



"LA ESENCIA DE LA GRANDEZA RADICA EN LAS RAÍCES"

INFORME TÉCNICO DE RESIDENCIAS PROFESIONALES

TÍTULO DEL PROYECTO:

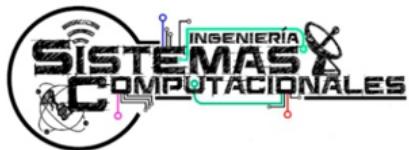
**SISTEMA INTEGRAL DE INDICADORES PARA EL SGC DEL
ITT**

PRESENTA:

URBANO BALLESTEROS RODRIGUEZ
VIANET VARELA CHIRINOS

ASESORES:

ING. JUAN CARLOS CAMPOS CABELLO
ING. EDSON JESÚS ROSAS MARTÍNEZ



Fecha de entrega: 15 de febrero de 2017.

Tabla de contenido

Lista de figuras	III
Lista de tablas	v
1. Introducción	1
2. Generalidades	5
2.1. Objetivos	5
2.1.1. Objetivo general	5
2.1.2. Objetivos específicos	5
2.2. Justificación	5
2.3. Caracterización de la empresa en la que participó	6
2.3.1. Datos generales de la empresa	6
2.3.2. Descripción del departamento o área de trabajo.	7
2.4. Problemas a resolver	7
3. Fundamento teórico	9
3.1. Indicador	9
3.2. Sistema de gestión	9
3.3. Arquitectura Cliente-Servidor	11
3.4. Lenguajes en cliente (Frontend)	12
3.4.1. HTML Y HTML 5	12
3.4.2. CSS	13
3.4.3. Javascript	13
3.5. Lenguajes en servidor (Backend)	14
3.5.1. C#	14
3.6. Frameworks	18
3.6.1. Frameworks en cliente	18
3.6.2. Frameworks en servidor	19
3.7. Arquitectura de Software	20
3.7.1. MVC (Modelo Vista Controlador)	21
3.8. Base de Datos	23
3.8.1. Lenguaje SQL	24
3.8.2. Microsoft SQL Server	24
3.9. Metodología de desarrollo	27
3.9.1. Modelo en cascada	28

4. Desarrollo	31
4.1. Análisis del sistema	31
4.2. Análisis de requisitos del software	32
4.2.1. Requerimientos físicos	32
4.2.2. Interfaces	33
4.2.3. Usuarios y factores humanos	33
4.2.4. Funcionalidad	34
4.3. Diseño del sistema	37
4.3.1. Estructura de los datos	37
4.4. Caracterización de la interfaz	41
4.5. Programación	42
4.5.1. Manejo de interfaces de mensajes emergentes	42
5. Pruebas y resultados	44
5.1. Pruebas y resultados de las páginas principales del SII.	44
5.2. Pruebas y resultados de módulo Departamentos.	45
5.2.1. Alta de departamento	47
5.2.2. Actualización de departamentos	48
5.2.3. Baja de departamentos	48
5.3. Pruebas y resultados de módulo Fórmulas.	51
5.3.1. Alta de fórmula	51
5.3.2. Actualización de fórmulas	53
5.3.3. Baja de fórmulas	53
5.4. Pruebas y resultados de módulo Colecciones.	55
5.4.1. Alta de colección	55
5.4.2. Actualización de colecciones	57
5.4.3. Baja de colección	57
5.4.4. Detalle de colecciones	60
5.5. Pruebas y resultados de módulo Proceso.	63
5.5.1. Alta de Procesos	63
5.5.2. Actualización de procesos	64
5.5.3. Baja de procesos	67
5.5.4. Proceso de periodo de indicadores	67
5.5.5. Visualización del detalle de procesos	67
6. Conclusiones	70
Bibliografía	71

Lista de figuras

2.1.	Croquis de ubicación.	6
2.2.	Croquis de ubicación.	7
3.1.	Diagrama de modelo cliente-servidor.	12
3.2.	Diagrama de flujo de una petición en MVC.	22
4.1.	Diagrama del proceso de configuración de fórmulas de indicadores.	35
4.2.	Diagrama del proceso de información de indicadores.	36
4.3.	Diagrama entidad relación del sistema de indicadores.	38
4.4.	Diagrama entidad relación con tablas compuestas del sistema de indicadores.	39
4.5.	Diagrama relacional con campos del sistema de indicadores.	39
5.1.	Pantalla de inicio del Sisitema Integral de Información del ITT.	45
5.2.	Pantalla de inicio del Sisitema Integral de Indicadores.	46
5.3.	Pantalla principal de departamentos.	46
5.4.	Ventana emergente para creación de departamentos.	47
5.5.	Mensaaje de validación departamentos.	48
5.6.	Mensaaje de grabación satisfactoria de departamento.	49
5.7.	Listado de departamentos.	49
5.8.	Ventana emergente de actualización de departamento.	50
5.9.	Mensaje de advertencia al eliminar departamento.	50
5.10.	Pantalla principal de fórmulas.	51
5.11.	Ventana emergente para creación de fórmulas.	52
5.12.	Mensaaje de validación fórmulas.	52
5.13.	Mensaaje de grabación satisfactoria de fórmula.	53
5.14.	Listado de fórmulas.	54
5.15.	Ventana emergente de actualización de fórmula.	54
5.16.	Mensaje de advertencia al eliminar fórmula.	55
5.17.	Pantalla principal de colecciones.	56
5.18.	Ventana emergente para creación de colecciones.	56
5.19.	Mensaaje de validación colecciones.	57
5.20.	Mensaaje de grabación satisfactoria de colección.	58
5.21.	Listado de colecciones.	58
5.22.	Ventana emergente de actualización de colecciones.	59
5.23.	Mensaje de advertencia al eliminar colección.	59
5.24.	Pantalla de inicio del detalle de colecciones.	60
5.25.	Ventana emergente para agregar formulas a colección.	61
5.26.	Mensaje de grabación satisfactoria de formulas en colección.	61
5.27.	Listado de fórmulas en colección.	62
5.28.	Mensaje de advertencia al eliminar una formula de colección.	62

5.29.	Pantalla principal de procesos.	63
5.30.	Ventana emergente para creación de periodos de indicadores.	64
5.31.	Mensaje de validación en proceso de indicadores.	65
5.32.	Mensaje de grabación satisfactoria de procesos de indicadores.	65
5.33.	Listado de procesos.	66
5.34.	Ventana emergente de actualización de periodos de indicadores.	66
5.35.	Mensaje de advertencia al eliminar un periodo de indicadores.	67
5.36.	Pantalla de procesos de indicadores listos para procesar.	68
5.37.	Alerta emergente con alerta de reproceso.	68
5.38.	Pantalla con el detalle de periodo de indicadores.	69

Lista de tablas

3.1.	Tabla de departamentos organizadas por subdirecciones	10
3.2.	Requisitos de Hardware de Microsoft SQL Server	25
4.1.	Detalle de los campos de la tabla departamentos.	40
4.2.	Detalle de los campos de la tabla formulasc.	40
4.3.	Detalle de los campos de la tabla gformulase.	40
4.4.	Detalle de los campos de la tabla gformulasr.	40
4.5.	Detalle de los campos de la tabla procesoe.	41
4.6.	Detalle de los campos de la tabla procesod.	41

Capítulo 1

Introducción

En la actualidad la calidad de los servicios entendida como la satisfacción a las necesidades y expectativas de las organizaciones debe estar en constante mejora, es decir, aumentar la eficiencia de un sistema aplicando una política de calidad, los objetivos de calidad, el análisis de los datos y las acciones correctivas y preventivas, identificando de que manera cada uno de estos procesos contribuyen a la mejora constante y optimización, buscando simplificar las operaciones más complejas, mejorando los tiempos de respuesta en un Sistema de Gestión de Calidad, por lo que es necesario realizar revisiones constante de cada proceso, de esta manera podremos examinar con atención y cuidado para corregir errores o para comprobar que funcione correctamente la organización y así poder tomar acciones para perfeccionarnos

Hoy en día las organizaciones cuentan con un Sistema de Gestión de Calidad, sin tener la idea exacta de lo que esto significa, su concepto y los beneficios que se pueden tener cuando este se implementa adecuadamente con compromiso y liderazgo. Sistema de Gestión de calidad significa planear, ejecutar y controlar las actividades realizadas dentro de una organización, permitiendo con ello alcanzar el objetivo estipulado en la misión y brindar servicios con altos estándares de calidad, los cuales son medidos a través de indicadores de satisfacción de los usuarios. Como concepto tenemos que es una herramienta que permite planear, controlar y mejorar aquellos elementos de una organización que influyen en el cumplimiento de los objetivos. Logrando como beneficio el perfeccionamiento de la organización

El Tecnológico Nacional de México (TecNM) cuenta con un Sistema para la Gestión de la Calidad en el cual contiene las normativas para la realización de las mejoras prácticas y con ello brindar a los estudiantes el mejor servicio de educación

Por lo anterior dicho proyecto va orientado a la realización de un Sistema Integral de Indicadores para el Sistema de Gestión de Calidad del Instituto Tecnológico de Tláhuac

Con la realización de dicho Sistema de indicadores se pretende que los datos proporcionados por los indicadores de gestión cuenten con la características de relevancia, comprensibilidad, comparabilidad, oportunidad, consistencia y fiabilidad, así mismo, reunan los rasgos de relevancia, verificabilidad, ausencia de sesgos, posibilidad de cuantificación, aceptabilidad institucional, factibilidad económica, comparabilidad y oportunidad. Dichos indicadores reunirán las siguientes características

- a) Relevante para la gestión: Que aporte información para informar, controlar, evaluar y tomar decisiones.
- b) No ambiguo e inequívoco: Que no permita interpretaciones contrapuestas.
- c) Pertinente: Que resulte adecuado a lo que se pretende medir.
- d) Objetivo: Que no esté influenciado por factores externos.

En la actualidad la calidad de los servicios entendida como la satisfacción a las necesidades y expectativas de las organizaciones debe estar en constante mejora, es decir, aumentar la eficiencia de un sistema aplicando una política de calidad, los objetivos de calidad, el análisis de los datos y las acciones correctivas y preventivas, identificando de que manera cada uno de estos procesos contribuyen a la mejora constante y optimización, buscando simplificar las operaciones más complejas, mejorando los tiempos de respuesta en un Sistema de Gestión de Calidad, por lo que es necesario realizar revisiones constante de cada proceso, de esta manera podremos examinar con atención y cuidado para corregir errores o para comprobar que funcione correctamente la organización y así poder tomar acciones para perfeccionarnos.

Hoy en día las organizaciones cuentan con un Sistema de Gestión de Calidad, sin tener la idea exacta de lo que esto significa, su concepto y los beneficios que se pueden tener cuando este se implementa adecuadamente con compromiso y liderazgo. Sistema de Gestión de calidad significa planear, ejecutar y controlar las actividades realizadas dentro de una organización, permitiendo con ello alcanzar el objetivo estipulado en la misión y brindar servicios con altos estándares de calidad, los cuales son medidos a través de indicadores de satisfacción de los usuarios. Como concepto tenemos que es una herramienta que permite planear, controlar y mejorar aquellos elementos de una organización que influyen en el cumplimiento de los objetivos. Logrando como beneficio el perfeccionamiento de la organización.

El Tecnológico Nacional de México (TecNM) cuenta con un Sistema para la Gestión de la Calidad en el cual contiene las normativas para la realización de las mejoras prácticas y con ello brindar a los estudiantes el mejor servicio de educación.

Por lo anterior dicho proyecto va orientado a la realización de un Sistema Integral de Indicadores para el Sistema de Gestión de Calidad del Instituto Tecnológico de Tláhuac.

Los indicadores son herramientas necesarias para poder medir, y con ello, controlar los procesos con el objetivo de realizar una gestión eficaz de los mismos.

Según la AECA, los indicadores son unidades de medida que permiten el seguimiento y la evaluación periódica de las variables clave de una organización, mediante su comparación con los correspondientes referentes internos y externos. Por su parte, Gómez Rodríguez expone que “un indicador debe representar las magnitudes más importantes del sistema así como dar respuesta a todo tipo de variaciones del objeto de medición”. De manera más concreta, y específica para los indicadores de gestión, De Forn señala que estos indicadores tienen que permitir la medición en un doble sentido: desde la vertiente de los resultados obtenidos y desde los recursos utilizados.

Independientemente de la tipología del indicador, hay que destacar que un indicador:

- a) Es una síntesis cuantitativa de uno o varios aspectos concretos de una determinada realidad.
- b) Es una medida estadística, de resumen, referida a la cantidad o magnitud de un conjunto de parámetros o atributos. Permite ubicar o clasificar las unidades de análisis (personas, organizaciones, etc.) con respecto al concepto o conjunto de variables o atributos que se están analizando.
- c) Es una magnitud utilizada para medir o comparar los resultados efectivamente obtenidos, en la ejecución de un proyecto, programa o actividad.
- d) Permite identificar las acciones cuyo efecto no se asemejan al estándar planteado.

Con la realización de dicho Sistema de indicadores se pretende que los datos proporcionados por los indicadores de gestión cuenten con las características de relevancia, comprensibilidad, comparabilidad, oportunidad, consistencia y fiabilidad, así mismo, reunan los rasgos de relevancia, verificabilidad, ausencia de sesgos, posibilidad de cuantificación, aceptabilidad institucional, factibilidad económica, comparabilidad y oportunidad.

Dichos indicadores reuniran las siguientes características:

- a) Relevante para la gestión: Que aporte información para informar, controlar, evaluar y tomar decisiones.
- b) No ambiguo e inequívoco: Que no permita interpretaciones contrapuestas.
- c) Pertinente: Que resulte adecuado a lo que se pretende medir.
- d) Objetivo: Que no esté influenciado por factores externos.
- e) Sensible: Que capte también los cambios pequeños.
- f) Accesible: Que sea fácil de calcular y de interpretar.

Esta es la introducción de mi proyecto

Una forma efectiva de escribir la introducción es tratar de seguir el patrón de un triángulo, esto significa que primero se comienza con afirmaciones generales relativas al tema del trabajo reportado en el informe y poco a poco se va precisando más lo que se dice hasta llegar al meollo del trabajo descrito. Al final de este triángulo se debe decir alguna frase o párrafo que resuma el contenido del reporte.

En términos prácticos podría decirse que una introducción obedece a la formulación de las siguientes preguntas que antes de hacerlas el lector; se ha hecho ya el propio investigado.

- a) ¿Cuál es el tema del trabajo?
- b) ¿Por qué se hace el trabajo?
- c) ¿Cómo está pensado el trabajo?
- d) ¿Cuál es el método empleado en el trabajo?
- e) ¿Cuáles son las limitaciones del trabajo?

Se recomienda que la información se haga una vez terminado el desarrollo del trabajo, a fin de que no se proponga metas que no se van a alcanzar. En este sentido, no debe confundirse la introducción con el plan de trabajo que el investigador elabora al principio, antes de abordar el estudio de los temas propuestos.

El último párrafo de la introducción debe contener una descripción muy corta, no más de una frase, del contenido de cada uno de los capítulos o secciones en los cuales está dividido el trabajo. Tal y como se hizo en capítulo de introducción de esta guía. La numeración se registra a partir de la introducción, siguiendo la numeración subsiguiente.

Algunos puntos que el residente debe cubrir en este capítulo son:

- a) Qué movió al residente a investigar precisamente este tema o problema y no otro.
- b) Cómo abordó el tema o el problema (principios, criterios, métodos y procedimientos).
- c) Una descripción breve de cada una de las partes del informe desde la justificación hasta las conclusiones y recomendaciones.
- d) En el caso de que por políticas de la empresa se exija confidencialidad en algunos aspectos del proyecto, esto deberá ser mencionado en la introducción.

Dependiendo de la extensión del informe completo y de la complejidad, la longitud de la introducción puede variar, de una página completa como mínimo a unas cinco o diez páginas. La introducción debe tener sustancia, es decir, su contenido debe transmitir lo más importante del trabajo reportado en el informe. No debe entrarse en demasiado detalle, pero tampoco deben decirse vaguedades, sino aspectos bien concretos del trabajo. Muchas personas lo primero que leen son la introducción y las conclusiones, para tener una buena idea del trabajo realizado, por lo tanto es importante redactar con cuidado estas dos secciones, pues son la primera impresión que se va a llevar el lector del trabajo. En general, la forma como está escrita esta guía debe servir también de ejemplo de lo que es un reporte, cómo organizarlo y presentarlo.

Capítulo 2

Generalidades

En este capítulo se presenta de manera detallada los objetivos del proyecto, además de la información de la institución donde se desarrolló el mismo.

2.1. Objetivos

En esta sección se detallan los objetivos tanto generales como específicos a cumplir a lo largo del desarrollo del Sistema Integral de Indicadores, el cual coadyuvara con el Sistema de Gestión de la Calidad(SGC).

2.1.1. Objetivo general

Desarrollar un sistema que permita medir, evaluar y controlar los procesos realizados por cada departamento en el Instituto Tecnológico de Tláhuac, basado en los lineamientos establecidos por el TecNM en el Sistema de Gestión de Calidad, para lograr realizar una gestión eficaz y de calidad de los mismos. Este sistema permitirá visualizar los resultados de los indicadores de tal manera que los procesos podrán ser monitoreados de una manera práctica, accesible y en tiempo real.

2.1.2. Objetivos específicos

- Análisis de requerimientos.
- Toma de datos y entrevistas.
- Desarrollo del sistema.
- Implementación de sistema en ambiente de pruebas.
- Implementación de sistema en ambiente de producción.

2.2. Justificación

Dicho proyecto está basado en la necesidad de contar con una herramienta que permita la evaluación automática de los procesos pertenecientes en el Sistema de Gestión de Calidad, esto debido a que no se cuenta con ningún sistema para la realización de dicha evaluación, ya que actualmente los procesos se realizan de manera manual, teniendo así procesos extensos y complicados.

El Sistema Integral de Indicadores facilitará el trabajo realizado, agilizará los tiempos requeridos y permitirá con esto brindar un mejor servicio a la comunidad, además de que dichos indicadores influyen directamente para la toma de decisiones ya que muestran información acerca del cumplimiento de los objetivos, logrando como beneficio el perfeccionamiento de los procesos de la institución.

2.3. Caracterización de la empresa en la que participó

En esta sección se detalla la información general de la institución, así como el área para la cual se desarrolla el proyecto.

2.3.1. Datos generales de la empresa

- **Empresa:** Instituto Tecnológico de Tláhuac.
- **Dirección:** Av. Estanislao Ramírez No.301 Colonia Ampliación Selene C.P. 13420 Tláhuac D.F.

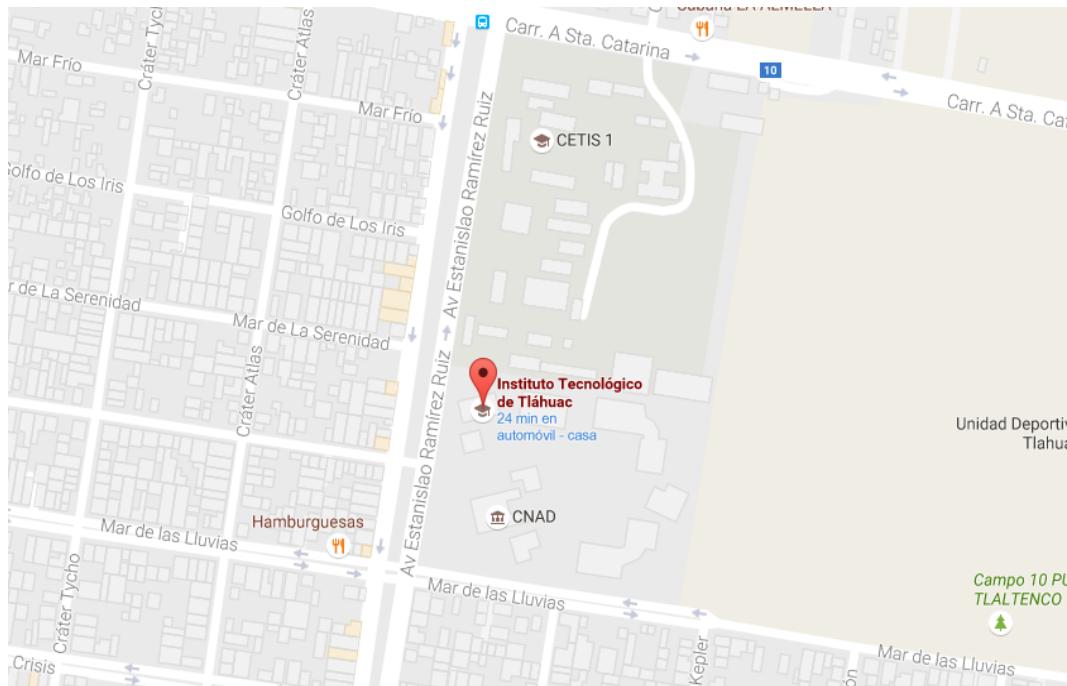


Figura 2.1: Croquis de ubicación.

- **Teléfono:** 5866-0927, 7312-5616, 5841-0560, 2594-4096
- **Dirección de correo electrónico:** sub.academica@ittlahuac.edu.mx
- **Giro:** Educativo.
- **Misión:** Ofrecer un servicio educativo de calidad a través de la mejora continua, personal capacitado, compromiso global e infraestructura de vanguardia.
- **Visión:** Ser una institución de alto desempeño académico, basada en la formación integral de profesionales competitivos a nivel internacional con responsabilidad global.
- **Valores:** Los valores que manejamos en la institución son los que nos ayudan a identificarnos como seres humanos, como personas que están al servicio de la educación, que les gusta la actividad que desarrollan y que están orgullosos de promover un servicio de calidad a la comunidad.
 - Honestidad.
 - Respeto.

- Trabajo en equipo.
 - Vocación de servicio.
 - Comunicación.
- **Políticas de calidad:** La organización establece el compromiso de implementar todos sus procesos orientándolos hacia la satisfacción de sus estudiantes, sustentada en la calidad del proceso educativo, para cumplir con sus requisitos, mediante la eficacia de un sistema de gestión de calidad de mejora continua, conforme a la norma ISO 9001:2008/NMX-CC-9001-IMNC-2008.
 - **Estructura organizacional:** La estructura organizacional del Instituto Tecnológico de Tláhuac se encuentra como se muestra en la figura 2.2

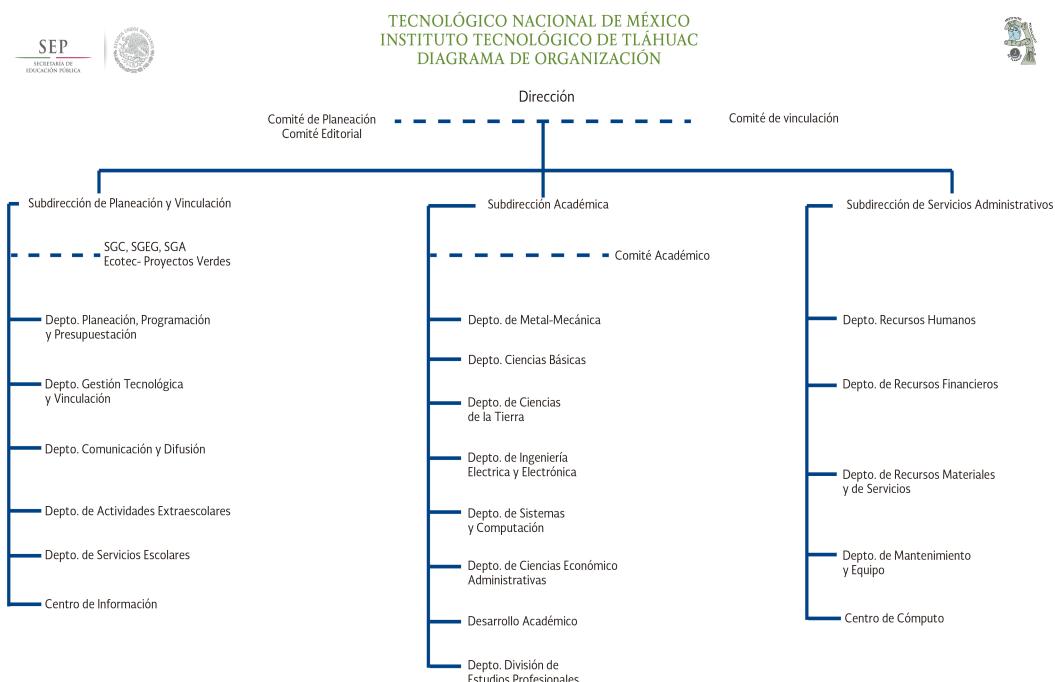


Figura 2.2: Croquis de ubicación.

2.3.2. Descripción del departamento o área de trabajo.

Definición y detalle de el departamento.

2.4. Problemas a resolver

A continuación se detallaran los problemas a resolver con el Sistema Integral de Indicadores.

Dentro de los problemas más importantes a resolver se encuentra la digitalización de los procesos de indicadores se realizan de manera manual. Esto por ende lleva demasiado tiempo y requiere de una gran sincronización entre departamentos para que la información sea confiable. Realizar estas actividades requiere de demasiado esfuerzo por parte de los trabajadores del Instituto Tecnológico de Tláhuac lo que provoca que los trámites sean lentos y tediosos. El Sistema Integral de Indicadores pretende solucionar este problema permitiendo tener la información de los procesos en cualquier momento y de una manera sencilla, dando como resultado una sincronización de información eficiente y por consecuencia procesos más rápidos.

Como segundo problema a resolver es la generación de datos indicadores para la toma de decisiones, la cual de igual manera que los procesos realizados en los departamentos, el sistema permitirá configurar reportes de indicadores para los directivos, permitiendo con esto tener información general de los procesos.

Como último punto el sistema contará con la facilidad de adaptarse a nuevos procesos de indicadores, lo cual permite configurarse de tal manera que cuando la información de un indicador cambie, esta modificación no necesite realizarse un cambio en código.

Capítulo 3

Fundamento teórico

En el tercer capítulo se presentan las técnicas y herramientas utilizadas para adoptar una buena orientación que permita sustentar el desarrollo del Sistema Integral de Indicadores mostrando una investigación detallada con todo lo necesario para entender cada fase del mismo.

3.1. Indicador

El primer concepto que nos debe quedar claro es saber que es un indicador, un indicador es un dato que nos ayuda a medir de manera objetiva la evolución de un sistema de gestión, por consiguiente un sistema de indicadores son datos que nos brindan información cualitativa o cuantitativa, que permiten seguir el desarrollo de un proceso y su evaluación.

3.2. Sistema de gestión

Es una herramienta que permite a una organización planear, ejecutar y controlar las actividades realizadas en esta, aquellas que sean necesarias para el buen desarrollo de su misión. El objetivo de los sistemas es aportar a la institución un camino correcto para lograr el cumplimiento de las metas establecidas.

Todo sistema de medición debe satisfacer los siguientes objetivos:

- Comunicar la estrategia.
- Comunicar las metas.
- Identificar problemas y oportunidades.
- Diagnosticar problemas.
- Entender procesos.
- Definir responsabilidades
- Mejorar el control de la institución.
- Identificar iniciativas y acciones necesarias.
- Medir comportamientos.
- Facilitar la delegación en las personas.
- Integrar la compensación con la actuación.

El Sistema Integral de Indicadores es un sistema de gestión que organiza indicadores, permitiendo mediante su implementación obtener múltiples beneficios a la comunidad del Instituto Tecnológico de Tláhuac los cuales son:

- Mejorará la imagen ante los alumnos que deseen ingresar a dicha institución.
- Se logrará brindar un servicio caracterizado por la tolerancia y la responsabilidad.
- Permitirá contar con información útil para la mejora continua de procesos.
- Disminuirá las demoras en la realización de trámites internos.
- Se realizará una gestión enfocada al beneficio del alumno.
- Se Logrará el compromiso de los directivos con los objetivos organizacionales.
- Permitirá conocer la información de los avances en cuanto a la evaluación de procesos para poder tomar decisiones estratégicas y permitir alcanzar los objetivos.

Con dicho Sistema se pretende abarcar cada uno de los departamentos estratégicos que sustentan al Instituto Tecnológico de Tláhuac permitiendo a los directivos tener a su alcance los datos de los indicadores en cada uno de ellos, buscando una vinculación entre estos para obtener estadísticas correctas y precisas. Los departamentos abordados se muestran en la tabla 3.1.

Tabla 3.1: Tabla de departamentos organizadas por subdirecciones

SUBDIRECCION ACADÉMICA	CIENCIAS BÁSICAS
	CIENCIAS DE LA TIERRA
	ELECTRICA Y ELECTRONICA
	SISTEMAS Y COMPUTACION
	CIENCIAS ECONÓMICO-ADMINISTRATIVAS
	METAL-MECÁNICA
	DESARROLLO ACADÉMICO
	DIVISIÓN DE ESTUDIOS PROFESIONALES
	RECURSOS HUMANOS
SUBDIRECCIÓN ADMINISTRATIVA	RECURSOS FINANCIEROS
	RECURSOS MATERIALES Y DE SERVICIOS
	CENTRO DE COMPUTO
	MANTENIMIENTO Y EQUIPO
	GESTIÓN TECNOLÓGICA Y VINCULACIÓN
SUBDIRECCIÓN DE PLANEACIÓN Y VINCULACIÓN	COMUNICACIÓN Y DIFUSIÓN
	ACTIVIDADES EXTRAESCOLARES
	SERVICIOS ESCOLARES
	CENTRO DE INFORMACIÓN
	PLANEACIÓN, PROGRAMACIÓN Y PRESUPUESTACIÓN

Los departamentos antes mencionados se encuentran detallados en el sitio oficial del Instituto Tecnológico de Tláhuac en la sección de departamentos.

Para el desarrollo del Sistema Integral de Indicadores se realizó una investigación acerca de las tecnologías que se utilizarían. Estas se encuentran clasificadas en dos partes importantes en el desarrollo web, la parte del cliente y la parte del servidor.

Las tecnologías que se encuentran en la parte del cliente son todas aquellas que son ejecutadas por el navegador web como son Javascript, HTML y CSS.

Las tecnologías del lado del servidor son todas aquellas en las cuales su ejecución se encuentra del lado del servidor como son C#, VB, Java, Python y PHP. Estos lenguajes tienen la capacidad de comunicarse directamente con la base de datos, y con esto realizar los procesos necesarios para formar el resultado solicitado por el cliente.

Dentro de los lenguajes de servidor se encuentran los lenguajes de consulta de base de datos, los cuales como su nombre lo dice sirven para manipular la información de una base de datos. El lenguaje de manipulación de datos que se utiliza en los diferentes motores de base de datos es SQL.

A continuación se definirán las tecnologías del cliente utilizadas para el desarrollo del sistema.

3.3. Arquitectura Cliente-Servidor

Se puede definir la computación cliente-servidor como una arquitectura distribuida que permite obtener acceso a información de manera transparente para el usuario final aun en entornos multiplataforma.

La arquitectura cliente-servidor es el resultado de la integración de dos culturas, por un lado la del Mainframe que aporta la capacidad de almacenamiento, integridad y acceso a la información, y por otro lado la del computador, la cual aporta la facilidad de uso a bajo costo además, provee de una presentación atractiva y una gran variedad en productos y aplicaciones.

El funcionamiento de la arquitectura cliente-servidor es la siguiente:

- Solicitud de información del cliente al servidor.
- El servidor recibe la petición del cliente.
- El servidor procesa la solicitud.
- El servidor regresa el resultado con la información solicitada.
- El cliente recibe la información y la procesa.

La arquitectura cliente-servidor está basada en la idea del servicio, en el que el cliente es un proceso consumidor de servicios y el servidor es un proceso proveedor de servicios, esta relación está establecida en función del intercambio de mensajes que es el único elemento de acoplamiento entre ambos.

Las partes fundamentales de la arquitectura cliente-servidor son las siguientes:

- **Proceso en el cliente:** Es el que inicia el diálogo mediante una petición.
- **Proceso en el servidor:** Es quien se encuentra en espera de que lleguen peticiones de servicio y da respuesta a las mismas.

-
- **Middleware:** Es el medio por el cual se transmite información entre cliente y servidor, permitiendo el envío y recepción de mensajes.

En la figura 3.1 se muestra el diagrama que ilustra lo anterior.

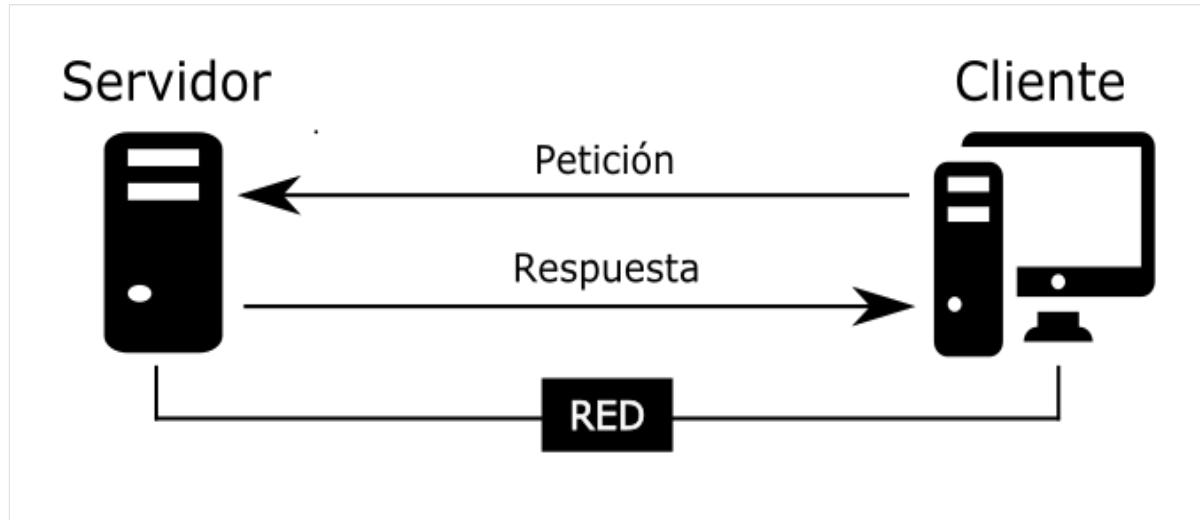


Figura 3.1: Diagrama de modelo cliente-servidor.

3.4. Lenguajes en cliente (Frontend)

3.4.1. HTML Y HTML 5

Definiéndolo de forma sencilla, *"HTML es lo que se utiliza para crear todas las páginas web de Internet"*. Más concretamente, HTML es el lenguaje con el que se “escriben” la mayoría de páginas web.

Los diseñadores utilizan el lenguaje HTML para crear sus páginas web, los programas que utilizan los diseñadores generan páginas escritas en HTML y los navegadores que utilizamos los usuarios muestran las páginas web después de leer su contenido HTML.

Aunque HTML es un lenguaje que utilizan los ordenadores y los programas de diseño, es muy fácil de aprender y escribir por parte de las personas.

El lenguaje HTML es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado World Wide Web Consortium, más conocido como W3C. Como se trata de un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de forma muy similar en cualquier navegador de cualquier sistema operativo.

El propio W3C define el lenguaje HTML como *“un lenguaje reconocido universalmente y que permite publicar información de forma global”*. Desde su creación, el lenguaje HTML ha pasado de ser un lenguaje utilizado exclusivamente para crear documentos electrónicos a ser un lenguaje que se utiliza en muchas aplicaciones electrónicas como buscadores, tiendas online y banca electrónica.

El origen de HTML se remonta al año de 1980, cuando el físico Tim Berners-Lee, trabajador del CERN propuso un nuevo sistema de hipertexto para compartir documentos.

Los sistemas de hipertexto habían sido desarrollados años antes. En el ámbito de la informática el hipertexto permitía que los usuarios accedieran a la información relacionada con los documentos electrónicos que estaba visualizando. De cierta manera, los primitivos sistemas de hipertexto podrían asimilarse a los enlaces de las páginas web actuales.

Con el tiempo este sistema de hipertexto fue evolucionando convirtiéndose en el lenguaje de marcado más utilizado en la actualidad, siendo su última versión la conocida HTML5.

HTML5 establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos, permitiendo concentrar básicamente tres características:

- Estructura.
- Estilo.
- Funcionalidad.

Aunque oficialmente no fue declarado, HTML5 es considerado como la combinación de HTML, CSS y Javascript ya que estas tecnologías son altamente dependientes y actúan como una sola unidad organizada bajo la especificación de HTML5. HTML está a cargo de la estructura, CSS presenta esta estructura y Javascript se encarga de darle funcionalidad.

3.4.2. CSS

Como ya se mencionó anteriormente el conjunto de tecnologías que trabajan en conjunto con HTML para formar HTML5 no solamente incluye los nuevos elementos que permiten que la definición de la estructura de un sistema sea más fácil, si no también es de vital importancia hablar de la parte que permite darle una buena presentación.

Oficialmente CSS no tiene nada que ver con HTML5 ya que es un complemento desarrollado para superar las limitaciones y reducir la complejidad de HTML. En un principio HTML proveía de atributos que permitían definir estilos esenciales para cada elemento, pero a medida que el lenguaje evolucionó, la escritura de códigos se volvió compleja y no pudo satisfacer las necesidades demandadas por los diseñadores, en consecuencia, CSS pronto fue adoptado gracias a que este permite separar la estructura de la presentación. Desde entonces CSS se ha convertido en uno de los lenguajes más importantes de la actualidad enfocado en las necesidades de los diseñadores.

En la versión 3 de CSS es considerado en el desarrollo de HTML5, debido a esto la integración entre ambos lenguajes es vital para el desarrollo web y es la razón por la que cada vez se habla de HTML5 también se hace referencia a CSS3 aunque oficialmente se trate de dos tecnologías completamente separadas. En estos momentos las nuevas características que se incorporan a CSS3 se han ido incorporando a los navegadores web al igual que las características de HTML5.

3.4.3. Javascript

Javascript es un lenguaje interpretado usado para múltiples propósitos pero solo considerado como un complemento hasta ahora. Una de las innovaciones que ayudó a cambiar el modo en que vemos Javascript fue el desarrollo de nuevos motores de interpretación, creados para acelerar el procesamiento de código. La clave de los motores más exitosos fue transformar el código Javascript en código máquina para lograr velocidades de ejecución similares a aquellas encontradas en aplicaciones de escritorio. Esta mejorada capacidad permitió superar viejas limitaciones de rendimiento y confirmar el lenguaje Javascript como la mejor opción para la web.

Para aprovechar esta prometedora plataforma de trabajo ofrecida por los nuevos navegadores, Javascript fue expandido en relación con portabilidad e integración. Al igual, interfaces de programación de aplicaciones (APIs) fueron incorporadas por defecto en cada navegador para asistir al lenguaje en funciones elementales. Estas nuevas APIs (como Web Storage, Canvas, y otras) son interfaces para librerías incluidas en navegadores. La idea es hacer disponible poderosas funciones a través de técnicas de programación sencillas y estándares, expandiendo el alcance del lenguaje y facilitando la creación de programas útiles para la web.

Después de hablar de las tecnologías del lado del cliente, a continuación hablaremos de las que se ejecutan en el servidor.

3.5. Lenguajes en servidor (Backend)

3.5.1. C#

C# es el nuevo lenguaje de propósito general diseñado por Scott Wiltamuth y Anders Hejlsberg, este último también es conocido por diseñar el lenguaje Turbo Pascal y la herramienta RAD Delphi.

Aunque es posible escribir código para la plataforma .NET en distintos lenguajes, C# es el único diseñado específicamente para ser utilizado en ella, por lo que programarla usando C# es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes ya que C# carece de elementos heredados innecesarios en .NET. Por esta razón, se suele decir que C# es el lenguaje nativo de .NET.

En resumen, C# es un lenguaje de programación que toma las mejores características de lenguajes pre-existentes como Visual Basic, Java o C++ y los combina en uno solo. El hecho de ser relativamente reciente no implica que sea inmaduro, pues Microsoft ha escrito la mayor parte de la BCL usándolo, por lo que su compilador es el más depurado y optimizado de los incluidos en el .NET Framework SDK.

Las características principales de C# son las siguientes:

- **Sencillez:** C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET. Por ejemplo:
 - El código escrito en C# es autocontenido, lo que significa que no necesita de ficheros adicionales al propio fuente tales como ficheros de cabecera o ficheros IDL.
 - El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile (no como en C++), lo que facilita la portabilidad del código.
 - No se incluyen elementos poco útiles de lenguajes como C++ tales como macros, herencia múltiple o la necesidad de un operador diferente del punto (.) acceder a miembros de espacios de nombres (::)
- **Modernidad:** C# incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como Java o C++ hay que simular, como un tipo básico decimal que permita realizar operaciones de alta precisión con reales de 128 bits, la inclusión de una instrucción foreach que permita recorrer colecciones con facilidad y es ampliable a tipos definidos por el usuario, la inclusión de un tipo básico string para representar cadenas o la distinción de un tipo bool específico para representar valores lógicos.
- **Orientación a objetos:** Como todo lenguaje de programación de propósito general actual, C# es un lenguaje orientado a objetos, aunque eso es más bien una característica del CTS que de C#. Una diferencia de este enfoque orientado a objetos respecto al de otros lenguajes como C++ es que el de C# es más puro en tanto que no admiten ni funciones ni variables globales sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos

de nombres y facilita la legibilidad del código.

C# soporta todas las características propias del paradigma de programación orientada a objetos: encapsulación, herencia y polimorfismo.

En lo referente a la encapsulación es importante señalar que aparte de los típicos modificadores public, private y protected, C# añade un cuarto modificador llamado internal, que puede combinarse con protected e indica que al elemento a cuya definición precede sólo puede accederse desde su mismo ensamblado.

Respecto a la herencia -a diferencia de C++ y al igual que Java- C# sólo admite herencia simple de clases ya que la múltiple provoca más quebraderos de cabeza que facilidades y en la mayoría de los casos su utilidad puede ser simulada con facilidad mediante herencia múltiple de interfaces. De todos modos, esto vuelve a ser más bien una característica propia del CTS que de C#.

Por otro lado y a diferencia de Java, en C# se ha optado por hacer que todos los métodos sean por defecto sellados y que los redefinibles hayan de marcarse con el modificador virtual (como en C++), lo que permite evitar errores derivados de redefiniciones accidentales. Además, un efecto secundario de esto es que las llamadas a los métodos serán más eficientes por defecto al no tenerse que buscar en la tabla de funciones virtuales la implementación de los mismos a la que se ha de llamar. Otro efecto secundario es que permite que las llamadas a los métodos El lenguaje de programación C# Tema 2: Introducción a C# José Antonio González Seco Página 23 virtuales se puedan hacer más eficientemente al contribuir a que el tamaño de dicha tabla se reduzca.

- **Orientación a componentes:** La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas. Es decir, la sintaxis de C# permite definir cómodamente propiedades (similares a campos de acceso controlado), eventos (asociación controlada de funciones de respuesta a notificaciones) o atributos (información sobre un tipo o sus miembros).
- **Gestión automática de memoria:** Como ya se comentó, todo lenguaje de .NET tiene a su disposición el recolector de basura del CLR. Esto tiene el efecto en el lenguaje de que no es necesario incluir instrucciones de destrucción de objetos. Sin embargo, dado que la destrucción de los objetos a través del recolector de basura es indeterminista y sólo se realiza cuando éste se active ya sea por falta de memoria, finalización de la aplicación o solicitud explícita en el fuente-, C# también proporciona un mecanismo de liberación de recursos determinista a través de la instrucción using.
- **Seguridad de tipos:** C# incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente, lo que permite evita que se produzcan errores difíciles de detectar por acceso a memoria no perteneciente a ningún objeto y es especialmente necesario en un entorno gestionado por un recolector de basura. Para ello se toman medidas del tipo:
 - Sólo se admiten conversiones entre tipos compatibles. Esto es, entre un tipo y antecesores suyos, entre tipos para los que explícitamente se haya definido un operador de conversión, y entre un tipo y un tipo hijo suyo del que un objeto del primero almacenase una referencia del segundo (downcasting) Obviamente, lo último sólo puede comprobarlo en tiempo de ejecución el CLR y no el compilador, por lo que en realidad el CLR y el compilador colaboran para asegurar la corrección de las conversiones.
 - No se pueden usar variables no inicializadas. El compilador da a los campos un valor por defecto consistente en ponerlos a cero y controla mediante análisis del flujo de control del fuente que no se lea ninguna variable local sin que se le haya asignado previamente algún valor.

-
- Se comprueba que todo acceso a los elementos de una tabla se realice con índices que se encuentren dentro del rango de la misma.
 - Se puede controlar la producción de desbordamientos en operaciones aritméticas, informándose de ello con una excepción cuando ocurra. Sin embargo, para conseguirse un mayor rendimiento en la aritmética estas comprobaciones no se hacen por defecto al operar con variables sino sólo con constantes (se pueden detectar en tiempo de compilación).
 - A diferencia de Java, C# incluye delegados, que son similares a los punteros a funciones de C++ pero siguen un enfoque orientado a objetos, pueden almacenar referencias a varios métodos simultáneamente, y se comprueba que los métodos a los que apunten tengan parámetros y valor de retorno del tipo indicado al definirlos.
 - Pueden definirse métodos que admitan un número indefinido de parámetros de un cierto tipo, y a diferencia lenguajes como C/C++, en C# siempre se comprueba que los valores que se les pasen en cada llamada sean de los tipos apropiados.
 - **InSTRUCCIONES SEGURAS:** Para evitar errores muy comunes, en C# se han impuesto una serie de restricciones en el uso de las instrucciones de control más comunes. Por ejemplo, la guarda de toda condición ha de ser una expresión condicional y no aritmética, con lo que se evitan errores por confusión del operador de igualdad (==) con el de asignación (=); y todo caso de un switch ha de terminar en un break o goto que indique cuál es la siguiente acción a realizar, lo que evita la ejecución accidental de casos y facilita su reordenación.
 - **Sistema de tipos unificado:** A diferencia de C++, en C# todos los tipos de datos que se definen siempre derivarán, aunque sea de manera implícita, de una clase base común llamada System.Object, por lo que dispondrán de todos los miembros definidos en ésta clase (es decir, serán "objetos").

A diferencia de Java, en C# esto también es aplicable a los tipos de datos básicos. Además, para conseguir que ello no tenga una repercusión negativa en su nivel de rendimiento, se ha incluido un mecanismo transparente de boxing y unboxing con el que se consigue que sólo sean tratados como objetos cuando la situación lo requiera, y mientras tanto puede aplicárseles optimizaciones específicas.

El hecho de que todos los tipos del lenguaje deriven de una clase común facilita enormemente el diseño de colecciones genéricas que puedan almacenar objetos de cualquier tipo.

- **Extensibilidad de tipos básicos:** C# permite definir, a través de estructuras, tipos de datos para los que se apliquen las mismas optimizaciones que para los tipos de datos básicos. Es decir, que se puedan almacenar directamente en pila (luego su creación, destrucción y acceso serán más rápidos) y se asignen por valor y no por referencia. Para conseguir que lo último no tenga efectos negativos al pasar estructuras como parámetros de métodos, se da la posibilidad de pasar referencias a pila a través del modificador de parámetro ref.
- **Extensibilidad de operadores:** Para facilitar la legibilidad del código y conseguir que los nuevos tipos de datos básicos que se definen a través de las estructuras estén al mismo nivel que los básicos predefinidos en el lenguaje, al igual que C++ y a diferencia de Java, C# permite redefinir el significado de la mayoría de los operadores -incluidos los de conversión, tanto para conversiones implícitas como explícitas- cuando se apliquen a diferentes tipos de objetos.

Las redefiniciones de operadores se hacen de manera inteligente, de modo que a partir de una única definición de los operadores ++ y - el compilador puede deducir automáticamente como ejecutarlos de manera prefijas y postfija; y definiendo operadores simples (como +), el compilador deduce cómo aplicar su versión de asignación compuesta (+=). Además, para asegurar la consistencia, el compilador vigila que los operadores con opuesto siempre se redefinan por parejas (por ejemplo, si se redefine ==,

también hay que redefinir !=).

También se da la posibilidad, a través del concepto de indizador, de redefinir el significado del operador [] para los tipos de dato definidos por el usuario, con lo que se consigue que se pueda acceder al mismo como si fuese una tabla. Esto es muy útil para trabajar con tipos que actúen como colecciones de objetos.

- **Extensibilidad de modificadores:** C# ofrece, a través del concepto de atributos, la posibilidad de añadir a los metadatos del módulo resultante de la compilación de cualquier fuente información adicional a la generada por el compilador que luego podrá ser consultada en tiempo ejecución a través de la librería de reflexión de .NET . Esto, que más bien es una característica propia de la plataforma .NET y no de C#, puede usarse como un mecanismo para definir nuevos modificadores.
- **Versionable:** C# incluye una política de versionado que permite crear nuevas versiones de tipos sin temor a que la introducción de nuevos miembros provoquen errores difíciles de detectar en tipos hijos previamente desarrollados y ya extendidos con miembros de igual nombre a los recién introducidos.

Si una clase introduce un nuevo método cuyas redefiniciones deban seguir la regla de llamar a la versión de su padre en algún punto de su código, difícilmente seguirían esta regla miembros de su misma signatura definidos en clases hijas previamente a la definición del mismo en la clase padre; o si introduce un nuevo campo con el mismo nombre que algún método de una clase hija, la clase hija dejará de funcionar. Para evitar que esto ocurra, en C# se toman dos medidas:

- Se obliga a que toda redefinición deba incluir el modificador override, con lo que la versión de la clase hija nunca sería considerada como una redefinición de la versión de miembro en la clase padre ya que no incluiría override. Para evitar que por accidente un programador incluya este modificador, sólo se permite incluirlo en miembros que tengan la misma signatura que miembros marcados como redefinibles mediante el modificador virtual. Así además se evita el error tan frecuente en Java de creerse haber redefinido un miembro, pues si el miembro con override no existe en la clase padre se producirá un error de compilación.
- Si no se considera redefinición, entonces se considera que lo que se desea es ocultar el método de la clase padre, de modo que para la clase hija sea como si nunca hubiese existido. El compilador avisará de esta decisión a través de un mensaje de aviso que puede suprimirse incluyendo el modificador new en la definición del miembro en la clase hija para así indicarle explícitamente la intención de ocultación.
- **Eficiente:** En principio, en C# todo el código incluye numerosas restricciones para asegurar su seguridad y no permite el uso de punteros. Sin embargo, y a diferencia de Java, en C# es posible saltarse dichas restricciones manipulando El lenguaje de programación C# Tema 2: Introducción a C# José Antonio González Seco Página 26 objetos a través de punteros. Para ello basta marcar regiones de código como inseguras (modificador unsafe) y podrán usarse en ellas punteros de forma similar a cómo se hace en C++, lo que puede resultar vital para situaciones donde se necesite una eficiencia y velocidad procesamiento muy grandes.
- **Compatible:** Para facilitar la migración de programadores, C# no sólo mantiene una sintaxis muy similar a C, C++ o Java que permite incluir directamente en código escrito en C# fragmentos de código escrito en estos lenguajes, sino que el CLR también ofrece, a través de los llamados Platform Invocation Services (PInvoke), la posibilidad de acceder a código nativo escrito como funciones sueltas no orientadas a objetos tales como las DLLs de la API Win32. Nótese que la capacidad de usar punteros en código inseguro permite que se pueda acceder con facilidad a este tipo de funciones, ya que éstas muchas veces esperan recibir o devuelven punteros.

También es posible acceder desde código escrito en C# a objetos COM. Para facilitar esto, el .NET Framework SDK incluye una herramientas llamadas tlbimp y regasm mediante las que es posible generar automáticamente clases proxy que permitan, respectivamente, usar objetos COM desde .NET como si de objetos .NET se tratase y registrar objetos .NET para su uso desde COM.

Finalmente, también se da la posibilidad de usar controles ActiveX desde código .NET y viceversa. Para lo primero se utiliza la utilidad aximp, mientras que para lo segundo se usa la ya mencionada regasm.

3.6. Frameworks

El concepto framework se emplea en muchos ámbitos del desarrollo de sistemas software, no solo en el ámbito de aplicaciones Web. Podemos encontrar frameworks para el desarrollo de aplicaciones médicas, de visión por computador, para el desarrollo de juegos, y para cualquier ámbito que pueda ocurrírseños.

En general, con el término framework, nos estamos refiriendo a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.

Existen frameworks tanto para lenguajes del cliente como del servidor, permitiendo realizar desarrollos mas rápidos y eficientes. Dentro de los frameworks utilizados se encuentran los siguientes:

3.6.1. Frameworks en cliente

Bootstrap

Bootstrap es un framework de Twitter el cual está enfocado para el desarrollo Web, específicamente diseñado para la maquetación de sitios Web mediante el lenguaje de estilos CSS y Javascript. Este está conformado por un conjunto de archivos CSS y Javascript los cuales al incluirlos en un sitio Web se facilita el maquetado y la edición de estilos de la misma sin tocar en absoluto el código CSS, esto agiliza el desarrollo de aplicaciones Web dándonos un diseño elegante y bueno gracias a sus clases predefinidas.

El Framework trae varios elementos con estilos predefinidos fáciles de configurar: Botones, Menús desplegables, Formularios incluyendo todos sus elementos e integración jQuery para ofrecer ventanas y mensajes dinámicos.

JQuery

JQuery es una biblioteca de JavaScript creada inicialmente por John Resig, la cual nos permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar integración con la técnica AJAX a páginas Web.

El DOM (Document Object Model), proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos. A través del DOM, los programas pueden acceder y modificar el contenido, estructura y estilo de los documentos HTML y XML, que es para lo que se diseñó principalmente.

3.6.2. Frameworks en servidor

.NET Framework

El framework utilizado en el servidor es .NET Framework de Microsoft el cual es un componente de software que puede ser añadido al sistema operativo Windows, este provee un extenso conjunto de soluciones predefinidas para necesidades generales de la programación de aplicaciones y administra la ejecución de los programas escritos específicamente con la plataforma.

.NET Framework podría ser considerada una respuesta de Microsoft al creciente mercado de los negocios en entornos WEB, como competencia a la plataforma Java de Oracle Corporation y a los diversos Frameworks de desarrollo web usados en PHP. Dicho Framework propone ofrecer una manera rápida y económica, a la vez que segura y robusta para desarrollar aplicaciones permitiendo una integración rápida y ágil, permitiendo con esto acceso a la información desde cualquier tipo de dispositivo.

A continuación se muestra una breve descripción de las versiones de .NET Framework:

- **.NET Framework 1.0:** Lanzado en 2002 (Visual Studio .NET), Esta es la primera versión de .NET Framework, publicado el 13 de febrero de 2002 y disponible para Windows 98, ME, NT 4.0, 2000 y XP. El soporte estándar de Microsoft para esta versión finaliza 10 de julio de 2007 y el soporte extendido terminó el 14 de julio de 2009, con la excepción de XP Media Center y Tablet PC ediciones.
- **.NET Framework 1.1:** Lanzado en 2003 (Visual Studio 2003), El soporte integrado para teléfonos ASP.NET controles. Previamente disponible como un add-on para .NET Framework, que ahora forma parte del marco. Los cambios en la seguridad permiten utilizar ensamblados para ejecutar aplicaciones en internet con restricciones de confianza permitiendo accesos restringidos a las aplicaciones de ASP.NET. El soporte para ODBC y bases de datos le permite una mayor funcionalidad ademas de implementar el soporte para Internet versión 6.
- **.NET Framework 2.0:** Lanzado en 2005 (Visual Studio .NET 2005), con un nuevo CLR (para manejar los genéricos y tipos anulables) y los compiladores de C# y VB 2.8. El paquete redistribuible 2.0 se puede descargar de forma gratuita desde Microsoft, y fue publicado el 22 de enero de 2006. 2.0 El SDK(Software Development Kit) se puede descargar de forma gratuita desde Microsoft. Se incluye como parte de Visual Studio 2005 y Microsoft SQL Server 2005. La versión 2.0 sin ningún Service Pack es la última versión con soporte para Windows 98 y Windows Me. Versión 2.0 con Service Pack 2 es la última versión con soporte oficial para Windows 2000, aunque ha habido algunas soluciones no oficiales publicados en línea para utilizar un subconjunto de la funcionalidad de la Versión 3.5 en Windows 2000.
- **.NET Framework 3.0:** Lanzado en 2006 , básicamente es sólo .NET 2.0 con nuevas bibliotecas como Windows Presentation Foundation, Windows Communication Foundation, Workflow Foundation y Cardspace, .NET Framework 3.0 anteriormente llamado WinFX, fue lanzado el 21 de noviembre de 2006 e incluye un nuevo sistema de código administrado API que son una parte integral de Windows Vista y Windows Server 2008 sistemas operativos. También está disponible para Windows XP SP2 y Windows Server 2003 como descarga. No hay grandes cambios de arquitectura que se incluyen con esta versión; .NET Framework 3.0 utiliza el mismoCLR (Common Language Runtime) como .NET Framework 2.0 .
- **.NET Framework 3.5:** Lanzado en 2007, basado en .NET 3.0 incluyendo nuevas bibliotecas como LINQ y TimeZoneInfo, además de nuevos compiladores de C# y VB 3.9.
- **.NET Framework 4.0:** Lanzado en 2010 con un nuevo CLR, nuevas bibliotecas y el DLR(Dynamic Language Runtime). La parte principal de esta versión son las extensiones que permiten mejorar el apoyo para el computo paralelo, dirigida a multiples núcleos o sistemas distribuidos. Se implementa la tecnología de computo paralelo para LINQ llamada PLINQ (Parallel LINQ).

-
- **.NET Framework 4.5:** Lanzado en 2012 permite el desarrollo en Windows 8, así como bibliotecas adicionales que permiten la resolución más eficiente de expresiones regulares. También permite definir una cultura para un dominio de aplicación y soporte para la Unicode(UTF-16).

Estas tecnologías en conjunto son de vital importancia debido a que las funcionalidades que se agregan al sistema dependen del trabajo en conjunto de las mismas.

Para mejorar el funcionamiento del SII fue necesario aplicar un modelo de arquitectura de software ya que en la actualidad de no hacerlo de esta manera, puede que el sistema no tenga el rendimiento o funcionamiento esperado.

A continuación se describen los conceptos de arquitectura de software utilizados en el SII.

3.7. Arquitectura de Software

En un principio la programación fue considerada un arte y se desarrollaba como tal, esto debido a la dificultad que representaba realizar dicha actividad. Con el paso del tiempo se han ido desarrollando nuevas formas para el desarrollo de software con las cuales la resolución de problemas ha reducido su complejidad. A estas formas de desarrollo se les denomina Arquitectura de Software.

La arquitectura de software es definida como un nivel de diseño que pone especial interés en aspectos más allá de los algoritmos y estructuras de datos de la computación; el diseño y especificación de la estructura global del sistema es un nuevo tipo de problema.

La arquitectura de software es el diseño de mas alto nivel de la estructura de un sistema permitiendo con esto contar con un panorama amplio acerca del funcionamiento del sistema.

Para elegir una arquitectura de software se necesita realizar un análisis de objetivos y requerimientos, además de las restricciones, por ejemplo, no es recomendable utilizar una arquitectura de 3 capas para un sistema que manejará información en tiempo real.

Generalmente, no es necesario inventar una nueva arquitectura para cada sistema a desarrollar, mas bien es recomendable adoptar alguna de las existentes.

Dentro de las arquitecturas mas conocidas se encuentran las siguientes:

- **Descomposición modular:** Donde el software se estructura en grupos funcionales muy acoplados.
- **Cliente-Servidor:** Donde el software reparte su carga de cómputo en dos partes independientes pero sin reparto claro de funciones.
- **Arquitectura de tres niveles:** Especialización de la arquitectura cliente-servidor donde la carga se divide en tres partes (o capas) con un reparto claro de funciones: una capa para la presentación (interfaz de usuario), otra para el cálculo (donde se encuentra modelado el negocio) y otra para el almacenamiento (persistencia). Una capa solamente tiene relación con la siguiente.
- **Modelo Vista Controlador:** Es un patrón de arquitectura de software, que separa los datos y la lógica de negocios de un aplicación de la interfaz de usuario y el modulo encargado de gestionar los eventos y las comunicaciones.

La arquitectura seleccionada para el SII es el Modelo Vista Controlador (MVC) el cual debido a la forma de organizar el funcionamiento del sistema cubre muy bien las necesidades del sistema.

3.7.1. MVC (Modelo Vista Controlador)

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.

Los tres componentes en los que se separa el Modelo Vista Controlador son:

- **Modelo:** Contiene una representación de los datos que maneja el sistema, su lógica de negocio y sus mecanismos de persistencia.
- **Vista:** Esta es en otras palabras la interfaz, la cual contiene la información que se envía al cliente y los mecanismos de interacción con el mismo.
- **Controlador:** Este actúa como intermediario entre el modelo y la vista, gestionando el flujo de información entre ellos y las transformaciones para adoptar los datos a las necesidades.

El modelo se encarga de las siguientes acciones:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: "Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor".
- Lleva un registro de las vistas y controladores del sistema.
- Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero por lotes que actualiza los datos, un temporizador que desencadena una inserción, etc.).

El controlador se encarga de las siguientes acciones:

- Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc).
- Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método `.Actualizar()`. Una petición al modelo puede ser `ObtenerTiempoDeEntrega(nuevaOrdenDeVenta)`.

La vista se encarga de las siguientes acciones:

- Recibir datos del modelo y la muestra al usuario.
- Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- Pueden dar el servicio de `Actualizar()`, para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

El flujo que sigue una petición dentro del MVC se describe en la figura 3.2

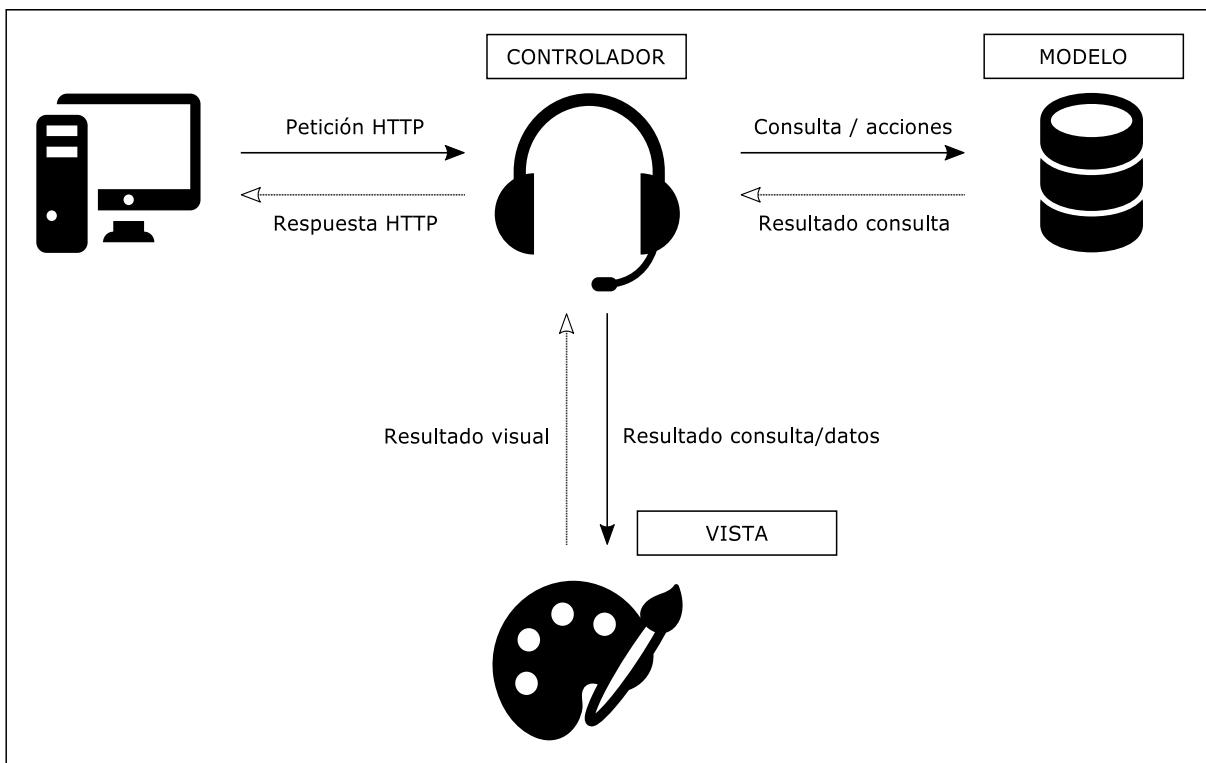


Figura 3.2: Diagrama de flujo de una petición en MVC.

Los pasos que sigue la petición son los siguientes:

- El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc).
- El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
- El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
- El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, se podría utilizar el patrón Observador para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice. Nota: En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.
- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

3.8. Base de Datos

En los últimos tiempos, se ha empezado a considerar la información como un recurso estratégico de una organización; pues facilita su supervivencia y le permite, además, ser competitiva en el mercado. Dicha información le permite tener una visión de los posibles cambios, y con esto mejorar la toma de decisiones.

Una base de datos se define como una colección de datos almacenados de una manera permanente, que pueden ser compartidos y usados con distintos propósitos por múltiples usuarios.

La información se debe considerar como otro recurso corporativo de la misma manera que se considera el recurso humano o los recursos físicos. Por lo tanto, la administración de la información debe implicar:

- Planear su adquisición por anticipado.
- Protegerla contra la destrucción o mal uso.
- Asegurar su calidad.
- Retirarla de la organización cuando ya no se le requiera.

Debido que la cantidad de información que una organización necesita para subsistir es cada vez mayor, deben existir métodos eficientes tanto para el almacenamiento rápido como para la consulta ágil. La tecnología más utilizada para manejar estas grandes cantidades de información son las Bases de Datos.

La tecnología de bases de datos proporciona los medios para que se cumplan los objetivos de lograr máximos beneficios en la administración de información debido a las siguientes razones:

- Se logra que el desarrollo de aplicaciones sea más rápido debido a que los programadores reutilizan datos u procedimientos almacenados en la base de datos.
- Hay una mayor participación del usuario final en la creación de las aplicaciones, haciendo software más tangible y de mayor valor inmediato.
- El acceso a los datos es flexible y rápido.

Gracias a esto las organizaciones tienen más y mejor información para la toma de decisiones, además de incrementar su productividad.

A continuación se presentan las ventajas y desventajas de un sistema de base de datos:

a) Ventajas

- 1) **Economía de escala:** esencialmente, la concentración de aplicaciones en una sola localidad puede reducir costos.
- 2) **Mayor información:** existe una mayor facilidad para el análisis y la toma de decisiones.
- 3) **Datos y programas compartidos:** la reutilización de los mismos datos y programas, permiten minimizar o controlar la redundancia.
- 4) **Declaración de estándares:** Debido a que la información se vuelve muy grande, es necesario declarar estándares para el acceso y manejo de los datos.
- 5) **Consistencia de datos:** Esta dada por el control o eliminación de la redundancia.
- 6) **Integridad:** El sistema de base de datos debe velar por el grado de validez y de corrección de los datos. Debe permitir definir reglas que deben cumplir los datos en la base de datos.
- 7) **Seguridad:** Se pueden especificar niveles de acceso mediante perfiles de usuario.

b) Desventajas

- **Tamaño:** Un sistema de base de datos es un gran conjunto de programas que a lo largo de su uso su tamaño incrementa considerablemente.
- **Mayor susceptibilidad a fallas:** Debido a la cantidad de información es más probable que se generen fallas.
- **Recuperación a las fallas:** La recuperación de un DBMS interactivo y multiusuario puede ser muy compleja.

Los sistemas de base de datos utilizan un lenguaje el cual le permite realizar modificaciones sobre ella. Este lenguaje es el llamado SQL.

3.8.1. Lenguaje SQL

El lenguaje SQL es un conjunto de construcciones que permite crear, consultar y modificar una base de datos. Para realizar estas acciones, SQL se divide en tres grupos de lenguajes de base de datos:

- **DML (Data Manipulation Language):** Permite consultar y modificar datos.
- **DDL (Data Definition Language):** Permite definir la base de datos.
- **DAL (Data Access Language):** Permite administrar el acceso a la base de datos.

Todos los sistemas gestores de base de datos cuentan con estos tres grupos para poder realizar las acciones sobre las bases de datos. Dentro de los sistemas gestores de base de datos más comunes se encuentran los siguientes:

- SQL Server.
- MySQL.
- PostgreSQL.
- Oracle.

El lenguaje elegido para realizar el SII es SQL Server, esto debido a la integración que tiene con el ambiente de desarrollo y los lenguajes seleccionados.

3.8.2. Microsoft SQL Server

Microsoft SQL Server es un sistema de gestión de bases de datos relacional producido por Microsoft. Su principal lenguaje de consulta es Transact-SQL, una aplicación de las normas ANSI/ISO Standard Structured Query Language (SQL) utilizando por ambas Microsoft y Sybase.

Las características de Microsoft SQL Server son las siguientes:

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.

-
- Además permite administrar información de otros servidores de datos.

Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños, que en su versión 2005 pasa a ser el SQL Express Edition, que se distribuye en forma gratuita.

Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle, Sybase ASE, PostgreSQL o MySQL.

Requisitos de hardware

Tabla 3.2: Requisitos de Hardware de Microsoft SQL Server

COMPONENTE	REQUISITO
	Minimo:
	Ediciones Express: 512 MB
	Todas las demás ediciones: 1 GB
Memoria	Recomendaciones:
	Ediciones Express: 1 GB
	Todas las demás ediciones: Al menos 4 GB y debe aumentar a medida que el tamaño de la base de datos aumente para asegurar un rendimiento óptimo.
	Minimo
Velocidad del procesador	- Procesador x86: 1 GHz. - Procesador x64: 1.4 Ghz.
	Recomendado: 2 GHz o más.
Tipo de procesador	- Procesador x64: AMD Opteron, AMD Athlon 64, Intel Xeon compatible con Intel EM64T Intel Pentium IV. - Procesador x86: compatible con Pentium III o superior.

Rendimiento

Respecto al rendimiento de Microsoft SQL Server, este se desempeña dentro de cualquier arquitectura .NET o J2EE.

Del mismo modo este se beneficia de RAID (Redundant Array of Independent Disks) el cual es un sistema de almacenamiento de datos entre los que se distribuye o replican los datos, es decir, este se desempeña mejor si se ejecuta en un servidor dedicado específicamente a él.

Microsoft SQL Server como cualquier otro sistema manejador de bases de datos depende de la estructura de las instrucciones que recibe para determinar su rendimiento, esto debido a que existen distintas maneras de consultar o modificar la información, es por ello que es importante seguir la documentación del producto para realizar las instrucciones más rápidas posibles.

Administración y mantenimiento

Microsoft SQL Server ofrece un grupo de acciones que permiten administrar los recursos dentro del mismo, las cuales se enlistan a continuación:

- Se provee de un sistema de copias de seguridad en la cual se vuelcan los datos y estructura en un archivo con extensión ".bak".
- Soporta tres tipos de replicació:
 - **Instantánea:** En la replicación de instantáneas los datos se copian tal y como aparecen exactamente en un momento determinado.
 - **Transaccional:** En este caso se propaga una instantánea inicial de datos a los suscriptores, y después, cuando se efectúan las modificaciones en el publicador, las transacciones individuales se propagan a los suscriptores. Microsoft SQL Server almacena las transacciones que afectan a los objetos replicados y propaga esos cambios a los suscriptores de forma continua o a intervalos programados. Al finalizar la propagación de los cambios, todos los suscriptores tendrán los mismos valores que el publicador.
 - **Mixta:** Permite que varios sitios funcionen en línea o desconectados de manera autónoma, y mezclar más adelante las modificaciones de datos realizadas en un resultado único y uniforme. La instantánea inicial se aplica a los suscriptores; a continuación SQL Server 2000 hace un seguimiento de los cambios realizados en los datos publicados en el publicador y en los suscriptores.

Estabilidad

Microsoft SQL Server tiene una gran resistencia a la corrupción de datos ya que los datos van a través de distintos puestos de control y previene contratiempos en caso de que el sistema colapse sin previo aviso. Se puede considerar como un contratiempo a un fallo inesperado del equipo huésped de Microsoft SQL Server o un apagado inesperado del mismo.

Desarrollo de aplicaciones

- Se cuenta con múltiples APIs para acceder a él.
- Se apoya en ODBC y JDBC para conectividad en red, así como los métodos de acceso de base de datos nativos.
- Soporta conexiones desde distintos lenguajes como C/C++, Java, Perl, Python, PHP, C# y Visual Basic.
- Soporta métodos de cifrado SSL.

Licencias

El licenciamiento de Microsoft SQL Server esta disponible de las siguientes dos maneras:

- **Por Procesador:** Requiere una licencia única para cada CPU en el equipo que ejecuta e incluye el acceso limitado de clientes.
- **Servidor/por asiento:** Se requiere una licencia para el servidor y las licencias para cada cliente.

Ya terminados los putos de la parte tÃ©cnica del proyecto, hablaremos de la parte de la metodologÃa utilizada para realizar el mismo.

3.9. Metodología de desarrollo

Se le llama metodología a el conjunto de técnicas y herramientas que dan una documentación formal al proyecto referente a los procesos, las políticas y procedimientos que ayudan a los desarrolladores para la realizaron de un producto (software) .

El objetivo principal de una metodología es tener mayor eficacia al cumplir con los requisitos que se plantearon desde un principio en el proyecto, además de prevenir las pérdidas de tiempo en la realización del mismo, siendo un guía que llevará de manera ordenada el equipo de trabajo en el desarrollo de software con respecto a las actividades y a que el software tiene un ciclo de vida, es decir, tiene un tiempo límite para su cumplimiento.

De forma general el ciclo de vida del software está definido de los siguientes puntos:

- **Definición de objetivos:** se definen todas las ideas concretas del proyecto permitiendo tener una visión general del mismo, además de establecer un panorama amplio para la definición de alcances.
- **Análisis de requisitos y su viabilidad:** Se recopilan los requisitos del cliente y examina cualquier restricción que se pueda aplicar, es aquí donde se comienzan a visualizar los alcances del proyecto.
- **Diseño general:** Son los requisitos generales de la arquitectura del proyecto.
- **Diseño en detalle:** Es la definición precisa de cada bloque de la aplicación, esto hace referencia al detalle del funcionamiento y acciones a realizar por el sistema.
- **Programación:** Es la implementación de un lenguaje de programación para crear los modulos o funciones definidas en el proyecto.
- **Prueba de unidad:** Es la prueba individual de cada bloque para verificar que cumplen con las especificaciones de diseño y funcionalidad del proyecto.
- **Prueba beta:** Esta prueba se encarga de garantizar que el software cumple con las especificaciones originales y cumpla con todos los objetivos establecidos.
- **Documentación:** Es la guía que servirá a los usuarios que utilizaran el software a aclarar sus dudas respecto al funcionamiento del mismo en caso de usuarios finales, además de la creación de un manual técnico que servirá como guía para los encargados del soporte.
- **Implementación:** Es la puesta en marcha del proyecto para su uso, para llegar a esta parte es necesario que los puntos anteriores se evaluaran de manera satisfactoria y con esto validar que el software esta listo para ponerse en un ambiente de producción.
- **Mantenimiento:** Es la corrección de los posibles errores resultantes del uso cotidiano de los usuarios finales, estos se pueden presentar de dos maneras, errores de datos y errores de sistema.
 - Los errores de datos hacen referencia a una mala captura de información por parte del sistema o el usuario final, truncando con esto alguno de los procesos posteriores debido a información incorrecta o falta de la misma.
 - Los errores de sistema son aquellos en los que la falla es ocasionada por un fallo directamente en la programación del sistema.

Los modelos para el desarrollo de software están clasificados en tres grupos, de los cuales cada uno de ellos representa el proceso de desarrollo de software de una manera particular, y a pesar de estar definidos claramente, no representan necesariamente la realidad de cómo se debe desarrollar el software, si no que establece un enfoque común. Un modelo puede ser modificado y adaptado de acuerdo a las necesidades de

software en desarrollo.

Los tres grupos de modelos de desarrollo son los siguientes:

- Modelo en cascada.
- Modelo evolutivo.
- Modelo basado en componentes.

El modelo que se utilizará para el desarrollo de este proyecto es el modelo en cascada, del cual hablaremos a continuación.

3.9.1. **Modelo en cascada**

Modelo conocido también como modelo clásico, modelo tradicional o modelo lineal secuencial.

Es el más básico de todos los modelos y sirve como bloque de construcción para los demás paradigmas de ciclo de vida del desarrollo de sistemas, se puede decir que es un método puro que implica un desarrollo rígido y lineal. Está basado en el ciclo convencional de una ingeniería y su visión es muy simple: el desarrollo de software se debe realizar siguiendo una secuencia de fases. Cada etapa tiene un conjunto de metas bien definidas y las actividades dentro de cada una contribuyen a la satisfacción de metas de esa.

Las fases normalmente aplicadas son:

- **Análisis del Sistema:** En esta fase se establecen los requisitos de todos los elementos del sistema y luego se asigna algún subconjunto de estos requisitos al software.
- **Análisis de requisitos del software:** El proceso de recopilación de los requisitos se centra e intensifica especialmente en el software. Es necesario comprender el ámbito de la información del software así como la función, el rendimiento y las interfaces requeridas.
- **Diseño del sistema:** Se enfoca en cuatro atributos distintos del programa:
 - La estructura de los datos.
 - La arquitectura del software.
 - El detalle procedimental.
 - La caracterización de la interfaz.
- **Codificación:** El diseño debe traducirse en una forma legible para la máquina. Si el diseño se realiza de una manera detallada, la codificación puede realizarse mecánicamente.
- **Pruebas:** Generado el código se inicia la prueba del programa. Que se centra en la lógica interna del software y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren.
- **Implantación:** Significa programación. Producto de esta etapa es el código en cualquier nivel, incluido el producido por sistemas de generación automática.
- **Mantenimiento:** el software puede sufrir algunos cambios después de que se entrega al cliente. Los cambios ocurrirán debidos a que se haya encontrado errores, a que el software deba adaptarse a cambios del entorno externo (sistema operativo o dispositivos periféricos) o a que el cliente requiera ampliaciones funcionales o del rendimiento.

Sin embargo debemos saber que los proyectos reales raramente siguen el flujo secuencial que propone el modelo. Siempre hay iteraciones y se crean problemas en la aplicación del paradigma.

Es difícil para el cliente establecer todos los requisitos explícitamente. El ciclo de vida clásico lo requiere y tiene dificultades en acomodar posibles incertidumbres que pueden existir al comienzo de muchos productos.

Por lo tanto se inicia con la especificación de requerimientos del cliente, continua con la planificación, el modelado, la construcción y el despliegue para finalizar en el enfoque del software. El modelo está dirigido por documentos y no proporciona resultados tangibles de software hasta el final del ciclo de vida de algunas herramientas. El diseño en cascada es una secuencia definida de los acontecimientos y los resultados finales para proporcionar una estructura para cualquier proyecto que siga el contenido específico y detallado. Puede ser apropiado para proyectos de software que son estables especialmente cuando sus requisitos no cambian. Es caracterizado por ordenar de manera rigurosa las etapas del ciclo de vida de software, dado que el comienzo de cada etapa debe esperar a la finalización de la inmediata anterior.

La metodología en cascada es esencialmente:

- El inicio y el alcance del proyecto.
- La planificación del proyecto (calendario, recursos necesarios, costo).
- Definición de las necesidades del negocio y el análisis en detalle de la solución.
- La creación de la solución.
- Prueba que la solución funciona.
- La entrega de la solución a su público objetivo.
- Cierre del proyecto.

Ventajas

- Permite el control de gestión.
- Plazos normalmente adecuados para cada etapa de desarrollo.
- Este proceso conduce a entregar el proyecto a tiempo.
- Es sencilla y facilita la gestión de proyectos.
- Permite tener bajo control el proyecto.
- Limita la cantidad de interacción entre equipos que se produce durante el desarrollo.

Desventajas

- No conocer si la solución es correcta hasta estar cerca de su lanzamiento.
- Poco tiempo para corregir fallas.
- Depuración complicada.
- Los cambios introducidos durante el desarrollo pueden confundir al equipo.
- No es frecuente que el cliente o usuario final explice clara y completamente los requisitos.
- Es necesaria la paciencia del cliente.
- El cliente podría detectar un error.
- El proceso es lento y pesado.

Probable fracaso del modelo de cascada

- Los clientes quieren ver el producto a medida que se desarrolla.
- El cliente no puede validar el producto a medida que lo desarrollamos.
- No poder tomar decisiones por la falta de avances o resultados.
- Las actividades están estrechamente ligadas causando así impedir que una actividad empiece antes de determinar totalmente la anterior, provocando invertir algún tiempo de desarrollo innecesariamente.

Capítulo 4

Desarrollo

Dentro de este capítulo se desglosarán las etapas que se siguieron en el desarrollo del SII, cada una de estas es parte de la metodología de cascada que fue la que se eligió para este desarrollo.

4.1. Análisis del sistema

Una de las actividades más importantes del Instituto Tecnológico de Tláhuac es el cálculo de indicadores, esto debido a que estos brindan información para la mejora de los procesos. El Sistema Integral de Indicadores debe ser capaz de mostrar la información requerida por los usuarios de una manera fácil, flexible y rápida.

La información que se requiere del sistema es el seguimiento de los procesos de los diferentes departamentos. Los departamentos a administrar por el sistema son los siguientes:

- Académico.
- Vinculación.
- Planeación.
- Administración de los recursos.
- Calidad.

Cada uno de ellos requiere obtener información específica la cual permitirá mejorar cada uno de sus procesos.

El sistema será capaz de recalcular los indicadores con los cambios que se le presenten a lo largo del periodo declarado, permitiendo con esto tomar decisiones estratégicas para mejorar en caso de que los indicadores sean bajos, o mantener las buenas prácticas en caso de que los indicadores sean los esperados.

Los reportes a configurar dentro del sistema contendrán un número indefinido de cálculos, es decir, el número de datos a calcular es dinámico permitiendo con esto darle flexibilidad y adaptación a diferentes circunstancias.

Se necesita que el sistema tenga las siguientes características:

- **Configurable:** El sistema debe tener la capacidad de ser configurado de acuerdo a las necesidades de los usuarios.
- **Escalable:** El sistema debe de ser capaz de adaptarse a nuevas funcionalidades y nuevos módulos, esto quiere decir que debe estar listo para que nuevos desarrolladores puedan usar como base este desarrollo para futuras adaptaciones.

-
- **Amigable:** Es importante que la presentación del sistema sea agradable para el usuario, debido a que esto mejora la experiencia del mismo permitiendo al usuario tener un ambiente de trabajo agradable.

Con estas tres características el sistema debe ser capaz de calcular y mostrar las cifras de los indicadores por departamento.

4.2. Análisis de requisitos del software

Para la realización del Sistema Integral de Indicadores se requiere que este se desarrolle en un ambiente Web, esto para permitir que múltiples usuarios puedan acceder a estos datos simultáneamente.

A lo largo de esta sección se detallaran los distintos tipos de requerimientos necesarios para el desarrollo del sistema.

4.2.1. Requerimientos físicos

Para el correcto funcionamiento del sistema es necesario contar con el equipo y software adecuado, del cual hablaremos a continuación:

- Hardware

- **Servidor 1:** Equipo con Windows Server 2012 R2 (recomendado) o algún sistema operativo con Windows a 64 bits dedicado para el servidor de base de datos. Mínimo 4GB de RAM, para su correcto funcionamiento se necesitan 6 GB o más, Procesador x64 a 1.4 GHz, para su correcto funcionamiento procesadores a 2.0 GHz o más. Mínimo 6GB de disco duro para su instalación, el tamaño del disco duro depende a las demandas de espacio al ir almacenando información.
- **Servidor 2:** Equipo con Windows Server 2012 R2 (recomendado) o algún sistema operativo con Windows a 64 bits dedicado para el servidor de aplicación. Mínimo 2GB de RAM, para su correcto funcionamiento se necesitan 4 GB o más, Procesador x64 a 1.4 GHz, para su correcto funcionamiento procesadores a 2.0 GHz o más. Disco duro de 500GB para almacenamiento de sitios y archivos en servidor.
- **Desarrollo:** Equipo con Windows Server 7 Ultimate a 64 bits dedicado para el desarrollo del sistema, 6GB de RAM o más. Procesador x64 a 1.4 GHz, para su correcto funcionamiento procesadores a 2.0 GHz o más. Disco duro de 500GB para almacenamiento de fuentes y archivos de instalación.

- Software

- Microsoft SQL Server 2012.
- IIS Server 8.0 o superior.
- Navegador Web (Mozilla Firefox preferentemente)
- Microsoft Visual Studio 2013 Test Premium

El servidor 1 deberá tener instalado Microsoft SQL Server 2012 para fungir como servidor de base de datos.

El servidor 1 deberá tener instalado IIS Server 8.0 para fungir como servidor de aplicación y almacenamiento de archivos.

El equipo de desarrollo deberá tener instalado Microsoft Visual Studio 2013 Test Premium, Navegador Web además de IIS Server 8.0 para fungir como ambiente de desarrollo.

4.2.2. Interfaces

El Sistema Integral de Indicadores será un sistema alimentado de todas las tablas del sistema general. Para esto es necesario establecer los métodos de entrada y salida de datos.

Entrada de datos

Para la entrada de datos es necesario contar con una serie de configuraciones que permitan obtener la información de cualquier tabla de la base de datos, esto permitirá obtener datos dinámicos adaptables a las necesidades que tenga en ese momento el usuario.

Los módulos para realizar la configuración del sistema son:

- **Departamentos:** En este módulo se permitirá declarar la información de los departamentos que se manejarán en el SII, los datos requeridos hasta el momento son Id, nombre y descripción.
- **Formulas:** En este módulo se configurarán las fórmulas que extraerán los datos de la base de datos y realizaran los cálculos establecidos en estas. Los campos requeridos para configurar una formula son id formula, nombre y la definición de la formula.
- **Colecciones:** Este módulo permite establecer las fórmulas que se aplicaran a cada proceso. Las formulas pueden estar en más de un proceso, esto debido a que se puede reutilizar un cálculo de un departamento a otro. Los campos necesarios en este módulo solamente son el id de formula y el id de colección.
- **Procesos:** En este se definen los datos del periodo dentro de los cuales se necesitan el departamento al que pertenece, numero interno de proceso, descripción, grupo de fórmulas a aplicar, fecha de inicio y fin, responsable y unidad de medida.

Salida de datos

Para la salida de datos se necesita hasta el momento una sola vista en donde se ve el resultado del proceso del periodo de indicadores, esta sección mostrara los siguientes datos:

- **Número interno de periodo de indicadores:** El número interno del periodo de indicadores es el identificador único para cada periodo de indicadores.
- **Número consecutivo:** Este se asignará al momento de resolver la formula dándole a cada registro del detalle del proceso un identificador.
- **Formula sistema:** Aquí se mostrará la formula tal cual está en el sistema al momento del proceso.
- **Formula pre compilado:** En esta se resuelven las funciones propias del sistema, es decir, las formulas programadas en el sistema, dando como resultado una formula la cual se encuentra lista para ser resuelta por el módulo de fórmulas de Microsoft Excel.
- **Resultado:** En este se mostrara el resultado de la formula resuelta por Excel y será el dato más interesante ya que será el resultado del indicador. De los datos insertados en el sistema, el que requiere de especial atención es la definición de la formula, esto debido a que si existe algún error en sintaxis al momento de procesar la formulas, esta retornara un valor de 0.

4.2.3. Usuarios y factores humanos

El Sistema Integral de Indicadores está pensado para tres tipos de usuarios de usuarios los cuales tendrán roles y permisos diferentes. Los usuarios que se podrán usar en el sistema son los siguientes:

- Usuario administrador.
- Usuario soporte
- Usuario consulta

Usuario administrador

Este usuario tendrá permisos para de acceso a todos los módulos teniendo la capacidad de modificar cualquier dato dentro del sistema, además de tener la capacidad de procesar y reprocesar los periodos de indicadores.

Este usuario debe tener conocimientos de programación, ya que tendrá que programar y modificar funciones para los cálculos de indicadores. Además deberá de tener conocimientos de rastreo de errores y base de datos para poder consultar los errores guardados en el log de base de datos cuando estos se generen.

El usuario de soporte debe hacerse cargo de dar soporte a los tickets reportados referentes a errores lógicos y ajustes de programación que se requieran según las necesidades, por lo que este deberá tener conocimientos en desarrollo Web con C# y MVC 4, además de conocer JQuery y bases de datos con Microsoft SQL Server 2012.

Usuario soporte

Este usuario tendrá la capacidad de modificar los datos de los módulos de fórmulas y grupos de fórmulas con lo cual podrá configurar los datos resultantes del proceso de indicadores. Este además podrá tener la capacidad de reprocesar periodos de indicadores con autorización de un usuario de soporte.

Este usuario deberá de tener conocimientos básicos en programación para realizar la creación y modificación de fórmulas de sistema.

Usuario consulta

Este usuario es el usuario más limitado del sistema, pues este solamente podrá consultar los resultados de los periodos procesados.

Este usuario no necesita conocimientos específicos ya que su función es solamente de consulta.

4.2.4. Funcionalidad

En esta sección se detalla la funcionalidad que el Sistema Integral de Indicadores tendrá.

La funcionalidad del Sistema Integral de Indicadores se verá marcada por 3 etapas muy importantes, las cuales son:

- Configuración.
- Proceso de datos.
- Visualización de información.

Configuración

En esta etapa se realizara la configuración necesaria la cual servirá para alimentar el motor de información, además de ser un paso necesario ya que sin él no será posible configurar los parámetros de los indicadores.

El proceso de configuración consta de 3 pasos los cuales se muestran en la figura 4.1.

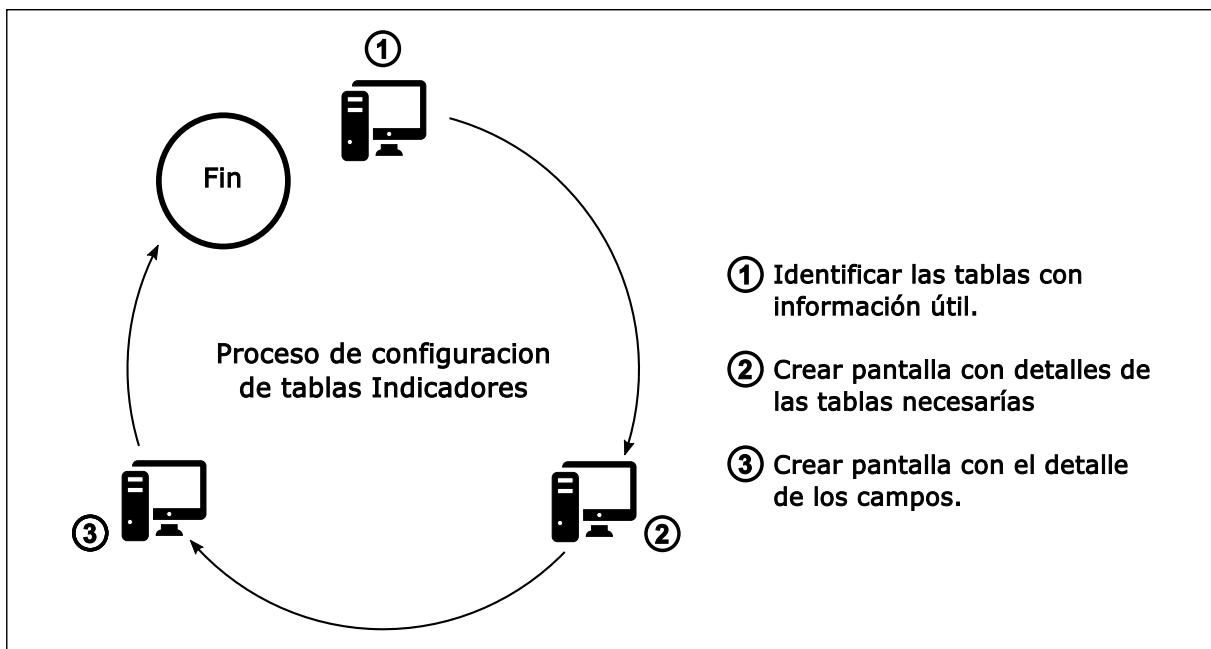


Figura 4.1: Diagrama del proceso de configuración de fórmulas de indicadores.

1. Identificar las tablas con información útil.

El Sistema Integral de Indicadores necesitará contar con la información de las tablas del sistema para que él pueda generar consultas dinámicas de información, es por esto que se pide identificar las tablas que contienen información que sea de utilidad para los fines de los indicadores. Estas tablas estarán disponibles para su consulta y creación de fórmulas permitiendo al usuario la facilidad de consultar cualquier campo que le sea de utilidad.

2. Crear pantalla con detalles de las tablas necesarias.

Este punto consta de realizar el diseño de una pantalla que permita mostrar las tablas en una pantalla para poder ser consultadas y aplicadas directamente a las fórmulas del sistema. Los datos a mostrar en la pantalla son solamente el nombre de la tabla, permitiendo con esto navegar y ubicar los datos de mejor manera. La pantalla de tablas por solicitud del usuario mostrara todas las tablas de la base de datos. Estas tablas se mostrarán en una lista en el módulo de creación de fórmulas.

3. Crear pantalla con el detalle de los campos.

Esta pantalla será para complementar la pantalla tablas ya que el detalle de los campos de cada tabla se mostrará en una segunda lista en el módulo de creación de fórmulas, permitiendo con esto tener a la mano todos las herramientas necesarias para el diseño de estas.

Proceso de datos

En este proceso se realizará una de las tareas más delicadas del sistema, ya que es aquí donde se procesará la información de los indicadores.

En este proceso se realizarán los cálculos de los indicadores, realizando el registro de los datos para tener la información organizada según los datos reales y además para hacer más eficiente la consulta de información.

En este paso se realizará lo descrito en la figura 4.2

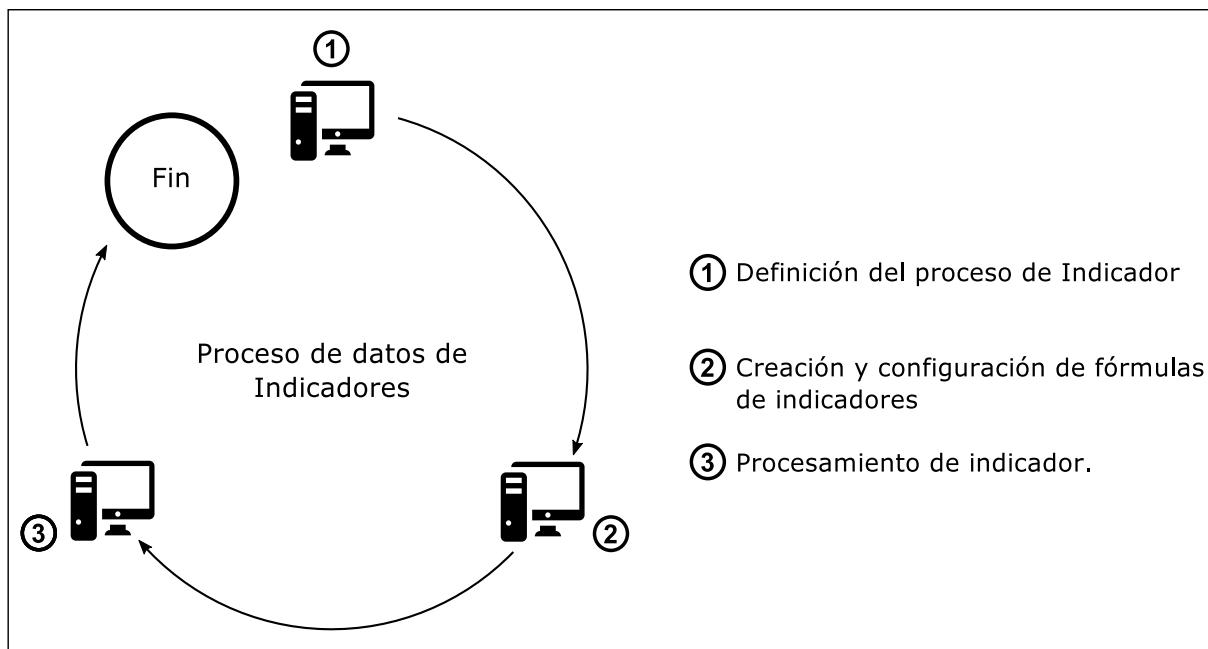


Figura 4.2: Diagrama del proceso de información de indicadores.

La definición de los indicadores es la parte en donde se definen los datos para generar los resultados de los indicadores, el cual se realizará de acuerdo a los lineamientos del Instituto Tecnológico de Tláhuac el cual establece que la información se organizará primeramente por institución, después se especificará el departamento al que pertenece el indicador y por último se organizarán por un rango de fechas especificado manualmente. Este rango de fechas podrá ser utilizado en la programación de las fórmulas permitiendo filtrar los datos.

1. Definición del proceso de indicador.

Este proceso consta de realizar la creación del proceso de indicador que se desee, para el cual se necesitan los siguientes datos:

- **ID institución:** Este campo contendrá el ID de la institución a la que pertenece el indicador, este campo formará parte de la llave para identificar el indicador.
- **Departamento:** Este campo es el número de departamento.
- **Número interno del proceso:** Este campo se autogenerara cuando se crea un nuevo proceso, ya que es único para poder identificar los procesos dentro de todo el sistema.
- **Descripción:** Este campo contendrá el detalle del contenido del indicador, permitiendo tener un resumen rápido del mismo.
- **Colección de fórmulas:** Este campo contendrá el número de colección que se utilizará para realizar el cálculo del indicador.
- **Fecha inicial:** Fecha de inicio del proceso.
- **Fecha final:** Fecha de fin del proceso.
- **Responsable:** Este campo es para almacenar el puesto o el nombre del responsable de calcular la información de este indicador.
- **Unidad de medida:** Especifica la unidad que se utiliza para medir el indicador, este regularmente es un porcentaje.

2. Configuración de fórmulas de indicadores.

Este proceso consiste en realizar la configuración de las fórmulas que permitirán de manera dinámica obtener los datos configurados en la misma.

Para realizar la configuración de la misma es necesario conocer algunos de los elementos que se pueden utilizar en las mismas los cuales son:

- Fórmulas definidas en el sistema
- Fórmulas de Microsoft Excel

Fórmulas definidas en el sistema: Esta funcionalidad permite realizar acciones generales basadas en funciones parametrizables, las cuales se podrán acceder con la siguiente nomenclatura NOMBRE_FUNCION (PARAMETRO_1, PARAMETRO_2 ... PARAMETRO_N).

Con esto la primera capa de resolución de fórmulas nos permite obtener datos procesados en fórmulas que se pueden programar en código, ampliando la potencia de la resolución de las mismas.

Fórmulas de Microsoft Excel: Esta funcionalidad es el segundo y último de los niveles de resolución de fórmulas del sistema de indicadores, ya que al tener los datos obtenidos mediante la resolución de fórmulas de sistema, podemos realizar la resolución de fórmulas de Excel, permitiendo usar la potencia del motor de fórmulas de dicho software, y al tener esta posibilidad se pueden obtener los resultados como si estuviéramos en el mismo software.

3. Procesamiento del indicador.

Este es el paso final del proceso de datos el cual consiste en realizar el procesamiento de las fórmulas y almacenamiento de dichos datos.

Este proceso se planeó con el fin de tener información específica además de registrar estos datos en la base de datos para mejorar la eficiencia de la visualización de los indicadores. Si este paso no existiera, la resolución de las fórmulas se tendría que realizar cada que un indicador se visualice generando con esto posibles inconsistencias en la información por la adición nuevos datos, además de lentitud al momento de usar el sistema.

4.3. Diseño del sistema

Para el diseño del sistema se tomarán dos puntos de la metodología de cascada los cuales son:

- Estructura de los datos.
- Caracterización de la interfaz.

4.3.1. Estructura de los datos

Para la creación de la base de datos del Sistema Integral de Indicadores es necesario identificar las entidades principales para poder comenzar con el proceso de diseño de la misma. Las entidades que se utilizarán en el sistema son:

- Departamentos.
- Fórmulas.
- Grupos de fórmulas.
- Proceso.

Estas entidades están ligadas con una que ya existe llamada Institución, la cual tiene la información de las instituciones que administra el sistema.

Para relacionar las entidades identificadas se creó un diagrama entidad-relación como se muestra en la figura 4.3.

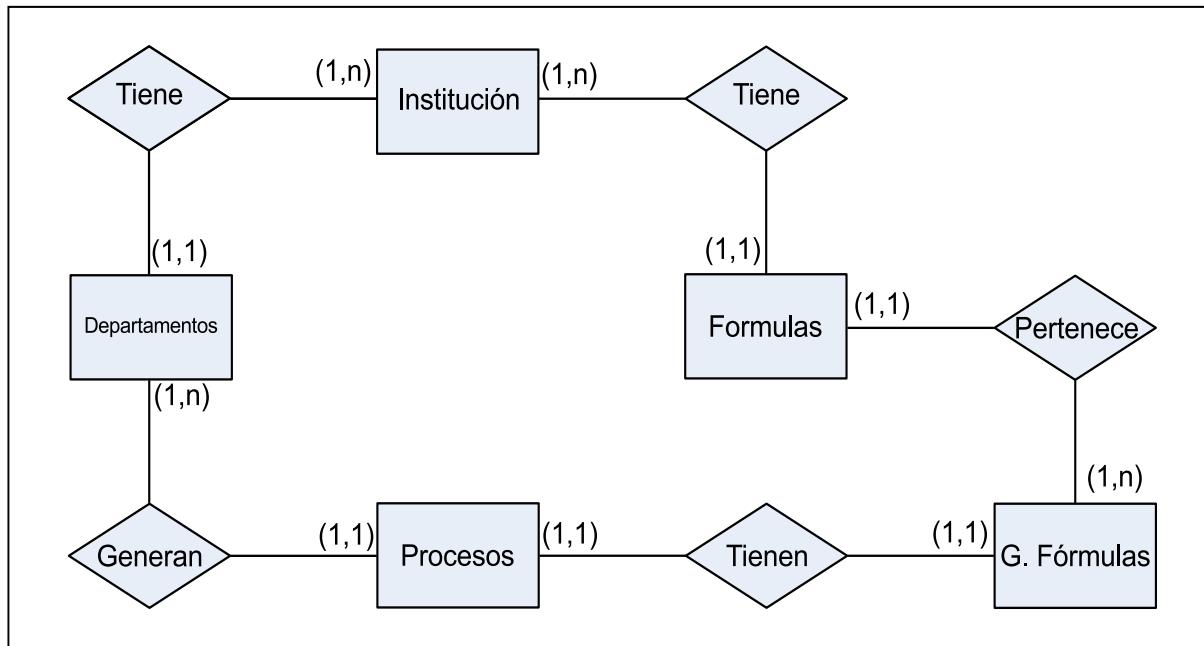


Figura 4.3: Diagrama entidad relación del sistema de indicadores.

- **institución:** Esta entidad es parte del sistema general del Tecnológico de Tláhuac por lo que solamente haremos referencia a ella para validar la integridad de los datos verificando que los indicadores estén ligados a una institución existente.
- **Departamentos:** La entidad departamentos contendrá la información necesaria de cada departamento como son un identificador, su nombre y descripción. Posteriormente se podrán agregar campos para futuras funcionalidades.
- **Formulas:** Esta entidad contendrá la información de las formulas necesaria para ser identificada en el proceso. La información que se requiere es la institución, identificador, nombre y la definición de la fórmula. Posteriormente se pueden añadir campos para futuras funcionalidades.
- **GFormulas:** Esta entidad está encargada de agrupar las formulas en grupos. Estos grupos serán utilizados para ser procesados en un proceso de indicadores, esta entidad se considerará una entidad compuesta partiendo los datos en encabezado y detalle, permitiendo con esto evitar la redundancia de datos.
- **Procesos:** Esta entidad es la parte más importante del sistema de indicadores, pues es en ella donde se almacena el resultado del cálculo y avance de los procesos de indicadores. Esta es considerada una entidad compuesta partiendo los datos en encabezado y detalle.

Después de identificar las entidades compuestas el diagrama sufre algunos cambios, los cuales se muestran en la figura 4.5.

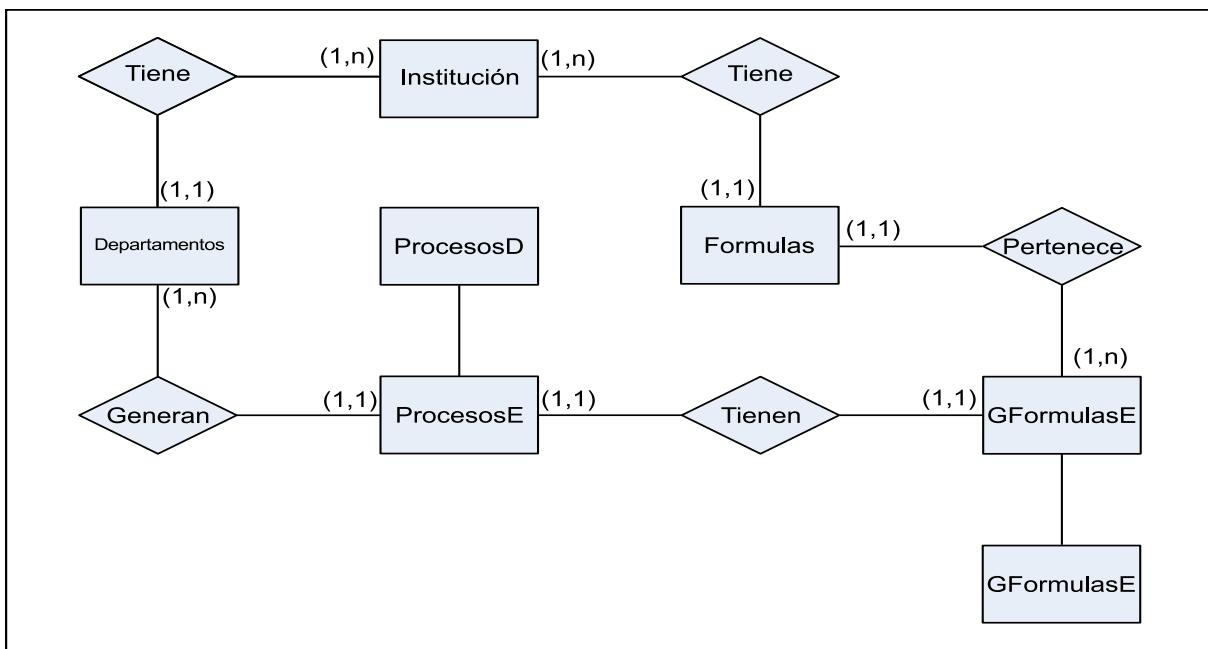


Figura 4.4: Diagrama entidad relación con tablas compuestas del sistema de indicadores.

Definición del diccionario de datos del sistema

De acuerdo a la definición de los datos, se puede generar el diagrama relacional indicando como es que los datos estarán organizados en la base de datos. La figura 4.5 ilustra este diagrama.

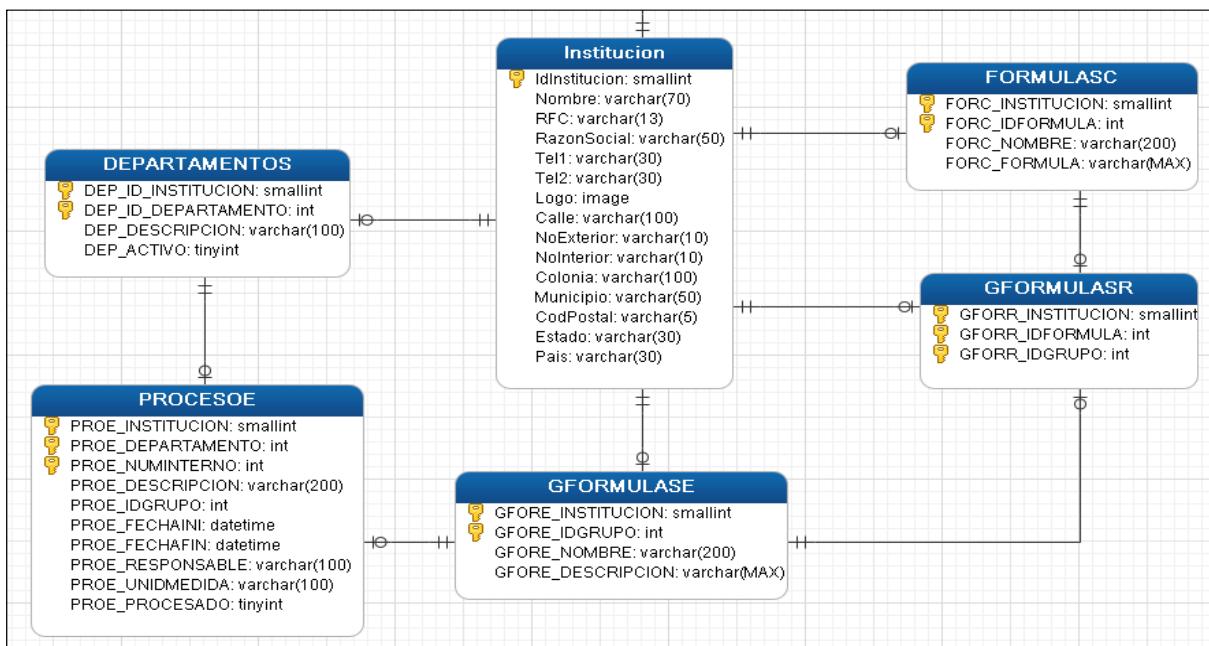


Figura 4.5: Diagrama relacional con campos del sistema de indicadores.

Definición del diccionario de datos del sistema

De acuerdo a los diagramas anteriores, se procede a realizar la definición del diccionario de datos para la

Tabla 4.1: Detalle de los campos de la tabla departamentos.

DEPARTAMENTOS						
Nombre	Tipo	TipoNativo	Longitud	Precisión	Orden	PK
DEP_ACTIVO	tinyint	tinyint	1	3	4	No
DEP_ID_INSTITUCION	smallint	smallint	2	5	1	Si
DEP_ID_DEPARTAMENTO	int	int	4	10	2	Si

Tabla 4.2: Detalle de los campos de la tabla formulasc.

FORMULASC						
Nombre	Tipo	TipoNativo	Longitud	Precisión	Orden	PK
FORC_INSTITUCION	smallint	smallint	2	5	1	Si
FORC_IDFORMULA	int	int	4	10	2	Si
FORC_NOMBRE	varchar	varchar	200	200	3	No
FORC_FORMULA	varchar	varchar	Max	Max	4	No

Tabla 4.3: Detalle de los campos de la tabla gformulase.

GFORMULASE						
Nombre	Tipo	TipoNativo	Longitud	Precisión	Orden	PK
GFORE_INSTITUCION	smallint	smallint	2	5	1	Si
GFORE_IDGRUPO	int	int	4	10	2	Si
GFORE_NOMBRE	varchar	varchar	200	200	3	No
GFORE_DESCRIPCION	varchar	varchar	Max	Max	4	No

Tabla 4.4: Detalle de los campos de la tabla gformulasr.

GFORMULASR						
Nombre	Tipo	TipoNativo	Longitud	Precisión	Orden	PK
GFORR_INSTITUCION	smallint	smallint	2	5	1	Si
GFORR_IDFORMULA	int	int	4	10	2	Si
GFORR_IDGRUPO	int	int	4	10	3	Si

Tabla 4.5: Detalle de los campos de la tabla procesoe.

PROCESOE						
Nombre	Tipo	TipoNativo	Longitud	Precisión	Orden	PK
PROE_PROCESADO	tinyint	tinyint	1	3	10	No
PROE_INSTITUCION	smallint	smallint	2	5	1	Si
PROE_DEPARTAMENTO	int	int	4	10	2	Si
PROE_NUMINTERNO	int	int	4	10	3	Si
PROE_IDGRUPO	int	int	4	10	5	No
PROE_FECHAINI	datetime	datetime	8	23	6	No
PROE_FECHAFIN	datetime	datetime	8	23	7	No
PROE_DESCRIPCION	varchar	varchar	200	200	4	No
PROE_RESPONSABLE	varchar	varchar	100	100	8	No
PROE_UNIDMEDIDA	varchar	varchar	100	100	9	No

Tabla 4.6: Detalle de los campos de la tabla procesod.

PROCESOD						
Nombre	Tipo	TipoNativo	Longitud	Precisión	Orden	PK
PROD_INSTITUCION	smallint	smallint	2	5	1	Si
PROD_NUMINTERNO	int	int	4	10	2	Si
PROD_CONSECUITIVO	int	int	4	10	3	Si
PROD_FORMULASIS	varchar	varchar	Max	Max	4	No
PROD_FORMULAPRE	varchar	varchar	Max	Max	5	No
PROD_FORMULARES	varchar	varchar	Max	Max	6	No

Con esto la definición de los datos queda completa, permitiendo tener los argumentos suficientes para la descripción de las interfaces.

4.4. Caracterización de la interfaz

Para la definición de la funcionalidad de la interfaz se tomaran en cuenta los siguientes requerimientos generales:

- Los tipos de dato de los componentes de entrada de las interfaces estarán definidos de acuerdo al tipo de dato nativo en la BD, esto para mejorar la experiencia del usuario.
- Todas las entradas de datos se mostrarán en ventanas emergentes.
- La visualización de los datos se hará en una tabla para mejorar la accesibilidad de los datos
- Las acciones propias a los registros serán colocadas a la derecha de la tabla, estas no tendrán título de encabezado.
- Se crearan los botones de navegación del sistema en la parte superior derecha de las tablas, estas acciones serán determinadas por módulo.
- En el caso de que no existan registros para mostrar en el módulo, deberá de mostrar un mensaje descriptivo a la situación, dejando invisibles los encabezados de la tabla propia del módulo.

En el caso de algunos módulos, se deberán de tomar algunas consideraciones especiales las cuales se describe continuación:

-
- **Colecciones:** En el caso de este formulario se deberá agregar una columna que muestre un botón, este botón re direccionara al formulario DetalleColecciones.
 - **DetalleColecciones:** Este formulario contara únicamente con las funciones de agregar y eliminar, se deshabilita la función de actualizar ya que aquí no se pueden modificar las formulas si no solo agregarlas o eliminarlas.
La funcionalidad de agregar se mostrara en una ventana emergente la cual tendrá únicamente una lista multi-selección que mostrara las fórmulas que están disponibles para ser agregadas a la colección.
 - **Proceso:** Este formulario mostrara los registros del encabezado de los procesos con las siguientes funcionalidades.
 - **Crear:** Esta funcionalidad permitirá crear un nuevo proceso.
 - **Actualizar:** Esta funcionalidad permite actualizar los datos del periodo. Cuando se realiza esta acción es importante mencionar que el periodo puede que necesite reprocesarse, ya que los datos como las fechas de inicio y fin o el número de colección de fórmulas afectan directamente en el resultado.
 - **Eliminar:** Esta funcionalidad permite eliminar un proceso. Es importante destacar que al eliminar un proceso se eliminaran tanto los registros en el encabezado y el detalle.
 - **Procesar:** Esta funcionalidad ejecuta la resolución de fórmulas del sistema guardando los resultados en el detalle del proceso.
 - **Detalle:** Esta funcionalidad muestra los registros que se encuentran en el detalle del proceso. Este formulario es solo de consulta por lo que no se tendrá ninguna funcionalidad.

4.5. Programación

Dentro de los puntos destacables para la codificación de este proyecto, se detectaron procesos los cuales, debido a su constante repetición, se hizo código reutilizable que permitiera manejar estos casos.

Para dar una solución a este problema, se utilizó el framework que el Instituto Tecnológico de Tláhuac está desarrollando, el cual cuenta con muchas tareas programadas que pueden ser de gran utilidad.

4.5.1. Manejo de interfaces de mensajes emergentes

Los mensajes emergentes dentro del sistema son una de las tareas que a nivel de programación requieren realizar la repetición de código, haciendo con esto que este se vuelva ilegible y muy difícil de entender. Para solucionar este punto se creó una clase dentro del proyecto de las utilerías del Framework del Instituto Tecnológico de Tláhuac la cual se encarga de realizar el manejo de la creación del código que contendrá el cuerpo de este mensaje.

Las acciones que soporta esta clase son las siguientes:

- Creación de ventana emergente para creación, actualización y eliminación de datos del sistema.
- Ejecución de acciones de creación, actualización y eliminación de información.
- Acciones especiales de módulos.

Cada una de estas acciones está delimitada a usarse de acuerdo al nombre del módulo que se ejecuten, es decir, si se quisiera la ventana emergente para la creación de un nuevo registro en el módulo de departamentos sea visible, basta con indicar que se requiere el alta del formulario junto con el nombre del departamento.

Las acciones a ejecutar dentro de cada uno de estas ventanas emergentes se encuentra también en esta clase, la cual realiza las actividades necesarias para realizar la actualización o eliminación de estos modulo.

Capítulo 5

Pruebas y resultados

En este apartado se muestran las pruebas y resultados del funcionamiento del sistema, las cuales serán organizadas por módulo.

Los modulos que se someterán a revisión son los siguientes:

- Departamentos.

- Fórmulas.

- Grupos de fórmulas.

- Proceso.

Las pruebas que se realizaran son las siguientes:

- Visualización del módulo y su respectiva información.

- Creación de un nuevo registro.

- Eliminación de un nuevo registro.

- Actualización de un registro existente.

- Acciones adicionales en caso de que existan.

A continuación se procede a mostrar las pruebas de los módulos.

5.1. Pruebas y resultados de las páginas principales del SII.

Como primera prueba se realizara la ejecución de la página principal del sistema, la cual se muestra en la figura 5.1



Figura 5.1: Pantalla de inicio del Sisitema Integral de Información del ITT.

En esta pantalla se pueden visualizar las diferentes opciones que tendrá el sistema, en esta ocasión el único con funcionalidad es el de indicadores. Estas funcionalidades se mostraran en un futuro dependiendo del tipo de usuario que ingrese al sistema, ya que en este caso los indicadores, solamente podrán ser visualizados por personal autorizado.

Para entrar a las opciones permitidas en los indicadores se tiene que dar click el el botón con la leyenda “indicadores” que se encuentra en el menú principal mostrando la pantalla que se muestra en la figura 5.2.

Esta pantalla muestra dos funcionalidades las cuales son:

- **Configuración:** Pantalla con las opciones para configurar las funciones y datos necesarias para realizar un periodo de indicadores.
- **Proceso:** Pantalla con todas las funcionalidades para consultar y procesar los periodos de indicadores.

5.2. Pruebas y resultados de módulo Departamentos.

Para ingresar al módulo de configuración de departamentos, es necesario dar click en el botón Configuración → Departamentos, lo cual nos mostrara la ventana ilustrada en la figura 5.3.



Figura 5.2: Pantalla de inicio del Sisitema Integral de Indicadores.



Figura 5.3: Pantalla principal de departamentos.

El módulo de departamentos cuenta con las siguientes funciones para validar:

- Alta de departamento.
- Actualización de departamentos.
- Baja de departamentos.

Estas acciones se muestran a continuación:

5.2.1. Alta de departamento

Para crear un departamento, se tiene que dar click en el botón crear del módulo de configuración de departamentos, mostrando con esto la ventana emergente que se puede ver en la figura 5.4.

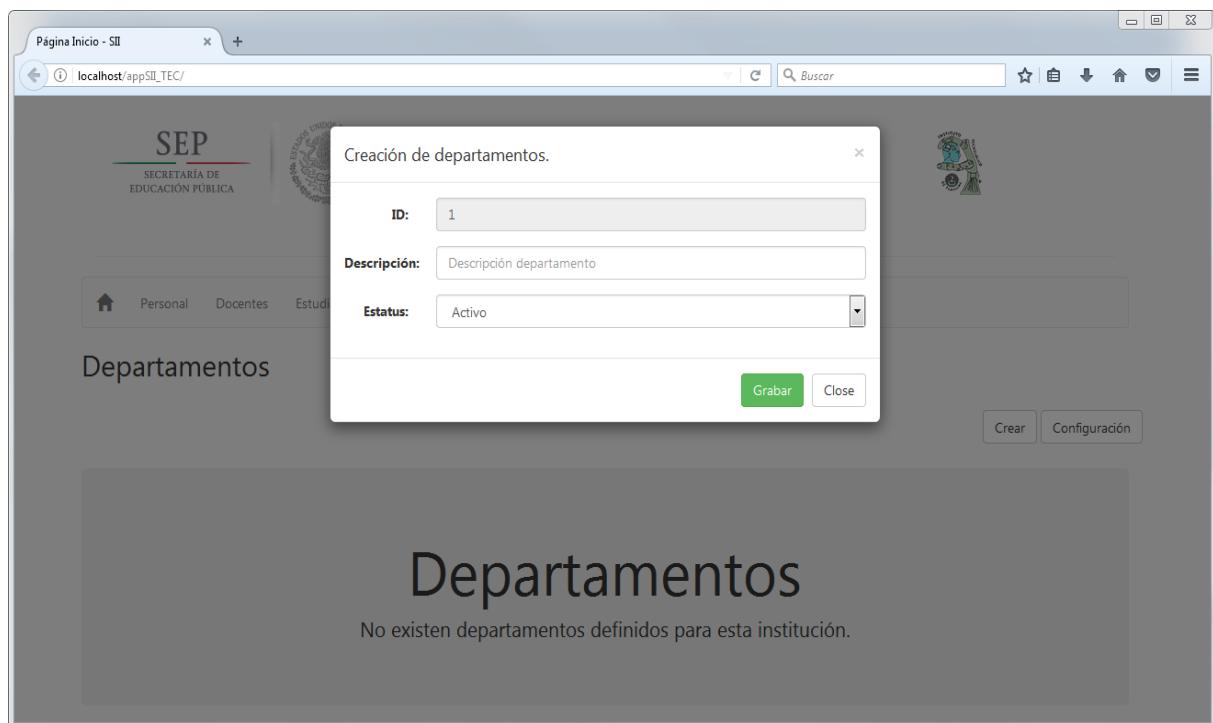


Figura 5.4: Ventana emergente para creación de departamentos.

Los datos que se solicitan para la creación de un departamento son los siguientes:

1. **ID:** Número identificador, este se genera de manera automática, es por esto que se muestra deshabilitado.
2. **Descripción:** Campo para almacenar una breve descripción del departamento.
3. **Estatus:** Campo para indicar si el departamento es activo o no.

Cuando uno de estos datos no se ingresa correctamente, se muestra un mensaje de error tal como se muestra en la figura 5.5.

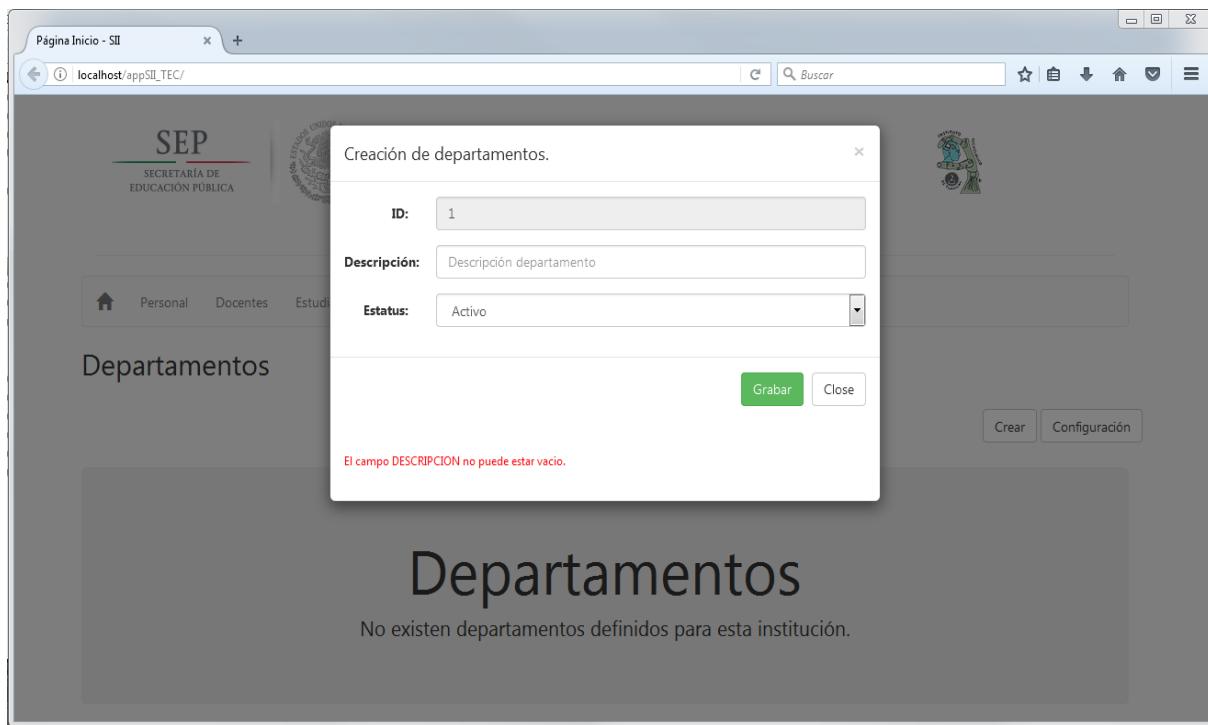


Figura 5.5: Mensaje de validación departamentos.

Al momento de ingresar los datos correctamente, el sistema graba el registro en la base de datos y manda un mensaje indicando que el proceso se realizó correctamente, esto se muestra en la figura 5.6.

Una vez que existen departamentos registrados en la base de datos, estos se muestran en una tabla, la cual tendrá dos acciones específicas por departamento, una es actualizar y la otra es eliminar. Esta tabla de departamentos se muestra en la figura 5.7.

5.2.2. Actualización de departamentos

Para la actualización de departamentos se muestra la misma ventana emergente que se usa para crear uno, pero en este caso se visualizan los datos del departamento que se seleccionó. En este caso se muestra la ventana emergente en la que se quiere actualizar el departamento académico, esta ventana se muestra en la figura 5.8.

5.2.3. Baja de departamentos

Como última prueba se muestra en la figura 5.9 la ventana emergente mostrada al momento de querer eliminar un departamento, el sistema pregunta que el usuario este seguro de eliminar este registrado.

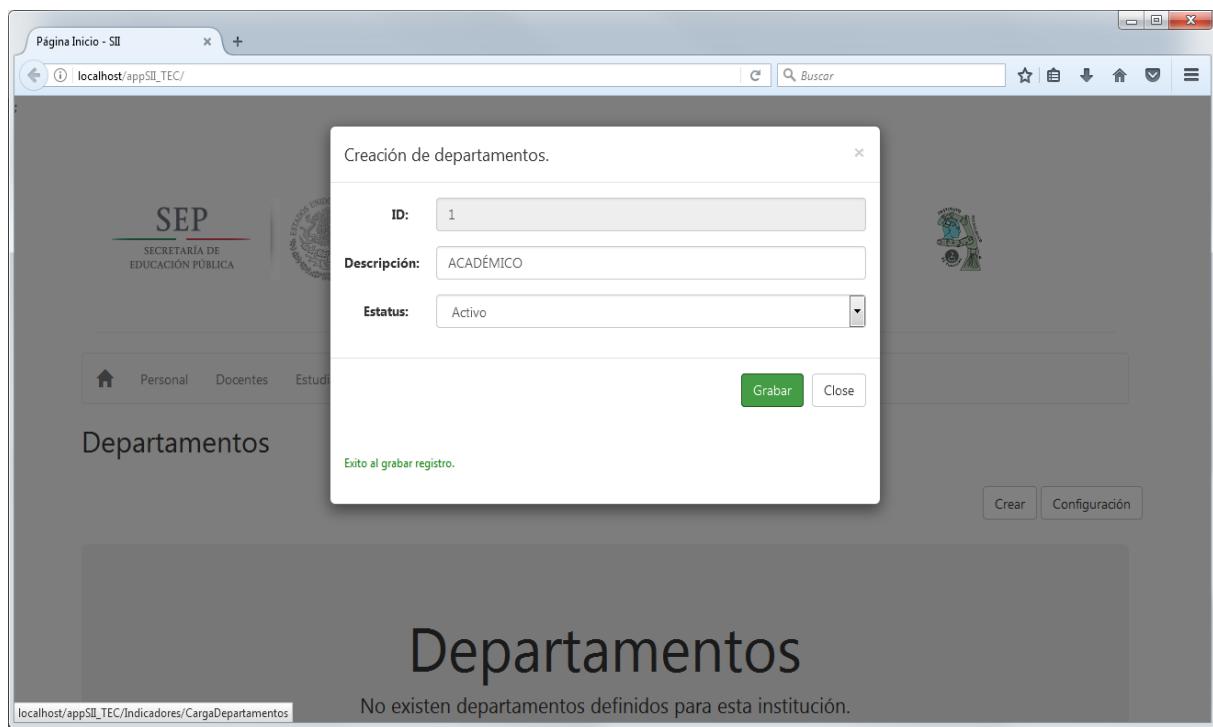


Figura 5.6: Mensaje de grabación satisfactoria de departamento.

The screenshot shows a web browser window titled 'Página Inicio - SII' with the URL 'localhost/appSII_TEC/'. The main content area displays the header 'TECNOLÓGICO NACIONAL DE MÉXICO' and 'INSTITUTO TECNOLÓGICO DE TLÁHUAC'. Below the header, there is a navigation bar with links: 'Personal', 'Docentes', 'Estudiantes', 'Aspirantes', and 'Indicadores'. A 'Departamentos' section is shown with a table listing one department:

ID Institución	ID Departamento	Descripción	Estatus
1	1	ACADÉMICO	Activo

At the bottom of the page, there is a copyright notice: '© 2017 Instituto Tecnológico de Tláhuac - Sistema Integral de Información - Centro de Cómputo'.

Figura 5.7: Listado de departamentos.

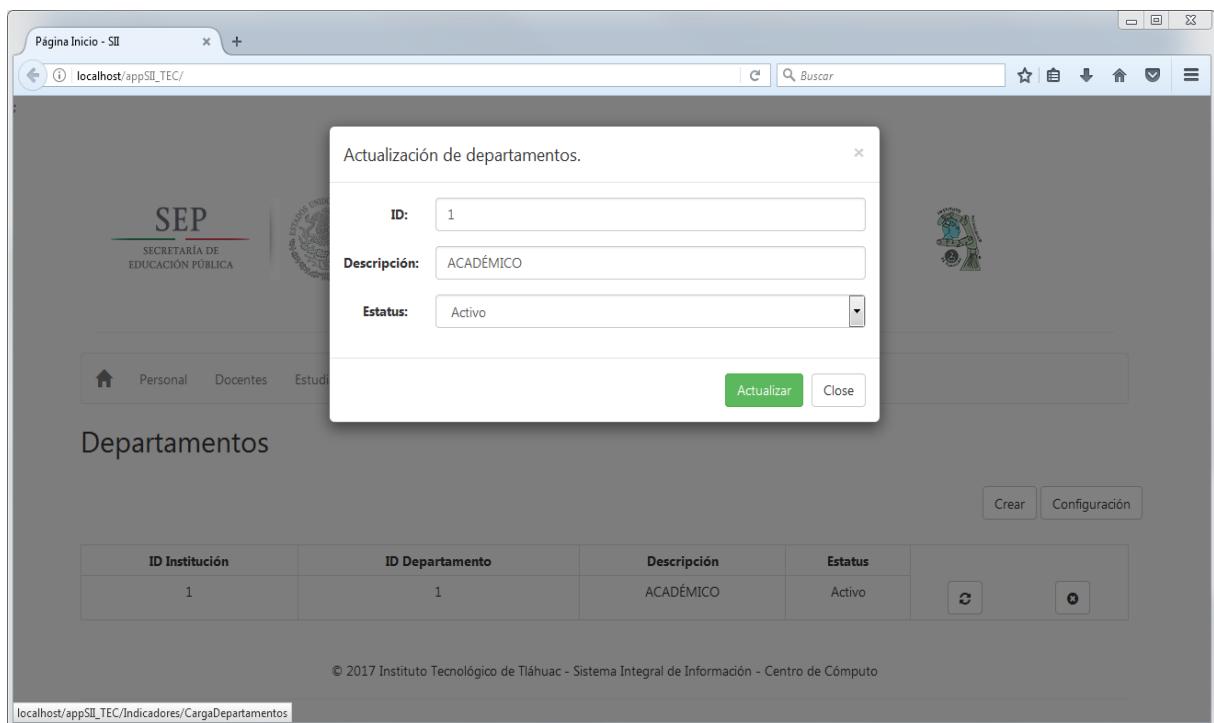


Figura 5.8: Ventana emergente de actualización de departamento.

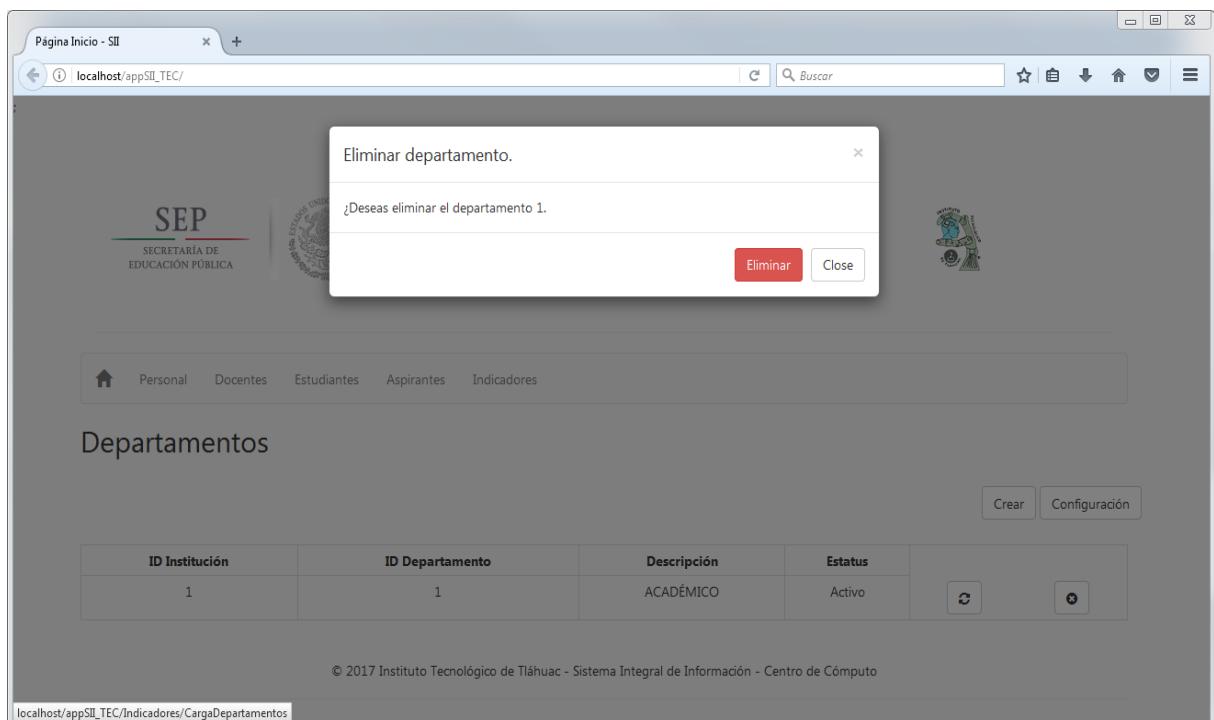


Figura 5.9: Mensaje de advertencia al eliminar departamento.

5.3. Pruebas y resultados de módulo Fórmulas.

Para ingresar al módulo de configuración de fórmulas, es necesario dar click en el botón Configuración → Formulas, lo cual nos mostrara la ventana ilustrada en la figura 5.10.



Figura 5.10: Pantalla principal de fórmulas.

El módulo de fórmulas cuenta con las siguientes funciones para validar:

- Alta de fórmulas.
- Actualización de fórmulas.
- Baja de fórmulas.

Estas acciones se muestran a continuación:

5.3.1. Alta de fórmula

Para crear una fórmula, se tiene que dar click en el botón crear del módulo de configuración de fórmulas, mostrando con esto la ventana emergente que se puede ver en la figura 5.11.

Los datos que se solicitan para la creación de una fórmula son los siguientes:

1. **ID:** Número identificador, este se genera de manera automática, es por esto que se muestra deshabilitado.
2. **Nombre:** Campo para especificar el nombre con el que se identificará la fórmula.
3. **fórmula:** Campo para definir la fórmula.

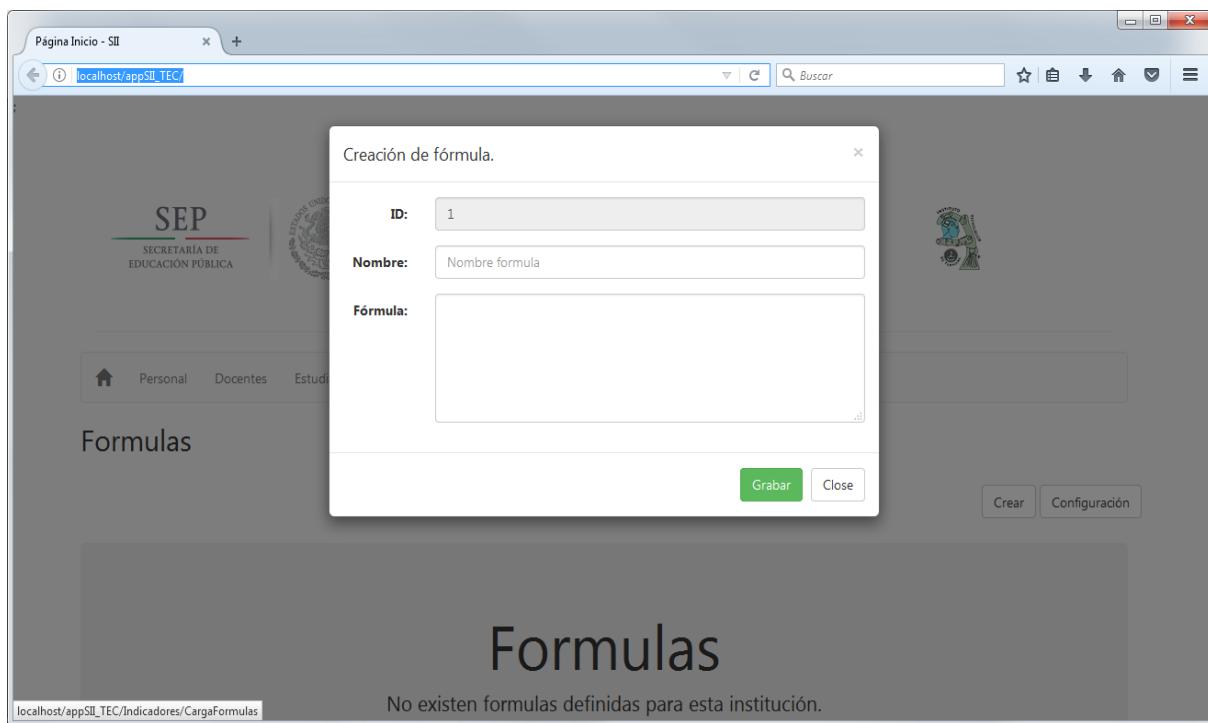


Figura 5.11: Ventana emergente para creación de fórmulas.

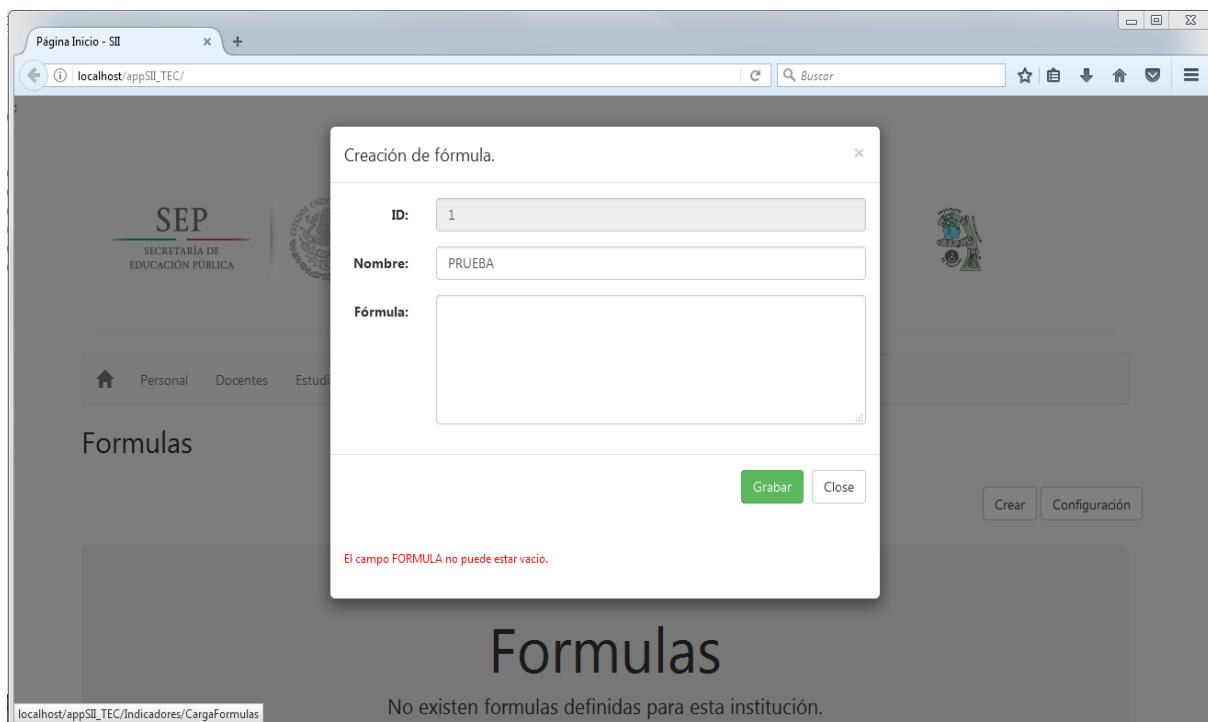


Figura 5.12: Mensaje de validación fórmulas.

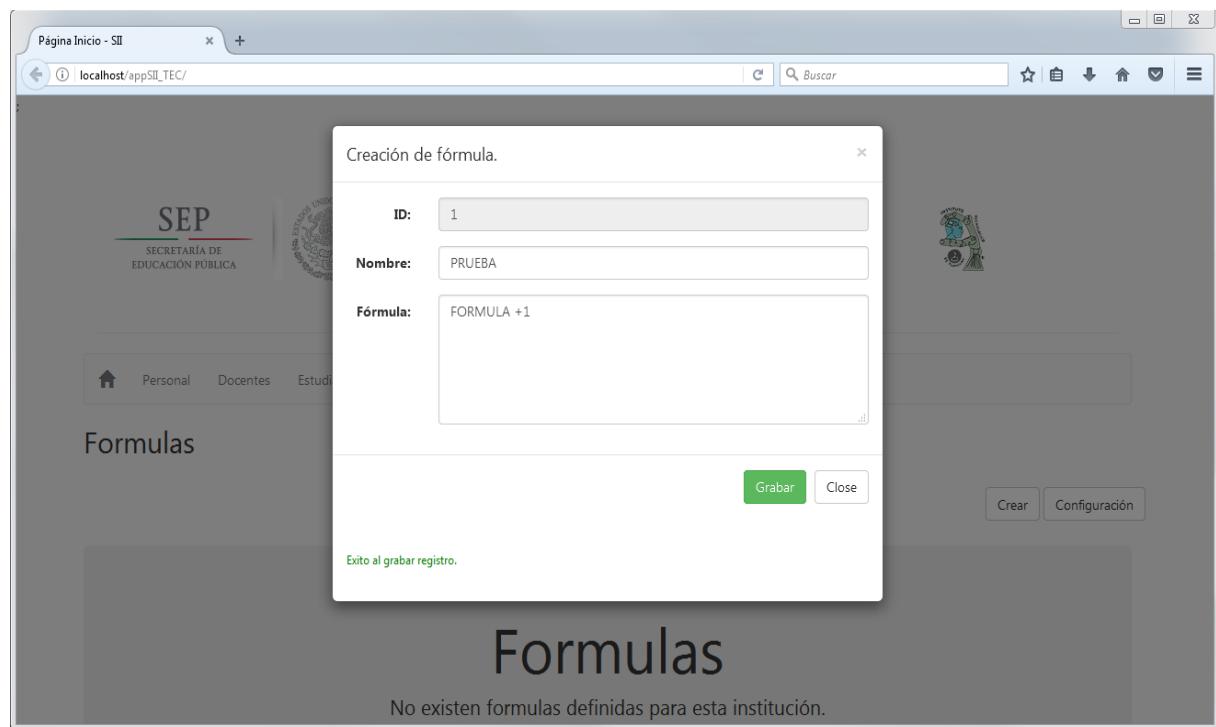


Figura 5.13: Mensaje de grabación satisfactoria de fórmula.

Cuando uno de estos datos no se ingresa correctamente, se muestra un mensaje de error tal como se muestra en la figura 5.12.

Al momento de ingresar los datos correctamente, el sistema graba el registro en la base de datos y manda un mensaje indicando que el proceso se realizó correctamente, esto se muestra en la figura 5.13.

Una vez que existen fórmulas registrados en la base de datos, estos se muestran en una tabla, la cual tendrá dos acciones específicas por fórmula, una es actualizar y la otra es eliminar. Esta tabla de fórmula se muestra en la figura 5.14.

5.3.2. Actualización de fórmulas

Para la actualización de fórmulas se muestra la misma ventana emergente que se usa para crear una, pero en este caso se visualizan los datos de la fórmula que se seleccionó. En este caso se muestra la ventana emergente en la que se quiere actualizar la fórmula de PRUEBA, esta ventana se muestra en la figura 5.15.

5.3.3. Baja de fórmulas

Como última prueba se muestra en la figura 5.16 la ventana emergente mostrada al momento de querer eliminar una fórmula, el sistema pregunta al usuario si está seguro de eliminar este registró.

The screenshot shows a web browser window titled 'Página Inicio - SII'. The URL in the address bar is 'localhost/appSII_TEC/'. The page header includes the SEP logo, the Mexican National Seal, and the text 'TECNOLÓGICO NACIONAL DE MÉXICO' and 'INSTITUTO TECNOLÓGICO DE TLÁHUAC'. A navigation menu at the top has links for 'Personal', 'Docentes', 'Estudiantes', 'Aspirantes', and 'Indicadores'. Below the menu, the title 'Formulas' is displayed. To the right of the title are two buttons: 'Crear' and 'Configuración'. A table lists one formula entry: ID Formula 1, Nombre PRUEBA, and Fórmula FORMULA +1. There are edit and delete icons next to the row. At the bottom of the page, a copyright notice reads '© 2017 Instituto Tecnológico de Tláhuac - Sistema Integral de Información - Centro de Cómputo'.

Figura 5.14: Listado de fórmulas.

The screenshot shows a modal dialog box titled 'Actualizar de fórmula.' It contains three input fields: 'ID:' with value '1', 'Nombre:' with value 'PRUEBA', and 'Fórmula:' with value 'FORMULA +1'. At the bottom of the dialog are two buttons: 'Actualizar' (highlighted in green) and 'Close'. The background of the main application window is dimmed. The main application window shows the same 'Formulas' list as Figure 5.14, with the updated entry visible. A copyright notice at the bottom of the main window reads '© 2017 Instituto Tecnológico de Tláhuac - Sistema Integral de Información - Centro de Cómputo'.

Figura 5.15: Ventana emergente de actualización de fórmula.

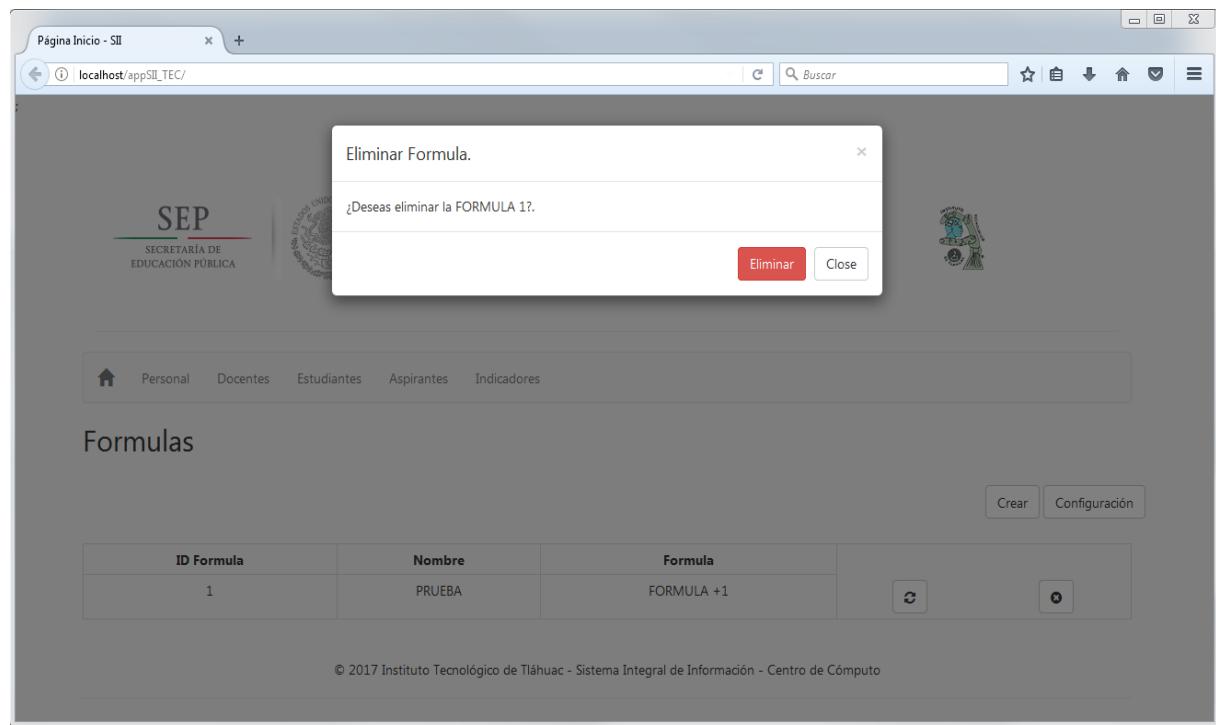


Figura 5.16: Mensaje de advertencia al eliminar fórmula.

5.4. Pruebas y resultados de módulo Colecciones.

Para ingresar al módulo de configuración de colecciones, es necesario dar click en el botón Configuración → Colecciones, lo cual nos mostrara la ventana ilustrada en la figura 5.17.

El módulo de colecciones cuenta con las siguientes funciones para validar:

- Alta de colecciones.
- Actualización de colecciones.
- Baja de colecciones.
- Detalle de colecciones.

Estas acciones se muestran a continuación:

5.4.1. Alta de colección

Para crear una colección, se tiene que dar click en el botón crear del módulo de configuración de colecciones, mostrando con esto la ventana emergente que se puede ver en la figura 5.18.

Los datos que se solicitan para la creación de una colección son los siguientes:

1. **ID:** Número identificador, este se genera de manera automática, es por esto que se muestra deshabilitado.
2. **Nombre:** Campo para especificar el nombre con el que se identificará la fórmula.
3. **Descripción:** Campo para almacenar una breve descripción de la colección.



Figura 5.17: Pantalla principal de colecciones.

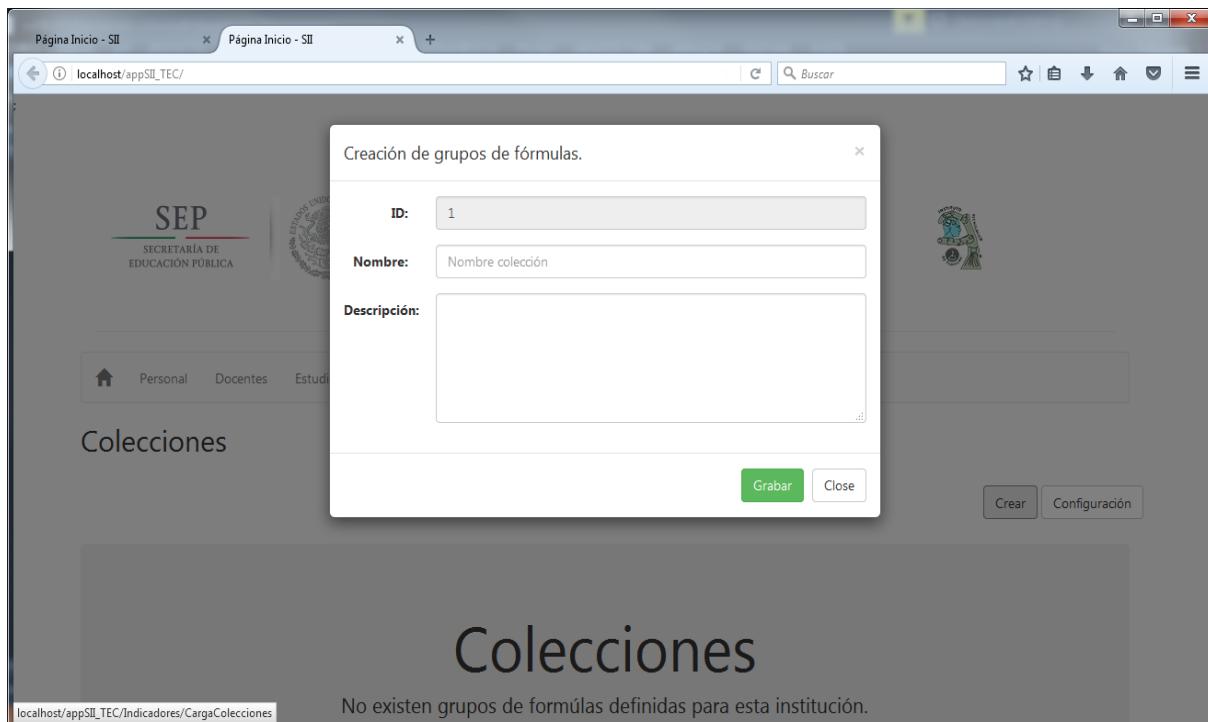


Figura 5.18: Ventana emergente para creación de colecciones.

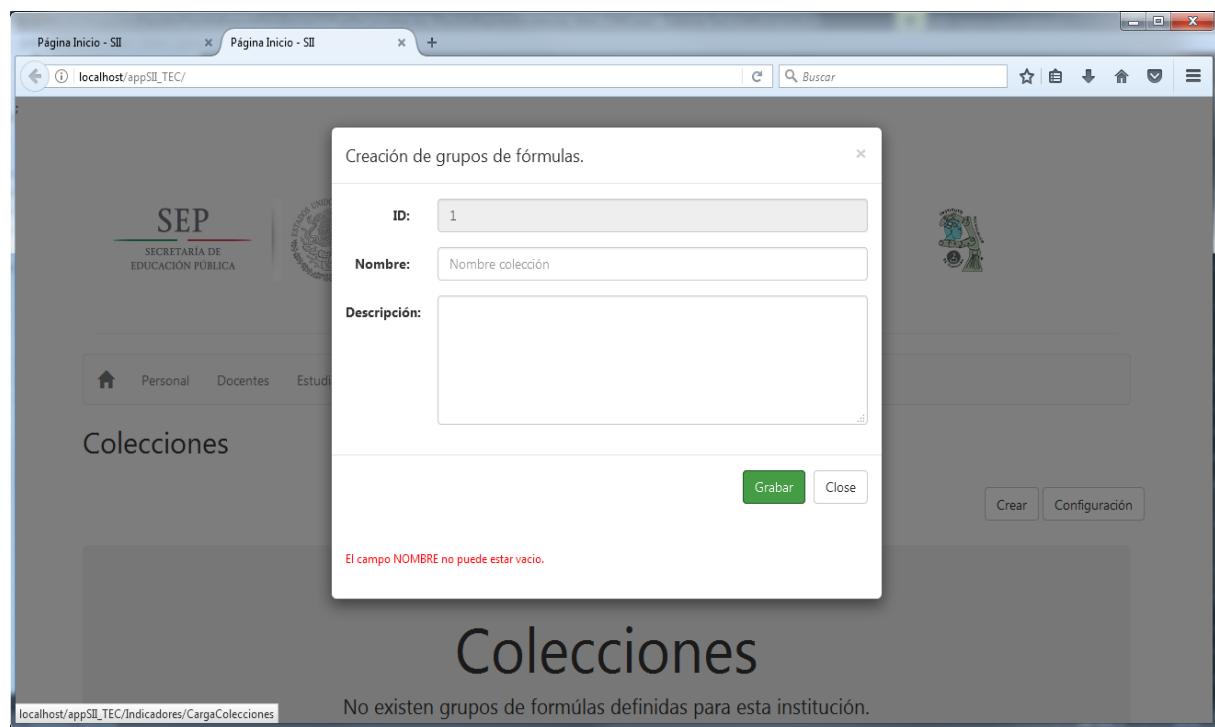


Figura 5.19: Mensaje de validación colecciones.

Cuando uno de estos datos no se ingresa correctamente, se muestra un mensaje de error tal como se muestra en la figura 5.19.

Al momento de ingresar los datos correctamente, el sistema graba el registro en la base de datos y manda un mensaje indicando que el proceso se realizó correctamente, esto se muestra en la figura 5.20.

Una vez que existen colecciones registrados en la base de datos, estos se muestran en una tabla, la cual tendrá dos acciones específicas por colección, una es actualizar, la otra es mostrar el detalle de la colección y por ultimo eliminar. Esta tabla de fórmula se muestra en la figura 5.21.

5.4.2. Actualización de colecciones

Para la actualización de colecciones se muestra la misma ventana emergente que se usa para crear una, pero en este caso se visualizan los datos de la colección que se seleccionó. En este caso se muestra la ventana emergente en la que se quiere actualizar la colección de SERVICIO SOCIAL, esta ventana se muestra en la figura 5.22.

5.4.3. Baja de colección

Como última prueba se muestra en la figura 5.23 la ventana emergente mostrada al momento de querer eliminar una colección, el sistema pregunta al usuario si está seguro de eliminar este registró.

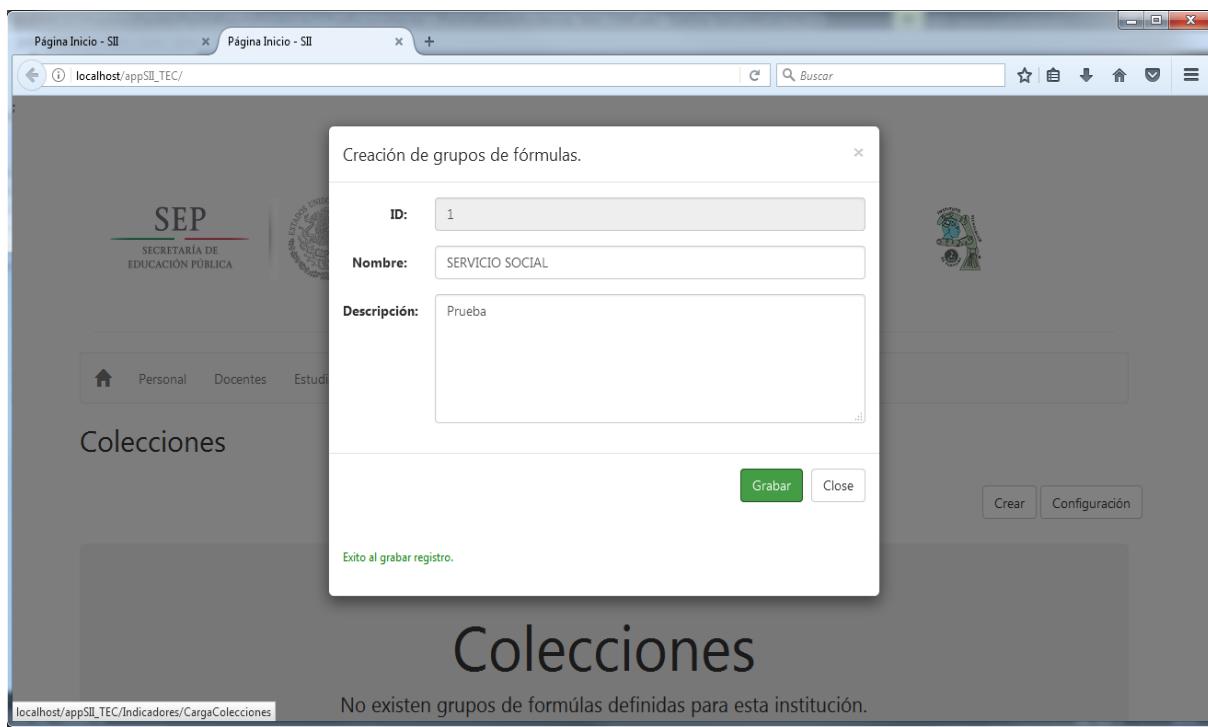


Figura 5.20: Mensaje de grabación satisfactoria de colección.

The screenshot shows a web browser window with two tabs: 'Página Inicio - SII' and 'Página Inicio - SII'. The URL in the address bar is 'localhost/appSII_TEC/'. The main content area displays the SEP logo, the TECNOLÓGICO NACIONAL DE MÉXICO logo, and the INSTITUTO TECNOLÓGICO DE TLÁHUAC logo. Below the logos is a navigation menu with links: 'Personal', 'Docentes', 'Estudiantes', 'Aspirantes', and 'Indicadores'. A large 'Colecciones' section is visible, with the subtext 'No existen grupos de formúlas definidas para esta institución.' and the URL 'localhost/appSII_TEC/Indicadores/CargaColecciones'. At the bottom of the page is a copyright notice: '© 2017 Instituto Tecnológico de Tláhuac - Sistema Integral de Información - Centro de Cómputo'. The page also features a table listing a single collection entry:

ID Grupo	Nombre	Descripción	Detalles		
1	SERVICIO SOCIAL	Prueba	Detalle		

Figura 5.21: Listado de colecciones.

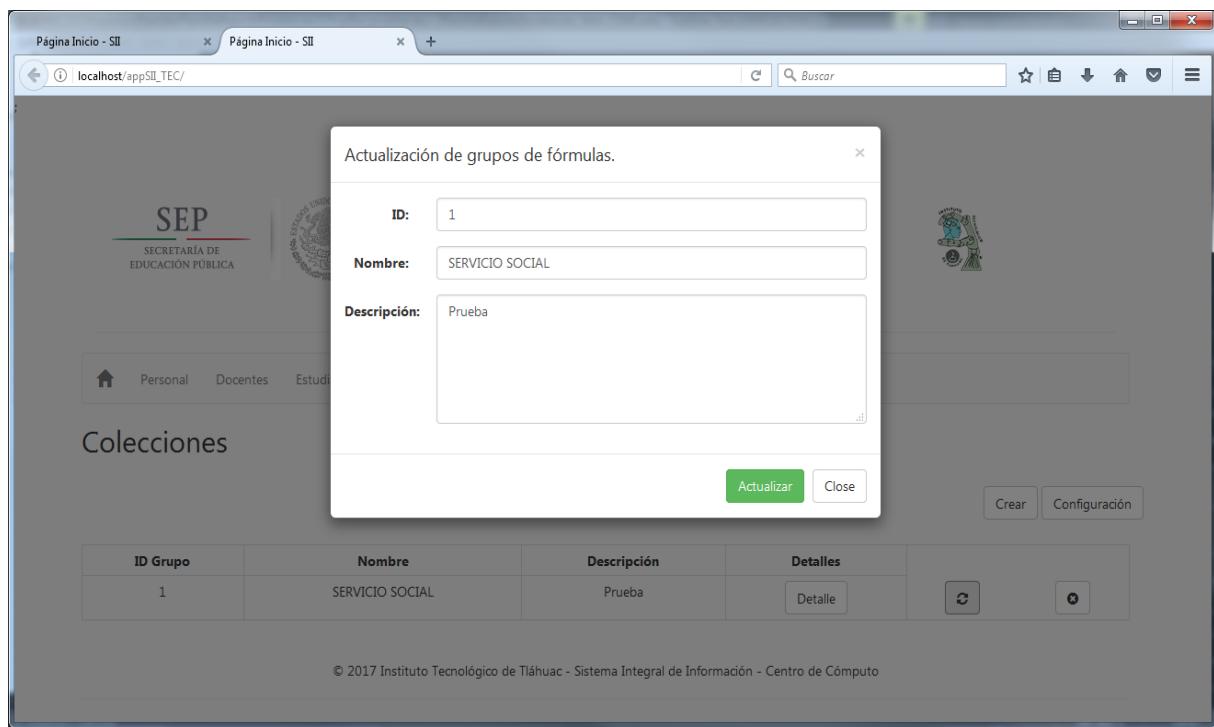


Figura 5.22: Ventana emergente de actualización de colecciones.

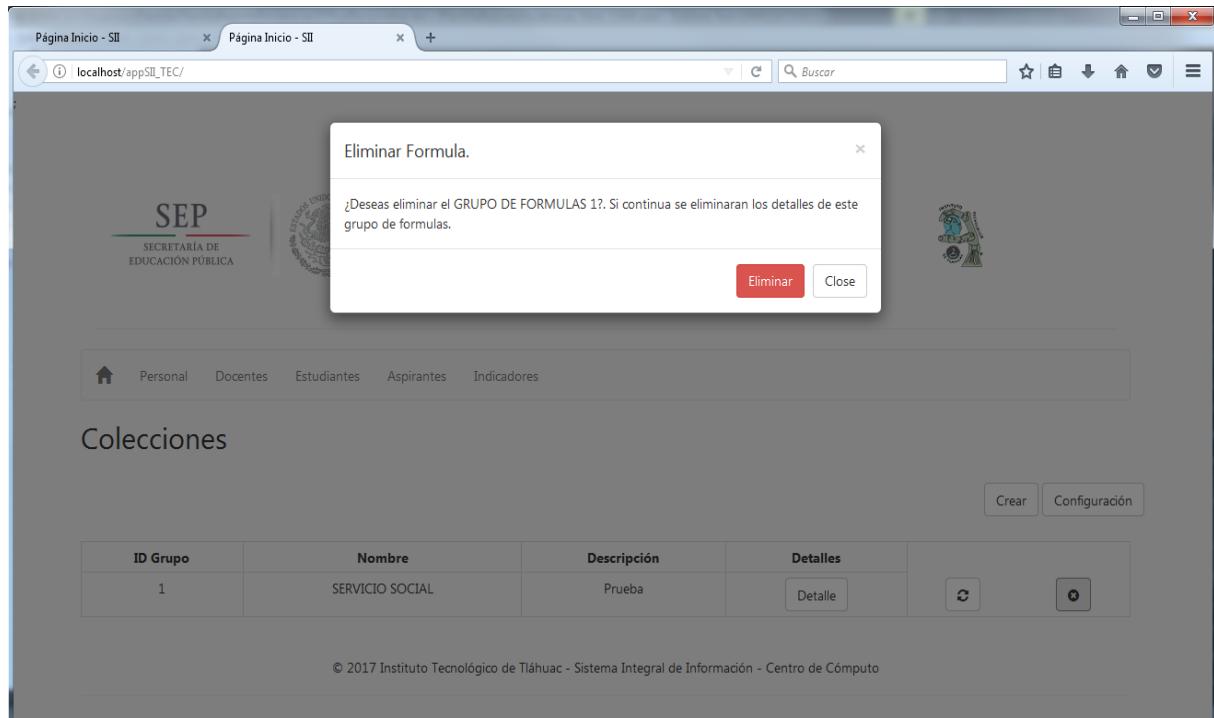


Figura 5.23: Mensaje de advertencia al eliminar colección.



Figura 5.24: Pantalla de inicio del detalle de colecciones.

5.4.4. Detalle de colecciones

Esta funcionalidad es para mostrar las fórmulas que están contenidas dentro de esta colección, para esto al dar click en el botón Detalle que se encuentra en cada fila de las colecciones. Al ejecutar esta función se mostrara una nueva pantalla como se muestra en la figura

Las acciones que se pueden realizar dentro de este módulo son:

- Agregar formulas a colección.
- Quitar fórmulas de colección.

Agregar formulas a colección

Esta funcionalidad se ejecuta al momento de presionar el botón agregar, el cual abre una ventana emergente que contiene la lista de fórmulas disponibles para ser agregadas a la colección. Estas se describen en la figura 5.25 la cual muestra cómo se pueden seleccionar múltiples fórmulas para ser agregadas.

Al dar click en Grabar, el sistema mostrara un mensaje indicando si el proceso se ejecutó de manera correcta o no como se muestra en la figura 5.26. El listado de las fórmulas que pertenecen a la colección se visualiza en la figura 5.27.

Quitar fórmulas de colección

En esta permite quitar una fórmula de la colección. Esta función se ejecuta cuando presionamos el botón Eliminar que se encuentra en cada registro. Al ejecutar dicha funcionalidad se preguntara al usuario si desea continuar como se muestra en la figura 5.28.

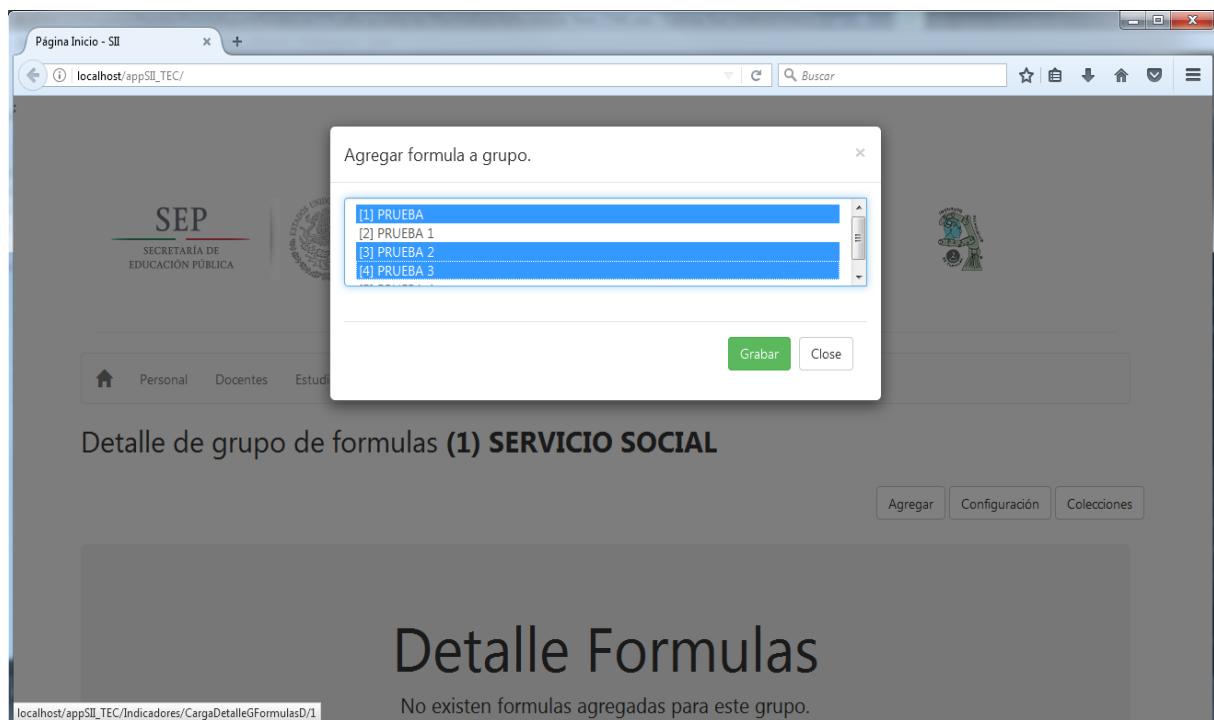


Figura 5.25: Ventana emergente para agregar formulas a colección.

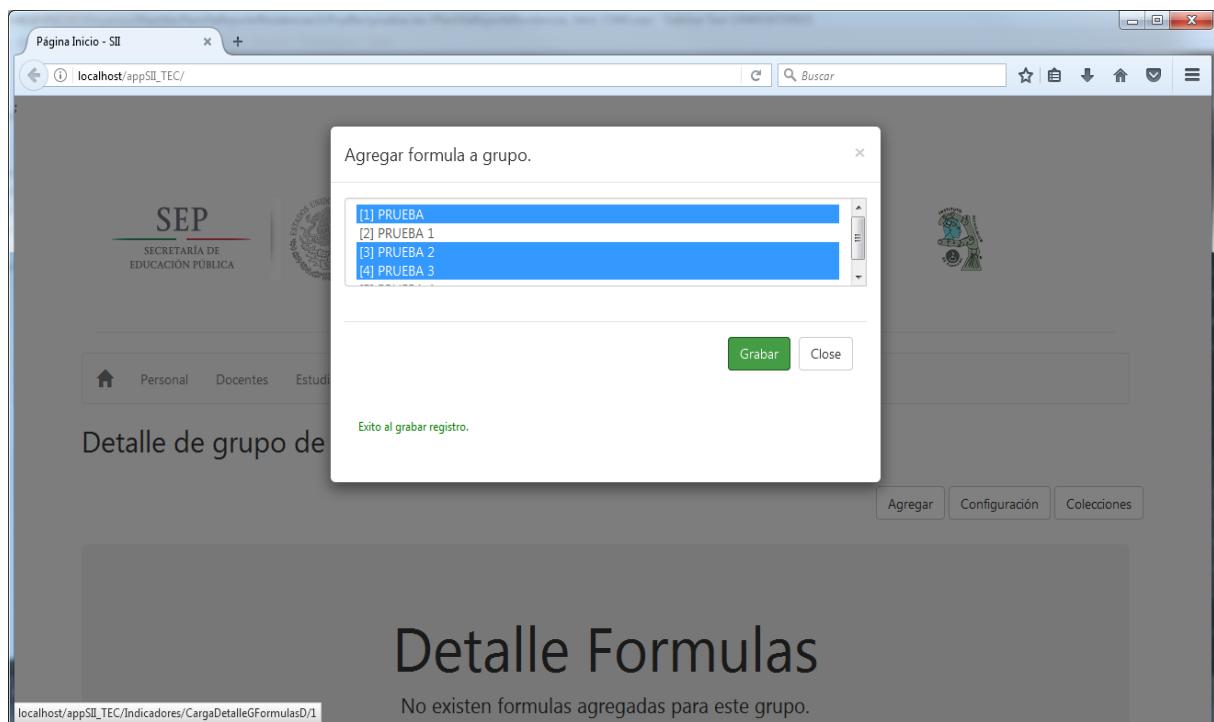


Figura 5.26: Mensaje de grabación satisfactoria de formulas en colección.

The screenshot shows a web browser window titled 'Página Inicio - SII'. The URL is 'localhost/appSII_TEC/'. The page header includes the SEP logo, the Mexican National Seal, and the text 'TECNOLÓGICO NACIONAL DE MÉXICO' and 'INSTITUTO TECNOLÓGICO DE TLÁHUAC'. A navigation menu at the top has links for 'Personal', 'Docentes', 'Estudiantes', 'Aspirantes', and 'Indicadores'. Below the menu, the title 'Detalle de grupo de formulas (1) SERVICIO SOCIAL' is displayed. To the right of the title are three buttons: 'Agregar', 'Configuración', and 'Colecciones'. A table lists four formulas with their IDs and names: ID Formula 1 (PRUEBA), ID Formula 3 (PRUEBA 2), and ID Formula 4 (PRUEBA 3). Each row has a small circular icon with a delete symbol to its right. At the bottom of the page, a copyright notice reads '© 2017 Instituto Tecnológico de Tláhuac - Sistema Integral de Información - Centro de Cómputo'.

Figura 5.27: Listado de fórmulas en colección.

The screenshot shows a modal dialog box titled 'Eliminar fórmula de grupo.' with the message '¿Deseas eliminar la fórmula 1 del GRUPO DE FORMULAS 1?'. At the bottom of the dialog are two buttons: 'Eliminar' (in red) and 'Close'. The background of the main page is dimmed. The main page title is 'Detalle de grupo de formulas (1) SERVICIO SOCIAL'. It features the same navigation and table as Figure 5.27, but with only three rows visible: ID Formula 1 (PRUEBA), ID Formula 3 (PRUEBA 2), and ID Formula 4 (PRUEBA 3). The copyright notice at the bottom is identical to Figure 5.27.

Figura 5.28: Mensaje de advertencia al eliminar una fórmula de colección.

5.5. Pruebas y resultados de módulo Proceso.

Para ingresar al módulo de proceso de indicadores, es necesario dar click en el botón Proceso que se encuentra en la pagina principal, lo cual nos mostrara la ventana ilustrada en la figura 5.10.



Figura 5.29: Pantalla principal de procesos.

El módulo de procesos cuenta con las siguientes funciones para validar:

- Alta de periodos.
- Actualización de periodos.
- Baja de periodos.
- Proceso de periodos.
- Visualización del detalle de procesos.

Estas acciones se muestran a continuación:

5.5.1. Alta de Procesos

Para crear una periodo de indicadores, se tiene que dar click en el botón crear del módulo de configuración de periodos, mostrando con esto la ventana emergente que se puede ver en la figura 5.30.

Los datos que se solicitan para la creación de un periodo de indicadores son los siguientes:

1. **Departamento:** Número de departamento al que pertenece el periodo.
2. **Número interno:** Valor identificador del periodo de indicadores.

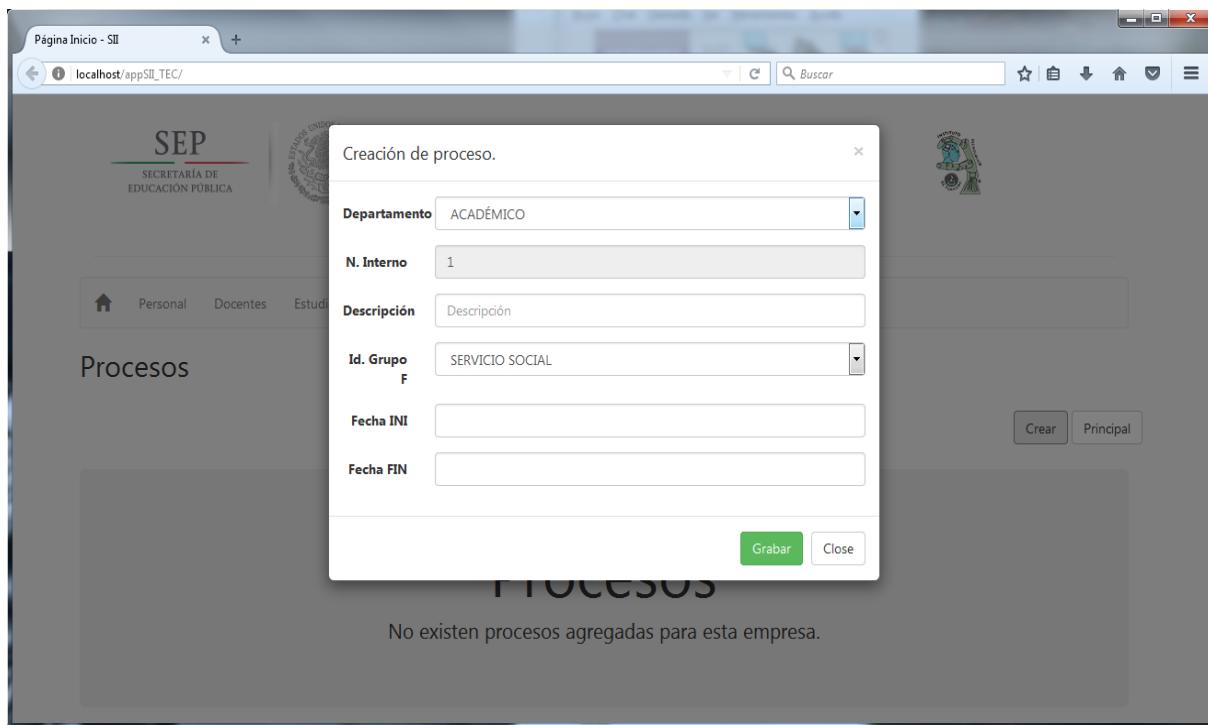


Figura 5.30: Ventana emergente para creación de periodos de indicadores.

3. **Descripción:** Campo para dar detalle de lo que se está procesando en el periodo.
4. **ID Colección de fórmulas:** Número de colección de fórmulas a procesar.
5. **Fecha inicial:** Fecha en la que el periodo inicia.
6. **Fecha final:** Fecha en la que el periodo termina.

Cuando uno de estos datos no se ingresa correctamente, se muestra un mensaje de error tal como se muestra en la figura 5.31.

Al momento de ingresar los datos correctamente, el sistema graba el registro en la base de datos y manda un mensaje indicando que el proceso se realizó correctamente, esto se muestra en la figura 5.32.

Una vez que existen procesos registrados en la base de datos, estos se muestran en una tabla, la cual tendrá acciones específicas por proceso, las acciones por proceso son actualizar, eliminar, procesar y mostrar detalle de proceso. Esta tabla de procesos se muestra en la figura 5.33.

5.5.2. Actualización de procesos

Para la actualización de procesos se muestra la misma ventana emergente que se usa para crear uno, pero en este caso se visualizan los datos del proceso que se seleccionó. En este caso se muestra la ventana emergente en la que se quiere actualizar el proceso con el nombre PERÍODO DE PRUEBA, esta ventana se muestra en la figura 5.34.

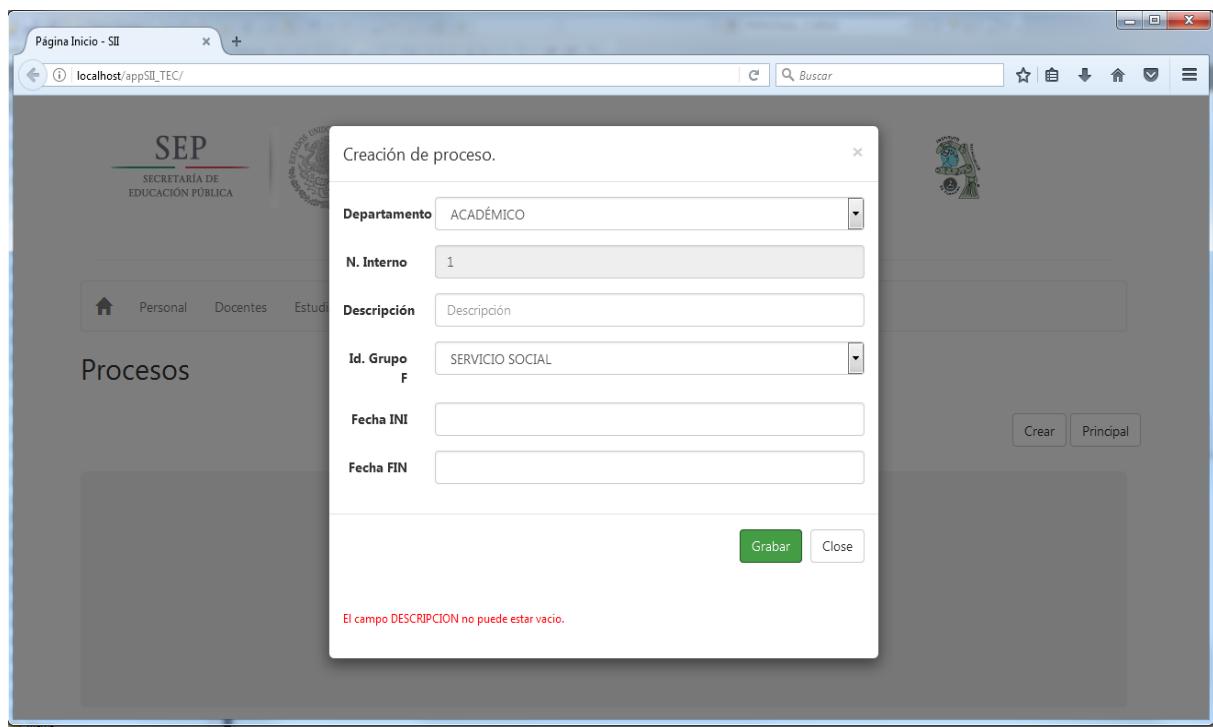


Figura 5.31: Mensaje de validación en proceso de indicadores.

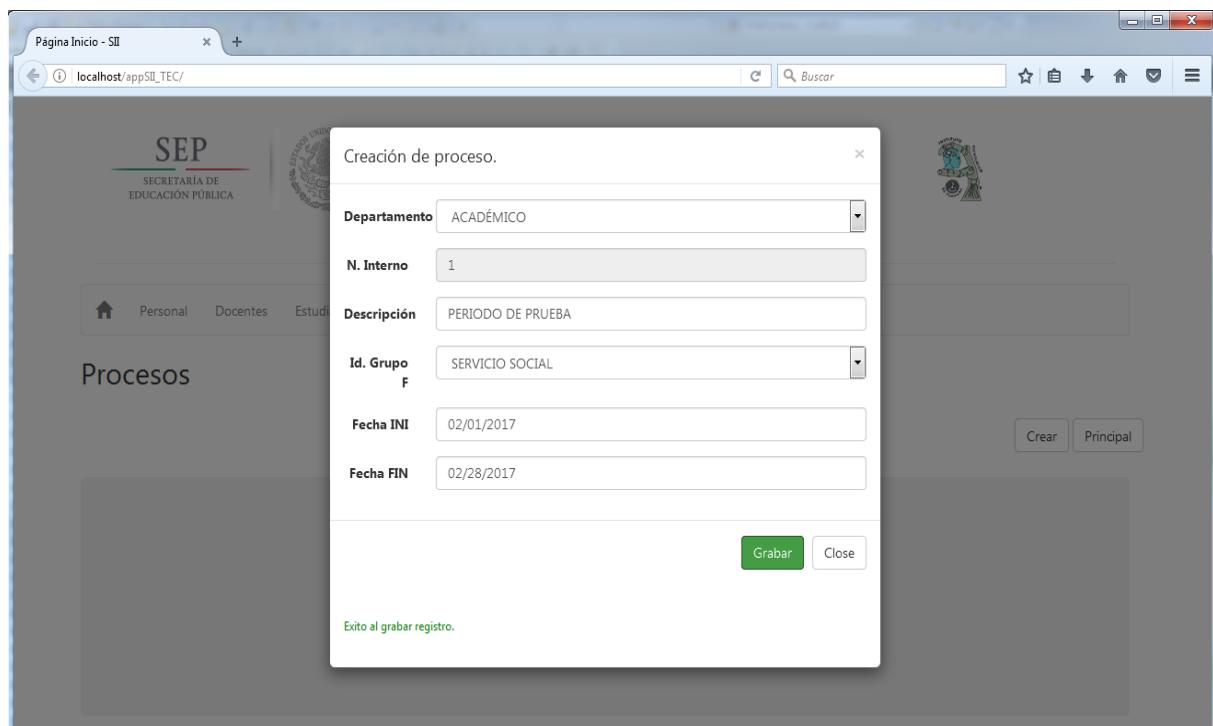


Figura 5.32: Mensaje de grabación satisfactoria de procesos de indicadores.

Departamento	Id	Descripción	G. Formulas	F. Inicial	F. Final
1	1	PERIODO DE PRUEBA	1	miércoles, 01 de febrero de 2017	martes, 28 de febrero de 2017

Figura 5.33: Listado de procesos.

Figura 5.34: Ventana emergente de actualización de periodos de indicadores.

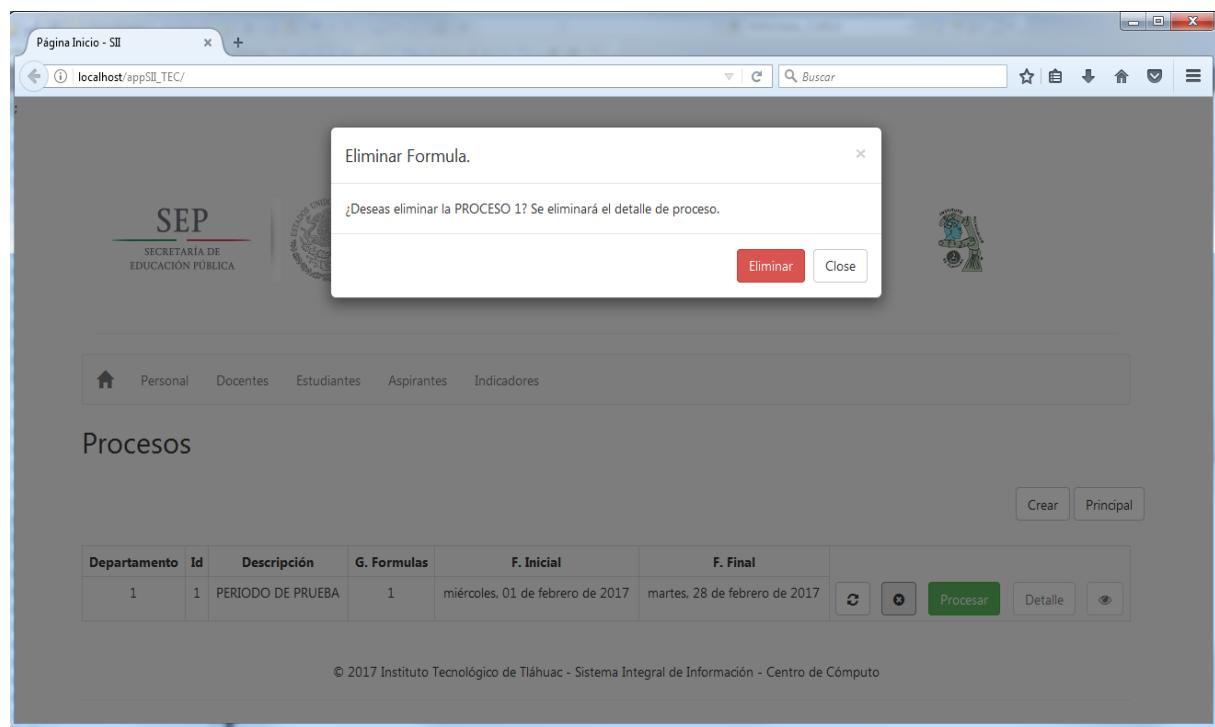


Figura 5.35: Mensaje de advertencia al eliminar un periodo de indicadores.

5.5.3. Baja de procesos

En esta prueba prueba se muestra en la figura 5.16 la ventana emergente mostrada al momento de querer eliminar un proceso, el sistema pregunta al usuario si esta seguro de eliminar este registró.

5.5.4. Proceso de periodo de indicadores

Esta funcionalidad permite ejecutar el proceso que resolverá y calculara los resultados de los indicadores. Pasando por toda la configuración de los mismos al presionar el botón de Procesar de cada uno de los periodos enlistados, el sistema resolverá cada una de las fórmulas agrupadas en la colección de fórmulas seleccionada.

Los periodos de indicadores tienen dos estatus, uno cuando fue creado y otro cuando ya fue procesado. El primero muestra un botón de color verde con la leyenda Procesar, esto indicando que el periodo no ha sido procesado anteriormente. Este se ilustra en la figura 5.36.

Cuando el periodo ya fue procesado, el botón se mostrara de color rojo con la leyenda Reprocesar, el cual al presionarlo mostrara al usuario la leyenda indicando que ya existen datos para este periodo. Este mensaje se ilustra en la figura 5.37.

5.5.5. Visualización del detalle de procesos

Al momento de procesar o reprocesar algún periodo de indicadores, se podrá visualizar el detalle con los resultados del proceso. Esto se ilustra en la figura 5.38.

Página Inicio - SII

localhost/appSII_TEC/

SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA

ESTADOS UNIDOS MEXICANOS

TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE TLÁHUAC

Procesos

Crear Principal

Departamento	Id	Descripción	G. Formulas	F. Inicial	F. Final			
1	1	PERIODO DE PRUEBA	1	miércoles, 01 de febrero de 2017	martes, 28 de febrero de 2017			

© 2017 Instituto Tecnológico de Tláhuac - Sistema Integral de Información - Centro de Cómputo

Figura 5.36: Pantalla de procesos de indicadores listos para procesar.

Página Inicio - SII

localhost/appSII_TEC/

SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA

ESTADOS UNIDOS MEXICANOS

Reproceso periodo.

¿Desea reprocesar el periodo 1? Al realizar esto los datos procesados se reemplazarán por los nuevos.

Reprocesar Cerrar

Procesos

Crear Principal

Departamento	Id	Descripción	G. Formulas	F. Inicial	F. Final			
1	1	PERIODO DE PRUEBA	1	miércoles, 01 de febrero de 2017	martes, 28 de febrero de 2017			

© 2017 Instituto Tecnológico de Tláhuac - Sistema Integral de Información - Centro de Cómputo

Figura 5.37: Alerta emergente con alerta de reproceso.

Página Inicio - SII

localhost/appSII_TEC/

SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA

ESTADOS UNIDOS MEXICANOS

TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE TLÁHUAC

Detalle de proceso 1

Configuración Proceso

Consecutivo	Formula Sistema	Formula Precompilada	Resultado
1	CONTEO("Estudiantes","", "ALU_SERV","1")	6	6
2	CONTEO("ESTUDIANTES","", "ALU_SERV","1") +20	6+20	26
3	AVERAGE(CONTEO("ESTUDIANTES","", "ALU_SERV","1"),9,20,CONTEO("ESTUDIANTES","", "ALU_SERV","0"))	AVERAGE(6,9,20,7)	10.5

© 2017 Instituto Tecnológico de Tláhuac - Sistema Integral de Información - Centro de Cómputo

Figura 5.38: Pantalla con el detalle de periodo de indicadores.

El detalle de indicadores muestra datos que nos permiten realizar un mejor análisis y seguimiento de los datos procesados, ya que muestra las etapas por las que las fórmulas fueron avanzando par a lo largo del proceso. Esto es útil cuando existe algún error y se quisiera analizar en qué parte de la fórmula se encuentra el error.

Con la visualización del resultado del proceso de un periodo de indicadores se concluyen las pruebas del sistema.

Capítulo 6

Conclusiones

Con el presente desarrollo de puede concluir que haciendo uso del Sistema Integral de Indicadores, los procesos serán más fáciles de realizar permitiendo con esto obtener resultados en tiempo real y de una manera muy simple ahorrando tiempo invertido en actividades repetitivas.

Gracias a la flexibilidad del sistema este permitirá adaptarse a cualquier módulo que se le incorpore, con esto se ahorra tiempo invertido en nuevos desarrollos para la adaptación del mismo.

Además de las facilidades de proceso que brindará el sistema, este ayudará gracias a su dinamismo a desarrollar herramientas de cálculos dinámicos más poderosas y flexibles.

Bibliografía

- [1] M. Del Carmen Gómez, *Análisis de requerimientos*. Universidad Autónoma Metropolitana, 2011.
- [2] José Luis Comesaña. (2015) http://mundosica.github.io/tutorial_hispano_jQuery/.
- [3] J. Pavon Maestras, *Aplicaciones Web/Sistemas Web*. Universidad Computense de Madrid, 2014.
- [4] J. Vallés Botella, *ASP.NET MVC 3 y 4, CONTACTO CON MVC*. UNIVERSIDAD DE ALICANTE, 2014.
- [5] J. S. y Javier Hernández, *SQL Server Vs MySQL*. Documento de internet, 2014.