Universidad
Nacional de
Ingeniería

Área de Conocimiento de Tecnología de la Información y Comunicación

Desarrollo de Sistema Web de Gestión Financiero Administrativo (SIGFA) para la Universidad Nacional de Ingeniería

Trabajo monográfico para optar al título de Ingeniería en Computación

Elaborado por:

Br. Ronnie Aaron Acevedo Cisnero
Carnet: 2017-1335U

Br. Alfredo Antonio Eva Vargas
Carnet: 2017-0210U

Tutor:

TeknL.Ing. Anayanci López Poveda

Noviembre del 2024

Managua, Nicaragua

Área de Conocimiento de
Tecnología de la Información
y Comunicación

SECRETARIA DE ÁREA ACADÉMICA

F-8:CARTA DE FINALIZADO PLAN DE ASIGNATURA

El Suscrito Secretario del **ÁREA DEL CONOCIMIENTO DE TECNOLOGÍA DE LA INFORMACIÓN Y COMUNICACIÓN** hace constar que:

EVA VARGAS ALFREDO ANTONIO

Carné: **2017-0210U** Turno: **Diurno** Plan de Asignatura: **2015** de conformidad con el Reglamento Académico vigente en la Universidad, ha aprobado todas las asignaturas correspondientes a la carrera de **INGENIERÍA EN COMPUTACIÓN**, en el año 2022 y solo tiene pendiente la realización de una de las formas de culminación de estudio.

Se extiende la presente **CARTA DE FINALIZADO PLAN DE ASIGNATURA**, a solicitud del interesado en la ciudad de Managua, a los trece días del mes de junio del año dos mil veinte y cuatro.

Atentamente,

Ing. Cedrick Elksnherr DallaTorre Rorales
SECRETARIO DE ÁREA ACADÉMICA

UNIVERSIDAD NACIONAL DE INGENIERIA · REPÚBLICA DE NICARAGUA · AMERICA CENTRAL · SECRETARIA DE FACULTAD

Móvil: (505) 8588 8333

Recinto Universitario Simón Bolívar
Avenida Universitaria.
Managua, Nicaragua.
Apdo: 5595

www.uni.edu.ni

Área de Conocimiento de
Tecnología de la Información
y Comunicación

SECRETARIA DE ÁREA ACADÉMICA

F-8:CARTA DE FINALIZADO PLAN DE ASIGNATURA

El Suscrito Secretario del **ÁREA DEL CONOCIMIENTO DE TECNOLOGÍA DE LA INFORMACIÓN Y COMUNICACIÓN** hace constar que:

ACEVEDO CISNERO RONNIE AARON

Carné: **2017-1335U** Turno: **Diurno** Plan de Asignatura: **2015** de conformidad con el Reglamento Académico vigente en la Universidad, ha aprobado todas las asignaturas correspondientes a la carrera de **INGENIERÍA EN COMPUTACIÓN**, en el año 2022 y solo tiene pendiente la realización de una de las formas de culminación de estudio.

Se extiende la presente **CARTA DE FINALIZADO PLAN DE ASIGNATURA**, a solicitud del interesado en la ciudad de Managua, a los trece días del mes de junio del año dos mil veinte y cuatro.

Atentamente,

Ing. Cedrick Elksnherr DallaTorre Parrales
SECRETARIO DE ÁREA ACADÉMICA

UNIVERSIDAD NACIONAL DE INGENIERIA
REPUBLICA DE NICARAGUA
AMERICA CENTRAL
SECRETARIA DE FACULTAD F.E.C.

Móvil: (505) 8588 8333

Recinto Universitario Simón Bolívar
Avenida Universitaria.
Managua, Nicaragua.
Apdo: 5595

www.uni.edu.ni

Programa Académico de Ingeniería en Computación

Managua, 06 de noviembre de 2024.

MSc. Claudia Lucia Benavidez Rugama
Directora DACTIC
Sus manos.

Estimada MSc. Benavidez,

Asunto: Carta Aval de Tutor para Monografía

A través de la presente hago de su conocimiento que los estudiantes **Alfredo Antonio Eva Vargas (2017-0210U)** y **Ronnie Aaron Acevedo Cisnero (2017-1335U)** han finalizado su proyecto monográfico para optar al título de **Ingeniería en Computación**, el cual se titula "**Desarrollo de Sistema Web de Gestión Financiero Administrativo (SIGFA) para la Universidad Nacional de Ingeniería**". He dado seguimiento a este proyecto monográfico y considero que su informe final (el cual se adjunta y **cumple con los requisitos indicados en las normativas correspondientes**: Normativa de Culminación de Estudio de la UNI, y normativa interna correspondiente de la DACTIC) **está listo para pasar al proceso de defensa.**

Sin más que agregar, me despido agradeciendo su amable atención a esta comunicación.

Cordialmente,

TeknL.Ing. Anayanci López P.
Profesor Titular
Ingeniería en Computación
DACTIC
Universidad Nacional de Ingeniería
anayanci.lopez@dactic.uni.edu.ni

Área de Conocimiento de
Tecnología de la Información
y Comunicación

Managua, 19 de Febrero 2024

Bachilleres
Alfredo Antonio Eva Vargas – 2017-0210U
Ronnie Aarón Acevedo Cisnero – 2017-1335U
Estudiantes Egresados del Programa Académico Ingeniería en Computación
Sus Manos. -

Estimados Bachilleres:

Por medio de la presente, les comunico la aprobación e inscripción del Protocolo de trabajo monográfico, titulado: **"Desarrollo de Sistema Web de Gestión Financiero Administrativo (SIGFA) para la Universidad Nacional de Ingeniería"**, el cual cumple con los requisitos establecidos en el capítulo II de la Normativa para los Trabajos Monográficos de la Universidad Nacional de Ingeniería (UNI), como Forma de Culminación de Estudios.

No omito manifestar, que el **TeknL. Ing. Anayanci López Poveda**, es la docente encargada de acompañarlos con responsabilidad y compromiso como tutor en el proceso de desarrollo de la monografía.

Así mismo, en correspondencia con la Normativa para los Trabajos Monográficos, a partir de la fecha de aprobación e inscripción tendrá un máximo de 12 meses para la ejecución y entrega de los ejemplares para la organización de la predefensa y posterior defensa del mismo.

Sin más a que hacer referencia, les deseo s el mejor de los éxitos en la culminación de esta etapa, les saludo.

Atentamente,

Msc. Claudia Lucía Benavidez Rugama
Directora Área de Conocimiento de
Tecnología de la Información y Comunicación

Cc: TeknL. Ing. Anayanci López Poveda – Tutor
Ing. Cedrick Dalla Torre – Secretario Académico DAC-TIC
Archivo DAC-TIC 2024

Móvil: (505) 8588 8333

Recinto Universitario Simón Bolívar
Avenida Universitaria.
Managua, Nicaragua.
Apdo: 5595

www.uni.edu.ni

UNIVERSIDAD NACIONAL DE INGENIERIA
MANAGUA

DIRECCION DE REGISTRO
CERTIFICACION DE ESTUDIOS No. 0044183

LA DIRECCION DE REGISTRO DE LA UNIVERSIDAD NACIONAL DE INGENIERIA, CERTIFICA QUE SEGUN LOS ARCHIVOS ACADEMICOS QUE ESTA DIRECCION LLEVA A SU CARGO, RONNIE AARON ACEVEDO CISNERO

NUMERO DE IDENTIDAD: 2017-1335U ALUMNO (A) DE LA CARRERA DE: INGENIERÍA EN COMPUTACIÓN

QUE OFRECE LA FACULTAD DE: **DACTIC** OBTUVO EN LOS EXAMENES DE FIN DE CURSO, LAS CALIFICACIONES QUE A CONTINUACION SE DETALLAN:

CODIGO	ASIGNATURA	NOTA	CALIFICACION
	PRIMER SEMESTRE 2017		
CO0437	FILOSOFIA	85.00	ochenta y cinco
CON426	GEOMETRIA ANALITICA DESCRIPTIVA	60.00	sesenta
CO0440	INGLES I	61.00	sesenta y uno
CON425	INTRODUCCION A LA INGENIERIA EN COMPUTACION	73.00	setenta y tres
	SEGUNDO SEMESTRE 2017		
CO0162	CONCEPTOS DE LENGUAJES	80.00	ochenta
CO0441	INGLES II	63.00	sesenta y tres
CO0292	LENGUAJE DE PROGRAMACION	75.00	setenta y cinco
CO0N44	MATEMATICA I PARA COMPUTACION	69.00	sesenta y nueve
CO3002	REDACCION TECNICA	73.00	setenta y tres
	PRIMER SEMESTRE 2018		
CO0430	GEOMETRIA COMPUTACIONAL	60.00	sesenta
CO0N45	MATEMATICA II PARA COMPUTACION	60.00	sesenta
CO0442	PROGRAMACION ORIENTADA A OBJETO	83.00	ochenta y tres
CO8N29	SOCIOLOGIA	100.00	cien
	SEGUNDO SEMESTRE 2018		
CO0135	ALGORITMIZACION Y ESTRUCTURAS DE DATOS	90.00	noventa
CO8004	CULTURA DE PAZ Y DERECHOS HUMANOS	81.00	ochenta y uno
CO0025	FISICA I	60.00	sesenta
CO0N46	MATEMATICA III PARA COMPUTACION	91.00	noventa y uno
CO1001	MATEMATICAS DISCRETAS	82.00	ochenta y dos
	PRIMER SEMESTRE 2019		
CO0026	FISICA II	60.00	sesenta
CO0450	PROGRAMACION GRAFICA	100.00	cien
CO0388	SISTEMAS DIGITALES	80.00	ochenta
CO0453	TEORIA DE LA COMPUTACION	89.00	ochenta y nueve
	SEGUNDO SEMESTRE 2019		
CO0432	BASES DE DATOS	81.00	ochenta y uno
CO1000	ECONOMIA	72.00	setenta y dos
CO0N16	ESTADISTICA I	60.00	sesenta
CO0445	HARDWARE REPARACION Y MANTENIMIENTO	82.00	ochenta y dos
CO0444	SISTEMAS OPERATIVOS Y REDES	91.00	noventa y uno
	PRIMER SEMESTRE 2020		
CO0448	ECONOMIA DE PROYECTOS INFORMATICOS	68.00	sesenta y ocho
CO0435	SIMULACION	79.00	setenta y nueve
	SEGUNDO SEMESTRE 2020		
CO0145	ARQUITECTURA DE SISTEMAS OPERATIVOS	87.00	ochenta y siete
CO0182	CONTROL	75.00	setenta y cinco
CO8000	METODOLOGIA DE LA INVESTIGACION	71.00	setenta y uno
	PRIMER SEMESTRE 2021		
CO0443	ADMINISTRADORES DE BASES DE DATOS	87.00	ochenta y siete
CO0446	ARQUITECTURA DE MAQUINAS COMPUTADORAS I	81.00	ochenta y uno
	SEGUNDO SEMESTRE 2021		

ESCALA DE CALIFICACIONES: DE CERO (0) A CIEN (100) NOTA MINIMA PARA APROBAR: SESENTA (60)

FORMULARIOS STANDARD INC. S.A. RUC No. J031000001569 C.T. No. 3021969 9 M. CEL No. 41501 AL 44600 08/2023

CODIGO	ASIGNATURA	NOTA	CALIFICACION
CO0447	ARQUITECTURA DE MAQUINAS COMPUTADORAS II	100.00	cien
CO0272	INGENIERIA DE SOFTWARE I	89.00	ochenta y nueve
	CURSO DE VERANO 2021		
CO8028	REDES DE COMPUTADORAS	75.00	setenta y cinco
	PRIMER SEMESTRE 2022		
CO0273	INGENIERIA DE SOFTWARE II	83.00	ochenta y tres
CO8019	TENDENCIAS TECNOLOGICAS	96.00	noventa y seis
	SEGUNDO SEMESTRE 2022		
CO8030	ADMINISTRACION DE SISTEMAS DE INFORMACION	85.00	ochenta y cinco
CO8024	INGENIERIA DE SOFTWARE III	90.00	noventa
	CURSO DE VERANO 2022		
CO0449	ARQUITECTURA DE MAQUINAS COMPUTADORAS III	75.00	setenta y cinco
CO0434	INTELIGENCIA ARTIFICIAL	96.00	noventa y seis

Esta es una certificación parcial de la historia académica del estudiante y consta de una hoja escrita en las dos caras. Es conforme con su original con el que fue debidamente cotejado. A solicitud de parte interesada, se extiende la presente Certificación en la ciudad de Managua, a los veintiséis días del mes de agosto del año dos mil veinticuatro.

MBA. Ronald Antonio Torres Mercado
Director del Departamento de Registro - UNI

Notas: Facultad= Área del Conocimiento
DACTIC: Dirección del Área del Conocimiento de Tecnología de la Información y Comunicación

UNIVERSIDAD NACIONAL DE INGENIERIA
MANAGUA

DIRECCION DE REGISTRO
CERTIFICACION DE ESTUDIOS No. 0044184

LA DIRECCION DE REGISTRO DE LA UNIVERSIDAD NACIONAL DE INGENIERIA, CERTIFICA QUE SEGUN LOS ARCHIVOS ACADEMICOS QUE ESTA DIRECCION LLEVA A SU CARGO, ALFREDO ANTONIO EVA VARGAS

NUMERO DE IDENTIDAD: 2017-0210U ALUMNO (A) DE LA CARRERA DE: INGENIERÍA EN COMPUTACIÓN

QUE OFRECE LA FACULTAD DE: **DACTIC** OBTUVO EN LOS EXAMENES DE FIN DE CURSO, LAS CALIFICACIONES QUE A CONTINUACION SE DETALLAN:

CODIGO	ASIGNATURA	NOTA	CALIFICACION
	PRIMER SEMESTRE 2017		
CO0437	FILOSOFIA	85.00	ochenta y cinco
CON426	GEOMETRIA ANALITICA DESCRIPTIVA	74.00	setenta y cuatro
CO0440	INGLES I	71.00	setenta y uno
CON425	INTRODUCCION A LA INGENIERIA EN COMPUTACION	85.00	ochenta y cinco
	SEGUNDO SEMESTRE 2017		
CO0162	CONCEPTOS DE LENGUAJES	80.00	ochenta
CO0441	INGLES II	60.00	sesenta
CO0292	LENGUAJE DE PROGRAMACION	97.00	noventa y siete
CO0N44	MATEMATICA I PARA COMPUTACION	80.00	ochenta
CO3002	REDACCION TECNICA	78.00	setenta y ocho
	PRIMER SEMESTRE 2018		
CO0430	GEOMETRIA COMPUTACIONAL	60.00	sesenta
CO0N45	MATEMATICA II PARA COMPUTACION	100.00	cien
CO0442	PROGRAMACION ORIENTADA A OBJETO	100.00	cien
CO8N29	SOCIOLOGIA	100.00	cien
	SEGUNDO SEMESTRE 2018		
CO0135	ALGORITMIZACION Y ESTRUCTURAS DE DATOS	95.00	noventa y cinco
CO8004	CULTURA DE PAZ Y DERECHOS HUMANOS	73.00	setenta y tres
CO0025	FISICA I	80.00	ochenta
CO0N46	MATEMATICA III PARA COMPUTACION	94.00	noventa y cuatro
CO1001	MATEMATICAS DISCRETAS	80.00	ochenta
	PRIMER SEMESTRE 2019		
CO0026	FISICA II	76.00	setenta y seis
CO0450	PROGRAMACION GRAFICA	100.00	cien
CO0388	SISTEMAS DIGITALES	100.00	cien
CO0453	TEORIA DE LA COMPUTACION	100.00	cien
	SEGUNDO SEMESTRE 2019		
CO0432	BASES DE DATOS	100.00	cien
CO1000	ECONOMIA	83.00	ochenta y tres
CO0N16	ESTADISTICA I	80.00	ochenta
CO0445	HARDWARE REPARACION Y MANTENIMIENTO	86.00	ochenta y seis
CO0444	SISTEMAS OPERATIVOS Y REDES	91.00	noventa y uno
	PRIMER SEMESTRE 2021		
CO0443	ADMINISTRADORES DE BASES DE DATOS	95.00	noventa y cinco
CO0446	ARQUITECTURA DE MAQUINAS COMPUTADORAS I	90.00	noventa
CO0448	ECONOMIA DE PROYECTOS INFORMATICOS	100.00	cien
CO0435	SIMULACION	87.00	ochenta y siete
	SEGUNDO SEMESTRE 2021		
CO0447	ARQUITECTURA DE MAQUINAS COMPUTADORAS II	100.00	cien
CO0145	ARQUITECTURA DE SISTEMAS OPERATIVOS	87.00	ochenta y siete
CO0182	CONTROL	90.00	noventa
CO0272	INGENIERIA DE SOFTWARE I	95.00	noventa y cinco
CO8000	METODOLOGIA DE LA INVESTIGACION	100.00	cien
	PRIMER SEMESTRE 2022		

ESCALA DE CALIFICACIONES: DE CERO (0) A CIEN (100) NOTA MINIMA PARA APROBAR: SESENTA (60)

FORMULARIOS STANDARD INC. S.A. RUC No. J0310000010001 C.T. No. 3821969 S.M. DEL No. 41501 AL 44500 08/2023

CODIGO	ASIGNATURA	NOTA	CALIFICACION
CO0273	INGENIERIA DE SOFTWARE II	89.00	ochenta y nueve
CO0434	INTELIGENCIA ARTIFICIAL	98.00	noventa y ocho
CO8019	TENDENCIAS TECNOLOGICAS	99.00	noventa y nueve
	SEGUNDO SEMESTRE 2022		
CO8030	ADMINISTRACION DE SISTEMAS DE INFORMACION	90.00	noventa
CO8024	INGENIERIA DE SOFTWARE III	90.00	noventa
	CURSO DE VERANO 2022		
CO0449	ARQUITECTURA DE MAQUINAS COMPUTADORAS III	83.00	ochenta y tres
CO8028	REDES DE COMPUTADORAS	100.00	cien

Esta es una certificación parcial de la historia académica del estudiante y consta de una hoja escrita en las dos caras. Es conforme con su original con el que fue debidamente cotejado. A solicitud de parte interesada, se extiende la presente Certificación en la ciudad de Managua, a los veintisiete días del mes de agosto del año dos mil veinticuatro.

MBA. Ronald Antonio Torres Mercado
Director del Departamento de Registro - UNI

Notas: Facultad= Área del Conocimiento
DACTIC: Dirección del Área del Conocimiento de Tecnología de la Información y Comunicación

Dedicatoria

Este trabajo monográfico está dedicado a Dios, por su guía y fortaleza en cada paso de este camino.

A mis padres, Dora Cisnero y Rubén Acevedo, por su amor incondicional, apoyo constante y por ser mi fuente de inspiración.

A mi tutora, TeknL. Ing. Anayanci López Poveda, por su invaluable guía, paciencia y sabiduría, que han sido cruciales para la realización de este proyecto.

Con profundo agradecimiento y dedicación,

Ronnie Aaron Acevedo Cisnero.

Le dedico a Dios, por la gran bendición de escribir esto, por brindarme el deseo de superación y lo que existe a mi alrededor para poder desarrollarme; y por las personas increíbles que Dios destina en el camino de la vida, que han influido en mi crecimiento personal, académico y profesional.

A mis Padres (Evelin Vargas y Wilmer Eva), por alentarme a establecer metas en mi vida y su apoyo incondicional en todo momento para alcanzar este logro compartido,

Alfredo Antonio Eva Vargas.

Resumen

En la Universidad Nacional de Ingeniería (UNI) la gestión financiera y administrativa previo a este proyecto se ha realizado a través del Sistema Integral Financiero y Administrativo (SIFA), el cual presenta varias problemáticas, como la descentralización de la contabilidad y la falta de manejo adecuado de ciertos documentos contables.

Para solucionar estos problemas, se desarrolló el **Sistema Web de Gestión Financiero Administrativo (SIGFA)**, con la aprobación de la Dirección de Normas y Procedimientos (DNP) de la UNI. Este nuevo sistema web incorpora nuevas funcionalidades y mayor seguridad de datos, atendiendo a diversas áreas de la universidad; centraliza las gestiones financieras y administrativas de la UNI, incluyendo sus sedes departamentales; y mejora funciones críticas como asientos contables, manejo de cheques y generación automática de información relevante para la toma de decisiones. Además, el SIGFA será un punto de centralización para otros sistemas informáticos de la universidad.

En particular, el sistema web desarrollado brinda los siguientes beneficios a la institución:

1. Parametrización de cuentas contables[1] y centralización de la contabilidad.
2. Manejo de nuevos documentos contables.
3. Protección adecuada de datos financieros y administrativos sensibles.
4. Automatización de procesos financieros y administrativos para mayor eficiencia.

El proceso de desarrollo y el resultado obtenido han permitido cumplir con los objetivos planteados en este proyecto; aplicándose las metodologías, buenas prácticas, y herramientas seleccionadas para obtener un producto con aceptación de parte de los tomadores de decisión de la institución.

[1] Parametrización contable en este contexto consiste en la correcta armonía y unión del SIPPSI y el sistema "*Tesorería SIFA"* en el nuevo sistema SIGFA.

Índice

I. Introducción 1
II. Antecedentes 3
III. Justificación 5
IV. Objetivos 7
4.1 Objetivo General 7
4.2 Objetivos Específicos 7
V. Marco Teórico 8
5.1. Metodología de desarrollo de Software a utilizar: Scrum 8
5.1.1. Eventos de Scrum 10
5.1.1.1. El Sprint 10
5.1.1.2. Sprint Planning 11
5.1.1.3. Daily Scrum 11
5.1.1.4. Sprint Review 12
5.1.1.5. Sprint Retrospective 12
5.1.2. Artefactos de Scrum 13
5.1.2.1. Product backlog 13
5.1.2.2. Sprint backlog 13
5.1.2.3. Increment 14
5.2. Herramientas de desarrollo a utilizar en este proyecto 14
5.3. Prácticas de seguridad a implementar en el proyecto basadas en OWASP 17
VI. Análisis y Presentación de Resultados 22
6.1. Diseño Metodológico 22
6.1.1. Scrum Team 22
6.1.2. Product Backlog y Sprint Planning 23
6.1.3. Sprints 23
6.1.3.1. Historias de Usuario y Sprint Backlog 23
6.1.3.2. Reuniones de trabajo Scrum 24
6.1.3.3. Construcción del Software 24
6.1.3.4. Despliegue de versión completa del sistema SIGFA 25
6.1.3.5. Pruebas de seguridad OWASP 26
6.1.4. Gestión del proyecto con Microsoft Project 26

6.2. Resultados 26
6.2.1. Procesos administrativos financieros a cubrir en este sistema 28
6.2.1.1. Conceptos fundamentales sobre actividades, procesos y entidades abordadas en este sistema: 28
6.2.1.2. Aspectos sobre el manejo contable de la Universidad Nacional de Ingeniería 34
6.2.2. Vinculación de los Módulos del Sistema SIGFA con los Procesos Administrativos a Cubrir 37
6.2.3. Product Backlog 40
6.2.4. Requerimientos No Funcionales del Sistema SIGFA 50
6.2.5. Planeación y preparación de condiciones iniciales del proyecto 50
6.2.5.1. Medidas de seguridad de OWASP implementadas en el proyecto 52
6.2.6. Sprint 2 55
6.2.6.1. Sprint Planning 55
6.2.6.2. Resumen del proceso de desarrollo de software del Sprint 2 58
6.2.6.3. Revisión del Sprint 2: entrega del producto terminado, Cuentas Contables 61
6.2.6.4. Retrospectiva de Sprint 2 63
6.2.7. Sprint 7 64
6.2.7.1. Sprint Planning 64
6.2.7.2. Resumen del proceso de desarrollo de software del Sprint 7 66
6.2.7.3. Revisión del Sprint 7: entrega de producto terminado, Gestionar Asiento Contable 69
6.2.7.4. Retrospectiva de Sprint 7 70
6.2.8. Sprint 8 72
6.2.8.1. Sprint Planning 72
6.2.8.2. Resumen del proceso de desarrollo de software del Sprint 8 75
6.2.8.3. Revisión del Sprint 8: entrega del producto terminado Nota de Crédito 78
6.2.8.4. Retrospectiva de Sprint 8 79
6.2.9. Sprint 10 80
6.2.9.1. Sprint Planning 80
6.2.9.2. Resumen del proceso de desarrollo de software del Sprint 10 82
6.2.9.3. Revisión del Sprint 10: entrega del producto terminado Parametrización 84
6.2.9.4. Retrospectiva de Sprint 10 85

6.3. Diseño de Base de Datos Completa del SIGFA 87
6.4. Pruebas.....88
6.4.1. Pruebas Funcionales.....88
6.4.2. Pruebas No Funcionales (Pruebas de seguridad): 91
6.5. Despliegue de versión finalizada del SIGFA en entorno de prueba 94
VII. Conclusiones..... 97
VIII. Recomendaciones 98
IX. Referencias..... 99
X. ANEXOS..... 101
10.1. Cartas de soporte del proyecto 101
10.2. Detalle de Sprints 1, 3, 4, 5, 6, 9 y 11.....106
10.2.1. Sprint 1 106
10.2.1.1. Sprint Planning.....106
10.2.1.2. Resumen del proceso de desarrollo de software del Sprint 1 109
10.2.1.3. Revisión del Sprint 1: entrega del producto terminado, Periodos Contables Generales y Específicos 113
10.2.1.4. Retrospectiva de Sprint 1 115
10.2.2. Sprint 3 116
10.2.2.1. Sprint Planning.....116
10.2.2.2. Resumen del proceso de desarrollo de software del Sprint 3 118
10.2.2.3. Revisión del Sprint 3: entrega del producto terminado, Cuentas Bancarias.....122
10.2.2.4. Retrospectiva de Sprint 3 123
10.2.3. Sprint 4 124
10.2.3.1. Sprint Planning.....124
10.2.3.2. Resumen del proceso de desarrollo de software del Sprint 4 126
10.2.3.3. Revisión del Sprint 4: entrega de producto terminado, Asientos Contables Tipo.....129
10.2.3.4. Retrospectiva de Sprint 4 130
10.2.4. Sprint 5 132
10.2.4.1. Sprint Planning.....132
10.2.4.2. Resumen del proceso de desarrollo de software del Sprint 5.....134
10.2.4.3. Revisión del Sprint 5: entrega del producto terminado, Tipo Documentos Contables.....137
10.2.4.4. Retrospectiva de Sprint 5 138

10.2.5. Sprint 6 140
10.2.5.1. Sprint Planning 140
10.2.5.2. Resumen del proceso de desarrollo de software del Sprint 6 142
10.2.5.3. Revisión del Sprint 6: entrega del producto terminado, Gestionar Cheque 145
10.2.5.4. Retrospectiva de Sprint 6 146
10.2.6. Sprint 9 148
10.2.6.1. Sprint Planning 148
10.2.6.2. Resumen del proceso de desarrollo de software del Sprint 9 150
10.2.6.3. Revisión del Sprint 9: entrega del producto terminado Nota de Débito 152
10.2.6.4. Retrospectiva de Sprint 9 153
10.2.7. Sprint 11 154
10.2.7.1. Sprint Planning 154
10.2.7.2. Resumen del proceso de desarrollo de software del Sprint 11 157
10.2.7.3. Revisión del Sprint 11: entrega del producto terminado Reporte e Informes Financieros 159
10.2.7.4. Retrospectiva de Sprint 11 160
10.3. Pruebas 162
10.3.1. Pruebas Funcionales 162
10.3.2. Pruebas de Seguridad 171
10.4. Índice de CD adjunto 173

I. Introducción

Los sistemas contables son softwares informáticos que permiten gestionar el manejo del área contable de las compañías, de manera fácil y segura. Su importancia radica en la automatización de los procesos, lo cual permite optimizar el tiempo de manera significativa (Soluciones Alegra S.A.S., 2021).

El sistema contable que actualmente utiliza el área financiera de la Universidad Nacional de Ingeniería (UNI) es un sistema de escritorio llamado Sistema Integral Financiero y Administrativo (SIFA). Este sistema se encuentra desarrollado en tecnología obsoleta y no incluye reglas de negocio de importancia por lo que presenta muchas problemáticas; por ejemplo: la descentralización de la contabilidad institucional que genera retrasos, la falta de manejo de algunos documentos contables relevantes como las notas de crédito y débito, entre otros (Montiel, 2023).

Por tanto, en este trabajo se propone el desarrollo del **Sistema Web de Gestión Financiero Administrativo**, al que también por preferencia de los beneficiarios se refirieren en este documento como “Sistema Integrado de Gestión Financiero Administrativo” (SIGFA). Este sistema solventará las problemáticas existentes actuales, habilitando la realización de todas las gestiones bancarias y financieras que se efectuarán en la UNI, incluyendo sus sedes departamentales. Entre las gestiones que realizará el (SIGFA) se encuentran estas funciones: asiento contable, manejo de cheques, notas de crédito, notas de débito, reportes automáticos, entre otros[2].

Cabe mencionar que el (SIGFA) será el punto de centralización de otros sistemas informáticos como el Sistema de Información de Planificación, Presupuesto y

[2] Para mayor claridad al respecto se recomienda la lectura de la sección “6.2.1. Procesos administrativos financieros a cubrir en este sistema”, y la sección “6.2.2. Vinculación de los Módulos del Sistema SIGFA con los Procesos Administrativos a Cubrir”

Seguimiento Institucional (SIPPSI) y el sistema de tesorería. Este sistema lo usarán una amplia variedad de áreas de la UNI como:

1. División de Finanzas.
2. Instituto de Estudios Superiores (IES).
3. Sede UNI-Norte.
4. Dirección de Estudios de Postgrados y Educación Continua.
5. Administración Unidades de Servicios (AUS).

La Dirección de Normas y Procedimientos (DNP) de la UNI aprobó el desarrollo de este nuevo sistema que incluirá las soluciones de las problemáticas que presenta el anterior SIFA, además de nuevas funcionalidades que faciliten el trabajo de los usuarios finales, así como una mayor seguridad y protección de datos.

II. Antecedentes

Según indicó Yasser Montiel, Responsable de la Oficina de Sistemas de la UNI, actualmente el área financiera y administrativa de la Universidad Nacional de Ingeniería trabaja con un sistema de escritorio antiguo llamado Sistema Integral Financiero Administrativo (SIFA), el cual está desarrollado en Visual Basic 6.0, una tecnología de codificación ya obsoleta, la cual tiene que estar en constante mantenimiento ya que no atiende todas las nuevas modificaciones y se llegó a un punto que no es escalable, y no permite responder a las expectativas de los usuarios sobre la usabilidad de los sistemas que van cambiando (Montiel, 2023). La tecnología de desarrollo mencionada no va evolucionando a la par en temas de usabilidad, soporte, escalabilidad, seguridad, entre otros.

Asimismo, el Ing. Montiel indicó que el sistema SIFA presenta una serie de problemas en los cuales destacan los siguientes (Montiel, 2023):

1. En la estructura programática, ya que no contempla la categorización de los indicadores financieros de la UNI.
2. Eliminación de los datos registrados en los casos que haya corte de energía eléctrica o del internet (ya que no cuenta con un autoguardado de los elementos que se están digitando en el momento).
3. A la hora de eliminar una *cuenta parcial* se elimina también la *cuenta padre* y el detalle.
4. Por ultimo y la más importante la descentralización de las áreas de tesorería, Finanzas y SIPPSI, ya que cuando se realizan algunas transacciones como anulaciones de cheques, el monto queda flotante y esto afecta la contabilidad presupuestaria ya que no es de conocimiento de las otras áreas vía el sistema actual.
5. Entre otros.

Debido a lo anterior el SIFA ha tenido que estar en constante mantenimiento, lo que genera pérdidas monetarias y de tiempo, y afecta la ejecución de los procesos

de las diversas áreas de la UNI. En consecuencia, en la UNI, específicamente el área de Finanzas y la Dirección de Normas y Procedimientos (DNP) se vio en la necesidad de desarrollar un nuevo sistema web que permita gestionar todos estos procesos reflejando las nuevas necesidades institucionales. Así surgió la propuesta para el desarrollo de un nuevo sistema web financiero administrativo que en este caso será llamado SIGFA; iniciativa que está siendo coordinada por la Dirección de Normas y Procedimientos (DNP) de la UNI con el involucramiento de todas las áreas interesadas. Este nuevo sistema web vendrá a reemplazar de forma permanente al anterior SIFA.

III. Justificación

La justificación para implementar este sistema web financiero administrativo radica en la necesidad de: Parametrización de las cuentas contables; centralización de la contabilidad de toda la UNI; incluir nuevas gestiones de documentos contables; la creación automática de reportes en tiempo real; mejorar la protección de los datos financieros y administrativos sensibles; garantizar la eficiencia de los procesos; y mantener la confianza de los usuarios. Cada uno de estos motivos clave se desarrolla a continuación:

1. La **Parametrización** (parametrización contable), la cual consiste en la correcta armonía y unión del SIPPSI y el sistema "*Tesorería SIFA"* en el nuevo sistema SIGFA. Es decir, que las cuentas contables, cheques, recibos, y demás documentos contables se ordenen y descuenten en sus respectivas cuentas.
2. La **centralización** de la información administrativa financiera de toda la contabilidad de la UNI central, así como de las sedes RUPAP y UNI Norte.
3. El manejo de **documentos contables** como son las notas de crédito y débito, además de los periodos generales y específicos por subprograma, que son nuevas funcionalidades que no tiene el SIFA, pero sí tendrá el nuevo SIGFA.
4. El envío de **cheques** desde el sistema SIPPSI en el cual los cheques llegarán creados y aprobados para las gestiones necesarias que se vayan a realizar en el SIGFA con un determinado cheque.
5. La creación de **reportes automáticos** de ciertas gestiones que realizará el sistema SIGFA, como pueden ser los pagos de aranceles en un determinado tiempo, como el cierre diario de caja.
6. La **naturaleza sensible de los datos financieros y administrativos** que requiere una protección adecuada. Estos datos incluyen información confidencial relacionada con transacciones financieras, estados financieros, datos de estudiantes, trabajadores y proveedores. La filtración o el acceso

no autorizado a estos datos podrían tener consecuencias financieras y legales graves, así como dañar la reputación de la Universidad Nacional de Ingeniería (UNI). Por lo tanto, es fundamental la implementación de medidas de seguridad sólidas que garanticen la confidencialidad, integridad y disponibilidad de la información financiera y administrativa.

7. Los **sistemas web financieros** administrativos son utilizados para agilizar y optimizar los procesos financieros y administrativos de la UNI. Este sistema proporcionará herramientas y funcionalidades que permitirán la automatización de tareas, la gestión eficiente de transacciones, el seguimiento de presupuestos y la generación de informes precisos y oportunos.

Dado lo anterior, con la implementación de este sistema web se obtendrán los siguientes beneficios:

1. Prevención de pérdida de datos.
2. Ahorro de tiempo en gestiones contables.
3. Ahorro de tiempo al generar reportes.
4. Procesos y gestiones más rápidas, seguras y eficientes.
5. Mayor control en la administración financiera contable.
6. Auditorias más eficientes.
7. Mitigación de riesgos financieros.

IV. Objetivos

4.1 Objetivo General

Desarrollar un sistema web para llevar a cabo las gestiones financieras administrativas de la UNI de forma centralizada y segura.

4.2 Objetivos Específicos

1. Realizar el análisis requerido para el desarrollo del nuevo sistema SIGFA, tomando en cuenta a las áreas beneficiarias (finanzas, contabilidad y tesorería) y el área técnica de la UNI, División de Tecnologías de la Información (DTI).
2. Realizar el diseño del sistema web SIGFA teniendo como base el análisis previamente realizado con un enfoque de metodología ágil.
3. Codificar el sistema web SIGFA implementando buenas prácticas de seguridad, respetando la lógica de negocio, y utilizando las herramientas de desarrollo relevantes para la institución.
4. Realizar las pruebas necesarias en el sistema web SIGFA para el aseguramiento del funcionamiento correcto del mismo.
5. Desplegar el nuevo sistema SIGFA en entorno de prueba provisto por la División de Tecnologías de la Información (DTI).

V. Marco Teórico

En esta sección se aborda la estructura fundamental del marco teórico que respalda al sistema web SIGFA. Comienza por presentar los fundamentos de la metodología empleada para el desarrollo de este sistema web: Scrum. Esta elección no es casual; la agilidad, la adaptabilidad y la flexibilidad a Scrum hacen que sea una opción muy adecuada para un proyecto de este tipo.

Luego, se presentan las herramientas de desarrollo para la creación del SIGFA. Se mencionan las tecnologías que se utilizarán para la construcción del sistema, y las prácticas de seguridad fundamentadas en los estándares de OWASP que se utilizarán. Estas prácticas aseguran la protección rigurosa de los datos financieros y administrativos que la UNI maneja, así como los de sus valiosos usuarios y clientes.

Consecuentemente, este marco teórico profundizará en la metodología, herramientas de desarrollo y aspectos de seguridad que convergen en la creación y funcionamiento del sistema web SIGFA.

5.1. Metodología de desarrollo de Software a utilizar: Scrum

Scrum es una forma de hacer el trabajo en equipo en pequeñas partes a la vez, con experimentación continua y ciclos de retroalimentación en el camino para aprender y mejorar a medida que avanza. Scrum ayuda a las personas y los equipos a generar valor de forma incremental y colaborativa. Como marco ágil, Scrum proporciona la estructura suficiente para que las personas y los equipos se integren en su forma de trabajar, al tiempo que agrega las prácticas correctas para optimizar sus necesidades específicas. (Scrum.org, 2023)

La información a continuación proviene de (Schwaber & Sutherland, 2020):

Scrum requiere un Scrum Master para fomentar un entorno donde:

1. Un Product Owner ordena el trabajo de un problema complejo en un Product Backlog.

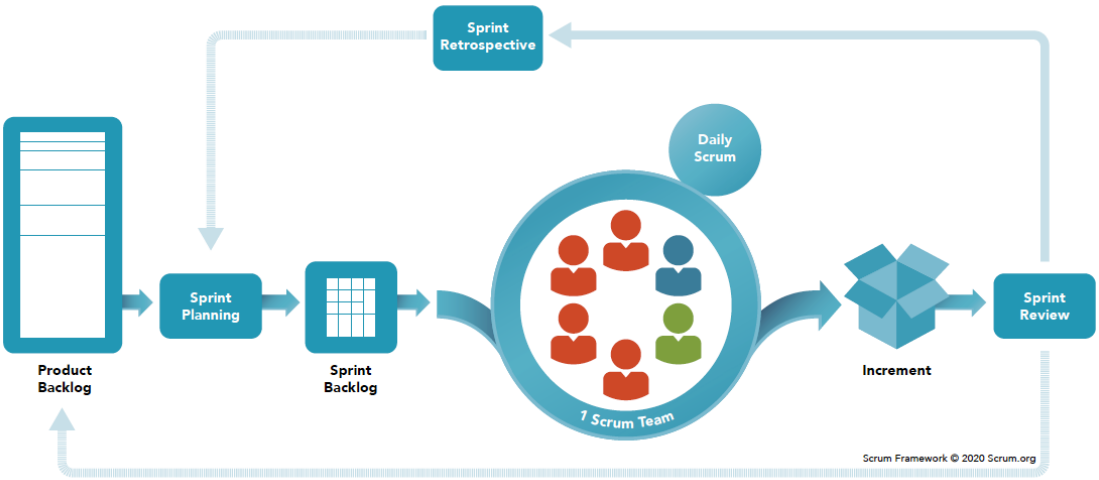
2. El Scrum Team convierte una selección del trabajo en un Increment de valor durante un Sprint.
3. El Scrum Team y sus interesados inspeccionan los resultados y se adaptan para el próximo Sprint.
4. Repita.

La unidad fundamental de Scrum es un pequeño equipo de personas, un Scrum Team. El Scrum Team consta de un Scrum Master, un Product Owner y Developers. Dentro de un Scrum Team, no hay subequipos ni jerarquías. Es una unidad cohesionada de profesionales enfocados en un objetivo a la vez, el Objetivo del Producto.

Scrum emplea un enfoque iterativo e Incremental para optimizar la previsibilidad y controlar el riesgo. Scrum involucra a grupos de personas que colectivamente tienen todas las habilidades y experiencia para hacer el trabajo y compartir o adquirir dichas habilidades según sea necesario.

Figura 1

Metodología Scrum (Scrum.org, 2023)



Nota: Esta figura muestra la estructura de la metodología scrum.

Como se observa en la Fig. 1, Scrum combina cuatro eventos formales para inspeccionar y adaptar dentro de un evento contenedor, el Sprint. Estos eventos

funcionan porque implementan los pilares empíricos de Scrum de transparencia, inspección y adaptación.

5.1.1. Eventos de Scrum

5.1.1.1. El Sprint

La información a continuación proviene de (Schwaber & Sutherland, 2020):

El Sprint es un contenedor para todos los demás eventos. Cada evento en Scrum es una oportunidad formal para inspeccionar y adaptar los artefactos Scrum. Estos eventos están diseñados específicamente para habilitar la transparencia requerida. No operar cualquier evento según lo prescrito resulta en la pérdida de oportunidades para inspeccionar y adaptarse. Los eventos se utilizan en Scrum para crear regularidad y minimizar la necesidad de reuniones no definidas en Scrum. Lo óptimo es que todos los eventos se celebren al mismo tiempo y en el mismo lugar para reducir la complejidad. Los Sprints son el corazón de Scrum, donde las ideas se convierten en valor.

Son eventos de duración fija de un mes o menos para crear consistencia. Un nuevo Sprint comienza inmediatamente después de la conclusión del Sprint anterior.

Todo el trabajo necesario para lograr el Objetivo del Producto, incluido la Sprint Planning, Daily Scrums, Sprint Review y Sprint Retrospective, ocurre dentro de los Sprints.

Durante el Sprint:

1. No se realizan cambios que pongan en peligro el Objetivo del Sprint.
2. La calidad no disminuye.
3. El Product Backlog se refina según sea necesario.
4. El alcance se puede aclarar y renegociar con el Product Owner a medida que se aprende más.

Un Sprint podría cancelarse si el Objetivo del Sprint se vuelve obsoleto. Solo el Product Owner tiene la autoridad para cancelar el Sprint.

5.1.1.2. Sprint Planning

La información a continuación proviene de (Schwaber & Sutherland, 2020):

> La Sprint Planning inicia el Sprint al establecer el trabajo que se realizará para el Sprint. El Scrum Team crea este plan resultante mediante trabajo colaborativo.
> El Product Owner se asegura de que los asistentes estén preparados para discutir los elementos más importantes del Product Backlog y cómo se relacionan con el Objetivo del Producto. El Scrum Team también puede invitar a otras personas a asistir a la Sprint Planning para brindar asesoramiento.
> La Sprint Planning aborda los siguientes temas:
> 1. Tema uno: ¿Por qué es valioso este Sprint?
> 2. Tema dos: ¿Qué se puede hacer en este Sprint?
> 3. Tema tres: ¿Cómo se realizará el trabajo elegido?

El Objetivo del Sprint, los elementos del Product Backlog seleccionados para el Sprint, más el plan para entregarlos se denominan juntos Sprint Backlog. La Sprint Planning tiene un límite de tiempo de máximo ocho horas para un Sprint de un mes. Para Sprints más cortos, el evento suele ser de menor duración.

5.1.1.3. Daily Scrum

La información a continuación proviene de (Schwaber & Sutherland, 2020):

El propósito de la Daily Scrum es inspeccionar el progreso hacia el Objetivo del Sprint y adaptar el Sprint Backlog según sea necesario, ajustando el trabajo planificado entrante.

La Daily Scrum es un evento de 15 minutos para los Developers del Scrum Team. Para reducir la complejidad, se lleva a cabo a la misma hora y en el mismo lugar todos los días hábiles del Sprint. Si el Product Owner o Scrum Master están trabajando activamente en elementos del Sprint Backlog, participan como

Developers. Las Daily Scrums mejoran la comunicación, identifican impedimentos, promueven la toma rápida de decisiones y, en consecuencia, eliminan la necesidad de otras reuniones.

La Daily Scrum no es el único momento en el que los Developers pueden ajustar su plan. A menudo se reúnen durante el día para discusiones más detalladas sobre cómo adaptar o volver a planificar el resto del trabajo del Sprint.

5.1.1.4. Sprint Review

La información a continuación proviene de (Schwaber & Sutherland, 2020):

El propósito de la Sprint Review es inspeccionar el resultado del Sprint y determinar futuras adaptaciones. El Scrum Team presenta los resultados de su trabajo a los interesados clave y se discute el progreso hacia el Objetivo del Producto.

Durante el evento, el Scrum Team y los interesados revisan lo que se logró en el Sprint y lo que ha cambiado en su entorno. Con base en esta información, los asistentes colaboran sobre qué hacer a continuación. El Product Backlog también se puede ajustar para satisfacer nuevas oportunidades. La Sprint Review es una sesión de trabajo y el Scrum Team debe evitar limitarla a una presentación.

La Sprint Review es el penúltimo evento del Sprint y tiene un límite de tiempo de máximo cuatro horas para un Sprint de un mes. Para Sprints más cortos, el evento suele ser de menor duración.

5.1.1.5. Sprint Retrospective

La información a continuación proviene de (Schwaber & Sutherland, 2020):

El propósito de la Sprint Retrospective es planificar formas de aumentar la calidad y la efectividad. El Scrum Team inspecciona cómo fue el último Sprint con respecto a las personas, las interacciones, los procesos, las herramientas y su definición de terminado. Los elementos inspeccionados suelen variar según el ámbito del trabajo. Se identifican los supuestos que los llevaron por mal camino y

se exploran sus orígenes. El Scrum Team analiza qué salió bien durante el Sprint, qué problemas encontró y cómo se resolvieron (o no) esos problemas.

El Scrum Team identifica los cambios más útiles para mejorar su efectividad. Las mejoras más impactantes se abordan lo antes posible. Incluso se pueden agregar al Sprint Backlog para el próximo Sprint. La Sprint Retrospective concluye el Sprint. Tiene un tiempo limitado a máximo tres horas para un Sprint de un mes. Para Sprint más cortos, el evento suele ser de menor duración.

5.1.2. Artefactos de Scrum

5.1.2.1. Product backlog

La información a continuación proviene de (Schwaber & Sutherland, 2020):

El Product Backlog es una lista emergente y ordenada de lo que se necesita para mejorar el producto. Es la única fuente del trabajo realizado por el Scrum Team.

Los elementos del Product Backlog que el Scrum Team puede dar por terminados dentro de un Sprint se consideran preparados para ser seleccionados en un evento de Sprint Planning. Suelen adquirir este grado de transparencia tras las actividades de refinamiento. El refinamiento del Product Backlog es el acto de dividir y definir aún más los elementos del Product Backlog en elementos más pequeños y precisos. Esta es una actividad continua para agregar detalles, como una descripción, orden y tamaño. Los atributos suelen variar según el ámbito del trabajo.

5.1.2.2. Sprint backlog

La información a continuación proviene de (Schwaber & Sutherland, 2020):

El Sprint Backlog se compone del Objetivo del Sprint (por qué), el conjunto de elementos del Product Backlog seleccionados para el Sprint (qué), así como un plan de acción para entregar el Increment (cómo).

El Sprint Backlog es un plan realizado por y para los Developers. Es una imagen muy visible y en tiempo real del trabajo que los Developers planean realizar durante el Sprint para lograr el Objetivo del Sprint.

En consecuencia, el Sprint Backlog se actualiza a lo largo del Sprint a medida que se aprende más. Debe tener suficientes detalles para que puedan inspeccionar su progreso en la Daily Scrum.

5.1.2.3. Increment

La información a continuación proviene de (Schwaber & Sutherland, 2020):

Un Increment es un peldaño concreto hacia el Objetivo del Producto. Cada Increment se suma a todos los Increments anteriores y se verifica minuciosamente, lo que garantiza que todos los Increments funcionen juntos. Para proporcionar valor, el Increment debe ser utilizable.

Se pueden crear múltiples Increments dentro de un Sprint. La suma de los Increments se presenta en la Sprint Review apoyando así el empirismo. Sin embargo, se puede entregar un Increment a los interesados antes del final del Sprint.

5.2. Herramientas de desarrollo a utilizar en este proyecto

A continuación, se presenta un resumen de las herramientas a utilizar para el diseño y desarrollo del SIGFA son:

Tabla 1: Herramientas de desarrollo y trabajo para el desarrollo del Sistema SIGFA

Herramienta	Descripción	Versión	¿Requiere licencia?
C#	Es un lenguaje de programación moderno, orientado a objetos y con seguridad de tipos. C# se ejecutan en .NET. (earn-microsoft-com., 2023)	10.0	No

Herramienta	Descripción	Versión	¿Requiere licencia?
Entity Framework	Entity Framework es un conjunto de tecnologías en ADO.NET que respaldan el desarrollo de aplicaciones de software orientadas a datos. (learn-microsoft-com, 2019)	6.0	No
Microservicios	Los microservicios son un enfoque arquitectónico y organizativo para el desarrollo de software donde está compuesto por pequeños servicios independientes que se comunican a través de API bien definidas. (aws.amazon.com)		No
Microsoft Visual Studio 2022	IDE de Visual Studio es una plataforma de lanzamiento creativa que puede utilizar para editar, depurar y compilar código y, finalmente, publicar una aplicación. (visualstudio.microsoft.com)	2022	No
Angular	Angular es una plataforma de desarrollo, construida sobre TypeScript. Es un Framework basado en componentes para crear aplicaciones web escalables. (Concavales, 2021)	15.1.1	No
Node.js	Es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript.	18.13	No
Npm	Es el sistema de gestión de paquetes por defecto para Node.js, un entorno de ejecución para JavaScript, bajo Artistic License 2.0. (docs.npmjs.com/)	9.2.0	No

Herramienta	Descripción	Versión	¿Requiere licencia?
Visual Studio Code	Es un editor de código fuente desarrollado por Microsoft. Es software libre y multiplataforma. (Flores, 2022)	2022	No
SQL Server 2022	Es un entorno integrado para administrar cualquier infraestructura de SQL. (learn-microsoft-com)	18.12	No
FastReport	Es una herramienta que permite incorporar a las aplicaciones una plataforma completa de informes. (www.danysoft.com/fast-report)		Si
Microsoft Project	es un software de administración y programas de proyectos para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo. (www.microsoft.com)	2022	Si
GitLab	es una compañía de núcleo abierto y es la principal proveedora del software GitLab, un servicio web de forja, control de versiones y DevOps basado en Git. (gitlab.com)	2022	No

Nota: Esta tabla muestra las herramientas de desarrollo y trabajo para la planificación, construcción, supervisión, pruebas y despliegue el Sistema SIGFA.

5.3. Prácticas de seguridad a implementar en el proyecto basadas en OWASP

Open Web Application Security Project (Proyecto de seguridad de aplicaciones web abiertas), promueve el desarrollo de software seguro, orientado a la prestación de servicios orientados a la web. (Fernandez, 2010)

Se centra principalmente en el back-end más que en cuestiones de diseño web, es un foro abierto para el debate, además de ser un recurso gratuito para cualquier equipo de desarrollo de software. (Fernandez, 2010)

Para asegurar el desarrollo de una aplicación web segura se utilizará el Top Ten de OWASP versión 2021, la cual lista una serie de recomendaciones para cubrir todos los posibles fallos de seguridad que pueda tener una aplicación web, como el SIGFA. A continuación, se lista los puntos que se utilizarán de la metodología OWASP: (OWASP, 2021)

Tabla 2: Top ten de la metodología OWASP 2021

Problemas	Soluciones
1. A01:2021 – Pérdida de Control de Acceso	¿Qué es? El control de acceso implementa el cumplimiento de política de modo que los usuarios no pueden actuar fuera de los permisos que le fueron asignados. Medidas a tomar: 1. A excepción de los recursos públicos, denegar por defecto. 2. Implemente mecanismos de control de acceso una única vez y reutilícelos en toda la aplicación. 3. El control de acceso debe implementar su cumplimiento a nivel de dato. 4. Deshabilite el listado de directorios del servidor web y asegúrese de que los archivos de metadatos. 5. Registre las fallas de control de acceso (loggin).
2. A02:2021 – Fallas Criptográficas	¿Qué es? Lo primero es determinar las necesidades de protección de los datos en tránsito y en reposo. Medidas a tomar:

Problemas	Soluciones
	1. Clasifique los datos procesados, almacenados o transmitidos por una aplicación. 2. No almacene datos sensibles innecesariamente. 3. Asegúrese de cifrar todos los datos sensibles en reposo (almacenamiento). 4. Garantice la implementación de algoritmos, protocolos y claves que utilicen estándares sólidos y actualizados. 5. Cifre todos los datos en tránsito con protocolos seguros como SSL con cifradores de confidencialidad adelantada.
3. A03:2021 – Inyección	¿Qué es? Algunas de las inyecciones más comunes son SQL, NoSQL, comandos de sistema operativo, Object-Relational Mapping (ORM), LDAP, expresiones de lenguaje u Object Graph Navigation Library (OGNL). Medidas a tomar: 1. La opción preferida es utilizar una API segura. 2. vite el uso de un intérprete por completo y proporcione una interfaz parametrizada o utilizar una herramienta de ORM. 3. Implemente validaciones de entradas de datos en el servidor, utilizando "listas blancas". 4. Utilice LIMIT y otros controles SQL dentro de las consultas para evitar la fuga masiva de registros en caso de inyección SQL.
4. A04:2021 – Diseño Inseguro	¿Qué es? El diseño inseguro es una categoría amplia que representa diferentes debilidades, expresadas como "diseño de control faltante o ineficaz". Medidas a tomar: 1. Limitar el consumo de recursos por usuario o servicio. 2. Separe a los tenants de manera robusta por diseño en todos los niveles. 3. Separe las capas del sistema y las capas de red según las necesidades de exposición y protección. 4. Integre verificaciones de viabilidad en cada capa de su aplicación (desde el front-end al Backend). 5. Integre el lenguaje y los controles de seguridad en las historias de usuario.
5. A05:2021 – Configuración de	¿Qué es? El 90% de las aplicaciones se probaron para detectar algún tipo de configuración incorrecta, con una

Problemas	Soluciones
Seguridad Incorrecta	tasa de incidencia promedio del 4,5% y más de 208.000 ocurrencias de CWEs en esta categoría de riesgo. Medidas a tomar: 1. Una plataforma mínima sin funciones, componentes, documentación ni ejemplos innecesarios. 2. Envío de directivas de seguridad a los clientes. 3. Un proceso automatizado para verificar la efectividad de las configuraciones y ajustes en todos los entornos.
6. A06:2021 – Componentes Vulnerables y Desactualizados	¿Qué es? Los componentes vulnerables son un problema conocido que es difícil de probar y evaluar el riesgo. Medidas a tomar: 1. Elimine las dependencias que no son utilizadas, funcionalidades, componentes, archivos y documentación innecesarios. 2. Realice un inventario continuo de las versiones de los componentes en el cliente y en el servidor. 3. Solo obtenga componentes de fuentes oficiales a través de enlaces seguros. 4. Supervise las bibliotecas y los componentes que no sea mantenidos o no generen parches de seguridad para versiones anteriores.
7. A07:2021 – Fallas de Identificación y Autenticación	¿Qué es? La confirmación de la identidad, la autenticación y la gestión de sesiones del usuario son fundamentales para protegerse contra ataques relacionados con la autenticación. Medidas a tomar: 1. No incluya o implemente en su software credenciales por defecto, particularmente para usuarios administradores. 2. Implementar un control contra contraseñas débiles. 3. Limite o incremente el tiempo de espera entre intentos fallidos de inicio de sesión. 4. Utilice un gestor de sesión en el servidor, integrado, seguro y que genere un nuevo ID de sesión aleatorio con alta entropía después de iniciar sesión.
8. A08:2021 – Fallas en el Software y	¿Qué es? Los fallos de integridad del software y de los datos están relacionados con código e infraestructura no protegidos contra alteraciones (integridad).

Problemas	Soluciones
en la Integridad de los Datos	Medidas a tomar: 1. Utilice firmas digitales o mecanismos similares para verificar que el software o datos provienen efectivamente de la fuente esperada y no fueron alterados. 2. Asegúrese que las bibliotecas y dependencias, tales como npm o maven son utilizadas desde repositorios confiables. 3. Asegúrese que se utilice un proceso de revisión de cambios de código y configuraciones para minimizar las posibilidades de que código. 4. Asegúrese que su pipeline CI/CD posee adecuados controles de acceso, segregación y configuraciones que permitan asegurar la integridad del código a través del proceso de build y despliegue.
9. A09:2021 – Fallas en el Registro y Monitoreo	¿Qué es? La intención es apoyar la detección, escalamiento y respuesta ante brechas activas. Sin registros y monitoreo, las brechas no pueden ser detectadas. Medidas a tomar: 1. Asegúrese de que los registros se generen en un formato fácil de procesar por las herramientas de gestión de registros. 2. Asegúrese de que los datos de registros son correctamente codificados para prevenir inyecciones o ataques en el sistema de monitoreo o registros. 3. Asegúrese de que las transacciones de alto valor poseen una traza de auditoria con controles de integridad para evitar la modificación o el borrado, tales como permitir únicamente la inserción en las tablas de base de datos o similares. 4. Los equipos de DevSecOps debe establecer alertas y monitoreo efectivo tal que se detecte actividades sospechosas y responderlas rápidamente.
10. A10:2021 – Falsificación de Solicitudes del Lado del Servidor (SSRF)	¿Qué es? Las fallas de SSRF ocurren cuando una aplicación web está obteniendo un recurso remoto sin validar la URL proporcionada por el usuario Medidas a tomar:

Problemas	Soluciones
	1. Sanítese y valide todos los datos de entrada proporcionados por el cliente 2. Haga cumplir el esquema de URL, el puerto y destino a través de una lista positiva de items permitidos. 3. No envíe respuestas en formato "crudo" a los clientes. 4. Deshabilite las redirecciones HTTP. 5. Tenga en cuenta la coherencia de la URL para evitar ataques como el enlace de DNS y las condiciones de carrera de "tiempo de verificación, tiempo de uso" (TOCTOU por sus siglas en inglés).

Nota: Esta tabla lista los problemas de seguridad de los sistemas web, centrándose en los diez más importantes. Llamándose Top Ten de la metodología OWASP, la cual se actualiza cada cierto tiempo para evaluar las últimas tendencias de ciberseguridad. (OWASP, 2021)

VI. Análisis y Presentación de Resultados

En esta sección se incluye el diseño metodológico contemplado para la ejecución de este trabajo, y seguidamente se presentan los resultados de la ejecución del mismo.

6.1. Diseño Metodológico

Se propone la utilización de la metodología de desarrollo Scrum como la base para organizar y llevar a cabo el trabajo de manera efectiva y colaborativa.

6.1.1. Scrum Team

El equipo de trabajo está conformado por los roles principales que se muestran a continuación:

Tabla 3: Miembros del Scrum Team SIGFA

Scrum Team	
Product Owner[3]	Presidente de Comisión de Proyecto SIGFA, y director de Normas y Procedimientos de la UNI (Isuhi Liberta Gutiérrez Ubau, Omar Quintana y Pablo David Trejos)
Scrum master [4]	Responsable de seguimiento de proyectos informáticos (Ing. Rodrigo Diaz)
Developer team	Desarrolladores (Ronnie Acevedo, Alfredo Eva, entre otros)

Nota: Esta tabla muestra los roles del Scrum Team y los respectivos miembros de cada rol.

Desarrolladores Frontend y Backend: Los autores de este trabajo (Alfredo Eva y Ronnie Acevedo) tienen como rol principal, en el equipo, el de desarrollador

[3] Product Owner: El Product Owner es el vínculo con el cliente y usuarios. Se asegura de que se comprendan las necesidades y expectativas, y trabaja para priorizar y refinar el Product Backlog. Su orientación guiará al equipo hacia la creación de funcionalidades significativas para el producto.

[4] Scrum Master: Es el líder que facilita el proceso Scrum, elimina obstáculos y fomenta un ambiente de colaboración y aprendizaje.

Frontend, pero también tendrán la oportunidad de colaborar en aspectos del desarrollo Backend cuando sea necesario. Alfredo y Ronnie, trabajan con otros desarrolladores Backend (Gabriel García y Henry Herrera). La combinación de habilidades del equipo permite abordar todas las capas del desarrollo del SIGFA. Se trabajará estrechamente en la creación de interfaces de usuario atractivas y funcionales, así como en la implementación y optimización de la lógica de negocio en el Backend.

El Product Owner se encargará de generar el Product backlog recibiendo insumos de las partes interesadas de este software, que son el área de finanzas, tesorería y contabilidad de la UNI, todo con ayuda del Scrum Master. Esto se documentará tanto físicamente (papel) como en digital para una mayor seguridad y resguardo.

6.1.2. Product Backlog y Sprint Planning

El Product backlog será organizado según la prioridad en el desarrollo de la programación en el sistema web. Por último, se llevará un registro y seguimiento de estos hasta que se hayan concluido de forma exitosa.

Teniendo listo el **Product backlog** (ya ordenados en importancia y complejidad, así como en digital y puestos en el Microsoft Project con su tiempo de duración, con los recursos que tendrá), se procederá a planear el **Sprint Planning** que no es más que la selección de forma colaborativa de las tareas a trabajar en un sprint.

6.1.3. Sprints

Los **sprint** tendrán una duración fija de tres semanas (21 días). En cada **Sprint**, se colaborará en la planificación y estimación de las **historias de usuario** seleccionadas para el **Sprint Backlog**. El **Product Owner** proporcionará claridad sobre las expectativas del cliente; y los desarrolladores Frontend trabajarán para que las interfaces de usuario sean atractivas, accesibles y responsivas.

6.1.3.1. Historias de Usuario y Sprint Backlog

Las historias de usuario serán tomadas directamente del cliente final, que en este caso son los usuarios de finanzas, contabilidad y tesorería, a través de reuniones

y entrevistas con los usuarios. La responsabilidad de registrar las historias de usuario recaerá principalmente en el Product Owner y en colaboración con el equipo de desarrollo.

Estas historias de usuario primeramente serán apuntadas en papel siguiendo la estructura de Scrum, (*Como [rol], quiero [acción] para [beneficio]*), luego de ser apuntadas en papel se procederá a registrarlas en digital.

Posteriormente, se procederá a ordenar estas historias según la importancia para el sistema y la dificultad para desarrollar dicha historia, según estos parámetros (importancia y dificultad), se calculará un tiempo estimado para culminar dicha tarea (**Sprint Backlog**).

6.1.3.2. Reuniones de trabajo Scrum

Cada mañana a primera hora se realizará un **Daily Scrum** que no es más que una reunión de diez a quince minutos donde el team de desarrollo conversara respecto a cómo va la tarea que tengan asignada, si se tiene problemas, si ya se terminó la tarea, si se va a necesitar más tiempo o ayuda.

Al finalizar un sprint está el **Sprint Review** que son las reuniones donde se presentarán los avances del proyecto, y se recibirán consejos, mejoras y retroalimentación de los responsables de cada área (Finanzas, Contabilidad y Tesorería). Esta Revisión de Sprint se realizará también con el Product Owner para demostrar funcionalidades completadas y recibir retroalimentación.

Por último, se realizará el **Sprint Retrospective (reunión de retrospectiva)** que es otra reunión, pero esta es al final de cada Sprint, ya cuando las tareas que se le asignaron a cada desarrollador se terminaron con éxito, se hacen reflexiones si hubo discusiones, malas prácticas de desarrollo, cualquier actitud negativa, con el fin de mejorar como equipo y ser mejores para el desarrollo del siguiente Sprint.

6.1.3.3. Construcción del Software

El desarrollo del software se llevará a cabo utilizando las herramientas reportadas en la sección 5.5. Específicamente para la construcción del software el equipo de

desarrollo Frontend utilizará Visual Studio Code, Angular, Node.JS, GIT y GitLab; por otro lado, el equipo de desarrollo Backend utilizará Microsoft Visual Studio, Entity Framework, C#, GIT y GitLab.

Conforme el equipo de desarrollo valla ejecutando dichas tareas se irán haciendo las pruebas necesarias para asegurar el correcto funcionamiento del software.

6.1.3.3.1. Control de versiones de Código con GitLab

Para el control de versiones se utilizará GitLab ya que es una plataforma en la nube que proporciona una variedad de herramientas para supervisar, probar y desplegar el código además de poder guardar repositorios completos. Cada desarrollador (Alfredo Eva, Ronnie Acevedo, Gabriel García y Henry Herrera) tendrá una rama en el proyecto que se titulará “apisifa” y estará alojado en GitLab.

Cada desarrollador tendrá su propia rama donde trabajará individualmente en las tareas que se le asignen. Luego de hacer las pruebas necesarias, ya sea una tarea, código, módulo, reporte, etc. y ser aprobadas por Alfredo Eva que en este caso es el responsable de la rama principal (dev), se subirán a dicha rama.

Por último, todo este proceso se volverá a repetir para el siguiente sprint.

6.1.3.4. Despliegue de versión completa del sistema SIGFA

Una vez se cuente con la versión completa del Sistema SIGFA, se llevará a cabo el despliegue en un entorno de pruebas. En esta fase, los usuarios finales tendrán la oportunidad de poner a prueba el sistema con el propósito de identificar posibles fallos. En caso de que se detecten errores, se tomarán medidas inmediatas para corregirlos y garantizar un funcionamiento óptimo.

La responsabilidad de gestionar la publicación de los sistemas en la Dirección de Tecnología de Información (DTI) recae en la Ingeniera Nubia Kenia Altamirano, quien apoyaría a los miembros del equipo de trabajo a ejecutar esta tarea.

6.1.3.5. Pruebas de seguridad OWASP

Igualmente, se llevarán a cabo exhaustivas pruebas de seguridad en ambos niveles, el Backend y el Frontend, con el propósito de identificar y eliminar posibles vulnerabilidades o fallos de seguridad; esto como complemento a las pruebas funcionales realizadas durante el desarrollo de cada módulo. Estas pruebas se realizarán siguiendo las pautas establecidas en el top ten de OWASP, que enumera las diez principales vulnerabilidades de seguridad en aplicaciones web.

6.1.4. Gestión del proyecto con Microsoft Project

Para la administración, planificación y distribución de tareas se utilizará la herramienta Microsoft Project. Microsoft Project es utilizado ya que es la herramienta que se utiliza en la UNI por defecto para la elaboración de las gestiones de proyectos; esto a solicitud de los interesados, ya que están familiarizados con dicha herramienta.

6.2. Resultados

En esta sección de la monografía se abordan los resultados obtenidos en el desarrollo, pruebas y despliegue del sistema financiero SIGFA. Debido a limitaciones de espacio, se seleccionó un subconjunto de los Sprints ejecutados para presentarse en esta sección, y los restantes Sprints se documentan en Anexos. También se describen aspectos importantes del proceso por el cual paso dicho sistema, incluyendo:

1. Reuniones iniciales de planificación con el Product Owner para el levantamiento del Product Backlog con sus respectivas historias de usuario.
2. Ejecución y documentación de las reuniones (Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective).
3. La implementación de prácticas de seguridad y pruebas de seguridad que se utilizó para la debida protección de datos.
4. El control de versiones de código que se utilizó, para el resguardo y respaldo del código fuente del sistema.

Todos estos puntos se abordan en esta sección a más detalle.

Como se menciona anteriormente este proyecto del desarrollo del sistema SIGFA surge como una actualización del anterior sistema financiero de la UNI llamado SIFA. Con la aprobación de la Dirección de Normas y Procedimientos (DNP) de la UNI, el Ing. Omar Quintana (Product Owner en ese momento) inició con las reuniones y levantamiento de la información principal utilizada para la documentación de las historias de usuario.

En las primeras reuniones en febrero 2023 se comenzó con las propuestas, las explicaciones del modelo de negocio del SIFA, los procesos que ejecutaba, las funciones que se pueden realizar.

Los encargados tanto de la supervisión y desarrollo del sistema SIGFA se incluyen en la Tabla 3 (cabe mencionar que, debido a los cambios estructurales de la UNI, algunos miembros del Equipo de Desarrollo, así como el Product Owner y el Scrum Master cambiaron al transcurrir el tiempo):

Tabla 4: Miembros del Equipo de trabajo

Product Owner:	**MSc. Pablo David Trejos (Director de la División de Finanzas)**
Scrum Master:	Ing. Rodrigo Diaz (responsable de Oficina de Sistema)
Equipo de Desarrollo:	Ing. Jorge Abrahán Torres; Ing. Yasser Montiel; Ing. Carlos Martínez; Ing. Gabriel García; **Alfredo Antonio Eva Vargas, Ronnie Aaron Acevedo Cisnero.**

Debido a cambios estructurales en la Universidad Nacional de Ingeniería (UNI), algunos miembros del equipo Scrum fueron reemplazados a lo largo del tiempo. Como parte de estos cambios, el Product Owner, en este caso el Director de Finanzas, fue reemplazado, y el rol de Scrum Master también experimentó modificaciones, siendo cambiado en tres ocasiones, lo cual generó el mayor impacto en el equipo.

Tabla 5**:** Responsables quienes aprobaron los Sprint

Responsable	Cargo	Área	Rol	Sprint
Lic. Manuel Olivares Solís	Director de Finanzas	División de Finanzas	Product Owner	1-8
MSc. Pablo David Trejos Aguilar	Director de Finanzas	División de Finanzas	Product Owner	9-11
Lic. Gustavo Adolfo Silva Bermúdez	Responsable de la Oficina de Contabilidad	División de Finanzas	Stakeholders	1-11
Ing. Rodrigo Diaz	Responsable de Oficina de Sistema	División de Tecnología de la Información	Scrum Master	1-11

En estas primeras reuniones se obtuvo mejor claridad del modelo de negocio asociado a estos procesos, el cual se describe muy brevemente a continuación:

6.2.1. Procesos administrativos financieros a cubrir en este sistema

Esta sección aborda conceptos fundamentales asociados a los procesos a cubrir en este sistema, y luego cómo estos son realizados en la UNI.

6.2.1.1. Conceptos fundamentales sobre actividades, procesos y entidades abordadas en este sistema:

El Sistema Web desarrollado se enfoca en facilitar las gestiones administrativas y financieras de la UNI, consecuentemente se enfoca en apoyar las funciones de tres unidades organizacionales que colaboran estrechamente y trabajan en conjunto para asegurar la salud financiera de la organización, en este caso la UNI, estas son: Finanzas, Tesorería y Contabilidad. Particularmente, el sistema web desarrollado se enfoca en atender con mayor énfasis procesos de Contabilidad que están relacionados a las Unidades Organizacionales (UO) mencionadas.

La contabilidad registra, clasifica, analiza e interpreta todas las transacciones económicas de una empresa. Es como un historial financiero detallado que permite conocer la situación económica de un negocio en un momento dado. Se puede imaginar la contabilidad como un diario financiero de la universidad; en este diario se registran todos los ingresos (dinero que entra) y gastos (dinero que sale) de manera ordenada.

La contabilidad se basa en el método de la partida doble, lo que significa que cada transacción afecta al menos dos cuentas contables. Por ejemplo, si una empresa compra una computadora, se registra un aumento en el activo (la computadora) y una disminución en el efectivo o en otra cuenta de pasivo si se financia a crédito.

La contabilidad proporciona la información básica que las finanzas y la tesorería utilizan para tomar decisiones y gestionar los recursos. A continuación, se menciona brevemente lo que Finanzas, Tesorería y Contabilidad abordan respecto al ciclo contable:

- **Contabilidad:** Es el corazón del ciclo contable. Se encarga de:
    - **Identificar y medir:** Reconoce las transacciones económicas que afectan a la empresa, en este caso la UNI, y les asigna un valor monetario.
    - **Registrar:** Anota todas las transacciones en los libros contables (diario, mayor).
    - **Clasificar:** Agrupa las transacciones por tipo (ingresos, gastos, activos, etc.) para facilitar el análisis.
    - **Resumir:** Prepara los estados financieros (balance general, estado de resultados) que resumen la situación financiera de la empresa.
    - **Analizar:** Interpreta los datos contables para identificar tendencias y tomar decisiones.
- **Finanzas:** Utiliza la información contable para:
    - **Planificar:** Establece objetivos financieros y elabora presupuestos.

    - **Decidir:** Toma decisiones de inversión, financiamiento y distribución de dividendos.
    - **Controlar:** Evalúa el desempeño financiero y toma medidas correctivas si es necesario.
- **Tesorería:** Se enfoca en la gestión del efectivo y está estrechamente relacionada con el ciclo contable, ya que:
    - **Recibe:** Registra los ingresos de efectivo (ventas, cobros).
    - **Paga:** Registra los egresos de efectivo (compras, pagos a proveedores).
    - **Concilia:** Compara los registros contables con los extractos bancarios.
    - **Pronostica:** Estima los flujos de efectivo futuros para asegurar la liquidez.

Para diferenciar un poco el ámbito de cada área se puede considerar el **enfoque temporal**, en cuyo caso:

- **Contabilidad:** Se enfoca en el pasado y presente, registrando las transacciones y generando estados financieros históricos.
- **Finanzas:** Se enfoca en el futuro, realizando proyecciones y tomando decisiones estratégicas basadas en la información contable.
- **Tesorería:** Se enfoca en el corto plazo, gestionando el flujo de efectivo diario y asegurando la liquidez.

La contabilidad es esencial para cualquier organización, incluyendo una universidad, ya que permite: saber cuánto dinero se tiene (por tanto, se sabe cuánto se tiene disponible para pagar gastos); tomar decisiones informadas (por ejemplo: invertir en nuevos proyectos, aumentar los salarios o reducir gastos); y cumplir con las leyes fiscales y otras regulaciones.

6.2.1.1.1. El Ciclo de la Contabilidad

El ciclo contable es un proceso continuo que involucra la identificación, medición, registro y comunicación de transacciones económicas. Es como un flujo de información que inicia con una operación y culmina con la generación de estados financieros. Parte de este proceso será administrado por el sistema de información desarrollado.

Para aclarar mejor el ciclo contable, se presenta un escenario a continuación: Al comprar un libro nuevo para la biblioteca; esto es una transacción financiera. El ciclo contable es el proceso que sigue esta transacción desde que ocurre hasta que se refleja en los estados financieros, y se vería como sigue:

1. **Registro:** Se anota la compra del libro en un registro contable, como si fuera una entrada en un diario.
2. **Clasificación:** La compra se clasifica como un gasto.
3. **Resumen:** Al final del mes, se suman todos los gastos y se compara con los ingresos para ver si la universidad tuvo ganancias o pérdidas.
4. **Informes:** Se generan informes, como el estado de resultados, que muestran los resultados financieros de la universidad.

6.2.1.1.2. Algunas actividades y conceptos relevantes alrededor del ciclo contable

Dentro de las actividades que realizan las áreas mencionadas, se resumen las siguientes:

- Las transacciones se registran en el **libro diario** y se respaldan con **comprobantes**.
- Los datos del libro diario se resumen en el **libro mayor**.
- Al finalizar el período, se realizan los ajustes contables y se elaboran los **estados financieros** basados en la información del **libro mayor**.

- Los **estados financieros** proporcionan una visión general de la salud financiera de la organización (en este caso la UNI) y son utilizados por diversos usuarios (administradores, tomadores de decisiones, etc.).

A continuación, se definen algunos de los términos mencionados previamente:

Libros Contables: Los libros contables son registros donde se anotan todas las transacciones económicas de una empresa de manera ordenada y cronológica. Son la base de la información contable y sirven como evidencia de las operaciones realizadas.

- **Libro diario:** Es el primer registro de las transacciones. En él se anotan todas las operaciones en orden cronológico, indicando la fecha, las cuentas afectadas y los importes.
- **Libro mayor:** Es un resumen de todas las cuentas contables. Cada cuenta tiene una página donde se registran todos los movimientos (débitos y créditos) relacionados con esa cuenta.

Cuentas contables: Una cuenta contable es un registro individualizado que permite llevar un control detallado de cada elemento del patrimonio de una empresa. Por ejemplo: Caja (registra el dinero en efectivo que tiene la empresa), Bancos (registra el dinero que la empresa tiene en el banco), Clientes (registra las facturas que los clientes aún no han pagado), etc.

Asiento contable: Es la anotación que se realiza en el libro diario de contabilidad para registrar una transacción económica. Proporcionan un registro preciso y detallado de cada transacción, incluyen: Fecha, Cuentas afectadas, debe y haber, montos.

Los asientos contables son las operaciones que se realizan en las cuentas contables. Cada vez que ocurre una transacción (comprar un producto, pagar un sueldo, etc.), se registra un asiento contable que afecta a una o más cuentas. Por

ejemplo: Si una empresa compra una computadora por $1,000 pagando en efectivo, se realizaría un asiento contable que aumentaría la cuenta "Maquinaria" en $1,000 y disminuiría la cuenta "Caja" en $1,000.

Comprobantes de Diario: Los comprobantes de diario son documentos que respaldan las transacciones registradas en el libro diario. Estos comprobantes pueden ser facturas, recibos, notas de crédito, etc. Sirven como evidencia de la ocurrencia de una transacción y facilitan la auditoría de los registros contables.

El comprobante de diario es la fuente de datos para elaborar el asiento contable. El asiento contable es el registro contable formal, mientras que el comprobante de diario es el documento que lo respalda.

Cierre Contable: El cierre contable es un proceso periódico, generalmente al final de cada mes, trimestre o año fiscal, donde se actualizan los registros contables para reflejar la situación financiera real de la empresa en ese momento, y es fundamental para mantener la información contable actualizada y precisa. Es como tomar una "foto" de la salud financiera de la empresa en un punto específico del tiempo. El cierre contable es importante porque ayuda a tener: Precisión de los estados financieros; cumplimiento normativo con las obligaciones fiscales y legales; información actualizada para la toma de decisiones gerenciales; preparación para el siguiente período (limpiar las cuentas y prepararlas para el nuevo período contable).

El cierre contable implica realizar un registro de ajustes (se realizan ajustes contables para reconocer ingresos no registrados, gastos devengados, depreciación, etc.); elaboración de estados financieros (se preparan el balance general, estado de resultados, estado de flujo de efectivo y otros estados financieros según sea necesario); cierre de cuentas temporales; apertura de cuentas para el siguiente período.

6.2.1.1.3. Principales resultados de la contabilidad:

Estados financieros: Son los documentos que resumen la situación financiera de una empresa en un período determinado. Los estados financieros son como "fotografías" de la situación financiera de la universidad en un momento dado. Los principales son:

- **Balance general:** Muestra lo que la universidad posee (activos), lo que debe (pasivos) y el valor de la universidad (patrimonio neto) en un momento determinado. Es como un balance bancario, pero para toda la universidad.
- **Estado de resultados:** Muestra los ingresos y gastos de la universidad durante un período determinado (por ejemplo, un mes o un año). Es como un resumen de ganancias y pérdidas.

Informes contables: Se elaboran informes personalizados para analizar aspectos específicos de la gestión financiera, como la rentabilidad, la liquidez o la solvencia.

6.2.1.2. Aspectos sobre el manejo contable de la Universidad Nacional de Ingeniería

La contabilidad de la Universidad Nacional de Ingeniería se rige por un modelo presupuestario clasificado según las fuentes de financiamiento (ingresos propios o aporte estatal). Estas fuentes son utilizadas por las unidades ejecutoras (áreas, proveedores, estudiantes y trabajadores) en las operaciones contables mensuales (compras, gastos, ventas, préstamos, aranceles y otros), correspondientes a un período general (año contable). Además, la contabilidad se organiza de acuerdo con un catálogo contable que agrupa este modelo presupuestario.

La base de la contabilización de estas operaciones reside en los asientos contables (comprobantes de diario y pago), que son responsables de distribuir el detalle contabilizado en los datos contables. Estos los conforman el detalle

general (Fuente de Financiamiento, Periodo Especifico, Concepto y Tipo de Asiento Contable) y las líneas contables (cuenta, unidad ejecutora y monto).

6.2.1.3. Procesos y Funciones de Finanzas, Contabilidad y Tesorería en la Universidad Nacional de Ingeniería:

El área de finanzas de la Universidad Nacional de Ingeniería (UNI) juega un rol fundamental en la administración, gestión, y optimización de los recursos financieros, con el objetivo de asegurar un funcionamiento eficiente y sostenible de la institución. El área de Contabilidad y el área de Tesorería están por debajo del área de Finanzas, es decir le rinden cuentas a esta última. A continuación, se detallan los procesos clave que realizan Finanzas, Contabilidad y Tesorería en la UNI (Solis, 2023; UNIVERSIDAD DE NEGOCIOS ISEC , 2022).

1. Finanzas: Presupuesto, Planificación y Gestión Financiera

Esta área se encarga de coordinar los recursos económicos disponibles para satisfacer las necesidades institucionales y apoyar los objetivos estratégicos de la universidad. Entre sus funciones principales se encuentran:

Presupuesto y Planificación Financiera:

1. Establecer un plan de ingresos y gastos alineado con las prioridades de la universidad.
2. Asignar recursos a diferentes departamentos, proyectos y programas de manera eficiente.
3. Definir metas financieras a corto, mediano y largo plazo, garantizando la sostenibilidad financiera.

Gestión de Tesorería:

1. Supervisar los flujos de efectivo para mantener una operación eficiente y asegurar liquidez.
2. Realizar pronósticos financieros, garantizando que se cuente con los fondos suficientes para cumplir compromisos operativos.
3. Implementar políticas de control para minimizar riesgos financieros.

Reportes Financieros:

1. Registrar todas las transacciones financieras y mantener un registro preciso y actualizado.
2. Generar informes financieros periódicos que respalden la toma de decisiones estratégicas.
3. Proveer información clara y precisa para auditorías y para la rendición de cuentas ante las autoridades pertinentes.

2. Tesorería: Control y Ejecución de Pagos

Tesorería se especializa en la administración de los pagos y movimientos de efectivo de la universidad. Su objetivo es garantizar que las transacciones financieras se realicen de manera fluida y eficiente. Sus responsabilidades incluyen:

Manejo de Pagos:

1. Coordinar y autorizar pagos a proveedores, personal, y estudiantes, asegurando puntualidad y precisión en los desembolsos.
2. Supervisar pagos electrónicos, transferencias y cheques.
3. Mantener una comunicación constante con las diferentes áreas para cumplir los cronogramas de pago.

Control de Flujos Financieros:

1. Gestionar las cuentas bancarias y saldos disponibles para cubrir las obligaciones diarias.
2. Asegurar el cumplimiento de normativas internas y externas en la ejecución de pagos.

3. Contabilidad: Registro y Control de la Información Financiera

El área de contabilidad es responsable de registrar, supervisar y analizar todas las transacciones financieras, asegurando la precisión y la integridad de la información contable. Las funciones más relevantes incluyen:

Registro de Transacciones:

1. Registrar de manera oportuna ingresos, gastos, activos, pasivos y patrimonio en los sistemas contables.

2. Garantizar que todas las transacciones cumplan con las normativas contables vigentes.

Libros Contables:

1. Mantener libros oficiales como el libro diario y el libro mayor, donde se registran todas las operaciones de manera organizada.
2. Facilitar el acceso a registros históricos para fines de análisis y auditoría.

Elaboración de Estados Financieros:

1. Preparar estados financieros clave, como el balance general, estado de resultados y estado de flujos de efectivo. Estos documentos ofrecen una visión integral del desempeño financiero de la universidad.

Auditorías Internas y Externas:

1. Coordinar las auditorías internas y externas, garantizando el acceso a la información solicitada por los auditores.
2. Implementar las recomendaciones derivadas de las auditorías para mejorar los controles internos.

Gestión de Cuentas por Pagar y Cobrar:

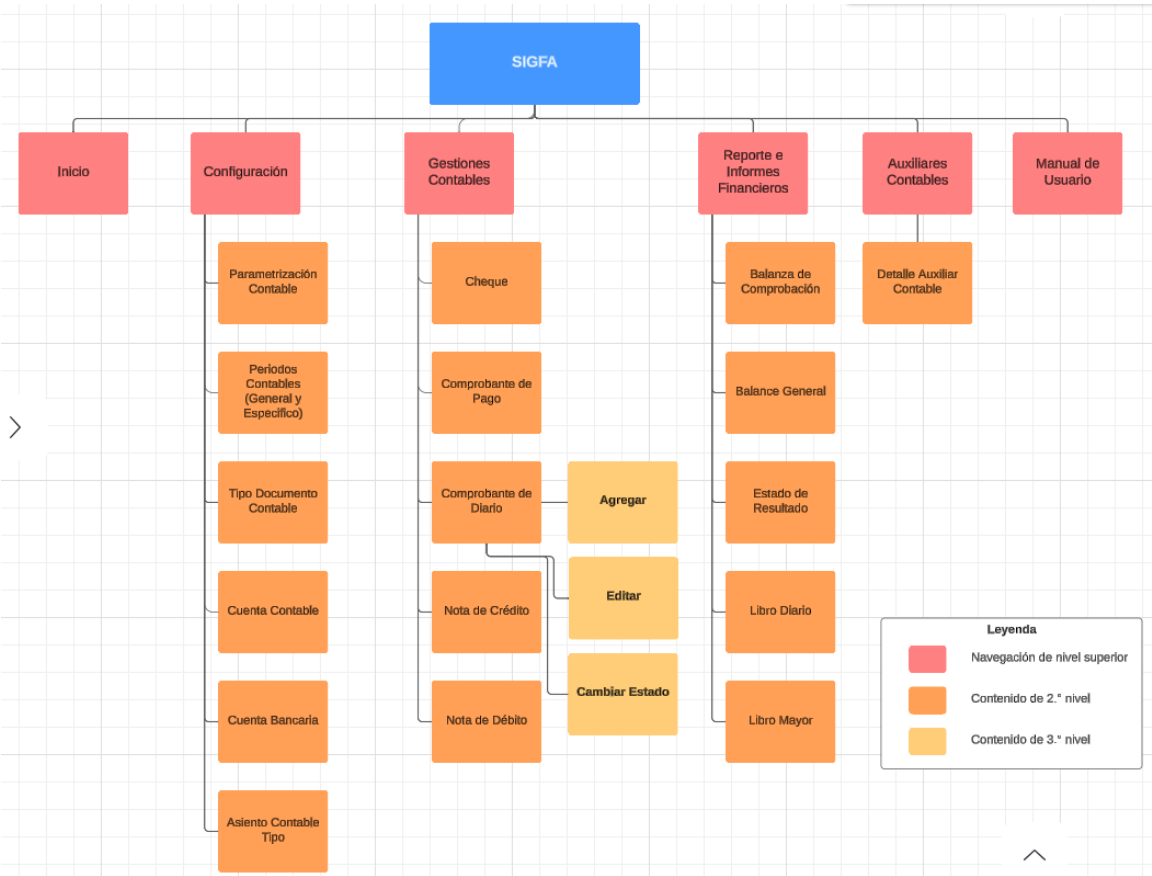
1. Administrar las cuentas por pagar a proveedores y empleados, asegurando que los pagos se registren correctamente y se realicen en tiempo y forma.
2. Controlar las cuentas por cobrar, supervisando los pagos pendientes de estudiantes y trabajadores.

6.2.2. Vinculación de los Módulos del Sistema SIGFA con los Procesos Administrativos a Cubrir

A continuación, se detalla cómo los módulos del sistema SIGFA se alinean y apoyan los procesos administrativos financieros descritos anteriormente. Esta vinculación es fundamental para asegurar que el nuevo sistema cubra de manera integral las necesidades de finanzas, contabilidad y tesorería en la UNI, y ayuda a dimensionar el alcance del sistema desarrollado. A continuación, se presenta un diagrama que resume la estructura de módulos del sistema, y seguidamente se

explica brevemente cada módulo y su vinculación con los procesos administrativos:

Figura 2
Alcance del Sistema SIGFA



Módulo de Inicio

Resumen Anual y Gráficos para Toma de Decisiones

1. **Función:** Este módulo proporciona una visión global del estado financiero de la UNI a lo largo del año, apoyando la toma de decisiones estratégicas mediante gráficos que reflejan el desempeño general de la institución, incluyendo el estado patrimonial y los resultados del periodo.
2. **Vinculación con Procesos Administrativos**:

2.1. **Presupuesto y Planificación Financiera**: Los gráficos del resumen anual permiten evaluar el cumplimiento de los presupuestos general de la UNI.

2.2. **Reportes Financieros**: El estado patrimonial (balance general específico) y el estado de resultados (resultados del período) ofrecen información para el análisis financiero y para medir el desempeño institucional.

Módulo de Configuraciones

Submódulos de Catálogo: Agregar Cuentas Contables, Bancarias, etc.

1. **Función**: Facilita la configuración del sistema mediante la creación y actualización de cuentas contables, bancaria, programas, asiento contable tipo y otros catálogos esenciales para la operación financiera.
2. **Vinculación con Procesos Administrativos**:

2.1. **Gestión de Tesorería**: Las cuentas bancarias definidas en este módulo son esenciales para la ejecución de pagos y manejo de fondos.

2.2. **Registro de Transacciones y Libros Contables (Contabilidad)**: La configuración de las cuentas contables permite que los registros se realicen de forma precisa y organizada, facilitando los procesos contables y la elaboración de informes financieros.

Módulo de Gestiones Contables

Submódulos: Cheque, Comprobante de Pago, Comprobante de Diario, Nota de Crédito

1. **Función**: Este módulo centraliza los procesos contables, permitiendo gestionar diferentes tipos de comprobantes y documentos necesarios para registrar las transacciones financieras.
2. **Vinculación con Procesos Administrativos**:

2.1. **Manejo de Pagos (Tesorería)**: Los cheques y comprobantes de pago aseguran que los desembolsos a proveedores, empleados y estudiantes se registren y ejecuten de manera oportuna.

2.2. **Registro de Transacciones (Contabilidad)**: Los comprobantes de diario y las notas de crédito garantizan que todas las operaciones financieras se registren en los libros contables.

2.3. **Control de Cuentas por Pagar y Cobrar (Contabilidad)**: Este módulo facilita la gestión de pagos pendientes y cobros, asegurando precisión y transparencia en las transacciones.

Módulo de Reportes e Informes Financieros

Informes: Balanza de Comprobación, Balance General, Estado de Resultados

1. **Función**: Provee reportes financieros detallados que ofrecen una visión integral de la situación financiera de la universidad, esenciales para la evaluación del desempeño institucional.
2. **Vinculación con Procesos Administrativos**:

2.1. **Elaboración de Estados Financieros (Finanzas)**: Este módulo genera informes como la **balanza de comprobación**, **balance general** y **estado de resultados**, fundamentales para medir el desempeño financiero de la UNI.

2.2. **Auditorías Internas y Externas (Finanzas)**: Los informes financieros son claves para las auditorías, permitiendo validar el cumplimiento normativo y las políticas internas.

2.3. **Reportes Financieros (Finanzas)**: Los reportes periódicos ayudan a tomar decisiones informadas y a rendir cuentas ante las autoridades.

6.2.3. Product Backlog

Con la información anterior se prosiguió a concretar lo referente a funcionalidades, para esto el Product Owner (Ing. Omar Lenin Quintana Blandón) indicó que el nuevo SIGFA debería incluir las funcionalidades y procesos principales del SIFA, y adicionalmente solicitó otras funcionalidades; todo esto se documenta en el Product Backlog a continuación con las respectivas escalas de esfuerzo (Tabla 6) y de prioridad (Tabla 7) que se aplicarán a las historias de usuario.

Tabla 6**:** Escala de esfuerzo

Numero	Color	Significado
1	Rojo	Grado de complejidad de la tarea a realizar, en este caso es alta.
2	Amarillo	Grado medio de complejidad de la tarea a realizar.
3	Verde	Tarea relativamente poco compleja.

Tabla 7**:** Escala de prioridad

Prioridad	Color	Significado
Alta	Rojo	Grado de prioridad de la tarea a realizar, en este caso es alta.
Media	Amarillo	Grado medio de prioridad de la tarea a realizar.
Baja	Verde	Tarea relativamente baja prioridad.

Tabla 8**:** Product Backlog basado en Historias de Usuario

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Esfuerzo	Prioridad
1	Contador	Generar periodos contables generales.	Tener estados financieros claros anuales.	1	Alta
2	Director	Editar los periodos generales.	Tener la opción de arreglar errores financieros.	2	Media
3	Director	Cambiar estado de registrado a activo en el Periodo General.	Tener control de los estados de los periodos.	2	Media
4	Director	Cambiar estado de activo a cerrado en Periodo General.	Tener control de los estados de los periodos.	2	Media
5	Contador	Agregar Periodos Específicos.	Tener estados financieros claros mensuales.	1	Alta

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Esfuerzo	Prioridad
6	Director	Cambiar estado de registrado a activo en los periodos específicos.	Tener la opción de activar o desactivar los estados.	2	Media
7	Director	Cambiar estado de activo a precierre en los periodos específicos.	Tener la opción de activar o desactivar los estados.	2	Media
8	Director	Cambiar estado de precierre ha cerrado en los periodos específicos.	Tener control de los estados de los estados.	2	Media
9	Contador	Buscar ce por rango de fecha.	Buscar de una manera más fácil periodos.	3	Baja
10	Contador	Agregar Cuenta Contable.	Añadir cuentas al sistema SIGFA.	1	Alta
11	Director	Editar Cuenta Contable.	Arreglarlas en caso de que haya cambios.	2	Media
12	Director	Generar árbol de cuentas contables.	Tener un control ordenado de las cuentas padres e hijas de la UNI.	1	Media
13	Director	Cambiar estado de activo a cerrado en Periodo General.	Tener control de los estados de los periodos.	2	Media
14	Contador	Anular cuenta contable.	Ponerlas en estado inhabilitadas en el sistema SIGFA.	3	Alta
15	Contador	Buscar Cuenta Contable por Cuentas	Encontrar información de esa cuenta con rapidez.	3	Baja

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Esfuerzo	Prioridad
		Contable y por nombre.			
16	Contador	Agregar Cuenta Bancaria.	Añadir cuentas bancarias al sistema SIGFA.	1	Alta
17	Director	Editar Cuenta Bancaria.	Arreglarlas en caso de que haya cambios.	2	Media
18	Director	Anular Cuenta Bancaria.	Ponerlas en estado inhabilitadas en el sistema SIGFA.	1	Media
19	Director	Buscar Cuenta Bancaria por Cuentas Contable y Banco.	Encontrar información de esa cuenta bancaria con rapidez.	2	Media
20	Contador	Agregar Asiento Contable Tipo.	Añadir Asientos Contables al sistema SIGFA.	1	Alta
21	Director	Editar Asiento Contable Tipo.	Arreglarlas en caso de que haya cambios.	2	Media
22	Director	Anular Asiento Contable Tipo.	Ponerlas en estado inhabilitadas en el sistema SIGFA.	1	Media
23	Director	Buscar Asiento Contable Tipo por Nombre.	Encontrar información de ese asiento contable con rapidez.	2	Media
24	Contador	Agregar Tipo Documento Contable.	Añadir Tipo Documentos Contables al sistema SIGFA.	1	Alta

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Esfuerzo	Prioridad
25	Director	Editar Tipo Documento Contable.	Arreglarlos en caso de que haya cambios.	2	Media
26	Director	Anular Tipo Documento Contable.	Ponerlas en estado inhabilitadas en el sistema SIGFA.	1	Media
27	Director	Buscar Tipo Documentos Contables por nombre y visualizar.	Encontrar información de ese Tipo Documento Contable con rapidez.	3	Baja
28	Director	Editar Cheque.	Cambiar datos del cheque en caso de que haya alteraciones o incidencias.	1	Alta
29	Director	Cambiar estados de Cheque.	Llevar la secuencia de la vida de cada cheque	1	Alta
30	Director	Mostrar los estados del cheque.	Tener control del estado de cada cheque.	3	Baja
31	Director y Contador	Anular Cheque.	Anularlo en caso de que haya una incidencia con el beneficiario.	1	Alta
32	Director y Contador	Generar vista previa del cheque.	Tener información completa de ese cheque.	2	Media
33	Director y Contador	Buscar Cheque por Beneficiario, Fecha de Creación y Área.	Tener información de un cheque de una manera más rápida.	3	Baja
34	Contador	Agregar Asiento Contable.	Añadir Asiento Contable al sistema SIGFA.	1	Alta

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Esfuerzo	Prioridad
35	Director	Editar Asiento Contable.	Cambiar datos del Asiento Contable. en caso de que haya alteraciones o incidencias.	1	Alta
36	Director	Cambiar estados de Asiento Contable.	Llevar la secuencia de la vida de Asiento Contable	1	Alta
37	Director	Mostrar los estados del Asiento Contable.	Tener control del estado de cada Asiento Contable.	3	Baja
38	Director y Contador	Anular Asiento Contable.	Anularlo en caso de que haya una incidencia con un dato o cifra en el Asiento Contable.	1	Alta
39	Director y Contador	Generar vista previa del Asiento Contable.	Tener información completa de ese Asiento Contable.	2	Media
40	Director y Contador	Buscar Asiento Contable por Beneficiario, Fecha de Creación y Área.	Tener información de un Asiento Contable de una manera más rápida.	3	Baja
41	Contador	Agregar Nota de Crédito.	Añadir Agregar Nota de Crédito al sistema SIGFA.	1	Alta
42	Director	Editar Nota de Crédito.	Cambiar datos de la Nota de Crédito en caso de que haya alteraciones o incidencias.	1	Alta
43	Director	Cambiar estados de Nota de Crédito.	Llevar la secuencia de la vida de una Nota de Crédito.	1	Alta

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Esfuerzo	Prioridad
44	Contador	Mostrar los estados de la Nota de Crédito.	Tener control de cada estado de cada Nota de Crédito.	3	Baja
45	Director y Contador	Anular Nota de Crédito.	Anularlo en caso de que haya una incidencia con un dato o cifra con la Nota de Crédito.	1	Alta
46	Director y Contador	Generar vista previa de la Nota de Crédito.	Tener información completa de esa Nota de Crédito.	2	Media
47	Director y Contador	Buscar una Nota de Crédito por Beneficiario, Fecha de Creación y Área.	Tener información de una Nota de Crédito de una manera más rápida.	3	Baja
48	Contador	Agregar Nota de Débito.	Añadir Agregar Nota de Débito al sistema SIGFA.	1	Alta
49	Director	Editar Nota de Débito.	Cambiar datos de la Nota de Débito en caso de que haya alteraciones o incidencias.	1	Alta
50	Director	Cambiar estados de Nota de Débito.	Llevar la secuencia de la vida de una Nota de Débito.	1	Alta
51	Contador	Mostrar los estados del Nota de Débito.	Tener control de cada estado de cada Nota de Débito.	3	Baja
52	Director	Anular Nota de Débito.	Inhabilitarlo en caso de que haya una incidencia con un dato o cifra con la Nota de Débito.	1	Alta
53	Director y Contador	Generar vista previa del Nota de Débito.	Tener información completa de esa Nota de Débito.	2	Media

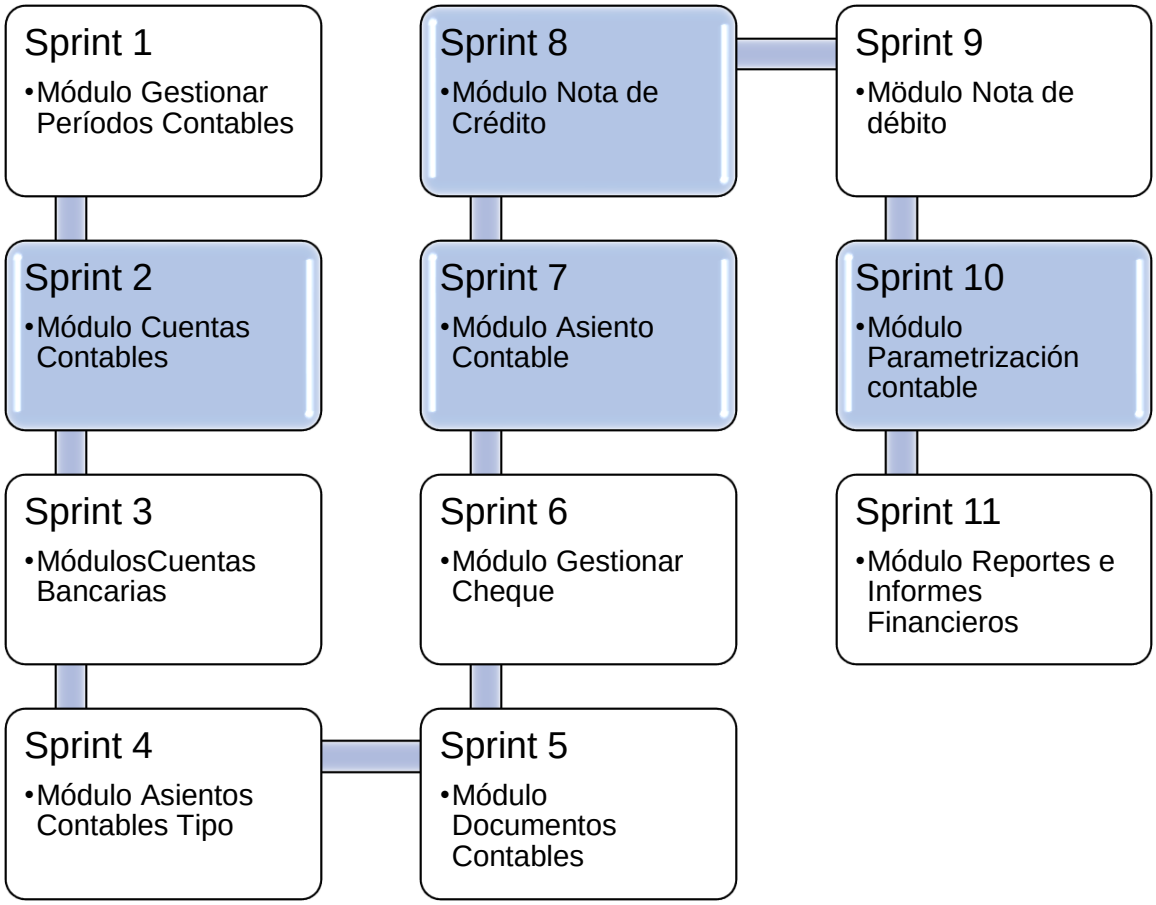
N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Esfuerzo	Prioridad
54	Director y Contador	Buscar Nota de Débito por Beneficiario, Fecha de Creación y Área.	Tener información de una Nota de Débito de una manera más rápida.	3	Baja
55	Director y Contador	Agregar Parametrización Contable.	Añadir Parametrización Contable al SIGFA.	1	Alta
56	Director	Editar Parametrización Contable.	Cambiar datos a la Parametrización Contable en caso de que haya alteraciones o incidencias.	1	Alta
57	Director y Contador	Anular Parametrización Contable.	Inhabilitarla en caso de que haya una incidencia con un dato o cifra con la Parametrización Contable.	1	Alta
58	Director y Contador	Buscar una Parametrización Contable por tipo documento, cuenta debe o haber y código catálogo.	Tener información de una Parametrización Contable de una manera más rápida.	3	Baja
59	Director y Contador	Anular Nota de Débito.	Inhabilitarlo en caso de que haya una incidencia con un dato o cifra con la Nota de Débito.	1	Alta
60	Director y Contador	Generar vista previa del Nota de Débito.	Tener información completa de esa Nota de Débito.	2	Media
61	Director y Contador	Buscar Nota de Débito por Beneficiario,	Tener información de una Nota de Débito de una manera más rápida.	3	Baja

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Esfuerzo	Prioridad
		Fecha de Creación y Área.			
62	Director y Contador	Consultar datos de los Estados Financieros (Activo, Pasivo, Patrimonio, ingreso y egreso) en Landing Page por programa contable.	Tener una vista previa del estado actual de los datos financieros de la UNI.	1	Alta
63	Director y Contador	Generar reporte de Balance general por programa específico.	Tener información sobre las cuentas de pasivo, activo y patrimonio de la UNI.	1	Alta
64	Director y Contador	Generar reporte de Balanza de comprobación por programa específico.	Tener información de los saldos deudores y acreedores de la UNI.	1	Alta
65	Director y Contador	Generar reporte de Estado de Resultado por programa específico.	Tener información de que si la UNI tuvo ganancias o pérdidas económicas.	1	Alta
66	Director y Contador	Generar reporte de Libro Diario individual por rango de fecha y por programa específico.	Mantener un histórico completo y detallado de todos los movimientos financieros de la UNI.	1	Alta
67	Director y Contador	Generar reporte de Libro Mayor por programa específico.	Conocer el saldo que va sobrando en cada cuenta contable.	1	Alta

Las funcionalidades descritas en las historias de usuario del Product Backlog presentado anteriormente se fueron desarrollando a lo largo del proyecto considerando las prioridades de la institución y con base a la organización definida por el Scrum Master y Product Owner. El proyecto se llevó a cabo en un total de 11 Sprints, cada uno con una duración de 21 días. La figura a continuación muestra cada sprint con el módulo generado como resultado. Posteriormente, en las secciones 6.2.6 hasta la 6.2.9 se presenta un resumen de resultados de los sprint 2, 7, 8 y 10; debido a limitaciones de espacio, los restantes sprint se documentan en anexos (ver sección 10.2)

Figura 3

Resumen de Sprints y sus resultados



6.2.4. Requerimientos No Funcionales del Sistema SIGFA

La seguridad es un requerimiento no funcional muy relevante en este sistema, para documentar de manera más precisa, se desglosa en la tabla a continuación en requerimientos no funcionales más específicos esperados para el Sistema Integro de Gestión Financiero Administrativo SIGFA.

Tabla 9: Requerimientos No Funcionales

Requerimientos No Funcionales	Descripción
1.3. Autenticación y Autorización	1. Implementar un sistema de autenticación con usuario y contraseña seguros, con políticas de complejidad. 2. Controlar el acceso basado en roles (contador, director) para limitar las acciones según el tipo de usuario.
1.4. Gestión de Sesiones	1. Establecer un tiempo de inactividad tras el cual la sesión se cierre automáticamente. 2. Prohibir sesiones simultáneas para el mismo usuario en diferentes dispositivos.
1.5. Registro de Actividades (Auditoría)	1. Registrar los eventos relevantes del sistema (modificación, creación, y eliminación de periodos, cuentas, documentos, etc.). 2. Almacenar los registros de auditoría. 3. Los registros deben incluir: fecha/hora, acción realizada, usuario y estado previo del recurso.
1.6. Cifrado	1. Encriptar la comunicación entre el cliente y el servidor mediante HTTPS. 2. Cifrar los datos sensibles como contraseñas y datos financieros almacenados en la base de datos.
1.7. Control de Errores	1. Implementar políticas de gestión de errores para evitar divulgación de detalles técnicos sensibles.

6.2.5. Planeación y preparación de condiciones iniciales del proyecto

Para la planificación del proyecto se utilizó la herramienta Microsoft Project, donde cada Sprint tuvo una duración fija de 21 días, y cada sprint representa un conjunto de tareas del Product backlog acordadas con el Product Owner.

Primeramente, se asignaron los roles a cada miembro del equipo de desarrollo, esto con el fin de establecer de una vez la o las tareas que tendrá cada desarrollador en el SIGFA.

También, el equipo de desarrollo definió el tipo de arquitectura que se iba a utilizar para el desarrollo tanto del Frontend como del Backend. Para el Frontend se utilizó el tipo de arquitectura orientada por componentes e inyección de servicios; en el Backend se utilizó la arquitectura inyección de servicios. A continuación, un ejemplo que evidencia cada una de estas:

Figura 4
a) Estructura Backend del SIGFA

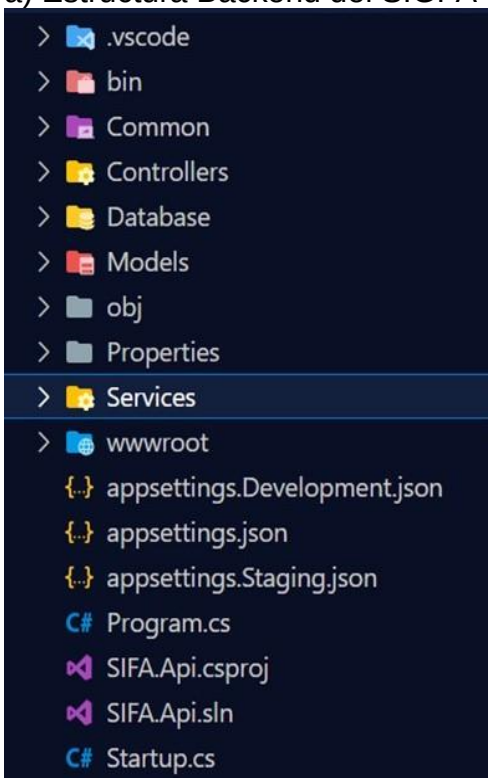
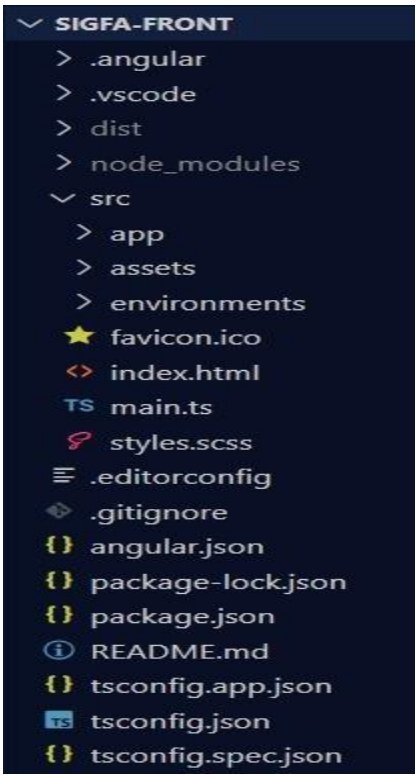


Figura 4
b) Estructura Frontend del SIGFA



Otro punto que se planteó fue el gestor de repositorios con que se iba a trabajar, esto con el fin de trabajar de forma colaborativa y por seguridad para el manejo de versiones de código. Se crearon dos repositorios: ApiSigfa y FrontSigfa, donde cada desarrollador del equipo de desarrollo tiene su rama en la que trabaja de

forma individual las tareas asignadas, para luego hacer “merge” (unión del código de las ramas) con la rama principal.

Así también, se sostuvieron reuniones con especialistas en contabilidad y finanzas del personal de la UNI para profundizar más y comprender mejor la lógica de negocios que se implementaría en el sistema.

6.2.5.1. Medidas de seguridad de OWASP implementadas en el proyecto

Se utilizó el top ten de OWASP con el fin de tener una codificación más segura y confiable. Estas medidas se implementaron en el Backend, Frontend y en la base de datos. Es importante recalcar que las medidas de seguridad recomendadas por OWASP están alineadas con las prácticas de seguridad implementadas en el Sistema Institucional de Seguridad y Autorización (SISA) existente en la UNI, el cual implementa todo un conjunto de prácticas de seguridad que deben utilizarse en todos los sistemas informáticos desarrollados; por tanto, se retomaron en el proyecto. A continuación, se listan las prácticas de seguridad retomadas del sistema SISA para el uso en este proyecto, según indicaciones del responsable de Oficina de Sistemas:

En esta tabla se muestra las medidas tomadas para implementar los Requerimientos No Funcionales de seguridad del SIGFA asociadas al Top Ten de OWASP.

Tabla 10: Asociación de Requerimientos NO-funcionales con las Pruebas OWASP

Top Ten de OWASP	Medidas a tomar para implementar los requerimientos No funcionales
1. Pérdida de Control de Acceso (A01:2021)	• Denegar accesos por defecto, solo habilitar los necesarios. • Aplicar control de acceso en todas las capas de la aplicación. • Registrar intentos fallidos de acceso para auditoría.
2. Fallas Criptográficas (A02:2021)	• Cifrar datos en tránsito y en reposo. • Evitar almacenamiento de datos sensibles innecesarios. • Usar protocolos seguros (SSL/TLS).
3. Inyección (A03:2021)	• Usar consultas parametrizadas para prevenir SQL Injection. • Implementar validaciones de entrada y filtros seguros (listas blancas). • Limitar la cantidad de datos retornados en consultas SQL.
4. Diseño Inseguro (A04:2021)	• Diseñar separación clara de roles desde el inicio. • Limitar recursos según el usuario o servicio. • Asegurar que el control de acceso esté integrado en todas las capas.
5. Configuración de Seguridad Incorrecta (A05:2021)	• Establecer un proceso automatizado para validar configuraciones. • Enviar directivas de seguridad a los usuarios. • Eliminar componentes o funcionalidades no necesarias.
6. Componentes Vulnerables y Desactualizados (A06:2021)	• Realizar un inventario continuo de versiones de componentes. • Supervisar bibliotecas que no reciben actualizaciones o parches. • Usar solo componentes de fuentes oficiales.
7. Fallas de Identificación y	• Evitar credenciales por defecto en el software.

Top Ten de OWASP	Medidas a tomar para implementar los requerimientos No funcionales
Autenticación (A07:2021)	• Implementar bloqueos tras múltiples intentos fallidos. • Regenerar IDs de sesión tras cada inicio de sesión exitoso.
8. Fallas en la Integridad del Software y Datos (A08:2021)	• Implementar firmas digitales para garantizar la integridad de los datos. • Usar pipelines CI/CD con controles de acceso adecuados. • Realizar revisiones continuas de código y configuraciones.
9. Fallas en el Registro y Monitoreo (A09:2021)	• Registrar acciones críticas en un formato fácil de procesar. • Implementar auditorías en operaciones sensibles. • Establecer alertas para detectar actividades sospechosas.
10. Falsificación de Solicitudes del Lado del Servidor (SSRF) (A10:2021)	• Validar y sanitizar todas las URLs recibidas del cliente. • Aplicar listas positivas para limitar las URLs permitidas. • Deshabilitar redirecciones HTTP para evitar abusos.

En la Fig. 5 se evidencia parte de la implementación de servicios de seguridad para el uso del sistema SIGFA, se evidencia con esto la implementación de roles de usuario en el código para asegurar que solamente los roles pertinentes tengan acceso a las pantallas correspondientes.

Figura 5

Servicio de seguridad de pantalla

```
export class SeguridadService {
  ObtenerMenus(entidades: EntidadMenu_Seguridad): Observable<Menu_Seguridad[]> {
    return this.http
      .post<Menu_Seguridad[]>(this.keyGlobal + this.keymenus, entidades)
      .pipe(
        map((respuesta: Menu_Seguridad[]) => {
          return respuesta;
        })
      );
  }

  ObtenerAcciones(acciones: Acciones) {
    return this.http
      .post<RespuestaAcciones>(this.keyGlobal + this.keyAcciones, acciones)

      .pipe(
        map((respuesta: RespuestaAcciones) => {
          return respuesta;
        })
      );
  }
  obtenerBotones(pantalla: string) {
    let arreglo: Acciones = {
      sistema: environment.claveSistema,
      entidad: '',
      pantalla: pantalla,
      entidades: [environment.claveEntidad],
    };

    return this.ObtenerAcciones(arreglo).pipe(
      map((respuesta: RespuestaAcciones) => {
        return respuesta;
      })
    );
  }
```

6.2.6. Sprint 2

Este sprint consistió en el desarrollo del módulo de las cuentas contables que administrará el sistema SIGFA, este módulo cuenta con las funcionalidades principales: agregar nueva Cuenta Contable, dar de baja a una cuenta contable, editar una cuenta contable. Estas funcionalidades y otros puntos se abordarán más a detalle en adelante.

6.2.6.1. Sprint Planning

Este segundo Sprint Planning se enfocará en el módulo para Cuentas Contables. El Product Backlog de este Sprint indica el desarrollo de las historias de usuario que se incluyen en la Tabla 11.

Durante el Sprint 2, con una duración de 21 días asignado para el módulo de cuentas contables, se llevaron a cabo reuniones diarias de Scrum con el objetivo de mantener a todos los miembros del equipo informados sobre el progreso de las tareas asignadas. Estas reuniones proporcionaron un espacio para compartir dificultades, recibir retroalimentación y compartir las tareas completadas, con el fin de facilitar la colaboración entre desarrolladores y resolver cualquier obstáculo que pudiera surgir.

Tabla 11**:** Sprint Backlog para el sprint 2

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Responsable	Estimación
1	Contador	Agregar Cuenta Contable.	Añadir cuentas al sistema SIGFA.	Jorge Torres; Alfredo Eva	5 días
2	Director	Editar Cuenta Contable.	Arreglarlas en caso de que haya cambios.	Carlos Martínez	2 días
3	Director	Generar árbol de Cuentas Contables.	Tener un control ordenado de las cuentas padres e hijas de la UNI.	Carlos Martínez; Alfredo Eva	8 días
4	Contador	Anular Cuenta Contable.	Ponerlas en estado inhabilitadas en el sistema SIGFA.	Ronnie Acevedo	2 días
5	Contador y Director	Mostrar Cuentas Contables.	Pueda visualizar de forma clara y precisa.	Alfredo Eva	3 días
6	Contador	Buscar por Cuentas Contable y por nombre.	Encontrar información de esa cuenta con rapidez.	Ronnie Acevedo	1 día

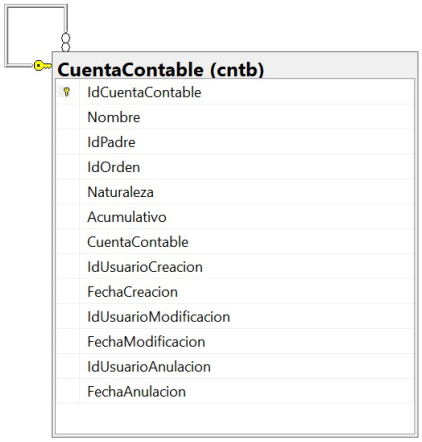
6.2.6.2. Resumen del proceso de desarrollo de software del Sprint 2

Al igual que en el sprint anterior, la implementación de las tablas se llevó a cabo directamente en Microsoft SQL Server, con un enfoque centrado en las historias de usuario definidas para este Sprint, así como en las instrucciones detalladas proporcionadas por expertos en los procesos institucionales pertinentes.

Este proceso incluyó la creación de la estructura de las tablas relacionadas con las cuentas contables, incluyendo las tablas con sus controladores que a su vez esto genera sus Endpoint; que son utilizados por el Frontend para mostrar la información de las cuentas contables, según fuera necesario para garantizar un funcionamiento óptimo del módulo de mencionado. Se muestra un ejemplo del diseño de la base de datos correspondiente a la historia de usuario 1, 2, 3, 4, 5 del sprint backlog 2.

Figura 6

Diseño de la base de datos, tabla para Cuentas Contables



Siguiendo con el proceso de desarrollo, se llevó a cabo la implementación tanto del Backend como del Frontend utilizando herramientas específicas. Para el Frontend, se utilizó Visual Studio Code junto con el Framework Angular, mientras que para el Backend se empleó Visual Studio en conjunto con Entity Framework 6.0. Se muestra un ejemplo del código correspondiente a la historia de usuario 1,2, 3, 4, 5 del sprint backlog 2.

Figura 7

Código del servicio para obtener todos los registros de Cuentas Contables, para diseñar el Árbol de Cuentas Contables

```
getArbol(nombre: string | null): Observable<CuentaContable[]> {
  let params = new HttpParams();
  nombre != null && nombre != ''
    ? (params = params.set('Nombre', nombre))
    : null;
  params = params.set('registroPorPagina', 0);
  params = params.set('OrientarPor', 'ASC');
  return this.http.get<CuentaContable[]>(this.API, {
    headers: {
      Authorization: `Bearer ${this.TOKEN}`,
      entidad: [environment.claveEntidad],
    },
    params,
  });
}
```

Figura 8

Código del servicio para registrar, modificar y anular una Cuenta Contable

```
post(cuenta: CuentaContableInsertar): Observable<RespuestaServidor> {
  return this.http.post<RespuestaServidor>(this.API, cuenta, {
    headers: {
      Authorization: `Bearer ${this.TOKEN}`,
    },
  });
}
update(
  cuenta: CuentaContableInsertar,
  id: number
): Observable<RespuestaServidor> {
  return this.http.put<RespuestaServidor>(`${this.API}/${id}`, cuenta, {
    headers: {
      Authorization: `Bearer ${this.TOKEN}`,
    },
  });
}
delete(id: number): Observable<RespuestaServidor> {
  return this.http.delete<RespuestaServidor>(`${this.API}/${id}`, {
    headers: {
      Authorization: `Bearer ${this.TOKEN}`,
    },
  });
}
```

Figura 9

Seguridad de componentes de botones en la pantalla de Cuenta Contable

```
<div class="" *ngFor="let element of banderaFiltro">
  <app-button
    *ngIf="element.accion == 'Buscar'"
    descripcion="{{ element.etiqueta }}"
    color="{{ element.color }}"
    icono="{{ element.icono }}"
    (disparador)="operacion(element.accion)"
    [visible]="element.visible"
  >
  </app-button>
</div>
```

También se llevó a cabo el diseño de las interfaces relacionadas con las cuentas Contables. Estos diseños, junto con la estructura de navegación de las pantallas, fueron inicialmente bosquejados en papel para luego ser implementados utilizando Visual Studio Code.

Además, el desarrollo del Backend estuvo estrechamente vinculado al del Frontend. A medida que surgían especificaciones de historias de usuario para la visualización o consumo de datos, se implementaban los Endpoint necesarios para completar el flujo de trabajo. Esta sincronización entre el Backend y el Frontend garantizó una integración coherente y eficiente a lo largo de todo el proceso de desarrollo. Se muestra un ejemplo del diseño de las pantallas correspondiente a la historia de usuario 5 y 6 del sprint backlog 2.

Figura 10

Pantalla principal, donde se listan las Cuentas Contables

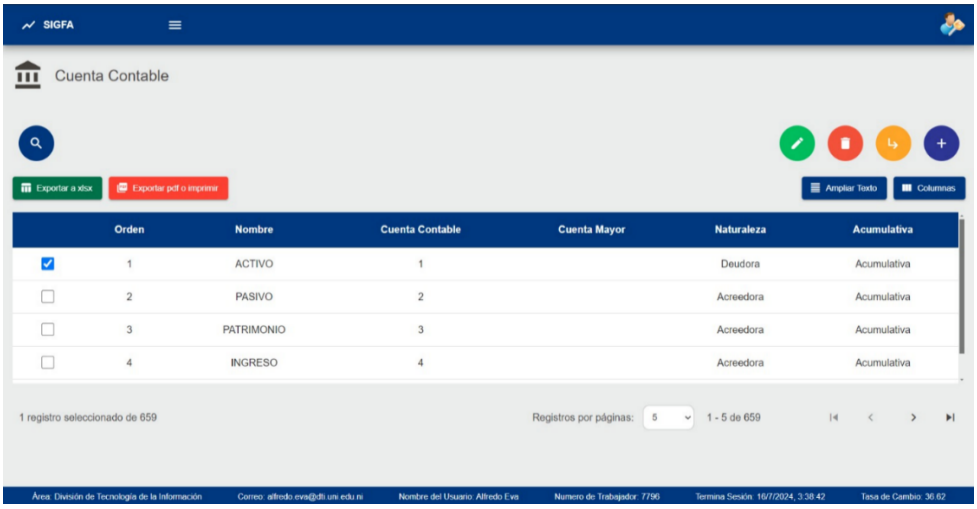
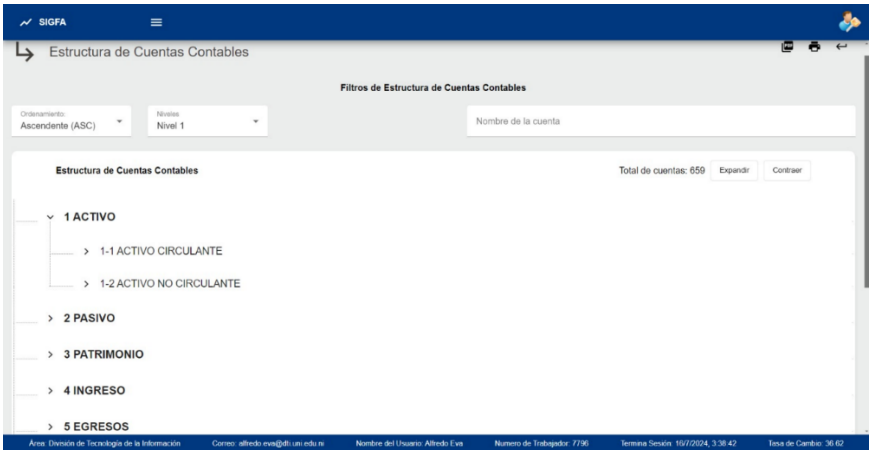


Figura 11

Árbol de Cuentas Contables



Los avances del módulo fueron subidos al repositorio GitLab, una vez estos eran revisados y aprobados por el ing. Carlos Martínez pasaban a la rama principal del proyecto.

6.2.6.3. Revisión del Sprint 2: entrega del producto terminado, Cuentas Contables

Una vez completado el desarrollo de acuerdo con los elementos definidos en el Product Backlog, se llevaba a cabo una presentación del producto al Product Owner, el Ingeniero Omar Quintana, así como a los stakeholders, entre los que se incluían el Licenciado Manuel Olivares Solís y el Licenciado Gustavo Adolfo Silva Bermúdez. Durante esta sesión, los responsables evaluaban el trabajo entregado, formalizando sus conclusiones, documentando los acuerdos alcanzados vía correo electrónico, que recogía los acuerdos alcanzados. Además, se aprovechaba la oportunidad para discutir cualquier pregunta, duda o posible mejora que pudiera surgir, con el objetivo de garantizar la alineación del producto con las expectativas y necesidades del cliente.

El equipo de desarrollo llevó a cabo la demostración y entrega del segundo sprint, que abarcaba el desarrollo completo de las cuentas contables dentro del sistema

SIGFA. Durante la entrega, se garantizó que el módulo estuviera completamente funcional.

Además, se cumplió la entrega en tiempo y forma según planificación. El módulo resultante abarca las siguientes funcionalidades.

1. Agregar Cuenta Contable.
2. Editar Cuenta Contable.
3. Generar árbol de Cuentas Contables.
4. Anular Cuenta Contable.
5. Mostrar Cuentas Contables.
6. Buscar por Cuentas Contable y Nombre.

Feedback y retroalimentación.

Durante la sesión, las partes interesadas plantearon diversas preguntas, entre las que se destacan:

1. ¿Cuál es el proceso para registrar una nueva Cuenta Contable en el sistema SIGFA?
2. ¿Cómo se procede para dar de baja una Cuenta Contable?
3. ¿Cuál es la metodología para llevar a cabo la búsqueda de una cuenta contable en el sistema?
4. ¿Qué pasos se siguen para editar la información de una Cuenta Contable específica?

Estas interrogantes fueron abordadas a través de la demostración del módulo desarrollado del sistema. Asimismo, se proporcionará un manual de usuario detallado que permitirá aclarar estas y otras dudas similares al utilizar el sistema, garantizando así una comprensión.

En la reunión de entrega de este segundo Sprint, el Product Owner Pablo David Trejos aprobó la ubicación y tamaño de los botones, consolidando así la estructura definitiva de las pantallas del sistema SIGFA.

Con esto, se cerró la reunión de entrega del Sprint, logrando un alto nivel de éxito al cumplir en tiempo y forma con los requisitos establecidos por el Product Owner. El Scrum Master Ing. Rodrigo Diaz coordinó la planificación de la siguiente reunión para la preparación del siguiente Sprint.

6.2.6.4. Retrospectiva de Sprint 2

El Scrum Master Ing. Rodrigo Diaz convocó al Equipo de Desarrollo para llevar a cabo una retrospectiva del segundo Sprint, con el propósito de identificar posibles fallos como equipo y áreas de mejora. Durante esta sesión reflexiva, se fomentó un ambiente de colaboración y retroalimentación, permitiendo que cada miembro del equipo de desarrollo expresara sus opiniones y sugerencias para optimizar el proceso de desarrollo en futuras iteraciones.

Identificación de fallos

Durante la revisión del proceso de desarrollo, el equipo identificó una problemática importante, se observó que algunos desarrolladores estaban introduciendo código basura en sus cambios, como líneas comentadas, variables no utilizadas y falta de comentarios en partes importantes del código. Estas prácticas dificultaban la revisión y comprensión del código por parte del equipo, lo que afectaba negativamente la calidad y eficiencia del desarrollo.

Análisis de la raíz del fallo

El equipo de desarrollo identifico la raíz del fallo de estas malas prácticas de codificación, entre las cuales se mencionaron las siguientes:

1. Falta de conciencia.
2. Presión por cumplir plazos.
3. Falta de estándares y prácticas de codificación.
4. Falta de capacitación.
5. Falta de supervisión.

Solución

Se resolvió esta problemática mediante la implementación de planes de capacitación para el equipo de desarrollo, enfocados en promover buenas prácticas de codificación, así como codificación limpia y clara. Además, se establecieron estándares claros de codificación y se implementó un proceso de revisión regular por parte del Ing. Carlos Martínez encargado de subir los cambios en la rama principal.

6.2.7. Sprint 7

Este sprint, se enfocó en el desarrollo del módulo de Asiento Contable (Comprobante de Pago y Diario) para la administración del sistema SIGFA. Este módulo se centra en los movimientos contables que son generados por las líneas de cuentas, beneficiarios (Trabajador, Estudiante, Proveedor y Unidad Ejecutara) y montos (Debe o Haber) y que estos contabilizados en la balanza de comprobación para llevar los estados financieros de la Universidad Nacional de Ingeniería (UNI).

6.2.7.1. Sprint Planning

Este séptimo Sprint Planning se enfocará en el módulo de Asiento Contable. El Product Backlog de este Sprint indica el desarrollo de las historias de usuario que se incluyen en la tabla 12.

Durante el séptimo sprint, con una duración de 21 días, designado para el módulo de “cheque”, se llevaron a cabo reuniones diarias de Scrum con el propósito de mantener a todos los integrantes del equipo al tanto del avance de las tareas asignadas.

Tabla 12**:** Sprint Backlog para el sprint 7

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Responsable	Estimación
1	Contador	Agregar Asiento Contable.	Añadir Asiento Contable al sistema SIGFA.	Alfredo Eva; Jorge Torres; Carlos Martínez	6 días
2	Director	Editar Asiento Contable.	Cambiar datos del Asiento Contable. en caso de que haya alteraciones o incidencias.	Alfredo Eva; Carlos Martínez	6 días
3	Director	Cambiar estados de Asiento Contable.	Llevar la secuencia de la vida de Asiento Contable	Alfredo Eva; Carlos Martínez	3 días
4	Director y Contador	Mostrar los estados del Asiento Contable.	Tener control del estado de cada Asiento Contable.	Alfredo Eva; Carlos Martínez	1 día
5	Director	Anular Asiento Contable.	Anularlo en caso de que haya una incidencia con un dato o cifra en el Asiento Contable.	Alfredo Eva; Carlos Martínez	1 día
6	Director y Contador	Generar vista previa del Asiento Contable.	Tener información completa de ese Asiento Contable.	Alfredo Eva; Carlos Martínez	3 días
7	Director y Contador	Buscar Asiento Contable por Beneficiario, Fecha de Creación y Área.	Tener información de un Asiento Contable de una manera más rápida.	Alfredo Eva; Carlos Martínez	1 día

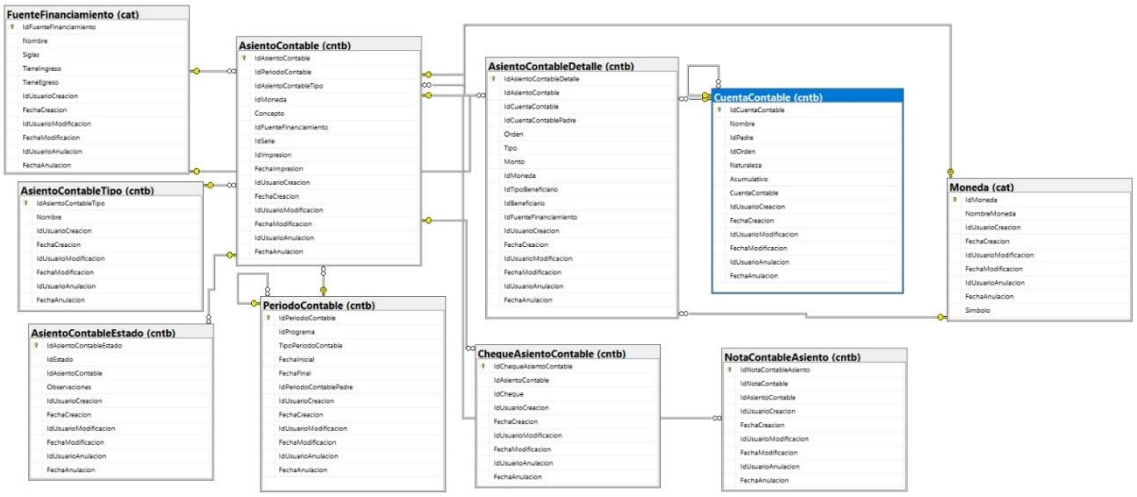
6.2.7.2. Resumen del proceso de desarrollo de software del Sprint 7

Tal como sucedió en Sprints previos, el desarrollo de las tablas se llevó a cabo directamente en Microsoft SQL Server, con un enfoque centrado en las historias de usuario definidas para este Sprint, así como en las instrucciones detalladas proporcionadas por expertos en los procesos institucionales pertinentes.

Este proceso incluyó la creación de la estructura de las tablas relacionadas con asiento contable, incluyendo las tablas con sus controladores que a su vez esto genera sus Endpoint; que son utilizados por el Frontend para mostrar la información de los asientos contables y sus estados. A continuación, en la Fig. 12. Se muestra un ejemplo del diseño de la base de datos correspondiente a la historia de usuario 1, 2, 3, 4, 5 del sprint backlog 7.

Figura 12

Diseño de la base de datos, del módulo de Asiento Contable



Continuando con el proceso de desarrollo, se procedió a implementar tanto el Backend como el Frontend utilizando herramientas específicas (Frontend: Visual Studio Code en combinación con el Framework Angular; Backend: Visual Studio junto con Entity Framework 6.0). A continuación, se presenta un ejemplo del código correspondiente a la historia de usuario 1,2, 3, 4, 5, 6 y 7 del sprint backlog 7.

Figura 13

Código del evento para registrar un Asiento Contable (Comprobante de Pago o Diario) junto a su detalle de cuentas contables, beneficiarios y montos

```
export class AsientocontablepartidadobleComponent
  // LLama al metodo post para insertar los datos
  insertarDatos(data: AsientoContableAdd, matgroup: any): void {
    let dataDetalle: DetalleLineaContable[] | null =
      this.validarEnvioDetalleAgregar();

    if (dataDetalle != null && dataDetalle.length > 1) {
      try {
        this.asientocontableService.post(data).subscribe({
          next: (res) => {
            var ddata = JSON.stringify(res);
            var dataMensaje = JSON.parse(ddata);
            let respuestaServerAsientoContableID =
              dataMensaje['idAsientoContable'];
            if (
              respuestaServerAsientoContableID == null &&
              respuestaServerAsientoContableID == 0
            ) {
              return this.toast.openSnackBar(
                'Error al agregar el Asiento Contable Padre.'
              );
            }
```

Figura 14

Código del controlador para registrar el Asiento Contable Padre

```
[HttpPost]
0 referencias
public async Task<IActionResult> PostAsync([FromBody] AsientoContableCreate c, [FromHeader] String[] entidad)
    {

        try
        {
            if (entidad.IsNullOrEmptyArray())
                return StatusCode(StatusCodes.Status401Unauthorized,
                    new { Mensaje = "Error, no autorizado." });
            ConsultaPermiso cp = new ConsultaPermiso(_configuration)
            {
                Accion = _configSecurity.HomologacionAccion(Accion.Guardar),
                Pantalla = _configSecurity.HomologacionPantalla(Pantalla.AsientoContableDiario),
                Entidades = entidad
            };
            var permiso = await _httpPermisoService.VerificarPermisoAsync(cp);
            if (!permiso.TienePermiso)
                return StatusCode(StatusCodes.Status403Forbidden,
                    new { Mensaje = "Error, no tiene asignado los permisos necesarios para realizar la acción." });



            c.IdUsuarioCreacion = await usuarioService.ObtenerIdUsuarioAsync(Request.ObtenerUsuarioGuidHeader());
            AsientoContable asientoContable = await asientoContableService.InsertarAsientoContable(c);

    return StatusCode(StatusCodes.Status201Created,
      new
      {
        Mensaje = "El asiento contable se ha guardado exitosamente.", asientoContable.IdAsientoContable
      });
  }
```

También se realizó el diseño de las interfaces asociadas al Asiento Contable (Comprobante de Pago y Diario). Estos diseños, junto con la estructura de

navegación de las pantallas, fueron inicialmente diseñadas en papel para luego ser implementados utilizando Visual Studio Code.

Además, el desarrollo del Backend estuvo estrechamente ligado al del Frontend. A medida que surgían requisitos para la visualización o consumo de datos, se implementaban los Endpoint necesarios para completar el flujo de trabajo. Esta sincronización entre el Backend y el Frontend garantizó una integración coherente y eficiente a lo largo de todo el proceso de desarrollo. A continuación, se presenta un ejemplo del diseño de las pantallas correspondiente a la historia de usuario del 1 al 7 del backlog del sprint 7.

Figura 15

Pantalla principal del Comprobante Pago

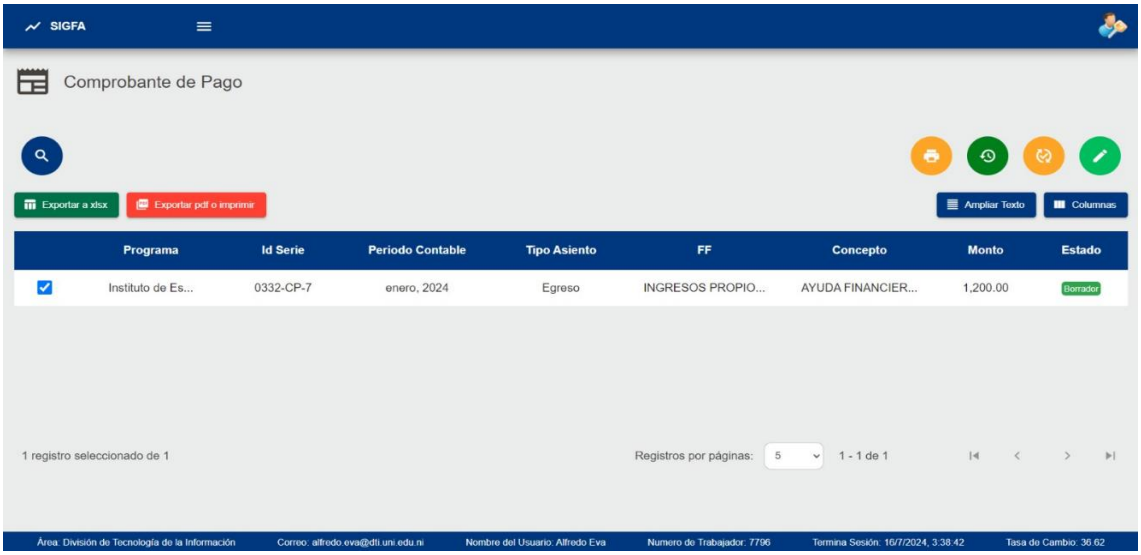
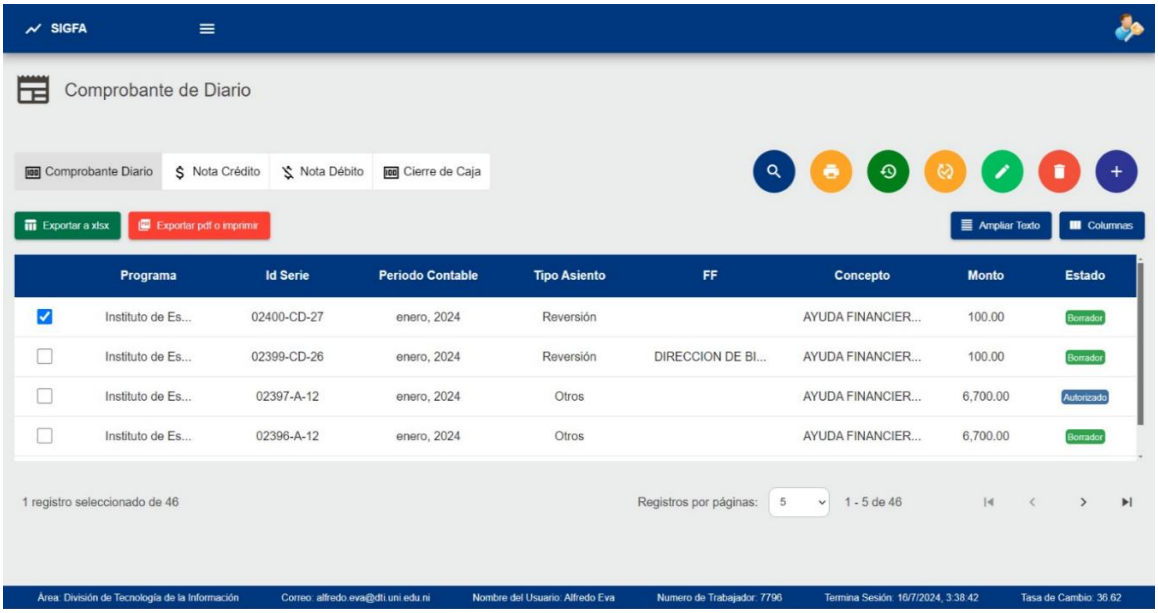


Figura 16

Pantalla principal del Comprobante Diario



Los avances del módulo fueron subidos al repositorio GitLab, una vez estos eran revisados y aprobados por el Ing. Carlos Martínez pasaban a la rama principal del proyecto.

6.2.7.3. Revisión del Sprint 7: entrega de producto terminado, Gestionar Asiento Contable

Una vez finalizado el desarrollo de acuerdo con los elementos definidos en el Product Backlog, se procedía a realizar una presentación del producto al Product Owner MSc. Pablo David Trejos y el stakeholder Lic. Gustavo Adolfo Silva Bermúdez. Durante esta sesión, los responsables evaluaban el trabajo entregado y formalizaban sus conclusiones mediante los acuerdos alcanzados vía correo electrónico.

El equipo de desarrollo llevó a cabo la demostración y entrega del séptimo sprint, que abarcaba el desarrollo completo de asiento contable dentro del sistema SIGFA. Durante la entrega, se garantizó que el módulo estuviera completamente funcional.

Además, se cumplió la entrega en tiempo y forma según planificación. El módulo resultante abarca las siguientes funcionalidades.

1. Editar Asiento Contable.
2. Anular Asiento Contable.
3. Mostrar Asiento Contable.
4. Buscar Asiento Contable.
5. Cambiar estado del Asiento Contable.
6. Histórico del Asiento Contable

Feedback y retroalimentación.

Durante la sesión, las partes interesadas plantearon diversas preguntas, entre las que se destacan:

1. ¿Cuándo se anula un comprobante establece en cero su detalle?
2. ¿Cuál es el proceso para editar una línea contable en el caso de las cuentas de ingreso o egreso?
3. ¿Cómo se registran los parciales?
4. ¿Se muestran las cuentas de mayor?

Estas interrogantes fueron abordadas a través de la demostración del módulo desarrollado del sistema. Asimismo, se proporcionará un manual de usuario detallado que permitirá aclarar estas y otras dudas similares al utilizar el sistema, garantizando así una comprensión.

Con esto, se cerró la reunión de entrega del Sprint 7, logrando un alto nivel de éxito al cumplir en tiempo y forma con los requisitos establecidos por el Product Owner. El Scrum Ing. Master Rodrigo Diaz coordinó la planificación de la siguiente reunión para la preparación del siguiente Sprint.

6.2.7.4. Retrospectiva de Sprint 7

El Scrum Master Ing. Rodrigo Diaz convocó al Equipo de Desarrollo para llevar a cabo una retrospectiva del séptimo Sprint, con el propósito de identificar posibles

fallos como equipo y áreas de mejora. Durante esta sesión reflexiva, se fomentó un ambiente de colaboración y retroalimentación, permitiendo que cada miembro del equipo de desarrollo expresara sus opiniones y sugerencias para optimizar el proceso de desarrollo en futuras iteraciones.

Identificación de fallos

Durante la revisión del proceso de desarrollo, se observó que el detalle del asiento contable no realiza la anulación de forma correcto, esto era causado por el uso la palabra reservada *return* en el validador de la anulación. Esto se solvento retirando esta operación al momento de terminar el proceso de anulación.

Análisis de la raíz del fallo

El equipo de desarrollo identifico que la falta de pruebas adecuadas en el Endpoint de anulación de asiento contable fue la causa principal de los problemas encontrados durante las pruebas finales.

1. Lógica programática incompleta.
2. Falla al interpretar las historias de usuario.

Solución

Se solucionó dicha problemática de la siguiente forma:

1. Aclaración de dudas en tiempo y forma.
2. Procesos complejos tienen que ser analizados con ejemplo simples.
3. Entendimiento del lenguaje de programación y sus funciones

6.2.8. Sprint 8

Este sprint, detalla el desarrollo del módulo de "Nota de Crédito ", responsable de gestionar los movimientos contables de tipo compra venta que son aplicados a los diferentes tipos de beneficiarios (Trabajadores, Estudiantes, Proveedores o Unidades Ejecutoras). Este módulo, será utilizado por los distintos centros de costos (UNI CENTRAL Y AUS) que realizan operaciones contables de tipo compra venta.

6.2.8.1. Sprint Planning

Este octavo Sprint Planning se enfocará en el módulo de “Nota de Crédito”. El Product Backlog de este Sprint indica el desarrollo de las historias de usuario que se incluyen en la tabla 13.

Durante el octavo sprint, con una duración de 21 días, designado para el módulo de “Nota de Crédito”, se llevaron a cabo reuniones diarias de Scrum con el propósito de mantener a todos los integrantes del equipo al tanto del avance de las tareas asignadas.

Tabla 13: Sprint Backlog para el sprint 8

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Responsable	Estimación
1	Contador	Agregar Nota de Crédito.	Añadir Agregar Nota de Crédito al sistema SIGFA.	Alfredo Eva	4 días
2	Director	Editar Nota de Crédito.	Cambiar datos de la Nota de Crédito en caso de que haya alteraciones o incidencias.	Jorge Torres; Carlos Martínez	4 días
3	Director	Cambiar estados de Nota de Crédito.	Llevar la secuencia de la vida de una Nota de Crédito.	Ronnie Acevedo; Gabriel García	4 días
4	Contador	Mostrar los estados de la Nota de Crédito.	Tener control de cada estado de cada Nota de Crédito.	Ronnie Acevedo	1 día
5	Contador	Anular Nota de Crédito.	Anularlo en caso de que haya una incidencia con un dato o cifra con la Nota de Crédito.	Alfredo Eva	4 días
6	Contador	Generar vista previa de las Nota de Crédito.	Tener información completa de esa Nota de Crédito.	Gabriel García	2 días
7	Contador	Buscar una Nota de Crédito por Beneficiario, Fecha de Creación y Área.	Tener información de una Nota de Crédito de una manera más rápida.	Ronnie Acevedo	1 día

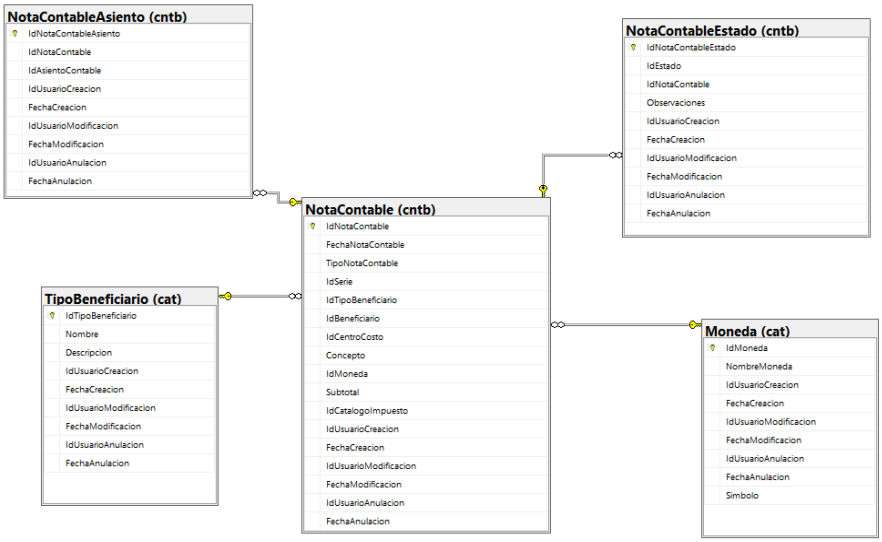
N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Responsable	Estimación
8	Contador	Mostrar las Nota de Crédito registrada.	Registros de las Notas de Crédito existentes	Alfredo Eva; Ronnie Acevedo	1 día

6.2.8.2. Resumen del proceso de desarrollo de software del Sprint 8

Tal como sucedió en Sprints previos, el desarrollo de las tablas se llevó a cabo directamente en Microsoft SQL Server, con un enfoque centrado en las historias de usuario definidas para este Sprint, así como en las instrucciones detalladas proporcionadas por expertos en los procesos institucionales pertinentes.

Figura 17

Diseño de la base de Nota de Crédito



Continuando con el proceso de desarrollo, se procedió a implementar tanto el Backend como el Frontend utilizando herramientas específicas (Frontend: Visual Studio Code en combinación con el Framework Angular; Backend: Visual Studio junto con Entity Framework 6.0). A continuación, se presenta un ejemplo del código correspondiente a la historia de usuario 1,2, 3, 4, 5 y 6 del sprint backlog 8

Figura 18

Código del evento para cambiar el estado de la Nota de Crédito

```
export class CambiarestadoComponent {
  actualizarDatos(data: CambiarEstado) {
    try {
      this.api.cambiarestado(data, this.editData.idNotaContable!).subscribe({
        next: (res) => {
          if (res.toString()) {
            this.toast.openSnackBar(res.mensaje);
            this.notaEstadoForm.reset();
            this.matDialogRef.close('update');
          } else {
            this.candado = false;
            this.toast.openSnackBar(
              'Advertencia al Modificar el Estado de la Nota Contable:' +
                res.mensaje
            );
          }
        },
        error: (err) => {
          this.errorPeticion(err, false, '');

        },
      });
    } catch (err) {
      this.errorPeticion(err, false, '');

    }
  }
```

Figura 19

Código del servicio para cambiar estado de la Nota de Crédito

```
export class NotacreditoService implements Servicio {

  cambiarestado(cambiarEstado: CambiarEstado, id: number): Observable<RespuestaServidor> {
    return this.http
      .put<RespuestaServidor>(
        `${this.API}/CambiarEstado/${id}`,
        cambiarEstado,
        {
          headers: {
            Authorization: `Bearer ${this.TOKEN}`,
          },
        }
      )

  }
```

Figura 20

Código del controlador para cambiar el estado de la Nota de Crédito

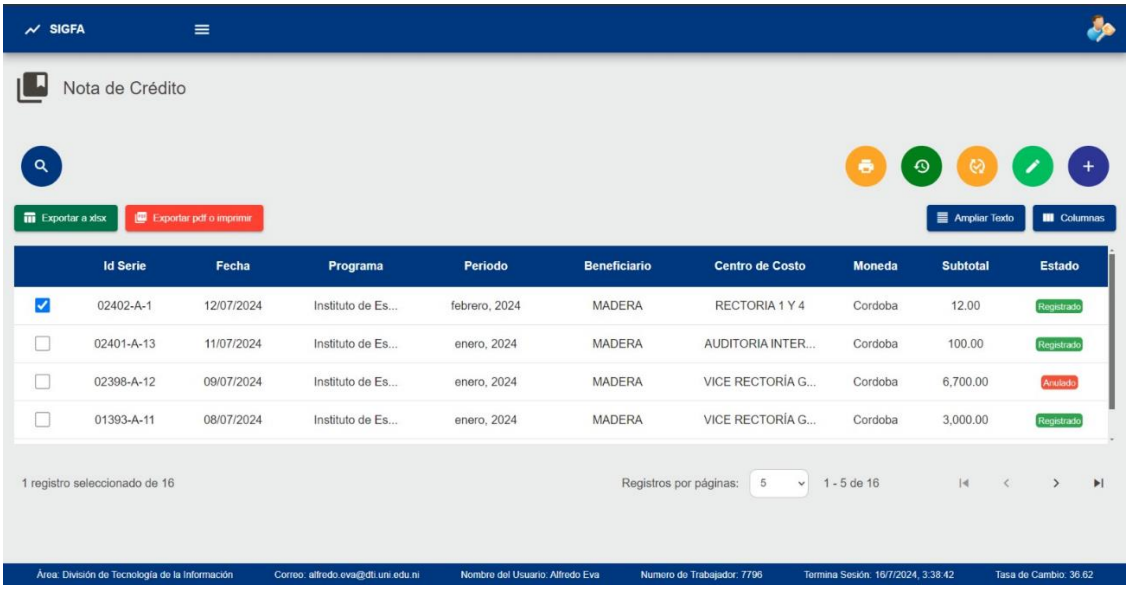
```
public class NotaContableController : ControllerBase
    [HttpPut("CambiarEstado/{idNotaContable:int}")]
    0 references
    public async Task<IActionResult> CambiarEstadoNotaContableAsync(int idNotaContable,
        [FromBody] NotaContableEstadoCreate c)
    {
        try
        {
            if (!ModelState.IsValid) return BadRequest(ModelState);

            c.IdNotaContable = idNotaContable;
            c.IdUsuarioCreacion = await usuarioService.ObtenerIdUsuarioAsync(Request.
            ObtenerUsuarioGuidHeader());

            notaContableService.CambiarEstadoNotaContable(c);

            string vmensaje = "";
            if (c.IdEstado == (int)EstadosNotaContable.Registrado)
                vmensaje = "La Nota Contable se ha cambiado a registrado exitósamente.";
            if (c.IdEstado == (int)Estados.Revisado)
                vmensaje = "La Nota Contable se ha cambiado a revisado exitósamente.";
            if (c.IdEstado == (int)Estados.Autorizado)
                vmensaje = "La Nota Contable se ha cambiado a autorizado exitósamente.";
            if (c.IdEstado == (int)Estados.Anulado)
                vmensaje = "La Nota Contable se ha a anulado exitósamente.";
            if (c.IdEstado == (int)Estados.Impreso)
                vmensaje = "La Nota Contable se ha a impreso exitósamente.";
```

También se realizó el diseño de las interfaces asociadas con las Notas de Créditos. Estos diseños, junto con la estructura de navegación de las pantallas, fueron inicialmente diseñadas en papel para luego ser implementados utilizando Visual Studio Code.

Además, el desarrollo del Backend estuvo estrechamente ligado al del Frontend. A medida que surgían requisitos para la visualización o consumo de datos, se implementaban los Endpoint necesarios para completar el flujo de trabajo. Esta sincronización entre el Backend y el Frontend garantizó una integración coherente y eficiente a lo largo de todo el proceso de desarrollo. A continuación, se presenta un ejemplo del diseño de las pantallas correspondiente a la historia de usuario del 1 al 7 del backlog del sprint 8.

Figura 21

Pantalla principal de Nota de Crédito



Los avances del módulo fueron subidos al repositorio GitLab, una vez estos eran revisados y aprobados por el ing. Carlos Martínez pasaban a la rama principal del proyecto.

6.2.8.3. Revisión del Sprint 8: entrega del producto terminado Nota de Crédito

Una vez finalizado el desarrollo de acuerdo con los elementos definidos en el Product Backlog, se procedía a realizar una presentación del producto al Product Owner MSc. Pablo David Trejos y el stakeholder Lic. Gustavo Adolfo Silva Bermúdez. Durante esta sesión, los responsables evaluaban el trabajo entregado y formalizaban sus conclusiones mediante los acuerdos alcanzados vía correo electrónico.

El equipo de desarrollo llevó a cabo la demostración y entrega del octavo sprint, que abarcaba el desarrollo completo de Nota de Crédito dentro del sistema SIGFA. Durante la entrega, se garantizó que el módulo estuviera completamente funcional.

Además, se cumplió la entrega en tiempo y forma según planificación. El módulo resultante abarca las siguientes funcionalidades.

1. Agregar Nota de Crédito.
2. Editar Nota de Crédito.
3. Anular Nota de Crédito.
4. Mostrar Nota de Crédito.
5. Buscar Nota de Crédito.
6. Cambiar estado Nota de Crédito.

Feedback y retroalimentación

Durante la sesión, las partes interesadas plantearon diversas preguntas, entre las que se destacan:

1. ¿Cuál es el proceso para editar una Nota de Crédito en el sistema SIGFA?
2. ¿Cómo se procede para anular una Nota de Crédito?
3. ¿Cuál es la metodología para llevar a cabo la búsqueda de una Nota de Crédito en el sistema?
4. ¿Qué pasos se siguen para cambiar de estado una Nota de Crédito?

Estas interrogantes fueron abordadas a través de la demostración del módulo desarrollado del sistema. Asimismo, se proporcionará un manual de usuario detallado que permitirá aclarar estas y otras dudas similares al utilizar el sistema, garantizando así una comprensión.

Con esto, se cerró la reunión de entrega del Sprint 8, logrando un alto nivel de éxito al cumplir en tiempo y forma con los requisitos establecidos por el Product Owner. El Scrum Master Ing. Rodrigo Diaz coordinó la planificación de la siguiente reunión para la preparación del siguiente Sprint.

6.2.8.4. Retrospectiva de Sprint 8

El Scrum Master Ing. Rodrigo Diaz convocó al Equipo de Desarrollo para llevar a cabo una retrospectiva del octavo Sprint, con el propósito de identificar posibles fallos como equipo y áreas de mejora. Durante esta sesión reflexiva, se fomentó un ambiente de colaboración y retroalimentación, permitiendo que cada miembro del equipo de desarrollo expresara sus opiniones y sugerencias para optimizar el proceso de desarrollo en futuras iteraciones.

Identificación de fallos

Se identificó una falla al registrar una nueva nota de crédito en el módulo del sistema SIGFA. Una posible causa de este error podía ser un problema en la validación de los datos ingresados, como inconsistencias en los campos obligatorios, formato incorrecto de la información, o conflicto con las reglas de negocio establecidas en el sistema.

Análisis de la raíz del fallo

Se identificó el fallo, cuya raíz se encontraba en la diferencia entre el tipo de dato en la base de datos y cómo se había declarado en el modelo durante la codificación. Esta diferencia de tipo de datos impedía que los registros se guardaran correctamente en la base de datos.

Solución

Para resolver esta diferencia entre el tipo de dato en la base de datos y el modelo se realizó lo siguiente:

1. Revisar y Alinear Tipos de Datos.
2. Se verifico los tipos de datos de los campos en la base de datos.
3. Se aseguró que estos tipos de datos coincidan con los tipos declarados en el código.
4. Modificar la declaración en código.
5. Se actualizo en el código para que los tipos de datos coincidan con los de la base de datos.

6.2.9. Sprint 10

Este sprint, se enfocó en el desarrollo del módulo de "Parametrización", que permite la parametrización de los documentos contables (nota de crédito, débito, cheque, o cierre de caja), que afectan las distintas cuentas y actividades contables, que se relacionan a un código de catálogo contable (Código Objeto de Gasto o Recibo Oficinal de Caja), fuente de financiamiento; que generan asientos contables (comprobante de pago o diario). El objetivo de esto es automatizar el proceso de codificación (Registro del asiento contable) de la propuesta contable (Debe y Haber).

6.2.9.1. Sprint Planning

Este décimo Sprint Planning se enfocará en el módulo de “Parametrización”. El Product Backlog de este Sprint indica el desarrollo de las historias de usuario que se incluyen en la tabla 14.

Durante el décimo sprint, con una duración de 21 días, designado para el módulo de “Parametrización”, se llevaron a cabo reuniones diarias de Scrum con el propósito de mantener a todos los integrantes del equipo al tanto del avance de las tareas asignadas.

Tabla 14: Sprint Backlog para el sprint 10

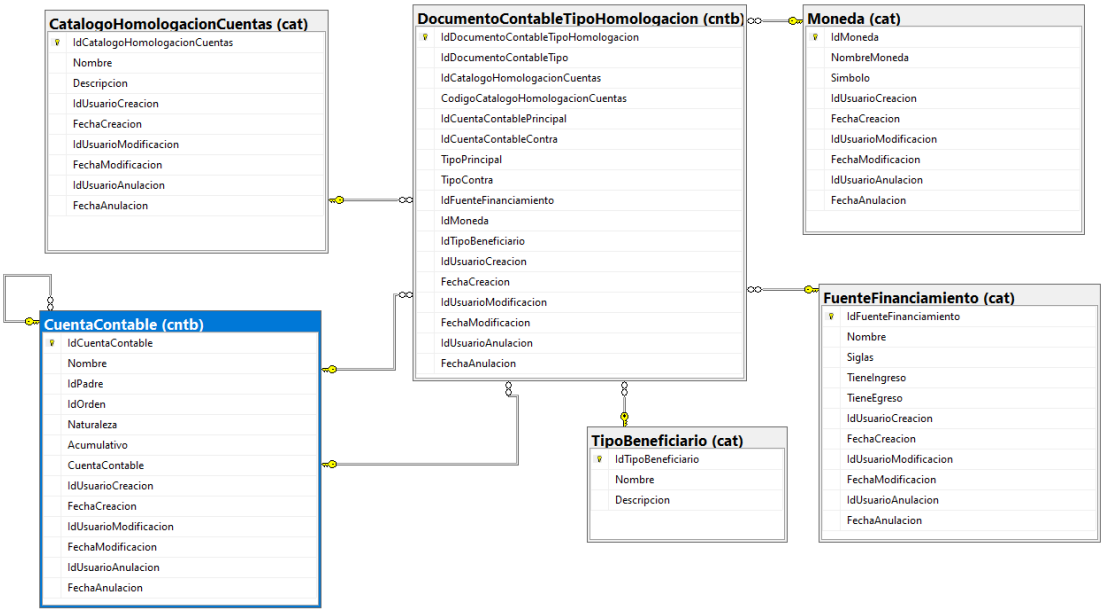
N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Responsable	Estimación
1	Director y Contador	Agregar Parametrización Contable.	Añadir Parametrización Contable al SIGFA.	Alfredo Eva; Carlos Martínez	5 días
2	Director y Contador	visualizar los tipos de documentos contables con su información.	Tener una vista clara y precisa de la información de los tipos de documentos.	Alfredo Eva; Carlos Martínez	5 días
3	Director	Editar Parametrización Contable.	Cambiar datos a la Parametrización Contable en caso de que haya alteraciones o incidencias.	Alfredo Eva; Carlos Martínez	5 días
4	Director y Contador	Anular Parametrización Contable.	Inhabilitarla en caso de que haya una incidencia con un dato o cifra con la Parametrización Contable.	Ronnie Acevedo	5 días
5	Director y Contador	Buscar una Parametrización Contable por Beneficiario y Área.	Tener información de una Parametrización Contable de una manera más rápida.	Ronnie Acevedo	1 día

6.2.9.2. Resumen del proceso de desarrollo de software del Sprint 10

Tal como sucedió en Sprints previos, el desarrollo de las tablas se llevó a cabo directamente en Microsoft SQL Server, con un enfoque centrado en las historias de usuario definidas para este Sprint, así como en las instrucciones detalladas proporcionadas por expertos en los procesos institucionales pertinentes.

Figura 22

Diseño de la base de Datos de Parametrización Contable



Continuando con el proceso de desarrollo, se procedió a implementar tanto el Backend como el Frontend utilizando herramientas específicas (Frontend: Visual Studio Code en combinación con el Framework Angular; Backend: Visual Studio junto con Entity Framework 6.0). A continuación, se presenta un ejemplo del código correspondiente a la historia de usuario 1,2, 3, 4 y 5 del sprint backlog 10.

Figura 23

Código de la petición del servicio GET, que consultas el catálogo de cuentas contables para Parametrización Contable

```
export class HomologacionService implements Servicio {

  getCatalogoHomologacionCuentas(): Observable<CatalogoHomologacionCuentas[]> {
    return this.http
      .get<CatalogoHomologacionCuentas[]>(
        `${environment.API_URL}/Catalogo/CatalogoHomologacionCuentas`,
        {
          headers: {
            Authorization: `Bearer ${this.TOKEN}`,
          },
        }
      )
```

Figura 24

Código del evento ngOnInit (Evento inicial del ciclo de vida del componente), que instancia el catálogo de cuentas para estar a la escuchar del cualquier cambio en el Frontend

```
export class InsertareditarComponent implements InsertarEditar {
  ngOnInit(): void {
    this.filteredOptionsCatalogoCuentaContable =
      this.api.getCatalogoHomologacionCuentas();

    this.filteredOptionsCatalogoCuentaContable.subscribe({
      next: (res) => {


        this.homologacionForm.controls[
          'idCatalogoHomologacionCuentas'
        ].setValue(res[0].idCatalogoHomologacionCuentas);
      },
      error: (err) => {
        this.toast.openSnackBar(
          'Error al Cargar en el Componente de Catalogo Cuentas'
        );
      },
    });
```

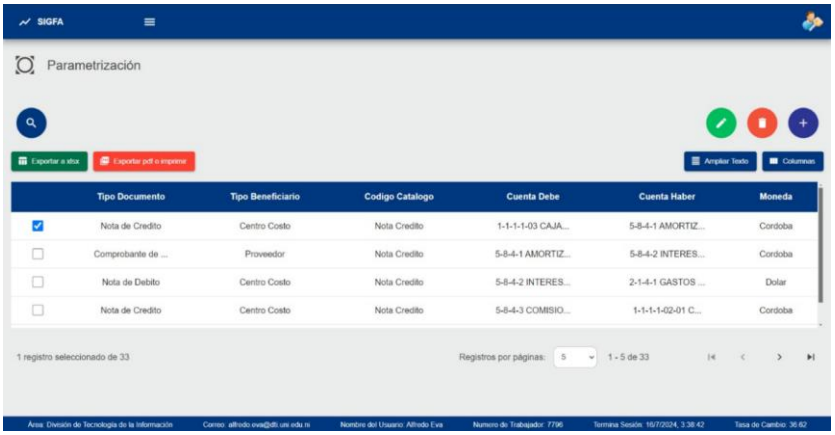
También se realizó el diseño de las interfaces asociadas con la Parametrización. Estos diseños, junto con la estructura de navegación de las pantallas, fueron inicialmente diseñadas en papel para luego ser implementados utilizando Visual Studio Code.

Además, el desarrollo del Backend estuvo estrechamente ligado al del Frontend. A medida que surgían requisitos para la visualización o consumo de datos, se implementaban los Endpoint necesarios para completar el flujo de trabajo. Esta sincronización entre el Backend y el Frontend garantizó una integración coherente y eficiente a lo largo de todo el proceso de desarrollo. A continuación, se presenta

un ejemplo del diseño de las pantallas correspondiente a la historia de usuario del 1 al 6 del backlog del sprint 10.

Figura 25

Pantalla principal del módulo Parametrización



Nota: se puede observar los datos de los tipos de documentos contables registrados en el sistema SIGFA, el código catalogo que es el tipo de documento, las cuentas deben y haber y el tipo de moneda de esa cuenta.

Los avances del módulo fueron subidos al repositorio GitLab, una vez estos eran revisados y aprobados por el Ing. Carlos Martínez pasaban a la rama principal del proyecto.

6.2.9.3. Revisión del Sprint 10: entrega del producto terminado Parametrización

Una vez finalizado el desarrollo de acuerdo con los elementos definidos en el Product Backlog, se procedía a realizar una presentación del producto al Product Owner MSc. Pablo David Trejos y el stakeholder Lic. Gustavo Adolfo Silva Bermúdez. Durante esta sesión, los responsables evaluaban el trabajo entregado y formalizaban sus conclusiones mediante los acuerdos alcanzados vía correo electrónico.

El equipo de desarrollo llevó a cabo la demostración y entrega del décimo sprint, que abarcaba el desarrollo completo de Parametrización dentro del sistema SIGFA. Durante la entrega, se garantizó que el módulo estuviera completamente

funcional. Además, se cumplió la entrega en tiempo y forma según planificación. El módulo resultante abarca las siguientes funcionalidades.

1. Agregar Parametrización.
2. Editar Parametrización.
3. Anular Parametrización.
4. Mostrar Parametrización.
5. Buscar Parametrización.

Feedback y retroalimentación.

Durante la sesión, las partes interesadas plantearon diversas preguntas, entre las que se destacan:

1. ¿Cuál es el proceso para editar una Parametrización en el sistema SIGFA?
2. ¿Cómo se procede para anular una Parametrización?
3. ¿Cuál es la metodología para llevar a cabo la búsqueda de una Parametrización en el sistema?
4. ¿Qué pasos se siguen para anular una Parametrización?

Estas interrogantes fueron abordadas a través de la demostración del módulo desarrollado del sistema. Asimismo, se proporcionará un manual de usuario detallado que permitirá aclarar estas y otras dudas similares al utilizar el sistema, garantizando así una comprensión.

Con esto, se cerró la reunión de entrega del Sprint 10, logrando un alto nivel de éxito al cumplir en tiempo y forma con los requisitos establecidos por el Product Owner. El Scrum Master Ing. Rodrigo Diaz coordinó la planificación de la siguiente reunión para la preparación del siguiente Sprint.

6.2.9.4. Retrospectiva de Sprint 10

El Scrum Master Ing. Rodrigo Diaz convocó al Equipo de Desarrollo para llevar a cabo una retrospectiva del décimo Sprint, con el propósito de identificar posibles fallos como equipo y áreas de mejora. Durante esta sesión reflexiva, se fomentó un ambiente de colaboración y retroalimentación, permitiendo que cada miembro

del equipo de desarrollo expresara sus opiniones y sugerencias para optimizar el proceso de desarrollo en futuras iteraciones.

Identificación de fallos

Se identificó un fallo al editar un tipo de documento en el módulo de Parametrización de cuentas en el sistema SIGFA. Aunque la información se ingresaba correctamente en los campos de texto, al intentar guardar los cambios en la base de datos, estos no se guardaban y se generaba un error.

Análisis de la raíz del fallo

Se identificó el fallo, el cual surgía porque se estaba pasando incorrectamente el ID de la cuenta en el Backend. Esto provocaba que la API no pudiera encontrar el dato correspondiente, generando un error de inserción en la base de datos debido a la discrepancia entre el dato que se intentaba guardar y el dato que debía guardarse.

Solución

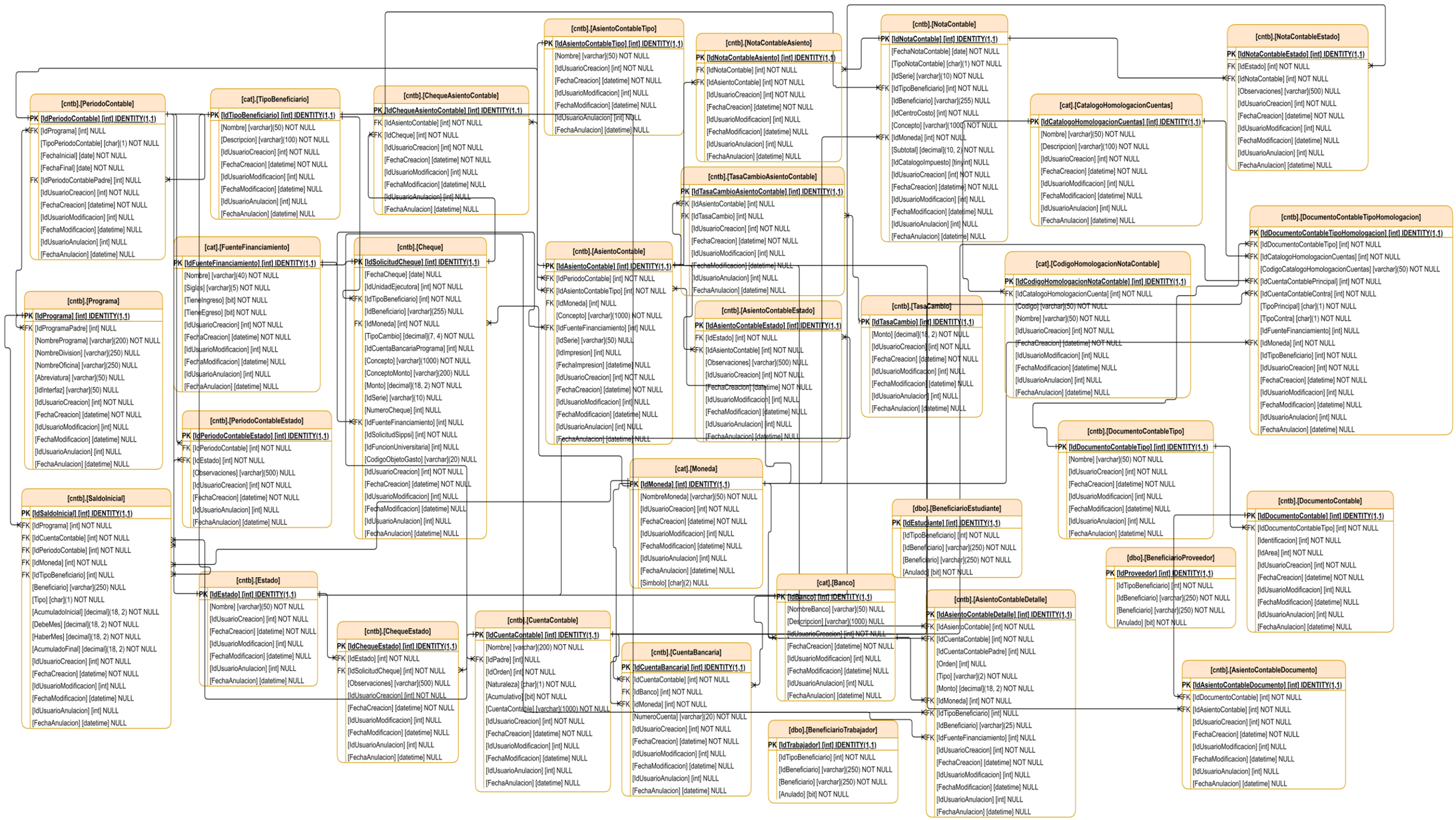
Se resolvió de la siguiente manera:

1. Se reviso y corrigió el código del Backend.
2. Se verifico que el ID de la cuenta se está pasando correctamente.
3. Se aseguró de que el ID de la cuenta se está recibiendo y procesando adecuadamente en el Backend.
4. Se añadió validaciones para confirmar que el ID de la cuenta recibido es válido y existe en la base de datos antes de intentar guardar los cambios.
5. Depuración y Pruebas.
6. Se realizó pruebas para asegurarse que el ID de la cuenta se pase correctamente y que los cambios se guardan sin errores.
7. Manejo de Errores.
8. Se implementó un manejo de errores para proporcionar mensajes de error claros cuando el ID de la cuenta no sea válido o no se pueda encontrar.

6.3. Diseño de Base de Datos Completa del SIGFA

Cada avance en el diseño de la base de datos que se generaba en cada sprint se fue integrando de manera que dicha base de datos fue creciendo paulatinamente con cada sprint. La base de datos finalizada se muestra a continuación:

Figura 26: Diseño de la base de datos SIGFA



6.4. Pruebas

6.4.1. Pruebas Funcionales

Para poder diseñar las pruebas funcionales se toman como base los “criterios de aceptación” asociados a las historias de usuario de los Sprints desarrollados. Debido a limitaciones de espacio, a continuación, se colocan como ejemplo los “criterios de aceptación” de las historias de usuario 10, 16 y 20, a los que se les da continuidad en esta sección, el resto se encuentra en anexos, en el punto 10.3.5.

Tabla 15

Criterios de aceptación para Agregar Cuenta Contable

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Criterios de aceptación
10	Director y Contador	Agregar Cuenta Contable	Añadir Cuentas Contables al sistema SIGFA.	• Las Cuentas Contables deben tener el siguiente formato 1-1 (iniciar, finalizar con números mayores a uno y separadas por guiones). • Todos los campos son requeridos, excepto la Cuenta de Mayor. • El número de Cuenta Contable, debe ser único.

Tabla 16

Criterios de aceptación para Agregar Cuenta Bancaria

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Criterios de aceptación
16	Director y Contador	Agregar Cuenta Bancaria	Añadir Cuenta Bancaria al sistema SIGFA.	• La Cuenta Bancaria debe ser de Activo (inicia con 1 su Cuenta Contable). • Todos los campos son requeridos. • El número de Cuenta Bancaria, debe ser único. • La Cuenta Bancaria, debe ser operativa.

Tabla 17

Criterios de aceptación para Agregar Asiento Contable Tipo

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Criterios de aceptación
20	Director y Contador	Agregar Asiento Contable Tipo.	Añadir Asiento Contable Tipo, al sistema SIGFA.	• Todos los campos son requeridos. • El nombre del Asiento Contable Tipo, debe ser único.

A continuación, se presentan las Pruebas Funcionales del sistema SIGFA. En total, se documenta la realización de once pruebas, una por cada sprint, debido a limitaciones de espacio. A continuación, se presentan tres de ellas; las restantes se encuentra en los anexos, punto 10.3.1. Pruebas Funcionales.

Tabla 18

Agregar Cuenta Contable

Agregar Cuenta Contable	Rol	Identificador
	Contador	01
Descripción: La prueba consiste en agregar una Cuenta Contable en el Sistema SIGFA.		
Prerrequisitos: Haber iniciado sesión como Contador.		
Pasos: 1. Ingresar a la pantalla de Cuenta Contable. 2. Ingresar a la opción de agregar Cuenta Contable. 3. Ingresar todos los datos necesarios para agregar una Cuenta Contable. 4. Si los datos son incorrectos o la Cuenta Contable ya existe el sistema desplegara un mensaje de error. 5. Si los datos son correctos el sistema guardara la Cuenta Contable.		
Resultado esperado: Nueva Cuenta Contable agregada exitosamente.		
Resultado obtenido: Nueva Cuenta Contable agregada exitosamente.		

Tabla 19

Agregar Cuenta Bancaria

Agregar Cuenta Bancaria	Rol	Identificador
	Contador	02
Descripción: La prueba consiste en agregar una Cuenta Bancaria en el Sistema SIGFA.		
Prerrequisitos: Haber iniciado sesión como Contador.		
Pasos: 1. Ingresar a la pantalla de Cuenta Bancaria. 2. Ingresar a la opción de agregar Cuenta Bancaria. 3. Ingresar todos los datos necesarios para agregar una Cuenta Bancaria 4. Si los datos son incorrectos o la Cuenta Bancaria ya existe el sistema desplegara un mensaje de error. 5. Si los datos son correctos el sistema guardara la Cuenta Bancaria.		
Resultado esperado: Nueva Cuenta Bancaria agregada exitosamente.		
Resultado obtenido: Nueva Cuenta Bancaria agregada exitosamente.		

Tabla 20

Agregar Asiento Contable Tipo

Agregar Asiento Contable Tipo	Rol	Identificador
	Contador	03
Descripción: La prueba consiste en agregar un Asiento Contable Tipo en el Sistema SIGFA.		
Prerrequisitos: Haber iniciado sesión como Contador.		
Pasos: 1. Ingresar a la pantalla de Asiento Contable Tipo. 2. Ingresar a la opción de agregar Asiento Contable Tipo. 3. Ingresar todos los datos necesarios para agregar un Asiento Contable Tipo 4. Si los datos son incorrectos o el Asiento Contable Tipo ya existe, el sistema desplegara un mensaje de error. 5. Si los datos son correctos el sistema guardara el Asiento Contable Tipo.		
Resultado esperado: Nuevo Asiento Contable Tipo agregado exitosamente.		
Resultado obtenido: Nueva Asiento Contable Tipo agregado exitosamente.		

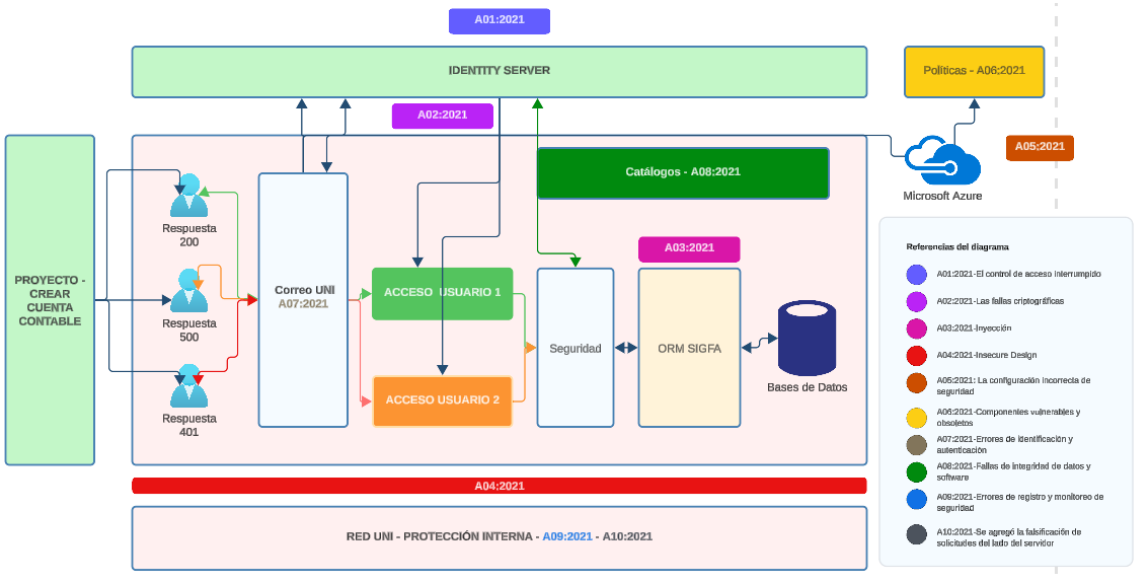
6.4.2. Pruebas No Funcionales (Pruebas de seguridad):

El sistema SIGFA maneja información financiera y administrativa sensible, lo que lo convierte en un objetivo potencial de múltiples tipos de ataques. Por lo tanto, es fundamental asegurar que tanto la infraestructura como la lógica del FrontEnd y Backend estén protegidas frente a las vulnerabilidades más comunes descritas en el OWASP Top 10.

En el siguiente diagrama se describe un entorno de prueba para el aplicativo del Backend SIGFA. Además, se detalla la Infraestructura de seguridad ya implementada y cómo cada componente se interrelaciona para proteger el entorno. Es importante mencionar, que las pruebas realizadas fueron sobre los Códigos de estado de respuesta HTTP que brinda el servidor, y el entorno que protege estas respuestas.

Figura 27

Diagrama de Pruebas de Seguridad para el Backend SIGFA según OWASP Top 10



A continuación, se muestra el proceso de las pruebas unitarias, tomando como punto de partida el punto A01:2021 – Control de Acceso Rotos del Top 10 de OWASP que está asociado a los requerimientos no funcionales 1.1 y 1.2. En el cual se abordaron los siguientes puntos:

1. Uso de Intercambio de recursos entre origen (CORS).
2. Servicio de navegación privado, permitido en Red UNI.
3. La duración del token, cuenta con un ciclo de vida de 8 horas, a partir de su inicio de sesión. Esta sesión es válida por el OAuth2 de Microsoft.
4. Modelo de acceso, basado por propiedades (Pantalla y Acciones) suministrado por el Sistema Institucional Seguridad y Autorización (SISA).

Pruebas de seguridad asociadas a autenticación y autorización:

El código a continuación verifica, que el usuario que realiza el Login, tenga los permisos para acceder al sistema financiero SIGFA. Esto siguiendo el tenant de

OWASP TOP 10. Además, si la respuesta es correcta se comparte el Token de la sesión. Los resultados de esta prueba fueron exitosos.

Figura 28

Petición de Inicio de Sesión (Login), con credenciales institucionales de prueba

```
[Test]
[TestCase(HttpStatusCode.OK)]
[TestCase(HttpStatusCode.InternalServerError)]
[TestCase(HttpStatusCode.Conflict)]
[TestCase(HttpStatusCode.Unauthorized)]
| 0 referencias
public void Test1(HttpStatusCode code)
{

    var builder = WebApplication.CreateBuilder();
    using (var client = new HttpClient() )
    {
        client.BaseAddress = new Uri(builder.Configuration.GetSection("ApiReferencia:ApiSIGFA").Value);
        var login = new Login()
        {
            usuario = builder.Configuration.GetSection("RecursoTesting:correo").Value,
            clave = builder.Configuration.GetSection("RecursoTesting:credencial").Value,
        };
        var json = JsonConvert.SerializeObject(login);
        var data = new StringContent(json, Encoding.UTF8, "application/json");
        var token =  client.PostAsync(
        (builder.Configuration.GetSection("ApiReferencia:ApiSeguridad").Value
            + "/oauth2/autorize"), data
       ).Result;
        string Ttoken = "";
        string result =  token.Content.ReadAsStringAsync().Result;
        Ttoken = JsonConvert.DeserializeObject<Token>(result)!.codigo;
        client.DefaultRequestHeaders.Add("Authorization", "Bearer " + Ttoken.ToString());
        client.DefaultRequestHeaders.Add("entidad", builder.Configuration.GetSection("Sistema:ClaveSecretaOauth").Value);
```

Pruebas de seguridad asociadas a gestión de sesiones:

La figura a continuación muestra los test de pruebas unitarias para registrar una cuenta contable asegurando el acceso de roles específicos. En esta prueba se validan las cuatro posibles respuestas más comunes que retorna el Endpoint de cuenta contables (OK, InternalServerError, Conflict, Unauthorized).

Figura 29

Pruebas unitarias de gestión de sesiones al registrar cuentas contables

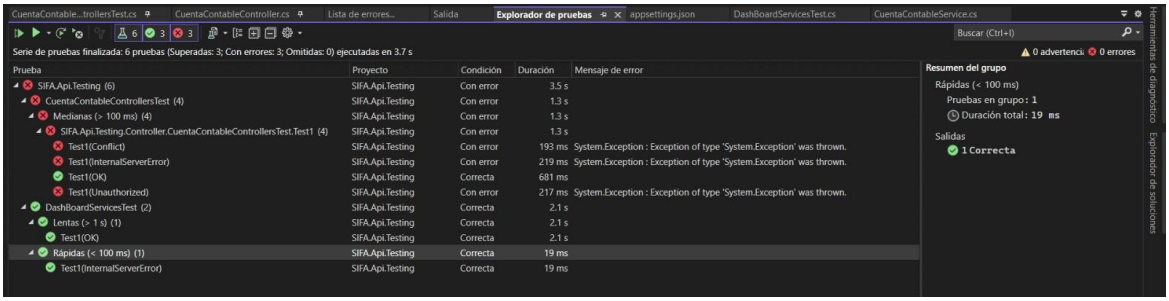


Figura 30

Validación de las cuatro posibles respuestas (OK, InternalServerError, Conflict, Unauthorized)

```
var cuentaContableResponse = client.PostAsync(
     (builder.Configuration.GetSection("ApiReferencia:ApiSIGFA").Value
         + builder.Configuration.GetSection("RecursoTesting:endPoint").Value), dataRequest
         ).Result;
result = cuentaContableResponse.Content.ReadAsStringAsync().Result;
var respuestaServidor = JsonConvert.DeserializeObject<Respuesta>(result)!.mensaje;
var respuestaServidorCode = JsonConvert.DeserializeObject<Respuesta>(result)!.statusCode;
if(respuestaServidorCode == 201)
{
    Assert.IsTrue(BigInteger.Parse(respuestaServidor) > 0, "La Cuenta Contable se ha registrado correctamente.");
    mapper.Map<CuentaContableCreateTest>(cuentaContableRequest).ReturnsForAnyArgs(cuentaContableRequest);
}
else if (respuestaServidorCode == 409)
{
    Assert.That(respuestaServidor.Equals("El número de cuenta contable ya se encuentra asignado."));
    mapper.Map<CuentaContableCreateTest>(cuentaContableRequest).ThrowsForAnyArgs(error => { throw new Exception(); });
}
else
{
    mapper.Map<CuentaContableCreateTest>(cuentaContableRequest).ThrowsForAnyArgs(error => { throw new Exception(); });
}
```

Por último, se muestra el código donde se validan las cuatro respuestas comunes mencionadas que puede retornar, al registrar una cuenta contable, con su usuario origen, que hace uso del Token de sesión para realizar la petición con éxito.

Evidencias adicionales sobre las pruebas de seguridad realizadas se incluyen en el Anexo 10.3.2, debido a limitaciones de espacio.

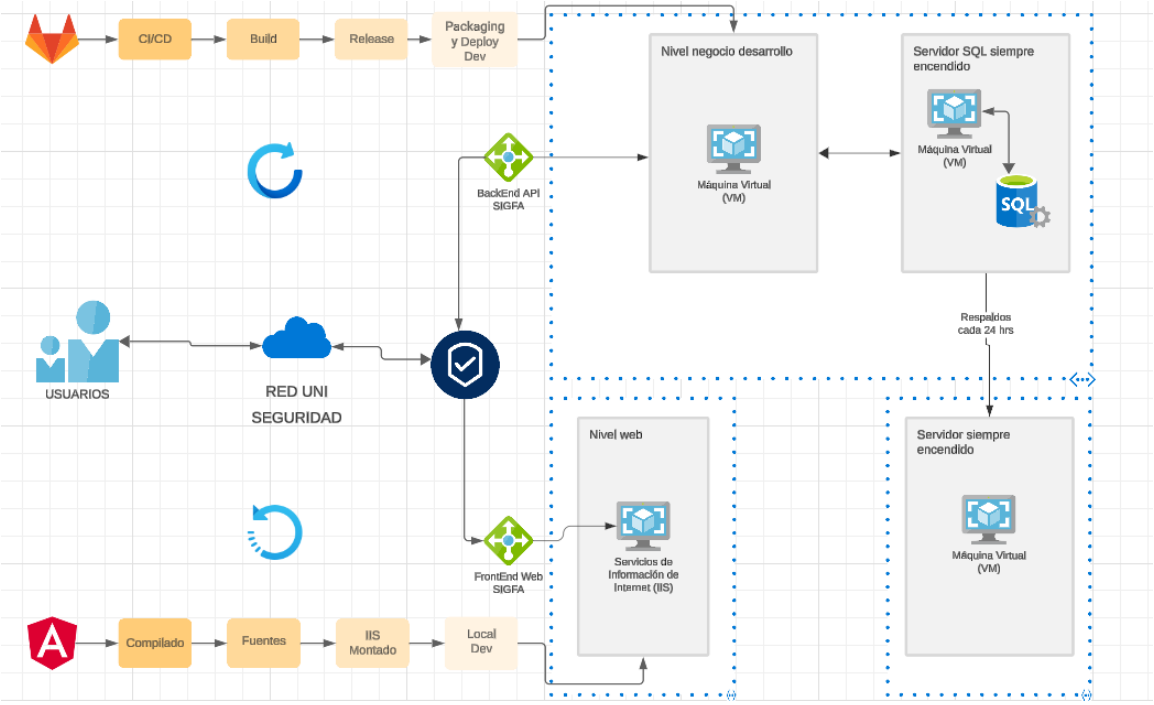
6.5. Despliegue de versión finalizada del SIGFA en entorno de prueba

A continuación, en la siguiente figura se detallan los despliegues en el entorno de prueba UNI. Iniciando con el despliegue del BackEnd, que es preparado por el

GITLAB, después de actualizar el repositorio (git push origin dev) con la versión aprobadas. Para el despliegue, del FrontEnd, el CLI de Angular, nos permite generar el compilado (archivos, html, js y css, assets) a publicar, y alojarlo en el IIS (Servicios de Información de Internet) para su posterior salida a la Red local. Para la base de datos, se estableció una VM con el motor de SQL SERVER, para los planes de mantenimientos existe otra máquina que almacena esta copia de seguridad cada 24 horas. Además, la comunicación de estos ambientes es garantizada por la Red UNI.

Figura 31

Diagrama de Despliegue para Backend y el FrontEnd SIGFA



A continuación, se muestra el entorno de pruebas donde se está ejecutando el sistema SIGFA y el repositorio donde se ejecuta el proceso de ejecución del Backend.

Figura 32

Entorno de pruebas en el IIS.

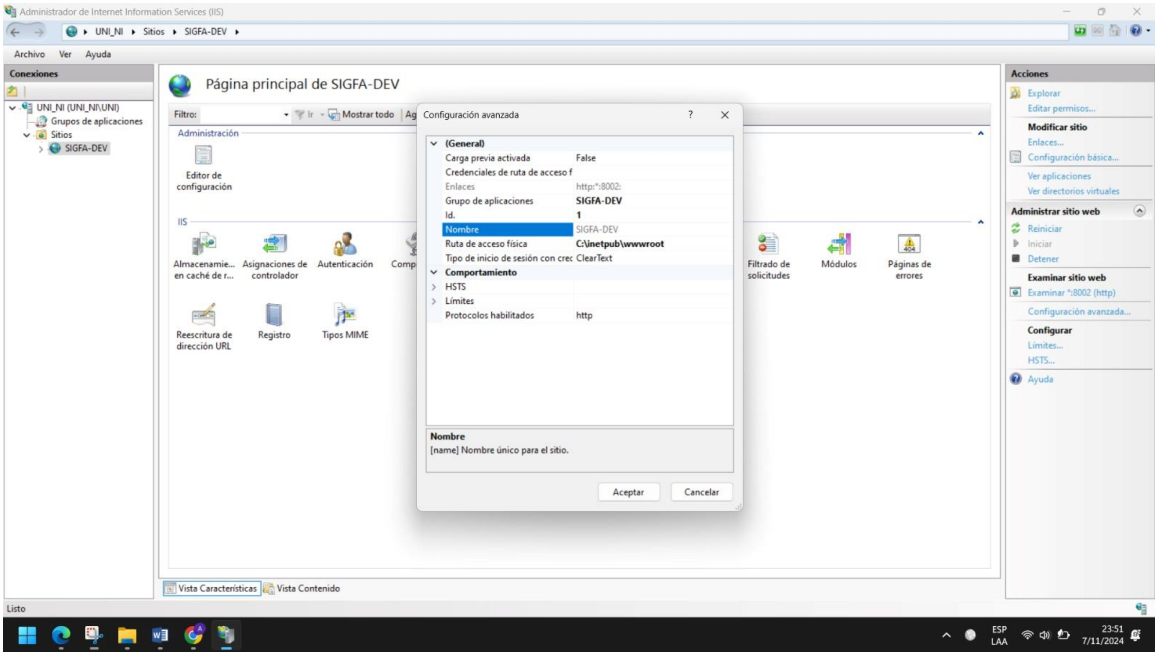
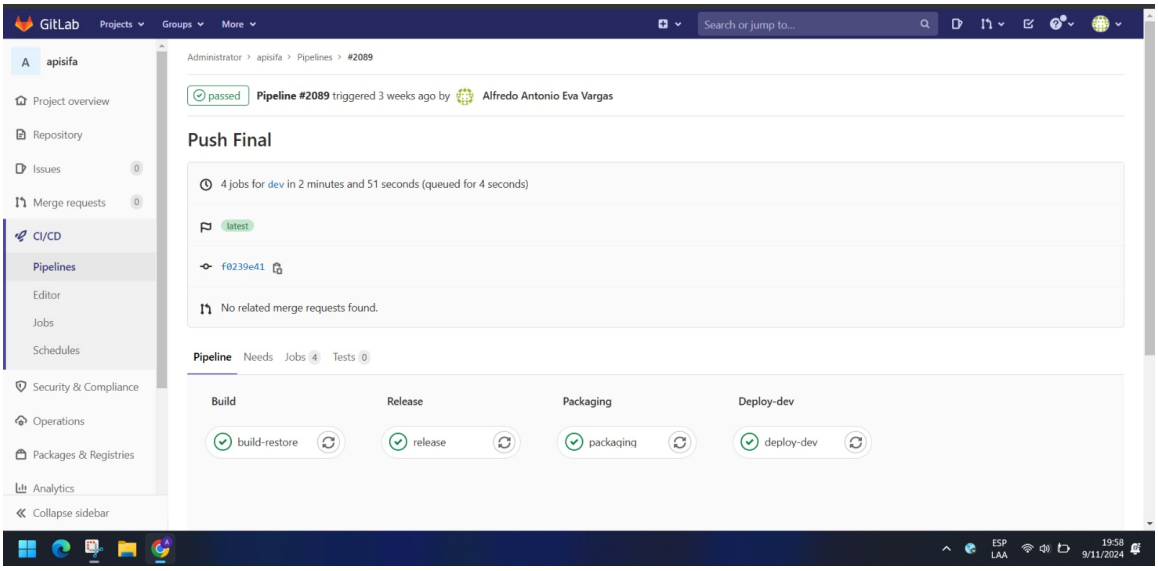


Figura 33

Repositorio GITLAB donde se ejecuta el Backend



VII. Conclusiones

El desarrollo del SIGFA en la UNI se ha realizado exitosamente de acuerdo a los objetivos establecidos en este trabajo. Mediante la aplicación de la metodología SCRUM, y tecnologías web relevantes y adecuadas para la institución se ha desarrollado una aplicación informática que ofrece los elementos para satisfacer las necesidades urgentes existentes respecto al manejo de las gestiones financieras y administrativas de la UNI, incluyendo la centralización de la información administrativa financiera, la parametrización de cuentas contables y la creación automática de reportes en tiempo real, las cuales mejorarán significativamente los procesos internos, ahorrando tiempo y reduciendo riesgos financieros. Esto ha sido validado por los tomadores de decisiones institucionales que han sido delegados con esta responsabilidad, los cuales han dado seguimiento a todo el proceso, y la debida aprobación del resultado alcanzado (como se observa en las cartas de aceptación y documentación asociada incluidas en anexos “10.1. Cartas de soporte del proyecto”), obteniéndose una aceptación positiva del producto resultante.

VIII. Recomendaciones

A raíz del desarrollo de este proyecto, el BPO y los stakeholders observaron algunas recomendaciones que consideraron pertinentes para futuras mejoras o actualizaciones del sistema web, dado el alcance planteado para este proyecto:

1. Incorporar un filtro por Programas en el Árbol de Cuentas Contables para permitir una visión segmentada y detallada de las cuentas, facilitando una gestión financiera más eficiente y precisa.
2. Implementar una sección dedicada a la visualización detallada de los beneficiarios, incluyendo información específica y relevante sobre estudiantes, trabajadores y proveedores, mejorando así la transparencia y el control sobre las transacciones realizadas con cada grupo.
3. Además, se recomienda la utilización de Socket.IO para el dashboard, lo que permitirá la actualización en tiempo real de la información cuando se produzcan cambios en los balances, mejorando la toma de decisiones y la supervisión continua del estado financiero.

IX. Referencias

UNIVERSIDAD DE NEGOCIOS ISEC . (20 de Junio de 2022). *uneg.edu.mx/.* Obtenido de uneg.edu.mx/: https://uneg.edu.mx/que-hace-el-departamento-de-finanzas/

aws.amazon.com. (s.f.). Obtenido de aws.amazon.com: https://aws.amazon.com/es/microservices/#:~:text=Los%20microservicios%20son%20un%20enfoque,servicios%20son%20equipos%20peque%C3%B1os%20independientes.

Concavales, M. J. (13 de 10 de 2021). *www.hiberus.com.* Obtenido de www.hiberus.com: https://www.hiberus.com/crecemos-contigo/que-es-angular-y-para-que-sirve/

docs.npmjs.com/. (s.f.). Obtenido de docs.npmjs.com/: https://docs.npmjs.com/about-npm

earn-microsoft-com. (04 de 05 de 2023). Obtenido de earn-microsoft-com.: https://learn-microsoft-com.translate.goog/en-us/dotnet/csharp/tour-of-csharp/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es-419&_x_tr_pto=sc

Fernandez, I. C. (Octubre de 2010). *owasp.org.* Obtenido de owasp.org: https://owasp.org/www-pdf-archive/Introduccion_a_la_OWASP.pdf

Flores, F. (22 de 07 de 2022). *openwebinars.net.* Obtenido de openwebinars.net: https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/

Gamboa, D. (02 de Noviembre de 2020). *www.stelorder.com.* Obtenido de www.stelorder.com: https://www.stelorder.com/blog/funciones-contable/

gitlab.com. (s.f.). Obtenido de gitlab.com: https://docs.gitlab.com/

learn-microsoft-com. (s.f.). Obtenido de learn-microsoft-com: https://learn-microsoft-com.translate.goog/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16&_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es-419&_x_tr_pto=sc

learn-microsoft-com. (15 de 09 de 2019). Obtenido de learn-microsoft-com: https://learn-microsoft-com.translate.goog/en-us/dotnet/framework/data/adonet/ef/overview?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es-419&_x_tr_pto=sc

Montiel, Y. (10 de Enero de 2023). Entrevista sobre el SIFA a Responsable de Oficina de Sistemas y Servidores (NIC.NI). (A. Eva, Entrevistador)

OWASP. (2021). *owasp.org*. Obtenido de owasp.org: https://owasp.org/Top10/es/

Schwaber, K., & Sutherland, J. (1 de Noviembre de 2020). La Guía de Scrum. *La Guía Definitiva de Scrum: Las Reglas del Juego*. Scrum Guide. Obtenido de https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf

Scrum.org. (2023). *What is Scrum?* Obtenido de Scrum.org: The home of Scrum: https://www.scrum.org/learning-series/what-is-scrum

Solis, M. O. (2023). Director de finanzas UNI. (R. Acevedo, Entrevistador)

Soluciones Alegra S.A.S. (01 de Octubre de 2021). *escuela-emprendedores.alegra.com.* Obtenido de escuela-emprendedores.alegra.com: https://escuela-emprendedores.alegra.com/contabilidad-facil/importancia-de-un-sistema-contable-en-costa-rica/#:~:text=Los%20sistemas%20contables%20son%20software,el%20tiempo%20de%20manera%20significativa.

visualstudio.microsoft.com. (s.f.). Obtenido de visualstudio.microsoft.com: https://visualstudio.microsoft.com/es/#:~:text=IDE%20de%20Visual%20Studio%20es,%2C%20finalmente%2C%20publicar%20una%20aplicaci%C3%B3n.

www.danysoft.com/fast-report. (s.f.). Obtenido de www.danysoft.com/fast-report: https://www.danysoft.com/fast-reports/#:~:text=Fast%20Reports%20es%20una%20herramienta,la%20productividad%20de%20los%20desarrolladores.

www.fp-santagema.es. (20 de 09 de 2022). Obtenido de www.fp-santagema.es: https://www.fp-santagema.es/funciones-auxiliar-de-tesoreria-empresa

www.microsoft.com. (s.f.). Obtenido de www.microsoft.com: https://www.microsoft.com/es/microsoft-365/project/project-management-software

X. ANEXOS

10.1. Cartas de soporte del proyecto

Figura 34

Carta de solicitud dirigida el Ing. Manuel Godínez, Director de la DTI para utilizar y presentar el sistema SIGFA para tesis monográfica

Managua, 11 de julio del 2024

Estimado Ing. Manuel Godínez Duriez:
Director
División de Tecnología de la Información

Nos dirigimos a usted para solicitar su autorización, para presentar la primera etapa del **Desarrollo del Sistema Integrado de Gestión Financiero Administrativo (SIGFA)** como Forma de Culminación de Estudio, que hemos hecho uso como tema monográfico (Autorización de anterior Dirección) denominado, **Desarrollo de Sistema Web de Gestión Financiero Administrativo (SIGFA) para la Universidad Nacional de Ingeniería**.

Es importante mencionar, que se están tomando todas las normas de seguridad, de los sistemas de información Institucionales en la etapa de desarrollo, utilizando datos de prueba, de baja criticidad y entornos controlados.

Agradezco su consideración a nuestra solicitud y estamos a su disposición para discutir cualquier aspecto adicional relacionado con nuestra solicitud.

Atentamente,

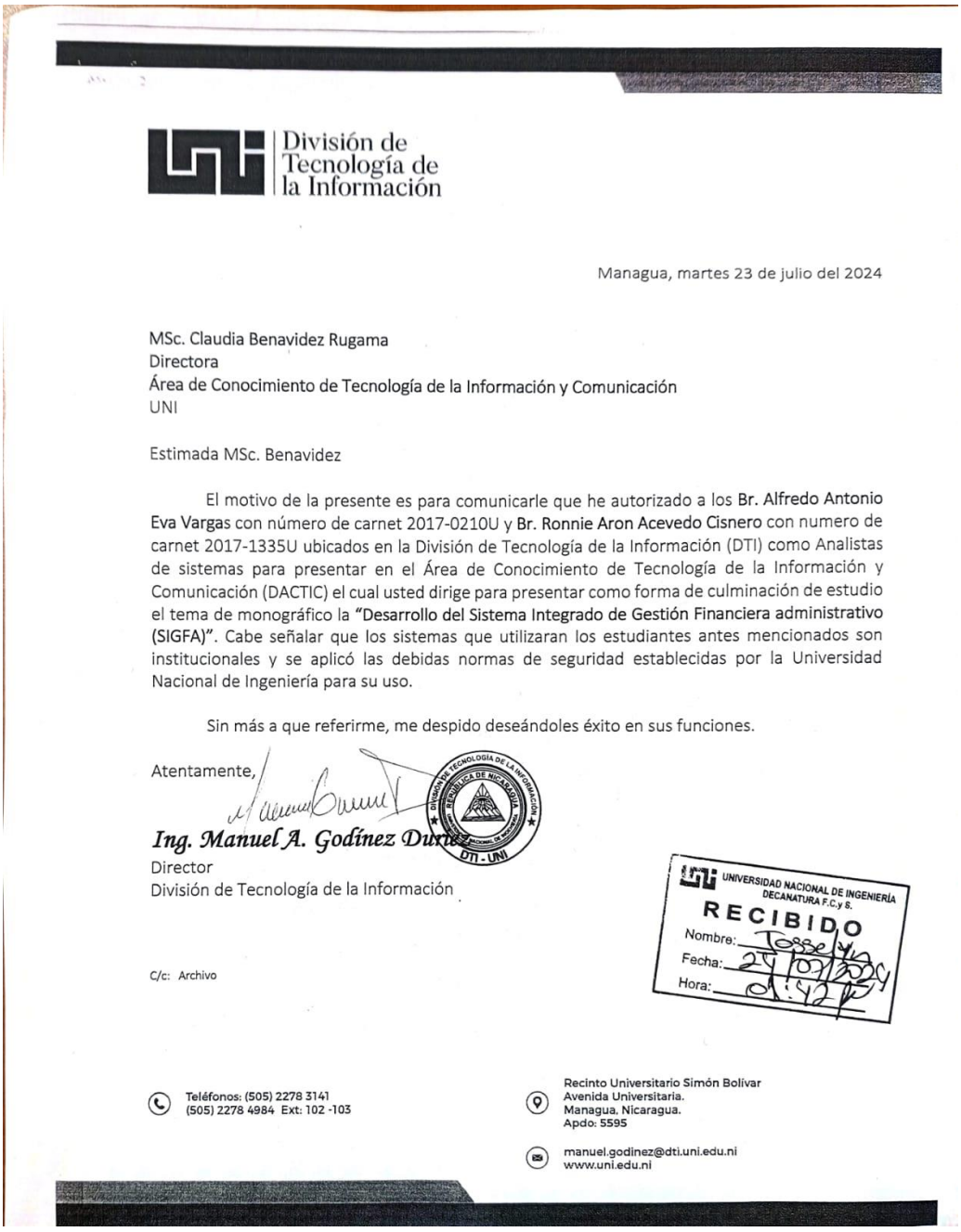
Br. Ronnie Aaron Acevedo Cisnero No carnet 2017- 1335U
Estudiante de la Carrera de Ing. en Computación
UNIVERSIDAD NACIONAL DE INGENIERÍA

Br. Alfredo Antonio Eva Vargas No carnet 2017-0210U
Estudiante de la Carrera de Ing. en Computación
UNIVERSIDAD NACIONAL DE INGENIERÍA

UNIVERSIDAD NACIONAL DE INGENIERÍA
DIRECCIÓN NIC.NI
RECIBIDO
Nombre: Claudia.
Fecha: 11/07/2024
Hora: 10:23.

Figura 35

Carta de aceptación de parte del Ing. Manuel Godínez, Director de la DTI dirigida a MSc. Claudia Benavidez, para utilizar y presentar el sistema SIGFA para tesis monográfica



División de Tecnología de la Información

Managua, martes 23 de julio del 2024

MSc. Claudia Benavidez Rugama
Directora
Área de Conocimiento de Tecnología de la Información y Comunicación
UNI

Estimada MSc. Benavidez

El motivo de la presente es para comunicarle que he autorizado a los **Br. Alfredo Antonio Eva Vargas** con número de carnet 2017-0210U y **Br. Ronnie Aron Acevedo Cisnero** con numero de carnet 2017-1335U ubicados en la División de Tecnología de la Información (DTI) como Analistas de sistemas para presentar en el Área de Conocimiento de Tecnología de la Información y Comunicación (DACTIC) el cual usted dirige para presentar como forma de culminación de estudio el tema de monográfico la **"Desarrollo del Sistema Integrado de Gestión Financiera administrativo (SIGFA)"**. Cabe señalar que los sistemas que utilizaran los estudiantes antes mencionados son institucionales y se aplicó las debidas normas de seguridad establecidas por la Universidad Nacional de Ingeniería para su uso.

Sin más a que referirme, me despido deseándoles éxito en sus funciones.

Atentamente,

Ing. Manuel A. Godínez Durán
Director
División de Tecnología de la Información

DIVISIÓN DE TECNOLOGÍA DE LA INFORMACIÓN REPÚBLICA DE NICARAGUA UNIVERSIDAD NACIONAL DE INGENIERÍA DTI - UNI

UNIVERSIDAD NACIONAL DE INGENIERÍA
DECANATURA F.C.y S.
RECIBIDO
Nombre: Josselyn
Fecha: 24/07/2024
Hora: 01:42 p

C/c: Archivo

Teléfonos: (505) 2278 3141
(505) 2278 4984 Ext: 102 -103

Recinto Universitario Simón Bolívar
Avenida Universitaria.
Managua, Nicaragua.
Apdo: 5595

manuel.godinez@dti.uni.edu.ni
www.uni.edu.ni

Figura 36

Carta de aceptación y validación del sistema SIGFA por parte de MSc. Pablo David Trejos Aguilar, Director de la División de Finanzas de la Universidad Nacional de Ingeniería (UNI)

División de Finanzas

Estimados Br. Alfredo Eva y Br. Ronnie Acevedo:

Me dirijo a ustedes en mi calidad de Director de la **División de Finanzas de la Universidad Nacional de Ingeniería,** con el propósito de notificarle formalmente la aceptación y validación del proyecto de software financiero, denominado **Sistema Integrado de Gestión Financiera Administrativo (SIGFA) para la Universidad Nacional de Ingeniería**. Nos complace informarles que, tras la revisión de los módulos presentados que se describen en la Tesis Monográfica, son suficientes para cumplir con la necesidad Operacional Contable Institucional, establecidas en los acuerdos de reuniones con la **División de Finanzas y la División de Tecnología de la Información**.

MSc. Pablo David Trejos Aguilar
Universidad Nacional de Ingeniería UNI
Director de la División de Finanzas
pablo.trejos@ddf.uni.edu.ni

Teléfono: (505) 2278 1453 / 54 Ext: 108 - 109

Recinto Universitario Simón Bolívar
Avenida Universitaria.
Managua, Nicaragua.
Apdo: 5595

manuel.olivares@finanzas.uni.edu.ni
www.uni.edu.ni

Figura 37

Carta de descripción por parte del Ing. Rodrigo Diaz, Responsable de Oficinas de Sistemas de la DTI, sobre entorno de prueba provisto por el DTI para la presentación del sistema SIGFA para la predefensa y defensa de la tesis monográfica

División de Tecnología de la Información

Managua, Nicaragua 16 julio 2024

Hoja descriptiva de las especificaciones del despliegue de prueba del Tema Monográfico de nombre: **Sistema web para llevar a cabo las gestiones financieras administrativas de la UNI (SIGFA)**, provisto por la División de la Tecnología de la Información (DTI).

1. Acceso al sistema de información (Usuarios):
   a. Acceso a dos usuarios por medio del correo institucional para realizar las operaciones necesarias en el entono de prueba provisto.
2. Entorno FrontEnd (Interfaces):
   a. Configuración de **Internet Information Services (IIS)** en entorno local (http://localhost:8002), estos se adecuaron en dos equipos proporcionado por la DTI a los analistas de sistemas que integran este tema monográfico, estos entornos se construyeron en conjunto con el encargado de estos servicios.
3. Entorno BackEnd (Servidor):
   a. API nube, se permitirá consumir este servicio de forma segura.

Teléfonos: (505) 2278 3141
(505) 2278 4984 Ext: 102 -103

Recinto Universitario Simón Bolívar
Avenida Universitaria.
Managua. Nicaragua.
Apdo: 5595

manuel.godinez@dti.uni.edu.ni
www.uni.edu.ni

Figura 38

División de Tecnología de la Información

Dirigido a:

Br. Ronnie Aaron Acevedo Cisnero No. Carnet: 2017-1335U
Analista de Sistema - DTI
Estudiante de la Carrera de Ing. en Computación
UNIVERSIDAD NACIONAL DE INGENIERÍA

Br. Alfredo Antonio Eva Vargas **No. Carnet: 2017-0210U**
Analista de Sistema - DTI
Estudiante de la Carrera de Ing. en Computación
UNIVERSIDAD NACIONAL DE INGENIERÍA

Atentamente,

Nota: Debido a que es un aplicativo institucional, está en proceso de prueba, una vez terminado esta verificación se les solicitara a las autorizadas el permiso de despliegue del mismo.

Ing. Rodrigo Diaz
Responsable de Oficina de Sistema - DTI
UNIVERSIDAD NACIONAL DE INGENIERIA

Teléfonos: (505) 2278 3141
(505) 2278 4984 Ext: 102 -103

Recinto Universitario Simón Bolívar
Avenida Universitaria.
Managua, Nicaragua.
Apdo: 5595

manuel.godinez@dti.uni.edu.ni
www.uni.edu.ni

10.2. Detalle de Sprints 1, 3, 4, 5, 6, 9 y 11

A continuación, se incluye el detalle de los Sprints 1, 3, 4, 5, 6, 9 y 11.

10.2.1. Sprint 1

Ya con las herramientas de desarrollo establecidas, la arquitectura que se utilizará, el sitio en la nube donde se almacenaran los repositorios del proyecto, los roles definidos de cada desarrollador, la seguridad que se utilizará para las pruebas, lo siguiente era la planificación del proyecto.

10.2.1.1. Sprint Planning

El resultado principal del sprint planning es el sprint backlog. El Sprint Backlog incluye al responsable a cargo de realizar la tarea asignada, de acuerdo al grado de esfuerzo. Si la tarea tiene un esfuerzo nivel tres (menor nivel de complejidad), se le asignó a un solo responsable; de lo contrario, si la tarea tuvo un esfuerzo uno (máximo nivel de complejidad) se le asignaron dos responsables a dicha tarea. Este primer Sprint Planning se enfocará en el módulo para Gestionar Periodos Contables. El Product Backlog de este Sprint indica el desarrollo de las historias de usuario que se incluyen en la Tabla 21.

Diariamente durante el periodo establecido para cada Sprint (21 días), se realizaron los Daily Scrum, pequeñas reuniones diarias a primeras horas de la mañana, esto con el fin de informar cómo iba el desarrollo de las tareas asignadas. Si tenían alguna dificultad, retroalimentación, o si se ya habían terminado y así ayudar a otro desarrollador que tuviera alguna dificultad.

Tabla 21: Sprint Backlog para el Sprint 1

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Responsable	Estimación
1	Contador	Generar Periodos Contables Generales.	Tener estados financieros claros anuales.	Jorge Torres; Alfredo Eva	4 días
2	Director	Editar los Periodos Generales.	Tener la opción de arreglar errores financieros.	Carlos Martínez; Alfredo Eva	2 días
3	Director	Cambiar estado de registrado a activo en el Periodo General.	Tener control de los estados de los Periodos.	Carlos Martínez; Alfredo Eva	2 días
4	Director	Cambiar estado de activo a cerrado en Periodo General.	Tener control de los estados de los Periodos.	Yasser Montiel; Alfredo Eva	2 días
5	Contador	Agregar Periodos Específicos.	Tener estados financieros claros mensuales.	Carlos Martínez; Alfredo Eva	3 días
6	Contador	Vincular los Periodos Específicos a un Periodo General.	Enlazarlos y así tener control de los periodos.	Carlos Martínez; Alfredo Eva	3 días
7	Director	Cambiar estado de registrado a activo en los Periodos Específicos.	Tener la opción de activar o desactivar los estados.	Carlos Martínez; Alfredo Eva	1 día
8	Director	Cambiar estado de activo a precierre en los Periodos Específicos.	Tener la opción de activar o desactivar los estados.	Yasser Montiel; Alfredo Eva	1 día

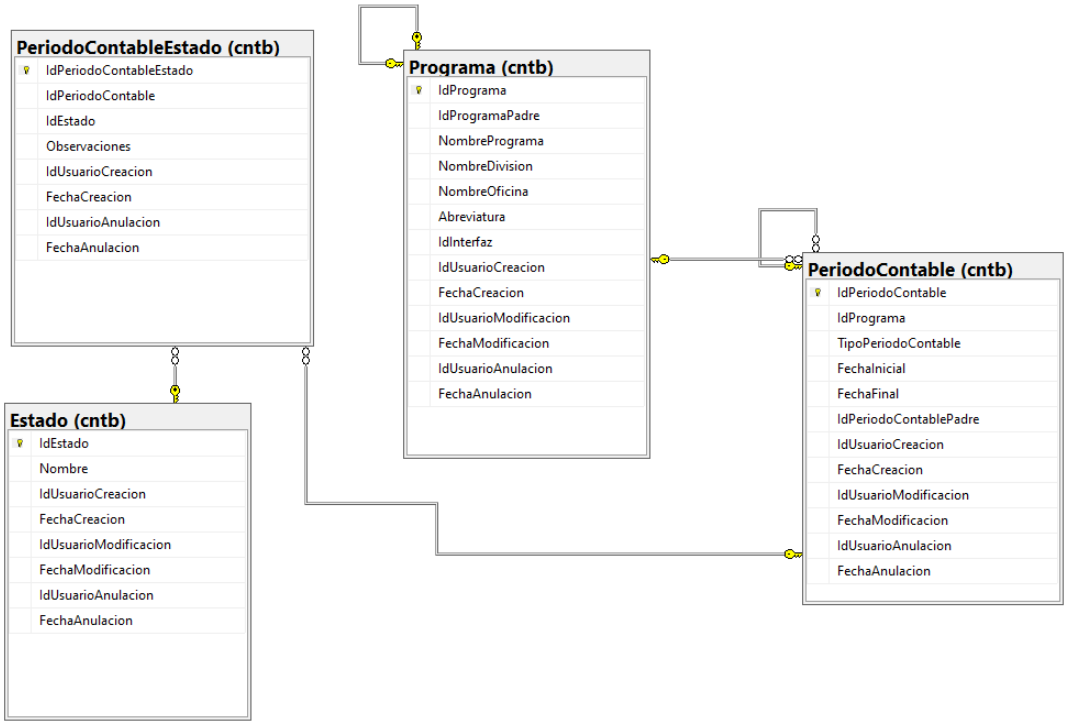
N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Responsable	Estimación
9	Director	Cambiar estado de precierre ha cerrado en los Periodos Específicos.	Tener control de los estados de los Periodos Específicos.	Yasser Montiel; Alfredo Eva	1 día
10	Contador y Director	Mostrar en una tabla los Periodos Contables con los datos relevantes.	Tener una visualización más amplia y clara.	Yasser Montiel; Alfredo Eva	1 día
11	Contador y Director	Buscar un Periodo Contable por rango de fecha.	Buscar de una manera más fácil periodos contables.	Ronnie Acevedo	1 día

10.2.1.2. Resumen del proceso de desarrollo de software del Sprint 1

El desarrollo de la base de datos se realizó directamente en la herramienta Microsoft SQL Server, teniendo en cuenta las historias de usuario de este Sprint, y las aclaraciones sobre los procesos institucionales que fueron provistas por personal especialista en la materia. El desarrollo de la base de datos de este sprint engloba la creación de la base de datos relacionada con los periodos contables, incluyendo las tablas con sus controladores que a su vez esto genera sus Endpoint; que son utilizados por el Frontend para mostrar la información de los periodos contables. A continuación, en la Fig. 39, se muestra un ejemplo del diseño de la base de datos correspondiente a la historia de usuario 1, 2, 5, 6 del sprint backlog 1.

Figura 39

Tablas creadas para los periodos Contables Generales y Específicos



Continuando el proceso de desarrollo, se realizó la implementación del Backend y Frontend utilizando las herramientas Visual Studio Code con el Framework Angular para Frontend y Visual Studio junto con Entity Framework 6.0 para el Backend. Como ejemplo, en la Fig. 40 se presenta una muestra del código desarrollado para la historia de usuario 11 y 12; posteriormente las interfaces gráficas (Frontend) asociadas.

Figura 40

Código del servicio para obtener los registros de los Periodos General, con su paginación

```
export class PeriodoContableService implements Servicio {
  get(
    let params = new HttpParams();
    const fechaAnual = new Date();
    params = params.set('TipoPeriodoContable', 'G');
    peridoContable?.anio != null && peridoContable?.anio.toString() != ''
      ? (params = params.set('Anio', peridoContable?.anio))
      : (params = params.set('Anio', fechaAnual.getFullYear()));
      peridoContable?.programa != null && peridoContable?.programa.toString() != ''
      ? (params = params.set('NombrePrograma', peridoContable?.programa))
      : null;
    peridoContable?.ordenamiento != null && peridoContable?.ordenamiento != ''
      ? (params = params.set('OrdenarPor', peridoContable?.ordenamiento))
      : null;
    peridoContable?.orientacion != null && peridoContable?.orientacion != ''
      ? (params = params.set('OrientarPor', peridoContable?.orientacion))
      : null;

    //Parametros de la paginacion
    pagionador?.pagina != null
      ? (params = params.set('pagina', pagionador?.pagina))
      : null;
    pagionador?.registroPorPagina != null
      ? (params = params.set(
          'registroPorPagina',
          pagionador?.registroPorPagina
        ))
      : (params = params.set('registroPorPagina', 5));

    //Retorno de arreglo de interfaz tipado
    return this.http.get<PeriodoContable[]>(this.API, {
      headers: {
        Authorization: `Bearer ${this.TOKEN}`,
      },
      params,
    });
```

Figura 41

Código del servicio para registrar, modificar y anular un Periodo General

```
//Metodo post para insertar un registro nuevo
post(periodo: PeriodoContable): Observable<RespuestaServidor> {
    return this.http.post<RespuestaServidor>(this.API, periodo, {
    headers: {
      Authorization: `Bearer ${this.TOKEN}`,
    },
  });
}
//Metodo update para actualizar un registro existente
update(periodo: PeriodoContable, id: number): Observable<RespuestaServidor> {
  return this.http.put<RespuestaServidor>(`${this.API}/${id}`, periodo, {
    headers: {
      Authorization: `Bearer ${this.TOKEN}`,
    },
  });
}
//Delete metodo para eliminar (borrado logico) un registro existente de la base de datos
delete(id: number): Observable<RespuestaServidor> {
  return this.http.delete<RespuestaServidor>(`${this.API}/${id}`, {
    headers: {
      Authorization: `Bearer ${this.TOKEN}`,
    },
  });
}
```

Figura 42

Código del controlador que valida y registra el Periodo General a la base de datos

```
[HttpPost]
0 referencias
public async Task<IActionResult> PostAsync([FromBody] PeriodoContableCreate c, [FromHeader] String[] entidad)
{
  try
  {

      if (!ModelState.IsValid) return BadRequest(ModelState);
      if (entidad.IsNullOrEmptyArray())
           return StatusCode(StatusCodes.Status401Unauthorized,
                 new { Mensaje = "Error, no autorizado." });
      ConsultaPermiso cp;
      if (c.TipoPeriodoContable.Equals("E"))
       {
              cp = new ConsultaPermiso(_configuration)
              {
                  Accion = _configSecurity.HomologacionAccion(Accion.Guardar),
                  Pantalla = _configSecurity.HomologacionPantalla(Pantalla.PeríodoEspecifico),
                  Entidades = entidad
              };
      }
      else
      {
             cp= new ConsultaPermiso(_configuration)
              {
                  Accion = _configSecurity.HomologacionAccion(Accion.Guardar),
                  Pantalla = _configSecurity.HomologacionPantalla(Pantalla.PeríodoGeneral),
                  Entidades = entidad
              };
      }

      var permiso = await _httpPermisoService.VerificarPermisoAsync(cp);
      if (!permiso.TienePermiso)
          return StatusCode(StatusCodes.Status403Forbidden,
                 new { Mensaje = "Error, no tiene asignado los permisos necesarios para realizar la acción." });
       c.IdUsuarioCreacion = await _usuarioService.ObtenerIdUsuarioAsync(Request.ObtenerUsuarioGuidHeader());

     _periodoContableService.InsertarPeriodoContable(c);

   string mensaje = c.TipoPeriodoContable.ToUpper().Equals("G")
     ? "El periodo contable general se ha guardado exitosamente."
     : "Los periodos contables especificos se han guardado exitosamente.";

   return StatusCode(StatusCodes.Status201Created, new { Mensaje = mensaje });
  }
```

De igual manera se diseñaron las pantallas relacionadas con los Periodos Contables (ver Fig. 43 y 44 con ejemplos de las pantallas realizadas). Estos diseños, así como la jerarquía de las pantallas primeramente se realizaron en papel para luego implementarlos en la herramienta Visual Studio Code.

El Backend va de la mano con el Frontend, porque conforme iban saliendo las necesidades de consumir o mostrar algún dato se realizaban los Endpoint necesarios para completar el proceso.

Los avances del módulo fueron subidos al repositorio GitLab, una vez estos eran revisados y aprobados por el Ing. Carlos Martínez pasaban a la rama principal del proyecto.

Figura 43

Pantalla principal donde se muestran los Periodos Generales

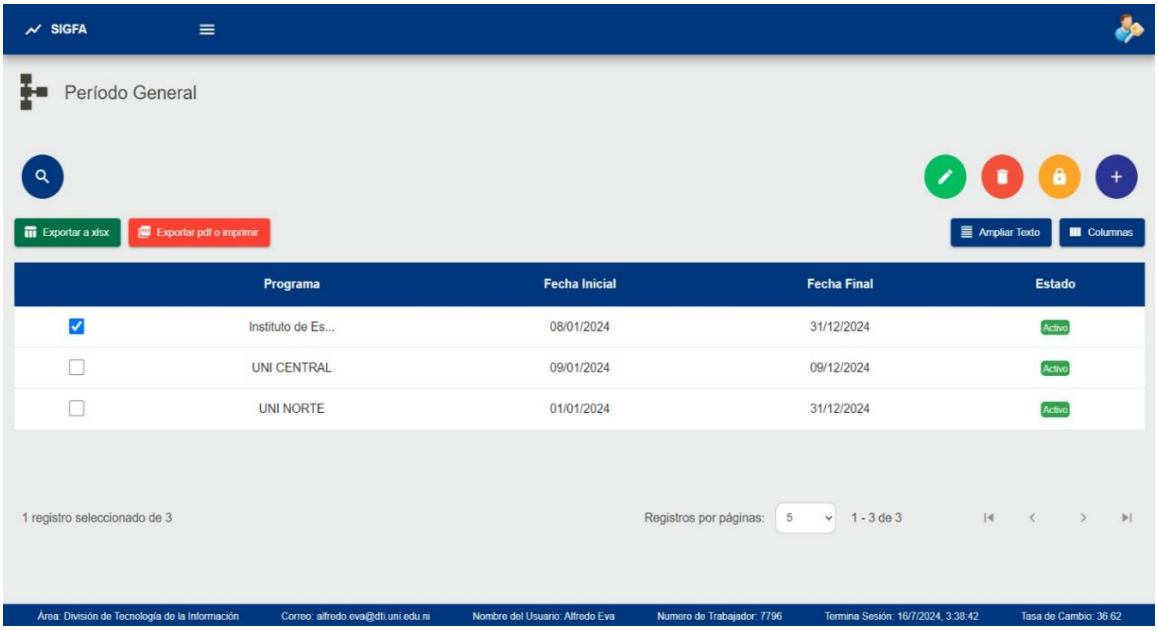
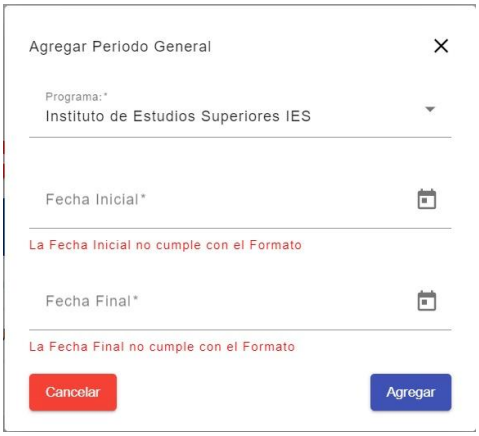


Figura 44

Pantalla emergente para agregar un nuevo Periodo General



10.2.1.3. Revisión del Sprint 1: entrega del producto terminado, Periodos Contables Generales y Específicos

Una vez finalizado el desarrollo conforme a lo establecido en el Product Backlog, se procedía a presentar el producto al Product Owner MSc. Pablo David Trejos y el stakeholder Lic. Gustavo Adolfo Silva Bermúdez. Estos responsables evaluaban el trabajo entregado, documentando los acuerdos alcanzados vía correo electrónico, con los acuerdos alcanzados y se discutía cualquier pregunta, duda o posible mejora a aplicar.

El equipo de desarrollo realizó la demostración y entrega de este primer sprint, el cual era todo el desarrollo de los periodos contables generales en el sistema SIGFA, cabe mencionar que este sprint es un módulo de todo el sistema SIGFA. Al momento de realizar la entrega, el módulo es completamente funcional; además, se cumplió la entrega en tiempo y forma según planificación. El módulo resultante abarca las siguientes funcionalidades.

1. Creación de nuevos Periodos Contables Generales.
2. Editar Periodo Contables General
3. Cambiar estado de activo a precierre en el Periodo Contables General.
4. Cambiar estado de precierre ha cerrado en Periodo Contables General.
5. Agregar Periodo Contables Específico.

6. Editar Periodo Contables Especifico.
7. Vincular los Periodos Específicos a un Periodo Contables General.
8. Cambiar estado de inactivo a activo en los Periodos Contables Específicos.
9. Cambiar estado de activo a Precierre en los Periodos Contables Específicos.
10. Cambiar estado de Precierre ha cerrado en los Periodos Contables Específicos.
11. Buscar Periodos Contable por rango de fecha.

Feedback y retroalimentación

Las partes interesadas realizaron algunas preguntas, entre las cuales destacaron:

1. ¿Cómo se registra en el sistema SIGFA un nuevo Periodo Contable?
2. ¿Cómo se da de baja a un Periodo Contable?
3. ¿Cómo se realiza la búsqueda de un Periodo Contable?
4. ¿Cómo se edita cierta información de un Periodo Contable?

Estas preguntas fueron respondidas mediante la demostración del módulo del sistema desarrollado; así también se proveerá manual de usuario que permita aclarar estas y otras dudas similares al utilizar el sistema.

El Product Owner MSc. Pablo David Trejos dio algunas recomendaciones sobre el diseño de la interfaz, en específico la ubicación y tamaño de los botones, ya que para algunos usuarios padecen de problemas oculares.

Con esto se cerró la reunión de entrega de este primer Sprint lográndose una aceptación positiva, cumpliendo en tiempo y forma con lo solicitado por el Product Owner.

El Scrum Master Ing. Rodrigo Diaz programó la siguiente reunión para la planificación del siguiente Sprint. Mientras tanto, el equipo de desarrollo se encargó de implementar las mejoras indicadas por el Product Owner y stakeholders.

10.2.1.4. Retrospectiva de Sprint 1

El Scrum Master Ing. Rodrigo Diaz reunió al Equipo de Desarrollo para reflexionar sobre el primer Sprint, esto con el fin de identificar fallos como equipo, aspectos en lo que se puede mejorar.

Identificación de fallos

El equipo de desarrollo identificó que hubo algunos fallos con respecto a que algunos miembros del equipo no subían sus cambios de forma consecutiva al repositorio en la nube, es decir GitLab. Lo cual causaba contratiempos a la hora de que el líder del Equipo de Desarrollo quería hacer *merge* de todas las ramas a la principal (rama Master).

También se abordó el tema de la ubicación y tamaño de los botones.

Análisis de la raíz del fallo

Se identificó que el problema principal era la mala práctica por parte de algunos miembros del Equipo de Desarrollo de no subir sus cambios y avances de las tareas asignadas al repositorio remoto GitLab.

Solución

Se estableció como una regla importante que al final de cada día cada miembro debe subir sus cambios al repositorio remoto, no importando si los cambios fuesen mínimos o grandes.

Además, se estableció un patrón de ubicación, de tamaño y de estilo de los iconos de los botones para todas las pantallas que se iban a desarrollar en el sistema SIGFA.

10.2.2. Sprint 3

Este sprint consistió en el desarrollo del módulo de las cuentas bancarias que administrará el sistema SIGFA, este módulo cuenta con las funcionalidades principales: agregar nueva cuenta bancaria, dar de baja a una cuenta bancaria, editar una cuenta bancaria. Estas funcionalidades y otros aspectos adicionales se abordarán a más a detalle en los siguientes puntos.

10.2.2.1. Sprint Planning

Este tercer Sprint Planning se enfocará en el módulo de Cuentas Bancarias. El Product Backlog de este Sprint indica el desarrollo de las historias de usuario que se incluyen en la Tabla 22.

Durante el Sprint 3, con una duración de 21 días, asignado para el módulo de cuentas bancarias, se llevaron a cabo reuniones diarias de Scrum con el objetivo de mantener a todos los miembros del equipo informados sobre el progreso de las tareas asignadas, al igual que se realizó en Sprints anteriores.

Tabla 22**:** Sprint Backlog para el sprint 3

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Responsable	Estimación
1	Contador	Agregar Cuenta Bancaria.	Añadir Cuentas Bancarias al sistema SIGFA.	Jorge Torres; Alfredo Eva	8 días
2	Director	Editar Cuenta Bancaria.	Arreglarlas en caso de que haya cambios.	Carlos Martínez	7 días
3	Director	Anular Cuenta Bancaria.	Ponerlas en estado inhabilitadas en el sistema SIGFA.	Ronnie Acevedo	2 días
4	Contador y Director	Mostrar Cuentas Bancarias.	Pueda visualizar de forma clara y precisa	Alfredo Eva	3 días
5	Director	Buscar Cuenta Bancaria por Cuentas Contable y Banco.	Encontrar información de esa Cuenta Bancaria con rapidez.	Alfredo Eva; Ronnie Acevedo	1 día

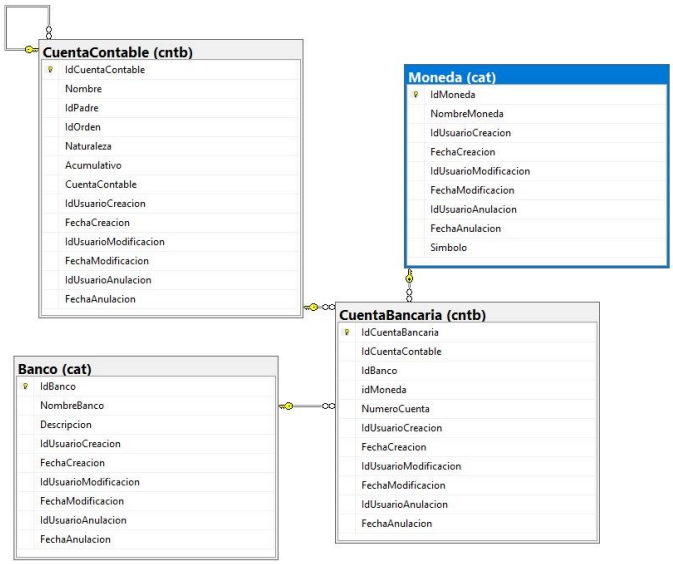
10.2.2.2. Resumen del proceso de desarrollo de software del Sprint 3

De la misma manera que en sprint anteriores, la implementación de las tablas se llevó a cabo directamente en Microsoft SQL Server, con un enfoque centrado en las historias de usuario definidas para este Sprint, así como en las instrucciones detalladas proporcionadas por expertos en los procesos institucionales pertinentes.

Este proceso incluyó la creación de la estructura de las tablas relacionadas con las cuentas contables, moneda, banco y cuentas bancarias, incluyendo las tablas con sus controladores que a su vez esto genera sus Endpoint; que son utilizados por el Frontend para mostrar la información de las cuentas bancarias. Se muestra un ejemplo del diseño de la base de datos correspondiente a la historia de usuario 1, 2, 3, 4, 5 del sprint backlog 3.

Figura 45

Diseño de la base de datos de Cuenta Bancaria



Siguiendo con el proceso de desarrollo, se llevó a cabo la implementación tanto del Backend como del Frontend utilizando herramientas específicas (Frontend: Visual Studio Code junto con el Framework Angular; Backend: Visual Studio en

conjunto con Entity Framework 6.0). Se muestra un ejemplo del código correspondiente a la historia de usuario 1,2, 3, 4, 5 del sprint backlog 3.

Figura 46

Código del servicio para obtener la Cuenta Bancaria por ID

```
getExclusivo(id: number): Observable<CuentaBancaria> {
  return this.http.get<CuentaBancaria>(`${this.API}/${id}`, {
    headers: {
      Authorization: `Bearer ${this.TOKEN}`
    },
  });
}
```

Figura 47

Código del servicio para registrar, modificar y anular una Cuenta Bancaria

```
post(cuenta: CuentaBancariaInsertar): Observable<RespuestaServidor> {
  return this.http.post<RespuestaServidor>(this.API, cuenta, {
    headers: {
      Authorization: `Bearer ${this.TOKEN}`,
    },
  });
}
update(
  cuenta: CuentaBancariaInsertar,
  id: number
): Observable<RespuestaServidor> {
  return this.http.put<RespuestaServidor>(`${this.API}/${id}`, cuenta, {
    headers: {
      Authorization: `Bearer ${this.TOKEN}`,
    },
  });
}

delete(id: number): Observable<RespuestaServidor> {
  return this.http.delete<RespuestaServidor>(`${this.API}/${id}`, {
    headers: {
      Authorization: `Bearer ${this.TOKEN}`,
    },
  });
}
```

Figura 48

Seguridad de controlador Cuenta Bancaria por ID

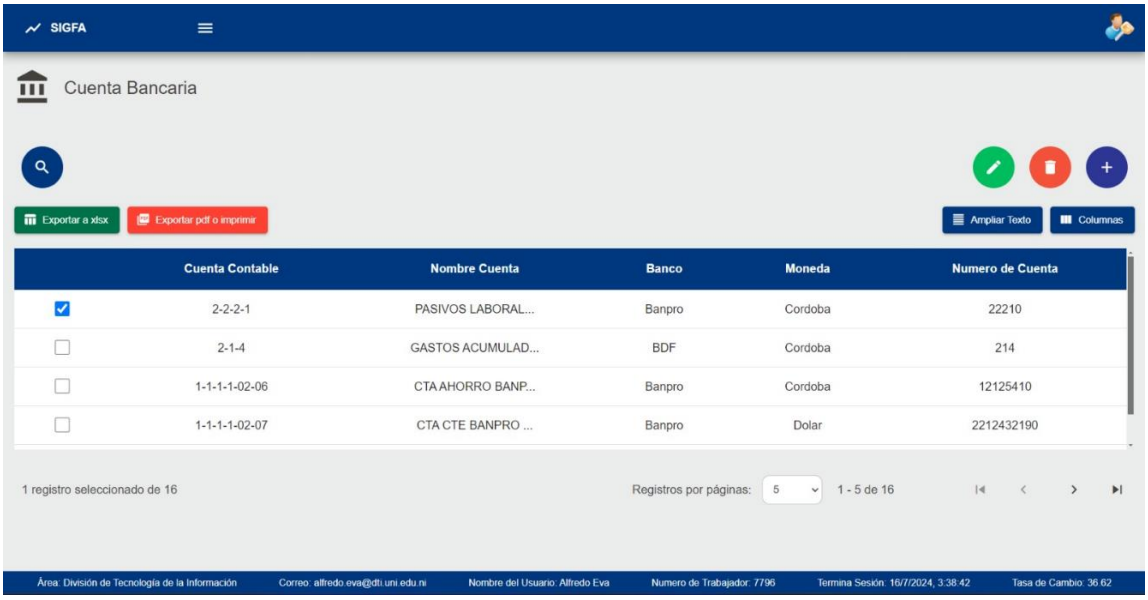
```
[HttpGet("{id:int}")]
0 references
public async Task<IActionResult> Get(int id, [FromHeader] String[] entidad)
{
    try
    {
        if (entidad.IsNullOrEmptyArray())
            return StatusCode(StatusCodes.Status401Unauthorized,
                new { Mensaje = "Error, no autorizado." });
        ConsultaPermiso cp = new ConsultaPermiso(_configuration)
        {
            Accion = _configSecurity.HomologacionAccion(Accion.Acceder),
            Pantalla = _configSecurity.HomologacionPantalla(Pantalla.CuentaBancaria),
            Entidades = entidad
        };
        var permiso = await _httpPermisoService.VerificarPermisoAsync(cp);
        if (!permiso.TienePermiso)
            return StatusCode(StatusCodes.Status403Forbidden,
                new { Mensaje = "Error, no tiene asignado los permisos necesarios para realizar la acción." });
        var cuentaBancaria = cuentaBancariaService.LeerCuentaBancaria(id);
        return Ok(cuentaBancaria);
    }
    catch (Exception e)
    {
        Console.WriteLine(e);
        throw;
    }
}
```

También se llevó a cabo el diseño de las interfaces de modales relacionado con las Cuentas Bancarias. Estos diseños, junto con la estructura de navegación de las pantallas, fueron inicialmente bosquejados en papel para luego ser implementados utilizando Visual Studio Code.

Además, el desarrollo del Backend estuvo estrechamente vinculado al del Frontend. A medida que surgían especificaciones de historias de usuario para la visualización o consumo de datos, se implementaban los Endpoint necesarios para completar el flujo de trabajo. Esta sincronización entre el Backend y el Frontend garantizó una integración coherente y eficiente a lo largo de todo el proceso de desarrollo. Se muestra un ejemplo del diseño de las pantallas correspondiente a la historia de usuario 2 del sprint backlog 3.

Figura 49

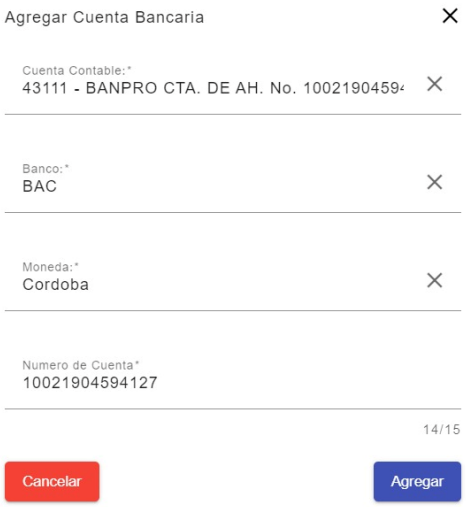
Pantalla principal, donde se listan las Cuentas Bancarias



Los avances del módulo fueron subidos al repositorio GitLab, una vez estos eran revisados y aprobados por el ing. Yasser Montiel pasaban a la rama principal del proyecto.

Figura 50

Pantalla para editar las Cuentas Bancarias



10.2.2.3. Revisión del Sprint 3: entrega del producto terminado, Cuentas Bancarias

Una vez completado el desarrollo de acuerdo con los elementos definidos en el Product Backlog, se llevaba a cabo una presentación del producto al Product Owner MSc. Pablo David Trejos y el stakeholder Lic. Gustavo Adolfo Silva Bermúdez. Durante esta sesión, los responsables evaluaban el trabajo entregado, formalizando sus conclusiones mediante un correo institucional se recogía los acuerdos alcanzados. Además, se aprovechaba la oportunidad para discutir cualquier pregunta, duda o posible mejora que pudiera surgir, con el objetivo de garantizar la alineación del producto con las expectativas y necesidades del cliente.

El equipo de desarrollo llevó a cabo la demostración y entrega del tercer sprint, que abarcaba el desarrollo completo de las cuentas bancarias dentro del sistema SIGFA. Durante la entrega, se garantizó que el módulo estuviera completamente funcional.

Además, se cumplió la entrega en tiempo y forma según planificación. El módulo resultante abarca las siguientes funcionalidades.

1. Agregar Cuenta Bancaria.
2. Editar Cuenta Bancaria.
3. Anular Cuenta Bancaria.
4. Mostrar Cuentas Bancaria.
5. Buscar Cuenta Bancaria por Cuentas Contable y Nombre.

Feedback y retroalimentación

Durante la sesión, las partes interesadas plantearon diversas preguntas, entre las que se destacan:

1. ¿Cómo se relaciona una cuenta contable con un número de Cuenta Bancaria?
2. ¿Cuál es el proceso para registrar una nueva Cuenta Bancaria en el sistema SIGFA?

3. ¿Qué pasos se siguen para editar la información de una Cuenta Bancaria específica?

Estas interrogantes fueron abordadas a través de la demostración del módulo desarrollado del sistema. Asimismo, se proporcionará un manual de usuario detallado que permitirá aclarar estas y otras dudas similares al utilizar el sistema, garantizando así una comprensión.

Con esto, se cerró la reunión de entrega del Sprint, logrando un alto nivel de éxito al cumplir en tiempo y forma con los requisitos establecidos por el Product Owner. El Scrum Master Ing. Rodrigo Diaz coordinó la planificación de la siguiente reunión para la preparación del siguiente Sprint.

10.2.2.4. Retrospectiva de Sprint 3

El Scrum Master Ing. Rodrigo Diaz convocó al Equipo de Desarrollo para llevar a cabo una retrospectiva del primer Sprint, con el propósito de identificar posibles fallos como equipo y áreas de mejora. Durante esta sesión reflexiva, se fomentó un ambiente de colaboración y retroalimentación, permitiendo que cada miembro del equipo de desarrollo expresara sus opiniones y sugerencias para optimizar el proceso de desarrollo en futuras iteraciones.

Identificación de fallos

Durante la revisión del proceso de desarrollo, el equipo identificó una problemática importante, se estaban nombrando de forma errónea el controlador y modelo de las cuentas bancarias y falta de comentarios en partes importantes del código. Estas prácticas dificultaban la revisión y comprensión del código por parte del equipo, lo que afectaba negativamente la calidad y eficiencia del desarrollo.

Análisis de la raíz del fallo

El equipo de desarrollo identifico la raíz del fallo de estas malas prácticas de codificación, entre las cuales se mencionaron las siguientes:

1. Presión por cumplir plazos.
2. Falta de estándares y prácticas de codificación.
3. Falta de capacitación.

Solución

Se resolvió esta problemática mediante la implementación de planes de capacitación para el equipo de desarrollo, enfocados en promover buenas prácticas de codificación, así como codificación limpia y clara. Además, se establecieron estándares claros de codificación y se implementó un proceso de revisión regular por parte del ing. Carlos Martínez encargado de subir los cambios en la rama principal.

10.2.3. Sprint 4

Este sprint consistió en el desarrollo del módulo de “asientos contables tipo” que administrará el sistema SIGFA, es la base para clasificar el tipo de asiento contable (comprobante de pago o diario) de la contabilidad financiera y sirve para registrar todas las transacciones financieras de manera ordenada. Estos registros son importantes para mantener una búsqueda precisa de los ingresos, gastos, activos, pasivos y el patrimonio de la UNI.

10.2.3.1. Sprint Planning

Este cuarto Sprint Planning se enfocará en el módulo de “asiento contable tipo”. El Product Backlog de este Sprint indica el desarrollo de las historias de usuario que se incluyen en la tabla 23. Durante el cuarto sprint, con una duración de 21 días, designado para el módulo de “asiento contable tipo”, se llevaron a cabo reuniones diarias de Scrum con el propósito de mantener a todos los integrantes del equipo al tanto del avance de las tareas asignadas.

Tabla 23: Sprint Backlog para el sprint 4

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Responsable	Estimación
1	Contador y Director	Agregar Asiento Contable Tipo.	Añadir Asientos Contables Tipo al sistema SIGFA.	Jorge Torres; Alfredo Eva	6 días
2	Director y Contador	Editar Asiento Contable Tipo.	Arreglarlas en caso de que haya cambios.	Carlos Martínez	3 días
3	Director y Contador	Anular Asiento Contable Tipo.	Ponerlas en estado inhabilitadas en el sistema SIGFA.	Gabriel García	4 días
4	Contador y Director	Mostrar los Asientos Contables Tipo.	Tener una visualización clara y precisa de los asientos contables tipo.	Carlos Martínez	6 días
5	Director y Contador	Buscar Asiento Contable Tipo por Nombre.	Encontrar información de ese Asiento Contable Tipo con rapidez.	Ronnie Acevedo	2 día

10.2.3.2. Resumen del proceso de desarrollo de software del Sprint 4

Tal como sucedió en Sprints previos, el desarrollo de las tablas se llevó a cabo directamente en Microsoft SQL Server, con un enfoque centrado en las historias de usuario definidas para este Sprint, así como en las instrucciones detalladas proporcionadas por expertos en los procesos institucionales pertinentes.

Este proceso incluyó la creación de la estructura de las tablas relacionadas con Asiento Contable Tipo, incluyendo las tablas con sus controladores que a su vez esto genera sus Endpoint; que son utilizados por el Frontend para mostrar la información de las cuentas bancarias, para garantizar el funcionamiento óptimo del módulo de asientos contables tipo. Se muestra un ejemplo del diseño de la base de datos correspondiente a la historia de usuario 1, 2, 3, 4, 5 del sprint backlog 4.

Figura 51

Diseño de la base de datos, tabla de Asiento Contable Tipo

Continuando con el proceso de desarrollo, se procedió a implementar tanto el Backend como el Frontend utilizando herramientas específicas (Frontend: Visual Studio Code en combinación con el Framework Angular; Backend: Visual Studio junto con Entity Framework 6.0). A continuación, se presenta un ejemplo del código correspondiente a la historia de usuario 1,2, 3, 4, 5 del sprint backlog 4.

Figura 52

Código del evento para registrar un Asiento Contable Tipo

```
export class InsertareditarComponent implements OnInit, InsertarEditar {

  // Llama al metodo post para insertar los datos
  insertarDatos(): void {
    try {
      this.api.post(this.asientoContableForm.value).subscribe({
        next: (res) => {
          if (res.mensaje) {
            this.toast.openSnackBar(res.mensaje);
            this.asientoContableForm.reset();
            this.matDialogRef.close('save');
          } else {
            this.errorPeticion(
              null,
              false,
              'Advertencia al Registrar el Asiento Contable Tipo.'
            );
          }
        },
        error: (err) => {
          this.errorPeticion(err, false, '');
        },
      });
    } catch (err) {
      this.errorPeticion(err, false, '');
    }
  }
```

Figura 53

Código del modelo de datos del formulario para registrar, actualizar y buscar en el módulo de Asiento Contable Tipo

```
export interface AsientoContableTipo {
  idAsientoContableTipo: null | number;
  nombre: string;
}
export interface AsientoContableTipoFiltro extends   Omit<AsientoContableTipo,'idAsientoContableTipo'>{
  ordenamiento: string| null;
  orientacion: string| null;
}
```

Figura 54

Código del evento de anulación lógica en el controlador de Asiento Contable Tipo

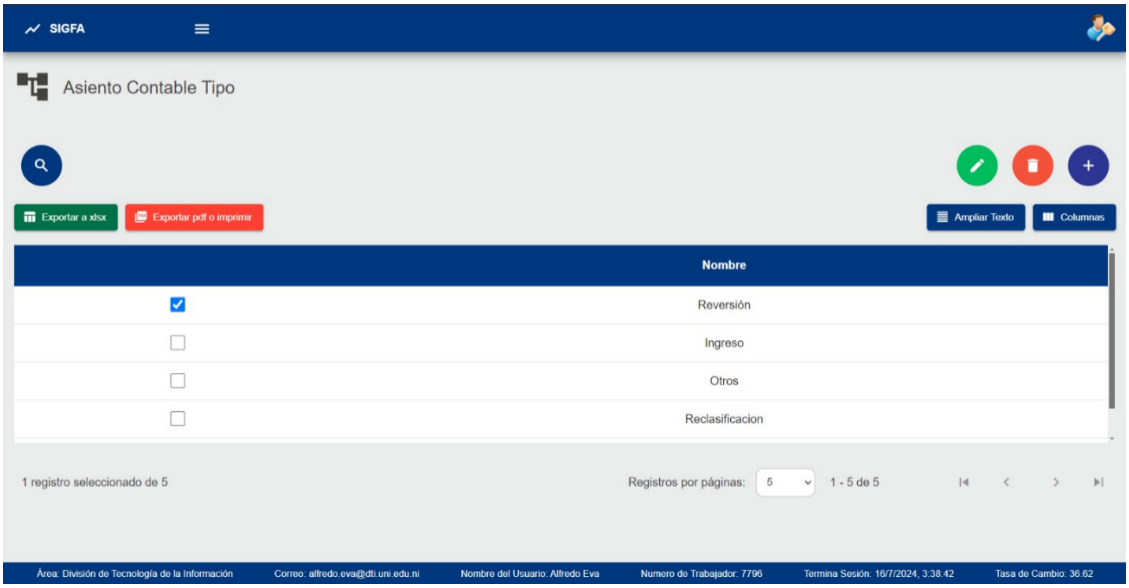
```
[HttpDelete("{idAsientoContableTipo:int}")]
0 referencias
public async Task<IActionResult> DeleteAsync(int idAsientoContableTipo, [FromHeader] String[] entidad)
{
    try
    {
        if (!ModelState.IsValid) return BadRequest(ModelState);
        if (entidad.IsNullOrEmptyArray())
            return StatusCode(StatusCodes.Status401Unauthorized,
                new { Mensaje = "Error, no autorizado." });
        ConsultaPermiso cp = new ConsultaPermiso(_configuration)
        {
            Accion = _configSecurity.HomologacionAccion(Accion.Eliminar),
            Pantalla = _configSecurity.HomologacionPantalla(Pantalla.AsientoContableTipo),
            Entidades = entidad
        };
        var permiso = await _httpPermisoService.VerificarPermisoAsync(cp);
        if (!permiso.TienePermiso)
            return StatusCode(StatusCodes.Status403Forbidden,
                new { Mensaje = "Error, no tiene asignado los permisos necesarios para realizar la acción." });

        AsientoContableTipoDelete d = new AsientoContableTipoDelete()
        {
            IdUsuarioAnulacion = await usuarioService.ObtenerIdUsuarioAsync(Request.ObtenerUsuarioGuidHeader())
        };

        asientoContableTipoService.EliminarAsientoContableTipo(idAsientoContableTipo, d);

        return StatusCode(StatusCodes.Status200OK, new { Mensaje = "El tipo de asiento contable se ha eliminado exitosamente." });
    }

    catch (Exception ex)
    {
        //db.Database.RollbackTransaction();
        return HandleException.GetExceptionResult(ex);
    }
}
```

También se realizó el diseño de las interfaces asociadas con los Asientos Contables Tipo. Estos diseños, junto con la estructura de navegación de las pantallas, fueron inicialmente diseñadas en papel para luego ser implementados utilizando Visual Studio Code.

Además, el desarrollo del Backend estuvo estrechamente ligado al del Frontend. A medida que surgían requisitos para la visualización o consumo de datos, se implementaban los Endpoint necesarios para completar el flujo de trabajo. Esta sincronización entre el Backend y el Frontend garantizó una integración coherente y eficiente a lo largo de todo el proceso de desarrollo. A continuación, se presenta un ejemplo del diseño de las pantallas correspondiente a la historia de usuario 5 y 6 del backlog del sprint 4.

Figura 55

Pantalla principal de visualización de Asiento Contables Tipo



Los avances del módulo fueron subidos al repositorio GitLab, una vez estos eran revisados y aprobados por el ing. Carlos Martínez pasaban a la rama principal del proyecto.

10.2.3.3. Revisión del Sprint 4: entrega de producto terminado, Asientos Contables Tipo

Una vez finalizado el desarrollo de acuerdo con los elementos definidos en el Product Backlog, se procedía a realizar una presentación del producto al Product Owner MSc. Pablo David Trejos y el stakeholder Lic. Gustavo Adolfo Silva Bermúdez. Durante esta sesión, los responsables evaluaban el trabajo entregado y formalizaban sus conclusiones mediante los acuerdos alcanzados vía correo electrónico.

El equipo de desarrollo llevó a cabo la demostración y entrega del cuarto sprint, que abarcaba el desarrollo completo de los Asientos Contables Tipos dentro del sistema SIGFA. Durante la entrega, se garantizó que el módulo estuviera completamente funcional.

Además, se cumplió la entrega en tiempo y forma según planificación. El módulo resultante abarca las siguientes funcionalidades.

1. Agregar Asiento Contable Tipo.
2. Editar Asiento Contable Tipo.
3. Anular Asiento Contable Tipo.
4. Mostrar Asiento Contable Tipo.
5. Buscar Asiento Contable Tipo por nombre.

Feedback y retroalimentación

Durante la sesión, las partes interesadas plantearon diversas preguntas, entre las que se destacan:

1. ¿Cuál es el proceso para registrar una nueva Asiento Contable Tipo en el sistema SIGFA?
2. ¿Cómo se procede para dar de baja un Asiento Contable Tipo?
3. ¿Cuál es la metodología para llevar a cabo la búsqueda de un Asiento Contable Tipo en el sistema?
4. ¿Qué pasos se siguen para editar la información de un Asiento Contable Tipo específica?

Estas interrogantes fueron abordadas a través de la demostración del módulo desarrollado del sistema. Asimismo, se proporcionará un manual de usuario detallado que permitirá aclarar estas y otras dudas similares al utilizar el sistema, garantizando así una comprensión.

Con esto, se cerró la reunión de entrega del Sprint, logrando un alto nivel de éxito al cumplir en tiempo y forma con los requisitos establecidos por el Product Owner. El Scrum Master Ing. Rodrigo Diaz coordinó la planificación de la siguiente reunión para la preparación del siguiente Sprint.

10.2.3.4. Retrospectiva de Sprint 4

El Scrum Master Ing. Rodrigo Diaz convocó al Equipo de Desarrollo para llevar a cabo una retrospectiva del cuarto Sprint, con el propósito de identificar posibles

fallos como equipo y áreas de mejora. Durante esta sesión reflexiva, se fomentó un ambiente de colaboración y retroalimentación, permitiendo que cada miembro del equipo de desarrollo expresara sus opiniones y sugerencias para optimizar el proceso de desarrollo en futuras iteraciones.

Identificación de fallos

Durante la revisión del proceso de desarrollo, el equipo detectó una preocupación, ciertos desarrolladores estaban escribiendo código de manera desorganizada, no alineada con la arquitectura de desarrollo establecida para el SIGFA (arquitectura orientada por componentes). Uno de los problemas más frecuentes era la combinación de la codificación de las solicitudes y el diseño en un solo archivo. Esta práctica es incorrecta, ya que las solicitudes deben estar separadas en un archivo distinto ubicado en un directorio separado, llamado 'services'.

Análisis de la raíz del fallo

El equipo de desarrollo identifico la raíz del fallo de estas malas prácticas de codificación, entre las cuales se mencionaron las siguientes:

1. Presión por cumplir plazos.
2. Falta de estándares y prácticas de codificación.
3. Falta de capacitación.
4. Falta de supervisión.

Solución

Se soluciono dicha problemática de la siguiente forma:

1. Capacitación.
2. Revisión de código.
3. Herramientas de análisis estático.
4. Plantillas y directrices de codificación.
5. Refactorización gradual.
6. Automatización de tareas repetitivas.

10.2.4. Sprint 5

Este sprint consistió en el desarrollo del módulo de documentos contables que administra el sistema SIGFA, los cuales listan los tipos de documentos contables que pueden ser emitidos en la UNI, por el sistema financieros y que son utilizados en los movimientos contables.

10.2.4.1. Sprint Planning

Este quinto Sprint Planning se enfocará en el módulo de documentos contables. El Product Backlog de este Sprint indica el desarrollo de las historias de usuario que se incluyen en la siguiente tabla:

Durante el cuarto sprint, con una duración de 21 días, designado para el módulo de documentos contables, se llevaron a cabo reuniones diarias de Scrum con el propósito de mantener a todos los integrantes del equipo al tanto del avance de las tareas asignadas.

Tabla 24**:** Sprint Backlog para el sprint 5

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Responsable	Estimación
1	Contador	Agregar Tipo Documento Contable.	Añadir Tipo Documentos Contables al sistema SIGFA.	Jorge Torres; Alfredo Eva	7 días
2	Director	Editar Tipo Documento Contable.	Arreglarlos en caso de que haya cambios.	Jorge Torres; Alfredo Eva	5 días
3	Director	Anular Tipo Documento Contable.	Ponerlas en estado inhabilitadas en el sistema SIGFA.	Jorge Torres; Alfredo Eva	4 días
4	Contador y Director	Mostrar los Tipo Documentos Contables.	Tener una visualización clara y precisa de los Tipo Documentos Contables.	Jorge Torres; Alfredo Eva	4 días
5	Director	Buscar Tipo Documentos Contables por nombre.	Encontrar información de ese Tipo Documento Contable con rapidez.	Jorge Torres; Alfredo Eva	1 día

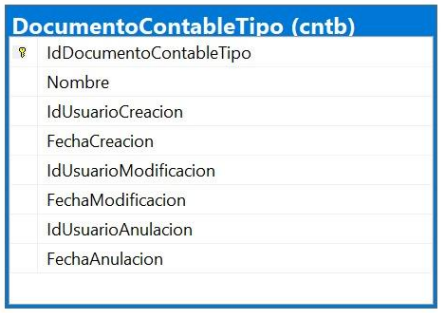
10.2.4.2. Resumen del proceso de desarrollo de software del Sprint 5

Al igual que en Sprints anteriores, el desarrollo de las tablas se llevó a cabo directamente en Microsoft SQL Server, con un enfoque centrado en las historias de usuario definidas para este Sprint, así como en las instrucciones detalladas proporcionadas por expertos en los procesos institucionales pertinentes.

Este proceso incluyó la creación de la estructura de las tablas relacionadas con Tipo Documento Contable, incluyendo las tablas con sus controladores que a su vez esto genera sus Endpoint; que son utilizados por el Frontend para mostrar la información del Documento Contable Tipo, para lograr el funcionamiento óptimo del módulo de documentos contables. Se muestra un ejemplo del diseño de la base de datos correspondiente a la historia de usuario 1, 2, 3, 4, 5 del sprint backlog 5.

Figura 56

Diseño de la base de datos, tabla de Tipo Documento Contable



Continuando con el proceso de desarrollo, se procedió a implementar tanto el Backend como el Frontend utilizando herramientas específicas (Frontend: Visual Studio Code en combinación con el Framework Angular; Backend: Visual Studio junto con Entity Framework 6.0). Esta selección de herramientas facilitó una integración eficiente entre las distintas partes del sistema, lo que contribuyó a agilizar el desarrollo y mejorar la mantenibilidad del proyecto. A continuación, se

presenta un ejemplo del código correspondiente a la historia de usuario 1, 2, 3, 4, 5 del sprint backlog 5.

Figura 57

Código del evento para anular un Tipo Documento Contable

```
export class TipodocumentoComponent implements OnInit, OnDestroy, Modulo {
  onEliminarModal() {
    if (this.tipoDocumentoEdit) {
      this.mensajeFlotante(
        'Eliminar',
        'delete',
        'red',
        '',
        '',
        '¿Esta seguro que quiere eliminar el Tipo Documento?',
        () => {
          if (
            this.tipoDocumentoEdit!.idDocumentoContableTipo?.toString() !=
              '' &&
            this.tipoDocumentoEdit!.idDocumentoContableTipo?.toString() !=
              null
          ) {
            this.eliminarElemento(
              this.tipoDocumentoEdit!.idDocumentoContableTipo!
            );

            this.reiniciarBandera('actualizado', '');
          }
        },
        () => {}
      );
    } else {
      this.toast.openSnackBar('Seleccione un Tipo Documento');
    }
  } catch (err) {
    this.toast.openSnackBar('Tipo Documento no Eliminado:' + err);
  }
}
```

Figura 58

Código del servicio de Tipo Documento Contable para anulación lógica del registro seleccionado

```
export class TipodocumentoService implements Servicio {
  delete(id: number): Observable<RespuestaServidor> {

    return this.http.delete<RespuestaServidor>(`${this.API}/${id}`, {
      headers: {
        Authorization: `Bearer ${this.TOKEN}`,
      },
    });
  }
```

Figura 59

Código del evento de anulación lógica en el controlador de Tipo Documento Contable

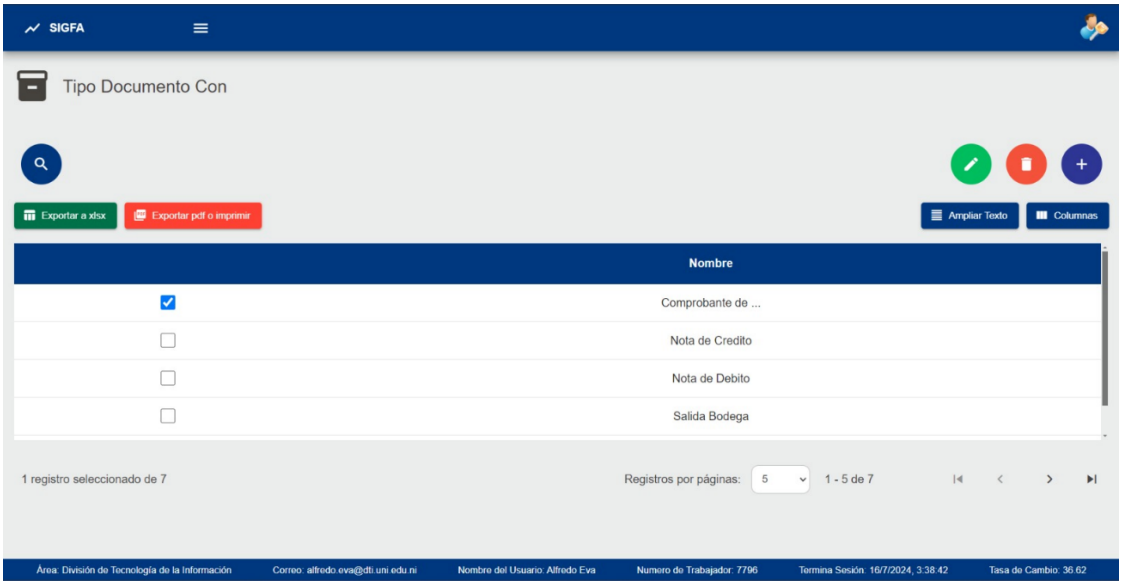
```
public class DocumentoContableController : ControllerBase
    [HttpDelete("{idDocumentoContable:int}")]
    0 references
    public async Task<IActionResult> DeleteAsync(int idDocumentoContable)
    {
        try
        {
            if (!ModelState.IsValid) return BadRequest(ModelState);

            DocumentoContableDelete d = new DocumentoContableDelete()
            {
                IdUsuarioAnulacion = await usuarioService.ObtenerIdUsuarioAsync(Request.ObtenerUsuarioGuidHeader())
            };

            documentoContableService.EliminarDocumentoContable(idDocumentoContable, d);

            return StatusCode(StatusCodes.Status200OK, new { Mensaje = "El documento contable se ha eliminado exitosamente." });
        }

        catch (Exception ex)
        {
            return HandleException.GetExceptionResult(ex);
        }
    }

}
```

También se realizó el diseño de las interfaces asociadas con los documentos contables. Estos diseños, junto con la estructura de navegación de las pantallas, fueron inicialmente diseñadas en papel para luego ser implementados utilizando Visual Studio Code.

Además, el desarrollo del Backend estuvo estrechamente ligado al del Frontend. A medida que surgían requisitos para la visualización o consumo de datos, se implementaban los Endpoint necesarios para completar el flujo de trabajo. Esta sincronización entre el Backend y el Frontend garantizó una integración coherente y eficiente a lo largo de todo el proceso de desarrollo. A continuación, se presenta un ejemplo del diseño de las pantallas correspondiente a la historia de usuario 5 y 6 del backlog del sprint 5.

Figura 60

Pantalla principal de visualización de Tipo Documentos Contables



Los avances del módulo fueron subidos al repositorio GitLab, una vez estos eran revisados y aprobados por el ing. Carlos Martínez pasaban a la rama principal del proyecto.

10.2.4.3. Revisión del Sprint 5: entrega del producto terminado, Tipo Documentos Contables

Una vez finalizado el desarrollo de acuerdo con los elementos definidos en el Product Backlog, se procedía a realizar una presentación del producto al Product Owner MSc. Pablo David Trejos y el stakeholder Lic. Gustavo Adolfo Silva Bermúdez. Durante esta sesión, los responsables evaluaban el trabajo entregado y formalizaban sus conclusiones mediante, los acuerdos alcanzados vía correo electrónico.

El equipo de desarrollo llevó a cabo la demostración y entrega del quinto sprint, que abarcaba el desarrollo completo de los asientos contables dentro del sistema SIGFA. Durante la entrega, se garantizó que el módulo estuviera completamente funcional.

Además, se cumplió la entrega en tiempo y forma según planificación. El módulo resultante abarca las siguientes funcionalidades.

1. Agregar Tipo Documento Contable.
2. Editar Tipo Documento Contable.
3. Anular Tipo Documento Contable.
4. Mostrar Tipo Documento Contable.
5. Buscar Tipo Documento Contable por nombre.

Feedback y retroalimentación

Durante la sesión, las partes interesadas plantearon diversas preguntas, entre las que se destacan:

1. ¿Cuál es el proceso para registrar un nuevo Tipo Documento Contable en el sistema SIGFA?
2. ¿Cuál es la metodología para llevar a cabo la búsqueda de un Tipo Documento Contable en el sistema?
3. ¿Cuál rol tiene permitido dar de baja un Tipo Documento Contable?

Estas interrogantes fueron abordadas a través de la demostración del módulo desarrollado del sistema. Asimismo, se proporcionará un manual de usuario detallado que permitirá aclarar estas y otras dudas similares al utilizar el sistema, garantizando así una comprensión.

Con esto, se cerró la reunión de entrega del Sprint, logrando un alto nivel de éxito al cumplir en tiempo y forma con los requisitos establecidos por el Product Owner. El Scrum Master Ing. Rodrigo Diaz coordinó la planificación de la siguiente reunión para la preparación del siguiente Sprint.

10.2.4.4. Retrospectiva de Sprint 5

El Scrum Master Ing. Rodrigo Diaz convocó al Equipo de Desarrollo para llevar a cabo una retrospectiva del cuarto Sprint, con el propósito de identificar posibles fallos como equipo y áreas de mejora. Durante esta sesión reflexiva, se fomentó

un ambiente de colaboración y retroalimentación, permitiendo que cada miembro del equipo de desarrollo expresara sus opiniones y sugerencias para optimizar el proceso de desarrollo en futuras iteraciones.

Identificación de fallos

Durante la revisión del proceso de desarrollo, el equipo detectó, que ciertos desarrolladores no estaban siguiendo el patrón y lineamientos base de la arquitectura de desarrollo establecida para el SIGFA (arquitectura orientada por componentes). Uno de los problemas más frecuentes era la compatibilidad visual del componente formulario, no se adaptaban correctamente a la resolución de los dispositivos móviles. Esta práctica es incorrecta, ya que los componentes son reutilizables por distintos dispositivos y deben tener sus propios estilos en su archivo.

Análisis de la raíz del fallo

El equipo de desarrollo identifico la raíz del fallo de estas malas prácticas de codificación, entre las cuales se mencionaron las siguientes:

1. Presión por cumplir plazos.
2. Falta de estándares y prácticas de codificación.
3. Falta de capacitación.
4. Falta de supervisión.
5. Desconocimiento de los Tipo Documentos Contables.

Solución

Se soluciono dicha problemática de la siguiente forma:

1. Explicación del funcionamiento de los Tipo Documentos Contables.
2. Plantillas y directrices de codificación.
3. Refactorización gradual.
4. Automatización de tareas repetitivas.

10.2.5. Sprint 6

Este sprint, se enfocó en el desarrollo del módulo de "cheque" para la administración del sistema SIGFA. Este módulo se encarga de gestionar la edición de cheques, los cuales son previamente generados desde el sistema SIPPSI. Además, al editar un cheque también cambian la cuenta bancaria asociada a cada cheque, así como la chequera a la que pertenece dicho cheque. Estas funcionalidades no solo simplificarán el proceso de registro, sino que también mejorarán la auditoría contable en la UNI.

10.2.5.1. Sprint Planning

Este sexto Sprint Planning se enfocará en el módulo de “Cheque”. El Product Backlog de este Sprint indica el desarrollo de las historias de usuario que se incluyen en la siguiente tabla 25. Durante el sexto sprint, con una duración de 21 días, designado para el módulo de “cheque”, se llevaron a cabo reuniones diarias de Scrum con el propósito de mantener a todos los integrantes del equipo al tanto del avance de las tareas asignadas.

Tabla 25**:** Sprint Backlog para el sprint 6

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Responsable	Estimación
1	Director	Editar Cheque.	Cambiar datos del Cheque en caso de que haya alteraciones o incidencias.	Jorge Torres; Alfredo Eva	5 días
2	Director	Cambiar estados de Cheque.	Llevar la secuencia de la vida de cada Cheque	Jorge Torres; Carlos Martínez	5 días
3	Director	Mostrar los estados del Cheque.	Tener control del estado de cada Cheque.	Ronnie Acevedo	2 día
4	Director	Anular Cheque.	Anularlo en caso de que haya una incidencia con el beneficiario.	Ronnie Acevedo; Gabriel García	5 días
5	Director	Generar vista previa del Cheque.	Tener información completa de ese Cheque.	Gabriel García	3 días
6	Director	Buscar Cheque por Beneficiario, Fecha de Creación y Área.	Tener información de un Cheque de una manera más rápida.	Ronnie Acevedo	1 día

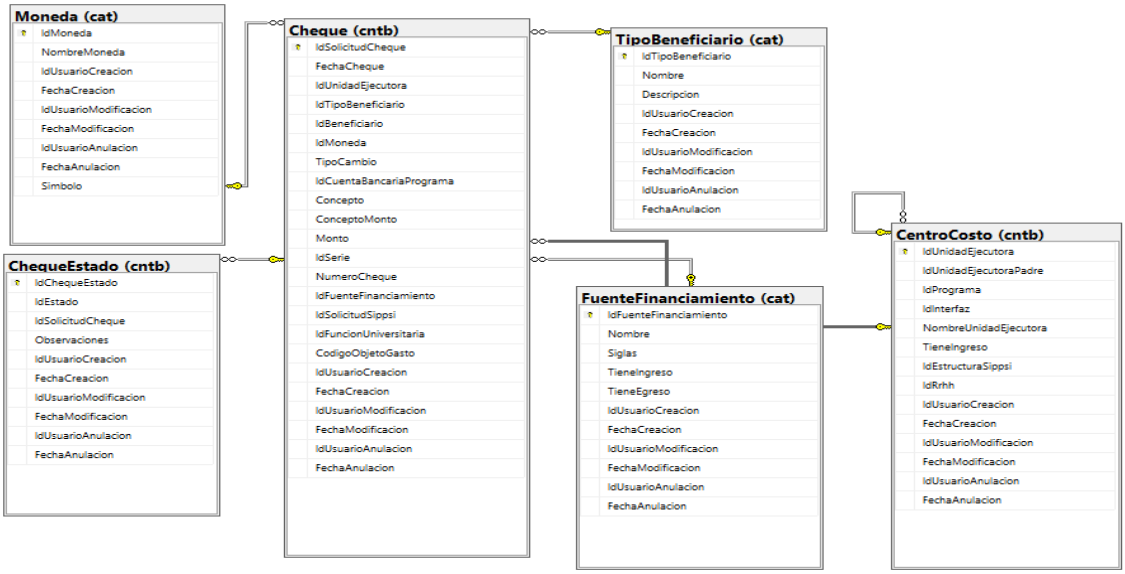
10.2.5.2. Resumen del proceso de desarrollo de software del Sprint 6

Tal como sucedió en Sprints previos, el desarrollo de las tablas se llevó a cabo directamente en Microsoft SQL Server, con un enfoque centrado en las historias de usuario definidas para este Sprint, así como en las instrucciones detalladas proporcionadas por expertos en los procesos institucionales pertinentes.

Este proceso incluyó la creación de la estructura de las tablas relacionadas con cheque, incluyendo las tablas con sus controladores que a su vez esto genera sus Endpoint; que son utilizados por el Frontend para mostrar la información de los cheques con su comprobante de pago y sus estados. A continuación, en la Fig. 61 Se muestra un ejemplo del diseño de la base de datos correspondiente a la historia de usuario 1, 2, 3, 4, 5 del sprint backlog 6.

Figura 61

Diseño de la base de datos, tabla de Cheque



Continuando con el proceso de desarrollo, se procedió a implementar tanto el Backend como el Frontend utilizando herramientas específicas (Frontend: Visual Studio Code en combinación con el Framework Angular; Backend: Visual Studio

junto con Entity Framework 6.0). A continuación, se presenta un ejemplo del código correspondiente a la historia de usuario 1,2, 3, 4, y 5 del sprint backlog 6.

Figura 62

Código método para editar el concepto, fuente de financiamiento y cuenta bancaria del cheque enviado desde SIPPSI

```
export class InsertarEditarComponent implements InsertarEditar {
  actualizarDatos(data: ChequeEdit): void {
    try {
      this.chequeService
        .update(data, this.editData.idSolicitudCheque!)
        .subscribe({
          next: (res) => {
            if (res.mensaje) {
              this.toast.openSnackBar(res.mensaje);
              this.chequeGeneralForm.reset();
              this.matDialogRef.close('update');
            } else {
              this.errorPeticion(
                null,
                false,
                'Advertencia al Modificar el Cheque.'
              );
            }
          },
          error: (err) => {
            this.errorPeticion(err, false, '');
          },
        });
    } catch (err) {
      this.errorPeticion(err, false, '');
    }
  }
```

Figura 63

Modelo de datos del editar Cheque

```
export interface ChequeEdit {
  idCuentaBancariaPrograma: number | null;
  concepto: string | null;
  idBeneficiario: string | null;
  idFuenteFinanciamiento: number | null;
}
```

Figura 64

Controlador para editar un Cheque por IdSolicitudCheque

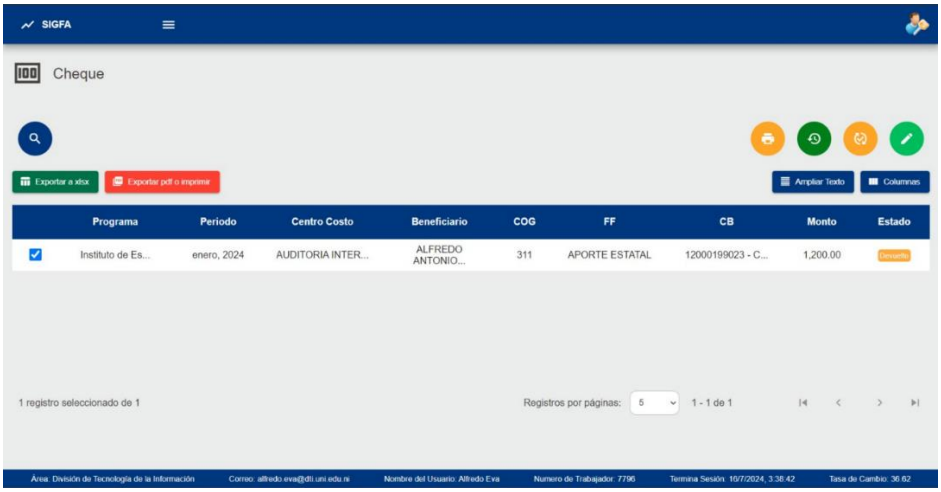
```
[HttpPut("{idSolicitudCheque:int}")]
0 referencias
public async Task<IActionResult> PutAsync(int idSolicitudCheque, [FromBody] ChequeUpdate u, [FromHeader] String[] entidad)
{

        try
        {
            if (entidad.IsNullOrEmptyArray())
                return StatusCode(StatusCodes.Status401Unauthorized,
                    new { Mensaje = "Error, no autorizado." });
            ConsultaPermiso cp = new ConsultaPermiso(_configuration)
            {
                Accion = _configSecurity.HomologacionAccion(Accion.Editar),
                Pantalla = _configSecurity.HomologacionPantalla(Pantalla.Cheque),
                Entidades = entidad
            };
            var permiso = await _httpPermisoService.VerificarPermisoAsync(cp);
            if (!permiso.TienePermiso)
                return StatusCode(StatusCodes.Status403Forbidden,
                    new { Mensaje = "Error, no tiene asignado los permisos necesarios para realizar la acción." });


           u.IdSolicitudCheque = idSolicitudCheque;
            u.IdUsuarioModificacion = await _usuarioService.ObtenerIdUsuarioAsync(Request.ObtenerUsuarioGuidHeader());

            _chequeService.ModificarCheque(u);

            return StatusCode(StatusCodes.Status200OK, new { Mensaje = "El Cheque se ha modificado exitosamente." });
 }
```

También se realizó el diseño de las interfaces asociadas con los Cheques. Estos diseños, junto con la estructura de navegación de las pantallas, fueron inicialmente diseñadas en papel para luego ser implementados utilizando Visual Studio Code.

Además, el desarrollo del Backend estuvo estrechamente ligado al del Frontend. A medida que surgían requisitos para la visualización o consumo de datos, se implementaban los Endpoint necesarios para completar el flujo de trabajo. Esta sincronización entre el Backend y el Frontend garantizó una integración coherente y eficiente a lo largo de todo el proceso de desarrollo. A continuación, se presenta un ejemplo del diseño de las pantallas correspondiente a la historia de usuario del 1 al 6 del backlog del sprint 6.

Figura 65

Pantalla principal de Cheque



Los avances del módulo fueron subidos al repositorio GitLab, una vez estos eran revisados y aprobados por el ing. Carlos Martínez pasaban a la rama principal del proyecto.

10.2.5.3. Revisión del Sprint 6: entrega del producto terminado, Gestionar Cheque

Una vez finalizado el desarrollo de acuerdo con los elementos definidos en el Product Backlog, se procedía a realizar una presentación del producto al Product Owner MSc. Pablo David Trejos y el stakeholder Lic. Gustavo Adolfo Silva Bermúdez. Durante esta sesión, los responsables evaluaban el trabajo entregado y formalizaban sus conclusiones mediante los acuerdos alcanzados vía correo electrónico.

El equipo de desarrollo llevó a cabo la demostración y entrega del sexto sprint, que abarcaba el desarrollo completo de Cheque dentro del sistema SIGFA. Durante la entrega, se garantizó que el módulo estuviera completamente funcional.

Además, se cumplió la entrega en tiempo y forma según planificación. El módulo resultante abarca las siguientes funcionalidades.

1. Editar Cheque.
2. Anular Cheque.
3. Mostrar Cheque.
4. Buscar Cheque.
5. Cambiar estado del Cheque.

Feedback y retroalimentación

Durante la sesión, las partes interesadas plantearon diversas preguntas, entre las que se destacan:

1. ¿Cuál es el proceso para editar un Cheque en el sistema SIGFA?
2. ¿Cómo se procede para anular un Cheque?
3. ¿Cuál es la metodología para llevar a cabo la búsqueda de un Cheque en el sistema?
4. ¿Qué pasos se siguen para cambiar de estado un Cheque?

Estas interrogantes fueron abordadas a través de la demostración del módulo desarrollado del sistema. Asimismo, se proporcionará un manual de usuario detallado que permitirá aclarar estas y otras dudas similares al utilizar el sistema, garantizando así una comprensión.

Con esto, se cerró la reunión de entrega del Sprint 6, logrando un alto nivel de éxito al cumplir en tiempo y forma con los requisitos establecidos por el Product Owner. El Scrum Master Ing. Rodrigo Diaz coordinó la planificación de la siguiente reunión para la preparación del siguiente Sprint.

10.2.5.4. Retrospectiva de Sprint 6

El Scrum Master Ing. Rodrigo Diaz convocó al Equipo de Desarrollo para llevar a cabo una retrospectiva del sexto Sprint, con el propósito de identificar posibles fallos como equipo y áreas de mejora. Durante esta sesión reflexiva, se fomentó un ambiente de colaboración y retroalimentación, permitiendo que cada miembro del equipo de desarrollo expresara sus opiniones y sugerencias para optimizar el proceso de desarrollo en futuras iteraciones.

Identificación de fallos

Durante la revisión del proceso de desarrollo, surgió una preocupación significativa: ciertos desarrolladores no estaban llevando a cabo las pruebas necesarias y pertinentes en los Endpoint que estaban siendo codificados e implementados. Esto resultó en problemas al realizar las pruebas finales con el Frontend y el Backend integrados, ya que los Endpoint fallaban o no hacían los que se esperaba debido a la falta de pruebas adecuadas.

Análisis de la raíz del fallo

El equipo de desarrollo identifico que la falta de pruebas adecuadas en los Endpoint del sistema fue la causa principal de los problemas encontrados durante las pruebas finales.

1. Falta de planificación de pruebas.
2. Falta de documentación adecuada.
3. Comunicación deficiente.
4. Falta de automatización de pruebas.

Solución

Se soluciono dicha problemática de la siguiente forma:

1. Planificación y asignación de recursos.
2. Comunicación efectiva.
3. Implementación de procesos de desarrollo robustos.
4. Automatización de pruebas.

10.2.6. Sprint 9

Este sprint, detalla el desarrollo del módulo de "Nota de Débito", responsable de gestionar los movimientos contables de tipo cobro, que son aplicados a los diferentes tipos de beneficiarios (Trabajadores, Estudiantes, Proveedores o Unidades Ejecutoras). Este módulo, será utilizado por los distintos centros de costos (UNI CENTRAL Y AUS) que realizan operaciones contables de tipo cobro.

10.2.6.1. Sprint Planning

Este noveno Sprint Planning se enfocará en el módulo de “Nota de Débito”. El Product Backlog de este Sprint indica el desarrollo de las historias de usuario que se incluyen en la tabla 26.

Durante el octavo sprint, con una duración de 21 días, designado para el módulo de “Nota de Débito”, se llevaron a cabo reuniones diarias de Scrum con el propósito de mantener a todos los integrantes del equipo al tanto del avance de las tareas asignadas.

Tabla 26**:** Sprint Backlog para el sprint 9

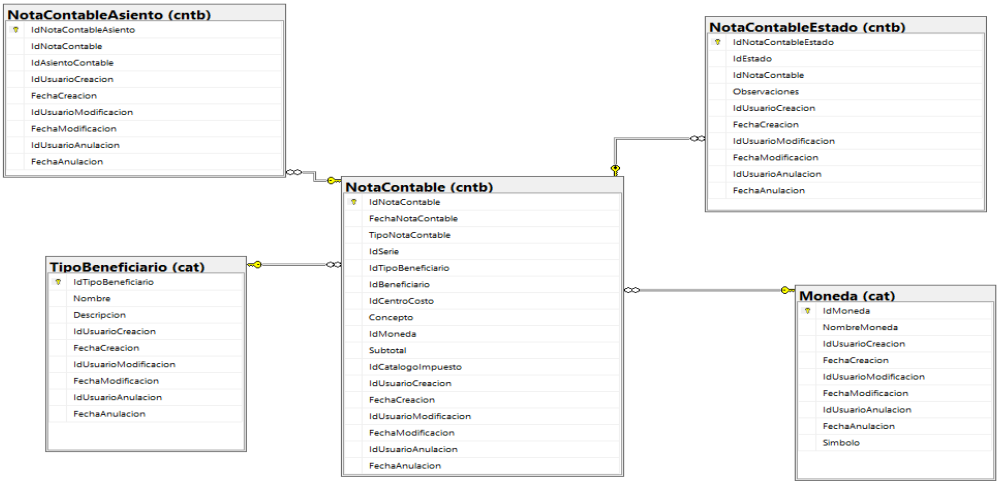
N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Responsable	Estimación
1	Contador	Agregar Nota de Débito.	Añadir Agregar Nota de Débito al sistema SIGFA.	Alfredo Eva; Jorge Torres	4 días
2	Director	Editar Nota de Débito.	Cambiar datos de la Nota de Débito en caso de que haya alteraciones o incidencias.	Alfredo Eva; Jorge Torres	4 días
3	Director	Cambiar estados de Nota de Débito.	Llevar la secuencia de la vida de una Nota de Débito.	Alfredo Eva; Jorge Torres	4 días
4	Contador	Mostrar los estados del Nota de Débito.	Tener control de cada estado de cada Nota de Débito.	Alfredo Eva; Jorge Torres	1 día
5	Director y Contador	Anular Nota de Débito.	Inhabilitarlo en caso de que haya una incidencia con un dato o cifra con la Nota de Débito.	Alfredo Eva; Jorge Torres	4 días
6	Director y Contador	Generar vista previa del Nota de Débito.	Tener información completa de esa Nota de Débito.	Alfredo Eva; Jorge Torres	2 días
7	Director y Contador	Buscar Nota de Débito por Beneficiario, Fecha de Creación y Área.	Tener información de una Nota de Débito de una manera más rápida.	Alfredo Eva; Jorge Torres	1 día
8	Contador	Mostrar las Nota de Débitos registradas.	Registros de las Notas de Débitos existentes.	Alfredo Eva	1 día

10.2.6.2. Resumen del proceso de desarrollo de software del Sprint 9

Tal como sucedió en Sprints previos, el desarrollo de las tablas se llevó a cabo directamente en Microsoft SQL Server, con un enfoque centrado en las historias de usuario definidas para este Sprint, así como en las instrucciones detalladas proporcionadas por expertos en los procesos institucionales pertinentes.

Figura 66

Diseño de la base de Datos de Nota de Débito



Continuando con el proceso de desarrollo, se procedió a implementar tanto el Backend como el Frontend utilizando herramientas específicas (Frontend: Visual Studio Code en combinación con el Framework Angular; Backend: Visual Studio junto con Entity Framework 6.0). A continuación, se presenta un ejemplo del código correspondiente a la historia de usuario 1, 2, 3, 4, 5 y 6 del sprint backlog 9.

Figura 67

Código del evento para filtras las Notas de Débitos ya registradas

```
export class NotacreditoComponent implements OnInit, OnDestroy, Modulo {
  onFiltroEmitter() {
    try {
      this.dialog
        .open(FiltroComponent, {
          width: '30%',
          data: this.notaCreditoFilter,
        })
        .afterClosed()
        .subscribe((val: 'close' | 'clear' | NotaCreditoFilter) => {
          if (val == 'close') {
            return;
          }
          this.notaCreditoFilter = val == 'clear' ? null : val;
          this.paginacion =
            this.tablapaginacionService.reiniciarPaginacionLocal(
              this.paginacion!
            );
          this.getElemento();
        });
    } catch (err) {
      this.toast.openSnackBar('Error al Buscar las Notas Contables:' + err);
    }
  }
```

Figura 68

Código del controlador que recibe los filtros enviados por el usuario

```
[HttpGet]
0 referencias | 0 cambios | 0 autores, 0 cambios
public ActionResult Get([FromQuery] NotaContableRead r)
{
    var notaContable = notaContableService.ObtenerListadoNotasContables(r);
    return Ok(notaContable);
}
```

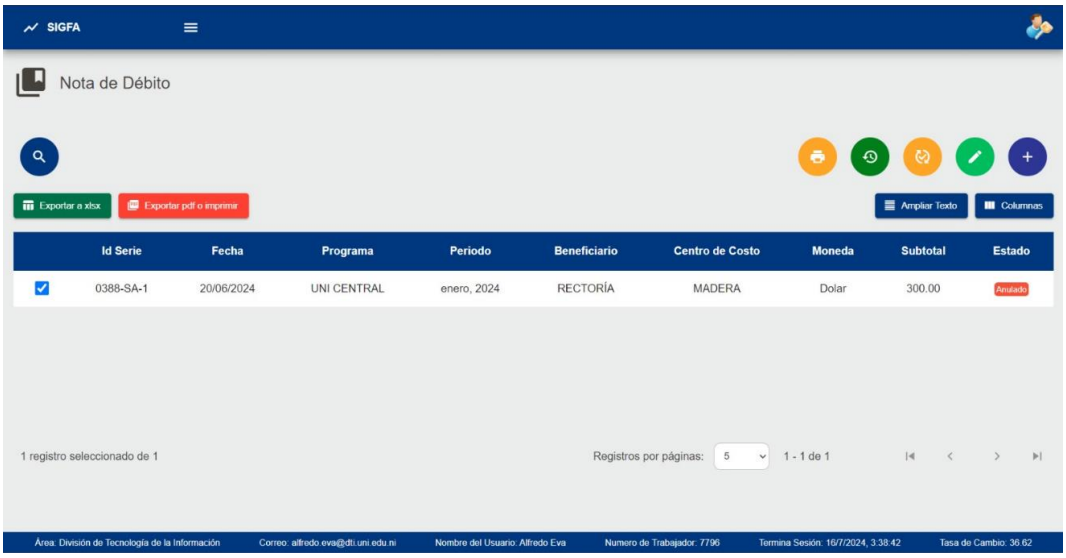
También se realizó el diseño de las interfaces asociadas con las Notas de Débito. Estos diseños, junto con la estructura de navegación de las pantallas, fueron inicialmente diseñadas en papel para luego ser implementados utilizando Visual Studio Code.

Además, el desarrollo del Backend estuvo estrechamente ligado al del Frontend. A medida que surgían requisitos para la visualización o consumo de datos, se implementaban los Endpoint necesarios para completar el flujo de trabajo. Esta sincronización entre el Backend y el Frontend garantizó una integración coherente y eficiente a lo largo de todo el proceso de desarrollo. A continuación, se presenta

un ejemplo del diseño de las pantallas correspondiente a la historia de usuario del 1 al 8 del backlog del sprint 9.

Figura 69

Pantalla principal de Nota de Débito



Los avances del módulo fueron subidos al repositorio GitLab, una vez estos eran revisados y aprobados por el Ing. Carlos Martínez pasaban a la rama principal del proyecto.

10.2.6.3. Revisión del Sprint 9: entrega del producto terminado Nota de Débito

Una vez finalizado el desarrollo de acuerdo con los elementos definidos en el Product Backlog, se procedía a realizar una presentación del producto al Product Owner MSc. Pablo David Trejos y el stakeholder Lic. Gustavo Adolfo Silva Bermúdez. Durante esta sesión, los responsables evaluaban el trabajo entregado y formalizaban sus conclusiones mediante los acuerdos alcanzados vía correo electrónico.

El equipo de desarrollo llevó a cabo la demostración y entrega del octavo sprint, que abarcaba el desarrollo completo de Nota de Débito dentro del sistema SIGFA. Durante la entrega, se garantizó que el módulo estuviera completamente

funcional. Además, se cumplió la entrega en tiempo y forma según planificación. El módulo resultante abarca las siguientes funcionalidades.

1. Agregar Nota de Débito.
2. Editar Nota de Débito.
3. Anular Nota de Débito.
4. Mostrar Nota de Débito.
5. Buscar Nota de Débito.
6. Cambiar estado Nota de Débito.

Feedback y retroalimentación.

Durante la sesión, las partes interesadas plantearon diversas preguntas, entre las que se destacan:

1. ¿Cuál es el proceso para editar una Nota de Débito en el sistema SIGFA?
2. ¿Cómo se procede para anular una Nota de Débito?
3. ¿Cuál es la metodología para llevar a cabo la búsqueda de una Nota de Débito en el sistema?
4. ¿Qué pasos se siguen para cambiar de estado una Nota de Débito?

Estas interrogantes fueron abordadas a través de la demostración del módulo desarrollado del sistema. Asimismo, se proporcionará un manual de usuario detallado que permitirá aclarar estas y otras dudas similares al utilizar el sistema, garantizando así una comprensión.

Con esto, se cerró la reunión de entrega del Sprint 9, logrando un alto nivel de éxito al cumplir en tiempo y forma con los requisitos establecidos por el Product Owner. El Scrum Master Ing. Rodrigo Diaz coordinó la planificación de la siguiente reunión para la preparación del siguiente Sprint.

10.2.6.4. Retrospectiva de Sprint 9

El Scrum Master Ing. Rodrigo Diaz convocó al Equipo de Desarrollo para llevar a cabo una retrospectiva del octavo Sprint, con el propósito de identificar posibles

fallos como equipo y áreas de mejora. Durante esta sesión reflexiva, se fomentó un ambiente de colaboración y retroalimentación, permitiendo que cada miembro del equipo de desarrollo expresara sus opiniones y sugerencias para optimizar el proceso de desarrollo en futuras iteraciones.

Identificación de fallos

No se identificaron fallos, debido a que, con la retroalimentación del módulo de Nota de Crédito, se aplicaron al desarrollo de este sprint, ya que presentaron la misma lógica programática a nivel de Backend y Bases de Datos.

Análisis de la raíz del fallo

No se identificaron fallos.

10.2.7. Sprint 11

Este sprint, se enfocó en el desarrollo de la sección de Reportes e Informes Financieros, que son vitales para la UNI para consultar sobre su posición financiera y resultado del periodo contable, esta información es crucial en la toma de decisiones operacionales administrativas. Además, estos informes son herramientas que solicitan las distintas autoridades como Rectoría, Auditoria (Auditoría Interna, Sub dirección de Finanzas).

10.2.7.1. Sprint Planning

Este Sprint Planning se enfocará en la sección de Reportes e Informes Financieros. El Product Backlog de este Sprint indica el desarrollo de las historias de usuario. Durante el décimo primer sprint, con una duración de 21 días, designado en la sección de Reportes e Informes Financieros, se llevaron a cabo reuniones diarias de Scrum con el propósito de mantener a todos los integrantes del equipo al tanto del avance de las tareas asignadas.

Tabla 27: Sprint Backlog para el sprint 11

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Responsable	Estimación
1	Director y Contador	Consultar datos de los Estados Financieros (Activo, Pasivo y Patrimonio, Ingresos y Egresos) en Landing Page del periodo general de Programa actual.	Tener una vista previa del estado actual de los datos financieros de la UNI.	Alfredo Eva	3 días
2	Director y Contador	Generar reporte de Balance general individual por periodo específico de Programa.	Tener información sobre las cuentas de pasivo, activo y patrimonio de la UNI.	Alfredo Eva	3 días
3	Director y Contador	Generar reporte de Balanza de comprobación individual por periodo específico de Programa.	Tener información de los saldos deudores y acreedores de la UNI.	Alfredo Eva	3 días
4	Director y Contador	Generar reporte de Estado de Resultado individual por periodo específico de Programa.	Tener información de que si la UNI tuvo ganancias o pérdidas económicas.	Alfredo Eva	3 días

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Responsable	Estimación
5	Director y Contador	Generar reporte de Libro Diario por periodo específico de Programa.	Mantener un histórico completo y detallado de todos los movimientos financieros de la UNI.	Alfredo Eva; Ronnie Acevedo	4 días
6	Director y Contador	Generar reporte de Libro Mayor por periodo específico de Programa.	Conocer el saldo que va sobrando en cada cuenta contable.	Alfredo Eva; Ronnie Acevedo	5 días

10.2.7.2. Resumen del proceso de desarrollo de software del Sprint 11

Tal como sucedió en Sprints previos, el desarrollo de los procedimientos almacenados y funcionales que se llevó a cabo directamente en Microsoft SQL Server y Fast Report, con un enfoque centrado en las historias de usuario definidas para este Sprint, así como en las instrucciones detalladas proporcionadas por expertos en los procesos institucionales pertinentes.

Figura 70

Diseño visual del reporte en Fast Report de Balanza de Comprobación

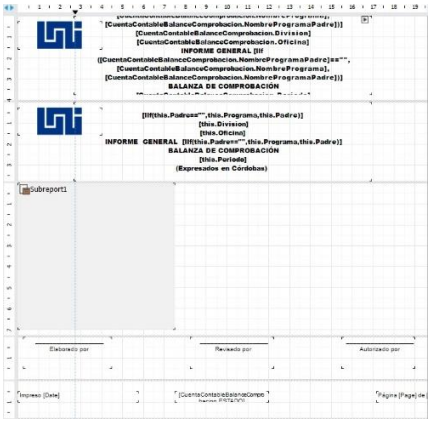


Figura 71

Vista previa del reporte con datos del Periodo Contable Programa al 31 de enero del 2024

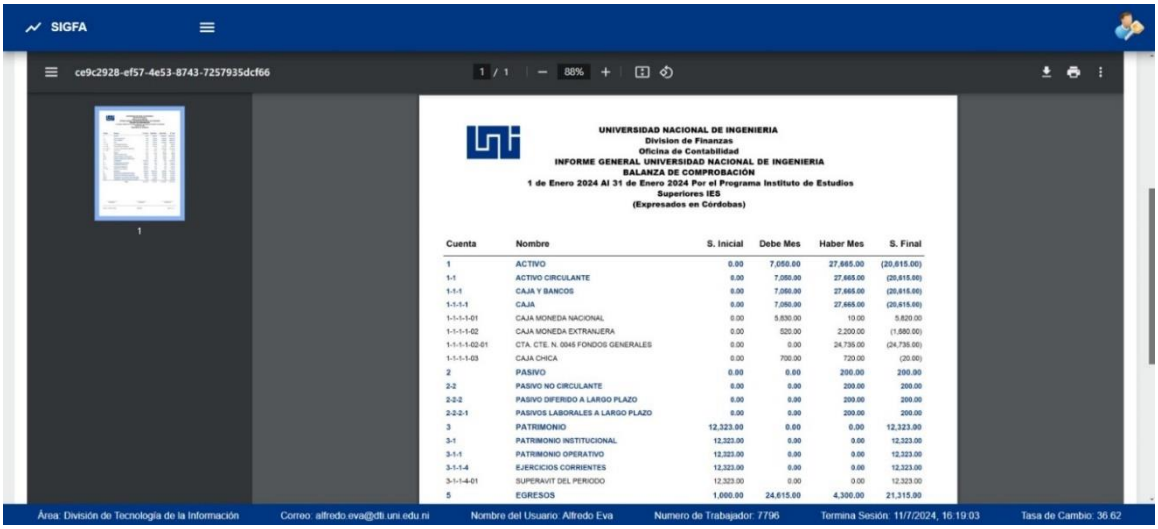


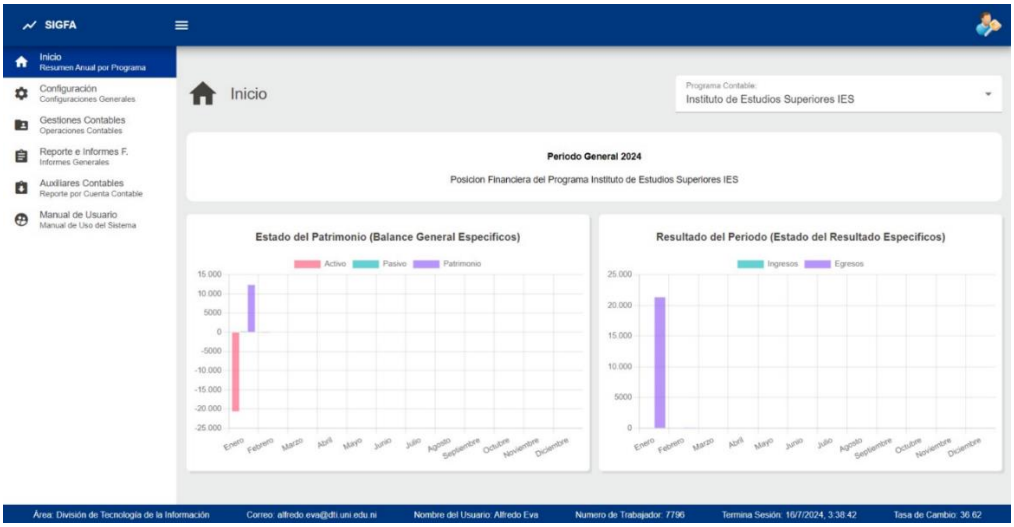
Figura 72

Lógica programática y contable Reporte Balanza de Comprobación

```
USE [SIGFA]
GO
/****** Object:  StoredProcedure [dbo].[SP_BALANZA_COMPROBACION]    Script Date: 11/7/2024 10:02:12 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[SP_BALANZA_COMPROBACION](@tipoPeriodo varchar(130),@idPeriodoContable int )
AS
SELECT T.IdCuentaContable,T.CuentaContable, T.Nombre, T.ACUMULADOINICIAL, T.DEBEMES,T.HABERMES,
        (
     select PP.nombrePrograma from  cntb.Programa as Pr  inner join
     cntb.Programa as PP on PP.IdPrograma=PR.IdProgramaPadre
    inner join cntb.PeriodoContable as P  on Pr.idPrograma=P.idPrograma
    where P.IDPERIODOCONTABLE=@idPeriodoContable
    ) as 'NombreProgramaPadre',
    (
     select Pr.nombrePrograma from  cntb.Programa as Pr  inner join
    cntb.PeriodoContable as P  on Pr.idPrograma=P.idPrograma

    where P.IDPERIODOCONTABLE=@idPeriodoContable
    ) as 'NombrePrograma',
    (
     select Pr.nombreDivision from  cntb.Programa as Pr  inner join
    cntb.PeriodoContable as P  on Pr.idPrograma=P.idPrograma
    where P.IDPERIODOCONTABLE=@idPeriodoContable

    ) as 'Division',
        (
     select Pr.NombreOficina from  cntb.Programa as Pr  inner join
    cntb.PeriodoContable as P  on Pr.idPrograma=P.idPrograma
    where P.IDPERIODOCONTABLE=@idPeriodoContable

    ) as 'Oficina',
CASE  SUBSTRING(T.CuentaContable,1,1)
  WHEN '1' THEN (T.DEBEMES-T.HABERMES+T.ACUMULADOINICIAL) -- DEUDORA
  WHEN '2' THEN (T.HABERMES-T.DEBEMES+T.ACUMULADOINICIAL) -- ACREDEDORA
  WHEN '3' THEN (T.HABERMES-T.DEBEMES+T.ACUMULADOINICIAL) -- ACREDEDORA
  WHEN '4' THEN (T.HABERMES-T.DEBEMES+T.ACUMULADOINICIAL) -- ACREDEDORA
  WHEN '5' THEN (T.DEBEMES-T.HABERMES+T.ACUMULADOINICIAL) -- DEUDORA
```

Figura 73

Pantalla de Landing Page de la posición financiera por Programa



También se realizó el diseño de las interfaces asociadas con la Landing Page. Estos diseños, junto con la estructura de navegación de las pantallas, fueron inicialmente diseñadas en papel para luego ser implementados utilizando Visual Studio Code.

Además, el desarrollo del Backend estuvo estrechamente ligado al del Frontend. A medida que surgían requisitos para la visualización o consumo de datos, se implementaban los Endpoint necesarios para completar el flujo de trabajo. Esta sincronización entre el Backend y el Frontend garantizó una integración coherente y eficiente a lo largo de todo el proceso de desarrollo. A continuación, se presenta un ejemplo del diseño de las pantallas correspondiente a la historia de usuario 1 del backlog del sprint 11.

10.2.7.3. Revisión del Sprint 11: entrega del producto terminado Reporte e Informes Financieros

Una vez finalizado el desarrollo de acuerdo con los elementos definidos en el Product Backlog, se procedía a realizar una presentación del producto al Product Owner MSc. Pablo David Trejos y el stakeholder Lic. Gustavo Adolfo Silva Bermúdez. Durante esta sesión, los responsables evaluaban el trabajo entregado y formalizaban sus conclusiones mediante los acuerdos alcanzados vía correo electrónico.

El equipo de desarrollo llevó a cabo la demostración y entrega del décimo primer sprint, que abarcaba el desarrollo completo de Reportes e Informes Financieros dentro del sistema SIGFA. Durante la entrega, se garantizó que el módulo estuviera completamente funcional. Además, se cumplió la entrega en tiempo y forma según planificación.

Además, se cumplió la entrega en tiempo y forma según planificación. La sección resultante abarca las siguientes funcionalidades.

1. Generar reporte de Balance General por Periodo Especifico de Programa.
2. Generar reporte de Balanza de Comprobación por Periodo Especifico de Programa.
3. Generar reporte de Estado de Resultado por periodo Especifico de Programa.
4. Generar reporte de Libro Diario por Periodo Especifico de Programa.

5. Generar reporte de Libro Mayor por Periodo Especifico de Programa.
6. Generar Posición Financiera y Resultado del Periodo por Periodo General (Conformado por todos los periodos específicos) de Programa en el Landing Page.

Feedback y retroalimentación

Durante la sesión, las partes interesadas plantearon diversas preguntas, entre las que se destacan:

1. ¿Cuál es el proceso para generar el reporte de Balance General en el SIGFA?
2. ¿Cómo se imprime el Reporte Balance General?

Estas interrogantes fueron abordadas a través de la demostración del módulo desarrollado del sistema. Asimismo, se proporcionará un manual de usuario detallado que permitirá aclarar estas y otras dudas similares al utilizar el sistema, garantizando así una comprensión.

Con esto, se cerró la reunión de entrega del Sprint 11, logrando un alto nivel de éxito al cumplir en tiempo y forma con los requisitos establecidos por el Product Owner. El Scrum Master Ing. Rodrigo Diaz coordinó la planificación de la siguiente reunión para la preparación del siguiente Sprint.

10.2.7.4. Retrospectiva de Sprint 11

El Scrum Master Ing. Rodrigo Diaz convocó al Equipo de Desarrollo para llevar a cabo una retrospectiva del décimo Sprint, con el propósito de identificar posibles fallos como equipo y áreas de mejora. Durante esta sesión reflexiva, se fomentó un ambiente de colaboración y retroalimentación, permitiendo que cada miembro del equipo de desarrollo expresara sus opiniones y sugerencias para optimizar el proceso de desarrollo en futuras iteraciones.

Identificación de fallos

Se identificó un fallo al consultar el reporte de Balanza de Comprobación. que los programadores confundían el periodo general con los específicos, esto causaba que los reportes no presentaran información.

Análisis de la raíz del fallo

Se identificó el fallo, el cual surgió por la falta de capacitación por parte del equipo referente a los periodos general y específicos.

Solución

Se resolvió de la siguiente manera:

1. Se revisó y corrigió el código del Backend.
2. Se ajusto el Backend para que apuntara a los periodos específicos relacionado a los asientos contables.
3. Se ajusto los procedimientos almacenados y la funciones, para que reconociera los periodos específicos.

10.3. Pruebas

10.3.1. Pruebas Funcionales

Tabla 28

Criterios de aceptación para Agregar Documento Contable Tipo

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Criterios de aceptación
24	Director y Contador	Agregar Documento Contable Tipo.	Añadir Documento Contable Tipo, al sistema SIGFA.	• Todos los campos son requeridos. • El nombre del Documento Contable Tipo, debe ser único.

Tabla 29

Criterios de aceptación para Mostrar los estados del cheque

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Criterios de aceptación
30	Director y Contador	Mostrar los estados del cheque.	Tener control del estado de cada cheque.	• El listado de estado único. • Cada estado debe tener un orden cronológico (Fecha, Usuario).

Tabla 30

Criterios de aceptación para Agregar Asiento Contable

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Criterios de aceptación
34	Contador	Agregar Asiento Contable.	Añadir Asiento Contable al sistema SIGFA.	• Debe existir un programa contable relacionado a un periodo especifico (programa contable especifico). • Todos los campos son requeridos. • Él debe total y haber total son del mismo monto, la resta de estos dos da como resultado 0 (debe total-haber total). • El usuario, debe estar vinculado a un programa contable

Tabla 31

Criterios de aceptación para Agregar Nota de Crédito

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Criterios de aceptación
41	Contador	Agregar Nota de Crédito.	Añadir Nota de Crédito al sistema SIGFA.	• Debe existir un programa contable relacionado a un periodo especifico (programa contable especifico). • Todos los campos son requeridos. • Los impuestos de esta nota, se debe tener vinculado al catálogo de impuesto.

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Criterios de aceptación
				• El usuario, debe estar vinculado a un programa contable

Tabla 32

Criterios de aceptación para Agregar Nota de Debito

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Criterios de aceptación
48	Contador	Agregar Nota de Débito.	Añadir Agregar Nota de Debito al sistema SIGFA.	• Debe existir un programa contable relacionado a un periodo especifico ((programa contable especifico)). • Todos los campos son requeridos. • Los impuestos de esta nota, se debe tener vinculado al catálogo de impuesto. • El usuario, debe estar vinculado a un programa contable.

Tabla 33

Criterios de aceptación para Generar reporte de Balance general por programa específico

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Criterios de aceptación
63	Contador y Director	Generar reporte de Balance general por programa específico.	Tener información sobre las cuentas de pasivo, activo y patrimonio de la UNI.	• Debe existir un programa contable relacionado a un periodo especifico (programa contable especifico). • El usuario, debe estar vinculado a un programa contable. • Luego de generar el reporte se debe relacionar con el programa contable especifico. • Debe mostrar las cuentas de Activo, Pasivo y Patrimonio, sin importar si es negativo o positivo.

Tabla 34

Criterios de aceptación para Generar reporte de Balanza de comprobación por programa específico

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Criterios de aceptación
64	Contador y Director	Generar reporte de Balanza de comprobación por	Tener información de los saldos deudores y	• Debe existir un programa contable relacionado a un periodo especifico (programa contable especifico).

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Criterios de aceptación
		programa específico	acreedores de la UNI.	• El usuario, debe estar vinculado a un programa contable. • Luego de generar el reporte se debe relacionar con el programa contable especifico • Solo debe mostrar las cuentas que tiene un saldo (monto), sin importar si es negativo o positivo.

Tabla 35

Criterios de aceptación para Generar reporte de Estado de Resultado por programa específico

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Criterios de aceptación
65	Contador y Director	Generar reporte de Estado de Resultado por programa específico	Tener información de que si la UNI tuvo ganancias o pérdidas económicas.	• Debe existir un programa contable relacionado a un periodo especifico (programa contable especifico). • El usuario, debe estar vinculado a un programa contable. • Luego de generar el reporte se debe relacionar con el

N.	Como <Tipo de usuario>	Quiero <realizar alguna tarea>	Para que pueda <lograr algún objetivo>	Criterios de aceptación
				programa contable especifico • Solo debe mostrar las cuentas de ingresos y gastos (egresos, dependiente del nombre de la cuenta).

Tabla 36:

Agregar Documento Contable Tipo

Agregar Documento Contable Tipo	Rol	Identificador
	Contador	04
Descripción: La prueba consiste en agregar un Documento Contable Tipo en el Sistema SIGFA.		
Prerrequisitos: Haber iniciado sesión como Contador.		
Pasos: 1. Ingresar a la pantalla de Documento Contable Tipo. 2. Ingresar a la opción de agregar Documento Contable Tipo. 3. Ingresar todos los datos necesarios para agregar un Documento Contable Tipo 4. Si los datos son incorrectos o el Documento Contable Tipo ya existe el sistema desplegara un mensaje. 5. Si los datos son correctos el sistema guardara el Documento Contable Tipo.		
Resultado esperado: Nuevo Documento Contable Tipo agregado exitosamente.		
Resultado obtenido: Nueva Documento Contable Tipo agregado exitosamente.		

Tabla 37

Mostrar los estados del cheque

Mostrar los estados del cheque	Rol	Identificador
	Director	05
Descripción: La prueba consiste en mostrar todos los estados por el cual ha pasado un cheque en el Sistema SIGFA.		
Prerrequisitos: Haber iniciado sesión como Director.		
Pasos: 1. Ingresar a la pantalla de Cheque. 2. Seleccionar el Cheque en el cual desea ver los estados. 3. Ingresar a la opción de estados del Cheque. 4. Se desplegará otra pantalla en la cual se muestran todos los estados por el cual ha pasado el Cheque.		
Resultado esperado: Visualización de los estados del Cheque.		
Resultado obtenido: Visualización de los estados del Cheque.		

Tabla 38

Agregar Asiento Contable

Agregar Asiento Contable	Rol	Identificador
	Contador	06
Descripción: La prueba consiste en agregar un Comprobante de Diario en el Sistema SIGFA.		
Prerrequisitos: Haber iniciado sesión como Contador.		
Pasos: 1. Ingresar a la pantalla de Asiento Contable. 2. Ingresar a la opción de agregar Comprobante de Diario. 3. Ingresar todos los datos necesarios para agregar un Comprobante de Diario. 4. Agregar las líneas Contables en el Comprobante de Diario. 5. Si los datos son incorrectos el sistema desplegara un mensaje. 6. Si los datos son correctos el sistema guardara el Comprobante de Diario.		
Resultado esperado: Nuevo Comprobante de Diario agregado exitosamente.		
Resultado obtenido: Nueva Comprobante de Diario agregado exitosamente.		

Tabla 39

Agregar Nota de Crédito

Agregar Nota de Crédito	Rol	Identificador
	Contador	07
Descripción: La prueba consiste en agregar una Nota de Crédito en el Sistema SIGFA.		
Prerrequisitos: Haber iniciado sesión como Contador.		
Pasos: 1. Ingresar a la pantalla de Nota de Crédito. 2. Ingresar a la opción de agregar Nota de Crédito. 3. Ingresar todos los datos necesarios para agregar una Nota de Crédito. 4. Si los datos son incorrectos o están duplicados el sistema desplegara un mensaje de error. 5. Si los datos son correctos el sistema guardara la Nota de Crédito.		
Resultado esperado: Nueva Nota de Crédito agregada exitosamente.		
Resultado obtenido: Nueva Nota de Crédito agregada exitosamente.		

Tabla 40

Agregar Nota de Debito

Agregar Nota de Debito	Rol	Identificador
	Contador	08
Descripción: La prueba consiste en agregar una Nota de Debito en el Sistema SIGFA.		
Prerrequisitos: Haber iniciado sesión como Contador.		
Pasos: 1. Ingresar a la pantalla de Nota de Débito. 2. Ingresar a la opción de agregar Nota de Débito. 3. Ingresar todos los datos necesarios para agregar una Nota de Débito. 4. Si los datos son incorrectos o están duplicados el sistema desplegara un mensaje de error. 5. Si los datos son correctos el sistema guardara la Nota de Débito.		
Resultado esperado: Nueva Nota de Débito agregada exitosamente.		
Resultado obtenido: Nueva Nota de Débito agregada exitosamente.		

Tabla 41

Generar reporte de Balance general por programa específico

Generar reporte de Balance general por programa específico	Rol	Identificador
	Contador, Director	09
Descripción: La prueba consiste en Generar reporte de Balance general por programa específico en el Sistema SIGFA.		
Prerrequisitos: Haber iniciado sesión como Contador o Director.		
Pasos: 1. Ingresar a la pantalla de Generar reporte de Balance general. 2. Seleccionar la opción de Programa Específico. 3. Seleccionar el Periodo en el cual desea visualizar el reporte de Balance general. 4. El sistema generara el Reporte de Balance general.		
Resultado esperado: Reporte de Balance general por programa específico Generado.		
Resultado obtenido: Reporte de Balance general por programa específico Generado.		

Tabla 42

Generar reporte de Balanza de comprobación por programa específico

Generar reporte de Balanza de comprobación por programa específico	Rol	Identificador
	Contador, Director	10
Descripción: La prueba consiste en Generar reporte de Balanza de comprobación por programa específico en el Sistema SIGFA.		
Prerrequisitos: Haber iniciado sesión como Contador o Director.		
Pasos: 1. Ingresar a la pantalla de Generar reporte de Balanza de comprobación. 2. Seleccionar la opción de Programa Específico. 3. Seleccionar el Periodo en el cual desea visualizar el reporte de Balanza de comprobación. 4. El sistema generara el Reporte de Balance general.		
Resultado esperado: Reporte de Balanza de comprobación por programa específico Generado.		
Resultado obtenido: Reporte de Balanza de comprobación por programa específico Generado.		

Tabla 43

Generar reporte de Estado de Resultado por programa específico

Generar reporte de Estado de Resultado por programa específico	Rol	Identificador
	Contador, Director	11
Descripción: La prueba consiste en Generar reporte de Estado de Resultado por programa específico en el Sistema SIGFA.		
Prerrequisitos: Haber iniciado sesión como Contador o Director.		
Pasos: 1. Ingresar a la pantalla de Generar Reporte de Estado de Resultado. 2. Seleccionar la opción de Programa Específico. 3. Seleccionar el Periodo en el cual desea visualizar el reporte de Estado de Resultado. 4. El sistema generara el Reporte de Estado de Resultado.		
Resultado esperado: Reporte de Estado de Resultado por programa específico Generado.		
Resultado obtenido: Reporte de Estado de Resultado por programa específico Generado.		

10.3.2. Pruebas de Seguridad

Figura 74

Código de la prueba unitaria realizada al servicio Dashboard y generar la posición financiera de las cuentas de patrimonios

```
1 referencia | 2/2 pasando
private IServiceProvider CreateContext(string nameDB)
{
    var services = new ServiceCollection();
    var builder = WebApplication.CreateBuilder();
    services.AddDbContext<DataBaseContext>(o =>{o.UseSqlServer(builder.Configuration.GetSection("ConnectionStrings:SigfaDatabase").Value);}
    , ServiceLifetime.Transient);
    return services.BuildServiceProvider();
}

[Test]
[TestCase(HttpStatusCode.OK)]
[TestCase(HttpStatusCode.InternalServerError)]
| 0 referencias
public  void Test1(HttpStatusCode code)
{
    var nameDb=Guid.NewGuid().ToString();
    var serviceProvider=CreateContext(nameDb);
    var db = serviceProvider.GetService<DataBaseContext>();
    DashboardService service=new DashboardService(db);
    var responseService = service.ObtenerGraficaActivoPasivoPatrimonios(new DashBoard { IdPrograma = 4 });
    Assert.AreEqual(12, (responseService.Count));

}
```

Esta figura muestra el código fuente de la prueba unitaria para consultar el Dashboard, mediante inyección de servicios e instancia física de la base de datos

existente. El cual debe retornar 12 registros, que pertenece a los meses contables y el total de saldos de estas cuentas contables de patrimonio.

Figura 75

Pruebas unitarias del servicio Dashboard

DashBoardServicesTest (2)	SIFA.Api.Testing	Correcta	2.1 s
Lentas (> 1 s) (1)	SIFA.Api.Testing	Correcta	2.1 s
Test1(OK)	SIFA.Api.Testing	Correcta	2.1 s
Rápidas (< 100 ms) (1)	SIFA.Api.Testing	Correcta	19 ms
Test1(InternalServerError)	SIFA.Api.Testing	Correcta	19 ms

Resultado del test unitario, luego de la ejecución.

Figura 76

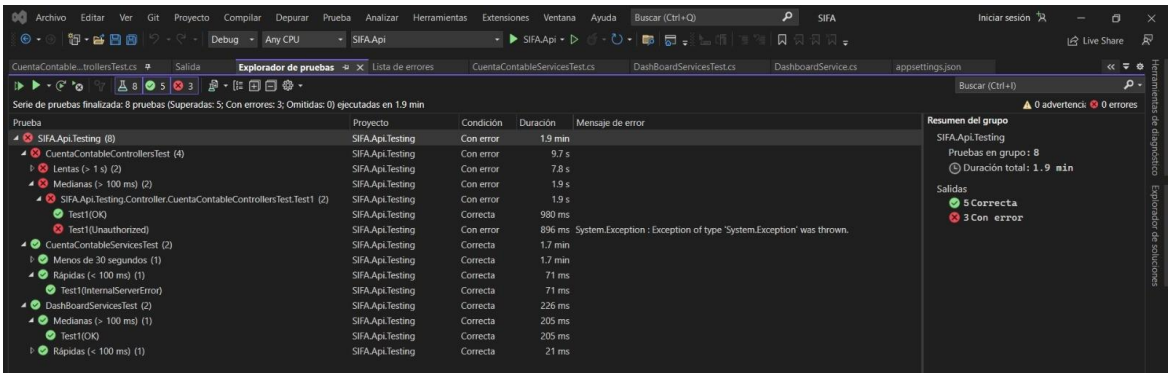
Código de las pruebas unitarias para obtener las Cuentas Contables

```
[Test]
[TestCase(HttpStatusCode.OK)]
[TestCase(HttpStatusCode.InternalServerError)]
| 0 referencias
public  void Test1(HttpStatusCode code)
{
    var nameDb=Guid.NewGuid().ToString();
    var serviceProvider=CreateContext(nameDb);


    var db = serviceProvider.GetService<DataBaseContext>();
    ProgramaService programa = new ProgramaService(db);
    CuentaContableService cuentaContableService = new CuentaContableService(db, programa);
    var responseService = cuentaContableService.ObtenerListadoCuentasContables(new Models.CuentaContableRead
    { OrdenarPor = "IdCuentaContable" },2);


    Assert.IsNotNull(responseService);
    Assert.That(responseService.TieneItems,"Si tiene items");


}
```

Esta figura muestra el código fuente de la prueba unitaria para consultar el listado de Cuentas Contables, mediante la inyección de servicios e instancia física de la base de datos existente. El cual debe retornar la lista de cuentas contables activas ordenadas por IdCuentaContable y se verifica que existan registros.

Figura 77

Pruebas unitarias de las Cuentas Contables



Resultado del test unitario, luego de la ejecución.

Figura 78

Resultado a nivel de datos, de la prueba unitarias del servicio de las Cuentas Contables

Registros de cuentas contables ordenados por IdCuentaContable descendente paginado por 10 registros activos.

10.4. Índice de CD adjunto

1. Monografía en PDF.
2. Manual de Usuario.
3. Manual Técnico.
4. Código Fuente.