



**Universidad de Chile**

Departamento de Ciencias de la Computación

## SPRINT 2

*Ingeniería de Software*

CC4401

Autores:

Benjamín Contreras, Eduardo Reyes, José Videla,  
Mario Calderón, Matías Cisterna

Profesor:

Jocelyn Simmonds

4 de julio de 2023

# Índice de Contenidos

<b>1. Sección 1</b>	<b>3</b>
<b>2. Sección 2</b>	<b>4</b>
2.1. Estado del backlog . . . . .	4
<b>3. Sección 3</b>	<b>6</b>
3.1. Modelo de datos . . . . .	6
<b>4. Sección 4</b>	<b>7</b>
4.1. Recapitulación de features . . . . .	7

# 1. Sección 1

## Resumen

El proyecto a desarrollar en este segundo Sprint contempla uno de los proyectos sugeridos por el equipo docente, en particular, el Proyecto de **Tipo A**: un sistema de información financiera. Este proyecto está construido como una Aplicación Web bajo el framework de Django.

En este proyecto pretendemos ayudar a personas que quieren ordenar sus finanzas. Para esta oportunidad los usuarios de esta aplicación podrían:

1. Crear, modificar y borrar ingresos/egresos (Sprint 1)
2. Ver su saldo disponible (Sprint 1)
3. Asignar categorías a sus ingresos/egresos (Sprint 1)
4. Listar sus ingresos/egresos por categoría (Sprint 2)
5. Listar sus ingresos/egresos en un rango de fechas (Sprint 2)

Además, cuenta con características destacables de mencionar como:

1. Login page (Sprint 1)
2. Register page (Sprint 1)
3. User page (Sprint 1)
4. Panel de transacciones (Sprint 1)
5. Formulario creador/editor y borrador de transacciones (Sprint 1)
6. Landing page (Sprint 2)
7. Diseño y estética del panel de usuario (Sprint 2)
8. Manejo de algunos bugs/errores menores en la implementación de las features (Sprint 2)

## 2. Sección 2

### 2.1. Estado del backlog

Como punto inicial fabricamos un backlog con la plantilla propuesta por el equipo docente lo que nos dio una distribución de las horas para el equipo. Este consiste en un orden cronológico de tareas por realizarse para implementar cada una de las 2 features restantes para este segundo Sprint.

Durante las reuniones con nuestro equipo auxiliar hicimos modificaciones al backlog pues tenía instrucciones demasiado grandes que hacían difícil rastrear el progreso que estábamos realizando. El siguiente es el estado final del proyecto:

Horas por persona para Sprint 2:	28			Total Hrs Disponible	140
Total Estimado (suma columna D)				133	
Horas por asignar				7	
Id	Descripción	Hrs	Sprint	Encargado	Estado
1	<b>Coordinación:</b> Reunión semanal (recurrente) (Fijada los viernes a las 6 de la tarde)	1	2	Todos	Terminado ▼
2	<b>Informe y Presentación:</b> Definir Usuarios (el usuario podría con trabajo que necesite llevar registro de sus ingresos y gastos en una plataforma portable como una página web, de manera que sus datos se conservan en la nube y no se maneje mucho con la tecnología)	8	2	José	Terminado ▼
3	<b>Informe:</b> Crear el contenido del informe	4	2	Benjamín y Eduardo	Terminado ▼
4	<b>Template y View:</b> Manejo de mensajes de error	10	2	Benjamín	Terminado ▼
5	<b>Presentación:</b> Crear las slides de la presentación	10	2	Benjamín y Eduardo	Terminado ▼
6	<b>Template (F4):</b> Hacer que la misma página muestre la tabla ordenada por un rango de fechas	10	2	José y Matías	Terminado ▼
7	<b>Template (F5):</b> Hacer que la misma página muestre la tabla ordenada por categoría	10	2	José y Matías	Terminado ▼

Figura 1: Backlog

8	<b>Template y View (Mejora F1):</b> Agregar modificar para template y view para que sea desplegable como un popup, dependiendo de la columna seleccionada	10	1	José y Matías	Terminado ▼
9	<b>Template y View (Mejora F1):</b> Modificar borrado de manera que se pueda borrar una columna a través del boton de borrar	10	1	José y Matías	Terminado ▼
10	<b>Template:</b> Realizar el estilo CSS de la página de inicio	10	2	Eduardo	Terminado ▼
11	<b>Template:</b> Realizar el template de la página de inicio	6	2	Eduardo	Terminado ▼
12	<b>Template:</b> Resolver conflictos de estilos con Bootstrap	2	2	José y Matías	Terminado ▼
13	<b>View (F4):</b> Modificar view transacciones para ordenar por rango de fechas	10	2	Mario	Terminado ▼
14	<b>View (F5):</b> Modificar view transacciones para ordenar por categoría	10	2	Mario	Terminado ▼
15	<b>View:</b> Verificar que no entre usuarios que no estén autenticados en las views que corresponda	8	2	Benjamín	Terminado ▼
16	<b>Template:</b> Modificar estructura de templates para que hereden de un base.html	8	2	José	Terminado ▼
17	<b>Template y View (Mejora F1):</b> Hacer que el modal de la opción modificar en la tabla muestre los valores iniciales por defecto.	2	2	José	Terminado ▼
18	<b>Template y View:</b> Hacer port de las forms normales a forms de Django.	2	2	José	Terminado ▼
19	<b>Template y View:</b> Borra categorías.	2	2	Mario	Terminado ▼

Figura 2: Backlog

Como se ven en las imágenes, todas las tareas propuestas fueron completadas. No hay casos de tareas iniciadas sin completar o tareas no iniciadas.

### 3. Sección 3

#### 3.1. Modelo de datos

El modelo de datos no tuvo mayores cambios respecto al primer sprint. Este consta de 3 entidades, Usuario, Transacciones y Categorías:

- El usuario llevará el registro de su nombre y su saldo registrado actualmente.

Cada usuario podrá crear, modificar o eliminar cuantas transacciones quiera, siempre que estas le pertenezcan.

- Las transacciones tendrán registro del usuario al cual pertenecen como llave primaria, mientras que el nombre de la transacción, monto y fecha contendrán información descriptiva de esta. Finalmente, el campo Categoría llevará una llave foránea para una entidad Categoría.

Cada transacción contendrá 0 o 1 categoría asociada.

- Las Categorías solo llevarán registro de su nombre y del usuario que las creó.

Cada usuario puede crear 0 o más Categorías y cada Categoría puede existir en 0 o más transacciones.

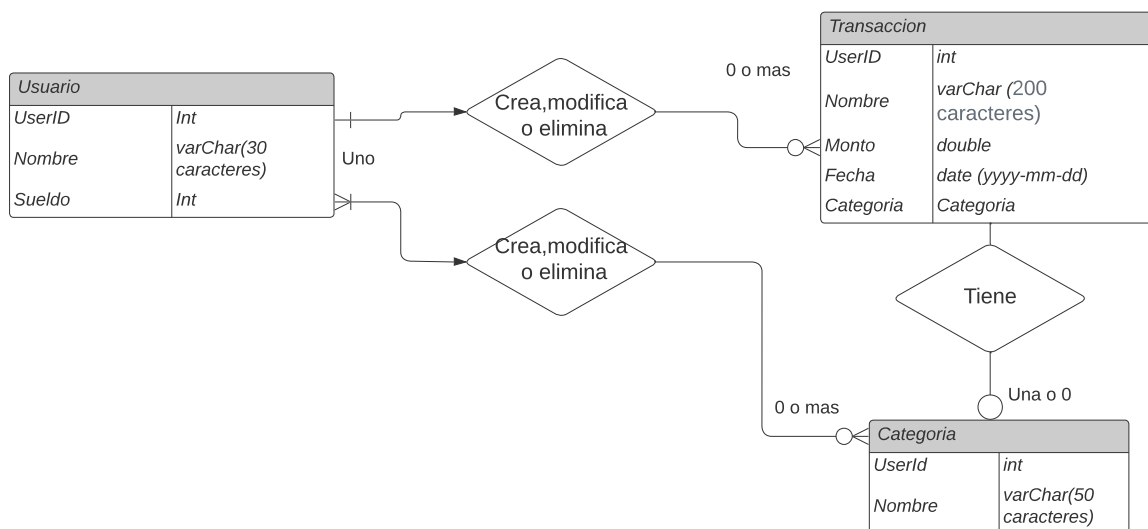


Figura 3: Diagrama del modelo de datos

## 4. Sección 4

### 4.1. Recapitulación de features

1. Crear, modificar y borrar ingresos/egresos (Sprint 1)
2. Ver su saldo disponible (Sprint 1)
3. Asignar categorías a sus ingresos/egresos (Sprint 1)
4. Borrar categorías existentes (Sprint 2)
5. Mejorar el apartado visual de la aplicación (Sprint 2)
6. Listar sus ingresos/egresos por categoría (Sprint 2)
7. Listar sus ingresos/egresos en un rango de fechas (Sprint 2)

Este segundo Sprint tenía por meta lograr que nuestro usuario pudiera manipular de mejor manera una gran cantidad de transacciones si así lo desease. Para ello se trabajó en las últimas dos features que tienen lugar en una representación mediante botones y selectores alrededor de la tabla que lista todas las transacciones para el usuario (limitada por un display finito y ajustable).

Se creó una *landing page* que permitiera redirigir a el login o a la página de transacciones si ya se está loggeado, esta contiene información sobre la aplicación.



Figura 4: Parte superior de la *landing page*

Se mejoró el apartado visual en las páginas de login y registro de usuario, también se implementó manejo de errores para inputs inválidos y se informa al usuario en caso de que estos ocurran.

**Registro**

Nombre de Usuario

Contraseña

¿Olvidaste tu contraseña?

**\* Por favor ingrese un nombre de usuario.**  
**\* Por favor ingrese una contraseña.**  
**\* Tu contraseña debe de ser de al menos 6 caracteres.**

**Crear Usuario**

¿Tienes cuenta? [Ingresa](#)

Figura 5: Registro de usuario.

**Login**

Nombre de Usuario

Contraseña

¿Olvidaste tu contraseña?

**\* Usuario o contraseña incorrectos, por favor intente de nuevo.**

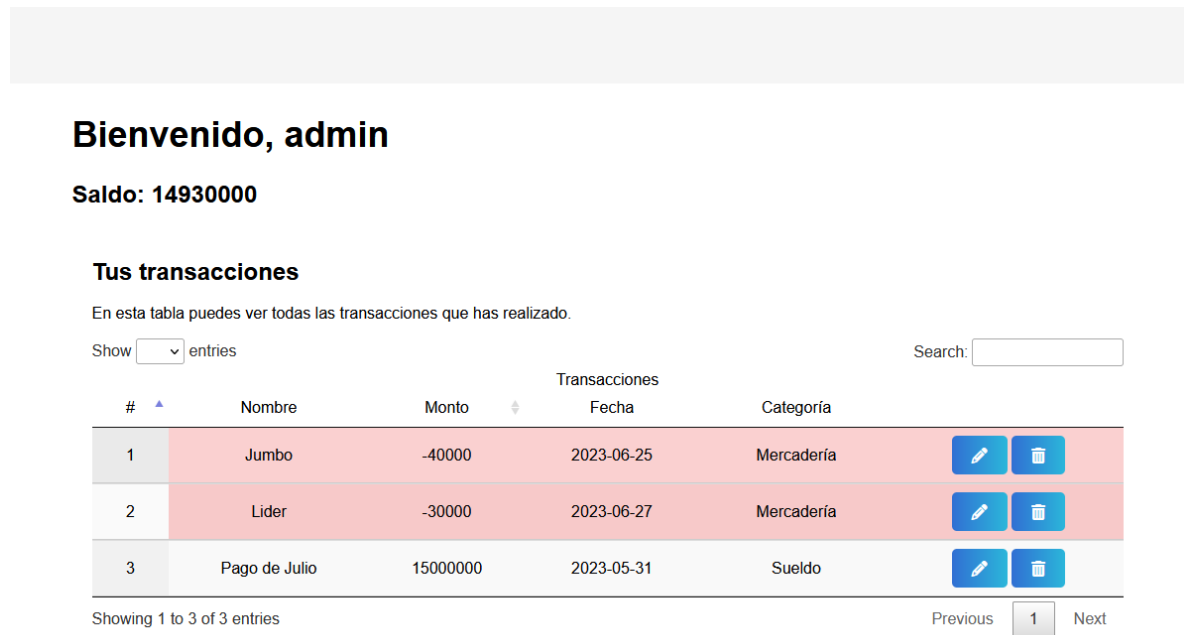
**Ingresar**

¿No tienes cuenta? [Regístrate](#)

Figura 6: Login de usuario.



La imagen siguiente corresponde a la lista de transacciones, en esta se pueden ver todas las transacciones hechas, es posible modificar o eliminar cualquier transacción presionando el ícono correspondiente, al editar se abre un popup en el que se puede modificar los parámetros de la transacción seleccionada. Es posible elegir cuántas transacciones mostrar a la vez en *Show entries* y se puede buscar una transacción por su nombre con *Search*.



**Bienvenido, admin**

**Saldo: 14930000**

**Tus transacciones**

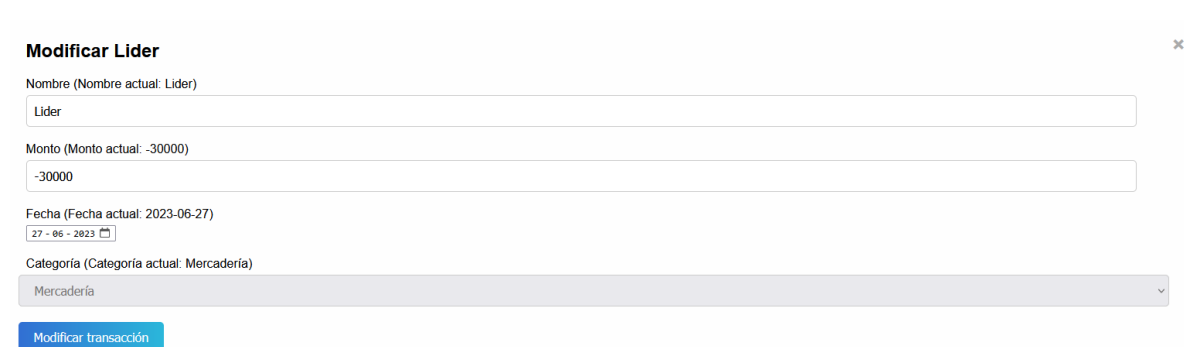
En esta tabla puedes ver todas las transacciones que has realizado.

Show  entries Search:

#	Nombre	Monto	Fecha	Categoría	
1	Jumbo	-40000	2023-06-25	Mercadería	
2	Lider	-30000	2023-06-27	Mercadería	
3	Pago de Julio	15000000	2023-05-31	Sueldo	

Showing 1 to 3 of 3 entries Previous **1** Next

Figura 7: Lista de transacciones realizadas.



**Modificar Lider** ✕

Nombre (Nombre actual: Lider)

Monto (Monto actual: -30000)

Fecha (Fecha actual: 2023-06-27)

Categoría (Categoría actual: Mercadería)

Figura 8: Popup al modificar una transacción.

En la siguiente imagen se puede observar la interfaz de filtro de transacciones, la primera opción de filtros de fecha permite mostrar solo transacciones en el rango de fechas elegidos, la segunda

opción permite mostrar solo las transacciones de la categoría seleccionada, si se deja en *Elige una categoría* se muestran transacciones de cualquier categoría, clicar *Restablecer filtros* muestra todas las transacciones sin ningún filtro. Al cambiar las opciones de filtro se actualiza en tiempo real (sin recargar la página) los resultados de la tabla de transacciones (Figura 7).

**Filtrar transacciones**

Si deseas filtrar una lista específica de transacciones puedes hacerlo con los siguientes atributos:

- Si deseas filtrar tus transacciones por fecha, debes seleccionar una fecha de inicio y una de fin.
- Si deseas filtrar tus transacciones por categoría, selecciona la categoría que desees filtrar.
- También puedes filtrar usando ambos atributos a la vez.

**Fecha de inicio:**

**Fecha de término:**

**Filtrar por categoría:**

Figura 9: Opciones de filtro para las transacciones.

Respecto a los formularios para crear categorías y transacciones, se agregó la opción de eliminar categorías y se cambió el estilo de los formularios por uno más atractivo visualmente.

**Agregar transacción**

Nombre de tu transacción

Monto

Fecha

Categoría

**Agregar categoría**

Nombre de tu nueva categoría

**Eliminar categoría**

Selecciona la categoría a eliminar

Figura 10: Creación de transacción, creación y eliminación de categoría.