

**UNIVERSIDAD DE TARAPACÁ**



**UNIVERSIDAD DE TARAPACÁ**  
*Universidad del Estado*

**FACULTAD DE INGENIERÍA**

Departamento de Ingeniería en Computación e Informática



**Avance 1**

**“Sistema de Gestión y Control de  
Presupuesto para la Facultad de  
Odontología de la Universidad de  
Chile”**

**Autor(es): Diego Baltazar Fernández**

**Dylan Choque Nina**

**Asignatura : Proyecto IV**

**Profesor: Diego Aracena Pizarro**

**Cliente: Roberto Gamboa Aguilar**

Arica, 29 de abril de 2023

## ÍNDICE DE CONTENIDO.

<b>ÍNDICE DE CONTENIDO.....</b>	<b>2</b>
<b>1. INTRODUCCIÓN.....</b>	<b>5</b>
<b>2. DEFINICIÓN DEL PROBLEMA.....</b>	<b>6</b>
2.1. Contexto.....	6
2.2. Problema.....	6
2.3. Solución.....	6
<b>3. OBJETIVOS.....</b>	<b>7</b>
3.1. Objetivo General.....	7
3.2. Objetivos Específicos.....	7
3.3. Alcance.....	7
<b>4. PLANIFICACIÓN.....</b>	<b>8</b>
4.1. Metodología.....	8
4.1.1 Marco de trabajo.....	8
4.1.2 ¿Por qué Scrum?.....	8
4.2. Mecanismos de Organización.....	9
4.2.1 Estructura del Equipo.....	9
4.2.2 Herramientas y Software.....	9
4.3. Carta Gantt.....	10
<b>5. RESUMEN DEL PROYECTO.....</b>	<b>11</b>
5.1. Documento de Requisitos de Alto Nivel.....	11
5.1.1. Características Principales.....	11
5.1.1.1. Requerimientos Funcionales.....	11
5.1.1.2. Requerimientos No Funcionales.....	11
5.1.1.3. Restricciones.....	12
5.1.1.4. Entregables.....	12
5.1.2. Alcance del Ambiente del Software.....	12
5.2. Diseño del Proyecto.....	13
5.2.1. Arquitectura del Sistema.....	13
5.2.2. Modelo de Contexto.....	13
5.2.3. Identificación de Subsistemas.....	14
5.2.4. Modelo de Casos de Uso.....	15
5.2.5. Desarrollo de un BPM.....	20
5.2.6. Herramientas para el Desarrollo del Software.....	22
5.2.7. Aspectos Iniciales del Front-End.....	23
<b>6. CONCLUSIÓN.....</b>	<b>28</b>
<b>7. REFERENCIAS.....</b>	<b>29</b>

## ÍNDICE DE FIGURAS.

Figura 1. Logo de la Facultad de Odontología de la Universidad de Chile.....	5
Figura 2. Carta Gantt del Proyecto.....	10
Figura 3. Firma Digital de Don Roberto Gamboa Aguilar.....	13
Figura 4. Arquitectura de Sistema Inicial.....	14
Figura 5. Diagrama de Modelo de Contexto.....	14
Figura 6. Diagrama de Modelo de Subsistema.....	15
Figura 7. Casos de Uso para “Servidor de Presupuesto”.....	16
Figura 8. Casos de Uso para “Administrar Usuario”.....	16
Figura 9. Casos de Uso para “Administrar Centro de Costos”.....	17
Figura 10. Casos de Uso para “Administrar Subcentro de Costos”.....	17
Figura 11. Casos de Uso para “Administrar Presupuesto Anual”.....	18
Figura 12. Casos de Uso para “Administrar Presupuesto para Centro de Costos”.....	18
Figura 13. Casos de Uso para “Administrar Presupuesto para Subcentro de Costos”.....	19
Figura 14. Casos de Uso para “Administrar Categorías”.....	19
Figura 15. Casos de Uso para “Administrar Transacción”.....	20
Figura 16. Casos de Uso para “Servidor de Autenticación”.....	20
Figura 17. Diagrama BPMN del Proceso “Gestionar Centros de Costo”.....	21
Figura 18. Diagrama BPMN del Proceso “Asignar Presupuesto”.....	21
Figura 19. Diagrama BPMN del Proceso “Gestionar Gastos”.....	22
Figura 20. Boceto de Pantalla “Inicio de Sesión”.....	24
Figura 21. Boceto de Pantalla Inicial del Sistema Administrativo.....	24
Figura 22. Boceto de Pantalla “Gestión de Centros de Costo”.....	25
Figura 23. Boceto de Pantalla “Gestión de Subcentros de Costo”.....	25
Figura 24. Boceto de Pantalla “Gestión de Usuarios”.....	26
Figura 25. Boceto de Pantalla “Creación de Usuario”.....	26
Figura 26. Boceto de Pantalla “Gestión de Presupuesto General Anual”.....	27
Figura 27. Boceto de Pantalla “Gestión de Presupuesto de Centros de Costo”.....	27
Figura 28. Boceto de Pantalla “Asignación de Presupuesto de Centro de Costo”.....	28
Figura 29. Boceto de Pantalla “Panel de Responsable de Centro de Costo”.....	28

## ÍNDICE DE TABLAS.

Tabla 1. Estructura del Equipo.....	9
Tabla 2. Herramientas y Software.....	9
Tabla 3. Requerimientos Funcionales.....	11
Tabla 4. Requerimientos no Funcionales.....	11

## 1. INTRODUCCIÓN

La gestión eficiente de los recursos financieros es un aspecto fundamental para el funcionamiento óptimo de cualquier institución educativa. En el contexto de la Facultad de Odontología de la Universidad de Chile [1], se enfrentan desafíos significativos en la asignación y control de presupuestos destinados a diferentes áreas académicas y clínicas. La complejidad aumenta al tener que distribuir un presupuesto general entre múltiples centros de costos y mantener un registro preciso de los gastos realizados.

El presente proyecto tiene como objetivo desarrollar un sistema de gestión y control de presupuesto web que permita gestionar de manera eficiente y transparente los recursos financieros de la Facultad de Odontología. Esta solución integrará tecnologías modernas como la autenticación con Google, un sistema de mensajería para comunicaciones internas, y funcionalidades que faciliten la asignación y seguimiento de presupuestos en tiempo real.

A través de este proyecto, se busca mejorar la toma de decisiones basadas en información actualizada y precisa, optimizar los procesos de asignación y reducción de presupuestos, y garantizar la transparencia y eficiencia en el uso de los recursos financieros disponibles.



***Figura 1. Logo de la Facultad de Odontología de la Universidad de Chile.***

## **2. DEFINICIÓN DEL PROBLEMA**

### **2.1. Contexto**

El proyecto a desarrollar tiene como objetivo abordar una problemática de control de presupuesto dentro de la Facultad de Odontología de la Universidad de Chile. La Clínica Odontológica de la Universidad de Chile es una institución de salud que cuenta con 148 unidades dentales y ofrece atención de problemas odontológicos y tratamiento de enfermedades bucales.

### **2.2. Problema**

En la actualidad, la gestión de presupuestos en la Facultad de Odontología se lleva a cabo de forma manual y descentralizada, lo que complica la asignación eficiente de recursos y el seguimiento de gastos. La ausencia de un sistema centralizado y automatizado ocasiona retrasos en los procesos y dificulta la toma de decisiones fundamentadas en información actualizada y precisa.

### **2.3. Solución**

Para abordar esta problemática, se propone el desarrollo e implementación de un sistema de gestión y control de presupuesto a través de una plataforma web. Esta solución permitirá optimizar la asignación y distribución de recursos financieros en la Facultad de Odontología. Entre las características principales de esta solución se encuentra la creación de perfiles de usuario con diferentes niveles de acceso, funcionalidades para gestionar el presupuesto de los distintos centros de costos de la Clínica Odontológica de la Universidad de Chile y para visualizar detalladamente los gastos producidos por cada centro de costo.

### **3. OBJETIVOS**

#### **3.1. Objetivo General**

Desarrollar un sistema de gestión y control de presupuesto para la Facultad de Odontología de la Universidad de Chile que permita asignar, distribuir y controlar los recursos financieros de manera eficiente y transparente.

#### **3.2. Objetivos Específicos**

- Analizar y comprender la problemática existente en el proceso de control de presupuesto actual de la Facultad de Odontología.
- Establecer los requerimientos funcionales y no funcionales del sistema que dará solución a la problemática.
- Diseñar e implementar una interfaz web intuitiva y funcional para la asignación y distribución de presupuestos.
- Integrar funcionalidades de reducción de presupuestos por medio de compras o contrataciones en los centros de costos.
- Implementar mecanismos de seguimiento y control para garantizar la transparencia y eficiencia en el uso de los recursos financieros.
- Realizar las pruebas de funcionamiento y análisis de resultados
- Desplegar el sistema en servidores en la nube para garantizar la disponibilidad y escalabilidad del sistema.

#### **3.3. Alcance**

El sistema abarca la asignación y distribución de presupuestos anuales para áreas académicas y clínicas específicas dentro de la Facultad de Odontología. Se incluirá la funcionalidad para reducir presupuestos en centros de costos mediante compras o contrataciones. Además, se desarrollará un módulo de control de acceso y autenticación para garantizar la seguridad de la plataforma.

## **4. PLANIFICACIÓN**

### **4.1. Metodología**

#### **4.1.1 Marco de trabajo**

El marco de trabajo a utilizar en este proyecto es Scrum. Scrum es un marco de trabajo basado en metodologías ágiles para la gestión de proyectos, enfocados en un conjunto de valores, principios y prácticas. Scrum propone un enfoque iterativo e incremental que se ajusta perfectamente a proyectos que requieren adaptabilidad y respuesta rápida a los cambios [1].

#### **4.1.2 ¿Por qué Scrum?**

Se escogió Scrum ya que este marco de trabajo impulsa la adaptabilidad, la flexibilidad, la comunicación y la retroalimentación. Estos pilares son de suma importancia al momento de entablar acuerdos con el cliente alcanzando un mayor conocimiento y apoyo mutuo en los requisitos que se deben cumplir en el proyecto, gracias a la gran comunicación y retroalimentación que este marco de trabajo propone.



## 4.2. Mecanismos de Organización

### 4.2.1 Estructura del Equipo

**Tabla 1. Estructura del Equipo.**

Rol	Responsabilidad	Responsable
Jefe del Proyecto	<ul style="list-style-type: none"> <li>• Coordinación general del proyecto, asegurando el cumplimiento de plazos y objetivos.</li> <li>• Gestión de recursos y asignación de tareas.</li> <li>• Monitoreo del progreso del proyecto y ajustes de planificación según sea necesario.</li> </ul>	Diego Baltazar
Desarrollador	<ul style="list-style-type: none"> <li>• Participar en la definición de los requisitos del software junto con el equipo y los usuarios.</li> <li>• Diseñar la arquitectura del sistema y el diseño de interfaces.</li> <li>• Desarrollar el software siguiendo las mejores prácticas de desarrollo.</li> </ul>	Dylan Choque

### 4.2.2 Herramientas y Software

**Tabla 2. Herramientas y Software.**

Nombre	Logo	Funcion
Whatsapp		Comunicación rápida y efectiva entre miembros del equipo.
Notion		Organización de la documentación del proyecto, notas, y tareas pendientes.
Drive		Almacenamiento de documentos y archivos importantes del proyecto en la nube.
Draw.io		Herramienta de diagramación para la creación de diagramas arquitectónicos, flujos de trabajo y otros diagramas relacionados con el proyecto.
Visual Studio Code		Editor de código fuente, utilizado para la programación y desarrollo del proyecto.
Github		Control de versiones y almacenamiento del código fuente del proyecto.
Redmine		Gestión de proyectos y seguimiento de errores; subida y evaluación de documentos.



### 4.3. Carta Gantt

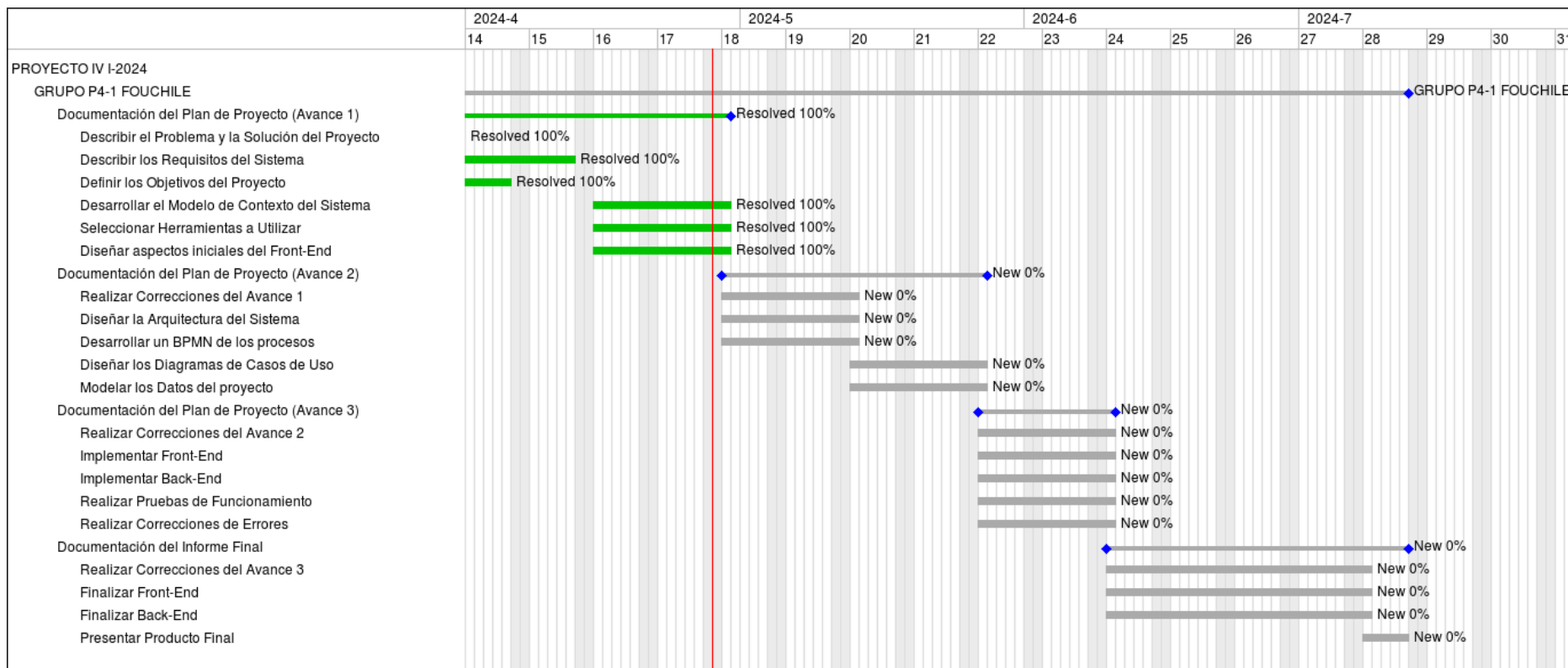


Figura 2. Carta Gantt del Proyecto.

## 5. RESUMEN DEL PROYECTO

### 5.1. Documento de Requisitos de Alto Nivel

#### 5.1.1. Características Principales

El sistema de control de presupuesto de la Facultad de Odontología incluirá las siguientes características principales:

##### 5.1.1.1. Requerimientos Funcionales.

**Tabla 3. Requerimientos Funcionales.**

NºRequerimiento	Descripción
RF_1	Los usuarios deben poder iniciar sesión según su rol en el sistema utilizando credenciales válidas.
RF_2	Permitir a los administradores asignar y modificar los presupuestos de cada uno de los centros y subcentros de costos de la facultad.
RF_3	Asignar y modificar usuarios responsables para cada centro de costos de la facultad.
RF_4	Visualizar la ramificación de los gastos realizados por cada centro de costos, incluyendo los ítems específicos en los que se ha gastado.
RF_5	Generar alertas automáticas para los responsables de los centros de costos cuando el saldo disponible se encuentre en un porcentaje bajo.
RF_6	Generar informes financieros por periodos de los gastos producidos en distintos centros de costos de la facultad.

##### 5.1.1.2. Requerimientos No Funcionales.

**Tabla 4. Requerimientos no Funcionales.**

NºRequerimiento	Descripción
RnF_1	El sistema debe cumplir con los estándares de seguridad requeridos para mantener la integridad y seguridad de los datos.
RnF_2	Interfaz interactiva y amigable con el usuario para facilitar la navegación y gestión dentro del sistema.
RnF_3	Los usuarios del sistema deben poder acceder mediante navegador web en computadores y dispositivos móviles.
RnF_4	El sistema debe adoptar una arquitectura de microservicios y ser capaz de comunicarse con sistemas externos.
RnF_5	El código debe estar bien documentado y estructurado para facilitar futuras actualizaciones y mantenimiento.

#### **5.1.1.3. Restricciones.**

- Plataforma web: El sistema estará disponible como una aplicación web accesible desde navegadores modernos.
- Integridad de datos: Implementar mecanismos de control de versiones y copias de seguridad periódicas para mantener la integridad de los datos y la capacidad de recuperación en caso de fallos o pérdida de información.
- Escalabilidad: Diseñar el sistema de manera que pueda escalar fácilmente para manejar un aumento en el número de usuarios, centros de costos y transacciones, sin comprometer el rendimiento ni la seguridad.

#### **5.1.1.4. Entregables.**

- Bitácoras Semanales
- Documentos de Requerimientos Firmado por el Cliente
- Informe de Avance 1
- Informe de Avance 2
- Informe de Avance 3
- Informe Final
- Manuel de Usuario
- Propuesta de Despliegamiento
- Documentación del Producto
- Producto

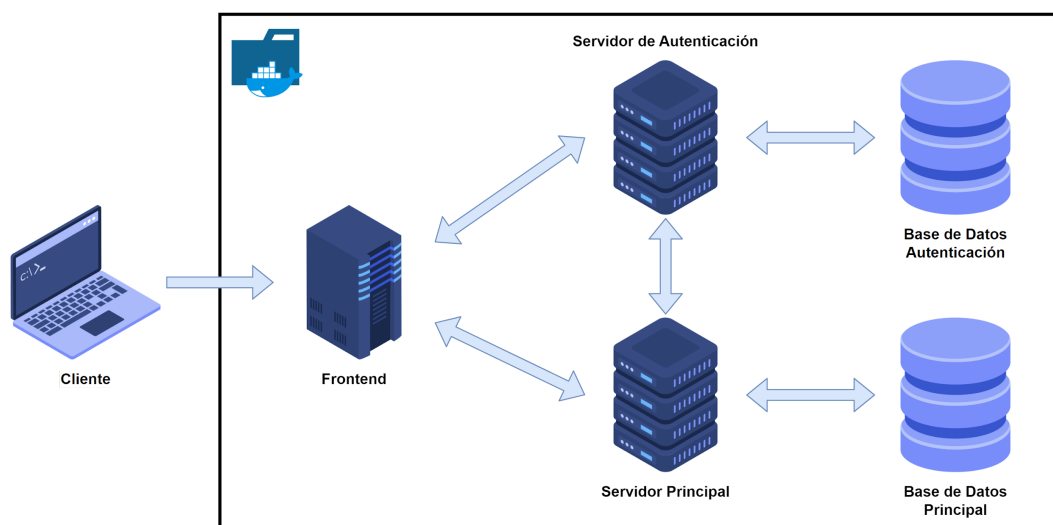
#### **5.1.2. Alcance del Ambiente del Software**

El alcance del ambiente del software incluirá la implementación de todas las funcionalidades mencionadas anteriormente en un entorno web seguro y escalable. Se desarrollarán módulos específicos para la gestión de presupuestos, autenticación y otras características esenciales para el funcionamiento del sistema.

## 5.2. Diseño del Proyecto

### 5.2.1. Arquitectura del Sistema

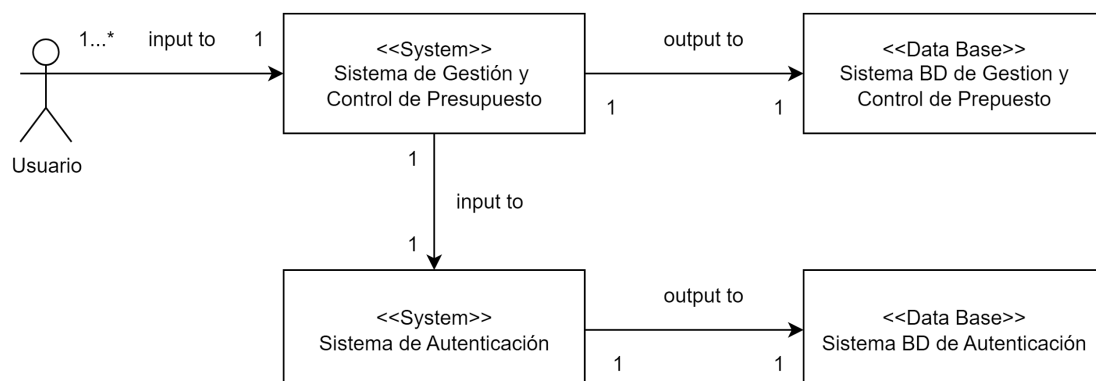
La arquitectura del sistema propuesto se basa en un enfoque de microservicios utilizando Angular para el frontend y NestJS para los servicios backend, organizados en dos microservicios principales: uno para la gestión del presupuesto y otro para la autenticación. La comunicación entre los microservicios se facilita mediante RabbitMQ, asegurando un flujo de datos eficiente y desacoplado. Cada servicio cuenta con su propia base de datos MongoDB. Todo el sistema está contenerizado con Docker, permitiendo una implementación y mantenimiento flexibles y consistentes en diferentes entornos.



**Figura 4. Arquitectura de Sistema Inicial.**

### 5.2.2. Modelo de Contexto

El modelo de contexto ilustrado en la figura 5 muestra cómo el sistema de gestión y control de presupuesto interactúa con los usuarios dentro de la Facultad de Odontología.

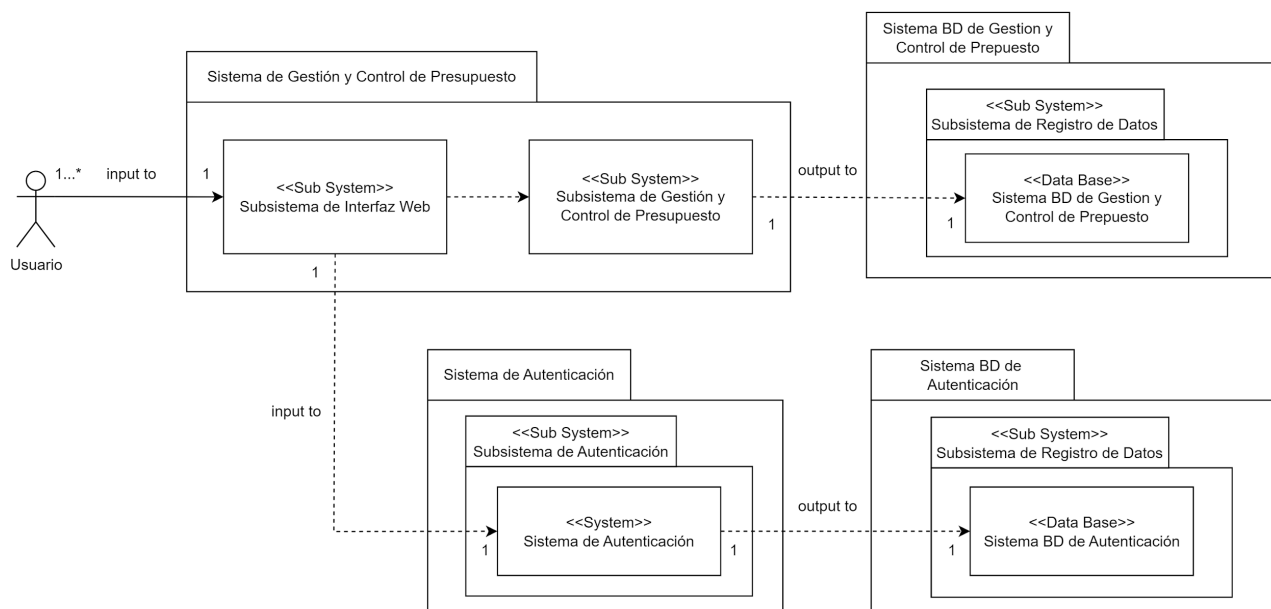


**Figura 5. Diagrama de Modelo de Contexto.**

### 5.2.3. Identificación de Subsistemas

El diagrama de subsistemas ilustrado en la figura 6 muestra la estructura y las interacciones del sistema de gestión y control de presupuestos de la Facultad de Odontología, podemos organizar los subsistemas basándonos en las principales funcionalidades y responsabilidades.

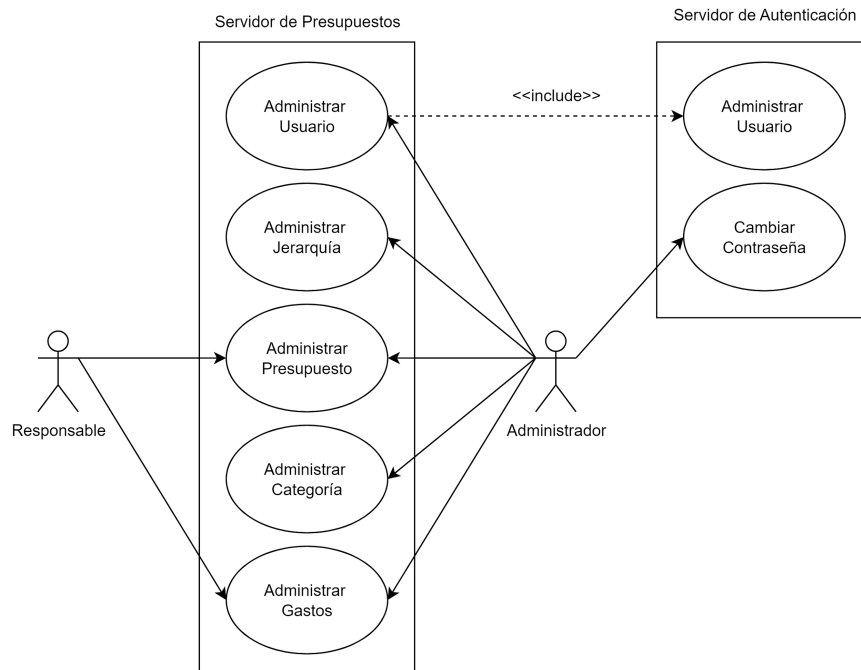
- Subsistema de Interfaz Web: Este módulo es responsable de interactuar con los usuarios finales y comunicarse con los microservicios de gestión de presupuesto y autenticación para obtener y enviar datos.
- Subsistema de Gestión y Control de Presupuesto: Este módulo es responsable del manejo de los presupuestos, encargándose de las operaciones relacionadas con la asignación, modificación y seguimiento de los fondos a través de los centros de costos.
- Subsistema de Base de Datos del Sistema de Gestión y Control de Presupuesto: Este módulo es responsable de almacenar y gestionar todos los datos, incluyendo registros de transacciones y ajustes de fondos.
- Sistema de Autenticación: Este módulo es responsable de la gestión de la autenticación y autorización de los usuarios que acceden al sistema. Este módulo asegura que solo los usuarios autorizados puedan realizar operaciones dentro del sistema de gestión presupuestaria.
- Subsistema de Base de Datos del Sistema de Autenticación: Este módulo es responsable de proporcionar una gestión segura de las credenciales y los roles de acceso.



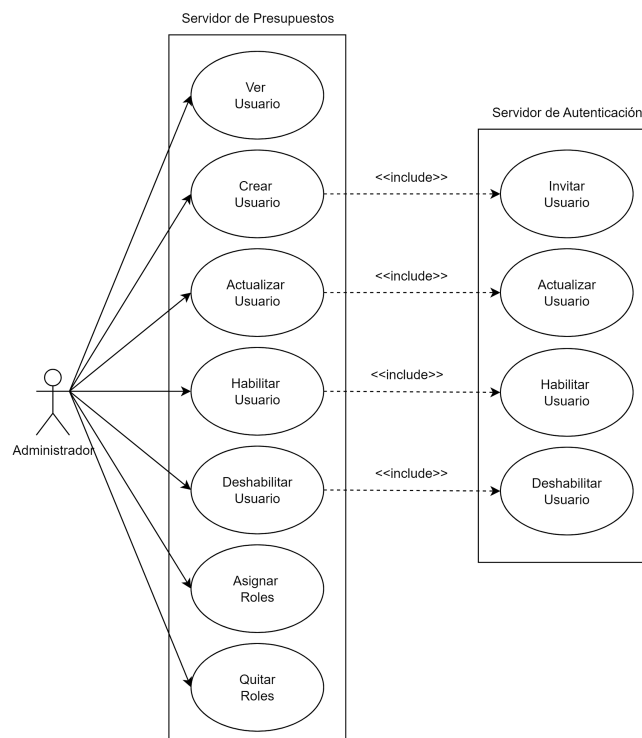
**Figura 6. Diagrama de Modelo de Subsistema.**

### 5.2.4. Modelo de Casos de Uso

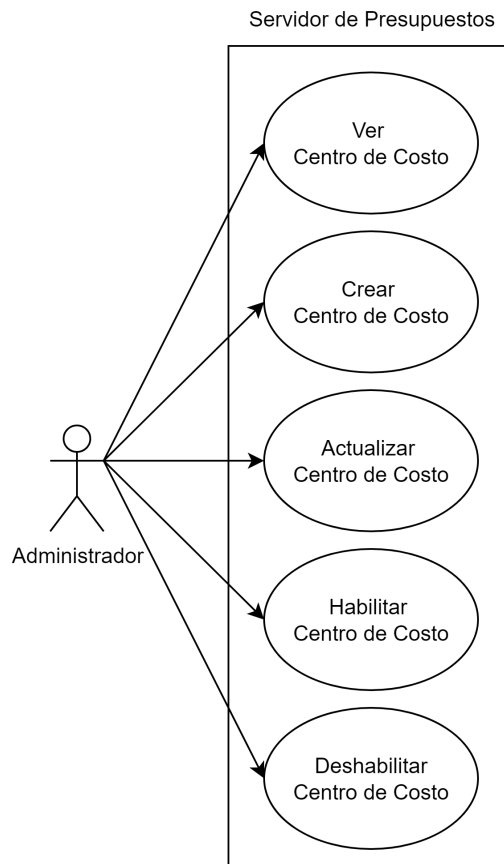
A continuación, se presentan descripciones iniciales para los diagramas de casos de uso específicos de diferentes componentes del sistema de gestión y control de presupuestos:



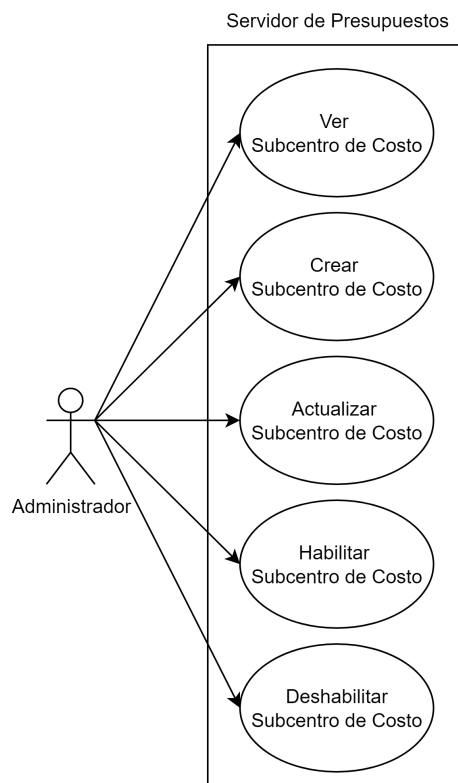
**Figura 7. Casos de Uso para “Servidor de Presupuesto”.**



**Figura 8. Casos de Uso para “Administrar Usuario”.**

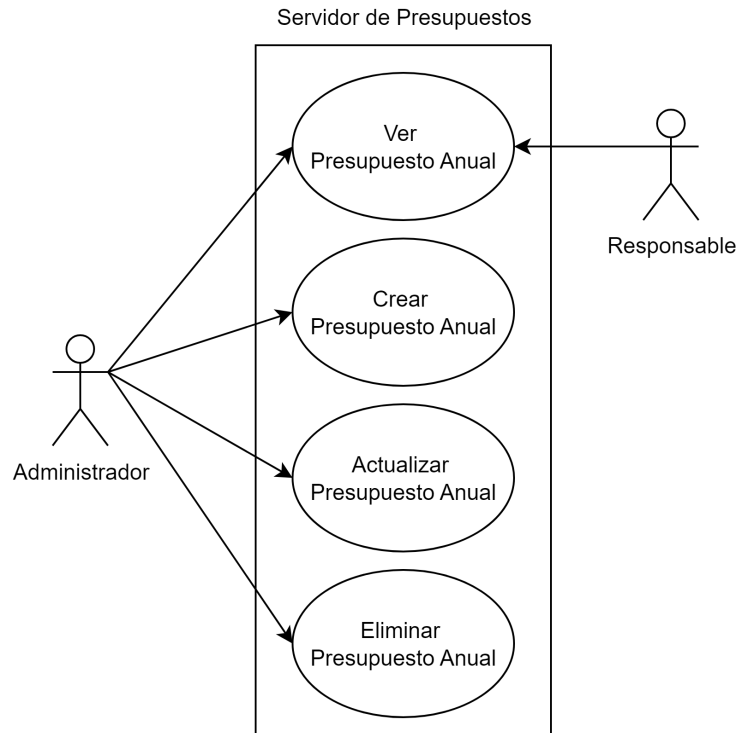


**Figura 9. Casos de Uso para “Administrar Centro de Costos”.**

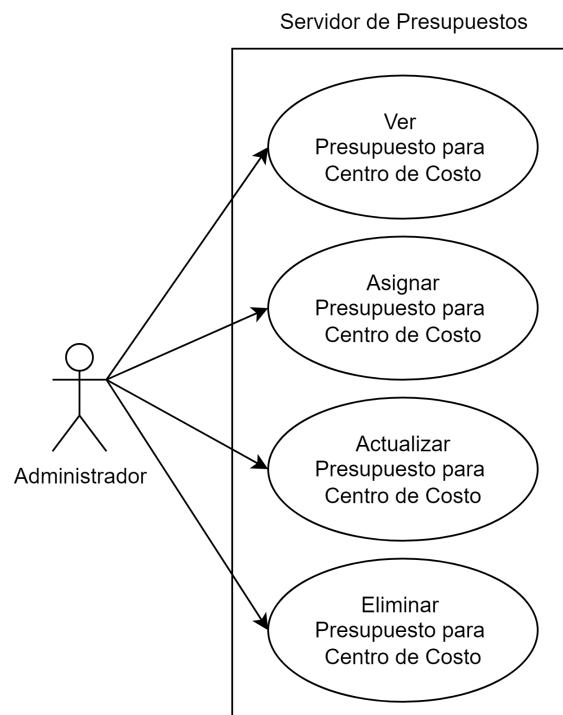


**Figura 10. Casos de Uso para “Administrar Subcentro de Costos”.**

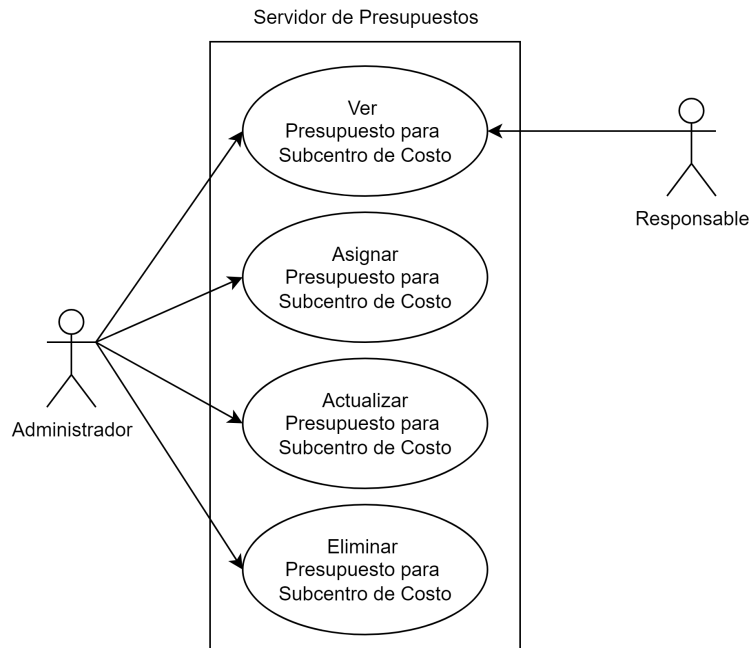




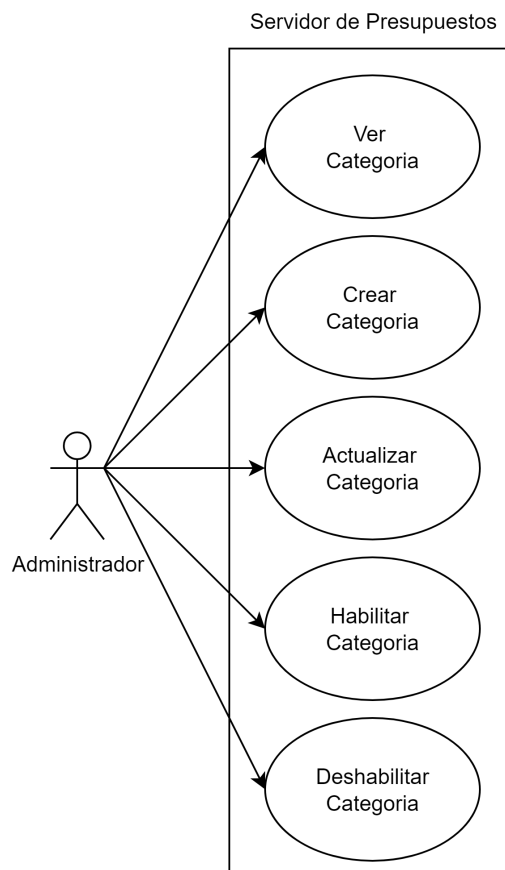
**Figura 11. Casos de Uso para “Administrar Presupuesto Anual”.**



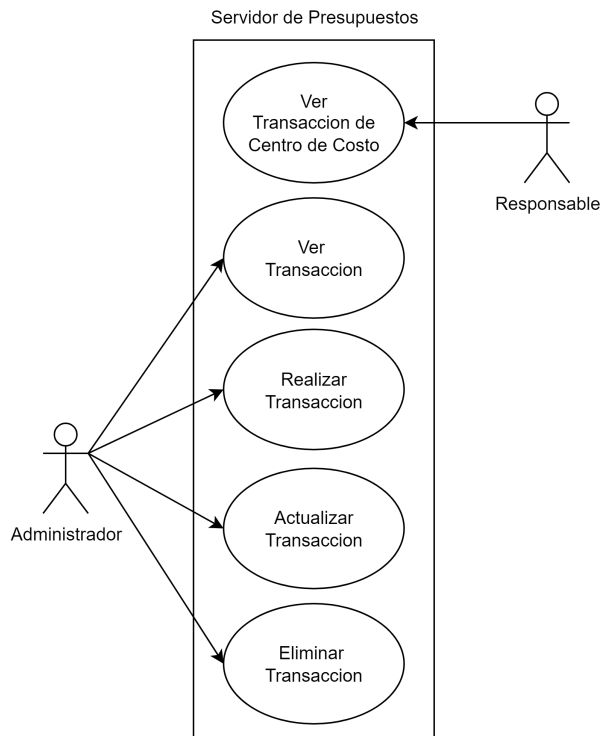
**Figura 12. Casos de Uso para “Administrar Presupuesto para Centro de Costos”.**



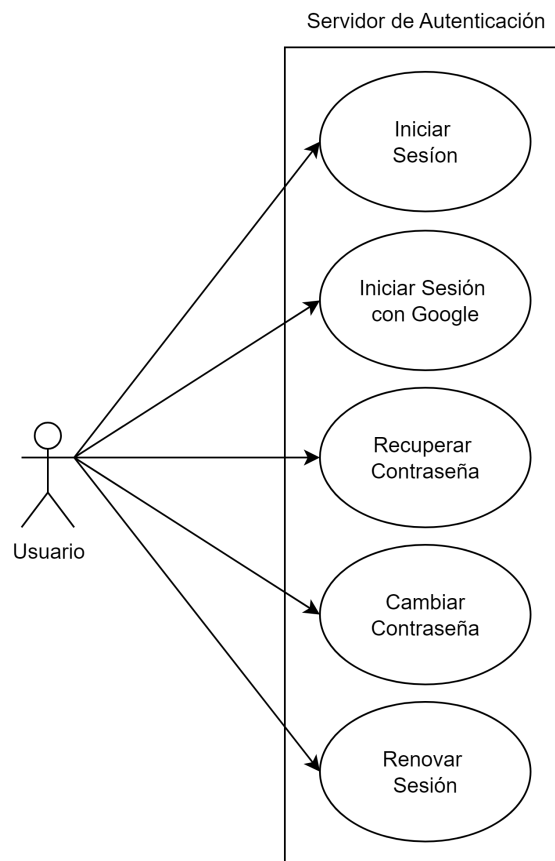
**Figura 13. Casos de Uso para “Administrar Presupuesto para Subcentro de Costos”.**



**Figura 14. Casos de Uso para “Administrar Categorías”.**



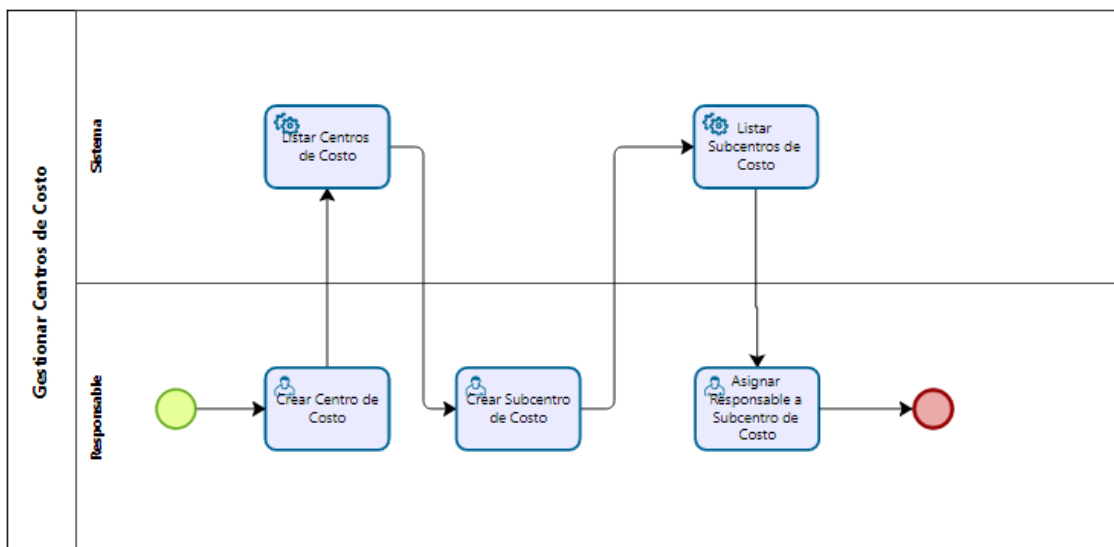
**Figura 15. Casos de Uso para “Administrar Transacción”.**



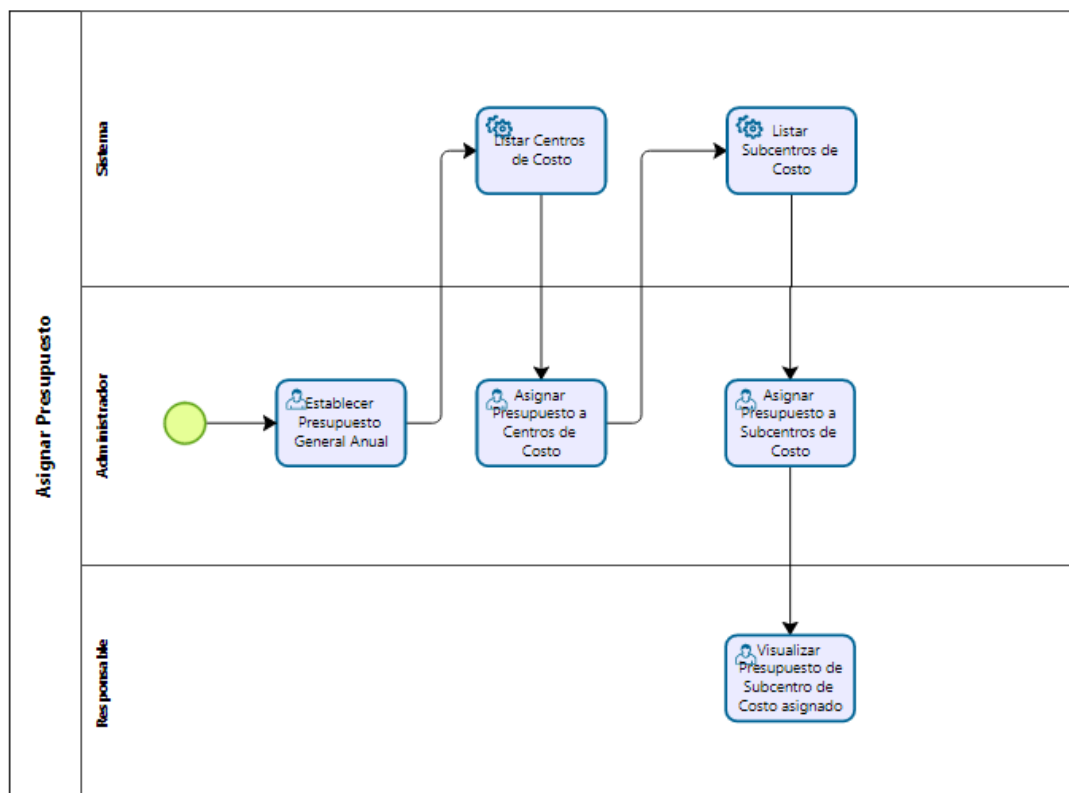
**Figura 16. Casos de Uso para “Servidor de Autenticación”.**

### 5.2.5. Desarrollo de un BPM.

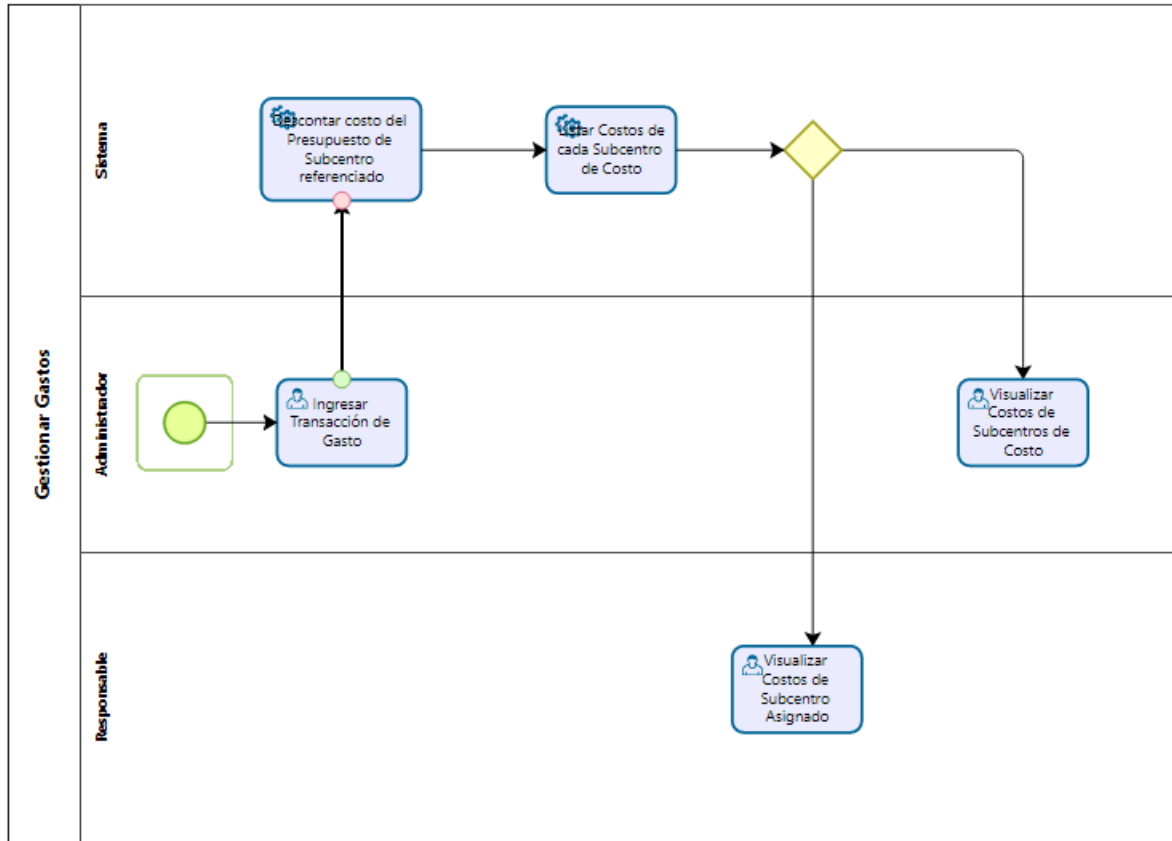
El Modelado de Procesos de Negocio (BPM) es un método de diagrama de flujo que modela los pasos de un proceso de negocio planificado desde un inicio a un fin. Es clave para la gestión de procesos de negocio y representa visualmente una secuencia detallada de actividades comerciales y flujos de información necesarios para completar un proceso. Se utilizó la herramienta Bizagi Modeler [3] para el modelado de los distintos procesos para la Gestión de Presupuesto de la Facultad de Odontología mediante el sistema a implementar.



**Figura 17. Diagrama BPMN del Proceso “Gestionar Centros de Costo”.**



**Figura 18. Diagrama BPMN del Proceso “Asignar Presupuesto”.**



**Figura 19. Diagrama BPMN del Proceso “Gestionar Gastos”.**

### 5.2.6. Herramientas para el Desarrollo del Software

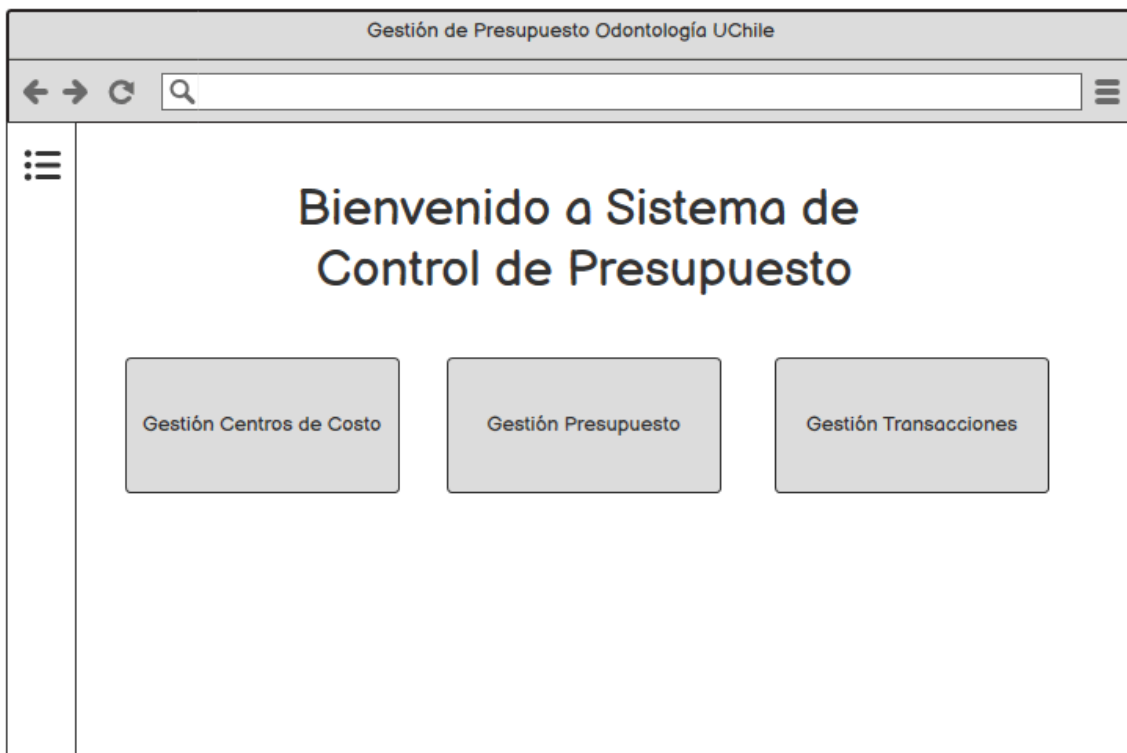
1. Angular:
  - Función: Framework de desarrollo para construir la interfaz de usuario del sistema. Facilita la creación de componentes reutilizables y maneja eficientemente el estado y la lógica de la interfaz.
2. NestJS:
  - Función: Framework de Node.js que se utiliza para construir aplicaciones del lado del servidor. Es ideal para la creación de servicios robustos y escalables, apoyando el modelo de microservicios con su arquitectura modular y facilidad de integración con otros sistemas y bases de datos [\[4\]](#).
3. MongoDB:
  - Función: Base de datos NoSQL utilizada para almacenar y recuperar datos de manera eficiente. Su esquema flexible es ideal para manejar las estructuras de datos complejas requeridas por los microservicios [\[5\]](#).
4. Docker:
  - Función: Plataforma de contenerización que facilita el desarrollo, el testing y la producción de aplicaciones al permitir que los microservicios se empaqueten y se ejecuten en contenedores independientes. Esto asegura consistencia entre los entornos de desarrollo, prueba y producción [\[6\]](#).
5. RabbitMQ:
  - Función: Middleware de mensajería que permite la comunicación entre diferentes microservicios a través de la transmisión de mensajes de manera confiable y eficiente, lo cual es crucial en una arquitectura de microservicios [\[7\]](#).
6. Visual Studio Code:
  - Función: Editor de código que soporta múltiples lenguajes de programación y tiene extensiones para integrar otros servicios y herramientas como Docker y Git, lo que lo hace ideal para el desarrollo de software moderno.
7. Git (con GitHub):
  - Función: Sistema de control de versiones que permite gestionar el código fuente y colaborar en el desarrollo del proyecto. GitHub proporciona repositorios remotos, revisión de código, gestión de contribuciones y más.

### 5.2.7. Aspectos Iniciales del Front-End.

Para el diseño de los bocetos iniciales del Frontend del Sistema, se utilizó la herramienta Balsamiq Wireframes [8]. A continuación se presentan los bocetos de las distintas pantallas que tendrá el sistema.



**Figura 20. Boceto de Pantalla “Inicio de Sesión”.**



**Figura 21. Boceto de Pantalla Inicial del Sistema Administrativo.**

En la pantalla “Gestión de Centros de Costos” el administrador podrá crear y modificar los distintos Centros y Subcentros de Costo de la Facultad de Odontología de la Universidad de Chile, además podrá deshabilitarlos o asignar a un responsable.

CR	CC	SCC	Nombre	Acciones
01	00		DECANATO	Modificar
02	00		VICEDECANATO	Modificar
03	00		ESCUELA DE PREGRADO	Modificar
03	04		DAA	Modificar
03	05		DAE	Modificar
33	01		DIFO	Modificar
04	00		ESCUELA DE GRADUADOS	Modificar
31	59	00	CLINICA DE PROYECTOS	Modificar

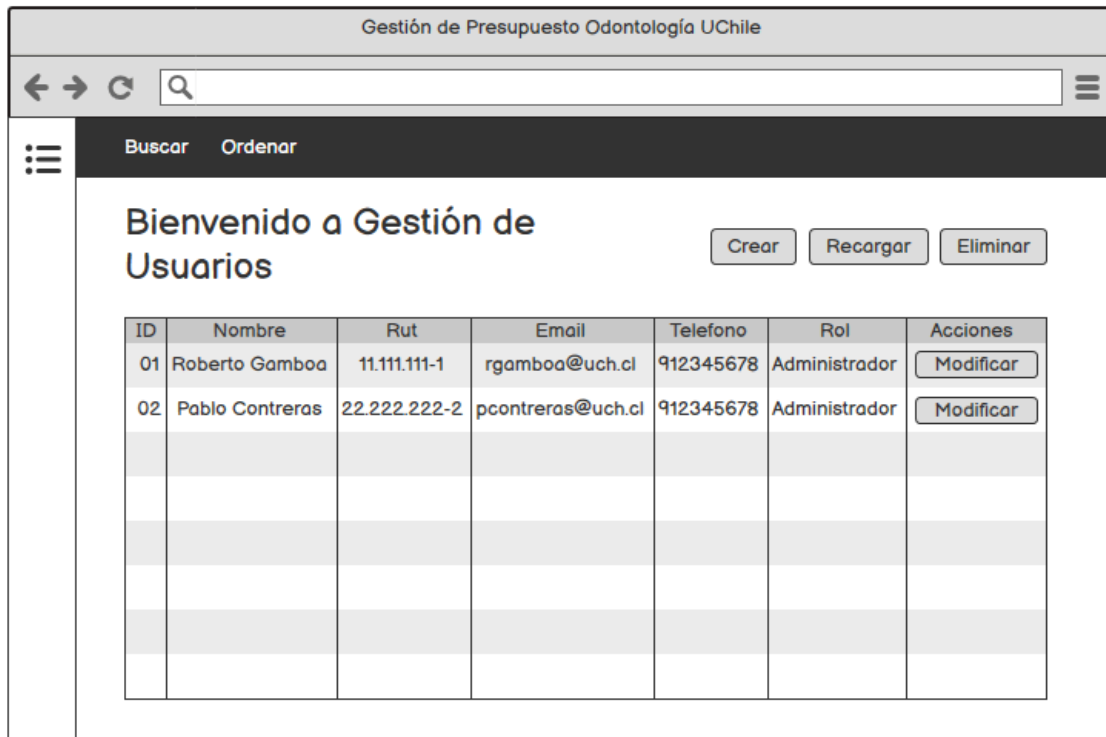
**Figura 22. Boceto de Pantalla “Gestión de Centros de Costo”.**

CR	CC	SCC	Nombre	Acciones
01	01	001	DIRECCION DECANATO	Habilitado Modificar Asignar Res
01	01	005	CURSO CLINICA (INYECCIÓN...	Habilitado Modificar Asignar Res
01	01	006	CURSO CLINICA REGULACIÓN...	Habilitado Modificar Asignar Res
01	01	007	COMITÉ DE ÉTICA CIENTÍFICO	Habilitado Modificar Asignar Res
01	01	008	CONTINGENCIA NACIONAL (PROTESIS...	Habilitado Modificar Asignar Res
01	01	009	PROYECTO COVID VAEGI	Habilitado Modificar Asignar Res
01	01	010	Estacionamiento de Quimica	Habilitado Modificar Asignar Res

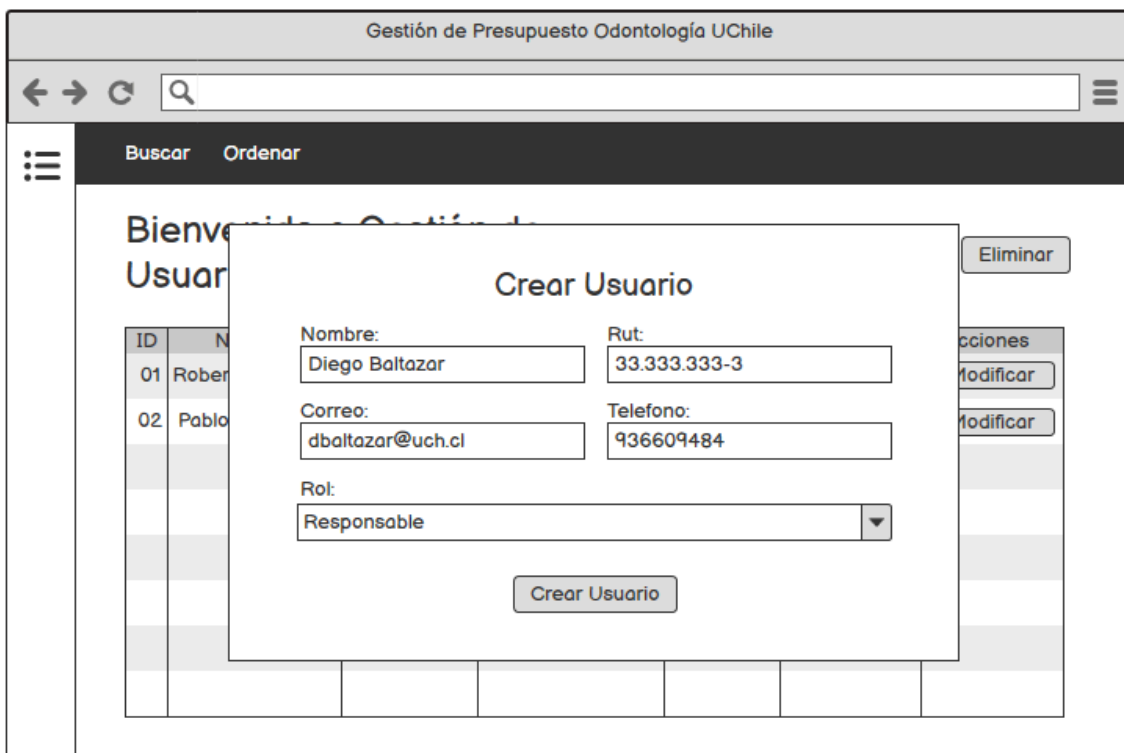
**Figura 23. Boceto de Pantalla “Gestión de Subcentros de Costo”.**



En la pantalla “Gestión de Usuarios”, el Administrador podrá crear a los usuarios que podrán acceder al sistema, los usuarios pueden ser “Administrador” o “Responsable”.

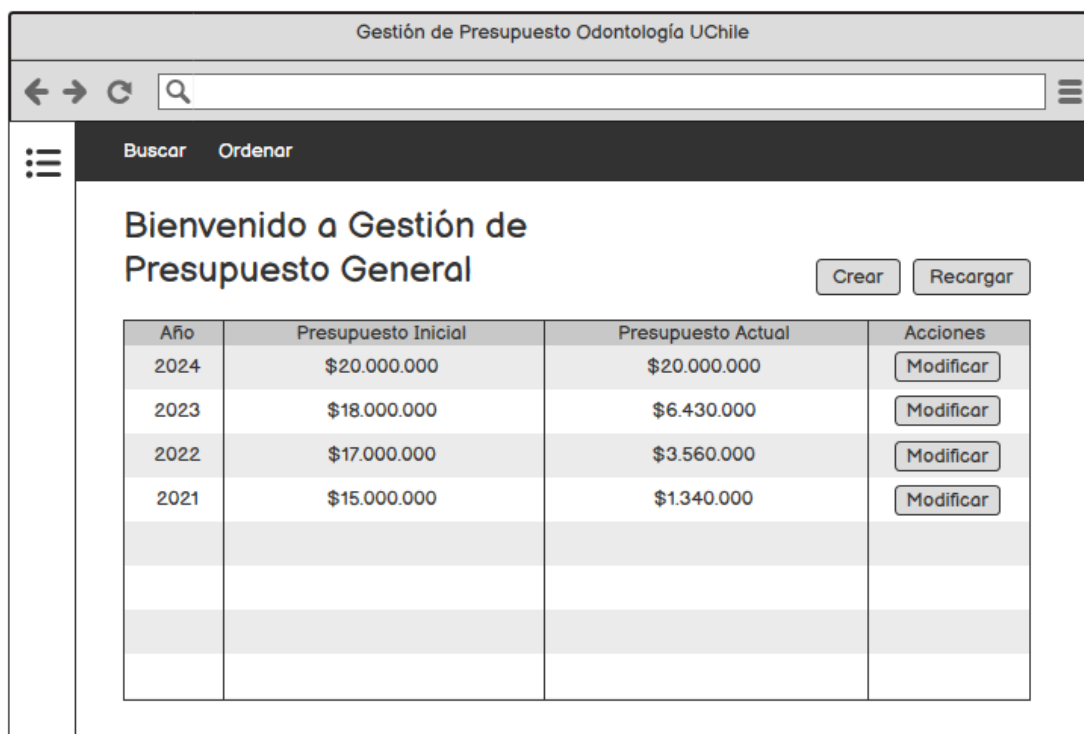


**Figura 24. Boceto de Pantalla “Gestión de Usuarios”.**

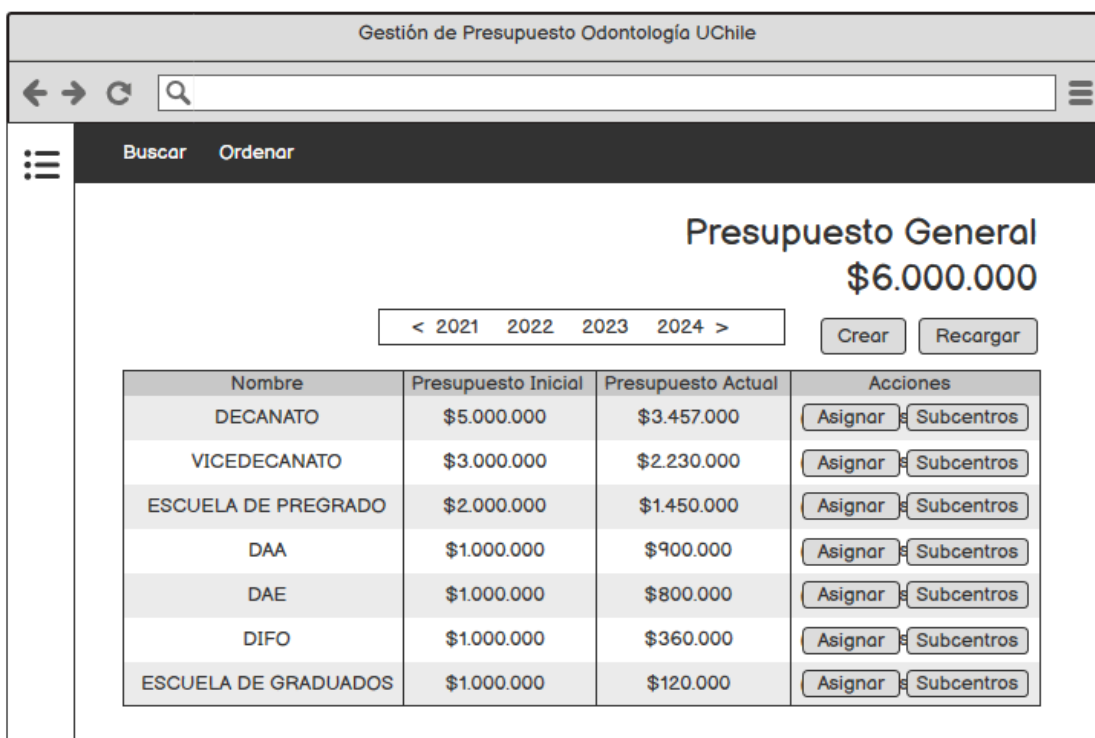


**Figura 25. Boceto de Pantalla “Creación de Usuario”.**

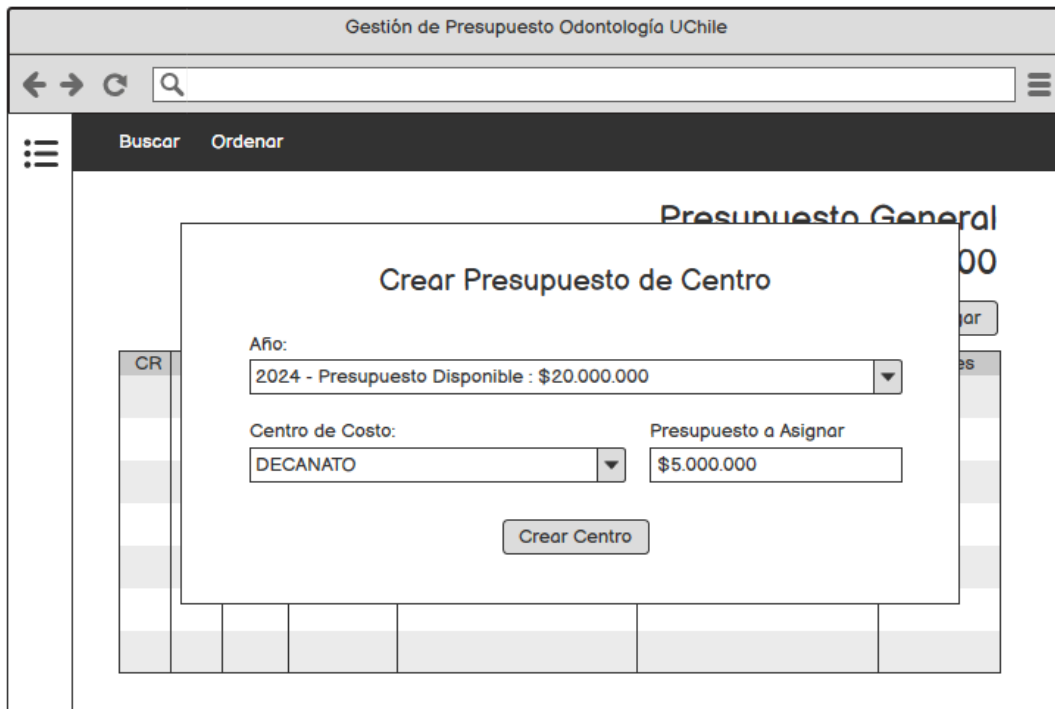
En la pantalla “Gestión de Presupuesto”, el Administrador podrá asignar el Presupuesto General Anual de la Facultad de Odontología, además podrá asignar también el Presupuesto para los Centros y Subcentros de la Facultad.



**Figura 26. Boceto de Pantalla “Gestión de Presupuesto General Anual”.**

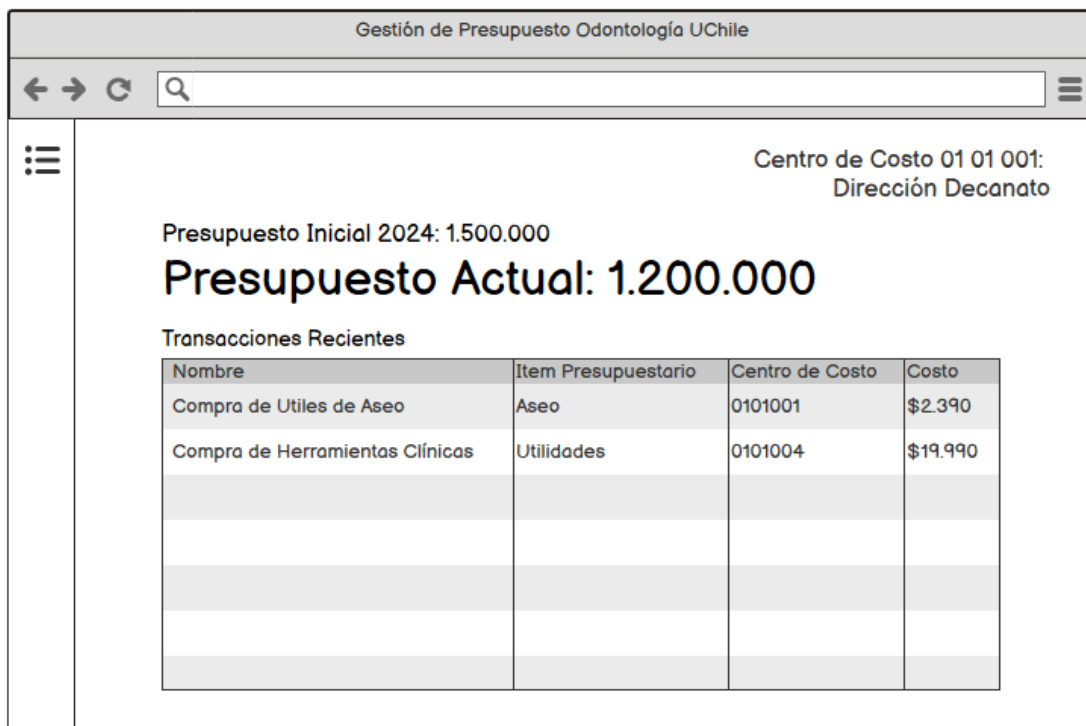


**Figura 27. Boceto de Pantalla “Gestión de Presupuesto de Centros de Costo”.**



**Figura 28. Boceto de Pantalla “Asignación de Presupuesto de Centro de Costo”.**

Los usuarios con rol “Responsable” al iniciar sesión, accederán a su Panel de Centro de Costo donde podrán ver la información del presupuesto de su Centro de Costo asignado.



**Figura 29. Boceto de Pantalla “Panel de Responsable de Centro de Costo”.**

## **6. CONCLUSIÓN**

En esta primera etapa del proyecto de desarrollo del sistema de gestión y control de presupuesto para la Facultad de Odontología de la Universidad de Chile, se ha realizado un análisis de los requisitos para acordar las funcionalidades a implementar, se han establecido la herramientas necesarias, el alcance y los objetivos del sistema. La colaboración con el cliente ha sido clave para lograr una clara comprensión de las necesidades y problemas que tienen actualmente en el proceso de Gestión de Presupuesto de la Facultad de Odontología de la Universidad de Chile.

En términos técnicos, se ha definido la arquitectura del sistema, optando por un enfoque de microservicios utilizando tecnologías como Angular para el frontend y NestJS para los servicios backend. Esta elección se basa en la escalabilidad, eficiencia y desacoplamiento que ofrecen los microservicios, así como en la capacidad de comunicación entre ellos mediante RabbitMQ.

En resumen, esta primera etapa del proyecto ha sentado las bases necesarias para un desarrollo exitoso, con una comprensión clara de los requisitos, una comunicación efectiva con el cliente y una planificación detallada de los pasos a seguir en las siguientes etapas del proyecto.

## 7. REFERENCIAS

- [1] Facultad de Odontología. Universidad de Chile. (s. f.). <https://odontologia.uchile.cl/>
- [2] Atlassian. (s. f.). ¿Qué es scrum?. Atlassian. <https://www.atlassian.com/es/agile/scrum>
- [3] Página Web de Bizagi. (2024, febrero). Software gratuito de mapeo y modelamiento de procesos de negocio. Bizagi Modeler. <https://www.bizagi.com/es/plataforma/modeler>
- [4] NestJS - A progressive Node.js framework. (s. f.). NestJS. <https://nestjs.com/>
- [5] MongoDB. (s. f.). MongoDB: the Developer Data platform. <https://www.mongodb.com/>
- [6] Docker: Accelerated Container Application Development. (2024, abril). Docker. <https://www.docker.com/>
- [7] RabbitMQ: One broker to queue them all. RabbitMQ. (s. f.). <https://www.rabbitmq.com/>
- [8] Balsamiq: Fast, focused wireframing for teams and individuals. Balsamiq. (s. f.). <https://balsamiq.com/>