#### MEJORA DE UN SISTEMA DE MESA DE AYUDA PARA ESTUDIANTES DEL DCC

# MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

MARCELO ESTEBAN BECERRA ALARCÓN

PROFESOR GUÍA: JOCELYN SIMMONDS

MIEMBROS DE LA COMISIÓN:

# Resumen

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

 $Una\ dedicatoria\ corta.$ 

# Agradecimientos

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

# Tabla de Contenido

# Índice de Tablas

# Índice de Ilustraciones

# Capítulo 1

# Introducción

La cantidad de alumnos del Departamento de las Ciencias de la Computación (DCC), ha aumentado considerablemente en los últimos 5 años, 200 alumnos aproximadamente (212) [CADCC2016], [CaDCC2018], [CADCC2021]. Este aumento viene acompañado de una serie de desafíos, no solo en la docencia, sino en los procesos administrativos. Contexto (Antecedentes)

### 1.1. Contexto

Uno de los focos de estos desafíos son los procesos de titulación, que también se han visto afectados por este aumento, casi el doble de alumnos en 7 años [ARANCIBIA2021]. Asociados a ellos, encontramos procedimientos, reglas de formato, un tipo específico de contenido, que junto a otros elementos causan incertidumbre por parte de los alumnos. En relación con ellos hay varias fuentes de información que atañen a diversos aspectos de estos trabajos como la wiki de titulación, las páginas del centro de alumnos, publicaciones oficiales en U-Cursos, etc. A pesar de tener estos recursos disponibles, muchas veces, la mayoría de los estudiantes suele preguntar directamente a sus profesores y, funcionarios como la secretaria del departamento suelen recibir la mayor cantidad de dudas no resueltas [ARANCIBIA2021].

Una de las soluciones a considerar, podría ser ampliar personal, lo cual debe ser debidamente justificado, y consensuado [Chile2014]. Por otra parte, tampoco es lo que buscan los estudiantes, pues a la mayoría le gustaría un acceso directo a la información sin pasar por intermediarios [ARANCIBIA2021]. Y es justamente en este ambiente, que el proyecto Mesa de Ayuda Virtual se posiciona como una buena alternativa para asistir a los estudiantes.

El proyecto Mesa de Ayuda Virtual, es un proyecto de soporte o asistencia digital, 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: La primera es un bot de Telegram, con funcionalidades como preguntas frecuentes agrupadas por proceso, capacidad de feedback por parte del usuario y contactar a un asistente. La segunda, es una página web, que cuenta con un front-end para alumnos y otro para administradores,

en ellos se puede categorizar preguntas, acceder y modificar los procesos vigentes con sus preguntas, dispone de chat directo con los estudiantes, 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.

### 1.2. Problema

Como se habló anteriormente, la solución consta de dos partes vitales la web y el bot. 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. Sin embargo, este proyecto no está implementado y presenta algunas falencias en su desarrollo que impiden hacerlo extensible, principalmente en el código del bot. En ese sentido, el que una parte vital del proyecto no pueda tener soporte es algo crítico.

Además la ausencia de personalización, en un sistema de información, podría hacer que los usuarios lo dejen de usar, ya que este es un factor clave en un sistema como este [Paz2021].

El presente trabajo busca aportar en la continuidad del proyecto, enfocándose en resolver los problemas descritos Objetivos

# 1.3. Objetivos

## Objetivo General

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

## Objetivos Específicos

- 1. Analizar la solución existente, con el fin de identificar qué modificaciones son relevantes, tanto para que el modelo de datos soporte las nuevas funcionalidades, como para saber la manera en que se debe reestructurar el código.
- 2. Rediseñar los modelos y funcionalidades, para que el sistema acepte las modificaciones necesarias.
- 3. Diseñar nuevos componentes 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. Diseñar un modelo de subscripción personalizada: que contempla 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 y probar las reestructuraciones, asegurando la integridad del modelo de datos, del código y permitiendo su extensibilidad a través de la modularidad lógica.
- 6. Implementar y probar las nuevas funcionalidades buscando que sean aquellas relevantes para los alumnos.

#### **Evaluación**

Para validar que el objetivo se cumplió, hay dos áreas que tomar en cuenta: La integridad del sistema y la validez de las nuevas funcionalidades.

En relación con la primera, se espera que las modificaciones hechas le permitan al sistema seguir cumpliendo con sus objetivos. Para lo cual se creará y usará una serie de tests que permitan asegurar sus funcionalidades críticas y la integridad del sistema. Por ejemplo, en el back-end probando los end-points, que los modelos permitan operaciones CRUD, entre otros.

En relación con la validez de las nuevas funcionalidades, se propone hacer tests de usabilidad, tanto entrevistas individuales como focus groups.

Adicionalmente, solo si es posible, se propone beta testing, porque permitiría integrar ambas áreas y acercar el producto al usuario. Pero esto depende de la disponibilidad del CADCC, así como de que el sistema ya este desplegado por ellos. Solución Propuesta

# 1.4. Solución Propuesta

Para poder llevar a cabo los objetivos anteriormente descritos, se propone agruparlos en 3 fases: Análisis, Diseño e Implementación. Pensadas en la lógica que abarca cada uno de estos objetivos.

El Análisis tiene el propósito de asegurar la mantenibilidad y rescatar las funcionalidades críticas. Para lo cual, se estudiará en detalle la solución actual. Con esto se busca identificar todos aquellos aspectos relevantes, así como aquellos que necesiten una modificación, ya sea una reestructuración o reimplementación. Entre ellos: las modificaciones en el modelo de datos, las refactorizaciones en el código para modularizar la solución actual y darle una estructura lógica, que permitan entender el sistema y hacerlo extensible.

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 con el conjunto de test diseñados, cuando existan modificaciones, se probará que no afectaron la funcionalidad original ya implementada.

Durante el Diseño, se buscará agregar elementos de confiabilidad y generar un modelo de subscripción.

Para mejorar la confiabilidad se agregará a las respuestas, la fuente de la información y su última actualización. 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.

En relación con el modelo de subscripción, este permitirá buscar y seleccionar un proceso de forma individual, más específicamente una instancia a un cierto proceso, ya que los procesos cómo tesis, memorias, funcionan como clases de los objetos individuales, que serían los procesos vigentes cada semestre (ver [ARANCIBIA2021]). A partir de esta búsqueda, se debe permitir suscribirse 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. A estos elementos se les denominó en los objetivos elemento subscriptor.

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.

Al final de esta fase, se debe validar estos modelo con usuarios, y entidades relevantes para hacer un ajuste, antes de comenzar con la etapa de implementación. Estas validaciones, deben 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, puede levantar problemas de privacidad que deben ser revisados junto a los encuestados.

La última 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.

#### 1.4.1. 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 requeire tiempo, especial en cada iteración, para realizar una buena documentación del sistema.

### 1.4.2. 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 *Celeri* para el menejo 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*.
- Pagina 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.

# 1.5. Metodologias

a Continuación se detalla el plan de trabajo con sus partes más relevantes: Análisis, Diseño, Implementación, Validación y Documentación.

				Agosto		Septiembre				Octubre				Noviembre				
			S1	S2	S3	S4		S5	S6	S7	S8	S9	S10		S11	S12	S13	S14
	Inicio	Fin	16-8	23-8	30-8	6-9	13-9	20-9	27-9	4-10	11-10	18-10	25-10	1-11	8-11	15-11	22-11	29-11
Análisis del 16-Aug al 05-Sep																		
Deployment y análisis de código	16-8	23-8																
Refactorización preliminar	16-8	30-8																
Análsis de nuevos casos de Uso	30-8	30-8																
Diseño del 06-Sep al 26-Sep																		
Diseño de Diagramas UML	6-9	13-9																
Diseño de la nueva arquitectura lógica	13-9	13-9																
Diseño de los nuevos modelos de datos	13-9	13-9																
Diseño de las nuevas GUI	6-9	20-9																
Implementación del 20-Sep al 03-Oct																		
Modificaciones a la BD	20-9	20-9																
Creación de los nuevos modulos para la API	20-9	27-9																
Inserción de las nuevas UI	20-9	27-9																
Validación del 27-Sep al 31-Oct																		
Testing Funcional preliminar	27-9	27-9																
Testing Funcional completo	4-10	25-10																
Validación de las nuevas funcionalidades	4-10	18-10																
Validación de las nuevas interfaces	25-10	25-10																
Documentación del 13-Sep al 31-Oct																		
Docuementar los modulos del bot	20-9	20-9																
Documentar la estructura lógica del código	4-10	11-10																
Documentar el modelo de datos final	13-9	13-9																
Documentar las interfaces de Usuario	18-10	25-10																
Informe del 01-Nov al 05-Dec																		
Análisis de resultados	1-11	1-11																
Redacción del Informe final	8-11	29-11																

Figura 1.1: Carta Gantt del plan de trabajo propuesto para el desarrollo de la memoria

# Capítulo 2

# Marco Teórico

# 2.1. Estado del Arte

Para el desarrollo de este trabajo se hizo una investigación sobre los sistemas que solucionan problemas similares y a que objetivos responden. Al mismo tiempo como abordar el hecho de que el usuario es parte fundamental del flujo del sistema tanto en su funcionamiento como imponiendo restricciones dinámicas sobre el mismo, las que responderían a preferencias o valores de cada usuario. Por este motivo los temas abordados a continuación son *User Centered Desing (UCD)*, human in the loop, information systems y chatbots.

## 2.1.1. User centered desing - privacy, cutomization and confiability

Este es un sistema que indudablemente se basa en el usuario y sus preferencias, es por esta razón que para que el desarrollo sea exitoso no basta con que se desarrolle de la manera correcta y asumiendo buenas prácticas, sino que debe hacerse en torno al usuario y rescatando no sólo su input [Karat1997], sino sus objetivos y valores. Dentro de las maneras de enfocar un desarrollo centrado en el usuario encontramos 3 alternativas la especialista, generalista y mixto [Fox2008].

la manera especialista hace referencia a que hay 3 actores encargados del proceso, los usuarios, los encargados de UCD y el resto del equipo de desarrollo. Por otro lado, la generalista asume dos roles: el de usuario y el de desarrollador encargado de esta área.

En esta línea de *User Centered Desing*, el diseño propiamente tal, se entiende como un proceso, en el que debe estar involucrado el usuario final, de forma constante. De modo que la personalización llega a ser una pieza fundamental del diseño exitoso es aterrizar el desarrollo de *features* y herramientas sobre el valor del usuario final"[**Kramer2000**].

De aquí se obtiene que el sistema debería ser personalizable, y rescatar lo que sea valorado por los usuarios. Dada la naturaleza del proyecto, se decide adoptar un enfoque generalista.

### 2.1.2. Information Sistem and Human in the loop

Al ser un sistema que depende del usuario para determinar el curso de acción, el humano en este sistema no solo está en ciertos momentos críticos del flujo, sino que es el centro del flujo de información, modelos como este han sido validados con buenos resultados [Smith2018] recientemente.

Los sistemas de información asociados a chatbots, generalmente buscan la integración de información, este proceso es decir mostrarle al usuario una gran variedad de información de diferentes fuentes a través de una vista unificada viene asociado a varias dificultades técnicas y teóricas [Li2017].

En este sentido, se propone que los sistemas de información que dependen de un humano son altamente complejos, porque los humanos, en este caso usuarios no solo son observadores, sino que tienen la capacidad de incidir en el ambiente no solo reaccionar al ambiente [McBride2021], sería iluso desarrollar un sistema para las personas sin considerar los objetivos de ellas de forma explícita en el diseño del sistema.

Estos descubrimientos e investigaciones ponen varias restricciones en cuanto al desarrollo de un sistema que depende los usuarios para funcionar y que al mismo tiempo está centrado en el usuario. Pero las podemos condensar en al menos 2: El diseño del sistema debe considerar de forma explícita los objetivos del usuario en cada interacción y debe integrar la información.

#### 2.1.3. Chabots

Los chatbots se utilizan desde hace varios años para variados y diversos fines, en múltiples áreas. En particular ahora los abordaremos desde la esfera académica, centrándonos específicamente en instituciones de educación superior. En estas se utilizan principalmente como un medio de apoyo al proceso de aprendizaje, como soporte para procesos universitarios, como ayuda para la gestión académica y cómo un sistema de consulta que responde a dudas de diversos entes de estas instituciones.

Como apoyo al proceso de aprendizaje los chatbots se han usado tanto dentro como fuera de la sala de clases.

En la *University of Salerno*, se creó un sistema para responder de manera efectiva las preguntas de los estudiantes, el contexto de *e-learning*. A grandes rasgos es un sistema que usa procesamiento de lenguaje natural como técnicas de ontología, para ligar las preguntas de los estudiantes al contenido disponible y de esta forma poder responder eso que los alumnos están buscando [Clarizia2018]

El Dr. Sherif Abdelhamid del Virginia Polytechnic Institute and State University también propuso un sistema de características similares [Abdelhamid2020], con las distinciones de que uso el sistema de reconocimiento de voz para traducir las dudas verbales de los estudiantes a texto y así procesarlas y que su objetivo estaba ligados a los elementos que acompañaban al estudio más que al aprendizaje académico en sí. Algunos ejemplos como cuál es la bibliografía pertinente al curso, que recursos tiene disponibles, que debería aprender, etc.

En este aspecto, investigadores del Department of Mathematics and Computer Science, University of Kabianga, Kericho, Kenya con miembros del Department of Curriculum Instruction Educational Media, School of Education, Moi University, Kenya analizaron la respuesta de los mismos profesores a la introducción de estas herramientas en la rutina de enseñanza [K2018]

Como soporte para procesos universitarios los chatbots se han usado para entregar el horario de clases o las clases siguientes así cómo ayudar a los alumnos a elegir ramos.

Desarrolladores de la Bogdan Khmelnytsky Melitopol State Pedagogical University en Ucrania, desarrollaron un servicio de 3 capas para poder implementar un servicio de ayuda al entregar el horario de clases a través de un chatbot, esto se hacía no solo con la finalidad de ayudar a alumnos que hayan olvidado o no un horario o su sala, sino también porque los horarios pueden sufrir cambios o modificaciones, además de que si bien hay alternativas simples como fotos estas se pierden en un gran volumen de elementos similares sin relación, lo que las hace difícil de volver a obtener, según la investigación. [Priadko2019]

En *The Open University of Hong Kong Hong Kong* desarrolladores implementaron un sistema para facilitar a los alumnos la decisión sobre qué cursos tomar a través de un chatbot [ChunHo2018].

En el área de los desarrollos de ayuda para la gestión académica, se encontraron desarrollos ligados a: Hacer disponibles servicios en horarios no hábiles, Mantener, monitorear y mostrar la información del alumno y Acercar la implementación de chatbots a usuarios de más alto nivel y con menos conocimiento técnico.

Los trabajos de Vaishnavi Ajay Inamdar1 y Shivanand R.D en el *BIET College* de la India, van enfocados en poder proveer servicios de gestión académica en cualquier momento, de este modo bajar la carga de los funcionarios y proveer de una *Efective GUI* [Journal2019]

Del Engineering Rajalakshmi Institute of Technology una profesora y sus alumnos, desarrollaron un sistema para administrar la información escolar de los alumnos y está ligado la base de datos de administración. Este sistema tenía el foco en el personal administrativo de la institución y dependía de la información entregada por el alumno.

Con el foco en los apoderados o tutores de un alumno, en la *Universitas Komputer Indonesia* A Heryandi, desarrolló un sistema de chatbots para poder desplegar las notas y el rendimiento de los alumnos a los padres usualmente menos versados en tecnología y más familiarizados con plataformas de mensajería. Por esto mismo se desarrolló un sistema con *Easy access*, es decir, con un login o un proceso de autenticación sencillo y con información conocida por los tutores [Heryandi2020].

En Indonesia en la *Universitas Sebelas Maret* investigadores desarrollaron un sistema con un bot de Telegram para incluir a funcionarios no expertos en el diseño de funcionalidades para la universidad. Este proveía funcionalidades como agregar botones, comandos, acciones, mensajes directos, transmitir mensajes, configuraciones y análisis [Hasyim\_2021].

Como un sistema de consulta se han desarrollado sistemas de preguntas frecuentes, algunos ejemplos son el sistema de Mesa de ayuda del DCC [ARANCIBIA2021] y un trabajo

de la *Manipal University* en la India [Ranoliya2017].

Aunque ambos son similares, responden a necesidades completamente distintas, el sistema del DCC es un proyecto centrado en el usuario, mientras que el trabajo de la India tiene un foco en la performance y el uso de IA.

#### 2.1.4. Análisis de las soluciones actuales

Como se puede observar los bots cumplen variadas funciones dentro de la vida universitaria a través del mundo, apoyando a toda la comunidad escolar, tanto tutores, como funcionarios y alumnos, haciendo más accesibles, disponibles y eficientes diversos servicios, incluso apoyando procesos de aprendizaje virtual y presencial.

A pesar de la gran diversidad de funcionalidades que están presentes en estos sistemas, se logra rescatar que en la mayoría de los casos se diseña basándose en un modelo de inputs, procesamiento y respuestas. En la mayoría de los casos a pesar de ser desarrollos que buscan acercar o facilitar la vida de sus usuarios, suelen tomar un enfoque técnico y centrarse en aspectos como tener una , son desarrollos que en general carecen de un modelo que integre al usuario directamente, y se enfocan en solucionar los problemas de manera analítica o teórica. Esto podría ser una enorme dificultad para el proyecto e iría en contra de los descubrimientos en el área de *User Centered Desing*.

Por estas razones se opta por tomar los buenos resultados funcionales de los chatbots, pero agregarles de alguna forma las preferencias de los usuarios de forma explícita en el diseño de los nuevos modelos de interacción.

# 2.2. Notacion i\*

La notación i\*, se usará como se explicó anteriormente para explicar no solo la interacción del usuario con el sistema, sino también que persiguen los mismos al realizar dicha interacción.

Las bases del lenguaje se pueden encontrar en la guía del lenguaje [Dalpiaz2016] y las imágenes en la wiki del lenguaje [No-mention2011].

esto se hace a través de la sintaxis de este lenguaje que detallaremos brevemente a continuación:

• Objetivos: los objetivos o *Goals* por su nombre en inglés, son en palabras simples un estado que desea alcanzar el usuario relativo a algo. Por ejemplo: "Tener los pasajes de avión comprados"



Figura 2.1: Objetivo i\*

• Tareas: las tareas (*Task*) son acciones que el usuario puede realizar para lograr su objetivo. Ejemplo: "Buscar pasajes de avión", "Comprar pasajes en la aerolínea".



Figura 2.2: Tarea i\*

• Cualidades: o *Qualities* son maneras o elementos de valor que se quiere preservar u obviar al completar un objetivo. Ejemplo: "Rápido", en esta serie de ejemplo la frase completa se entendería como "Tener los pasajes de avión comprados rápido" o de manera "rápida".



Figura 2.3: Cualidad i\*, antes softgoal en i\* 1.0

• Recursos: Los recursos son elementos ligados a las tareas, como una "tarjeta de crédito", en esa línea se podría "comprar los pases en la aerolínea con una tarjeta de crédito"



Figura 2.4: Recurso i\*

• Actores: Los actores son elementos principales del diagrama, realiza acciones, dispone los recursos y tiene objetivos. Pueden ser de tres tipos: un tipo general del cuando no se desea especificar, un agente que representa una instancia particular de un actor: como "Juan" o el "Bot" y un rol, que viene a ser una categoría o clase como "Estudiante". Las boundaries o límites relativos a un actor determinan que elementos le "pertenecen"

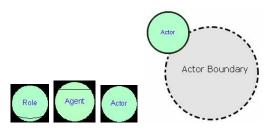


Figura 2.5: Actores i\*

- Enlaces: o *Links* son lo que representa las relaciones entre los diferentes elementos del sistema, pueden representar dependencias, contribuciones, especificaciones o incluso notar una contribución. En el caso de este trabajo se usan principalmente 3 tipos de enlaces:
  - Dependencias: Se indican con una letra D y expresan que el elemento a la izquierda de la dependencia, depende del elemento a la derecha, de manera un poco simplista.



Figura 2.6: Dependencia i\*

Contribuciones: Indican que cierto elemento contribuye a una Cualidad, puede ser de 4 formas, satisfaciendo completamente lo que se nota con make sobre el enlace, Imposibilitando su realización lo que se denota con break, o ayudando un poco o debilitando un poco lo que se denota con help y hurt respectivamente.



Figura 2.7: Contribuciones i\*

- Refinamiento: Expresan que un objetivo o tarea tiene otros elementos que lo definen. Pueden ser O (OR) o Y (AND), lo que determinan es que para que el objetivo se satisfaga se debe satisfacer a su vez todos los elementos que lo refinan (AND) o alguno de ellos (OR).

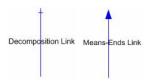


Figura 2.8: Refinamiento i\*

# Capítulo 3

# Análisis y Diseño

A partir de lo expresado anteriormente, se puede observar que existen variados desafíos en torno al diseño del sistema pero también a la implementación y uso del mismo. Una de las grandes preocupaciones al desarrollar el sistema era diseñar una infraestructura tanto lógica como de software que permitiera abarcar no solo las necesidades de los alumnos, sino también la manera en que ellos esperan que estas necesidades sean resueltas, ya que esto es un elemento clave en la adopción de tecnologías como los chatbots ().

Para ello el proceso de análisis y diseño tomó como input las valoraciones de los alumnos, potenciales usuarios del sistema, o de ex-alumnos que ya hubieran pasado por la mayoría de los procesos académicos de pregrado. Y partir de estos resultados, se esquematizan sus necesidades para desarrollar manteniendo no solo las funcionalidades y contenidos deseados, sino también la manera en que se quieren obtener estos requisitos.

Con el fin de poder llevar a cabo esto se realizaron: Dos Focus Groups, su metodología y resultados se detallan a continuación. Un diseño basado en i\* de necesidades y objetivos. Un diseño basado en UML para la capa de software.

# 3.1. Exploración a través de Focus Groups

Una de las problemáticas del trabajo realizado, era validar con usuarios del DCC las propuestas obtenidas de la investigación teórica. A modo de recoger sus preferencias de manera explícita en la interacción con el sistema. Para esto, se llevaron a cabo dos *Focus Group*. En ellos se realizaron consultas en torno a la información a la que se puede acceder a través del bot, información sugerida a través del comportamiento social, el tutelaje, la motivación para obviar el contacto directo con personas, privacidad y opciones de notificación.

### 3.1.1. Caracterización de los participantes

Los participantes del primer FG, son alumnos y ex-alumnos del DCC. De estos, ocho de los nueve han presentado problemas relativos a información académica, es decir, la falta de información o información errónea les ha dificultado el avance en algunos procesos académicos. Cinco de nueve no han cursado ningún ramo relativo a la memoria, sin embargo todos han participado de procesos con plazos como las prácticas profesionales o CC5402 Proyecto de software. En el caso del segundo *Focus Group* las características son similares, pero son solo cuatro alumnos

### 3.1.2. Respuestas de los estudiantes

A continuación se detallan los resultados obtenidos para las categorías anteriormente descritas. Los resultados mostrados responden a análisis en conjunto de ambos *Focus Group*, por lo tanto de aquí en adelante cuando refiramos este término nos estaremos refiriendo al proceso de consulta completo.

• Información a la que se puede acceder a través del bot: El contenido al que se puede acceder actualmente a través del bot, son solamente las preguntas frecuentes. En relación con las nuevas funcionalidades de suscripción se les preguntó a los estudiantes cuál era la información relevante para ellos, sobre qué cosas querían un recordatorio o sugerencias.

La mayoría está de acuerdo en que las fechas, ya sean de apertura o cierre de procesos, incluso tareas o avances parciales serían un contenido que ellos desearían del bot, cómo una notificación. También los cambios en la información relativa a un proceso, se deberían comunicar oportunamente, por otro lado, no se debería notificar un cambio de plazo, sino ajustar los recordatorios a los nuevos plazos. Cómo sugerencias, se admiten temas relacionados con los procesos en los que se esté suscrito o se haya preguntado reiteradamente ya sea por parte del usuario o por usuarios "similares".

Aparte de las notificaciones, se sugirió dar acceso a información relevante del proceso en cuestión, como el contacto público de personas relacionadas con el proceso (correo el coordinador del CC6908, contacto de la secretaria docente, por decir algunos) y ejemplos de válidos de los requisitos para cierta etapa (preinscripciones de práctica, ejemplos de propuestas de memoria).

- Información Social: Sobre este punto la consulta venía en torno a si estarían dispuestos a recibir sugerencias a partir de otros usuarios con perfiles similares. Aquí las opiniones estaban divididas. Aunque en el alumnado se es reticente a compartir información con terceros, no habría problemas en que fuera información exclusivamente académica, relacionada con fines laborales o información que otros organismos ya tengan.
- Tutelaje: La idea de esta pregunta era saber qué tan dispuestos están en involucrar un tercer actor que pueda ayudar en un proceso académico, al que denominaremos tutor, por ejemplo que el profesor guía recibiera un aviso cuando el alumno no respondiera ni

interactuara con el bot durante el proceso de titulación. Las respuestas dan cuenta de que no hay un acuerdo claro en cuanto a esto. Para algunos era algo viable, para otros era algo opcional, mientras que unos pocos preferían descartar de plano.

Finalmente a pesar de los cuestionamientos a su funcionalidad y a la forma de usar los bots de cada alumno. Se pudo consensuar que: Fuera una opción habilitada por el usuario y con posibilidad de deshabilitar en cualquier momento, coloquialmente çómo hacer un sudo". Así mismo, los parámetros asociados a este tercero como el contacto e información a la que accede también deberían ser definidos por el usuario. De esta forma, elegir al "tutor" que cada usuario decida y asociado al aspecto que él decida.

- Motivación para obviar el contacto: Esta pregunta busca profundizar en las razones de por qué los alumnos podrían preferir el bot al contacto directo con otra persona. En este caso, las razones son diferentes para cada alumno, pero concuerdan con la literatura de lo que se busca en un sistema de información [Thurman]. Algunas de ellas son ansiedad producto de la interacción, para no molestar, porque las preguntas son repetitivas y algunas mejoras funcionales como la rapidez, sugerencias a temas que tal vez no surjan de una pregunta directa a un encargado humano, la libertad de acudir cuando se desee y la sensación de control sobre la interacción.
- Privacidad: Dentro de este ítem, se preguntó específicamente por la privacidad de la información que el bot pudiera almacenar, y que trade-off estarían dispuestos a realizar por más funcionalidades. Aquí hay acuerdo en que debería ser la mínima posible y en caso de ser más solo información académica. Por otro lado, la mayoría estuvo de acuerdo en conceder permisos temporales y sin guardar información de forma permanente, en caso de que alguna funcionalidad requiera más información. De este modo se vuelve a repetir la configuración y permisos personalizados a cada alumno.
- Opciones de notificación: En el caso de las opciones de notificación se puede observar se puede observar que las notificaciones deberían ser en general cortas con poca información y que también fueran personalizables. Cada notificación debería mostrar el área a la que pertenece o una forma de saber qué temática o relacionado con qué es esa notificación. Al mismo tiempo, debería ser capaz de explicar en breves líneas que es la información nueva y en el caso de requerir más espacio, texto o se necesite mostrar más contenido debería agregarse un link o un enlace en el que se pueda desplegar a través de una página o un blog la información necesaria pero no directamente en el Chat.

## 3.1.3. Análisis y conclusiones

A pesar de que en el focus group se puede ver una ligera tendencia a ciertas características a partir del grado de avance de los alumnos, en general lo que se puede observar es que no hay una correlación directa entre el grado de avance y las preferencias personales de cada alumno, así que se propone una caracterización alternativa, asociar a los estudiantes en tres grupos: Aquellos que prefieren definir absolutamente todos los parámetros de su interacción, otros que prefieren que se les sugiera o lo que se les facilite la interacción, mientras que los terceros serían algo así como un usuario mixto que prefiere cierto grado de personalización mientras que prefiere una simplificación de otras tareas.

Generalmente los alumnos que prefieren una alta especificación o definir cada una de las características, están también ligados a un alto grado de valoración de la privacidad y configuración manual. Se perciben a ellos mismos como autosuficientes,

Por otro lado, los alumnos que suelen tener un interés más práctico, valoran entablar la menor cantidad de interacciones posibles y valoran mucho más la eficiencia del flujo de trabajo y su tiempo. Generalmente se perciben a sí mismos como no tan atentos a las tareas y algo desconcentrados.

A partir de esto vamos a caracterizar los tres grupos de estudiantes con tres nombres el primero sería un alumno auto suficiente  $(Alumno\ S)$ , el segundo vendría ser un alumno regular  $(Alumno\ R)$  y el tercero vendría ser un alumno práctico  $(Alumno\ P)$ .

También podemos caracterizar la forma en que persiguen sus objetivos, los distintos actores en el sistema, esta puede variar dependiendo del proceso en el que se esté, la experiencia previa y las preferencias personales. Cabe destacar que no responden específicamente las caracterizaciones de cada uno de los alumnos es decir un  $Alumno\ P$  aun así puede querer valorar o perseguir la privacidad al consultar información y también un  $Alumno\ S$  puede perseguir la funcionalidad de los recordatorios.

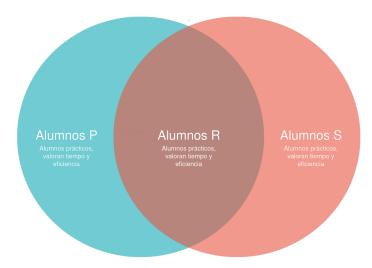


Figura 3.1: Diagrama de Venn: que representa los tipos de estudiantes encontrados en el Focus Group.

Como conclusión se puede ver una tendencia a la personalización de la mayoría de los elementos funcionales, a sí mismo el contenido. En el caso de la privacidad, se debería poder elegir tanto que permisos son dados así como su duración, en el caso de las notificaciones poder establecer su frecuencia en cualquier momento, en el caso de las sugerencias se accede a entregar más información o recibir sugerencias siempre y cuando esto sea establecido por el usuario y que la información dad tenga el carácter de input temporal y no almacenado. Así mismo se establece que el tutelaje debe ser definido por el usuario. Todo esto va en línea con la investigación preliminar que sugiere que los usuarios valoran la personalización y lo relevante de la información para cada usuario suele ser completamente diferente. Todo lo anterior, hace que este trabajo esté orientado en la dirección de las necesidades y preferencias del estudiantado, al mismo tiempo que va en línea con los descubrimientos y los trabajos

relacionados.

#### 3.1.4. Nuevos Modelos de Interacción

En esta sección se detalla las problemáticas de diseño de los nuevos modelos, luego se condensan los tipos de contenido que debería tener el sistema, las funcionalidades y las modalidades de interacción que surgen de las preferencias de los usuarios

#### Problemáticas de diseño de los nuevos modelos

A partir de la diversidad de opinión y de preferencia que se extrae de la investigación a través de FG, podemos ver que la complejidad del modelo subyacente a tales impresiones es alta. Por la misma razón, el sistema va a tener una manera diferente de concretar los objetivos de estos actores dependiendo de cuáles sean estos objetivos y cuáles sean las tareas o requisitos asociados a tales objetivos. A raíz de esto, se hace extremadamente necesario poder especificar con claridad cada uno de los requisitos y sin embargo poder englobarse en funcionalidades que respondan de manera unilateral a las necesidades de las personas, es decir a través de la misma interfaz, la del chatbot. Por eso es que antes de involucrarnos en el diseño funcional o computacional o generar algún diagrama como el de arquitectura o de clases. Se hace necesario plasmar de manera clara las diferentes necesidades de los alumnos en el sistema, las características de estos objetivos, sus relaciones y como pueden ser llevados a la concreción.

Estas diferencias de parecer y de uso, hacen que el sistema a crear deba ser altamente configurable, es decir que cada usuario puede definir que quiere y cómo lo quiere. El cómo responder a las necesidades de los distintos usuarios determina las funcionalidades del sistema, así mismo el qué o la propia necesidad por un contenido específico determina cuál será la información del sistema. A partir de esto, podemos resumir lo que buscan los diferentes usuarios en el sistema en dos grandes categorías contenido, es decir fechas, definiciones, actores relacionados, requisitos, consecuencias, etc. y funcionalidades, tales como recordatorios, consultas, entre otros.

Desde esta separación preliminar se proponen diferentes definiciones y diagramas, los que ayudan a encapsular y explicar de manera concisa y clara el modelo de interacción de alumnos y alumnas, sin perder el foco en los objetivos de los mismos.

Para lograr esto se propone la adopción de la simbología del lenguaje gráfico i\* [Dalpiaz2016], con el fin expresar de forma correcta a los actores involucrados, sus preferencias y asimismo las tareas subyacentes que el sistema debería realizar para completar los objetivos que cada uno de estos actores tiene.

A partir de la investigación del estado del arte, la exploración a través de focus groups y el análisis posterior, se define lo siguiente:

#### Contenidos del Sistema

Los contenidos que buscan los alumnos a través del bot los podemos agrupar en cuatro categorías Definiciones, Requisitos, Fechas y Actores.

las Definiciones son la descripción o detalle del proceso, a grandes rasgos, es una explicación de se trata el proceso, como su nombre lo expresa muy bien, cumplen un rol *educativo*, que es informar al usuario de que es lo que se trata este proceso. La idea de este tipo de información es que el usuario puede entender en una sola consulta que es el proceso, de que trata, que consigue con él, de forma eficiente y amigable.

Los Requisitos son todas aquellas cosas que serán solicitadas durante el proceso, y que permiten el avance durante el mismo. El alumno busca enterarse de cuáles son estas cosas, que tareas o asignaciones debe realizar para lograrlo y ejemplos válidos de una tarea completada. Por ejemplo, un ejemplo de requisito para el proceso de práctica sería realizar la preinscripción de práctica. Al consultar en el sistema el alumno debería obtener el requisito junto a una explicación de en qué consiste y un ejemplo válido de cómo realizarla.

Las fechas son todas aquellas fecha relevantes en un proceso, en general definen plazos o hitos durante el proceso. Por ejemplo la fecha de entrega del primer avance en el CC6908

Los actores son todos aquellos entes relacionados con un proceso, tienen cuatro características que serían: el rol que cumplen dentro del proceso (el cargo o responsabilidad dentro del proceso), el contacto (medio para localizarlos), proveen de un servicio dentro del proceso (las tareas asignadas a su rol) y tendrían una asociación o un grupo al que pertenecen, por ejemplo dentro de una categoría "tipo profesor. este podría pertenece al rol de . Académicos. Universidad", mientras que una institución que provee de prácticas podría pertenecer a la categoría de . Externos. "Lugares de práctica".

#### Funcionalidades relativas a la información

Las funcionalidades buscadas se puede agrupar también en 4 grupos: Consultas, Sugerencias, Intervenciones y Recordatorios.

Las Consultas son un tipo de interacción ya presente y que consiste en acceder al bot para realizar alguna consulta sobre la información de un determinado proceso, esta información podría ser cualquiera de los contenidos listados arriba.

Las Sugerencias son de un formato similar a las consultas, pero son entregadas sin ser solicitadas directamente, debe poder activarse y desactivarse, además de que no deben ser entregadas en cualquier momento, sino ojalá con una periodicidad predeterminada por el usuario, buscan ayudar al usuario a encontrar o enterarse de contenido que podría serle de utilidad sin ser buscado directamente, es decir a partir de su interacción y usuarios con características similares.

Los recordatorios son un tipo especial de notificación que es gatillada expresamente por el usuario quien les fija una periodicidad o frecuencia, buscan ayudarlo a programar y efectuar

los trabajos de manera efectiva. Tienen la opción de requerir feedback si el usuario así lo desea, para motivar el compromiso.

Las intervenciones son habilitadas por el usuario y permiten al bot enviar notificaciones que alteren o podrían alterar el curso conocido del proceso, estas notificaciones pueden ser enviadas tanto al estudiante cómo al estudiante, cómo a un tercero que estudiante defina, en variados casos. Por ejemplos, los cambios en los requisitos serían una intervención, y esa modificación sería enviada al alumno. Por otro lado un cambio de fechas no siempre se categorizan como intervención, dependiendo del alumno, y del plazo. Por esta razón las intervenciones tienen un periodo de vigencia, en el que son relevantes y deberían proveer idealmente suficiente marco de acción para tomar medidas pertinentes al caso. También podrían ser provocadas por el estudiante al no responder a una interacción pactada como los recordatorios, siempre y cuando esto haya sido habilitado y parametrizado por el alumno.

#### Modalidades de Interacción

la mayoría de reticencias o preferencias de los alumnos, respecto a como se deben llevar a cabo las tareas asociadas a los objetivos de contenido o funcionalidad, se pueden agrupar en 5 categorías Privacidad, Parametrización, Eficiencia, Individualidad y Confiabilidad.

la Privacidad hace referencia a la información que el alumno debe entregar para obtener cierta funcionalidad o contenido. En general se busca que exista la mayor privacidad posible, pero algunas interacciones contienen un *trade-off* entre privacidad y eficiencia por ejemplo, que algunos alumnos están dispuestos a aceptar.

La Parametrización tiene que ver con la configurabilidad de la funcionalidad en cuestión, dependiendo del estudiante es más o menos valorada, y esto implica que el sistema debe dar las opciones de parametrización al tiempo que simplificarlas dependiendo del alumno.

La Eficiencia hace referencia a que tan bien puedo cumplir la tarea en cuestión, usualmente relacionada a obtener más recursos o simplificar las parametrizaciones de una funcionalidad determinada.

La Individualidad propone la autonomía del usuario en esa funcionalidad, por ejemplo, buscar contenido de memoria o de práctica sin avisar a otros actores. O no avisar inmediatamente al profesor guía de una omisión sin que el alumno haya consentido esto.

La Confiabilidad por último hace referencia tanto a la validez de la interacción como la confianza que alumno puede descargar en tal interacción. Por ejemplo, que los contactos consultados de un determinado actor estén vigentes.

#### Resumen conceptual del modelo de Interacción

este resumen pretende expresar de manera breve, los diferentes contenidos, funcionalidades y modalidades a través de un diagrama jerárquico de clases o de conjuntos. Cómo se podrá observar más tarde en la sección ??, las funcionalidades determinan de qué forma se

quiere obtener un determinado recurso del sistema, son la interfaz funcional del sistema. Por otro lado los contenidos, son los datos que el sistema debe almacenar para posteriormente servir, estos a su vez definen modelos de datos de la información del sistema. Finalmente las Modalidades hacen eco de los valores presentados en la interacción alumno-bot.

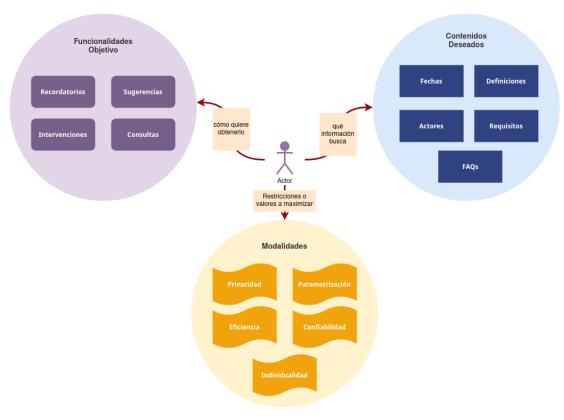


Figura 3.2: Diagrama que busca plasmar las superclases en las que se agrupan los objetivos de los alumnos a sí como las funcionalidades del sistema. Todo agrupado en categorías

# 3.2. Análisis de requisitos

->Conclusión y esquema de focus groups.

# 3.2.1. Diagramas de Funcionalidades y Objetivos

Para poder plasmar nociones como seguridad, individualidad y personalización, es necesario interactuar varias veces con un cliente. Además, para un diseño que integre visiones opuestas de lo que se requiere, es necesario plasmar alternativas diferentes de un mismo servicio. Estas complejidades se expresan y se detallan a través de de diagramas i\* (Cuya sintaxis básica y fundamentación se puede encontrar en el marco teórico).

Por lo tanto los siguientes diagramas plasman las preferencias de los usuarios obtenidas de los focus groups, y permiten el diseño y discusión del proceso cómo un todo. Este proceso

simplifica el diseño general y permite que la discusión se centre en lo que se requiere lograr y cómo lograrlo, y no en cada camino para llegar a ello especificamente.

A continuación se explican las funcionalidades y objetivos esperados por los usuarios, a través de 4 diagramas Consultas, Intervenciones, Recordatorios y Sugerencias.

#### Consultas

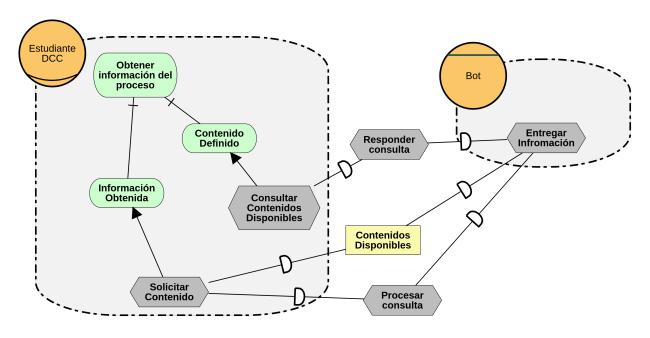


Figura 3.3

Una de las principales necesidades de los estudiantes en torno a los procesos académicos y administrativos es obtener información, en el diagrama esquematizamos este proceso. A través de los focus groups, (más la observación de las interacciones actuales de los alumnos con otros como: chats de alumnos, correos a secretaria docente, profesores entre otros. Cómo se puede apreciar un modelo de información simple de consulta y respuesta, son los dos objetivos secundarios del estudiante, es decir, para obtener información sobre un proceso debe tener una inquietud y ser respondido, es decir Definir un contenido que desea conocer y luego recibir la información relativa al mismo.

Ahora para lograr eso el usuario debe solicitar un contenido al bot, quien procesara la consulta, luego entregará el contenido disponible y el alumno podrá seleccionar que contenido desea conocer, a lo que el bot ayudará a completar respondiendo la información por el proceso seleccionado. Esta interacción simple en dos pasos, engloba varias funcionalidades especificas. Por ejemplo los contenidos se entregan y consultan de formas diferentes. También hay ciertos contenidos que se requieren con más o menos frecuencia. Sin embardo como se puede observar, este diagrama representa lo que se quiere lograr y cómo lograrlo a modo general.

Esto permite un diseño lógico posterior de todas las funcionalidades que se comportan de manera similar puedan ser entendidas bajo el alero de esta descripción.

Cómo veremos en detalle en la sección siguiente esto permite: Diseñar los diferentes mecanísmos con un ordén lógico y manteniendo siempre el foco en el problema. Por otro lado, permiten que la comunicación y extensión de funcionalidades similares se pueda explicar de forma sencilla a nuevos desarrolladores uno de los objetivos del proyecto.

Ejemplos de lo expresado son que por ejemplo el FAQ es una forma de consultar información. Otra modalidad de consultas también recae sobre este modelo, y a pesar de que un método use una vía de comunicación distinta, incluso ejecuten procesamientos de datos difenrentes, todos ellos responden a la misma funcionalidad, esto permite extender el esta área del sistema, aun estando bajo la lógica de retribuir información que el usuario consulta dentro de un catálogo

#### Recordatorios

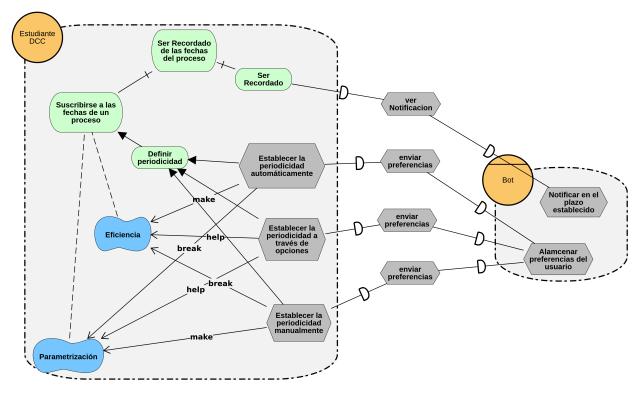


Figura 3.4

Otra de las funcionalidades que estudiamos fue la de ser recordados sobre las fechas de algún proceso académico. Cómo, por ejemplo, durante el proceso de titulación: la fecha para definir el tema, definir profesor guía, entregar el informe de propuesta, etc.

A partir del análisis también se obtuvo que existían 2 cualidades o características que se valoran en cuanto a ser recordados, 1 es la eficiencia y otro es la parametrización (nota al pie: la definición de cada una se puede ver en la sección anterior). La eficiencia en este caso hace referencia a que si me suscribo a un recordatorio quiero que el sistema sepa como debo ser recordado, de forma de hacer el proceso simple, rápido y por sobre todo que me

asegure el resituado. La parametrización, por otro lado, tiene que ver con establecer todos los parámetros que se consideren relevantes, cómo: Cuando soy notificado.

En este sentido, los alumnos propusieron 3 maneras de establecer la periodicidad de una subscripción o recordatorios: automáticamente (el sistema define predeterminadamente cuando), a través de opciones predefinidas: 2 semanas, 1 mes, 1 día, etc., algo que aplicaciones como outlook o google calendar han recogido y permiten en sus sistemas.

En el diagrama, si bien se puede observar que existen las tres alternativas favorecen de maneras diferentes las cualidades valoradas por los alumnos, si bien, por un lado, establecer la periodicidad manualmente hace que la aplicación sea parametrizable, la ha menos eficiente, el caso contrario ocurre al establecer la periodicidad automáticamente.

Una de las conclusiones importantes que se saca de este diagrama es que tener las 3 alternativas es la única forma de favorecer a todos los usuarios de sistema en cuanto a sus preferencias. Sin embargo, eso implica desarrollar 3 vías adecuadas especialmente para ese objetivo.

Otra conclusión interesante que se extrae de diagrama es que el bot, debe guardar una configuración específica para cada estudiante, en orden de favorecer las cualidades que son importantes para él. Si bien esto podría eventualmente ser un riesgo de privacidad, tanto desde lo extraído de los alumnos como lo la forma en que está creado el bot, hacen que se puedan guardar estas preferencias de forma anónima.

Y finalmente, como es de esperarse, el bot debe ser capaz de notificar también en una fecha particular a cada alumno.

Este diagrama, si bien en pocos pasos, da cuenta de una complejidad subyacente en torno a diseñar e implementar una funcionalidad puede hacer que un grupo de alumnos se vea favorecido y otro perjudicado, estas conclusiones son relevantes a la hora de priorizar o establecer metas de desarrollo.

#### Sugerencias

Las sugerencias son un tipo de funcionalidad pensada en alumnos que no tienen claro que deben hacer, se adelanta a las posibles dudas del alumno y entrega recomendaciones sobre alguna de las alternativas disponibles. Esta fue una de a las funcionalidades más contradictorias en términos de lo valorado por los alumnos y esto se puede observar en las cualidades adjuntas a esta funcionalidad.

En primer lugar, este es un sistema particularmente individualizado, esto quiere decir que las sugerencias que son particularmente específicas a cada alumno, sobre todo porque algunos de lleno no quieren recibir sugerencias.

Es una funcionalidad también que es recibida con un poco más de recelo por parte de los mismos estudiantes, porque para que el sistema pueda dar sugerencias debe tener más información de los alumnos, la que podría hacerlos identificables y con ellos las problemáticas que podrían surgir a partir de exponer las preferencias de cada uno.

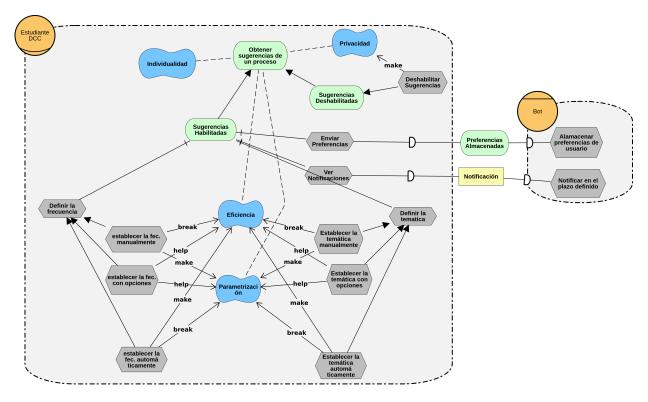


Figura 3.5

Al mismo tiempo, es extremadamente eficiente para un grupo de alumnos que hubieran querido más guía y asesoría en su proceso académico, sobre todo que alguien se adelantara a lo que les pudiera hacer falta y les avisara con tiempo.

También se observa una discusión similar a los recordatorios en cuanto, a la parametrización y eficiencia.

Ahora algo interesante, aunque muy similar a establecer los plazos, sin duda una de las cualidades clave de este sistema, sería establecer la temática de las sugerencias. Uno de los grandes problemas que el bot busca resolver, es el hecho de que a pesar de tener mucha información disponible, los alumnos continúan con dudas e incertidumbres. Esto debido a dos razones principales, 1 es el exceso de información no relevante y la otra el desconocimiento de los canales existentes. Un sistema que agrupe la información sería valioso para solucionar el problema de tener múltiples canales, y una mejora significativa en la experiencia se daría si los alumnos reciben información que es importante para ellos y no cualquier cosa.

Esta es una discusión particularmente especial, porque es similar a las disyuntivas que presenta el desarrollo de redes sociales, y este aspecto del bot se ve muy influenciado por las percepciones que los alumnos tienen de las políticas establecidas en cada una de ellas.

#### Intervenciones

Las intervenciones, soporte o tutelaje tienen el objetivo de incluir a un tercero en el recorrido del alumno por un proceso, y que este pueda ser una voz amigable para revisar

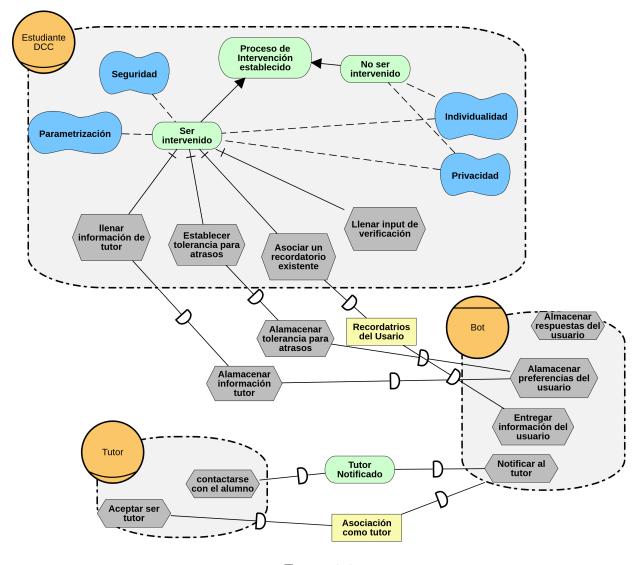


Figura 3.6

el estado del proceso. Este "Tutor" cómo se denomina, es notificado cuando el alumno no interactúa con los elementos a los que el mismo se suscribió, esta funcionalidad, busca que el alumno pueda recibir apoyo cuando por diversas razones podría no estar dándole la atención necesaria al proceso. Esta funcionalidad, al igual que las sugerencias, se adelanta a que el alumno pudiese tener un problema o atrasarse en los plazos.

Cómo se puede ver en el diagrama, las cualidades de individualidad y privacidad son factores determinantes a la hora de decidir si usar o no esta funcionalidad. Por otro lado la tranquilidad que puede brindar al estudiante el tener un soporte externo que puede ser cualquier persona, es uno de los mayores incentivos para agregar esta feature.

Otra de las razones por las que puede ser controversial es porque debe almacenar interacciones del alumno en el sistema, se deben guardar contactos de terceros, etc.

# 3.3. Conclusiones generales de Análisis

A modo general, desde los diagramas se pueden obtener varias nociones que pueden guiar el trabajo. Se presentan las funcionalidades a realizar por el sistema en las tareas que deben realizar los distintos agentes involucrados (hexagonos grises). Y, por otro lado, las cualidades vienen a ser restricciones y alternativas a las mismas funcionalidades. También algunos de los contenidos quedan explícitos como recursos (rectángulos amarillos). Así mismo todos los actores relevantes para una funcionalidad. Siempre teniendo a la vista cuál es el objetivo final de cada una de las funcionalidades globales.

Esto permite distinguir entre los requisitos de usuarios (objetivos y cualidades) y los requisitos de software (tareas, recursos y cualidades) de manera global, muy útil en la dirección estratégica de proyectos. Mantener este tipo de nociones resulta fundamental en el desarrollo orientado a usuarios, porque es muy fácil que el quehacer del proyecto se puedan perder los horizontes originales y perder de vista el objetivo por el cual se desarrolla el sistema.

Así mismo, hay que notar que estos diagramas representan un escenario ideal, en el que se extraen las necesidades, objetivos y restricciones de manera explícita en el planteamiento de una solución, pero como todo desarrollo de software, se deben priorizar las funcionalidades de mayor importancia y también las que sean alcanzables durante la implementación, respetando las reglas tiempo de todo desarrollo.

A partir entonces del análisis del focus group, de los diagramas, y el traspaso del trabajo anterior. Se establecen a modo general que los principales requerimientos que cuadran mejor con los objetivos de: "extender el sistema actual, agregando funcionalidades que permitan crear un sistema extensible, personalizado y confiable." Son las consultas y recordatorios. Ambos agregan valor deseado por los usuarios, permiten personalizar su experiencia en cuanto a la información que desean conocer y son los que presentan la mayor aceptación en torno a privacidad y otras cualidades valoradas por los usuarios.

También se hace necesario analizar y modificar el sistema existente para que sea capaz no solo de cumplir con la funcionalidad esperada, sino ser extensible de manera lógica, para garantizar el desarrollo continuo de la plataforma.

# 3.4. Diseño de plataformas

A partir de los requerimientos recabados en el trabajo anterior, se diseñaron varios diagramas UML, que permitieron esquematizar las principales funcionalidades y cambios a realizar en el sistema. Eso sí, es fundamental decir que a estos diseños sufrieron modificaciones durante el transcurso del proyecto y algunos como el diagrama de objetos obtiene su versión final en el modelo de datos. La versión de los diseños que se presenta aquí es la obtenida a partir del análisis de los requerimientos.

#### 3.4.1. UML

En primer lugar se diseñaron los casos de uso a considerar en el sistema:

img Casos DE USO

Durante el planteamiento de los casos de uso, se agruparon varias de las tareas, sobre todo porque desde los diagramas se extrae que si bien el contenido puede variar, la funcionalidad deseada es la misma. En resumen del diagrama se puede extraer que el alumno podrá hacer 3 cosas principalmente en el sistema: Buscar información, suscribirse a información y habilitar permisos.

Cada una de estas funcionalidades clave tiene muchos pasos clave y diseños asociados, por eso se mostrará a continuación el diagrama de estados porque es representativo de las interacción más compleja del sistema.

En el diagrama anterior se puede observar que el alumno debe seleccionar primero un proceso vigente y a partid de ahí seleccionar alguno de los contenidos disponibles. Definido esto, se procede a elegir la funcionalidad deseada en torno al contenido y finalmente se solicita al alumno aceptar o no el almacenamiento de los datos, equivalente a permitir o rechazar la funcionalidad. La unica salvedad es que si estos hubieron sido aceptados en el pasado o configurados globalmente se puede pasar directamente a la funcionalidad esperada en este caso los recordatorios.

Una de las principales conclusiones de este diseño es que un diseño basado en una interfaz, es que necesariamente consta de varios pasos, los cuales se pueden simplificar y trabajar, pero añade muchas permutaciones en torno a los posibles viajes del usuario, y también tiene un alto costo de diseño. A sí mismo, si además de establecer este viaje default agregamos la opción de que el usuario pueda parametrizar la frecuencia, se añaden muchas opciones de interfaz, lo que termina haciendo al sistema engorroso y lento de usar.

Es por esto que luego del diseño UML de varias de las plataformas se diseñaron dos formas paralelas de acceder a las funcionalidades, para mantener la simplicidad y la eficiencia, por un lado, y la parametrización y control sobre otro.

Diseño Dual de interfaces y comandos Uno de los grandes hallazgos del proceso de diseño, es que las interfaces no se podían comportar de forma dual (eficiente y parametrizable), ya que se habrían tenido que añadir una gran cantidad de opciones de ajuste, esto al final redunda en un viaje de usuario muy largo, lo que claramente perjudica a los alumnos de tipo P.

A partir de esto entonces se propuso un nuevo sistema que funciona de manera paralela a las interfaces basadas en botones y es una basada en comandos. Los comandos son una función bastante común en Telegram, ya que la mayoría de los bots ofrece un funcionamiento basados en comandos, esto permite no solo identificar una opción a realizar, sino que agregar en el mismo comando texto y valores que pueden servir cómo parámetros para dichas funciones. Este tipo de interfaz, más parecido a una shell o terminal, es bastante familiar para los alumnos del dcc y además usuarios de Telegram, en ese mismo sentido proporcionan una enorme flexibilidad y extensibilidad, dado que a partir de un mismo identificador se pueden

efectuar muchos ajustes y configuraciones.

#### Diagrama de objetos

A partir de los casos de usos se creó el siguiente diagrama de objetos que da cuenta de las principales relaciones entre los objetos del sistema. Uno de los grandes cambios al modelo de original es la creación de un Usuario (o perfil) que debe servir como el centro para las suscripciones, los permisos y el usuario de apoyo. Así mismo se identifica que deben agregarse tablas de contenido para las instancias. Finalmente, una de las grandes modificaciones es que se necesita una forma de mantener un catálogo de las cosas a las que los usuarios pueden acceder, particularmente suscribirse, las que no pueden quedar simplemente en el código, ya que por naturaleza son mutables de instancia a instancia, eso implicaría agregar muchas líneas de código o quitar muchas para cada instancia de cada proceso en particular.

El problema principal con este catálogo, es que se debe diseñar alguna forma de mantenerlo actualizado, por ahora el diseño es manual, pero en un futuro podría ser de forma automática.

#### 3.4.2. Modelo de datos

• User: Información de acceso al sistema de administración.

Relativos al Proceso

- Proceso: Contiene los procesos que existan. Ejs: Proceso de Titulación, Prácticas, etc.
- Categorías: Categorías de las preguntas frecuentes.
- FAQ: Preguntas Frecuentes.

Relativos a una instancia

- Instancias: Instancias de un proceso, por ejemplo: Proceso de Titulación 2022.
- Novedades: Son las noticias, pero relacionadas a una instancia en particular del proceso.

Usuarios del Bot

- Usuarios del Bot: identificador del chat.
- Permisos de los Usuarios del Bot: Configuraciones de funcionalidad y seguridad de los usuarios.

# Capítulo 4

# Implementación

### 4.1. Reestructuración

El código existente para el bot contenía varios problemas de diseño que hacían compleja su extensibilidad y se pueden revisar detalladamente en la sección de trabajo previo.

Para solucionar estos problemas primero se extrajeron las funcionalidades clave del bot: Obtener información desde la base de datos, recibir request, procesar request y enviar mensajes. La extracción de información, la recepción y envío de mensajes en sí no presentaba mayores problemas de diseño.

## 4.1.1. Procesar request

#### De if a Parser

Cómo se vio en la sección de trabajo previo, la API de telegram permite varias formas de comunicación. De ellas se utilizan los websockets. A partir de aquí, los bots pueden recibir todos los mensajes enviados por el usuario, sin embargo, telegram permite agrupar los mensajes en más o menos 4 modalidades: primero texto, archivos o media adjunta, comandos y los keyboards.

En el diseño anterior, como se explicó, se seleccionaron principalmente una manera de comunicación a través de inline\_keyboards. (sección anterior: En este tipo de comunicación se presentan al usuario una serie de botones que el puede apretar y a partir de los se envia un mensaje al servidor, cuando este es procesado se envía la respuesta al usuario de manera asincrona). Para poder desarrollar las nuevas features, se añadieron comandos, en miras tambien de procesar texto. Se amplió el diseño para pasar de una decisión condicional a un parsing con double dispatch.

#### Sistema de procesamiento anterior

Sección de Código 4.1: Ejemplo de fixture para el proceso de postulación a becas

```
if label is None:
                  if message == '/start':
                  messages = [ {"text": 'Hola_...',
                   "keyboard": {}},...]
                  elif message == '/preguntasFrecuentes': ...
                  elif message == '/asistente': ...
              Else: ...
                  elif label == 'Process': ...
                  elif label == 'Category': ...
                  elif label == 'Question': ...
                  elif label == 'Helper':
              for msg in messages:
                  self.send_message(msg['text'],
                   t_chat["id"], msg['keyboard'])
Nuevo sistema de procesamiento:
          def post(self, request: AsgiRequest, *args, **kwargs):
          update = request.body
          parser.decode_update(update)
          return JsonResponse({"ok": "POST request processed"})
         # Parser
          def decode_update(self, update: bytes):
                  # Se detecta que el update es un mensaje
                  if update.match(message-regex):
                  return self.parse expression (message)
                  # Se detecta que el update es una call_bacK_query
                  elif update.match(call_bacK_query-regex):
                  self.parse expression(label, chat id)
                  # No see detecta que es el update
                  else:
                      return Updates. UpdateProccesingResult()
```

messages = []

def message\_processing(self, t\_chat, message, label=None, que

Cómo se puede apreciar en este nuevo sistema, el parser se hace cargo de determinar que tipo de update o actualización fue recibida desde telegram y a partir de esa detección se pasa la expresión recibida a un handler, este determina que acción debe realizarse y luedo llama a la instancia determinada de la acción para que se procese la request. Este nuevo modelo es util principalmente por 5 razones. Mantiene la integridad del procesamiento sin importar si el tipo mensaje de telegram es conocido o no. Separa el parsing de la estructura de la api de Telegram del procesamiento de la acción requerida por el usuario en el sistema. Dualidad Parser-Handler. Permite escalar de forma ordenada el código para recibir muchas nuevas funcionalidades siguiendo un diseño estándar de procesamiento. Esto porque separa la

identificación de la acción de su código en particular, estandariza el llamado de las acciones y permite reutilizar patrones de código sin duplicar código. Utiliza un parsing basado en regex y no en comparaciones de strings, lo que permite una mayor flexibilidad y robustez del procesamiento. Da la posibilidad de añadir nuevos tipos de procesamiento de acciones sin cambiar la lógica de traducción de la API del bot. En otras palabras, para añadir procesamiento de lenguaje natural, no hay que hacer grandes modificaciones al parser, solo hay que añadir el handler correspondiente.

something in the appendiex

something in the appendiex