

## 1. Título del proyecto.

Desarrollo de estrategias de modelado y diseño que contribuyan en la calidad de la construcción de Sistemas de Software.

## 2. Líneas de investigación.

- Ingeniería de Software: Modelado y Procesos.

## 3. Responsable del proyecto, participantes y adscripción de cada uno de ellos.

Serán los siguientes profesores (adscritos al Departamento de Matemáticas Aplicadas y Sistemas):

### **Dra. María del Carmen Gómez Fuentes (Responsable)**

Su principal área de investigación es la ingeniería de software, tiene experiencia en la construcción de sistemas de software (Especificación de Requerimientos, Diseño, Implementación y Pruebas). Sus responsabilidades serán coordinar el proyecto y llevar a cabo las tareas especificadas en el punto 13 de este protocolo. La Dra. María del Carmen Gómez es integrante del Cuerpo Académico “Teoría de gráficas y Teoría de la Computación” con grado *en consolidación*.

### **Dr. Jorge Cervantes Ojeda**

Creador de los Diagramas de Transición entre Interfaces de Usuario (DTIU), la ingeniería de software es una de sus líneas de investigación. Sus responsabilidades serán colaborar con el diseño de los estudios propuestos, obtener gráficas de resultados, colaborar con la obtención de conclusiones y con la redacción de los artículos que serán producto de este proyecto de investigación. El Dr. Jorge Cervantes es integrante del Cuerpo Académico “Teoría de gráficas y Teoría de la Computación” con grado *en consolidación*.

### **Dr. Abel García Nájera**

Una de sus áreas de investigación es la ingeniería de software. Sus responsabilidades serán colaborar en el diseño, construcción y puesta en marcha del estudio sobre la diferencia entre las relaciones de asociación y agregación entre clases. También colaborará en la redacción del artículo en el que se reportarán los resultados. El Dr. Abel García es integrante del Cuerpo Académico “Optimización, Sistemas Complejos e Interfaces Cerebro Computadora” con grado *en consolidación*.

**Dr. Pedro Pablo González Pérez**

Una de sus áreas de docencia, investigación y desarrollo es la ingeniería de software. Ha sido líder de varios proyectos de desarrollo de software patrocinados, tanto por la UAM como por patrocinadores externos. Sus responsabilidades serán colaborar en el diseño y construcción de casos de estudio en el que se demuestre la utilidad de lo explicado en el punto c) de los objetivos particulares. También será el autor principal de un libro de texto como soporte al desarrollo de software a gran escala. Dirigirá un proyecto terminal y dos proyectos de servicio social.

**Dr. Guillermo Chacón Acosta**

Una de sus especialidades es la estadística. El Dr. Chacón se encargará de llevar a cabo el análisis estadístico del estudio sobre la expresividad de los Diagramas de Transición entre Interfaces de Usuario. También colaborará en la redacción y obtención de conclusiones del artículo en el que se reportarán los resultados. El Dr. Guillermo Chacón es integrante del Cuerpo Académico “Modelos Matemáticos Continuos y Aplicaciones en Física y Geometría” con grado *en formación*.

**4. Orientación (se puede seleccionar más de una opción):**

- Investigación básica ( )
  - Investigación aplicada (X)
  - Desarrollo o adaptación (X)
  - Transferencia de tecnología ( )
  - Desarrollo de tecnología (X)
  - Otros ( )
- Especificar: \_\_\_\_\_

**5. Inicio y duración.**

**5.1 Fecha de inicio.**

Mayo de 2019.

**5.2 Duración.**

2 años de duración.

## 6. Propuesta

### 6.1 Resumen

En este proyecto de investigación, proponemos desarrollar estrategias innovadoras en el área de modelado y diseño, que contribuyan en la mejora de la construcción de Sistemas de Software, tanto a nivel del proceso de desarrollo como del producto final software. Concretamente, se harán investigaciones en el área de modelado de software, procesos de desarrollo, así como en el uso y propuesta de patrones de diseño. Los avances que logren los profesores de la Licenciatura de Ingeniería en Computación, que participan en este proyecto, dentro del área, contribuyen directamente al incremento del nivel con el que se forman los alumnos. Por otra parte, la participación de los alumnos en un proyecto de investigación, les acerca al conocimiento de punta en la Ingeniería de Software y les permite insertarse en un escenario de desarrollo muy cercano al que encontrarán en el mercado laboral, una vez egresado de sus estudios.

### 6.2 Antecedentes (máxima 2 cuartillas).

El modelado de sistemas de software es una técnica encaminada a comprender y visualizar cómo será el sistema que se pretende construir. Comúnmente, el modelado de sistemas de software constituye un proceso incremental e iterativo, que parte de un nivel de máxima abstracción, y a través de diferentes niveles de refinamiento, concluye en un nivel de mínima abstracción y máximo detalle. El modelo obtenido en el nivel de máxima abstracción puede resultar de gran utilidad para el establecimiento de la comunicación con el cliente, mientras que el nivel de máximo detalle juega un papel fundamental como guía para la implementación.

UML es actualmente el lenguaje de modelado más utilizado para el desarrollo de sistemas de software [Milicév, 2007; Budgen et al., 2011] porque es muy útil para esbozar el diseño y la arquitectura de un sistema de software. Sin embargo, sus inconvenientes y defectos han sido identificados y señalados históricamente [Grossman et al., 2005; Budgen et al., 2011]. En la industria del software, es un hecho que no todos modelan sus sistemas antes de implementarlos [Gorschek et al., 2014], pero cuando este modelado se realiza con UML, el diagrama de clase es la herramienta más utilizada para modelar la vista estática de un sistema de software [Dobing, 2006; Erickson y Siau, 2007]. Por otro lado, en el contexto del modelo de detección de olores para la refactorización de software, el diagrama de clase es el modelo UML más investigado [Mumtaz et al., 2019]. Dada la importancia de los diagramas de clase, consideramos que es relevante hacer un análisis cualitativo y cuantitativo de la diferencia entre estas dos relaciones de clases y así hacer frente a la *indefinición histórica de la diferencia entre Asociación y Agregación en términos de implementación*. Aclarar si existe o

no esta diferencia es útil para los modeladores, así como para aquellos que implementan los modelos.

Como señala [Stevens, 2002], la confusión en el concepto de Asociación conduce a malentendidos al modelar los diagramas de clase y al desarrollar herramientas de generación de código. Esta confusión deja brechas en la comunicación que pueden llevar a implementaciones incorrectas de las relaciones de clase modeladas en diagramas de clase.

Una notación de modelado debería facilitar la implementación de sistemas de software. Según [Moody, 2009], una de las principales características deseables de una notación de modelado es la claridad semiótica, es decir, que debería haber una correspondencia de 1 a 1 entre construcciones semánticas y símbolos gráficos. Esto significa que no debe haber sinónimos ni símbolos con más de una interpretación semántica. Si no hay una diferencia clara entre Asociación y Agregación, entonces pueden usarse como sinónimos, y no tiene sentido usar los dos símbolos al modelar las relaciones de clase. Incluso cuando ambos conceptos están claramente diferenciados en un nivel abstracto, no es práctico crear un diseño de sistema de software utilizando ambos conceptos y luego pegarlos en un solo concepto en la implementación porque de esta manera el modelo es inconsistente con la implementación.

[Grossman et al., 2005] realizaron un estudio dedicado a investigar la adopción y el uso de UML en la comunidad de desarrollo de software. Uno de los resultados de este estudio es que existe "cierto grado de confusión entre la comunidad de usuarios que rodea a UML". Mencionan los siguientes problemas como las posibles causas que hacen de UML un estándar inmaduro: su naturaleza esquivada, su complejidad, es poco conocido y, a veces, se usa de manera inconsistente. A pesar de todos estos problemas, UML se ha convertido en el estándar de facto para el desarrollo de sistemas, por lo que es importante continuar haciendo cambios para mejorarlo.

Por otra parte, aunque UML es el lenguaje estándar para representar gráficamente los requerimientos de un sistema de software, no tiene ningún soporte de notación especial para especificar los aspectos dinámicos de la interacción usuario-sistema [Markopoulos y Marijnissen, 2000]. Por lo tanto, en UML es difícil identificar qué Interfaz de Usuario está relacionada con cada acción que involucra la interacción del usuario [Pinheiro Da Silva & Paton, 2003]. De hecho, la única notación de modelado formal especialmente diseñada para describir la interfaz de usuario en función de las transiciones entre estas interfaces es la de los Diagramas de Transición entre Interfaces de Usuario (DTIU) [Gómez y Cervantes, 2013].

Durante el desarrollo de este proyecto, pretendemos demostrar la expresividad de los DTIU como herramienta de modelado para establecer la comunicación entre clientes y desarrolladores. Concretamente, se investigará la capacidad de los DTIU para comunicar el funcionamiento de un sistema en términos de las interfaces que se presentan al usuario y las posibles acciones que éste puede llevar a cabo en cada interfaz. Además, se analizará la forma en que los DTIU se pueden conectar e integrar con los patrones de diseño, durante la construcción de sistemas de software.

En Ingeniería de Software un patrón se refiere a la forma en la que un determinado problema de análisis, diseño, arquitectura, implementación, etc. fue solucionado, y cómo reutilizar la

esencia de esta solución para resolver nuevos problemas. Un patrón encierra una experiencia, un conocimiento en la solución de problemas previos, es un resultado de la experiencia práctica, que contribuye significativamente en el desarrollo de nuevos sistemas de software, al incorporar soluciones previas que han demostrado funcionar bien. Un patrón maneja un problema de construcción de software (análisis, diseño arquitectónico, diseño detallado, implementación, etc.) recurrente que aparece en situaciones específicas durante las fases de desarrollo del proyecto de software, presentando una solución a éste. Además, un patrón documenta experiencia de construcción de software existente que ha demostrado funcionar bien, identifica y especifica abstracciones que están por encima del nivel de simples clases o instancias, o de componentes.

## **7. Objetivos**

### **7.1 Objetivo general.**

Contribuir en el área de Ingeniería de Software con métodos y técnicas innovadores en el modelado y construcción de sistemas, que fomenten tanto la calidad del proceso de desarrollo como del producto software.

### **7.2 Objetivos particulares.**

Las mejoras propuestas se especifican en los siguientes objetivos particulares.

- a) Investigar la efectividad de los Diagramas de Transición entre Interfaces de Usuario (DTIU) para comunicar el comportamiento de las interacciones usuario-sistema.
- b) Investigar el problema de la confusión que existe al diferenciar las relaciones de Asociación y Agregación en un diagrama de clases, cuando se implementan en código.
- c) Proponer una metodología de Construcción de Sistemas de Software en base a los DTIU en el diseño de alto nivel y los diagramas de secuencia en el diseño detallado.

## **8. Descripción, incluyendo hipótesis y metodología (máximo 2 cuartillas).**

### **8.1 Hipótesis**

A continuación, se presenta una hipótesis para cada uno de los objetivos particulares planteados.

- a) La notación de los DTIU puede ser bien entendida por la mayoría de las personas, con solo una breve explicación, y sin haber trabajado nunca con ellos. Incluso si tienen poca o ninguna capacidad de programación.
- b) La confusión entre la implementación de la Asociación y de la Agregación se puede resolver descartando la Agregación. De esta forma evitamos su uso innecesario en el diseño de software. Al hacer esto, se simplifican los procesos de diseño, implementación y mantenimiento.
- c) Los DTIU son de mucha utilidad para ligar la fase de requerimientos con la del diseño de alto nivel. Además, los diagramas de secuencia y los patrones de diseño pueden tener un uso novedoso que ayuda durante la fase del diseño detallado.

## 8.2 Metodología

Este proyecto consta de tres partes: a) un estudio estadístico sobre la expresividad de los Diagramas de Transición entre Interfaces de Usuario (DTIU), b) un estudio cualitativo y cuantitativo sobre la implementación de las relaciones de Asociación y Agregación entre clases y, c) casos de estudio del desarrollo de sistemas de software tomando como base los DTIU en el diseño de alto nivel y, en el diseño detallado, los diagramas de secuencia y patrones de diseño. A continuación, se detalla cada una de las partes.

- a) Los Diagramas de Transición entre Interfaces de Usuario (DTIU)<sup>1</sup> es una notación de modelado que sirve para describir las interfaces que se presentan al usuario de un sistema de software, así como las acciones del usuario que desencadenan una transición entre estas interfaces. Se planea hacer un estudio estadístico que abarque por lo menos a 300 sujetos, para demostrar que los DTIU pueden ser bien entendidos por la mayoría de las personas, incluso si tienen poca o ninguna capacitación en la programación de sistemas.
- b) La falta de una semántica bien definida para las relaciones de Asociación y Agregación en un diagrama de clases ha producido históricamente confusión al interpretar sus símbolos al momento de su implementación. Haremos un estudio cualitativo y cuantitativo sobre las diferencias entre la Agregación y la Asociación a nivel conceptual y de implementación. Para el estudio cualitativo, revisaremos la literatura sobre la brecha entre el diseño y la implementación de las relaciones de Agregación y Asociación. También estudiaremos las contradicciones que hemos encontrado en las definiciones de algunos libros de Modelado Orientado a Objetos. Por otra parte, haremos un estudio cuantitativo entre los desarrolladores de software acerca de cómo usan la Agregación y la Asociación en la práctica para concluir que tan útil es, desde el

---

<sup>1</sup> Gómez M. C., Cervantes J., (2013). *User Interface Transition Diagrams for Customer-Developer Communication Improvement in Software Development Projects*. Journal of Systems and Software Vol 86, Issue 9, pp. 2394-2410.

punto de vista del diseño de software, diferenciar entre estas dos relaciones entre clases.

- c) Demostraremos, mediante uno o varios casos de estudio, la utilidad de los DTIU durante la fase del diseño de alto nivel y de un uso nuevo de los diagramas de secuencia y los patrones de diseño durante el diseño detallado.

Para lograr cada uno de los objetivos descritos en el presente proyecto se propone la siguiente metodología:

- a) Estudio de la expresividad del DTIU
  - A.1 Diseño y construcción del estudio.
  - A.2 Aplicar el estudio a por lo menos 300 sujetos.
  - A.3 Obtener los resultados.
  - A.4 Hacer el análisis estadístico y obtener conclusiones.
  - A.5 Redactar un artículo para revista indexada (JCR).
- b) Estudio cualitativo y cuantitativo de las relaciones de Asociación y de Agregación
  - B.1 Estudio cualitativo (Revisión de la literatura en libros y artículos de investigación)
  - B.2 Diseño y construcción del estudio cuantitativo.
  - B.3 Aplicación del estudio cuantitativo.
  - B.4 Obtención de resultados y conclusiones.
  - B.5 Redactar un artículo para revista indexada.
- c) Construcción de sistemas en base a DTIU, diagramas de secuencia y patrones de diseño
  - C.1 Análisis y Especificación de Requerimientos de uno o varios casos de estudio.
  - C.2 Elaboración de los prototipos de interfaces, DTIU y base de datos.
  - C.3 Definición de la arquitectura del sistema y del diseño detallado con diagramas de secuencia y patrones de diseño.
  - C.4 Implementación y pruebas del sistema.
  - C.5 Redactar un artículo para revista indexada o congreso.
  - C.6 Concluir la redacción de un libro de texto como soporte al desarrollo de software a gran escala, que integre e ilustre el uso de las técnicas de los DTIU, los diagramas de secuencias y los patrones de ingeniería del software.
  - C.7 Formación de recursos humanos.

## **9. Formación de recursos humanos.**

A partir de la presente propuesta se espera la participación de alumnos de la licenciatura de Ingeniería en Computación, que participen en el desarrollo del tercer objetivo con:

- Al menos 2 proyectos terminales por año
- Al menos 2 servicios sociales.

## **10. Impacto esperado del proyecto.**

Con esta propuesta se pretende demostrar a nivel de investigación, desarrollo y docencia, la utilidad de los Diagramas de Transición entre Interfaces de Usuario (DTIU). Además, demostraremos mediante uno o varios casos de estudio, que la construcción de un sistema de software se facilita cuando se aplican los DTIU, los diagramas de secuencia y los patrones de diseño. Por otro lado, si logramos que sea aceptada la hipótesis del punto 8.1 b), evitaremos la confusión para distinguir la diferencia en la implementación de la relación de Asociación y la de Agregación entre clases. Hemos detectado que esta confusión surgió desde finales del siglo pasado y sigue sin aclararse.

## **11. Recursos necesarios para el proyecto**

Respecto al equipo y espacios necesarios para el desarrollo del presente proyecto, cada uno de los académicos participantes cuenta con laptop y con computadora de escritorio. Asimismo, los alumnos participantes podrán hacer uso de la Sala General de Cómputo y, en caso de requerirse, se solicitará acceso al Laboratorio de Redes y/o a la Fábrica de Software, en horarios que no interfieran con la impartición de UEA. .

A continuación, se desglosan los recursos económicos.

- *Gastos de Viaje.* - Se requerirán recursos para transporte y viáticos para la participación en algún evento nacional o internacional estrechamente relacionado con el área de ingeniería de software.
- *Publicaciones.* - Se requerirán recursos para el pago de la publicación del libro de texto y, de ser el caso, para el pago de una publicación en revista.

Se harán las gestiones para buscar financiamiento para el desarrollo del proyecto . Se atenderán a las publicaciones de convocatorias, tanto de la UAM como de patrocinadores externos, para someter el proyecto a alguna de éstas, siempre que el alcance de la convocatoria lo permita.



## 12. Calendario de actividades en períodos trimestrales.

Cronograma de actividades						
Actividad	1° año			2° año		
	Trim 1	Trim 2	Trim 3	Trim 1	Trim 2	Trim 3
A.1	X					
A.2	X	X				
A.3		X				
A.4			X			
A.5			X			
B.1				X	X	
B.2				X	X	
B.3					X	
B.4					X	X
B.5					X	X
C.1	X					
C.2	X					
C.3		X				
C.4		X	X			
C.5			X	X		
C.6	X	X	X	X	X	X
C.7	X	X	X	X	X	X

## 13. Información para el seguimiento del proyecto

### 13.1 Calendarización de productos esperados a lo largo del proyecto.

A continuación, las siglas de cada uno de los profesores que participan en este proyecto.

- María del Carmen Gómez Fuentes (MCGF)
- Jorge Cervantes Ojeda (JCO)
- Abel García Nájera (AGN)
- Pedro Pablo González Pérez (PPGP)
- Guillermo Chacón Acosta (GCA)

En cada uno de los entregables se especifica con siglas, a él o los responsables de la entrega.

Calendario de metas y entregables						
Metas y entregables	1° año			2° año		
	Trim 1	Trim 2	Trim 3	Trim 1	Trim 2	Trim 3
A.1 Presentación .ppt con la explicación de los DTIU (MCGF)	X					
A.1 Cuestionario en alguna plataforma de internet (JCO, MCGF)	X					
A.2 Obtención de voluntarios que resuelvan el cuestionario (MCGF)	X	X				
A.3 Datos recopilados y validados (MCGF)		X				
A.3 Gráficas con los resultados (JCO)		X				
A.4Análisis estadístico (GCA)			X			
A.4 Conclusiones (JCO, MCGF y GCA)			X			
A.5 Artículo para revista internacional indizada (MCGF, JCO, GCA)		X	X	X		
B.1 Revisión de la literatura (MCGF)			X	X	X	
B.2 Diseño y construcción del cuestionario en alguna plataforma en internet (MCGF, JCO, AGN)				X	X	
B.3 Obtención de voluntarios que resuelvan el cuestionario (MCGF, AGN)				X	X	
B.4 Datos recopilados y validados (MCGF)					X	X
B.4 Gráficas con los resultados (JCO)					X	X
B.4 Conclusiones (MCGF, JCO, AGN)					X	X
B.5 Artículo para revista internacional indizada (MCGF, JCO, AGN)					X	X
C.1 Especificación de Requerimientos de uno o varios sistemas de SW (MCGF, PPGP)	X	X	X	X		
C.2 Prototipos de interfaces de usuario (JCO)	X	X	X	X		
C.2 DTIU de los sistemas (JCO, MCGF)	X	X	X	X		
C.3 Documento de Diseño (JCO, MCGF, PPGP)	X	X	X	X	X	X
C.4 Implementación y pruebas de los sistemas (JCO. MCGF, PPGP, Alumnos )	X	X	X	X	X	X
C.5 Artículo para congreso o revista (MCGF, JCO, PPGP)			X	X	X	X
C.6 Libro de texto (PPGP, MCGF, JCO)	X	X	X	X	X	X
C.7 Formación de Recursos humanos (MCGF, PPGP, JCO)	X	X	X	X	X	X

### 13.2 Resultados esperados, en términos de productividad científica y formación de recursos humanos.

Al concluir el tiempo de vigencia del presente proyecto se esperan contar, al menos, con los siguientes resultados:

- 4 alumnos con proyecto terminal concluido.
- 2 alumnos de servicio social.
- Dos artículos publicados en revista internacional indexada (al menos uno en JCR).
  - Un artículo en revista indexada o Congreso de Ingeniería de Software con el trabajo en el que colaboren los alumnos.
- Un libro de texto.

## 14. Referencias

- Budgen D., Burn A. J., Brereton O. P., Kitchenham B. A., Pretorius R. (2011). Empirical evidence about the UML: a systematic literature review. *Software: Practice and Experience*, 41(4), 363-392.
- Dobing B., Parsons J. (2006). How UML is used. *Communications of the ACM*, 49(5), 109-113.
- Erickson, J., Siau, K. (2007). Can UML be simplified? Practitioner use of UML in separate domains. In *proceedings EMMSAD*, Vol. 7, pp. 87-96.
- Gómez M. C., Cervantes J., (2013). *User Interface Transition Diagrams for Customer-Developer Communication Improvement in Software Development Projects*. *Journal of Systems and Software* Vol 86, Issue 9, pp. 2394-2410.
- Gorschek T., Tempero E., Angelis L. (2014). On the use of software design models in software development practice: An empirical investigation. *Journal of Systems and Software*, 95, 176-193.
- Grossman, M., Aronson, J. E., McCarthy, R. V. (2005). Does UML make the grade? Insights from the software development community. *Information and Software Technology*, 47(6), 383-397.
- Markopoulos, P., Marijnissen, P. (2000). UML as a representation for interaction designs. In: *Proc. Australian Conf. Computer-Human Interaction*. CHISIG, pp. 240-249.
- Milicév D. (2007). On the semantics of associations and association ends in uml. *IEEE Transactions on Software Engineering*, 33(4), 238-251.
- Moody, D.L. (2009). The "Physics" of notations: towards a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on Software Engineering* 35 (6).

- Mumtaz, H., Alshayeb, M., Mahmood, S., Niazi, M. (2019). A survey on UML model smells detection techniques for software refactoring. *Journal of Software: Evolution and Process*, e2154.
- Pinheiro Da Silva, P., Paton, N. (2003). User interface modelling with UMLi. *IEEE Software* 20 (4), 62–69.
- Siau K., Loo P. P. (2006). Identifying difficulties in learning UML. *Information Systems Management*, 23(3), 43-51.
- Stevens P. (2002). On the interpretation of binary Associations in the Unified Modelling Language. *Software and Systems Modeling*, 1(1), 68-79.