

**FACULTAD DE INGENIERÍA**

CARRERA DE INGENIERÍA DE SISTEMAS COMPUTACIONALES

**Diseño e implementación del Sistema web e-commerce para mejorar el proceso de ventas online de la empresa WAYKI STORE de la ciudad de Lima en el año 2020**

Informe académico

**Autor(es):**

Maricielo Milagros Duran Torres

Luis Fernando Loli Motta

Abraham Giovanni Sánchez Cruces   
Alejandro Axel Holguín Andrade

**Curso:**

Diseño y arquitectura de software

**Docente:**

Jorge Alfredo Guevara Jiménez

LIMA – PERÚ

2020-2

INDICE GENERAL

[Capítulo 1 5](#_Toc52255717)

[IDENTIFICACIÓN DEL PROBLEMA 5](#_Toc52255718)

[PLANEAMIENTO DE LA SOLUCIÓN 5](#_Toc52255719)

[IMPLEMENTACIÓN DE LA SOLUCIÓN 6](#_Toc52255720)

[JUSTIFICACIONES DE LA INVESTIGACIÓN 6](#_Toc52255721)

[LIMITACIONES DE LA INVESTIGACIÓN 6](#_Toc52255722)

[VENTAJAS 7](#_Toc52255723)

[DESVENTAJAS 7](#_Toc52255724)

[OBJETIVOS 8](#_Toc52255725)

[Objetivo General: 8](#_Toc52255726)

[Objetivos Específicos: 8](#_Toc52255727)

[Capítulo 2 9](#_Toc52255728)

[MARCO TEÒRICO 9](#_Toc52255729)

[SOFTWARE: 9](#_Toc52255730)

[PROCESO DE NEGOCIO: 9](#_Toc52255731)

[REQUISITOS DE SOFTWARE: 9](#_Toc52255732)

[CASOS DE USO 9](#_Toc52255733)

[ESPECIFICACIÓN DE CASOS DE USO 9](#_Toc52255734)

[PLAN DE ITERACIONES 10](#_Toc52255735)

[MVC 10](#_Toc52255736)

[ACME 10](#_Toc52255737)

[DIAGRAMA DE SECUENCIA 10](#_Toc52255738)

[VISTA ARQUITECTÒNICA DE HW Y SW: 10](#_Toc52255739)

[PATRÒN DE ARQUITECTURA EN CAPAS: 10](#_Toc52255740)

[PATRÒN ARQUITECTÓNICO (TUBERIA Y FILTRO): 11](#_Toc52255741)

[HERRAMIENTAS 12](#_Toc52255742)

[GITHUB 12](#_Toc52255743)

[BALSAMIQ 12](#_Toc52255744)

[LUCIDCHART 12](#_Toc52255745)

[SUBLIME TEXT 12](#_Toc52255746)

[SPRING TOOLS SUITE 12](#_Toc52255747)

[MYSQL 12](#_Toc52255748)

[BOOTSTRAP 13](#_Toc52255749)

[HTML5 13](#_Toc52255750)

[JQUERY 13](#_Toc52255751)

[MARCO METODOLÒGICO 14](#_Toc52255752)

[Capítulo 3 16](#_Toc52255753)

[DESARROLLO DE LA SOLUCIÒN 16](#_Toc52255754)

[Capítulo 4 35](#_Toc52255755)

[REFERENCIAS 35](#_Toc52255756)

[ANEXO 36](#_Toc52255757)

**INDICE DE FIGURA**

[Figura 1 Mapa de procesos 16](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387412)

[Figura 2 Diagrama de casos de uso 17](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387413)

[Figura 3 especificación de casos de uso 18](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387414)

[Figura 4 Requisitos funcionales 19](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387415)

[Figura 5 plan de iteraciones 20](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387416)

[Figura 6 MVC-ITERACION 1 20](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387417)

[Figura 7 MVC-ITERACION 2 21](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387418)

[Figura 8 MVC- ITERACION 3 21](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387419)

[Figura 9 ACME 22](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387420)

[Figura 10 DER 23](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387421)

[Figura 11 ingresar cuenta administrador 24](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387422)

[Figura 12 registrar producto 24](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387423)

[Figura 13 buscar producto a modificar 25](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387424)

[Figura 14 modificar producto 25](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387425)

[Figura 15 buscar producto a eliminar 26](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387426)

[Figura 16 eliminar producto 26](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387427)

[Figura 17 visualizar categoría de productos 27](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387428)

[Figura 18 mostrar información/detalle del producto 27](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387429)

[Figura 19 DIAGRAMA DE CALLES 28](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387430)

[Figura 20 VISTA ARQUITECTÒNICADE HW Y SW 29](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387431)

[Figura 21 patrón de arquitectura en capas 30](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387432)

[Figura 22 PATRÓN ARQUITECTÒNICO (TUBERIA Y FILTRO) REQUISITO 1 31](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387433)

[Figura 23 Prototipo Reporte – requisito 1 31](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387434)

[Figura 24 GUI Reporte – requisito 1 32](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387435)

[Figura 25 PATRÓN ARQUITECTÒNICO (TUBERIA Y FILTRO) REQUISITO 2 33](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387436)

[Figura 26 Prototipo Reporte – Requisito 2 33](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387437)

[Figura 27 GUI Reporte – Requisito 2 34](file:///C:\Users\Eva\Documents\GitHub\Nueva%20carpeta\DIARS-PROYECTO\DIARS_INFORME.docx#_Toc52387438)

# Capítulo 1

## IDENTIFICACIÓN DEL PROBLEMA

En la actualidad, sobre todo en esta situación de pandemia (Covid-19) las personas se ven obligadas a evitar acercamiento o contacto físico con otras personas para evitar contagiar o ser contagiados. La comunicación de hoy en día es posible a distancia gracias al Internet y los medios de comunicación. Y no solo ello, sino también gracias a los dispositivos que son realmente necesarios como las computadoras. Es por ello que la empresa Wayki Store ubicado en Mza. I Lote. 2 urb Luis Vallejo Santoni ,con el ruc:20606201207 tiende a tener dificultades con el proceso de venta ya que los únicos canales de comunicación con los que cuentan no llegan a ser tan eficientes como un sistema web automatizado.

## PLANEAMIENTO DE LA SOLUCIÓN

Para solucionar los procesos de venta de la empresa Wayki Store

Se implementará un sistema web e-commerce en donde se mostrarán los productos ordenados en categorías y con una vista de catálogo para una mayor facilidad hacia los clientes.

## IMPLEMENTACIÓN DE LA SOLUCIÓN

Este sistema se implementará con el fin de obtener un manejo automatizado y eficiente de las ventas o control de los productos ya sea para mostrar la información como precios, valoración o stock de cada artículo a venderse.

## JUSTIFICACIONES DE LA INVESTIGACIÓN

* aprender sobre la aplicación y creación de un sistema web para una empresa.
* Poner en práctica los conocimientos adquiridos en la universidad durante los ciclos anteriores
* Ofrecer una alternativa para las actividades cotidianas que se dan día a día en la empresa

## LIMITACIONES DE LA INVESTIGACIÓN

* Acceso a internet de manera constante
* cumplir con los plazos de tiempo establecidos
* recursos (hardware) disponibles y óptimos

## VENTAJAS

* Facilidad para mostrar productos
* Abierto las 24 hr en el medio online
* Costo de mantenimiento barato
* Procesos automatizados
* Optimizar el tiempo
* Más facilidades para el consumidor

## DESVENTAJAS

* Desconfianza en los pagos online
* menor fidelidad del cliente
* fallos técnicos
* poseer un dispositivo electrónico
* Estar conectado a internet constantemente
* falta de contacto con el cliente

## OBJETIVOS

Objetivo General:Diseñar e implementar el sistema web e-commerce para mejorar el proceso de ventas online

### Objetivos Específicos:

-Diseñar e implementar el sistema web e-commerce para lograr ver y consultar los productos disponibles en un catálogo de productos.

-Diseñar e implementar el sistema web e-commerce para lograr agregar y modificar los productos

-Diseñar e implementar el sistema web e-commerce para tener control del registro y eliminación de los productos

-Diseñar e implementar el sistema web e-commerce para seleccionar los productos a comprar

-Diseñar e implementar el sistema web e-commerce para que los usuarios registran sus compras

-Diseñar e implementar el sistema web e-commerce para tener un control sobre los usuarios

-Diseñar e implementar el sistema web e-commerce para elegir el método de pago y visualizar los comprobantes de venta.

-Diseñar e implementar el sistema web e-commerce para darle opciones al usuario sobre el tipo de envío del producto

-Diseñar e implementar el sistema web e-commerce para tener más control sobre las compras e información del usuario.

# Capítulo 2

## MARCO TEÒRICO

SOFTWARE:Según (Sommerville, 2005) “son todos los documentos asociados y la configuración de datos que se necesitan para que los programas puedan operar de manera correcta”.

PROCESO DE NEGOCIO: Según (Sparks, 2000) “Un proceso de negocio es una colección de actividades diseñadas para producir una salida específica para un cliente o un mercado en particular”

REQUISITOS DE SOFTWARE:Según (Sommerville, 2005) son la “descripción de servicios proporcionados por el sistema y sus restricciones operativas.”

CASOS DE USO: Según (Fowler & Scott, 1999) un caso de uso “se obtiene hablando con los usuarios habituales y analizando con ellos las distintas cosas que deseen hacer con el sistema”.

ESPECIFICACIÓN DE CASOS DE USO**:** Según (Pressman, 2010) una especificación de caso de uso “se obtiene cuando un caso de uso básico presenta una historia de alto nivel que describe la interacción entre actor y sistema”.

PLAN DE ITERACIONES**:** Según (Pressman, 2010) afirma que “cada componente del software se elabora a través de cierto número de iteraciones que apliquen paso a paso el concepto de refinamiento”.

MVC**:** Según (Pressman, 2010) señala que “el CVM es en realidad un patrón de diseño arquitectónico desarrollado para el ambiente Smalltalk que puede usarse para cualquier aplicación interactiva”.

ACME**:** Según (Pressman, 2010) opina que ACME “puede considerarse como un lenguaje de descripción arquitectónica (LDA) de segunda generación, ya que su objetivo es identificar una clase de mínimo común denominador de los LDA”

DIAGRAMA DE CALLES**:** SO también llamado diagrama de actividad, según (Sommerville, Ian, 2011) ”Los diagramas de actividad intentan mostrar las actividades que incluyen un proceso de sistema, asi como el flujo de control de una actividad a otra.”

### VISTA ARQUITECTÒNICA DE HW Y SW:

Para (Sommerville, Ian, 2011) la vista arquitectónica “muestra como un sistema se descompone en módulos, cómo interactúan los procesos de tiempo de operación o las diferentes formas en que los componentes del sistema se distribuyen a través de una red local”

### PATRÒN DE ARQUITECTURA EN CAPAS:

El patrón de arquitecturas en capas representa servicios escalables ya que según (Sommerville, Ian, 2011) “Organiza el sistema en capas con funcionalidad relacionada con cada capa”

### PATRÒN ARQUITECTÓNICO (TUBERIA Y FILTRO):

Según afirma (Sommerville, Ian, 2011) “Es un modelo de la organización en tiempo de operaciones de un sistema”

### PATRONES GRASP:

Según (Larman, 2003) “El patrón GRASP ayuda a entender el diseño de objetos esencial, y aplica el razonamiento para el diseño de forma sistemática, racional y explicable”

### PATRÒN ARQUITECTÓNICO (DE REPOSITORIO):

Según afirma (Sommerville, Ian, 2011) menciona que “Todos los datos en un sistema se gestionan en un repositorio central, accesible a todos los componentes del sistema.”

DIAGRAMA DE SECUENCIA**:** Según (Pressman, 2010) “El diagrama de secuencia indica la forma en la que los eventos provocan transiciones de un objeto a otro”.

DIAGRAMA DE SUBSISTEMA**:**

Según (Sommerville, Ian, 2011) “El diagrama de subsistema no sirve para comprender las relaciones entre software y el entorno externo”

## HERRAMIENTAS

GITHUB**:** Según (Castillo, 2012) “GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones git””.

BALSAMIQ**:** Según (santamarìa, 2013) Balsamiq es una aplicación que facilita y agiliza la creación de bocetos con el fin de ayudar al desarrollo de aplicaciones”.

LUCIDCHART**:** Según (Gisbert, 2018) “Lucidchart es un software que permite diseñar y compartir todo tipo de representaciones gráficas con el objetivo de satisfacer las necesidades de comunicación de los procesionales”.

SUBLIME TEXT**:** Según (Genbeta, 2012) “sublime text es una herramienta muy ligera concebida para programar sin distracciones”

SPRING TOOLS SUITE**:** Según (davidmarco, 2010) “Spring tool es un IDE basado en la versión javaEE de eclipse, pero altamente customizado para trabajar con spring framework.”

MYSQL**:** Según (Margaret, s.f.) señala que “MySQL es un sistema de gestión de base de datos relacional de código abierto, basado en el lenguaje de consulta SQL”.

BOOTSTRAP**:** Bootstrap es un framework que permite la creación de interfaces con css y JavaScript. Según (Solis, 2014) “El sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo”

HTML5**:** Según (Perez , 2018) afirma que HTML “Es un estándar que sirve para definir la estructura y el contenido de una página”CSS3**:** Según (Pèrez & Merino, 2017) afirma que CSS “Es un lenguaje que permite presentar, de manera estructurada, un documento que fue escrito en un lenguaje de marcado” JAVA**:** Según señala (MX, Editorial Definición, 2014) “Java es un lenguaje de programación orientado a objetos que se incorporó al ámbito de la informática en los años noventa”JAVASCRIPT**:** Según señalan (Pèrez, juliàn; Gardey, Ana, 2018) “Es un lenguaje formal que brinda instrucciones a una computadora para generar ciertos datos”.

JQUERY**:** Según señalan (Pèrez , juliàn; Gardey, Ana, 2018)“Es una librería o biblioteca de JavaScript que facilita la programación en este lenguaje”.

**SOA:** Según señalan([PowerData](https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/author/redacci%C3%B3n-powerdata)**,2018** ) “**La**[arquitectura orientada a servicios (SOA)](https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/que-es-soa-y-su-diagrama-de-arquitectura-para-integracion-de-datos)**es el nexo que une las metas de negocio con el sistema de software,** de aportar flexibilidad, desde la automatización de las infraestructura y herramientas necesarias consiguiendo, al mismo tiempo, reducir los costes de integración”.

**Patrón CRC-Clase-responsabilidad-colaboración:** “Según señala (Sedeci) para modelar un patrón de seguridad mediante un sub-escenario y conducir el estudio y análisis de problemas que demandan servicios de seguridad.”

## MARCO METODOLÒGICO

Para la realización del proyecto se tomó en cuenta una serie de pasos para el desarrollo del estudio:

**1.-** Se tomó la decisión de contactar a una empresa del rubro de ventas de computadoras.

**2.-**Se procedió a contactarse con alguna empresa en dicho rubro para brindar la creación de un sistema web gratuito con fines educativos como parte de un proyecto.

**3.-**Luego de recibir la autorización de la empresa, como requisito se les pidió responder algunas preguntas cortas para poder modelar los procesos.

**4.-**Primero, se comenzó a modelar los procesos de la organización mediante el mapa de procesos.  
 **5.-** Luego se analizó los procesos del cliente en la empresa para ver que requisitos funcionales se necesitan en el sistema web.

**6.-** Después para ver el comportamiento de aquellos requisitos se usó el diagrama de casos de uso.

**7.-** Posteriormente se relacionaron cada caso de uso de acuerdo con sus procesos.

**8.-** A su vez se procedió a especificar los dos casos de uso principales.

**9.-** Seguidamente se dividió el total de requisitos en tres iteraciones usando el plan de iteraciones, añadiendo la meta y el valor en cada iteración.

**10.-** para tener una idea gráfica de como quedarían las iteraciones se realizaron los prototipos.

**11.-** Teniendo ya identificados los casos de uso de cada iteración, se empezó a agregar las tres clases principales de software aplicando el MVC (modelo vista controlador).

**12.-** Luego de definir los procesos de software se agregó los cuatro tipos principales de programación en base de datos aplicando el modelo ACME (adiciona, consulta, modifica y elimina).

**13.-** Seguido de ello, para tener una idea gráfica y a detalle del comportamiento de los procesos, se realizó el diagrama de calles

**14.-** Para ver las especificaciones del hardware y software que se usará a lo largo de desarrollo del proyecto, se aplicó la vista arquitectónica del HW y SW**.**

**15.-** Para ver las especificaciones del hardware y software que se usará a lo largo de desarrollo del proyecto, se aplicó la vista arquitectónica del HW y SW**.**

**16.-** Inmediatamente, se procedió a realizar el plan de desarrollo en capas para establecer una serie de pasos al momento de desarrollar y ejecutar el proyecto.

**17.-** También se realizó el patrón arquitectónico tubería y filtro para diagramar los procesos de la peticiones en base de datos

**18.-**Para entender el diseño del sistema se usó los patrones GRASP, comenzando con el patrón de repositorio para dar a conocer los archivos y códigos trabajados dentro del repositorio GitHub,

**19.-** Luego, se realizó el diagrama de casos de uso de la iteración 1 mostrando todos los requisitos que está conllevará.

**20.-** Seguidamente, se especificó el caso de uso principal mediante la especificación de casos de uso.

**22.-** Teniendo el diagrama de casos de uso de la iteración 1, se procedió a realizar el diagrama de secuencia con los procesos que tendrá el usuario o administrador en está iteración.

**23.-** Finalmente, se realizó el diagrama de subsistemas para visualizar las relaciones de cada subsistema con otros subsistemas.

# Capítulo 3

## DESARROLLO DE LA SOLUCIÒN

**MAPA DE PROCESOS**

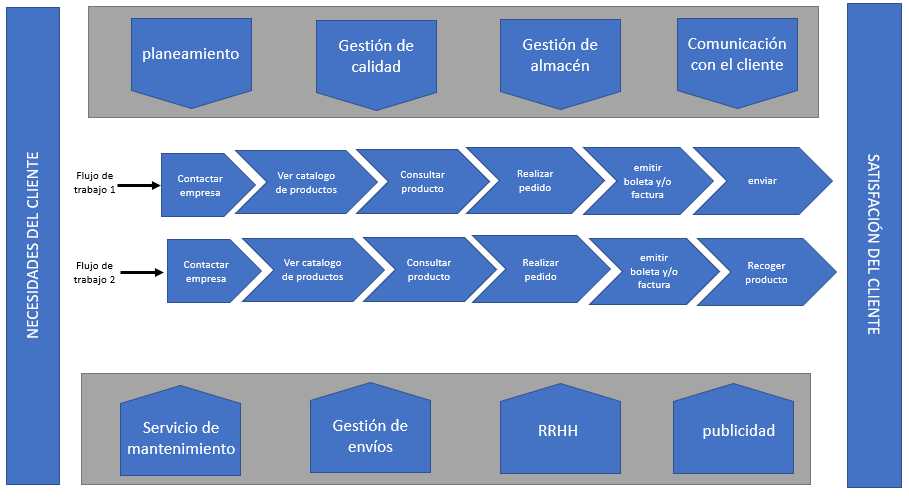


Figura 1 Mapa de procesos

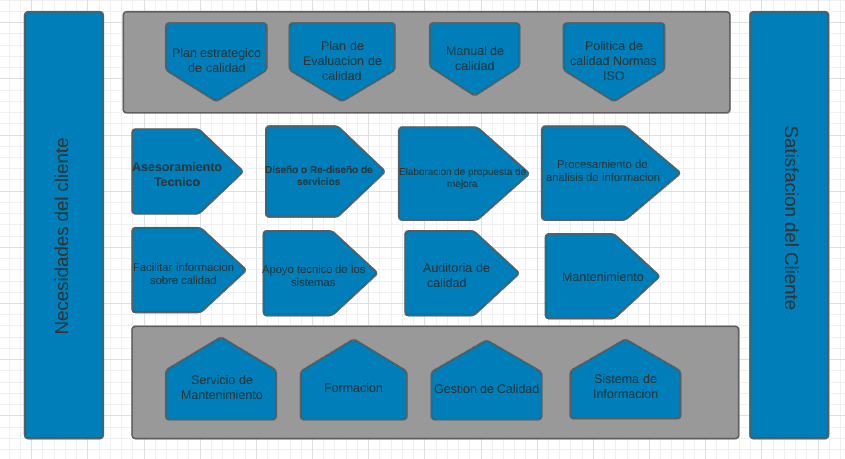


Figura 2 Mapa de procesos de Calidad

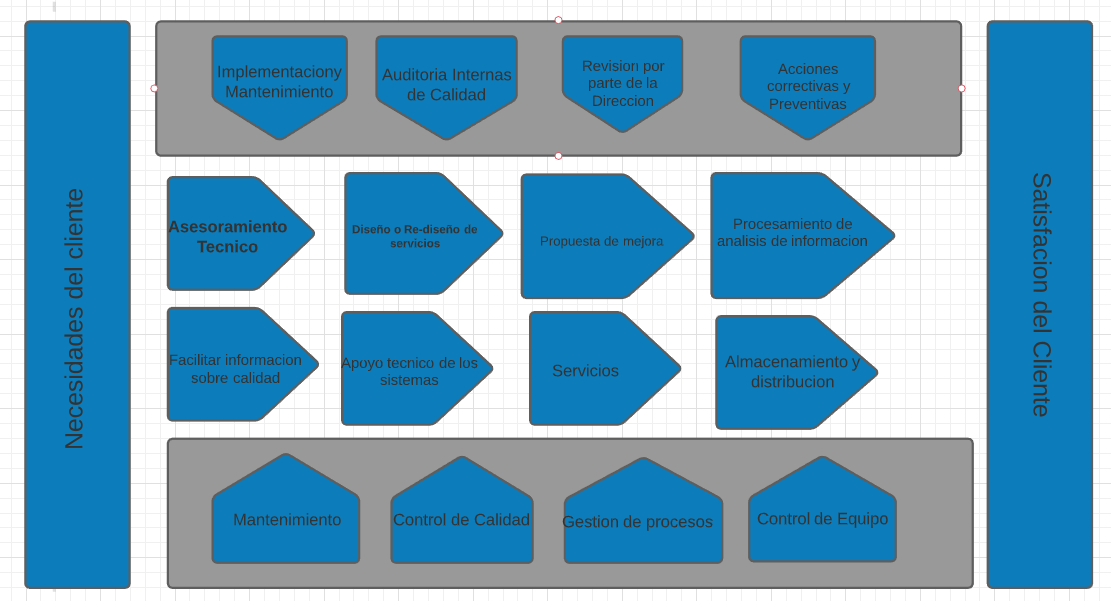


Figura 3 Mapa de procesos de Producción

**DIAGRAMA DE CASOS DE USO**

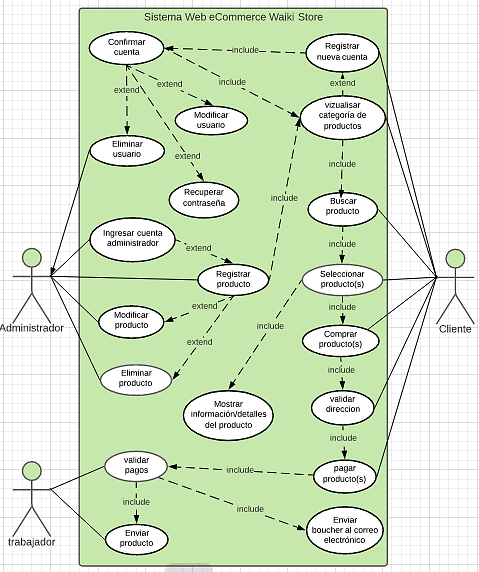


Figura 2 Diagrama de casos de uso

**ESPECIFICACION DE CASOS DE USO**



Figura 3 especificación de casos de uso

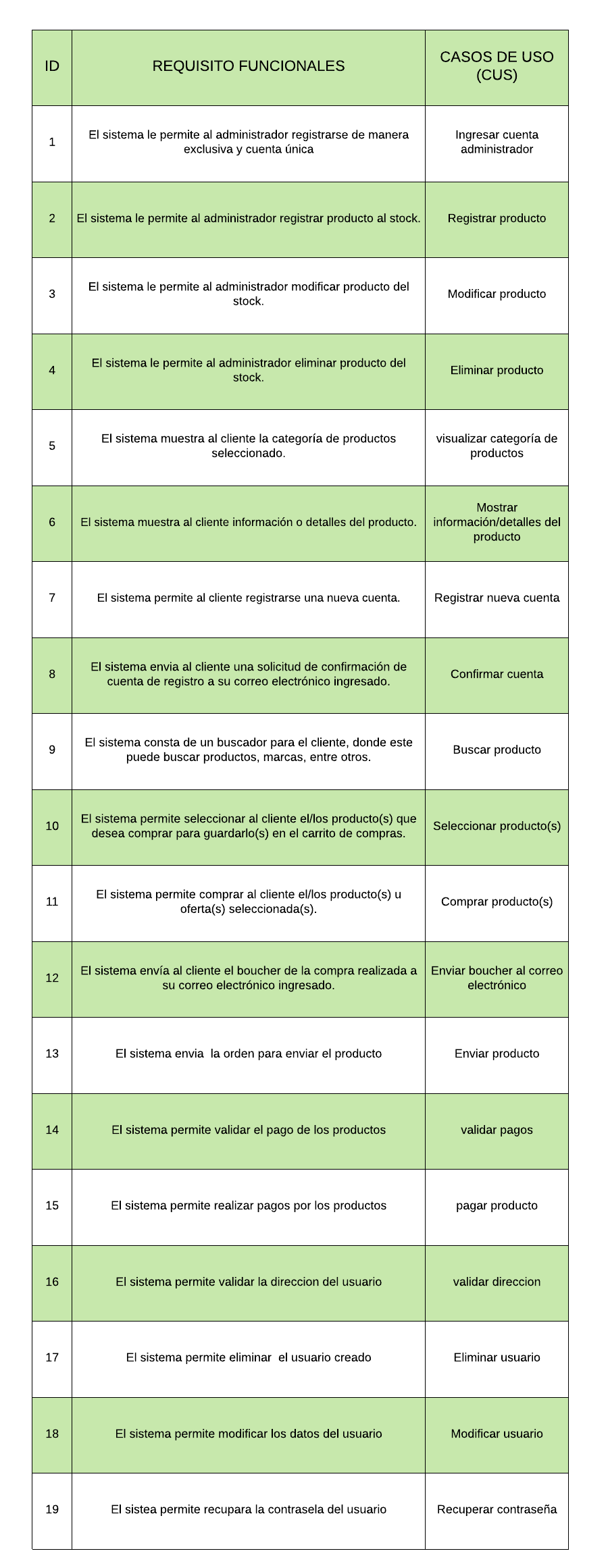
**REQUISITOS FUNCIONALES**

Figura 4 Requisitos funcionales

**PLAN DE ITERACIONES**

**MVC**

Figura 5 plan de iteraciones

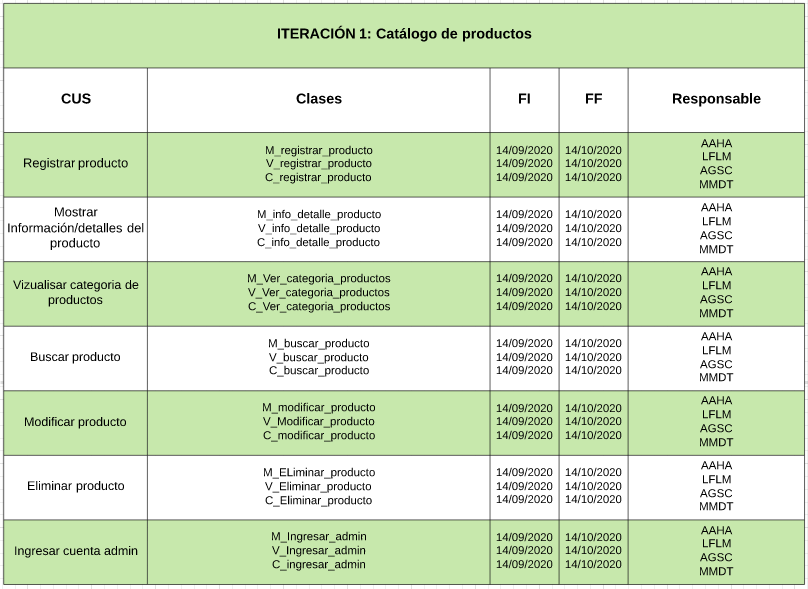


Figura 6 MVC-ITERACION 1

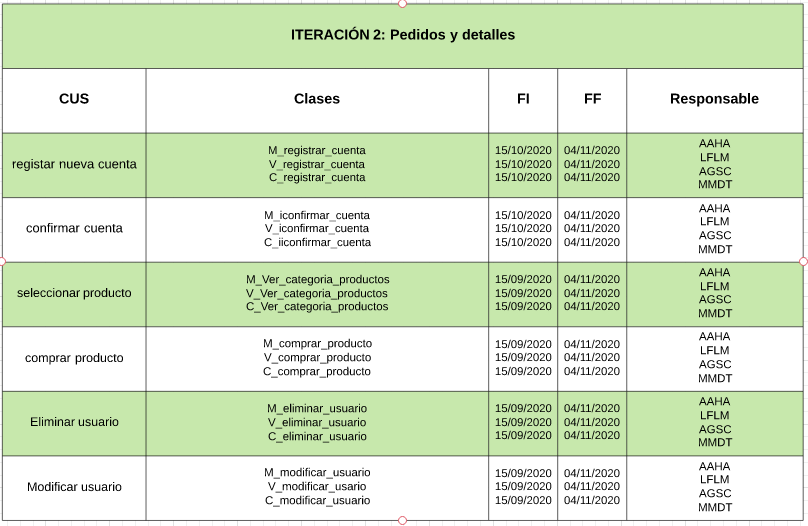


Figura 7 MVC-ITERACION 2

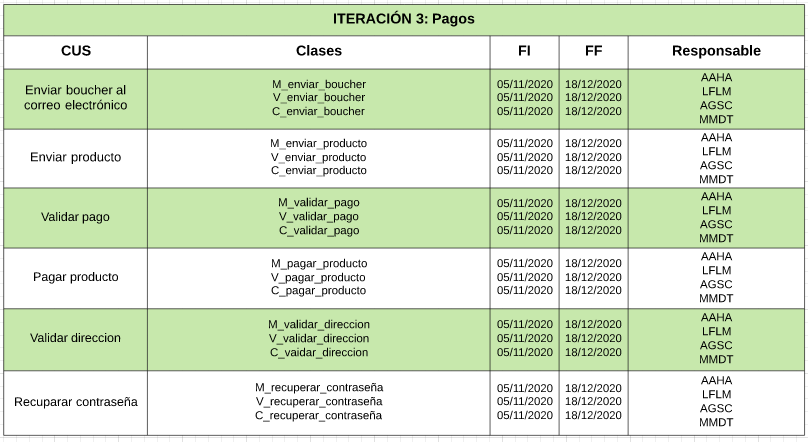


Figura 8 MVC- ITERACION 3

**ACME**

Figura 9 ACME

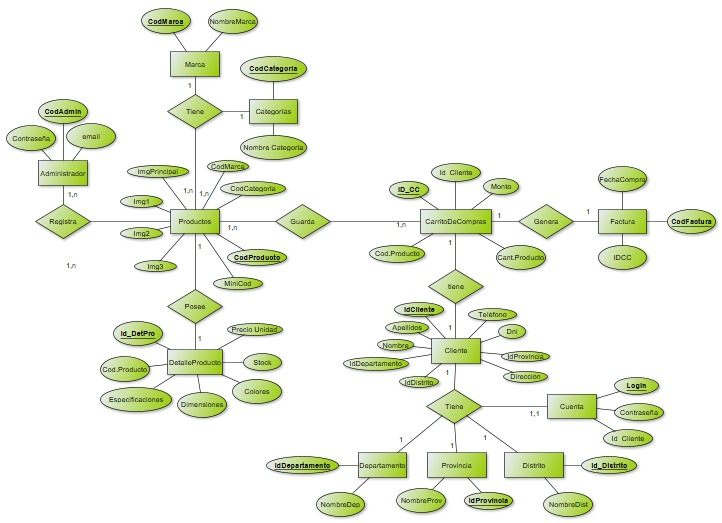
 **DER**

Figura 10 DER

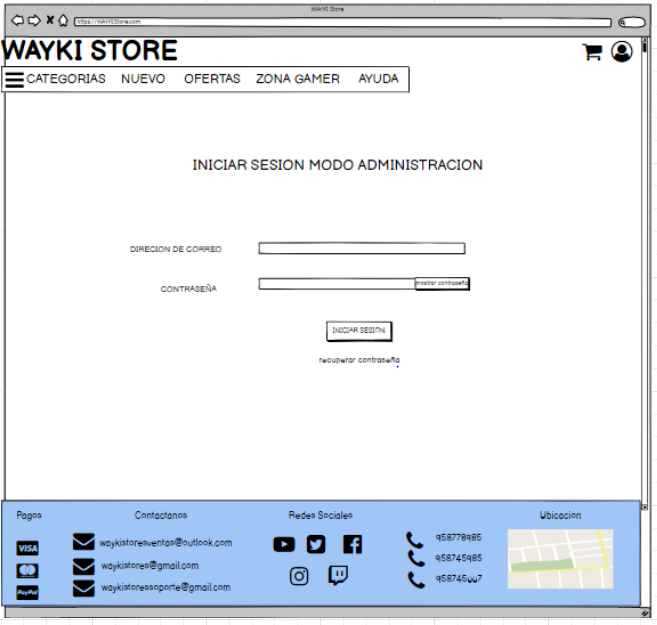
**PROTOTIPOS**

Figura 11 ingresar cuenta administrador

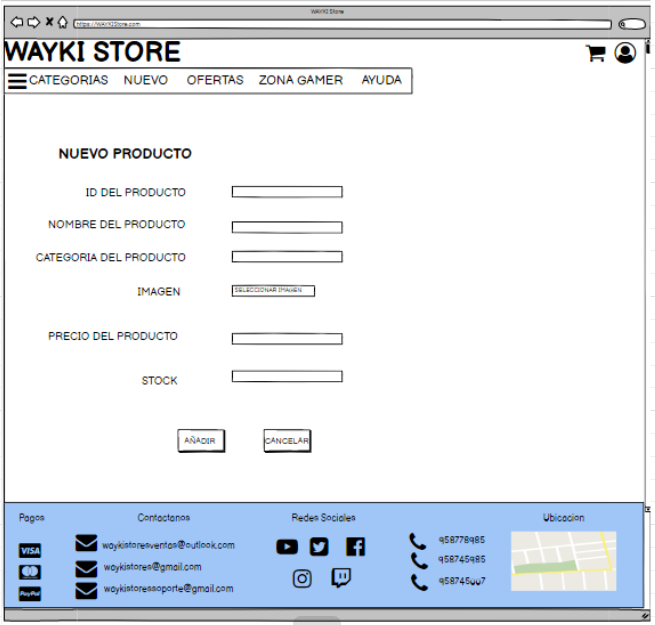


Figura 12 registrar producto

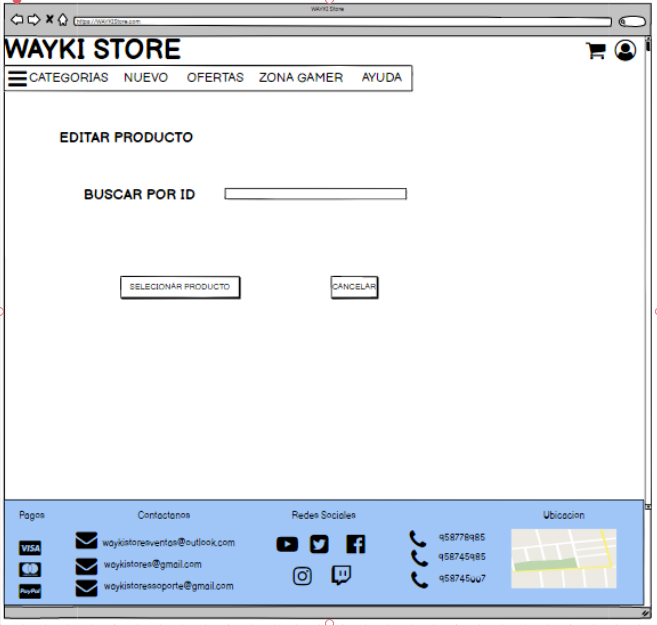


Figura 13 buscar producto a modificar

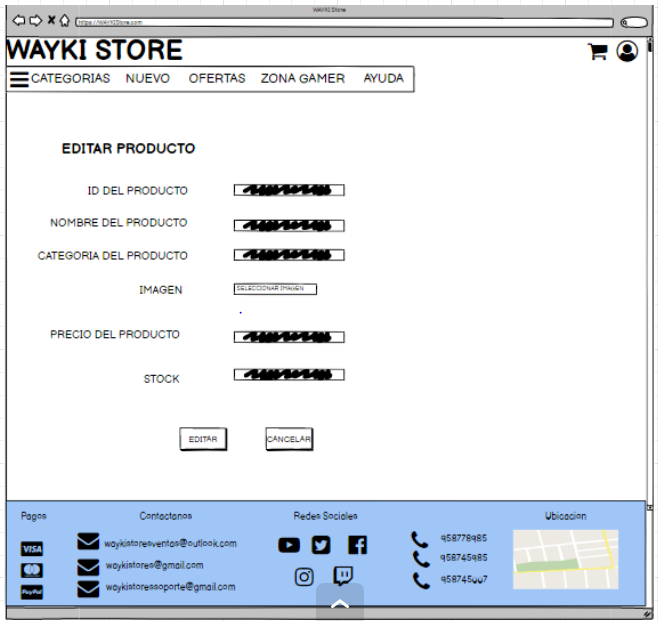
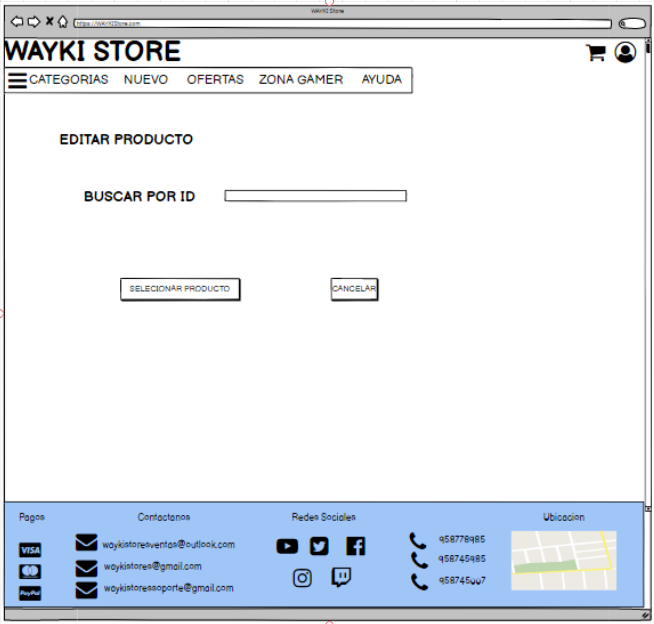


Figura 14 modificar producto



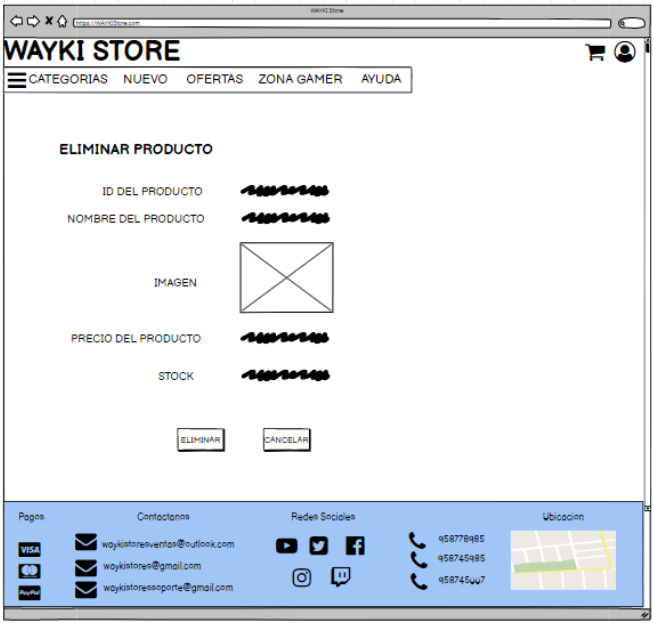


Figura 15 buscar producto a eliminar

Figura 16 eliminar producto

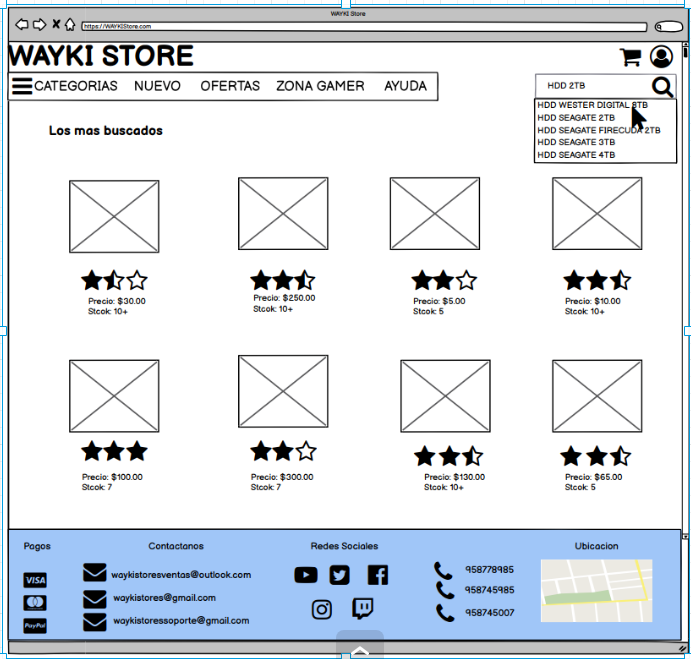
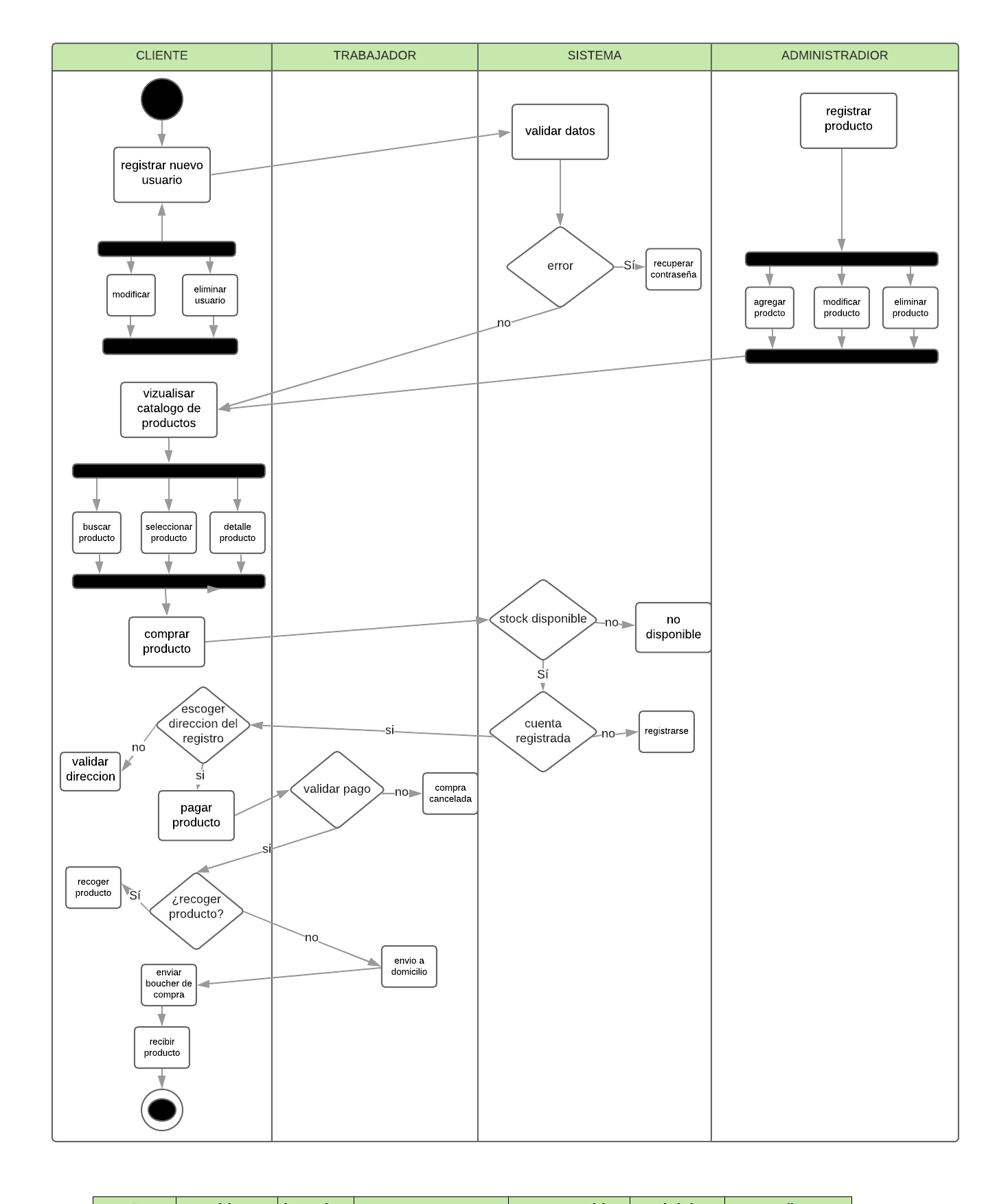


Figura 17 visualizar categoría de productos



Figura 18 mostrar información/detalle del producto

**DIAGRAMA DE CALLES**



**Figura 14 diagrama de calles**

Figura 19 DIAGRAMA DE CALLES

**VISTA ARQUITECTÒNICA DE HW Y SW**

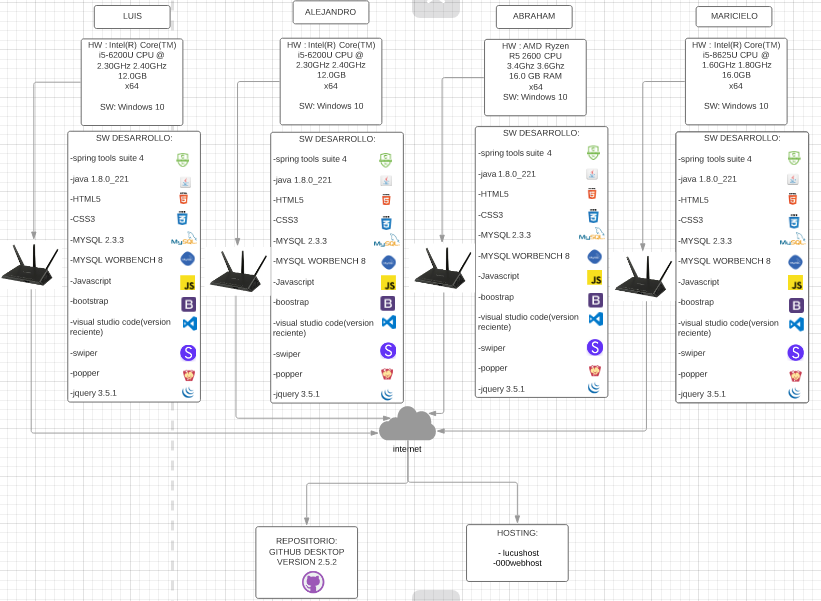


Figura 20 VISTA ARQUITECTÒNICADE HW Y SW

**PATRÒN DE AQUITECTURA EN CAPAS**

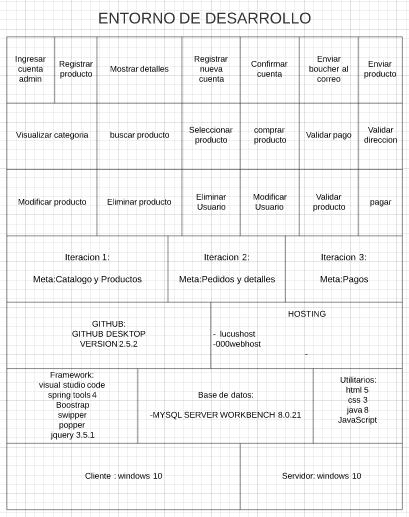


Figura 21 patrón de arquitectura en capas

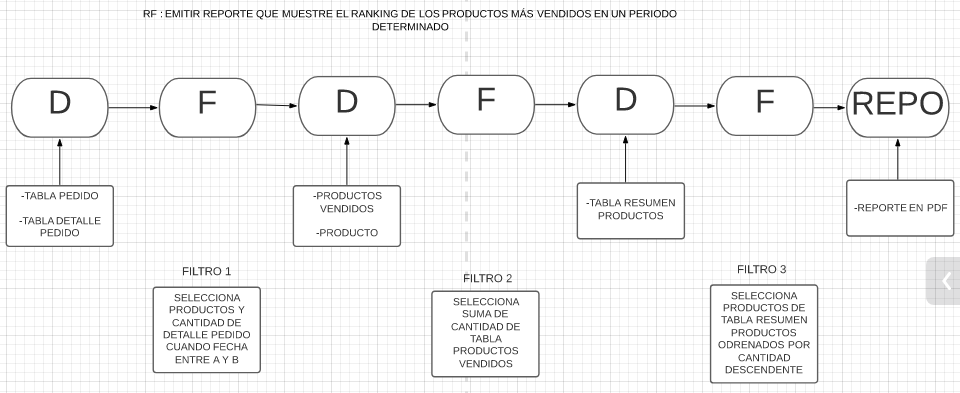
**PATRÒN ARQUITECTÓNICO (TUBERIA Y FILTRO)**

Figura 22 PATRÓN ARQUITECTÒNICO (TUBERIA Y FILTRO) REQUISITO 1

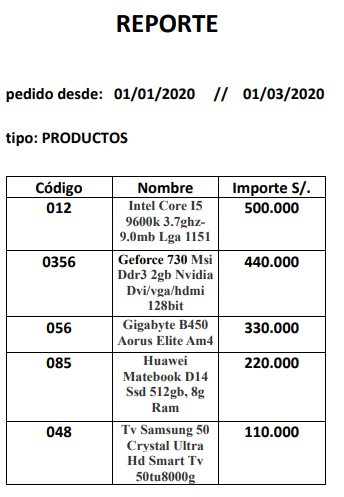


Figura 23 Prototipo Reporte – requisito 1

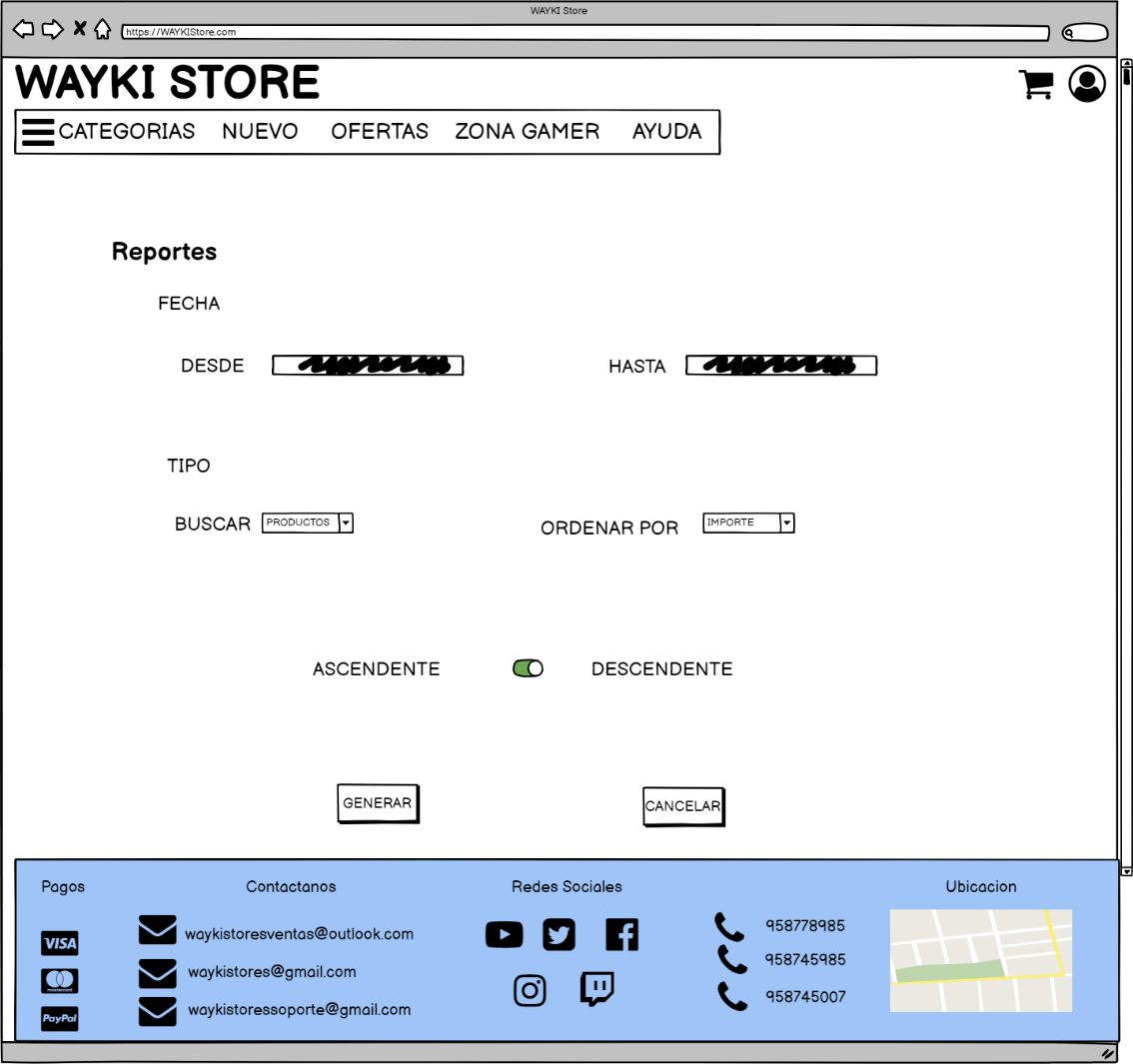


Figura 24 GUI Reporte – requisito 1

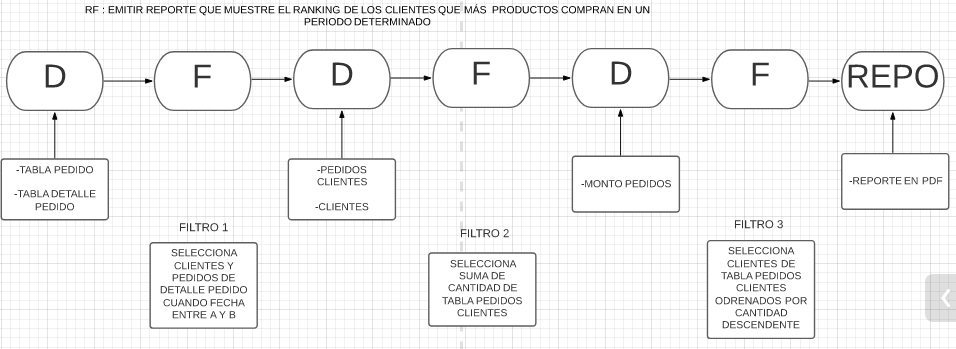


Figura 25 PATRÓN ARQUITECTÒNICO (TUBERIA Y FILTRO) REQUISITO 2

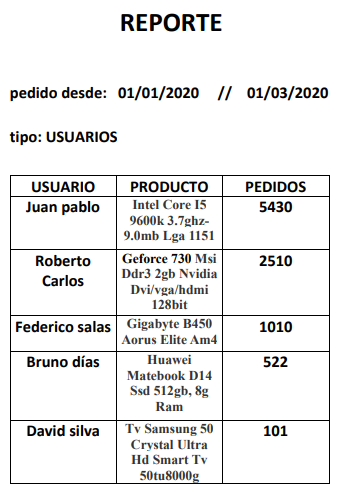


Figura 26 Prototipo Reporte – Requisito 2

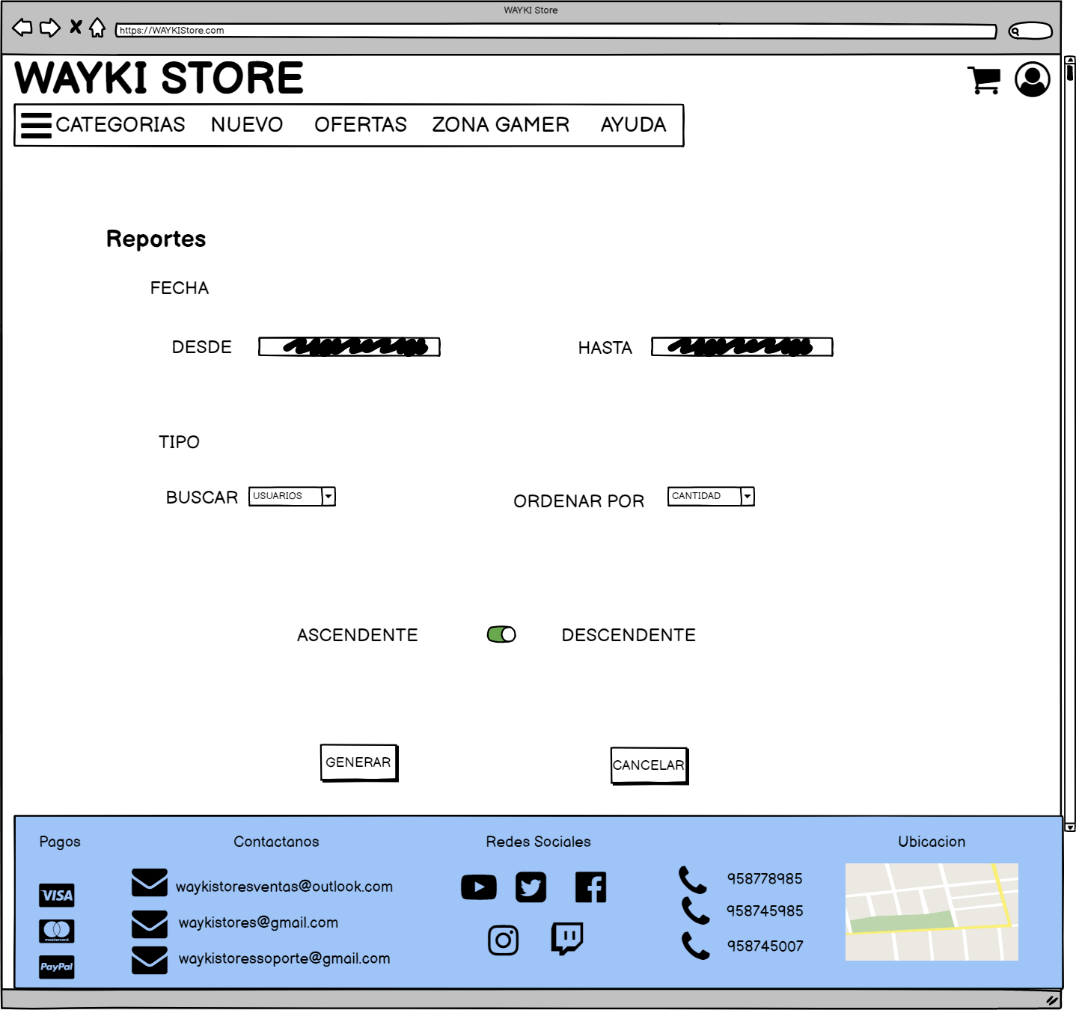
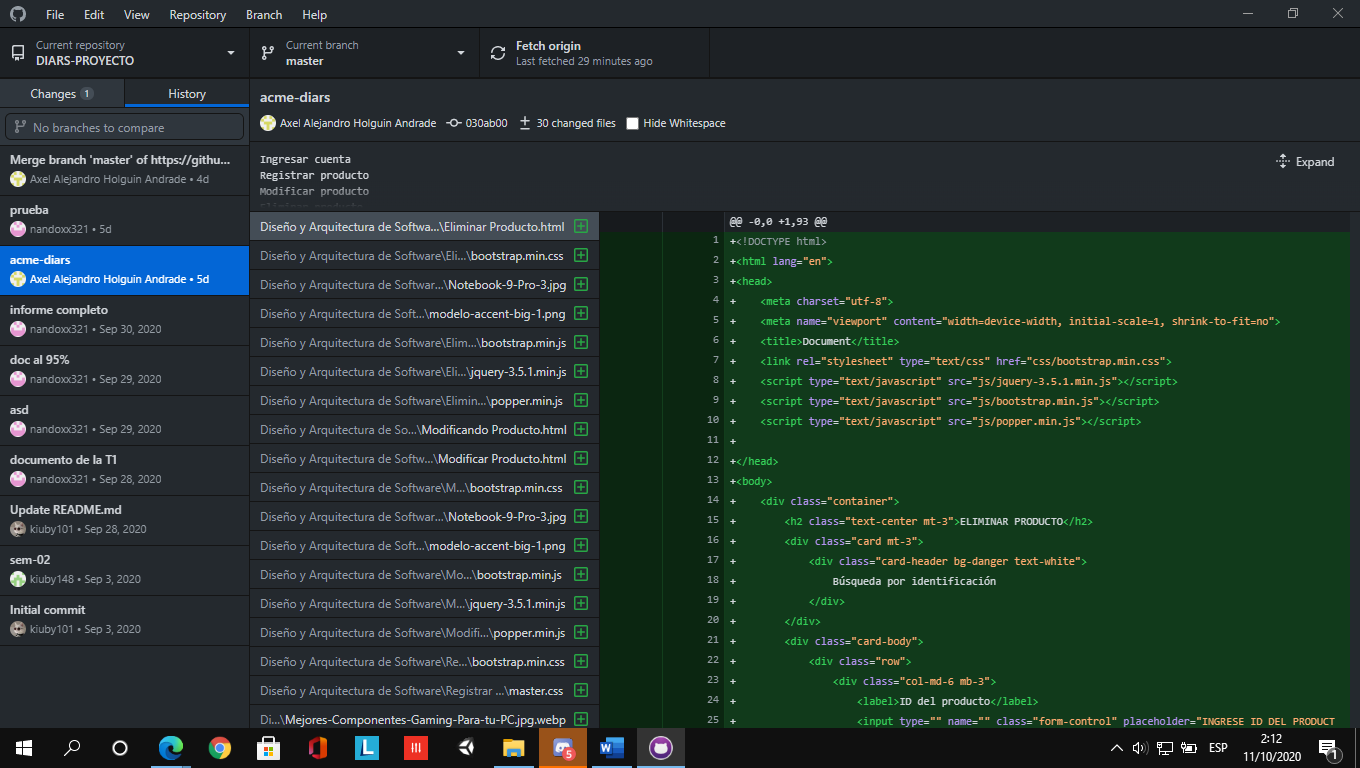


Figura 27 GUI Reporte – Requisito 2

**PATRONES GRASP:**

**PATRÒN ARQUITECTÒNICO (DE REPOSITORIO):**

CUS: REGISTRAR PRODUCTO

* VISTA:  
  (imagen del código desde la ventana del github)  
  ejemplo
* CONTROL:

* MODELO:
* PROCEDIMIENTO ALMACENADO

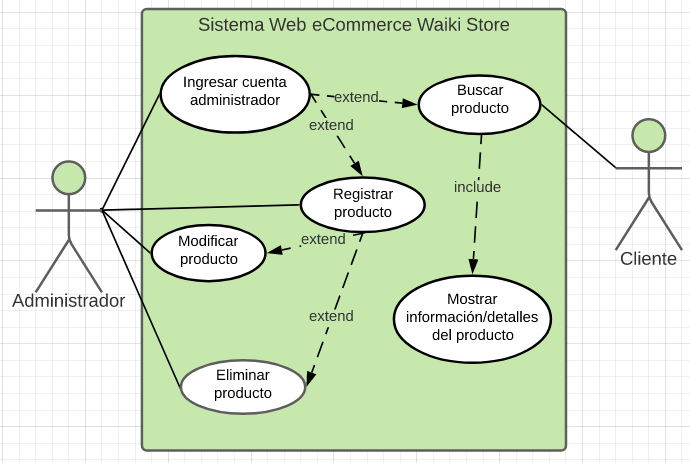
CUD: REGISTRAR USUARIO

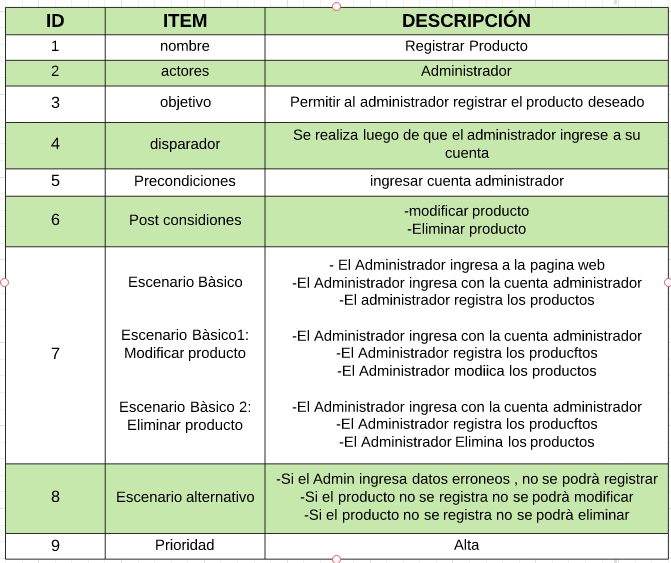
* VISTA:
* CONTROL:

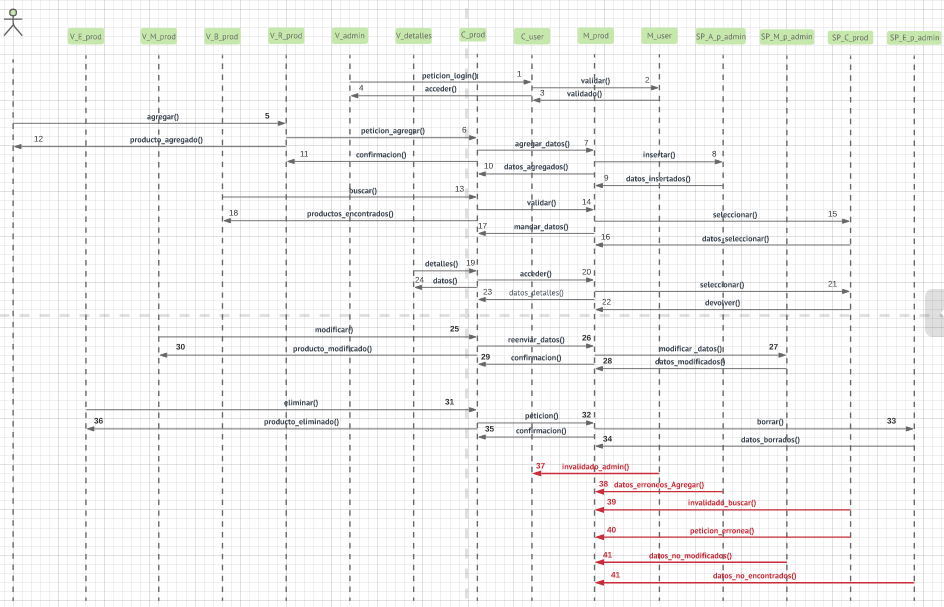
* MODELO:

* PROCEDIMIENTO ALMACENADO

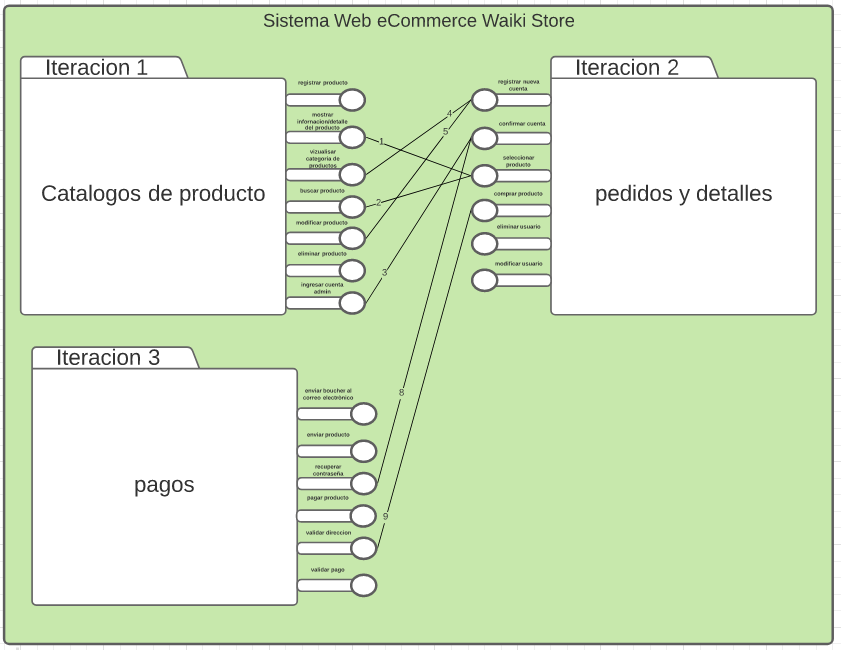
**CASO DE USO (ITERACIÒN 1):**



**ESPECIFICACION DEL CASO DE USO (ITERACIÒN 1):**

**DIAGRAMA DE SECUENCIA (ITERACIÒN 1):**

**DIAGRAMA DE SUBSISTEMA:**



# Capítulo 4

## CONCLUSIONES

* El sistema podrá mostrar, editar y eliminar los productos ya registrados
* Es fundamental la unión entre conocimientos teóricos de clase y prácticos al momento de programar
* La utilización de repositorios ayuda al grupo a realizar un trabajo en conjunto
* Se tiene pensado mejoras para las próximas iteraciones
* Es necesario investigar más para complementar los conocimientos adquiridos en clase

## RECOMENDACIONES

* Investigar más sobre el desarrollo y manejo de un sistema web en spring tools.
* Investigar el método de Login usando tanto Facebook o Gmail.
* Investigar el uso de pop-up para los sistemas web.
* Asignar y respetar los tiempos para el desarrollo del sistema.
* Investigar como implementar un carrito de compras para el sistema.

## REFERENCIAS

Castillo, L. (2012). *Introducción-Conociendo GitHub*. Obtenido de Conociendo GitHub: https://conociendogithub.readthedocs.io/en/latest/data/introduccion/

davidmarco. (2010). *Integraciòn de Eclipse y SpringSource Tool Suite*. Obtenido de davidmarco: http://www.davidmarco.es/articulo/integracion-de-eclipse-y-springsource-tool-suite#:~:text=SpringSource%20Tool%20Suite%20(STS)%20es,que%20STS%20proporciona%20se%20encuentran%3A&text=Herramientas%20de%20desarrollo%20para%20Spring%20Web%20Flow%20y%20Spring%

Fowler, M., & Scott, K. (1999). *UML gota a gota.* Obtenido de http://www.face.ubiobio.cl/~cvidal/modelamiento/libros/UMLgotaagota.pdf

Genbeta. (2012). *Sublime Text, un sofisticado editor de código multiplataforma*. Obtenido de Genbeta: https://www.genbeta.com/herramientas/sublime-text-un-sofisticado-editor-de-codigo-multiplataforma

Gisbert, E. (2018). *Lucidchart - Fundaciò Factor Humà*. Obtenido de FactorHumà: https://factorhuma.org/es/actualitat/2015-12-07-19-31-00/webs/13558-lucidchart

Larman, C. (2003). *UML y PATRONES.* Reino Unido: PERASON Education.

M. R. (s.f.). *¿Què es MySQL?-Definiciòn*. Obtenido de searchdatacenter: https://searchdatacenter.techtarget.com/es/definicion/MySQL

MX, Editorial Definición. (25 de 07 de 2014). *java*. Obtenido de definicìon: https://definicion.mx/?s=Java

Perez , j. D. (2018). *Què es HTML5*. Obtenido de OpenWebinars: https://openwebinars.net/blog/que-es-html5/

Pèrez , juliàn; Gardey, Ana. (2018). *Definicion de jQuery*. Obtenido de definicion.de: https://definicion.de/jquery/

Pèrez, j., & Merino, m. (2017). *definiciòn de css*. Obtenido de definicion.de: https://definicion.de/css/

Pèrez, juliàn; Gardey, Ana. (2018). *Definición de JavaScript*. Obtenido de definicion.de: https://definicion.de/javascript/

Pressman, R. S. (2010). *Ingeniería del software: un enfoque práctico Sèptima edición.* mèxico: University of Connecticut. Obtenido de http://artemisa.unicauca.edu.co/~cardila/Libro\_Pressman\_7.pdf

santamarìa, p. (2013). *Balsamiq mockup, una muy buena herramienta para esbozar futuras apps*. Obtenido de applesfera: https://www.applesfera.com/aplicaciones-os-x-1/balsamiq-mockup-una-muy-buen-herramienta-para-esbozar-tus-futuras-apps

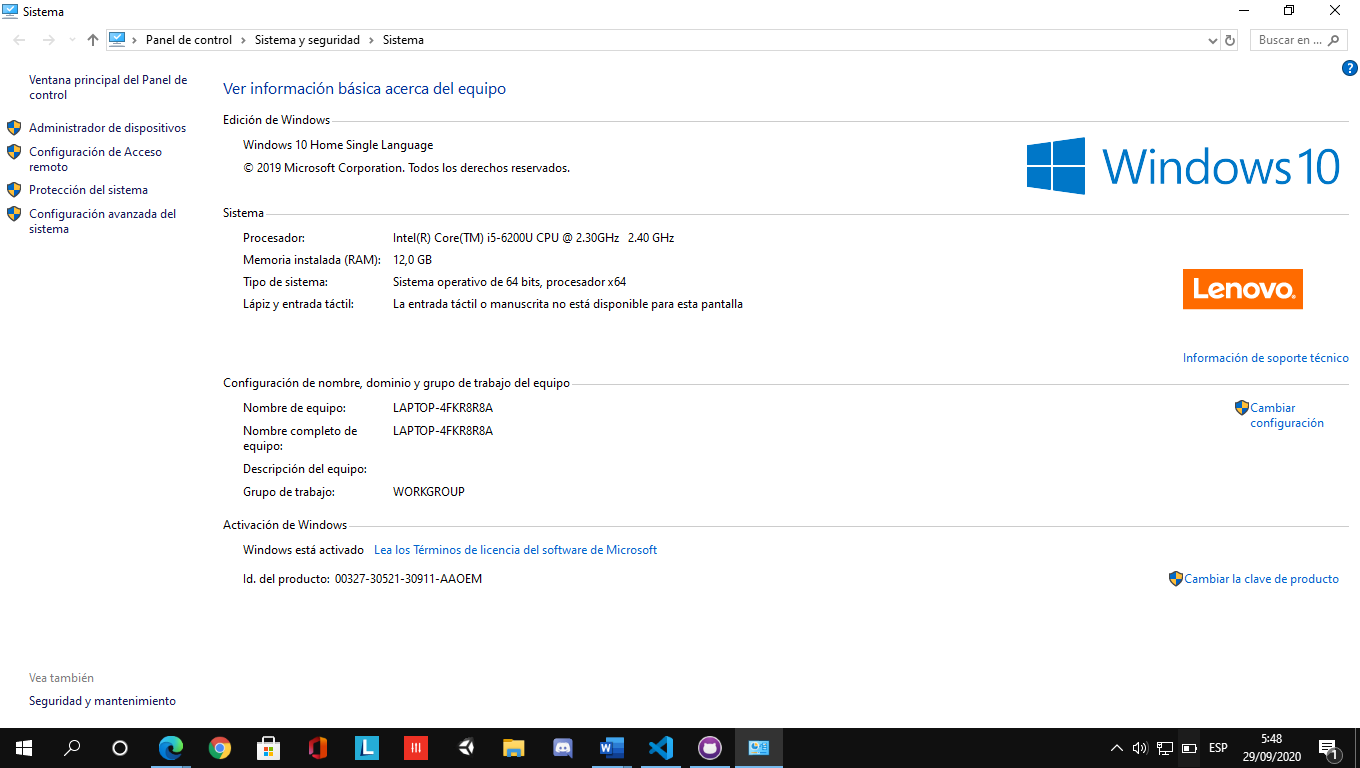
Solis, J. (2014). *¿què es bootstrap y còmo funciona en el diseño web?* Obtenido de arweb: https://www.arweb.com/blog/%C2%BFque-es-bootstrap-y-como-funciona-en-el-diseno-web/

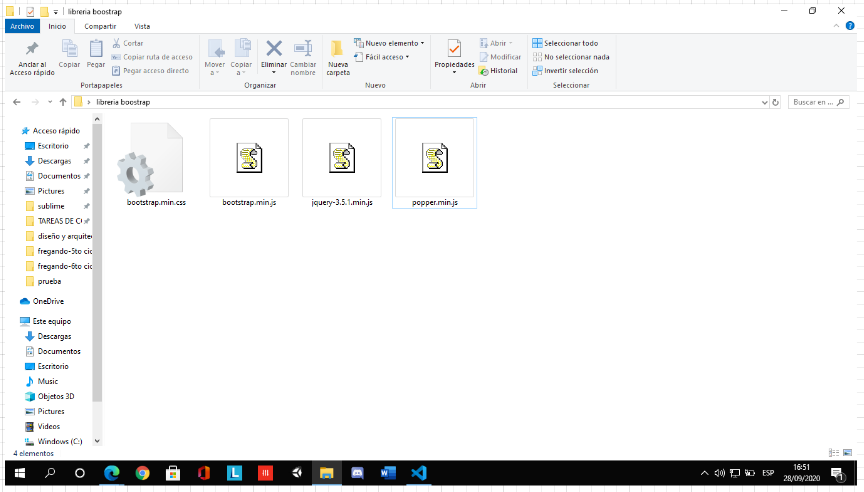
Sommerville, I. (2005). *Ingenierìa del software.* Obtenido de http://zeus.inf.ucv.cl/~bcrawford/AULA\_ICI\_3242/Ingenieria%20del%20Software%207ma.%20Ed.%20-%20Ian%20Sommerville.pdf

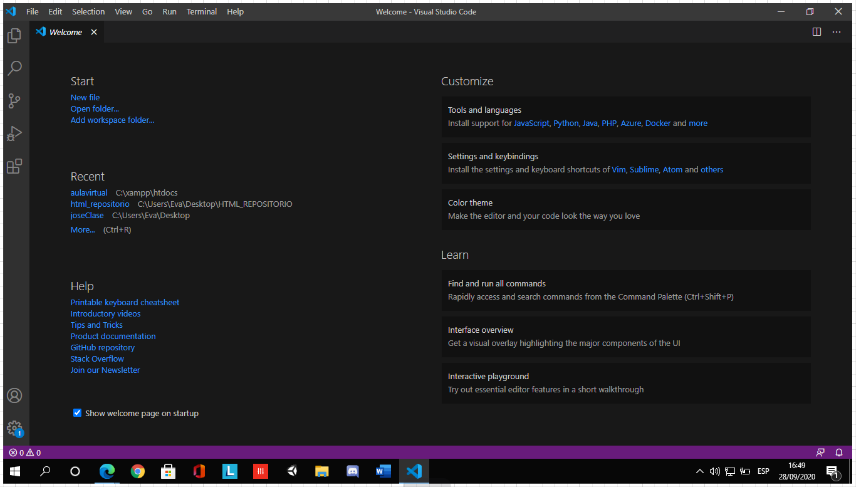
Sommerville, Ian. (2011). *Ingeniería de software- novena edición.* México: PEARSON EDUCACIÓN. Obtenido de http://artemisa.unicauca.edu.co/~cardila/Libro\_Sommerville\_9.pdf

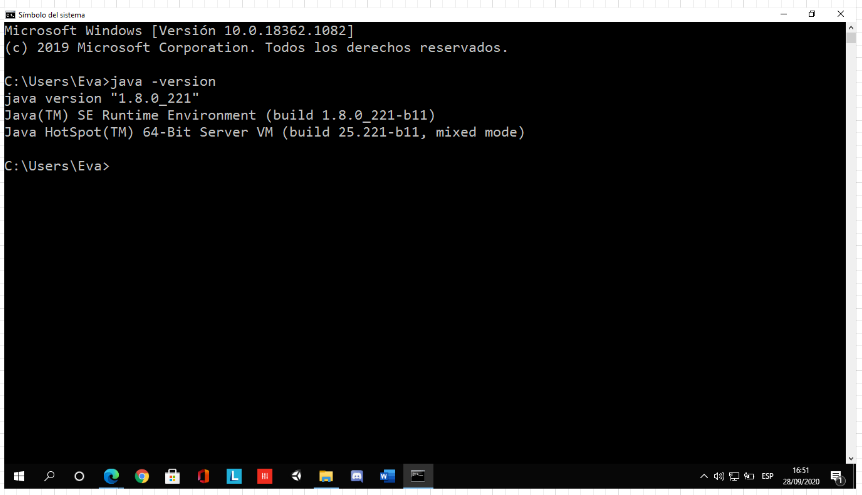
Sparks, G. (2000). *El modelo de proceso de negocio.* Obtenido de http://www.sparxsystems.com.ar/downloads/whitepapers/El\_Modelo\_de\_Proceso\_de\_Negocio.pdf

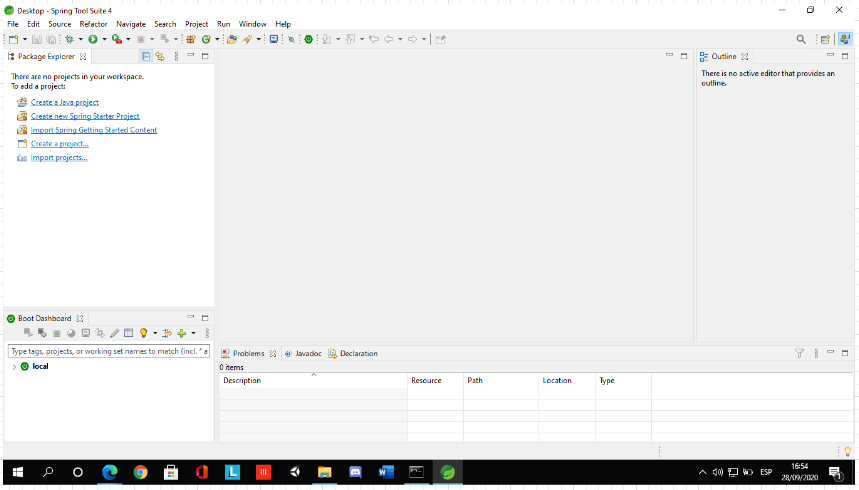
## ANEXO

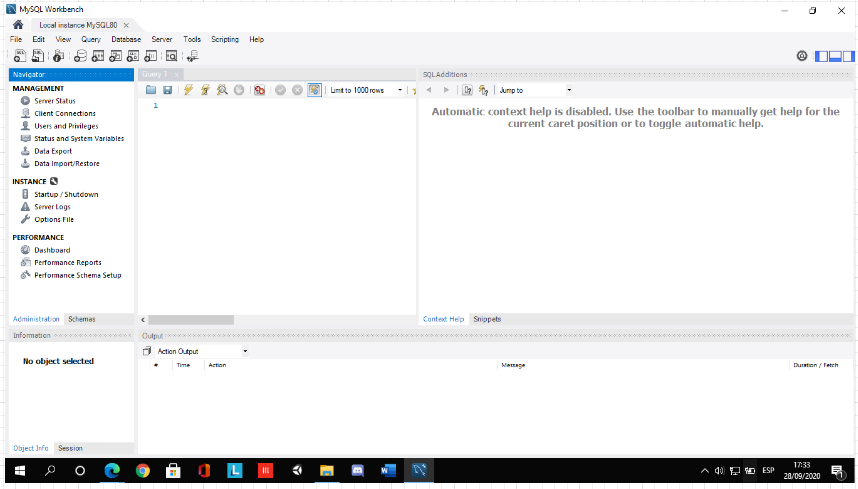
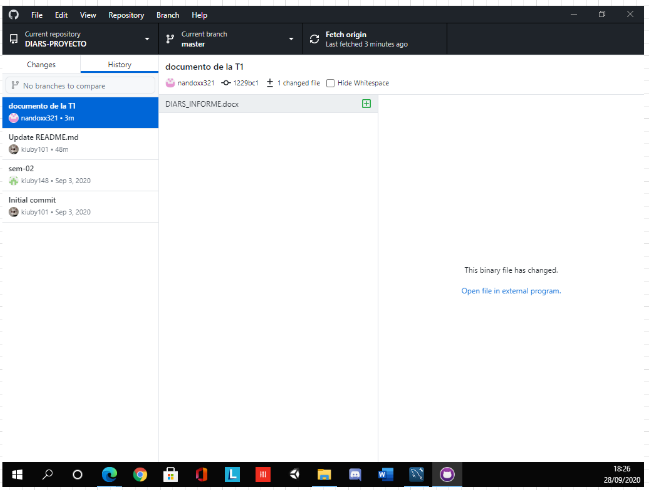
**SOFTWARE BASE LUIS LOLI:**



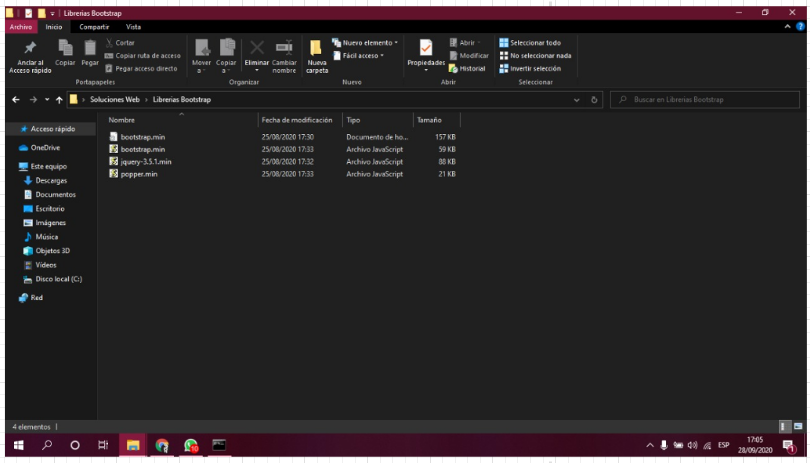
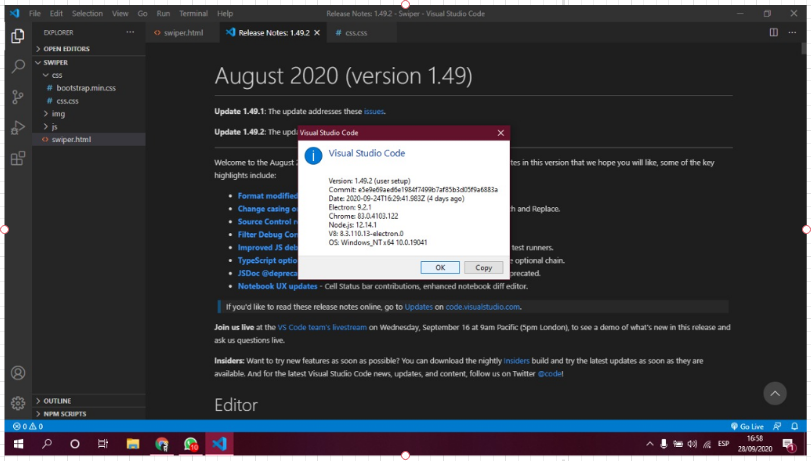
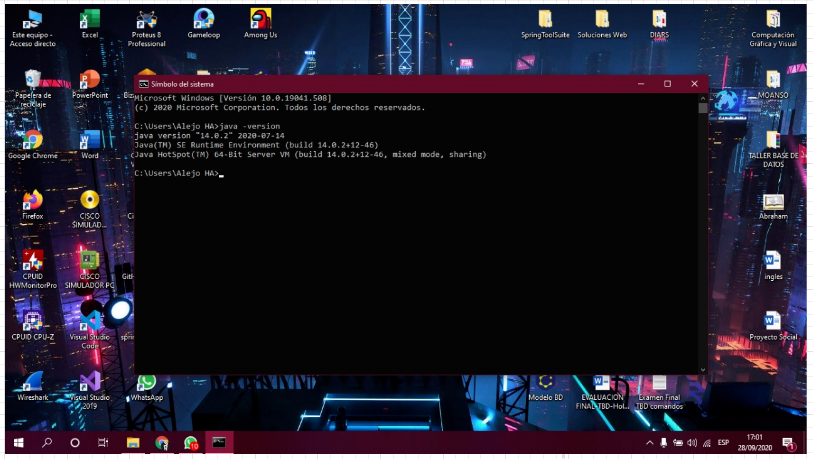
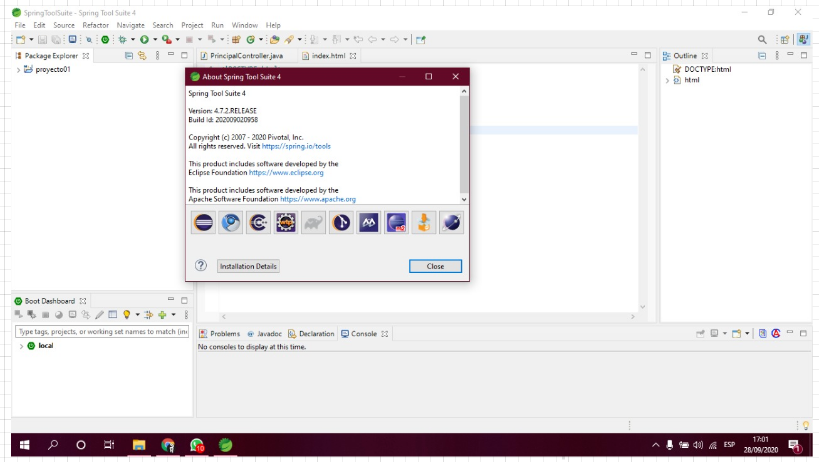


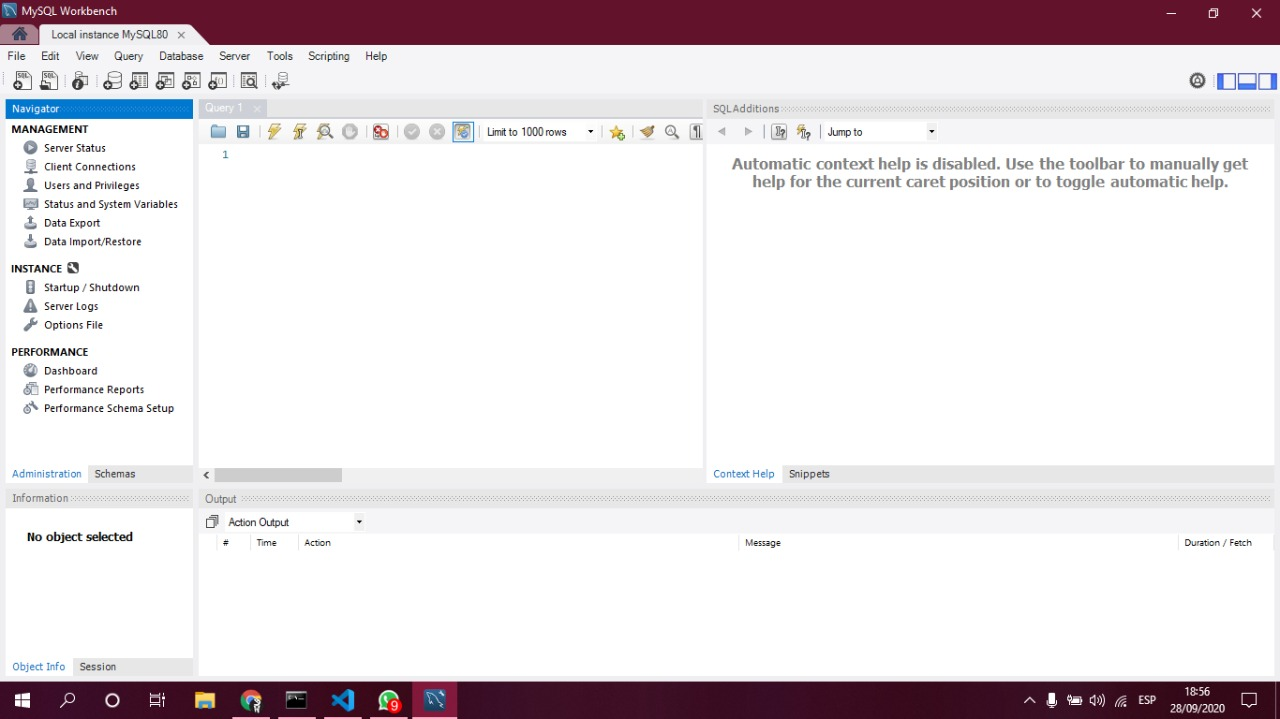


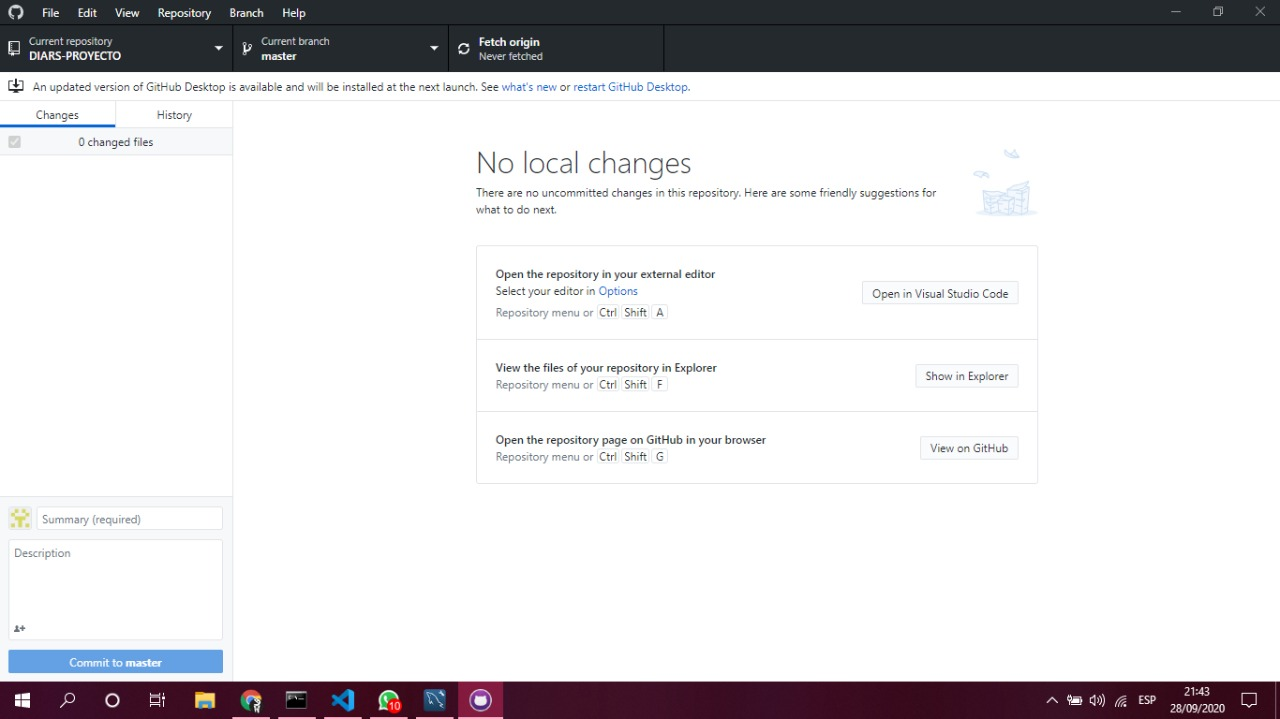


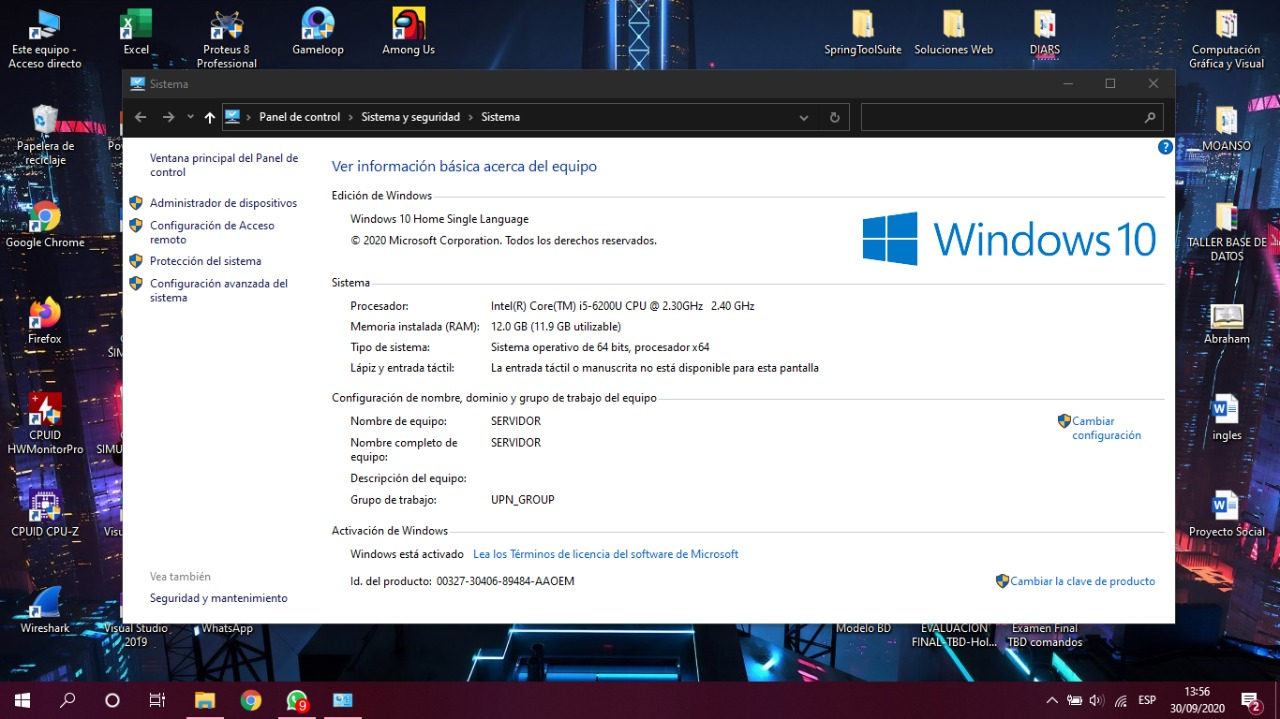
   
  


**SOFTWARE BASE ALEJANDRO HOLGUIN:**

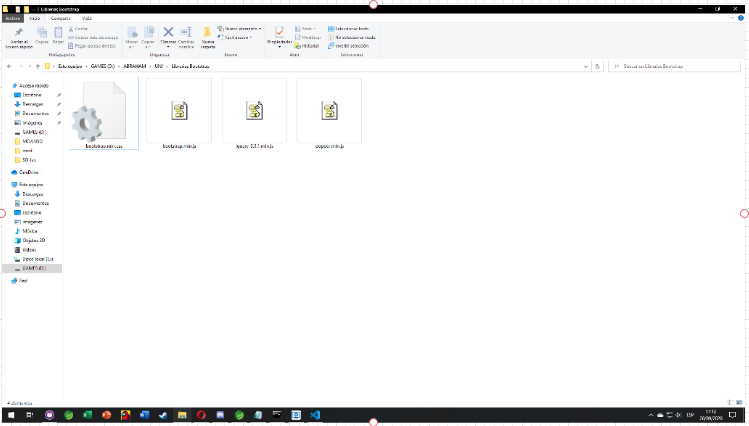
   
  
  
  
  
  
  
  


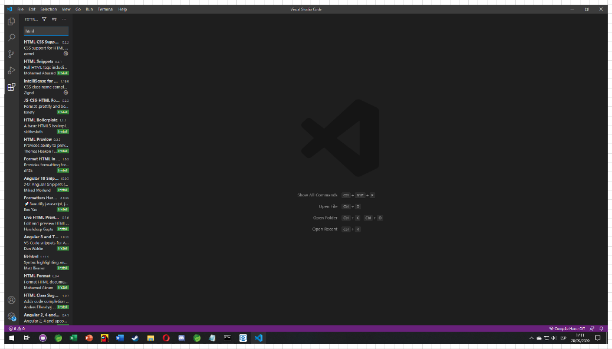


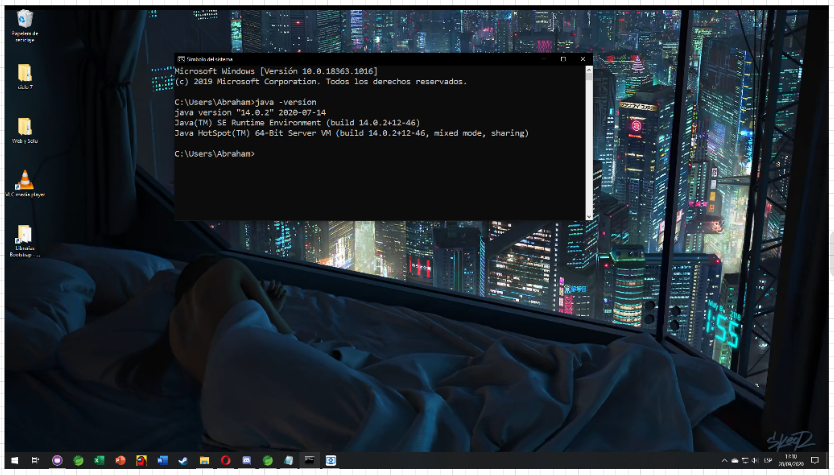


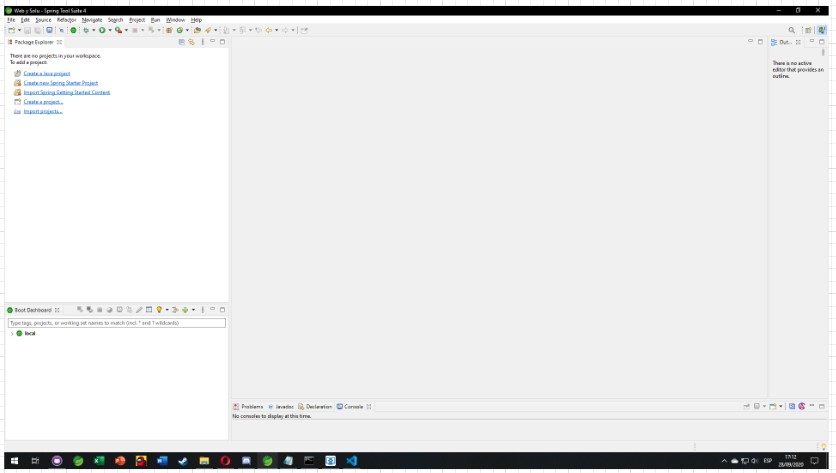


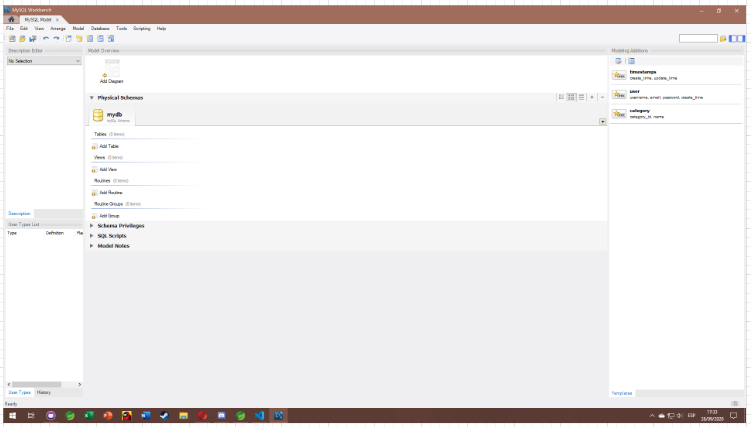
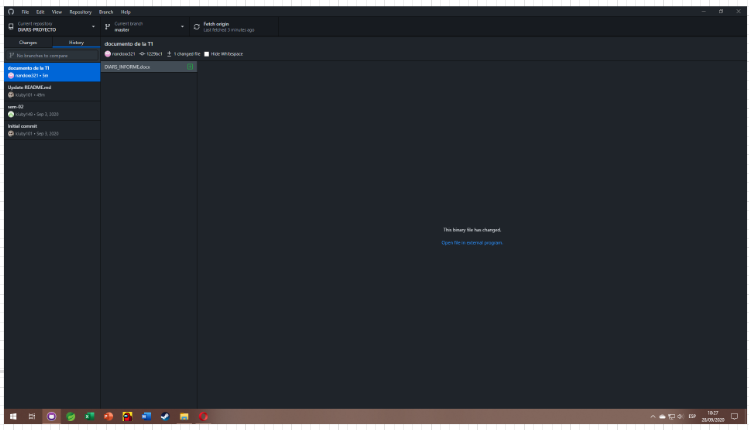
**SOFTWARE BASE ABRAHAM SANCHEZ:**

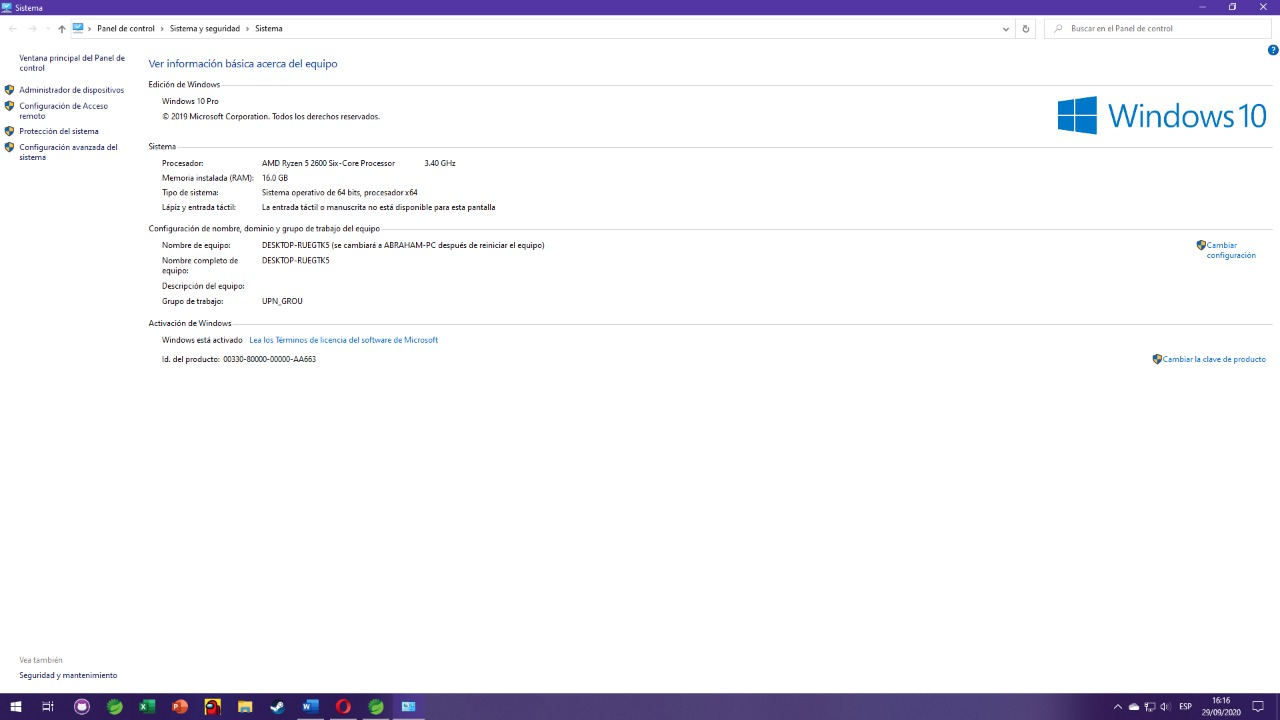




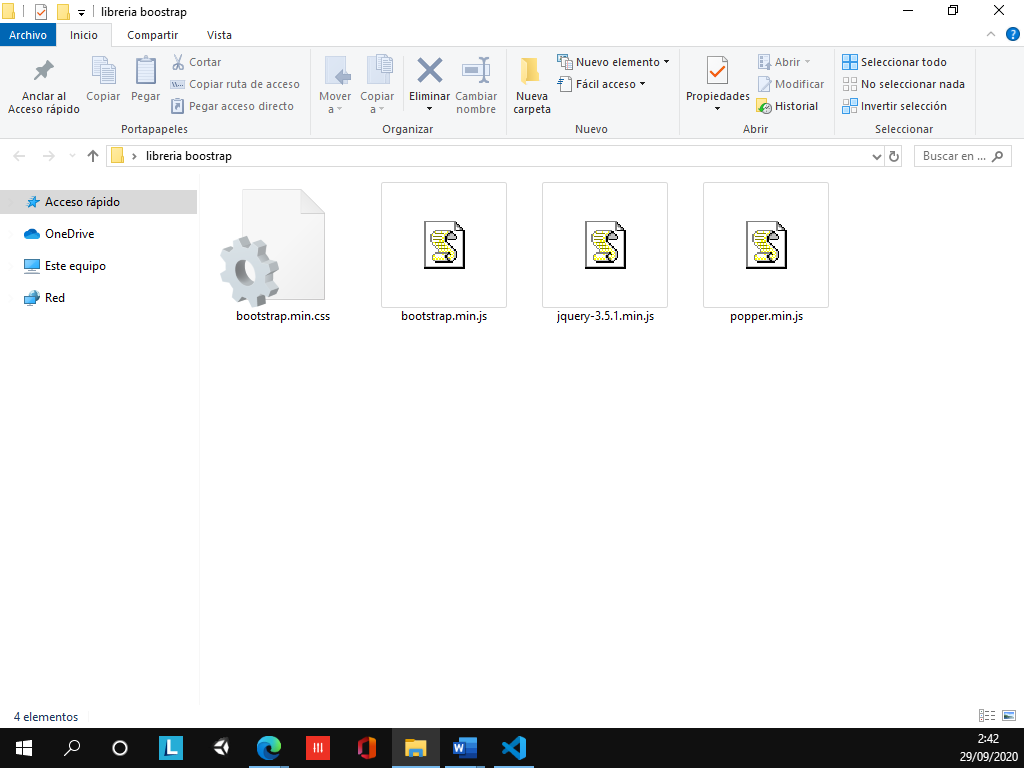


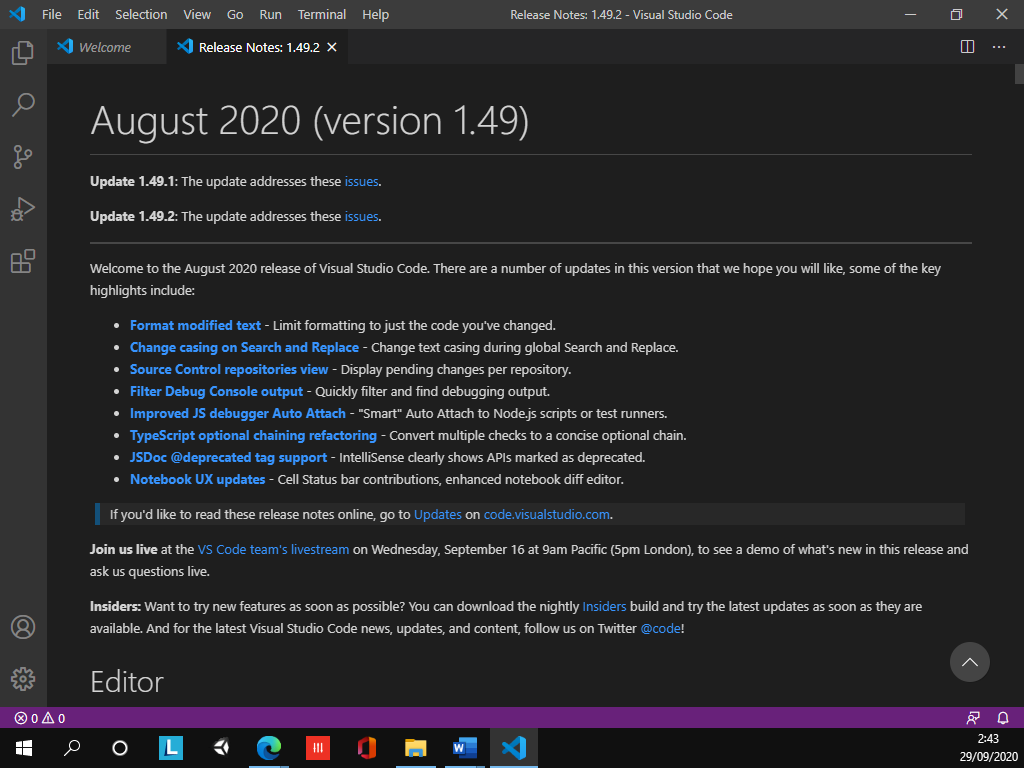


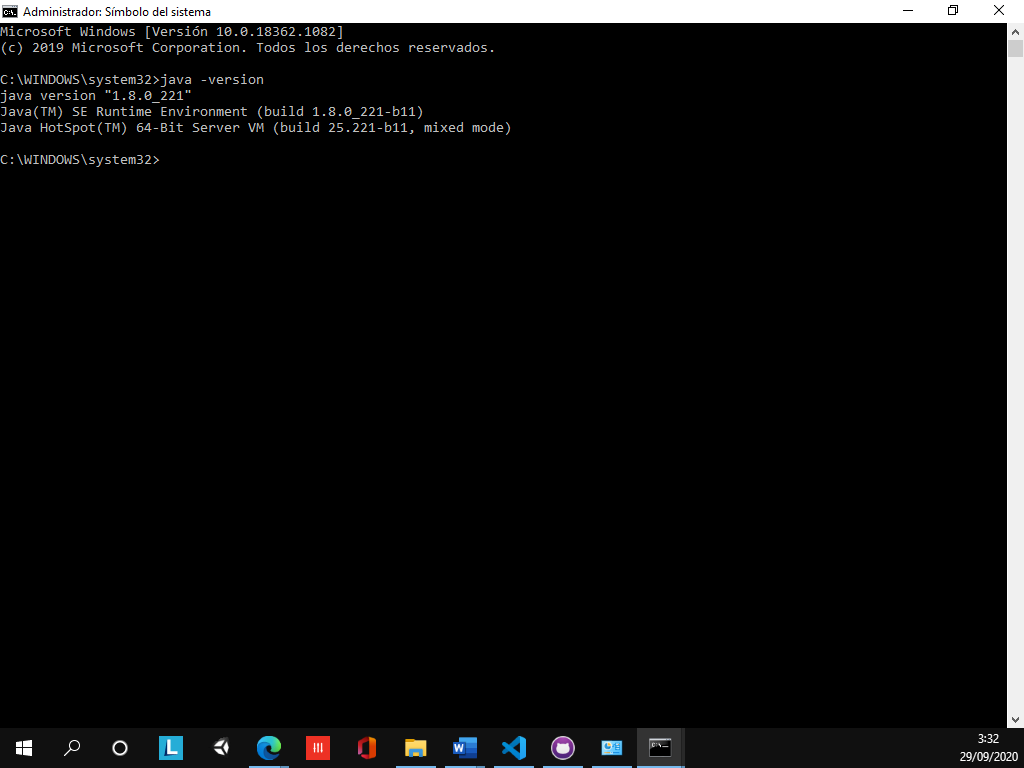
   


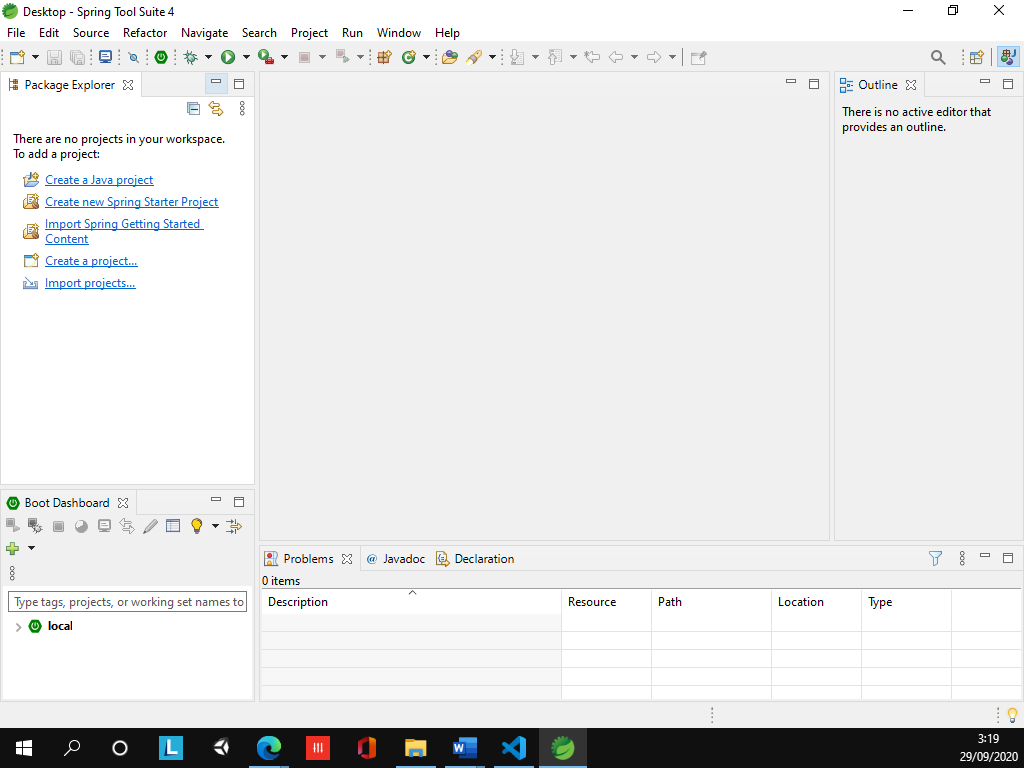


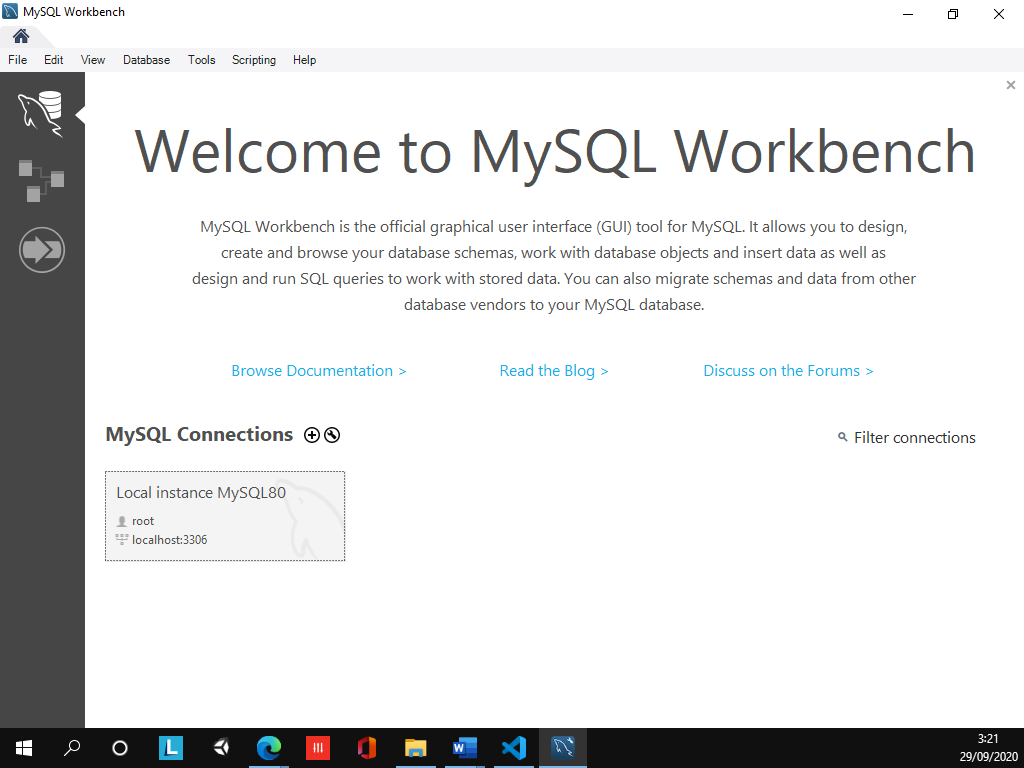
**SOFTWARE BASE MARICIELO DURÁN:**

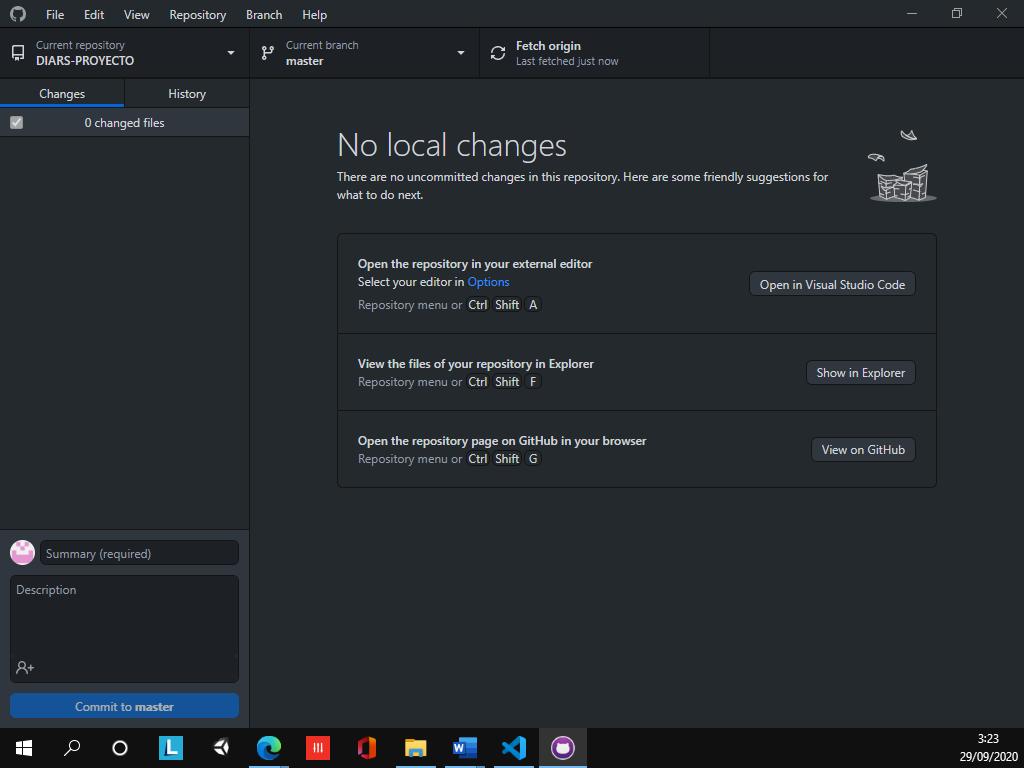












**IMÁGENES DE LOS CASOS DE USO (ITERACION 1)**

Imágenes del funcionamiento del sistema (iteración 1):