

# Análisis y Diseño del Software para el Control de la Acreditación de las Actividades Complementarias

M.G. Flores-Luevanos<sup>1</sup>, E. Moreno-Núñez<sup>1</sup>, E.V.J Solorio-Vega<sup>1</sup>

□ **Resumen**— La presente investigación se centrará en el análisis y diseño del software “Sistema de Actividades Complementarias” (SAC).

Se describe de manera general en qué consisten las actividades complementarias que se realizan en los Institutos Tecnológicos de nuestro país y en específico en el Instituto Tecnológico Superior de Lerdo, el proceso para llevarlas a cabo y acreditarlas.

Se detalla el objetivo del SAC, el cual es: “Coadyuvar en la formación integral de los estudiantes del Instituto Tecnológico Superior de Lerdo, brindando un conjunto de herramientas que faciliten el control de la acreditación de las actividades complementarias”; se presentan los productos principales del análisis y diseño del software y se describe la metodología utilizada para su desarrollo. Al final se incluyen las conclusiones de este proceso.

**Temas claves**—Actividades Complementarias, análisis, diseño, software.

**Abstract** – this research will focus on the analysis and design of the "Complementary Activities System" (SAC) software. It generally describes the complementary activities that take place in the Technological Institutes of our country (México) and specifically in the Instituto Tecnológico Superior de Lerdo, the process to carry out and accredit.

The objective of the SAC, which is detailed: "To assist in the formation of students of the Superior Technological Institute of Lerdo, providing a set of tools that facilitate the control and accreditation of complementary activities"; the main products of analysis and software design are presented and described the methodology used for development. Finally the conclusions of this process are included.

**Keywords** — Complementary activities, analysis, design, software.

## I. INTRODUCCIÓN

Las actividades complementarias son todas aquellas que realiza el estudiante de los Institutos Tecnológicos, en beneficio de su formación integral con el objetivo de

complementar sus competencias profesionales. Éstas pueden ser: tutorías, actividades extraescolares, proyectos de investigación, participación en eventos académicos, productividad laboral, emprendimiento, fomento a la lectura, construcción de prototipos y desarrollo tecnológico, conservación al medio ambiente y participación en ediciones, o aquellas que defina el comité académico.

El valor curricular para el conjunto de las actividades complementarias establecidas en el plan de estudios es de 5 créditos, considerando que por cada crédito equivale a 20 horas y su cumplimiento debe ser dentro de los seis primeros semestres.

Cada una de las actividades complementarias autorizadas por el plantel, no deben de tener más de 2 créditos. En el ITSL estas actividades tienen un valor curricular de un crédito cada una.

### Fundamento Teórico

Para llevar a cabo este proyecto, se hará uso de las herramientas tecnológicas siguientes: Project, UML así mismo para el desarrollo se usarán CSS, ASP, SQL, Visual Studio: C#.

### UML

Es una herramienta para el modelamiento de software basado en los estándares UML (Unified Modeling Language) y MDA (Model Driven Architecture), actualmente es de licencia abierta GNU/GPL. Es una de las mejores alternativas gratis que hay en Internet para el modelamiento de software y probablemente una gran ayuda a la hora de programar juegos. (Schmuller, 2001).

Entre sus principales características:

Soporte completo al diseño UML mediante el uso de.

- Diagrama de casos de uso
- Diagrama de clase
- Diagrama de secuencia
- Diagrama de colaboración
- Diagrama de estados
- Diagrama de actividad.

### Microsoft Project

Project ayuda a planificar proyectos y a colaborar con otras personas fácilmente teniendo todo organizado y realizando el seguimiento de los proyectos con el único sistema de administración de proyectos diseñado para trabajar sin ningún problema con otras aplicaciones de Microsoft y servicios en la nube. (Gamboa, 2011)

<sup>1</sup> María Guadalupe Flores Luévanos ([floresluevanos@itslerdo.edu.mx](mailto:floresluevanos@itslerdo.edu.mx)), Elda Moreno Núñez ([elda\\_ivan@hotmail.com](mailto:elda_ivan@hotmail.com)), Edna Velia Josefina Solorio Vega ([edna\\_solorio@yahoo.com.mx](mailto:edna_solorio@yahoo.com.mx)), Instituto Tecnológico Superior de Lerdo, Av. Tecnológico S/N, Col. Periférico C.P. 35150 Cd. Lerdo, Durango, México.

## Metodología.

La metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto con altas posibilidades de éxito, esta sistematización indica cómo se divide un proyecto en módulos pequeños llamados etapas y las acciones que corresponden a cada una de ellas, ayuda a definir entradas y salidas para cada una y normaliza el modo en que se administrará el proyecto.

Existen dos metodologías que tienen analogía en la práctica con los paradigmas de programación: metodología estructurada y metodología orientada a objetos.

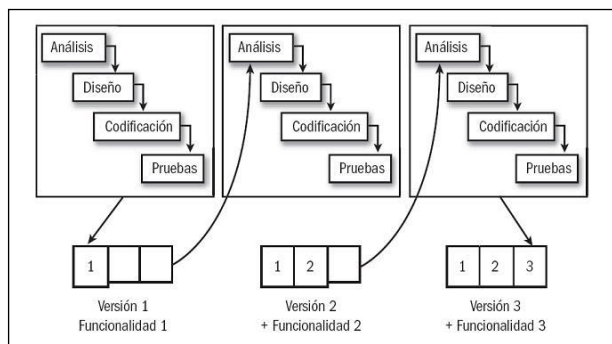
## II. PARTE TÉCNICA DEL ARTÍCULO

### A. Metodología

La metodología propuesta para este proyecto es la orientada a objetos. Ésta consiste en armar módulos basados en componentes y cada componente es independiente del otro. Esto permite que el código sea reutilizable y es más fácil de mantener pues los cambios están localizados en cada componente. Su principal diferencia con la metodología estructurada es que ésta última comprende los procesos como funciones y cada función a realizar por el sistema se descompone en pequeños módulos individuales y finalmente se unen las soluciones para crear la solución al problema.

La ISO (International Organization for Standardization) en su norma 12207 define al ciclo de vida de un software como un marco de referencia que contiene las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto software, abarcando desde la definición hasta la finalización de su uso.

Ciclo de vida propuesto: incremental.



Una forma de reducir los riesgos es construir solo una parte del sistema, reservando otros aspectos para niveles posteriores; el desarrollo incremental es el proceso de construcción siempre incrementando subconjuntos de requerimientos del sistema.

En este modelo se desarrolla el sistema para satisfacer un subconjunto de requisitos especificados y en posteriores versiones se incrementa el sistema con nuevas funcionalidades que satisfagan más requisitos.

### Características.

- ✓ Combina elementos del modelo de cascada con la filosofía interactiva de construcción de prototipos.
- ✓ Cada secuencia lineal produce un producto operacional con cada incremento de la misma forma que progresa el tiempo en el calendario.
- ✓ El primer incremento es a menudo el núcleo.
- ✓ Como un resultado de evaluación y/o utilización se desarrolla un plan para el incremento siguiente, este proceso se repite hasta llegar al producto completo.
- ✓ Este modelo es particularmente útil cuando la dotación de personal no es suficiente para una implementación completa.
- ✓ Los primeros incrementos se pueden implementar con menos recursos.
- ✓ Si es muy riesgoso desarrollar el sistema completo de una sola vez, entonces debería considerar este modelo.

### Ventajas.

- Construir un sistema pequeño es siempre menos riesgoso que construir un sistema grande.
- Al ir desarrollando parte de las funcionalidades, es más fácil determinar si los requerimientos planeados para los niveles subsiguientes son correctos.
- Si un error importante es realizado, sólo la última iteración necesita ser descartada y utilizar el incremento previo.

### Desventajas.

- Se presupone que todos los requisitos se han definido al inicio.
- Se requiere de una experiencia importante para definir los incrementos de forma de distribuir en ellos las tareas en forma proporcional.
- Si el sistema a desarrollar es de gran magnitud y se cuenta con un único grupo para construirlo se corre el riesgo que el desarrollo se prolongue demasiado en tiempo.

### B. Especificación de requisitos según el estándar de IEEE 830.

Después de la recopilación y verificación de los requisitos, éstos se detallan en un documento, el que ha sido elaborado en colaboración con las profesoras responsables del proyecto de desarrollo del software mencionado.

Esta especificación se ha estructurado basada en las directrices dadas por la última versión del estándar “IEEE Recommended Practice for Software Requirement Specifications (ANSI/IEEE) 830-1998”

*C. Análisis orientado a objetos.*

En esta fase se crearon diagramas de casos de uso, de clases, descripciones de escenarios, diagramas de proceso y de actividades, de secuencia, entre otros. A continuación se presentan los diagramas de caso de uso.

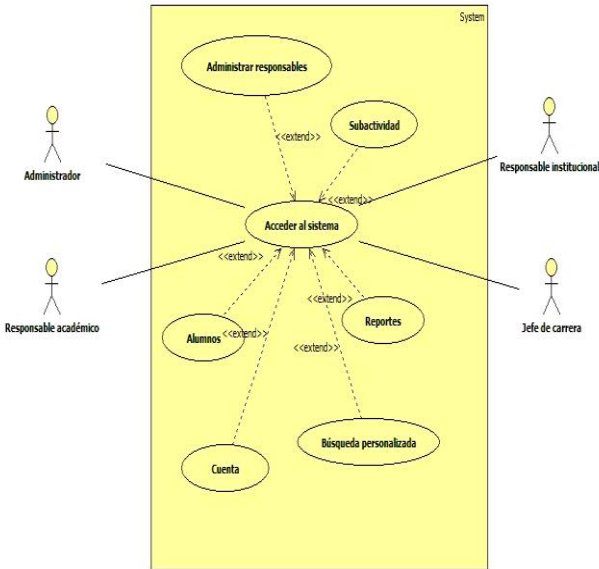


Figura 2 Casos de uso Acceder al Sistema

El sistema le solicita al usuario su número de checador y su contraseña de acceso.

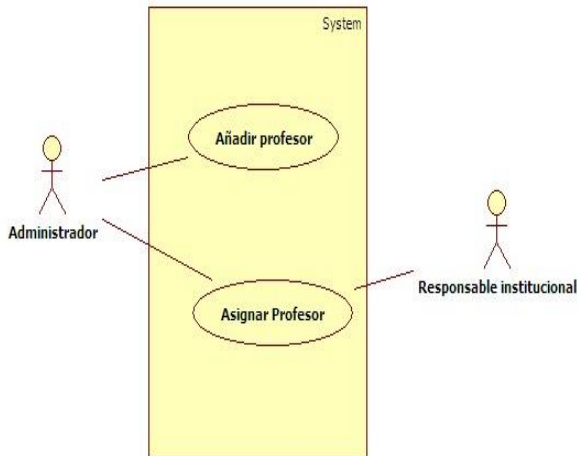


Figura 3 Casos de uso módulo asignar responsables

El sistema solicita al interesado el número de empleado del profesor responsable, el nombre de la actividad

complementaria y el nombre de la sub-actividad a las cuales se les asignarán.

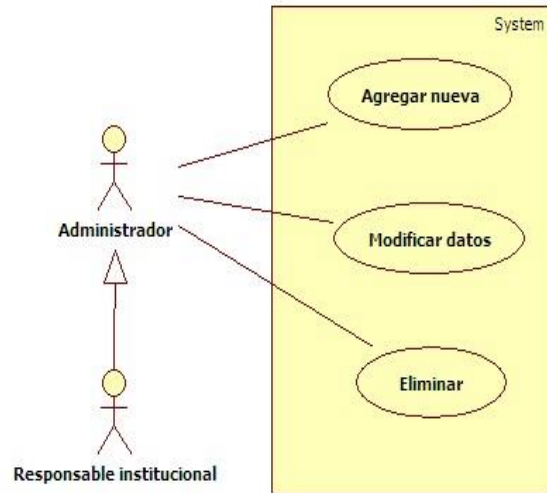


Figura 4 Casos de uso Módulo Subactividad

El sistema le solicita al interesado el nombre de la sub-actividad complementaria.

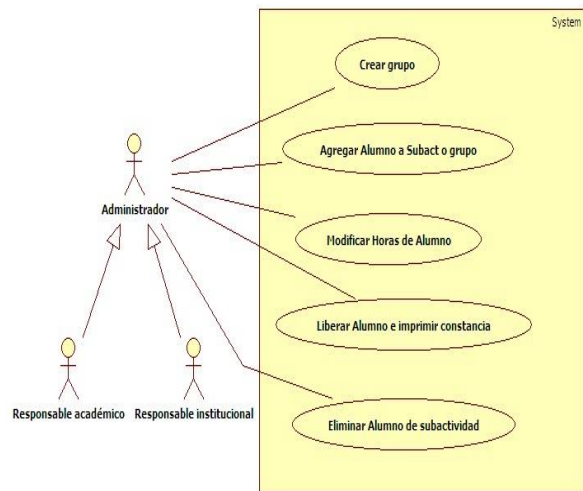


Figura 5 Casos de uso Módulo Alumnos

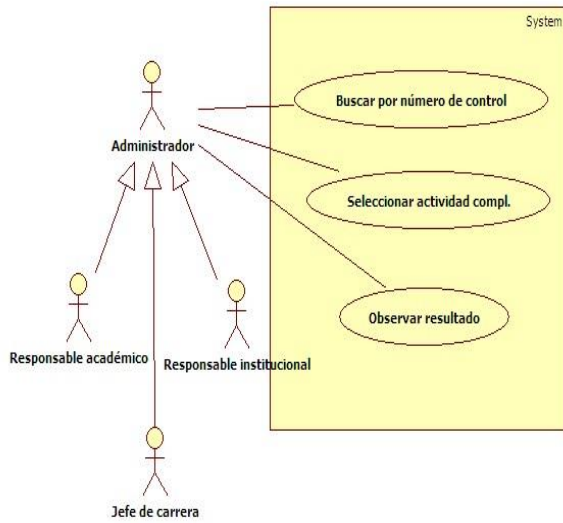


Figura 6 Casos de uso Módulo Búsqueda

El sistema le solicita al interesado el número de control del alumno y el nombre del grupo de la sub-actividad complementaria.

El sistema le solicita al responsable institucional o académico el número de control del alumno y le solicita que elija el nombre de la actividad complementaria

El sistema permite generar diferentes reportes para un mejor control de las actividades complementarias.

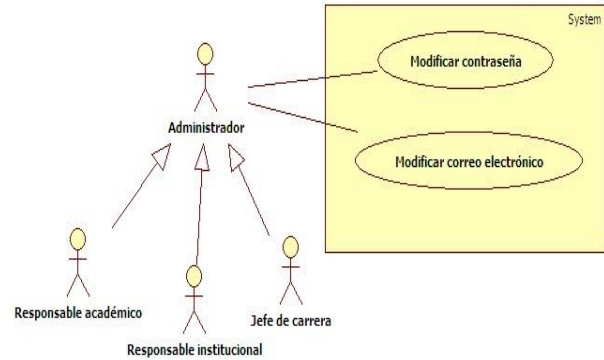


Figura 8 Casos de uso Módulo Reportes

### Modelado de clases

Mediante el Diagrama de Clases de la Figura 9 mostramos la estructura principal del Sistema.

### Diagrama de clases

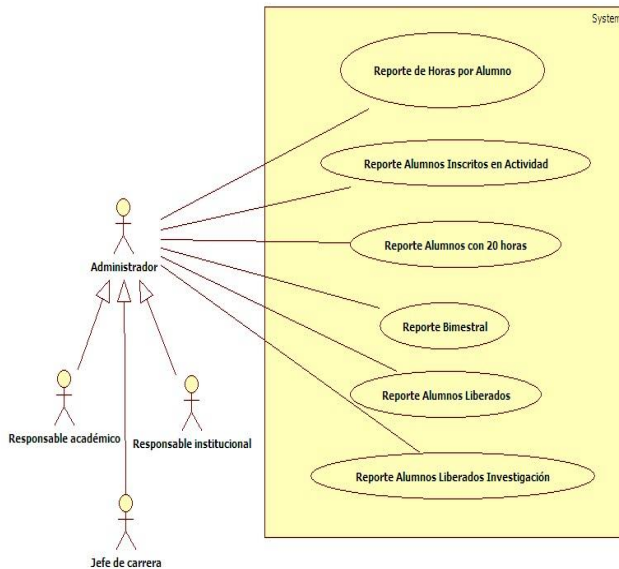


Figura 7 Casos de uso Módulo Reportes

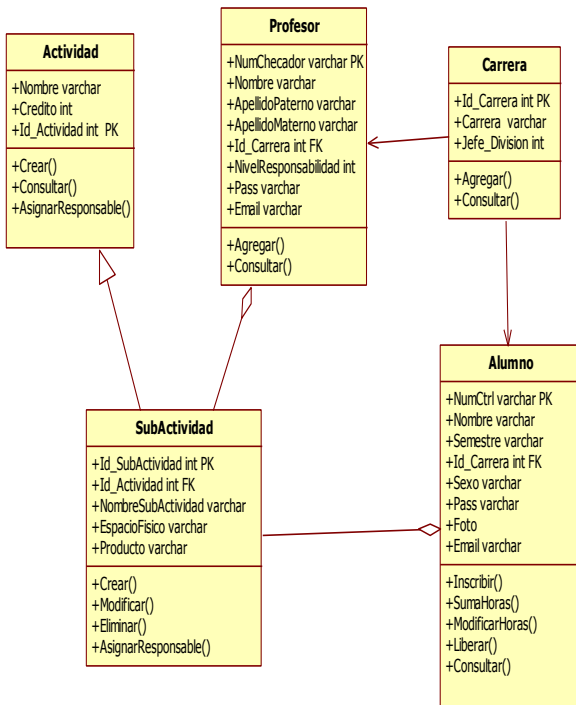


Figura 9 Diagrama de Clases del Sistema

**Modelo Relacional**

Mediante el Modelo Relacional mostrado en la figura 10 podemos visualizar la forma en que se almacenará y relacionará la información manejada por el sistema.

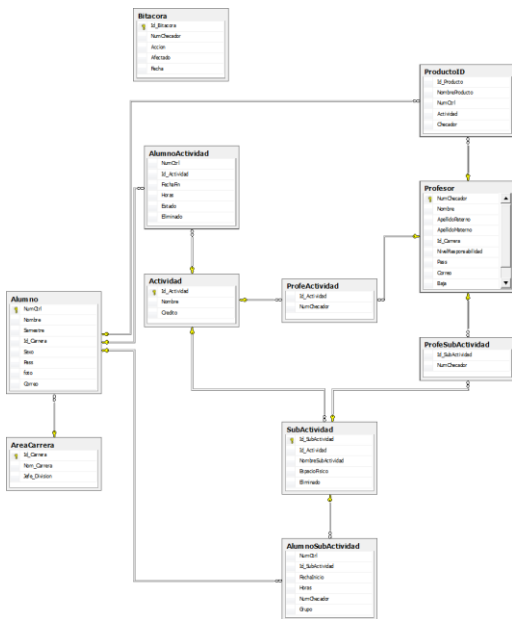


Figura 10 Esquema Relacional de la Base de Datos

Diccionario de Datos de las Tablas principales.

Mediante el Diccionario de datos, podemos enlistar las características lógicas de los datos que vamos a utilizar en el sistema, tales como nombre, descripción, contenido. En la figura 11 mostramos el diccionario de datos de las tres tablas principales del sistema a desarrollar

columna	tipo	EsNulo	EsAutonumerico	EsUnico	EsPrimaryKey	descripcion	max_length	precision
Id_Actividad	int	FALSO	VERDADERO	Verdadero	VERDADERO	NULL	4	10
Nombre	varchar	VERDADERO	FALSO	Verdadero	VERDADERO	NULL	30	0
Crédito	int	VERDADERO	FALSO	Verdadero	VERDADERO	NULL	4	10

columna	tipo	EsNulo	EsAutonumerico	EsUnico	EsPrimaryKey	descripcion	max_length	precision
NumCtrl	varchar	FALSO	FALSO	Verdadero	VERDADERO	NULL	50	0
Nombre	varchar	VERDADERO	FALSO	Verdadero	VERDADERO	NULL	70	0
Semestre	varchar	VERDADERO	FALSO	Verdadero	VERDADERO	NULL	2	0
Id_Carrera	int	VERDADERO	FALSO	Verdadero	VERDADERO	NULL	4	10
Sexo	varchar	VERDADERO	FALSO	Verdadero	VERDADERO	NULL	1	0
Pass	varchar	VERDADERO	FALSO	Verdadero	VERDADERO	NULL	30	0
Foto	varchar	VERDADERO	FALSO	Verdadero	VERDADERO	NULL	50	0
Correo	varchar	VERDADERO	FALSO	Verdadero	VERDADERO	NULL	50	0

Columna	tipo	EsNulo	EsAutonumerico	EsUnico	EsPrimaryKey	descripcion	max_length	precision
Id_SubActividad	int	FALSO	VERDADERO	Verdadero	VERDADERO	NULL	4	10
Id_Actividad	int	FALSO	FALSO	Verdadero	VERDADERO	NULL	4	10
NombreSubActividad	varchar	VERDADERO	FALSO	Verdadero	VERDADERO	NULL	50	0
EspacioFisico	varchar	VERDADERO	FALSO	Verdadero	VERDADERO	NULL	30	0
Eliminado	varchar	FALSO	FALSO	Verdadero	VERDADERO	NULL	2	0

Figura 11 Diccionario de datos de tablas principales del Sistema

(Coronel, 2011)

**Conclusiones**

El proyecto a realizar contribuirá de manera importante al control de la acreditación de las actividades complementarias realizadas por los alumnos del Instituto Tecnológico Superior de Lerdo.

Dado que se realizó un análisis detallado de los requerimientos y las necesidades que se quieren satisfacer, es posible realizar este sistema con el mínimo riesgo de cambios durante su desarrollo.

**III. AGRADECIMIENTOS**

Nuestro agradecimiento es para el Instituto Tecnológico Superior de Lerdo por su apoyo a este proyecto.

**IV. REFERENCIAS**

- [1] Coronel, C., Morris, S., & Rob, P. (2011). *Base de Datos, Diseño, Implementación y Administración*. México, DF: Cengage Learning Editores, S.A.
- [2] Gamboa, F. G. (2011). *Microsoft Project 2010: Microsoft Project 2010*. Microsoft.
- [3] Microsoft. (2010). *Microsoft Project 2010: guía interactiva de la correspondencia entre los menús y la cinta de opciones*
- [4] Pressman, R. S. (1993). *Ingeniería del Software Un Enfoque Práctico*. Madrid, España: Mc Graw Hill.
- [5] Schmuller, J. (2001.). *Aprendiendo UML en 24 horas*. Prentice Hall.

## V. BIOGRAFÍA



**María Guadalupe Flores Luévanos.** Nació en Torreón, Coahuila de Zaragoza, México, es Ingeniero en Sistemas Computacionales. Egresada del Instituto Tecnológico de la Laguna. Torreón, Coah., México (2002). Estudió la Maestría en Administración. Universidad Autónoma de Coahuila. Torreón, Coah., México. (2004).

Ella es docente del Instituto Tecnológico Superior de Lerdo, Cd. Lerdo. Durango, México. Sus áreas de interés son el Desarrollo de aplicaciones en el área de Sistemas Computacionales.



**Elda Moreno Núñez.** Nació en Gómez Palacio, Durango, México, es Ingeniero en Sistemas Computacionales. Egresada del Instituto Tecnológico de la Laguna. Torreón, Coah., México (1995). Estudió la Maestría en Administración. Universidad Autónoma de Coahuila. Torreón, Coah., México. (2004). Actualmente es docente del Instituto Tecnológico

Superior de Lerdo, Cd. Lerdo. Durango, México. Sus áreas de interés son la ingeniería de software y los sistemas geográficos.



**Edna Velia Josefina Solorio Vega.** Nació en Torreón, Coahuila de Zaragoza, México, es Ingeniero en Sistemas Computacionales. Egresada del Instituto Tecnológico de la Laguna. Torreón, Coah., México. Estudió la Maestría en Administración. Universidad Autónoma de Coahuila. Torreón, Coah., México. (2008). Ella es docente del Instituto Tecnológico Superior de Lerdo, Cd. Lerdo.

Durango, México. Su área de interés es el desarrollo de Proyectos de Software.