



Universidad Autónoma de San Luis Potosí
Facultad de Ingeniería
Centro de Investigación y Estudios de Posgrado

Modelo para la Recomendación de Patrones de Diseño de Interacción con Base en Requerimientos Funcionales, para el Apoyo en el Diseño de Interfaces Gráficas de Usuario

T E S I S

Que para obtener el grado de:

Doctorado en Ciencias de la Computación

Presenta:

M.C. Genoveva Viridiana Silva Rodríguez

Asesor:

Dra. Sandra Edith Nava Muñoz

San Luis Potosí, S. L. P.

Enero del 2021





Universidad Autónoma de San Luis Potosí
Facultad de Ingeniería
Centro de Investigación y Estudios de Posgrado

**Model for the Recommendation of Interaction
Design Patterns based on Functional
Requirements, to support the Design of Graphical
User Interfaces**

T H E S I S

In order to obtain the:

Doctor of Computer Science

Present:

M.C. Genoveva Viridiana Silva Rodríguez

Thesis Advisor:

Dra. Sandra Edith Nava Muñoz

San Luis Potosí, S. L. P.

January 2021





UASLP
Universidad Autónoma
de San Luis Potosí



FACULTAD DE
INGENIERÍA

19 de noviembre de 2020

**M.C GENOVEVA VIRIDIANA SILVA RODRÍGUEZ
P R E S E N T E.**

En atención a su solicitud de Temario, presentada por la **Dra. Sandra Edith Nava Muñoz** Asesor de la Tesis que desarrollará Usted, con el objeto de obtener el Grado de **Doctora en Ciencias de la Computación**, me es grato comunicarle que en la Sesión del H. Consejo Técnico Consultivo celebrada el día 19 de noviembre del presente año, fue aprobado el Temario propuesto:

TEMARIO:

“Modelo para la recomendación de patrones de diseño de interacción con base en requerimientos funcionales, para el apoyo en el diseño de interfaces gráficas de usuario”

Introducción.

1. Antecedentes.
2. Patrones de diseño de interacción en la ingeniería de software.
3. Identificación de patrones de diseño de interacción: Estudio preliminar.
4. Modelo de recomendación de patrones de diseño de interacción: UIPatternM.
5. Evaluación del modelo de recomendación UIPatterM.
6. Conclusiones, contribuciones y trabajo a futuro.

Referencias.

Anexos.

“MODOS ET CUNCTARUM RERUM MENSURAS AUDEBO”

A T E N T A M E N T E

**DR. EMILIO JORGE GONZÁLEZ GALVÁN
DIRECTOR.**



UNIVERSIDAD AUTÓNOMA
DE SAN LUIS POTOSÍ
FACULTAD DE INGENIERIA
DIRECCION

www.uaslp.mx

Copia. Archivo.
*etn.

Av. Manuel Nava 8
Zona Universitaria - CP 78290
San Luis Potosí, S.L.P.
tel. (444) 826 2330 al39
fax (444) 826 2336

“1945-2020: 75 años de formación de profesionales en la Facultad de Ingeniería”

Resumen

La Ingeniería de Software es una disciplina que engloba procesos asociados al desarrollo de sistemas interactivos. La calidad percibida de un sistema interactivo está fuertemente influenciada por el diseño de interfaces gráficas de usuario (IGU), lo que puede resultar en muchos desafíos. Uno de esos desafíos es el análisis de requerimientos a nivel de diseño. El éxito del sistema de software depende principalmente de qué tan bien los diseñadores hayan comprendido los requerimientos de los usuarios y los hayan traducido en funcionalidades adecuadas. Durante el proceso de diseño del sistema interactivo, es común encontrar problemas recurrentes en las interacciones entre humanos y computadoras, para los cuales la reutilización de soluciones es altamente factible. Dichas soluciones son representadas en una forma compacta llamada Patrones de Diseño de Interacción, los cuales buscan apoyar a los diseñadores en la toma de decisiones de diseño debido a que la tarea de diseño tiende a ser subjetiva y propensa a errores. Este trabajo tiene como objetivo presentar y evaluar un modelo de recomendación de patrones de diseño de interacción con base en la clasificación de requerimientos a nivel de diseño, a través de la aplicación de algoritmos de aprendizaje automático supervisados. Para validar el modelo, se consideraron las métricas de precisión y desempeño de los algoritmos de clasificación, y la percepción de utilidad brindada por 14 diseñadores profesionales de IGU a través de cuestionarios, como el SUS y el TAM. Como resultado de la evaluación con diseñadores profesionales de IGU, se obtuvo una percepción positiva de la utilidad del modelo propuesto denominado UIPatternM. Particularmente en el caso de los diseñadores de IGU novatos, que requieren experiencia y conocimiento para comprender la similitud entre los patrones de diseño de interacción y los requerimientos de software. Adicionalmente en este trabajo se presenta un comparativo del desempeño de cuatro algoritmos de clasificación, se realizó un estudio, en el que el método de Máquina de Vector de Soporte Lineal fue la más adecuada para este problema. Por lo tanto, estos resultados podrían ser atractivos para implementar marcos que ayuden a los diseñadores de IGU en la toma de decisiones de diseño.

Abstract

Software Engineering is a discipline that encompasses processes associated with the development of interactive systems. The perceived quality of an interactive system is heavily influenced by the graphical user interface (GUI) design, resulting in many challenges. One such challenge is design-level requirements analysis. The software system's success is mostly dependent on how the designers well users' requirements have been understood and translated into appropriate functionalities. During the interactive system design process, it is common to find recurring problems in human-computer interactions, for which reusing solutions is highly feasible. Interaction design patterns seek to support designers in decision-making during interactive systems because the design task tends to be subjective and prone to errors. This work aims to present and evaluate a model of interaction design patterns recommendation based on design-level requirements classification through supervised machine learning algorithms. To validate the model, we considered the metrics of precision and performance of the classification algorithms and the perception of usefulness provided by 14 professional GUI designers through questionnaires, such as the SUS and the TAM. As a result of the evaluation with professional GUI designers, we obtained a positive perception of the proposed model called UIPatternM utility. Particularly in the case of novice GUI designers, who require experience and knowledge to understand the similarity between interaction design patterns and software requirements. Additionally, to compare the performance of four classification algorithms, a study was carried out, in which the linear support vector machine algorithm was the most suitable for this problem. Thus, these results could be attractive for implementing frameworks that help GUI designers in design decisions making.

Dedicatoria

Ante todo a Dios, por darme fortaleza y sabiduría para llevar acabo este proyecto de vida y poder permitirme concluir este trabajo de tesis.

A mi amado esposo Alberto y mi querida hija Carolina que son mi motor en cada uno de mis proyectos y lo más maravilloso en mi vida.

A mi madre Josefina, mi padre Gerardo y mi hermano Gerardo, porque no lo hubiera logrado sin su apoyo incondicional y su presencia en cada momento de mi vida.

A mi abuelita Genoveva, mi eterno agradecimiento y un beso al cielo.

Agradecimientos

Parecía muy lejano este momento al inicio de este viaje profesional y personal, sin embargo me llevó grandes aprendizajes y experiencias de por vida.

Agradezco a la Dra. Sandra Edith Nava Muñoz por su apoyo en el proceso de ingreso al doctorado y por colaborar conmigo para llevar acabo este trabajo de tesis.

Mi agradecimiento al Comité de Tesis, Dr. Héctor Gerardo Pérez González, Dr. Francisco Eduardo Martínez Pérez y Dr. César Arturo Guerra García, cuyos comentarios y consejos contribuyeron a este trabajo de tesis.

Al Dr. Luis Adrián Castro Quiroa, muchas gracias por siempre estar disponible a la distancia y brindarle el impulso a las publicaciones producto de mi trabajo de tesis.

A mis compañeros del posgrado, junto a los cuales viví numerosas pláticas, convivencias y experiencias durante los últimos años. Las deliciosas carnes asadas y partidos de básquetbol en dónde nos relajábamos después de un día de trabajo.

A mis amigos y amigas que son la familia que escogí y que han estado presente en cada momento importante de mi vida. Gracias por su comprensión debido a mi ausencia en los últimos meses.

Al Centro de Investigación y Estudios de Posgrado de la Facultad de Ingeniería de la UASLP y al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el apoyo económico y todos los recursos brindados durante mis estudios de posgrado.

Índice General

| | |
|--|-----------|
| Lista de Figuras | XIII |
| Índice de Tablas | XV |
| 1. INTRODUCCIÓN | 1 |
| 1.1. Antecedentes | 2 |
| 1.2. Planteamiento del problema | 8 |
| 1.3. Preguntas de Investigación | 9 |
| 1.4. Objetivos de la Investigación | 10 |
| 1.4.1. Objetivo General | 10 |
| 1.4.2. Objetivos Específicos | 10 |
| 1.5. Metodología de investigación | 11 |
| 1.6. Contribuciones principales de esta tesis | 13 |
| 1.7. Contenido del documento | 13 |
| 2. ANTECEDENTES. | 16 |
| 2.1. Ingeniería de Software | 16 |
| 2.1.1. Ingeniería de Requerimientos | 17 |
| 2.1.2. Técnicas de Inteligencia Artificial en Ingeniería de Software | 22 |
| 2.2. Interacción Humano-Computadora | 23 |
| 2.2.1. Diseño de Interacción | 25 |

| | |
|--|-----------|
| 2.2.2. Patrones de Diseño de Interacción | 28 |
| 2.3. Resumen del capítulo | 31 |
| 3. PATRONES DE DISEÑO DE INTERACCIÓN EN LA INGENIERÍA DE SOFTWARE | 33 |
| 3.1. Uso de Patrones de Diseño de Interacción | 34 |
| 3.1.1. Identificación y aplicación de Patrones de Diseño de Interacción | 35 |
| 3.2. Sistemas de recomendación en la ingeniería de requerimientos | 40 |
| 3.2.1. Procesamiento del Lenguaje Natural de los Requerimientos de Software | 41 |
| 3.2.2. Clasificación de los requerimientos de software | 42 |
| 3.3. Hallazgos identificados | 46 |
| 3.4. Resumen del capítulo | 47 |
| 4. IDENTIFICACIÓN DE PATRONES DE DISEÑO DE INTERAC- CIÓN | 49 |
| 4.1. Participantes | 50 |
| 4.2. Actividad | 50 |
| 4.3. Herramientas | 51 |
| 4.4. Sesiones del Caso de Estudio | 52 |
| 4.4.1. Primera Sesión | 52 |
| 4.4.2. Segunda Sesión | 53 |
| 4.4.3. Tercera sesión | 54 |
| 4.4.4. Cuarta Sesión | 57 |
| 4.5. Resultados | 57 |
| 4.6. Discusión | 61 |
| 4.7. Resumen del capítulo | 63 |

| | |
|---|-----------|
| 5. MODELO DE RECOMENDACIÓN DE PATRONES DE DISEÑO DE INTERACCIÓN: UIPatternM. | 65 |
| 5.1. Patrones de diseño de interacción en el diseño de Interfaces de Usuarios | 66 |
| 5.2. Características de los Requerimientos de Software | 67 |
| 5.3. Modelo de recomendación de Patrones de Diseño de Interacción | 68 |
| 5.3.1. Módulo editor de texto de requerimientos | 69 |
| 5.3.2. Módulo de clasificación de requerimientos | 73 |
| 5.3.3. Módulo de recomendación de patrones de diseño de interacción . | 74 |
| 5.4. Resumen del capítulo | 77 |
| | |
| 6. EVALUACIÓN DEL MODELO DE RECOMENDACIÓN UIPatternM. | 79 |
| 6.1. Evaluación de desempeño de los Algoritmos de Clasificación | 80 |
| 6.1.1. Métricas de Rendimiento | 80 |
| 6.1.2. Conjuntos de Datos | 81 |
| 6.1.3. Extracción de características | 83 |
| 6.1.4. Entrenamiento y prueba de algoritmos | 83 |
| 6.1.5. Resultados | 85 |
| 6.2. Evaluación de utilidad del Modelo de Recomendación de Patrones de Diseño de Interacción | 88 |
| 6.2.1. Objetivos del estudio | 88 |
| 6.2.2. Hipótesis | 89 |
| 6.2.3. Diseño del estudio | 90 |
| 6.2.4. Participantes | 90 |
| 6.2.5. Instrumentos | 92 |
| 6.2.6. Procedimiento | 93 |
| 6.2.7. Resultados | 94 |

| | | |
|-----------|---|------------|
| 6.2.7.1. | Resultados sobre las percepciones de uso del modelo UI- PatternM por diseñadores profesionales | 94 |
| 6.2.7.2. | Selección de patrones de diseño de interacción por di- señadores profesionales | 95 |
| 6.2.7.3. | Percepción de los patrones de diseño de interacción re- comendados | 96 |
| 6.2.7.4. | Percepción de utilidad y facilidad de uso | 99 |
| 6.3. | Discusión | 101 |
| 6.4. | Resumen del capítulo | 105 |
| 7. | CONCLUSIONES, CONTRIBUCIONES Y TRABAJO A FUTURO. | 107 |
| 7.1. | Conclusiones | 108 |
| 7.2. | Contribuciones | 111 |
| 7.3. | Trabajo futuro | 115 |
| | Referencias | 117 |
| | Apéndice A | 130 |
| | A. Cuestionario Selección de Patrones de Diseño de Interacción | 130 |
| | Apéndice B | 132 |
| | B. Cuestionario Evaluación de Recomendación de Patrones de Diseño de Interacción | 132 |
| | Apéndice C | 134 |
| | C. Cuestionario SUS (System Usability Scale, por sus siglas en inglés) | 134 |
| | Apéndice D | 139 |

D. Cuestionario TAM (Technology Acceptance Model, por sus siglas en inglés)

139

Lista de Figuras

| | |
|--|----|
| 1.1. Áreas involucradas en el desarrollo de sistemas interactivos [1]. | 4 |
| 1.2. Cinco capas para el diseño y desarrollo web, de acuerdo con James Garrett [2] | 6 |
| 1.3. Estructura básica de un patrón de diseño de acuerdo a Alexander [3]. | 7 |
| 1.4. Metodología de Investigación. | 11 |
| 2.1. Tipos de requerimientos [4] | 19 |
| 2.2. Usuarios del documento de especificaciones de requerimientos de software (SRS) [5]. | 20 |
| 2.3. Pasos del Procesamiento del Language Natural (PLN) [6] | 23 |
| 2.4. Esquema de Interacción Humano-Computadora (IHC) [7]) | 25 |
| 2.5. Comparación de Paradigmas de Interacción [8] | 26 |
| 2.6. Dimensiones de reglas de diseño de acuerdo a Dix et al. [9] | 28 |
| 4.1. Contenido del formato SRS establecido por el IEEE [10] | 52 |
| 4.2. Generación de documento SRS por programadores Junior | 53 |
| 4.3. Aplicación de patrones de diseño de interacción en el diseño de prototipos por programadores Junior | 55 |
| 4.4. Entrevista estructurada a programadores Junior | 58 |
| 5.1. Operación del <i>Modelo de recomendación de patrones de diseño de interacción UIPatternM</i> | 70 |

| | |
|--|----|
| 5.2. Modelo de recomendación de patrón de diseño de interacción basado en requerimientos a nivel de diseño UIPatternM. | 71 |
| 5.3. Arquitectura de canalización de Diamantopoulos [11] para analizar los requerimientos de software en el módulo <i>Editor de Requerimientos Textuales</i> | 72 |
| 5.4. Procesamiento de un requerimiento textual con un enfoque de tarea de análisis semántico [11] | 73 |
| 5.5. Arquitectura del módulo de Clasificación de requerimientos. | 76 |
| 5.6. Arquitectura del módulo de recomendación de patrones de diseño de interacción. | 77 |
| 6.1. Evaluación con Diseñadores de IGUs profesionales | 95 |
| 6.2. Porcentaje del grupo de participantes (diseñadores <i>junior</i> y <i>senior</i>) que seleccionaron patrones de diseño de interacción similares a los del recomendador UIPatternM. | 99 |

Índice de Tablas

| | |
|--|----|
| 2.1. Elementos de información de un patrón propuesto por Tidwell [12] . . . | 30 |
| 3.1. Esquema XML de X-BSPDL | 38 |
| 3.2. Propuestas relacionadas con los requerimientos de software y los patrones de diseño de interacción reportados en la literatura (A = Análisis, D = Diseño, I = Fases de implementación). | 45 |
| 4.1. Distribución de los participantes en los equipos de trabajo. | 50 |
| 4.2. Preguntas asociadas a cada una de las reglas heurísticas de Nielsen [13] | 56 |
| 4.3. Escala de severidad atribuida en la evaluación de usabilidad, con base en las diez reglas Heurística Nielsen [14]. | 57 |
| 4.4. Resultados de la evaluación de las IGUs diseñadas por cada equipo con base en las heurísticas de Nielsen | 59 |
| 5.1. Un conjunto de muestra de requerimientos de software que se encuentran en el conjunto de datos corpus PROMISE [15]. La columna Clase puede ser: Availability (Disponibilidad) = A, Look and feel (Apariencia) = LF, Performance (Desempeño) = PE, Usability (Usabilidad) = US, y Portability (Portabilidad) = PO. | 75 |
| 6.1. Un conjunto de muestra de patrones de diseño de interacción que se encuentran en la colección Toxboe [16]. | 82 |

| | |
|---|-----|
| 6.2. El conjunto de unigramas y bigramas obtenidos por clase. | 84 |
| 6.3. Rendimiento global de los algoritmos de clasificación. | 85 |
| 6.4. Rendimiento del algoritmo Máquina de vectores de soporte lineal. . . . | 86 |
| 6.5. Matriz de confusión del algoritmo Máquina de vectores de soporte lineal en la Clasificación de requerimientos. | 87 |
| 6.6. Información de los participantes en la evaluación del modelo UIPatternM. | 91 |
| 6.7. Conjunto de requerimientos funcionales a nivel de diseño para identifi- cación de patrones de diseño de interacción, por parte de los diseñadores de IGUs profesionales. | 97 |
| 6.8. Predicción emitida por el recomendador de patrones de diseño de inter- acción a 10 requerimientos funcionales a nivel de diseño. | 98 |
| 6.9. Análisis estadístico de la evaluación de los resultados del recomendador de patrones de diseño de interacción por parte de diseñadores profesionales. | 100 |
| 6.10. Resultados de aplicación del cuestionario TAM. | 101 |

Lista de Acrónimos

IGUs Interfaces Gráficas de Usuario

IHC Interacción Humano-Computadora

IS Ingeniería de Software

IR Ingeniería de Requerimientos

IA Inteligencia Artificial

PLN Procesamiento del Lenguaje Natural

RF Requerimientos Funcionales

RNF Requerimientos No-Funcionales

DI Diseño de Interacción

SR Sistemas de Recomendación

SRS-Software Requirement Specification Especificaciones de Requerimientos de Software

XML-Extensible Markup Language Lenguaje Extensible de Etiquetado

MBUID-Model-Based User Interface Development Desarrollo de Interfaces de Usuario basados en Modelos

UIPatternM - Model of User Interface Pattern Modelo de Patrón de Interfaz de Usuario

TF-IDF - Term frequency-Inverse document frequency Término de Frecuencia Inversa del Documento

SUS-System Usability Scale Escala de Usabilidad del Sistema

TAM-Technology Acceptance Model Modelo de Aceptación de la Tecnología

Capítulo 1

INTRODUCCIÓN

En el diseño de sistemas interactivos, es necesario tener en cuenta las consideraciones humanas, los contextos de uso, las arquitecturas de los sistemas interactivos y las Interfaces Gráficas de Usuario (IGUs) a partir de la definición de requerimientos. Para poder interpretar las necesidades y predecir el comportamiento del usuario, los diseñadores de IGUs realizan interpretaciones personales de los casos de uso, las cuales se refieren a su propia experiencia y conocimiento. Por lo tanto, los errores de diseño surgen de las diferencias que existen entre los diseñadores y el uso real de los usuarios. Sin embargo, es común encontrar problemas recurrentes en las interacciones humano-computadora, por lo que la reutilización de soluciones es viable. En la actualidad, una de las herramientas que se ha estado usando son los patrones de diseño. Estos tienen como propósito el apoyar a los diseñadores en la toma de decisiones en el proceso de diseño de sistemas interactivos, debido a que la toma de decisiones puede ser subjetiva y propensa a errores.

Esta tesis aborda de manera particular el uso de patrones de diseño de interacción en el diseño de IGUs. Con esto se propone que el diseñador de IGUs cuente con las especificaciones de los requerimientos a nivel de diseño. Con base en dicha información, el diseñador podrá identificar los patrones de diseño de interacción apropiados para

aplicar al diseño de IGUs.

En este capítulo se describen antecedentes y planteamiento del problema, luego se presentan las preguntas de investigación que guiaron el trabajo de tesis, para enseguida definir los objetivos a alcanzar. Posteriormente, se detalla la metodología de trabajo utilizada durante el desarrollo del mismo, las principales contribuciones y se finaliza con la descripción del contenido del resto del documento.

1.1. Antecedentes

En los últimos años se observa el incremento en el uso de sistemas interactivos, tales como en la telefonía móvil, dispositivos portátiles, sistemas de información comercial, sistemas de información pública, médica o de ingeniería, etc. En el área de la Interacción Humano-Computadora (IHC), estas adopciones tecnológicas ponen especial atención a los aspectos cognitivos que intervienen en dichas interacciones, de manera que atiendan aquellas características que hacen que un sistema interactivo sea fácil de usar e intuitivo al momento de aprender a utilizarlo. A lo que los investigadores han llamado usabilidad [17,18], concepto que hace referencia a la facilidad de uso de un sistema interactivo y para lo cual, los diseñadores deben basarse siempre en la información obtenida en estudios centrados en los usuarios. Así como la experiencia de usuario, término acuñado por Donald Norman y el cual hace referencia al conjunto de factores y elementos que se ven involucrados en la experiencia del usuario al interactuar con un sistema. Además de otros aspectos como los requerimientos de diseño, que en conjunto con la usabilidad y experiencia de usuario, son fundamentales para brindar experiencias significativas y relevantes a los usuarios de sistemas interactivos, a través de los cuales sea posible realizar sus tareas con eficiencia, efectividad y satisfacción en diversos contextos, ambientes y actividades.

Con base en lo anterior, en el diseño de sistemas interactivos se sigue un proceso

de desarrollo donde participa, individualmente o de manera conjunta, una variedad de áreas [1] (ver Figura 1.1). Por su parte, la Ingeniería de Software (IS) es una colección sistemática de prácticas y técnicas de desarrollo de software [19]. Un proceso de IS tiene como fase inicial la Ingeniería de Requerimientos (IR), la cual va desde la recopilación de requerimientos hasta la verificación de su implementación, es decir, se verifica que cada una de las funcionalidades establecidas sea conforme a los requerimientos establecidos formalmente de manera textual y los cuales contienen información detallada de las necesidades de los usuarios. Mientras que IHC se centra en el estudio del fenómeno de interacción entre los usuarios y los sistemas interactivos, a través del conocimiento de usuarios, tareas y contexto de uso de los sistemas interactivos. De manera que los diseñadores sean capaces de comprender y crear software u otra tecnología que la gente querrá usar, podrá usar y resultará efectiva de utilizar. Es por ello, que en IHC se utiliza la usabilidad como atributo de calidad para evaluar la facilidad de uso de las interfaces de usuario. Así mismo, una área que surge de IHC es el diseño de productos de software, llamada diseño de interacción. Dicha área busca que los productos resulten útiles, además de tener un diseño que facilite su uso, así como generar una experiencia de usuario agradable, divertida y satisfactoria cuando los sistemas interactivos son usados por los usuarios objetivo.

Cuando las IGUs están bien diseñadas, el usuario será capaz de encontrar la respuesta a su pregunta cuando interactúe con ellas. Si éste no es el caso, la interacción con las IGUs puede ser frustrante, ya que el usuario generalmente se culpa a sí mismo por no saber cómo usar los componentes gráficos incluidos en las mismas [17,20,21]. Por lo tanto, es necesario tomar en cuenta que los sistemas interactivos son utilizados por usuarios con diferentes niveles de conocimiento (desde principiantes hasta expertos). Estas consideraciones humanas, contexto de uso, arquitecturas de los sistemas y las IGUs, deben abordarse en el proceso de diseño y desarrollo. Por lo tanto, el diseño de interacción no sólo se trata de diseñar productos y experiencias de usuario, sino también

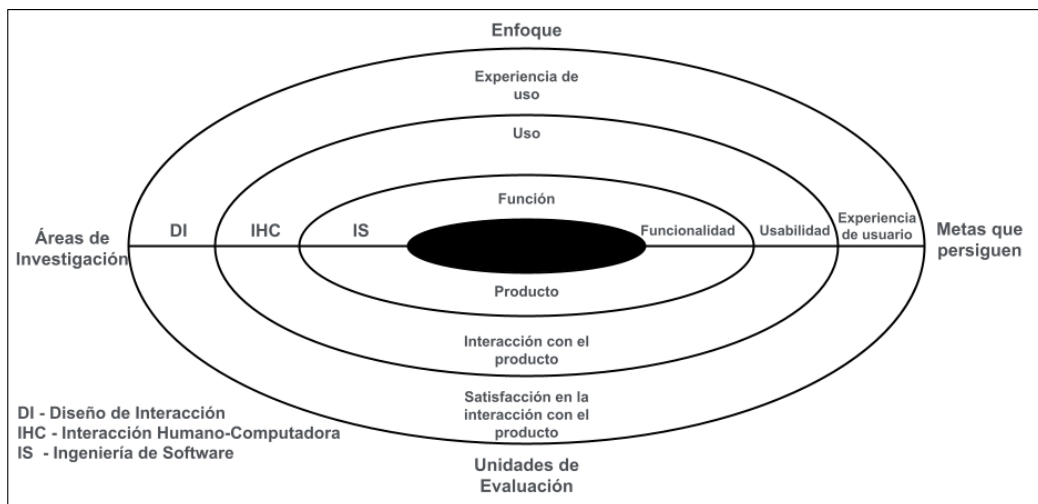


Figura 1.1: Áreas involucradas en el desarrollo de sistemas interactivos [1].

de entender el producto vinculado con su uso, su contexto, las necesidades y motivaciones del usuario final. Si la concepción de un producto no parte de las necesidades reales e intereses de sus usuarios, poco podrán aportar las siguientes etapas de desarrollo a la aceptación del producto por parte del usuario final. Estas especificaciones de diseño, son parte de los requerimientos que se definen previo al diseño de sistemas interactivos.

En este sentido, Jesse James Garrett [2] conceptualiza el diseño y desarrollo de un sitio web en términos de cinco capas: estrategia, alcance, estructura, esqueleto y superficie (ver Figura 1.2). En donde las capas superiores se construyen como resultado del entendimiento en las capas inferiores. Para comenzar cualquier experiencia es necesario iniciar con un claro entendimiento de las necesidades que el sistema debe resolver para el usuario, además del objetivo de negocio que el sistema debe cumplir, lo cual se establece en la capa de **estrategia**. Mientras que en la capa de **alcance**, se definen los requerimientos, los cuales son especificados como funcionales y de contenido. Los requerimientos funcionales son las tareas que el sistema debe cumplir para poder atender las necesidades del usuario y alcanzar los objetivos definidos por parte del negocio. Mientras que los requerimientos de contenido identifican el contenido multimedia que

se necesitará para el cumplimiento de las necesidades y objetivos. Posteriormente, como parte de la capa de **estructura** se deben de establecer los sistemas de navegación, jerarquía y búsqueda que facilitarán que el usuario interactúe con el contenido. Por lo tanto, esta capa se divide en dos: Diseño de interacción y Arquitectura de información. Diseño de interacción hace referencia a cómo los usuarios van a interactuar con el contenido definido en la etapa anterior. Por ejemplo, se define si habrá botones, barras de desplazamiento, contadores o algún otro control gráfico a nivel funcional. En cuanto a la arquitectura de información, ésta hace referencia a la estructura que tendrá el contenido dentro del sistema, cuál es su relación con otros contenidos y cómo el usuario llegará a él. Con base en lo anterior, será posible hacer los primeros bocetos del sistema web para visualizar los diferentes contenidos, lo que compone la capa de **esqueleto**. Una vez que se visualicen interactuando unos con otros, se podrá asegurar que la estructura que se propone es funcional, clara y cumple con los objetivos del usuario y del negocio. Para finalmente pasar a la capa de **superficie** que contiene el diseño visual, en donde se asegurará que los componentes interactivos y estáticos se distingan, que el texto sea legible y que el sistema sea congruente a nivel experiencia.

Por lo anterior, para proporcionar soluciones adecuadas es necesario poner especial énfasis en obtener, especificar y documentar los requerimientos. Los cuales se basan en aspectos normativos, sociales y técnicos, que posteriormente se definen como requerimientos funcionales para el desarrollo del sistema. Dentro del proceso de diseño y desarrollo de sistemas interactivos, es común encontrar algunos problemas que se generan de forma recurrente, y para los cuales la reutilización de soluciones de diseño resultan viables [22]. Dichas soluciones capturan el conocimiento y experiencia de diseño de sistemas interactivos y son representadas en una forma compacta llamada Patrones de Diseño de Interacción. El término patrón fue acuñado por Alexander [3], el cual describe un problema y una solución dentro de un contexto (ver Figura 1.3), a través de los cuales la experiencia y el conocimiento empírico de un desarrollador son reflejados. Por

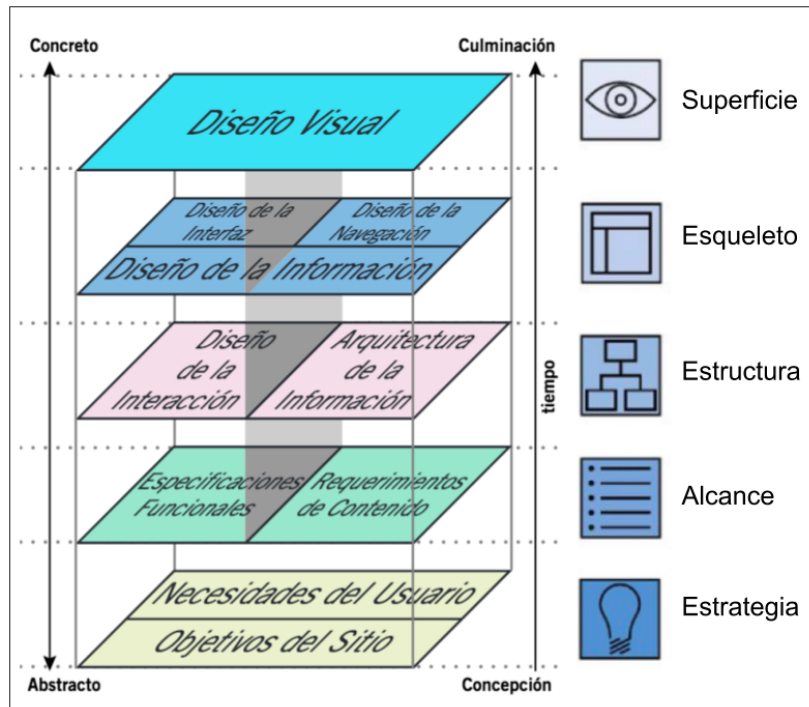


Figura 1.2: Cinco capas para el diseño y desarrollo web, de acuerdo con James Garrett [2]

su parte, Riehle y Zullighoven [23] definieron un patrón de diseño como una construcción de un software específico para manejar un problema recurrente. Los patrones de diseño de interacción capturan la esencia de una solución general de una manera que permite que otras personas lo diseñen, evalúen y construyan con mayor facilidad y éxito [24]. Lo anterior, a través de incorporar aspectos centrados en el usuario en el proceso de diseño de sistemas interactivos. De esta manera, otro aspecto importante de estos patrones es que también se utilizan para facilitar la comunicación entre personas involucradas en el desarrollo de sistemas interactivos de diferentes áreas, por ejemplo, ingenieros de software, diseñadores de interacción, expertos en usabilidad, usuarios finales, entre otros. Por lo tanto, los patrones centrados en el diseño de interacción idealmente deben de ser legibles y comprensibles por cualquier profesional interesado.

Con base en la definición de patrón de diseño antes mencionada, investigadores como Borchers [25] y Welie [26] se dieron a la tarea de recopilar sus patrones de diseño



Figura 1.3: Estructura básica de un patrón de diseño de acuerdo a Alexander [3].

en colecciones, las cuales son en su mayoría de manera descriptiva. Los patrones de diseño de interacción son problemas recurrentes que se presentan durante el uso real con sistemas interactivos, por parte de los usuarios. De manera que se definan soluciones a dichos problemas, con base en lo que funciona bien o se considera exitoso durante las interacciones de los usuarios con los sistemas interactivos [27, 28]. Conforme con estas definiciones de patrones de diseño de interacción, es posible la identificación de los problemas a los cuales se da una solución, con base en un contexto de uso y un tipo de usuario en específico. Sin embargo, también varios problemas han sido identificados, tales como la falta de un formato estándar y un principio organizador [25]. Así mismo, otro de los problemas identificados es la falta de experiencia en la identificación del patrón de diseño que mejor se adapte a las necesidades del usuario [29, 30]. Actualmente existe una amplia variedad de colecciones de patrones de diseño de interacción, lo que dificulta que los diseñadores puedan identificar cuáles patrones de diseño de inter-

acción aplicar en relación con sus necesidades. Este tipo de colecciones deben de tener la capacidad de servir como un medio para intercambiar ideas entre diferentes disciplinas y partes interesadas, haciendo posible que los ingenieros, diseñadores, evaluadores, comercializadores, financiadores e incluso los usuarios, se comuniquen de manera más efectiva [27, 31].

1.2. Planteamiento del problema

El desarrollo de sistemas interactivos es una actividad intelectual compleja que consiste en generar aplicaciones a partir de una necesidad planteada o identificada por un cliente, y que en principio debe garantizar que la aplicación desarrollada satisface dicha necesidad. Así mismo, en los requerimientos se establece la exigencia de llegar a soluciones con la calidad a través de factores como la usabilidad y experiencia de usuario. Por ello, los diseñadores deben reutilizar soluciones dadas por los profesionales a problemas similares que se han presentado en proyectos anteriores, para que éstos no tengan que pensar en cómo resolver problemas de diseño que ya han sido resueltos.

Los patrones de diseño de interacción deben reforzar los principios de usabilidad, fomentar el cumplimiento de las normas y promover las mejores prácticas. Lo cual, convierte a los patrones de diseño de interacción en un conjunto de herramientas informales [22, 28]. Sin embargo, dado que los patrones de diseño de interacción carecen de una representación de patrones efectiva, es decir, se entregan principalmente en texto narrativo, no existe una validación verdadera y no se ha proporcionado apoyo a los diseñadores para su correcta identificación y selección [32]. Los diseñadores han encontrado que los patrones de diseño de interacción son altamente beneficiosos, sin embargo enfrentan algunas dificultades cuando se trata de encontrarlos, comprenderlos y aplicarlos correctamente en un contexto de uso. Debido a que se requiere de experiencia y trabajo para modificar los diseños existentes para su reutilización [33–35]. En el caso

de un diseñador que no tenga suficiente experiencia, tendría que generar desde cero los prototipos de IGUs; salvo que exista la posibilidad de utilizar patrones de diseño de interacción para generar diferentes propuestas de IGUs sin mucho esfuerzo y siempre con el objetivo del cumplimiento de las necesidades de los usuarios.

En la definición de los requerimientos se establecen y especifican las necesidades de los usuarios, las cuales en algunos casos serán cubiertas a través de IGUs. Por lo tanto, para lograr el entendimiento e identificación de patrones de diseño de interacción, es necesario analizar la información contenida en los requerimientos. Sin embargo, hasta el momento existen esfuerzos aislados respecto a la asociación de patrones de diseño de interacción con requerimientos a nivel de diseño.

1.3. Preguntas de Investigación

Con base en la problemática descrita previamente, se definen las siguientes preguntas que guían este trabajo de tesis:

- *¿Cómo se puede explotar el establecimiento de requerimientos para el diseño de interfaces gráficas de usuario?*
- *¿Cuáles elementos de información de un requerimiento son tomados como referencia para el diseño de interfaces gráficas de usuario?*
- *¿Cómo se relacionan los requerimientos con los patrones de diseño de interacción en el proceso de diseño de interfaces gráficas de usuario?*
- *¿Cómo se pueden utilizar los enfoques de aprendizaje automático para identificar patrones de diseño de interacción con base en los requerimientos?*
- *¿Cuál es el impacto del diseño de la interfaz gráfica de usuario con base en la recomendación del patrón de diseño de interacción?*

1.4. Objetivos de la Investigación

1.4.1. Objetivo General

Generar un modelo de recomendación de patrones de diseño de interacción con base en los requerimientos y utilizando técnicas de Inteligencia Artificial (IA) en apoyo al diseño de Interfaces Gráficas de Usuario (IGUs).

1.4.2. Objetivos Específicos

Del objetivo general, se derivan los siguientes objetivos específicos:

- Estructurar los elementos de información contenidos en un requerimiento funcional en formato textual, para analizar lo solicitado en cada uno de los requerimientos.
- Clasificar los elementos de información contenidos en el texto de un requerimiento funcional de acuerdo a sus características, para asociarlos con un conjunto de patrones de diseño de interacción.
- Generar un conjunto de datos con instancias de requerimientos, para poder implementar modelos de aprendizaje automático para asociar instancias de requerimientos y patrones de diseño de interacción.
- Generar un modelo de recomendación de patrones de diseño de interacción con base en las asociaciones aprendidas en relación con los elementos de información contenidos en los requerimientos, para apoyar a los diseñadores en la toma de decisiones de diseño de interfaces gráficas de usuario.
- Validar el impacto del modelo de recomendación de patrones de diseño de interacción en el diseño de interfaces gráficas de usuario, para determinar su precisión y utilidad en el proceso de diseño de interfaces gráficas de usuario.

1.5. Metodología de investigación

La metodología de investigación utilizada para el desarrollo de este trabajo de tesis, consta de siete etapas (ver Figura 1.4). Las actividades de cada etapa se describen a continuación:

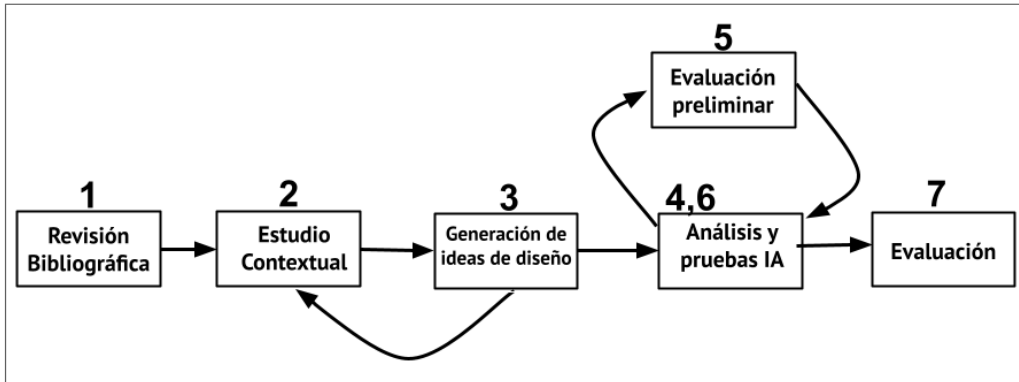


Figura 1.4: Metodología de Investigación.

1. **Revisión bibliográfica.** En esta etapa se realizó un extenso análisis bibliográfico en relación al proceso de diseño de IGUs desde la perspectiva de ingeniería de software y la interacción humano-computadora. Dicho análisis tuvo como objetivo identificar el trabajo relacionado con el tema propuesto y establecer el marco teórico de esta investigación.
2. **Estudio Contextual.** Esta etapa consistió en realizar un análisis de los elementos de información considerados por los diseñadores al diseñar interfaces gráficas de usuario, con el objetivo de identificar los elementos de información necesarios para encontrar y aplicar patrones de diseño de interacción durante el diseño de las interfaces gráficas de usuario con base en requerimientos de software. Para este propósito, se realizó un estudio de caso que analiza la identificación de patrones de diseño de interacción, a través de la comprensión de los requerimientos funcionales por parte de programadores junior. Para lo cual, se realizaron una serie

de sesiones en las cuales se le dio seguimiento al proceso de diseño de un sistema interactivo, y en las cuales se les solicitó a los programadores junior la generación de especificaciones de requerimientos con base en las cuales identificaron los patrones de diseño de interacción para el diseño de IGUs.

3. **Generación de ideas de diseño.** Sobre la base de la revisión de la literatura y el estudio contextual, se establecieron las pautas para el diseño de la primera versión del modelo para identificar y aplicar patrones de diseño de interacción, en apoyo en el diseño de IGUs con base en requerimientos de software. Las etapas dos y tres deben realizarse de manera iterativa para refinar las ideas de diseño. Como parte de la primera iteración, se definió un proceso para la identificación de patrones de diseño de interacción a través de requerimientos.

Se propone que las tareas de interacción se extraigan de los requerimientos, para luego ser parte de la entrada a los algoritmos de las máquinas de aprendizaje. Para lo cual, se propuso la generación de un conjunto de datos de instancias de requerimientos para poder entrenar a los algoritmos. Esto resultó en una recomendación de patrones de diseño de interacción apropiados para aplicar en ciertas tareas.

4. **Análisis y pruebas de algoritmos de IA.** En esta etapa, se analizaron y probaron las técnicas de IA, identificando la técnica más apropiada en la que se recomendarán los patrones de diseño de interacción aplicables de acuerdo con los parámetros establecidos. Sobre la base de la propuesta de la primera iteración de diseño, se utilizaron técnicas de Procesamiento del Lenguaje Natural (PLN) y aprendizaje automático.
5. **Evaluación preliminar.** Teniendo en cuenta el diseño de las interfaces gráficas de usuario, se procedió a realizar una evaluación de los algoritmos de IA aplicados, con base en métricas de precisión.

6. **Rediseño del modelo.** Sobre la base de los resultados de la evaluación preliminar, se ajustaron los parámetros para la evaluación de los algoritmos del modelo de recomendación propuesto.
7. **Evaluación.** Dentro de esta etapa, se llevó a cabo la planificación y ejecución de un caso de estudio, donde se presentó la visión tecnológica a diseñadores de IGUs profesionales. Además, se realizó un análisis detallado de los datos obtenidos durante el estudio, donde se generaron las conclusiones de la investigación.

En la siguiente parte de este trabajo de tesis se definen las partes en las que está constituida la misma.

1.6. Contribuciones principales de esta tesis

Las principales contribuciones derivadas de este trabajo de tesis son presentadas a continuación:

- Un modelo de recomendación de patrones de diseño de interacción con base en requerimientos funcionales asociados con patrones de diseño de interacción.
- Un conjunto de datos con instancias de requerimientos a nivel de diseño.
- Una aplicación de métodos de aprendizaje automático para asociar instancias de requerimientos y patrones de diseño de interacción.

1.7. Contenido del documento

El resto de este documento está dividido en 7 capítulos y 4 apéndices, el contenido de cada uno de ellos se describe brevemente a continuación:

En el capítulo 2 se presenta la descripción de un conjunto de trabajos de distintas áreas de investigación, con los cuales se argumenta el marco teórico para entender y

establecer en qué consiste el diseño de IGUs. De manera particular, se centran en trabajos de patrones de diseño de interacción destacando los beneficios y sus oportunidades de investigación.

En el capítulo 3 se presenta un resumen general de trabajos de investigación que contribuyen con algunos elementos que inciden de alguna manera en la recomendación de patrones de diseño de interacción con base en requerimientos a nivel de diseño. Estos trabajos son parte de los campos de estudio: procesamiento de lenguaje natural y clasificación de texto.

En el Capítulo 4 se describe un caso de estudio con el objetivo de conocer cómo se identifican patrones de diseño de interacción en el proceso de diseño de IGUs. Además de que se plantea un conjunto de hallazgos que motivan la necesidad especificada en el capítulo 1.

En el capítulo 5 se describe de manera detallada el modelo de recomendación de patrones de diseño de interacción a través de requerimientos a nivel de diseño en formato textual. Así mismo, se detalla como se aplican algoritmos de IA para el procesamiento de texto y recomendación de patrones de diseño de interacción.

En el capítulo 6 se describe la evaluación de la propuesta a través de métricas para determinar el desempeño y precisión de los algoritmos de IA. Además de un caso de estudio, el cual consistió en presentar el modelo de recomendación de patrones de diseño de interacción a diseñadores de IGUs profesionales. Dicha evaluación es presentada en cuatro partes: la delimitación de los objetivos que deben alcanzarse, la proposición de las hipótesis, la validación de las hipótesis y el análisis de los resultados. La finalidad de esta evaluación fue obtener la percepción de utilidad del modelo de recomendación de patrones de diseño de interacción, así como la facilidad de uso de la visión tecnológica con los usuarios finales, en apoyo al diseño de IGUs.

Al final se presentan las conclusiones y reflexiones derivadas de este trabajo de tesis; las principales aportaciones, entre las que se encuentran las publicaciones realizadas

durante el desarrollo de la tesis; así como algunas oportunidades de trabajo futuro.

El apéndice A presenta el cuestionario escala Likert-5 que el grupo de participantes respondieron para la selección de patrones de diseño de interacción, a un conjunto de 10 requerimientos funcionales a nivel de diseño.

El apéndice B muestra el cuestionario escala Likert-5 que el grupo de participantes respondieron para evaluar las predicciones de patrones de diseño de interacción, emitidas para un conjunto de 10 requerimientos funcionales a nivel de diseño.

El apéndice C presenta el cuestionario para la Escala de Usabilidad del Sistema (SUS-System Usability Scale) aplicado en el caso de estudio realizado con diseñadores de IGUs profesionales.

El apéndice D muestra el cuestionario del Modelo de Aceptación de la Tecnología (TAM-Technology Acceptance Model) aplicado al grupo de participantes para la evaluación del caso de estudio con diseñadores de IGUs profesionales.

Capítulo 2

ANTECEDENTES.

En este capítulo se presenta un análisis de trabajos de investigación relevantes para el tema del diseño de IGUs. Con base en dicho análisis se argumenta en qué consiste el diseño de interacción de sistemas interactivos para posteriormente lograr un entendimiento inicial de dicho proceso. Así mismo, se establece la necesidad de contar con información que apoye a los diseñadores para el diseño de IGUs intuitivas y usables.

Finalmente, se identifica una serie de aspectos que deben ser cubiertos para lograr los objetivos de este trabajo de tesis.

2.1. Ingeniería de Software

En los últimos años, ha sido desarrollada una variedad de técnicas y métodos de IS con el fin de mejorar y apoyar el proceso de desarrollo de software profesional. IS es una disciplina de ingeniería que comprende todos los aspectos que se involucran en la producción de software, desde la especificación del sistema hasta el mantenimiento del mismo, una vez que se pone en operación.

El proceso de desarrollo de software requiere producir software de alta calidad, para lo cual se sigue una secuencia de actividades. De acuerdo con Sommerville [5] existen

cuatro actividades fundamentales que son comunes en todos los procesos de software y las cuales son:

1. Especificación del software, actividad en la cual se definen el software que se producirá y las restricciones en su operación.
2. Desarrollo del software, en donde se diseña y se codifica el software.
3. Validación del software, en donde se verifica que el software cumpla con los requerimientos del cliente.
4. Evolución del software, en donde se modifica el software para reflejar los nuevos requerimientos del cliente y del mercado.

Estas actividades fundamentales se llevan a cabo de acuerdo al tipo de sistema interactivo que se requiere desarrollar, del personal y de la inclusión de estructuras organizativas. Sin embargo, todos los procesos de software inician a partir de la actividad de especificación del software o IR para comprender y definir qué servicios se requieren del sistema interactivo, así como la identificación de especificaciones sobre la operación y el desarrollo del mismo.

2.1.1. Ingeniería de Requerimientos

La IR comprende una etapa crítica del proceso de software, la cual tiene como objetivo el recopilar los requerimientos más significativos y correctamente establecidos por parte de los clientes y usuarios. Para lo cual es importante la aplicación de diferentes técnicas de IR a lo largo del proceso de desarrollo de sistemas interactivos. A partir de la aplicación de dichas técnicas es posible la identificación crítica de las necesidades de las partes involucradas, así como el apoyo a la toma de decisiones de manera eficiente. El éxito o el fracaso de un proyecto de software dependen de la precisión y gestión efectiva de dichos requerimientos [5]. Lo anterior, debido a que los errores en esta etapa

conducen a problemas en etapas posteriores como el diseño y la implementación del sistema.

Estudios como el de Vaz et al. [36] han demostrado que el costo de corregir errores de ingeniería en sistemas y software aumenta exponencialmente a lo largo del ciclo de vida del proyecto. La cantidad de errores debidos a requerimientos deficientes, ambiguos e incoherentes se está volviendo inmanejable y conduce a sobrecostes dramáticos y demoras sistémicas.

De acuerdo con Hussain [37], la falta de información del usuario (12.80%), los requerimientos y especificaciones incompletas (12.30%) y los requerimientos cambiantes (11.80%) son los tres factores principales responsables de los proyectos cuestionados. Así como, los requerimientos incompletos (13,10%), la falta de participación del usuario (12,40%) y la falta de recursos (10,60%) son los tres factores principales que causan proyectos deteriorados o fallidos. Por lo tanto, los problemas relacionados con los requerimientos son factores fundamentales que intervienen en el éxito o el fracaso de un proyecto de software.

En IS hay una búsqueda constante para producir sistemas de software que satisfagan las necesidades de sus entornos y sus usuarios. El éxito del software depende en gran medida de qué tan bien se hayan entendido y traducido los requerimientos de los usuarios en las funcionalidades apropiadas durante el proceso de diseño. Las fallas de los proyectos de software suelen ser costosas, riesgosas y plantean serios desafíos en el mercado competitivo actual [38], ya que afectan negativamente la imagen de la empresa, la lealtad del cliente, los ingresos y también pueden disminuir la satisfacción percibida de los clientes.

Los requerimientos son declaraciones establecidas por todas las partes interesadas sobre el desarrollo de un sistema. Estas declaraciones van orientadas hacia el cumplimiento de los objetivos para que los problemas de los clientes y usuarios se resuelvan realmente. Sin embargo, los requerimientos se pueden clasificar en varios grupos de

acuerdo a la necesidad de los involucrados. Tal es el caso de los requerimientos comerciales, requerimientos de usuario, Requerimientos Funcionales (RF), Requerimientos No-Funcionales (RNF), requerimientos de rendimiento y requerimientos de seguridad (ver Figura 2.1) [4, 5].

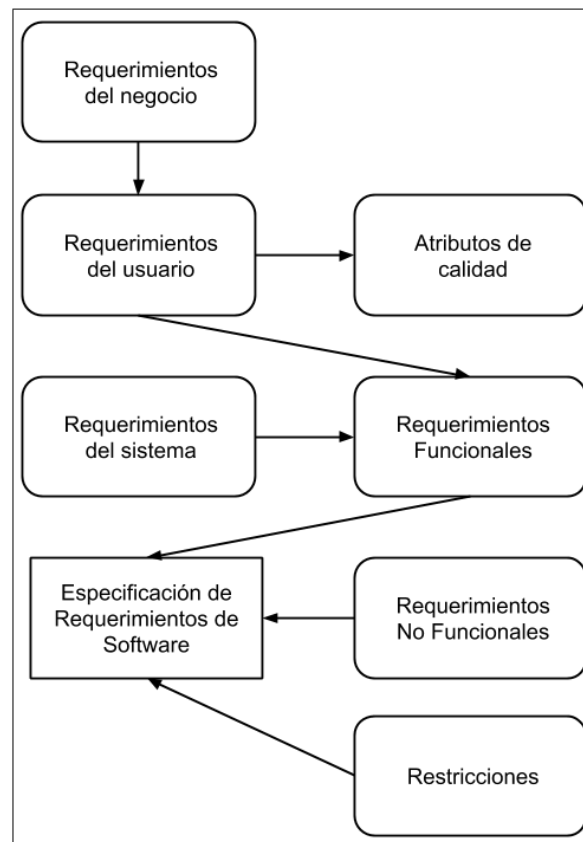


Figura 2.1: Tipos de requerimientos [4]

Cada uno de los requerimientos se puede identificar a través del proceso de IR y son escritos con diferentes niveles de detalle. Para satisfacer estas necesidades, los requerimientos de software se definen en el proceso de IR, que implica obtención, análisis, especificación y verificación. El análisis de requerimientos puede considerarse como uno de los pasos más complejos en el proceso de IR para desarrollar un sistema de software [39, 40]. En IS, los requerimientos de software se definen a través del documento de especificaciones de Especificaciones de Requerimientos de Softwa-

re (SRS-Software Requirement Specification) y desempeñan un papel fundamental en los proyectos de desarrollo de software. Un documento SRS tiene un conjunto diverso de partes interesadas, tales como, arquitectos, diseñadores de IGU, evaluadores, entre otros (ver Figura 2.2); los cuales utilizan estas especificaciones para la comunicación y la información [5,41].

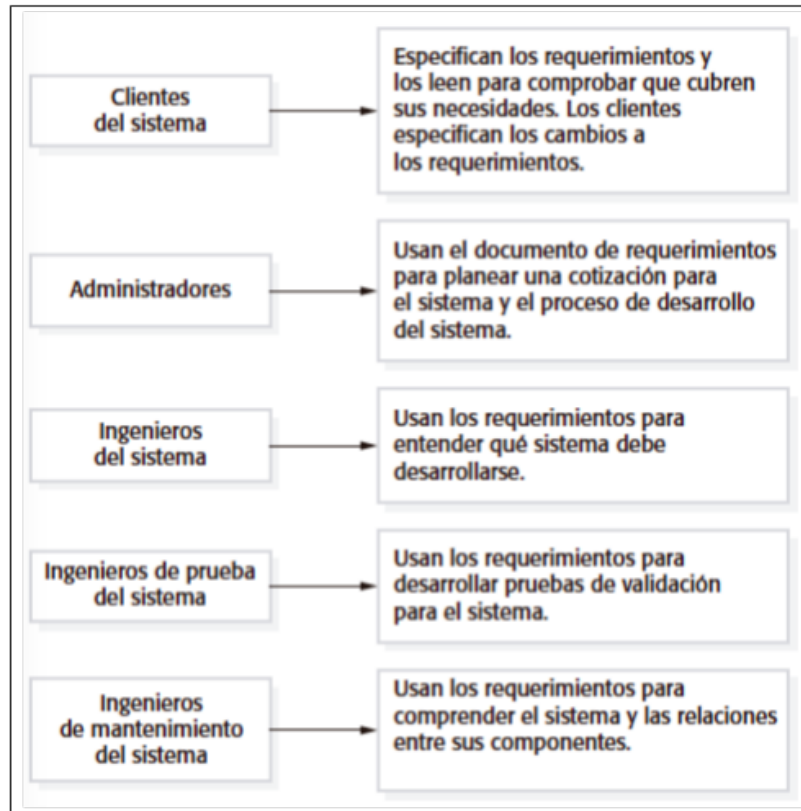


Figura 2.2: Usuarios del documento de especificaciones de requerimientos de software (SRS) [5].

El SRS tiene como base los RF y RNF. Los RNF son limitaciones sobre servicios o funciones que ofrece el sistema interactivo. Se relacionan con propiedades emergentes del sistema, tales como la fiabilidad, tiempo de respuesta y uso de almacenamiento. Los RNF se establecen a partir de las necesidades de los usuarios, a las restricciones de presupuesto, políticas de la empresa, necesidades de interoperabilidad con otros

sistemas, regulaciones de seguridad o legislaciones de privacidad. Por otro lado, los RF son las declaraciones de los servicios proporcionados por el sistema interactivo, el modo de reaccionar a las entradas particulares y las salidas esperadas. Son básicamente los requerimientos establecidos por el cliente y usuario que se pueden ver directamente en el producto final, a diferencia de los RNF.

El documento SRS consta de texto (es decir, datos) que cada parte interesada puede interpretar de manera diferente. Para poder comprender, modelar y presentar una posible solución al problema, las partes interesadas deben ser competentes para traducir el texto (es decir, los requerimientos) en conocimiento [42]. Para lo cual es necesario que los requerimientos sean claros, sin ambigüedades, fáciles de entender, consistentes y completos. Para minimizar la interpretación errónea al escribir requerimientos en lenguaje natural, Sommerville [5] establece los siguientes lineamientos:

- a) Elaborar un formato estándar en dónde se especifiquen todas las definiciones de requerimientos en una sola oración.
- b) Utilizar un lenguaje claro para distinguir requerimientos obligatorios de requerimientos deseables.
- c) Resaltar las partes claves del requerimiento.
- d) Evitar utilizar lenguaje tan técnico, abreviaturas y acrónimos.
- e) Definir las razones por las cuáles fue incluido un requerimiento. Esto resulta de utilidad cuando los requerimientos cambian y se requiere tomar decisiones sobre cuáles cambios serían indeseables.

Sin embargo, esto en la práctica resulta complejo dado lo expresivo, intuitivo y universal del lenguaje natural. Debido a esta situación, los investigadores han aprovechado técnicas de IA para automatizar actividades como la identificación de inconsistencias,

búsqueda de redundancias, mapeo de texto al código, identificando deficiencias y defectos en el texto de los requerimientos, entre otros [43].

2.1.2. Técnicas de Inteligencia Artificial en Ingeniería de Software

La IA es una área de las Ciencias en la Computación que estudia la simulación de inteligencia humana por parte de las computadoras. Para lo cual se emplean diferentes técnicas con la finalidad de que los sistemas inteligentes sean capaces de recopilar, analizar datos, aprender de la experiencia de otras máquinas y así poder comunicarse con otros sistemas. En los últimos años, las técnicas de IA han sido utilizadas ampliamente para apoyo en la clasificación de documentos, identificación de temas y extracción de conceptos a partir de documentos textuales. En el caso particular de RE, la clasificación de requerimientos en el análisis de requerimientos es una tarea que requiere mucho tiempo. Para la cual, la técnica de PLN proporciona apoyo a las partes interesadas en el análisis de dichos requerimientos [6, 43, 44].

En este sentido, los conceptos de PLN son útiles para el análisis. El sistema PLN procesa los datos escritos en lenguaje natural de manera inteligente y genera información que es comparativamente más fácil de analizar. El PLN se ocupa de los esfuerzos combinados de la informática y la lingüística y de cómo la computadora y el lenguaje humano interactúan entre sí. Una vez proporcionado el texto, la computadora a través de algoritmos extrae el significado asociado a cada oración y recopila los datos esenciales para poder interpretarlo [45]. Para el procesamiento del texto en lenguaje natural es importante el análisis sintáctico y semántico. El análisis sintáctico se usa para evaluar cómo el lenguaje natural se alinea con las reglas gramaticales. Mientras que la semántica se refiere al significado que transmite un texto [6, 42, 44].

En general, las técnicas de PLN incluyen 4 pasos principales: 1) Análisis léxico,

2) Análisis sintáctico, 3) Análisis semántico y 4) Transformación de salida (ver Figura 2.3). El análisis léxico es un proceso importante que analiza las oraciones del lenguaje natural que se divide en pequeñas unidades llamadas Token junto con el tipo y la información esencial utilizada en el siguiente paso. Sin embargo, un análisis defectuoso y las palabras incorrectas segmentadas contribuirán a una interpretación incorrecta, un significado incorrecto y resultados de salida erróneos [46].

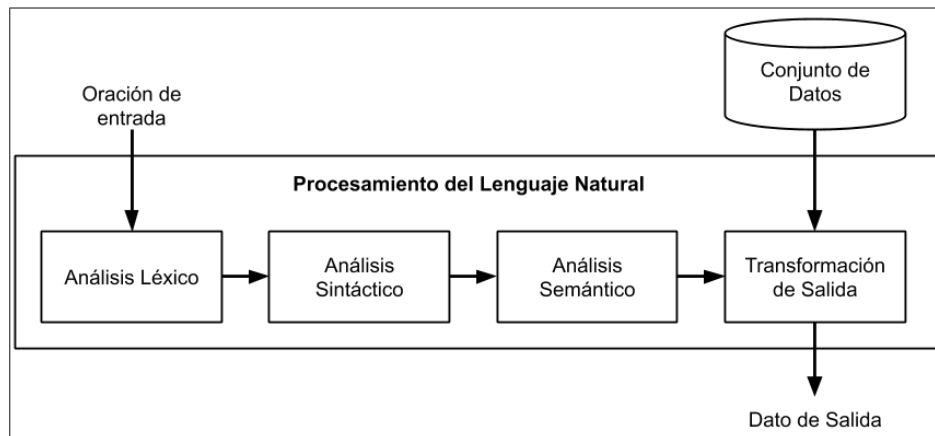


Figura 2.3: Pasos del Procesamiento del Lenguaje Natural (PLN) [6]

Los requerimientos son fundamentales para la planeación de proyectos de desarrollo de software. Además, de permitir verificar y medir si los objetivos establecidos han sido alcanzados, debido a que éstos son un reflejo de las necesidades de los clientes o usuarios del sistema [5]. La calidad de software implica aspectos de usabilidad, los cuales son validados por el área de IHC, utilizando prototipos. De esta manera, ambas disciplinas coinciden en que necesitan métodos para medir qué tan bien sus productos y procesos de desarrollo cumplen con los requerimientos previstos [47, 48].

2.2. Interacción Humano-Computadora

Los sistemas interactivos se encuentran inmersos en las actividades de la vida diaria de las personas. En donde el comportamiento de los usuarios de sistemas interactivos

toma mayor relevancia y se enmarca dentro de las ciencias cognitivas [48, 49]. IHC es un campo de estudio multidisciplinario que involucra disciplinas tales como, Ciencia de la Computación, Psicología y Antropología, entre otras. Así mismo, IHC provee de un entendimiento de la forma en que los usuarios interactúan con los sistemas interactivos, para poder diseñar sistemas interactivos efectivos, eficientes, fáciles de utilizar y estéticamente placenteros, con la finalidad de facilitar el logro de tareas por parte de los usuarios [48–50]. Para ello, se establece la necesidad de contar con métodos para analizar las tareas, las necesidades y comportamientos de los usuarios, lo cual es parte central del diseño de un sistema interactivo.

IHC abarca el diseño, implementación y evaluación de sistemas interactivos en un contexto específico, para tareas específicas y usuarios específicos. En la Figura 2.4 se muestra al humano (se refiere a un grupo de usuarios) con sus características del procesamiento de la información, de comunicación (lenguaje, comunicación, interacción) y físicas (ergonomía) que interactúa con una computadora (que se refiere a cualquier sistema interactivo) que tiene una serie de características tecnológicas que soportan esta interacción. En el otro lado, están los dispositivos de entrada y de salida que “conectan” a la persona con una computadora y se comunican mediante determinadas técnicas o reglas de diálogo que manejan distintos elementos de diseño (como por ejemplo las metáforas), todo esto soportado por la arquitectura interna del diálogo y por distintas técnicas de computación gráfica [7].

Esta interacción se lleva a cabo en un contexto de uso, que puede variar de acuerdo al área de aplicación. Así mismo, el sistema con el que se interactúa previamente pasó por un proceso de desarrollo, lo cual involucra al área de la ingeniería de software. Sin embargo, la interacción entre el humano y la computadora es parte esencial en el estudio de IHC. Para lo cual han sido abstraídos modelos de interacción para entender dicho proceso y poder proponer nuevos diseños potencialmente disruptivos pero útiles y con una buena experiencia de uso.

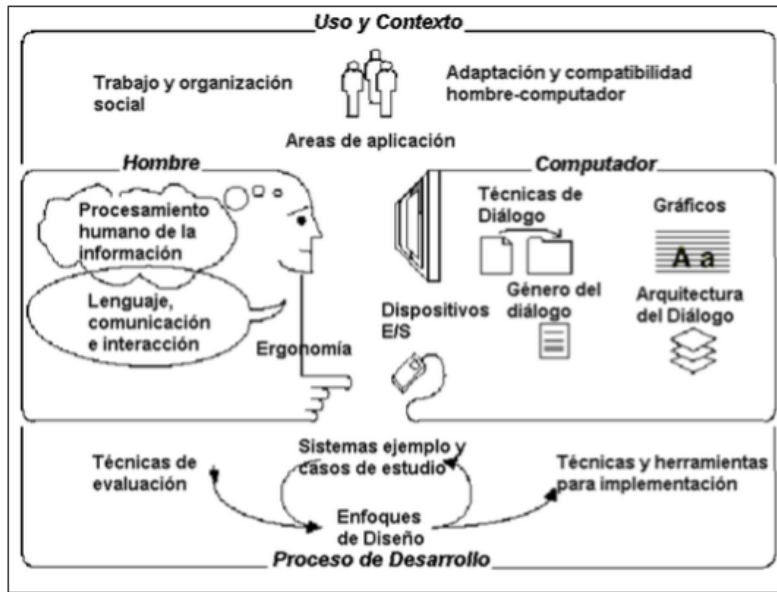


Figura 2.4: Esquema de Interacción Humano-Computadora (IHC) [7]

De acuerdo con Myers et al. [51], el desarrollo de IGUs va desde los primeros requerimientos hasta la obsolescencia del software, lo que lo convierte en un proceso costoso y que lleva mucho tiempo. Normalmente, la IGU de un sistema interactivo representa aproximadamente el 48 % del código fuente, requiere aproximadamente el 45 % del tiempo de desarrollo y el 50 % del tiempo de implementación, y cubre el 37 % del tiempo de mantenimiento. Debido a la importancia de proveer de IGUs usables y que brinden una buena experiencia de usuario, el diseño de interacción define la estructura y el comportamiento de sistemas interactivos.

2.2.1. Diseño de Interacción

El Diseño de Interacción (DI) surge a partir de las primeras IGUs. Sin embargo, la evolución de las formas de interacción y la tecnología impulsaron la aplicación del término "facilidad de uso". Además del surgimiento de otras subáreas para apoyar en el cumplimiento de crear relaciones significativas entre las personas, los productos y

servicios que utilizan.

Los sistemas interactivos cuando resultan efectivos y eficientes para los usuarios, asociados al aporte que brindan los avances en la tecnología y su aplicación creativa, son tomados como paradigmas para futuros sistemas y aportan a la mejora de la interacción [50, 52]. Con base en esto, a través del tiempo han surgido modelos de IHC, los cuales de acuerdo a Rekimoto [8] son clasificados en cuatro paradigmas de interacción: IGU, Realidad Virtual, Cómputo Ubicuo y Realidad Aumentada (ver Figura 2.5).

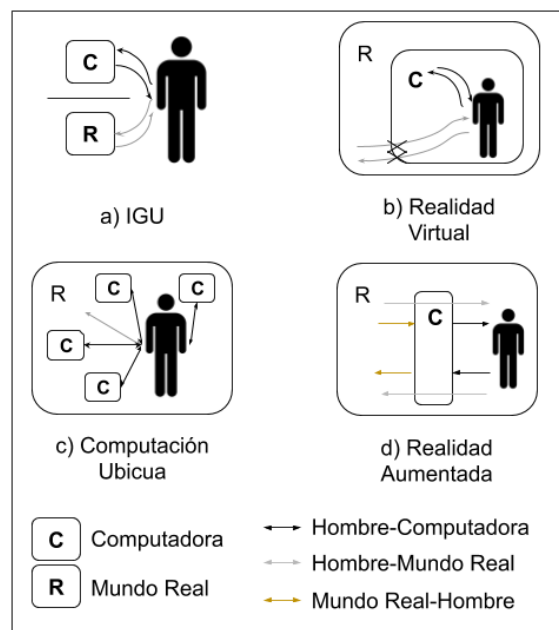


Figura 2.5: Comparación de Paradigmas de Interacción [8]

En el paradigma de una IGU, la interacción del usuario con la computadora es de manera aislada de la interacción entre el usuario y el mundo real. Mientras que en la realidad virtual, la computadora cubre totalmente al usuario y la interacción de éste con el mundo real desaparece. En cuanto a la computación ubicua, el usuario interactúa con el mundo real pero también puede interactuar con las computadoras disponibles en el mundo real. Finalmente, en la realidad aumentada se soporta la interacción entre el usuario y el mundo real utilizando la información aumentada de la computadora [8].

Estos paradigmas generan la existencia de usuarios más diversos y menos técnicos, es decir, que no se preocupan por el funcionamiento interno del software, sino en cómo usarlo [53,54]. Para lo cual, la interfaz es el punto donde las personas y las computadoras convergen y transmiten información, órdenes y datos que provocan sensaciones, intenciones y nuevas formas de ver las cosas. Sin embargo, para una buena interacción es necesario diseñar sistemas interactivos que cumplan con las expectativas de usabilidad de los usuarios actuales. Es decir, que se adecuen a las capacidades cognitivas de las personas que lo usan [50,52].

La usabilidad es definida por la Organización Internacional de Normalización (ISO 9241-11) como “la capacidad de un producto de software para ser entendido, aprendido, utilizado y atractivo para el usuario, cuando se utiliza en condiciones específicas” [55]. Además de que la usabilidad es considerada como un aspecto de calidad integral en el desarrollo de software. Por lo cual, es necesario proporcionar a los diseñadores reglas de diseño, que les brinden la capacidad de determinar las consecuencias de sus decisiones de diseño en el aspecto de usabilidad. Los tipos de reglas de diseño son: principios, estándares y lineamientos o heurísticas. Los cuales son clasificados por Dix et al. [9] en dos dimensiones: con base en su generalidad y su autoridad (ver Figura 2.6).

En términos de autoridad los principios son determinados por la comunidad, pero sin llegar a ser obligatorios. En cuanto a la dimensión de generalidad de la aplicación, es más compleja dado que las pautas a menudo son específicas de algún fabricante. Posiblemente los estándares sean menos generalmente aplicables, pero el rango de situaciones al que se aplica suele ser mucho más estricto [9].

Una forma de abordar el DI es tomando como referencia ejemplos que han sido aplicados de manera exitosa. Para lo cual, los patrones de diseño de interacción pueden mejorar las actividades de comunicación y presentación de informes; permiten a los diseñadores que están familiarizados con los patrones de diseño de interacción discutir, documentar y compartir especificaciones o alternativas de diseño utilizando nombres

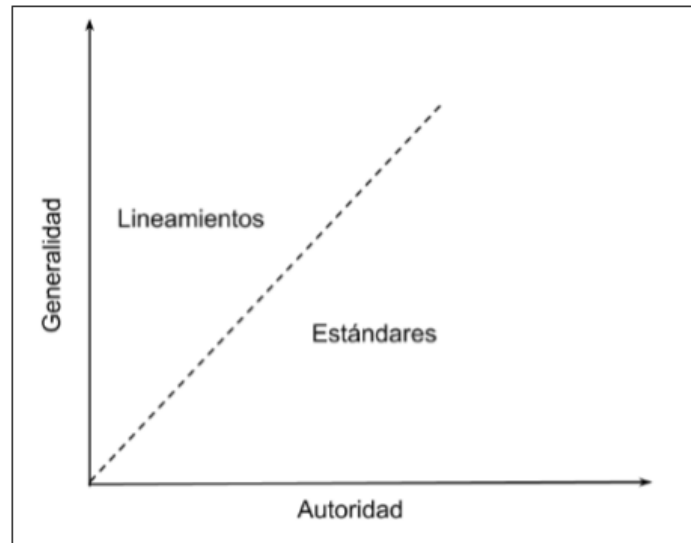


Figura 2.6: Dimensiones de reglas de diseño de acuerdo a Dix et al. [9]

conocidos y bien entendidos para soluciones de diseño en lugar de descripciones largas. En el dominio de IHC, la adopción de soluciones de diseño de interfaz populares y ampliamente utilizadas hace que la experiencia de usuario sea más intuitiva y predecible, aprovechando la familiaridad del usuario.

2.2.2. Patrones de Diseño de Interacción

Los patrones se originaron en la arquitectura, en donde fueron utilizados con éxito, así como en el desarrollo de software para capturar soluciones comprobadas a problemas comunes de programación. Según Alexander [3], un patrón se compone de tres partes: contexto, un problema y una solución.

Los patrones de diseño se usan cada vez más para capturar y difundir las mejores prácticas, así como para el diseño de interacción de sistemas interactivos con base en los requerimientos. Considerando en todo momento los criterios de satisfacción del usuario cuando utiliza un sistema interactivo como por ejemplo la facilidad de uso, la asistencia en caso de errores, la facilidad de aprender y el tiempo de respuesta de la

aplicación. Además, los patrones de diseño se convierten en un vocabulario compartido que ayuda a expresar y comunicar conocimiento técnico. Los patrones permiten crear componentes que brindan soluciones efectivas, por lo que los equipos de trabajo no requieren invertir horas intentando reinventar la rueda. Los diseñadores sólo requieren adaptar los patrones de diseño de interacción para un contexto específico, de manera que se agilizan los procesos de diseño y desarrollo de sistemas interactivos.

A partir del libro de Gamma et al. [56] (conocidos como The Gang of Four - GoF) los patrones de software se han expandido a otros dominios. Recientemente, los patrones han sido utilizados en el diseño de interfaces. Debido a que capturan las propiedades del buen diseño a través de los elementos comunes que se mantienen entre todas las instancias de la solución. La implementación específica de un patrón dependerá del contexto de uso y creatividad del diseñador. El primer conjunto de patrones de diseño de interacción fue la colección de patrones “Terreno común” desarrollada por Jenifer Tidwell [12]. Posteriormente surgieron otras colecciones, tales como los Patrones de Diseño de Interacción de Martijn van Welie [57], los enfoques basados en patrones de Borchers [58], etc.

Diferentes autores han propuesto estructuras de patrones de diseño de interacción, las cuales deben de facilitar la comunicación entre los diseñadores [12, 57, 58]. La estructura de un patrón hace referencia a propiedades y atributos externos que abordan los riesgos de la aplicación de patrones de diseño. No existe un formato comúnmente acordado para los patrones de diseño, por lo que los diferentes formatos contienen diferentes elementos. La mayoría de los formatos incluyen al menos las siguientes cuatro partes [59]:

- **Nombre:** un nombre significativo para identificar los patrones y se relaciona con el problema o la solución.
- **Contexto:** la(s) situación(es) donde el patrón es relevante para determinar cuándo

Tabla 2.1: Elementos de información de un patrón propuesto por Tidwell [12]

| | |
|----------------------------------|--|
| Nombre: | Nombre por el cual se identifica el patrón. |
| Problema: | Describe la situación que el patrón resolverá. |
| Contexto: | Características del usuario y características de las tareas que se realizarán. |
| Fortalezas: | Cómo influyen en diferentes aspectos del problema. |
| Solución: | Descripción de la solución propuesta. |
| Consecuencias: | Descripción de los resultados de aplicar el patrón. |
| Principios de Usabilidad: | Describe los principios o criterios ergonómicos sobre los cuales se basa. |
| Ejemplo: | Un ejemplo ilustrativo de una solución exitosa. |

se debe aplicar este patrón.

- **Fortalezas:** las fortalezas presentes, que pueden restringir o sugerir soluciones alternativas. Cuando estas fortalezas están en tensión unas con otras, el problema es más difícil de resolver y puede ser necesario un compromiso.
- **Solución:** una solución que resuelve, en la medida de lo posible, las diversas fortalezas.

De acuerdo con la revisión de [60], se presentan las diferentes estructuras propuestas por varios autores. En la Tabla 2.1 se muestran los elementos de la propuesta de Tidwell [12], donde se agrega a las cuatro partes básicas la descripción del problema que se resolverá con la aplicación del patrón, consecuencias de la misma aplicación, así como la descripción de los principios de usabilidad que toma en consideración el patrón, y ejemplos ilustrativos de una aplicación exitosa.

En el caso de diseño de IGUs, un diseñador se basa en información detallada sobre las características de los usuarios finales, así como en descripciones del flujo de trabajo

en el que se introducirá el sistema de software. Esta información tiene una importante influencia en las decisiones de diseño de la IGU [47]. A lo largo de los años, los diseñadores de IGU han encapsulado y sugerido soluciones probadas a problemas recurrentes de diseño. Las soluciones a los problemas recurrentes que suelen conocerse y organizarse como patrones de diseño de interacción [61]. Sin embargo, los diseñadores de IGU inexpertos normalmente luchan por encontrar patrones de diseño de interacción apropiados y determinar su aplicabilidad a ciertos requerimientos de diseño. Si la toma de decisiones es defectuosa en esta etapa, la selección y aplicación inapropiada de los patrones de diseño de interacción podrían causar interacciones ineficientes, ineficaces y decepcionantes para los usuarios.

En este sentido, para apoyar a los diseñadores de IGUs es necesario un análisis enfocado a lo que se aporta en la literatura en cuanto a la integración de los requerimientos relacionados a IGUs y los patrones de diseño de interacción. Este análisis conducirá a modelar la recomendación para apoyar a los diseñadores de IGUs a identificar patrones de diseño de interacción. Así mismo, con base en dicho análisis se establecerá la propuesta del conjunto de datos para el recomendador.

2.3. Resumen del capítulo

En este capítulo se ha dado una descripción de las principales áreas y conceptos relacionados con el diseño de IGUs en sistemas interactivos. Con base en dicho análisis de la literatura, se obtuvo una descripción del proceso de diseño de sistemas interactivos desde el punto de vista de IS y IHC. Además de las facilidades que proporcionan las técnicas de IA, para que el análisis de requerimientos se realice de manera efectiva y eficiente. Finalmente, se definió como los patrones de diseño de interacción se utilizan para la reutilización de ideas de diseño, que previamente han sido comprobadas.

En el siguiente capítulo se presenta un análisis basado en un conjunto de trabajos

de investigación de las áreas del uso de Patrones de Diseño de Interacción, así como la interpretación de requerimientos de software a nivel de diseño. Con base en esta revisión de la literatura, son identificados los hallazgos para un modelo en apoyo a las tareas de diseño de interacción de los diseñadores de IGUs.

Capítulo 3

PATRONES DE DISEÑO DE INTERACCIÓN EN LA INGENIERÍA DE SOFTWARE

En el capítulo anterior se describió un análisis de la literatura, con base en las áreas relacionadas en el proceso de desarrollo de sistemas interactivos. Dicho análisis fue soporte para establecer el marco teórico y entender cómo y qué elementos de información intervienen en el diseño de interfaces de sistemas interactivos. En este capítulo se presenta un análisis con base en una serie de trabajos, los cuales aportan elementos que sirven como complemento para establecer ideas de diseño para la identificación y aplicación de patrones de diseño de interacción, en el diseño de IGUs.

Este capítulo está compuesto de dos secciones, en las cuales se muestra cómo los trabajos citados inciden en la presente investigación. Estas secciones son *el uso de patrones de diseño de interacción y análisis de requerimientos de software*.

En relación con el uso de patrones de diseño de interacción, algunos investigadores reportan cómo los diseñadores encuentran los patrones de diseño de interacción altamente beneficiosos cuando enfrentan algunas dificultades en el proceso de diseño [12,28,35,62]

y requieren establecer una correcta comunicación con el resto del equipo involucrado en el desarrollo de sistemas interactivos [25,28]. Por lo tanto, es necesario un análisis más profundo cuando los diseñadores intentan encontrar, comprender y aplicar correctamente patrones de diseño de interacción en sus proyectos de diseño de sistemas interactivos, de manera que logren satisfacer las necesidades de los usuarios con su producto final.

En relación con la interpretación de requerimientos de software a nivel de diseño de interacción, se han realizado grandes esfuerzos. Algunos de ellos enfocados en tratar el problema del análisis de requerimientos, dado que es la parte más difícil del proceso de IR para desarrollar un sistema de software, debido a que es responsable de decidir qué es lo que se deberá de construir como producto final [6, 44, 63]. Por lo anterior, la clasificación de requerimientos representa una fase temprana pero crítica en la etapa de análisis de requerimientos. Es por ello, que para entender de mejor manera la relación de estos trabajos con el diseño de IGUs, éstos fueron divididos con base en el tipo de requerimientos, los métodos y los conjuntos de datos utilizados.

3.1. Uso de Patrones de Diseño de Interacción

Inicialmente el término patrón fue acuñado para el diseño de edificios y planificación arquitectónica. Posteriormente, el término fue adoptado en otros campos como el de IHC y la IS. En el contexto del diseño de IGUs, los patrones de diseño de interacción están destinados a ayudar en la identificación de elementos de diseño concretos que proporcionan soluciones a los problemas de diseño de interacción [64].

Los patrones de diseño de interacción son el resultado de aspectos comunes en el uso real de sistemas interactivos que identifican lo que funciona bien o lo que ellos consideran exitoso en situaciones cotidianas [27, 28]. Los patrones de diseño de interacción suelen constar de tres elementos esenciales: problema, solución y contexto. El problema se refiere a problemas de diseño comunes que enfrenta el usuario al utilizar el sistema [65].

En este sentido, los desarrolladores y diseñadores tienen poco apoyo en el proceso de toma de decisiones de diseño de interacción, ya que tienen esa responsabilidad exclusiva. Es tarea de los desarrolladores y diseñadores transformar su conocimiento sobre el usuario en decisiones de diseño de interacción rentables que determinen la experiencia de uso en un sistema interactivo. Si las decisiones fueran inadecuadas, esto derivaría en una mala selección y aplicación de los patrones de diseño de interacción, lo que a su vez podría causar una interacción del usuario ineficiente, ineficaz e insatisfactoria.

Los patrones de diseño de interacción en la interfaz de usuario se han estudiado intensamente en diferentes áreas de problemas de diseño. La mayoría de los patrones de diseño de interacción se han centrado en galerías web y catálogos, donde hay una gran cantidad de ejemplos disponibles listos para ser utilizados. Tidwell [12] describe una colección de patrones de diseño de interacción para hacer frente a problemas de diseño comunes de la interfaz de usuario web, de escritorio, móvil o de redes sociales. Así mismo el diseño de patrones de interacción de Martijn et al. [62] presentan la primera colección que hace la distinción explícita entre la perspectiva del usuario o la perspectiva del diseñador, y establece un lenguaje propio. Sin embargo, los formatos basados en texto para documentar patrones de diseño de interacción tienden a ser abstractos y ambiguos y, por lo tanto, difíciles de usar y reutilizar [66, 67].

3.1.1. Identificación y aplicación de Patrones de Diseño de Interacción

El conocimiento expresado por los patrones de diseño de interacción captura la esencia de una solución exitosa dentro de un contexto dado a un problema recurrente con respecto al diseño de IGUs. Además de ser una herramienta para facilitar la comunicación entre las personas involucradas en el desarrollo de software, tales como, los ingenieros de software, diseñadores de interacción, expertos en usabilidad, usuarios

finales y otros *stakeholders*. Para lo cual, los patrones de diseño de interacción requieren ser legibles y comprensibles tanto para profesionales como para no profesionales [67]. Sin embargo, con una gran cantidad de colecciones de patrones de diseño de interacción disponibles, los diseñadores deben leer manualmente algunas pilas de textos o navegar un tiempo en la web para buscar algunos patrones de diseño de interacción concretos para aplicar en el diseño de sistemas interactivos. En este sentido, Gaffar et al. [35] propuso un “Patrón Diseminador” que puede ser una persona o un agente de software, que interactúa con los autores de los patrones de diseño de interacción para apoyarlos en una generación semiestructurada de los diseños. El “Patrón Diseminador” puede ser parte de herramientas de software que ayudan a los diseñadores a encontrar, filtrar y combinar los patrones de diseño de interacción en un nuevo diseño. El “Patrón Diseminador” se fundamenta en un conjunto de datos en Lenguaje Extensible de Etiquetado (XML-Extensible Markup Language) subyacente que, a su vez, es retroalimentada. Sin embargo, la propuesta requiere una adecuada estructuración de los datos que conforma cada uno de los patrones de diseño de interacción, para poder consultarlos. Así mismo, el desarrollador es responsable de realizar consultas que representen la mayoría de las posibilidades de necesidades del usuario.

Para integrar patrones de diseño de interacción en cada etapa del proceso de desarrollo, en los últimos años algunos investigadores han propuesto una colección de modelos que impulsan el proceso de desarrollo de IGU. Desde el siglo pasado (los años 80s), la generación gráfica y textual de las IGU se basó en metamodelos (gramáticas) lo que demuestra que la comunidad de IHC lleva tiempo trabajando en estos temas [68]. El modelo puede ser semiautomatizado, como en el caso de la ingeniería dirigida por modelos y los entornos de Desarrollo de Interfaces de Usuario basados en Modelos (MBUID-Model-Based User Interface Development).

El entorno de MBUID tiene como objetivo identificar modelos de alto nivel que permitan a los diseñadores especificar y analizar aplicaciones de software interactivas desde

una perspectiva más semántica [69]. MBUID contribuye a esto, a través de la definición explícita de modelos relacionados con la IGU que se capturan y actualizan durante todo el ciclo de vida del desarrollo. Sin embargo, MBUID pone especial énfasis en una fase de modelado para garantizar que la implementación aborde los requerimientos centrados en el usuario, a partir de los cuales se define un conjunto de modelos de IGU.

Así mismo, MBUID hace explícito un conjunto de modelos de IGU y sus relaciones a través del marco de referencia de Cameleon (CRF-Cameleon Reference Framework) [70]. En el cual, los modelos de Tarea y Dominio corresponden al primer nivel de CRF, donde se establece una jerarquía de tareas. De esta manera, el establecimiento debe realizarse en un orden lógico para lograr los objetivos de los usuarios. Posteriormente, el modelo Abstract expresa la IGU en términos de Unidades de Interacción Abstracta, que pueden agruparse para asociar tareas relacionadas lógicamente. Siguiendo el modelo Concrete, que expresa la IGU en términos de Unidades de Interacción Concretas y define cómo se percibe la IGU y cómo puede ser manipulada por los usuarios finales. Finalmente, el modelo Final expresa la IGU en términos de código fuente dependiente de la tecnología de implementación que luego puede ser compilado o interpretado [69].

Se han propuesto varios trabajos que combinan el uso de MBUID y patrones de diseño de interacción en el desarrollo de IGU, tales como el de Delgado et al. [71], en donde un MBUID llamado WAINE facilitó a través de la inclusión, la parametrización y las subespecificaciones, la modularidad y la reutilización interna de las especificaciones de la IGU en el momento del desarrollo. Por otro lado, el MBUID llamado PD-MBUI (Pattern-Driven and MBUI) [72], ofrece un enfoque para instanciar y adaptar todos los modelos involucrados en el proceso de diseño de la IGU y no solo los modelos de tareas. PD-MBUI tiene como propósito adaptar y unir en un solo marco los enfoques basados en modelos y los dirigidos por patrones de diseño de interacción, a través del uso de patrones como bloques de construcción y poder crear los modelos adecuados, así como la tarea de uso, el diálogo y la presentación. Para ampliar el concepto de

patrón, Seffah et al. [72] proponen una interfaz para poder personalizar la aplicación en un contexto de uso dado y poder combinar patrones de diseño de interacción. La transformación automatizada entre modelos se realiza a través de archivos XML, que ayudan a adaptar la aplicación y los modelos a diferentes plataformas y usuarios. Sin embargo, la estructura de los patrones de diseño de interacción son definidos por el mismo modelo, además de soportarse en el uso del modelo de tareas de Paterno [73].

Así mismo, Li et al. [74] basado en la descripción de patrones de diseño de interacción, proponen un lenguaje descriptivo de patrones estructurados basado en XML (X-BSPDL - XML-Based structured pattern descriptive language). El marco sugerido se compone de seis módulos para la descripción de patrones de diseño de interacción (ver Tabla 3.1). X-BSPDL apoya en analizar los patrones de diseño de interacción e integrarlos con herramientas automatizadas. De manera, que los desarrolladores pueden liberarse del trabajo laboral, es decir, la escritura de código de IGU.

Tabla 3.1: Esquema XML de X-BSPDL

| Módulo | Descripción |
|---------------|---|
| Estructura | Elementos de IGU de acuerdo con el dispositivo. |
| Datos | Información que se muestra en la IGU. |
| Esquema | Elementos y Funciones de IGU. |
| Estilo | Hoja de estilo de dispositivos específicos. |
| Propiedad | Información de atributos de los controles de IGU. |
| Acción | Información operativa de los controles de IGU. |

Por otro lado, Thanh-Diane et al. [31] proponen un metamodelo que tiene como objetivo expresar patrones de diseño de interacción para IGUs de contextos múltiples. Como parte del metamodelo, el lenguaje de marcado de lenguaje de patrones de interfaz de usuario (UIPLML-User Interface Pattern Language Markup Language) se definió como un lenguaje de marcado compatible con XML para una estructura más formal de

patrones y poder exportar e importar patrones de diseño de interacción. Los patrones denominados patrones de interfaces de usuario multicontexto (MUI-Multimodal User Interface) se expresaron de acuerdo con varias dimensiones: (1) usuario, (2) plataforma o dispositivo, (3) entorno y (4) modelo de interfaz de usuario. Basado en diferentes niveles de abstracción recomendados por la W3C ¹, y por lo tanto compatible con el MBUID. Dentro del software UIPLML se crearon cuatro bases de datos de patrones MUI con un total de 340 entradas (42 para interfaces sensibles al contexto, 237 para sistemas multiplataforma, 52 para accesibilidad y 10 para interfaces culturalmente conscientes). Finalmente, para la consulta del conjunto de datos de patrones MUI, se establecieron seis esquemas: (1) búsqueda de criterio único / clase única, (2) búsqueda de criterio único / clase múltiple, (3) criterio múltiple / clase múltiple búsqueda, (4) búsqueda de texto completo, (5) búsqueda de un solo modelo y (6) búsqueda de varios modelos.

Mientras que Vanderdonck et al. [75] proponen un método para el desarrollo de IGU basado en el uso de un patrón de diseño generativo llamado lenguaje de marcado de lenguaje de patrones (PLML-Pattern Language Markup Language). Un patrón generativo contiene porciones de IGU diseñadas previamente, y que se expresan a través de modelos que se instancian parcial o totalmente. De manera que las porciones en el patrón generativo puedan identificarse y volver a aplicarse a un nuevo estudio de caso de diseño, generando código mediante la instanciación de las especificaciones contenidas en los modelos. Al igual que propuestas anteriores, este método considera la especificación de tareas a través del modelo de tareas de Paterno [73].

Un Modelo de tareas describe las tareas que los usuarios pueden llevar a cabo en las IGUs. Dicho modelo involucra entidades como metas, acciones y objetos de dominio. Las metas especifican cuando se ha alcanzado un estado o meta, las acciones definen los procedimientos para alcanzar dichas metas, y los objetos de dominio representan los

¹World Wide Web Consortium (W3C) - España www.w3c.es

elementos que han de presentarse en la IGU para completar cada tarea en el modelo [73].

Los trabajos descritos anteriormente cubren las fases del desarrollo de sistemas interactivos. Sin embargo, el diseño del modelo de tareas depende de cómo los diseñadores interpreten los requerimientos de software en función de sus conocimientos y experiencia [76]. En lo que respecta al catálogo de diseño de patrones, estos enfoques se basan en una biblioteca de patrones que utiliza una estructura formalizada y predefinida.

3.2. Sistemas de recomendación en la ingeniería de requerimientos

Los Sistemas de Recomendación (SR) tienen un papel importante en la toma de decisiones, ayudando a los usuarios a maximizar las ganancias o minimizar los riesgos. Suelen orientar a los usuarios que no tienen los antecedentes suficientes para evaluar la gran cantidad de alternativas. De acuerdo con Ninaus et al. [77] los SRs apoyan la identificación de requerimientos relacionados en los casos en que la complejidad del conjunto de requerimientos excede la capacidad de un usuario para examinarlo y tomar una decisión. Por lo que investigadores como Mohebzada et al. [78] han introducido los SRs para ayudar mejorar las actividades asociadas a la IR.

En los últimos años los investigadores de los SRs han implementado técnicas de IA y algoritmos de aprendizaje automático, para proporcionar a los usuarios mejores recomendaciones. Las cuales van enfocadas en hacer que la detección de dependencias entre requerimientos sea más inteligente en términos de hacer posible una predicción. Lo que implica un proceso de adquisición de conocimientos, es decir, aprendizaje. Los seres humanos aprenden naturalmente de la experiencia debido a su capacidad para razonar. Por el contrario, las computadoras no aprenden mediante el razonamiento, sino que aprenden a través de la aplicación de algoritmos [79]. En el caso del aprendizaje supervisado, ocurre cuando los algoritmos reciben datos de entrenamiento y respuestas

correctas. La tarea del algoritmo es aprender en función de los datos de entrenamiento y aplicar el conocimiento que se obtuvo utilizando datos reales [80]. Así mismo, el proceso de aprendizaje requiere considerar conceptos relacionados con las áreas de PLN y minería de texto.

La IR se reconoce como un área crítica en la IS porque afecta los riesgos del proyecto cuando se ejecuta de manera deficiente. En muchos casos, existe una brecha entre el software entregado y el esperado por las partes interesadas. La precisión en la captura de los requerimientos del sistema es el mayor factor de falla, con el 90 % de los grandes proyectos de software. Una parte importante de las dificultades para especificar los requerimientos de software se debe a la complejidad de escribir con precisión las frases que detallan los requerimientos [81].

Afortunadamente, ahora están surgiendo nuevas herramientas de análisis de requerimientos basadas en el PLN, las cuales prometen reducir significativamente el costo de corregir los errores de requerimientos al encontrarlos antes y más rápido, y liberar a los expertos del dominio de tareas tediosas y lentas que desperdician su experiencia. Un sistema de PLN procesa datos escritos en lenguaje natural de manera inteligente y genera información que es comparativamente más fácil de analizar. Así mismo, el PLN se ocupa de los esfuerzos combinados de la informática y la lingüística y cómo el lenguaje humano y la computadora interactúan entre sí.

3.2.1. Procesamiento del Lenguaje Natural de los Requerimientos de Software

El PLN es una subárea de la IA, que ha sido aplicada de manera recurrente en la IR debido al uso predominante del lenguaje natural en la especificación de requerimientos y el impacto de éstos en el desarrollo de sistemas interactivos. Así como Pérez-González [82], el cual realizó la identificación de clases utilizando las frecuencias de todos los

sustantivos con base en el texto de los requerimientos de software.

Por otro lado, el análisis sintáctico y el análisis semántico han sido las principales técnicas utilizadas para las tareas de PLN. La sintaxis se refiere a la disposición de las palabras en una oración de modo que tengan sentido gramatical. Mientras que la semántica se refiere al significado que transmite un texto. Kof et al. [83] aborda la aplicación de ingeniería ontológica en la ingeniería de requerimientos, específicamente durante la transición de requerimientos funcionales expresados en lenguaje natural a la fase de diseño de un sistema, utilizando lematización ² y etiquetado.

El PLN es aplicado para clasificar datos en formato textual, por lo que la clasificación de texto es el problema de asignar categorías a los datos de texto de acuerdo a su contenido.

3.2.2. Clasificación de los requerimientos de software

La clasificación de requerimientos de software facilita la toma de decisiones al agrupar una gran cantidad de requerimientos en menos grupos, que principalmente apoyan algunas actividades de IS como análisis de requerimientos, errores, características de software, riesgos de software y pruebas de software [84]. Para algunos investigadores, la clasificación puede ser beneficiosa, ya que puede mejorar la comprensión de los requerimientos del usuario, establecer prioridades y evaluar la calidad de los requerimientos [85].

La clasificación de los requerimientos de software se ha abordado mediante enfoques que se basan en técnicas de extracción de información y PLN, ya que están escritos en lenguaje natural. Sin embargo, se requieren conjuntos de datos de requerimientos para respaldar la capacitación, las pruebas y la validación adecuadas. Uno de los conjuntos de datos más utilizados es el conjunto de datos PROMISE corpus [15], que contiene

²La lematización es el proceso mediante el cual las palabras de un texto que pertenecen a un mismo paradigma flexivo o derivativo son llevadas a una forma normal que representa a toda la clase.

un conjunto de 625 requerimientos etiquetados en 11 clases de RNF y 1 clase de RF, es decir, 12 clases en total. En cuanto a las clases RNF, éstas se refieren a aspectos de la calidad del software (Disponibilidad, Legal, Sentir, Mantenibilidad, Operacional, Desempeño, Escalabilidad, Seguridad, Usabilidad, Tolerancia a fallas y Portabilidad). El trabajo de Licea et al. [44] propuso un método para clasificar oraciones en tipos de requerimientos utilizando redes neuronales convolucionales, y utilizaron el conjunto de datos de corpus PROMISE.

Por otro lado, Kurtanović et al. [6] reportan respecto las 10 características más informativas de los requerimientos de software, con características seleccionadas manualmente que emplean una bolsa de palabras, bigramas y trigramas, y filtran palabras vacías y puntuación. Mientras que Tarawneh et al. [63] presentaron una nueva técnica para la clasificación de requerimientos de software utilizando análisis semántico latente que consiste en la representación de archivos y la medida de similitud de coseno. Rahman et al. [86] procesaron RNF a través de una red neuronal de dos capas que vectoriza palabras (word2vec) para eliminar contenido innecesario y extraer características relevantes. El resultado de este procesamiento se utilizó como entrada de la red neuronal recurrente. Aunque estos trabajos presentan métodos de IA para la clasificación de requerimientos, hay una falta de enfoque en el análisis de los requerimientos de nivel de diseño para el diseño de interfaces. El análisis a nivel de diseño es intensivo en conocimiento porque los diseñadores de IGU deben conocer los principios de diseño, la heurística y los patrones de diseño de interacción para la toma de decisiones de diseño. Tomar malas decisiones de diseño puede tener un fuerte impacto en el desarrollo del sistema interactivo [87].

En la Tabla 3.2 se muestran los trabajos identificados y descritos en esta sección. Sin embargo, estos trabajos son esfuerzos aislados debido a que dificultan la integración de patrones de diseño de interacción en relación con requerimientos a nivel de diseño. Además de que no presentan apoyo en la fase de análisis de requerimientos. En caso

contrario, este trabajo de tesis pretende superar tales limitaciones proponiendo la construcción de un sistema de recomendación basado en la clasificación de requerimientos a nivel de diseño para apoyar a los diseñadores en el proceso de diseño de IGU. De esta manera, en lugar de depender de reglas creadas manualmente como los trabajos relacionados, la clasificación de texto con aprendizaje automático aprende a realizar clasificaciones con base en requerimientos recopilados y aprendidos con anterioridad. Por lo tanto, a continuación se describe cómo el enfoque de este trabajo de tesis puede ayudar a mitigar el problema planteado inicialmente.

Tabla 3.2: Propuestas relacionadas con los requerimientos de software y los patrones de diseño de interacción reportados en la literatura ($A = \text{Análisis}$, $D = \text{Diseño}$, $I = \text{Fases de implementación}$).

| Propuesta | Software | | Patrones de | | Clasificación de Requerimientos | | |
|--------------------------|---------------------|---|---------------------------------------|---|--|---|---|
| | Fases de Desarrollo | Diseño de Interacción | Métodos | Análisis semántico | Conjunto de Datos | Categorías | |
| Soffiah et al. [72] | D, I | 12 Patrones de Diseño | - | - | - | - | - |
| Thanh et al. [31] | D, I | Multiplataforma, Sensible al contexto, Consciente de la cultura, y Patrones de accesibilidad | - | - | - | - | - |
| Vanderdonckt et al. [75] | D, I | Multiplataforma | - | - | - | - | - |
| Gaffar [35] | D, I | Generador de Patrones de Diseño | - | - | - | - | - |
| Licea et al. [44] | A | - | Red Neuronal Convolutional | word2vec | Promise corpus | Funcional, Disponibilidad, Legal, Sentir, Mantenibilidad, Escalabilidad, Seguridad, Usabilidad, Tolerancia a fallos y Portabilidad. Operacional, Rendimiento, | - |
| Kurtanović et al. [6] | A | - | Máquinas de Vectores de Soporte | Natural Language Toolkit NLTK y el Stanford Parser | Opiniones de Amazon software | Funcional; No funcional | - |
| Tarawneh [63] | A | - | Descomposición de valores singulares | Análisis semántico latente | Tera Promise | Funcional, Disponibilidad, Legal, Sentir, Mantenibilidad, Escalabilidad, Seguridad, Usabilidad, Tolerancia a fallos y Portabilidad. Operacional, Rendimiento, | - |
| Rashwan [86] | A | - | Máquinas de Vectores de Soporte | Procesamiento del Lenguaje Natural y extracción de textos | Promise corpus, Concordia corpus | Calidad funcional, externa e interna, restricciones, otros. No Funcionales | - |
| This work | A | 9 Patrones de Diseño de Interacción UI-Patterns | Máquina de Vectores de Soporte Lineal | TF-IDF | Promise corpus, otros conjuntos de datos | No Funcional, Filtro de tabla, Tablero, Ordenar por columna, Controles de transformación, Filtros de Búsqueda, Formato Estructurado, Notificaciones, Formato de Perción, Solicitud de Entrada | - |

3.3. Hallazgos identificados

Después de haber analizado la literatura se ha encontrado una serie de elementos que pueden ser útiles para una propuesta de apoyo a los diseñadores de IGUs, y las cuales pueden ser:

- La falta de adopción de patrones de diseño de interacción por parte de la industria del software. A pesar de las propuestas existentes para el enfoque de MBUID, no se ha encontrado evidencia de que alguna de ellas haya sido adoptada por alguna empresa o fábrica de software. De acuerdo con Rodriguez [88], la escasa capacidad de reutilización de las especificaciones del modelo de interfaz de usuario podría explicar esta falta de interés. Igualmente, Meixner et al. [69] indican que existe una falta de armonización entre las propuestas basadas en modelos, lo que genera ausencia en la industria. Debido a que las propuestas de MBUID se enfocan principalmente, en apoyar en el proceso de desarrollo de sistemas interactivos. Por tanto, será necesario identificar los elementos de información que permitan aprobar los enfoques en apoyo al proceso de diseño de IGUs específicamente.
- Falta de estructura y estandarización de los patrones de diseño de interacción. Las colecciones de patrones de diseño de interacción actualmente existentes, presentan los elementos de información como una lista plana de atributos, que dificulta la interpretación de los patrones de diseño de interacción por parte de los diseñadores de IGUs. Además de encontrar diferentes estructuras en cada una de las colecciones de patrones de diseño de interacción existentes, con base a las especificaciones de cada autor.
- Interpretación de requerimientos de software a través de PLN. Los requerimientos describen lo que el sistema debe realizar, los servicios que debe proveer y las restricciones de su operación. Por lo tanto, son una pieza fundamental en un

proyecto de diseño de sistemas interactivos, ya que marcan su punto de partida y permiten verificar si los objetivos establecidos fueron alcanzados. Por lo anterior, es necesario generar especificaciones de requerimientos correctas, las cuales describan con claridad, sin ambigüedades, en forma consistente y compacta, las necesidades de usuarios.

Los trabajos descritos anteriormente soportan el proceso de desarrollo de sistemas interactivos. Sin embargo, cada una de las propuestas apoyan de manera aislada al diseño de IGUs y en algunos otros casos, no consideran la información referente a dicho proceso. Por lo que, para un proceso de diseño de IGUs, en donde se define la estructura y el comportamiento de sistemas interactivos es necesario apoyar a los diseñadores a un entendimiento y seguimiento de las necesidades de los usuarios con base en los requerimientos establecidos previamente.

Con base en el entendimiento adquirido del proceso de diseño de sistemas interactivos (descrito en el Capítulo 2) y en los hallazgos planteados en este capítulo, se cuenta con elementos que pueden guiar el diseño de un estudio contextual para obtener ideas de diseño para la propuesta del modelo. Por ello, es necesario abordar en el siguiente capítulo las preguntas de investigación de esta tesis: ¿Cómo se puede explotar el establecimiento de requerimientos para el diseño de IGUs?, ¿Cuáles elementos de información de un requerimiento son tomados como referencia para el diseño de IGUs? y ¿Cómo se relacionan los requerimientos con los patrones de diseño de interacción en el proceso de diseño de IGUs?.

3.4. Resumen del capítulo

En este capítulo se ha descrito una serie de trabajos relacionados con el uso de patrones de diseño de interacción e interpretación de requerimientos de software a nivel de diseño de interacción. A partir de dichos trabajos, se han obtenido hallazgos

importantes para guiar una propuesta en apoyo al diseño de IGUs.

En el siguiente capítulo se presenta un estudio contextual para analizar los elementos de información que son considerados por los diseñadores en el proceso de diseño de IGUs y el cual fue diseñado con base en los hallazgos obtenidos de este capítulo.

Capítulo 4

IDENTIFICACIÓN DE PATRONES DE DISEÑO DE INTERACCIÓN

En los capítulos anteriores se estableció en qué consiste el proceso de diseño de IGUs y qué trabajos de investigación contribuyen de alguna manera a apoyar a los diseñadores en las tareas de diseño de interacción.

Como resultado de ello, se estableció un conjunto de hallazgos que motivan la necesidad planteada de analizar los requerimientos de software para poder identificar patrones de diseño de interacción. Por lo anterior, en este capítulo se presenta un caso de estudio que analiza la identificación de patrones de diseño de interacción, a través de la comprensión de los requerimientos funcionales por parte de programadores junior. Este estudio tiene como objetivo entender un proceso de diseño de IGUs basado en la aplicación de patrones de diseño de interacción.

Para esto se diseñaron cuatro actividades compatibles en el proceso de diseño de un proyecto de la materia de Ingeniería de Software B, la cual se imparte para las carreras de Ingeniería en Computación e Ingeniería en Informática que oferta la Facultad de

Ingeniería de la UASLP ¹.

4.1. Participantes

En este caso de estudio fueron invitados a participar veintidós estudiantes (10 mujeres y 12 hombres) de sexto y séptimo semestre de sus respectivas carreras, con una edad promedio de 21 años (ver Tabla 4.1) Estos veintidós estudiantes se integraron en ocho equipos, que trabajaron durante el curso en el diseño y desarrollo de un producto de Software con base en la idea de una primera versión de diseño que fue generada en un curso anterior por ellos mismos (Ingeniería de Software A).

Tabla 4.1: Distribución de los participantes en los equipos de trabajo.

| Equipo | Distribución | Edad promedio |
|---------------|-----------------------|----------------------|
| 1 | 2 hombres y 1 mujer | 22 años |
| 2 | 2 hombres | 21 años |
| 3 | 1 hombre y 2 mujeres | 23 años |
| 4 | 2 hombres y 1 mujer | 22 años |
| 5 | 1 hombre y 2 mujeres | 21 años |
| 6 | 2 mujeres | 22 años |
| 7 | 2 hombres y 2 mujeres | 22 años |
| 8 | 2 hombres | 23 años |

4.2. Actividad

El estudio se realizó en el contexto de la materia de Ingeniería de Software, por lo que se compaginó con la etapa de diseño del proceso de Desarrollo de Software. De manera específica, se realizaron tareas asociadas a las etapas de definición de requeri-

¹<http://www.ingenieria.uaslp.mx/>

mientos, análisis de los requerimientos funcionales a nivel de diseño, establecimiento de requerimientos y el diseño de IGUs en cumplimiento de los requerimientos.

4.3. Herramientas

Para el estudio se consideró la información de diseño previamente generada en un curso anterior (Ingeniería de Software A), la cual incluía la definición del proyecto, la especificación de algunos requerimientos y el lenguaje unificado de modelado (UML- Unified Modeling Language). Sin embargo, para unificar la definición de requerimientos se estableció el uso del formato estándar SRS del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE-The Institute of Electrical and Electronics Engineers) en su estándar 830-1998 para la especificación de requerimientos de software.

De acuerdo con el IEEE [10], el SRS es un conjunto de recomendaciones para la especificación de los requerimientos de software. El cual conjunta en un documento los acuerdos entre el cliente y el grupo de desarrollo, para de esta manera poder tener el listado de requerimientos que se deberán cumplir para dar por aceptado el producto final (ver Fig. 4.1).

Una vez que se establecieron los requerimientos en el formato de SRS, los participantes los analizaron en función de las tareas a realizar, tipo de información relacionada, usuario relacionado y tipo de entorno. Posteriormente, los participantes analizaron la información del catálogo de patrones de diseño de interacción para identificar los asociados con las necesidades expresadas en la información de los requerimientos. A los participantes se les presentó un conjunto de patrones de diseño de interacción de una colección en línea del autor Anders Toxboe ². Dicho catálogo considera los patrones de diseño de interacción más usuales para la plataforma web, bajo la cual los participantes generarían sus proyectos.

²<http://ui-patterns.com/patterns>

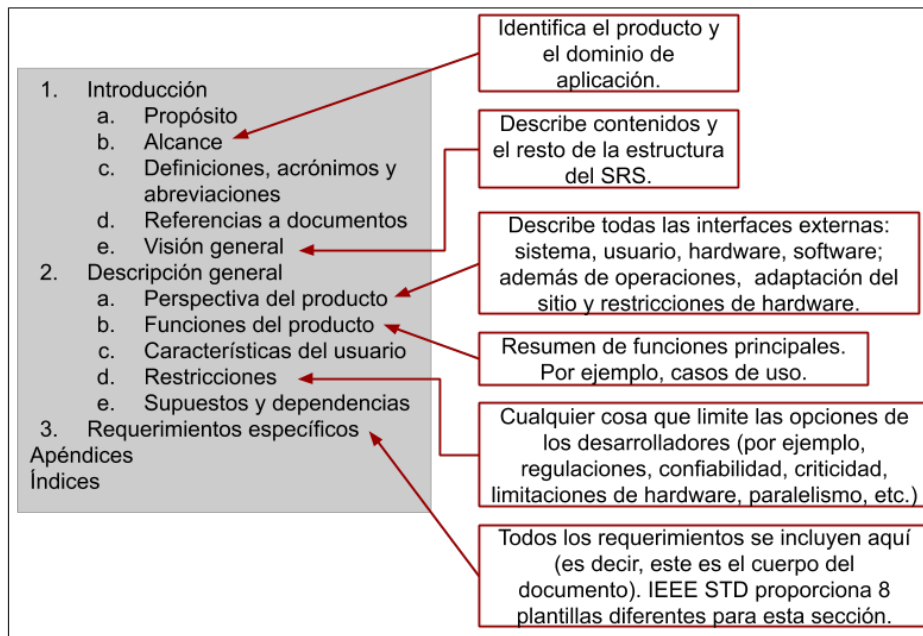


Figura 4.1: Contenido del formato SRS establecido por el IEEE [10]

Cada una de las sesiones fueron documentadas con fotografías y grabaciones de audio de las entrevistas con los participantes. De manera que la información recolectada, fuera analizada posteriormente para la identificación de hallazgos para las primeras ideas de diseño de la solución al problema de investigación de este trabajo de tesis.

4.4. Sesiones del Caso de Estudio

Para este caso de estudio se realizaron cuatro sesiones de tres horas cada una, las cuales se describen a continuación.

4.4.1. Primera Sesión

Con el propósito de organizar y estandarizar el proceso de documentación de proyectos de Software. Se propuso el uso de los estándares establecidos por el IEEE y el estándar 830-1998 para IS [10]. El documento SRS sustenta la documentación de los

requerimientos acordados con el supervisor del proyecto, estableciendo las bases para el diseño y para la formación del propio sistema.

Para una mejor comprensión, a los programadores Junior se les presentó un ejemplo práctico para que pudieran asociarlo con sus propios proyectos. La Figura 4.2 muestra un ejemplo de las primeras versiones de los documentos SRS generados por los programadores Junior.

Características de los usuarios
 A continuación, se muestran los tipos de usuario que estarán manejando el sistema, así como de los privilegios y requerimientos que llevara a cabo en el sistema:

| Usuarios | Privilegios | Requerimientos |
|---------------|--|--|
| Administrador | Dar de alta y baja a usuarios y acceso al historial. | REQ_001, REQ_002, REQ_004, REQ_007, REQ_008, REQ_009 |
| Secretaria | Agendar citas, agregar y visualizar información básica del paciente. | REQ_001, REQ_003, REQ_004, REQ_005, REQ_006 |
| Dentista | Agendar citas, visualización de expediente del paciente | REQ_001, REQ_004, REQ_007, REQ_009, REQ_009 |

Especificación de requerimientos

a) Requerimientos Funcionales

Manejo de usuarios

REQ_001 "Acceso al Usuario"
 El sistema identificará a los usuarios que accedan al sistema con los siguientes roles: Administrador, dentista o secretaria, a través de un usuario o clave y contraseña.

REQ_002 "Gestión de Usuario"
 El sistema proporcionará una interfaz para el mantenimiento de usuarios, y las actividades que se pueden realizar son las siguientes:

- **Baja de un usuario:** En la cual se solicitará la clave o el nombre para una búsqueda rápida y así poder darlo de baja.
- **Alta de un usuario:** Para dar de alta un usuario se necesitan los siguientes datos: una clave de usuario que el sistema generará automáticamente, el nombre, la dirección los cuales son obligatorios a llenar, en cambio otros datos como el teléfono o celular y e-mail pueden ser dejados en blanco, así como también una contraseña que será dada por el administrador, al dar de alta se le otorgarán los permisos necesarios para que pueda realizar sus actividades correspondientes.
- **Modificación de un Usuario:** Al realizar una modificación se puede buscar el usuario con clave o nombre y al darle aceptar se cargarán los datos en pantalla para poder hacer el cambio, los datos que no puedan ser cambiados estarán inhabilitados.

Figura 4.2: Generación de documento SRS por programadores Junior

4.4.2. Segunda Sesión

La sesión dio inicio con una presentación sobre la definición y origen de los patrones de diseño de interacción, así como la explicación de la manera que son utilizados los patrones de diseño de interacción en el diseño de IGUs. Dicha presentación le dejó en claro a los participantes, que los patrones de diseño de interacción capturan las propie-

dades del buen diseño a través de los elementos de información comunes en todas las instancias de la solución.

Una vez que los participantes se pusieron en contexto, se les presentaron ejemplos con base en algunos de los requerimientos definidos por algunos equipos. Lo anterior, con el propósito de brindarles una mejor comprensión de cómo analizar los requerimientos y relacionarlos con los estándares de diseño de interacción proporcionados de referencia.

Posteriormente, se mostraron a los programadores Junior las colecciones de patrones de diseño de interacción más representativas, tales como el conjunto de patrones de diseño de interacción de Jenifer Tidwell [12]. Además se mostraron otras colecciones de patrones de diseño de interacción, como los presentados por Martijn van Welie [26], y los enfoques basados en los patrones de Borchers [25], etc. Estas colecciones representativas fueron presentadas a los programadores Junior con el propósito de que sean conscientes de la importancia de conocer y aplicar patrones de diseño de interacción, además del ahorro de tiempo que se puede obtener en resolver problemas que ya tienen solución.

Al finalizar la sesión, se mostró a los programadores Junior un catálogo en línea de patrones de diseño de interacción, para que conocieran la estructura e información que se brinda para cada uno de los patrones de diseño de interacción. Así mismo, se mostraron algunos ejemplos en los que se identificaron patrones de diseño de interacción con base en el análisis de los requerimientos previamente establecidos en el formato SRS.

4.4.3. Tercera sesión

Una vez que los participantes tuvieron claro el proceso para identificar patrones de diseño de interacción en función de los requerimientos funcionales, se les solicitó a ellos mismos repetir los pasos para identificar los patrones de diseño de interacción correspondientes y aplicarlos en el diseño de las IGUs de su propio proyecto (ver Fig. 4.3). Posteriormente, los prototipos generados por los participantes, fueron sometidos a una evaluación heurística.

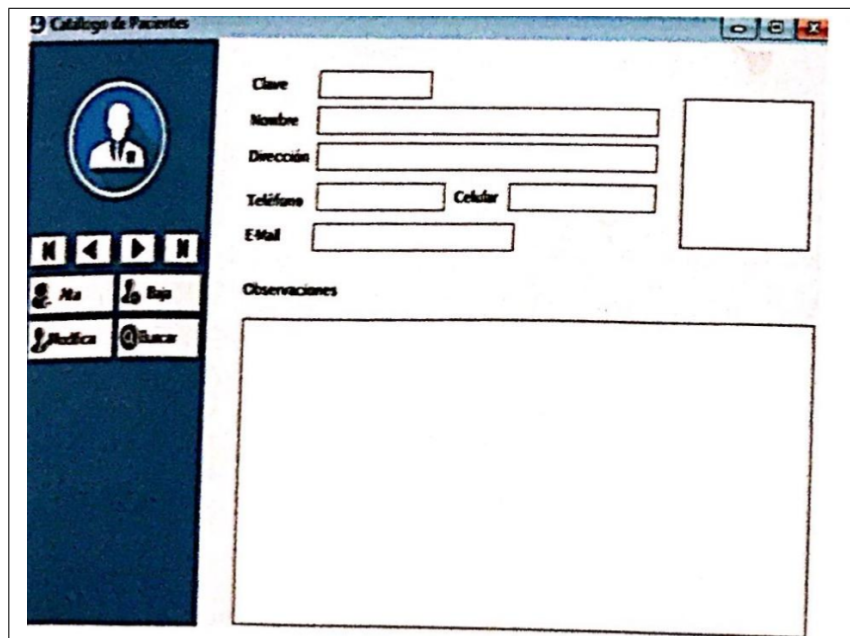


Figura 4.3: Aplicación de patrones de diseño de interacción en el diseño de prototipos por programadores Junior

Una Evaluación Heurística es un método de inspección de usabilidad sin usuarios, que consiste en examinar la calidad de uso de una IGU por evaluadores expertos que verifican el cumplimiento de cada una de las heurísticas. Para ello, se tomó como referencia un conjunto de preguntas relacionadas con cada una de las reglas heurísticas (ver Tabla 4.2). En este caso de estudio, la evaluación se enfocó en elementos específicos para la búsqueda de problemas de usabilidad, siguiendo las diez reglas heurísticas de Nielsen [18].

La evaluación heurística permite detectar errores de usabilidad sin necesidad de tener usuarios probando los diseños. Cada una de las reglas heurísticas fue verificado su cumplimiento en función de los grados de gravedad, los cuales pueden variar en una escala de severidad de cero a cuatro (ver Tabla 4.3). Siendo cero el valor más bajo que indica que no se considera un problema de usabilidad en absoluto. Mientras que cuatro es el valor más alto y el cual indica una catástrofe de usabilidad que se requiere

Tabla 4.2: Preguntas asociadas a cada una de las reglas heurísticas de Nielsen [13]

| Heurística | Pregunta |
|--|--|
| H1 - Visibilidad del estado del sistema | ¿Se informa a los usuarios sobre el avance del sistema con la respuesta adecuada y en un tiempo aceptable? |
| H2 - Coincidencia entre el sistema y el mundo real | ¿El sistema utiliza conceptos y lenguaje familiar para los usuarios en lugar de términos técnicos? ¿El sistema utiliza convenciones del mundo real y presenta la información de forma natural y en orden lógico? |
| H3 - Control y libertad del usuario | ¿Pueden los usuarios hacer lo que quieran y cuando quieran? |
| H4 - Coherencia y estándares | ¿Los elementos de diseño, como objetos y acciones, tienen el mismo significado o efecto en diferentes situaciones? |
| H5 - Prevención de errores | ¿Podrían los usuarios cometer errores que no harían en mejores interfaces? |
| H6 - Reconocimiento en lugar de recordar | ¿Son visibles los elementos del proyecto, como objetos, acciones y opciones? ¿El usuario se ve obligado a recordar información de una parte a otra del sistema? |
| H7 - Flexibilidad y eficiencia de uso | ¿Los métodos de las tareas son eficientes? ¿Pueden los usuarios personalizar acciones frecuentes o atajos? |
| H8 - Diseño estético y minimalista | ¿Los diálogos contienen información irrelevante o poco utilizada? |
| H9 - Ayude a los usuarios a reconocer, diagnosticar y recuperarse de errores | ¿Los mensajes de error se expresan en un lenguaje sencillo (sin códigos)? ¿Describen con precisión el problema y sugieren una solución? |
| H10 - Ayuda y documentación | ¿Se proporciona un apoyo adecuado? ¿Es esa información fácil de encontrar y enfocada en la tarea de los usuarios? |

Tabla 4.3: Escala de severidad atribuida en la evaluación de usabilidad, con base en las diez reglas Heurística Nielsen [14].

| Severidad | Significado |
|-----------|--|
| 0 | No estoy de acuerdo en que este sea un problema de usabilidad en absoluto. |
| 1 | Solo problema cosmético: no es necesario solucionarlo a menos que haya tiempo adicional disponible en el proyecto. |
| 2 | Problema menor de usabilidad: la solución de este problema debe tener poca prioridad. |
| 3 | Problema de usabilidad importante: es importante solucionarlo, por lo que se le debe dar una alta prioridad. |
| 4 | Catástrofe de usabilidad: imperativo solucionar este problema antes de que se pueda lanzar el producto. |

solucionar antes del lanzamiento del producto.

4.4.4. Cuarta Sesión

Finalmente, se realizó una entrevista estructurada a cada equipo con el objetivo de conocer la percepción de los programadores Junior sobre la identificación de patrones de diseño de interacción en función de los requerimientos funcionales (ver Fig. 4.4), la cual constaba de 14 preguntas que se organizaban en cuatro partes: datos demográficos, conocimientos previos, percepción de facilidad de uso y percepción de utilidad.

4.5. Resultados

En este caso de estudio, la identificación de patrones de diseño de interacción se analizó mediante la comprensión de los requerimientos funcionales por parte de los

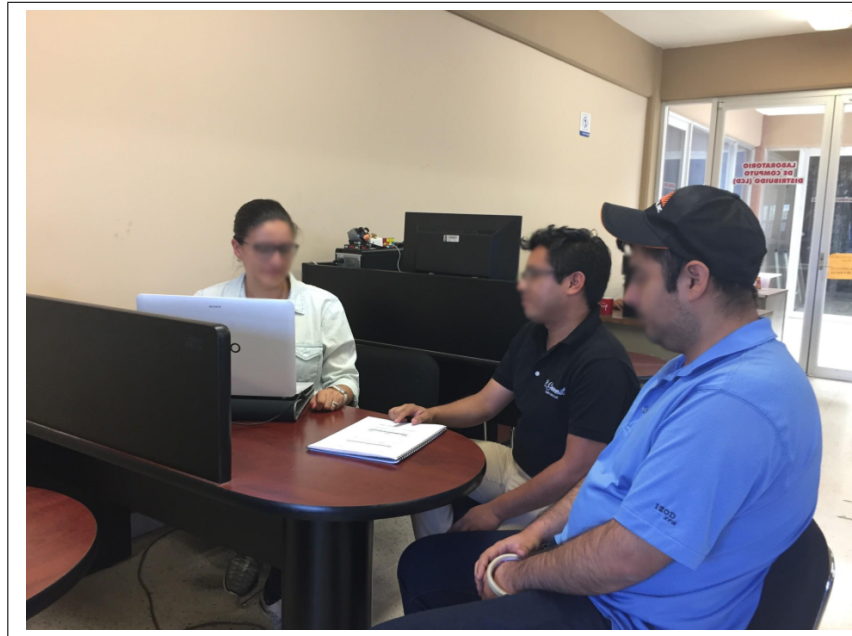


Figura 4.4: Entrevista estructurada a programadores Junior

programadores Junior. Para lo cual, se distribuyeron las cuatro sesiones a lo largo de cuatro meses, dentro de las cuales se consideraron los temas que se imparten en el curso de Ingeniería de Software B.

Debido a la identificación de los patrones de diseño de interacción y su aplicación, los programadores Junior generaron diseños de prototipos que posteriormente fueron evaluados con base en las 10 heurísticas de Nielsen [18]. La evaluación resultó en errores heurísticos de los ocho equipos (ver Tabla 4.4).

Los resultados mostraron que las heurísticas “Visibilidad del estado del sistema”, “Coincidencia entre el sistema y el mundo real”, “Consistencia y estándares” y “Estética y diseño minimalista” fueron las reglas heurísticas evaluadas con menor severidad. Así lo confirma lo expresado en la entrevista por parte de los programadores Junior, donde en relación con la heurística “Visibilidad del estado del sistema”, se indicó que el patrón de diseño de interacción con el que estaban más familiarizados era el de retroalimentación.

“La retroalimentación sobre el inicio de sesión... bueno, fue donde vimos más uso...”

Tabla 4.4: Resultados de la evaluación de las IGUs diseñadas por cada equipo con base en las heurísticas de Nielsen

| Heurística | Severidad por Equipo | | | | | | | | Media |
|--|----------------------|---|---|---|---|---|---|---|-------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | |
| H1 - Visibilidad del estado del sistema | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 |
| H2 - Coincidencia entre el sistema y el mundo real | 1 | 1 | 1 | 1 | 2 | 3 | 2 | 2 | 2 |
| H3 - Control y libertad del usuario | 1 | 3 | 3 | 2 | 3 | 4 | 3 | 3 | 3 |
| H4 - Coherencia y estándares | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | 1 |
| H5 - Prevención de errores | 2 | 2 | 2 | 3 | 3 | 3 | 2 | 3 | 3 |
| H6 - Reconocimiento en lugar de recordar | 1 | 2 | 2 | 1 | 2 | 3 | 3 | 3 | 2 |
| H7 - Flexibilidad y eficiencia de uso | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| H8 - Diseño estético y minimalista | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 1 |
| H9 - Ayude a los usuarios a reconocer, diagnosticar y recuperarse de errores | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| H10 - Ayuda y documentación | 2 | 2 | 2 | 2 | 2 | 3 | 2 | 2 | 2 |

porque decía que era como la retroalimentación de sí mismo, comenzó la sesión correctamente o algo faltaba o estaba mal... o algo así". Equipo 1

Otro aspecto que influyó en que los programadores Junior diseñaron IGUs consistentes fue que se estableció una base de diseño para las principales acciones para la gestión de catálogos, lo que restaba tiempo en la toma de decisiones respecto al diseño en este aspecto.

Por otro lado, las heurísticas “Control y libertad del usuario”, “Prevención de errores” y “Flexibilidad y eficiencia de uso” fueron las reglas heurísticas evaluadas con mayor severidad. Como se menciona en los siguientes comentarios de los Equipos 3 y 8, en los cuales se resalta la complejidad de establecer requerimientos en el documento SRS, dada su poca experiencia.

“Bueno, creo que realmente se trataba de identificar dónde iban a estar los elementos... por ejemplo, en la parte de... no era, uh... por ejemplo, en un requerimiento no decía nada más que... registro de usuario, pero bueno... Aún faltan datos que íbamos a poner todo eso”. Equipo 3

“Bueno... sí, algo complicado porque tuvimos que hacer varios cambios, porque no los definimos bien desde el principio”. Equipo 8

Los problemas de usabilidad relacionados con estas reglas heurísticas pueden disminuir la interacción entre el sistema y el usuario. Por tanto, los diseñadores deben estar atentos al desarrollo de las IGUs para que sean intuitivas y accesibles, incluso para usuarios inexpertos, evitando esfuerzos extra para aprender a utilizar el sistema. Estos problemas se pueden asociar con patrones de diseño de interacción para relacionar el conocimiento de diseño de interacción con casos de uso de IS [89]. Sin embargo, éstos raras veces han sido utilizados explícitamente por diseñadores expertos.

Los problemas de comprensión del concepto de patrones de diseño de interacción, provocan que los desarrolladores utilicen incorrectamente los patrones de diseño de interacción, lo cual queda claro en algunos de los comentarios realizados por programadores Junior.

“La falta de experiencia, es decir, nunca lo habíamos visto... entonces si estuviéramos más ocupados buscando detalles e investigando cosas que no entendíamos...”. Equipo 7

“Bueno... más o menos complicado porque nunca habíamos trabajado con patrones... nunca habíamos tenido un estándar que nos pusieran para el diseño de software”. Equipo 8

Sin embargo, los programadores Junior tienen claros los beneficios que conlleva la aplicación de patrones de diseño de interacción, por lo que su percepción sobre la utilidad de los mismos fue positiva. Además de eso, manifestaron interés en seguir aplicándolos en sus proyectos futuros.

“Sí, definitivamente los usaremos (los patrones de diseño de interacción)... porque está más ordenado, y es más fácil detectar los errores que puedan ocurrir”. Equipo 2

“Sí, porque te da una guía a seguir (patrones de diseño de interacción) y... es más fácil concentrarte en qué hacer y cómo programarlo”. Equipo 5

4.6. Discusión

Cuando un diseñador de IGUs hace uso de los patrones de diseño de interacción, generalmente encuentra que son múltiples, simplistas y abstractos, lo que complica

la selección e interpretación de los mismos. Además de que al utilizarlos, resulta en una difícil percepción del problema que se quiere resolver, en el cumplimiento de lo establecido por los requerimientos.

Por lo tanto, el análisis de los requerimientos es un proceso complejo que depende de la toma de decisiones por parte de los diseñadores de IGUs. Sin embargo, la falta de un proceso adecuado de definición y comprensión del problema y la definición poco clara de las necesidades del cliente, generan problemas a lo largo del proceso de desarrollo, entre los que se encuentran los problemas en el diseño de IGU.

Basado en el problema anterior, este caso de estudio fue diseñado para recopilar la percepción de los programadores Junior que identificarían y aplicarían patrones de diseño de interacción en el diseño de IGUs. Para lo cual, previamente se brindó información relacionada con la definición de requerimientos, así como información sobre los patrones de diseño de interacción. Las percepciones obtenidas de estas actividades serán consideradas para la clasificación de requerimientos de Software y la generación de un modelo.

Se esperaba que los programadores Junior pudieran establecer requerimientos funcionales bien definidos desde las primeras versiones del SRS. Sin embargo, la falta de experiencia y práctica en estos temas hizo que tuvieran varias iteraciones en la definición de los requerimientos. Además, se notó confusión en los programadores Junior al distinguir los patrones de diseño de interacción y los patrones de diseño orientados a objetos.

Al final de manera general, los programadores Junior estaban convencidos de la importancia de establecer bien los requerimientos funcionales y los beneficios de aplicar los patrones de diseño de interacción de manera efectiva. Sin embargo, si los programadores principiantes aprenden a establecer correctamente los requerimientos de software y utilizan con frecuencia patrones de diseño de interacción, mejorarán enormemente la identificación y selección de patrones de diseño de interacción.

La IR sigue siendo un desafío importante para la industria del desarrollo de software, por un lado debido a la ambigüedad del proceso de Desarrollo de Software, y por otro lado por el desafío de obtener requerimientos precisos por parte del cliente y usuarios. Por lo tanto, es necesario llevar a cabo un proceso continuo de unión de los procesos internos de una empresa desarrolladora, con el objetivo del cumplimiento de las necesidades del cliente y usuarios, las cuales son establecidas a través de los requerimientos de software expresados en un lenguaje natural.

En el caso de proyectos grandes con muchos requerimientos, se requerirá mucho tiempo y esfuerzo [44]. Posteriormente, se explorarán líneas de investigación para evitar el esfuerzo humano de crear y analizar las características de los datos, lo que conducirá a una identificación y selección eficiente y efectiva de patrones de diseño de interacción.

4.7. Resumen del capítulo

En este capítulo se han presentado los resultados de un caso de estudio para analizar la identificación y aplicación de patrones de diseño de interacción, a través del establecimiento y análisis de requerimientos de software. El estudio fue realizado con programadores Junior en cuatro etapas, las cuales generaban como producto final el diseño de IGUs. Posteriormente las IGUs fueron evaluadas con base en las reglas heurísticas de Nielsen.

Los resultados de este caso de estudio dan respuesta a las preguntas de investigación de tesis: ¿Cómo se puede explotar el establecimiento de requerimientos para el diseño de IGUs?, ¿Cuáles elementos de información de un requerimiento son tomados como referencia para el diseño de IGUs? y ¿Cómo se relacionan los requerimientos con los patrones de diseño de interacción en el proceso de diseño de IGUs?. Debido a la importancia de establecer de manera clara y precisa los requerimientos de software, para poder llevar a cabo una interpretación de las necesidades de los clientes y usuarios. Por

lo que los participantes del estudio tomaron como base elementos de información como: actor, acción y objeto, como los elementos de información básicos para la interpretación de requerimientos a nivel de diseño. De manera que sea posible la reutilización de soluciones a problemas de diseño recurrentes, a través de la identificación y aplicación de los patrones de diseño de interacción. Dichas respuestas dan indicios que motivan el diseño de un modelo de recomendación de patrones de diseño de interacción con base en el análisis de requerimientos de software. Por lo tanto, es necesario abordar en el siguiente capítulo la pregunta de investigación: ¿Cómo se pueden utilizar los enfoques de aprendizaje automático para identificar patrones de diseño de interacción con base en los requerimientos?. Además, se describen los detalles del diseño del modelo propuesto.

Capítulo 5

MODELO DE RECOMENDACIÓN DE PATRONES DE DISEÑO DE INTERACCIÓN: UIPatternM.

En el capítulo anterior se presentó un caso de estudio, que tuvo como objetivo analizar el proceso de identificación de patrones de diseño de interacción, a través de la comprensión del texto de requerimientos funcionales. Como resultado se obtuvieron los hallazgos, tales como la importancia de la información contenida en la especificación de requerimientos, que su nivel de interpretación depende de la experiencia del diseñador de IGUs y los retos que se enfrentan al tratarse de información textual. Por lo tanto, se trabajó en una propuesta de modelo de recomendación de patrones de diseño de interacción para apoyo en las tareas de diseño de IGUs.

Los diseñadores de IGUs enfrentan una falta de apoyo en el proceso de tomar decisiones de diseño de interacción, debido a que son los únicos responsables de transformar los requerimientos a nivel de diseño en decisiones de diseño de interacción rentables. Por lo que si las decisiones de diseño resultaran inadecuadas, como consecuencia el sistema de software resultaría afectado en última instancia en términos de usabilidad (por

ejemplo, eficiencia del usuario, efectividad y experiencia del usuario).

Por lo tanto, en este capítulo se describe de manera detallada el modelo propuesto para relacionar los patrones de diseño de interacción con base en los requerimientos a nivel de diseño. Además de los detalles de la aplicación de técnicas de IA para dicho propósito.

5.1. Patrones de diseño de interacción en el diseño de Interfaces de Usuarios

Como se mencionó en el Capítulo 3, los patrones de diseño de interacción son soluciones de diseño a problemas recurrentes, y que permiten identificar características de las situaciones y dichos problemas, a través de sus elementos de información (problema, solución y contexto). Por lo tanto, la reutilización de soluciones de diseño basadas en patrones de diseño de interacción, en lugar de crear nuevas soluciones desde cero permite a los diseñadores novatos y expertos crear buenas IGUs para sistemas interactivos, las cuales pueden mejorar la experiencia de usuario.

En el diseño de IGUs, los diseñadores suelen considerar varios factores al diseñar un sistema interactivo con el objetivo de cumplir las demandas de los usuarios para usos específicos. Los patrones de diseño de interacción ofrecen la oportunidad de mantener las necesidades y expectativas de los usuarios en las primeras etapas de diseño al mismo tiempo que permiten unificar conceptos provenientes de diferentes áreas de estudio como los factores humanos, la ingeniería y el diseño de interacción, con el fin de garantizar un nivel óptimo de aceptación [90]. Los patrones de diseño de interacción no quitan la libertad creativa del proceso de diseño para instanciar diseños relevantes en el contexto del informe de diseño particular. Algunas propuestas proponen procesos de identificación de patrones de diseño de interacción, tales como [31, 35, 72, 75], los cuales han identificado los patrones de diseño de interacción en relación con las tareas

principalmente y para lo cual se basan en el modelo de tareas de Calvary [69].

Los modelos de tareas describen las actividades lógicas que deben llevarse a cabo para cumplir las necesidades de los usuarios, por lo tanto, los modelos de tarea son útiles al diseñar y desarrollar sistemas interactivos. Las actividades son descripciones con un nivel de detalle, las cuales se basan en los requerimientos.

5.2. Características de los Requerimientos de Software

El ciclo de vida del desarrollo de software inicia con la obtención de los requerimientos del cliente y usuarios, que posteriormente se detallan en el formato SRS [43]. A medida que los clientes utilizan su propio idioma para establecer sus requerimientos, se observa que a menudo es ambiguo y confuso para que el equipo de desarrollo analice y posteriormente diseñe la solución deseada. Por ello, se requiere de la comprensión de lo que tiene que hacer el sistema, cómo debe ser y cómo debe encajar con otras cosas: con los requerimientos del producto, sistema o servicio. Los diseñadores deben investigar la variedad de usuarios, actividades y contextos relevantes para el dominio que están investigando para que puedan comprender los requerimientos (tanto funcionales como no funcionales) del sistema que están desarrollando. Los requerimientos funcionales están relacionados con lo que el sistema debería poder hacer y con las limitaciones funcionales de un sistema. Por lo que es importante que el diseñador piense en toda la experiencia de interacción de forma abstracta. Decidir quién hace qué, cuándo se debe mostrar algo o la secuencia en la que se llevan a cabo las acciones debe venir más adelante en el proceso de diseño [5].

Para lograr un entendimiento de los requerimientos, es necesario que se encuentren claramente escritos, es decir, que contengan toda la información necesaria para que el diseñador los comprenda. Cada requerimiento debe describir con precisión una capacidad

que cumpla las necesidades de algunas partes interesadas y debe describir claramente la funcionalidad que se va a construir. De acuerdo con Wiegers [91], se recomienda escribir los requerimientos funcionales desde la perspectiva de lo que hace el sistema o de lo que el usuario puede hacer. Indicando los requerimientos de manera coherente, como “El sistema debe” o “El usuario deberá”, seguido de un verbo de acción, seguido del resultado observable sobre un objeto. Así mismo, se requiere especificar la acción o condición que hace que el sistema realice un comportamiento en específico.

En el caso de estudio del Capítulo 4, se identificó la importancia de establecer requerimientos bien definidos; sin embargo, el nivel de experiencia de los diseñadores repercute en la manera en la que se definen los mismos. Por lo cual, en este trabajo de tesis se propone un modelo de recomendación, que se basa en el ingreso de requerimientos en formato textual. A partir de los cuales, es posible la extracción de información relacionada con las tareas que deberán de realizar los sistemas interactivos.

5.3. Modelo de recomendación de Patrones de Diseño de Interacción

En la Figura 5.1, se muestra el modelo de recomendación llamado Modelo de Patrón de Interfaz de Usuario (UIPatternM - Model of User Interface Pattern) que se propone en este trabajo de tesis, desde un punto de vista funcional. El modelo UIPatternM parte de un conjunto de *requerimientos de software* como entrada, de manera que el módulo *Editor de texto de requerimientos* los valide en términos de semántica y sintáctica. Posteriormente, una vez que se garantiza que el texto del requerimiento tiene congruencia textual, el módulo *Clasificación de requerimientos* predice a que tipo de requerimiento corresponde (RF o RNF). Por ejemplo, el requerimiento # 3 (R3): “Un usuario podrá ingresar información sobre un incidente, incluyendo ubicación, descripción y período de tiempo” se clasifica como RF, ya que describe una función que deberá realizar el usuario

en el sistema interactivo.

El conjunto de RFs a nivel de diseño que dio como resultado el módulo *Clasificación de requerimientos*, se ingresan al módulo de *Recomendación de patrones de diseño de interacción* para ser asociados con los patrones de diseño de interacción. Como resultado se obtiene la predicción de un patrón de diseño de interacción para cada uno de los requerimientos ingresados inicialmente. En el caso del requerimiento de nivel de diseño mencionado anteriormente, la predicción es un “Patrón de formato estructurado”. Dicho patrón de diseño de interacción aborda el problema de que “el usuario necesita ingresar datos rápidamente en el sistema, pero el formato de los datos debe adherirse a una estructura predefinida” (ejemplo de [16]). Dicha predicción representa la salida del modelo UIPatternM, la cual se deberá presentar al diseñador de IGUs para que apoye a su toma de decisiones de diseño de IGUs.

A continuación, se describen los módulos que comprende el modelo UIPatternM, para extraer los elementos de los requerimientos de software (ver Figura 5.2). Los módulos descritos son: Editor de texto de requerimientos, Clasificación de requerimientos y Recomendación de patrones de diseño de interacción.

5.3.1. Módulo editor de texto de requerimientos

Los requerimientos de nivel de diseño se expresan con mayor frecuencia en lenguaje natural. No obstante, el análisis de requerimientos puede verse afectado por cuestiones como la ambigüedad, que da lugar a diversas interpretaciones. La ambigüedad puede ser léxica, sintáctica, semántica, pragmática, vaga, general y ambigua en el lenguaje error [92]. Por esta razón, los requerimientos a nivel de diseño deben procesarse previamente. En este trabajo de tesis, se consideró un conjunto de datos de instancias de requerimientos a nivel de diseño (es decir, *requerimiento de software*) como una entrada en el módulo *Editor de texto de requerimientos*, que verifica que los elementos de información (es decir, el usuario, la acción y los objetos) son gramaticalmente adecuados.

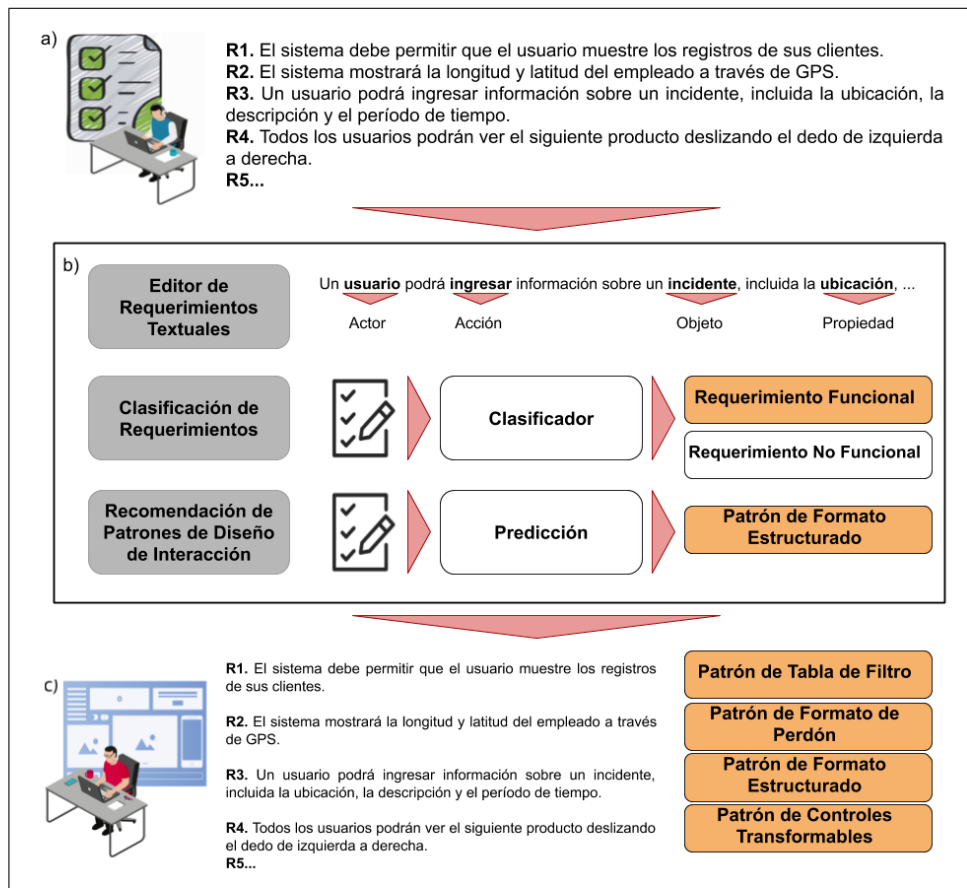


Figura 5.1: Operación del *Modelo de recomendación de patrones de diseño de interacción UIPatternM*.

En el Capítulo 4 se detalló el caso de estudio contextual, en el cual se determinó que la identificación de patrones de diseño de interacción se analizó mediante la comprensión de los requerimientos a nivel de diseño. Considerando las ventajas de la PLN y las representaciones formales, el módulo *Editor de Requerimientos Textuales* incluye una verificación de texto basada en una tarea de análisis semántico propuesta por Diamantopoulos et al. [11], que empleó un método de extracción de relaciones basado en características que permite implementar una canalización de análisis basada en el etiquetado de roles semánticos (ver Figura 5.3). Por ejemplo, dado el siguiente requerimiento a nivel de diseño: “El sistema debe mostrar eventos en una tabla vertical por

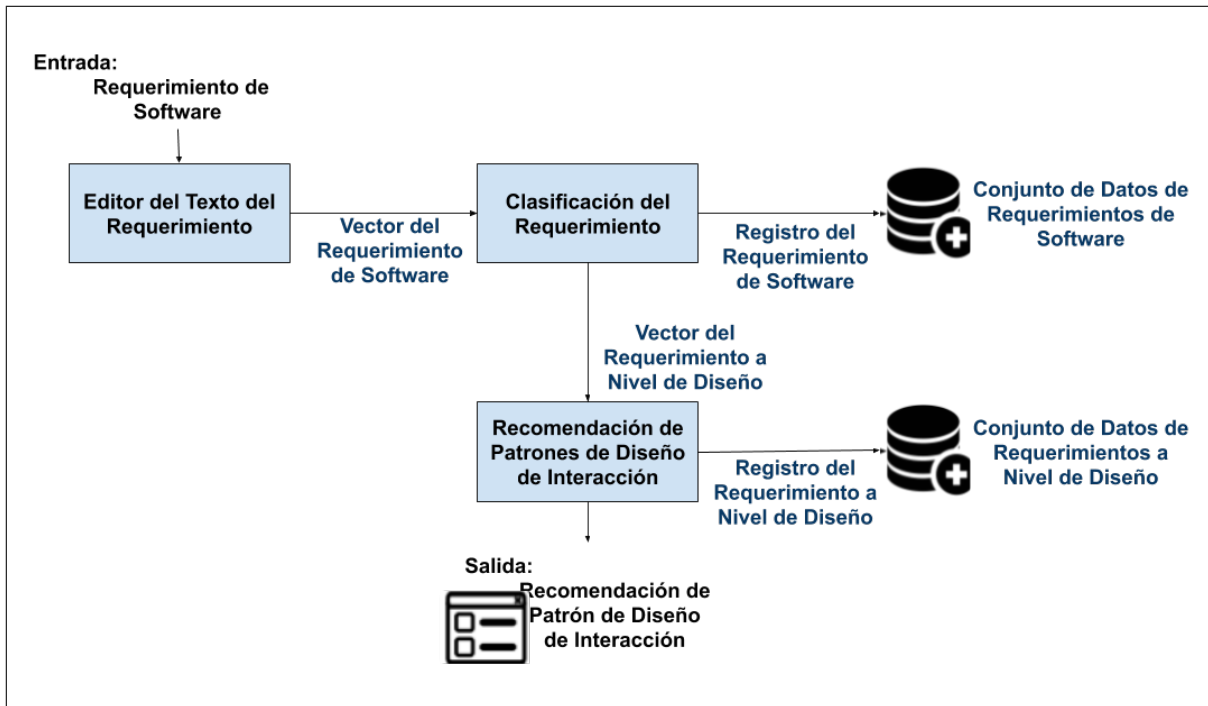


Figura 5.2: Modelo de recomendación de patrón de diseño de interacción basado en requerimientos a nivel de diseño UIPatternM.

tiempo”. La oración incluye: “sistema”, “tabla”, “eventos” como tipo de cosa y “display” como tipo de operación. La estructura de estas propiedades es similar a la forma en que se estructuran las oraciones. La definición de entidades es posible a través de una ontología, que almacena información derivada de requerimientos funcionales. Una ontología proporciona un medio estructurado para organizar la información, admite métodos para recuperar datos almacenados a través de consultas y permite razonar sobre las relaciones implícitas entre los elementos de datos.

Siguiendo la Figura 5.3, para la extracción de las instancias de los conceptos Actor, Acción, Objeto y Propiedad. Actor se refiere a los actores del proyecto, incluidos los usuarios, el sistema en sí o cualquier sistema externo. Las instancias de tipo Objeto incluyen cualquier objeto o recurso del sistema que recibe alguna acción, mientras que las instancias de Propiedad incluyen todos los modificadores de objetos o acciones. La

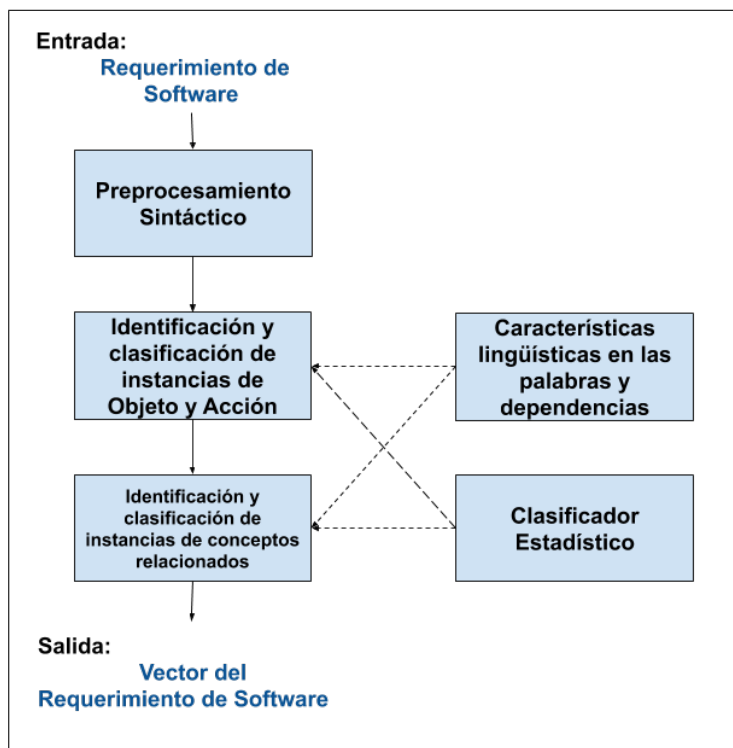


Figura 5.3: Arquitectura de canalización de Diamantopoulos [11] para analizar los requerimientos de software en el módulo *Editor de Requerimientos Textuales*.

propiedad expresa posesión (por ejemplo, “cada usuario tiene una cuenta”) y Acción describe una operación realizada por un Actor en algún Objeto. La arquitectura de Diamantopoulos Pipeline [11] aprende a mapear automáticamente los requerimientos de software escritos en textos de lenguaje natural para conceptos y relaciones definidos en una ontología. La anotación de requerimientos permite especificar términos (o frases) como una de las entidades Actor, Acción, Objeto o Propiedad. Para las relaciones entre estos términos, [11] definió tres relaciones: *IsActorOf*, *ActsOn* y *HasProperty*. *IsActorOf* se declara de Actor a Acción, y *ActsOn* se define de Acción a Objeto o de Acción a Propiedad. Finalmente, *HasProperty* se declara de Actor a Propiedad, o de Objeto a Propiedad, o de Propiedad a Propiedad (ver Figura 5.4).

Con base en el análisis semántico de representaciones fundamentadas en ontologías del texto [11], se preprocesó un conjunto de requerimientos de software. Este preproce-

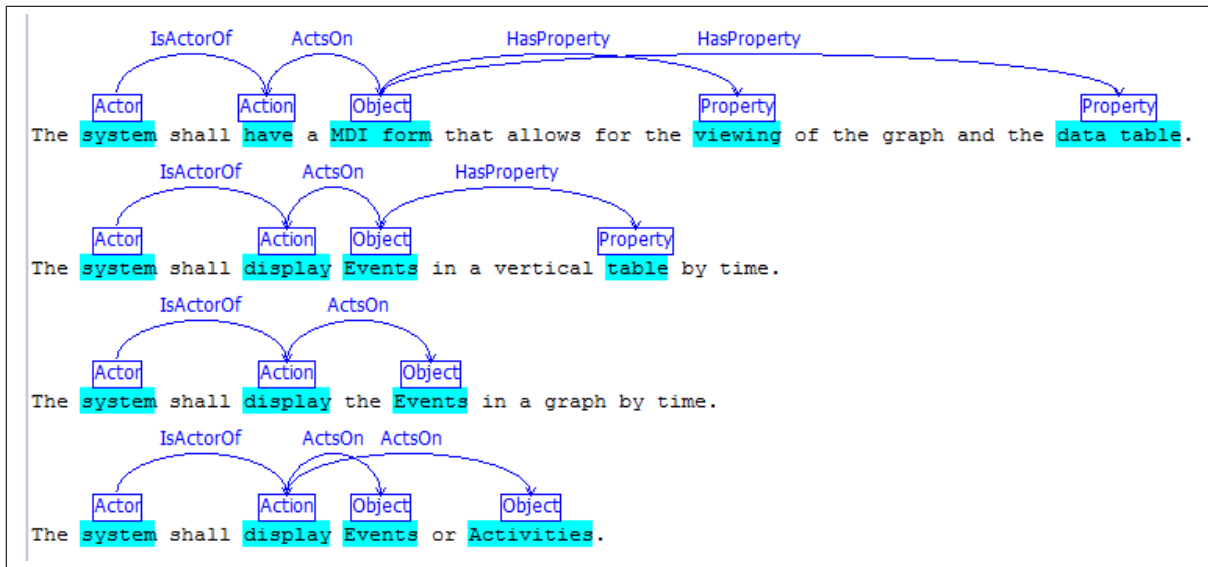


Figura 5.4: Procesamiento de un requerimiento textual con un enfoque de tarea de análisis semántico [11]

samiento tiene como objetivo identificar oraciones ambiguas y ayudar al diseñador de la interfaz de usuario a comprender por qué son ambiguas, para que puedan eliminarse y, por lo tanto, clasificar mejor los requerimientos a nivel de diseño.

5.3.2. Módulo de clasificación de requerimientos

En IS, la clasificación es la disposición de los requerimientos de software en diferentes clases para que puedan proporcionar un apoyo significativo para la toma de decisiones y otros procesos analíticos. Los requerimientos de software pueden ser funcionales o no funcionales. Los requerimientos no funcionales pueden ser más complejos de diseñar y probar, ya que obtener un diseño que los cumpla a menudo lleva tiempo. En comparación, los requerimientos funcionales pueden ser más sencillos.

En grandes proyectos de software con numerosos requerimientos, se requerirá un tiempo y esfuerzo considerable. Una práctica común es usar humanos para clasificarlos en RF y RNF, ya que en la mayoría de los casos están escritos en lenguaje natural. En

cambio, aunque es una tarea compleja para la ingeniería de requerimientos, es posible utilizar técnicas de aprendizaje automático como PLN [93]. En este trabajo de tesis, se propone un módulo de *Clasificación de requerimientos* (vea la Figura 5.5).

Para entrenar el *Algoritmo de clasificación*, primero es necesario transformar el texto del requerimiento en un vector de números. Posteriormente las representaciones de vectores se exploraron como vectores ponderados de Término de Frecuencia Inversa del Documento (TF-IDF - Term frequency-Inverse document frequency). Esta técnica proporciona una puntuación que representa el término frecuencia multiplicado por la frecuencia inversa del documento [94, 95]. La ponderación TF-IDF se usa comúnmente en la extracción de texto y la recuperación de información para evaluar la importancia de un término lingüístico (generalmente unigrama o bigrama) en un conjunto de datos.

La importancia del término (es decir, el peso) aumenta con la frecuencia del término en el texto, pero se compensa con la frecuencia del término en el dominio de interés (por ejemplo, palabras frecuentes como “el” o “para”). Para este propósito, se usaron algunos registros del corpus PROMISE, que es uno de los conjuntos de datos más utilizados para los requerimientos de software [15] (consulte la Tabla 5.1 para ver algunos ejemplos del conjunto de datos). Este conjunto de datos utiliza requerimientos categorizados para entrenar a un clasificador de texto mediante un algoritmo de aprendizaje supervisado. Así, el principal aspecto a evaluar es el desempeño de este tipo de clasificadores en la categorización de un conjunto de requerimientos textuales.

5.3.3. Módulo de recomendación de patrones de diseño de interacción

Los componentes principales de los sistemas de recomendación basados en contenido son el preprocesamiento, el aprendizaje y la predicción en línea. Dado que los sistemas basados en contenido funcionan con una amplia variedad de conocimientos, estos dife-

Tabla 5.1: Un conjunto de muestra de requerimientos de software que se encuentran en el conjunto de datos corpus PROMISE [15]. La columna Clase puede ser: Availability (Disponibilidad) = A, Look and feel (Apariencia) = LF, Performance (Desempeño) = PE, Usability (Usabilidad) = US, y Portability (Portabilidad) = PO.

| ID del Proyecto | Requerimiento de Software | Clase |
|-----------------|--|-------|
| 1 | El sistema actualizará la pantalla cada 60 segundos. | PE |
| 1 | La solicitud coincidirá con el color del esquema establecido por el Departamento de Seguridad Nacional. | LF |
| 1 | Si se proyectan, los datos deben ser legibles. En una pantalla de proyección de 10x10, el 90 % de los espectadores deben poder leer los datos de eventos/actividades desde una distancia de visualización de 30. | US |
| 1 | El producto estará disponible durante el horario laboral normal. Siempre que el usuario tenga acceso al PC cliente, el sistema estará disponible el 99 % del tiempo durante los primeros seis meses de funcionamiento. | A |

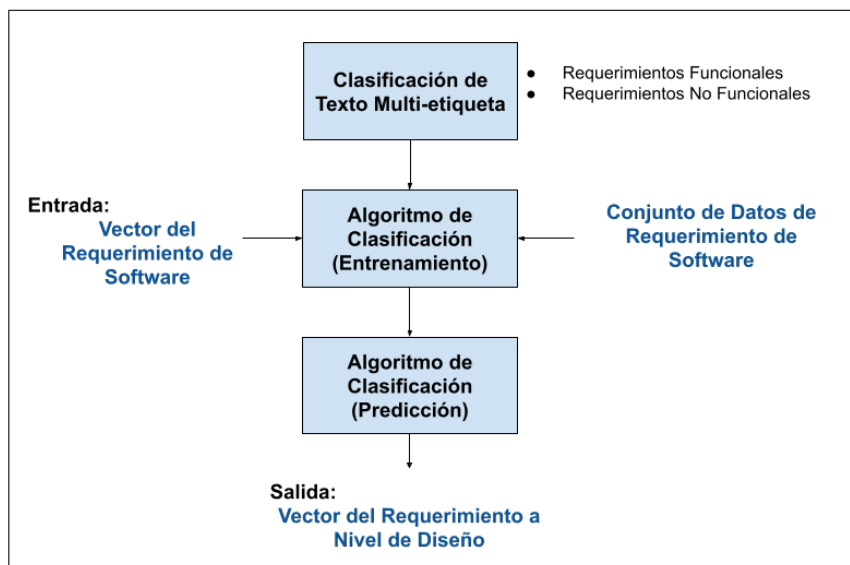


Figura 5.5: Arquitectura del módulo de Clasificación de requerimientos.

rentes tipos de datos no estructurados deben convertirse en información estandarizada. Por lo tanto, los sistemas basados en contenido operan principalmente en el dominio de texto [96].

La reutilización de patrones de diseño de interacción se considera especialmente adecuada para acelerar el diseño de la interfaz de usuario, pero también porque los patrones de diseño de interacción son soluciones comprobadas a problemas que surgen en diversos contextos y proyectos. Además de fomentar la reutilización de las mejores prácticas, mejora la comunicación entre las partes interesadas y permite a los diseñadores menos experimentados producir mejores propuestas de diseño [50, 97]. Por lo tanto, se considera un módulo de *Recomendación de patrones de diseño de interacción*, el cual predice los patrones de diseño de interacción asociados con un conjunto de requerimientos a nivel de diseño. El módulo *Recomendación de patrones de diseño de interacción* utiliza una técnica de filtrado basado en contenido, cuyo objetivo es etiquetar los requerimientos de nivel de diseño utilizando ciertas palabras clave para comprender qué acción se realizará en la IGU (ver la Figura 5.6). Posteriormente, esas palabras clave se buscan en el conjunto de datos y se predice un patrón de diseño de interacción que aborde

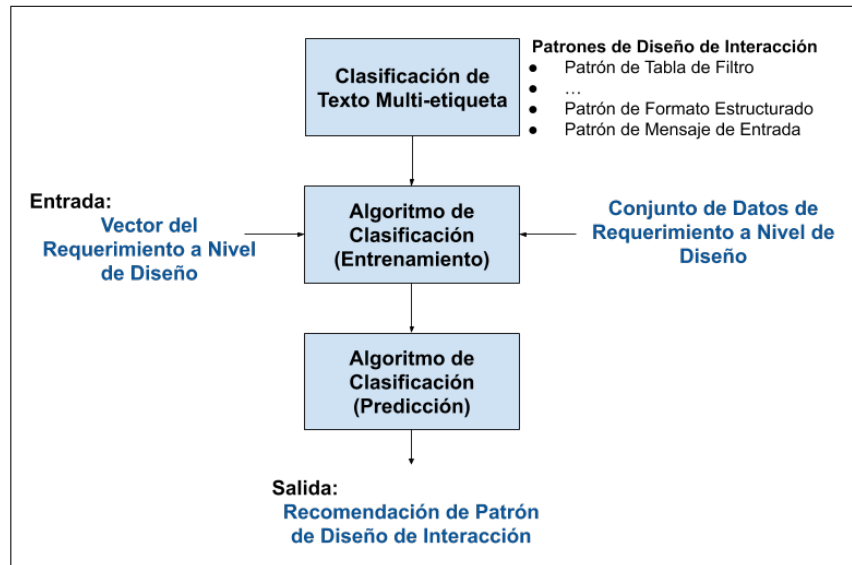


Figura 5.6: Arquitectura del módulo de recomendación de patrones de diseño de interacción.

un problema de diseño similar. Un módulo *Recomendación de patrones de diseño de interacción* aplica la técnica TF-IDF para transformar el texto del requerimiento en un vector de números, como en el módulo *Clasificación de requerimientos*.

5.4. Resumen del capítulo

En este capítulo se ha proporcionado una descripción del modelo de recomendación de patrones de diseño de interacción, el cual se propuso con base en los hallazgos identificados en el estudio contextual del capítulo 4. Como parte de los resultados de este capítulo se presentan:

- Se dio respuesta a la pregunta de investigación: ¿Cómo se pueden utilizar los enfoques de aprendizaje automático para identificar patrones de diseño de interacción con base en los requerimientos?. Debido a que se aplicaron técnicas de PLN para extraer información de los requerimientos funcionales a nivel de diseño a partir de

su formato en texto. Además de la aplicación de algoritmos de clasificación para la predicción de patrones de diseño de interacción.

- Se determinó como punto de entrada al modelo UIPatternM las especificaciones de requerimientos de software, dado que son parte fundamental en el proceso de diseño y desarrollo de sistemas interactivos. Dado que a partir de los requerimientos de software es posible inferir información relevante para el diseño de IGUs. Sin embargo, tuvo que ser considerado el nivel de interpretación por parte de los diseñadores de IGUs, debido a que éste depende entre otros aspectos, de la experiencia de los diseñadores en el diseño de IGUs.
- Se definió el modelo de recomendación de patrones de diseño de interacción (UIPatternM) con base en el análisis de requerimientos funcionales a nivel de diseño.
- Se generó un conjunto de datos de instancias de requerimientos de software, a través de la recolección de diferentes corpus y fuentes de información de proyectos de sistemas interactivos.

En el siguiente capítulo se presenta la evaluación del modelo de recomendación UIPatternM, en términos del rendimiento de los algoritmos de clasificación aplicados por el mismo, además de la evaluación de utilidad por parte de un grupo de diseñadores de IGUs profesionales. Para lo cual será necesario abordar la siguiente pregunta de investigación: ¿Cuál es el impacto del diseño de la IGU con base en la recomendación del patrón de diseño de interacción?.

Capítulo 6

EVALUACIÓN DEL MODELO DE RECOMENDACIÓN UIPatternM.

En el capítulo anterior se detalló el modelo de recomendación de patrones de diseño de interacción llamado UIPatternM, el cual con base en el análisis del texto de requerimientos a nivel de diseño proporciona apoyo a los diseñadores de IGUs en la toma de decisiones de diseño.

En este capítulo se presenta la evaluación de dicho modelo de recomendación de patrones de diseño de interacción desde la perspectiva del rendimiento de los algoritmos de clasificación aplicados. Así como la obtención de la percepción por parte de un grupo de diseñadores de IGUs profesionales. Con base en lo cual, se demuestra que la recomendación de patrones de diseño de interacción asociados a un requerimiento a nivel de diseño específico, apoya a los diseñadores de IGUs a tomar decisiones de diseño precisas en cumplimiento de factores de calidad como la usabilidad y la experiencia de usuario.

6.1. Evaluación de desempeño de los Algoritmos de Clasificación

El rendimiento del módulo *Recomendación de patrones de diseño de interacción* se obtuvo comparando cuatro algoritmos de aprendizaje automático: (1) Regresión logística, (2) Bayes Ingenuo multinomial, (3) Máquina de vectores de soporte lineal y (4) Bosque aleatorio [98, 99]. Estos algoritmos se corroboraron mediante una validación cruzada de 10 veces [100].

Uno de los algoritmos de clasificación más simples es el de Regresión Logística, que ha sido abordado en el dominio de la minería de datos. Mientras que en la historia más temprana de la recuperación de información como una aplicación factible, el clasificador Bayes Ingenuo multinomial fue muy popular. El clasificador Bayes Ingenuo multinomial es computacionalmente económico y también necesita una cantidad muy baja de memoria. Por otro lado, se han utilizado técnicas no paramétricas como tareas de clasificación de Máquina de vectores de soporte lineal, la cual es una técnica que también se puede utilizar en todos los dominios de la minería de datos, como la bioinformática, la imagen, el vídeo, la clasificación de la actividad humana, la seguridad y la protección, etc. Así mismo, se han estudiado los clasificadores con base en árboles, como el bosque aleatorio para la categorización de documentos [101].

6.1.1. Métricas de Rendimiento

La exactitud, la precisión, la recuperación y la puntuación F1 se utilizan como métricas de rendimiento de los algoritmos de clasificación. A continuación, se describen cada una de las métricas en términos de verdaderos positivos (TP-True Positives) y falsos positivos (FP-False Positives), y verdaderos negativos (TN-True Negatives) y falsos negativos (FN-False Negatives).

Exactitud una proporción de textos predichos correctamente al total de textos.

$$Exactitud = \frac{TP + TN}{TP + TN + FP + FN} \quad (6.1)$$

Precisión es la capacidad de un clasificador de no etiquetar una instancia como positiva que en realidad es negativa. Para cada clase, se define como la relación entre verdaderos positivos y la suma de verdaderos y falsos positivos.

$$Precisión = \frac{TP}{TP + FP} \quad (6.2)$$

Recuperación es la capacidad de un clasificador para encontrar todas las instancias positivas. Para cada clase, se define como la proporción de verdaderos positivos a la suma de verdaderos positivos y falsos negativos.

$$Recuperación = \frac{TP}{FN + TP} \quad (6.3)$$

Puntuación F1 es una media armónica ponderada de precisión y recuperación tal que la mejor puntuación es 1.0 y la peor es 0.0. En términos generales, los puntajes F1 son más bajos que las medidas de precisión, ya que incorporan precisión y recuerdan en su cálculo. Como regla general, el promedio ponderado de F1 debe usarse para comparar modelos de clasificadores, no la precisión global.

$$Puntuación F1 = 2 \times \left(\frac{precisión \times recuperación}{precisión + recuperación} \right) \quad (6.4)$$

6.1.2. Conjuntos de Datos

Se creó un conjunto de datos compuesto por 307 requerimientos etiquetados a nivel de diseño. El conjunto de datos se creó en función de los requerimientos de nivel de diseño que se identificaron y recopilaron de algunos conjuntos de datos de proyectos de software como [15,102]. Los requerimientos a nivel de diseño del conjunto de datos pertenecen a 9 clases diferentes, que corresponden a 9 patrones de diseño de interacción de la colección Toxboe [16] (ver Tabla 6.1). El conjunto de datos se dividió aleatoriamente en entrenamiento (75 %) y conjunto de prueba (25 %).

Tabla 6.1: Un conjunto de muestra de patrones de diseño de interacción que se encuentran en la colección Toxboe [16].

| Patrón de Diseño de Interacción | Problema |
|--|---|
| Filtro de tabla | El usuario necesita filtrar categóricamente los datos mostrados en tablas por columnas. |
| Tablero | El usuario desea asimilar datos de varias fuentes de un vistazo. |
| Ordenar por columna | El usuario debe poder ordenar los datos en una tabla de acuerdo con los valores de una columna. |
| Controles Transformables | El usuario solo quiere que se le presenten los controles disponibles para el modo actual. |
| Filtros de búsqueda | Los usuarios deben realizar una búsqueda utilizando filtros contextuales que reducen los resultados de la búsqueda. |
| Formato estructurado | El usuario necesita ingresar datos en el sistema rápidamente, pero el formato de los datos debe adherirse a una estructura predefinida. |
| Notificaciones | El usuario desea estar informado sobre actualizaciones y mensajes esenciales. |
| Formato de tolerancia | El usuario necesita ingresar datos en el sistema rápidamente, que luego a su vez, interpreta la entrada de los usuarios. |
| Solicitud de entrada | El usuario debe ingresar datos en el sistema. |

6.1.3. Extracción de características

Como primera etapa se llevó a cabo un proceso de extracción de características, el cual es un método que se utiliza para transformar cada requerimiento en una representación numérica en forma de vector. Posteriormente, cada algoritmo de aprendizaje automático se entrenó con el conjunto de datos creado y la clase (es decir, patrones de diseño de interacción). Para que después el algoritmo de aprendizaje automático entrenado, fuera utilizado para hacer predicciones utilizando requerimientos de software basados en texto. Dicho extractor de características es utilizado para transformar el texto de requerimientos a nivel de diseño en conjuntos de características, que son utilizadas como entrada en el algoritmo de aprendizaje automático para obtener predicciones de patrones de diseño de interacción.

Una vez que se completa la extracción de características basada en TF-IDF, se aplica el método de selección de características de estadísticas de chi-cuadrado. Este método de Chi-cuadrado es de selección de características que resulta muy útil en la categorización de texto, además de que refleja la asociación entre una característica y clase. La prueba TF-IDF determina si la asociación entre dos variables categóricas refleja su asociación real en la población [103]. Siguiendo la Tabla 6.2, los unigramas y bigramas obtenidos demuestran la congruencia que existe de la definición del problema de cada patrón de diseño de interacción con cada texto de requerimiento funcional. Para ello, es necesario contar con un conjunto de datos confiable, es decir, con clasificaciones de requerimientos funcionales validadas por diseñadores de sistemas interactivos capacitados.

6.1.4. Entrenamiento y prueba de algoritmos

Como resultado del preprocesamiento del conjunto de datos, se obtuvieron todas las características y etiquetas. Posteriormente, se entrenaron cuatro algoritmos de clasificación (regresión logística, Bayes Ingenuo multinomial, máquina de vectores de soporte

Tabla 6.2: El conjunto de unigramas y bigramas obtenidos por clase.

| Clase | Unigramas | Bigramas |
|--------------------------|-------------------------|--|
| Filtro de tabla | resumen imprimible | resumen imprimible capaz de desplegar |
| Tablero | desplegar eventos | proporcionará deberá mostrar |
| Ordenar por columna | reciclado auditoría | informe de auditoría deberá informar |
| Controles Transformables | evitar usuarios | listado de jugadores permitir a los jugadores |
| Filtros de búsqueda | marcadores buscar | permitir a los usuarios deberá buscar |
| Formato estructurado | borrar clínico | capaz de crear capaz de borrar |
| Notificaciones | cambios notificar | notificar a los afectados deberá notificar |
| Formato de tolerancia | lavado funcionalidad | guías de lavado funcionalidad de lavado |
| Solicitud de entrada | registro permitir | permitirá deberá registrar |

lineal y bosque aleatorio) para predecir patrones de diseño de interacción. Para entrenar y probar los algoritmos de clasificación fue utilizada la librería Scikit-Learn sobre el entorno de desarrollo Python.

Una vez que el 75 % del conjunto de datos se aplicó para la etapa de entrenamiento, se tomó como referencia para predecir las etiquetas de clase (patrones de diseño de interacción) para el conjunto de datos de prueba (25 % restante). Por lo tanto, se ejecutó la tarea de clasificación en la que el sistema de recomendación propuesto procesaba el texto de un requerimiento y tiene que predecir a qué patrón de diseño de interacción se asocia.

6.1.5. Resultados

Una vez ejecutados los algoritmos de clasificación, la máquina de vectores de soporte lineal resultó con una ligera ventaja en términos de precisión, resultando con una media de alrededor de 45 % (ver tabla 6.3). En comparación con Bayes Ingenuo, el algoritmo Máquina de vectores de soporte lineal superó la eficacia de clasificación [104]. Debido a esto, los siguientes experimentos se basaron en el algoritmo Máquina de vectores de soporte lineal.

Tabla 6.3: Rendimiento global de los algoritmos de clasificación.

| Algoritmos | Exactitud | Precisión | Recuperación | Puntuación F1 |
|---------------------------------------|-------------|-------------|--------------|------------------|
| Regresión logística | 0.40 | 0.53 | 0.53 | 0.49 |
| Bayes Ingenuo multinomial | 0.43 | 0.43 | 0.49 | 0.44 |
| Máquina de vectores de soporte lineal | 0.45 | 0.58 | 0.54 | 0.52 |
| Bosque aleatorio | 0.33 | 0.55 | 0.39 | 0.33 |

Tabla 6.4: Rendimiento del algoritmo Máquina de vectores de soporte lineal.

| Patrón de Diseño de Interacción | Precisión | Recuperación | Puntuación F1 |
|---------------------------------|-------------|--------------|---------------|
| Filtro de tabla | 0.22 | 0.40 | 0.29 |
| Tablero | 1.00 | 0.20 | 0.33 |
| Ordenar por columna | 0.40 | 0.29 | 0.33 |
| Controles Transformables | 0.44 | 0.67 | 0.53 |
| Filtros de búsqueda | 0.50 | 0.11 | 0.18 |
| Formato estructurado | 0.67 | 0.87 | 0.75 |
| Notificaciones | 0.80 | 0.50 | 0.62 |
| Formato de tolerancia | 0.50 | 0.60 | 0.55 |
| Solicitud de entrada | 0.71 | 0.38 | 0.50 |
| Promedio / Total | 0.59 | 0.54 | 0.52 |

La Tabla 6.4 muestra los resultados obtenidos con el algoritmo Máquina de vectores de soporte lineal. Precisión, Recuperación y Puntuación F1 son valores medios de todas las predicciones de requerimientos a nivel de diseño. De éstos, el 59% de los datos se predijeron con precisión. Los patrones de diseño de interacción "Tablero" (100%) y "Notificación" (80%) obtenidos fueron la mejor predicción. Además, la Tabla 6.5 muestra la matriz de confusión del algoritmo Máquina de vectores de soporte lineal por clase. La mayoría de las predicciones se encuentran en la diagonal, es decir, verdaderos positivos. Sin embargo, hay casos en los que la recuperación es baja, como la clase *Filtros de búsqueda* con solo uno (1) verdadero positivo (recuperación = 0.11).

Tabla 6-5: Matriz de confusión del algoritmo Máquina de vectores de soporte lineal en la Clasificación de requerimientos.

| | Filtro de tabla | Tablero | Ordenar por columna | Controles Transformables | Filtros de búsqueda | Formato estructurado | Notificaciones | Formato de tolerancia | Solicitud de entrada |
|--------------------------|-----------------|---------|---------------------|--------------------------|---------------------|----------------------|----------------|-----------------------|----------------------|
| Filtro de tabla | 2 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 |
| Tablero | 3 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| Ordenar por columna | 1 | 0 | 2 | 1 | 1 | 0 | 0 | 2 | 0 |
| Controles Transformables | 1 | 0 | 0 | 12 | 0 | 4 | 0 | 0 | 1 |
| Filtros de búsqueda | 2 | 0 | 2 | 2 | 1 | 1 | 0 | 0 | 1 |
| Formato estructurado | 0 | 0 | 0 | 2 | 0 | 20 | 1 | 0 | 0 |
| Notificaciones | 0 | 0 | 0 | 2 | 0 | 1 | 4 | 1 | 0 |
| Formato de tolerancia | 0 | 0 | 0 | 2 | 0 | 2 | 0 | 6 | 0 |
| Solicitud de entrada | 0 | 0 | 0 | 3 | 0 | 2 | 0 | 3 | 5 |

6.2. Evaluación de utilidad del Modelo de Recomendación de Patrones de Diseño de Interacción

El modelo UIPatternM a partir de la especificación textual de requerimientos de software a nivel de diseño, genera como resultado una predicción de patrones de diseño de interacción. El modelo UIPatternM asocia instancias de requerimientos de software con patrones de diseño de interacción, a través del entrenamiento de un clasificador de texto que aplica algoritmos de clasificación. Lo anterior con el objetivo de apoyar a los diseñadores de IGUs en la toma de decisiones de diseño. Por lo tanto, es necesario obtener el punto de vista de diseñadores profesionales en el contexto del diseño de IGUs de sistemas interactivos, para lo cual se llevó a cabo una evaluación que consistió en cuatro etapas, con el objetivo de determinar la viabilidad del concepto presentado.

6.2.1. Objetivos del estudio

Los siguientes son los objetivos que guiaron el estudio:

- La utilidad percibida de la aplicación del modelo de recomendación (UIPatternM) para respaldar las decisiones de diseño en el diseño de interacción de sistemas interactivos.
- La efectividad percibida de la recomendación de patrones de diseño de interacción con base en el modelo UIPatternM, para respaldar las decisiones de diseño en el proceso de diseño de sistemas interactivos.

6.2.2. Hipótesis

En el diseño de interacción el desafío es diseñar IGUs que requieran una atención mínima y un esfuerzo cognitivo de los usuarios para comprender la información presentada. Además, las necesidades explícitas deben ser percibidas como estéticas y naturalmente integradas en su entorno. Sin embargo, la selección de patrones de diseño de interacción es subjetiva y propensa a errores. Por lo tanto, al utilizar una herramienta de recomendación con base en el modelo UIPatternM, es posible obtener recomendaciones de patrones de diseño de interacción relacionados con las necesidades de los usuarios. En consecuencia, y para enmarcar el estudio, se establecieron las siguientes hipótesis de trabajo:

- H1: ¿Cuál es la opinión de los diseñadores de interacción profesionales, respecto a la presentación de la visión tecnológica de un recomendador de patrones de diseño de interacción?
- H2: ¿Cuál es el porcentaje de coincidencia de la predicción emitida por el modelo de recomendación de patrones de diseño de interacción, respecto a la selección de patrones de diseño de interacción por parte de los diseñadores de interacción profesionales?
- H3: ¿Los diseñadores de interacción profesionales consideran útil la visión de la herramienta de recomendación con base en el modelo UIPatternM, para apoyar a los diseñadores de IGUs?
- H4: ¿Los diseñadores de interacción profesionales consideran efectiva la visión de la herramienta de recomendación con base en el modelo UIPatternM, para apoyar a los diseñadores de IGUs?

6.2.3. Diseño del estudio

Este estudio consistió en un diseño *within-subjects*, debido a que todos los tratamientos se aplicaron a todos los participantes. La primera parte consistió en una sesión de formación sobre los nueve patrones de diseño de interacción, durante la cual todos los participantes se situaron mentalmente en un patrón de diseño de interacción de la colección de Toxboes. Posteriormente, los participantes tuvieron que seleccionar un patrón de diseño apropiado para un conjunto de diez requerimientos a nivel de diseño en formato textual. Los participantes tuvieron la libertad de seleccionar el patrón de diseño de interacción y dar respuestas a tres preguntas posteriores. Para la etapa de recomendación del patrón de diseño de interacción, los participantes tuvieron que validar un conjunto de diez requerimientos a nivel de diseño procesados y clasificados previamente. En la última etapa, los participantes debían responder los cuestionarios del TAM [105] y el SUS [106].

6.2.4. Participantes

Para garantizar la participación de los diseñadores de interacción a nivel profesional, fue publicada una convocatoria en redes sociales. Los diseñadores interesados, previamente respondieron un cuestionario de selección, posteriormente se les informó que podían participar en el estudio completo. Los participantes fueron 14 diseñadores profesionales de diferentes empresas de México y Ecuador (ver Tabla 6.6). Todos ellos participaban en proyectos de diseño interactivo para diversas áreas (educativo, sanitario, Internet de las cosas, administración en general, etc.). Su edad promedio era de 32 años, con cinco años de experiencia en promedio. Todos ellos habían concluido su licenciatura (BSc) relacionada con áreas de IHC.

Tabla 6.6: Información de los participantes en la evaluación del modelo UIPatternM.

| Identificador | Sexo | Edad | Años de Experiencia |
|----------------------|-------------|-------------|----------------------------|
| Participante 1 | Mujer | 32 años | 2 años |
| Participante 2 | Mujer | 33 años | 4 años |
| Participante 3 | Hombre | 37 años | 9 años |
| Participante 4 | Hombre | 32 años | 9 años |
| Participante 5 | Mujer | 31 años | 1 año |
| Participante 6 | Hombre | 42 años | 5 años |
| Participante 7 | Hombre | 32 años | 5 años |
| Participante 8 | Mujer | 29 años | 3 años |
| Participante 9 | Mujer | 27 años | 6 años |
| Participante 10 | Hombre | 29 años | 5 años |
| Participante 11 | Hombre | 31 años | 4 años |
| Participante 12 | Mujer | 29 años | 5 años |
| Participante 13 | Hombre | 28 años | 7 años |
| Participante 14 | Hombre | 32 años | 8 años |

6.2.5. Instrumentos

A continuación, se presentan los instrumentos desarrollados para realizar este estudio.

1. Escenario de uso. Este escenario estaba relacionado con un diseñador de sistemas interactivos que trabajaba con un equipo multidisciplinario, en un proyecto relacionado con el desarrollo de un sistema interactivo (sistema de servicios turísticos). A partir de los requerimientos recopilados, el diseñador tenía que generar propuestas de diseño de interacción. Sin embargo, el diseñador se enfrentó a la confusión de identificar qué patrones de diseño de interacción implementar en su diseño, por lo que tuvo que consultar una herramienta con base en el modelo de recomendación para obtener una predicción de patrones de diseño de interacción fundamentada en el conocimiento de proyectos anteriores.
2. Presentación de patrones de diseño de interacción. Cada participante tuvo que revisar nueve patrones de diseño de interacción de la colección de Toxboes.
3. Cuestionario para seleccionar patrones de diseño de interacción. Cada participante tuvo que analizar diez requerimientos a nivel de diseño y seleccionar el patrón de diseño de interacción apropiado. Se preparó un cuestionario utilizando una escala Likert-5 y se centró en la comprensión del requerimiento a nivel de diseño (ver Apéndice A).
4. Cuestionario para validar las predicciones de patrones de diseño de interacción. Cada participante tuvo que revisar diez predicciones de patrones de diseño de interacción relacionados con diez requerimientos de nivel de diseño diferentes, para predecir el resultado de la herramienta de recomendación basada en el modelo UI-PatternM. Por ejemplo, para el requerimiento “El sistema validará que el número PIN ingresado por el usuario sea correcto.”, el sistema de recomendación con base

en el modelo UIPatternM predijo que el patrón de diseño de interacción recomendado es Notificaciones. Para que los diseñadores de IGUs brinden su opinión respecto a las recomendaciones emitidas por el recomendador, se preparó un cuestionario utilizando una escala Likert-5 y se centró en la predicción del patrón de diseño de interacción por requerimiento a nivel de diseño (ver Apéndice B).

5. Cuestionario SUS. Se elaboró un cuestionario basado en el SUS [106] utilizando una escala Likert-5 y enfocado en la herramienta de recomendación basada en el modelo UIPatternM (ver Apéndice C).
6. Cuestionario TAM. Se elaboró un cuestionario basado en el TAM [105] utilizando una escala Likert-7 y enfocado en la herramienta de recomendación basada en el modelo UIPatternM (ver Apéndice D).

6.2.6. Procedimiento

El experimento se desarrolló en cuatro sesiones con una duración total de dieciocho horas. Sin embargo, no todos los participantes realizaron la prueba el mismo día. Las etapas de la sesión se explican a continuación.

- Introducción (duración \approx 5 min). Los participantes recibieron una descripción general del estudio, junto con sus objetivos.
- Mostrar escenario de uso (duración \approx 5 min). Los participantes fueron puestos en contexto a través de un escenario de uso. Además, se mostró el uso de una herramienta de recomendación de patrones de diseño de interacción.
- Muestra patrones de diseño de interacción (duración \approx 10min). Los participantes recibieron una breve sesión de presentación sobre la identificación de los patrones de diseño de interacción de la colección de Toxboes.

- Muestra un conjunto de diez requerimientos a nivel de diseño (duración \approx 20 min). Se pidió a cada diseñador que analizara diez requerimientos a nivel de diseño y seleccionara un patrón de diseño de interacción para cada uno. Posteriormente de su selección, se pidió a los participantes que indicaran qué tan bien estaban de acuerdo con la comprensión del requerimiento de nivel de diseño para seleccionar un patrón de diseño de interacción (ver Figura 6.1).
- Muestra predicciones de patrones de diseño de interacción (duración \approx 15 min). Se mostraron las predicciones de patrones de diseño de interacción correspondientes de diez requerimientos a nivel de diseño. A cada diseñador se le pidió que respondiera un cuestionario para indicar qué tan bien estaba de acuerdo con cada recomendación generada por la herramienta de recomendación.
- Una vez que los participantes terminaron de evaluar las predicciones de los patrones de diseño de interacción, respondieron los cuestionarios SUS y TAM (duración \approx 5 min).

6.2.7. Resultados

Como resultado del estudio, se obtuvo un conjunto de datos cualitativos y cuantitativos. Estos datos fueron analizados y los resultados obtenidos se presentan a continuación.

6.2.7.1. Resultados sobre las percepciones de uso del modelo UIPatternM por diseñadores profesionales

Los resultados obtenidos se presentan en tres partes: (1) los resultados de la selección del patrón de diseño interactivo por diseñadores profesionales, (2) sobre la evaluación por diseñadores profesionales al conocer las predicciones del recomendador UIPatternM,

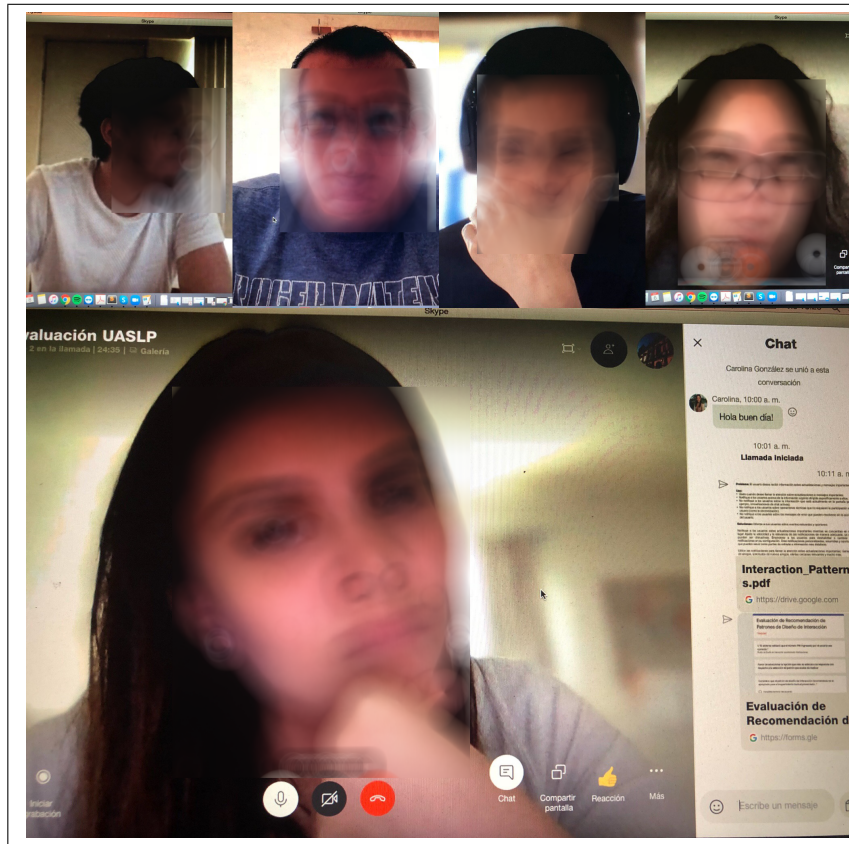


Figura 6.1: Evaluación con Diseñadores de IGUs profesionales

y (3) los resultados de las percepciones de los participantes sobre la aplicación del modelo UIPatternM a través de un prototipo de herramienta de recomendación. Los datos obtenidos se analizaron con la herramienta Minitab versión 19.1.1 para Windows [107] para analizar los datos obtenidos de los cuestionarios.

6.2.7.2. Selección de patrones de diseño de interacción por diseñadores profesionales

La precisión de la recomendación de patrones de diseño de interacción se determinó validando los patrones de diseño de interacción seleccionados por los participantes para 10 requerimientos funcionales a nivel de diseño (ver Tabla 6.7). Posteriormente, las selecciones de los participantes fueron comparadas con las predicciones del reco-

mendador UIPatternM, para obtener el número de selecciones válidas e inválidas por participante.

Para analizar los datos, los participantes se dividieron en dos grupos: diseñadores *junior* (de 1 a 4 años de experiencia) y diseñadores *senior* (6 o más años de experiencia). La figura 6.2 muestra que los diseñadores *senior* son la mayoría de los profesionales que seleccionaron patrones de diseño de interacción similares a los predichos por el recomendador de UIPatternM. Lo anterior debido a que los diseñadores *senior* seleccionaron patrones de diseño de interacción similares en cinco de diez requerimientos de nivel de diseño (50%). Por otro lado, los diseñadores *junior* solo seleccionaron patrones de diseño de interacción similares en dos de diez requerimientos a nivel de diseño (20%). Este hecho podría significar que las decisiones de diseño dependan de la experiencia de los participantes.

6.2.7.3. Percepción de los patrones de diseño de interacción recomendados

Los participantes evaluaron las recomendaciones de patrones de diseño de interacción para diez requerimientos de nivel de diseño diferentes (ver Tabla 6.8). Para obtener las percepciones de los participantes se utilizó un cuestionario de escala Likert de 5 puntos. El cuestionario varía de 1 ('completamente en desacuerdo') a 5 ('completamente de acuerdo'). El estadístico utilizado para comparar si hubo una diferencia significativa entre las medias en las muestras de cada requerimiento es la t-Student. La prueba *t* calcula un valor *p* basado en los datos de desempeño para decidir si los participantes estuvieron de acuerdo con la recomendación de un patrón de diseño de interacción. Si el valor *p* es lo suficientemente pequeño (Valor $p < 0.05$) en la mayoría de los diez requerimientos a nivel de diseño, se concluye que la diferencia es significativa. La Tabla 6.9 muestra los participantes que estuvieron de acuerdo con la mayoría de las recomendaciones predichas por el recomendador de UIPatternM (7 de 10).

Tabla 6.7: Conjunto de requerimientos funcionales a nivel de diseño para identificación de patrones de diseño de interacción, por parte de los diseñadores de IGUs profesionales.

| Requerimiento a nivel de diseño | Predicción |
|--|----------------------|
| El sistema no permitirá al cliente reservar una habitación que ya está reservada. | Notificaciones |
| El sistema debe permitir al usuario imprimir los registros de sus clientes por nombre, fecha, nombre comercial, etc. | Filtro de Tabla |
| El sistema validará que la tarjeta insertada sea válida para transacciones financieras en este cajero automático. | Notificaciones |
| El sistema debe registrar todos los cambios en cualquier parte del módulo del cliente. Los detalles de estos cambios se pueden ver a través del sistema o mediante informes impresos en una base de datos diaria. | Filtros de búsqueda |
| La facturación de los pedidos de ventas se realizará en lotes, a través de una pantalla de pedidos pendientes de facturación, que mostrará los pedidos no facturados. Una vez facturados, los pedidos no se mostrarán en esta lista. | Filtro de Tabla |
| El sistema permitirá a los usuarios ver las calificaciones parciales, los promedios y la calificación académica de los estudiantes y clasificarlos por estudiante, área, fecha, etc. | Ordenar por columna |
| El sistema permitirá a los usuarios autorizados ingresar planes y cronogramas del proyecto. | Solicitud de entrada |
| El sistema permitirá al cliente registrar sus datos personales, esta información debe contener los siguientes datos, nombre, apellidos, edad. | Solicitud de entrada |
| El sistema permitirá al usuario seleccionar un radio de búsqueda, no mayor que los límites de la ciudad donde vive. | Formato de perdón |
| El sistema indicará el contexto actual de los cuidadores, que incluye: identidad, ubicación y actividad; en apoyo de la toma de decisiones. | Tablero |

Tabla 6.8: Predicción emitida por el recomendador de patrones de diseño de interacción a 10 requerimientos funcionales a nivel de diseño.

| Requerimiento a nivel de diseño | Predicción |
|---|--------------------------|
| El sistema validará que el número PIN ingresado por el usuario sea correcto | Notificaciones. |
| El sistema clasificará los artículos de inventario en orden alfabético. | Ordenar por columna |
| Los nuevos usuarios podrán registrarse en el sistema. | Solicitud de Entrada |
| El sistema permitirá a la recepcionista observar las habitaciones reservadas a través de la web. Además de mostrar la información del cliente registrado en la base de datos. | Filtros de Búsqueda |
| El sistema debe mostrar al cliente la disponibilidad de habitaciones. | Controles Transformables |
| El sistema indicará al usuario cuándo y dónde se realizarán actividades recreativas de acuerdo con sus afinidades, dentro de la ciudad. | Formato Estructurado |
| Un usuario puede buscar un vuelo. Si se selecciona un vuelo, el usuario tiene 2 minutos para confirmar la reserva o se liberará el vuelo. | Filtros de Búsqueda |
| El sistema debe proporcionar alertas cuando un artículo en el inventario cae por debajo de un nivel de umbral. | Notificaciones |
| El sistema permitirá que el cuidador conozca la identidad del adulto mayor con quién está ocupado cada cuidador. | Notificaciones |
| La herramienta de autoría debería permitir una fácil modificación del comportamiento de las interfaces de usuario. | Controles Transformables |

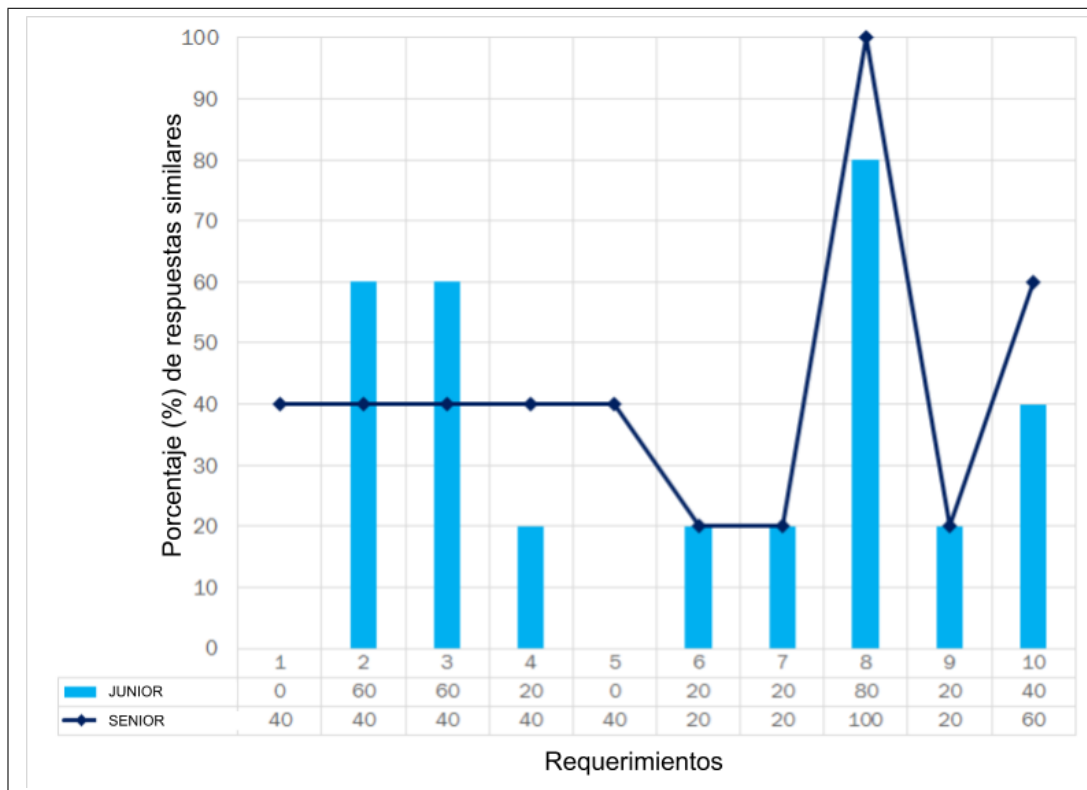


Figura 6.2: Porcentaje del grupo de participantes (diseñadores *junior* y *senior*) que seleccionaron patrones de diseño de interacción similares a los del recomendador UI-PatternM.

6.2.7.4. Percepción de utilidad y facilidad de uso

Respecto a la percepción de utilidad de la herramienta de recomendación, se utilizó el cuestionario SUS, y las primeras seis preguntas del cuestionario TAM. Se obtuvieron los puntajes de todos los cuestionarios del SUS y se transformaron según la escala de calificación curva [108]. El recomendador de patrones de diseño de interacción obtuvo una puntuación SUS promediada de 77 (Media = 78 = B +, desv. est. = 15), correspondiente a una nota B, y representa una buena percepción de utilidad [109]. Así mismo, se aplicó en paralelo el cuestionario TAM para verificar con seis de sus ítems el nivel de utilidad de la herramienta de recomendación. El cuestionario TAM utiliza una escala Likert de 7 puntos, que va de 1 ('completamente en desacuerdo') a 7 ('completamente

Tabla 6.9: Análisis estadístico de la evaluación de los resultados del recomendador de patrones de diseño de interacción por parte de diseñadores profesionales.

| Requerimiento a nivel de diseño | Media | Desv. Estándar | Valor P (t-student) |
|--|--------------|-----------------------|----------------------------|
| Requerimiento 1 | 4.214 | 1.122 | 0.001 |
| Requerimiento 2 | 4.571 | 0.514 | 0.000 |
| Requerimiento 3 | 4.857 | 0.363 | 0.000 |
| Requerimiento 4 | 4.214 | 0.893 | 0.000 |
| Requerimiento 5 | 4.143 | 0.663 | 0.000 |
| Requerimiento 6 | 3.429 | 1.505 | 0.153 |
| Requerimiento 7 | 3.714 | 1.541 | 0.053 |
| Requerimiento 8 | 4.429 | 1.453 | 0.001 |
| Requerimiento 9 | 2.929 | 1.439 | 0.572 |
| Requerimiento 10 | 4.143 | 1.167 | 0.001 |

de acuerdo'), por lo que se considera un nivel aceptable de utilidad en un rango de 25 a 42 (ver Tabla 6.10). Las medias del cuestionario TAM en su factor de utilidad (primeros seis ítems) indican una tendencia favorable aceptable para la herramienta de recomendación (Media = 6, desv. est. = 0.663).

Para validar la facilidad de uso se obtuvieron los datos de las últimas seis preguntas del cuestionario TAM. El cuestionario TAM utiliza una escala Likert de 7 puntos, que va de 1 ('completamente en desacuerdo') a 7 ('completamente de acuerdo'), por lo que se considera un nivel aceptable de facilidad de uso en un rango de 25 a 42 (ver Tabla 6.10). Las medias del cuestionario TAM en su factor de facilidad de uso indican una tendencia aceptable para la herramienta de recomendación (Media = 6, desv. est. = 0.730).

Tabla 6.10: Resultados de aplicación del cuestionario TAM.

| Pregunta | Media |
|---|--------------|
| Utilidad | |
| P1: Usar el sistema me permitiría realizar tareas más rápidamente | 5.631 |
| P2: Usar el sistema mejoraría mi desempeño | 5.722 |
| P3: Usar el sistema facilitaría las tareas del trabajo. | 6.021 |
| P4: Usar el sistema mejoraría mi efectividad en el trabajo | 5.814 |
| P5: Usar la pantalla del sistema aumentaría mi productividad | 5.247 |
| P6: Encontraría útil el sistema en mi trabajo | 5.902 |
| Promedio | 5.907 |
| Facilidad de uso | |
| P7: Me resulta fácil conseguir que el sistema haga lo que quiero que haga | 5.468 |
| P8: Mi interacción con el sistema es clara y comprensible | 5.710 |
| P9: Aprender a operar el sistema sería fácil para mí. | 5.702 |
| P10: Sería fácil para mí volverme hábil en el uso del sistema | 5.211 |
| P11: Encontraría el sistema fácil de usar | 5.495 |
| P12: Encontraría el sistema flexible para interactuar | 5.609 |
| Promedio | 5.530 |

6.3. Discusión

La mayoría de los enfoques reportados en la literatura [6, 44, 63, 86] se enfocan en clasificar los RF y RNF con respecto a la calidad del software, mientras que este trabajo de tesis analiza los RF a nivel de diseño (para respaldar el trabajo de los diseñadores de IGUs). Además de que los conjuntos de datos actuales, consideran mayor número de clases para los RNF para análisis con respecto a la calidad del software, mientras que los RF sólo se considera como una única clase. Por lo que fue necesario la creación de un conjunto de datos con base en diferentes corpus y especificaciones de requerimientos de diferentes proyectos de software. El número de instancias de RF a nivel de diseño

recolectados representó un número menor en relación a los trabajos reportados en la literatura, sin embargo el conjunto de datos de este trabajo de tesis permitió contribuir al cuerpo de trabajos en esta área.

Respecto a la evaluación con diseñadores de IGU expertos, se encontró que en cuanto a la hipótesis H1 se considera aceptable la opinión de los diseñadores relacionada con la recomendación de un patrón de diseño de interacción. La hipótesis de $\mu > 3$ fue aceptada en siete de las diez recomendaciones para patrones de diseño de interacción basados en el análisis de medias realizado. La posible razón de esto es que los diseñadores analizan el requerimiento a nivel de diseño a partir de la recomendación proporcionada por la herramienta de recomendación, delimitando el número de patrones de diseño de interacción. Sin embargo, los diseñadores mencionaron la importancia de tener la justificación de la recomendación, para estar seguros de que es el patrón de diseño de interacción correcto.

“Lo que necesito es algo que esté respaldado por datos, en este caso, el sistema me dice si esto funciona, esto no funciona. ¿Por qué? ¿Dónde están los datos? ¿Dónde está el estudio detrás de estas sugerencias? el sistema está haciendo la selección y cómo puedo estar seguro de que el sistema está haciendo la selección correcta”. [Participante 4]

Por otro lado, la hipótesis H2 fue aceptada, aunque los resultados reflejan que los participantes seleccionaron patrones de diseño de interacción subjetivamente de acuerdo con su interpretación de los requerimientos a nivel de diseño. Los participantes con más años de experiencia (diseñador *senior*) realizaron un análisis más profundo (50 % de similitud con UIPatternM) que los participantes con menos años de experiencia (diseñador *junior*). Sin embargo, los requerimientos de nivel de diseño donde la estructura gramatical era simple, tanto los diseñadores *senior* como los *junior* lograron identificar el patrón de diseño de interacción similar al predicho por la herramienta de recomendación.

En el caso de la hipótesis H3, se aceptó; esto significa que los diseñadores consideraron que una herramienta de recomendación de UIPatternM es fácil de usar, de acuerdo con los resultados de SUS y TAM. Los diseñadores expresaron que los requerimientos son proporcionados de forma textual por un miembro del equipo del proyecto. Por lo tanto, no es una actividad complicada para ellos ingresar requerimientos a nivel de diseño en una herramienta de recomendación.

“Tengo mis datos en mis carpetas y los sigo guardando; al tener algo como esto es como tener algún tipo de biblioteca, entonces sería más fácil aplicar el proceso. Es más fácil, más rápido y pienso más eficiente. No tengo que ir a buscar mis archivos...”[Participante 12]

Finalmente, el hecho de que se haya aceptado H4 indica que la herramienta de recomendación es efectiva para apoyar decisiones de diseño interactivo, en comparación con hacer un proceso subjetivo. La propuesta fue aplicar el modelo UIPatternM para diseñar una herramienta de recomendación que respaldaría las decisiones de diseño en el diseño de sistemas interactivos. Los resultados de la evaluación (ver Tabla 6.9) brindan evidencia que indica que se podría obtener la recomendación de patrones de diseño de interacción congruentes, y por lo tanto, la herramienta de recomendación sería útil y bien aceptada en el contexto del diseño de interacción (ver Tabla 6.10).

Al respecto uno de los participantes comentó: *“Si eres novato y no tienes experiencia, el sistema te enseñará... quizás entenderás un poco más cómo es el comportamiento del sistema, al usar el sistema... además te ayuda a mejorar tus tiempos, te enseñará si eres un novato, a utilizar correctamente las herramientas, en base a las mejores prácticas. Probablemente, en base a los proyectos que se han hecho antes, mi enfoque como alguien que ha un poco más de experiencia, aunque todavía tengo mis propios cinco años de experiencia, siempre busco proyectos relacionados en Internet; veo lo que usan y trato de analizar en base a requerimientos... qué me funciona, qué no funciona para mí”. [Participante 7]*

Por un lado, estos resultados podrían ser atractivos para implementar marcos que ayuden a los diseñadores de IGUs a tomar decisiones de diseño. La implementación del *Modelo de recomendación de patrones de diseño de interacción* podría reducir la cantidad de tiempo invertido, tratando de encontrar patrones de diseño de interacción con los requerimientos de nivel de diseño anteriores. Incluso cuando la precisión resulta ser baja para algunas clases, reducir el espacio de búsqueda sugiriendo patrones de diseño de interacción potenciales a los diseñadores puede ser suficiente. El uso de patrones de diseño de interacción es favorable para las tareas de diseño de IGU de los diseñadores. Además, el uso de patrones de diseño de interacción ayuda a cumplir con los principios de usabilidad y experiencia del usuario, promoviendo así el uso de estándares y mejores prácticas. La búsqueda y selección de patrones de diseño de interacción son los dos problemas principales que se pueden abordar antes de la aplicación del patrón correcto. La primera cuestión es la búsqueda de patrones de diseño de interacción asociados, que dependen de los esquemas de clasificación de las organizaciones de patrones de diseño de interacción. El segundo tema se refiere a la selección de los patrones de diseño de interacción correctos dentro de la gran cantidad de catálogos existentes. Además, si el diseñador es un novato, se requiere conocimiento para comprender la similitud entre los patrones de diseño de interacción [110].

El alcance de la aplicación desarrollada es para un catálogo de patrones de diseño de interacción que cubren el diseño de IGU. Sin embargo, el *Modelo de recomendación de patrones de diseño de interacción* apunta más allá de eso al tener un conjunto de datos y considerar diferentes catálogos de patrones de diseño de interacción para cubrir los requerimientos de cualquier sistema. Además, dichos resultados se apoyaron de una segunda evaluación del módulo de Recomendación de patrones de diseño de interacción con diseñadores expertos que trabajan con patrones de diseño de interacción en su práctica diaria. La propuesta de este trabajo de tesis apoya a los diseñadores profesionales a obtener recomendaciones de patrones de diseño de interacción con base en

el texto de los requerimientos a nivel de diseño y un conjunto de datos de proyectos de interacción. Por lo tanto, se fomenta la reutilización de soluciones de diseño mediante el uso de patrones de diseño de interacción. En consecuencia, el diseño del sistema interactivo con el recomendador UIPatternM podría ser más productivo y menos subjetivo y sujeto a errores.

Los diseñadores profesionales que participaron en el estudio conocen los métodos y prácticas aceptables del diseño de interacción. Por tanto, las percepciones de los diseñadores dotaron a este trabajo de tesis de una visión real de los proyectos actuales, así como la viabilidad de la propuesta para adaptarse a las diferentes metodologías de diseño bajo las que actualmente trabajan.

6.4. Resumen del capítulo

En este capítulo se describe la evaluación realizada a los algoritmos de clasificación aplicados al *Modelo de recomendación de patrones de diseño de interacción*, para la clasificación de requerimientos a nivel de diseño. Con el objetivo de apoyar a los diseñadores de IGUs en la toma de decisiones de diseño. Los algoritmos de clasificación aplicados fueron: regresión logística, Bayes Ingenuo multinomial, máquina de vectores de soporte lineal y bosque aleatorio [98,99]. Si bien los resultados de los experimentos determinan que el método de la Máquina de Vector de Soporte Lineal es el más apropiado para este problema, un desafío continuo es determinar el conjunto de datos ya que se formó a partir de la literatura, en lugar de opiniones de expertos en el campo.

Como parte de los resultados de las evaluaciones, se dio respuesta a la pregunta de investigación: ¿Cuál es el impacto del diseño de la IGU con base en la recomendación del patrón de diseño de interacción?. A través de la opinión de los diseñadores profesionales, respecto a obtener información de patrones de diseño de interacción asociada con los requerimientos funcionales para la toma de decisiones de diseño de IGUs. Además

se identificaron algunos puntos que se deben considerar para mejorar la aplicación del modelo UIPatternM. Este tipo de observaciones dan pauta para definir en el siguiente capítulo las principales conclusiones, aportaciones y trabajo futuro derivados del presente trabajo de tesis.

Capítulo 7

CONCLUSIONES, CONTRIBUCIONES Y TRABAJO A FUTURO.

En este trabajo de tesis se aborda la problemática de identificación de patrones de diseño de interacción en el diseño de IGUs. Para lo cual, se realizó un estudio preliminar para la identificación de elementos de información asociados al diseño de interfaces gráficas de usuario en el proceso de desarrollo de sistemas interactivos, y el cual es detallado en el Capítulo 4. Lo anterior, conllevó al diseño de un modelo de recomendación con base en la clasificación de requerimientos funcionales, en apoyo a la toma de decisiones en el diseño de interfaces gráficas de usuario. Para la predicción de patrones de diseño de interacción fueron aplicados algoritmos de clasificación sobre un conjunto de datos de requerimientos de software.

En relación a la validación del modelo, se llevaron a cabo pruebas en relación a la precisión y desempeño de los algoritmos de clasificación. Además de la ejecución de una evaluación con diseñadores profesionales para conocer su percepción de utilidad de la visión tecnológica.

En las siguientes secciones se presentan las conclusiones de este trabajo de tesis, las principales aportaciones del mismo, y las propuestas de trabajo futuro.

7.1. Conclusiones

En este trabajo de tesis, se identificó un área de oportunidad para el apoyo a los diseñadores de IGUs en la toma de decisiones de diseño de IGUs. Para lo cual, se diseñó un modelo de recomendación llamado UIPatternM a partir de los hallazgos identificados en el caso de estudio inicial, para el entendimiento del proceso de diseño de IGUs desde la perspectiva de IS e IHC. Como resultado del estudio se identificaron y analizaron los elementos de información asociados en el diseño de IGUs (ver Capítulo 4). De esta manera, se definió que se contaba con elementos suficientes para considerar la propuesta de apoyar en la identificación de patrones de diseño de interacción, las cuales considerarán las necesidades expresadas por los usuarios finales. Con base en estos requerimientos e ideas de diseño, fueron tomados en cuenta los elementos de información incluidos en la especificación de requerimientos de software.

A partir del entendimiento del proceso de diseño de IUs, se derivó el diseño del modelo UIPatternM, el cual establece un conjunto de módulos que consideran el procesamiento de texto de los requerimientos de software con base en algoritmos de clasificación (ver Capítulo 5). Sin embargo, fue necesario considerar la forma en que son expresados los requerimientos funcionales (formato textual), para considerar que al menos expresaran los elementos básicos de usuario, acción, objeto y propiedad. Para este tipo de retos, el área de IA brinda soluciones a través de los algoritmos de procesamiento de lenguaje natural y la cual ha sido abordada por algunos investigadores sobre la problemática de análisis de requerimientos de software. Sin embargo, un obstáculo que se tuvo que enfrentar fue la limitante en los conjuntos de datos para este tipo de aplicaciones, ya que en el área de diseño de interacción no se analiza este tipo de información.

En el caso de la ingeniería de software, se tiene registro de algunos conjuntos de datos pequeños orientados al análisis para determinar la calidad de software o la reutilización de código. Por ello, en este trabajo de tesis se creó un conjunto de datos compuesto por 307 requerimientos etiquetados a nivel de diseño. El conjunto de datos se creó en función de los requerimientos de nivel de diseño que se identificaron y recopilamos de algunos conjuntos de datos de proyectos de software como [15,102]. Los requerimientos a nivel de diseño del conjunto de datos pertenecen a nueve clases diferentes, que corresponden a nueve patrones de diseño de interacción de la colección Toxboe [16]. Para la selección de dicho catálogo de patrones, se analizaron varios y se consideró la colección de Toxboe por su estructura sencilla y clara para los patrones de diseño de interacción más comunes dentro del entorno web.

En el desarrollo del trabajo de tesis se estableció la necesidad de evaluar la propuesta (ver Capítulo 6). Dicha evaluación tuvo como objetivo validar el rendimiento de los algoritmos de clasificación aplicados, además de una evaluación más para obtener la percepción por parte de diseñadores de IGUs profesionales. Para llevar a cabo la primera evaluación fue necesario realizarla a través de métricas específicas de rendimiento de algoritmos de clasificación sobre un conjunto de datos formado por requerimientos de sistemas interactivos. La evidencia de la primera evaluación mostró que el método de la Máquina de Vector de Soporte Lineal es el más apropiado para este problema. Sin embargo un desafío continuo fue determinar el conjunto de datos para la aplicación de los algoritmos de clasificación, ya que el utilizado para este trabajo de tesis se formó a partir de la literatura, en lugar de opiniones de expertos en el campo.

En relación al estudio de caso para obtener la percepción de utilidad por parte de los diseñadores de IGUs profesionales, se solicitó la participación de diseñadores de IGUs con conocimientos en los métodos y prácticas aceptables del diseño de interacción. Por lo cual, el desafío se presentó en el reclutamiento de diseñadores de IGUs profesionales que tuvieran conocimientos de patrones de diseño y sobre todo, de disposición para colaborar

con el proyecto de investigación. A pesar de ello, se logró obtener la percepción de 14 diseñadores de IGUs profesionales, que dotaron a este trabajo de tesis de una visión real de los proyectos actuales, así como la viabilidad de nuestra propuesta para adaptarse a las diferentes metodologías de diseño bajo las que actualmente trabajan.

El análisis de los resultados obtenidos en estas evaluaciones y de los estudios de caso presentados a lo largo de este documento, permiten concluir lo siguiente:

- Los requerimientos de software marcan el punto de partida del proceso de desarrollo de un sistema interactivo, por lo tanto es de suma importancia poner énfasis en el establecimiento de los mismos. De manera que las especificaciones de requerimientos representen las necesidades de los clientes y usuarios, para su cumplimiento por parte del equipo de trabajo involucrado en el proyecto. Así mismo, las especificaciones textuales de los requerimientos de software contienen información, que el equipo de trabajo analiza para identificar los problemas, para los que posteriormente buscarán alternativas y soluciones.
- El uso de patrones de diseño de interacción permite la reutilización de soluciones a problemas de diseño, ya que transfieren su conocimiento a los diseñadores, de una manera clara y comprensible para mejorar sus diseños, por lo que son percibidas por ellos como útiles y con intención de volver a utilizarlos para resolver problemas semejantes. Sin embargo, el identificar los patrones de diseño de interacción no debe ser una actividad compleja y propensa a errores costosos para el proyecto.
- La recomendación de patrones de diseño de interacción debe considerar el problema de diseño al que se debe dar solución. Para lo cual, un sistema de recomendación a partir de un conjunto de datos de instancias de requerimientos de software asociados a instancias de patrones de diseño de interacción llamado UIPatternM, puede realizar recomendaciones precisas y acertadas para apoyar a los diseñadores en la toma de decisiones de diseño de IGUs.

Finalmente, podemos concluir que a través del modelo UIPatternM es posible la reutilización de conocimiento de diseño de interacción, además de agilizar el diseño de IGUs apoyando las decisiones de diseñadores de IGUs.

7.2. Contribuciones

La principal contribución de este trabajo de tesis es en torno al modelo de recomendación de patrones de diseño de interacción con base en requerimientos funcionales, en apoyo al diseño de interfaces gráficas de usuario. Lo anterior, a través de la aplicación de algoritmos de máquinas de aprendizaje. Para lo cual, el modelo de recomendación fue sometido a una evaluación con base a métricas de precisión y desempeño de los algoritmos aplicados, así como la percepción de utilidad por parte de diseñadores profesionales. Además, otras aportaciones derivadas de este trabajo de tesis fueron la generación de un conjunto de datos de instancias de requerimientos funcionales a nivel de diseño asociados a un conjunto de patrones de diseño de interacción y la aplicación de algoritmos de clasificación para la predicción de patrones de diseño de interacción. Finalmente, las publicaciones derivadas de este trabajo de tesis constituyen una contribución importante del mismo. A continuación se enlistan las principales aportaciones seguidas de la lista de publicaciones desarrolladas con base en el mismo.

1. Un modelo de recomendación de patrones de diseño de interacción con base en la correlación de instancias referentes a patrones de diseño de interacción y requerimientos funcionales. Este modelo fue construido con base en la literatura de IS y IHC. En cuanto a la IS, se retomó la importancia de los requerimientos de software en el desarrollo de sistemas interactivos [40]. Mientras que en IHC con el diseño de interacción, se ocupa de la asignación de funciones a la agencia humana o de la tecnología y con la estructuración y secuenciación de las interacciones. La asignación de funciones tiene un impacto significativo en la facilidad y la experiencia de

uso de un sistema interactivo. Los diseñadores de IGUs crean tareas, de manera que los usuarios tengan la manera de ejecutar acciones sobre las funciones del sistema interactivo [111]. Para lo cual, en el diseño de IGUs, resulta útil la aplicación de patrones de diseño de interacción, los cuales encapsulan una solución probada para un problema de usabilidad. Seffah [64] y algunos otros autores, basan sus propuestas en el desarrollo en torno al enfoque de que los patrones de diseño de interacción puede complementar a un modelo de tareas, al proveer las experiencias acumuladas a través de la retroalimentación del usuario final. Sin embargo, la especificación del modelo de tareas se genera a partir de la interpretación subjetiva del analista y la cual está sujeta a errores. Por lo anterior, el modelo UIPatternM considera procesamiento del lenguaje natural para determinar los elementos de información relacionados a las funciones del sistema y para lo cual es necesaria la implementación de controles gráficos en las IGUs. Esta situación se puede repetir consecutivamente en el diseño de un sistema interactivo o en algunos similares, de manera que con esos datos acumulados sea posible determinar diferente información, como predecir patrones de diseño de interacción. Es por ello que con este modelo UIPatternM se logra obtener argumentos suficientes para la comprensión de la identificación de patrones de diseño de interacción. Esta aportación puede ser considerada como parte complementaria a las literaturas tanto de IS como de IHC.

2. Un conjunto de datos representativo de instancias de requerimientos funcionales. Para obtener este conjunto de datos el área de IS resultó importante, dada los lineamientos establecidos para la especificación de requerimientos de software. Particularmente el trabajo de Diamantopoulos [11], referente al análisis semántico y sintáctico de requerimientos de software, se retomaron elementos para determinar las características o atributos del conjunto de datos. Las cuales representan un dato medible que se puede utilizar para la predicción de patrones de diseño de

interacción. Esta propuesta contribuyó a la identificación de áreas de oportunidad para apoyar a diseñadores de IGUs en la toma de decisiones de diseño.

3. Implementación de métodos de aprendizaje automático para correlacionar instancias referentes a patrones de diseño de interacción y requerimientos funcionales. La aplicación de algoritmos de clasificación sobre representaciones de vectores ponderados de TF-IDF. La ponderación TF-IDF se usa comúnmente en la extracción de texto y la recuperación de información para evaluar la importancia de un término lingüístico (generalmente unigrama o bigrama) en un conjunto de datos. La importancia del término (peso) aumenta con la frecuencia del término en el texto, pero se compensa con la frecuencia del término en el dominio de interés. Esta implementación en buena medida fue parte esencial para la evaluación en términos de rendimiento y utilidad por parte de diseñadores profesionales. Este modelo UIPatternM contribuye de manera diferente a los modelos propuestos para el diseño de IGUs, pues no se encontraron modelos o propuestas que apoyen a la toma de decisiones de diseño a través de la identificación de patrones de diseño de interacción.

Un resultado importante de este trabajo de tesis, lo constituyen las publicaciones realizadas.

Publicación en revista:

1. **Silva-Rodríguez, V.**, Nava-Muñoz, S. E., Castro, L. A., Martínez-Pérez, F. E., Pérez-González, H. G., Torres-Reyes, F. (2020). Classifying design-level requirements using Machine Learning for a Recommender of Interaction Design Patterns. In IET Software Journal. DOI: 10.1049/iet-sen.2019.0291.
2. **Silva-Rodríguez, V.**, Nava-Muñoz, S. E., Castro, L. A., Martínez-Pérez, F. E., Pérez-González, H. G., Torres-Reyes, F. Predicting interaction design patterns

for designing explicit interactions in ambient intelligence systems: A case study. In *Personal and Ubiquitous Computing Journal*. Aceptado para publicación.

Publicación en congresos internacionales:

1. **Silva-Rodríguez, V.**, Nava-Muñoz, S. E., Castro, L. A., Martínez-Pérez, F. E., Pérez-González, H. G., Torres-Reyes, F. (2019). Machine Learning Methods for Inferring Interaction Design Patterns from Textual Requirements. In *Multidisciplinary Digital Publishing Institute Proceedings* (Vol. 31, No. 1, p. 26).
2. **Silva-Rodríguez, V.**, Nava-Muñoz, S. E., Martínez-Pérez, F. E., Pérez-González, H. G. (2018, October). How to Select the Appropriate Pattern of Human-Computer Interaction?: A Case Study with Junior Programmers. In *2018 6th International Conference in Software Engineering Research and Innovation (CONISOFT)* (pp. 66-71). IEEE.

Publicación en congresos nacionales:

1. **Viridiana Silva-Rodríguez** and Sandra E. Nava-Muñoz. 2018. Identifying Human-Computer Interaction Patterns in Support of the User Interfaces Design. In *Proceedings of the 7th Mexican Conference on Human-Computer Interaction (MexIHC '18)*. ACM, New York, NY, USA, Article 21, 4 pages.
DOI: <https://doi.org/10.1145/3293578.3298781>

Publicación en foros locales:

1. **Póster de Investigación** - "Diseño de Interfaces de Usuario mediante el uso de Patrones de Interacción Humana-Computadora", en 2017 Concurso de Exhibición UASLP de Carteles de Proyectos de Investigación.
2. **Póster de Investigación** - "Diseño de Interfaces de Usuario mediante el uso de Patrones de Interacción Humano-Computador", en 2017 y 2018 UASLP Puertas Abiertas.

Es importante mencionar, que el desempeño y la facilidad de uso proporcionada por el modelo UIPatternM fue bien percibida por los diseñadores de IGUs profesionales, lo cual no significa que no pueda ser mejorada en más aspectos. Por ello, en la siguiente sección se presenta y describe un conjunto de propuestas de investigación para trabajo futuro que surgen a partir de este trabajo de tesis.

7.3. Trabajo futuro

En esta sección se establecen algunas líneas que pueden complementar esta investigación.

- Funcionalidad de la visión tecnológica aplicando el modelo UIPatternM. Se plantea el desarrollo de una IGU para que los diseñadores de IGUs puedan interactuar con el modelo UIPatternM. De manera de que los diseñadores ingresen sus propios requerimientos, a partir de los cuales se genere predicción y a su vez, retroalimenten al conjunto de datos.
- Aplicación del modelo UIPatternM para entornos inteligentes en los cuales se involucren interacciones tanto explícitas como implícitas. En el diseño de ambientes inteligentes se presenta la necesidad de diseñar sistemas bajo un contexto de uso específico y considerando aspectos contextuales del entorno, lo que le daría cierta “inteligencia” al sistema interactivo.
- Alimentación del conjunto de datos y evaluación de métricas de rendimiento. Hasta el momento, los conjuntos de datos identificados son en relación a la clasificación de clases específicas entre los requerimientos funcionales y no funcionales para aspectos en la calidad del software. Sin embargo, no se tiene referencia de conjuntos de datos asociados a un catálogo específico de patrones de diseño de interacción. Para lo cual se requiere seguir alimentando el conjunto de datos pro-

puesto, de manera que se mejoren los resultados de las métricas de desempeño y precisión al aplicar algoritmos de clasificación.

Bibliografía

- [1] Victoria Meza-Kubo. *Guías para el diseño de aplicaciones de estimulación cognitiva utilizables por el adulto mayor*. PhD thesis, Facultad de Ingeniería, Arquitectura y Diseño. Universidad Autónoma de Baja California, Feb 2013.
- [2] Jesse James Garrett. User experience and why it matters. *JJ Garrett, the Elements of User Experience: User-Centered Design for the Web*, pages 6–19, 2003.
- [3] Christopher Alexander, Sara Ishikawa, and Murray Silverstein. *A pattern language: towns, buildings, construction*. Oxford university press, 1977.
- [4] Vivek Shukla, Dharendra Pandey, and Raj Shree. Requirements engineering: A survey. *Requirements Engineering*, 3(5):28–31, 2015.
- [5] Ian Sommerville. *Software Engineering*. Addison-Wesley Publishing Company, USA, 9th edition, 2010.
- [6] Zijad Kurtanović and Walid Maalej. Automatically classifying functional and non-functional requirements using supervised machine learning. In *Requirements Engineering Conference (RE), 2017 IEEE 25th International*, pages 490–495. IEEE, 2017.
- [7] Thomas T Hewett, Ronald Baecker, Stuart Card, Tom Carey, Jean Gasen, Marilyn Mantei, Gary Perlman, Gary Strong, and William Verplank. *ACM SIGCHI curricula for human-computer interaction*. ACM, 1992.

- [8] Jun Rekimoto and Katashi Nagao. The world through the computer: Computer augmented interaction with real world environments. In *Proceedings of the 8th annual ACM symposium on User interface and software technology*, pages 29–36, 1995.
- [9] Alan Dix. *Human-Computer Interaction*, pages 1327–1331. Springer US, Boston, MA, 2009.
- [10] IEEE Computer Society. Software Engineering Standards Committee and IEEE-SA Standards Board. Ieee recommended practice for software requirements specifications. Institute of Electrical and Electronics Engineers, 1998.
- [11] Themistoklis Diamantopoulos, Michael Roth, Andreas Symeonidis, and Ewan Klein. Software requirements as an application domain for natural language processing. *Language Resources and Evaluation*, 51(2):495–524, 2017.
- [12] Jenifer Tidwell. *Designing interfaces: Patterns for effective interaction design*. ‘Reilly Media, Inc.’, 2010.
- [13] Lilian Regina de Carvalho, Yolanda Dora Martinez Évora, and Silvia Helena Zem-Mascarenhas. Assessment of the usability of a digital learning technology prototype for monitoring intracranial pressure. *Revista latino-americana de enfermagem*, 24, 2016.
- [14] Jakob Nielsen. Severity ratings for usability problems. *Papers and Essays*, 54:1–2, 1995.
- [15] Jane Cleland-Huang, Sepideh Mazrouee, Huang Ligu, and Dan Port. nfr, 2007.
- [16] Anders Toxboe. User interface design patterns.
- [17] Donald A Norman. Do companies fail because their technology is unusable? *Interactions*, 12(4), 2005.

- [18] Jakob Nielsen and Rolf Molich. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 249–256. ACM, 1990.
- [19] Rajib Mall. *Fundamentals of software engineering*. PHI Learning Pvt. Ltd., 2018.
- [20] Steve Krug. *Don't make me think!: a common sense approach to Web usability*. Pearson Education India, 2000.
- [21] Ben Shneiderman. *Designing the user interface: strategies for effective human-computer interaction*. Pearson Education India, 2010.
- [22] Michael Bradley, Per Ola Kristensson, Patrick Langdon, and P John Clarkson. Interaction patterns: The key to unlocking digital exclusion assessment? In *International Conference on Applied Human Factors and Ergonomics*, pages 564–572. Springer, 2018.
- [23] Dirk Riehle and Heinz Züllighoven. Understanding and using patterns in software development. *Tapos*, 2(1):3–13, 1996.
- [24] Miguel Carvalhais. Learning and studying interaction design through design patterns. In *Proceedings of the 15th Conference on Pattern Languages of Programs*, page 11. ACM, 2008.
- [25] Jan O Borchers. Interaction design patterns: twelve theses. In *Workshop, The Hague*, volume 2, page 3. Citeseer, 2000.
- [26] Matijn Van Welie. Pattern library for interaction design. *Website*. URL: <http://www.welie.com>, 2008.
- [27] Aaron Marcus. Patterns within patterns. *interactions*, 11(2):28–34, 2004.

- [28] Ahmed Seffah, Jan Gulliksen, and Michel C Desmarais. *Human-Centered Software Engineering-Integrating Usability in the Software Development Lifecycle*, volume 8. Springer Science & Business Media, 2005.
- [29] Paloma Díaz, Mary Beth Rosson, Ignacio Aedo, and John M Carroll. Web design patterns: Investigating user goals and browsing strategies. In *International Symposium on End User Development*, pages 186–204. Springer, 2009.
- [30] Luis Rivero and Tayana Conte. How novice software engineers apply user interface design patterns: An empirical study. In *Proceedings of the International Conference Software Engineering and Knowledge Engineering SEKE 2016-Janua*, pages 600–604, 2016.
- [31] Nguyen Thanh-Diane, Jean Vanderdonckt, and Ahmed Seffah. Uiplml: Pattern-based engineering of user interfaces of multi-platform systems. In *Research Challenges in Information Science (RCIS), 2016 IEEE Tenth International Conference on*, pages 1–12. IEEE, 2016.
- [32] Ahmed Seffah. *HCI Pattern Capture and Dissemination: Practices, Lifecycle, and Tools*, pages 219–242. Springer International Publishing, Cham, 2015.
- [33] A Gaffar. The other side of patterns: A user centered analysis. *Usability Professional Association, UPA Montreal, Retrieved on March, 2:2004*, 2004.
- [34] A Gaffar. Needles and the haystack: The visibility of information on the internet. *Usability Professional Association, UPA Montreal, Retrieved on February, 24:2005*, 2005.
- [35] Ashraf Gaffar. Application-agnostic interactive big data: managing hci complexity at the source. *International Journal of New Computer Architectures and their Applications (IJNCAA)*, 4:1–16, 2014.

- [36] Eduardo Vaz. Delivering better projects on time by ensuring requirements quality upfront. In *INCOSE International Symposium*, volume 28, pages 575–586. Wiley Online Library, 2018.
- [37] Azham Hussain, Emmanuel Mkpojiogu, and Fazillah Kamal. The role of requirements in the success or failure of software projects. *EJ Econjournals*, 6:6–7, 10 2016.
- [38] Azham Hussain, Emmanuel OC Mkpojiogu, and Fazillah Mohmad Kamal. The role of requirements in the success or failure of software projects. *International Review of Management and Marketing*, 6(7S):306–311, 2016.
- [39] Klaus Pohl. *Requirements engineering: fundamentals, principles, and techniques*. Springer Publishing Company, Incorporated, 2010.
- [40] Ian Sommerville, Russell Lock, and Tim Storer. Information requirements for enterprise systems. In *Monterey Workshop*, pages 266–282. Springer, 2012.
- [41] Jeremy Dick, Elizabeth Hull, and Ken Jackson. *Requirements engineering*. Springer, 2017.
- [42] Irma Patricia Delgado-Solano, Alberto S Núñez-Varela, and G Héctor Pérez-González. Keyword extraction from users’ requirements using textrank and frequency analysis, and their classification into iso/iec 25000 quality categories. In *2018 6th International Conference in Software Engineering Research and Innovation (CONISOFT)*, pages 88–92. IEEE, 2018.
- [43] Abinash Tripathy, Ankit Agrawal, and Santanu Kumar Rath. Requirement analysis using natural language processing. In *Fifth International Conference on Advances in Computer Engineering*, pages 26–27, 2014.

- [44] Guillermo Licea et al. Towards supporting software engineering using deep learning: A case of software requirements classification. In *2017 5th International Conference in Software Engineering Research and Innovation (CONISOFT)*, pages 116–120. IEEE, 2017.
- [45] K. R. Chowdhary. *Natural Language Processing*, pages 603–649. Springer India, New Delhi, 2020.
- [46] Chalernpol Tapsai, Phayung Meesad, and Choochart Haruechaiyasak. Tls-art: Thai language segmentation by automatic ranking trie. In *9th International Conference Autonomous Systems*, 2016.
- [47] Eelke Folmer, Martijn van Welie, and Jan Bosch. Bridging patterns: An approach to bridge gaps between se and hci. *Information and Software Technology*, 48(2):69–89, 2006.
- [48] Luis A Castro and M Rodríguez. Interacción humano-computadora y aplicaciones en méxico. *México: Academia Mexicana de Computación*, 2018.
- [49] Alan Cooper, Robert Reimann, and Hugh Dubberly. *About face 2.0: The essentials of interaction design*. John Wiley & Sons, Inc., 2003.
- [50] Alan Dix. Human–computer interaction, foundations and new paradigms. *Journal of Visual Languages & Computing*, 42:122–134, 2017.
- [51] Brad Myers, Scott E. Hudson, and Randy Pausch. Past, present, and future of user interface software tools. *ACM Trans. Comput.-Hum. Interact.*, 7(1):3–28, March 2000.
- [52] Benjamin Weyers, Judy Bowen, Alan Dix, and Philippe Palanque. *The handbook of formal methods in human-computer interaction*. Springer, 2017.

- [53] Toni Granollers i Saltiveri, Jesús Lorés Vidal, and José Juan Cañas Delgado. *Diseño de sistemas interactivos centrados en el usuario*. Editorial UOC, 2011.
- [54] Jonathan Grudin. The computer reaches out: the historical continuity of interface design. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 261–268, 1990.
- [55] Timo Jokela, Netta Iivari, Juha Matero, and Minna Karukka. The standard of user-centered design and the standard definition of usability: analyzing iso 13407 against iso 9241-11. In *Proceedings of the Latin American conference on Human-computer interaction*, pages 53–60, 2003.
- [56] Scott Henninger and Padmapriya Ashokkumar. An ontology-based metamodel for software patterns. *CSE Technical reports*, page 55, 2006.
- [57] Martijn Van Welie and Gerrit C Van der Veer. Pattern languages in interaction design: Structure and organization. In *Proceedings of interact*, volume 3, pages 1–5, 2003.
- [58] Jan O Borchers. A pattern approach to interaction design. In *Cognition, Communication and Interaction*, pages 114–131. Springer, 2008.
- [59] Rike Brecht. User interface patterns for digital libraries. *TCDL Bulletin*, 6(1), 2010.
- [60] Yuliana Puerta Cruz, César A Collazos, Josefina Guerrero Garcia, and Juan González Calleros. Patterns of interaction description including aspects of constraints. In *Software Engineering Research and Innovation (CONISOFT), 2016 4th International Conference in*, pages 136–140. IEEE, 2016.
- [61] Carlos Iñiguez-Jarrín, José Ignacio Panach, and Oscar Pastor López. Defining interaction design patterns to extract knowledge from big data. In *Internatio-*

- nal Conference on Advanced Information Systems Engineering*, pages 490–504. Springer, 2018.
- [62] Martijn Van Welie, Gerrit C Van Der Veer, and Anton Eliëns. Patterns as tools for user interface design. In *Tools for Working with Guidelines*, pages 313–324. Springer, 2001.
- [63] Mohammad Mahmoud Tarawneh. Software requirements classification using natural language processing and svd. *International Journal of Computer Applications*, 164(1), 2017.
- [64] Ahmed Seffah. *Patterns of HCI Design and HCI Design of Patterns: Bridging HCI Design and Model-Driven Software Engineering*. Springer, 2015.
- [65] Ana Paula O Bertholdo, Fabio Kon, and Marco Aurélio Gerosa. Agile usability patterns for user-centered design final stages. In *International Conference on Human-Computer Interaction*, pages 433–444. Springer, 2016.
- [66] Christine E Wania. Patterns and pattern sites in hci: An analysis. *SAIS 2017 Proceedings (9)*, 2017.
- [67] Joelma Choma, Rodrigo Lucas Barbosa Teixeira, Eduardo Martins Guerra, and Tiago Silva Da Silva. Interaction patterns for user interface design of large web applications. In *Proceedings of the 11th Latin-American Conference on Pattern Languages of Programming*, pages 1–11, 2016.
- [68] Joëlle Coutaz. User interface plasticity: Model driven engineering to the limit! In *Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems*, pages 1–8, 2010.
- [69] Gaëlle Calvary and Joëlle Coutaz. Introduction to model-based user interfaces. *Group Note*, 7:W3C, 2014.

- [70] G Calvary et al. The cameleon reference framework, cameleon project, september 2002, 2002.
- [71] Antonio Delgado, A Estepa, JA Troyano, and Rafael Estepa. Reusing ui elements with model-based user interface development. *International Journal of Human-Computer Studies*, 86:48–62, 2016.
- [72] Ahmed Seffah and Mohamed Taleb. Tracing the evolution of hci patterns as an interaction design tool. *Innovations in Systems and Software Engineering*, 8(2):93–109, 2012.
- [73] Fabio Paterno. *Model-based design and evaluation of interactive applications*. Springer Science & Business Media, 1999.
- [74] Nannan Li, Qingyi Hua, Shasha Wang, Kai Yu, and Liting Wang. Research on a pattern-based user interface development method. In *2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics*, volume 1, pages 443–447. IEEE, 2015.
- [75] Jean Vanderdonckt and Francisco Montero Simarro. Generative pattern-based design of user interfaces. In *Proceedings of the 1st International Workshop on Pattern-Driven Engineering of Interactive Computing Systems*, pages 12–19. ACM, 2010.
- [76] Hong Zhu, Lingzi Jin, Dan Diaper, and Ganghong Bai. Software requirements validation via task analysis. *Journal of Systems and Software*, 61(2):145–169, 2002.
- [77] Gerald Ninaus, Alexander Felfernig, Martin Stettinger, Stefan Reiterer, Gerhard Leitner, Leopold Weninger, and Walter Schanil. Intellireq: Intelligent techniques for software requirements engineering. In *ECAI*, pages 1161–1166, 2014.

- [78] Jamshaid G Mohebzada, Guenther Ruhe, and Armin Eberlein. Systematic mapping of recommendation systems for requirements engineering. In *2012 International Conference on Software and System Process (ICSSP)*, pages 200–209. IEEE, 2012.
- [79] Kunal Shah, Akshaykumar Salunke, Saurabh Dongare, and Kisandas Antala. Recommender systems: An overview of different approaches to recommendations. In *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, pages 1–4. IEEE, 2017.
- [80] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160(1):3–24, 2007.
- [81] Rafael Gorski M Souza and Paulo César Stadzisz. Problem-based software requirements specification. *Revista Electronica de Sistemas de Informacao*, 15(2):1, 2016.
- [82] Hector Gerardo Perez Gonzalez. *Automated Techniques for Object Oriented Analysis and Design from Natural Language Problem Statements*. PhD thesis, University of Colorado at Colorado Springs, 2003.
- [83] Leonid Kof, Ricardo Gacitua, Mark Rouncefield, and Peter Sawyer. Ontology and model alignment as a means for requirements validation. In *2010 IEEE Fourth International Conference on Semantic Computing*, pages 46–51. IEEE, 2010.
- [84] Elke Hochmüller. *Requirements classification as a first step to grasp quality requirements*. REFSQ’97, Presses universitaires de Namur, 1997.

- [85] Hala Alrumaih, Abdulrahman Mirza, and Hessah Alsalamah. Toward automated software requirements classification. In *2018 21st Saudi Computer Society National Computer Conference (NCC)*, pages 1–6. IEEE, 2018.
- [86] Abderahman Rashwan. Semantic analysis of functional and non-functional requirements in software requirements specifications. In *Canadian Conference on Artificial Intelligence*, pages 388–391. Springer, 2012.
- [87] Natalia Juristo, Ana M Moreno, and Maria-Isabel Sanchez-Segura. Analysing the impact of usability on software design. *Journal of Systems and Software*, 80(9):1506–1516, 2007.
- [88] Alberto Rodrigues Da Silva. Model-driven engineering: A survey supported by the unified conceptual model. *Computer Languages, Systems & Structures*, 43:139–155, 2015.
- [89] David Benyon and Catriona Macaulay. Scenarios and the hci-se design problem. *Interacting with computers*, 14(4):397–405, 2002.
- [90] Hosam Al-Samarraie and Yusof Ahmad. Use of design patterns according to hand dominance in a mobile user interface. *Journal of Educational Computing Research*, 54(6):769–792, 2016.
- [91] Karl Wieggers and Joy Beatty. *Software requirements*. Pearson Education, 2013.
- [92] Unnati S Shah and Devesh C Jinwala. Resolving ambiguities in natural language software requirements: a comprehensive survey. *ACM SIGSOFT Software Engineering Notes*, 40(5):1–7, 2015.
- [93] Farhana Nazir, Wasi Haider Butt, Muhammad Waseem Anwar, and Muazzam A Khan Khattak. The applications of natural language processing (nlp) for

- software requirement engineering—a systematic literature review. In *International conference on information science and applications*, pages 485–493. Springer, 2017.
- [94] Gerard Salton and Chung-Shu Yang. On the specification of term values in automatic indexing. *Journal of documentation*, 29(4):351–372, 1973.
- [95] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 2004.
- [96] Charu C Aggarwal. Content-based recommender systems. In *Recommender Systems*, pages 139–166. Springer, 2016.
- [97] Diana MacDonald. Introducing ui patterns. In *Practical UI Patterns for Design Systems*, pages 1–21. Springer, 2019.
- [98] Tomas Pranckevičius and Virginijus Marcinkevičius. Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification. *Baltic Journal of Modern Computing*, 5(2):221, 2017.
- [99] Monica Bali and Deipali Gore. A survey on text classification with different types of classification methods. *International Journal of Innovative Research in Computer and Communication Engineering*, 3:4888–4894, 2015.
- [100] Jose García Moreno-Torres, José A Sáez, and Francisco Herrera. Study on the impact of partition-induced dataset shift on k -fold cross-validation. *IEEE Transactions on Neural Networks and Learning Systems*, 23(8):1304–1312, 2012.
- [101] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. Text classification algorithms: A survey. *Information*, 10(4):150, 2019.

- [102] Dmitry Garbar. Custom software requirements specification document example (international standard), Jan 2020.
- [103] Guozhong Feng, Shaoting Li, Tieli Sun, and Bangzuo Zhang. A probabilistic model derived term weighting scheme for text classification. *Pattern Recognition Letters*, 110:23–29, 2018.
- [104] Amani Khelifa, Mariem Haoues, and Asma Sellami. Towards a software requirements change classification using support vector machine. In *Proceedings of the second Conference on Language Processing and Knowledge Management*, volume 2279, pages 1–10. CEUR-WS.org, 2018.
- [105] Fred D Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, pages 319–340, 1989.
- [106] John Brooke. *"SUS-A quick and dirty usability scale. Usability evaluation in industry*. CRC Press, June 1996. ISBN: 9780748404605.
- [107] LLC Minitab. Minitab. *Inc., versão, 19.1.1*, 2019.
- [108] Jeff Sauro and James R Lewis. *Quantifying the user experience: Practical statistics for user research*. Morgan Kaufmann, 2016.
- [109] Aaron Bangor, Philip T Kortum, and James T Miller. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction*, 24(6):574–594, 2008.
- [110] Shahid Hussain, Jacky Keung, Muhammad Khalid Sohail, Arif Ali Khan, and Manzoor Ilahi. Automated framework for classification and selection of software design patterns. *Applied Soft Computing*, 75:1–20, 2019.
- [111] David Benyon. Designing interactive systems: A comprehensive guide to hci. *UX and interaction design*, 3, 2014.

Apéndice A

Cuestionario Selección de Patrones de Diseño de Interacción

Selección de Patrones de Diseño de Interacción

Favor de elegir el patrón de diseño de interacción que considere que mejor se adecua al requerimiento que se presenta.

***Obligatorio**

1. El sistema no permitirá al cliente reservar una habitación que ya está reservada. *

Marca solo un óvalo.

- Filtro de Tabla
- Tablero
- Ordenar por columna
- Controles Transformables
- Filtros de búsqueda
- Formato estructurado
- Notificaciones
- Formato de perdón
- Solicitud de entrada

Favor indique la opción que más se adecue a su respuesta con respecto a la elección de patrón que acaba de realizar

2. Elegir un patrón de diseño de interacción me resultó sencillo. *

Marca solo un óvalo.

- Completamente en desacuerdo
- Ligeramente en desacuerdo
- Ni de acuerdo, ni en desacuerdo
- Ligeramente de acuerdo
- Completamente de acuerdo

3. Entendí lo que el requerimiento solicitaba en la primera lectura. *

Marca solo un óvalo.

- Completamente en desacuerdo
- Ligeramente en desacuerdo
- Ni de acuerdo, ni en desacuerdo
- Ligeramente de acuerdo
- Completamente de acuerdo

4. Considero que el patrón de diseño de interacción más adecuado no se encuentra dentro de los 9 patrones proporcionados. *

Marca solo un óvalo.

- Completamente en desacuerdo
- Ligeramente en desacuerdo
- Ni de acuerdo, ni en desacuerdo
- Ligeramente de acuerdo
- Completamente de acuerdo

Apéndice B

Cuestionario Evaluación de Recomendación de Patrones de Diseño de Interacción

Evaluación de Recomendación de Patrones de Diseño de Interacción

*Obligatorio

Favor de seleccionar la opción que más se adecue a su respuesta con respecto a la selección de patrón que acaba de realizar

1. Considero que el patrón de diseño de interacción recomendado es el apropiado para el requerimiento textual presentado. *

Marca solo un óvalo.

- Completamente en desacuerdo
- Ligeramente en desacuerdo
- Ni de acuerdo, ni en desacuerdo
- Ligeramente de acuerdo
- Completamente de acuerdo

2. Entendí lo que el requerimiento solicitaba en la primera lectura. *

Marca solo un óvalo.

- Completamente en desacuerdo
- Ligeramente en desacuerdo
- Ni de acuerdo, ni en desacuerdo
- Ligeramente de acuerdo
- Completamente de acuerdo

3. Considero que el patrón de diseño de interacción más adecuado no se encuentra dentro de los 9 patrones proporcionados. *

Marca solo un óvalo.

- Completamente en desacuerdo
- Ligeramente en desacuerdo
- Ni de acuerdo, ni en desacuerdo
- Ligeramente de acuerdo
- Completamente de acuerdo

Apéndice C

Cuestionario SUS (System Usability Scale, por sus siglas en inglés)

Cuestionario SUS

Escala de facilidad de uso del sistema

Instrucciones: Para cada uno de los enunciados seleccione la opción que mejor describa su reacción a la herramienta que se le acaba de presentar. Asume que dicha herramienta será desarrollado como un producto funcional.

***Obligatorio**

Dirección de correo electrónico *

1. Creo que me gustaría usar esta herramienta a menudo. *

Marca solo un óvalo.

- Completamente en desacuerdo
- Ligeramente en desacuerdo
- Neutral
- Ligeramente de acuerdo
- Completamente de acuerdo

2. La herramienta me pareció innecesariamente compleja. *

Marca solo un óvalo.

- Completamente en desacuerdo
- Ligeramente en desacuerdo
- Neutral
- Ligeramente de acuerdo
- Completamente de acuerdo

3. Me pareció que el sistema es fácil de usar. *

Marca solo un óvalo.

- Completamente en desacuerdo
- Ligeramente en desacuerdo
- Neutral
- Ligeramente de acuerdo
- Completamente de acuerdo

4. Creo que necesitaría la ayuda de un técnico para poder usar esta herramienta. *

Marca solo un óvalo.

- Completamente en desacuerdo
- Ligeramente en desacuerdo
- Neutral
- Ligeramente de acuerdo
- Completamente de acuerdo

5. Me pareció que las distintas funciones de esta herramienta estaban bien integradas. *

Marca solo un óvalo.

- Completamente en desacuerdo
- Ligeramente en desacuerdo
- Neutral
- Ligeramente de acuerdo
- Completamente de acuerdo

6. Me pareció que había demasiadas incoherencias en esta herramienta. *

Marca solo un óvalo.

- Completamente en desacuerdo
- Ligeramente en desacuerdo
- Neutral
- Ligeramente de acuerdo
- Completamente de acuerdo

7. Creo que casi todo el mundo aprendería a usar esta herramienta rápidamente. *

Marca solo un óvalo.

- Completamente en desacuerdo
- Ligeramente en desacuerdo
- Neutral
- Ligeramente de acuerdo
- Completamente de acuerdo

8. Me pareció que la herramienta es muy complicada de usar. *

Marca solo un óvalo.

- Completamente en desacuerdo
- Ligeramente en desacuerdo
- Neutral
- Ligeramente de acuerdo
- Completamente de acuerdo

9. Me encontraría muy a gusto al utilizar la herramienta. *

Marca solo un óvalo.

- Completamente en desacuerdo
- Ligeramente en desacuerdo
- Neutral
- Ligeramente de acuerdo
- Completamente de acuerdo

10. Tuve que aprender muchas cosas antes de poder hacerme con esta herramienta. *

Marca solo un óvalo.

- Completamente en desacuerdo
- Ligeramente en desacuerdo
- Neutral
- Ligeramente de acuerdo
- Completamente de acuerdo

Apéndice D

Cuestionario TAM (Technology
Acceptance Model, por sus siglas en
inglés)

Cuestionario TAM

Escala de percepción de utilidad y facilidad de uso.

Instrucciones: Para cada uno de los enunciados seleccione la opción que mejor describa su reacción a la herramienta que se le esta presentando. Asume que dicha visión tecnológica será desarrollada como un producto funcional.

***Obligatorio**

Dirección de correo electrónico *

1. Utilizando esta herramienta en mi trabajo podría llevar a cabo mis tareas más rápidamente. *

Marca solo un óvalo.

- Extremadamente en desacuerdo
- Muy en desacuerdo
- Ligeramente en desacuerdo
- Neutral
- Ligeramente de acuerdo
- Muy de acuerdo
- Extremadamente de acuerdo

2. Utilizando esta herramienta mejoraría mi desempeño laboral. *

Marca solo un óvalo.

- Extremadamente en desacuerdo
- Muy en desacuerdo
- Ligeramente en desacuerdo
- Neutral
- Ligeramente de acuerdo
- Muy de acuerdo
- Extremadamente de acuerdo

3. Utilizando esta herramienta haría que fueran más fáciles las tareas de mi trabajo. *

Marca solo un óvalo.

- Extremadamente en desacuerdo
- Muy en desacuerdo
- Ligeramente en desacuerdo
- Neutral
- Ligeramente de acuerdo
- Muy de acuerdo
- Extremadamente de acuerdo

4. Utilizando esta herramienta mejoraría mi actividad en el trabajo. *

Marca solo un óvalo.

- Extremadamente en desacuerdo
- Muy en desacuerdo
- Ligeramente en desacuerdo
- Neutral
- Ligeramente de acuerdo
- Muy de acuerdo
- Extremadamente de acuerdo

5. Utilizando esta herramienta incrementaría mi productividad. *

Marca solo un óvalo.

- Extremadamente en desacuerdo
- Muy en desacuerdo
- Ligeramente en desacuerdo
- Neutral
- Ligeramente de acuerdo
- Muy de acuerdo
- Extremadamente de acuerdo

6. Encontraría útil esta herramienta en mi trabajo. *

Marca solo un óvalo.

- Extremadamente en desacuerdo
- Muy en desacuerdo
- Ligeramente en desacuerdo
- Neutral
- Ligeramente de acuerdo
- Muy de acuerdo
- Extremadamente de acuerdo

7. Encontraría fácil que la herramienta hiciera lo que yo quiero que haga. *

Marca solo un óvalo.

- Extremadamente en desacuerdo
- Muy en desacuerdo
- Ligeramente en desacuerdo
- Neutral
- Ligeramente de acuerdo
- Muy de acuerdo
- Extremadamente de acuerdo

8. Interactuar con la herramienta sería claro y entendible. *

Marca solo un óvalo.

- Extremadamente en desacuerdo
- Muy en desacuerdo
- Ligeramente en desacuerdo
- Neutral
- Ligeramente de acuerdo
- Muy de acuerdo
- Extremadamente de acuerdo

9. Aprender a operar esta herramienta sería fácil para mí. *

Marca solo un óvalo.

- Extremadamente en desacuerdo
- Muy en desacuerdo
- Ligeramente en desacuerdo
- Neutral
- Ligeramente de acuerdo
- Muy de acuerdo
- Extremadamente de acuerdo

10. Es fácil llegar a ser hábil en el uso de la herramienta. *

Marca solo un óvalo.

- Extremadamente en desacuerdo
- Muy en desacuerdo
- Ligeramente en desacuerdo
- Neutral
- Ligeramente de acuerdo
- Muy de acuerdo
- Extremadamente de acuerdo

11. Encontraría fácil de usar la herramienta. *

Marca solo un óvalo.

- Extremadamente en desacuerdo
- Muy en desacuerdo
- Ligeramente en desacuerdo
- Neutral
- Ligeramente de acuerdo
- Muy de acuerdo
- Extremadamente de acuerdo

12. Encontraría flexible interactuar con la herramienta. *

Marca solo un óvalo.

- Extremadamente en desacuerdo
- Muy en desacuerdo
- Ligeramente en desacuerdo
- Neutral
- Ligeramente de acuerdo
- Muy de acuerdo
- Extremadamente de acuerdo