



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

Mejora de un sistema interactivo de información para estudiantes del DCC

PROPUESTA DE TEMA DE MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

Marcelo Becerra A.

PROFESOR GUÍA:
Jocelyn Simmonds

SANTIAGO DE CHILE
2021

1. Introducción

Actualmente en Chile, el que las organizaciones sigan la ruta de las transformaciones que la sociedad requiere y, que tengan un rol activo (liderazgo) en ellas, influye el devenir de la misma. [8]. En este mismo contexto, se destaca que la cantidad de alumnos del Departamento de las Ciencias de la Computación (DCC), ha aumentado considerablemente en los últimos 5 años, algo así como 200 alumnos aproximadamente [2], [3], [4]. Este aumento viene acompañado de una serie de desafíos, no solo en la docencia, sino en la logística de la escuela. Así mismo, un aumento tal, hace crecer la cantidad de dudas, incertidumbre y posibles problemas de los estudiantes.

Para resolver tales incertidumbres la mayoría preferiría un acceso directo a la información, es decir sin pasar por un intermediario. Por otro lado, contactar a un funcionario para resolver estas inquietudes se vuelve algo más frecuente, considerando que son estos los que reciben la mayor cantidad de peticiones sin solución desde los estudiantes [1].

Por esta razón, muchas organizaciones proveen mesas de ayuda, en diversas vías. En repetidos casos, las respuestas dependen de la preparación humana, así como de la cantidad de personal que se tiene. Por lo tanto, al aumentar la cantidad de clientes, aumenta también el costo para la organización de mantener un buen servicio. Usualmente este trabajo conlleva que muchas preguntas se repitan y, por lo tanto, se ocupa mucho tiempo en responder las mismas preguntas una y otra vez. Lo anterior puede generar varios problemas y, para organizaciones con escasos recursos, este puede no ser un escenario ideal [9].

Esta vía de ampliar personal, debe ser debidamente justificada, y consensuada [5]. Por otra parte, tampoco es lo que buscan los estudiantes. Y es justamente en este ambiente, que el proyecto Mesa de Ayuda Virtual se posiciona como una buena alternativa para asistir a los estudiantes y al mismo tiempo lograr moverse junto con una sociedad en cambio.

El proyecto Mesa de Ayuda Virtual, es un proyecto iniciado por Pablo Arancibia Barahona, egresado del DCC, que busca responder a la necesidad de mejorar el proceso de titulación de alumnos(as). Actualmente dividido en dos áreas: Un *bot*, con funcionalidades como preguntas frecuentes etiquetadas por proceso, capacidad feedback por el usuario y contactar a un asistente personal. La página web, cuenta con un *front-end* para alumnos y otro para administradores, en ellos se puede acceder a los procesos vigentes con sus preguntas frecuentes, a un chat directo con los estudiantes, categorizar preguntas, abrir y modificar procesos, entre otros.

Este proyecto, responde a las necesidades del estudiantado, pues fue diseñado a partir de la información levantada en el departamento y la consideración de las necesidades allí expuestas. Fue validado por la docencia y el centro de estudiantes y ha finalizado su primera etapa de desarrollo. Actualmente, soporta más procesos como prácticas profesionales.

Dentro de las conclusiones más relevantes del trabajo de Arancibia, destaca el hecho de que la mayoría de los encuestados, dijo que optaría por usar el servicio del *bot* de *Telegram*. Esto en un contexto dónde no hay un servicio ya implementado, es de suma relevancia, puesto que a pesar de que se podría generar la mejor herramienta, si esta no tiene público, no tendría razón de ser.

2. Situación Actual

El proyecto actualmente, aunque no esta desplegado, si está desarrollado y se han validado sus funcionalidades con usuarios. Y algunos actores se han comprometido con su desarrollo. Uno de los agentes relevantes en el proceso actual es el Centro de Alumnos del DCC (en adelante CADCC), ya que es uno de los más comprometidos con el proyecto. Y qué será fundamental en su primer arranque, así como en las validaciones. Posiblemente, será el principal contribuyente a la información.

Hay posibles participantes como académicos y funcionarios, que si bien han validado la propuesta, hoy no son parte del ecosistema que se busca generar. Esto, porque actualmente no consideran que la solución alivie su carga laboral, y por ende no tienen una motivación para integrarse al proyecto.

Tiene un modelo *API Rest*: el *back-end*, administra los datos en MongoDB, dentro de los cuales están los procesos (cómo titulación, memoria, prácticas, proyecto de Software), sus instancias y sus preguntas frecuentes, también maneja los usuarios con sus respectivos roles. El bot considera en su modelo de atención llegar a la información deseada en aproximadamente 3 clicks, permite enlazar la comunicación con un asistente en tiempo real, y permite al usuario dar feedback, sobre si la información fue útil o no. Ocupa *django channels* para enlazar la conversación con un asistente en la página web y *redis* para la memoria caché. La página web, permite acceder a toda la información categorizada por parte del alumno, mientras que a los administradores, les permite categorizar preguntas, administrar los procesos y sus instancias, agregar información y contestar a los alumnos de manera anónima.

Una de las principales oportunidades en relación a lo anterior y la vez una de las grandes restricciones, es la capacidad de hacer este sistema extensible. En ese sentido surge la pregunta ¿a dónde apuntar el desarrollo? Dado que el sistema no está desplegado, sería prudente integrar funcionalidades que apunte a favorecer la adopción de esta herramienta en el departamento. Pero surge la pregunta ¿cómo lograr esta adopción? ¿y cómo lograr que se mantenga en el tiempo? (*engagement*).

Dado los resultados de Arancibia, se concluye, que uno de los elementos que mejor se integra al flujo natural del alumnado es el *bot* [1]. Tanto por aceptación original descrita en su trabajo, como por la inserción y presencia natural que tienen estos servicios en la comunidad del DCC. A pesar de esto, hay ciertos elementos de en la solución actual, que pueden frenar la adopción del bot.

Primero, su código no es extensible, dado que no hay una implementación modular y bien estructurada, tal y como fue expresado por su creador, es funcional, sin embargo, con el objetivo de que el proyecto sea escalable en el tiempo, se plantea una modularización del actual código para así hacerlo mas legible y extensible. Hay que optimizar el manejo de tareas asíncronas como lo son los mensajes por múltiple usuarios en el bot. Se debe corregir la view (Django) del bot, porque el código presenta una carencia de entendibilidad, debido a que toda la lógica se concentra en un solo modulo, a pesar de la naturaleza de las interacciones del usuario.

La confianza en un sistema de información, es de suma relevancia [7]. En ese sentido, se

sabe que a medida que aumenta el número de traspasos de información, sobre todo entre agentes humanos, hay un riesgo de pérdida y modificación, tanto por las fallas propias de los canales de comunicación, como por la acción de los agentes involucrados. Actualmente, este sistema no tiene ninguna forma de generar información por si mismo, tampoco puede acceder a las fuentes primarias de información por su cuenta. Por lo tanto los encargados de guardar la información son vitales en la calidad de esta. Por otro lado las fuentes que los mismos usen, determinarán su veracidad, lo que afecta directamente su confiabilidad.

Debido a lo anterior, incluir actores que sean fuentes primarias de información es relevante y minimiza los cambios a la misma. Así como también es importante que se pueda verificar su fuente y vigencia.

Dentro de sus capacidades, el *bot* solo guarda información temporalmente, no guarda ningún tipo de preferencias del alumno, más allá de la interacción puntual. La retroalimentación corresponde solo a una evaluación simple del servicio, y no permite generar ningún otro tipo de información, como la calidad de la información o su relevancia real para el estudiante. Al ir más allá y examinar detenidamente el modelo de datos, se puede corroborar que no existe ningún tipo de *features* ajustadas a un usuario en particular.

La inclusión de un valor personal, en modelos de suscripción pagados es uno de los factores que define el *engagement*[6] (o la adopción por los usuarios). Aunque este sistema no es pagado, generar un buen modelo de suscripción es clave para darle un valor personal, que sería el gran diferenciador con soluciones existentes como una wiki o novedades en u-cursos, porque permitiría al usuario estar directamente relacionado con aquello que es importante para él, disminuye la interacción con información irrelevante, y mejora la experiencia general del usuario.

3. Objetivos

Objetivo General

El objetivo general es extender el sistema actual, agregando funcionalidades que permitan crear un modelo de suscripción, que sea: extensible, personalizado y confiable. El cual será actualizado principalmente por el CADCC, con miras de integrar otros actores posteriormente. Favoreciendo la adopción y continuidad de este servicio.

Objetivos Específicos

1. Analizar la solución existente, con el fin de identificar que modificaciones son relevantes. Tanto para que el modelo de datos soporte las nuevas funcionalidades y cómo se debe reestructurar el código.
2. Implementar y probar las modificaciones para asegurar la integridad del modelo de datos, del código y permitir su extensibilidad a través de la modularidad lógica.
3. Agregar features de confiabilidad que permitan: añadir información a las respuestas y mejorar el modelo de feedback del usuario, para que se pueda obtener más información como su vigencia.
4. Crear un modelo y funcionalidades de suscripción personalizadas: que contemplen una búsqueda personalizada de procesos, un elemento suscriptor, variables relevantes de notificar y un modelo de notificaciones y data del alumno, entre otras features relevantes para los alumnos.
5. Implementar las nuevas funcionalidades asegurando que sean aquellas relevantes para los alumnos.

Evaluación

1. Se pueden identificar en el modelo las nuevas entidades y relaciones que se corresponden con las funcionalidades descritas. puede asociar cada componente de la arquitectura lógica del bot.
2. El sistema soporta las funcionalidades fundamentales que ya presenta, se pueden hacer operaciones CRUD a través de los endpoints. Cada componente de la arquitectura lógica del bot tiene su representación en código.
3. Se puede ver la fuente de la respuesta en un mensaje, así como la última actualización de esta información. El usuario puede catalogar una respuesta como no-vigente.
4. Validar el modelo de suscripción con estudiantes, profesores y funcionarios a través de entrevistas y focus groups (de entre 5 a 10 personas). Con el foco en las funcionalidades, relevancia para el usuario y privacidad.
5. Validar la implementación de las nuevas funcionalidades, a través de una investigación con demos a estudiantes.

4. Solución Propuesta

Para poder llevar a cabo los objetivos anteriormente descritos, se propone agruparlos en x fases. Pensadas en la lógica que abarca cada uno de estos objetivos.

4.1. Análisis y modificaciones de la solución actual

Para asegurar la mantenibilidad y rescatar las funcionalidades críticas. Se estudiará en detalle la solución actual. Con esto se busca identificar todos aquellos aspectos relevantes de la solución actual, así como aquellos que necesiten una modificación, ya sea reestructuración o re-implementación. A partir de este proceso se diseñarán y usarán *tests* que sirvan para validar como el sistema actual responde a los objetivos originales y los planteados en esta propuesta.

Luego se implementarán: todas las modificaciones en el modelo de datos, para que admita las nuevas funcionalidades de confiabilidad, y algunas métricas que puedan ser relevantes y, las refactorizaciones en el código (del bot) para modularizar la solución actual y darle una estructura lógica, que permitan entender el sistema y hacerlo extensible.

luego con el conjunto de test diseñados se probará que las modificaciones no afectaron la funcionalidad original ya implementada. También se espera poder realizar test de los nuevos endpoints para las operaciones CRUD, que se requieran para las funcionalidades de confiabilidad.

Finalmente se espera poder obtener métricas de satisfacción que garanticen los objetivos uno al tres planteados para esta fase, desde el back-end del sistema.

4.2. Integración de nuevas capacidades

En primer lugar se buscará agregar a la interfaz del bot los elementos ya testeados de confiabilidad. Agregando a las respuestas, la fuente de la información y su ultima actualización. Se hace notar, que la fuente de la información no es quién pueda atender una consulta no resulta, ni quien esté disponible en el chat, manteniendo a estos actores anónimos. Y junto con esto, la capacidad del usuario de retroalimentar al sistema, permitiendo ayudar a establecer métricas de vigencia, reales, basadas en las calificaciones de los usuarios, así como proporcionar ayuda a los administradores del sistema, al poner atención a aquellos procesos que requieren realmente una actualización, lo que permite enfocar los esfuerzos.

Luego, se generará un modelo de subscripción, que permita buscar y seleccionar un proceso de forma individual, más específicamente una instancia a un cierto proceso, ya que los procesos como tesis, memorias, funcionan como clases de los objetos individuales, que serían los procesos vigentes cada semestre (ver [1]).

Se hace notar que esta búsqueda debe permitir generar modelos de subscripción no solo a procesos globales, sino que de forma más general, a información que sea relevante para el alumno, por ejemplo deadlines, requisitos, consecuencias, entre otros. Permitiendo de este modo, que

A partir de este proceso de búsqueda, no se debe mostrar las preguntas frecuentes, sino elementos que componen este proceso, y su flujo, de modo que el estudiante, pueda interactuar de forma personalizada con aquellos elementos, que considere que son relevantes para el, y no entregar únicamente una experiencia empaquetada que puede terminar generando información de bajo impacto para el usuario. A estos elementos se les denominó en los objetivos elemento subscriptor.

Es muy importante que estos elementos sean validados en su mayoría por usuarios, porque tanto su representación como sus desencadenantes, son diferentes en cada caso.

Así mismo, hay que generar un modelo de notificaciones, que se ajuste a estas necesidades, y que sea lo suficientemente simple como para poder extenderlo sin problemas. Es importante notar, que *Telegram* hay diferentes formas de “notificar” a un usuario, o de hacer cierta información presente, se puede realizar mediante nuevos mensajes, se pueden enviar mensajes que queden fijados en la parte superior del chat, así cómo se pueden editar mensajes anteriores, por lo que se puede actualizar el contenido de un mensaje. También, el formato, es muy amplio. Por lo tanto, lo que como un mensaje escrito puede ser mejor par aun caso, puede ser una imagen para otro. Por lo tanto hay que evaluar bien que es lo que está disponible al bot y la vez se ajusta las preferencias de los usuarios.

Finalmente se debe validar este modelo con usuarios, y entidades relevantes para hacer un ajuste, antes de comenzar con la etapa de implementación. Esta validaciones, debe contemplar todos los aspectos ya mencionados como nuevas funcionalidades, centrándose en la relevancia de cada una para del estudiante. Al mismo tiempo, el agregar nuevas funcionalidades y personalización que dependa de datos del usuario, pude levantar problemas de privacidad que deben ser revisados junto a los encuestados.

4.3. Implementación

La ultima fase es la implementación de las funcionalidades ya validadas. Este proceso busca integrar estas funcionalidades, tanto al *back-end* como al *front-end*. Por lo tanto nuevamente se requerirá de un alto grado de cuidado para no romper las funcionalidades ya implementadas.

Estas *features* deben ser testeadas. Además se incluirá un proceso de validación con los usuarios que incluya demos. Opcionalmente pero no depende del memorista, se plantea la posibilidad de incluir estas funcionalidades en un sistema de producción. Pero esto depende del CADCC y de la disponibilidad de sus miembros.

4.4. Documentación

Aunque la documentación no representa un desafío técnico elevado, se quiere destacar su rol, en la extensibilidad de un sistema, por lo que, se requiere tiempo, especial en cada iteración, para realizar una buena documentación del sistema.

4.5. Tecnologías

Si bien la decisión de qué tecnologías usar, puede verse modificada en un futuro, según los resultados del trabajo, en primera instancia se pretende continuar con el *setup* original del proyecto.

Dentro de los lenguajes a utilizar se destacan: Python, JavaScript y SQL. Estos están fuertemente ligados a las herramientas que se usan en los distintos aspectos del proyecto.

- **Servidor:** El servidor funciona principalmente en *django*, a través de la librería *django channels* se sincronizan los chats del administrador con el bot. También se pretende integrar *Celery* para el manejo de proceso asíncronos.
- **Motor de Base de datos:** La base de datos, inicialmente pensada para preguntas repetitivas y frecuentes, usa *MongoDB*, que como herramienta *NoSQL*, está especializada en este tipo de *requests*. También, puede que sea necesario agregar otro tipo de motores en el transcurso del proyecto, ya que hay varias áreas sobre todo auditoría, en las que el modelo, no se ajusta completamente al enfoque *NoSQL*.
- **Cache sesión usuario:** Se usa *Redis*, para administrar las conversaciones a través de *Django Channels*.
- **Página Web:** La página web, esta principalmente diseñada en *React*, más algunas librerías para facilitar el desarrollo de componentes visuales. Se destaca, que idealmente esta no se encuentra en el scope del proyecto.

Referencias

- [1] ARANCIBIA, PABLO IGNACIO ARANCIBIA BARAHONA: *Creación de un sistema interactivo de información para mejorar el proceso de titulación en el DCC*. Informe técnico, Universidad de Chile, Santiago de Chile, 2021.
- [2] CADCC: *Padrón 2016-2 Elecciones CaDCC 2017.xlsx*, 2016. https://www.u-cursos.cl/uchile/2008/0/COMCADCC/1/material_docente/detalle?id=1619365.
- [3] CaDCC: *Padrón Oficial Votaciones CaDCC 2019*, 2018. <https://www.cadcc.cl/tricel-padron-oficial-votaciones-cadcc-2019/>, visitado el 2021-05-03.
- [4] CADCC: *Elección directiva CaDCC 2021*, 2021.
- [5] Chile, Rectoría de la Universidad de: *Reglamento de Presupuesto - Universidad de Chile*, 2014. <https://www.uchile.cl/portal/presentacion/senado-universitario/reglamentos/reglamentos-aprobados-o-modificados-por-el-senado-universitario/104521/reglamento-de-presupuesto>, visitado el 2021-05-16.
- [6] Hansen, Elizabeth y Emily Goligoski: *Tow Center for Digital Journalism A Tow/Knight Report GUIDE TO AUDIENCE REVENUE AND ENGAGEMENT*. Informe técnico, 2018. <https://academiccommons.columbia.edu/doi/10.7916/D8BG410Whttps://doi.org/10.7916/D8BG410W>.
- [7] Marsh, Stephen, Stephen Marsh y Mark R. Dibben: *The Role of Trust in Information Science and Technology*. Annual Review of Information Science and Technology (ARIST), 37:465–98, 2003, ISSN 0066-4200.
- [8] Paz, María, Vásquez Pailaqueo, Romina Fernanda, Inostroza Naranjo y Hedy Acosta Antognoni: *Liderazgo transformacional: su impacto en la confianza organizacional, work engagement y desempeño laboral en trabajadores millennials en Chile*. Revista de Psicología, 30(1):1–17, mar 2021, ISSN 0719-0581. www.revistapsicologia.uchile.cl.
- [9] Thurman, David A, Jeffrey S Tracy y Christine M Mitchell: *Design of an Intelligent Web-Based Help Desk System*. Informe técnico. <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=337433F8ECB944A5FD6CC08B0191DE20?doi=10.1.1.34.1442&rep=rep1&type=pdf>.

5. Anexos

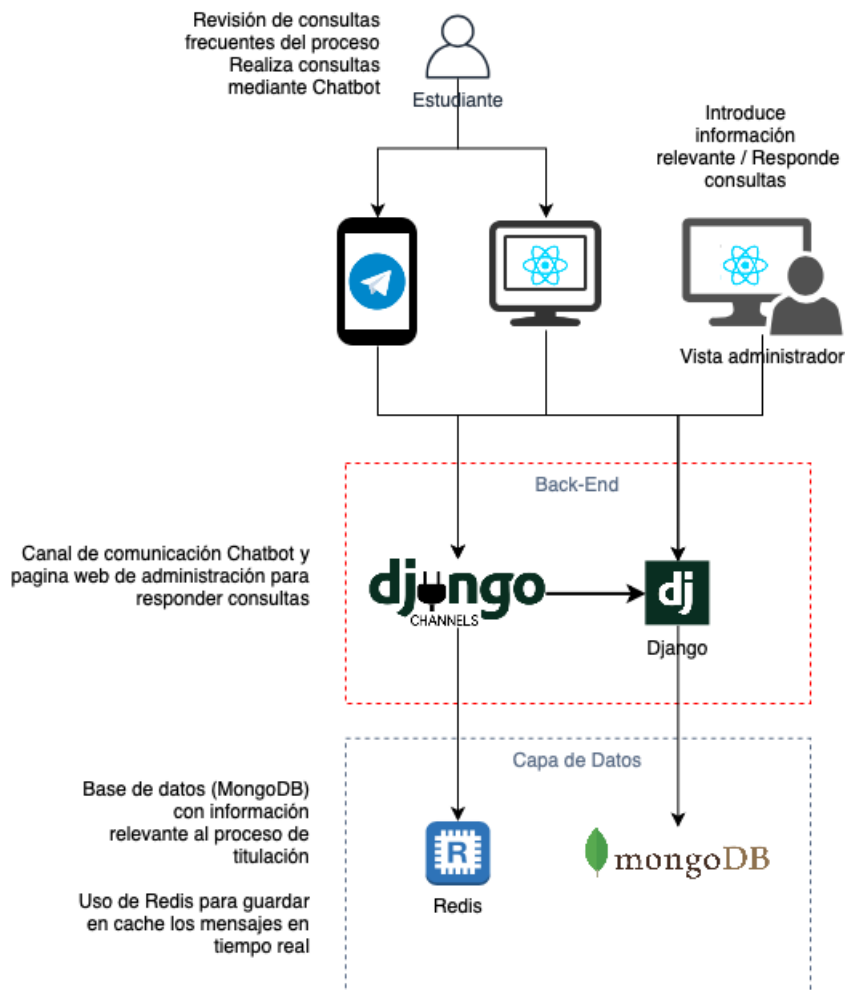


Figura 1: Diagramas de arquitectura del sistema Mesa de Ayuda Virtual

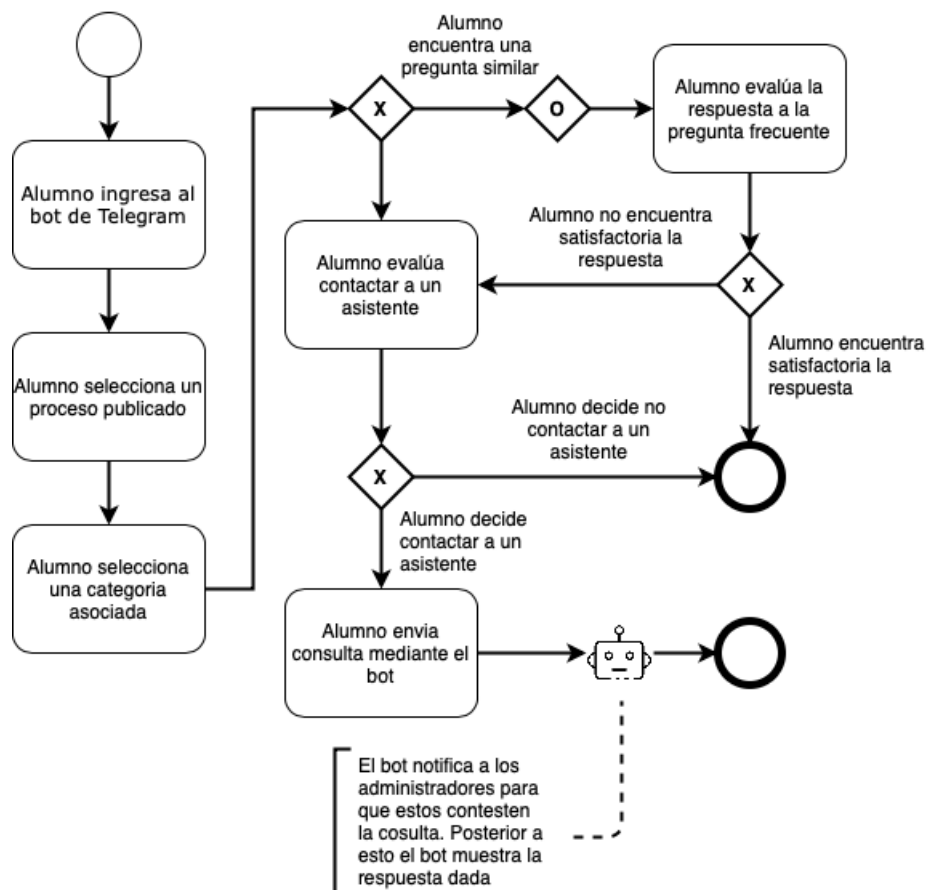


Figura 2: Diagrama flujo de la interacción con el usuario sistema Mesa de Ayuda Virtual

Base de Datos - Mesa de ayuda DCC

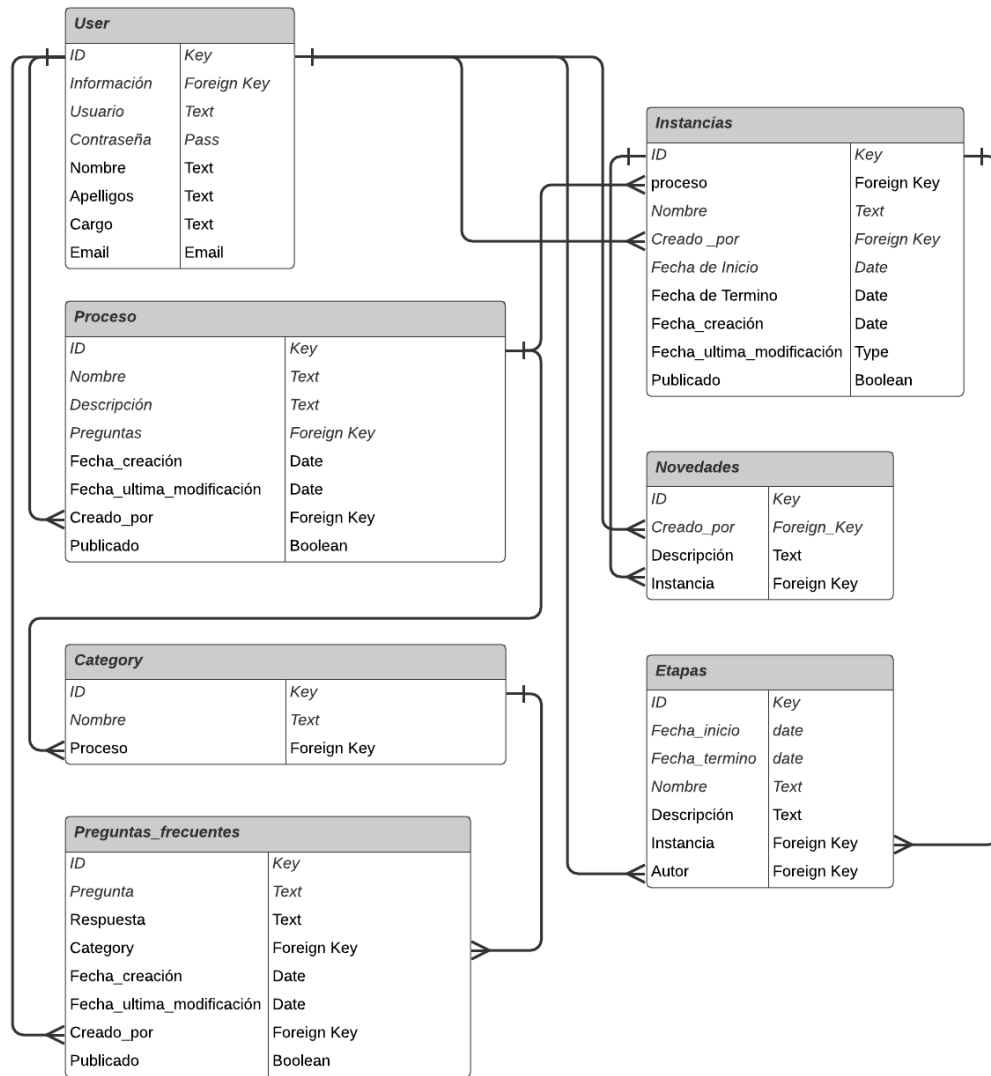


Figura 3: Modelo de datos actual del sistema Mesa de ayuda

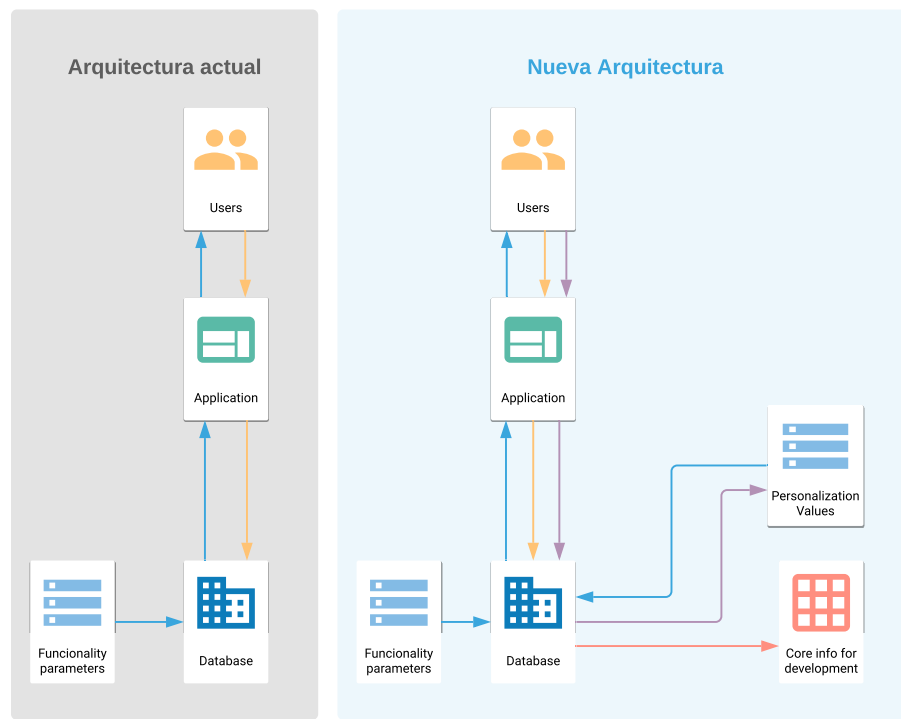


Figura 4: Propuesta de Arquitectura: En la figura las líneas azules denotan el flujo de información desde el sistema, las amarillas las peticiones puntuales del usuario, las moradas las subscripciones o contenido personalizado, y la roja la información valiosa del sistema para la extensión del mismo