

**UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL
CUSCO**

**FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA,
INFORMÁTICA Y MECÁNICA**

**ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA Y
DE SISTEMAS**



“PROYECTO DE INVESTIGACIÓN DE E-COMMERCE. UWUNSAAC”

DOCENTE: QUINTANILLA PORTUGAL ROXANA LISETTE

ALUMNOS:

- | | |
|------------------------------------|--------|
| • Araujo Lechuga Luis Augusto | 170748 |
| • Carcausto Mamani Elizon Frank | 170427 |
| • Deza Condori Rosmel Uriel | 171058 |
| • Flores Aquino Luis Enrique | 171564 |
| • Monzon Montalvo Alexander Junior | 171914 |
| • Salcedo Hurtado Jorge Andre | 171572 |
| • Velarde Flores Manuel Humberto | 171918 |
| • Yarahuaman Rojas Milagros | 171071 |

CUSCO – PERÚ
2020-II

1. Introducción

UwUnsaac es un sitio web que nos sirve como un catálogo de productos que están en venta, donde los docentes, estudiantes y personal de la UNSAAC puedan subir sus productos y publicitar su producto de tal forma que todas las personas que tengan acceso a la página pueden ver los productos en la página. A Través de la página UWUnsaac los interesados con el producto podrán visualizar todos los productos de nuestro catálogo, utilizar filtros para ver productos que están al alcance de su bolsillo y por ultimo ,comunicarse con el propietario del producto con el cual esta interesado para así poder completar su compra.

Puesto que cuando empezamos a hacer la página se tenía que manejar los conceptos de Reutilización de software y la Reingeniería de Software. Existían problemas como encontrar un código que sea adaptable al proyecto que nos teníamos planteados, como consecuencia de ello se tenía que entender el código para así poder hacer las modificaciones necesarias para cumplir con nuestros objetivos. Otro de los problemas que surgieron fueron los requisitos tanto funcionales como no funcionales. Por un lado los requisitos funcionales no se tenían claros, por otro lado teníamos requisitos no funcionales muy abstractos.

A lo largo del desarrollo e implementación de la página UWUnsaac se utilizó la metodología RUP el cual nos permite seguir mejorando la página en cada iteración. Para escoger el código se analizaron los posibles candidatos. Una vez seleccionado el mejor de los candidatos se analiza más a fondo, se hizo un diagrama de componentes y se empezó a modificar según a las necesidades y problemas que nos íbamos encontrando. Para los requisitos funcionales se comenzó a revisar cuales serian los problemas reales con los cuales nuestros usuarios se enfrentan. Por otro lado para tratar los requisitos no funcionales se fueron creando modelos, los cuales nos ayudan a descomprimir la abstracción que presentaba cada uno y convertirlo ya sea en tareas o metas.

2. Detallamiento del Desarrollo

Proyecto seleccionado de Github

El proyecto tomado como base, para el desarrollo del proyecto UwUnsaac, es el software Zack Market, dicha plataforma es de código abierto lo cual favorece bastante la investigación con respecto a la Reutilización de Software y Reingeniería de Software. Para la obtención del código de la aplicación se utilizó los repositorios de Github, donde el usuario “leveloper” público y por ende facilitó la adquisición del código.

Zack Market es una aplicación de tienda en línea orientada al comercio electrónico de prendas de vestir; Zack Market propone herramientas informáticas necesarias para realizar compras o adquirir productos posteados en su plataforma.

Zack Market está desarrollado en los lenguajes de programación:

LENGUAJE	PORCENTAJE
● JavaScript	78.5%
● CSS	19.8%
● HTML	1.7%

Front - end

La aplicación hace uso de la librería de React JS para la implementación de interfaces gráficas. La estructura interna o el código sigue un paradigma orientado a componentes por lo que la React es una herramienta idónea para su desarrollo.

URL : <https://github.com/levelopers/Ecommerce-Reactjs>.

Back - end

Para la gestión de datos Zack Market hace uso del entorno de JavaScript Node JS, el cual gracias a su arquitectura orientada a eventos facilita la tarea de ejecutar un servidor el cual gestione los datos de manera asíncrona y teniendo la información con una alta disponibilidad.

URL : <https://github.com/levelopers/Ecommerce-Nodejs>

3. Identificación de Funcionalidades

Funcionalidades genéricas

Son funcionalidades que poseen la mayoría de sistemas informáticos orientados ya sea a la web o de escritorio, analizando el software tomado como base se tiene en consideración las siguientes funcionalidades:

- Log In/Iniciar sesión : permite tener acceso a la plataforma.
- Sign In/Registrarse : permite generar un acceso a la plataforma.
- Dashboard/Tablero : permite la navegación en la plataforma.

Funcionalidades del Dominio

Dominio:

UwUnsaac está orientado a una tienda virtual o web store, a diferencia de Zack Market, nuestra plataforma no se limitará al comercio exclusivo de ropa.

Funcionalidades:

Dado el dominio de la plataforma a desarrollar se toma en consideración las siguientes funcionalidades:

- Product Overview/Describir el Producto
- Shopping Bag/Bolsa de compras
- Checkout/Confirmación de la compra
- Checkout Success/Comprobación exitosa de la compra

Otras Funcionalidades

- Autocomplete/Autocompletado
- Header/Encabezamiento

Descripción de Funcionalidad del Header

Debido a una incertidumbre con respecto a la funcionalidad del Header, se realizó un análisis minucioso por parte del equipo de desarrollo para facilitar el trabajo de reingeniería. Consecuentemente, tras el análisis se llegó a la siguiente interpretación. Para mejorar la comprensión del funcionamiento se utilizó el caso de uso descriptivo.

El header deberá mostrar:

- Un menú desplegable para que pueda acceder a la sección de departamentos, productos por categoría y vea todos los productos
- Drop list para categorizar los productos de manera instantánea.
- Una sección del usuario donde se mostrará su datos, este podrá acceder al login y logout, carro de compras
- Un buscador autocompletable

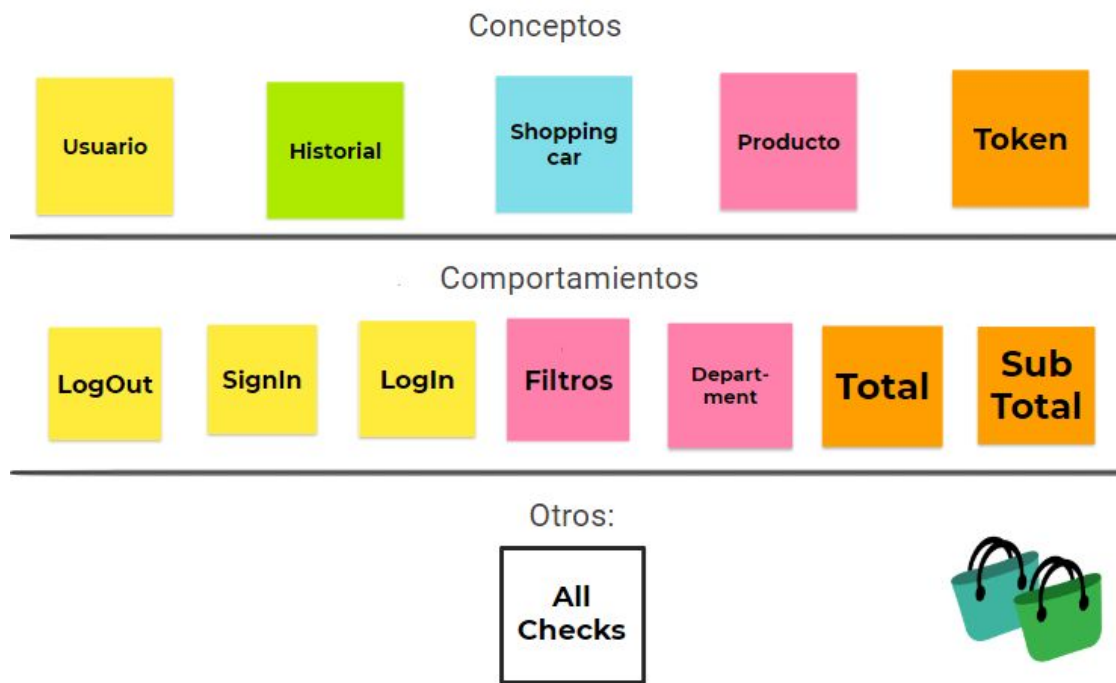


Figura 1 – Modelo basado en conceptos y comportamientos. *Fuente: Elaboración propia*

También podríamos basarnos en el diagrama de base de datos incluido en la documentación del proyecto:

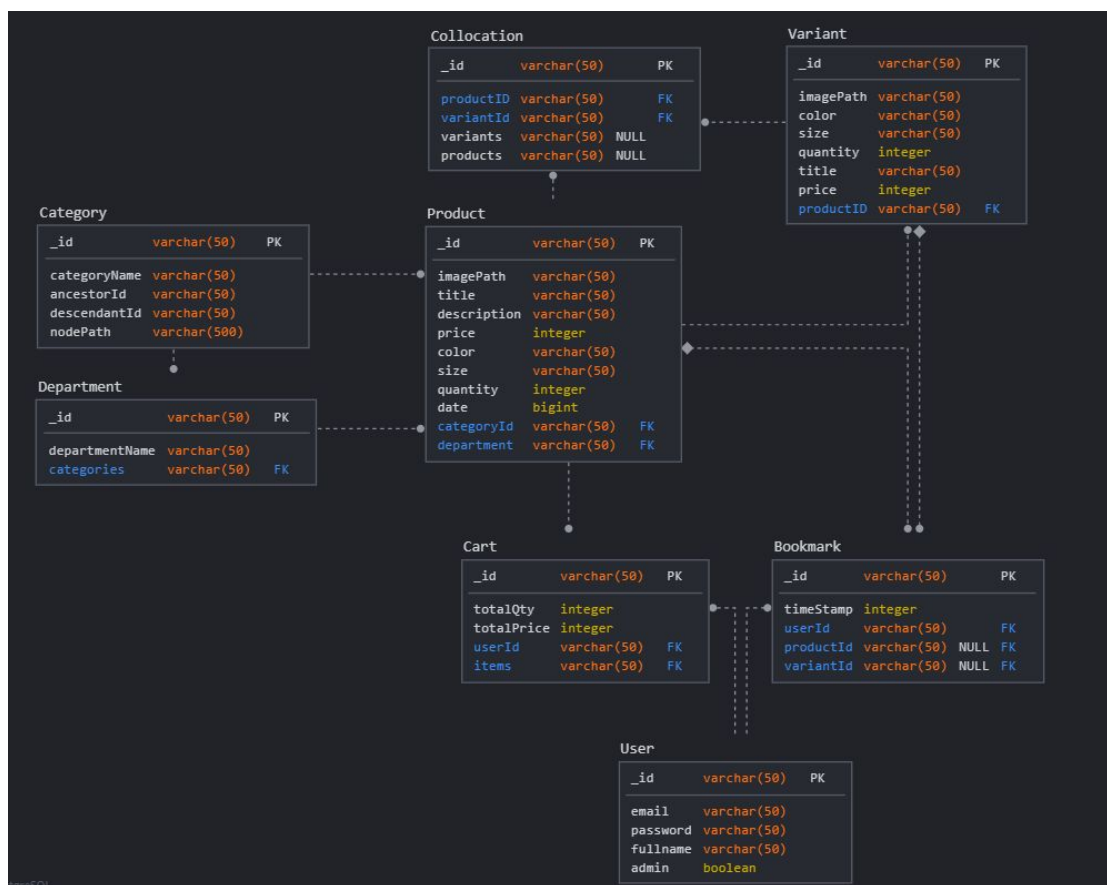


Figura 2 – Diagrama de clases de Zack Market . *Fuente: levelopers/ZackMarket de github*

4. Identificación de Problemas Potenciales

Falta de Documentación en el Repositorio

La documentación del software Zack Market es muy vaga, ya que no existen comentarios claros y concisos que expliquen el funcionamiento del código, lo cual dificulta la reutilización del código debido a una desventaja a la hora de comprender su estructura y patrón.

Las conexiones a la Base de Datos no están claras

Las conexiones que tiene para la Base de Datos tienen una deficiencia en su codificación ya que presenta partes en blanco, al parecer esperando ser completadas aunque no especifica el puerto de conexión; por ende la dificultad para buscar un puerto compatible aumenta.

ERROR: “database name must be a string”

El módulo de creación de la Base de Datos presenta deficiencias o aparenta estar incompleto; el nombre de la Base de Datos no es compatible con el formato que pide en el código del Back-end del software Zack Market.

5. Identificación de Requisitos Funcionales

La página e-commerce es capaz de filtrar los productos por precios y departamentos. El filtro de precios se divide en 4 rangos (a,b,c,d) y los departamentos son 2 (F,M). Además es posible ordenar los productos de mayor a menor precio.

En cualquier momento se puede limpiar las opciones del filtro

El proyecto e-Commerce posee un header en el cual se pueden realizar 2 acciones:

1. Categorizar todos los productos bajo los siguientes criterios:

La página e-commerce posee la capacidad de segmentar los productos por categorías referentes a género y edad, teniendo las opciones: Kids, Men y Woman. Cada uno de estos segmentos se subdivide

- Kids, que a su vez se divide en boys and girls.
- Men, que a su vez se divide en jeans, knitwear.
- Women, que a su vez se divide en básicos, chaqueta.
- All products, que muestra todos los productos sin distinción.

* La sección boys no comparte ningún producto con cualquiera que esté alojada en men, e igualmente con girls & women; y viceversa.

2. Realizar búsquedas con autocompletado

La página web posee la capacidad de realizar búsquedas con una funcionalidad agregada el cual trata de predecir lo que el usuario está buscando para así minimizar el tiempo en que el usuario se pasa digitando su producto en la barra de búsqueda.

6. Identificación de Requisitos No Funcionales

Usabilidad:

Es una cualidad que permite al usuario una facilidad en la navegación dentro del propio sistema, además que ayuda en gran medida al desempeño de los requisitos funcionales.

Precisión:

Quiere decir que los puntos principales que posee la aplicación deberán estar lo más visible posible para el usuario.

Eficacia:

Consiste en alcanzar las metas establecidas en la empresa.

Eficiencia:

Se refiere a lograr las metas con la menor cantidad de recursos. Obsérvese que el punto clave en esta definición es ahorro o reducción de recursos al mínimo. El sistema debe ser capaz de operar con normalidad incluso cuando existe una cantidad inmensa de usuarios con sesiones concurrentes.

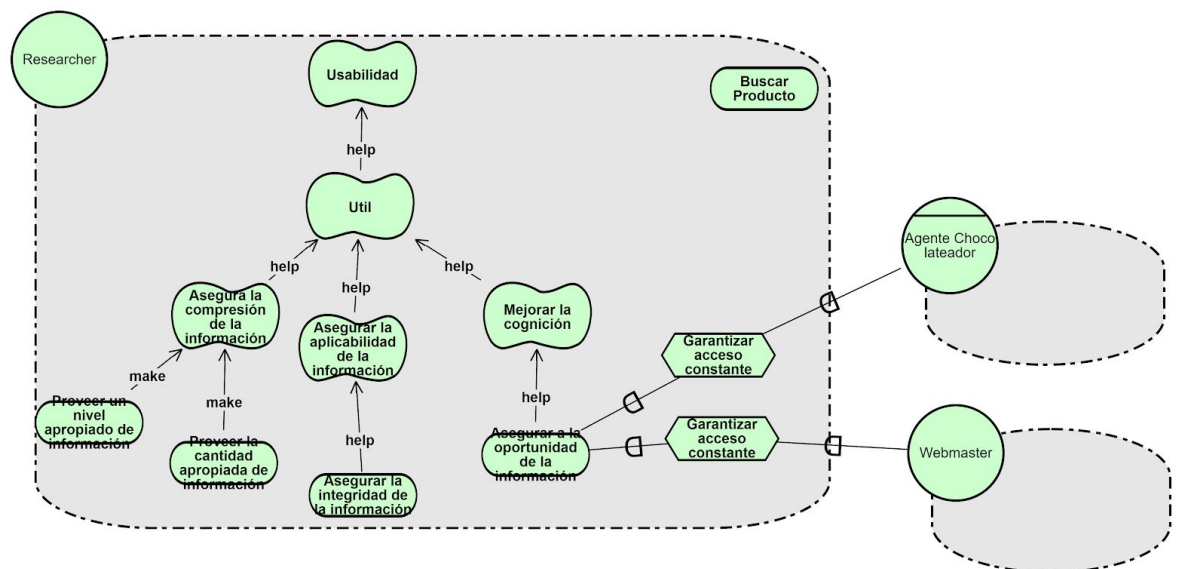


Figura 3 –Modelo de Usabilidad hecha con piStar. Fuente: Elaboración propia.

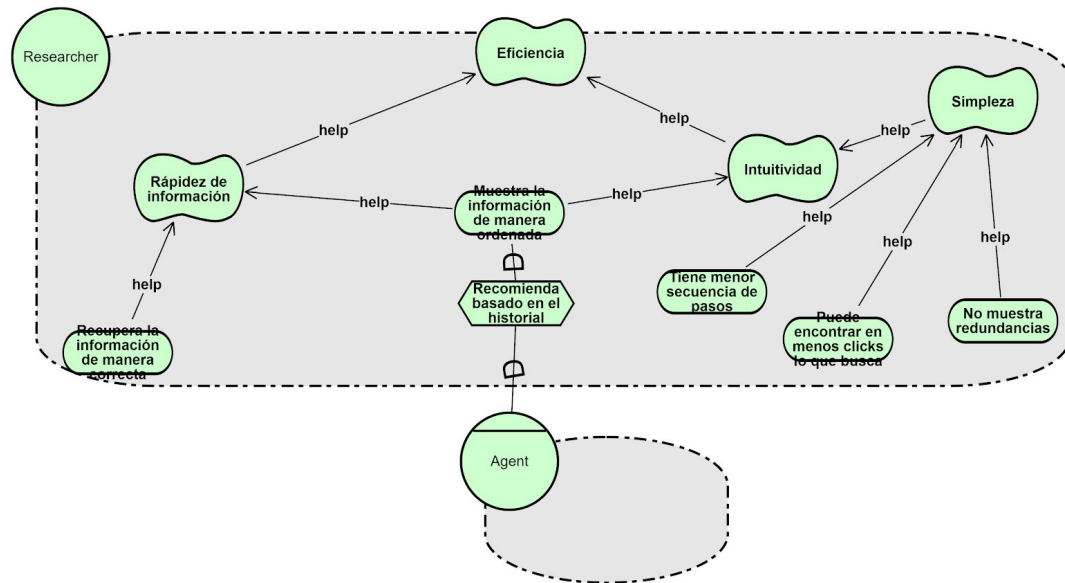


Figura 4 –Modelo de eficiencia hecha con piStar. *Fuente: Elaboración propia.*

7. Herramientas de Desarrollo

- React Developer Tools
- MOZILLA: [React Developer Tools – Consigue esta extensión para Firefox \(es\)](#)
- Chrome: [React Developer Tools](#)
- Visual Studio Code: <https://code.visualstudio.com/download>
- MongoDB: [MongoDB Community Download](#)
- NodeJS: [Descarga](#)

8. ¿Qué somos?

Vendemos un espacio en nuestra web para que los vendedores puedan publicar sus productos y que ellos mismos ofrezcan un medio de contacto para sus posibles clientes. No nos encargamos del proceso de compra. Sólo haremos un catálogo en base a los productos ofrecidos por los vendedores. Nuestra justificación está dada por la necesidad de características (como la intuitividad), faltantes en otras plataformas que puedan ofrecer servicios similares (como el marketplace de facebook).

Problemas:

- La estrategia del google form y el conversor csv /json toma mucho tiempo.

Solución:

- Seguir con google form o hacer el form en react.

9. Producto Mínimo Viable

Tener el programa con requerimientos funcionales (RF) mínimos publicados

Lo mínimo para poder utilizar

- Avance de lo asignado
- Problemas

MPV 1

Tabla 1 – Planeación de soluciones para el MPV.

PROBLEMA	SOLUCIÓN
No conocer la tecnología	Capacitaciones: <ul style="list-style-type: none">• Augusto• Andre
Idioma	Traducir
El vendedor no puede cargar sus productos	Formulario CRUD
No hay estrategia de trabajo	Pair Programming <ul style="list-style-type: none">• Milagros y Roxana• Alexander y Manuel• Frank y Rosmel• Andre y Enrique
Los requisitos no funcionales (NFRs) están aún abstractos	Crear tareas

Avance de lo asignado: Pull Requests

Problemas :

- Repartimos tareas que son dependientes ejem. archivo filter.js
- Modificar los componentes

10. Análisis de Componentes

Diagrama de componentes Back - end

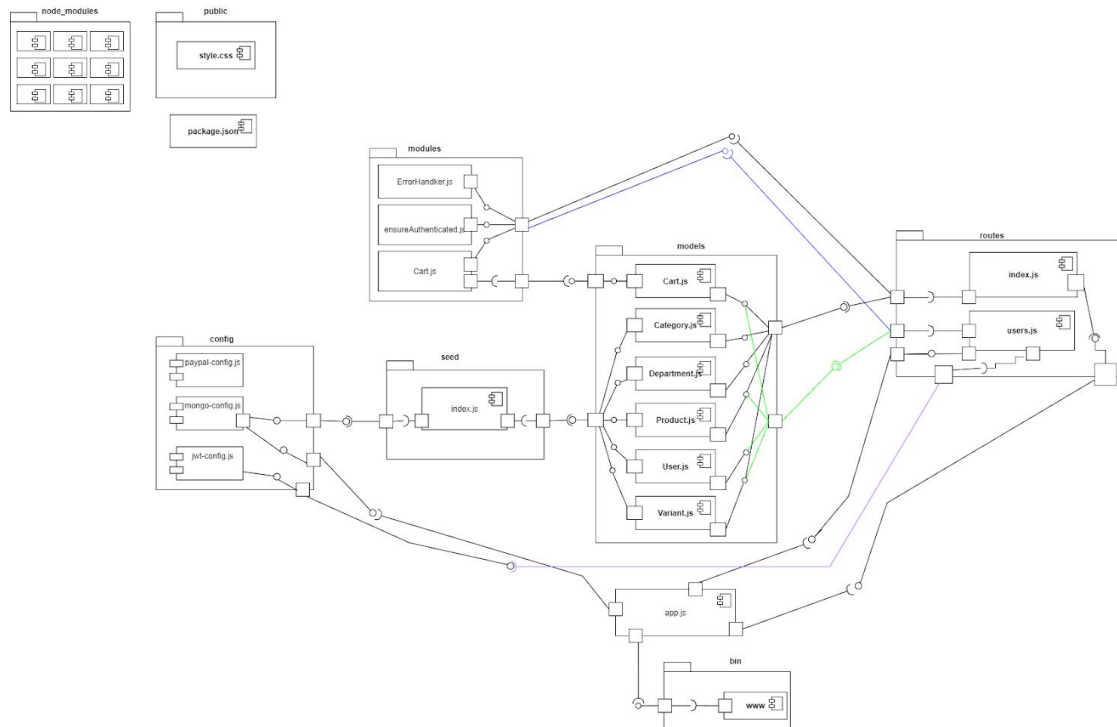


Figura 5 –Diagrama de componentes back-end. *Fuente: Elaboración propia.*

Observaciones:

- El contenido de la carpeta “public\stylesheets” no tiene una relación con otros componentes, dado que el componente “style.css” no es requerido por ningún otro componente, e incluso posee un contenido inservible para esta parte del proyecto.
- El componente “package.json” no tiene una relación directa con el resto de componentes, dado que este archivo contiene las dependencias generales del programa almacenadas en formato json.
- Los require a la carpetas “configs/jwt-config” y “configs/paypal-config” son innecesarias ya que ambos archivos .js no tienen código funcional.
- La carpeta node_modules es el almacén de paquetes que utiliza NodeJS a través de NPM y donde se guardan todas las dependencias de nuestro proyecto.

Observaciones:

- Se generalizaron los contenidos de las carpetas `action` y `reducer`, dado por la gran cantidad de archivos que presentan.
- Las carpetas `stylesheet` contienen las características de los componentes como el tamaño, color, alineación de texto, el padding, el display, flex-direction y las márgenes.

- c. La carpeta checkoutCancel no se relaciona con otros componentes porque se encuentra incompleta.
- d. La carpeta assets presenta archivos del tipo imágenes el cual se relaciona con 4 componentes y que no consideramos por ser imágenes.
- e. La carpeta de loadingAnimation contiene un atributo animado.

11. Metodología

Proceso para producir software de forma organizada, utilizando una variedad de técnicas y convención de notación predefinidas

Aspectos

1. Análisis de un sistema legado
2. La transformación lógica
3. El desarrollo de un nuevo sistema

Procesos

1. Reconstruir la arquitectura
2. Transformación de la arquitectura
3. Desarrollo basado en arquitectura

OAR

Actividad 1: Establecer un contexto de extracción.

Actividad 2: Inventario de componentes

Patrones de Reingeniería

Estableciendo la dirección (metas)(as is) vs (to be)

- Unificar intereses.
- Designar a un navegador
- Hablar alrededor de la mesa.
- Primero lo más valioso
- Arreglar los problemas, no los síntomas.
- Buscar una solución simple

win-software: El feedback de los usuarios.

LOGO

Logotipo:

Tenemos una representación tipográfica del nombre de una marca.



Figura 8 –Logotipo de UWUnsaac. *Fuente: Elaboración propia.*

Isologo:

Unión de logotipo e isotipo, para nuestro proyecto elegimos un isologo.



Figura 9 –Isologo de UWUnsaac. *Fuente: Elaboración propia.*

13. Interacción Intuitiva

Gut feeling

Es un proceso inconsciente, no analítico. Es similar a la toma de decisiones intuitiva en general. La intuición se percibe simplemente como un presentimiento.

Según Hammon(1993) describe la intuición como un proceso cognitivo que de alguna manera produce respuesta, solución o idea sin el uso de un proceso paso a paso consciente y lógicamente defendible.

Verbalizability

Hay la posibilidad de que los usuarios no puedan verbalizar las decisiones individuales y la secuencia de pasos en la interacción intuitiva.

Según Wickens y sus colegas (1998) argumentan que esto se debe a que las decisiones intuitivas se basan en asociaciones de memoria almacenada en lugar de razonamiento per se.

Según Gigerenzer (2013) argumenta que especialmente las personas con alta experiencia en un tema específico toman las mejores decisiones pero, sin embargo, son incapaces a la hora de explicar sus decisiones.

Se puede decir que la intuición ocurre cuando se llega a una conclusión sobre la información menos explicada que obviamente se requiere llegar a una conclusión.

Effortlessness

La intuición intuitiva suele parecer rápida y sin esfuerzo, muchos de los usuarios enfatizan manipula el producto sin ningún tipo de tensión, antes de razonar claramente ya se tiene un objetivo, lo que implica en las descripción en la toma de decisiones intuitiva.

Experiencia mágica (Magical experience)

Muchas de las veces la Interacción Intuitiva provoca que el usuario experimente la Experiencia Mágica, donde los usuarios califican la interacción como “impresionante”, “asombroso” o incluso “mágica”. La investigación refiere a que para lograr dicha experiencia es provocada por el

conocimiento previo que el usuario posee, pero a la misma vez esta se respalda mediante la introducción de nuevas tecnologías.

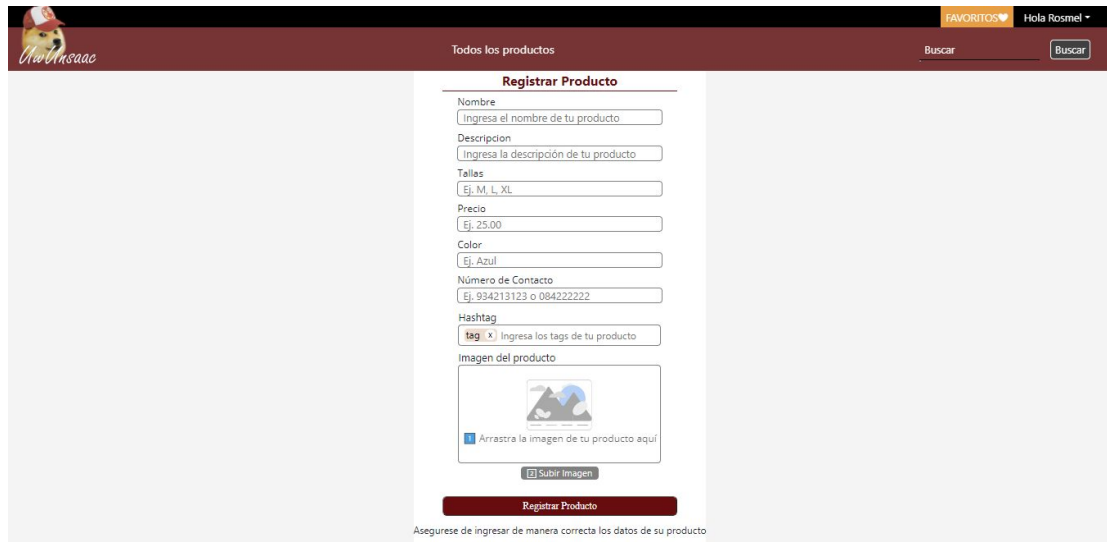
Tabla 2 – Funcionalidad/aspecto de intuitividad: Nos muestra en qué intuitivos ayudaran las funcionalidades ya descritas .

Funcionalidad / aspecto de intuitividad	effortlessness	magical experience	gut feeling	verbalizability
categorias y departamentos	Ayuda para la subida de productos	Ayuda	Ayuda	Ayuda
CARGADOR DE IMÁGENES	Ayuda para la subida de productos		Ayuda	
BARRA DE PRECIOS	Ayuda para la selección de productos usando el filtrado			
barra para seleccionar categorías y departamentos	Ayuda para la selección de productos usando el filtrado	Ayuda	Ayuda	Ayuda
ICONOS	No ayudará en gran medida	Ayuda		ayuda
BUSCADOR	Ayudará en búsquedas precisas	Ayuda		

14. Deployment

Inserción de productos

La inserción de los productos consta del nombre del producto, la descripción del producto, las tallas M, L, XL. El precio del producto, el color, el número de contacto, “hashtag” es decir, una etiqueta clicable, una palabra clave sobre el producto. También el acceso a subir la imagen del producto a la página web. Finalmente registrar el producto en la página web.



The screenshot shows a web application interface for registering a product. The header includes the 'UWUnsaac' logo, a navigation bar with 'Todos los productos', and a user profile section with 'FAVORITOS' and 'Hola Rosmel'. The main content area is titled 'Registrar Producto' and contains a form with the following fields: 'Nombre' (with placeholder 'Ingresa el nombre de tu producto'), 'Descripción' (with placeholder 'Ingresa la descripción de tu producto'), 'Tallas' (with placeholder 'Ej. M, L, XL'), 'Precio' (with placeholder 'Ej. 25.00'), 'Color' (with placeholder 'Ej. Azul'), 'Número de Contacto' (with placeholder 'Ej. 934213123 o 084222222'), and 'Hashtag' (with a 'tag' icon and placeholder 'Ingresa los tags de tu producto'). Below these fields is an 'Imagen del producto' section with a placeholder image and the text 'Arrastra la imagen de tu producto aquí'. A 'Subir Imagen' button is located below the image placeholder. At the bottom of the form is a red 'Registrar Producto' button. A footer note states: 'Asegúrese de ingresar de manera correcta los datos de su producto'.

Figura 10 –Inserción de productos en la página de UWUnsaac. *Fuente: Elaboración propia.*

Añadiendo travis para el despliegue

Creamos un archivo llamado `travis.yml` y le colocamos la siguiente configuración para el proyecto actual:

Posteriormente instalamos el Heroku CLI, para luego logearnos desde consola, con el comando “`heroku login`”.

```
! travis.yml
D: > Desktop > ! travis.yml
1  language: node_js
2  node_js:
3    - "stable"
4  cache: npm
5
6  install:
7    - npm ci
8
9  script:
10   - CI='' npm run
11
12  deploy:
13    provider: heroku
14    api_key: $HEROKU_API_KEY
15    app: e-commerce-uwunsaac-backend
16    skip-cleanup: true
```

Figura 11 –Configuraciones de travis. *Fuente: Travis documentation*

```
C:\Users\manuv>heroku login
heroku: Press any key to open up the browser
Opening browser to https://cli-auth.heroku.com/
2gDbQAAAA8xOTAuMjM5LjE5Ny4xNjNuBgBaICRldwFj
Logging in... done
Logged in as 170748@unsaac.edu.pe
```

Figura 12 –Login en heroku.

Y una vez logueados en heroku, recuperamos el token

```
C:\Users\manuv>heroku config:get GITHUB_USERNAME
» Error: Missing required flag:
»   -a, --app APP  app to run command against
»   See more help with --help

Creating OAuth Authorization... done
Client: <none>
ID: 70061026-7df0-4cf5-a71c-48044f11dff2
Description: Long-lived user authorization
Scope: global
Token: 5a3048ad-8125-43eb-a0ab-7a4f0bb79465
Updated at: Tue Feb 02 2021 19:00:45 GMT-0500 (GMT-05:00) (less than a minute ago)
```

Figura 13 – Recuperación de Token

Luego nos vamos al travis y añadimos el token.

Figura 14 – Conexión de travis con github

Y finalmente habilitamos el despliegue desde GitHub en el heroku.



Figura 15 – Habilidad de despliegue de Github a heroku

15. Patrones de Diseño

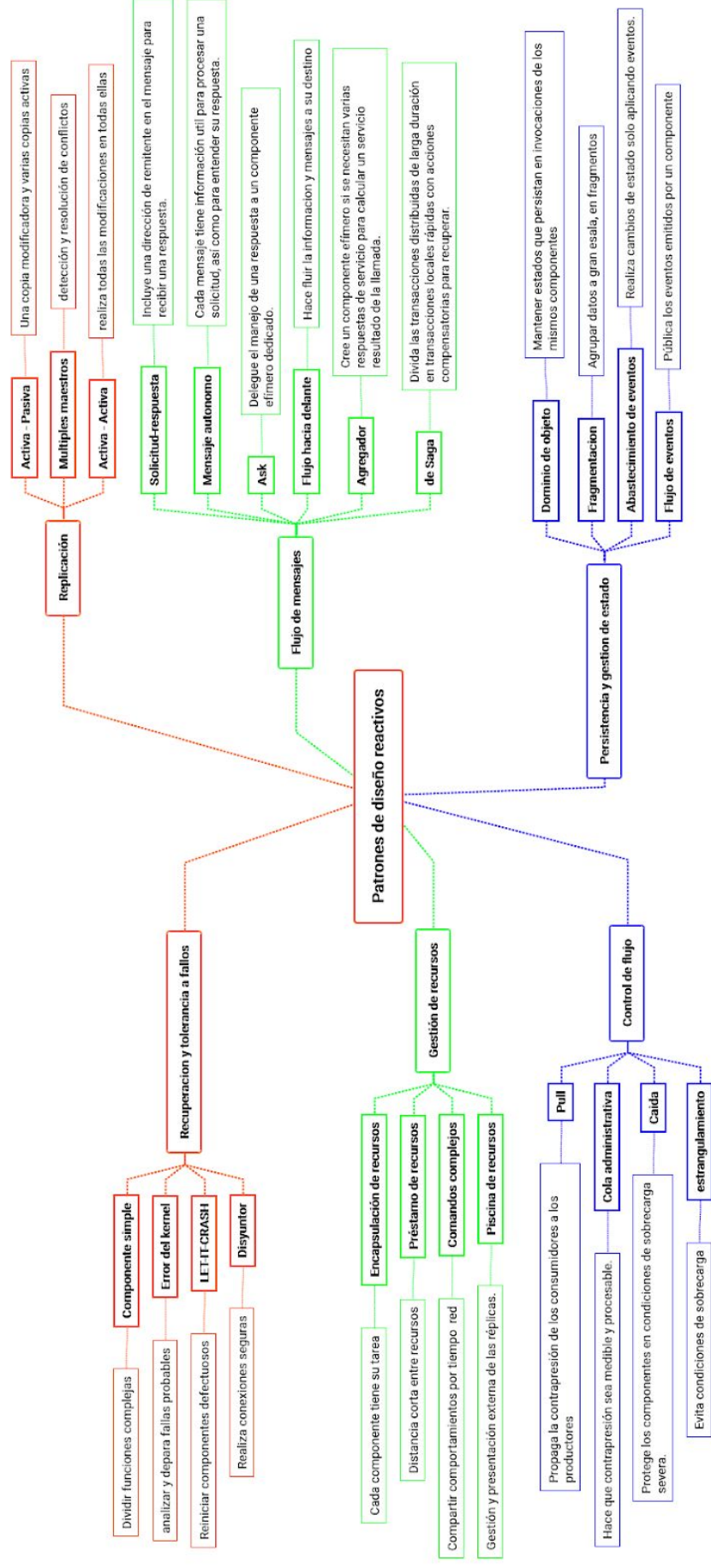


Figura 16 — Mapa mental de los patrones de diseño. Fuente: *Elaboración propia*.

16. Modelo de Características de UWUnsaac

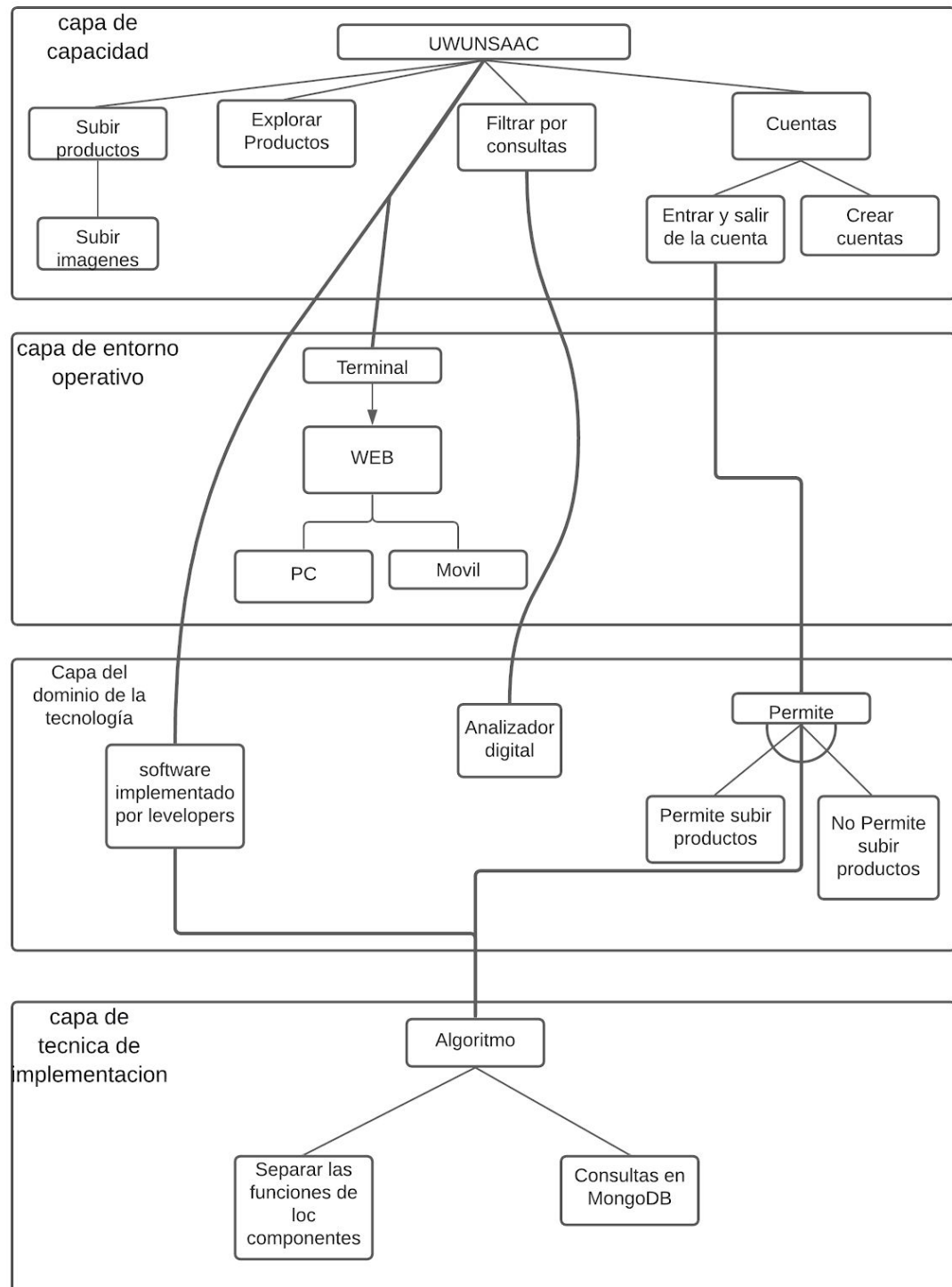


Figura 17 – Modelo de características de UWUnsaac. *Fuente: Elaboración propia.*

17. Resultados

Logramos adaptar la estética de la página a una más acorde con la de la institución, en todas las páginas, incluidas la principal y la de login.

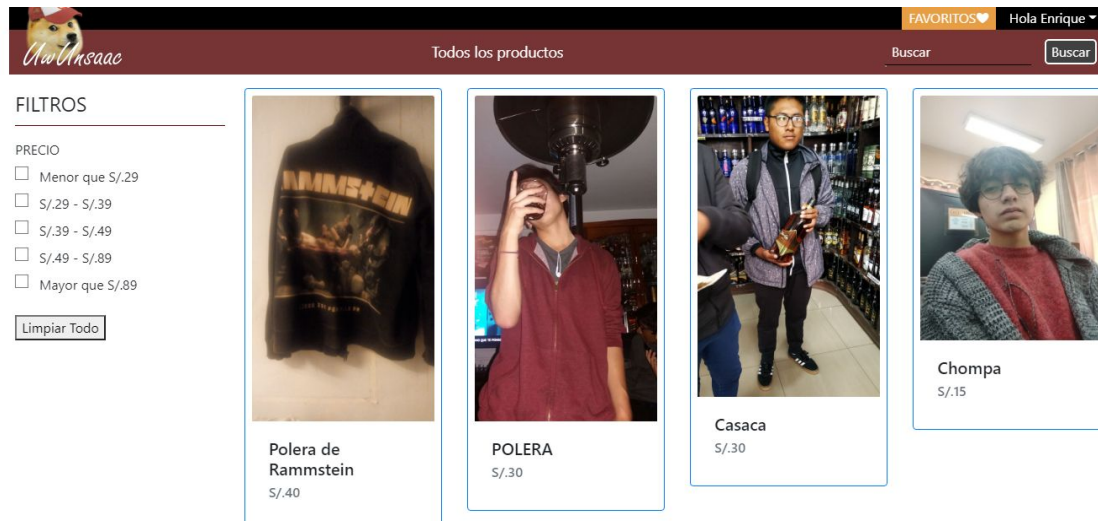


Figura 18 – Página principal de UWUnsaac. *Fuente: Elaboración propia.*

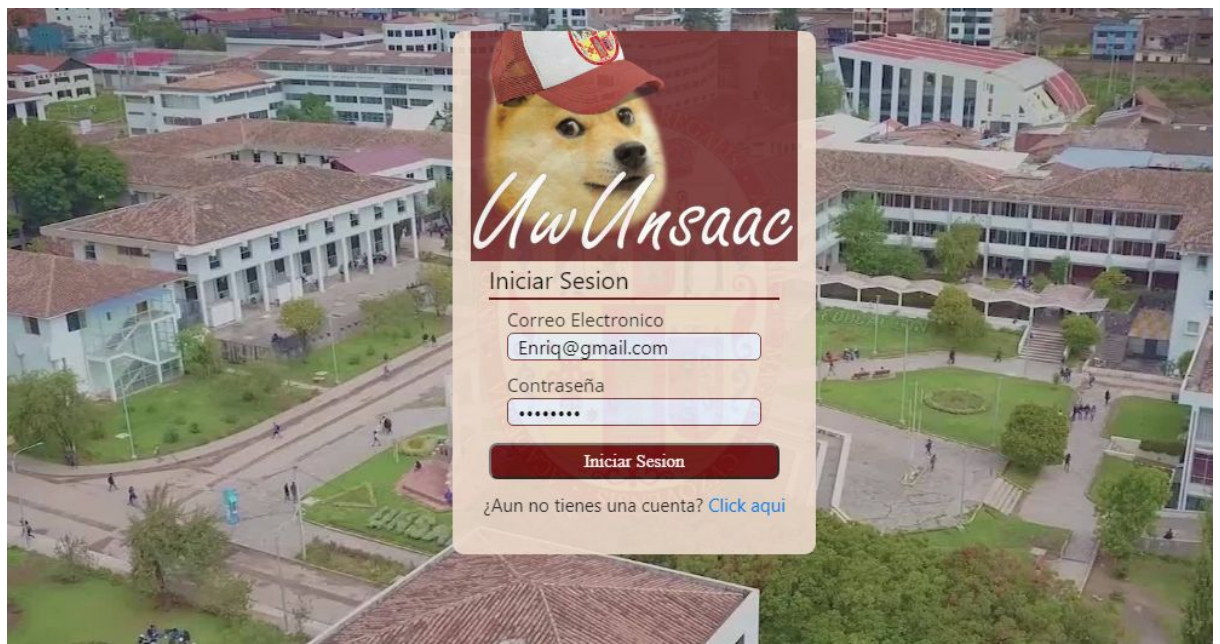


Figura 19 – Página de iniciar sesión de UWUnsaac. *Fuente: Elaboración propia.*

Cambiamos completamente la funcionalidad del carrito hacia la de favoritos

Favoritos

Fotos	Titulo	Contacto	Precio	Eliminar
	<p>Polera de Rammstein</p> <p>Descripción: Ya me exorcizaron y ahora solo escucho arjona</p>		40	
	<p>Poncho andino</p> <p>Descripción: Poncho tejido de lana de alpaca con colores naturales</p>		200	
	<p>POLERA</p> <p>Descripción: polera liviana</p>		30	

Agregar a favoritos ♥

Figura 20 – Página de favoritos de UWUnsaac. *Fuente: Elaboración propia.*

Se añadió la funcionalidad de añadir productos

Registrar Producto

Nombre

Ingresar el nombre de tu producto

Descripción

Ingresar la descripción de tu producto

Tallas

Ej. M, L, XL

Precio

Ej. 25.00

Color

Ej. Azul

Número de Contacto

Ej. 934213123 o 084222222

Hashtag

tag x Ingresar los tags de tu producto

Imagen del producto



Arrastra la imagen de tu producto aquí

Subir Imagen

Registrar Producto

Figura 21 – Registro de productos de UWUnsaac. Fuente: *Elaboración propia*.

18. Bibliografía

- Mateusz Papiernik (Diciembre 5, 2019). Como instalar MongoDB en Ubuntu 18.04.
URL: [Como instalar MongoDB en Ubuntu 18.04](#)
- Aleksander Groth and Daniel Haslwanter (Enero 22, 2016). Usabilidad percibida, atractiva e intuitiva de sitios web receptivos de turismo móvil. Una experiencia de usuario Estudio
URL: [Usabilidad percibida, atractiva e intuitiva de sitios web receptivos de turismo móvil](#)
- Daniel Ullrich, Sarah Diefenbach (Mayo 15, 2014). INTUI. Explorando las facetas de la interacción intuitiva.
URL: [INTUI. Explorando las facetas de la interacción intuitiva](#)
- Ullrich, D., Diefenbach, S. (2010). Cuestionario sobre intuitividad.
URL: http://intuitiveinteraction.net/resource/files/INTUI_en.pdf
- Daniel Ullrich, TU Darmstadt, Alexanderstrabe (2010). Interacción Intuitiva.
URL: <http://intuitiveinteraction.net/model/>
- KevinZhang (Octubre 21, 1994). Design Patterns Elements of Reusable Object-Oriented Software.
URL: [Design Patterns Elements of Reusable Object-Oriented Software](#)
- Jorge Melegati, Alfredo Goldman (2016). Ingeniería de requisitos en startups de software: un enfoque de teoría fundamentada.
URL: [Ingeniería de requisitos en startups de software: un enfoque de teoría fundamentada](#)
- Krzysztof Czarnecki, Simon Helsen, and Ulrich Eisenecker (2005). Formalizar modelos de características basados en cardinalidad y sus especialización.

URL: [Formalizar modelos de características basados en cardinalidad y sus especialización.](#)

- Lee K., Kang K.C., Lee J. (2002) Concepts and Guidelines of Feature Modeling for Product Line Software Engineering. In: Gacek C. (eds) Software Reuse: Methods, Techniques, and Tools. ICSR 2002. Lecture Notes in Computer Science, vol 2319. Springer, Berlin, Heidelberg.
URL: https://doi.org/10.1007/3-540-46020-9_5