

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA INFORMÁTICA

## **Análisis modular de las funcionalidades de Discord**

**Felipe Balbontín Gallegos  
Christopher Catalan Atton  
Joel Flores Sottile  
Patricio San Martín Carreño**

INFORME DE AVANCE DE PROYECTO  
PARA EL RAMO DE MODELAMIENTO DE  
SISTEMAS DE SOFTWARE

NOVIEMBRE, 2016

## Resumen

Poder hablar en vivo y compartir información en un contexto hablado mediante micrófono y/o por mensajería instantánea es de gran provecho para equipos de trabajo que necesitan de una conversación fluida para llevar a cabo sus tareas con el mayor entendimiento posible como base y reduciendo los tiempos de realización gracias a la comunicación continua.

Mediante este informe se pretende ilustrar los modelos de las funcionalidades más relevantes del sistema que compone a Discord e indicar los requerimientos asociados a las esas funcionalidades. En pos de esto es que se adjuntan tablas de requerimientos, diagramas de los distintos servicios e información que guía a comprender mejor aquellos diagramas.

**Palabras Clave:** comunicaciones, modelos, canal, Discord, interacción, diagramas.

# Índice

<b>1. Introducción</b>	<b>1</b>
<b>2. Objetivos del trabajo</b>	<b>2</b>
2.1. Objetivos Generales . . . . .	2
2.2. Objetivos Específicos . . . . .	2
<b>3. Planificación</b>	<b>3</b>
3.1. Modelo de proceso a utilizar . . . . .	3
3.2. Equipo de trabajo . . . . .	3
3.3. Carta Gantt . . . . .	4
3.3.1. Carta Gantt General . . . . .	4
3.3.2. Carta Gantt Felipe Balbontin . . . . .	5
3.3.3. Carta Gantt Christopher Catalan . . . . .	5
3.3.4. Carta Gantt Joel Flores . . . . .	6
3.3.5. Carta Gantt Patricio San Martin . . . . .	6
3.4. Evaluacion de riesgos y propuestas . . . . .	7
<b>4. Descripción del problema</b>	<b>8</b>
4.1. Origen de la problemática y motivación . . . . .	8
4.2. Justificación de la necesidad presentada . . . . .	8
4.3. Proyecciones de implementación y puesta en marcha . . . . .	9
<b>5. Especificación de requerimientos</b>	<b>10</b>
5.1. Requerimientos funcionales . . . . .	11
5.1.1. Módulo 1: Enviar archivos . . . . .	11
5.1.2. Módulo 2: Gestión de usuario . . . . .	11
<b>6. Modelos</b>	<b>13</b>
6.1. Diagramas de Flujos de Datos . . . . .	13
6.1.1. Módulo 1: Enviar archivos . . . . .	13
6.1.2. Módulo 2: Gestión de cuenta . . . . .	15
6.1.3. Módulo 3: Administrar usuarios de un servidor . . . . .	18
6.2. Diccionario de Datos (DFD) . . . . .	20
6.2.1. Módulo 1: Enviar archivos . . . . .	20
6.2.2. Módulo 2: Administrar usuarios	
21	
6.2.3. Módulo 3: Gestión de usuario . . . . .	22
6.3. Modelo entidad relacion . . . . .	24

6.3.1. Diccionario de datos ER . . . . .	25
<b>7. Enfoque orientado a objetos</b>	<b>29</b>
7.1. Diagramas de casos de uso . . . . .	29
<b>8. Conclusión</b>	<b>30</b>

## Lista de Figuras

1.	Carta Gantt General . . . . .	4
2.	Carta Gantt Felipe Balbontin . . . . .	5
3.	Carta Gantt Christopher Catalan . . . . .	5
4.	Carta Gantt Joel Flores . . . . .	6
5.	Carta Gantt Patricio San Martin . . . . .	6
6.	DFD de contexto del módulo 1 . . . . .	13
7.	DFD de nivel 1 del módulo 1 . . . . .	13
8.	DFD de nivel 2 (1.1) del módulo 1 . . . . .	14
9.	DFD de nivel 2 (1.2) del módulo 1 . . . . .	14
10.	DFD de nivel 2 (1.3) del módulo 1 . . . . .	14
11.	DFD de contexto del módulo 2 . . . . .	15
12.	DFD de nivel 1 del módulo 2 . . . . .	15
13.	DFD de nivel 2 (1.1) del módulo 2 . . . . .	16
14.	DFD de nivel 2 (1.2) del módulo 2 . . . . .	16
15.	DFD de nivel 2 (1.3) del módulo 2 . . . . .	17
16.	DFD de contexto del módulo 3 . . . . .	18
17.	DFD de nivel 1 del módulo 3 . . . . .	18
18.	DFD de nivel 2 (1.1) del módulo 3 . . . . .	19
19.	DFD de nivel 2 (1.2) del módulo 3 . . . . .	19
20.	Modelo Entidad Relacion . . . . .	24

## Lista de Tablas

1.	Tabla Carta Gantt . . . . .	4
2.	Requerimientos funcionales del módulo de envío de archivos .	11
3.	Requerimientos no funcionales del módulo de gestión de usuario	12
4.	Diccionario de Enviar archivos . . . . .	20
5.	Diccionario de Administrar usuarios . . . . .	21
6.	Diccionario Gestión de usuario Parte 1 . . . . .	22
7.	Diccionario Gestión de usuario Parte 2 . . . . .	23
8.	Diccionario de la Entidad Usuario . . . . .	25
9.	Diccionario de la Entidad Servidor . . . . .	26
10.	Diccionario de la Entidad Mensaje . . . . .	26
11.	Diccionario de la Entidad Canal . . . . .	27
12.	Diccionario de la Entidad Canal Servidor . . . . .	27
13.	Diccionario de la Entidad Canal Interno . . . . .	28
14.	Diccionario de la Entidad Canal Voz . . . . .	28
15.	Diccionario de la Entidad Canal Texto . . . . .	28

## 1. Introducción

En la actualidad el hombre hace uso de las tecnologías muy frecuentemente gracias a que ha encontrado ventajas notables en ellas, tanto para la vida diaria como para acciones en específico. El desarrollo de software ha incidido de buena manera en la actividad humana y en especial en el ámbito laboral, donde constantemente se proponen sistemas que aumenten la productividad y mejoren el resultado del producto. Sin embargo, el proceso para llegar a obtener software de calidad y que realmente sea funcional y aporte a organizaciones o a la comunidad es un proceso lento que requiere un estudio extenso de las necesidades del usuario y también una discusión con los clientes en la que se definan con exactitud los servicios que el sistema debe satisfacer.

Los modelos son un resultado del proceso de conversación con los clientes y futuros usuarios. El modelo es el que orientará el desarrollo de software para que el resultado obtenido coincida con lo conversado entre el desarrollador y los clientes y usuarios. Modelar es importante no sólo porque deja claramente documentado lo que se hará, sino también porque implica la constante comunicación entre los participantes del ciclo de vida del software y esto proporciona un trabajo más eficaz sin confusiones.

En este informe se quiere dar a conocer las funcionalidades de un sistema de comunicación muy útil llamado Discord, y para trabajar en pos de ello se describirán sus procesos más importantes y los requerimientos más relevantes con los que cumple, junto con las razones que nos llevan a estudiar un software gratuito cuyo uso aumenta de manera desproporcionada debido a lo interesante que resulta la propuesta para una comunidad específica de la sociedad.

## **2. Objetivos del trabajo**

Para continuar, enseñaremos los objetivos generales y específicos del trabajo, con los que detallaremos en qué se basará nuestro proyecto y su problemática.

### **2.1. Objetivos Generales**

- Conocer los conceptos necesarios para el diseño de la aplicación.
- Identificar la problemática de Discord, para analizarla detalladamente mediante módulos.
- Modelar 3 módulos de la aplicación Discord, usando los conceptos y habilidades ya conocidas.

### **2.2. Objetivos Específicos**

- Definir una planificación, sabiendo que modelo de proceso utilizar, el equipo de trabajo, una carta Gantt con sus respectivas actividades y recursos, por último, una evaluación de riesgos y propuestas.
- Determinar el origen de la problemática y la motivación para solucionarla, justificando la necesidad de conseguirla.
- Estimar las proyecciones de implementación, para luego determinar el sistema a modelar.
- Especificar los requerimientos funcionales y no funcionales de la aplicación.
- Especificar un diagrama de flujo de datos, con sus correspondientes subniveles y diccionario para los flujos de datos.
- Sintetizar todo lo realizado con una conclusión.

## **3. Planificación**

### **3.1. Modelo de proceso a utilizar**

La metodología de proceso que es necesaria para Discord es el desarrollo incremental. Como Discord va mejorando la calidad de software en todo momento, es inevitable ir actualizándolo cada cierto tiempo, por lo que primero se analiza que se quiere actualizar, para luego finalizar con la instauración y aprobación del sistema. El modelo permite al desarrollador sacar ventaja de lo que se ha aprendido del desarrollo anterior, concretando una versión más completa, realizando cambios en el diseño, agregando nuevas funcionalidades y capacidades al sistema.

### **3.2. Equipo de trabajo**

El equipo de trabajo esta constituido por cuatro estudiantes de la carrera ingenieria civil informatica que imparte la pontificia universidad catolica de valparaiso. Optamos por organizar la mayoría de las actividades en conjunto, con el fin de llegar a una opinión certera de los objetivos

### 3.3. Carta Gantt

#### 3.3.1. Carta Gantt General

Actividades	Fecha de inicio	Duración	Fecha de termino	Integrantes
Duración total proyecto	11-sept	52	02-nov	Todos los integrantes
Elección de temática	12-sept	3	15-sept	Todos los integrantes
Elegir módulos apropiados	29-sept	3	02-oct	Todos los integrantes
Requerimientos de cada módulo	02-oct	7	09-oct	Todos los integrantes
Creación MER	09-oct	7	16-oct	Christopher Catalan
Creación DFD	09-oct	11	20-oct	Felipe Balbontin, Joel Flores, Patricio San Martin
Revisión de modelos creados	20-oct	3	23-oct	Todos los integrantes
Corrección de errores	24-oct	7	31-oct	Todos los integrantes
Creación de diccionarios	27-oct	5	01-nov	Felipe Balbontin, Christopher Catalan, Joel Flores
Creación de informe	15-oct	18	02-nov	Felipe Balbontin, Joel Flores, Patricio San Martin
Preparación presentación	31-oct	2	02-nov	Todos los integrantes
Entrega proyecto parte A	02-nov	1	03-nov	Todos los integrantes

Tabla 1: Tabla Carta Gantt



Figura 1: Carta Gantt General

### 3.3.2. Carta Gantt Felipe Balbontin



Figura 2: Carta Gantt Felipe Balbontin

### 3.3.3. Carta Gantt Christopher Catalan



Figura 3: Carta Gantt Christopher Catalan

### 3.3.4. Carta Gantt Joel Flores



Figura 4: Carta Gantt Joel Flores

### 3.3.5. Carta Gantt Patricio San Martin



Figura 5: Carta Gantt Patricio San Martin

### **3.4. Evaluacion de riesgos y propuestas**

Al tener una gran cantidad de servidores, cada uno debe tener una constante mantención, lo que implica que al ser gratuita no se tendrá una fuente de ingresos. Por lo que para solucionar este riesgo se propone generarlos a través de ventas de skin y paquetes de stickers, un modelo de negocio implementado en otras aplicaciones que ha dado cierto éxito, lo que no influye a su funcionamiento, ya que son detalles estéticos y entretenimiento.

## **4. Descripción del problema**

### **4.1. Orígen de la problemática y motivación**

El desarrollo de las comunicaciones online, en estos tiempos, es vital para el ser humano moderno que busca mantenerse conectado y en contacto con amistades, familiares o colegas. Por otro lado, el progreso en esta área es de gran incidencia para actividades que comprenden el trabajo en equipo. Distintas organizaciones con distintas metas hacen uso de software que les permita trabajar en conjunto sin importar la distancia que separe a los miembros de una mesa de discusión. Para satisfacer estas necesidades, surgen Skype, FaceTime, TeamSpeak, Ventrilo, RaidCall y Mumble, entre los más destacados. Estos medios se distinguen entre sí por pequeñas variables, como el uso de cámara web, el pago por el servicio, la compatibilidad con otros sistemas operativos y muchas otras características propias de cada uno. El problema de estos servicios es que raramente eran actualizados y nunca lograron funcionar como una red social en la que uno diariamente haría ingreso para compartir con sus círculos sociales a través de internet.

### **4.2. Justificación de la necesidad presentada**

En vista de la escasa atención al cliente y el poco tino demostrado en no responder a las sugerencias de la comunidad, nace Discord, una aplicación semejante a las ya nombradas, pero con un enfoque distinto y con soporte para móviles, equipos de escritorio y navegador web, compatibles en distintos sistemas operativos y juntando distintas características de las aplicaciones nombradas anteriormente, pero con la diferencia de que quienes la desarrollan constantemente se preocupan del feedback.

Discord presta servicio gratis a una gran comunidad gamer que puede hacer uso de los servidores que ofrece la aplicación para administrar canales e invitar gente a que se una a ellos para comunicarse mediante voz o texto, además de compartir imágenes, enlaces y distintos archivos de documento digital. Esta característica significa que quien crea el servidor no necesita estar conectado a internet para que su servidor siga activo, por lo que los usuarios invitados al servidor pueden seguir accediendo a los canales del mismo y al contenido compartido en ellos. Considerando que el desarrollo de software de entretenimiento y videojuegos crece a un ritmo acelerado, con distintas plataformas de distribución digital, Discord ayuda a sus usuarios a

encontrar amistades en distintos juegos gracias a la conexión a esas plataformas, además de agregar amigos mediante aplicaciones independientes muy simbólicas del auge de los juegos electrónicos.

### **4.3. Proyecciones de implementación y puesta en marcha**

Implementar un sistema que abarque tantas áreas como lo hace Discord requiere de la participación continua de la comunidad para dar a conocer a los desarrolladores los errores que la aplicación presenta y también para que sepan de lo que la aplicación está careciendo, tomando en cuenta de que tiene un enfoque dado al jugador de videojuegos electrónicos y que cualquier cambio o innovación en la aplicación debería seguir este rumbo orientado a la comodidad para el jugador promedio. Es por eso que las implementaciones de Discord actúan en base a lo que se propone en los foros que ofrecen y en la atención de soporte a los jugadores, donde reciben comentarios muy notables que sirven de feedback para la mejora de este software.

## **5. Especificación de requerimientos**

En esta sección se hace una recopilación de los requerimientos funcionales y no funcionales que se encuentran en Discord, en base a sus funcionalidades más relevantes y las características que hacen de esta aplicación un servicio tan ventajoso y pragmático, para lo que se han propuesto distintos módulos con distinta especificación de requerimientos entre sí.

## 5.1. Requerimientos funcionales

### 5.1.1. Módulo 1: Enviar archivos

Identificación del requerimiento	Descripción del requerimiento
RF001	Los usuarios invitados a un canal de un servidor pueden compartir archivos a través del canal
RF002	Los archivos enviados no superan los 8 megabytes de memoria
RF003	Cada servidor tiene URL única
RF004	Cada canal tiene URL única
RF005	El volumen recibido por cada usuario puede ajustarse
RF006	El usuario puede silenciar a otro usuario
RF007	El usuario se puede cambiar de canal
RF008	El usuario puede bloquear a usuarios del mismo servidor
RF009	Cada usuario tiene una cuenta de correo única asociada a su usuario
RF010	El inicio de sesión se hace con el mail como usuario
RF011	El usuario puede agregar como amigo a otros usuarios
RF012	El usuario puede eliminar de amigos a otros usuarios
RF013	El usuario no puede tener un nickname vacío
RF014	El usuario es capaz de cambiar su contraseña
RF015	El cambio de nickname es confirmado mediante contraseña

Tabla 2: Requerimientos funcionales del módulo de envío de archivos

### 5.1.2. Módulo 2: Gestión de usuario

Identificación del requerimiento	Descripción del requerimiento
RNF001	El emisor de un archivo puede hacer un comentario que es enviado junto con el archivo
RNF001.1	El emisor de un archivo puede editar el comentario hecho sobre el archivo
RNF001.2	El emisor de un archivo puede eliminar el comentario hecho sobre el archivo
RNF002	El archivo enviado al canal puede ser eliminado por quien lo envió
RNF003	Una imagen compartida en un canal puede ser vista con sus propiedades reales en el navegador
RNF003.1	Se puede copiar en enlace de una imagen compartida en un canal sin tener que expandirla
RNF004	Se pueden arrastrar archivos hasta la casilla de texto del canal para enviar archivos
RNF005	El usuario decide si quiere ligar su cuenta a alguna cuenta de otra plataforma
RNF005.1	El usuario puede ligar su cuenta a otra de Battlenet
RNF005.2	El usuario puede ligar su cuenta a una cuenta de Twitch
RNF005.3	El usuario puede ligar su cuenta a una cuenta de Steam
RNF005.4	El usuario puede ligar su cuenta a una cuenta de Skype
RNF005.5	El usuario puede ligar su cuenta a una cuenta de Youtube
RNF005.6	El usuario puede ligar su cuenta a una cuenta de League Of Legends
RNF006	El usuario puede buscar amigos que también usen Discord en las listas de amigos de las cuentas ligadas
RNF007	El usuario decide quién puede agregarlo como amigo
RNF007.1	El usuario decide si miembros de un mismo servidor pueden agregarlo a su lista de amigos
RNF007.2	El usuario decide si cualquier usuario puede agregarlo a su lista de amigos
RNF007.3	El usuario decide si sólo amigos de amigos pueden agregarlo a su lista de amigos
RNF008	La cuenta queda como inactiva si no se ha hecho un inicio de sesión durante un largo período de tiempo
RNF008.1	La cuenta se reactiva al iniciar sesión nuevamente tras un largo período de tiempo
RNF009	El usuario puede asignarse una foto de perfil
RNF009.1	El usuario puede cambiar su foto de perfil
RNF010	El servidor despliega una lista de usuarios de los miembros históricos del servidor
RNF010.1	La lista especifica el estado de los miembros (desconectado, disponible y ocupado)
RNF010.2	La lista está ordenada alfabéticamente para conectados (disponible u ocupado) y desconectados
RNF011	Se puede ver una previa del perfil de algún miembro en el servidor
RNF012	El servidor informa al usuario acerca de su conexión

## 6. Modelos

### 6.1. Diagramas de Flujos de Datos

A continuación se muestran los diagramas de flujo de datos de los módulos estudiados del sistema. Estos fueron desglosados hasta el segundo nivel.

#### 6.1.1. Módulo 1: Enviar archivos

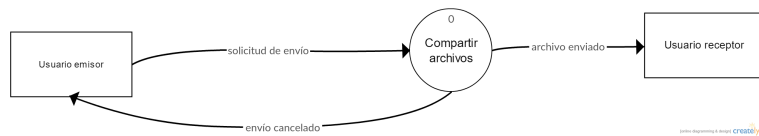


Figura 6: DFD de contexto del módulo 1

En este diagrama se puede visualizar los flujos de datos generales en el envío de archivos en Discord y los terminadores relacionados al proceso. En este caso es el usuario quien interactúa con el proceso para después recibir una respuesta.

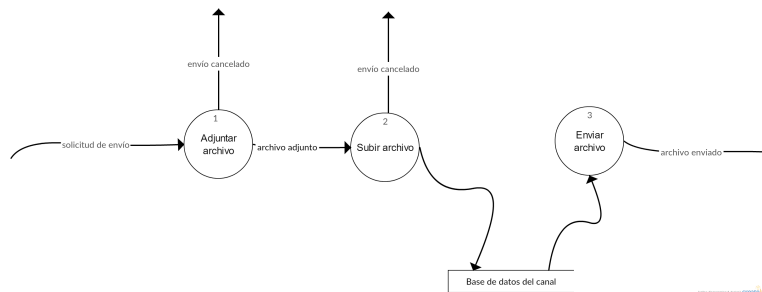


Figura 7: DFD de nivel 1 del módulo 1

Se logra apreciar un diagrama detallado de las interacciones, se desglosa el proceso principal para obtener así los procesos que lo componen. También se puede apreciar la aparición de un depósito de datos que permite la comunicación entre los dos últimos procesos.

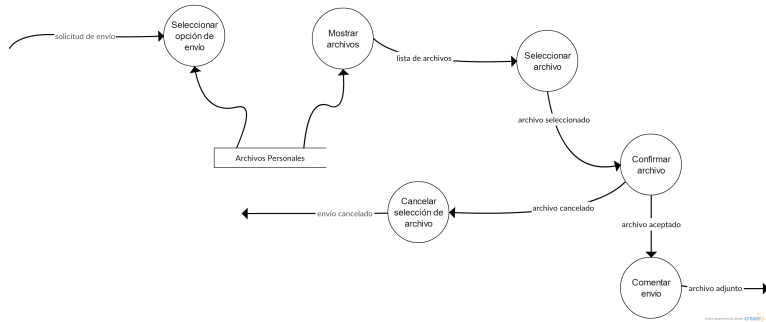


Figura 8: DFD de nivel 2 (1.1) del módulo 1

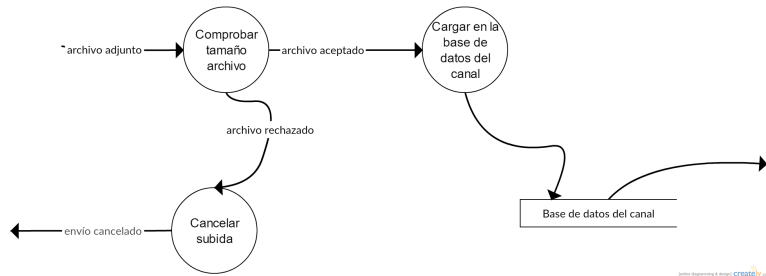


Figura 9: DFD de nivel 2 (1.2) del módulo 1

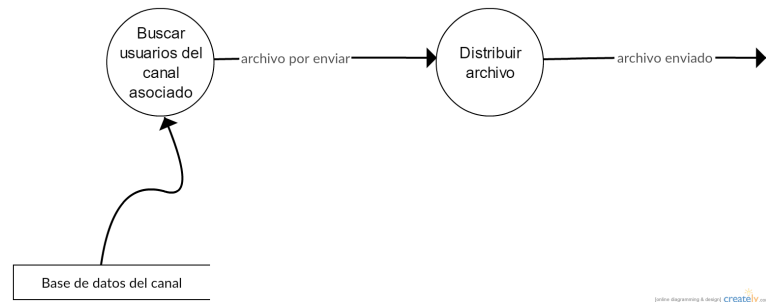


Figura 10: DFD de nivel 2 (1.3) del módulo 1

En las figuras anteriores se aprecia el segundo nivel, donde se ve explícitamente la interacción de los flujos en relación a sus terminadores, se visualiza



niveles de profundidad En la figura anterior se deslosa el proceso principal. Se pueden apreciar los procesos de creación, modificación y eliminación de la cuenta.

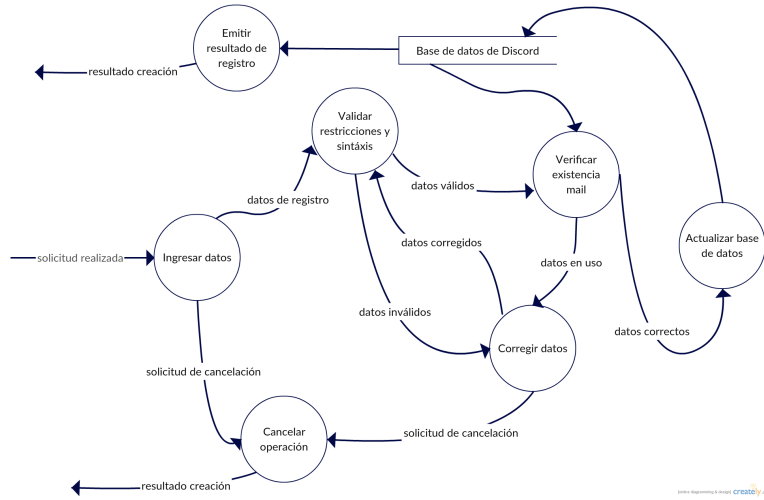


Figura 13: DFD de nivel 2 (1.1) del módulo 2

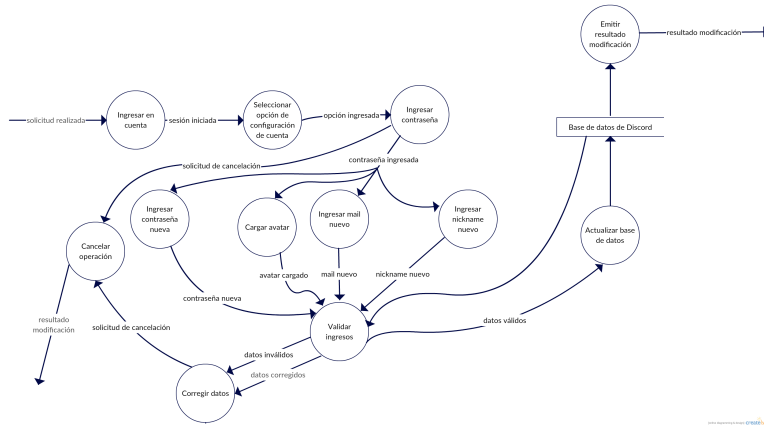


Figura 14: DFD de nivel 2 (1.2) del módulo 2

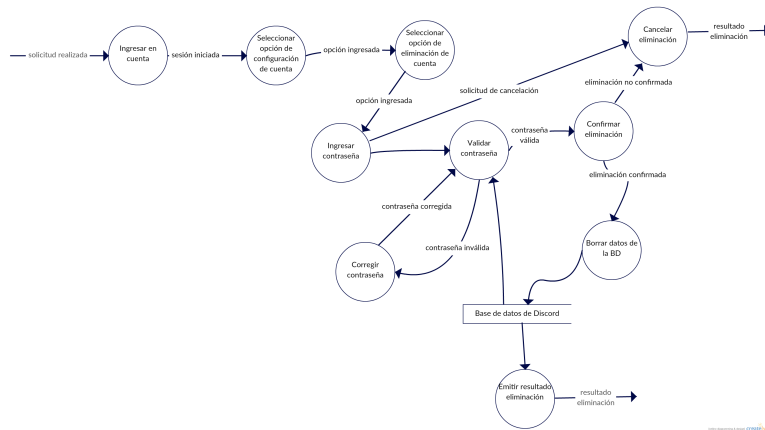


Figura 15: DFD de nivel 2 (1.3) del módulo 2

Finalmente en las figuras anteriores se desglosa cada uno de los procesos antes mencionados en el nivel anterior, donde se puede analizar cada una de las interacciones que permiten el desarrollo de los procesos.

### 6.1.3. Módulo 3: Administrar usuarios de un servidor



Figura 16: DFD de contexto del módulo 3

Este diagrama nos permite visualizar de forma general la solicitud de un usuario para administrar los usuarios asociados al servidor donde se encuentra el usuario.

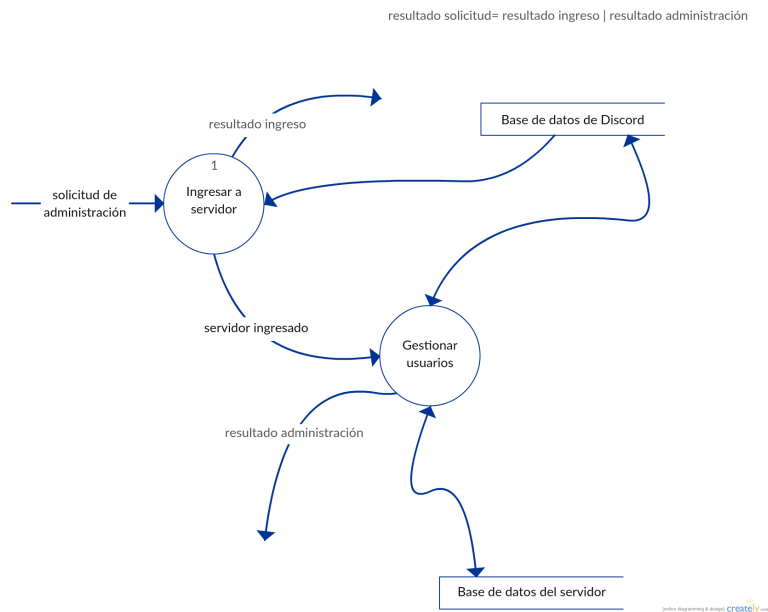


Figura 17: DFD de nivel 1 del módulo 3

En la figura anterior se pueden apreciar los pasos necesarios para poder realizar el proceso principal.

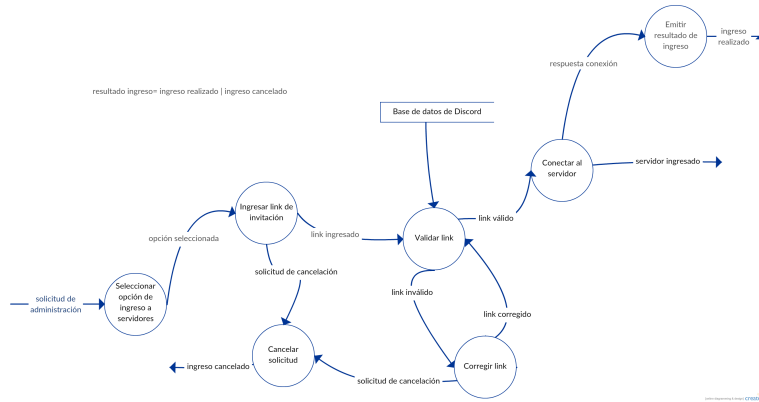


Figura 18: DFD de nivel 2 (1.1) del módulo 3

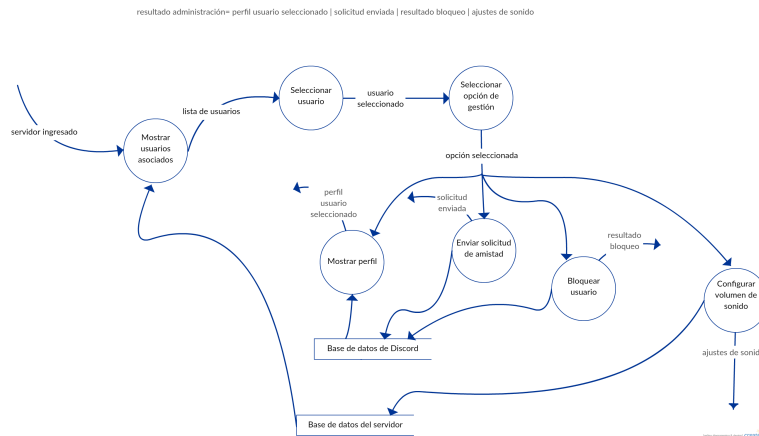


Figura 19: DFD de nivel 2 (1.2) del módulo 3

Por último, obtenemos un modelo del completo funcionamiento de la gestión de usuarios, donde explícitamente se detalla la relación entre los usuarios, para luego realizar una de las acciones que se requiera.

## 6.2. Diccionario de Datos (DFD)

### 6.2.1. Módulo 1: Enviar archivos

Nombre	Composición	Comentarios
Solicitud de envió	Booleano	Pide permiso para enviar el archivo.
Envió cancelado	CaracterPermitdo[30]	Mensaje que recibe el emisor por alguna falla en su envió.
Archivo enviado	CaracterPermitdo[30]	Archivo recibido por el receptor.
Archivo adjunto	CaracterPermitdo[30]	Archivo que se quiere enviar, además de un mensaje de ser necesario. Ej: Hola.png
Lista de archivos	CaracterPermitdo[30]	Lista que enseña todos los archivos en tu dirección de ruta.
Archivo seleccionado	Booleano	Elige el archivo a enviar.
Archivo cancelado	Booleano	Cancela el archivo seleccionado.
Archivo aceptado	Booleano	Acepta el archivo seleccionado.
Archivo rechazado	Booleano	Rechaza el archivo por tamaño inválido. Ej: ¡"hola.png" excede los 8mb!
Archivo por enviar	CaracterPermitdo[30]	Archivo a enviar.
CaracterPermitdo[30]	{Carácter}	Genera palabras.
Carácter	[ A-Z   a-z   0-9   '   -   ]	Cualquier tipo de letra.
Numérico	{0-9}	Numero.

Tabla 4: Diccionario de Enviar archivos

## 6.2.2. Módulo 2: Administrar usuarios

Nombre	Composición	Comentario
Solicitud de administración	Booleano	Solicita acceder a un servidor.
Resultado solicitud	Booleano	Estado de la solicitud.
Resultado ingreso	Booleano	Estado de ingreso.
Servidor ingresado	Booleano	Accede a un servidor.
Resultado administración	Booleano	Estado de administración.
Opción seleccionada	Booleano	Opción seleccionada por el administrador.
Link ingresado	CaracterPermitdo[40]	Link generado. Ej: <a href="https://discord.gg/UK89">https://discord.gg/UK89</a>
Solicitud de cancelación	Booleano	Cancela la solicitud por el usuario.
Ingreso cancelado	Booleano	Estado del ingreso.
Link inválido	Texto	Advierte que el link no es válido.
Link corregido	CaracterPermitdo[40]	Corrige el link que contiene caracteres inválidos.
Link válido	CaracterPermitdo[40]	Link validado.
Respuesta conexión	Texto	Estado de la conexión.
Lista de usuarios	CaracterPermitdo[40]	Lista que enseña todos los usuarios a seleccionar.
Usuario seleccionado	Booleano	Usuario seleccionado por el administrador.
Perfil usuario seleccionado	Resultado administración	Datos del usuario seleccionado.
Solicitud enviada	Resultado administración	Solicitud enviada al usuario.
Resultado bloqueo	Resultado administración	Bloqueo al usuario seleccionado. Ej: Basuraman, bloqueo del servidor.
Ajuste de sonido	Resultado administración	Ajustes de volumen.
CaracterPermitdo[40]	{Carácter}	Genera palabras.
Carácter	[ A-Z   a-z   0-9   '   -   ]	Cualquier tipo de letra.
Numérico	{0-9}	Numero.
Texto	CaracterPermitdo [2000]	Mensaje.

Tabla 5: Diccionario de Administrar usuarios

### 6.2.3. Módulo 3: Gestión de usuario

Nombre	Composición	Comentarios
Solicitud realizada	Booleano	Solicitud que realiza el usuario.
Resultado solicitud	Booleano	Conoce el estado de la solicitud.
Resultado modificación	Booleano	Conoce el estado de la modificación.
Resultado creación	Booleano	Conoce el estado de la creación.
Resultado eliminación	Booleano	Conoce el estado de la eliminación.
Datos de registro	CaracterPermitdo[30]	Datos ingresados por el usuario.
Datos válidos	CaracterPermitdo[30]	Datos válidos.
Datos corregidos	CaracterPermitdo[30]	Corrige los datos que contiene caracteres inválidos.
Datos inválidos	Texto	Advierte que los datos ingresados no son válidos.
Datos en uso	Texto	Indica que los datos ya fueron ocupados.
Datos correctos	Texto	Indica que los datos están correctos.
Solicitud de cancelación	Booleano	Cancela la solicitud por el usuario.
Sesión iniciada	Booleano	Ingreso a la cuenta del usuario.
Opción ingresada	Booleano	Elección de opción.
Contraseña ingresada	CaracterPermitdo[30]	Contraseña del usuario. Ej:hola123
Contraseña nueva	CaracterPermitdo[30]	Nueva contraseña elegida por el usuario.

Tabla 6: Diccionario Gestión de usuario Parte 1

Avatar cargado	CaracterPermitdo[30]	Foto elegida por el usuario.
Mail nuevo	CaracterPermitdo[30]	Nuevo correo para la cuenta. Ej:nuevo.correo@gmail.com
Nickname nuevo	CaracterPermitdo[30]	Nuevo nombre de usuario.
Contraseña válida	CaracterPermitdo[30]	Contraseña correcta.
Contraseña corregida	CaracterPermitdo[30]	Especifica el error de la contraseña.
Contraseña inválida	CaracterPermitdo[30]	Indica que la contraseña se encuentra inválida.
Eliminación no confirmada	Booleano	Cancela la eliminación por el usuario.
Eliminación confirmada	Booleano	Acepta la solicitud del usuario.
CaracterPermitdo[30]	{Carácter}	Genera palabras.
Carácter	[ A-Z   a-z   0-9   '   -   ]	Cualquier tipo de letra.
Numérico	{0-9}	Numero.
Texto	CaracterPermitdo [2000]	Mensaje.

Tabla 7: Diccionario Gestión de usuario Parte 2

### 6.3. Modelo entidad relacion

El siguiente modelo entidad relación consta de ocho entidades que en su conjunto describen el funcionamiento a nivel base de datos de la aplicación. Usuario que tiene por clave primaria un Id puede enviar cero o muchos mensajes, también puede tener cero o muchos servidores, los mensajes que pueden ser enviados por cero o un usuario y recibidos por un canal tienen como clave primaria un Id y representan los datos que envía un usuario, la entidad canal que tiene Id por clave primaria tiene como hijos a canalServidor y canalInterno, canalServidor representa a un canal alojado en un servidor específico mientras que canalInterno es cuando un usuario habla directamente con otro sin pasar por un servidor que haga de mediador, a su vez canalServidor tiene dos hijos que son canal texto y canal voz que se diferencian principalmente por el uso que se les da, por último la entidad servidor que tiene cero o muchos usuarios además de cero o muchos canalServidor posee como clave primaria un Id.

Todos los atributos de las entidades antes mencionadas son explicados detalladamente en sus diccionarios correspondientes.

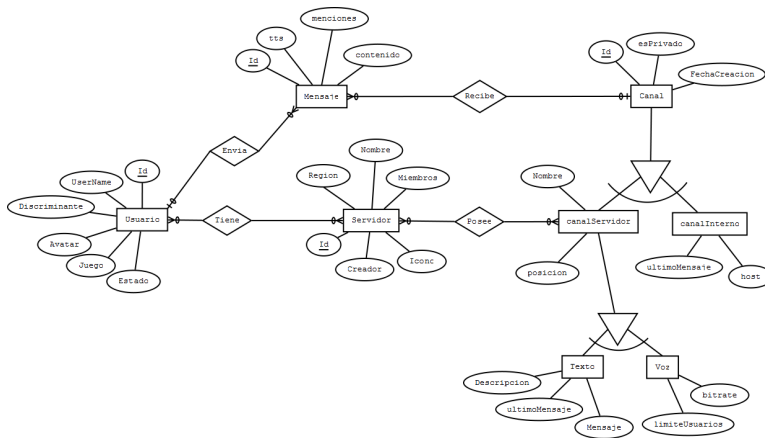


Figura 20: Modelo Entidad Relación

### 6.3.1. Diccionario de datos ER

Entidad:	Usuario	
Alias	No posee	
Persona que utiliza la aplicación		
Atributo	Descripción del atributo	Tipo
Id	String, nunca cambia, se utiliza para encontrar al usuario	{Digito}+{Letra}
UserName	String, nombre o alias que elige el usuario para ser visto dentro de la aplicación	{Letra}
Discriminante	Int, valor entero que varia entre 0-9999 junto al UserName se utiliza para diferenciar nombres iguales	{Digito}
Avatar	Imagen que selecciona el usuario para ser visto dentro de la aplicación	{Imagen}
Juego	String, contiene el nombre del videojuego que esta jugando actualmente el usuario	{Letra}
Estado	String, puede ser online u offline	{Letra}

Tabla 8: Diccionario de la Entidad Usuario

Entidad:	Servidor	
Alias	No posee	
Servidor que aloja canales y permita que usuarios interactúen entre si, guarda toda la información de la aplicación		
Atributo	Descripción del atributo	Tipo
Id	String, nunca cambia, se utiliza para encontrar al servidor	{Digito}+{Letra}
Creador	Usuario, es el fundador de un servidor específico	{Letra}
Icono	Imagen, imagen que selecciona el creador para representar el servidor	{Imagen}
Miembros	Lista de usuarios que han pasado por el servidor durante su existencia	{Letra}
Nombre	String, contiene el nombre del servidor es elegido por el creador	{Letra}
Región	String, locación física donde esta el servidor	{Letra}

Tabla 9: Diccionario de la Entidad Servidor

Entidad:	Mensaje	
Alias	No posee	
Entidad que representa los datos que envía un usuario aun canal en específico		
Atributo	Descripción del atributo	Tipo
Id	String, nunca cambia, se utiliza para encontrar al mensaje	{Digito}+{Letra}
tts	Booleano, true si es un texto para reproducir por voz	[true   false]
Menciones	Lista de usuarios que han sido mencionados en el mensaje	{Digito}+{Letra}
Contenido	String, contenido del mensaje	{Digito}+{Letra}

Tabla 10: Diccionario de la Entidad Mensaje

Entidad:	Canal	
Alias	No posee	
lugar donde se encuentran usuarios e intercambian datos		
Atributo	Descripción del atributo	Tipo
Id	String, nunca cambia, se utiliza para encontrar al canal	{Letra}
esPrivado	Booleano, true si es un canal privado entre dos usuarios o false si es uno publico dentro del servidor	[true   false]
fechaCreacion	Fecha, fecha en que se creo el canal	Dia+Mes+Año+{Letra}

Tabla 11: Diccionario de la Entidad Canal

Entidad:	canalServidor (extiende canal)	
Alias	No posee	
Canal perteneciente a un servidor		
Atributo	Descripción del atributo	Tipo
nombre	String, nombre que elije el creador de un canal	{Letra}
Posición	Int, posición del canal dentro del servidor con respecto a otros canales	{Digito}

Tabla 12: Diccionario de la Entidad Canal Servidor

Entidad:	canalInterno (extiende canal)	
Alias	Llamada	
Canal que no necesita de un servidor para existir, es una conexión directa entre dos usuarios		
Atributo	Descripción del atributo	Tipo
ultimoMensaje	Mensaje, copia del ultimo mensaje enviado	{Letra}
Host	Usuario, creador que genera el vinculo con otro usuario	{Letra}

Tabla 13: Diccionario de la Entidad Canal Interno

Entidad:	Canal de voz (extiende a canalServidor)	
Alias	No posee	
Canal específicamente de voz, usuarios hablan en directo mediante un enlace al servidor		
Atributo	Descripción del atributo	Tipo
bitrate	medida de calidad del sonido en kb/s	{Digito}
limiteUsuarios	Int, valor que indica el máximo de usuarios concurrentes en un canal, 0 indica que no hay limite	{Digito}

Tabla 14: Diccionario de la Entidad Canal Voz

Entidad:	Canal de texto (extiende a canalServidor)	
Alias	No posee	
Canal específicamente de texto, usuarios se envían mensajes escritos a través de este tipo de canal		
Atributo	Descripción del atributo	Tipo
Descripción	breve explicación del uso que se le dará al canal	{Letra}
ultimoMensaje	Mensaje, copia del ultimo mensaje enviado	{Letra}

Tabla 15: Diccionario de la Entidad Canal Texto

## 7. Enfoque orientado a objetos

### 7.1. Diagramas de casos de uso

Los diagramas de casos de usos representan bajo acciones y reacciones el comportamiento de Discord desde el punto de vista del usuario, las cuales están directamente relacionadas con los requerimientos funcionales explicados anteriormente en el informe. En el siguiente diagrama de casos de usos se puede apreciar que el usuario posee la facultad de enviar archivos, gestionar configuración y administrar usuarios en un servidor.

## 8. Conclusión

Gracias al estudio del sistema se pudo obtener una visión mucho más amplia de lo que significa desarrollar software y del cuidado que se debe tener para crear modelos que sean claros y no caigan en la ambigüedad.

En el comienzo del trabajo, modelar las funcionalidades de este servicio de comunicación parecía fácil, debido a que se pensó que se podrían realizar diagramas vagamente gracias a toda la documentación que Discord proporciona en su página oficial. Sin embargo, elaborar diagramas para un sistema que ya se encuentra hecho resultó más difícil de lo que se creía, porque los diagramas se debían adecuar a lo que ya está hecho. Por otro lado, esto es bueno porque enseña al ingeniero en formación a ponerse en distintas situaciones y razonar bajo otras condiciones.