APLICACIÓN DE NOTAS – **APUNT.ES**

Aplicaciones Web – Otoño 2020

25 de noviembre de 2020

Equipo 3 – Integrantes

* Flores Rivera José Carlos
* Pérez López Bernardo
* Hernández Álvarez Víctor

# Contenido

[INTRODUCCIÓN 5](#_Toc57243412)

[1.- Marco Teórico 6](#_Toc57243413)

[1.1 Desglose del Problema por acciones 6](#_Toc57243414)

[1.2 - Listado y Descripción de Operaciones 8](#_Toc57243415)

[2.- Descripción del Modelo de Procesos y Metodologías 9](#_Toc57243416)

[2.1- Metodología de Desarrollo - V 9](#_Toc57243417)

[2.2- Metodología Web - OOHDM (Método de Diseño e Hipermedia Orientado a Objetos) 10](#_Toc57243418)

[2.3 – Modelo Lógico General del sistema 11](#_Toc57243419)

[2.4 – Modelo Lógico de la base de Datos 13](#_Toc57243420)

[2.5 – Descripción técnica de las herramientas 13](#_Toc57243421)

[2.5.1 – Herramientas de Modelado 13](#_Toc57243422)

[3.- Diagramas de Casos de Uso 15](#_Toc57243423)

[3.1. Internauta 15](#_Toc57243424)

[15](#_Toc57243425)

[3.2. Alumno 16](#_Toc57243426)

[16](#_Toc57243427)

[3.3. Profesor 17](#_Toc57243428)

[3.4. Administrador 19](#_Toc57243429)

[IV. Diagramas de Navegación 21](#_Toc57243430)

[4.1. Internauta 21](#_Toc57243431)

[4.2. Alumno 22](#_Toc57243432)

[4.3. Profesor 23](#_Toc57243433)

[4.4. Administrador 24](#_Toc57243434)

[5.- Mockup 25](#_Toc57243435)

[5.1. Internauta 25](#_Toc57243436)

[5.2. Alumno 28](#_Toc57243437)

[5.3. Profesor 29](#_Toc57243438)

[5.4. Administrador 31](#_Toc57243439)

[6.- DESARROLLO 34](#_Toc57243440)

[7.- PRUEBAS 36](#_Toc57243441)

[7.1 -Creación de la base de datos 38](#_Toc57243442)

[7.4 - Modelado de las tablas. 38](#_Toc57243443)

[8.- REFERENCIAS 42](#_Toc57243444)

# INTRODUCCIÓN

El presente documento muestra la conceptualización y puesta en marcha del proyecto “APUNT.ES – Aplicación de Notas Escolares” perteneciente a la materia de Aplicaciones Web.

Se abarcarán conceptos de la definición de proyecto, pasando por la esquematización y muestreo de diagramas lógicos y de interfaz, además de la explicación de la metodología que se llevó acabo a lo largo del proyecto y las fases de la implementación logradas.

Para terminar, una muestra de las pruebas que se hicieron para comprobar que la aplicación sea de total fiabilidad.

# 1.- Marco Teórico

Se realizará un **sistema**​ **de apuntes electrónicos**, ​ donde los profesores registrados, podrán subir sus apuntes para compartir, los alumnos podrán revisar dichos apuntes y visualizarlos de forma dinámica como si fuese un libro. Los usuarios no registrados, solo podrán visualizar el resumen de cada apunte.

El sistema, además debe permitir:

* Registrar materia, unidad y tema.
* Se podrá consultar: Materia, Unidad y Tema
* Permitir Imprimir el apunte citado
* Un administrador debe permitir identificar o no a un usuario como Profesor.

## 1.1 Desglose del Problema por acciones

A partir del problema planteado, se puede desglosar el sistema en los usuarios ya que, debido al problema, no son muchos los mismos que podrán interactuar con el sistema, además de poder citar las funciones que podrá realizar cada uno. Comenzaremos describiendo los Usuarios:

|  |  |  |
| --- | --- | --- |
|  | Tabla de Usuarios | |
| Internauta (Usuario no registrado) |  | Usuario visitante de la web |
| Alumno (Usuario Registrado) |  | Usuario registrado en la aplicación |
| Profesor (Registrado y Concedido por Administrador) |  | Usuario que puede subir documentos |
| Administrador |  | Controla los usuarios registrados y el contenido. |

*Tabla de Usuarios que interactúan con el sistema*

Una vez definido los usuarios, podemos explicar que acciones realiza cada uno y el sistema de forma independiente.

Sistema:

* Permite registro de usuarios.
* Permite carga de archivos (Word / pdf / PowerPoint) así como los controles básicos para su actualización y eliminación
* Permite descarga e impresión de documentos.
* Visualiza los documentos como libros reales (animaciones)

Internauta:

* Poder visualizar el resumen de cada apunte.

Alumno:

* Podrá hacer una búsqueda de cada documento por Materia y/o Tema.
* Este usuario podrá visualizar e imprimir el apunte completo y los datos adjuntos al mismo.

Profesor:

* El profesor podrá subir los apuntes correspondientes a la materia solicitada, junto a su Unidad, además de colocar el tema correspondiente.

Administrador:

* El administrador controla qué usuarios son profesores y cuáles no.
* Además, verifica el sistema en busca de errores.

## 1.2 - Listado y Descripción de Operaciones

Las acciones se describirán conforme a las mencionadas en el apartado anterior, descrito como “Sistema”:

* **Acceso:**
* Cada usuario deberá poder ingresar con su usuario y contraseña correspondientes.
* **Registro:**
* El sistema permitirá el registro a cada usuario, eligiendo ser estudiante, y deberá corroborar que existe dentro de la institución.
* Si un usuario necesita registrarse como profesor, deberá llenar los campos propios a un docente, y se le notificará al administrador de la plataforma para poder ser dado de alta.
* **Inserción de información:**
* Cada profesor podrá subir sus apuntes en formato Word, PDF o PowerPoint y llenar los campos de acuerdo al tema del escrito, como Tema, Unidad y Materia.
* **Actualización de Información:**
* Cada usuario podrá actualizar la información de su perfil.
* Cada profesor podrá actualizar los apuntes y/o su información si lo considera necesario.
* **Dar de baja la Información:**
* Cada profesor podrá dar de baja sus apuntes si es pertinente.
* **Búsqueda de Información (Notas y Apuntes):**
* Cada usuario podrá buscar apuntes bajo tres categorías
* Tema: Se mostrarán todos los apuntes que coincidan con el título proporcionado.
* Materia: El sistema visualizará todos los archivos que se ajusten a la materia descrita.
* Profesor: Se mostrarán todos los apuntes publicados por ese profesor.
* Solo los usuarios registrados podrán visualizar el documento completo.
* **Previsualización principal del Contenido descrito:**
* Los apuntes se mostrarán de forma dinámica como un libro y simularán el movimiento de hoja al pasar página.

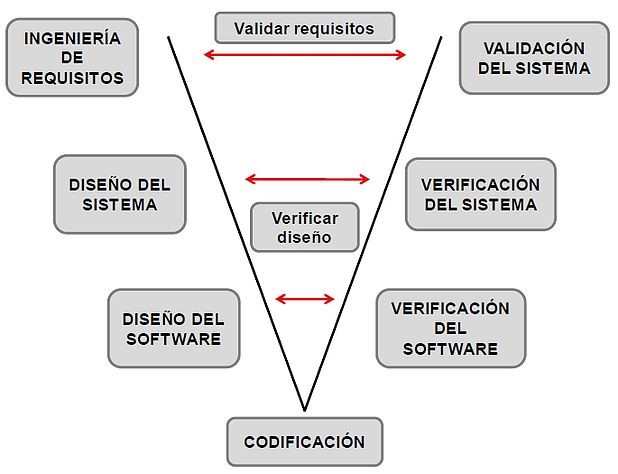
**- Impresión de Información:**

* Los usuarios y profesores pueden imprimir los apuntes y/o guardarlos en su dispositivo si lo consideran necesario.

# 2.- Descripción del Modelo de Procesos y Metodologías

## 2.1- Metodología de Desarrollo - V

La metodología de desarrollo de software a aplicar en el proyecto es el “modelo en V” que nos permite tener un temprano diseño e implementación del proyecto, lo que nos permite después, trabajar en la fase de pruebas en cada nivel con un periodo considerable de tiempo para poder detectar errores que puedan surgir en la aplicación.



*Diagrama del Modelo en “V”*

La principal estructura del modelo, la “V” significa Verificación y Validación, esto permite que en cada fase desde la toma de requisitos hasta el diseño del software tenga un homólogo en su fase de pruebas. Por ello, cuando vamos hacia abajo (el proceso habitual en cascada) y llegamos a la fase de codificación, se nos permite verificar cada parte correspondiente del diseño del sistema para poder finalmente validarlo.

## 2.2- Metodología Web - OOHDM (Método de Diseño e Hipermedia Orientado a Objetos)

Es una metodología basada en el paradigma OO la cual nos muestra una descripción precisa de los elementos que la aplicación poseerá. Este método toma como punto de partida el modelo de clases obtenido durante la primera fase del desarrollo de software denominado modelo conceptual, además permite modelar aplicaciones de grandes tamaños o con grandes volúmenes de información y pueden ser usados en diversos tipos de aplicaciones navegables, sitios Web, sistemas de información o presentaciones multimedia.

**Representación gráfica mediante:**

Diagrama de clase, diagrama navegacional, diagrama de configuración ADV

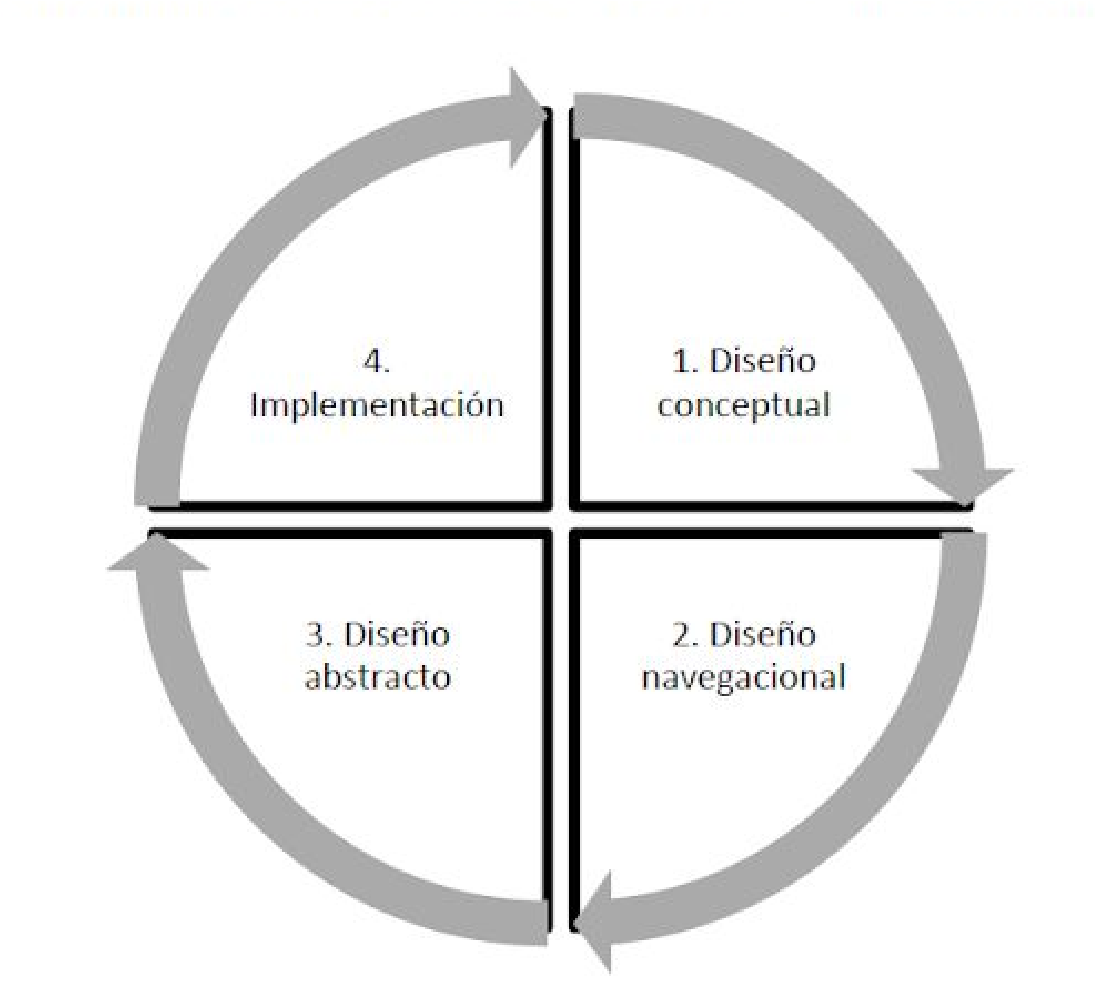
**Notación:**

UML y ADVs

**Herramientas:**

OOHDM-web y plugin para MagicDraw

**FASES DEL MODELADO OOHDM**



*Diagrama de fases del modelo OOHDM*

Ciclo de vida:

Fase Común:

* Obtención de requerimientos (Casos de Uso)

Entregables del modelo OOHDM

* Modelo conceptual -> (Diagrama de clases)
* Diseño navegacional -> (Diseño Navegacional)
* Diseño Abstracto -> (Prototipo)
* Implementación -> (Aplicación WEB)

## 2.3 – Modelo Lógico General del sistema

El modelo general del sistema se comprende de las funciones básicas que podrá realizar el usuario, no de hacia donde se debe dirigir.

Para comprender esto, a continuación, se mostrarán los diagramas lógicos usados para el desarrollo del sistema.

Para este diagrama se mostraron las principales funciones de la aplicación, en el que se puede ver que primero, cada usuario debe de pasar un registro o un inicio satisfactorio en el sistema.

Después, se procederá a mostrar la pantalla de home, en el caso de que sea un estudiante o profesor, se mostrará el dashboard correspondiente si así se desea.

El usuario puede ver los apuntes que se encuentran actualmente en el sistema, para ello debe dirigirse a la pantalla correspondiente y seleccionar un apunte, en caso de que el usuario esté registrado, podrá ver el apunte completo, caso contrario, será redirigido a la pantalla de registro / inicio de sesión.

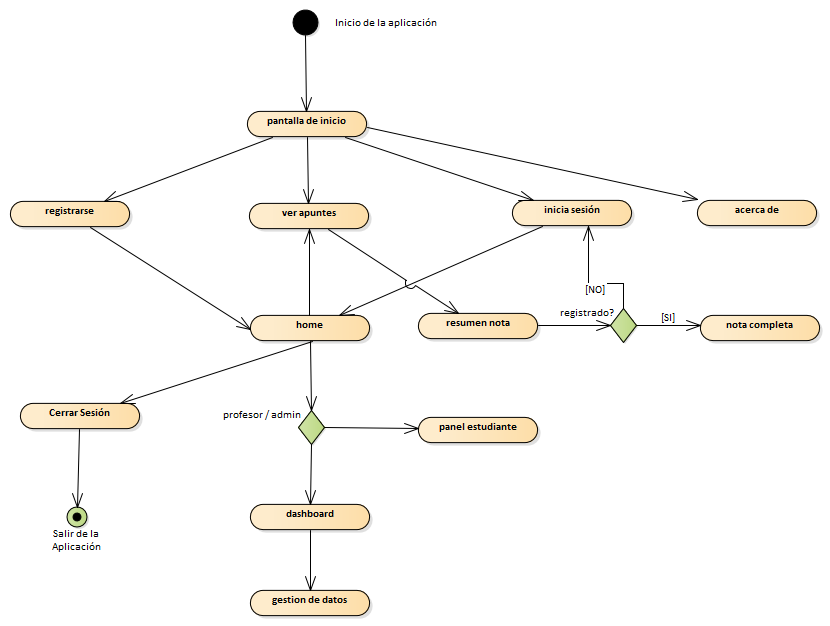


Figura 2.3 - 1 | Mapa de comportamiento general del sistema.

## 2.4 – Modelo Lógico de la base de Datos

Para el modelado de la base de datos se definieron 3 clases principales, la clase usuario, la clase apunte, y una clase intermedia llamada “libroApunte” que fungirá como conector entre la relación de usuario y apunte.

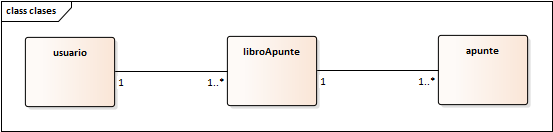


Figura 3.2.1 – 1 | Diagrama de Clases del Proyecto.

La clase de libro apuntes se usará mayormente a la hora de gestionar los profesores, ya que ellos tienen libros relacionas, un usuario registrado como alumno puede coexistir con el apunte sin tener una relación directa con el mismo.

## 2.5 – Descripción técnica de las herramientas

En esta sección se describirán las herramientas de modelado a utilizar, así como los lenguajes de programación que se usarán para implementar el proyecto.

### 2.5.1 – Herramientas de Modelado

El lenguaje de modelado primordial para este proyecto fue UML, que sirvió como lenguaje necesario para representar los diagramas de los cuales se basa el proyecto.

Las herramientas que se usaron para representar los diagramas mediante UML fueron:

* Enterprise Architech
* Magic Draw
* Star UML

Para implementar los diagramas realizados con los programas anteriores, y al ser una aplicación web, el proyecto se decidió realizar en los siguientes lenguajes:

Para la parte del Front-End, o la vista del usuario:

* HTML
* CSS
* Javascript

Para la parte del Backend, el proyecto fue entaramente realizado con:

* PHP

La base de datos fue realizada en Mysql / MariaDB, mediante el lenguaje estándar para bases de datos:

* SQL

# 3.- Diagramas de Casos de Uso

## 3.1. Internauta

## 

|  |  |
| --- | --- |
| Caso de Uso: **“buscando apuntes”** | |
| Actor: Internauta | |
| **Curso Normal** | **Alternativas** |
| 1) El usuario accede a la aplicación sin registro |  |
| 2) Ingresa el apunte a buscar |  |
| 3) El sistema obtiene los apuntes |  |
| 4) Se muestran los apuntes recolectados | 4.1 En caso de no obtener resultados, el sistema mostrará el mensaje correspondiente. |

|  |  |
| --- | --- |
| Caso de Uso: “**ver resumen de apuntes”** | |
| Actor: Internauta | |
| **Curso Normal** | **Alternativas** |
| 1) El usuario selecciona el apunte deseado |  |
| 2) Se muestra información y un resumen del apunte | 2.1 Si el usuario desea ver el documento completo, saltará una ventana de registro. |

## 3.2. Alumno

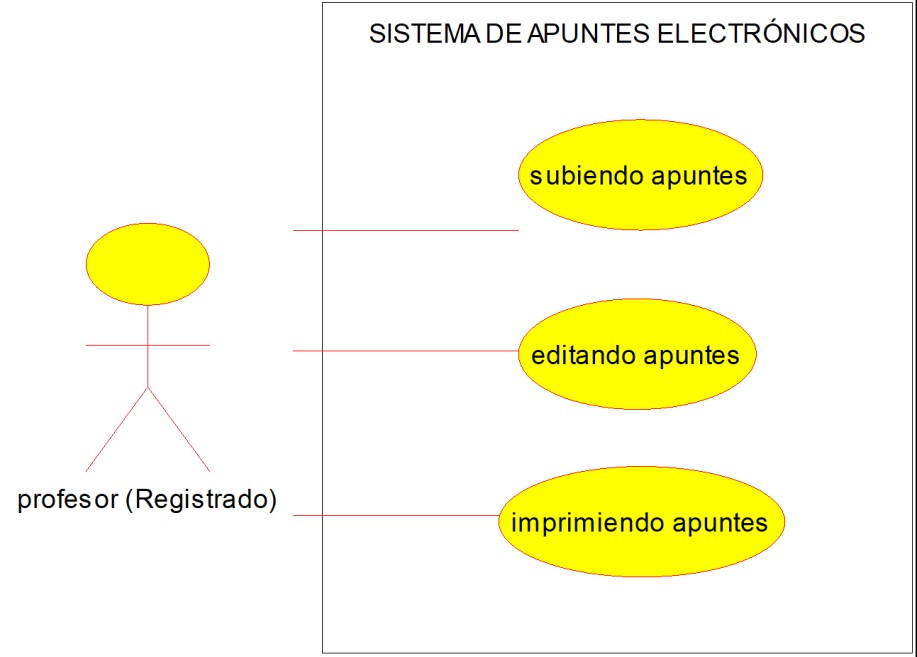
## 

|  |  |
| --- | --- |
| Caso de Uso: “b**uscando apuntes”** | |
| Actor: Alumno | |
| **Curso Normal** | **Alternativas** |
| 1) El usuario accede a la aplicación sin registro |  |
| 2) Ingresa el apunte a buscar |  |
| 3) El sistema obtiene los apuntes |  |
| 4) Se muestran los apuntes recolectados | 4.1 En caso de no obtener resultados, el sistema mostrará el mensaje correspondiente. |

|  |  |
| --- | --- |
| Caso de Uso: **“viendo apuntes completos”** | |
| Actor: Alumno | |
| **Curso Normal** | **Alternativas** |
| 1) El usuario selecciona el apunte de interés |  |
| 2) Se mostrará una página con información y un breve resumen |  |
| 3) El usuario dará click en el botón “ver apunte completo” |  |
| 4) Se mostrará el apunte completo con una vista dinámica para poder cambiar la página |  |

|  |  |
| --- | --- |
| Caso de Uso: **“imprimiendo apuntes”** | |
| Actor: Alumno | |
| **Curso Normal** | **Alternativas** |
| 1) El usuario selecciona el documento deseado abriendo la vista de libro. |  |
| 2) Se mostrará un botón en la interfaz que indicará la impresión del documento. |  |
| 3) Se mostrará la ventana de impresión por defecto del navegador |  |

## 3.3. Profesor

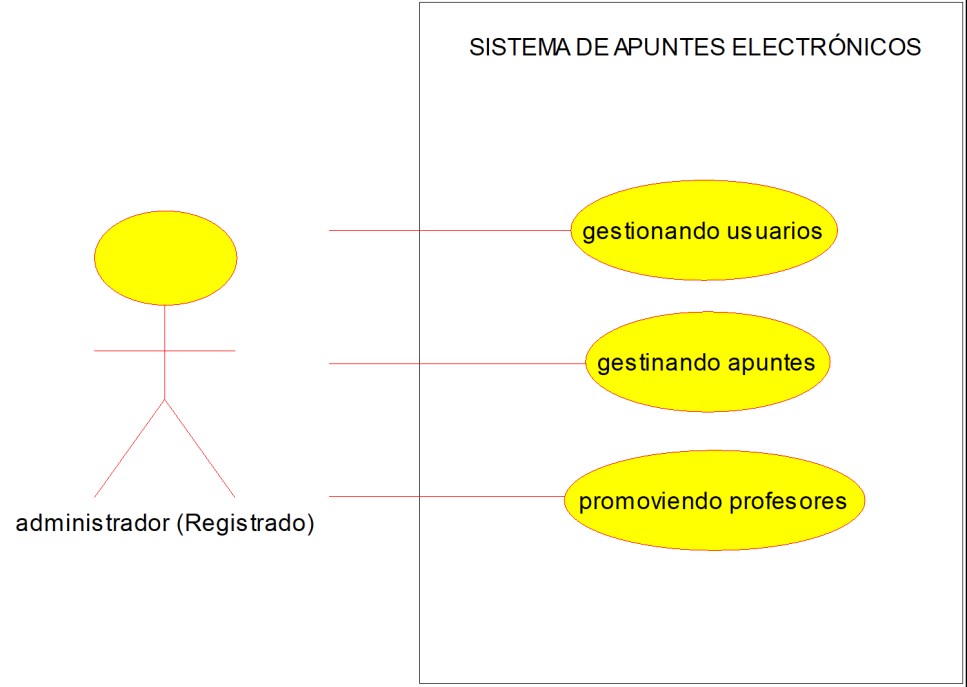


|  |  |
| --- | --- |
| Caso de Uso: **“subiendo apuntes”** | |
| Actor: Profesor | |
| **Curso Normal** | **Alternativas** |
| 1) El profesor entrará a su página Home |  |
| 2) Se mostrará un botón que indicará la creación de un nuevo apunte |  |
| 3) Se le llevará a una página para que se complete la información necesaria del documento. | 3.1) Si una información falta, se le notificará al usuario y no podrá publicar el apunte. |
| 4) El usuario podrá seleccionar un documento de su computador local. | 4.1) Si el usuario olvida de seleccionar el apunte, el sistema impedirá la publicación de la nota hasta que se seleccione. |
| 5) El usuario podrá dar click en aplicar cambios para publicar la nota | 5.1) El usuario podrá cancelar la publicación dando click en el campo “Descartar Cambios” |

|  |  |
| --- | --- |
| Caso de Uso: **“editando apuntes”** | |
| Actor: Profesor | |
| **Curso Normal** | **Alternativas** |
| 1) El profesor seleccionará el apunte indicado |  |
| 2) Se mostrará un botón que indicará el poder editar el apunte en concreto. |  |
| 3) Se le llevará a una página para que se pueda editar la información necesaria del documento. | 3.1) Si una información falta, se le notificará al usuario y no podrá publicar el apunte. |
| 4) El usuario podrá cambiar el documento de su computador local. | 4.1) Si el usuario olvida de seleccionar el apunte, el sistema impedirá la publicación de la nota hasta que se seleccione. |
| 5) El usuario podrá dar click en aplicar cambios para actualizar el apunte. | 5.1) El usuario podrá cancelar los cambios dando click en el campo “Descartar Cambios” |

|  |  |
| --- | --- |
| Caso de Uso: **“imprimiendo apuntes”** | |
| Actor: Profesor | |
| **Curso Normal** | **Alternativas** |
| 1) El usuario selecciona el documento deseado abriéndo la vista de libro. |  |
| 2) Se mostrará un botón en la interfáz que indicará la impresión del documento. |  |
| 3) Se mostrará la ventana de impresión por defecto del navegador |  |

## 3.4. Administrador



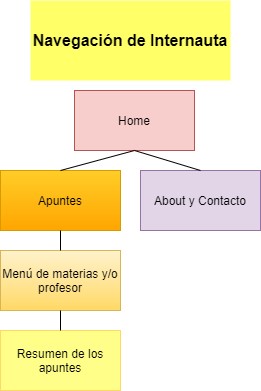
|  |  |
| --- | --- |
| Caso de Uso: **“gestionando usuarios”** | |
| Actor: Administrador | |
| **Curso Normal** | **Alternativas** |
| 1) El administrador seleccionará el alumno a editar |  |
| 2) Se mostrará en una página aparte la información del usuario, siendo capaz de modificar su información o eliminarlo. | 2.1) No se podrá aplicar los cambios si falta un campo obligatorio.    2.2) Si se cancelan los cambios, se regresará a la página de alumnos y se descartarán los cambios. |
| 3) Al aplicar los cambios, se regresará a la página que muestra la lista de alumnos. |  |

|  |  |
| --- | --- |
| Caso de Uso: **“gestionando apuntes”** | |
| Actor: Administrador | |
| **Curso Normal** | **Alternativas** |
| 1) El administrador seleccionará el apunte a revisar |  |
| 2) Se mostrará en una página aparte la información del apunte, siendo capaz de verlo y/o eliminarlo.    Se le notificará al profesor que su apunte fué eliminado, o que es necesaria una modificación. | 2.1) No se podrá aplicar los cambios si falta un campo obligatorio.    2.2) Si se cancelan los cambios, se regresará a la página de alumnos y se descartarán los cambios. |
| 3) Al aplicar los cambios, se regresará a la página que muestra la lista de alumnos. |  |

|  |  |
| --- | --- |
| Caso de Uso: **“promoviendo profesores”** | |
| Actor: Administrador | |
| **Curso Normal** | **Alternativas** |
| 1) El administrador seleccionará el profesor a promover. |  |
| 2) Se mostrará en una página aparte la información del apunte, donde el usuario registrado como profesor podrá ser aceptado o no.    Se le notificará al profesor que su solicitud fué aceptada. | 2.1) Si se deniega la solicitud, no procesará el registro del profesor. |
| 3) Al aceptar la solicitud de un profesor el sistema volverá a la página de profesores. |  |

# IV. Diagramas de Navegación

## 4.1. Internauta



*Mapa de navegación Principal del Internauta*

Descripción:

El diagrama de navegación muestra la capacidad de acceder al home, que, dependiendo del usuario, será distinto.

El “internauta” o usuario no registrado, solo podrá acceder al home principal, en el cual permitirá hacer búsquedas.

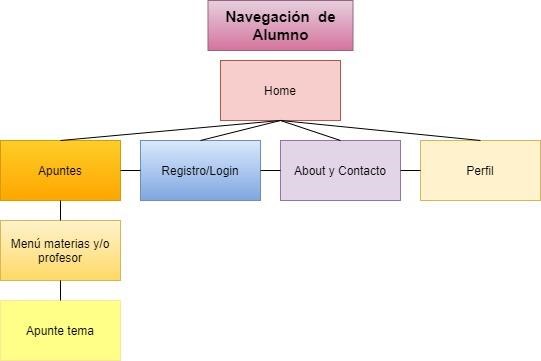
La página de apuntes muestra la interfaz general donde podrá hacer una nueva búsqueda, y mostrará los resultados de la anterior, en este caso, los apuntes obtenidos.

En la ventana de menú de materias, podrá visualizar algunas opciones extras para realizar la búsqueda, como búsqueda por Profesor, por materia, por Unidad y Tema.

En la ultima ventana, se podrá ver el resumen del apunte seleccionado.

La sección de About y contacto, muestran solo la información necesaria para ponerse en contacto con el equipo técnico de la aplicación, así como el FAQ e información general.

## 4.2. Alumno



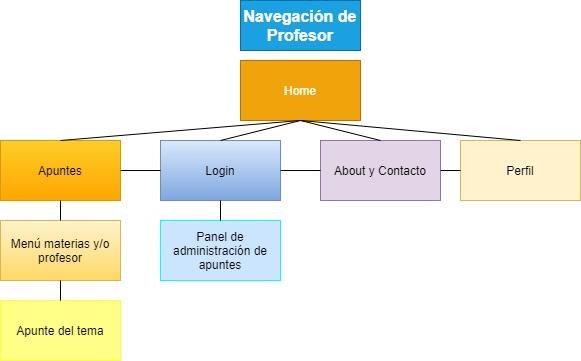
*Mapa de Navegación Principal del Alumno*

El mapa de navegación del alumno no cambia mucho respecto al de un internauta.

Ahora el Alumno podrá visitar su perfil o el de algún otro usuario.

Además, que en la ventana de “Apuntes Tema”, ahora podrá visualizar el apunte completo.

## 4.3. Profesor

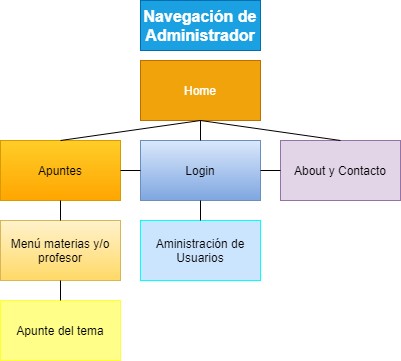


*Mapa de Navegación Principal del Profesor*

La navegación del profesor, es similar a la del alumno, pero con la diferencia de que el profesor ahora podrá acceder a una página dedicada a la administración de sus apuntes.

Esta página, se denomina “Panel de Administración de Apuntes” porqué esta dedicada a la función anteriormente descrita.

## 4.4. Administrador



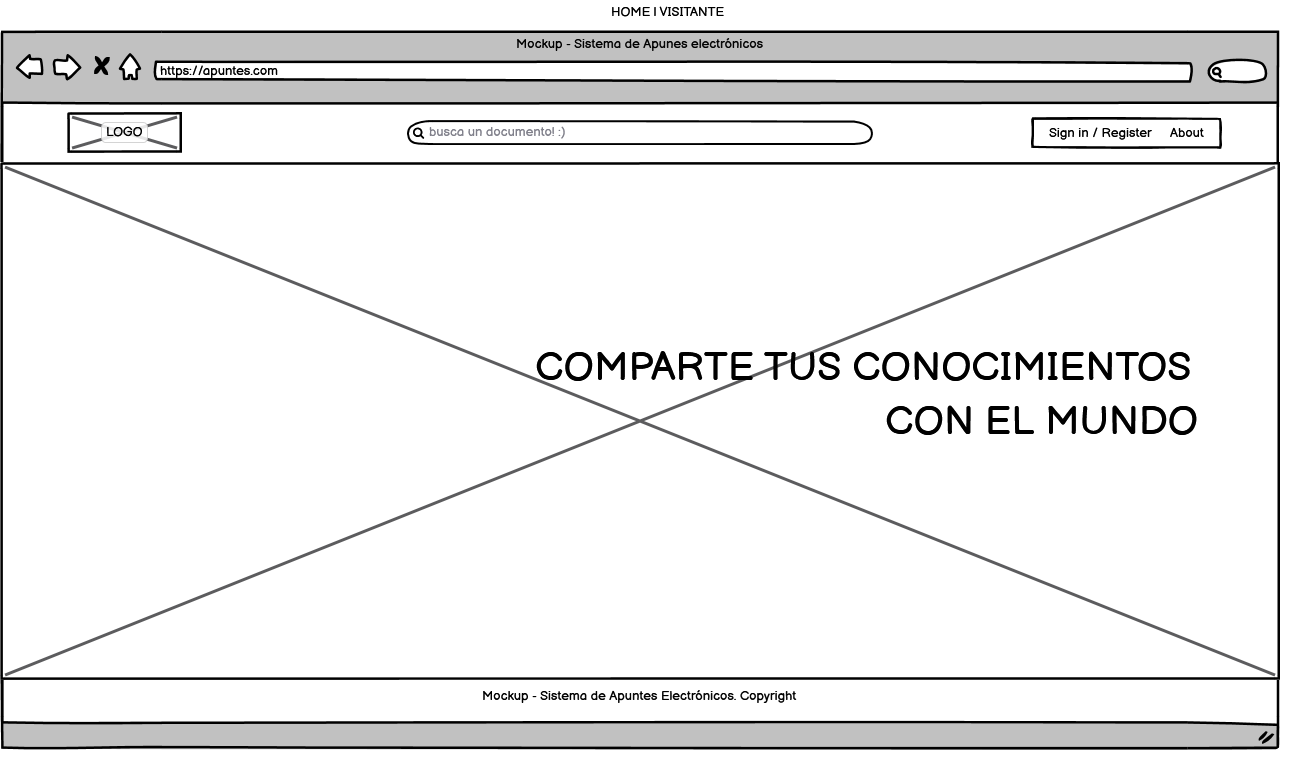
*Mapa de Navegación Principal del Administrador*

El mapa navegacional de un administrador se parece mucho al de un profesor, pero con la siguiente diferencia:

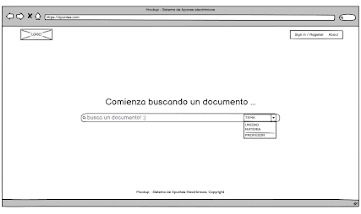
En el panel “Administración de Usuarios” o solo “Administración”, se gestionan todos los usuarios, y los apuntes en general. Así como la visualización de ciertas funciones relacionadas a ellos.

# 5.- Mockup

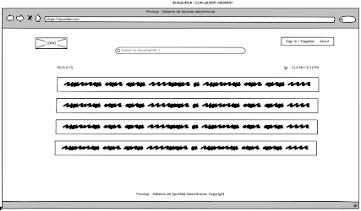
## 5.1. Internauta



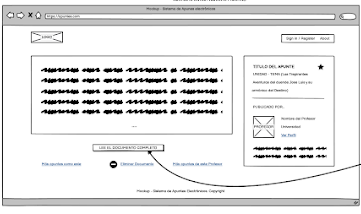
*1.1 - Página de visualización sin registro*



*1.2 - Página de búsqueda general*



*1.3 - Página de visualización de Resultados General*

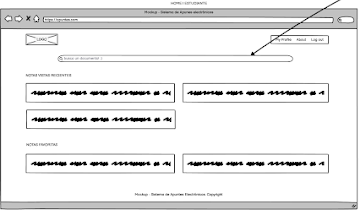


*1.4 - Página de resumen del Apunte*

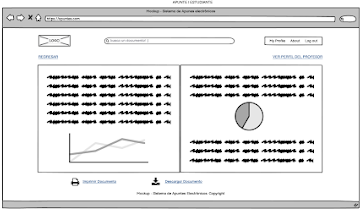


*1.5 - Estructura de los formularios*

## 5.2. Alumno

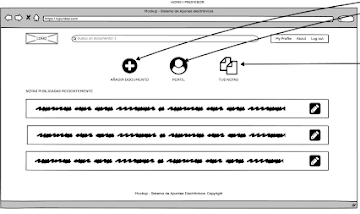


*2.1 - Home del Alumno*



*2.2 – Visualización del Apunte en formato de libro.*

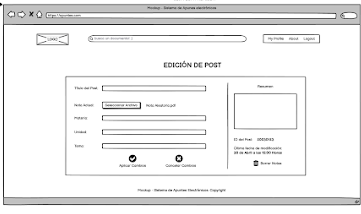
## 5.3. Profesor



*3.1 – Home / Dashboard del Profesor.*



*3.2 – Notas Publicadas del Profesor.*

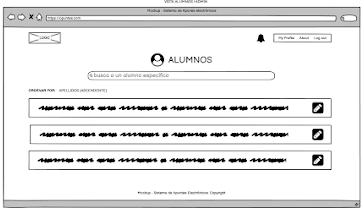


*3.3 – Edición de los apuntes del Profesor.*

## 5.4. Administrador



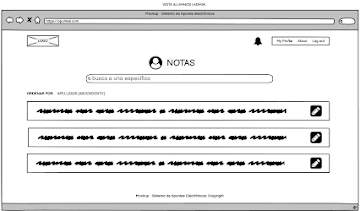
*4.1 – Home / Dashboard del Administrador.*



*4.1 – Vista de Alumnos del Administrador.*



*4.1 – Vista de Profesores del Administrador.*



*4.1 – Vista de Apuntes del Administrador.*

# 6.- DESARROLLO

Se construye un modelo orientado a objetos según (KOCH 2002) que represente el dominio de la aplicación usando las técnicas propias de la orientación a objetos. La finalidad principal durante esta fase es capturar el dominio semántico de la aplicación en la medida de lo posible, teniendo en cuenta el papel de los usuarios y las tareas que desarrollan. El resultado de esta fase es un modelo de clases relacionadas que se divide en subsistemas

Una vez definida la estructura navegacional, hay que prepararla para que sea perceptible por el usuario y esto es lo que se intenta en esta fase. Esto consiste en definir qué objetos de interfaz va a percibir el usuario, y en particular el camino en el cuál aparecerán los diferentes objetos de navegación, qué objeto de interfaz actuará en la navegación, la forma de sincronización de los objetos multimedia y el interfaz de transformaciones. Al haber una clara separación entre la fase anterior y esta fase, para un mismo modelo de navegación se pueden definir diferentes modelos de interfaces, permitiendo, así que el interfaz se ajuste mejor a las necesidades del usuario.

Una vez obtenido el modelo conceptual, el modelo de navegación y el modelo de interfaz abstracta, sólo queda llevar los objetos a un lenguaje concreto de programación, para obtener así la implementación ejecutable de la aplicación.

OOHDM es sin duda una de las metodologías que más aceptación ha tenido, y sigue teniendo, en el desarrollo de aplicaciones multimedia. Actualmente está sirviendo como base para el desarrollo de nuevas propuestas metodológicas para los sistemas de información web. OOHDM es una propuesta basada en el diseño, que ofrece una serie de ideas que han sido asumidas por bastantes propuestas y que han dado muy buenos resultados. La primera de ellas es que hace una separación clara entre lo conceptual, lo navegacional y lo visual. Esta independencia hace que el mantenimiento de la aplicación sea mucho más sencillo. Además, es la primera propuesta que hace un estudio profundo de los aspectos de interfaz, esencial no solo en las aplicaciones multimedia, sino que es un punto crítico en cualquiera de los sistemas que se desarrollan actualmente.

OOHDM hace uso también de la orientación a objetos y de un diagrama tan estandarizado como el de clases, para representar el aspecto de la navegación a través de las clases navegacionales: índices, enlaces y nodos. Esta idea ha dado muy buenos resultados y parece muy adecuada a la hora de trabajar.

Sin embargo, y a pesar de esto, OOHDM presenta algunas deficiencias. OOHDM ha dejado fuera de su ámbito un aspecto esencial que es el tratamiento de la funcionalidad del sistema. El qué se puede hacer en el sistema y en qué momento de la navegación o de la interfaz se puede hacer, es algo que no trata y que lo deja como tarea de implementación.

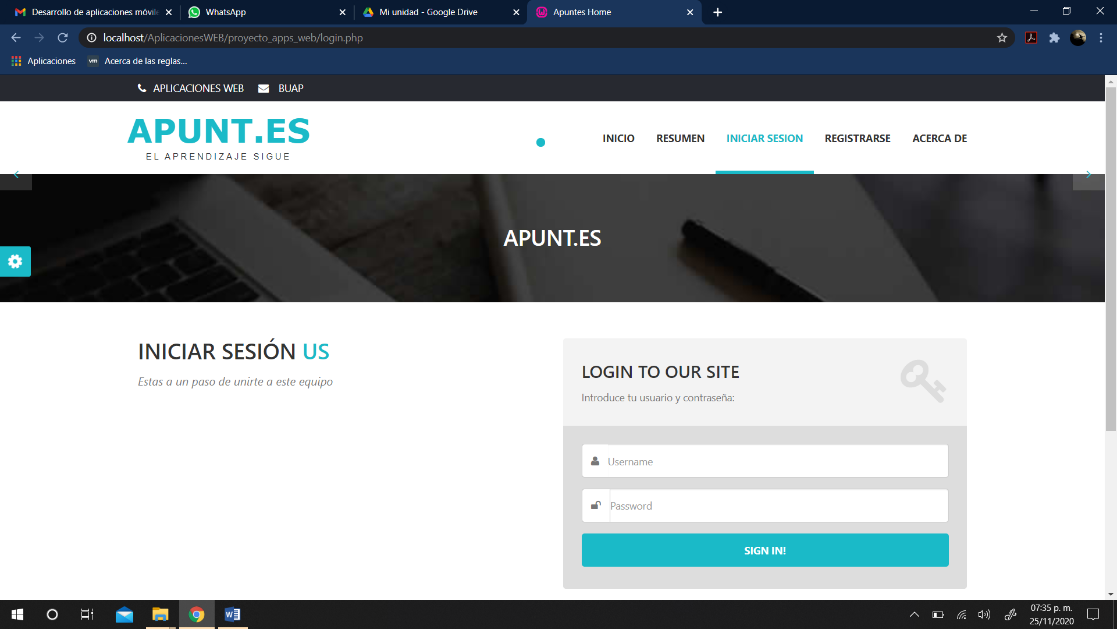
Además, OOHDM no ofrece ningún mecanismo para trabajar con múltiples actores. Por ejemplo, imaginemos que la interfaz y la navegación de la aplicación varia sustancialmente dependiendo de quién se conecte a la aplicación. El diagrama navegacional, los contextos navegacionales y los ADVs resultarían muy complejos para representar esta variabilidad.

# 7.- PRUEBAS

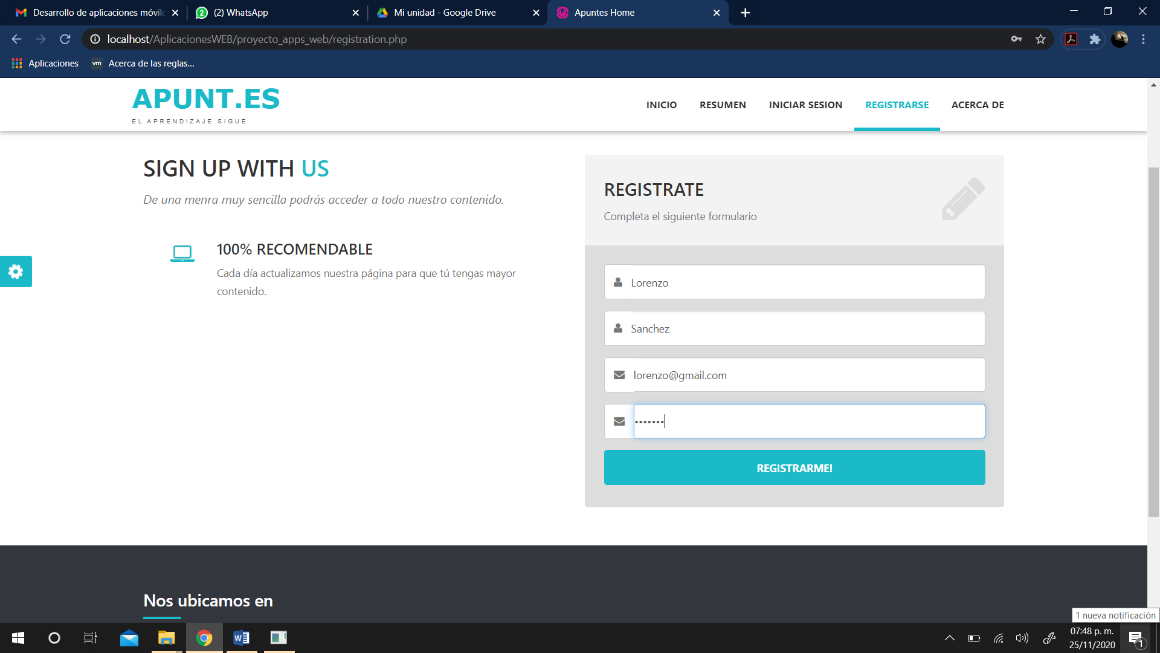
Se aprecia la página principal de nuestro sitio web, existe las diferentes paginas las cuales representan las principales funcionalidades de nuestro sistema.



Implementación de variables de sesión, como son abrir sesión, pasar usuario durante su estancia en el sitio web y cerrar sesión, mediante las funciones de php. Mediante variables de sesión que maneja php se inicia la sesión y se mantiene los datos del usuario en su navegación.



Cuando se llena el formulario estos datos son enviados a otro programa que los reproecsa y hace su insercción en la base de datos. Es funcional este resgitro.

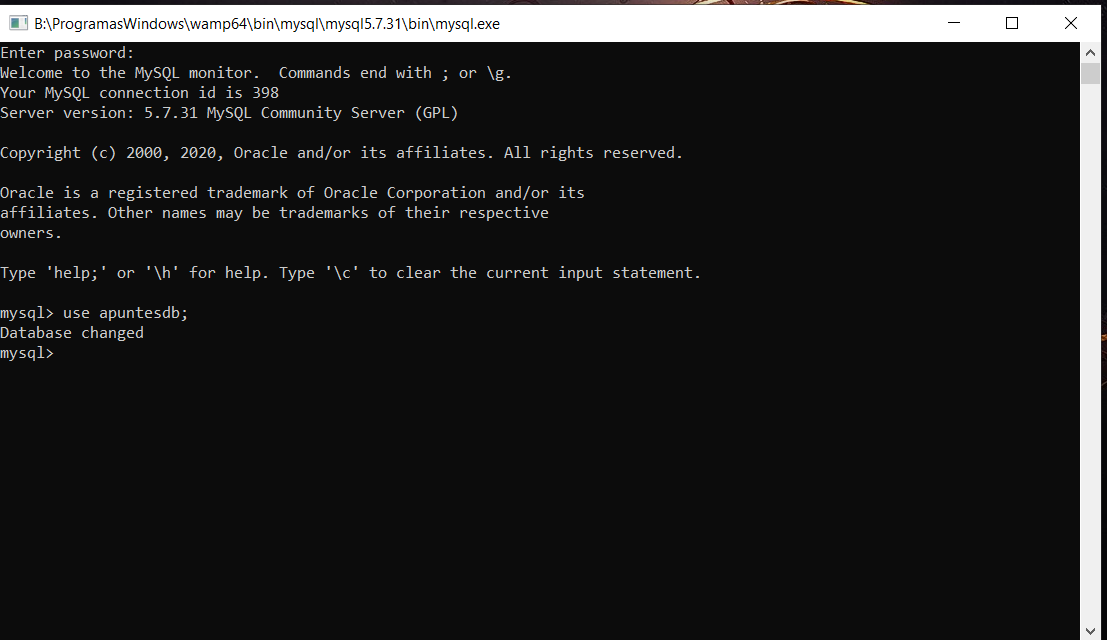


Se hace una presentación del equipo, se dan a conocer sus mayores fortalezas.

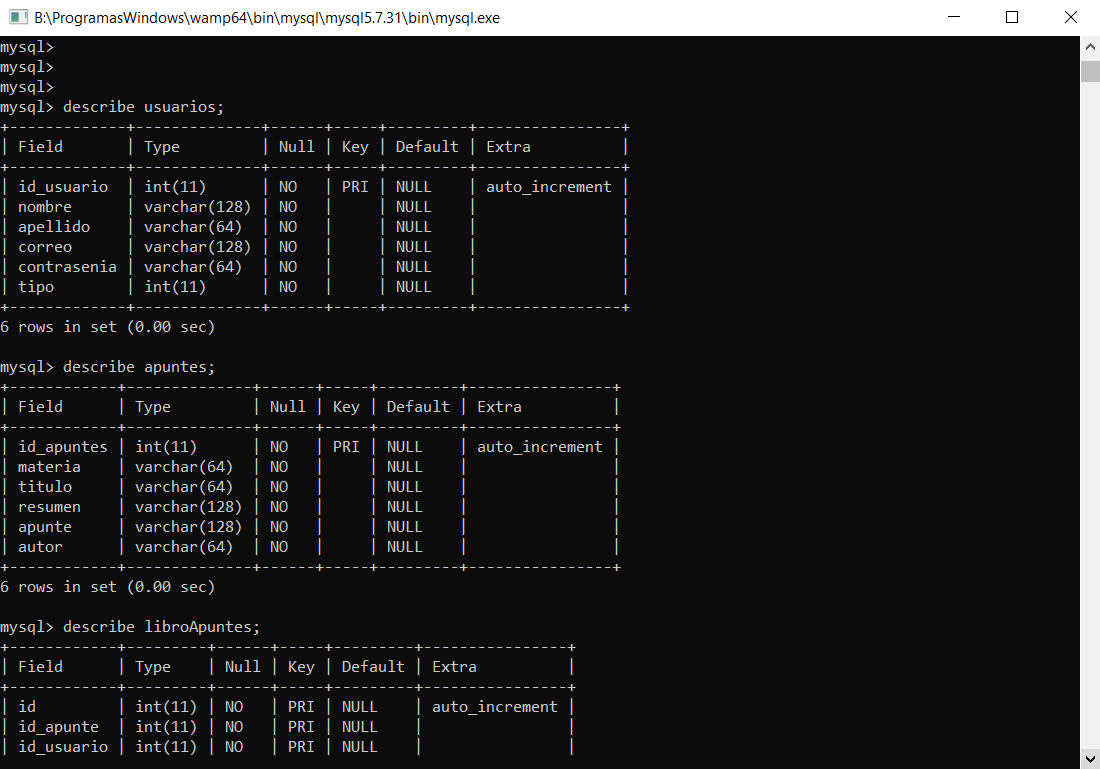


### 7.1 -Creación de la base de datos

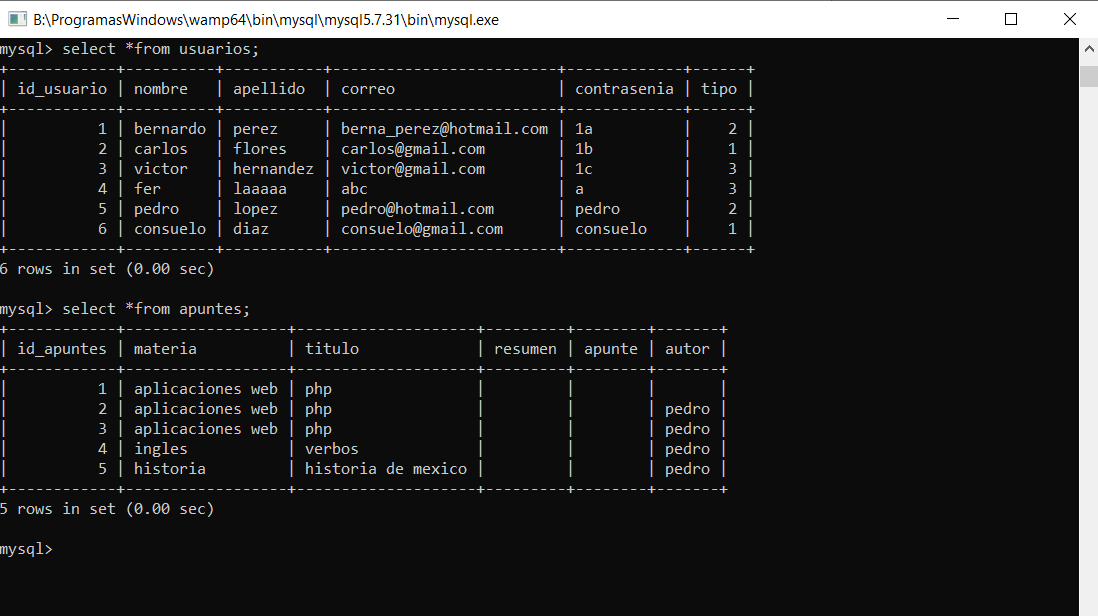
Se ha introducido registros la base de datos, los cuales han sido registrados de manera correcta mediante consola y mediante la inserción de la página web.



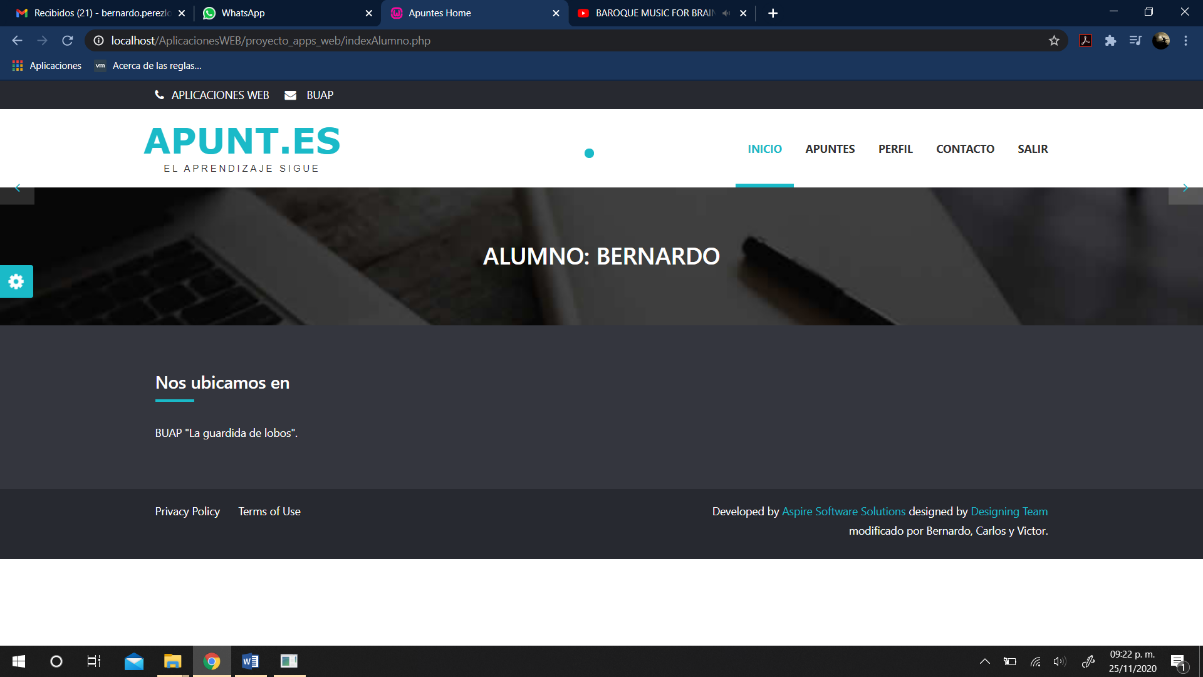
### 7.4 - Modelado de las tablas.



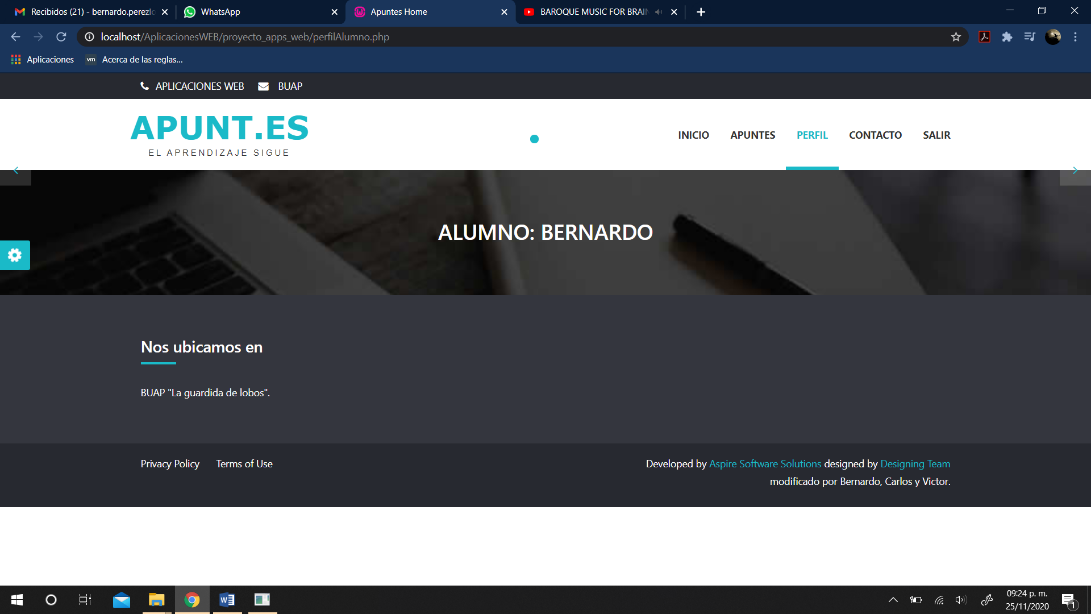
Llenado de las tablas, tanto de las tablas de usuarios y apuntes.



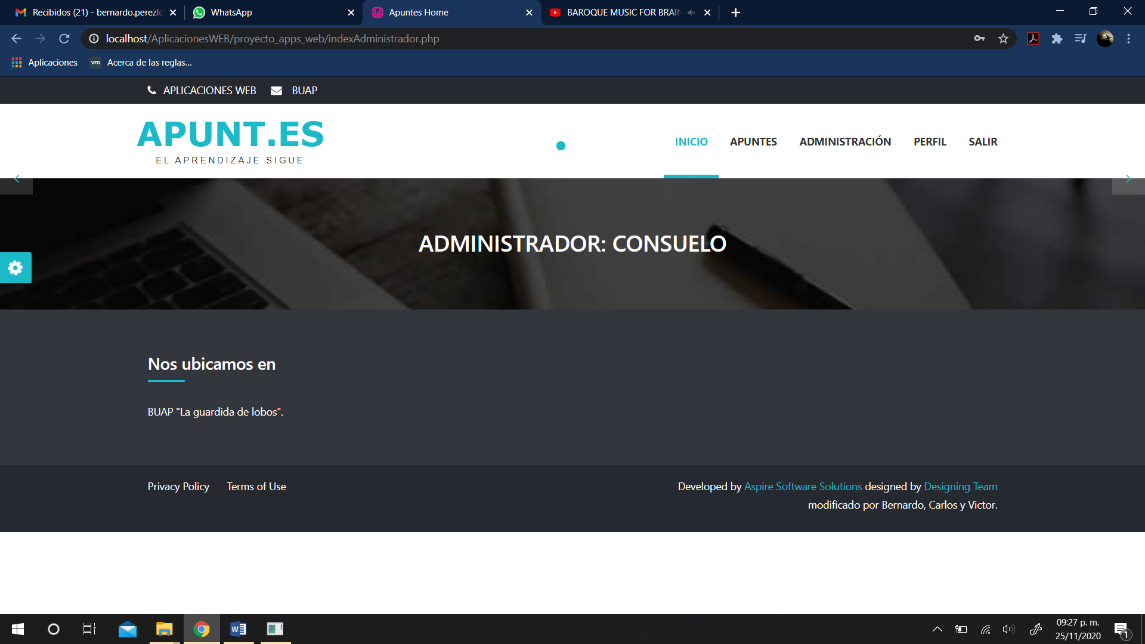
Aquí vemos que el registro ya se ha hecho en la base de datos, entra a la página correspondiente y las vistas que tiene son correctas.



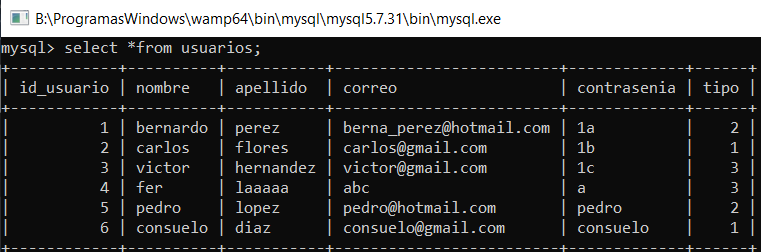
Hace falta desarrollar algunas funcionalidades en este apartado, serán para un trabajo posterior, solo está la vista.



Una vista del administrador.



Se puede observar que consuelo es de tipo, es decir, es administrador.



# 8.- REFERENCIAS

1. Definición SCRUM. 2018, <https://www.scrum.org/resources/blog/que-es-scrum>
2. Manual de PHP, <https://www.php.net/manual/es/intro-whatis.php>
3. CSS, Mozilla web docs, 2005-2020, <https://developer.mozilla.org/es/docs/Web/CSS>
4. MYSQL Y PHP, MYSQL (Original), <https://www.php.net/manual/es/book.mysql.php>
5. Preguntas con etiqueta PHP, <https://es.stackoverflow.com/questions/tagged/php>
6. Metodologias WEB,Abraham Sanches , FCC BUAP. <https://www.cs.buap.mx/~asanchez/iweb/MetodologiasWeb.pdf>
7. Html, Lenguaje de etiquetas de hipertexto <https://developer.mozilla.org/es/docs/Web/HTML>
8. JavaScript Guide, 2005-2020 <https://developer.mozilla.org/es/docs/Web/JavaScript>
9. Ingeniería de software I, Casos de uso, Santiago Ceria.
10. Metodologías de desarrollo de software, Esteban Gabriel, Universidad Católica de Argentina.
11. Modelo de diseño hipermedia orientada a objetos.