



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

Mejora en de Bot de Telegram, para servicio de Mesa de Ayuda

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

Muchas organizaciones proveen mesas de ayuda, ya sea por vía telefónica, *email* o similares. En muchos de estos 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 [8].

La universidad es un caso modelo, de una organización que no puede disponer de recursos libremente cada vez que se necesite ampliar un servicio.

En este mismo contexto, se destaca que la cantidad de alumnos del D.C.C., 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. 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].

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 en que la organización sea confiable, lo cual por supuesto tendrá un impacto el devenir de la misma [7]. Por este motivo, el incluir alternativas que aseguren un buen servicio de información, puede resultar crítico para la universidad y el departamento. 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 D.C.C, que busca responder a la necesidad de mejorar el proceso de titulación de alumnos(as). Posteriormente dicho proyecto se diversificó, incluyendo más procesos, como las prácticas profesionales. 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.

Dentro de las conclusiones más relevantes del trabajo de Arancibia, destaca el hecho de que la mayoría de los encuestados, dijo que usaría el servicio"[1]. Esto en un contexto donde 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.

Ahora, dadas las necesidades del departamento, es importante que proyectos como el de Arancibia tengan continuidad.

## 2. Situación Actual

El proyecto actualmente, tiene una arquitectura como la que se muestra en la figura 1. Como se puede visualizar, el bot, guarda solo información temporalmente, no guarda ningún tipo de preferencias, más allá de la interacción puntual. Y del flujo de datos se desprende que la información compartida al sistema, corresponde solo a una evaluación simple del servicio. Al ir más allá y examinar detenidamente el modelo de datos, ver 3, se puede corroborar que no existe ningún tipo de funcionalidad ajustada a un usuario en particular. Para añadir funcionalidades más específicas, y a la vez más relevantes, es preciso entender las preferencias del usuario, a la vez que la recolección de información puede generar potenciales dilemas de privacidad. Al mismo tiempo al diseño de la página web, del diseño de Arancibia, requiere de la intervención de otros desarrolladores para mejorar su flujo actual. En ese sentido surge la pregunta ¿a dónde apuntar el desarrollo?

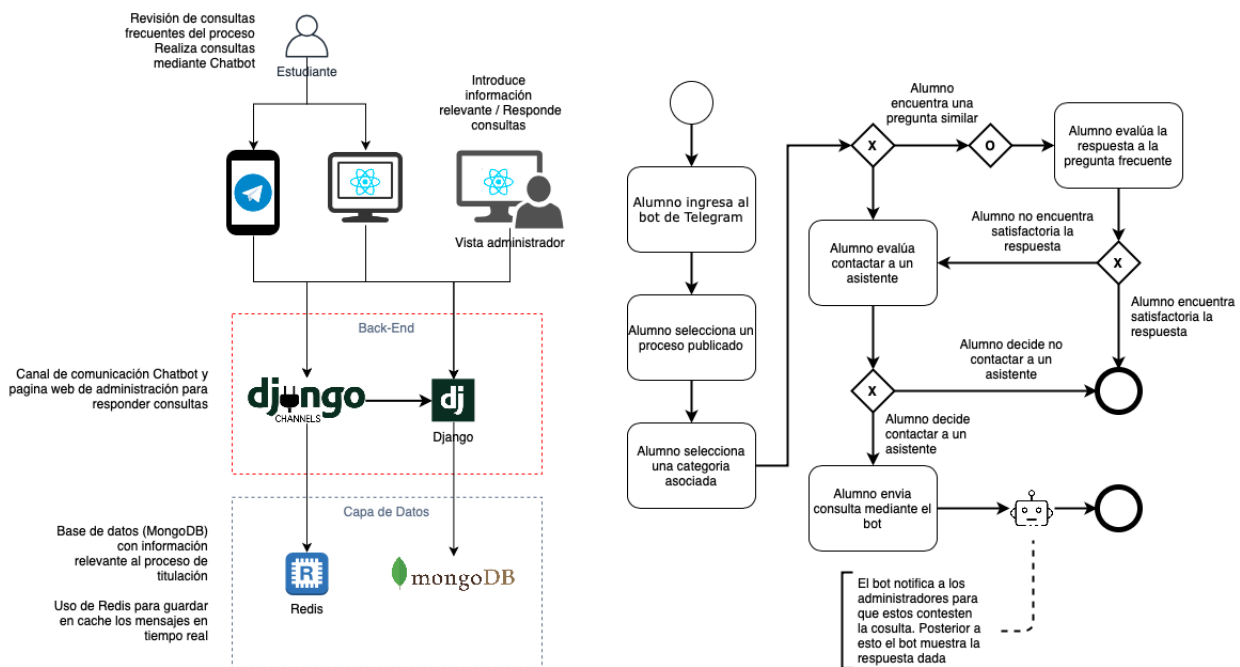


Figura 1: Diagramas de arquitectura y flujo de la interacción con el usuario respectivamente

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. Sin embargo, no tiene sentido extender un sistema que no tendrá uso, por eso lo principal, es garantizar 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. Y al mismo tiempo que el bot es una elección natural, hay ciertos elementos de en la solución actual, que pueden frenar la adopción del bot.

- **El diseño del bot:** Aunque funcional, el bot contiene aún varios aspectos a mejorar,

como fue expresado por su creador, el código actual "no se entiende del todo", y no es posible "darle un mantenimiento fácilmente", estas desventajas de la solución actual, hacen que sea necesario cambiar su diseño, para poder asegurar su continuidad. Y como se expresa en el flujo de la 1, el diseño actual solo cumple con una interacción básica.

- **posible ausencia de confiabilidad.** «la confianza es una fuerza fundamental positiva, que cuando es aplicada de la sociedad a la computación, dará buenos frutos» [6] El bot, actualmente no tiene ninguna forma de generar información por sí mismo, no puede acceder a las fuentes primarias de información por su cuenta. Por lo tanto los agentes encargados de guardar la información son vitales en la calidad de esta.

Uno de los actores relevantes en el proceso actual es el 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. Y posiblemente será el principal contribuidor a la información. Por otro lado, 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. Es en ese sentido, se puede concluir que a pesar de ser un conjunto funcional, el sistema requiere de mejora, para poder abarcar todo su potencial.

Debido a que todos estos actores son fundamentales, a la hora de otorgar confiabilidad al servicio (porque suelen ser las fuentes primarias de información), generar una solución que minimice la carga de tales actores y la distribuya, es fundamental para incluirlos o mantenerlos en diseños ulteriores.

- **No esta claro el horizonte de desarrollo:** Aunque el desarrollo actual no está completo, es posible implementarlo y usarlo. Sin embargo, este uso, no permite generar un horizonte claro de desarrollo, ya que como se describe en el trabajo de Arancibia, hay varias acciones que llevar a cabo en el área de mejora[1].

Una situación como esta puede frenar el desarrollo y ralentizar la llegada de los esperados beneficios. En ese sentido, es importante rescatar aquellos elementos transversales, a todo usuario, y a las aplicaciones, para que se pueda orientar el trabajo en la dirección adecuada. - es importante actualizar las next features cita

- **Ausencia de personalización:** Si bien hay un modelo y un sistema de información, para que un proyecto como este, se convierta en ecosistema de ayuda, y no solo en aplicaciones conectadas, uno de los ejes fundamentales es poder adaptarse a las necesidades de los usuarios individuales. Poder personalizar la ayuda tanto como sea posible le da un valor agregado para cada usuario, que hace al servicio propicio a mantener su uso.

La interfaz actual del bot, contiene las interacciones básicas esperadas, como mostrar el *faq* de forma categorizada, lo que permite llegar al tema más cercano en aproximadamente 3 *touches*. Pero aún no incluye avisos personalizados (suscripción específica), notificaciones, actualizar la información en pantalla y otras funcionalidades que podrían ser de utilidad, y que probablemente impacten en la UI/UX que los usuarios esperan obtener.

La personalización, en modelos de suscripción pagados es uno de los factores que define el *engagement*[5], en ese sentido, incluir algunas buenas prácticas de personalización, que permita generar contenido basado en los usuarios, es clave.

### 3. Objetivos

#### Objetivo General

**Generar Engagement** El objetivo principal es extender las funcionalidades del *bot*, tanto *back-end* como *front-end* para favorecer su adopción por los estudiantes y su posterior continuidad. Para lo cual se listan 5 objetivos específicos.

#### Objetivos Específicos

- **Objetivo 1: crear un sistema personalizable.** Se plantea generar un nivel de personalización del bot. Para esto se le añadirán *features* que permitan darle un valor propio a la aplicación, encontrar información selectivamente, además de dar un acompañamiento individual a cada alumno.
- **Objetivo 2: Hacer el sistema confiable.** Se propone entregar información clara, fidedigna y que no se contraponga a otras fuentes validadas. Con este fin, se presentará la información de manera simple, se mostrará su fuente y se pedirá feedback al usuario.
- **Objetivo 3: Generar un modelo de datos para orientar el desarrollo.** El bot en si es la punta del Iceberg de un modelo de atención de dudas y solicitudes, en ese sentido, una parte importante del proyecto debe ser generar un modelo de recolección de información que permita a su vez, determinar dónde se centran las mayores oportunidades del servicio, para que así, futuros trabajos puedan extender el funcionamiento en la dirección correcta.
- **Objetivo 4: Rescatar funcionalidades críticas.** Para poder ampliar las capacidades del sistema (que soporte a los nuevos usuarios) es necesario invertir tiempo en analizar la solución actual, para determinar aquellas áreas del desarrollo del bot que son críticas para obtener el funcionamiento esperado. Y mantenerlas en los diseños futuros.
- **Objetivo 5: Asegurar la Mantenibilidad.** A pesar del buen trabajo logrado, persisten muchas áreas sin terminar en el código, así como un código funcional pero no extensible. Por dichas razones es necesario reestructurar el código existente.
- **Reestructurar vs Reimplementar, depende pueden ser ambas, o solo una.**

#### Evaluación

Los objetivos son distintos en su naturaleza, pero en su mayoría son cualitativos. Dentro de las formas que se han analizado para evaluar su cumplimiento se encuentran.

1. **Métricas para personalización:** Que un sistema sea personalizable, puede incluir una infinidad de elementos desde aquellos netamente estéticos hasta los que tienen que ver con las características funcionales de la aplicación. Para este proyecto se adoptará un enfoque funcional. Y en ese sentido la personalización se medirá en 4 elementos:
  - Permitir al usuario suscribirse a un proceso en particular (Tesis, memoria, práctica, etc.).
  - Generar vistas de información relevante. Como por ejemplo, una vista cronológica para revisar fechas importantes
  - Notificar al usuario cuando se acerque algún deadline de sus procesos suscritos.

- Sugerir al usuario recursos que puedan ser útiles para los procesos a los que está suscrito.
2. **Para asegurar un sistema confiable:** Se propone, añadir a las respuestas, una fuente de información y mostrar cuando fue la última vez que fue validada esta información. Esto permite al usuario catalogar un mensaje como des actualizado, verificar si la respuesta tanto como su emisor son lo que está buscando. Esto además hace al sistema transparente en términos de responsabilidad. Al mismo tiempo, se propone llevar de manera interna un historial los agentes que han contribuido a cada una de las respuestas o tópicos emitidos, para poder auditar en caso de que ocurra alguna desinformación o mal-utilización de la plataforma. Se destaca eso si, que esto no afecta que quien esté respondiendo una consulta por el modelo de chat, permanezca como anónimo, solo se transparenta la fuente de la información. Se agregarán las opciones dentro de reatrealimentación de la consulta, un grado de confianza en la fuente y la opción de catalogar la información como vigente o desactualizada.
  3. **Validación del Modelo de datos:** . Para validar el modelo de datos se propone que este considere los siguientes aspectos:
    - Que permita identificar las componentes lógicas de la arquitectura
    - Que zonifique las componentes lógicas de acuerdo a su función.
    - Permita conocer el tráfico (en cantidad de usuarios) de cada zona.
    - caracterizar los procesos
    - Que permita saber cuales son las preguntas más problemáticas
  4. **Encuentro a las funcionalidades críticas y la mantenibilidad** Dentro de las funcionalidades críticas, no se conocen todos los aspectos del código que influyen en entregar la funcionalidad esperada, ya que esto jsutamente parte del análisis a realizar. En ese sentido se pretende que el sistema compla con:
    - Permitir al usuario acceder a la información.
    - Permitir a los administradores respodner a los usuarios.
    - permitir guardar información de la calidad de las preguntas.
 Así mismo, tanto para un buen manejo de las funcionalidades críticas como de la mantenibilidad, se propone la creación de una wiki de desarrollo que aloje:
    - Arquitectura y estructura lógica del código.
    - Tecnologías usadas
    - Modelos de datos
    - Fuentes de Información validada.
    - Manuales: Información de instalación y despliegue. Cómo usar distintas funcionalidades. Consensos y acuerdos de sintaxis.

## 4. Solución Propuesta

Lo esencial de la solución es la implementación de una interfaz, un modelo y set tests que permitan la mejora incremental y continua del servicio entregado. Por eso mismo se plantea que las tareas que se extiendan de manera cíclica.

Ya que por la naturaleza del proyecto habrá actualizaciones del modelo de datos, puesto, que un modelo incluso bien planteado, va a tener que ajustarse a los usuarios en distintos aspectos, como pueden ser constantes de sistema, fuentes de información, variables de personalización, entre otros. El mismo sentido, una interfaz inicialmente válida, puede tener que ajustarse en torno al uso que los usuarios den ella, alterándola, para este fin. Por lo tanto se propone realizar

- **Análisis y Test de la Solución Actual:** Para asegurar la mantenibilidad y rescatar las funcionalidades críticas, se diseñarán y usarán un *tests*. Estos servirán para validar como el sistema actual responde a los objetivos planteados, así como otras posibles *features*, que se puedan rescatar a partir de la interacción de los usuarios con el sistema.
- **Refactorización:** A partir del análisis se plantea, refactorizar el código (del bot) para modularizar la solución actual, darle una estructura lógica, que permitan entender el sistema y hacerlo extensible.
- **modelo de datos para suscripción:** Se plantea generar a partir del análisis preliminar un primer modelo, que contenga los elementos necesarios para responder a los objetivos del sistema. Este modelo debe considerar su posterior extensibilidad así como su uso continuo, así que la modularidad es clave en este diseño, ver figura 2. Dentro de otros aspectos fundamentales este modelo incluirá una forma de auditar el tiempo de respuesta, las fuentes de información confiable, identificación y preferencias de los usuarios, así como métricas de uso para su mejora continua.
- **UX/UI:** también a partir del testing, así como de una investigación del estado del arte, se plantea, implementar mejoras a la navegación del bot, al contenido desplegado, y a generar nuevas componentes que contemplen las funcionalidades requeridas. Y que permitan al usuario reutilizar el bot para nuevas tareas y generar los datos necesarios para su mejora.
- **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.

### 4.1. 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*, esta librería está diseñada para el uso con *chatbots* [1].
- **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.

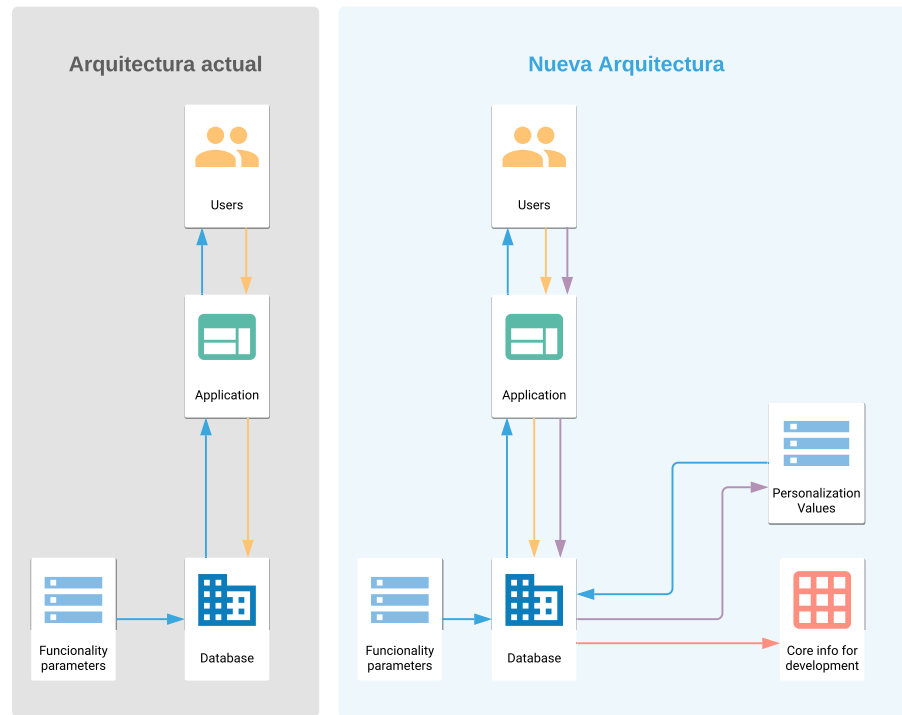


Figura 2: Propuesta de Arquitectura: En la figura las líneas azules denotan el flujo de información desde el sistema, las amarillas las peticiones puntuales del usuario, las moradas las subscripciones o contenido personalizado, y la roja la información valiosa del sistema para la extensión del mismo



## 5. Plan de Trabajo (Opcional)

1. **Diagnóstico (1s):** *Análisis preliminar:* Se plantea revisar las soluciones modelo del trabajo anterior esto consistirá en: revisar los fundamentos de las decisiones de diseño, contrastarlas con el estado del arte, revisar el modelo de datos actual, y hacer un testing cerrado de las funcionalidades. *Análisis de código:* Se plantea revisar a fondo el código para extraer las secciones críticas, extraer la semántica esencial y hacer una aproximación teórica en relación a cómo el sistema respondería a los distintos casos de uso.

*Pruebas de sistema:* Probar con usuarios y sistemas automáticos la respuesta del sistema a ciertas peticiones, permitiendo evidenciar cuellos de botella en el flujo, puntos críticos para la refactorización, así como determinar casos bordes que pudieran pasar inadvertidos.

*Pruebas de Usabilidad:* Probar con usuarios el sistema para recopilar datos sobre las mejoras a la interfaz. Enfocada en datos como: la navegación, la visibilidad, la confiabilidad, la claridad del contenido, entre otros.

*Análisis del Modelo de Datos:* analizar el modelo de datos existente, en contraste con los resultados obtenidos, para obtener y realizar de ser necesario, modificaciones, adiciones o incluso el rediseño, de lo que sea necesario.

2. **Refactorización de Código (1m):** A partir del análisis, llevar a cabo una refactorización de código para que este responda a al modelo de datos actualizado y los objetivos planteado.
3. **Extensión del *Back-end* para nuevas funcionalidades (2m):** Luego de actualizar el código, se agregarán todas las funcionalidades que se hayan considerado necesarias ya sea a partir del testing o del la refactorización, que no estén presentes en el modelo inicial.
4. **Proceso de Validación-Refactorización activa (2m)** A partir del trabajo anterior, se debería obtener una versión  $\gamma$  del proyecto. Con esta versión funcional y actualizada, se propone, realizar un periodo de test y mejora continua, seccionada, según los objetivos planteados. Para este periodo se propone una metodología inspirada en el desarrollo ágil. Alguna condiciones de éxito es que exista un pequeño grupo de testing (suscriptores del bot). Y estará basada el siguiente diseño de sprints:

sprint:  
Colección de Issues  
Sortig Issues  
Fast Fixing  
Validation-Check

Secciones:

- Primera validación: **funcionalidades** (2s)
- Segunda validación: **UI** (2s)
- Segunda validación: **Egagement** (2s)
- Cuarta validación: **Quality Assuring**

5. **Análisis de resultados y Escritura del Informe (1m):** Se dejará un tiempo para analizar los resultados obtenidos, validar los modelos, hacer pequeños ajustes de código y escribir el informe final.

## 5.1. Resumen

Semestre	Proceso	Duración
1	Diagnóstico	3 sem
1	Refactorización	4 sem
1	Extensión	3 sem
2	Extensión	5 sem
2	Validación	5 sem
2	Análisis	5 sem
total		15 semanas

Tabla 1: Resumen del plan de trabajo

## 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](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] 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>.
- [6] 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.
- [7] 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](http://www.revistapsicologia.uchile.cl).
- [8] 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>.

## 6. Anexos

### Base de Datos - Mesa de ayuda DCC

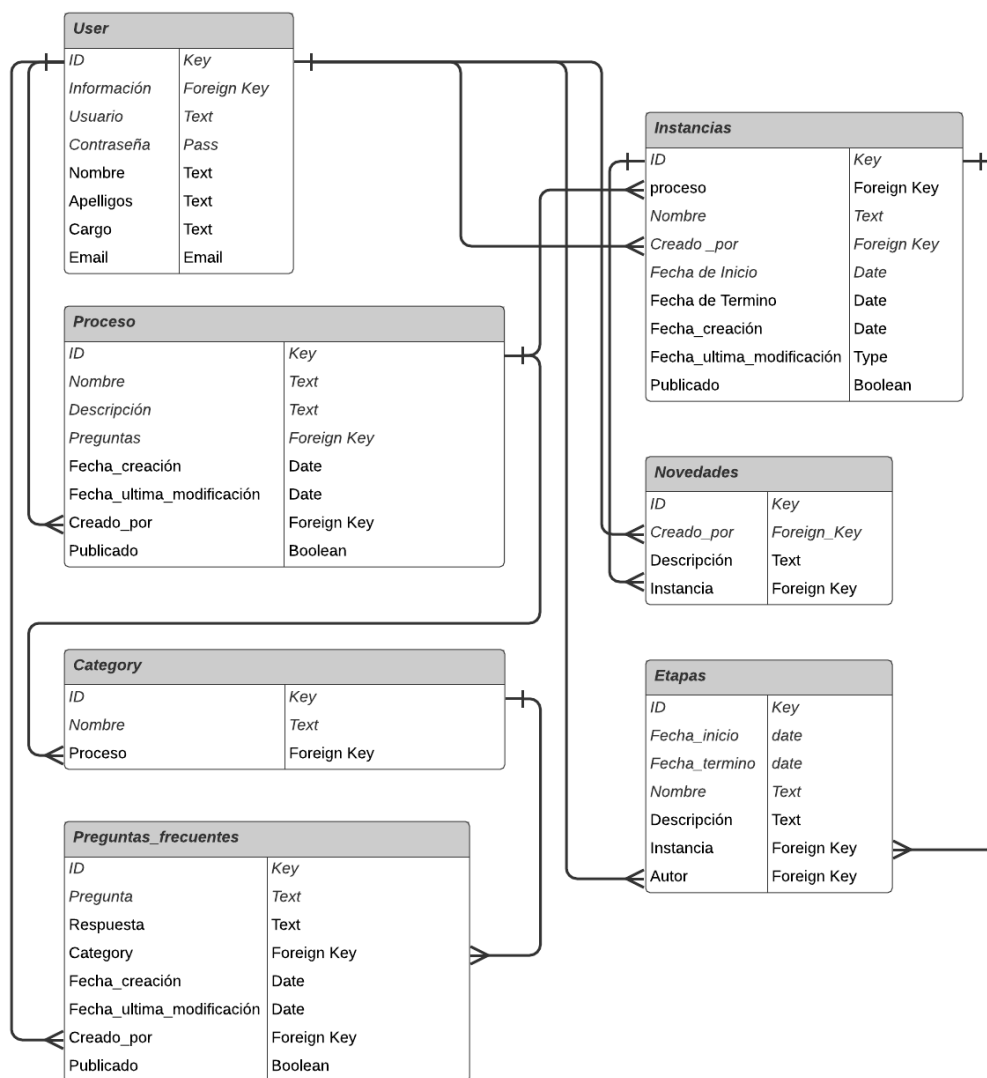


Figura 3: Modelo de datos actual del sistema Mesa de ayuda