los resultados obtenidos son insuficientes para sacar conclusiones definitivas respecto a la efectividad de la heurística propuesta,

los datos obtenidos nos dan fuertes indicios de que la estrategia de solución propuesta es factible de usar y es útil para la validación de información georreferenciada.

Agradecimientos

Agradezco a mis padres, quienes me apoyaron todo el tiempo, siempre dándome fuerzas para no rendirme, para seguir adelante todos estos años.

Quiero agradecer también a mis amigos, compañeros de trabajo, familiares, y a todos aquellos que, ya sea en persona o a la distancia, me brindaron siempre su apoyo incondicional y nunca dudaron en que lograría llevar a cabo este proyecto.

Finalmente, agradezco a mi profesor guía por su constante y continuo apoyo en la realización de este trabajo de tesis.

Este trabajo de tesis ha sido parcialmente financiado por el Proyecto Fondecyt Nro. 1120207.

Tabla de contenido

1. Introducción	1
1.1 Hipótesis del trabajo	3
1.2 Objetivos de la tesis	3
1.3 Metodología	4
1.4 Estructura de la tesis	4
2. Marco teórico	6
2.1 Crowdsourcing y validación	6
2.2 Context-awareness	8
2.3 Participatory sensing	11
2.4 Gamification	13
2.5 Captcha y Recaptcha	15
2.6 Información social para los sistemas de recomendación	17
2.7 Sistemas de recomendación	17
2.8 Aplicaciones basadas en crowdsourcing	18
2.8.1 Wikipedia	18
2.8.2 Foursquare	19
2.8.3 Facebook places	21
3. Definición de la heurística de validación	23
3.1 Motivación	23
3.2 Modelo de datos utilizado	25
3.2.1 Trabajo relacionado	26
3.2.2 Representación de eventos e ítems para la heurística	27
3.2.2.1 Lugar	27
3.2.2.2 Usuario	27

3.2.2.3	Ítem	28
3.2.2.4	Evento	28
3.2.2.5	Acotación de representación	29
3.3	Estrategia de validación	30
3.3.1	Agregar tag	33
3.3.2	Agregar lugar	38
3.3.3	Agregar propuesta de modificación de información de lugar	39
3.3.4	Agregar comentario/oferta	42
3.4	Robustez de la heurística	42
3.5	Ponderación de la fórmula	46
3.6	Temporalidad de la información	47
4.	Herramienta implementada	48
4.1	Ambiente operacional	48
4.2	Requisitos de la solución	49
4.3	Interfaces de la aplicación cliente	51
4.3.1	Interfaz de inicio de sesión	51
4.3.2	Interfaz de timeline	52
4.3.3	Interfaz de ranking	53
4.3.4	Interfaz de perfil de usuario	54
4.3.5	Interfaz de búsqueda de lugares cercanos	55
4.3.6	Interfaz de agregar nuevo lugar	56
4.3.7	Interfaz de información del lugar	57
4.3.8	Interfaz de votación de tags	58
4.3.9	Interfaz de propuesta de modificación del lugar	59
4.3.10	Interfaz de listado de comentarios	61

4.3.11	Interfaz de listado de ofertas.....	62
4.3.12	Interfaz de listado de tags.....	63
4.4	Interfaces API / mantención	64
4.4.1	Interfaz de registro de usuarios.....	64
4.4.2	Interfaz de mantención de usuario.....	65
4.4.3	Interfaz de actividad de usuario	66
5.	Resultados obtenidos.....	68
5.1	Escenario de experimentación.....	68
5.2	Proceso de Experimentación	68
5.3	Datos obtenidos	70
5.4	Validación de la hipótesis	72
6.	Discusión.....	75
7.	Conclusiones y trabajo a futuro	77
8.	Bibliografía	79

Índice de figuras

Figura 1: Arquitectura de un sistema sensible al contexto.....	10
Figura 2: Mapa entre deseos humanos y mecánicas de juego.....	15
Figura 3: Versión de la Wikipedia para dispositivos móviles.....	19
Figura 4: Aplicación de Foursquare para iOS	21
Figura 5: Aplicación de Facebook para Android.....	22
Figura 6: Ejemplo de la representación visual de un badge	24
Figura 7: Ejemplo de la representación visual de un leaderboard	25
Figura 8: Diagrama de flujo de validación inicial de tags.....	33
Figura 9: Diagrama de flujo de agregar tag	34
Figura 10: Diagrama de flujo de validación de un tag	35
Figura 11: Diagrama de flujo de ponderado de respuestas tag	36
Figura 12: Diagrama de flujo de votación de lugar	38
Figura 13: Diagrama de flujo validación de propuesta de modificación de información de lugar	41
Figura 14 Comportamiento esperado de un usuario malintencionado.....	44
Figura 15. Comportamiento esperado de un usuario bienintencionado.....	45
Figura 16: Casos de uso de la solución	50
Figura 17: Interfaz de inicio de sesión.....	52
Figura 18: Interfaz de timeline	53
Figura 19: Interfaz de ranking.....	54
Figura 20: Interfaz de perfil de usuario	55
Figura 21: Interfaz de búsqueda de lugares cercanos.....	56
Figura 22: Interfaz de agregar nuevo lugar.....	57

Figura 23: Interfaz de información de lugar	58
Figura 24: Interfaz de votación de tags	59
Figura 25: Interfaz de propuesta de modificación de información de lugar .	60
Figura 26: Interfaz de validación de propuesta de modificación de información de lugar	61
Figura 27: Interfaz de listado y agregado de comentarios.....	62
Figura 28: Interfaz de listado y agregado de ofertas	63
Figura 29: Interfaz de listado y agregado de tags	64
Figura 30: Interfaz de registro de usuarios	65
Figura 31: Interfaz de mantención de usuario	66
Figura 32: Interfaz de actividad de usuario	67
Figura 33: Interfaz de seguimiento de actividades de usuario	69

Índice de tablas

Tabla 1. Tabla de acrónimos de heurística	43
Tabla 2. Resultado de lugares validados	70
Tabla 3. Resultado de tags validados	71
Tabla 4. Resultado de modificaciones validadas	72
Tabla 5. Resultado de votos en comentarios	72
Tabla 6. Resultado de votos en ofertas	72
Tabla 7. Validación de resultados obtenidos en lugares validados con heurística	73

1. Introducción

Desde la aparición de la llamada Web 2.0, los usuarios de diversos servicios en Internet nos hemos vuelto colaboradores, pues día a día entregamos algo de información al servicio que estemos usando. Por ejemplo, escribiendo un comentario en un blog, subiendo una foto a Picasa, "tuiteando" sobre lo que estamos haciendo o viendo, o subiendo un video a YouTube. Es decir, nos hemos vuelto creadores, escritores y registradores de un pedazo de información de Internet.

Sumado a esto, debemos de tomar en cuenta el incremento de dispositivos móviles con capacidad de computación que han inundado a la sociedad. Debido a su rápida difusión y bajo coste de éstos, el incremento de usuarios de dispositivos móviles como celulares inteligentes (*smartphones*) ha llegado a una cifra de poco más de 3 mil millones. Considerando este dato, podemos ver que el ingreso de los dispositivos móviles al mercado ha generado más oportunidades, desde donde los usuarios podemos generar información y compartirla a través de Internet.

Debido a este incremento explosivo de la información en Internet, muchos sitios web hicieron uso de esta información pública para generar contenido en sus páginas. En paralelo a esto, otros sitios ofrecen plataformas para que los usuarios puedan generar contenido específico, como es el caso de los sitios de noticias, con secciones como "reportero ciudadano", o los motores de blogs que, valga la redundancia, permiten la generación de blogs. YouTube con su plataforma de videos es otro ejemplo, o Foursquare que permite agregar nuevos lugares de interés que nos parezcan pertinentes, por sólo mencionar algunos.

Estas plataformas y/o herramientas fueron muy útiles para el usuario generador de información. Sin embargo, la cantidad de información generada por éstos podría llegar a ser muy grande, e incluso repetitiva o contradictoria. Es debido a este tipo de situaciones que surge la necesidad de validar la información que se comparte, y la incógnita de cómo hacerlo. Para abordar este problema las plataformas actuales implementaron sus propios mecanismos de validación; en general éstos son ad hoc para cada tipo de aplicación. Por esto, hoy en día sitios como Wikipedia tienen una validación de información muy especializada para la calidad de los artículos que

contiene, así también se puede mencionar el caso de Slashdot, cuyo sistema de moderación y gobernación para la selección de artículos ha permitido mantener la calidad de información de su portal.

A pesar de la existencia de estos mecanismos, puede haber temas específicos para los cuales no se ha encontrado una forma de validación o no se ha probado una solución ya implementada en otros escenarios.

Por otro lado, el incremento de dispositivos disponibles para los usuarios y la facilidad de ingresar información por medio de éstos, ha dado lugar a un paradigma de computación conocido como sensado participativo (o *participatory sensing*, en inglés) [Goldman09]. Aquí los usuarios, con el apoyo de sensores embebidos en sus dispositivos móviles (como por ejemplo, cámaras fotográficas o de video, sensores de movimiento, y GPS, entre otros), ingresan información que es compartida con otras personas. Algunas aplicaciones típicas del sensado participativo es el estado del tráfico vehicular (como el caso de Waze), o el monitoreo de la contaminación de su ciudad (como la herramienta PEIR), entre otros. Aunque este nuevo paradigma es muy interesante y permite sensar una cierta variable usando a una colección de usuarios, nos encontramos con el problema de validar la información ingresada por estos usuarios, pues muchas veces ésta puede ser falsa o contar con poco grado de certeza.

Si tomamos en cuenta la existencia de temas específicos para los cuales no se ha probado aún una forma de validación, además de que el sensado participativo puede tener problemas de fiabilidad de la información sensada/generada, es claro que entre todos estos contextos, nos encontramos con un problema asociado a la confiabilidad de la información georreferenciada. Típicamente esta información, o metadata asociada (por ejemplo, los *tags* o descripciones del lugar) es de calidad dudosa, puesto que no se cuenta con mecanismos de validación adecuados, que sean públicamente conocidos.

Tal como es mencionado en [Gilbert10], los sistemas participativos tienen ciertas vulnerabilidades, entre las que destaca la integridad y veracidad de los datos. Un sistema participativo donde existen productores de contenido, así como consumidores de esto, es muy abierto, lo cual lo hace muy vulnerable ante usuarios maliciosos. En casos como Foursquare, donde se permite el libre ingreso de información, la validación, confirmación o

eliminación de ésta la llevan a cabo los usuarios de la misma, indicando si es o no correcta. Aunque este mecanismo aparenta funcionar bien, puede ser lento y requiere un número importante de detractores para que la información falsa sea eliminada del sistema. Así también, ciertos ítems, como los comentarios, no poseen mecanismo de validación, lo cual puede conllevar a generar una mala imagen de un sitio utilizando un grupo de usuarios maliciosos. Además de considerar la cantidad de usuarios sensando y agregando lugares a esta plataforma, este mecanismo puede verse inerme ante usuarios que decidan agregar información poco confiable, o en algunos casos falsa. Hoy en día, donde los sistemas móviles tienen gran auge y su disponibilidad se ha incrementado considerablemente, el contar con métodos efectivos para validar la información georreferenciada se vuelve mandatorio.

1.1 Hipótesis del trabajo

La hipótesis de trabajo planteada para esta tesis indica que *una heurística basada en crowdsourcing y computación social permite validar información georreferenciada, así como su información relacionada (tags), con un alto grado de confiabilidad*. La definición de esta hipótesis se basó en el hecho de que el uso de crowdsourcing en validación de información se ha estado usando con éxito en diversos ámbitos, por lo tanto el desafío radicó en encontrar buenas adaptaciones para poder validar información georreferenciada.

1.2 Objetivos de la tesis

El objetivo de este trabajo de tesis radica en definir una heurística que permita validar información georreferenciada utilizando conceptos de *crowdsourcing* y computación social. Esta definición fue implementada en un sistema móvil, destinado a validar la información georreferenciada en un escenario real, generando información confiable y que, con un alto grado de fiabilidad, puede ser replicado en diferentes escenarios. Los objetivos específicos definidos para alcanzar el objetivo general son los siguientes:

- Seleccionar los conceptos de *crowdsourcing*, ad hoc para el problema a abordar, a ser utilizados en la estrategia de validación propuesta.
- Definir componentes de computación social que permitan evitar o reducir la incertidumbre sobre la validez de información recolectada.
- Definir la heurística de validación de información georreferenciada.
- Diseñar e implementar una aplicación móvil que utilice la estrategia planteada.

- Determinar el nivel de confiabilidad de los datos resultantes del uso del sistema móvil.

1.3 Metodología

La metodología que se utilizó en la propuesta fue un desarrollo evolutivo, donde la heurística fue siendo ajustada a nivel de configuración, de acuerdo a los resultados de diversas pruebas de concepto que se llevaron a cabo. Para esto se realizó el primer diseño de la heurística de validación de información georreferenciada, así como la implementación de ésta en una API (Application Programming Interface). Usando esta API se hizo pruebas, mediante test unitarios, que simulaban diversas invocaciones a los servicios de la API. Estas invocaciones eran similares a las que realizarían los usuarios reales, lo cual permitió evaluar y ajustar el diseño de la heurística en forma evolutiva.

Considerando los cambios realizados a la solución propuesta, se desarrolló la versión final de la API. Además, se desarrolló una aplicación que trabaja en un entorno móvil, que utiliza soporte de geolocalización, y que hace uso de esta API. Los tests antes diseñados fueron nuevamente ejecutados para verificar si la API cumplía todo lo indicado por el diseño de la heurística.

Las pruebas finales se realizaron sobre una base de datos que contenía información ya validada por los dueños de ésta. La aplicación móvil se instaló en un conjunto de 34 usuarios de prueba. Para la validación de la heurística, el factor más importante que se llegó a medir con las pruebas realizadas sobre estos usuarios, fue la confiabilidad de la información indicada como válida por la heurística. El proceso de validación se realizó de forma manual debido al carácter de la misma. Como factor secundario, pero también válido a tomar en cuenta, se consideró la cantidad de información que, siendo válida, fue tomada como inválida por la heurística propuesta; es decir los falsos negativos.

Entre otros factores a medir en las pruebas, se consideraron los siguientes: número de usuarios necesarios para validar un lugar, cantidad de lugares que no se llegaron a validar, lugares inválidos que fueron tomados como válidos, y evolución de la cantidad de información, entre otros.

1.4 Estructura de la tesis

A continuación se describe brevemente la estructura de este documento de tesis. El capítulo 2 presenta el marco teórico y los trabajos relacionados

con "Participatory Sensing", "Estrategias de Validación de Información", "Context-Awareness" y "Gamification". El capítulo 3 describe de forma detallada la heurística propuesta para validar la información georreferenciada generada por un grupo de personas. El capítulo 4 expone, de forma general, el diseño de la herramienta desarrollada para la validación de la heurística. El capítulo 5 presenta los detalles del experimento realizado con la herramienta, así como los datos obtenidos en la validación de la misma. Además se realiza un breve análisis sobre estos resultados. El capítulo 6 muestra las conclusiones del trabajo realizado, así como un análisis de un posible trabajo a futuro.

2. Marco teórico

Este capítulo presenta los principales conceptos involucrados en la tesis, y los trabajos relacionados en las áreas correspondientes. Inicialmente se define el *crowdsourcing* y su uso en la validación de información. Después se definen conceptos generales como *context-awareness*, *gamification* y *sensado participativo (participatory sensing)*. Posteriormente se definen conceptos específicos como Captcha y Recaptcha, así como su utilidad, finalmente la información sobre los sistemas de recomendación que se considera en la tesis y mencionar un poco sobre los sistemas de recomendación en sí y el por qué es importante la heurística planteada en este tipo de sistemas.

2.1 Crowdsourcing y validación

El *crowdsourcing*, de acuerdo a [Yan09], es la utilización del poder de las multitudes para facilitar la realización de las tareas que pueden llegar a ser muy costosas en dinero o tiempo con los métodos tradicionales. Éste ha sido usado desde hace años para la realización de pequeñas tareas a cambio de dinero en sitios Web, como es el caso de Amazon Mechanical Turk [Turk14] o CrowdFlower [Crowdflower14]. Estas tareas pueden llegar a ser traducciones, tagging de imágenes, responder encuestas, entre otros.

Con el tiempo aparecieron soluciones donde se usaba el *crowdsourcing* para agregar contenidos (como es el caso de Wikipedia [Wikipedia14]), para la agregación de información geoespacial (como en OpenStreetMap [Streetmap14]), o como parte de los buscadores sociales donde se les paga a las personas por responder preguntas de los usuarios (como en ChaCha [Chacha14]). A pesar de los diversos usos del crowdsourcing, no se conocen públicamente soluciones que, en forma integral, permitan agregar nuevos lugares (por ejemplo tiendas, restaurantes, licorerías, entre otros), validar su existencia y administrar los *tags* generados en torno a éstos.

Diversas investigaciones se han hecho sobre este tema, incluso en algunos casos se han desarrollado plataformas móviles para conectarse a Amazon Mechanical Turk y a ChaCha, como es el caso de mCrowd [Yan09]. Sin embargo, estos trabajos apuntan a aprovechar el uso de *smartphones* para tener las tareas del usuario a la mano, y también para mejorar la participación de los mismos en escenarios móviles. En otras palabras, esas

soluciones no sacan provecho de los componentes que hoy en día trae un *smartphone*, como por ejemplo la cámara fotográfica o el GPS.

Como en toda plataforma donde se agrega información, existen diversos tipos de usuarios. Muchas veces esta tipificación de usuarios está relacionada al trabajo que éstos realizan, tal como se menciona en [Kazai11]. Debido a esto, el tema de control de calidad de la información es uno de los retos más importantes para sistemas basados en Crowdsourcing [Doan11]. Así lo toma en cuenta el trabajo de Zhu et al. [Zhu12], donde se propone un control de calidad basado en características de las tareas, y donde el principal factor a tomar en cuenta es el tiempo que le toma al usuario resolver una tarea. Sin embargo, estos investigadores mencionan que el factor de tiempo hace a su solución sensible a usuarios maliciosos, quienes se pueden tomar intencionalmente mucho tiempo en realizar una tarea y así atentar contra dicha solución. Este modelo sólo soporta tareas que demoran un tiempo considerable, no tomando en cuenta tareas muy puntuales y específicas, como la validación de la información de un lugar (o punto de interés) específico.

Tal como se ha mencionado anteriormente, la información georreferenciada puede ser ingresada por varios tipos de usuarios, entre los cuales pueden encontrarse usuarios maliciosos que buscan usualmente comprometer la fidelidad de la información. El trabajo de Mashhadi et al. [Mashhadi11] muestra a grandes rasgos el tipo de validación de la calidad de la información, que se puede obtener a través de una plataforma de *crowdsourcing*. En este tipo de validación se toma en cuenta los patrones de movilidad de los usuarios que ingresan información al sistema.

En dichas soluciones la opinión del usuario tiene gran validez, siempre que la ubicación de éste esté dentro del patrón de movilidad registrado para dicho usuario, no tomando en cuenta la dispersión de movilidad que éste pueda tener, o algún otro criterio para llevar a cabo la validación de la información ingresada. Por lo tanto, si el usuario (respetando su patrón de movilidad) ingresa información falsa de un cierto lugar, dicha información será considerada (a priori) como verdadera por adherir al patrón de movilidad del usuario.

Una de las mejores formas de control de calidad de información en sistemas de *crowdsourcing* fue diseñado para el sistema Recaptcha [Recaptcha14], el cual se explicará en mayor detalle en 2.5. La validación

involucra dos piezas de información, una conocida y otra desconocida. Debido a que el usuario no sabe cuál es la pieza de información conocida, éste se ve obligado a validar ambas. De esta forma el sistema tiene una validación inmediata de la calidad de información otorgada por el usuario. El caso de Amazon Mechanical Turk [Turk14] es muy similar, donde en lugar de palabras se colocan tareas con una o más respuestas válidas. Este tipo de control de calidad no ha sido utilizado en *crowdsourcing* relacionado con la validación de información sobre lugares físicos, y es uno de los puntos que se tomaron en cuenta en el desarrollo de este trabajo de tesis.

2.2 Context-awareness

Desde el punto de vista de las ciencias de la computación, *context-awareness* hace referencia tanto a la sensibilidad al contexto de los dispositivos, así como la reacción de estos de acuerdo al ambiente/contexto donde se encuentren y a los eventos que reciban. Por ejemplo, cuando una casa a través de los dispositivos que sensan calor, humedad y humo detecta un incendio, inmediatamente puede alertar al usuario del evento con una llamada telefónica, y a su vez también puede alertar a los organismos pertinentes, es decir tiene una reacción de acuerdo al contexto.

Según Charit Perera et al. [Perera13], esta técnica existe y ha sido utilizada desde principios de 1990s y se ha hecho más popular con la introducción del término de "computación ubicua". De acuerdo a Abowd et al. [Abowd99], "un sistema es sensible al contexto (context-aware) si éste utiliza el contexto para proveer información y/o servicios relevantes al usuario, donde la relevancia depende del usuario y del sistema".

Entonces, la percepción del contexto debe permitir abstraer la complejidad de la Internet de las cosas (*Internet of things*), donde cada elemento tiene múltiples de sensores, consolidando la información relevante de contexto para las aplicaciones, y que en base a ésta se puedan tomar acciones consecuentes según las necesidades del usuario. De acuerdo a Perera et al. [Perera13], estas acciones se pueden clasificar en tres categorías:

- **Presentación:** Cuando la aplicación decide qué debe ver el usuario. Por ejemplo, cuando un usuario entra a un supermercado y toma su *smartphone*, lo que él/ella desea ver es su listado de compras. Una aplicación móvil sensible al contexto podría conectarse, por ejemplo, al

refrigerador inteligente que posee el usuario en su casa, para que éste le indique que alimentos se necesitan [Sundmaecker10].

- **Ejecución:** Auto-ejecución de servicios, esto es muy utilizado en la implementación de ambientes inteligentes. Por ejemplo, cuando un usuario empieza a conducir el auto hacia su casa, la aplicación de IoT (*Internet of things*) instalada en casa podría querer encender el aire acondicionado, para que la casa esté temperada cuando arribe el usuario; o bien, encender la máquina de café y para que esté caliente para cuando el usuario llegue a su casa. Estas acciones deben de ser realizadas en forma automática, basadas en información de contexto y en reglas de decisión [Perera13].
- **Etiquetamiento:** Al producirse un gran volumen de información desde los diferentes sensores de los dispositivos, es deber de la aplicación sensible al contexto mezclar todas estas entradas y dar una información contextual más simple, indicando el estado de contexto.

Según su interacción con los usuarios, estas aplicaciones pueden clasificarse en tres tipos diferentes:

- **Personalización:** Cuando los sistemas siguen las preferencias del usuario, o cuando el usuario las configura manualmente. Por ejemplo, los usuarios pueden indicar la temperatura de preferencia en un ambiente de una casa, y luego un sistema inteligente se encarga de mantener esa temperatura a nivel de todos los dormitorios y espacios de la misma [Perera13].
- **Context-awareness pasivo:** Cuando el sistema constantemente monitoriza el ambiente, para ofrecer acciones al usuario. Por ejemplo, cuando un usuario entra a un supermercado y el celular del usuario empieza a notificarle que hay una lista de productos con descuento que podría interesarle [Perera13].
- **Context-awareness activo:** Cuando el sistema constantemente monitoriza el ambiente y toma acción ante determinados eventos. Por ejemplo, si los detectores de humo y temperatura detectan un incendio en algún ambiente de una casa inteligente, el sistema de ésta debería automáticamente notificar a los bomberos así como al dueño de la casa

de la forma apropiada, como por ejemplo, mediante un mensaje de voz y/o un SMS [Perera13].

Como ya se mencionó, el uso de *context-awareness* depende intrínsecamente de la capacidad de los dispositivos en disponibilizar la data y la generación de *middlewares*, *frameworks* y APIs, que permitan abstraer a las aplicaciones de la heterogeneidad de las “cosas” que existen en Internet y que están relacionadas con el usuario. La siguiente figura indica una arquitectura básica de un ambiente inteligente.

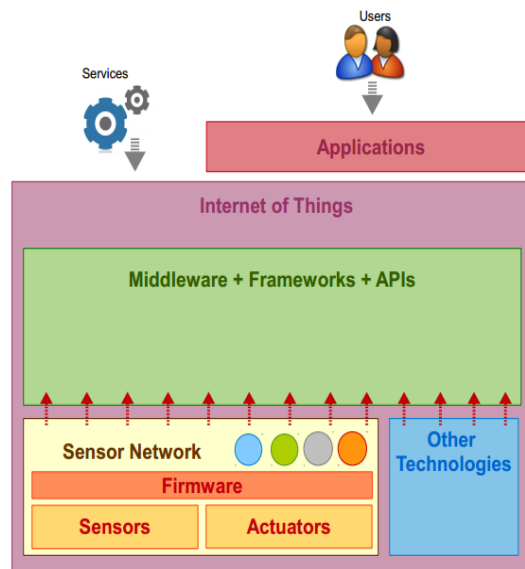


Figura 1: Arquitectura de un sistema sensible al contexto

Así es como los datos geográficos del GPS, que inicialmente sólo constituyen información de localización, pueden ser complementados por información contextual; por ejemplo, acerca de la ubicación de un supermercado, de una estación de policía o incluso del hogar del usuario. Esto permite a las aplicaciones, discriminar los servicios a brindar o el tipo de notificaciones a entregar dependiendo de lo que está sucediendo en el entorno del usuario.

El punto principal a recalcar en la Figura 1 es el Middleware + Frameworks + APIs, además del componente que implementa la “*Internet of things*”, a partir del cual se extraen los datos del contexto. Lo que muchos sistemas sensibles al contexto realizan es basar sus demás capas sobre unos datos que ya existen, es decir, sobre las bases de datos de diversos sitios de Internet o APIs públicas. Por lo cual la confianza de estos sistemas únicamente está basada en esos datos.

Existe gran cantidad de aplicaciones móviles sensibles al contexto, que se basan en la información de páginas Webs, APIs públicas y demás. Así también existen otras aplicaciones híbridas, que no sólo se basan en información que encuentran en componentes que implementan “*Internet of things*”, sino que además agregan a su base de datos, información que les otorga el usuario. Sin embargo, esta información nueva, así como la información consumida desde otros recursos no es validada. Es en este punto donde una heurística de validación de información puede apoyar a este tipo de aplicaciones.

2.3 Participatory sensing

Tal como indica Goldman et al. [Goldman09], *participatory sensing* (o sentido participativo) se define como comunidades contribuyendo con información sensada, con el objetivo de construir conocimiento. En sí, este conocimiento que se almacena es utilizado para el análisis e interpretación de estos mismos.

Participatory sensing hace bastante énfasis en involucrar a comunidades de usuarios en el proceso de sentir y documentar dónde ellos viven, trabajan o se divierten. El tipo de información recolectada y el cómo va ser usada, es determinada mediante un organismo central, o en algunos casos, lo decide la misma comunidad. La evolución del hardware de los *smartphones*, así como el incremento de uso de estos, ha permitido realizar con más facilidad experimentos de sentido participativo.

Según Goldman et al. [Goldman09], los procesos de sentido participativo involucran los siguientes pasos:

- *Coordinación*: Involucra recolectar, coordinar y guiar a la comunidad de usuarios.
- *Captura*: Se refiere a la adquisición de datos desde dispositivos (o sensores) externos.
- *Transferencia*: Haciendo uso de los *smartphones* y las redes inalámbricas se puede utilizar software para subir los datos recolectados a una base de datos central.
- *Almacenamiento*: La información recolectada suele almacenarse en servidores públicos o privados.

- *Acceso*: Se debería tener en claro el tipo de acceso a la información privada que está permitido a los usuarios que colaboran con el sensado.
- *Análisis*: Incluye una variedad de mecanismos para el análisis de los datos recolectados, para poder encontrar patrones a nivel de espacio y tiempo, o para realizar fusión de información.
- *Retroalimentación*: Es requerida cuando se desea automatizar eventos ante cambios que la aplicación detecte en el dispositivo a través del cual se está sensando.
- *Visualización*: Indica la data que es mostrada al usuario, la cual puede haber sido previamente analizada y procesada, por lo cual va de la mano con el paso de 'análisis'.

El sensado participativo es un concepto que tiene pocos años de investigación (aproximadamente desde el 2009), por lo cual tiene diversos puntos aún por trabajar. Entre estos pendientes se encuentra el de la confiabilidad de los datos sensados por los usuarios.

Tal como lo menciona Gilbert et al. [Gilbert10], los dispositivos que usualmente utilizan los usuarios para realizar esta labor, los cuales normalmente son *smartphones*, no son confiables debido a que la data sensada podría verse manipulada previamente a la subida de los datos sensados. Gilbert et al. [Gilbert10] y también Dua et al. [Dua09] usan el módulo TPM (Trusted Platform Module) para la validación de los datos generados por el hardware y/o software. Lamentablemente, realizar la validación del sensado de datos usando un hardware adicional podría ser un poco engorroso, dependiendo de la cantidad de usuarios de la comunidad, así como del tipo de experimento a realizar.

Aunque no está mal el uso de un hardware adicional para la validación de los datos sensados para ciertos tipos de experimentos, debido a la naturaleza del experimento a realizar en esta tesis, esa estrategia se ve como poco factible. Por otra parte, este trabajo busca mostrar que hay casos de *participatory sensing*, donde se puede hacer uso de *crowdsourcing* y otros elementos para la validación de los datos, donde no hay necesidad del uso adicional de hardware.

2.4 Gamification

Gamification [Donovan13] es el empleo de dinámicas de juego e incentivos a los jugadores en entornos no lúdicos. Cuando esta estrategia está diseñada correctamente, *gamification* aumenta la participación y fomenta ciertas conductas entre los usuarios.

En países como España, el concepto se ha utilizado como una herramienta de marketing. Según Game Marketing [GameMarketing12], el 87.5% de los profesionales de marketing y de tecnología conocen o han utilizado este instrumento. Igualmente de auspiciosos son los números que indican que el 82.03% de esos profesionales cree que esta herramienta permite resolver problemas de captación y fidelización de clientes.

Considerando que el 55% de ellos juega con dispositivos móviles, el 65% son jugadores sociales y que más del 70% de los altos ejecutivos juegan videojuegos [BBVA12], es natural que los videojuegos sean parte común de la vida cotidiana de las personas. Por lo tanto, es también normal mantener altas expectativas en la incursión transversal de *gamification* en la provisión de servicios a terceros, y en el mercado en general.

En Chile la PDI ha lanzado un juego móvil para darse a conocer [Publi14] y fomentar el ingreso de profesionales a su institución, Starbucks en estados unidos permite una experiencia compuesta de "Onboarding, rewards, levels, customization and gifting". Incluso se ha utilizado esta herramienta para motivar la realización de actividades cotidianas. Un ejemplo de aplicación que hace esto es "Zombies, RUN!", la cual utilizando el GPS del móvil provee al usuario la experiencia de ser perseguido por una horda de zombies, de la que debe liberarse corriendo "diariamente" [Zombie14].

Meloni [MELONI11] indica que *gamification* puede ayudar a las empresas a aumentar la participación del usuario (47%), la fidelización de la marca (22%), el reconocimiento de la marca (15%). También ayuda a motivar (9%), y a formar a los empleados (7%).

Por otra parte, el reparto de clientes interesados en usar *gamification* según industria es el siguiente: entretenimiento (42%), editoriales (18%), bienes de consumo (15%), salud/bienestar (10%), finanzas (5%), comercio minorista (5%), educación (3%), telecomunicaciones (1%). Siendo el crecimiento del año 2011 un 155% y 2012 un 197% es una tendencia al alza.

Para entender los aspectos psicológicos del comportamiento humano relacionado con *gamification*, Gears [Gears13] se basa en dos teorías: la teoría de los 16 deseos básicos [Reiss00] y la teoría de la autodeterminación

[Deci85]. La primera describe los 16 deseos básicos del ser humano: orden, poder, independencia, curiosidad, aceptación, seguridad, idealismo, honor, contacto social, familia, estatus, venganza, romance, hambre, actividad física y tranquilidad. La segunda explica el contexto en que los seres humanos pueden ser proactivos y comprometidos, o alternativamente, pasivos y alineados [Ryan00].

A partir de las dos teorías anteriores, se realizó un estudio en que se busca entender los factores que motivan a los empleados a tener un comportamiento participativo [Gears11]. En dicho estudio se encontraron fuertes deseos de poder, independencia, idealismo y curiosidad por incitar comportamientos de compromiso tales como agregar, modificar y comentar contenido. Así también se identificaron influencias contextuales, las cuales fueron asociadas a la participación y a la no-participación de la gente [Gears13]. Participación es un tema crítico, que buscan potenciar las empresas que brindan servicios o venden productos a terceros.

En el trabajo de Gears [Gears11] se propone un *role-motivation-interaction framework* que permite discriminar cuáles son los deseos y motivaciones para cada rol de usuario. Además, permite generar una experiencia más positiva y comprometida de los usuarios.

El banco español BBVA adopta un fundamento similar, para crear un propio esquema de cómo satisfacer sus necesidades básicas [BBVA12]. Este banco va un paso más allá, y especificando formas de abordar cada deseo, analizando la relación entre las mecánicas de juego y los deseos humanos (Figura 2).

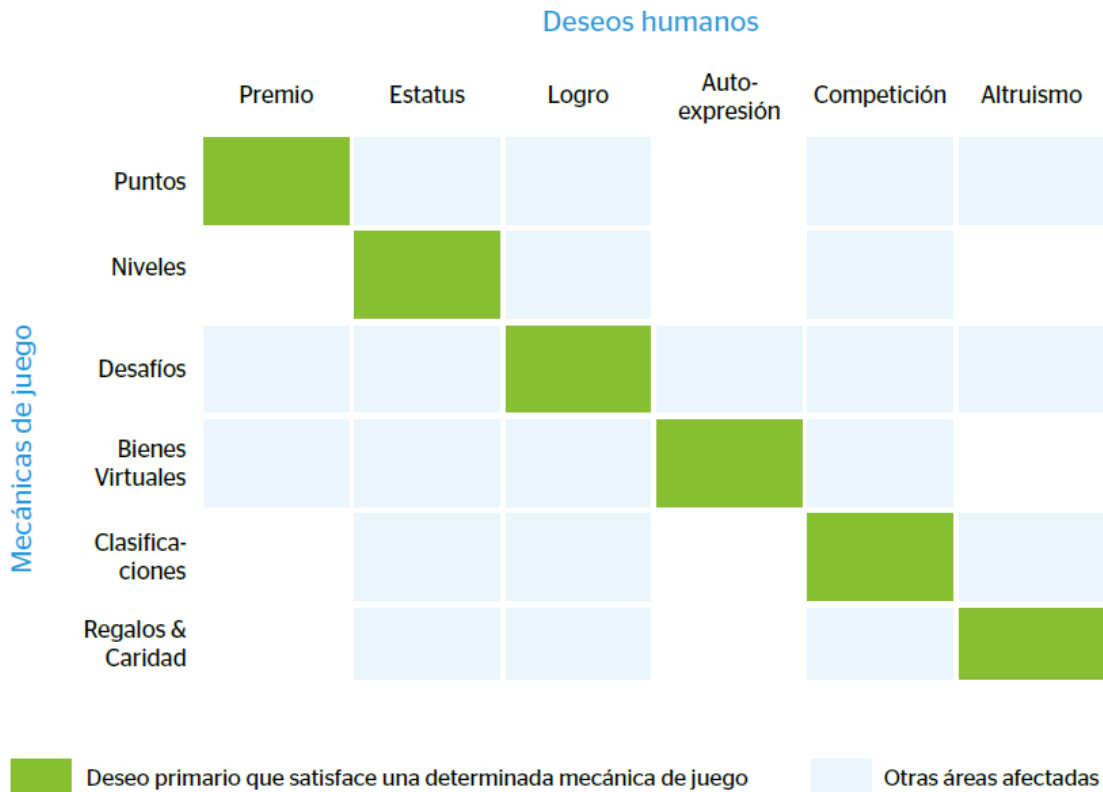


Figura 2: Mapa entre deseos humanos y mecánicas de juego

Entre todas estas investigaciones se ha podido ver el uso de *gamification*, pero además de esto, existe un estudio realizado por IBM titulado "Removing Gamification from an Enterprise SNS" [Thom12] donde se pudo demostrar que al remover los componentes de *gamification* de una red social, la participación en ésta disminuyó en casi un 50%.

Tal como se puede ver en estas investigaciones y estudios previos, se resalta el uso de *gamification* para fomentar comportamientos particulares, así como para animar a las personas a realizar tareas que normalmente podrían ser consideradas aburridas. Por lo tanto, se consideró el uso de algunos componentes de *gamification* en la definición de la heurística de validación propuesta en esta tesis, ya que la tarea de validación podría resultar relativamente aburrida para algunos usuarios. Este enfoque también podría ayudar a orientar el comportamiento del usuario hacia el deseado según la heurística.

2.5 Captcha y Recaptcha

Captcha (Completely Automated Public Turing test to tell Computers and Humans Apart) es una prueba de Turing completamente automática y

pública para diferenciar computadoras de humanos. De acuerdo a [Ahn08] es una medida de seguridad generalizada en la *World Wide Web* para prevenir a servicios online del uso excesivo de parte de programas automatizados. Esto se realiza normalmente preguntando a usuarios humanos realizar tareas que los computadores no serían capaces de realizar, como por ejemplo el descifrar caracteres distorsionados. El realizar esto previene el abuso de compras de tickets para un concierto, para después revenderlos a precios mayores, así también creación excesiva de mails en Yahoo Mail o Gmail, y de esta forma evitar el envío de spam mediante estos.

De acuerdo a [Ahn08] en el mundo se escriben cerca de 100 millones de CAPTCHAs al día, lo cual resulta en cientos de miles de horas por día gastados únicamente en esto. Al encontrar el tiempo y esfuerzo gastado al día por los CAPTCHAs surge reCAPTCHA, el cual reutiliza la técnica de CAPTCHA para ayudar en la digitalización de textos. En el proyecto de Google Books se escanean normalmente los libros, este escaneo es una imagen, la cual mediante OCR es transformada a texto, esta digitalización sirve para poder indexar y hacer búsquedas en los libros. El problema viene de la imperfección de los OCR en poder descifrar todos estos textos que se encuentran como imagen, calculando cerca de 20% de textos no reconocidos por estos.

Es acá donde ingresa reCAPTCHA y aprovecha el poder de la gente, categorizando las palabras en "conocidas" y "desconocidas", utiliza las primeras como "palabras control" que son aquellas palabras que reCAPTCHA sabe lo que dicen de acuerdo a los OCR utilizados. Haciendo uso combinado de las "palabras control" y las palabras "desconocidas" es que reCAPTCHA muestra al usuario un reto con ambas juntas en orden aleatorio, por lo cual el usuario debe de resolver ambas palabras sin tener conocimiento de cuál es la palabra "desconocida" ni la "palabra control", de esta forma se valida que el usuario resolvió correctamente el reto al acertar la "palabra control", mientras que la resolución de la palabra "desconocida" se le agrega un voto en el sistema interno de reCAPTCHA. La resolución que cuenta con mayor número de votos finalmente se toma como la correcta transcripción de la palabra "desconocida".

El uso de esta técnica, de acuerdo a [Ahn08], ha obtenido muy buenos resultado en un año donde se han resuelto más de 1200 millones de CAPTCHAs resueltos y ha colaborado con el descifrado de cerca de 440 millones de palabras, lo que es equivalente a 17.600 libros manualmente

transcritos. El éxito de esta técnica se puede ver claramente en los números obtenidos en sólo un año y en su continuo uso hasta ahora. Su uso ha cubierto incluso el área de mapas de Google, así también se propone el uso de éste para agregar *tags* a imágenes. Sin embargo, aún no ha sido usado en el área de validación de ciertos tipos de entidades como localización de lugares, comentarios, entre otros.

2.6 Información social para los sistemas de recomendación

Con el crecimiento exponencial de información en la Web 2.0 aparecen las interacciones tanto con el contenido como entre los usuarios. Todo tipo de interacción como: *rating*, *tagging*, *posting*, *liking*, entre otros; Son considerados información social. De acuerdo a [Bobadilla13] la información se utiliza con 3 objetivos:

1. Proponer y generar nuevos sistemas de recomendación.
2. Dilucidar las relaciones más importantes entre la información social y el proceso colaborativo.
3. La mejora de la calidad de las recomendaciones y predicciones.

De acuerdo al estudio de [Victor11] se ha podido utilizar información implícita y explícita de un usuario en una red social, es decir sus interacciones, con el objetivo de generar medidas que permitan verificar la confiabilidad y reputación de un usuario.

2.7 Sistemas de recomendación

Se define un sistema de recomendación [Burke02] como aquel que produce, como salida, recomendaciones personalizadas que tienen el efecto de guiar al usuario de una forma personalizada a productos interesantes o útiles entre una gran cantidad de productos disponibles. Esta definición no implica que tenga únicamente un dominio comercial, pues su aplicación se puede hacer en diferentes contextos como se demuestra en [Gonzalez12].

Los enfoques existentes basan a los sistemas de recomendación en ítems, usuarios e interacciones entre estos. De estos sistemas de recomendación se desprende una categoría llamada "Sistemas de recomendación sensibles al contexto (*context-aware*)", la cual se basa en [Adomavicius05], quien propuso el agregar la información contextual para dar las recomendaciones.

Esto se basa en el hecho de que no es lo mismo recomendar una canción estando en el trabajo, que estando en una fiesta, o estando feliz o estando triste.

Debido a esta extensión de los sistemas de recomendación y diversos dominios sobre los cuales estos trabajan, y a las nuevas interacciones que pueden tener los usuarios en estos dominios trabajados, es que surge la propuesta de *framework* de [Vasquez14], el cual permite la representación y construcción de sistemas de recomendación, manteniendo de esta forma estos sistemas bajo un esquema común de representación y construcción. Este *framework* es uno de los pocos que permite extensibilidad, de tal forma que pueda agregarse un componente adicional como el validar la información relacionada a los ítems que el sistema va a recomendar a los usuarios, así como la credibilidad de estos por su definición del modelo de usuario.

2.8 Aplicaciones basadas en crowdsourcing

En el escenario actual existen diversas aplicaciones que permiten a sus usuarios el agregado de información. Entre ellas encontramos muchas que permiten agregar información, sin ser ésta ad-hoc a lugares geolocalizados. Para aquellas aplicaciones cuyo objetivo secundario es el agregado de lugares por parte de los usuarios, no se brinda una validación de estos, o bien se proveen mecanismos de validación muy lentos. A continuación se muestran las aplicaciones disponibles en el mercado, cuyo objetivo es similar al que se persigue en esta tesis.

2.8.1 Wikipedia

Wikipedia, la cual es considerada la mayor enciclopedia libre, políglota y editada de forma colaborativa, cuenta con más de 37 millones de artículos que fueron creados colaborativamente. Esta enciclopedia cuenta con artículos de propósito general, donde uno puede encontrar información sobre historia, personajes famosos, o matemática, entre otros. Esta información es validada en forma colaborativa, lo cual le ha permitido crecer en forma sostenida y mantener un importante nivel de confiabilidad a lo largo de estos años. Este proceso de validación no es ad-hoc a los lugares geolocalizados, ni toma en cuenta la posición de sus usuarios, sino que más bien es una parte de la comunidad la encargada de determinar si la información ingresada es verdadera. Este mecanismo puede ser útil para validar información general, pero no es tan útil para validar geolocalizada.

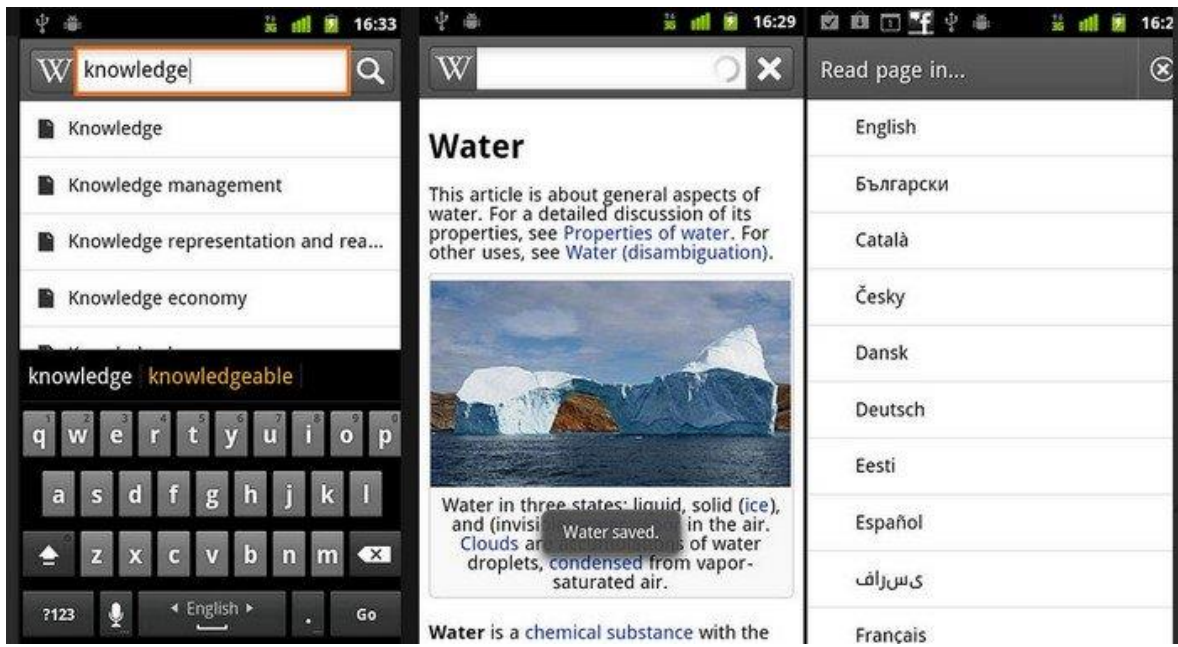


Figura 3: Versión de la Wikipedia para dispositivos móviles

2.8.2 Foursquare

Foursquare es una aplicación basada en geolocalización que cuenta con una base de datos de más de 30 millones de usuarios, y cerca de 32 millones de lugares registrados. Sin embargo, hay que tomar en cuenta que al no contar con un sistema rápido de validación de los lugares registrados, éstos perfectamente pueden ser desde falsos, estar mal posicionados, e incluso ser de poca utilidad para algunos usuarios (ejemplo: “mi casa”, o “sitio donde estuve”, entre otros).

A pesar de que ésta es una herramienta ad-hoc para registrar geolocalizadamente lugares sobre un mapa, y que además toma en cuenta la posición de los usuarios para agregar lugares a su base de datos, estos lugares son validados de forma manual, o incluso pueden no ser validados por mucho tiempo, y aun así permanecer públicos. Esto se pudo corroborar al usar la aplicación para agregar lugares que realmente no existían. El hecho de no contar con una validación usando crowdsourcing, genera este tipo de situaciones.

Entrando más a detalle a cuánto afecta que los lugares no existan o que los reviews sean falsos, mencionaremos algunas formas como un usuario llega a decidirse ir a un lugar:

- *Check-ins* de amigos.- Al ver los *reviews* o visitas que hacen sus "amigos" a diversos lugares, atrae la atención del usuario y decide ir a éste.
- *Reviews* de un lugar.- El usuario puede revisar los *reviews* que tiene un lugar cercano y debido a estos decidir en ir o no a éste.
- Búsqueda de un lugar cercano.- Existen ocasiones en que el usuario simplemente decide buscar un lugar cercano donde pueda realizar una actividad, por ejemplo, comer, tomar un café, entre otros.

Tomando en cuenta estos dos últimos, uno puede encontrar *reviews* de diverso tipo en un lugar, útiles o inútiles, pero estos no tienen una forma de denunciarlos, o mostrar disconformidad, en caso sean irreales o solo traten de generar una mala imagen al local. Un usuario puede agregar un *review*, llenarlo de *likes*, con un grupo de usuarios maliciosos, y de esta forma subir la prioridad de éste en el listado, lo cual demuestra lo fácil que puede ser generar una mala imagen a un competidor comercial o, simplemente, hablar mal de un lugar específico.

Respecto al tercer punto, algunos usuarios suelen buscar el local más cercano de una cadena conocida por ellos, por ejemplo McDonalds, e ir a éste sin importarles los *reviews* que éste tenga. Pero puede ser que cuando vayan a la dirección indicada por Foursquare, no se encuentren con el local que buscaban. La eliminación de esta información toma tiempo en dar efecto, lo cual podría afectar a más usuarios.

Esta apertura de creación de la plataforma de Foursquare es la que genera mucha información, pero también la expone ante posibles usuarios malintencionados.

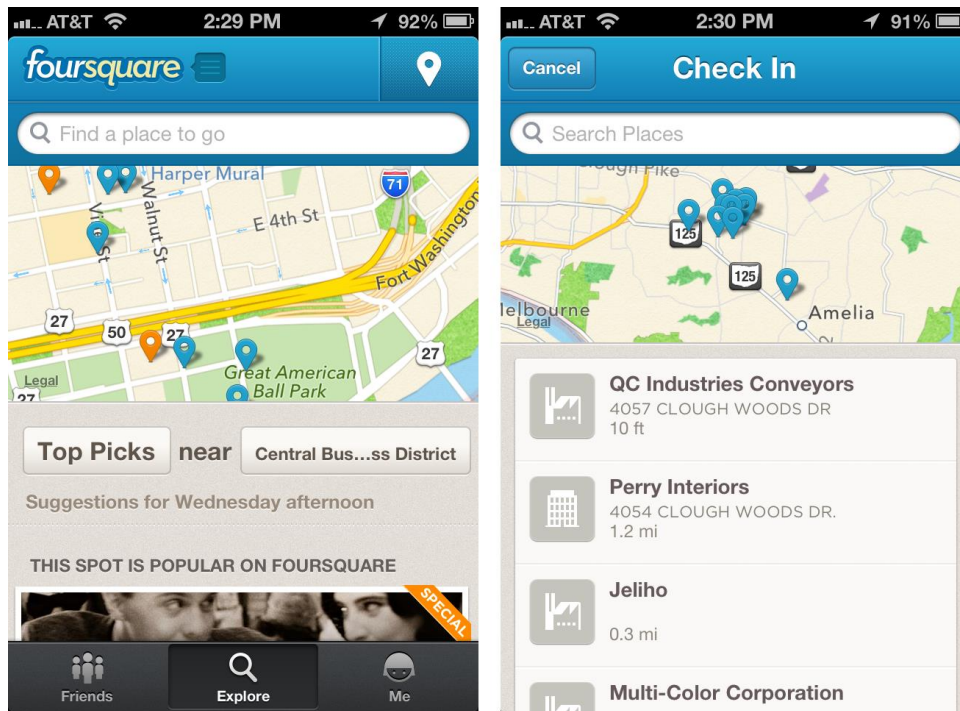


Figura 4: Aplicación de Foursquare para iOS

2.8.3 Facebook places

Facebook Places nació como una competencia para Foursquare, orientado netamente a lo mismo, pero teniendo por detrás la enorme cantidad de usuarios de Facebook. Este servicio fue integrado a Facebook después de agosto del 2011, como una característica adicional debido a su falta de uso. A pesar de todo esto cuenta con las mismas características de Foursquare, pero tiene un menor número de lugares registrados.

Facebook Places tiene una característica adicional a Foursquare, que es el poder registrar los lugares incorrectos de forma más explícita. Su orientación principal es la de hacer *check-in*, sin incluir componentes de *gamification* que les sirva de estimulación a los usuarios para el agregado de lugares. Por lo tanto, este servicio cuenta con una forma de validación basada en crowdsourcing, la cual es poco usada debido a su falta de visibilidad.



Figura 5: Aplicación de Facebook para Android

Si bien las tres herramientas presentadas son las más relevantes para el ingreso de información geolocalizada, en realidad existen cientos de éstas en las tiendas de aplicaciones de iOS y Android. Muchas de éstas incluyen *context-awareness* a diversos niveles, como de lugares físicos, eventos, y puntos de interés, entre otros. La mayoría de estas herramientas tienen limitaciones respecto a la fiabilidad de la información recolectada. Algunas de ellas usan bases de datos generadas de forma manual, y otras validan la información usando *scraping* para extraer datos de páginas web que publican eventos o recomiendan lugares. De este pool de herramientas, no se encontró ninguna que hiciera uso masivo del *crowdsourcing* para la generación y validación de información geolocalizada.

3. Definición de la heurística de validación

Para el diseño de la heurística, se ha planteado el problema de validación de información georreferenciada considerando dos aspectos: (1) el ingreso de la información por parte de los usuarios y (2) la participación de los usuarios para validar esta información ingresada. Tal como se puede ver, son dos aspectos que en sí se pueden resumir en la introducción y validación de la información. En ambas partes se debe abordar el problema motivacional del usuario para colaborar en esas tareas. Así también, se abordará el modelo de datos considerado en la heurística acotando el universo utilizado y mostrando su posible extensión.

3.1 Motivación

El agregado y la validación de la información son tareas que pueden ser consideradas como aburridas. Sin embargo, existen investigaciones previas que muestran dinámicas utilizadas para incentivar a los usuarios a realizar esas actividades. Es debido a esto, que se decidió utilizar técnicas de *gamification* para motivar al usuario a realizar el agregado y la validación de información. A partir de las técnicas de *gamification* analizadas, se decidió por la utilización de:

- **Experience Points (Puntos de Experiencia):** Los puntos de experiencia son el sistema de puntajes más importante de *gamification*, a pesar de no servir como un tipo de cambio de dinero real. Sirven para poder rankear a usuarios, así como para poder guiar a éstos en el comportamiento esperado en el sistema. Este tipo de puntos nunca llega a descender, pero sí pueden llegar a expirar para crear ciclos constantes de metas a obtener por el usuario. Para nuestro caso, haremos un reinicio semanal de los puntos para incentivar a los usuarios, semana tras semana, a interactuar con la aplicación [Zichermann11].
- **Badges (medallas):** Los *badges* son ítems que reflejan un estatus del usuario, y pueden ser asignados de forma virtual o física. Este tipo de ítems deben de ser visibles a los demás usuarios, caso contrario, su uso y valor sería muy limitado. El uso de los *badges* en nuestra heurística es para que el usuario acceda a un "premio" por las acciones realizadas en la aplicación implementada. De esta forma, generaremos una especie de comportamiento lúdico en el usuario, que hará que éste

busque conseguir más *badges*, lo cual generará más acciones positivas sobre la aplicación [Zichermann11].



Figura 6: Ejemplo de la representación visual de un badge

- **Ranking:** El propósito de un ranking o *leaderboard* va de la mano con los puntos de experiencia. Un ranking es usado para hacer comparación de los puntos de experiencia del usuario, en comparación con los demás usuarios. Esto sirve para ver en dónde se encuentra éste respecto a los demás, lo cual le genera motivación para encontrarse en una mejor posición en el futuro. Existen segmentaciones de rankings donde se puede segregar un ranking entre tu círculo social o con todos los usuarios. Muchas aplicaciones y juegos llegan a usar ambos rankings para que el usuario pueda tener un comportamiento competitivo, tanto entre sus amigos como con todos los demás. Sin embargo, debido a la cantidad de usuarios sobre la cual se va validar la heurística, se decidió en este caso usar un ranking que incluya a todos los usuarios [Zichermann11].

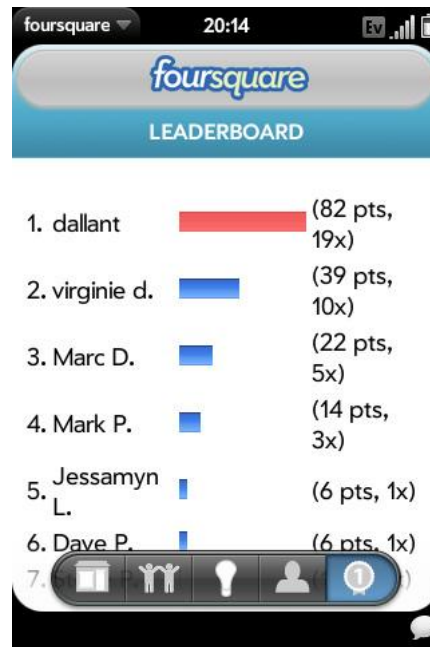


Figura 7: Ejemplo de la representación visual de un leaderboard

- Karma Points (Puntos de Karma):** En [Zichermann11] se define *karma points* como aquellos puntos que obtiene el usuario de acuerdo al comportamiento que éste tenga con el sistema. Mientras que se define, también en [Zichermann11], los *reputation points* como aquellos que indican cuán confiable es el usuario con lo que dice, hace u opina. Para esta tesis usaremos el término *karma points* como una fusión entre estos 2 términos definidos anteriormente. La elección del nombre *karma points* y no *reputation points* es debido a que la palabra "karma", proveniente de las religiones de origen indio, se define como la energía trascendente que se deriva de los actos de las personas, lo cual es análogo a la heurística que aplicamos, donde mientras mejor actitud tiene el usuario (dice la verdad, hace votaciones correctas, entre otros) mejor puntaje tendrá y más confiable serán las acciones que éste realice. Otra particularidad de estos puntos es que no son visibles para el usuario, lo cual de cierta forma permite ocultar el mecanismo de validación de *karma points* (KP).

3.2 Modelo de datos utilizado

Para la definición del modelo de datos a utilizar para la heurística primero se investigó sobre el modelamiento de las diferentes entidades en ambientes similares, como son los de los sistemas de recomendación. Tomando en

cuenta a estos, se elegirá el más cercano y se harán ciertas modificaciones que estén acorde a nuestro problema.

3.2.1 Trabajo relacionado

Los sistemas de recomendación, en su forma más básica, modelan las interacciones que existen entre los usuarios y los ítems de forma variada. Una propuesta de [Palomino12] nos da a conocer un modelo. A partir de eso, el autor basado en las interacciones entre los usuarios e ítems, lo cual se considera un evento, hace la siguiente definición:

$$e = \{(u, i, v) | s \in C, i \in S, v \text{ algún valor}\} \quad (1)$$

En donde C es el conjunto de usuarios, S el conjunto de ítems y el valor "v" es una preferencia del usuario como: un *like*, *tag*, comentario, entre otros. Por ejemplo, para una web como Youtube, el ítem puede corresponder a un video, mientras que el valor puede ser un *like* o *dislike*. Esta preferencia se asume como única para el usuario sobre el ítem en este modelo.

Así también existen modelos más evolucionados y detallados que éste, como el Synergy de [Tareen2010], quien toma en cuenta a los usuarios (U), los ítems (C), las posibles interacciones (I), el conjunto de eventos (E) y los valores que puede tomar el evento (V). Obteniendo un modelo así:

$$\begin{aligned} \mathbf{U} &= \{u_1, u_2, \dots, u_n\} \\ \mathbf{C} &= \{c_1, c_2, \dots, c_n\} \\ \mathbf{I} &= \{i_1, i_2, \dots, i_n\} \end{aligned} \quad (2)$$

$$\mathbf{E} = \{(s, p, o, t) | s \in U, p \in I, o \in C, t \text{ es el tiempo}\}$$

$$\mathbf{V} = \{(e, a) | e \in E, a \text{ es algún valor}\}$$

Este modelo ofrece una gran ventaja al ser flexible al momento de intentar describir los diversos tipos de interacciones que uno desee. Sin embargo, existen modelos más completos y extensos, como el propuesto por [Vasquez14] donde se representan las interacciones basadas para los sistemas de recomendación basados en la 3-Ontology, donde se puede ver mucho detalle en la definición de estos eventos, pues llega a ir muy profundo hasta tomar en cuenta los retratos (representaciones de comunidades) R, los mapas M, el conjunto de usuarios U, los ítems C y las interacciones I para la

definición de un evento, por lo tanto, este último se ve representado de la siguiente forma:

$$I = \{i_1, i_2, \dots, i_n\}$$

$$e = \{(u, c, i, v, \exists \text{onto}(co, l, t) | u \in U, c \in C, i \in I, co \in R, l \in M, t \text{ es el tiempo, } v \text{ es algún valor})\} \quad (3)$$

A pesar de que este último modelo permite ser más completo para los sistemas de recomendación, en esta heurística no se trata de llegar a tanto detalle, pero nuestra representación de interacciones e ítems se basará en estos modelos, presentados para facilitar el entendimiento de éste y su acotamiento.

3.2.2 Representación de eventos e ítems para la heurística

Tomando en cuenta el trabajo relacionado y usando la misma nomenclatura de interacción, procederemos a establecer nuestra propia representación de interacciones e ítems que participan en nuestra heurística.

3.2.2.1 Lugar

De acuerdo a nuestro caso, un lugar es considerado un espacio físico geolocalizado que puede ser visitado por los usuarios, posee ítems y atributos; además de ser el lugar donde ocurren las interacciones y además puede interactuar el usuario con éste. Teniendo definido $C = \{c_1, c_2, \dots, c_n\}$ como el conjunto de ítems y $A = \{attr^1_1, attr^1_2, \dots, attr^1_n\}$ el conjunto de atributos que definen a un lugar y P como el conjunto potencia de C , entonces podemos definir a un lugar ***l(ele)*** de la siguiente forma:

$$l = (P(C), List(attr^l)) \quad (4)$$

Más adelante acotaremos los tipos de ítems que puede haber en un lugar, mientras que los atributos corresponden a las características propias de un lugar como nombre, descripción, coordenadas geolocalizadas, entre otros.

3.2.2.2 Usuario

Los usuarios son aquellos que interactúan con los ítems del entorno, agregan nuevos y hacen votaciones respecto a estos. La definición de un usuario debe de ser minimalista, pero a la vez extensible para cualquier dominio adicional sobre el que se desee usar la heurística:

$$\mathbf{u} = (\text{attr}^{\mathbf{u}}_1, \text{attr}^{\mathbf{u}}_2, \dots, \text{attr}^{\mathbf{u}}_n) \quad (5)$$

De esta forma tenemos definido al usuario por un listado de atributos como nombre, puntos de experiencia, *karma points*, entre otros. Los sistemas de recomendación usan un modelo muy similar para la definición de los usuarios.

3.2.2.3 Ítem

Para nuestro caso, se asumirá que un ítem es todo aquello que el usuario puede agregar al contexto, pero además puede asignar una preferencia u opinión sobre éste. Por ejemplo, para nuestro caso pueden llegar a ser comentarios, ofertas, *tags* o, incluso, lugares. Tal como se menciona, todo ítem tiene su creador, el cual tiene que ser un usuario. Adicional a esto cada ítem puede tener un conjunto de atributos propios. Teniendo $\mathbf{B} = (\text{attr}^{\mathbf{c}}_1, \text{attr}^{\mathbf{c}}_2, \dots, \text{attr}^{\mathbf{c}}_n)$ como el conjunto de atributos de un ítem, procedemos a definir el ítem \mathbf{c} de la siguiente forma:

$$\mathbf{c} = (\mathbf{u}, \text{List}(\text{attr}^{\mathbf{c}})) \quad (6)$$

Donde \mathbf{u} es el usuario que agregó el ítem y además se tiene el listado de atributos que mencionábamos antes, los cuales pueden ser la fecha en que se agregó, la fecha en que se validó el ítem, el contenido de éste, entre otros. Como siempre, extensible para su uso.

Caso particular, mencionamos que un ítem es aquello sobre el cual el usuario puede mostrar una preferencia u opinión. Debido a esta definición, es que debemos de incluir a los lugares como un subconjunto de ítems, pues los usuarios agregan lugares e interactúan con estos al momento de hacer *check-in*, de darle *like* y otras interacciones posibles sobre éste. Por lo tanto, teniendo a \mathbf{L} como el conjunto de lugares y a \mathbf{C} como el conjunto de ítems, podemos definir:

$$\mathbf{L} \subseteq \mathbf{C} \quad (7)$$

De esta forma definimos al conjunto de lugares como un subconjunto de los ítems. En el acotamiento de esta representación se entenderá un poco mejor el porqué de esta consideración.

3.2.2.4 Evento

La definición utilizada para evento es similar a la utilizada por [Tareen2010], pues es la que mejor define el modelo para nuestro caso. No

tomamos en cuenta la definición de [Palomino12], pues llega a un nivel de comunidades, el cual no es necesario para nuestro caso particular de validación y preferencia del usuario. Tomando en cuenta las definiciones realizadas en los puntos anteriores y agregando las interacciones posibles que puede tener el usuario con los ítems, es posible definir un evento de la siguiente forma:

$$\begin{aligned}
 \mathbf{U} &= \{u_1, u_2, \dots, u_n\} \\
 \mathbf{C} &= \{c_1, c_2, \dots, c_n\} \\
 \mathbf{I} &= \{i_1, i_2, \dots, i_n\} \\
 \mathbf{E} &= \{(u, c, i, t) \mid u \in U, c \in C, i \in I, t \text{ es un } \textit{timestamp}\} \\
 \mathbf{V} &= \{(e, a) \mid e \in E, a \text{ es algún valor}\}
 \end{aligned}
 \tag{8}$$

En esta definición tenemos a E como un evento que sucede debido a una interacción de un usuario sobre un ítem, tomando en cuenta que un ítem puede ser un lugar o estar contenido en un lugar podemos tener la ubicación geolocalizada de la interacción, teniendo una marca de tiempo del momento que ocurrió esto. Así también tenemos V como el posible valor que haya agregado el usuario en esta interacción al ítem o lugar visitado.

3.2.2.5 Acotación de representación

El modelo definido para la representación de eventos, interacciones e ítems fue basado tomando en cuenta las recomendaciones de [Tareen2010] y [Vasquez14], pues nos permite definirlo para que pueda evolucionar de acuerdo a las necesidades de prueba de la heurística. Sin embargo, para el contexto sobre el cual trabajamos y validamos la heurística de validación, decidimos acotar cada uno de los puntos. Decidimos no tomar en cuenta por el momento el acotamiento de los atributos que puede tener un lugar, ni de las interacciones posibles, pues más adelante se explicarán más detalles sobre estos.

Respecto a los usuarios, los atributos que tomaremos en consideración en este caso son los siguientes: nombre de usuario, avatar o imagen, puntos de experiencia y los puntos de karma. Mientras que los ítems pueden ser de distinto tipo, como son lugares, comentarios, ofertas, propuesta de modificación o *tags*. Tal como se indicó previamente, estos son creados por los usuarios y para el caso de los 3 últimos tienen como atributo un texto y

la fecha en que fueron creados. Finalmente, los valores para los eventos variarán si es un *like* o *dislike*, en nuestro caso lo consideramos como un "1" y su negativo "-1".

Con esta acotación explicada, se procederá a explicar la estrategia de validación que se usará sobre este modelo. Así también se explayará sobre las implicancias que tienen las interacciones con los atributos del usuario y sobre los ítems mismos.

3.3 Estrategia de validación

Habiendo dado a conocer los componentes de motivación utilizados en la heurística, procederemos a explicar las acciones que se consideraron en la estrategia de validación. Además se describirá el comportamiento del sistema ante estas acciones y el uso de los KP para estas validaciones. Se explica también la lógica para incrementar o decrementar los KP.

Lo primero que debemos de conocer de esta estrategia de validación es "*¿Qué vamos a validar?*". Para el caso de esta heurística hemos mencionado siempre la validación de los lugares y su información georreferenciada, pero a continuación se listarán las características o atributos que para un usuario podría ser importante de conocer respecto a un lugar:

- Existencia
- Nombre
- Descripción
- Tipo de lugar (Restaurant, hospital, biblioteca, mall, entre otros)
- *Tags* relacionados (Restaurant: comida, completo, chorrillana, entre otros)
- Dirección
- Geocalización (Longitud y latitud)
- Comentarios
- Ofertas

Todas estas características de un lugar son validadas por la heurística. Además, la validez de esa información se ve afectada por las acciones de otros usuarios. Antes de pasar al listado de acciones que consideraremos como parte de esta heurística, definiremos bien la frase "votar

correctamente". Se define como votar correctamente, al hecho de votar en favor del resultado que fue elegido como "bueno" o "correcto". Por ejemplo, si un usuario vota indicando que un cierto *tag* para un lugar es correcto, y la heurística decide que efectivamente el *tag* es correcto, entonces dicho usuario habrá votado correctamente. En caso contrario, su voto será considerado como incorrecto, siendo considerado, solo en este caso validado, como usuario malintencionado.

Antes de ver el listado de acciones que modifican los KP del usuario, vamos a listar las acciones o interacciones que puede realizar el usuario:

- *Check-in* (indicar que se encuentra en un lugar)
- Validar *tags*
- Agregar *tags*
- Agregar nuevos lugares
- Validar lugares
- Escribir comentarios en lugares
- Agregar ofertas de lugares (una oferta del lugar, es muy diferente a un comentario sobre un lugar, por ejemplo: "martes 2x1", "viernes a mitad de precio", entre otros)
- Proponer modificación de información de lugar
- Validar modificación de información de lugar

Hay que tener presente que todas estas acciones pueden modificar los puntos de experiencia de los usuarios. Sin embargo, no necesariamente van a modificar sus KP. Las acciones que modificarán los KP de los usuarios, según la heurística, son las siguientes:

- Al momento de validar un *tag*, el cual se encuentra previamente validado, el usuario responde correctamente (respecto de si el *tag* es correcto o incorrecto), entonces se le incrementarán los KP a ese usuario. En caso contrario, se le procederá a decrementar los KP.
- Al agregar un lugar o un *tag* que después pasa a ser validado como correcto, entonces se le incrementarán los KP a dicho usuario. Caso contrario, se procederá a decrementarlos.

- Al votar correctamente en la validación de un lugar o un *tag*, se le incrementarán los KP. Caso contrario, se le decrementarán.
- Al agregar un comentario u oferta que después pasa a ser votado positivamente por cierta cantidad de usuarios, se le incrementarán los KP. En caso sea votado negativamente, se le decrementarán estos y su comentario u oferta tendrá menos prioridad para ser mostrada.
- Al votar correctamente por un comentario u oferta se le incrementarán los KP. En caso contrario, se le decrementarán.
- Al agregar una propuesta de modificación de información de un lugar, que después pasa a ser validado como correcto, entonces se le incrementarán los KP. En caso contrario pasa a ser validado como incorrecto, se le decrementan los KP a dicho usuario.
- Al votar correctamente en la validación de una propuesta de modificación de información de un lugar, se le incrementarán los KP. En caso contrario, se le decrementarán.

Tal como se puede ver, cada acción que realice un usuario tiene una implicancia sobre sus KP, con la única excepción de la acción *check-in* (es decir, indicar que el usuario está en un cierto lugar). Es decir, un usuario puede tener muchas acciones, donde algunas de éstas le incrementen los KP, mientras que otras los decrementen, lo que generará una variación de KPs de acuerdo a las acciones que tome el usuario durante el tiempo de prueba de la heurística.

Faltaría sólo un punto antes de explicar la lógica utilizada para incrementar o decrementar los KP de un usuario. La heurística asumirá que inicialmente se va contar con una base de datos de lugares validados, y *tags* válidos e inválidos para éstos. Estos últimos van a ser de mucha utilidad, para determinar la validez (potencial o efectiva) de la respuesta de un usuario.

Al contar con los *tags* válidos e inválidos, se aplicará una lógica muy similar a la de Recaptcha. Es decir, si el usuario indica que un lugar es válido, y en la base de datos ese lugar está marcado como válido, entonces se indicará que el usuario votó correctamente. Los votos correctos se verán recompensados, mientras que los incorrectos serán penalizados. A

continuación en la figura 8 se puede ver un diagrama de flujo explicando esta lógica.

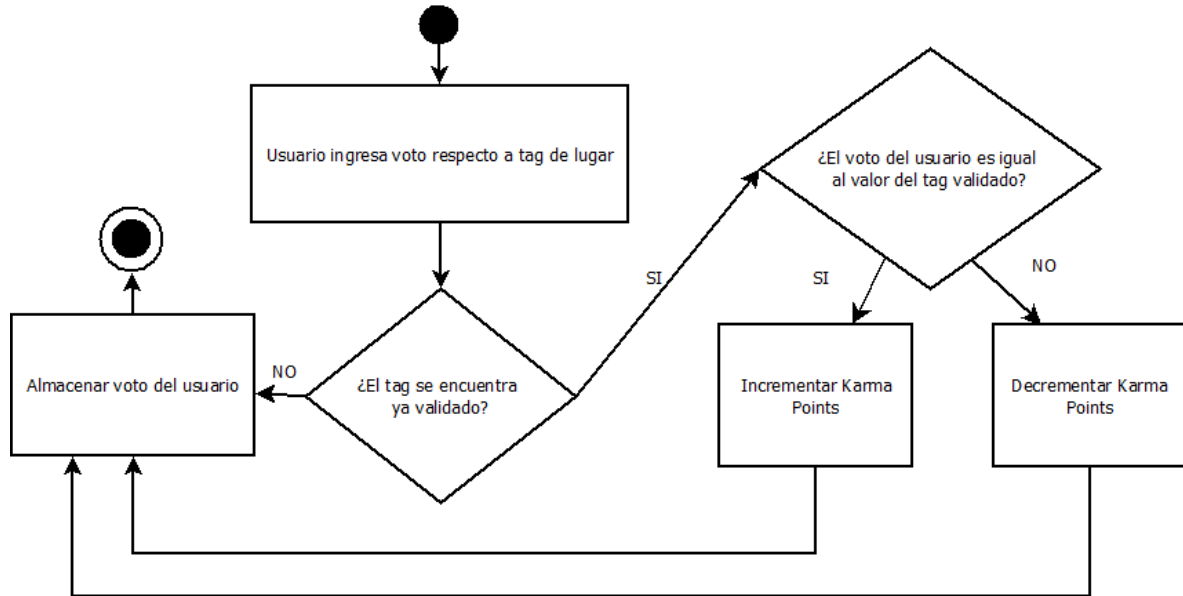


Figura 8: Diagrama de flujo de validación inicial de tags

Cabe recordar que la recompensa y penalización de los KP no es algo visible para el usuario. Por lo tanto, nuestra heurística empezará con esta información pre-validada, la cual será útil para analizar la intencionalidad de los usuarios, lo cual será medible por los KP. Teniendo en cuenta esto, las otras acciones que tendrá disponible el usuario son aquellas basadas en agregar información o recomendar modificaciones de éstas. En esos casos haremos uso de los KP, e iremos acción por acción analizando qué es lo que ocurre con los KP del usuario a través del tiempo. En los casos que el *tag* no se encuentre validado, sólo se almacenará el voto del usuario para después pasar por el proceso de validación de *tag* (Figura 10).

3.3.1 Agregar tag

La acción de agregar un *tag* está relacionada con la parte de validación de *tags*, ésta se realiza exactamente al momento que se agrega un voto de validación a un *tag* que no se encuentra validado. El proceso es el siguiente, primero se da el agregado del *tag*, después los usuarios proceden a votar si el *tag* es correcto o no para el lugar, y por último, se realiza la validación del *tag*. Tal como se puede ver en la siguiente figura, el proceso de agregado de un *tag* es muy simple y casi atómico:

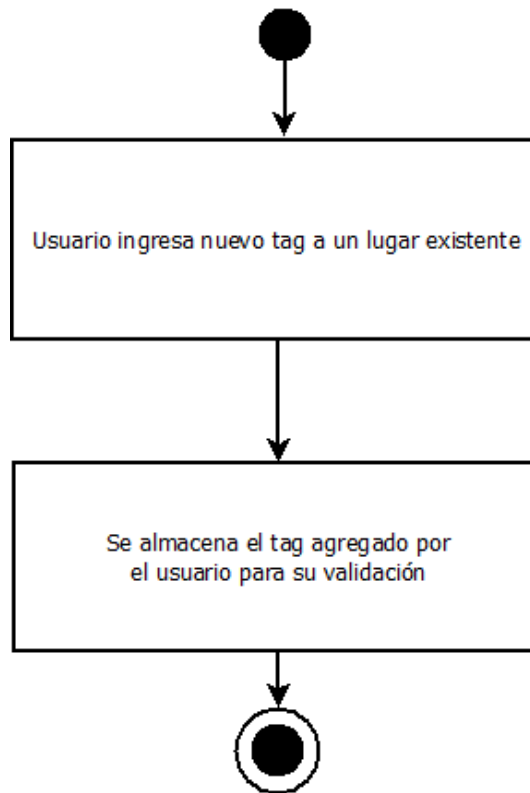


Figura 9: Diagrama de flujo de agregar tag

Después de este proceso, y tal como comentamos antes, se sigue con la votación del *tag* por parte de los usuarios. Allí se almacenan los votos, tanto a favor como en contra de los usuarios (Figura 8). Ahora procedemos a mostrar lo que ocurre en el proceso de validación de cada *tag*, que se realiza periódicamente cada un cierto tiempo (configurable). Para nuestro caso este tiempo fue de 30 minutos. En la figura 10 se puede ver claramente el proceso para que se valide un *tag*, así como las consecuencias que tiene sobre los KP de los usuarios votantes.

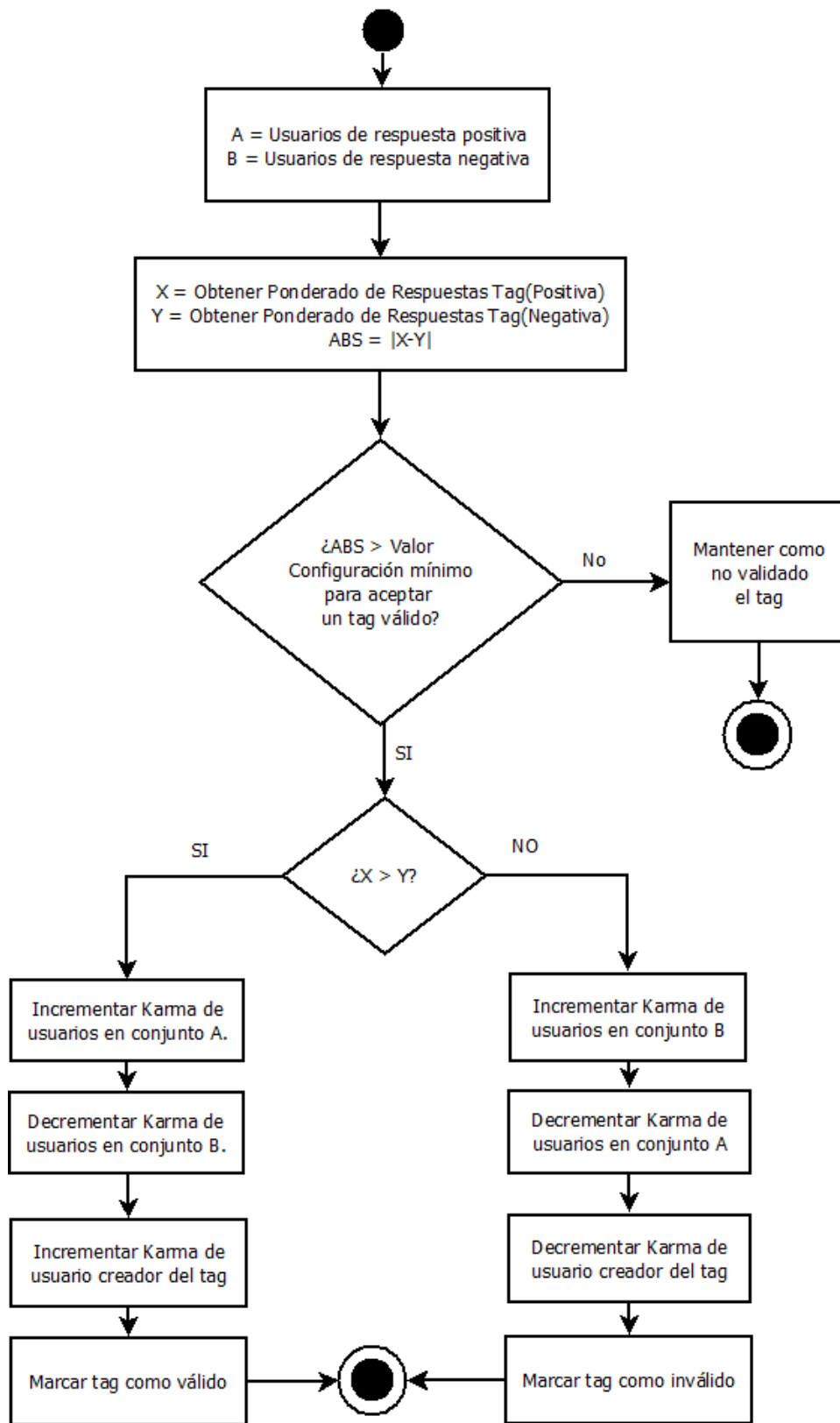


Figura 10: Diagrama de flujo de validación de un tag

El proceso de "Obtener ponderado de respuestas para un tag" se puede ver en el siguiente diagrama de flujo.

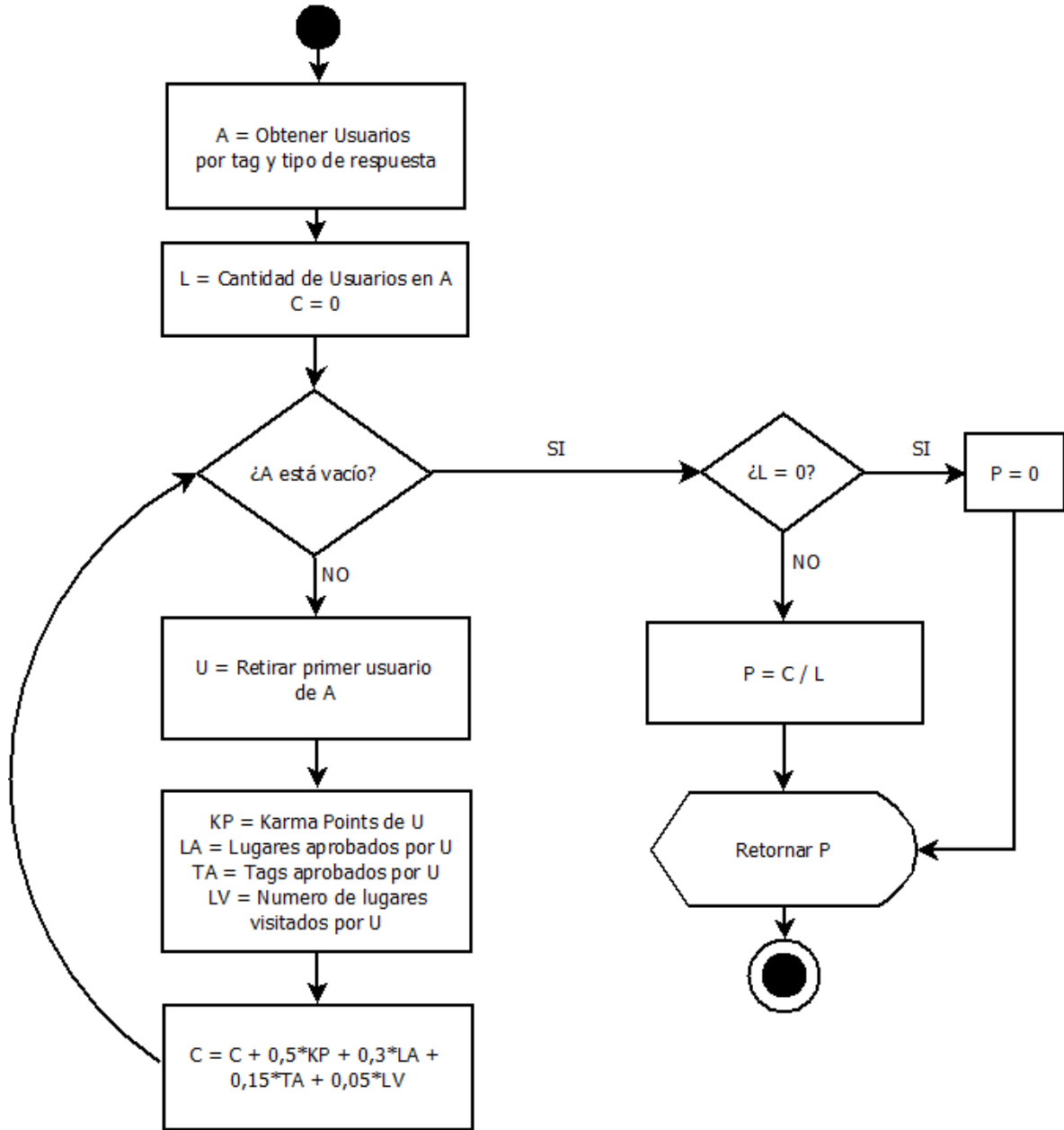


Figura 11: Diagrama de flujo de ponderado de respuestas tag

Se puede ver que la fórmula utilizada, la cual se explica más en la sección 3.4, para obtener el "puntaje" de cada usuario que está votando se basa en los KP que posee cada usuario, a pesar de que también participan variables como "lugares aprobados", "tags aprobados" y "número de lugares visitados". Sin embargo, estos no tienen el peso porcentual que poseen los KP, fortaleciendo así la idea del uso de los KP para medir el nivel de

confiabilidad de los usuarios. De esta forma, se entra en una especie de círculo vicioso/virtuoso, donde es necesario tener mayor cantidad de KP para así poder ganar más de estos, lo cual implica ser un usuario bienintencionado. En caso contrario, al comportarse el usuario en forma malintencionada, éste irá perdiendo sus KP en forma paulatina en función de sus acciones, y por lo tanto cada vez sus acciones tendrán menos peso en el proceso de validación de la información. La única forma que un usuario tiene de volverse "influyente", es sólo si primero se vuelve "bienintencionado".

Cabe recalcar dos temas antes de seguir con las siguientes acciones a considerar: *¿Qué ocurre si es que el tag pasa a ser válido cuando no lo es? ¿Los usuarios malintencionados mantendrán siempre estos KP ganados de forma indebida?* Para responder a estas preguntas hay que resaltar el hecho de que el proceso de validación de *tags* se repite cada cierto tiempo, por lo tanto las validaciones previas se pueden corregir en función de las acciones de los usuarios.

El período de demora hasta la próxima revalidación de la información puede discriminarse según se trate de *tags*, lugares o modificaciones a éstos. Este proceso no sólo considera a los *tags* o lugares que no tienen validación, sino también a los *tags* que ya han sido validados, pero cuentan con un "valor configuración mínimo para aceptar un *tag* válido" mayor al normal, pues esto evita que el proceso se ejecute muy frecuentemente. Al darse este caso, si el resultado de la validación de un *tag* es cambiado, se procederá a seguir el flujo normal determinado en la heurística, beneficiando a los que hayan votado correctamente y penalizando a los que no. Es en la etapa de la penalización, donde tenemos que atacar para tratar de salvaguardar al máximo la veracidad de la información ingresada al sistema. Es por ello que en los procesos de validación de los elementos de la heurística, las penalizaciones siempre van a ser mayores que los beneficios. Es decir, si es que un usuario creador agregó correctamente un *tag*, éste es beneficiado con "X" KP, mientras que un usuario creador que agregó un *tag* que pasó a ser inválido luego de la validación, será penalizado con "Y" KP, donde $Y > X$.

De esta forma, si un usuario malintencionado puede agregar un *tag* y hacer que éste pase a ser válido, a pesar de que no lo es, ganará KP debido a esto. Sin embargo, en un corto tiempo, los votos de los usuarios bienintencionados (los cuales normalmente poseen mayor KP) procederán a revertir la situación y el usuario malintencionado quedará con menos KP que

con los que empezó. De esta forma se trata en lo posible de mantener una constante validación de la información, y a los usuarios malintencionados con la menor cantidad de KP posibles. Cabe resaltar que esta penalización de los usuarios malintencionados creadores de *tags* también aplica para aquellos que votan incorrectamente por una opción que no es correcta.

3.3.2 Agregar lugar

El proceso de agregado de lugar muy similar al de agregado de *tag*. Sólo lo diferencia la cantidad de información que debe ingresar el usuario. Sin embargo, el proceso de votación de la validez de un lugar es diferente al de un *tag*, debido a que éste sólo se hace sobre lugares "no validados". Por lo tanto, en el momento de la votación, su opinión no causará una alteración en los KP de dicho usuario. Es decir, al agregar un voto sobre un lugar, sólo se registrará el voto del usuario, previo al proceso de validación del lugar.

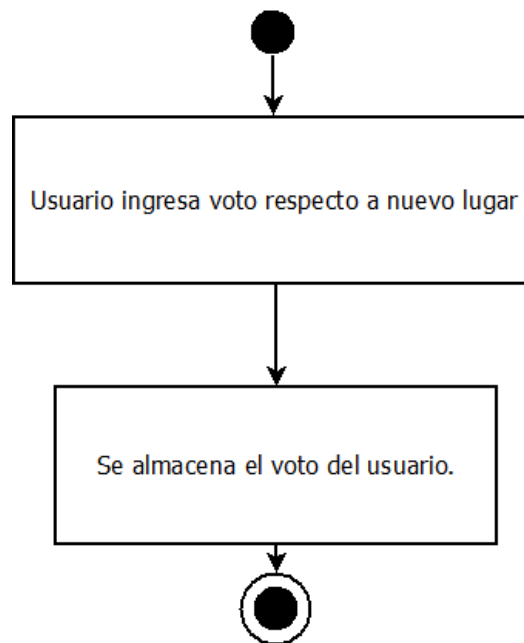


Figura 12: Diagrama de flujo de votación de lugar

Finalmente, se da el proceso de validación de lugar, donde se hace uso de la misma fórmula y proceso indicado en la figura 10. El proceso para la obtención del ponderado de las respuestas es similar al usado para la ponderación *tags*. La única diferencia es que los parámetros de entrada son el lugar y el tipo de respuesta que da el usuario, por lo que el flujo es netamente igual al de figura 11.

Tal como se puede ver, el proceso es totalmente similar al de los *tags*. Pero la única diferencia que se dio para el comportamiento del proceso de validación de los lugares es que estos tienen un tiempo máximo de vida para ser validados, en caso se sobrepase este tiempo, pasará al estado de “no validado” y no se mostrará más a los usuarios, quedando únicamente almacenado para estadísticas, así también, estos lugares no validados no afectarán los KP de los usuarios votantes.

Cabe resaltar que los casos especiales que se dieron en la validación de los *tags* son cubiertos de una forma similar, pero no igual, para los lugares. Debido a que estos al ser validados no pasan por un constante proceso de validación, no se puede usar una forma igual, pero sí la estrategia de modificación de información de lugar que se verá a continuación.

3.3.3 Agregar propuesta de modificación de información de lugar

La acción de agregar una propuesta de modificación de información de lugar nace bajo la necesidad de poder dar a los usuarios la posibilidad de corregir errores importantes en forma rápida respecto a los lugares, como son corrección del título, descripción, posición del lugar, así como indicar que un lugar no existe realmente o ya no se encuentra ahí, debido a mudanza, cierre de local u otros. El escenario que se da para la validación de una de estas propuestas de modificación de información de lugar es el siguiente: primero se da el agregado de la propuesta, después los usuarios son notificados de una nueva propuesta de modificación y proceden a votar si la propuesta es correcta o no, y por último, el proceso de validación de propuesta de modificación.

El agregado de la propuesta es sencillo como la figura 9, lo único complejo no está a nivel de heurística, sino a nivel de usuario, debido a que éste tiene que indicar los campos que desea proponer modificar, como por ejemplo puede elegir el título, descripción u otros. Después de esto, se realiza el proceso de votación de parte de los usuarios, similar al de votación de un lugar (figura 12), pues no influye a los KP del usuario al momento de emitir el voto. Finalmente, se da el proceso de validación de esta propuesta de modificación del lugar, donde se puede apreciar una leve diferencia con respecto a los demás procesos de validación (figura 13).

Tal como se puede apreciar, este proceso es similar a las anteriores validaciones vistas. Sin embargo, en la parte final cuenta con una validación

en caso la propuesta sea de eliminación, pues en este caso es que se cuida el punto visto en la sección de "agregar lugar", donde si un lugar inválido pasa a ser válido. Con una propuesta de eliminación que pase a ser válida, se penalizará al usuario creador del lugar inválido con una disminución de KP mayor a los que pudo obtener al pasar a ser validado incorrectamente. De esta forma se sigue manteniendo cuidado de los usuarios malintencionados penalizándolos de una forma constante.

Otro punto a recalcar en esta validación, es que se va contar con dos valores de configuración mínima para validar una propuesta, debido a que una propuesta de modificación de información básica del lugar, no afectaría tanto como una propuesta de modificación del tipo eliminación. Siendo esta última mayor a la anterior, pues es un proceso que toma un acción muy fuerte como el eliminar un lugar.

El proceso de obtención del ponderado de respuestas se mantiene similar a la de los anteriores, por lo tanto sigue el mismo diagrama de flujo de la figura 11. El proceso de ponderado sigue repitiéndose para cada ítem, haciendo hincapié en los KP de los usuarios para el ponderado.

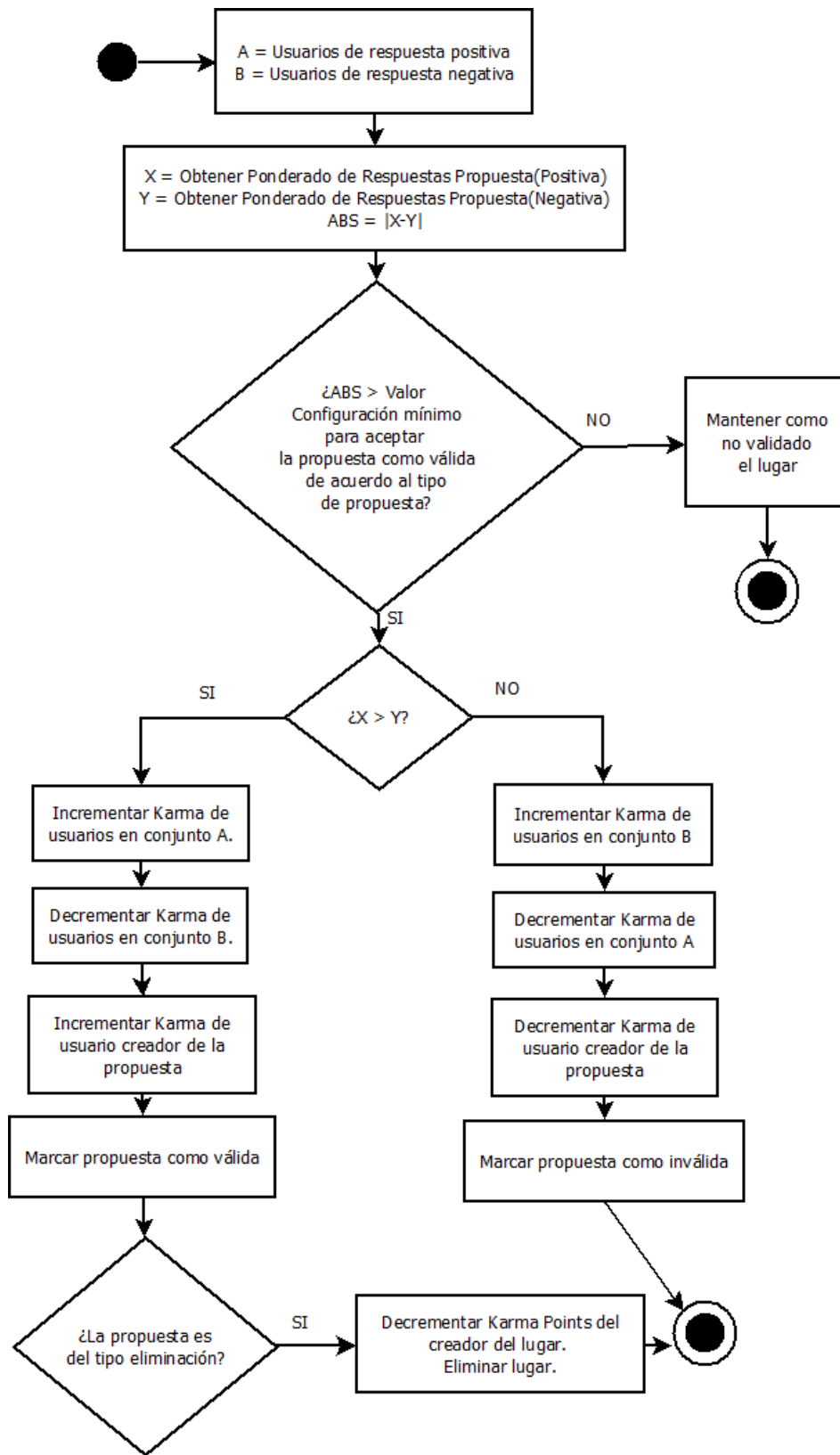


Figura 13: Diagrama de flujo validación de propuesta de modificación de información de lugar

3.3.4 Agregar comentario/oferta

El proceso de agregado de comentario y oferta es completamente igual, a diferencia del valor de configuración mínimo para validación, es por ello que se colocan en una misma sección. El agregado del comentario/oferta se da sin ninguna lógica adicional, únicamente registrándolo colocando como creador al usuario, por lo que es similar a la figura 9.

Después de este proceso, se procede a capturar los *likes* y *dislikes* de los usuarios, que contaremos como votaciones al respecto, por lo que nos basaremos en la figura 12. Finalmente, se da el proceso de validación, el cual, así como los otros, se ejecuta cada cierto tiempo configurable (figura 10). Así también, el proceso de ponderado no sufre cambio con respecto al ponderado de los otros ítems (figura 11).

Como único comentario a realizar, podemos decir que en la lógica de validación que se utiliza para comentarios/ofertas, se da el mismo caso que en *tags* para los usuarios malintencionados, donde aquellos que dieron como válidos sus comentarios/ofertas. Más adelante se validará nuevamente y se les penalizará a los creadores malintencionados, de esta forma se sigue manteniendo una tendencia descendente de los KP de los usuarios malintencionados.

Finalmente, después de analizar el comportamiento de la heurística ante cada una de las acciones que están permitidas para los usuarios, podemos ver que el factor más importante de ésta son los KP de los usuarios, que sirven para medir lo confiable que resultan estos. Así también, se puede ver que estos van incrementándose mientras el usuario siga teniendo un comportamiento correcto, lo cual genera un mayor peso en la opinión de estos usuarios al momento de votar por un lugar, *tag*, comentario u otro.

Además de esto, hay que notar que en cada validación existen siempre casos donde un usuario malintencionado puede haberse saltado la heurística, pero estos casos son cubiertos con una "re-validación", el cual mantendrá con menos KP a estos usuarios. Se recalcará un poco más la robustez de la heurística en el siguiente punto, explicando a detalle los comportamientos de los tipos de usuarios esperados.

3.4 Robustez de la heurística

Debido a que toda heurística tiene ciertas debilidades, se realizó un breve análisis respecto a las situaciones que se pueden dar en la validación de la

heurística, así como las respuestas esperadas en estos comportamientos. La idea de esto es analizar el soporte de la heurística ante ataques de usuarios maliciosos.

Dado que las fórmulas pueden resultar muy largas al momento del análisis, se tomará en cuenta las siguientes abreviaciones (Tabla 1).

Tabla 1. Tabla de acrónimos de heurística

Acrónimo	Descripción
KP	Karma points
MKP	Mínimo de karma points
KPI	Karma points iniciales. Es decir, cuando empieza el sistema, cada usuario empieza con 2 KP
VCI	Valor de configuración ítem. Es el valor que debe de sobrepasar la heurística para ser considerada válida para cada ítem (Lugares, tags, comentarios, ofertas)
MUM	El número mínimo de usuarios malintencionados
UB	Número de usuarios bienintencionados
MUV	Mínimo de usuarios votantes para considerar ejecutar la heurística. Si es que los usuarios votantes son menores al MUV, no se ejecuta la heurística y el ítem no se válida hasta llegar al MUV.

Estado inicial de la validación

En el estado inicial de la validación de la heurística, se toma como suposición que todos empiezan con $KP = KPI$, es un estado donde nadie ha hecho *check-in*, no han ganado KP, es tomado como un estado cero. Tomando en cuenta esto, la fórmula de:

$$0.5*(Karma Points) + 0.3* (Número de lugares aprobados) + 0.15* (Número de tags aprobados) + 0.05 * (Número de lugares visitados)$$

Quedaría reducida a:

$$0.5 * KP$$

Debido a que $KP = KPI$, entonces $KP = 2$, por lo tanto la fórmula dará un valor constante de 1. Entonces, tomando en cuenta que queremos encontrar el mínimo de personas malintencionadas (x) donde al momento de multiplicar por la fórmula (F) y obtener el promedio aritmético, debería verse así la fórmula:

$$(MUM * F) / MUM > VCI$$

Debido a que MUM se divide por MUM y a que el valor de la fórmula (F) es un valor constante, entonces se tiene lo siguiente:

$$1 > VCI$$

Por lo tanto, podemos ver que se obtendrá un valor constante, el cual, si es que VCI se configura mayor a 1, nunca sobrepasará el VCI . Por lo cual, sin importar la masiva cantidad de usuarios malintencionados al inicio de la validación traten de ingresar algún dato falso o incorrecto, éste no pasará la validación.

Estado avanzado de la validación

Debido a que son muchos los factores que influyen en la fórmula de la heurística, se estableció un tipo de gráfico de comportamiento esperado de los KP de los usuarios malintencionados, el cual se puede ver a continuación:

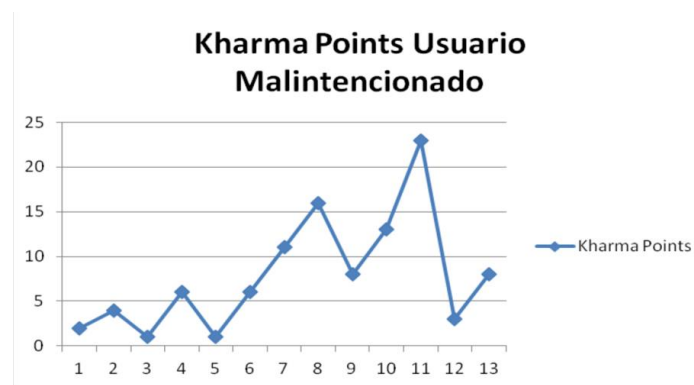


Figura 14 Comportamiento esperado de un usuario malintencionado

Este comportamiento esperado con tantas disminuciones de KP , es debido a las reglas que se mencionaron anteriormente, las cuales controlan el sancionar a los usuarios malintencionados con un monto mayor al que puedan haber ganado mediante otro medio KP . Por lo tanto, se espera que

lleguen a tener una especie de asíntota de la cual no puedan pasar durante toda la fase de validación.

Por el otro lado, se puede considerar una gráfica para los usuarios bienintencionados como la siguiente:

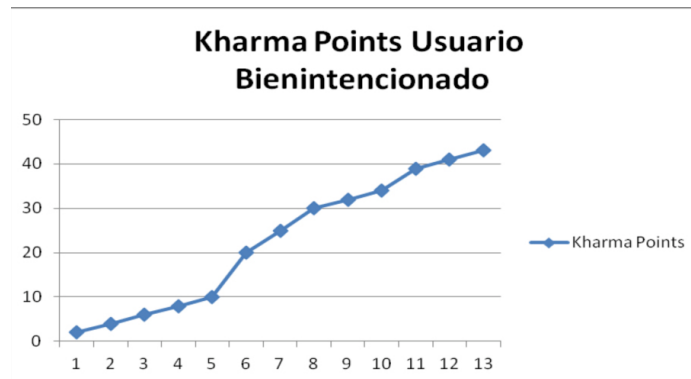


Figura 15. Comportamiento esperado de un usuario bienintencionado

Se puede ver que no hay una oscilación en comparación con la gráfica de los usuarios malintencionados. Al contrario, tiende a ser una función logarítmica en la que se ven incrementados los KP de los de usuarios.

Hasta el momento se podría indicar que la heurística tiene una robustez aceptable, debido a que los usuarios bienintencionados tienen un karma que se incrementa a través del tiempo, mientras que los usuarios malintencionados tienen una función oscilante con una especie de asíntota, por lo cual se espera que los resultados del cálculo de la heurística siempre den como ganador a los usuarios bienintencionados. Por lo tanto, se tendría lugares e información relacionada correctamente agregada a la base de datos.

Sin embargo, a pesar de todo esto, se puede dar casos en que no necesariamente en una votación se topan usuarios con el mismo estado, es decir, pueden aparecer usuarios malintencionados en estados como el 11 que se ve en la figura 14, muy por encima de los usuarios bienintencionados que se encuentren en los estados iniciales en los cuales tienen pocos KP, generando una votación a favor de los malintencionados, lo cual puede llegar a incurrir en la corrupción de los datos.

Se puede observar entonces que si el 100% de usuarios fueran malintencionados con estados como el 11 de la gráfica, se necesitaría un

mínimo igual al MUV, de tal forma que se ejecute la heurística, así también, los KP deberían de ser mayor o igual al VCI.

Entonces, tomando en cuenta lo último mencionado, se puede observar que se necesita al menos un usuario bienintencionado con unos KP mínimo necesarios para no validar incorrectamente el ítem, tal que estos $KP \cdot 0.5$ al restarlos por el valor de F de los usuarios malintencionados den un valor menor al VCI.

3.5 Ponderación de la fórmula

Se puede observar que en casi todos los casos se ha mostrado la fórmula de la siguiente forma:

$$0.5*(Karma Points) + 0.3* (Número de lugares aprobados) + 0.15* (Número de tags aprobados) + 0.05 * (Número de lugares visitados)$$

Se puede ver que cada una de las características que tiene el usuario dentro de la heurística posee un peso asignado. Estos pesos no fueron asignados de forma arbitraria, el proceso para colocar los pesos se basó inicialmente en el listado de las variables que describen al usuario como son: *karma points*, *experience points*, Número de lugares aprobados, número de *tags* aprobados, número de lugares visitados, número de comentarios votados como buenos, número de votaciones realizadas, entre otros. Se tomó en cuenta con mayor peso a aquellas que pueden indicarnos con mayor claridad que el usuario tiene la mejor intención de otorgar información fidedigna, así como colaborar correctamente con ésta.

Tomando en cuenta estas variables, se procedió a colocar valores arbitrarios inicialmente, para después proceder a ejecutar una simulación de usuarios, lo cual permitió ver cuánto se demoraba un lugar, *tag* o comentario/oferta en validarse al colocar estos valores. Iterando sobre estos tests, se pudo obtener los valores mostrados en la fórmula, los cuales nos dan una validación la cual no demora tanto, ni tampoco permite validar la información de forma demasiado sencilla, así como también, trata en lo posible de priorizar en lo mayor posible los KP para evitar a los usuarios malintencionados.

3.6 Temporalidad de la información

Debido a que muchas variables del sistema pueden variar durante el tiempo, por diversos motivos, se estableció el considerar una temporalidad para algunos de estos: *karma points*, *tags*, comentarios y ofertas.

La temporalidad que se le da a los *tags*, comentarios y ofertas es debido a muchos casos, entre estos puede darse que un restaurante cambie de dueño o chef, por lo cual la calidad de la comida se puede ver afectada, haciendo que los usuarios al ir al lugar debido a que estaba con muy buenos comentarios y *tags* resaltantes, se vean frustrados con que encuentren todo lo contrario. Incluso, los usuarios pueden llegar a pensar que la información provista por el sistema es incorrecta. Debido a esto es que esa información va tener una temporalidad. La temporalidad que se la va otorgar a la información va ser inversamente proporcional al número de votos que tenga cada ítem de la información, es decir, a mayor número de votos o interacciones tenga este ítem, menor va ser la temporalidad.

El tema de la temporalidad también va afectar a los KP, esto para no dar tanto "peso" de votación a los usuarios siempre, sino hacer que estos se mantengan activos validando información y ganando KP constantemente. Debido a esto los KP se configuraron para tener una caducidad de 2 meses desde su obtención.

4. Herramienta implementada

Para la validación de la hipótesis se realizó el desarrollo de la herramienta llamada Geocrowd, la cual considera la hipótesis de trabajo planteada y el estado del arte investigado. En este capítulo se describirá los requisitos a satisfacer, así como el diseño e implementación de la herramienta antes nombrada. Entre las descripciones se tendrá el ambiente operacional, los requerimientos a alto nivel, el diseño de la solución, así como unos vistazos de la solución implementada.

El objetivo principal de esta herramienta es validar la heurística propuesta, pero además ésta pretende animar a los usuarios a generar una especie de mapa colaborativo, donde cada uno de ellos pueda agregar lugares, modificarlos, validarlos, agregar información referente al lugar, entre otros; se podría resumir como una especie de mapa de lugares al estilo Wikipedia.

Todo este trabajo colaborativo, además de servir como una Wikipedia, pretende estimular su uso mediante los componentes de *gamification* necesarios para ésta, convirtiendo una actividad que de repente resulte aburrida, en algo más interesante para los usuarios.

4.1 Ambiente operacional

Debido a la naturaleza de la hipótesis a validar, la herramienta fue realizada en 2 partes: una API y una aplicación cliente móvil. Mientras que para la API se utilizó tecnología web como Python, Django y de base de datos PostgreSQL (PostGIS), para la aplicación cliente móvil, se desarrolló para celulares con Android.

Entrando a más detalle, la API fue desarrollada con Django, combinado con una modificación de Tastypie y un componente adicional llamado RestMocker [Restmocker14]. El formato de comunicación de entrada y salida de la API fue mediante Json, debido a su simplicidad, alta demanda de uso y poco procesamiento de parseo que necesita. La razón de combinar tecnologías como Django y Python para el desarrollo es debido a la facilidad de poder crear una API con éstas. Así también, el uso de RestMocker es para definir rápidamente la interfaz entre el cliente y la API, creando una API falsa que retorne datos estáticos, esto simplifica la dependencia entre ambos, cliente y servidor.

Para la ejecución de los procesos de validación de lugares, modificaciones, *tags*, comentarios y ofertas, se utilizó Celery [Celery14] como componente debido a su orientación de ejecutar tareas de forma periódica y, finalmente, fue administrada mediante Supervisor.

La API se puso en producción sobre un servidor Ubuntu 12.04, con un servidor web Nginx [Nginx14], base de datos PostgreSQL adicionado con el componente PostGIS [Postgis14], estos 2 últimos componentes se hicieron uso en conjunto debido a que lo que guardamos son datos geográficos.

Del lado de cliente, se trató de hacer una aplicación móvil basada en un sistema operativo de alta demanda como es Android, debido a que mientras más usuarios tenga disponibles la aplicación, generará un mejor resultado. Así también, la facilidad de distribución de aplicaciones que permite la plataforma fue un factor relevante para su elección. Respecto a las tecnologías utilizadas para este cliente móvil, se hizo uso del SDK básico de Android, así como de la librería GSON, la cual permitía establecer un parseo limpio de las comunicaciones con la API.

Para la distribución de la aplicación generada de Android (apk), se utilizó la vía mail, debido a la privacidad que se debía de mantener de ésta y, además, que son usuarios seleccionados.

4.2 Requisitos de la solución

Los casos de uso de alto nivel que se identificaron para la herramienta que valida la hipótesis se pueden ver representados en el siguiente diagrama de casos de uso:

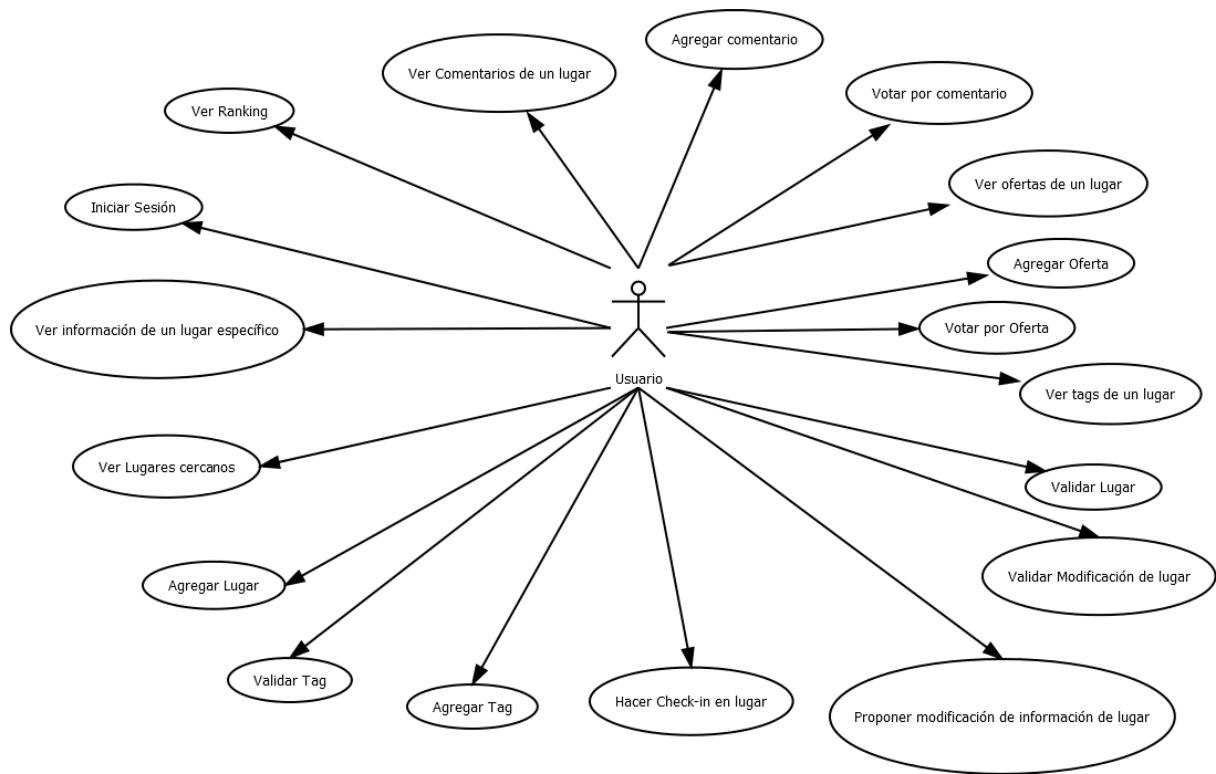


Figura 16: Casos de uso de la solución

Debido a la naturaleza de la aplicación cliente, sólo habrá un actor involucrado con los casos de uso principales. Esto no descarta que de parte del sistema se estén ejecutando procesos de validación con cierta frecuencia, y es en estos procesos donde aplicamos nuestra heurística para la validación de lugares, *tags*, comentarios y ofertas. El principal trabajo se enfocó en estos procesos, debido a su importancia y complejidad.

Tomando en cuenta estos casos de uso que se pudieron capturar en la fase inicial de la herramienta, y además considerando los componentes necesarios para la validación de la heurística, los macro-requisitos para la solución son:

- **Solución móvil bajo SO Android:** El sistema debe de ser capaz de ejecutarse en dispositivos Android con versión superior a 2.1 debido a compatibilidad de componentes.
- **Capacidad de Comunicación:** La aplicación cliente debe de poder comunicarse a los servicios web de la API, para esto debe de usar la red de datos del dispositivo, sea WiFi o red móvil.

- **Geolocalización:** Debido a la disponibilidad de tecnologías de geolocalización que otorgan los dispositivos con Android, se decidió el uso de A-GPS y WiFi, en este orden de prioridad debido a la naturaleza de la tesis, pues necesitamos la geolocalización con el mayor grado de exactitud posible.
- **Gestión de lugares geo-localizados:** La aplicación debe de darle la facilidad al usuario de poder ver lugares cercanos, su información relacionada, así como también el poder agregar nuevos lugares o modificar estos de una forma segura.
- **Gamification:** El componente que genere esa motivación y comportamiento deseado para esta validación es muy importante. Se estableció una forma sencilla mediante *badges*, puntos de experiencia y un control interno de KP. Así también, se utilizaron estos puntos de experiencia para usar el componente de ranking, lo cual permitió generar un nivel de competitividad entre los usuarios.

4.3 Interfaces de la aplicación cliente

La solución se implementó con todos los componentes deseados que se mencionaron anteriormente y tomando en cuenta los estándares de diseño de aplicaciones en Android. A continuación se muestran las principales pantallas que tuvo la aplicación, así como la razón de existencia de estos.

4.3.1 Interfaz de inicio de sesión

Tal como lo indica su nombre, esta pantalla permite que el usuario pueda acceder a la aplicación, así como también poder obtener sus datos básicos como nombres, puntos de experiencia y demás relacionados. Después de iniciar sesión, esta sesión es mantenida en el dispositivo para no solicitarle nuevamente las credenciales de ingreso.

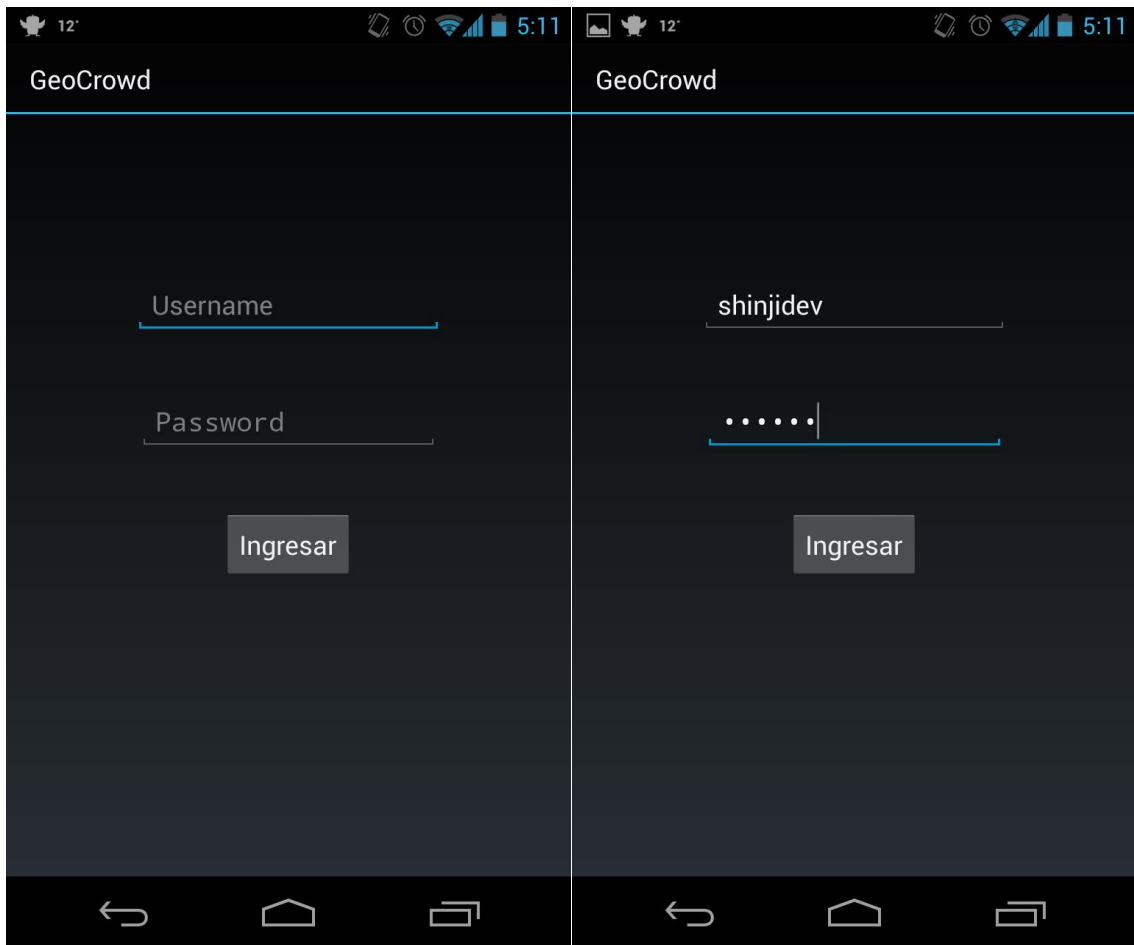


Figura 17: Interfaz de inicio de sesión

4.3.2 Interfaz de timeline

Luego de iniciar sesión, lo primero que se muestra es esta pantalla, la cual permite tener un conocimiento de lo que está sucediendo actualmente en su círculo social, de tal forma que genera visibilidad de los lugares que van visitando amigos, así como ver los puntos de experiencia que obtuvieron en ese lugar.



Figura 18: Interfaz de timeline

4.3.3 Interfaz de ranking

Esta pantalla, tal como fue comentado anteriormente, el usuario puede visualizar en qué posición se encuentra del ranking, así como también puede ver el status de su círculo de amigos. El objetivo de esta pantalla es primordial para generar la competitividad entre los usuarios de la aplicación.

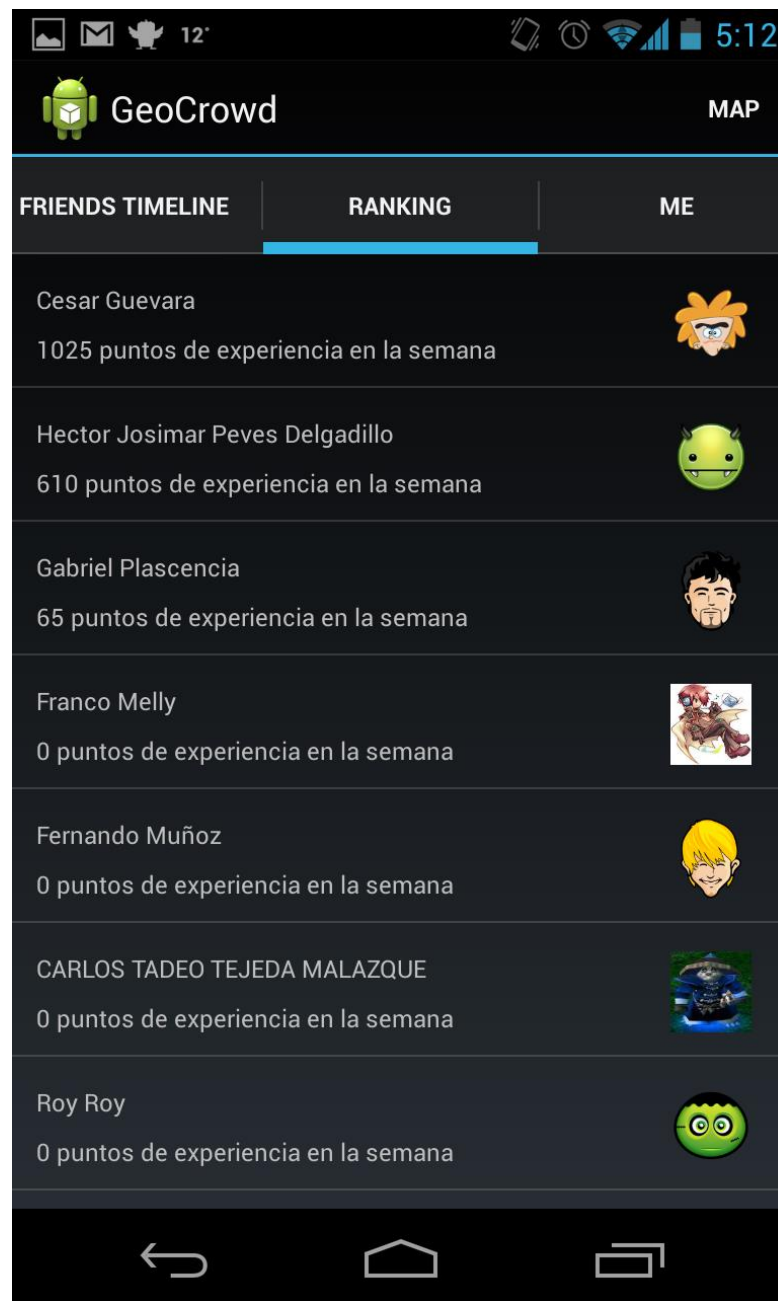


Figura 19: Interfaz de ranking

4.3.4 Interfaz de perfil de usuario

EL ingreso a esta pantalla es simplemente informativo, donde el usuario puede ver datos sencillos como su nombre, su foto actual, así como también sus puntos de experiencia acumulados hasta el momento. Además se puede seleccionar la visualización de los amigos que posee y de los *badges* ganados hasta el momento.

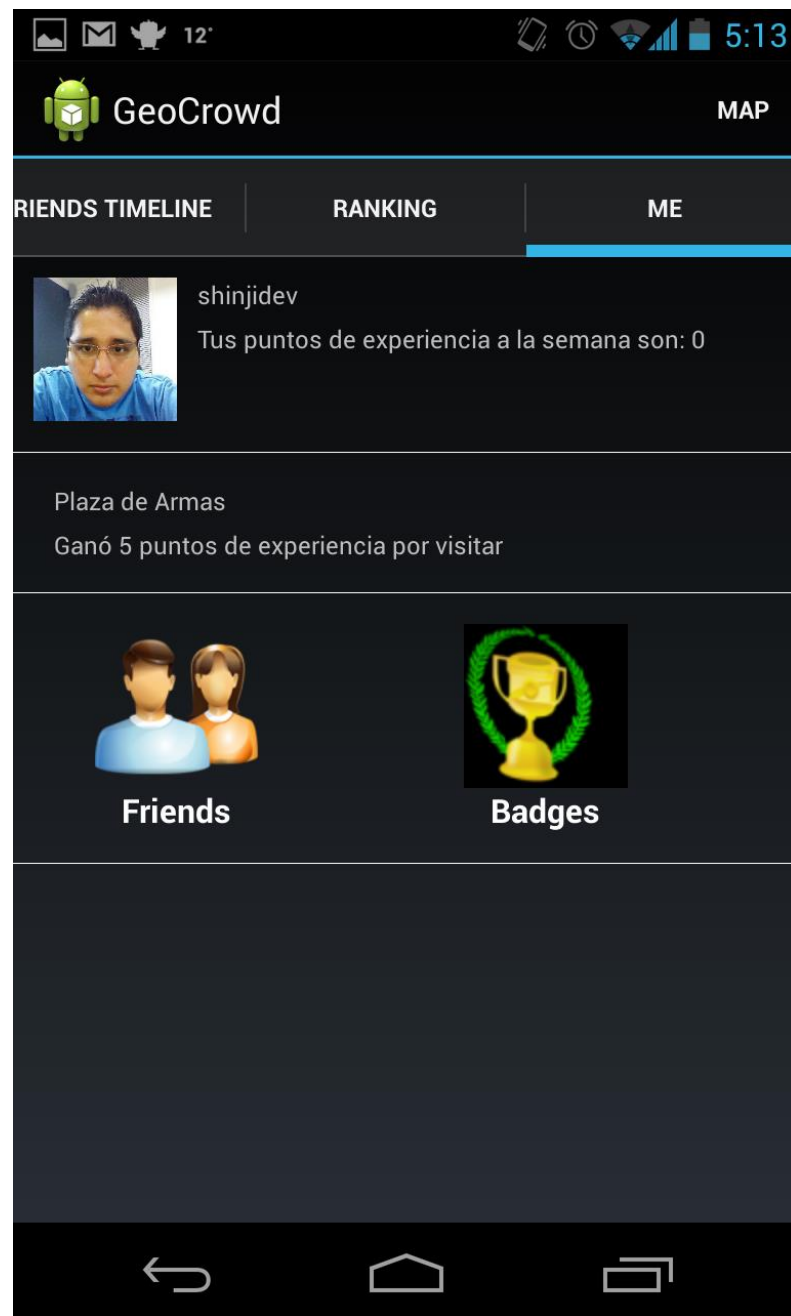


Figura 20: Interfaz de perfil de usuario

4.3.5 Interfaz de búsqueda de lugares cercanos

Esta interfaz es una de las esenciales donde se puede ver los lugares cercanos a donde uno se encuentra, dando un poco de *context-awareness* al usuario, además de esta forma puede ver si es que el lugar en que se encuentra está en el listado de lugares, sino en caso contrario puede agregarlo usando la aplicación mediante el botón ubicado en la parte superior derecha.

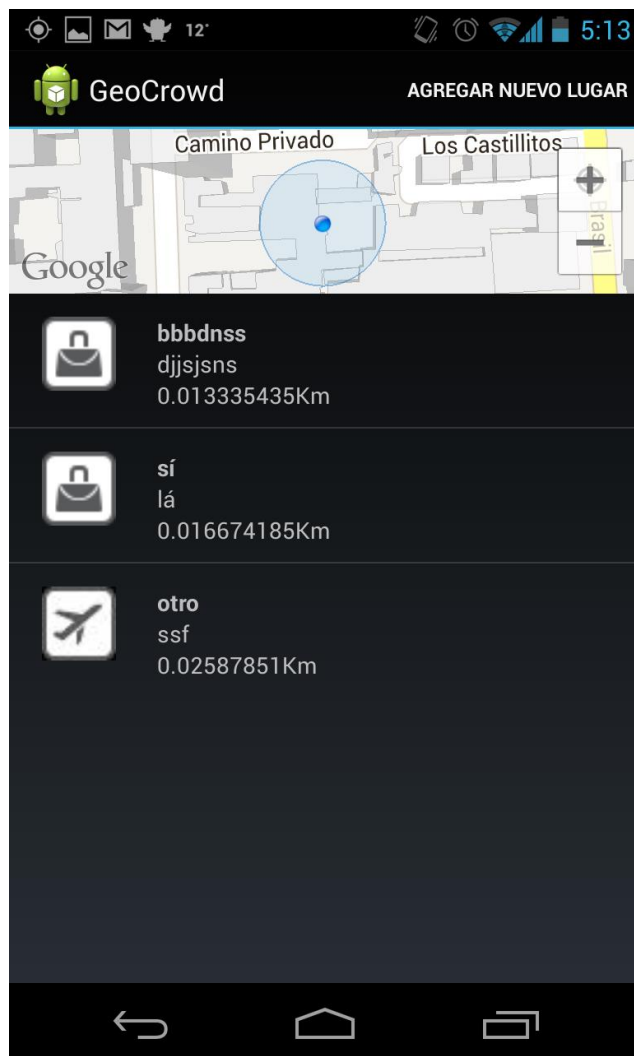


Figura 21: Interfaz de búsqueda de lugares cercanos

4.3.6 Interfaz de agregar nuevo lugar

El objetivo de esta interfaz es el esencial para que el usuario pueda agregar un nuevo lugar que no haya encontrado previamente en la búsqueda de lugares cercanos. Se le da la mayor sencillez posible para que agregue los datos necesarios, así como alguna foto que represente al lugar, adicionalmente puede agregarle unos *tags* que representen la información relacionada al lugar que agregan.

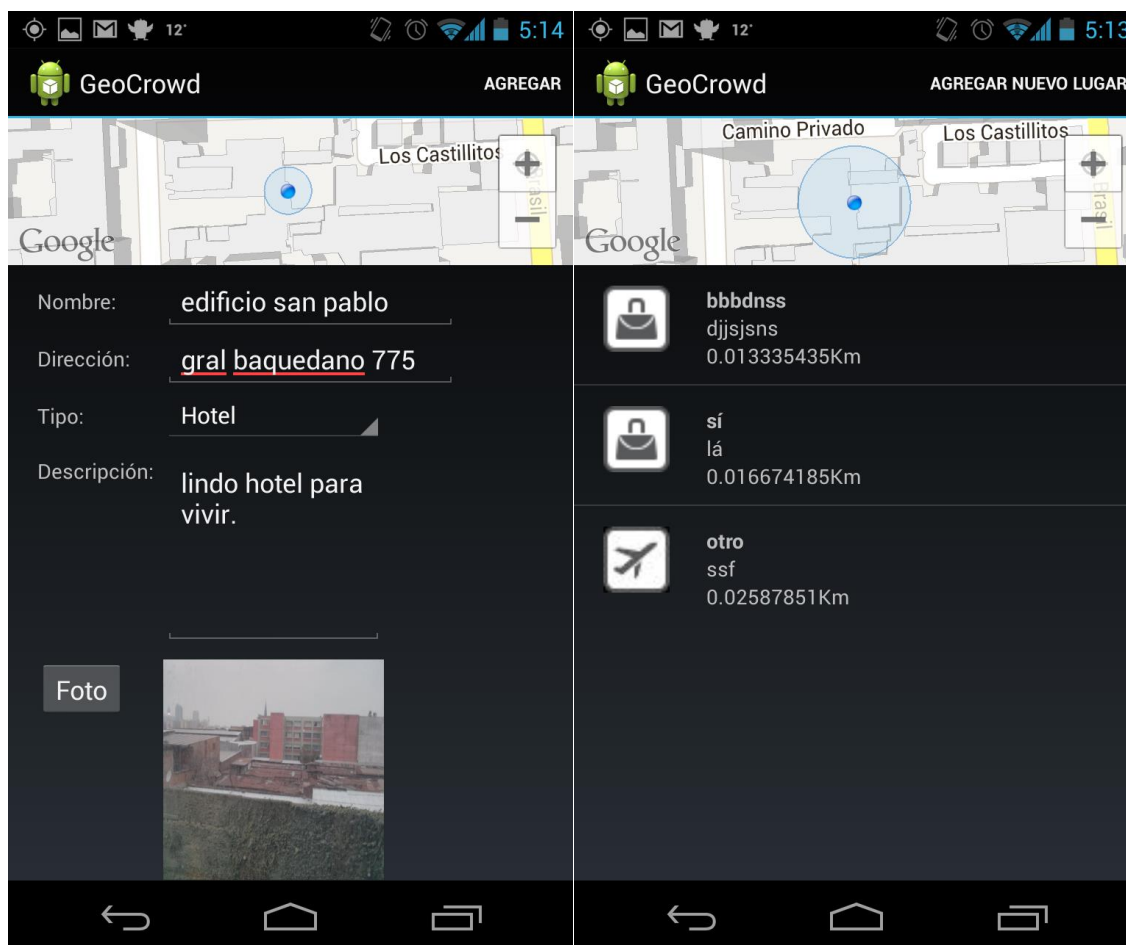


Figura 22: Interfaz de agregar nuevo lugar

4.3.7 Interfaz de información del lugar

Esta pantalla da una información completa, de forma sencilla, respecto al lugar que está visitando el usuario. Se muestran datos como el título, descripción, dirección, foto del lugar que está visitando. La priorización de esta información como inicial es que el usuario vea si es que la información del lugar es correcta, si en caso llega a encontrar algo incorrecto en esta información, se puede proceder a proponer editar la información del lugar. Además de esta información se muestran datos adicionales como los *tags* del lugar, ofertas que puede haber, así como los comentarios que hay en éste.

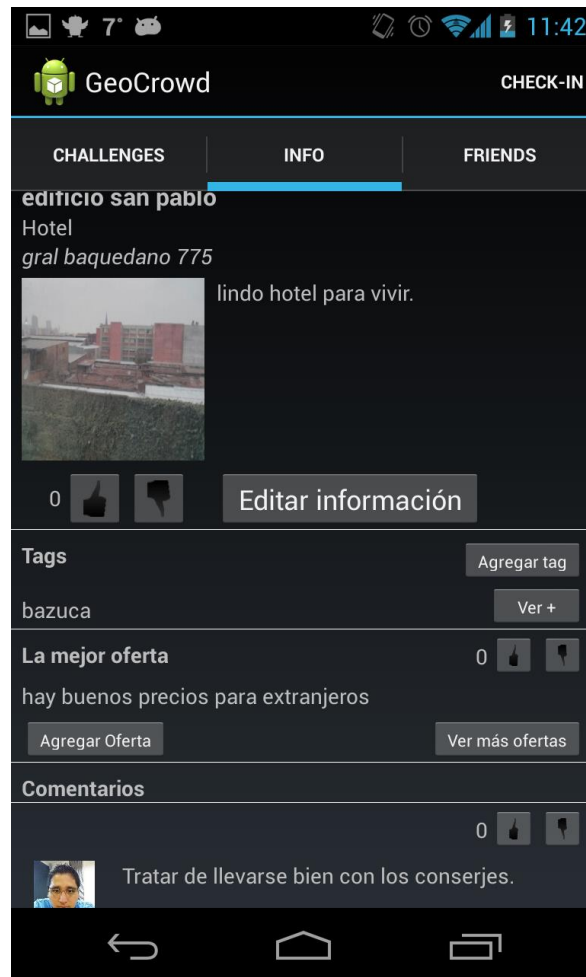


Figura 23: Interfaz de información de lugar

4.3.8 Interfaz de votación de tags

Esta pantalla es para algo elemental respecto a la información relacionada con el lugar que está visitando, lo cual es la votación a favor o en contra de esta información, que es llamado *tag*. Como se puede ver, se trata de incentivar al usuario a votar por cada *tag*, de tal forma que reciba algo a cambio, en este caso son puntos de experiencia lo cual le sirve para subir puestos en el ranking general.

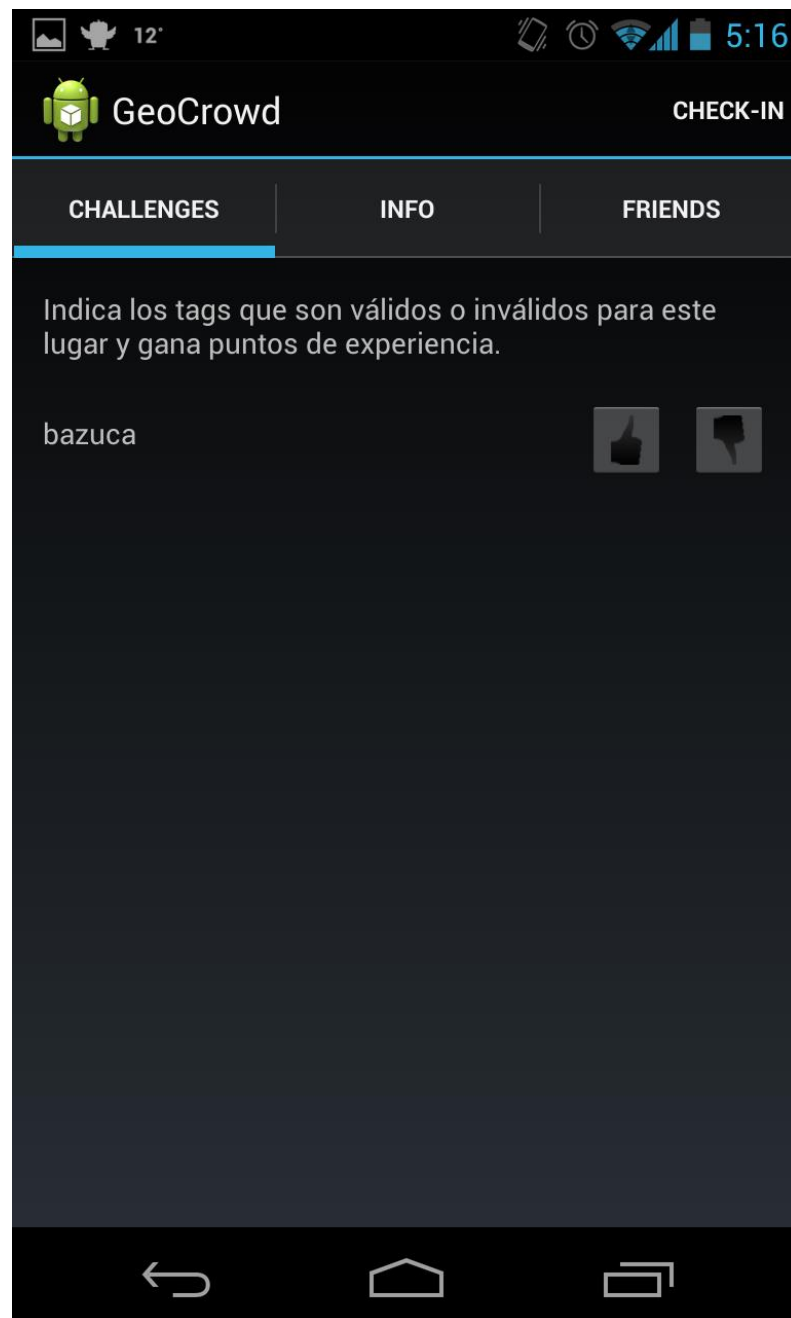


Figura 24: Interfaz de votación de tags

4.3.9 Interfaz de propuesta de modificación del lugar

Esta pantalla es la que da la posibilidad al usuario de poder proponer algún cambio al lugar, sea el título, descripción y/o dirección. Incluso, el usuario podría indicar que simplemente el lugar no existe o ya no se encuentra ahí, lo cual es útil para establecimientos que han cerrado o se han mudado de lugar, o incluso para eventos post-catástrofe donde muchos lugares suelen caer y desaparecer.

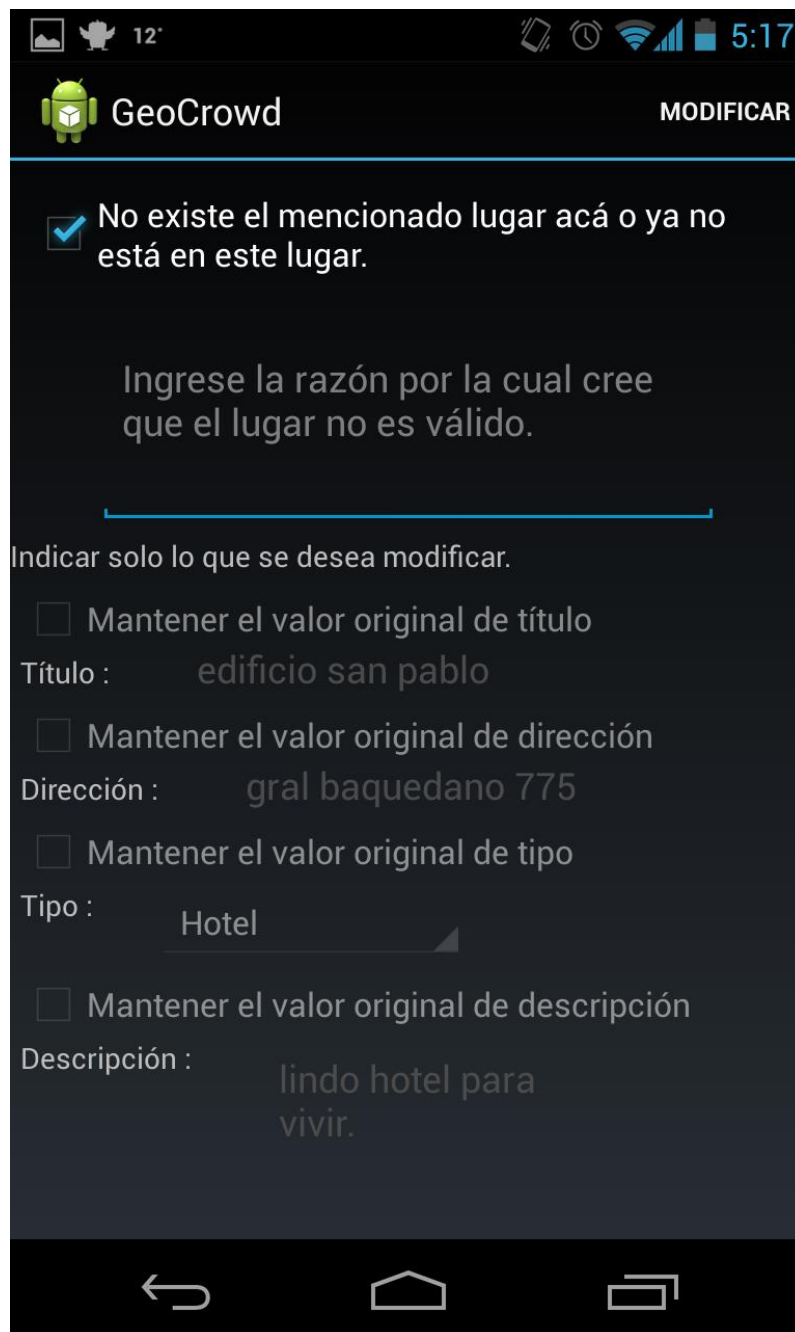


Figura 25: Interfaz de propuesta de modificación de información de lugar

Después de enviar una solicitud de edición de la información del lugar, cada vez que un usuario ingrese a ver la información del lugar verá un botón de “validar”, el cual le mostrará una pantalla donde sólo se verán los datos que se desean cambiar del lugar, de acuerdo al usuario elegirá votar a favor o en contra de la solicitud de edición de información.

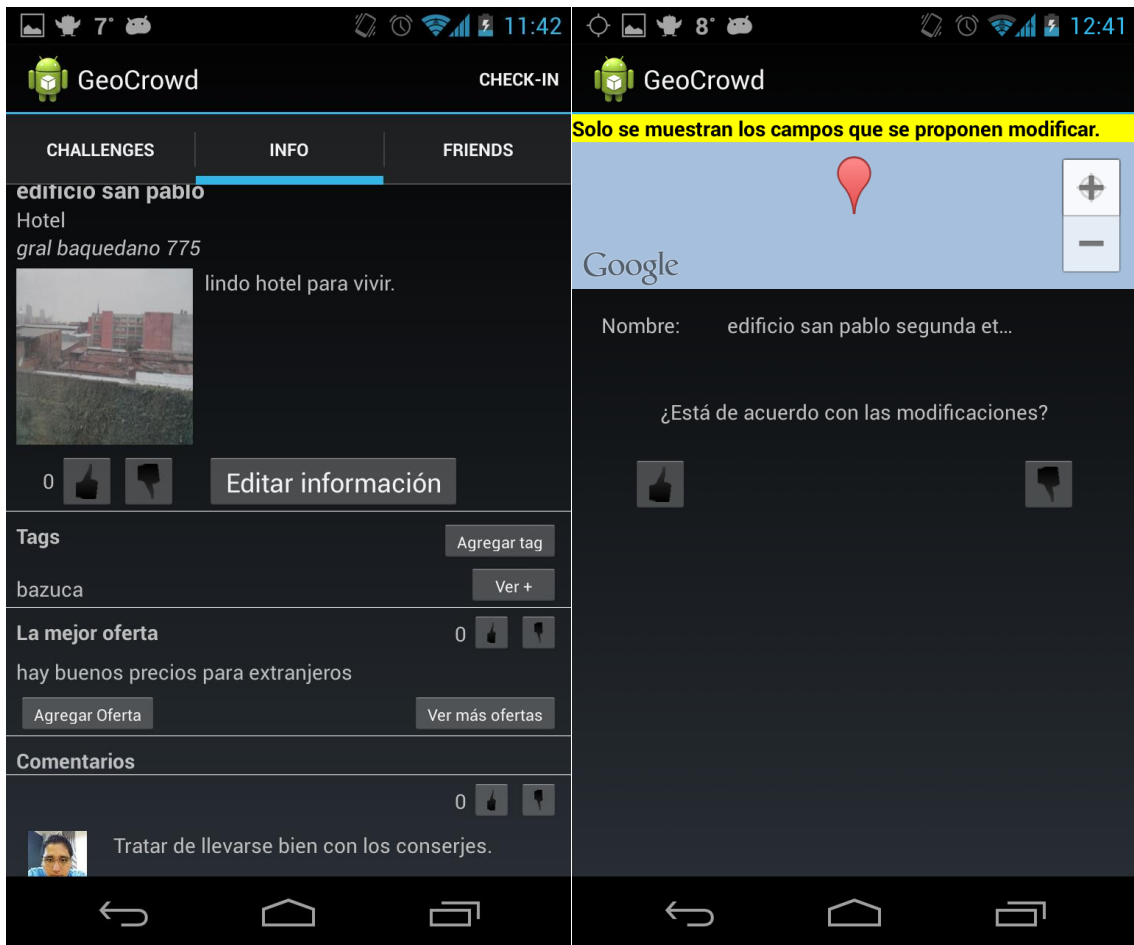


Figura 26: Interfaz de validación de propuesta de modificación de información de lugar

4.3.10 Interfaz de listado de comentarios

El usuario puede visualizar en esta pantalla el listado completo de comentarios del lugar, de esta forma podrá dar su *like* o *dislike* a estos según vea necesario. Además de permitirle agregar algún comentario adicional.

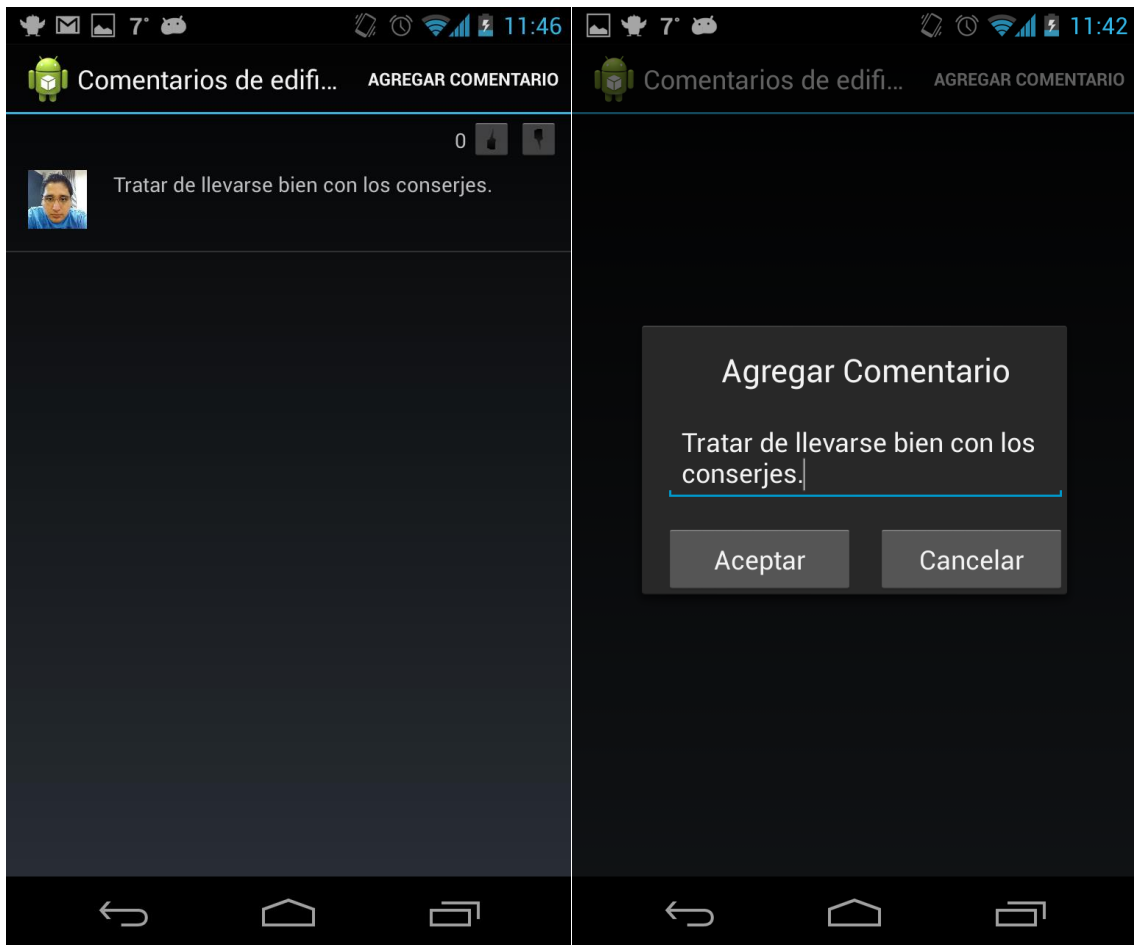


Figura 27: Interfaz de listado y agregado de comentarios

4.3.11 Interfaz de listado de ofertas

En esta pantalla se le da al usuario la posibilidad de ver el listado de ofertas del lugar, así como de poder agregar alguna oferta propia. Así también, puede dar *like* o *dislike* de estas ofertas según vea necesario.

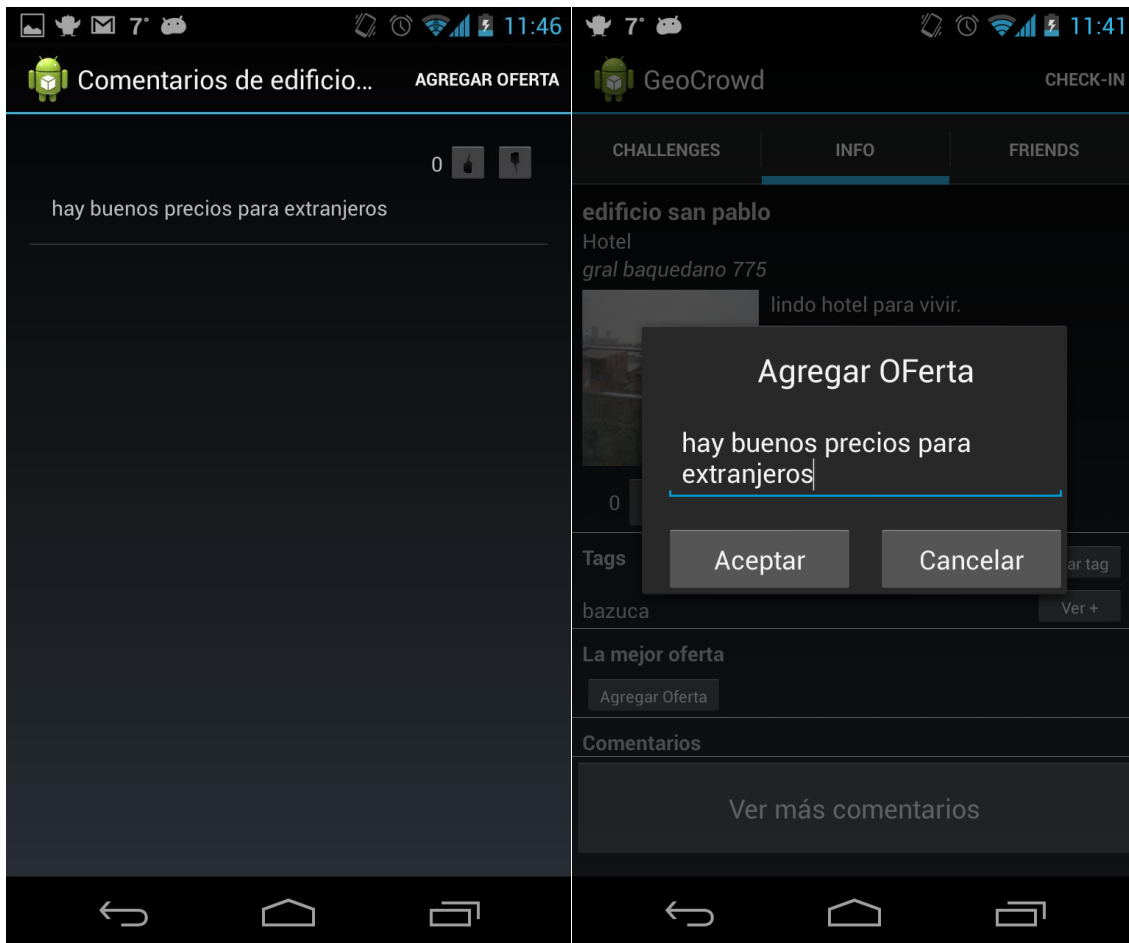


Figura 28: Interfaz de listado y agregado de ofertas

4.3.12 Interfaz de listado de tags

Esta pantalla da un vistazo a todos los *tags* con los que cuenta el lugar, es simplemente informativa, debido a que la cantidad de *tags* puede ser muy grande y no caber en la pantalla principal de información del lugar. Esta pantalla además da la opción de poder agregar un *tag* al lugar, dando recomendaciones basadas en los *tags* ya existentes.

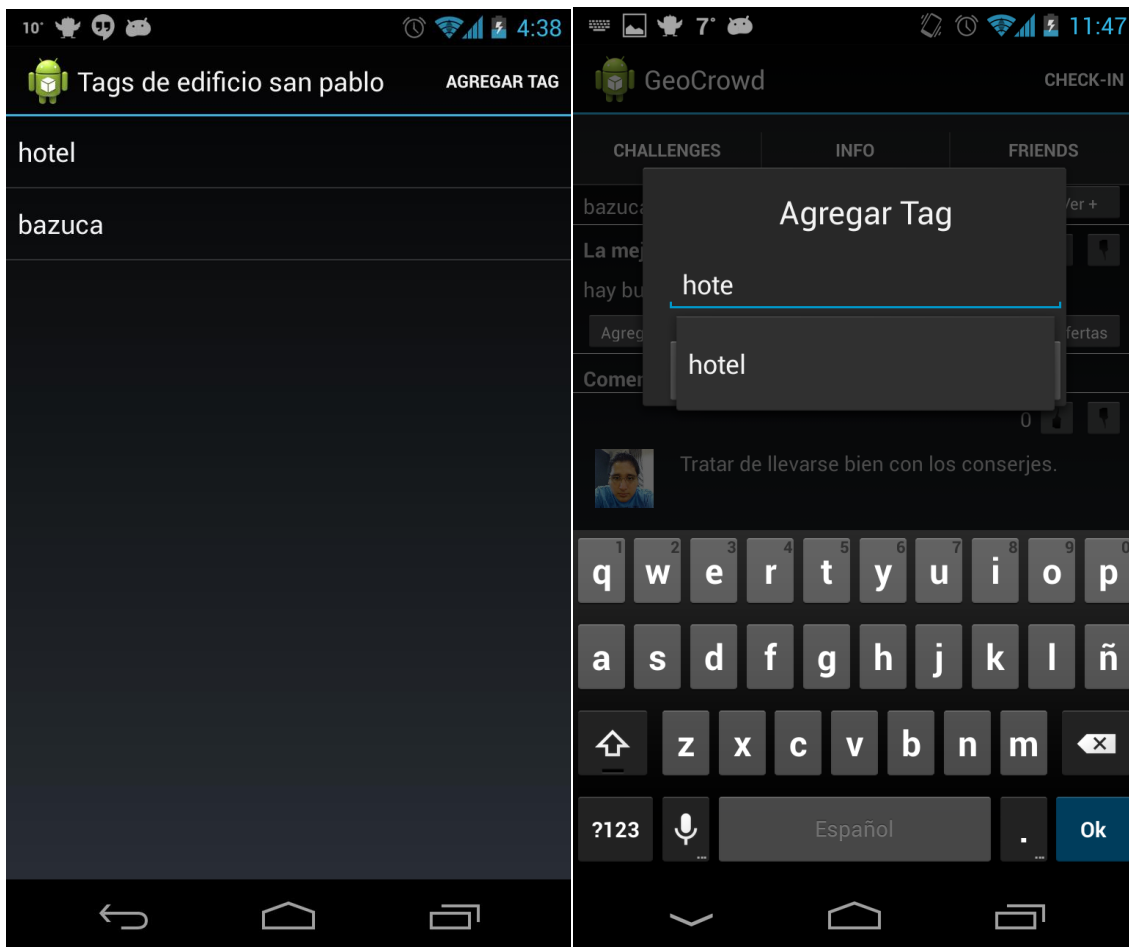


Figura 29: Interfaz de listado y agregado de tags

4.4 Interfaces API / mantención

En esta sección se mostrarán unas cuantas interfaces que permiten mostrar el administrador de los datos de la API, estas funcionalidades tienen como objetivo establecer una especie de configuración inicial a los datos del servidor sobre los cuales trabajará la API, así como también poder observar los datos que se van agregando. Este administrador puede realizar cambios del tipo CRUD en la base de datos del sistema, sin embargo se van a mostrar sólo las más relevantes para este documento.

4.4.1 Interfaz de registro de usuarios

Debido a la exclusividad de uso de esta aplicación, se realizó el registro de usuarios de forma manual mediante una interfaz que permite ingresar los datos básicos de estos, así como la personalización mediante una foto.

The image shows a user registration form for Geocrowd. The form is titled "Geocrowd" and includes the following fields and options:

- First Name**: A text input field.
- Last Name**: A text input field.
- Username**: A text input field.
- E-mail**: A text input field.
- Password**: A text input field.
- Gender**: Two radio button options: "Male" (selected) and "Female".
- Upload a Picture**: A button labeled "Seleccionar archivo" followed by the text "No se ha seleccionado ningún archivo".
- Join**: A button at the bottom of the form.

Figura 30: Interfaz de registro de usuarios

4.4.2 Interfaz de mantención de usuario

La idea de esta interfaz de administración es simplemente ver los datos del usuario, como sus KP, *experience points*, entre otros. Además de esto, se utilizó para configurar valores iniciales a los KP de los usuarios que se registraron.

Change User profile

User:	<input type="text" value="shinjidev"/> <input type="button" value="v"/> <input type="button" value="+"/>
Gender:	<input type="text" value="Male"/> <input type="button" value="v"/>
Profile Picture:	Currently: photos/user_profile/eaad88ce7cb04d08bfc0fc775fc503d6.jpg <input type="button" value="Clear"/> Change: <input type="button" value="Seleccionar archivo"/> No se ha seleccionado ningún archivo
Kharma points:	<input type="text" value="1734"/>
Experience points:	1970
Experience points in a week:	180
Number validated places:	19
Number validated tags:	28
Number visited places:	6
Number visited places in a week:	0
<input type="button" value="✖ Delete"/>	

Figura 31: Interfaz de mantención de usuario

4.4.3 Interfaz de actividad de usuario

Esta es la pantalla más interesante de la administración, debido a que nos permite ver la actividad completa, a detalle, de cada usuario que seleccionemos, de tal forma que podemos ver lo importante como los lugares que agregaron, los comentarios que hicieron, lugares visitados, ofertas agregadas, votaciones por lugares, comentarios u ofertas, entre otros.

Comments
Comment: lindo lugar
Details (Show)
Comment: comentario
Details (Show)
Comment: comentario
Details (Show)
Comment: Tratar de llevarse bien con los conserjes.
Details (Show)
+ Add another Comment
Comment Votes
+ Add another Comment Vote
Offers
Offer: 2 x 1 en churros
Details (Show)
Offer: oferta
Details (Show)
Offer: hay buenos precios para extranjeros
Details (Show)
+ Add another Offer
Offer Votes
+ Add another Offer Vote
Places
Place: Café la favorita
Details (Show)

Figura 32: Interfaz de actividad de usuario

5. Resultados obtenidos

En este capítulo se describe el escenario sobre el cual se realizó la experimentación de la heurística, haciendo uso de la solución desarrollada. Durante el desarrollo del experimento se fueron ajustando los valores configurables de la heurística para obtener mejores resultados.

5.1 Escenario de experimentación

Para llevar a cabo la experimentación se necesitó de usuarios de prueba para la aplicación cliente desarrollada, por lo cual estos usuarios deberían de tener un *smartphone* con SO Android, además de contar con plan de datos (3G) en su equipo.

De preferencia, para este experimento se seleccionó a los usuarios analizando de forma bastante superflua su uso de algunas aplicaciones conocidas como Google Maps, Foursquare o Facebook Places, así como su rutina y cuánto uso hacen de éstas en el día a día. Además de esto se limitó la cantidad de usuarios a 34, debido a que el proceso de validación se iba a realizar de forma manual, lo cual es un trabajo que toma un gran tiempo.

Los usuarios, por preferencia de la experimentación, viven en Lima, tienen entre 18 y 26 años de edad y algunas veces visitan ciertos lugares del Perú, por lo cual son considerados unos altos generadores de información para este experimento. Estos fueron conseguidos de forma voluntaria, únicamente se solicitó lo indicado párrafos arriba para que sean parte de las pruebas. Para este caso no se hizo sesgo alguno respecto a la profesión, nivel de educación, edad o sexo, únicamente se eligió a quienes cumplieron con los requerimientos definidos.

5.2 Proceso de Experimentación

Para el proceso de experimentación, se necesitó de una base de datos inicial, la cual tuviera datos ya verificados, así como que los lugares se encuentren con la información suficiente como son la geo-posición y los *tags* relacionados. Debido a esto se utilizó la base de datos de Limapoints.pe, por lo que se contó con un escenario inicial de 276 lugares, así como unos 2331 *tags* relacionados a los lugares. Cabe resaltar que esta base de datos inicial que se formó pudo haber sido creada a partir de diversas fuentes de datos como Foursquare, Wikipedia, entre otros. Así también, se pudo haber tomado no solo lugares que conocemos que son reales, sino también agregar

lugares que nosotros sabemos que no existen y esperamos que los usuarios corroboren esto.

Durante el proceso de experimentación, se estuvo indicando a los usuarios la forma como deberían de usar la aplicación, así como también solucionando diversos *bugs* de la aplicación para mantenerla estable en el flujo ideal. Respecto al uso de la aplicación, se les comentó que pueden optar libremente ser usuarios bienintencionados o malintencionados, esto para ver la efectividad de la heurística. Además, se monitoreó el avance de los usuarios mediante una página de administración de lado servidor que permitía ver las últimas acciones realizadas por los usuarios, para ver si estuvieron agregando lugares, *tags*, comentarios, u otros, tal como se puede ver en la siguiente imagen:



Figura 33: Interfaz de seguimiento de actividades de usuario

El tiempo de experimentación útil que se estableció fue de 21 días, donde los usuarios aprendieron de la aplicación, agregaron lugares, comentarios, publicaron ofertas, entre todas las acciones posibles. Así también, gracias al monitoreo, se pudo ajustar durante el tiempo el valor de configuración que permite aceptar un lugar, votación u otro como válido.

5.3 Datos obtenidos

A continuación se pueden ver unas tablas mostrando los resultados que hacen contraste con los datos iniciales con los que empezó el experimento, así como los datos finales.

En esta tabla (Tabla 2), la cual consideramos una de las más importantes, podemos ver los lugares que fueron agregados, así como la diferencia entre los lugares que existían inicialmente en la base de datos, así como los finales.

Tabla 2. Resultado de lugares validados

LUGARES		
Inicio	Lugares iniciales	276
Final	Total de lugares	587
	Lugares válidos	549
	Lugares no validados	9
	Lugares inválidos	29
	Total de lugares agregados	311
	Máximo de votos por lugar	13
	Mínimo de votos por lugar	1

Tal como se puede ver, en menos de un mes, se llegaron a agregar más lugares de los que había inicialmente, duplicando el valor inicial que había en nuestra base de datos. Además, cabe resaltar, que la mayor parte de estos lugares llegaron a ser validados, teniendo una baja cuota de lugares inválidos.

En la tabla de *tags*, tuvimos un resultado idéntico al anterior (ver Tabla 3).

Tabla 3. Resultado de tags validados

TAGS		
Inicio	Tags iniciales	2331
Final	Total de tags	3316
	Tags válidos	3290
	Tags no validados	5
	Tags inválidos	21
	Total de tags agregados	985
	Máximo de votos por tag	10
	Mínimo de votos por tag	1

La cantidad de *tags*, debido a que el número inicial era muy alto, no se pudo duplicar como en el caso de lugares, sin embargo, se agregaron cerca de 1000 *tags* a casi 300 lugares, lo cual en promedio son casi 3 *tags* por lugar, generando diversa información relacionada a estos. Así también, se resalta que la tasa de *tags* que no llegaron a validarse (*tags* no validados), es ínfima comparada con el total de *tags* agregados.

Respecto a las modificaciones que resultaron del experimento (ver Tabla 4):

Tabla 4. Resultado de modificaciones validadas

MODIFICACIONES LUGARES		
Final	Total de modificaciones	20
	Modificación de no existencia	3
	Modificación de algún campo	17
	Validados de forma correcta	16
	No validados	4
	Inválidos	0
	Máximo de votos por modificación	4
	Mínimo de votos por modificación	1

Se puede considerar, que hubo un total de 16 modificaciones que fueron validadas de las 20 que se propusieron, mientras que las otras 4 no se llegaron a validar. Para este experimento no se pudo llegar a un caso de modificación no válida, suponemos que debe ser a que no hubo un "uso indebido" de esta funcionalidad de la aplicación. A continuación se muestran las tablas de *likes/dislikes* de comentarios (Tabla 5) y de ofertas (Tabla 6).

Tabla 5. Resultado de votos en comentarios

COMENTARIOS	80
Likes/Dislikes	140

Tabla 6. Resultado de votos en ofertas

OFERTAS	36
Likes/Dislikes	40

5.4 Validación de la hipótesis

Para la validación de la hipótesis se consideró inicialmente, debido a la naturaleza de la heurística misma, que el proceso se iba a realizar de forma manual. Es decir, la validación se realizó punto por punto, mediante la búsqueda en diversos recursos de Internet, como mapas, buscadores, entre otros. A continuación se puede ver los recursos utilizados para la validación de los resultados:

- *Foursquare*.- Se llegaron a encontrar algunos lugares utilizando la búsqueda sobre esta plataforma, considerando la cercanía que tenía con el lugar buscado.

- *Google Maps.*- Se realizó una búsqueda mediante latitud y longitud, considerando un factor de cercanía, en los casos en que la diferencia de distancias era grande, se consideraba el lugar como no válido.
- *Street View.*- Basado en el sistema anterior, pero debido a que Google Maps no tiene todos los lugares marcados, se procedió a usar Street View para validar que el lugar realmente exista, mediante una vista de 360°.
- *Tripadvisor.*- Algunos lugares visitados por los usuarios eran hoteles o sitios turísticos, por lo que Tripadvisor fue una herramienta ideal, ya que se hizo la comparación entre la dirección otorgada por esta herramienta, combinada con Google Maps, para encontrar la exactitud de las coordenadas geolocalizadas otorgadas por nuestra solución.
- *Google.*- Debido a que algunos lugares no se encontraban en las anteriores fuentes, se procedió a realizar la búsqueda del lugar en Google mediante su nombre, topándonos muchas veces con las páginas de los mismos sitios donde se indica la dirección de estos, así como de *reviews* que realizan algunas personas sobre los lugares que visitaron, colocando datos de dónde encontrarlos. Esto combinado con Google Maps nos permitió validar casi el 30% de lugares.
- *Visita local.*- De los lugares agregados, muchos de estos fueron en locales que no tenían dirección o bien no tienen resultado en los buscadores, o en ciertos casos se encontraban dentro de otro local. Para la validación de este pequeño porcentaje se realizó la visita física al lugar, para de esta forma validar que el lugar existiera, haciendo un marcado en el mapa de dónde se encontraba la persona y comparándolo con las coordenadas del resultado a validar.

El resultado obtenido respecto a los lugares válidos validados mediante los recursos antes mencionados fue el siguiente (Tabla 7):

Tabla 7. Validación de resultados obtenidos en lugares validados con heurística

Bien validados	253	92,67%
Mal validados	13	4,76%
No posible validación	5	1,83%
Repetidos	2	0,73%

Tal como se puede apreciar, de los 273 lugares que se agregaron como válidos en la herramienta, un poco más del 92% resultó ser realmente válido. Aquellos que no fue posible la validación fueron debido a que representan un sector demasiado grande como para poder validar, o no hay un recurso que permita validar esto, pero cabe resaltar que representan menos del 2%. Así también, los mal validados fueron representados así debido a que la distancia de coordenadas entre el lugar real y el lugar a validar se encontraban muy lejos, por lo cual se decidió considerar como no válido (únicamente se llegó a dar en la provincia de Cuzco). Este error de validación del usuario más que considerarlo como error intencional, puede llegar a ser considerado un error de la precisión del GPS del dispositivo con el que contaba el usuario. Respecto a los repetidos, debido a la investigación y corroboración con los usuarios, se pudo ver que fue un error de usuario al momento de agregar un nuevo lugar a la herramienta.

Además de hacer la validación de los lugares que fueron tomados como válidos, se realizó la validación de aquellos que fueron tomados como no válidos y aquellos que no se pudieron validar. Respecto a los que fueron tomados como no válidos, se pudo corroborar que efectivamente el 100% de estos lugares no existía, ni se encontraba en una cercanía como para tomarlo como error de geo-posicionamiento.

Sobre los lugares que no se pudieron validar, se encontró que el 100% de estos eran válidos, sin embargo no tuvieron el quórum suficiente como para validarlos, por lo que hace suponer cierta necesidad de quórum mínimo cercano al lugar como para validarlo.

6. Discusión

Uno de los objetivos de esta tesis era la definición de la heurística para la validación de información, de esta forma podíamos determinar al final de ésta si es que era viable o no aplicarla en este contexto. A pesar de eso, cabe recalcar alguna otra posible solución que estuvo en mente aplicar en lugar de la presentada, así también presentar algunos factores que contribuyeron al resultado obtenido presentado en el capítulo anterior.

Uno de los factores más importantes del problema a resolver es validar que el usuario realmente se encuentre en el lugar que dice que se encuentra, una alternativa de solución a lo que se hizo en la tesis pudo haber sido el utilizar los TPM (Trusted Platform Module), mencionado en el capítulo 2, el cual permite evitar la manipulación de datos que son enviados por el Smartphone. Este hardware adicional hubiese ayudado muy bien para filtrado y validación de datos en este experimento, sin embargo, la viabilidad de obtención de muchos usuarios, así como la compra de estos para las pruebas, no era factible para este caso.

Otra solución alternativa hubiese sido el agregar, al planteamiento actual, una estrategia de *gamification* llamada motivación extrínseca, es decir, poner en juego un premio o premios a los usuarios con el objetivo de que puedan generar mayor información e interactúen más con la herramienta desarrollada. Esta alternativa se veía muy tentadora en el aspecto de participación y validación de información más fluida, pero posiblemente podía traernos el abuso de usuarios bienintencionados, los cuales, con tal de ser ganadores, hubiesen actuado de buena fe durante toda la etapa de pruebas y de esa forma no hubiésemos podido tener suficientes casos de usuarios malintencionados que validar.

Respecto a los factores que contribuyeron al resultado obtenido, comentaremos sobre los usuarios de prueba, que tal como se mencionó en el capítulo anterior, fueron seleccionados debido a que poseían plan de datos (3G), celulares con sistema operativo Android, son asiduos al uso de aplicaciones como Google Maps, Foursquare o Facebook, predisposición de apoyar en la prueba y viven en Lima, Perú. Un dato importante a resaltar, es que la edad de estos varía entre los 18 y 26 años. La razón de resaltar los datos de edad y localización es importante, pues el resultado obtenido, así como la cantidad de datos generados, pudo haber sido influenciado por esto.

Se conoce que el uso de aplicaciones en *smartphones* varía de acuerdo al rango etario de los usuarios, así como de la ubicación geográfica de los usuarios. Si es que se hubiese elegido otro rango etario y localización, donde el uso de aplicaciones, como la desarrollada para validar la heurística, es muy bajo, lo más probable es que el número acciones y datos generados sobre ésta hubiesen sido menor al que se obtuvo realmente. Al haber menos interacción con la aplicación, hubiese generado menos votos en las validaciones que se realizaron, lo cual hubiese conllevado a que se tuviera una mayor cantidad de ítems no validados, alterando, de esta forma, los resultados porcentuales obtenidos originalmente.

Analizando estos puntos, podemos discutir un poco sobre la robustez de los resultados. Obviamente, esto va depender, en un alto grado, del contexto en que se dio prueba a la heurística. Pero también hay que tomar en cuenta que la heurística, debido a la forma como se planteó, independiente del contexto en que se ejecute, podría llegar a obtener menos ítems validados o sin validar, pero, en lo posible, sin dar falsos positivos, es decir, sin que se indiquen ítems validados incorrectamente. Así también, la parametrización de la heurística con distintos umbrales permite que estos sean ajustados según la necesidad, contexto y objetivo a validar.

La mención de esta corta discusión invita a realizar estas pruebas con fines de investigación y encontrar otros posibles caminos que pudieran ayudar a obtener mejores resultados, sea con alternativas nuevas o comprobando la fiabilidad de la solución planteada en diferentes rangos etarios.

7. Conclusiones y trabajo a futuro

La validación de información, como área de investigación, ha ido evolucionando en paralelo con la cantidad de información nueva que se genera en el mundo del internet, esto debido a que los contextos y tipos de información se diversifican cada vez más. Por lo cual muchos métodos ya aplicados y validados en un contexto y con un tipo de información ya definidos no se han probado en todas las variedades, de contextos y tipos de información, o mixturas que estos generan. Siendo una de estas variedades de información, la georreferenciada.

La validación de la información georreferenciada ha ido encontrando ciertas metodologías propuestas para resolver esto. Sin embargo, en esta tesis proponemos una heurística diferente basada en crowdsourcing y computación social, la cual permitió validar la información georreferenciada con un alto grado de confiabilidad. Esta metodología no había sido probada en un contexto como el ya indicado.

Durante esta tesis se desarrolló una herramienta que nos permitió validar nuestra heurística de validación de información georreferenciada, cumpliendo con los objetivos planteados, así como también validar la hipótesis inicialmente planteada.

Los resultados de la evaluación de la heurística nos demostraron la gran posibilidad de usar una validación de información basada en el uso de crowdsourcing y computación social. La calidad de datos obtenida con esta evaluación, en especial la cantidad de datos que fue aceptada y rechazada, nos permiten ver que la gente participa del proceso de crowdsourcing con buena fe, con tal de obtener a cambio una fuente de información georreferenciada confiable.

A pesar de que los resultados obtenidos aún son insuficientes para sacar conclusiones definitivas respecto a la eficacia de la heurística aplicada, podemos afirmar, de acuerdo a los resultados obtenidos, que el uso de la heurística de validación nos generó una base de datos georreferenciada con un porcentaje de datos muy pequeño con validación errónea. Lo cual nos da un alto grado de confiabilidad de la heurística diseñada en esta tesis.

Pese a que el porcentaje de error fue bajo, se piensa a futuro mejorar la heurística utilizada para la validación, permitiendo utilizar alguna

herramienta que priorice alguna otra variable no considerada en la actual, o utilizar un mayor ponderado en alguna que se esté utilizando actualmente. Así también, mejorar el nivel participativo de los usuarios de la validación, no únicamente incrementando la cantidad de participantes, sino también dejar de tomar al usuario como un ente pasivo, sino poder hacerlo más activo y participativo, de esta forma evitaríamos en lo más posible aquellos lugares que no tuvieron validación por falta de participantes.

Otro punto a probar a futuro es el aplicar la heurística en otros contextos que no sea necesariamente el de lugares georreferenciados, observar el comportamiento de ésta en diferentes contextos puede ayudar a utilizarlo no como solución absoluta, sino como base para validar otros ambientes y así tener extensiones, modificaciones y/o mejoras para la heurística misma.

8. Bibliografía

[Abowd00] Gregory D. Abowd and Elizabeth D. Mynatt, "Charting past, present, and future research in ubiquitous computing," *ACM Trans. Comput.-Human Interaction*, vol. 7, pp. 29–58, March 2000.

[Abowd99] Gregory. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggle, "Towards a better understanding of context and context-awareness,". In *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing (HUC '99)*. London, UK: Springer-Verlag, pp. 304–307. 1999.

[Adomavicius05] Gediminas Adomavicius, Ramesh Sankaranarayanan, Shahan Sen, Alexander Tuzhilin. "Incorporating contextual information in recommender systems using a multidimensional approach". *ACM Transactions on Information Systems (TOIS)* 23(1), pp. 103-145, January 2005.

[Ahn08] Luis von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, Manuel Blum. "reCAPTCHA: Human-Based Character Recognition via Web Security Measures", *ScienceMan*, vol. 321, August 2008.

[Alt11] Florian Alt, Alireaza Sahami Shirazi, Albrecht Schmidt, URs Kramer, Zahid Nawaz. "Location-based Crowdsourcing: Extending Crowdsourcing to the Real World". *Proceedings of the 2nd International Workshop on Ubiquitous crowdsourcing (UbiCrowd 11)*. Beijing, China, 2011.

[BBVA12] BBVA. "Gamificación: el negocio de la diversión". *Innovation Edge*, Septiembre 2012.

[Bobadilla13] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez. "Recommender systems survey". *Knowledge-Based Systems* 46, pp. 109-132, July 2013.

[Burke02] Robin Burke. "Hybrid recommender systems: Survey and experiments". *User modeling and User-Adapted Interaction* 12(4), pp. 331-370, November 2002.

[Chacha14] <http://www.chacha.com/>. Última visita: 8 de Octubre de 2014.

[Chen09] Kuan-Ta Chen, Chen-Chi Wu, Yu-Chun Chang, Chin-Laung Lei. "A Crowdsourcable QoE Evaluation Framework for Multimedia Content".

Proceedings of the 17th ACM International Conference on Multimedia (MM 09). Beijing Hotel, Beijing, China, 2009.

[Celery14] <http://www.celeryproject.org/>. Última visita: 16 de Diciembre de 2014.

[Crowdflower14] <http://crowdflower.com/>. Última visita: 8 de Octubre de 2014.

[Deci85] Deci, E. L., and Ryan, R. M. Intrinsic motivation and self-determination in human behavior. New York: Plenum Press, 1985.

[Doan11] Doan, A., Ramakrishnan, R. and Halevy, A. Y. "Crowdsourcing systems on the World-Wide Web". Communications of the ACM 54(4), pp. 86-96, April 2011.

[Donovan13] Siobhan O'Donovan, James Gain, Patrick Marais. "A case study in the gamification of a university-level games development course". Proceedings of the SAICSIT 2013. New York, USA, 2013.

[Dua09] Akshay Dua, Nirupama Bulusu, Wu-Chang Feng, Wen Hu. "Towards trustworthy participatory sensing". Proceedings of the 4th USENIX Conference on Hot Topics in Security (HotSec'09). Berkeley, CA, USA, 2009.

[Eagle09] Nathan Eagle. "txteagle: Mobile Crowdsourcing". Proceedings of the 3rd International Conference on Internationalization, Design and Global Development (IDGD '09). Held as part of HCI International 2009. Springer-Verlag Berlin, Heidelberg, 2009.

[GameMarketing12] Game Marketing. "Estudio 2012. Gamificación expectativas y grado de adopción en España.". Estudio publicado el 9 de abril de 2012. <http://www.gamkt.com/estudio-2012/Estudio-2012-Gamificacion-Spanish-Version.pdf>. Última visita: 8 de Octubre de 2014.

[Gamificacion14] <http://www.gamificacion.com/que-es-la-gamificacion>. Última visita: 8 de Octubre de 2014.

[Gears11] Deborah Ann Gears. "Wiki Behavior in the Workplace: Emotional Aspects of Content Development". Journal of Management and Marketing Research, 2011.

[Gears13] Deborah Gears, Karen Braun. "Gamification in Business: Designing Motivating Solutions to Problem Situations". Proceedings of ACM CHI'13, Paris, France, April 2013.

[Gilbert10] Peter Gilbert, Landon P. Cox, Jaeyeson Jung, David Wetherall. "Toward trustworthy mobile sensing". Proceedings of the HotMobile 2010, New York, USA, pp31-36, 2010.

[Goldman09] Jeffrey Goldman, Katie Shilton, Jeff Burke, Deborah Estrin, Mark Hansen, Nithya Ramanathan, Sasank Reddy, Vids Samanta, Mani Srivastava, Ruth West. "Participatory Sensing. A citizen-powered approach to illuminating the patterns that shape our world". White Paper, May 2009.

[Gonzalez12] A. Gonzalez. "Comparando la funcionalidad y la usabilidad de dos aplicaciones basado en un proceso de negocios de generación de noticias". Trabajo de titulación de ingeniería civil informática. Departamento de Ingeniería Informática, Universidad de Santiago de Chile, 2012.

[Kazai11] Gabriella Kazai, Jaap Kamps, Natasa Milic-Frayling. "Worker Types and Personality Traits in Crowdsourcing Relevance Labels". Proceedings of the 20th ACM International Conference on Information and knowledge management (CIKM 11). Glasgow, Scotland, UK, 2011.

[Mashhadi11] Afra J. Mashhadi, Licia Capra. "Quality Control for Real-time Ubiquitous Crowdsourcing". Proceedings of the 2nd International Workshop on Ubiquitous Crowdsourcing (UbiCrowd 11). Beijing, China, 2011.

[Meloni11] Wanda Meloni. "Gamification - LEVEL 1". Proceedings of the Gamification Summit 2011. San Francisco, California, US, 2011.

[Nginx14] <http://www.nginx.com>. Última visita: 16 de Diciembre de 2014.

[Palomino12] M. Palomino. "Un framework para presentar y experimentar con sistemas de recomendación en la Web 2.0.". Tesis de Magíster en Informática, Departamento de Ingeniería Informática, Universidad de Santiago de Chile, 2012.

[Perera13] Charith Perera, Arkady Zaslavsky, Peter Christen, Dimitrios Georgakopoulos. "Context Aware Computing for the Internet of Things". IEEE Communications Surveys & Tutorials Journal, 2013. <http://arxiv.org/pdf/1305.0982v1.pdf>. Última visita: 8 de Octubre de 2014.

[Postgis14] <http://www.postgis.net>. Última visita: 16 de Diciembre de 2014.

[Publi14] <http://www.publimetro.cl/nota/cronica/juego-de-la-pdi-el-caso-de-laura-es-una-de-las-app-mas-descargadas/xIQmgd!yDtDIPs5SSA1E/> . Última visita: 8 de Octubre de 2014.

[Recaptcha14] <http://www.google.com/recaptcha/learnmore>. Última visita: 8 de Octubre de 2014.

[Reiss00] Reiss, S. Who am I? The 16 basic desires that motivate our actions and define our personality. New York: Penguin Publishing, 2000.

[Restmocker14] <https://github.com/gnrfan/restmock>. Última visita: 16 de Diciembre de 2014.

[Ryan00] Richard M. Ryan, Edward L. Deci. "Self-Determination Theory and the Facilitation of Intrinsic Motivation, Social Development, and Well-Being". American Psychologist 55(1), pp. 68-78, January 2000.

[Streetmap14] OpenStreet Map. <http://www.openstreetmap.org/>. Última visita: 8 de Octubre de 2014.

[Sundmaeker10] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelffle, "Vision and challenges for realising the Internet of things". European Commission Information Society and Media, Technical Report, March 2010.

[Tareen10] Babar Khan Tareen, Jae-won Lee, Sang-goo Lee. "Synergy: A workbench for collaborative filtering algorithms on user interaction data". Proceedings of the 2010 ACM International Workshop on User Data Interoperability in the Social Web, Hong Kong, China, 2010.

[Thom12] Jennifer Thom, David Millen, Joan DiMicco. "Removing gamification from an Enterprise SNS". Proceeding of the 2012 ACM Conference on Computer Supported Cooperative Work, New York, USA, 2012.

[Turk14] <https://www.mturk.com>. Última visita: 8 de Octubre de 2014.

[VanPelt12] Chris Van Pelt, Alex Sorokin. "Designing a Scalable Crowdsourcing Platform". Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data (SIGMOD 12). Scottsdale, Arizona, USA, 2012.

[Vasquez14] Rodrigo Vásquez. "Modelo de Representación y Construcción de Sistemas de Recomendación para Interacciones en la Web 2.0.". Tesis de

Magíster en Informática, Departamento de Ingeniería Informática, Universidad de Santiago de Chile.

[Victor11] Patricia Victor, Martine De Cock, Chris Cornelis. "Trust and Recommendations". Recommender System Handbook, pp. 645-675, 2011.

[Wikipedia14] <http://www.wikipedia.org/>. Última visita: 8 de Octubre de 2014.

[Yan09] Tingxin Yan, Matt Marzilli, Ryan Holmes, Deepak Ganesan, Mark Corner. "Demo Abstract: mCrowd - A platform for Mobile Crowdsourcing". Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (Sensys 09). Berkeley, California, USA, 2009.

[Zichermann11] Gabe Zichermann, Christopher Cunningham. "Gamification by Design. Implementing Game Mechanics in Web and Mobile Apps". O'Reilly Media. Agosto 2011.

[Zhu12] Shaojian Zhu, Shaun Kane, Jinjuan Feng, Andrew Sears. "A crowdsourcing quality control model for tasks distributed in parallel". Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems - Extended Abstracts (CHI EA 12). Austin, Texas, USA, 2012.

[Zombie14] <https://www.zombiesrungame.com/> . Última visita: 8 de Octubre de 2014.