

Arquitectura (40)				Modelo de Dominio (35)				Persistencia (25)		Final
A (10)	B (10)	C (5)	D (15)	A (20)	B (10)	C (5)	D (5)	A (15)	B (10)	

Apellido y Nombre:

Condiciones de aprobación: Para aprobar debe sumar como mínimo 60 puntos y no menos del 50 % en cada sección.

Contexto

ArgenPropi Social es una plataforma web orientada a la publicación y gestión de propiedades en alquiler y venta. En los últimos años, el crecimiento de los alquileres temporarios y la influencia de las reseñas en redes sociales han cobrado un papel central en la decisión de búsqueda y recomendación de inmuebles. Por eso, la plataforma debe integrar esta información para enriquecer la experiencia del usuario y mejorar la relevancia de los resultados de búsqueda.



Inmuebles

Casas o departamentos para alquilar y/o vender, con todo el detalle de ubicación, ambientes, fotos, etc.

Publicaciones

Oferta de alquiler o venta del inmueble, con todos los datos relevantes para su búsqueda: datos del inmueble, disponibilidad, formas de pago, etc. Existen dos tipos de publicaciones conceptualmente, la interna, que es la que almacenamos en nuestra base de datos, y la externa, que consiste básicamente en replicar la información de dicha publicación en alguna red social.

Publicación externa

Relaciona una publicación interna con los datos de una publicación en una red social. Tiene la red social, el identificador, el título, la descripción, fecha de creación, cantidad de comentarios, re-publicaciones, visualizaciones y "likes". Cada publicación interna tiene seleccionado con qué redes sociales se debe crear una publicación externa, no tiene por qué usar todas siempre.

Cambios en las publicaciones internas

Cada vez que en el "Editor" se da de alta o modifica una publicación, dicho cambio debe ser impactado en las publicaciones de las redes sociales, modificando su título o contenido.

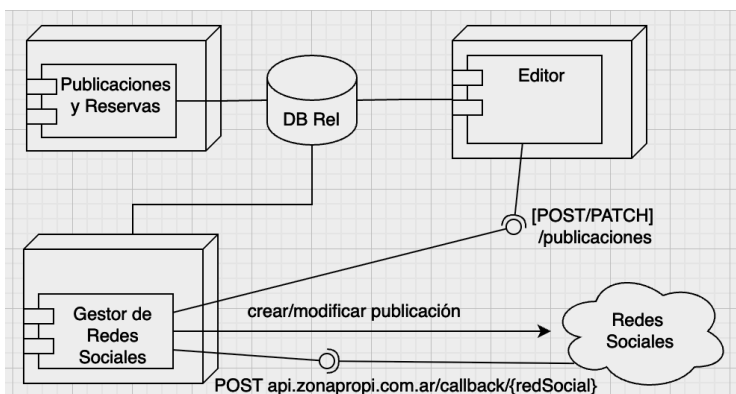
Obtención de información de las publicaciones externas

Las interacciones de las publicaciones en redes sociales deben ser registradas para optimizar los resultados de búsqueda en la plataforma. Para ello, al momento de crear una publicación, el sistema se suscribe a un servicio de notificaciones de cambios. Este servicio envía periódicamente la información actualizada (cada *N* minutos) mediante un webhook a una URL configurable.

Búsquedas

Las mismas se realizan sobre las publicaciones internas, sobre todos los parámetros, y con el agregado de estar ordenadas por "puntuación social", un valor calculado sobre la información obtenida en el punto anterior.

Arquitectura actual



La arquitectura actual del sistema se compone de 3 componentes, integrados a través de una única base de datos relacional. Los agentes inmobiliarios gestionan las propiedades y las publicaciones a través del componente "Editor" y los potenciales compradores / inquilinos, buscan, consultan las publicaciones a través del componente del componente "Publicaciones y reservas", **ambas son aplicaciones web render server / cliente liviano**. Y finalmente, en los últimos años, creció mucho la importancia y el uso de las redes sociales, tenemos el componente "Gestor de redes sociales".

Punto 1 – Arquitectura (40 puntos)

Hace menos de un mes se incorporó la funcionalidad de alquileres temporarios, lo que provocó un aumento significativo en el tráfico y en el volumen de uso de la aplicación, llegando en algunos momentos a multiplicarse por diez o más. Lo cual trajo varios problemas, que se enumeran a continuación. Teniendo como base la arquitectura **completa**, justifique por qué y cómo se resuelven los problemas. (Tip: diagrama/s es **mucho** mejor que texto)

- A. **(10 puntos)** Aumentaron notablemente los tiempos de respuesta de todos los componentes, en particular cuando se reciben muchas búsquedas, la latencia de un buen porcentaje de los pedidos puede tardar varios segundos. Cuando se relevó por qué sucede esto, se ve que los 3(tres) componentes están escalando horizontalmente de forma correcta y no parece explicar el problema.
- B. **(10 puntos)** Cuando el “Gestor de redes sociales” recibe las actualizaciones, si existe algún fallo en el procesamiento (falla la app, la base de datos, la conexión, etc), actualmente se está perdiendo la actualización (que luego se usa para calcular el puntaje). Por otro lado, las redes sociales cobran por cada reintento realizado.
- C. **(5 Puntos)** Se está obteniendo una mala percepción del usuario durante la ejecución de la reserva particularmente el que alquila temporariamente. Se detectó, que esto se debe a que el flujo que debe seguir requiere varios controles, cambios de pantalla, refrescar o cruzar información y muchas veces contacto con los dueños de la publicación, y genera una considerable demora en la operación. Si establecemos que el flujo de negocio es correcto, ¿qué podemos hacer para mejorar la experiencia de usuario?
- D. **(15 puntos)** Nueva Arquitectura Propuesta. No puede modificar las responsabilidades de los componentes. Comunique la arquitectura con un diagrama de despliegue y todo lo que considere necesario para comprender su funcionamiento.

Punto 2 – Modelado de dominio (35 puntos)

La implementación actual del “Gestor de Redes Sociales” se basa en dos scripts PHP que procesan las llamadas correspondientes. Sin embargo, su creciente complejidad ha derivado en un código extenso, difícil de interpretar y de mantener.

- A. **(20 Puntos)** Realice el re-diseño del componente “**Gestor de redes sociales**”, utilizando todos los recursos que considere necesarios. Documentar la solución utilizando diagramas UML (diagrama de clases obligatorio).
- B. **(10 puntos)** Con un diagrama de secuencia o pseudo código explique cómo se procesa una actualización de una publicación interna que proviene del “Editor”.
- C. **(5 puntos)** Explique cómo agregar una nueva red social para hacer publicaciones externas.

Observaciones

- Si se tiene más de una publicación externa, la puntuación social es el promedio
- Se tiene una clase Puntuador con el método puntaje(List<StatsPublicacion>), que calcula la “puntuación social”.
- Tanto las APIs para crear un post, como el body que envían para notificar las actualizaciones, son diferentes en todas las redes sociales, sin embargo requieren y proveen de la misma información.

Punto 3 – Datos (25 puntos)

- A. **(15 Puntos)** Armar la especificación usando un DER físico del componente “**Gestor de redes sociales**” a ser persistido en un ORM. Indicando las entidades, sus campos, claves primarias, las foráneas, cardinalidad, modalidad y las restricciones según corresponda.
- B. **(10 puntos)** Justificar
 - Qué elementos del modelo es necesario persistir y cuáles no
 - Cómo resolvió los impedance mismatches.
 - Las estructuras de datos que deban ser desnormalizadas, si corresponde.
 - Observaciones: persistir información que **no** es del componente es considerado un error grave

A continuación se muestra el **flujo actual** del sistema. Funciona CORRECTAMENTE, pero con la nueva arquitectura y el rediseño del componente puede mejorarlo.

