

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

HERRAMIENTA DE APOYO PARA ACTIVIDADES DE CARD SORTING

Eric Alonso Martínez Fonseca

Profesor Guía: Dr. Cristián Alexandru Rusu

Carrera: Ingeniería de ejecución en informática

Diciembre 2014

Dedicatoria

Dedico esta tesis a mis padres Jacqueline Verónica Fonseca Valdebenito y Patricio Ricardo Martínez Placencia quienes siempre me han apoyado de forma incondicional, moral y económicamente para lograr todas mis metas. Gracias a ustedes he llegado hasta aquí, es un privilegio ser su hijo.

A mi hermano Ignacio Eduardo Martínez Fonseca quien me ha motivado siempre a ser mejor persona, acompañándome en las buenas y en las malas.

A mis familiares y amigos quienes sin esperar nada a cambio compartieron su conocimiento, alegría y tristezas, y a todas aquellas personas que durante todo este tiempo estuvieron a mi lado apoyándome y lograron que este sueño se hiciera realidad.

Gracias a todos.

Índice

Resumen y Abstract	i
Resumen	i
Abstract	i
Índice de figuras	ii
Índice de Tablas.....	iii
Lista de Abreviaturas o Siglas	iv
1 Introducción.....	1
1.1 Descripción del tema	1
2 Definición de objetivos.....	2
2.1 Objetivos del Proyecto.....	2
2.1.1 Objetivo General.....	2
2.1.2 Objetivos Específicos.....	2
3 Marco Teórico.....	3
3.1 Interacción persona-computador	3
3.2 Usabilidad	3
3.3 Ordenamiento de tarjetas	5
3.3.1 Ventajas y desventajas del Card Sorting	6
3.3.2 Evaluación de prototipo de referencia.....	6
3.3.3 Desafíos para el Card Sorting virtual	7
3.3.4 Desarrollo de una prueba de ordenamiento de tarjetas.....	7
3.3.5 Análisis de resultados cualitativos y cuantitativos.....	8
4 Estudio de Factibilidad	11
4.1 Factibilidad Técnica.....	11
4.2 Factibilidad Económica	11
4.3 Factibilidad Operacional	12
4.4 Factibilidad Legal.....	12
5 Paradigma de proceso de software utilizado.....	13
6 Metodología de trabajo.....	15
7 Herramientas de desarrollo	16
7.1 Herramientas de análisis y diseño	16
7.2 Herramientas de programación.....	16
8 Requerimientos del prototipo.....	18

8.1	Requerimientos funcionales:	18
8.2	Requerimientos no funcionales:	18
9	Arquitectura del sistema	19
10	Modelado del sistema	20
10.1	Casos de uso	20
10.1.1	Diagrama de caso de uso general	20
10.1.2	Diagrama de caso de uso gestionar prueba	21
10.1.3	Diagrama de caso de uso revisar resultados individuales	22
10.1.4	Diagrama de caso de uso ver análisis de resultados	23
10.1.5	Diagrama de caso de uso realizar ordenamiento	25
10.1.6	Diagrama de caso de uso gestionar evaluadores	26
10.2	Diagrama de actividad	27
10.2.1	Diagrama de actividad: Revisar análisis	27
10.2.2	Diagrama de actividad: Gestión de pruebas	28
10.2.3	Diagrama de actividad: Realizar ordenamiento	28
10.3	Diagrama de secuencia	29
10.3.1	Diagrama de secuencia Revisar análisis	29
10.3.2	Diagrama de secuencia Gestionar pruebas	29
10.3.3	Diagrama de Secuencia Realizar ordenamiento	30
10.4	Diagrama de clases	31
11	Desarrollo del prototipo de sistema	32
11.1	Primer prototipo	32
11.2	Segundo prototipo	34
11.3	Tercer prototipo	36
12	Validación del prototipo	37
12.1	Participantes	37
12.2	Observaciones recibidas	37
12.3	Evaluación	37
12.4	Conclusión	38
13	Conclusiones y trabajo futuro	39
14	Referencias utilizadas	41

Resumen y Abstract

Resumen

En el desarrollo de sistemas software captar las reales necesidades del usuario es esencial para luego cumplir con las expectativas del usuario, es por esto que un análisis exitoso disminuye en gran medida los tiempo de trabajo en las etapas posteriores del proceso.

Para que la etapa de análisis sea exitosa se recurre a metodologías y técnicas del diseño centrado en el usuario, una de estas técnicas es la llamada ordenamiento de tarjetas, la cual se vale del modelo mental que el usuario registra a partir de su experiencia con el sistema para crear relaciones semánticas a través de conceptos.

El objetivo de este proyecto es desarrollar una herramienta de apoyo para la realización de pruebas de ordenamiento de tarjetas en pantallas táctiles en un entorno web. Con este software se busca facilitar el proceso de análisis de los resultados de las pruebas realizadas, agregando la sencillez de uso que pueden brindar las tecnologías táctiles.

Palabras claves: Ordenamiento de tarjetas, usabilidad, interacción persona computador, diseño centrado en el usuario.

Abstract

In the development of software systems to capture the actual user needs is essential to meet their expectations; therefore, a successful analysis greatly reduces work time in later stages.

Methods and techniques of user-centered designs are used so that the analysis stage is successful. One of these techniques is the "order of cards," which uses the mental model the user logs in from his experience with the system in order to create semantic relationships through concepts.

The objective of this project is to develop a support tool for testing card systems on touch screens in a web environment. This software aims to facilitate the process of analyzing the results of the tests, adding ease of use that tactile technologies can provide.

Key words: system of cards, usability, human-computer interaction, user-centered design.

Índice de figuras

Figura 3-1 Ejemplo método de Clustering jerárquico con criterio simple.	10
Figura 5-1 Modelo de desarrollo evolutivo	13
Figura 9-1 Arquitectura de tres capas	19
Figura 10-1 Diagrama de caso de uso general.....	20
Figura 10-2 Diagrama de caso de uso Gestión de pruebas	21
Figura 10-3 Diagrama de caso de uso Revisar resultados individuales	22
Figura 10-4 Diagrama de caso de uso Ver análisis de resultados.....	24
Figura 10-5 Diagrama de caso de uso Realizar ordenamientos	25
Figura 10-6 Diagrama de caso de uso Gestionar evaluadores.....	26
Figura 10-7 Diagrama de actividad Revisar análisis	27
Figura 10-8 Diagrama de actividad: Gestionar prueba	28
Figura 10-9 Diagrama de actividad: Realizar prueba	28
Figura 10-10 Diagrama de secuencia: Revisar análisis	29
Figura 10-11 Diagrama de secuencia: Gestionar prueba	29
Figura 10-12 Diagrama de secuencia: Realizar ordenamiento.....	30
Figura 10-13 Diagrama de clases	31
Figura 11-1 Prototipo gestionar proyectos	32
Figura 11-2 Prototipo tabla de coincidencias	32
Figura 11-3 Prototipo tabla de herencia	33
Figura 11-4 Prototipo gráfico Dendograma	33
Figura 11-5 Prototipo comentarios	34
Figura 11-6 Barra análisis de resultados	34
Figura 11-7 Tabla de coincidencias	35
Figura 11-8 Porcentaje de agrupación	35
Figura 11-9 Dendograma	35
Figura 11-10 Gestión de estudios	36

Índice de Tablas

Tabla 3-1 Tabla de ordenamientos.....	8
Tabla 3-2 Tabla co-ocurrencias	9
Tabla 3-3 Tabla de herencia	9
Tabla 10-1 Especificación formal: Gestión de pruebas	22
Tabla 10-2 Especificación formal: Revisar resultados individuales.....	23
Tabla 10-3 Especificación formal: Ver análisis de resultados	24
Tabla 10-4 Especificación formal: Realizar ordenamiento	26
Tabla 10-5 Especificación formal: Gestionar evaluadores	27

Lista de Abreviaturas o Siglas

HCI: Human Computer Interaction o Interacción Persona Computador.

ISO: International Organization for Standardization u Organización internacional de normalización.

IEC: International Electro technical Commission o Comisión Electrotécnica Internacional.

IBM: International Business Machines.

UML: Unified Modeling Language o Lenguaje unificado de modelado.

HTML: Hiper Text Markup Language o Lenguaje de marcas de hipertexto.

CSS: Cascading Style Sheets o hoja de estilo en cascada.

PHP: Acrónimo recursivo de Hypertext Pre-processor.

GNU: Acrónimo recursivo de GNU'sNot Unix.

GPL: General Public License o Licencia Pública General.

SQUARE: System and Software Quality Requirements Evaluation.

1 Introducción

1.1 Descripción del tema

En todo tipo de proyectos sean estos informáticos o no, donde se ofrece un producto personalizado, la etapa de análisis del sistema es crucial para el posterior desarrollo del proyecto. Y uno de los problemas más comunes en las etapas iniciales es reconocer las expectativas y necesidades del usuario. El no satisfacer al usuario desencadena problemas en el diseño del sistema, lo que provoca un cambio en la planificación y un aumento en el costo final del proyecto.

Cuando el usuario se enfrenta al sistema funcional es realmente cuando decide si seguirá utilizándolo, o buscará otra alternativa. Esto se puede deber a una mala disposición de las utilidades que el sistema ofrece y que muestra en la interfaz gráfica, por lo cual el usuario no percibe la totalidad de las funciones o no entiende cómo usarlas, bajando la eficiencia en la productividad y generando frustración al usuario. Por lo cual es necesario que la usabilidad del sistema genere un alto grado de satisfacción para el usuario.

En la actualidad existen diversas técnicas y estrategias para capturar los requerimientos de un sistema, como por ejemplo cuestionarios, estudios de documentación o entrevistas, entre otros. Pero estas soluciones pueden resultar poco prácticas para el usuario, dado que la falta de tiempo, recursos económicos y poca disponibilidad de los usuarios, dificultan la aplicación de estas técnicas. Por lo cual es necesario que la estrategia utilizada sea sencilla de aprender, fácil y rápida de utilizar, y que además entregue información relevante del sistema a desarrollar.

En el presente proyecto se propone el desarrollo de un prototipo de herramienta que facilite la obtención de los requerimientos del usuario, a través de la técnica de ordenamiento de tarjetas, conocida como CardSorting. Esta técnica consiste en la observación de como los usuarios agrupan y asocian entre si un numero predeterminado de tarjetas, etiquetadas en diferentes categorías temáticas del sistema. Este método es muy simple y útil, del cual se pueden obtener resultados cualitativos y cuantitativos, dependiendo del número de participantes.

En el mercado se ofrecen distintas herramientas para el desarrollo de CardSorting virtual, cada una con características propias, de ellas podemos reconocer tres módulos principales, un módulo para gestionar los estudios, uno para realizar los ordenamientos y otro para analizar los resultados, sin embargo estas herramientas están enfocadas en el evaluador de la prueba y no en el participante, quien ordena las tarjetas. Por lo que se desarrolló una herramienta pensada en mejorar la experiencia tanto del evaluador como del participante.

Es por esto que el proyecto se desarrolló para dispositivos táctiles, ofreciendo de este modo un mejor manejo de las tarjetas virtuales, ya que esta tecnología es más intuitiva para el usuario. Por otro lado se desarrolló como una aplicación web, facilitando al evaluador la disponibilidad de la información para realizar las pruebas correspondientes.

2 Definición de objetivos

2.1 Objetivos del Proyecto

Para la consecución de este proyecto se ha definido un objetivo general y tres objetivos específicos, los cuales se detallan a continuación. La realización de estos tres objetivos específicos permitieron que el proyecto llegue a buen término, marcando el camino hacia el cumplimiento del objetivo general.

2.1.1 Objetivo General

- Desarrollar un prototipo de herramienta para el apoyo de actividades de ordenamiento de tarjetas en dispositivos táctiles.

2.1.2 Objetivos Específicos

- Proporcionar una herramienta para la actividad de ordenamiento de tarjetas.
- Brindar un análisis de los ordenamientos realizados.
- Validar el prototipo propuesto.

3 Marco Teórico

3.1 Interacción persona-computador

La interacción persona computador (HCI del inglés Human Computer Interaction) es la disciplina que estudia el diseño, evaluación e implementación de sistemas informáticos interactivos para el uso de los seres humanos, y con el estudio de los fenómenos más importantes con los que está relacionado [6].

El tema principal de esta disciplina está en la interacción y más específicamente la interacción entre uno a más seres humanos y uno o más ordenadores. Aunque la situación clásica sería una persona sentada usando una aplicación en su ordenador, alrededor de las palabras interacción, persona y computador hay un amplio conjunto de opciones posibles, por lo cual no se puede asegurar una interacción convencional o un ambiente provechoso en el uso del sistema, por esto es importante que la interacción persona computador se produce dentro de un contexto social y organizacional y se requieren distintos tipos de aplicaciones para distintos propósitos.

Los objetivos de HCI son desarrollar o mejorar la seguridad, utilidad, efectividad, eficiencia y usabilidad de sistemas que incluyan computadoras. Cuando se menciona sistemas no se refiere tan solo al hardware y al software, sino también a todo el entorno.

Para desarrollar sistemas interactivos es necesario comprender como la gente trabaja y hace uso de los ordenadores, viendo factores psicológicos, ergonómicos, organizativos y sociales, y con esta información desarrollar herramientas y técnicas que ayuden a conseguir que los sistemas informáticos sean los idóneos y de esta forma conseguir una interacción eficiente, efectiva y segura, tanto a nivel individual como de grupo.

3.2 Usabilidad

La usabilidad es un término que proviene del inglés usability que significa facilidad de uso, y se refiere a la facilidad de un usuario de utilizar una herramienta para alcanzar un objetivo específico. Este neologismo puede ser utilizado para referirse a la interacción de un usuario con cualquier tipo de producto, sin embargo el uso más común es en el ámbito de la interacción persona computador.

Debido a que la usabilidad está involucrada en diversos ámbitos, existen diversas definiciones para el mismo concepto, por lo que desde la organización internacional para la estandarización (ISO) destacamos las siguientes definiciones de usabilidad:

- ISO/IEC 9126: "La usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso" [18].
- ISO/IEC 9241: "Medida en la que un producto puede ser usado por un grupo de usuarios determinados, para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso especificado" [19].

De lo anterior podemos distinguir que la norma ISO 9126 hace especial énfasis en los objetivos del usuario con el producto, mientras que el estándar ISO 9241 se enfoca en las características del producto para que el usuario pueda utilizarlo. Aunque el estándar

ISO/IEC 9126 ya no es vigente, ha sido reemplazado al estándar ISO/IEC 25000 [20] el cual se preocupa sobre la calidad del producto software y está basado en el estándar ISO/IEC 9126 y el estándar ISO/IEC 14598.

Para medir el grado de usabilidad de un producto existen cuatro conceptos principales atributos, paradigmas, principios y heurísticas de usabilidad.

- **Atributos de usabilidad:** Son las cualidades de la usabilidad.
 - Learnability (facilidad de aprendizaje): Rapidez con que un usuario aprende a utilizar un sistema con el que no ha tenido contacto anteriormente.
 - Efficiency (Eficiencia): El grado de productividad que puede lograr un usuario al saber utilizar el sistema.
 - Memorability (Memorabilidad): La facilidad para recordar cómo utilizar el producto luego de dejar de emplearlo durante un tiempo.
 - Errors (errores): La cantidad y gravedad de errores que comete el usuario junto con el deber de informar y ayudar al usuario.
 - Subjective Satisfaction (Satisfacción subjetiva): Referencia a si un usuario se siente cómodo o satisfecho con el producto.
- **Paradigmas de usabilidad:** características genéricas de un producto.
 - Acceso: El producto debe ser usable para personas con experiencia en el ámbito, pero no con el producto en sí.
 - Eficaz: El producto debe permitir un uso eficiente del tiempo de trabajo.
 - Avance: El producto debe permitir el avance continuo de nuevos conocimientos y habilidades del usuario.
 - Soporte: El producto debe apoyar al usuario para que éste cumpla los objetivos.
 - Contexto: El producto debe adaptarse a las condiciones de uso reales, en el entorno en el cual se va a utilizar.
- **Principios de usabilidad:** características específicas del producto.
 - Estructura: Debe tener una interfaz consistente, agrupando cosas relacionadas y separando las cosas diferentes.
 - Simplicidad: Simplificar la ejecución de tareas comunes, comunicando las cosas de manera simple y clara.
 - Visibilidad: Mostrar solo lo necesario al usuario, guardando tareas y herramientas para ciertas tareas visibles.
 - Retroalimentación: Mantener al usuario informado sobre las acciones que se están realizando.
 - Tolerancia: Debe reaccionar ante entradas o acciones inesperadas y ser flexible para deshacer o rehacer acciones.
 - Reusabilidad: Reutilizar los componentes y comportamientos para mantener consistencia con los propósitos.
- **Heurísticas de usabilidad:** son reglas generales para las propiedades comunes de las interfaces usables. Estas ofrecen pautas para comprender por qué los usuarios cometen determinados errores, un claro ejemplo de heurísticas son las heurísticas de Nielsen, inspiradas en aplicaciones web.

Luego de medir el grado de usabilidad del producto se pueden encontrar los principales problemas y confeccionar una estrategia para mejorar la usabilidad y consigo la calidad del producto [1][2].

3.3 Ordenamiento de tarjetas

El ordenamiento de tarjetas es una técnica que permitirá explorar el rotulado de los grupos, descubriendo el lenguaje de los usuarios y anticipar cuál será la organización de categorías o menús que mejor se adapte al modelo mental de los usuarios. Existen dos tipos de CardSorting: abierto y cerrado. En el “Card Sorting” abierto el usuario puede agrupar las categorías libremente en el número de conjuntos que crea necesario; mientras que en el cerrado, los grupos o conjuntos están predefinidos y etiquetados y el usuario únicamente deberá colocar cada categoría en el grupo que crea le corresponda. Este segundo tipo de “Card Sorting” está recomendado para verificar si un diseño de información es familiar y comprensible para el usuario, mientras que el “abierto” tiene el objetivo de descubrir qué tipo de clasificación de categorías sería más correcto utilizar.

La cantidad y calidad de la información que podamos extraer del empleo de esta técnica dependerán del tipo de análisis que realicemos. Entre los tipos de análisis, podemos distinguir entre cualitativos y cuantitativos. En el análisis cualitativo, el responsable de realizar la prueba prestará especial atención a todos los aspectos relativos a cómo cada usuario en concreto organiza las tarjetas, qué problemas tiene para realizar la tarea, qué categorías agrupa inmediata e intuitivamente y sobre cuáles duda más, qué preguntas hace durante la prueba, etc. El análisis cuantitativo, en cambio, se basa en la aplicación de técnicas estadístico automáticas sobre los grupos resultantes de la agrupación realizada por los usuarios.

La técnica se puede realizar manualmente, mediante tarjetas de papel o cartón en un lugar físico, pero también mediante aplicaciones de software con tarjetas virtuales en un ambiente simulado, y aunque el uso de dichas herramientas informáticas puede mejorar en ciertos aspectos la experiencia de la prueba, podría suponer en sí una barrera para los usuarios quienes no manejan las tarjetas virtuales con la misma facilidad y comodidad que una tarjeta de papel.

La ventaja del sistema manual con tarjetas de papel en una mesa es que es más natural e intuitivo para los usuarios que utilizar un programa. El problema es que el trabajo manual de registro de las agrupaciones y el posterior análisis estadístico para generar las categorías más adecuadas es muy laborioso.

Las aplicaciones de software tienen la ventaja de tener la capacidad para guardar automáticamente las agrupaciones de cada usuario, realizar rápidamente el análisis de datos de los ordenamientos realizados y entregar los resultados estadísticos requeridos sin la necesidad de que el evaluador tenga los conocimientos para realizar dicho análisis.

Algunas aplicaciones utilizadas como referencia son CardZort, UzCardSort, EZSort, UXSort, OptimalSort, entre otras [4] [5]. Al realizar pruebas en las distintas aplicaciones se pueden apreciar las ventajas y desventajas de cada una. Por ejemplo mientras la aplicación UXSort debe ser descargada e instalada para ser ejecutada, otras como OptimalSort solo requieren ingresar al sitio web y empezar a trabajar.

3.3.1 Ventajas y desventajas del Card Sorting

A continuación se presentaran las ventajas y desventajas del Card Sorting del tipo virtual, las cuales han sido consideradas luego de ocupar las herramientas de referencia mencionadas en el punto anterior:

Ventajas

- **Más pruebas en menos tiempo:** Realizar más pruebas en menos tiempo, porque sus estudios en línea están en marcha rápidamente. Estos son menos recursos y proporcionar más fácil la recopilación de datos automatizada y el análisis.
- **Más participantes:** Se pueden probar un número mayor de participantes para un mayor nivel de confianza en los datos.
- **Reducir el Costo:** No hay necesidad de alquilar espacio en el laboratorio o compensar los participantes los gastos de viaje. Los investigadores no están obligados a moderar las sesiones.
- **Más rápido análisis de datos:** Analizar los datos es más rápido. Análisis estadístico para construcción y representación de los datos se lleva a cabo de inmediato. Se ahorra el tiempo de analizar manualmente los datos.

Desventajas

- **No considera las tareas del usuario:** Las herramientas consultadas no consideran datos cualitativos del estudio.
- **Los resultados pueden ser muy variables:** Factores como la experiencia de los usuarios y la dimensión de la prueba pueden afectar la calidad de los resultados.
- **Poca comunicación entre usuario y evaluador:** No hay ayuda inmediata de parte del evaluador.

3.3.2 Evaluación de prototipo de referencia

Para iniciar el análisis del desarrollo de la herramienta se uso como referencia un modelo propuesto por un alumno de la Pontificia Universidad Católica de Valparaíso, a continuación se explicará a grandes rasgos el funcionamiento de la herramienta de apoyo para actividades de Card Sorting usado como referencia.

La herramienta evaluada solo puede ser accedida por usuarios previamente ingresados al sistema y como resumen es una herramienta de gestión de proyectos, cada proyecto contiene una lista de estudios y cada estudio contiene una serie de pruebas completadas (los ordenamientos realizados).

Primero evaluaremos los aspectos positivos de la herramienta: se puede ver las instrucciones para utilizar las opciones del sitio, permite solamente al administrador gestionar evaluadores, permite al evaluador gestionar proyectos, estudios y pruebas; diferencia las acciones que puede realizar con diferentes colores, además de ver la URL de la prueba para enviarla a los usuarios. En la prueba entrega las instrucciones de estas, las

tarjetas están en una misma columna y permite utilizar libremente las tarjetas y los grupos como el usuario desee.

Por otro lado los aspectos negativos de la herramienta: no se puede crear libremente una cuenta de evaluador, no hay una opción de "olvidé mi contraseña", algunas instrucciones están mal redactadas, problemas con la interfaz del sitio, no utiliza imágenes y/o símbolos a las opciones, no tiene buscar (por ejemplo buscar una tarjeta en específico), no se puede hacer zoom, mal orden de los elementos del sitio, no tiene "miga de pan" para que el evaluador sepa donde se encuentra y pueda volver, hay opciones que están ocultas como el volver y para finalizar, hay palabras u opciones que están demás en el sitio.

3.3.3 Desafíos para el Card Sorting virtual

El modo en que los usuarios han organizado las etiquetas nos debe servir para reflexionar acerca de la necesidad de comprobar o testear con los propios destinatarios cualquier diseño de un espacio virtual, más allá de las ideas preconcebidas acerca de cuál sería la óptima organización de los recursos en este espacio. Pero incluso más relevante es la metodología propuesta para el análisis de los resultados del "Card Sorting", destinada a simplificar y facilitar la comprensión del modelo mental del usuario al desarrollador. Esta técnica junto a algoritmos de agrupamiento resultan de gran utilidad para la interpretación de resultados y se complementan perfectamente entre sí para esta tarea.

Por otra parte se debe considerar satisfacción del usuario al realizar la prueba de Card Sorting, dado que realizarla de forma manual es más intuitivo para él. Para lo cual se propone incorporar tecnologías que ayuden al usuario a interactuar de forma más intuitiva con el sistema, simulando la experiencia de una prueba manual con tecnología táctil.

3.3.4 Desarrollo de una prueba de ordenamiento de tarjetas

El primer paso para realizar una prueba de ordenamiento de tarjetas es reclutar a los participantes de la prueba, en la selección se deben considerar participantes representativos con características y los perfil de usuarios del sistema estudiado. El número de participantes óptimo para realizar una prueba es de 20 o 30 personas aproximadamente [21], ya que un número menor no es representativo y un número mayor resulta un gasto excesivo de recursos.

Al principio de la prueba se debe explicar a los participantes el objetivo de la prueba, y que es el sistema el evaluado, no los participantes. Además de indicar que el criterio de agrupación de categorías es por similitud. Luego de esto se muestran al usuario las distintas tarjetas para que este pueda ordenar. Las tarjetas deben contener nombres claros y no ambiguos, para evitar confusión en los participantes, el número de tarjetas puede variar dependiendo del sistema a estudiar.

El participante puede agrupar y ordenar las tarjetas como estime conveniente y además puede crear y/o eliminar las tarjetas, categorías y sub-categorías que considere necesarias. El participante tiene la opción de guardar los cambios realizados para

continuarlos posteriormente o finalizar la prueba, no es necesario agrupar todas las tarjetas pues puede considerarse que no son necesarias en el ordenamiento. Antes de terminar la prueba se brinda la opción al participante de escribir un comentario sobre la prueba, de esta forma se obtiene una retroalimentación de parte del usuario.

3.3.5 Análisis de resultados cualitativos y cuantitativos

Como se mencionó previamente, existen dos tipos de análisis de resultados el análisis cualitativo y el análisis cuantitativo.

El análisis cualitativo en una prueba de Card Sorting virtual es difícil de implementar, por lo cual se limita a los **comentarios que puedan agregar los usuarios antes de terminar la prueba**. Este análisis se puede mejorar si la prueba se realiza en un ambiente controlado, como un laboratorio acondicionado para dicha actividad, donde el participante pueda ordenar las tarjetas de forma tranquila y si tiene alguna duda pueda pedir ayuda a algún encargado.

El análisis cuantitativo consiste en el procesamiento estadístico de los datos, para luego analizar los resultados obtenidos, para dicho efecto se crean distintas tablas para ordenar los resultados de las pruebas.

La primera tabla representa los ordenamientos de forma individual, es decir, de un solo ordenamiento y sirve para tener una representación numérica del ordenamiento realizado por el usuario.

ID	Nombre	ID Padre	Orden Padre	Orden	Sub Padre	Sub Orden

Tabla 3-1 Tabla de ordenamientos

La tabla 3-1 se utiliza para verificar los datos de cada tarjeta, su relación con la tarjeta padre y la profundidad de estos, y con estos datos es posible representar en la tabla el resultado del ordenamiento. El ID hace referencia a un identificador único para la tarjeta facilitando el análisis numérico, el orden hace referencia a la profundidad de la tarjeta, es decir en cuantos subgrupos está contenido y Sub hace referencia al identificador subconjunto al que pertenece la tarjeta. Esta tabla se rellena con los datos obtenidos en un ordenamiento, por lo que existirán tantas tablas como ordenamientos se realicen, y es un acercamiento a la digitalización de los datos del ordenamiento.

Una vez identificado los datos de cada ordenamiento realizado en el estudio se puede iniciar el análisis de los datos del estudio en general, se inicia con la confección de la tabla de co-ocurrencias. En esta tabla el número de filas y columnas es igual al número de conceptos del estudio, en cada celda se indica el número de veces que el concepto de una

fila ha sido ordenado junto a los conceptos de las columnas siguientes, construyendo así una matriz simétrica como la siguiente.

	Concepto 1	Concepto 2	Concepto 3
Concepto 1	---	15	9
Concepto 2	15	---	3
Concepto 3	9	3	---

Tabla 3-2 Tabla co-ocurrencias

Estos resultados pueden ser medidos probabilísticamente, obteniendo un primer resultado para el análisis del estudio, de forma que se considera el porcentaje dado el número total de pruebas del estudio, en el caso de ejemplo de la tabla 3-2 si el total de pruebas realizadas hubiese sido quince se puede concluir que el 100% de las veces el concepto 1 y el concepto 2 se ordenaron juntos, es decir, estuvieron en la misma categoría, el 60% de las veces el concepto 1 y el concepto 3 estuvieron en la misma categoría y el 20% de las veces el concepto 2 y el concepto 3 estuvieron agrupados juntos.

Así se pueden distinguir patrones de agrupación según los porcentajes obtenidos:

- Entre 90% y 100% el porcentaje de agrupación es Excelente.
- Entre 80% y 89% el porcentaje de agrupación es Muy bueno.
- Entre 70% y 79% el porcentaje de agrupación es Bueno.
- Entre 60% y 69% el porcentaje de agrupación es Aceptable.
- 59% o menos el porcentaje de agrupación es No suficiente.

De este modo se considera que las tarjetas que tengan un porcentaje de agrupación aceptable o superior, deberán formar parte de las categorías o grupos, para hacer más llamativo el resultado se pueden incluir códigos de colores para cada sector de porcentajes.

De igual modo se puede construir una tabla que denominamos tabla de herencia, donde en cada celda se indica el número de veces que el concepto de una fila fue padre de un concepto de alguna columna, construyendo así una matriz asimétrica como la siguiente.

	Concepto 1	Concepto 2	Concepto 3
Concepto 1	---	0	0
Concepto 2	12	---	2
Concepto 3	8	0	---

Tabla 3-3 Tabla de herencia

Al igual que la tabla anterior estos datos pueden ser medidos probabilísticamente, obteniendo otro tipo de resultados para el análisis del estudio. En el caso del ejemplo de la tabla 3-3 suponiendo un total de dieciséis pruebas realizadas, se observa que el 75% de las veces el concepto 2 fue padre del concepto 1, el 50% de la veces el concepto 3 fue padre del concepto 1 y el 12,5% de las veces el concepto 2 fue padre del concepto 3.

Con lo cual también se puede crear patrones de agrupación según los porcentajes obtenidos, aunque en este caso se pueden crear intervalos más extensos, dado que los

porcentajes suelen ser menores. De igual manera se pueden incluir códigos de colores para cada sector de porcentajes.

En casi la totalidad de los programas que existen actualmente el método escogido para analizar los datos obtenidos es el Clustering, en el presente proyecto también incluiremos este método para analizar resultados, por lo cual se analizará a continuación.

El método de Clustering consiste en agrupar los elementos en subgrupos ordenados según la cercanía dichos elementos. La agrupación se realiza en función de un parámetro llamado distancia[23], para obtener este parámetro utilizaremos los mismos datos de la tabla de co-ocurrencias transformando los valores a probabilidades, de este modo se puede saber la cercanía entre los conceptos, mientras más cercano a 1 el valor, será mayor la concordancia entre estos y menor la distancia.

Existen distintos tipos de algoritmos de Clustering, el método utilizado en el presente proyecto es el denominado **Clustering jerárquico con criterio simple**. Este tipo de Clustering se basa en la unión de los dos clúster más cercanos entre sí, se entiende como clúster a la unión en grupos de los conceptos cercanos entre sí, esta proximidad se mide en base al criterio de enlace simple, donde dos clúster's se unen si la distancia entre ellos es la mínima existente entre dos de sus elementos. A partir de esto se obtiene un diagrama de árbol llamado Dendograma [22].

Un Dendograma es una representación gráfica en forma de árbol, que agrupa los conceptos desde los más cercano al más lejano, separando los conceptos en subgrupos llamados cluster.

Para realizar este análisis se utilizará el criterio de enlace simple, donde se busca el concepto dentro del clúster más cercano a los otros conceptos o clusters. Es decir, cada clúster busca el clúster con el valor más cercano a 1 para enlazarse con él.

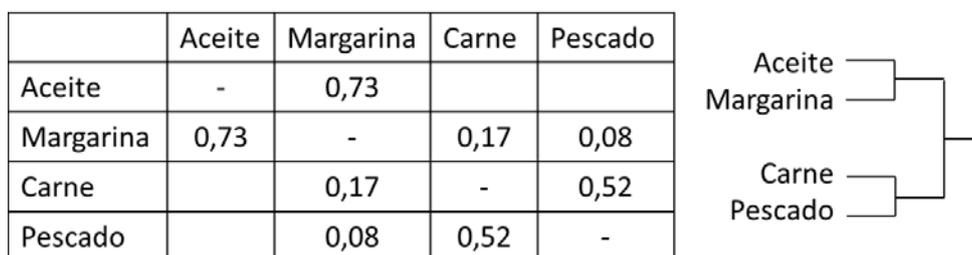


Figura 3-1 Ejemplo método de Clustering jerárquico con criterio simple.

En este caso del ejemplo de la figura 3-1, se observa que en la tabla de co-ocurrencia la relación más cercana al valor 1 es la relación Aceite – Margarina, por lo que forman un cluster, luego viene la relación Carne – Pescado, por lo que forman un nuevo cluster, el siguiente valor es el dado por la relación Margarina – Carne, pero en este caso como estos conceptos ya pertenecen a un cluster se enlazan los clusters de estos conceptos y no los conceptos por separado. Como resultado se puede observar en la Figura 3-1 el diagrama denominado Dendograma.

4 Estudio de Factibilidad

Como un estudio previo para evaluar cualquier proyecto, se debe llevar a cabo un previo estudio de factibilidad del proyecto. El resultado de dicho estudio es un informe que indica si es o no conveniente el llevar a cabo el proceso considerado en el proyecto.

El estudio de factibilidad considera los siguientes aspectos: la factibilidad técnica, económica y operacional. A su vez, también se considera si el proyecto es o no apropiado dado factores políticos y/o legales.

4.1 Factibilidad Técnica

Se refiere a los recursos necesarios como herramientas, conocimientos, habilidades, experiencia, etc., que son necesarios para efectuar las actividades o procesos que requiere el proyecto.

Para la realización del proyecto se contó con un notebook y un computador de escritorio para el desarrollo del proyecto, además de un computador táctil para realizar las respectivas pruebas del programa.

Se trabajó con el sistema operativo Windows en su versión 7 y otros software's de libre pago para el desarrollo del sistema. Es así que el grupo de trabajo debe tener los conocimientos necesarios para desarrollar una herramienta para pantallas táctiles en un entorno web.

4.2 Factibilidad Económica

Los estudios de factibilidad económica incluyen análisis de costos y beneficios asociados con cada alternativa del proyecto.

El proyecto "Herramienta de apoyo para actividades de CardSorting" tiene una finalidad puramente educativa y académica, por lo cual un análisis económico no es aplicable. Situación que se encuentra sujeta a través de una actividad académica para el Ramo Proyecto 1 de la carrera de Ingeniería Ejecución en Informática, en el que a su vez, tampoco percibe una finalidad lucrativa.

Costos asociados a la adquisición de software y hardware:

Se utilizaran herramientas de libre uso como JQuery Mobile. Por lo tanto, las herramientas que se deben utilizar tienen que cumplir con la misma característica; es por ello que no estará sujeto a costos monetarios en el desarrollo del sistema.

En este proyecto no hubo costos asociados a la compra de hardware, al ser una aplicación Web, los usuarios que usen el sistema, sólo necesitaran un equipo con conexión a Internet y que posea pantalla táctil.

4.3 Factibilidad Operacional

Se refiere a todos aquellos recursos donde interviene algún tipo de actividad, depende de los recursos humanos que participen durante la operación del proyecto. Durante esta etapa se identifican todas aquellas actividades que son necesarias para lograr el objetivo, se evalúa y determina todo lo necesario para llevarla a cabo.

En el presente proyecto el grupo de trabajo considera que cumple con los conocimientos básicos para desarrollar las actividades necesarias para llevar a cabo el desarrollo del sistema, y en el caso de no poseer el conocimiento, el plazo propuesto es suficiente para adquirir dicho conocimiento.

4.4 Factibilidad Legal

El objetivo de la Factibilidad Legal es el poder verificar que, al desarrollar un sistema, éste no incurre en infracciones, violaciones u otros delitos que podrían implicar la imposibilidad de poner en práctica y/o interrumpir el funcionamiento del sistema.

Para el presente proyecto no existen trabas legales que impidan el buen desempeño y funcionamiento del software, todo esto mientras no infrinja la propiedad intelectual de terceros.

Otro punto importante es velar por la confidencialidad de los datos de los usuarios que utilicen esta herramienta, al igual que al realizar una prueba de ordenamiento de tarjetas de forma manual el participante debe estar informado que los datos que entregue en la prueba solo serán confidenciales y ocupados solo para el uso por el cual son requeridos.

5 Paradigma de proceso de software utilizado

En la ingeniería de software el proceso de software es un conjunto de actividades y resultados asociados, que generan un producto software, las cuales son llevadas a cabo por los ingenieros de software, quienes incorporan estrategias de desarrollo para llevar a cabo su cometido. Para esto se selecciona un paradigma de proceso de software, los cuales son elegidos según la naturaleza propia del proyecto.

Un modelo de proceso de software es una representación abstracta de un proceso de software, puede pensarse en ellos como marcos de trabajo del proceso que pueden ser extendidos y adaptados para crear procesos más específicos de ingeniería de software. Dentro de los paradigmas estudiados el elegido fue el modelo de desarrollo evolutivo.

Desarrollo evolutivo: Este enfoque entrelaza las actividades de especificación, desarrollo y validación. Un sistema inicial se desarrolla rápidamente a partir de especificaciones abstractas. Este se refina basándose en las peticiones del cliente para producir un sistema que satisfaga sus necesidades.

Al desarrollar un prototipo rápidamente se pueden desatenderse aspectos importantes del sistema o se pueden tomar decisiones de desarrollo poco convenientes y el cliente al ver este primer prototipo se puede crear expectativas y se muestre reacio a reconstruirlo para asegurar la calidad del sistema, lo que convertiría el prototipo inicial en uno evolutivo, pero partiendo de un estado poco recomendado. Este modelo es útil cuando el cliente conoce los objetivos generales para el software, pero no identifica los requisitos detallados del mismo. También ofrece un mejor enfoque para el desarrollador al mezclar las cualidades de cada prototipo presentado.

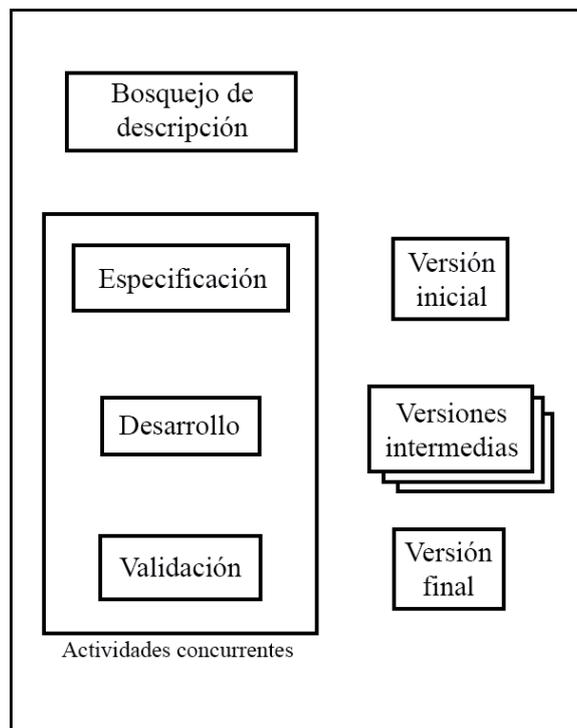


Figura 5-1 Modelo de desarrollo evolutivo

El **modelo de desarrollo evolutivo**, también conocido como paradigma de desarrollo basado en prototipos, fue elegido teniendo en cuenta el análisis realizado a los diversos paradigmas estudiados y en base a los requerimientos del presente proyecto, ya que como se menciona anteriormente este paradigma permite ir comprendiendo de mejor manera los requerimientos a medida que avanza el proyecto.

Al realizar diversos prototipos se puede validar continuamente que se cumple con los requerimientos, dado que cada versión pasa por un proceso de análisis, desarrollo y validación, de esta forma aunque la versión inicial presente una estructura deficiente esta puede ser mejorada y los requerimientos son comprendidos de mejor manera, a medida que se desarrollan más prototipos.

No necesariamente todos los prototipos deben contener todas las funcionalidades del sistema, pueden existir prototipos desarrollados para validar cierto módulo del sistema, de esta forma entre diversos módulos pueden entrelazar sus fortalezas. Esto permite desarrollar de forma acumulativa y poner énfasis en distintas funcionalidades a lo largo del desarrollo.

6 Metodología de trabajo

Al desarrollar un producto software una de las decisiones más importantes del analista es elegir la metodología que resulte más apropiada y eficiente para la realización del proyecto, esto dado a la amplia variedad en la naturaleza de los proyectos de desarrollo de software. Es importante considerar el tipo de proyecto y usuario, y también el tipo de herramienta que se desea ocupar para el desarrollo del software.

Al hablar sobre este tema podemos diferenciar tres tipos de desarrollo, desarrollo convencional, estructurado y orientado a objetos.

Desarrollo no estructurado: No se sigue una metodología formal, por lo cual los resultados finales son inesperados, no hay forma de controlar lo que está sucediendo en el proyecto y los cambios organizativos afectan negativamente al proceso de desarrollo.

Desarrollo estructurado: en esta metodología la programación, el diseño y el análisis se hacen de forma estructurada, tomando como elemento básico de desarrollo el módulo representados de forma jerárquica y descendente, donde se visualiza el sistema como entrada – proceso – salida, otorgando así mayor nivel de abstracción.

Desarrollo orientado a objetos: Su esencia es la identificación y organización de conceptos del dominio de la aplicación y no tanto su representación final en un lenguaje de programación. Con esto aparece una nueva forma de concebir los lenguajes de programación y su uso al incorporarse bibliotecas de clases y otros componentes reutilizables. Hay un alto grado de iteración y solapamiento, lo que lleva a una forma de trabajo muy dinámica, debido a su naturaleza iterativa es fácil dividir el sistema en varios subsistemas independientes, es decir, construir software complejos a partir de unidades de software modular y reutilizable.

En el presente proyecto se decidió utilizar una **metodología de desarrollo orientado a objetos**, dado que esta metodología propicia la creación de prototipos, facilita la creación de interfaces gráficas, y al fomentar la reutilización y extensión del código fuente permite crear sistema más complejos, por lo tanto agiliza el desarrollo y mantención del software [15].

7 Herramientas de desarrollo

7.1 Herramientas de análisis y diseño

Permiten al desarrollador crear un modelo del sistema que se va a construir y también la evaluación de validez y consistencia del modelo. Proporcionan un grado de confianza en la representación del análisis y ayudan a eliminar errores con anticipación.

Para el análisis y diseño del sistema se decidió utilizar el **Lenguaje Unificado de Modelado** (conocido como UML por sus siglas en inglés Unified Modeling Language) el cual es un lenguaje de modelado de sistema de software. Es un lenguaje gráfico que permite graficar, especificar, construir y documentar un sistema, esto incluye aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de base de datos y componentes reciclados.

La estandarización de un lenguaje de modelado es invaluable, ya que es la parte principal del proceso de comunicación que requieren todos los agentes involucrados en un proyecto informático. Si se quiere discutir un diseño con alguien más, ambos deben conocer el lenguaje de modelado y no así el proceso que se siguió para obtenerlo. Es por este motivo que el grupo de trabajo ha decidido modelar el sistema bajo este lenguaje, dado que además es el más conocido de los lenguajes de modelado [14].

7.2 Herramientas de programación

Este tipo de herramientas son aquellas que permiten realizar aplicativos, programas, rutinas, utilitarios y sistemas para que la parte física del computador u ordenador, funcione y pueda producir resultados. Se engloban aquí los compiladores, los editores y los depuradores de los lenguajes de programación convencionales.

Para el desarrollo del sistema, considerando una arquitectura de 3 capas, se trabajará con las siguientes herramientas:

HTML5: Es la quinta revisión de HTML (HiperText Markup Language), que es un lenguaje de marcado para la elaboración de páginas web. Esta versión todavía se encuentra en fase experimental, aunque ya es utilizado por múltiple desarrolladores web por sus mejoras y ventajas, se utilizará esta versión de HTML por su facilidad para arrastrar objetos como imágenes (proceso conocido como drag and drop) [9].

CSS3: Acrónimo de Cascading Style Sheets en su versión 3, es un lenguaje de hojas de estilo usado para describir la presentación semántica (aspecto y formato) de un documento escrito en lenguaje de marcas. Su aplicación más común es dar estilo a páginas webs escritas en lenguaje HTML o XHTML, aunque también da soporte a formatos como XML. Añadir un estilo a un sitio web es esencial para mejorar principios de la usabilidad del mismo como estructura, simplicidad y visibilidad [11].

PHP: Acrónimo de PHP Hypertext Pre-processor, es un lenguaje interpretado de propósito general ampliamente usado, diseñado especialmente para desarrollo Web, puede ser incrustado en HTML, desplegado en la mayoría de los servidores Web y en casi todos los sistemas operativos [10].

MySQL: Es un sistema de gestión de base de datos relacional, multihilo y multiusuario, por un lado se ofrece bajo la GNU GPL (del inglés GNU General Public License) para

cualquier uso compatible con esta licencia, pero también se puede adquirir una licencia específica para incorporarlo en productos privativos [12].

Xampp: Es un paquete de software con todo lo necesario para montar un prototipo de sistema, incluye Apache, MySQL, PHP, phpMyAdmin, Filezilla FTP Server, entre otras aplicaciones [13].

Eclipse: Es un entorno de desarrollo integrado compuesto por un conjunto de herramientas de código abierto multiplataforma, provee todas las herramientas para desarrollar un proyecto PHP.

8 Requerimientos del prototipo

8.1 Requerimientos funcionales:

- El administrador puede crear, modificar o eliminar los evaluadores.
- El evaluador puede crear, modificar o eliminar un proyecto en particular.
- El sistema permitirá al evaluador ver y/o exportar los resultados de la evaluación.
- El evaluador puede ver el análisis de resultados de un estudio en particular.
- El participante puede generar categorías a través del agrupamiento de tarjetas.
- El participante según su criterio puede agrupar las tarjetas dentro de una categoría o mover una tarjeta de una categoría a otra.
- El participante puede crear, renombrar o eliminar nuevas tarjetas o grupos.
- El participante puede agrupar categorías dentro de categorías, creando sub-categorías.
- El participante puede realizar una búsqueda de los conceptos en la prueba.
- El participante puede ver cuántas tarjetas quedan por agrupar.
- El participante puede ingresar comentarios sobre la prueba realizada.
- El participante puede guardar su avance para continuarlo posteriormente.

8.2 Requerimientos no funcionales:

- **Facilidad de uso:** El sistema debe ser de fácil uso y ejecución por parte de los usuarios, así como de fácil adaptación de la entidad con el mismo. Por lo cual se le ofrecerá ayuda al usuario en los aspectos que se pueda generar confusión y en las actividades que este realice.
- **Robustez:** El sistema debe estar en capacidad de permitir en el futuro su fácil mantenimiento con respecto a los posibles errores que se puedan presentar durante la operación del sistema.
- **Portabilidad:** El sistema debe ser compatible en los distintos dispositivos que cuenten con conexión a internet.

9 Arquitectura del sistema

La arquitectura del software es el diseño de más alto nivel de la estructura de un sistema, e indican la estructura, funcionamiento e interacción entre la partes del software. Una arquitectura de software se selecciona y diseña con base a objetivos y restricciones. Los objetivos son aquellos prefijados para el sistema de información, pero no solamente los objetivos funcionales, sino que también aspectos como mantenibilidad, flexibilidad e interacción con otros sistemas de información. Las restricciones son aquellas limitaciones derivadas de las tecnologías disponibles para implementar el sistema de información.

La elección de la arquitectura para el desarrollo de nuestro software se tomó en base a la mantenibilidad, flexibilidad y compatibilidad con sistemas web que ofrece la Arquitectura de 3 capas frente a otras alternativas como puede ser la arquitectura monolítica o cliente – servidor. La arquitectura elegida debe su nombre a la implementación de capas, cada una con un rol diferente. Se diferencia da la arquitectura de cliente – servidor en el objetivo primordial de la separación de la lógica de diseño de la lógica de negocio, de esta forma el desarrollo se puede llevar a cabo en varios niveles y al encontrar un error se soluciona solo el nivel requerido sin tener que revisar código entremezclado. A continuación se describen las capas del sistema [16].

- **Capa de Presentación:** Esta capa se encarga de proveer una interfaz entre el sistema y el usuario. Se responsabiliza de que se le comunique información al usuario por parte del sistema y viceversa.
- **Capa de Negocio:** Es la capa en la cual se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.
- **Capa de Datos:** En último caso, manejaremos la capa en donde almacenaremos los datos. La capa de negocio, se puede encargará de ofrecer, modificar, almacenar, borrar y recuperar datos, mediante el gestor de bases de datos diseñado para la aplicación que desarrollaremos.

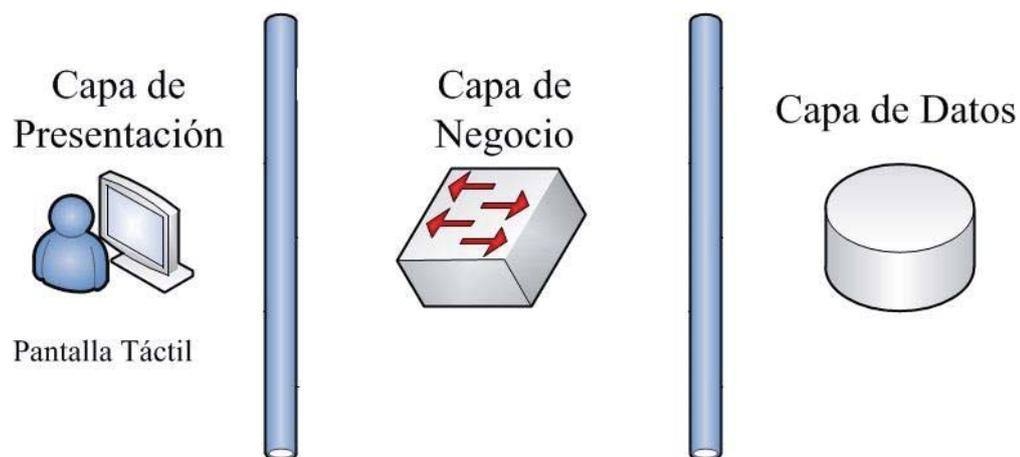


Figura 9-1 Arquitectura de tres capas

10 Modelado del sistema

10.1 Casos de uso

El modelado de Casos de Uso es la técnica más efectiva y a la vez la más simple para modelar los requisitos del sistema desde la perspectiva del usuario. Los Casos de Uso se utilizan para modelar cómo un sistema o negocio funciona actualmente, o cómo los usuarios desean que funcione. No es realmente una aproximación a la orientación a objetos; es realmente una forma de modelar procesos. Es, sin embargo, una manera muy buena de dirigirse hacia el análisis de sistemas orientado a objetos. Los casos de uso son generalmente el punto de partida del análisis orientado a objetos con UML.

El modelado de casos de uso se divide en una representación gráfica y una especificación formal, el diagrama de caso de uso muestra de forma gráfica la interacción de los actores con el sistema, mientras que la especificación formal se enfoca en describir la interacción paso a paso [14].

10.1.1 Diagrama de caso de uso general

En este diagrama se muestra de forma general la interacción de los actores con el sistema, se reconocen dos tipos de actores principales el evaluador y el participante, aunque también se identifica una extensión del evaluador llamado administrador, el cual se encarga de gestionar a los evaluadores. El actor evaluador se encarga de preparar la prueba y evaluar resultados y el actor participante es quien realiza el ordenamiento de tarjetas. Diagrama en Anexo B, sección Diagramas de casos de uso.

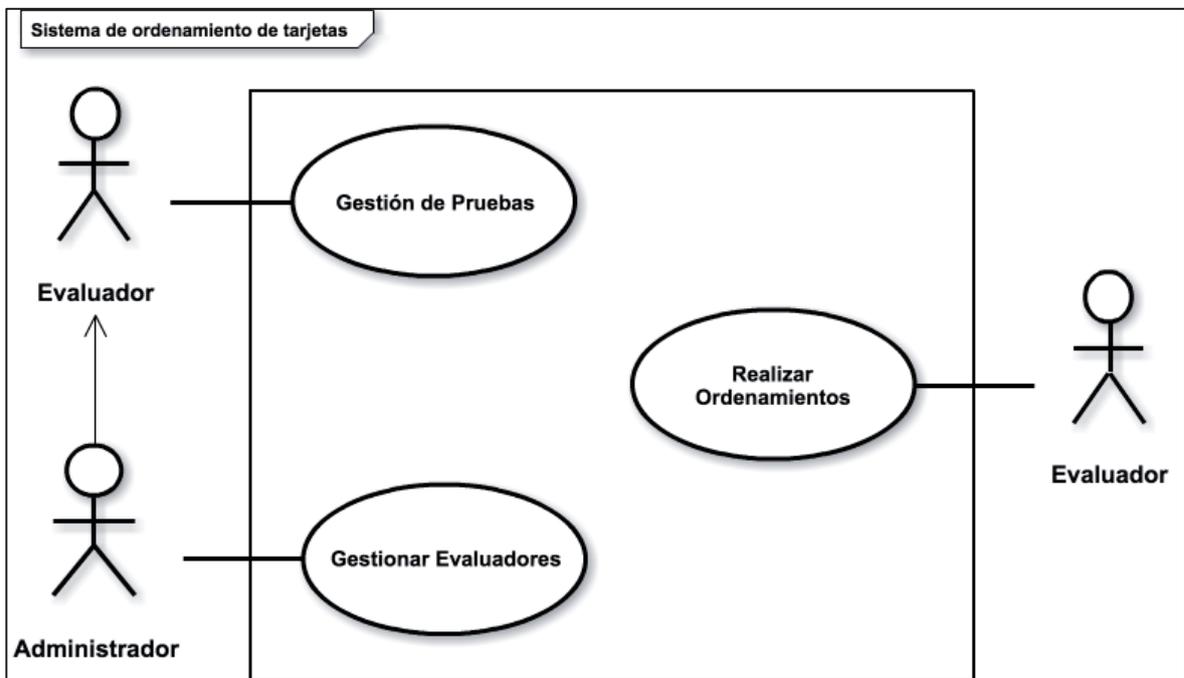


Figura 10-1 Diagrama de caso de uso general

10.1.2 Diagrama de caso de uso gestionar prueba

En este diagrama se muestra un mayor detalle sobre la gestión de la prueba por parte del actor evaluador. Acá se diferencia entre un estudio y ordenamiento, un estudio puede contener más de un ordenamiento y el ordenamiento es en sí, la prueba de ordenamiento de tarjetas desarrollada por el participante.

El evaluador puede gestionar estudios, es decir, crear un estudio nuevo, verificar sus ordenamientos, y además modificar o eliminar el estudio. El evaluador además podrá revisar los resultados de algún ordenamiento en particular o ver las estadísticas de una estudio.

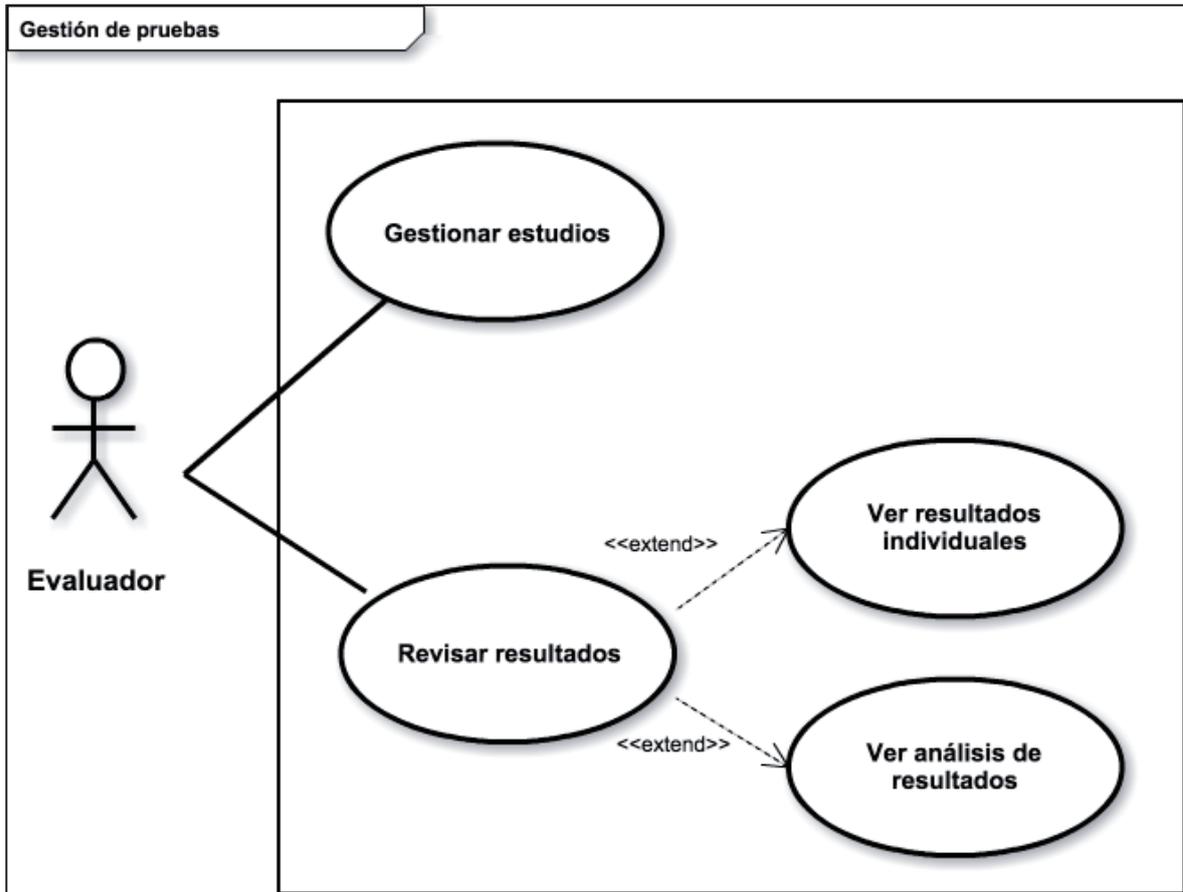


Figura 10-2 Diagrama de caso de uso Gestión de pruebas

Especificación formal: Gestión de pruebas

Caso De Uso	Gestión de Pruebas
Actor Principal	Evaluador
Participantes e intereses	El evaluador requiere ingresar al sitio para gestionar una prueba de CardSorting
Precondiciones	Evaluador requiere hacer uso de la herramienta e ingresar al sitio.
Postcondiciones	El evaluador logra gestionar una prueba satisfactoriamente
Escenario Principal	1. El evaluador debe ingresar al sitio 2. El evaluador debe ingresar sus credenciales de usuario y

	contraseña 3. El evaluador debe presionar login 4. El evaluador puede gestionar proyectos
Extensiones	2. El evaluador no se acuerde de su usuario y/o contraseña 4 El sistema crea el proyecto
Requisitos especiales	No aplica
Frecuencia de ocurrencia	Media

Tabla 10-1 Especificación formal: Gestión de pruebas

10.1.3 Diagrama de caso de uso revisar resultados individuales

En este diagrama se muestra las opciones que tiene el evaluador para revisar los resultados obtenidos en una prueba en particular, el evaluador puede ver cada ordenamiento en particular, viendo las tarjetas ordenadas por el participante y como estas están agrupadas. El evaluador además mientras observa el resultado de un ordenamiento tiene la opción de imprimir el ordenamiento.

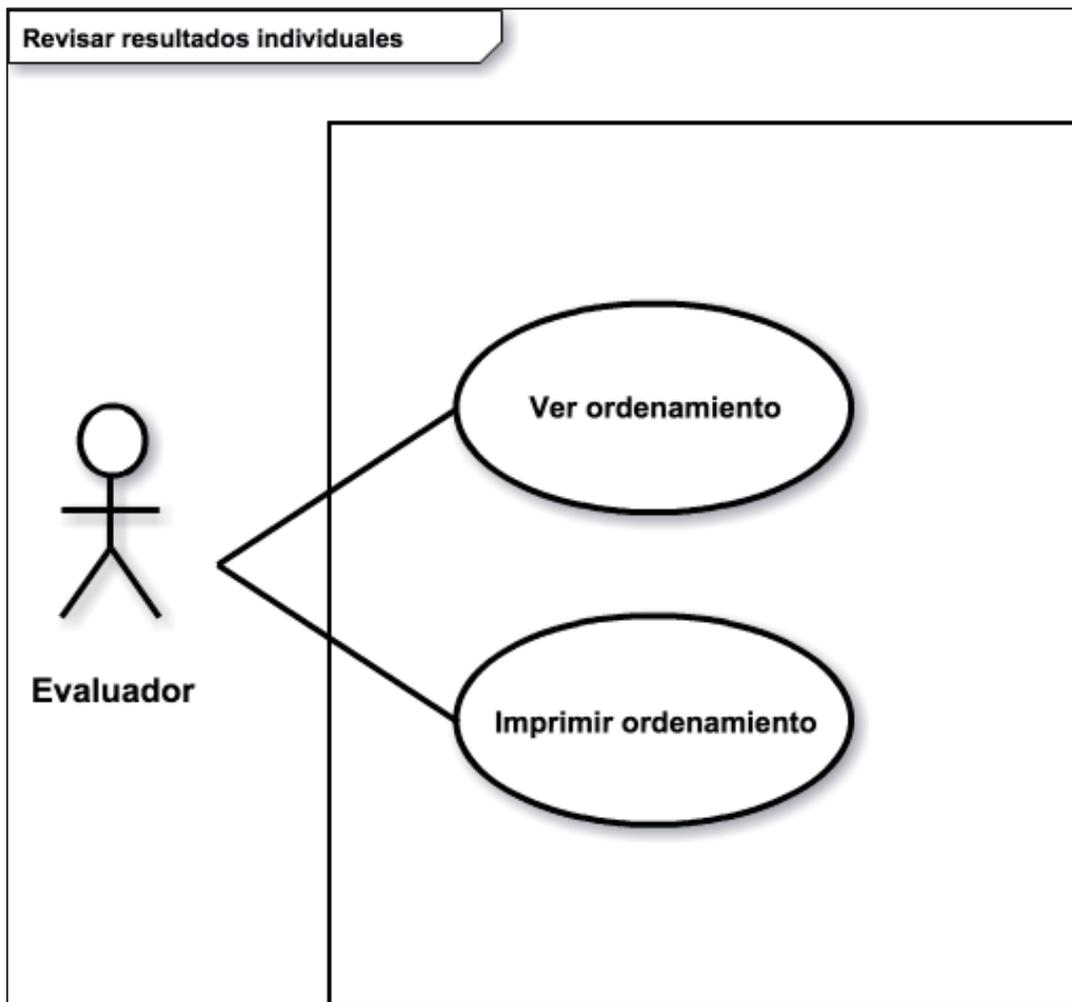


Figura 10-3 Diagrama de caso de uso Revisar resultados individuales

Especificación formal: Revisar resultados individuales

Caso De Uso	Revisar resultados individuales
Actor Principal	Evaluador
Participantes e intereses	Evaluador requiere ver los resultados individuales de los estudios
Precondiciones	Evaluador requiere ingresar a “Revisar Resultados individuales”
Postcondiciones	El sistema le muestra los resultados del estudio
Escenario Principal	1. El evaluador debe ingresar a “Revisar Resultados individuales” 2. El evaluador debe seleccionar el ordenamiento a revisar 3. El sistema muestra los resultados del ordenamiento
Extensiones	1. El evaluador debe ingresar en la opción correcta 2. El sistema acepta satisfactoriamente la petición 3. El sistema muestra correctamente la información
Requisitos especiales	No aplica
Frecuencia de ocurrencia	Alta

Tabla 10-2 Especificación formal: Revisar resultados individuales

10.1.4 Diagrama de caso de uso ver análisis de resultados

En este diagrama se muestra con mayor detalle la gestión de resultados por parte del evaluador, se puede distinguir entre gráficos, tablas y comentarios, los dos primeros son resultados cuantitativos, mientras el último es un dato cualitativo.

El evaluador puede elegir ver tablas ítem x ítem o tablas de herencia, o ver el gráfico de dendograma. Es significativo resaltar que estos análisis son de un estudio en particular.

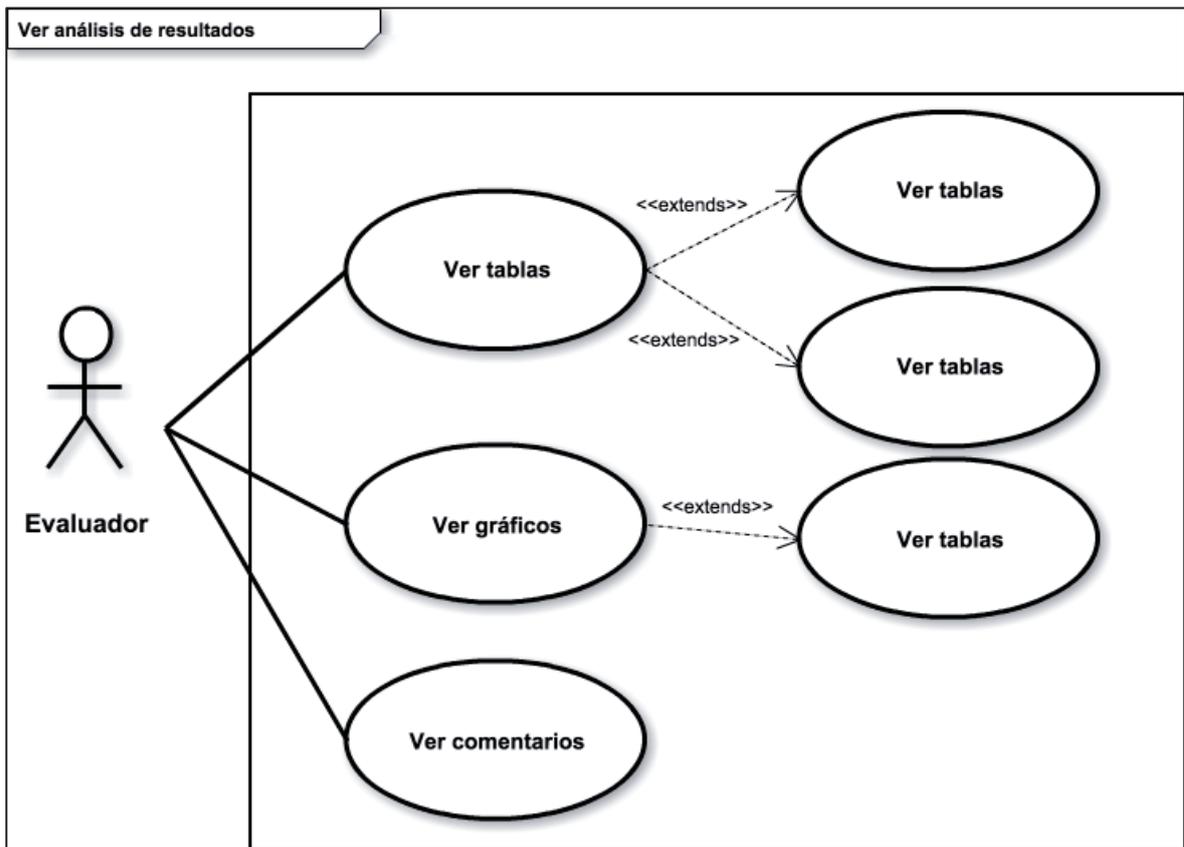


Figura 10-4 Diagrama de caso de uso Ver análisis de resultados

Especificación formal: Ver análisis de resultados

Caso De Uso	Ver análisis de resultados
Actor Principal	Evaluador
Participantes e intereses	Evaluador requiere ver los resultados de los estudios
Precondiciones	Evaluador requiere ingresar a “Revisar Resultados”
Postcondiciones	El sistema le muestra los resultados del estudio
Escenario Principal	<ol style="list-style-type: none"> 1. El evaluador debe ingresar a “Revisar Resultados” 2. El evaluador debe seleccionar el tipo de resultado a ver 3. El sistema muestra los resultados
Extensiones	<ol style="list-style-type: none"> 1. El evaluador debe ingresar en la opción correcta 2. El sistema acepta satisfactoriamente la petición 3. El sistema muestra correctamente la información
Requisitos especiales	No aplica
Frecuencia de ocurrencia	Alta

Tabla 10-3 Especificación formal: Ver análisis de resultados

10.1.5 Diagrama de caso de uso realizar ordenamiento

En este diagrama se detallan la interacción del participante de la prueba con el sistema. Acá el participante puede gestionar tarjetas y/o categorías, siendo una categoría un grupo de tarjetas. El participante puede crear nuevas tarjetas de considerarlo necesario o filtrar entre las tarjetas existentes mediante nombre para verificar si la tarjeta está creada, además de esto podrá modificar o eliminar dichas tarjetas, no así las elaboradas por el evaluador. También el participante puede crear o eliminar una categoría.

La función principal del participante radica en agrupar conceptos (tarjetas o categorías), es decir, puede agrupar tarjetas o un grupo de ellas como estime conveniente, modificando o eliminando las asociaciones existentes. También el participante tendrá la opción de guardar su progreso, para finalizar la prueba o seguir con ella posteriormente.

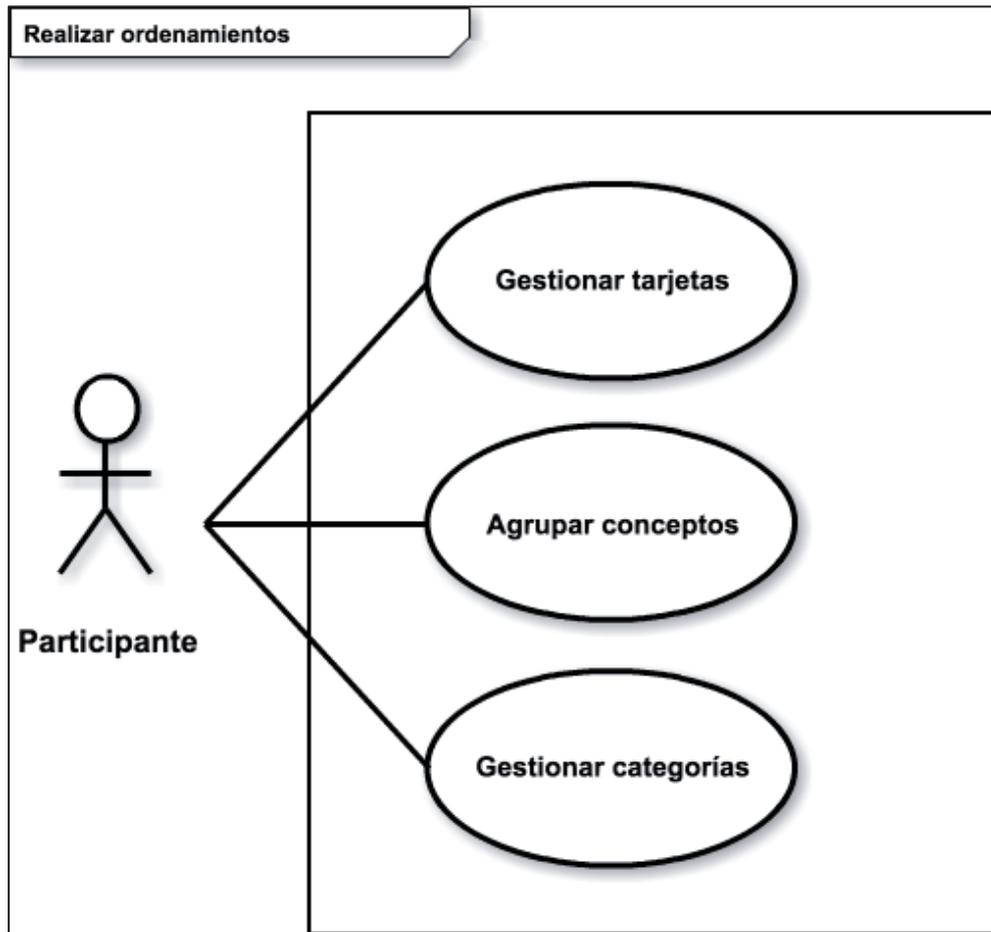


Figura 10-5 Diagrama de caso de uso Realizar ordenamientos

Especificación formal: Realizar ordenamientos

Caso De Uso	Realizar ordenamiento
Actor Principal	Participante
Participantes e intereses	El participante requiere realizar un ordenamiento de tarjetas
Precondiciones	El participante tenga conocimiento de cómo realizar el ordenamiento

Postcondiciones	El sistema guarda el ordenamiento realizado correctamente
Escenario Principal	1. El participante debe ingresar al ordenamiento 2. El participante puede gestionar tarjetas, gestionar categorías y agregar conceptos. 3. El participante puede terminar el ordenamiento
Extensiones	2. El participante requiere de conexión a internet para realizar la prueba. 3. El sistema envía la prueba satisfactoriamente
Requisitos especiales	No aplica
Frecuencia de ocurrencia	Alta

Tabla 10-4 Especificación formal: Realizar ordenamiento

10.1.6 Diagrama de caso de uso gestionar evaluadores

En este diagrama se detalla las acciones que puede realizar el administrador para gestionar a los evaluadores, pudiendo crear una cuenta de evaluador nueva o eliminar la cuenta de un evaluador. Además puede modificar la cuenta de un evaluador, cambiando los privilegios de dicha cuenta desde evaluador a administrador, o viceversa.

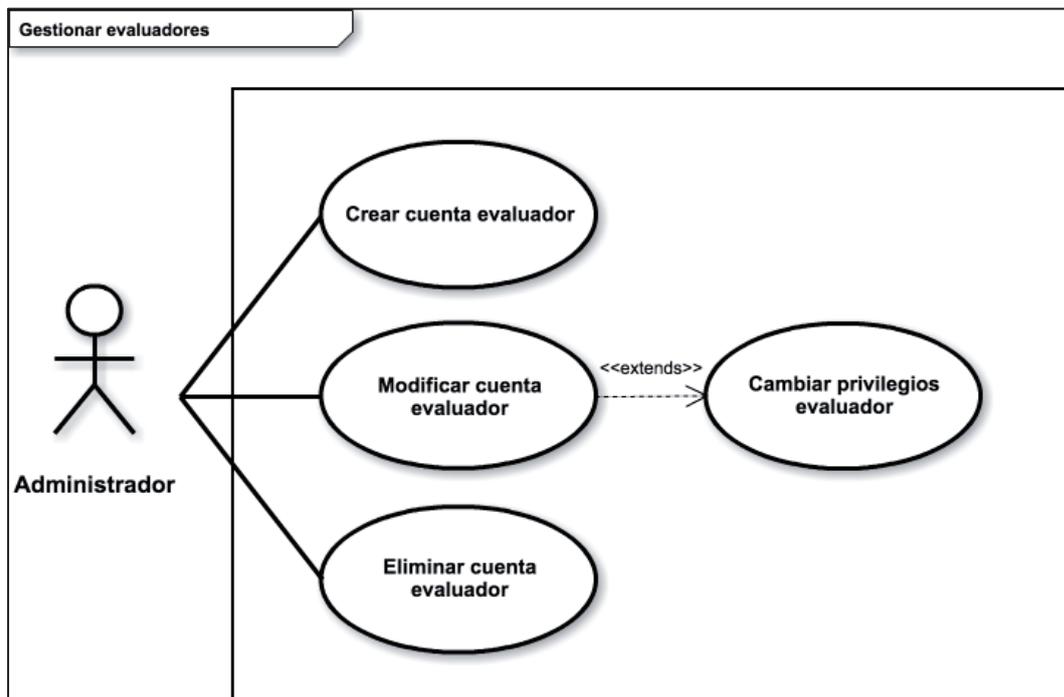


Figura 10-6 Diagrama de caso de uso Gestionar evaluadores

Especificación formal: Gestionar evaluadores

Caso De Uso	Gestionar evaluadores
Actor Principal	Administrador

Participantes e intereses	El administrador requiere gestionar la cuenta de un evaluador
Precondiciones	El administrador debe ingresar a gestión de evaluadores
Postcondiciones	El sistema guarda los cambios a la cuenta exitosamente
Escenario Principal	1. El administrador debe ingresar a gestión de evaluadores 2. El administrador puede crear una cuenta nueva, eliminar una cuenta o modificarla. 3. El administrador guarda los cambios realizados
Extensiones	2. El administrador requiere de conexión a internet para realizar los cambios. 3. El sistema almacena los cambios satisfactoriamente
Requisitos especiales	No aplica
Frecuencia de ocurrencia	Media

Tabla 10-5 Especificación formal: Gestionar evaluadores

10.2 Diagrama de actividad

10.2.1 Diagrama de actividad: Revisar análisis

Uno de los objetivos del presente proyecto es la obtención de datos para el análisis de los resultados de las pruebas, el siguiente diagrama es una representación de las actividades necesarias para revisar dichos datos.

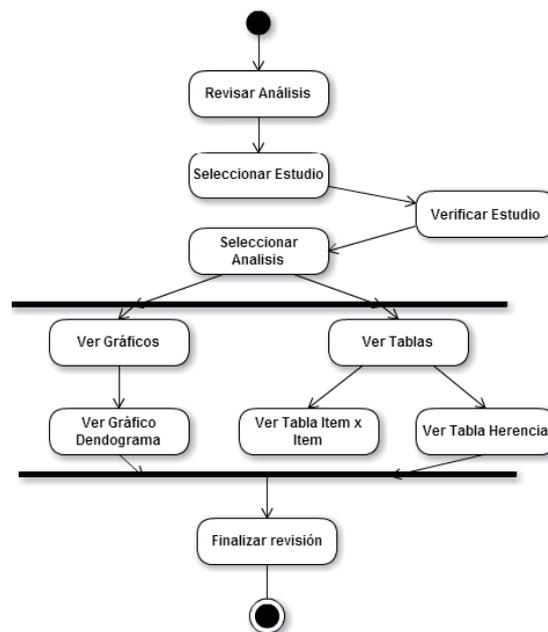


Figura 10-7 Diagrama de actividad Revisar análisis

10.2.2 Diagrama de actividad: Gestión de pruebas

En el presente diagrama se detallan las actividades para gestionar las pruebas

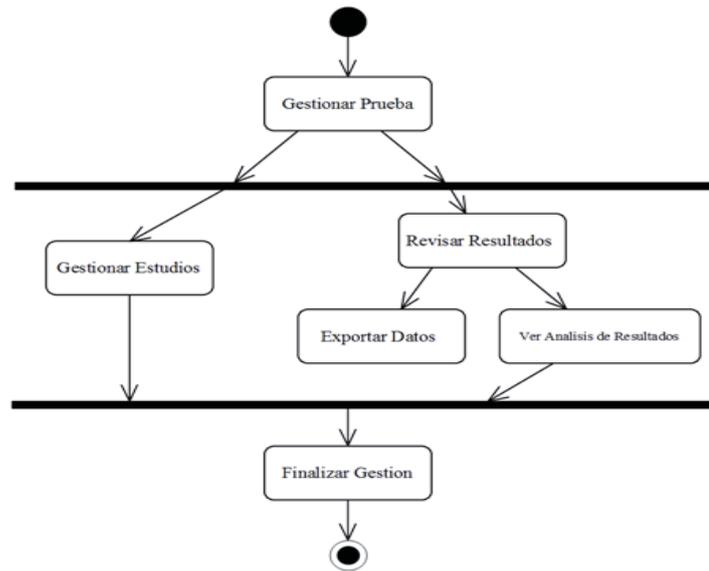


Figura 10-8 Diagrama de actividad: Gestionar prueba

10.2.3 Diagrama de actividad: Realizar ordenamiento

En el presente diagrama se detallan las actividades que se pueden realizar en un ordenamiento de tarjetas, gestionar tarjetas o categorías y agrupar conceptos, además de guardar los cambios y finalizar la prueba.

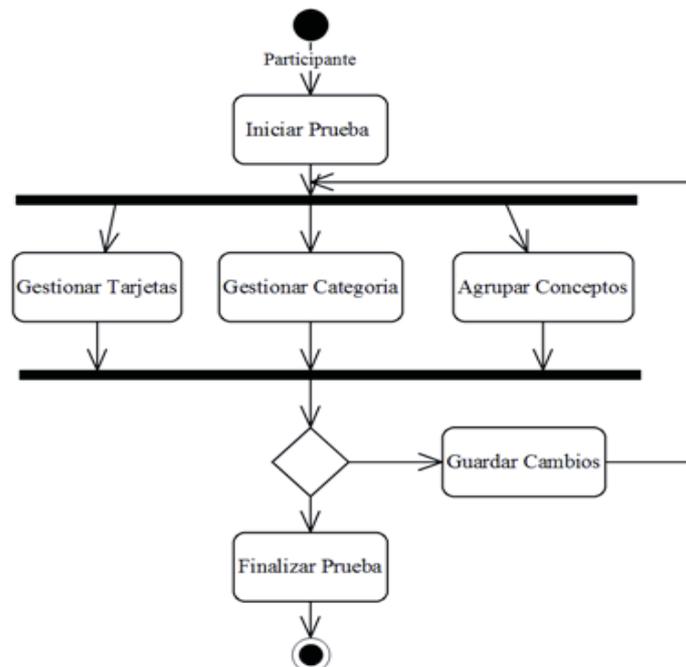


Figura 10-9 Diagrama de actividad: Realizar prueba

10.3 Diagrama de secuencia

10.3.1 Diagrama de secuencia Revisar análisis

Uno de los objetivos del presente proyecto es la obtención de datos para el análisis de los resultados de las pruebas, el siguiente diagrama es una representación de los pasos a realizar para obtener estos resultados.

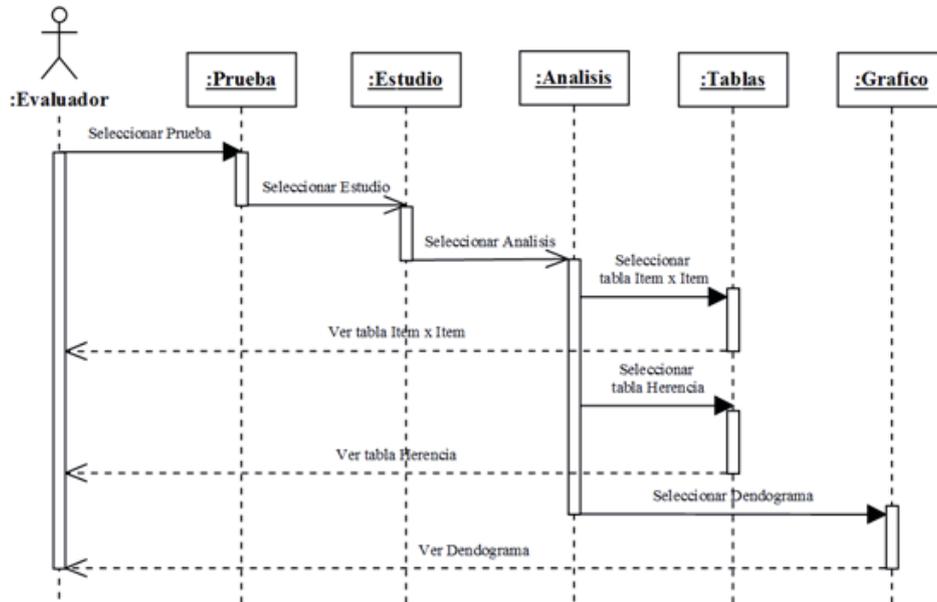


Figura 10-10 Diagrama de secuencia: Revisar análisis

10.3.2 Diagrama de secuencia Gestionar pruebas

En el siguiente diagrama de secuencia se detallan los pasos a realizar por el evaluador para gestionar las pruebas.

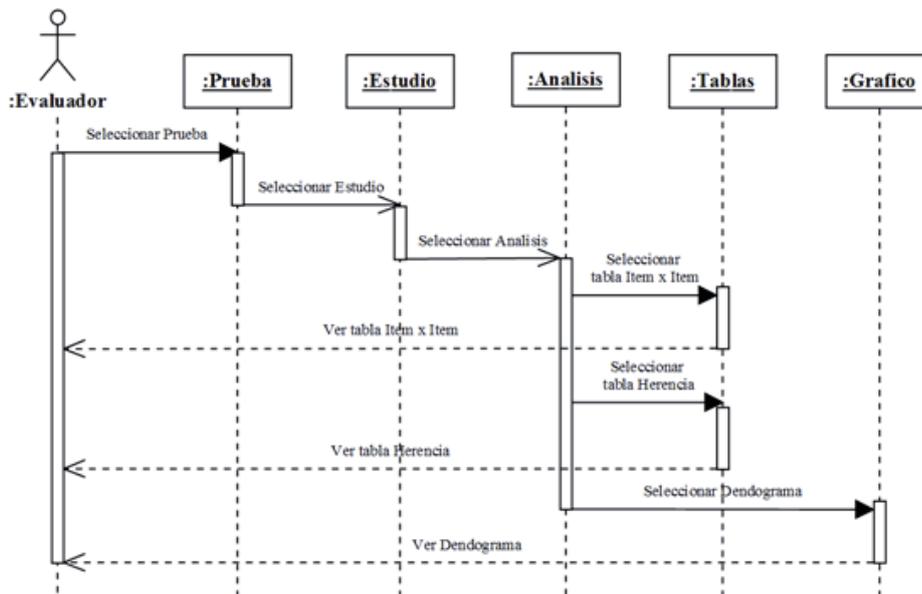


Figura 10-11 Diagrama de secuencia: Gestionar pruebas

10.3.3 Diagrama de Secuencia Realizar ordenamiento

En el siguiente diagrama de secuencia se detallan los pasos a realizar por el participante para realizar un ordenamiento de tarjetas.

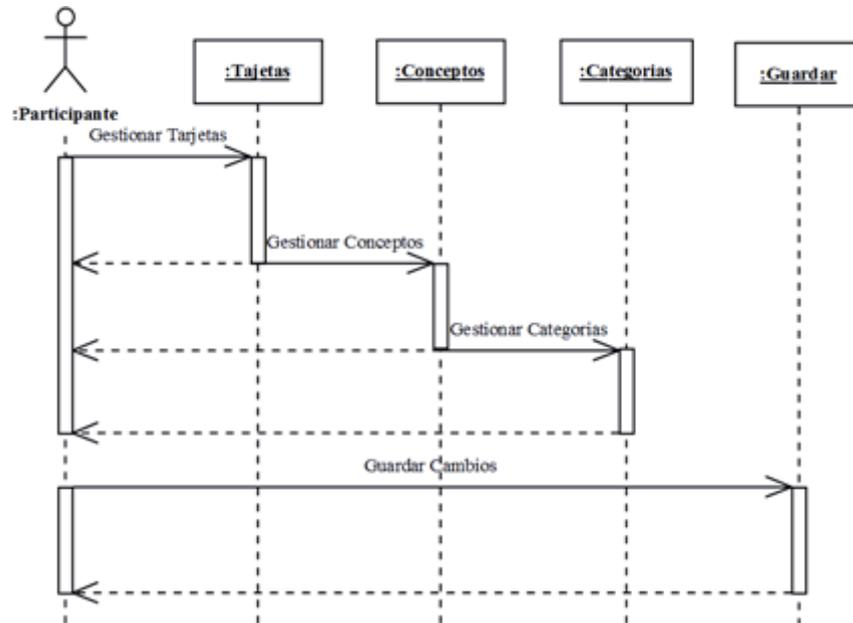


Figura 10-12 Diagrama de secuencia: Realizar ordenamiento

10.4 Diagrama de clases

El diagrama de clases es el diagrama principal para el análisis y diseño del sistema, es un tipo de diagrama estático que describe la estructura del sistema mostrando sus clases, atributos y las relaciones entre ellos.

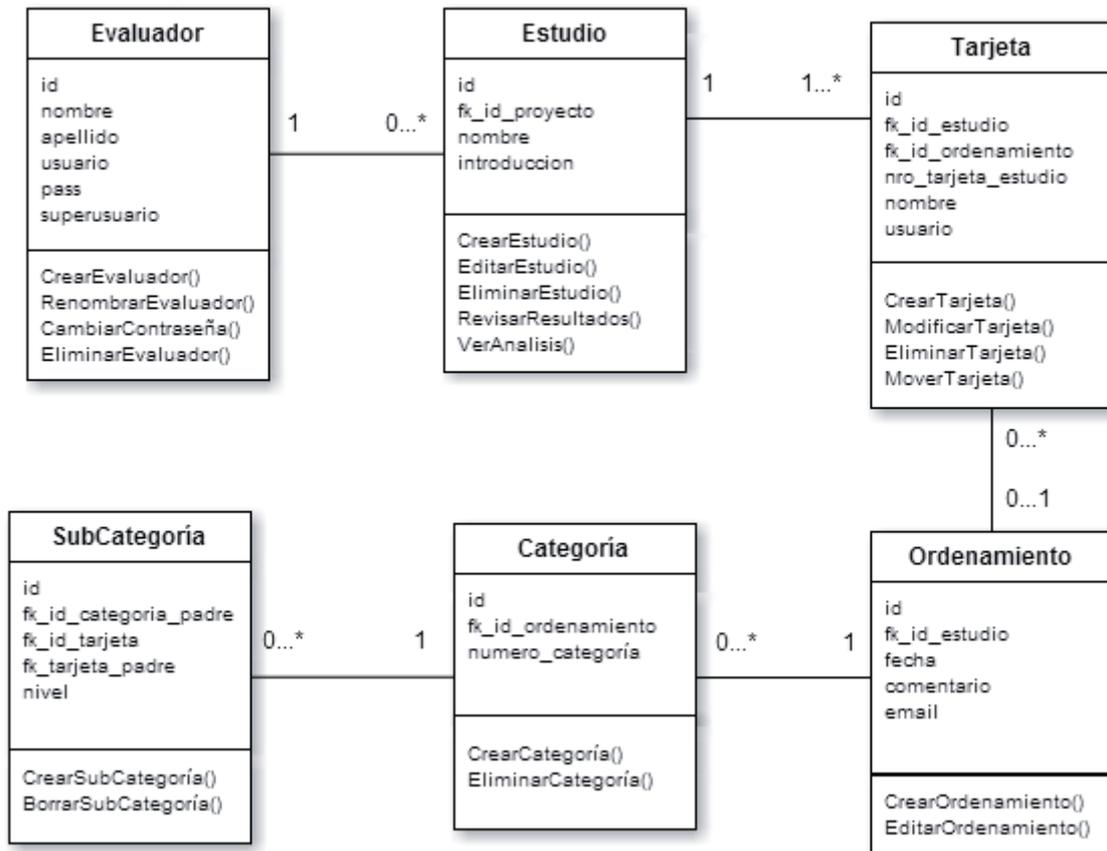


Figura 10-13 Diagrama de clases

11 Desarrollo del prototipo de sistema

11.1 Primer prototipo

El objetivo del primer prototipo tiene como objetivo considerar la interfaz visual con la que se encontrará el usuario, donde se comprueban los requerimientos del sistema para posteriormente agregar las respectivas funcionales.



Figura 11-1 Prototipo gestionar proyectos

La página de inicio del sitio consiste en un mantenedor de proyectos dando la opción de editar, eliminar o agregar proyectos.



Figura 11-2 Prototipo tabla de coincidencias

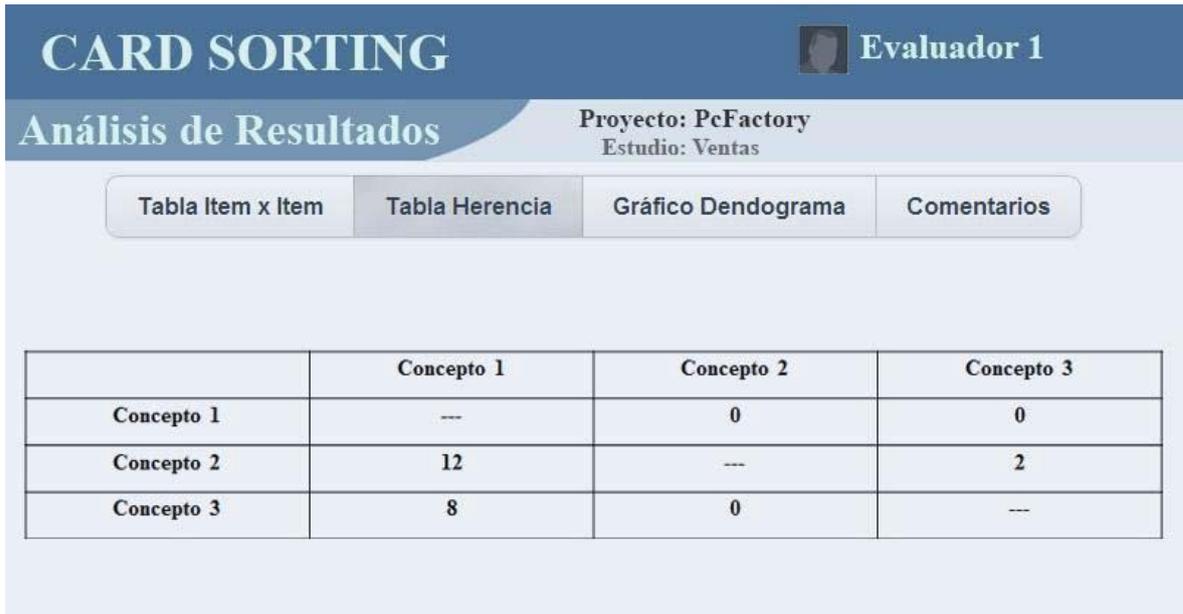


Figura 11-3 Prototipo tabla de herencia

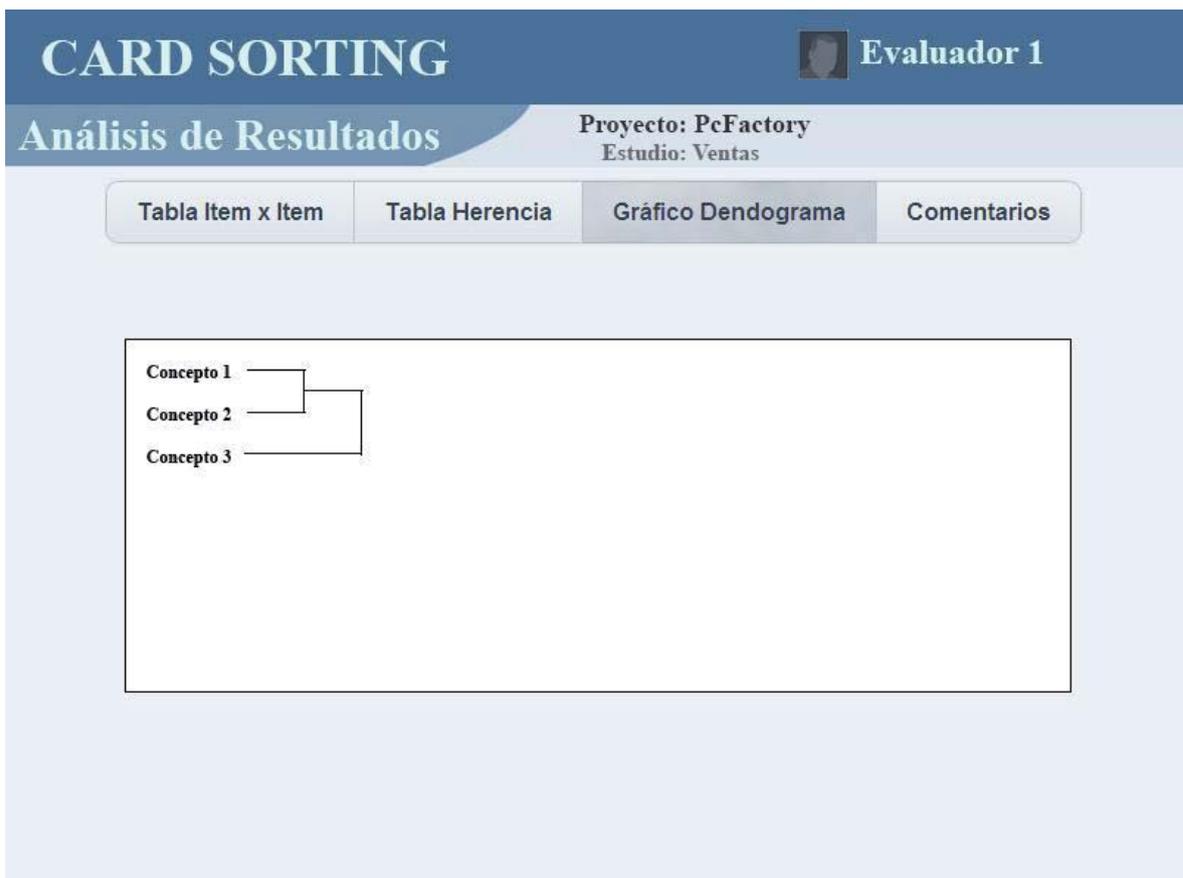


Figura 11-4 Prototipo gráfico Dendograma



Figura 11-5 Prototipo comentarios

Correspondiente al análisis de resultados de los ordenamientos se crean tablas y gráficos para los resultados cuantitativos de datos y se listan los comentarios de los usuarios que ordenan las tarjetas para obtener resultados cualitativos. El presente prototipo es usado para calificar la aceptación de los posibles usuarios a la herramienta.

11.2 Segundo prototipo

Parte fundamental de la herramienta es el análisis de los resultados entregados por los participantes, para lo cual se dispone de un apartado en el sitio para visualizarlos.



Figura 11-6 Barra análisis de resultados

Consiste en una barra horizontal desde donde se puede ingresar a los distintos tipos de análisis, ya sea tabla de coincidencias, tabla de herencia, dendograma o los comentarios de los participantes del estudio en particular.

Tabla de coincidencias

Numero de veces que una tarjeta ha sido agrupada junto a otra tarjeta

	chw	mega	facebook	twitter	hotmail	identi	jaidefinichon
chw	-	1	0	0	0	1	1
mega	1	-	0	0	1	2	0
facebook	0	0	-	2	1	0	0
twitter	0	0	2	-	1	0	0
hotmail	0	1	1	1	-	1	0
identi	1	2	0	0	1	-	0
jaidefinichon	1	0	0	0	0	0	-

Figura 11-7 Tabla de coincidencias

Tabla que muestra de forma gráfica el número de veces que dos conceptos fueron ordenados en el mismo grupo. Para esto es necesario revisar en todos los ordenamientos y tarjetas asociados al estudio, para finalmente contabilizar las tarjetas agrupadas en las categorías del ordenamiento.

Porcentaje de agrupación

Excelente (100% - 90%)	Bueno (89% - 75%)	Aceptable (74% - 50%)
mega - identi (100%) facebook - twitter (100%)		chw - mega (50%) chw - identi (50%) chw - jaidefinichon (50%) mega - hotmail (50%) facebook - hotmail (50%) twitter - hotmail (50%) hotmail - identi (50%)

Figura 11-8 Porcentaje de agrupación

También se incluye una pequeña tabla que resume los resultados de la tabla anterior, la cual filtra los resultados según su porcentaje de acuerdo al total de ordenamientos realizados.

DENDOGRAMA

Relacion grafica del nivel de agrupamiento de las tarjetas

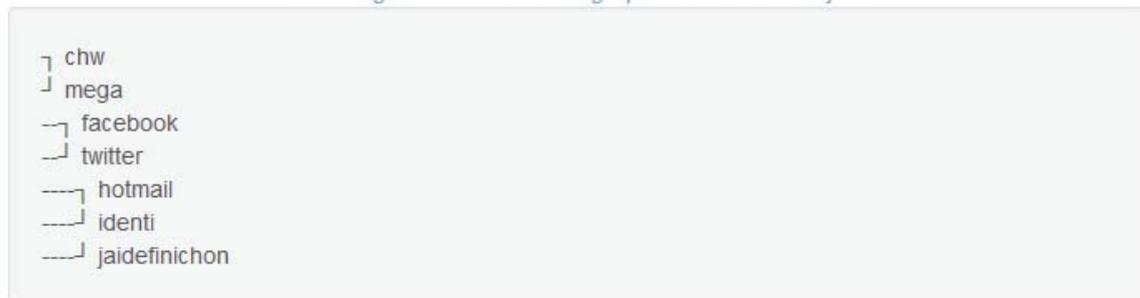


Figura 11-9 Dendograma

Gráfico que muestra de forma visual la relación entre las tarjetas, mostrando en el mismo nivel las tarjetas con el mismo nivel de agrupamiento, para luego descender en nivel. De esta manera se facilita al evaluador la división en grupos de los conceptos.

Para lograr realizar esta tarea es necesario realizar los mismos cálculos realizados en la tabla de coincidencias y en base a estos realizar un listado ordenado de mayor a menor que contiene los datos (tarjeta1 - tarjeta2 - número de coincidencias), de esta manera luego se puede recorrer este listado y agregar al gráfico los pares de tarjetas y su nivel de agrupamiento.

11.3 Tercer prototipo

El tercer prototipo tiene como objetivo integrar los cambios realizados a los prototipos anteriores, corrigiendo los errores y discordancias presentados al realizar la integración.

The screenshot shows the 'CARDSORTING' application interface. At the top, there is a blue header with the text 'CARDSORTING' on the left and 'Gestionar Evaluadores', 'Ayuda', and 'root toor' on the right. Below the header, the main content area is titled 'Listado de Estudios'. It contains a table with the following data:

Id	Nombre estudio	Accion
1	Estudio Alimentos	Ir a la prueba Editar estudio Pruebas completadas Estadísticas Eliminar
2	Proyecto Sitios de Internet	Ir a la prueba Editar estudio Pruebas completadas Estadísticas Eliminar
3	test	Ir a la prueba Editar estudio Pruebas completadas Estadísticas Eliminar

Below the table, there is a blue button labeled 'Nuevo Estudio'.

Figura 11-10 Gestión de estudios

Uno de los grandes cambios realizados al prototipo propuesto es la eliminación de la tabla de proyectos, dejando solamente la tabla de estudios, dado que no representaba una verdadera funcionalidad al sistema. Es por esto que la página principal del evaluador es ahora la gestión de estudios. Donde se puede ver añadida la funcionalidad para ingresar a las estadísticas del estudio, este apartado se refiere al análisis de resultados de dicho estudio.

12 Validación del prototipo

Para comprobar el nivel de aceptación de los usuarios hacia la herramienta se procedió a realizar una validación del prototipo, esta consiste en buscar participantes que cumplan con distintos perfil para validar la herramienta y que interactúen con ella, para finalmente pedir sus observaciones y una evaluación sobre el prototipo.

12.1 Participantes

Al realizar la prueba se buscaron dos grupos de participantes distintos, un grupo que tuvieran conocimiento sobre ordenamiento de tarjetas y sus propósitos, para cumplir con el perfil de usuario que usarían la herramienta.

Del mismo modo se buscaron participantes que no necesariamente tuvieran conocimientos sobre la prueba de ordenamiento de tarjetas, esto en específico para validar la sección de realizar un ordenamiento, puesto que estos usuarios no necesitan tener conocimientos previos para poder realizar la prueba.

El número de integrantes que realizaron la prueba de validación del prototipo fueron cinco personas de las cuales tres tenían conocimiento previo sobre el ordenamiento de tarjetas, entre ellos un ingeniero en informática, un alumno de técnico universitario en informática y un alumno de ingeniería ejecución en informática la escuela de informática de la universidad y las otras 2 personas que participaron no tenían experiencia anterior con el ordenamiento de tarjetas, uno de ellos familiarizado con tecnologías táctiles, mientras que el otro participante no.

12.2 Observaciones recibidas

A continuación se detallan las observaciones recibidas luego de la prueba de validación del prototipo.

- El encabezado del sitio da una clara referencia sobre el propósito del mismo.
- El sitio no da referencia hacia qué tipo de usuarios está enfocado el sitio.
- La navegación se hace de manera intuitiva, sin embargo algunos links o botones no dan una clara referencia de lo que hacen.
- El gráfico de Dendograma no muestra claramente cuál es el agrupamiento con mejor nivel de agrupamiento, los agrupados al lado izquierdo o derecho.
- No se explica de qué forma se pueden borrar las tarjetas creadas de un ordenamiento.
- En el ordenamiento no se puede volver una tarjeta al grupo de tarjetas no usadas.
- En el ordenamiento una vez creado un nuevo grupo, no se puede borrar.
- En el ordenamiento al crear una tarjeta nueva, no se puede borrar.

12.3 Evaluación

Al terminar la prueba se le pidió a los integrantes que evaluaran su experiencia con la herramienta con una puntuación del 1 al 10, para medir de forma objetiva el nivel de aceptación del prototipo.

Se determinó anteriormente a la realización de la prueba una escala de aceptación de 3 niveles sobre el promedio de dichas notas.

- **menor o igual a 6:** Se califica como mala.
- **mayor a 6 y menor o igual a 8:** Se califica como aceptable.
- **mayor a 8:** Se califica como bien.

El listado de puntuaciones obtenidas fue: 8, 9, 9, 8 y 7, y el promedio resultante fue: 8,2. Por lo que se puede calificar la puntuación obtenida como aceptable.

12.4 Conclusión

Luego de realizar la prueba de validación del prototipo se puede destacar que ningún usuario tuvo mayores problemas para terminar la tarea pedida, lo cual indica que no se presentaron problemas de navegabilidad del sitio y que los conceptos entregados fueron bien entendidos por los integrantes de la prueba.

Por otro lado los puntos negativos observados fueron la incapacidad de enmendar errores al realizar alguna acción, en este apartado los mayores problemas presentados fueron al intentar dejar una tarjeta agrupada nuevamente al grupo de tarjetas no seleccionadas y al intentar borrar una tarjeta creada por el usuario en el ordenamiento esta no puede ser modificada.

Al finalizar la prueba y revisar las puntuaciones y el promedio obtenido se puede decir que fue evaluada favorablemente, por lo que si necesitaran gestionar una prueba de ordenamiento de tarjetas recurrirían a la herramienta para hacerlo.

13 Conclusiones y trabajo futuro

A partir de lo expuesto en este informe, se puede concluir que el caso del prototipo desarrollado entrega una visión general sobre la disciplina de interacción persona ordenador, que ayuda a comprender porque cada día es más importante en los proyectos software e incluso en proyectos no relacionados con software. Esta disciplina aplicada al área informática ayuda a mejorar ciertos aspectos en los sistemas informáticos.

Se analizaron dos maneras de realizar esta prueba, clasificadas según el modo de efectuar la prueba. De forma manual y bajo aplicaciones software, cada una con sus ventajas y desventajas. La manera manual se refiere a la realización de la prueba de forma presencial ordenando tarjetas de papel en un tablero o mesa, mientras bajo la aplicación de algún software tanto las tarjetas como el tablero son virtualizados.

Las diferencias de aplicar un método de ordenamiento de tarjetas u otro radican en dos puntos importantes, la facilidad de realizar la prueba para el participante y la facilidad de obtener resultados cualitativos y/o cuantitativos para el evaluador. Mientras de forma manual es más sencillo ordenar los conceptos para el participante, el obtener y analizar los resultados es más tedioso para el evaluador, sin embargo, bajo la aplicación de algún software el ordenar conceptos se hace más tedioso para el usuario, pero se hace más sencillo para el evaluador la obtención e interpretación de resultados.

Al analizar esta situación se decidió desarrollar un prototipo de herramienta de apoyo para actividades de ordenamiento de tarjetas que mezcle las ventajas de ambos métodos, el manual y el virtual, que simplifique el análisis de los resultados para el evaluador mediante la automatización del análisis estadístico de resultados y, a la vez, mejore la interacción del participante con el sistema implementando alguna tecnología más intuitiva, como es el uso de tecnología táctil.

Luego de comparar algunas alternativas se determinó que el desarrollo de la herramienta se regiría bajo una metodología de desarrollo evolutivo, donde la entrega de prototipos permite una constante comprobación del cumplimiento de los requerimientos del proyecto. Para complementar el paradigma de proceso de software de desarrollo evolutivo se decidió trabajar bajo una metodología orientada a objetos, metodología de naturaleza modular y reutilizable, donde el modelado del sistema se basa en examinar el dominio del problema como un conjunto de objetos que interactúan entre sí, facilitando la creación y la integración de prototipos.

Como arquitectura para el desarrollo de la herramienta se resolvió utilizar una arquitectura de tres capas, implementando la solución en distintos niveles cada uno encargado de una simple tarea, esto permite trabajar en un nivel abstrayéndose del resto de los niveles.

Se logró desarrollar un prototipo de herramienta de apoyo para actividades de Card Sorting con tres módulos principales, gestión de datos, análisis de resultados y realización de una prueba. En la gestión de datos se desarrollaron mantenedores para administrar los estudios filtrando los estudios pertenecientes a un evaluador. En el análisis de resultados se desarrollaron algoritmos capaces de interpretar los datos entregados al sistema para elaborar tablas y gráficos que sirven de ayuda al evaluador para comprender los resultados

obtenidos. Y el módulo se basó en la realización de un espacio de trabajo virtual donde el usuario puede ordenar las tarjetas virtuales, mediante la ayuda de la tecnología de pantallas táctiles.

La realización del prototipo obligó al grupo de trabajo a refinar sus conocimientos sobre la programación, para la persistencia de datos se utilizó el lenguaje MySQL con el cual se desarrolló la base de datos, para la gestión de los datos se utilizaron los lenguajes PHP y JavaScript, y para presentar los datos se utilizó el lenguaje de programación HTML. Además el grupo de trabajo debió estudiar a fondo temas relacionados con la técnica de ordenamiento de tarjetas, la usabilidad en el área de las tecnologías y la interacción persona computador.

Para trabajos futuros se pueden tratar temas que acerquen aún más la forma de interactuar de las personas con la tecnología, como podría ser la realidad aumentada, logrando con esta tecnología mezclar la forma de realizar los ordenamientos de forma física y virtual, donde el participante pueda realizar la prueba con tarjetas reales y en un espacio de trabajo real y gracias a la realidad aumentada obtener información adicional que ayude a desarrollar la prueba.

Otro punto a considerar como trabajo futuro es la realización de herramientas de Card Sorting en aplicaciones móviles, la inclusión de tecnología táctil en el presente proyecto es una clara muestra de ello. La utilización de dispositivos móviles ayudaría a emular un espacio de trabajo para ordenamiento de tarjetas en cualquier lugar.

14 Referencias utilizadas

- [1] Nielsen J.: Usabilidad. Diseño de sitios web, editorial Prentice Hall, 2000.
- [2] Nielsen J., Loranger H.: Prioritizing Web Usability, New Riders Press, 2006.
- [3] Mitchell P.: A Step-by-Step Guide to Usability Testing, iUniverse, 2007.
- [4] Montero H., Fernández M., Rodríguez M.: Arquitectura de la información en los entornos virtuales de aprendizaje. Aplicación de la técnica de CardSorting y análisis cuantitativo de los resultados”, 2004.
- [5] Manchón E.: Diseña como piensan los usuarios. Técnica de agrupación de tarjetas o CardSorting, 2004.
- [6] ACM SIGCHI. Disponible vía web en <http://www.sigchi.org/> . Revisada por última vez el 30 de Julio de 2015.
- [7] Ingeniería del software, Séptima Edición, IAN SOMMERVILLE, 2005. Disponible vía web en http://www.inf.ucv.cl/~bcrawford/AULA_ICI441/Ingenieria%20del%20Software%207ma.%20Ed.%20-%20Ian%20Sommerville.pdf. Revisada por última vez el 30 de Julio de 2015.
- [8] JQuery Mobile. Disponible vía web en <http://sergioglez.webcindario.com/cargarArticulo.php?id=47>. Revisada por última vez el 30 de Julio de 2015.
- [9] IntroducciónHTML5. Disponible vía web en <http://www.w3.org/html/wg/drafts/html/master/introduction.html>. Revisada por última vez el 30 de Julio de 2015.
- [10] Que es PHP. Disponible vía web en <http://php.net/manual/es/intro-what-is.php>. Revisada por última vez el 30 de Julio de 2015.
- [11] Introducción a HTML y CSS. Disponible vía web en <http://www.w3.org/standards/webdesign/htmlcss>. Revisada por última vez el 30 de Julio de 2015.
- [12] Que es MySQL. Disponible vía web en <http://dev.mysql.com/doc/refman/5.0/en/what-is-mysql.html>). Revisada por última vez el 30 de Julio de 2015.
- [13] Que es Xampp. Disponible vía web en <http://www.apachefriends.org/es/xampp.html>). Revisada por última vez el 30 de Julio de 2015.
- [14] Que es UML. Disponible vía web en http://www.omg.org/gettingstarted/what_is_uml.htm<http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/doc-modelado-sistemas-uml.pdf>). Revisada por última vez el 30 de Julio de 2015.
- [15] Programación orientada a objetos. Disponible vía web en <http://www.desarrolloweb.com/articulos/499.php>. Revisada por última vez el 30 de Julio de 2015.

- [16] Arquitectura de tres niveles. Disponible vía web en http://www.ecured.cu/index.php/Arquitectura_de_tres_niveles. Revisada por última vez el 30 de Julio de 2015.
- [17] Análisis orientado a objetos, José Olgún Espinoza. Disponible vía web en <http://www.sisman.utm.edu.ec/libros/FACULTAD%20DE%20CIENCIAS%20INFORM%C3%81TICAS/CARRERA%20DE%20INGENIER%C3%8DA%20DE%20SISTEMAS%20INFORMATICOS/07/INGENIERIA%20DEL%20SOFTWARE%20I/MODELADO%20DEL%20ANALISIS/IngSoft%201-4.pdf>. Revisada por última vez el 30 de Julio de 2015.
- [18] ISO, International Organization for Standardization, "ISO 9126-1:2001, Software engineering - Product quality, Part 1: Quality model", 2001.
- [19] ISO, International Organization for Standardization, "ISO/IEC 9241-210:2010, Ergonomics of human-system interaction, Part 210, Human-centred design for interactive systems". Disponible vía web en http://www.iso.org/iso/catalogue_detail.htm?csnumber=52075. Revisada por última vez el 30 de Julio de 2015.
- [20] ISO/IEC 25000. Disponible vía web en <http://www.iso25000.com/>. Revisada por última vez el 30 de Julio de 2015.
- [21] How Many Users Are Enough for a Card-Sorting Study? - Tom Tullis y Larry Wood". Disponible vía web en <http://home.comcast.net/~tomtullis/publications/UPA2004CardSorting.pdf>. Revisada por última vez el 30 de Julio de 2015.
- [22] Ejemplo Análisis de clúster – M. Vargas Jiménez". Disponible vía web en <http://www.ugr.es/~mvargas/3.DosEjesanalisisclusteryCCAA.pdf>. Revisada por última vez el 30 de Julio de 2015.
- [23] Arquitectura de la información y card sorting, Victor Manuel Peñeñory. ". Disponible vía web en <http://es.scribd.com/doc/209200803/Card-Sorting>. Revisada por última vez el 30 de Julio de 2015.