

INFORMA&FORMA-T



Fco. José de Vicente

Trabajo Fin de Grado Superior de 2º de
Desarrollo de Aplicaciones Web.

Instituto Tecnológico
Edix

2020/2021

Después de dos años del ciclo formativo de Grado Superior, llega el momento de concluir con la realización de este proyecto. Durante este recorrido, he vivido multitud de experiencias con mucha gente que me ha ayudado en todo momento, tanto en el desarrollo del ciclo como en otros aspectos personales. Por ello, quiero agradecer a todas estas personas que forman parte de mi vida y a las nuevas incorporaciones que me han aportado durante esta formación.

- A mi familia, en primer lugar como no podía ser de otra manera. Por quererme y apoyarme en todo momento, por haber soportado mis cambios de humor constantes, mis frustraciones y celebraciones cuando salían las cosas, por esa paciencia infinita y los ánimos que me sacaban aunque a veces ni los mereciera.
- A mis amigos, comprensivos en todo momento incluso cuando no tenía ni tiempo para dedicarles. Por demostrar el significado de la palabra amistad y hacer que me sintiera en casa después de meses estudiando y desaparecido.
- A mis compañeros, a todos y cada uno de ellos, en especial a Pilar Bermejo y Raúl Montero que han estado ayudándome en cada ápice del curso, sin esperar nada a cambio y cediéndome su tiempo para que comprendiera el porqué de las cosas y la lucha constante por sacarlas, por su amistad y compañerismo incondicional y su compañía en los momentos más difíciles en los que sólo pensaba tirar la toalla.
- A mis profesores y el equipo del instituto, nacido en el instituto tecnológico Telefónica del primer año y creciendo como Edix en el segundo. Por enseñarme con entusiasmo, por hacer que durante la formación se convirtieran en mi segunda familia, acompañándome de la mano y despertar mis ganas y motivación de seguir aspirando a más en mi futuro profesional.

En definitiva y con la mano en el corazón, a todos y cada uno de ellos, **GRACIAS.**

ÍNDICE

1.- Memoria del proyecto	Pág. 07
1.1.- Memory (English Version)	Pág. 08
2.- Objetivos del proyecto	Pág. 09
2.1.- Estudios de la Situación Actual	Pág. 10
2.2.- Sistemas Actuales	Pág. 10
2.2.1.- Udemy	Pág. 10
2.2.2.- Discover	Pág. 11
2.2.3.- Adams	Pág. 11
3.- Módulos Formativos Aplicados en el TFG	Pág. 12
4.- Herramientas/Lenguajes Utilizados	Pág. 14
4.1.- Eclipse Java EE	Pág. 14
4.2.- MySQL WorkBench	Pág. 15
4.3.- Visual Studio Code	Pág. 16
4.4.- Git	Pág. 16
4.5.- GitHub	Pág. 17
4.6.- Software Ideas Modeler	Pág. 17
4.7.- StoryBoardThat	Pág. 17
5.- Fases del Proyecto	Pág. 18
5.1.- La Idea	Pág. 18
5.2.- Ficha del Perfil	Pág. 18
5.3.- StoryBoard	Pág. 19
5.4.- Caso de Uso	Pág. 20
5.5.- WireFrame	Pág. 21
5.6.- Guía de Estilos	Pág. 23
5.7.- Bases de Datos	Pág. 24
5.8.- HTML	Pág. 26
5.9.- CSS	Pág. 27
5.10.- JavaScript	Pág. 28
5.11.- Modelo, Vista, Controlador	Pág. 29
5.11.1.- Modelo Beans	Pág. 30
5.11.2.- Modelo DAO	Pág. 31
5.11.3.- Vista	Pág. 33
5.11.4.- Controlador	Pág. 35
5.11.5.- Servidor (Apache Tomcat)	Pág. 37
5.12.- Plan de Empresa	Pág. 38
5.12.01.- Expectativas	Pág. 38
5.12.02.- Localización del Negocio	Pág. 38
5.12.03.- Micro-Entorno	Pág. 39
5.12.04.- Factor Económico	Pág. 39
5.12.05.- Macro-Entorno	Pág. 40
5.12.06.- Competencia	Pág. 40

5.12.07.- Análisis DAFO	Pág. 40
5.12.08.- Plan de Marketing	Pág. 41
5.12.09.- Proveedores	Pág. 42
5.12.10.- Plan Financiero	Pág. 42
5.12.11.- Plan Económico	Pág. 43
6.- Conclusiones	Pág. 47
7.- Bibliografía	Pág. 48
8.- Anexos	Pág. 48

ÍNDICE DE IMÁGENES

• Portada	Pág. 00
• Imagen Portada Competencia	Pág. 10
○ Udemy	Pág. 10
○ Discover	Pág. 11
○ Adams	Pág. 11
• Eclipse	Pág. 14
• MySQL WorkBench (características)	Pág. 15
○ ID	Pág. 15
• Visual Studio Code	Pág. 16
• StoryBoardThat	Pág. 17
○ Fichas Perfiles	Pág. 18
○ StoryBoard	Pág. 19
• Casos de Uso	Pág. 20
• Wireframe	Pág. 21
○ Home	Pág. 21
○ Registro usuario	Pág. 22
○ Contacto	Pág. 22
• Guía de Estilos	Pág. 23
• Bases de Datos	Pág. 24
○ Script	Pág. 25
• HTML	Pág. 26
• CSS	Pág. 27
• JS	Pág. 28
○ Formulario	Pág. 28
• JPA	Pág. 29
• Modelo Beans	Pág. 30
• Modelo DAO	Pág. 32
○ Interfaz	Pág. 32
○ Implementación	Pág. 33
• Vista	Pág. 33
○ JSP	Pág. 34
○ JSTL/EL	Pág. 34
• Controlador	Pág. 35
○ Login	Pág. 36
○ Servlet	Pág. 36
○ Servlet proceso	Pág. 36
○ Servidor Apache Tomcat	Pág. 37
• Plan de Empresa	Pág. 38
○ Catálogo de productos	Pág. 38
○ Mapa	Pág. 38

○ Estadística _____	Pág. 39
○ Análisis DAFO _____	Pág. 40
○ Folleto publicitario _____	Pág. 41
○ Banner Redes Sociales _____	Pág. 41
○ Proveedores _____	Pág. 42
○ Plan de inversiones _____	Pág. 43
○ Plan Financiación _____	Pág. 44
○ Presupuesto tesorería _____	Pág. 44
○ Previsión Pérdidas y Ganancias _____	Pág. 45
○ Balance de situación _____	Pág. 45
○ Patrimonio _____	Pág. 46
○ Liquidación IVA _____	Pág. 46

1.- Resumen – Memoria del Proyecto

Este Trabajo Fin de Grado para el Ciclo Formativo de Grado Superior en Desarrollo de Aplicaciones Web, expone la gestión de información acerca de los diferentes cursos de formación que existen en el mercado.

Esta idea surge debido a que el presente alumno se ha dedicado a este sector durante seis años y medio, hasta que tuvo que finalizar su relación laboral para centrarse en sus estudios actuales. Durante su etapa laboral, ha visto y ha vivido en primera persona las necesidades que se pueden presentar a la hora de elegir una formación, así como la dificultad que genera la elección de una formación en concreto y porque no mencionarlo, el incordio que genera el recibo masivo de llamadas comerciales con las respectivas ofertas que hay en el mercado, generando así un agobio al estudiante/cliente y creando incluso una sensación de rechazo, no sólo por parte del centro en el cual formarse, sino que también en la decisión final sobre si estudiar o no dicha formación.

Dada la experiencia personal del alumno, tomó la decisión de formarme con el Instituto Tecnológico Telefónica ya que fue la única escuela que no se preocupó en acosarle a llamadas comerciales para venderle su formación, sino que atendió sus necesidades, explicándole su oferta formativa y dejando finalmente en sus manos la toma de su decisión final, sin ningún tipo de insistencia. Esta idea fue lo que llamó su atención para dar un giro radical sobre la gestión profesional que realizaba en aquel entonces y en la que aprendió a escuchar a sus clientes, trayéndose como consecuencia la generación casi de manera automática, de más matriculados y por consiguiente, más ventas. Llegando a superar incluso los objetivos estipulados por parte de su empresa.

Con esta página web lo que trata el alumno es de aliviar al futuro estudiante de las diversas llamadas comerciales y que si se siente en una situación de estancamiento personal y pretende dar un cambio profesional, que esta situación no afecte en su decisión, como ocurre en el 70% de los casos, en los que este tipo de perfil acaba por desestimar formarse y finalmente perder el interés por desarrollarse.

En conclusión, esta aplicación web expone los cursos que ofrece de manera que el alumno pueda ver con todo lujo de detalles sus contenidos, información adicional como a quién va dirigido, qué salidas laborales tiene, para qué sirve y todo lo que necesite sin tener que estar recibiendo llamadas comerciales y pueda realizar su compra sin ningún tipo de presión por parte de la compañía.

1.1.- Summary – Project Memory

This Final Degree Project for the Higher Degree Training Cycle in Web Application Development exposes the management of information about the different training courses that exist in the market.

This idea arises because the present student has been dedicated to this sector for six and a half years, until he had to end his employment relationship to focus on his current studies. During his work stage, he has seen and experienced first-hand the needs that may arise when choosing a specific training, as well as the difficulty generated by choosing a specific training and why not mention it, the nuisance generated by the massive receipt of commercial calls with the respective offers that are in the market, thus generating an overwhelm to the student / client and even creating a feeling of rejection, not only by the center in which to train but also in the final decision on whether to study or not said training.

Given the student's personal experience, he made the decision to train with the Telefónica Technological Institute because it was the only school that did not worry about harassing him on commercial calls to sell him his training, but rather attended to his needs, explaining his training offer and finally leaving in Your hands make your final decision, without any insistence. This idea was the one that caught his attention to take a radical turn on the professional management that he carried out at that time and in which he learned to listen to his clients, bringing as a consequence the generation almost automatically, of more enrolled and consequently, more sales. Even exceeding the objectives stipulated by your business.

With this web page what the student tries is to relieve the future student of the various commercial calls and that if he feels in a situation of personal stagnation and intends to give a professional change, that this situation does not affect his decision, as it happens in 70% of the cases, in which this type of profile ends up rejecting training and finally losing interest in developing.

In conclusion, this web application exposes the courses it offers so that the student can see its contents in great detail, additional information such as who it is addressed to, what job opportunities they have, what it is for and everything they need without having to be receiving commercial calls and can make your purchase without any pressure from the company.

2.- Objetivos del Proyecto

El proyecto que se ha desarrollado, pretende alcanzar estos objetivos y cubrir las necesidades que se enumeran a continuación:

1. Permitir al usuario valorar los diferentes cursos formativos sin necesidad de recibir llamadas comerciales para la insistencia de la venta posterior.
2. Suministrar una página web como herramienta para ser utilizada por diversos usuarios a la vez, pudiendo visualizar sus contenidos y realizar sus compras por iniciativa propia.
3. Permitir al usuario añadir productos a su carrito de compra, dando lugar después a poder eliminarlos o finalizar dicha compra, sin la presión de un/a agente comercial.
4. Dar una información clara y concisa sobre los objetivos del curso elegido por el usuario, así como los detalles más significantes que lo forma.

2.1.- Estudio de la situación actual

En esta sección se refleja un estudio de las páginas de formación actuales con su descripción, ventajas y desventajas y una evaluación de las diferentes alternativas para el desarrollo del proyecto.

2.2.- Sistemas actuales

Actualmente hay diversas plataformas y páginas web que se dedican a la venta de formación. Sin embargo, tienen como finalidad la realización posterior de llamadas comerciales con la excusa de explicar al alumno/cliente información detallada y aclarar las dudas de su elección.

A continuación, se muestran las tres páginas principales del mercado formativo con sus ventajas y desventajas.

2.2.1- Udemy

Es una de las mejores páginas de formación online y la que más se asemeja a los objetivos principales del alumno, como es la eliminación de llamadas comerciales. No obstante, si pasamos por su página web, la desventaja que vemos es como en todo momento destaca la venta indirecta del producto, haciendo que el usuario se desvincule de esa sensación de insistencia y se centre en el objetivo de formarse. Cada vez que pasamos el cursor del ratón para ver la información de alguno de sus cursos, no deja de visualizarse una ventana que “te dice: compra, compra, compra” o incluso, cuando se accede a ver la información detallada, aparece una nueva sección a mano derecha que se queda de manera permanente y bien visible, para que sus clientes no olviden en ningún momento de pasar por caja, sin importar la formación que elijan.



Imagen de la página principal de Udemy www.udemy.com

2.2.2.- Discover

Es otra de las principales páginas de formación online, una de las principales desventajas de esta web es que los cursos se imparten solamente en inglés, lo que desmarca a todo un país limitando a sus alumnos/clientes el poder realizar una formación con la comprensión necesaria que requiere para su desarrollo.

The screenshot shows the Discover website's interface. At the top, there's a navigation bar with the 'discover' logo, a dropdown menu for 'Categorías', and links for 'Talleres de trabajo', 'Crear curso', and 'Cursos gratuitos'. On the right, a link says 'Explorar los más populares'. The main content area features a large advertisement for 'El curso completo de Python' by Mosh Hamedani. The ad includes a video thumbnail of the instructor, a list of features (12 hours of HD content, coding exercises, practice activities, and lifetime access), a 'Mejor precio garantizado' badge, and a price of 'PS23 PS29'. A green 'Enlistate ahora' button is at the bottom right of the ad, with a 30-day money-back guarantee note below it.

www.discoverteachable.com

2.2.3.- Adams

La escuela formativa Adams, lleva más de una década vendiendo formación online. Aparentemente, su página web es atractiva e intuitiva. Incluso ofrece formación gratuita y hace despertar el interés del usuario para su visita. Sin embargo, una de las principales desventajas sin generar criterios personales, es la falta de claridad en sus objetivos ya que cuando se ingresa en su página web, al acceder a uno de los diferentes cursos que ofrece sin coste alguno, al tratar de adquirir dicha formación obliga a rellenar un formulario de inscripción en el que una vez cumplimentado, el alumno recibe una llamada comercial indicándole que debe abonar unos costes de matrícula sin mencionar, que en caso de negativa abordan la llamada para ofrecerte otras áreas y cursos “gratuitos” que disponen y la constante presión de sus agentes comerciales.

The screenshot shows the Adams website. The header includes the Adams logo, a 'Campus Virtual' button, a 'Contacto' link, and a phone number '617 109 109'. A search bar is also present. The main content area has two large advertisements for 'Cursos Gratuitos Para Desempleados' and 'Cursos Gratuitos Para Trabajadores'. To the right, there's a '¿Necesitas ayuda?' section with a form for users to provide their details (Name, Surname, Email, Phone, Postal Code, Occupation) for assistance.

3.- Módulos Formativos Aplicados en el TFG

Durante el desarrollo del proyecto, se ha marcado como objetivo abarcar el máximo número de módulos implicados durante los dos años de la formación, expresando lo más destacado en cada uno de ellos. Alguno de los módulos son:

1. Programación y Desarrollo Web Entorno Servidor

El desarrollo de la página web contiene programación orientada a objetos en la parte Back-End de la misma, abarcando a su vez la asignatura de desarrollo web entorno servidor y dando la funcionalidad dinámica de la web.

2. Bases de Datos

La aplicación web dispone de 4 tablas principales, siendo la tabla “compras” el foco de todas ellas. Pues aunque el objetivo principal sea evadir al usuario final de ello, no podemos olvidar su principal determinación.

Al ser una base de datos relacional, la tabla compras va relacionada con la tabla “cursos” en relación uno a muchos (N:M), en la que una compra puede contener muchos cursos pero un curso solo puede pertenecer a una compra (que no en cuanto a cantidades). Esta tabla está relacionada con la tabla “tipos” también en relación N:M en la que un tipo puede tener muchos cursos pero un curso solo pertenece a un tipo. Por último, tenemos la tabla “usuarios” relacionada con la tabla principal en relación N:M en la que un usuario podrá hacer muchas compras y las compras las realizará un solo usuario de cada sesión.

Por otro lado, el alumno ha contemplado realizar una tabla más, llamada “perfiles” para cuando impulse su página web, pueda generar el perfil del profesor además del ya existente perfil alumno. Esta tabla ha generado a su vez una tabla renacida llamada “usuario_perfiles”. Ver pág. 21

3. Entornos de Desarrollo

Durante todo el proceso del proyecto, se ha trabajado con distintos entornos de desarrollo. Esta parte se verá más adelante en la sección “herramientas” no obstante, cabe mencionar que se ha utilizado los principales objetivos de esta asignatura, como es la utilización de “debug” para testear el código fuente y subsanar errores que han ido surgiendo durante su desarrollo. También esta asignatura ha servido para crear el correspondiente diagrama de casos de uso para el usuario final. Por otro lado, no podemos dejar de mencionar una de las principales herramientas durante este proceso como es su control de versiones **git**, el cual también se detallará la versión utilizada en la próxima sección.

4. **Lenguajes de Marcas y Desarrollo Web Entorno Servidor**

Gracias a esta asignatura, el alumno ha sido capaz de desarrollar toda la parte Front-End con el conocimiento de HTML, CSS y JavaScript entre otros. Para la parte de la asignatura Desarrollo Web Entorno Servidor, ha realizado el contenido estático de la aplicación utilizando pequeñas validaciones en formularios.

5. **Diseño de Interfaces Web**

Este módulo recoge el mapa conceptual de la aplicación, así como la utilización de herramientas para la creación de su WireFrame, StoryBoard del usuario y la guía de estilos de dicha aplicación.

6. **Inglés**

El alumno añade en la página estática la posibilidad de traducir el contenido de la misma al idioma impartido en esta asignatura. Para ello, ha incluido el logo de la bandera inglesa para que al pulsar sobre ella, realice la función nombrada.

7. **Empresa e Iniciativa Emprendedora**

Durante el desarrollo del proyecto, ha valorado en todo momento la oferta que ofrece su competencia y cómo busca el alumno desmarcarse respecto a ellos, realizando un estudio de mercado para atraer a sus clientes/alumnos.

4.- Herramientas/Lenguajes Utilizados

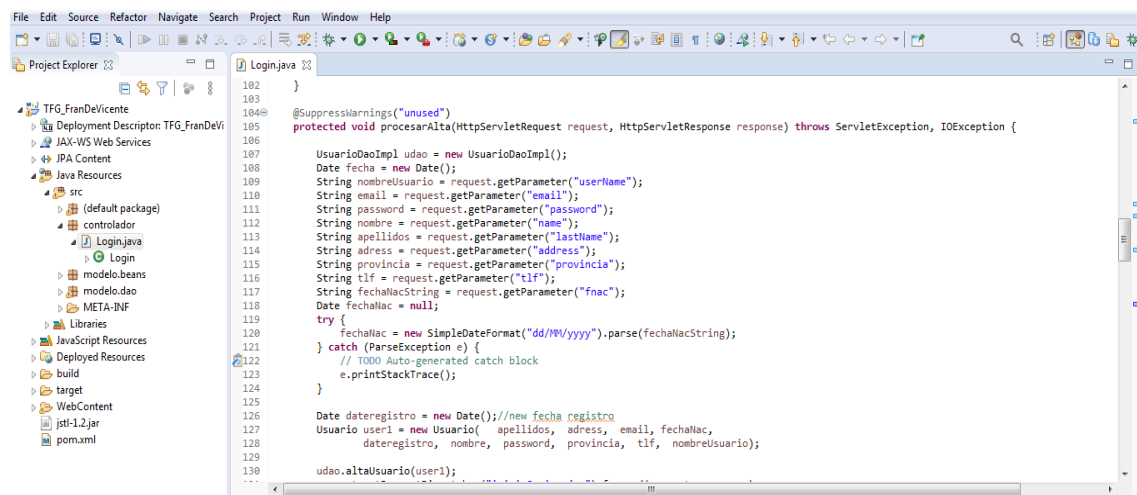
4.1.- Eclipse Java EE

Eclipse es un programa compuesto por un conjunto de herramientas de programación de código abierto que permiten programar en diferentes lenguajes. Originalmente fue desarrollado por IBM, pero ahora es desarrollado por la Fundación Eclipse, que es un consorcio de empresas fabricantes de software.

Emplea diferentes módulos (plug-in), que el usuario puede añadir si lo necesita. Dichos módulos permiten añadir diferentes lenguajes de programación, como por ejemplo C o C++, ya que el lenguaje principal empleado es Java.

Eclipse tiene un editor de texto con resaltado de sintaxis, mostrando posibles fallos y destacando el código dependiendo de lo escrito. Tiene un sistema de pruebas unitarias con JUnit, un control de versiones CVS, integración con Ant, asistentes de creación de la estructura de proyecto (paquetes, clases...), refactorización, un sistema de compilación en tiempo real y un sistema de depuración, entre otras muchas funciones.

Actualmente está empleándose aún más si cabe por ser la principal plataforma de programación para aplicaciones, mediante la integración de un SDK.



Texto: [https://es.wikipedia.org/wiki/Eclipse_\(software\)](https://es.wikipedia.org/wiki/Eclipse_(software)) Imagen: Archivo personal.

4.2.- MySQL WorkBench

MySQL Workbench es una herramienta visual de diseño de bases de datos que integra desarrollo de software, administración de bases de datos, diseño de bases de datos, gestión y mantenimiento para el sistema de base de datos MySQL. Es el sucesor de DBDesigner de fabFORCE.net, y reemplaza el anterior conjunto de software, MySQL GUI Tools Bundle.

Características [\[editar\]](#)

Las características destacadas de MySQL Workbench 5.2 son:

- General:
 - Conexión y gestión de instancia de base de datos.
 - Asistente de elementos de acción.
 - Completamente compatible con [guiones](#) escritos en [Python](#) y [Lua](#).
 - Soporte para complementos personalizados
- Editor de SQL:
 - Exploración de esquema de objetos.
 - Resaltado de sintaxis en SQL y analizador de declaraciones.
 - Conjuntos de resultados múltiples, editables.
 - Colecciones de fragmentos SQL.
 - Tunelización de conexión por SSH.
 - Soporte Unicode.
- Modelado de datos:
 - Diagrama entidad relación.
 - Modelado visual con [arrastrar y soltar](#).
 - Ingeniería inversa a guiones SQL y base de datos en vivo.
 - Ingeniería directa a guiones SQL y base de datos en vivo.
 - Sincronización de esquema.
 - Impresión de modelos.
 - Importación desde fabFORCE.net DBDesigner4.
- Administración de base de datos:
 - Iniciar y detener instancias de base de datos.
 - Configuración de instancias.
 - Administración de cuentas en base de datos.
 - Exploración de instancias variables.
 - Exploración de ficheros de registros.
 - Exportación e importación masiva de datos.

https://es.wikipedia.org/wiki/MySQL_Workbench

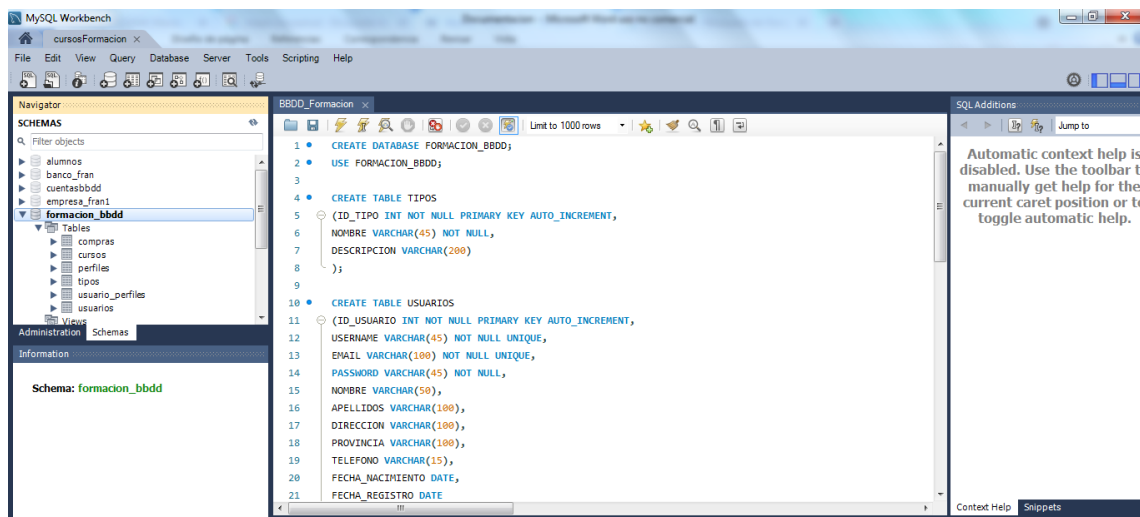


Imagen: Archivo personal.

4.3.- Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.

Visual Studio Code se basa en Electron, un framework que se utiliza para implementar Chromium y Node.js como aplicaciones para escritorio, que se ejecuta en el motor de diseño Blink. Aunque utiliza el framework Electron, el software no usa Atom y en su lugar emplea el mismo componente editor (Monaco) utilizado en Visual Studio Team Services (anteriormente llamado Visual Studio Online).



```
35
36 <article>
37   <header>
38     <h1>TFG FRANCISCO DE VICENTE</h1>
39   </header>
40
41   
42
43   <div class="content" style="text-align: justify;">
44     <p><b>Mirar al pasado hace replantearse tu futuro, y sólo dependerá de lo que hagas en tu presente.</b></p>
45     <p>Con estas palabras he elegido la presentación de mi página web porque sólo cuando vives ciertas situaciones de manera personal,
46     <p>El camino de la vida claro que no es nada fácil y aunque a veces pensemos que no podemos, tenemos que pensar que cada cambio que
47     <p>A veces nos cuesta mucho pedir ayuda, y hay que pensar y reconocernos que sólo con ayuda podemos continuar.</p>
48     <p>Hace dos años no sabía ni lo que era un código, "html" me sonaba a una firma de ropa. Cuando decidí realizar esta formación, empe
49     <p>Hoy no soy un gran experto, claro que no. Todavía me queda un largo recorrido por hacer, pero sí hace dos años me dicen que iba a
50     <p><b>A todos y cada uno de ellos, GRACIAS.</b></p>
51     
52   </div>
53 </article> <!-- /article -->
54
55 </section> <!-- / #main-content -->
56
```

Texto: https://es.wikipedia.org/wiki/Visual_Studio_Code Imagen: Archivo Personal.

4.4.- Git | GitHub

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora incluyendo coordinar el trabajo que varias personas realizan sobre archivos compartidos en un repositorio de código.

Al principio, Git se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir la interfaz de usuario o Front-End como Cogito o StGIT. Sin embargo, Git se ha convertido desde entonces en un sistema de control de versiones con funcionalidad plena. Hay algunos proyectos de mucha relevancia que ya usan Git, en particular, el grupo de programación del núcleo Linux. <https://es.wikipedia.org/wiki/Git>

4.5.- GitHub

GitHub es una forja (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador. El software que opera GitHub fue escrito en Ruby on Rails. Desde enero de 2010, GitHub opera bajo el nombre de GitHub. El código de los proyectos alojados en GitHub se almacena típicamente de forma pública.

<https://es.wikipedia.org/wiki/GitHub>

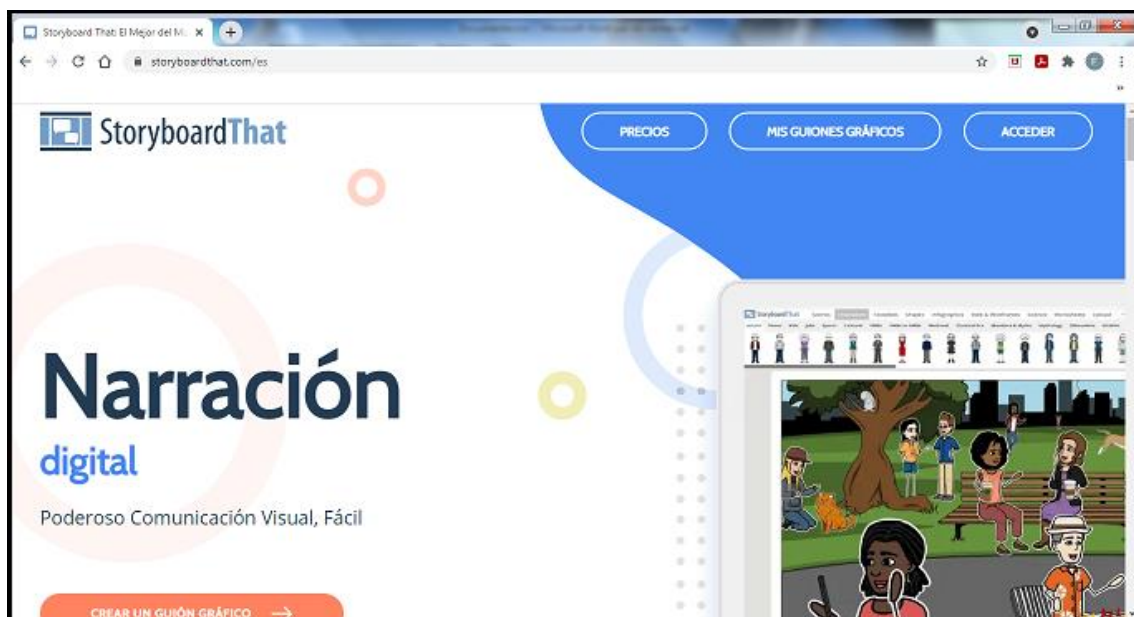
4.6.- Software Ideas Modeler

Software Ideas Modeler es una herramienta CASE y UML. El modelador admite los 14 tipos de diagramas especificados en UML. También admite, entre otros, los siguientes diagramas y estándares: Diagramas ER, BPMN 2.0, CMMN, SysML 1.5, ArchiMate 3, JSD, CRC, Diagramas de flujo, Diagrama de flujo de datos, Infografías, Wireframes, Mapas mentales, Historias de usuarios y Hojas de ruta.

https://en.wikipedia.org/wiki/Software_Ideas_Modeler

4.7.- StoryBoardThat

StoryboardThat, es una herramienta de forma gratuita con uso limitado con opción a uso ilimitado si abonas una cuota mensual. Esta herramienta, te permite realizar situaciones reales en las que un supuesto usuario hace en su día a día para entender mejor su situación y necesidades. Permite familiarizarte con tu cliente final.



5.1.- La Idea



El alumno, trata de pensar sobre qué hacer para su trabajo fin de grado (TFG). Le surgen varias ideas, entre las cuales se encuentran montar una página web que genere venta de entradas de ocio, otra idea que le sugiere es hacer una web a modo documental de animales con las curiosidades y los hábitats de los mismos. También le surge la idea de montar un blog de cocina donde poner sus recetas preferidas a modo biblioteca personal y así, una cantidad diversa de ideas. Finalmente opta por su idea presentada, en la que tras su experiencia profesional, considera montar una web de venta de cursos formativos en donde el usuario que la utilice, fuera como irse de compras. Una web en la que el usuario final busque su formación como el que busca pantalones hasta ver el que le gusta más y comprarlo, sin la presión de un/a agente comercial.

5.2.- Fichas de los Perfiles

Para poder hacer una comprensión mejor de a qué usuario quiere dirigirse, el alumno ha utilizado las técnicas empleadas en la asignatura Diseño de Interfaces Web y ha creado una ficha de un perfil muy común que encontró en sus años de trabajo. Este perfil, son alumnos comprendidos entre 30 y 50 años que están agotados de llevar una vida rutinaria y monótona, y quieren promocionar en sus puestos laborales así como también poder tener la oportunidad de buscar un nuevo empleo si no cumplen con sus objetivos.

INFORMA&FORMA-T

Marca tu Futuro

A cartoon illustration of a woman with blonde, curly hair, wearing a purple long-sleeved top and grey trousers. She is standing with her hands at her sides.

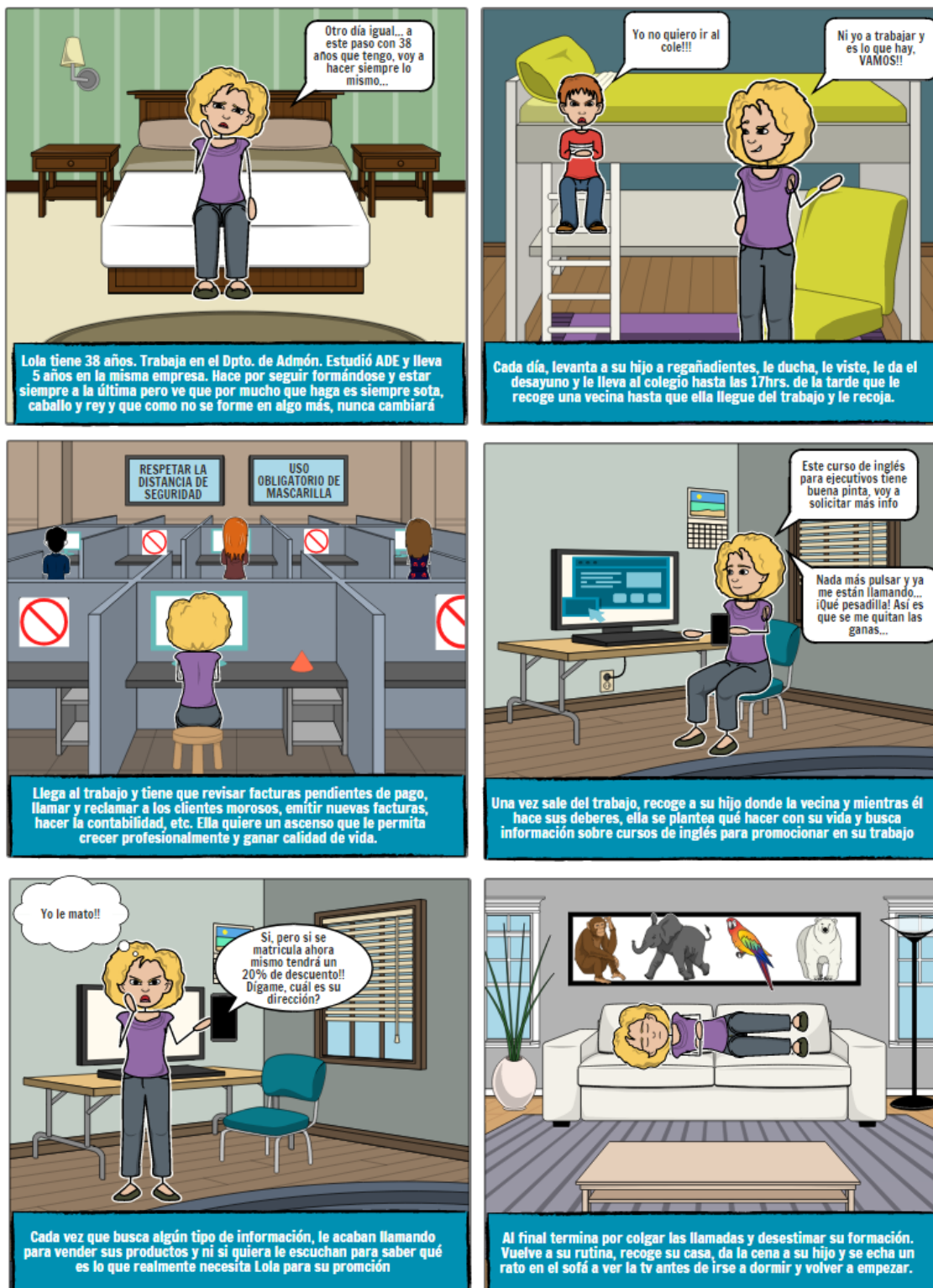
Nombre: Lola Martínez
Edad: 38 años
Profesión: Administrativa
Estado civil: Separada
Hijos: 1 hijo (8 años)
Hoobies: Yoga, lectura, pasear, ir a la playa, estar con su hijo.

Lola quiere promocionar en su trabajo porque lleva años en la misma empresa y haciendo las mismas funciones. Para promocionar, necesita tener un buen nivel de inglés. Promocionar le conllevaría más salario y más tiempo libre para su hijo. Lo que le frena a Lola para hacer la formación, es que cada vez que busca información le acosan a llamadas y no tiene ganas de estar atendiendo a nadie y mantener conversaciones incómodas si no le interesa lo que le ofrecen.

5.3.- StoryBoards

El guion gráfico (storyboard) es un conjunto de ilustraciones que aparecen en secuencia y que se utilizan como guía para entender una historia, para previsualizar una animación o para seguir la estructura de un supuesto antes de realizarlo.

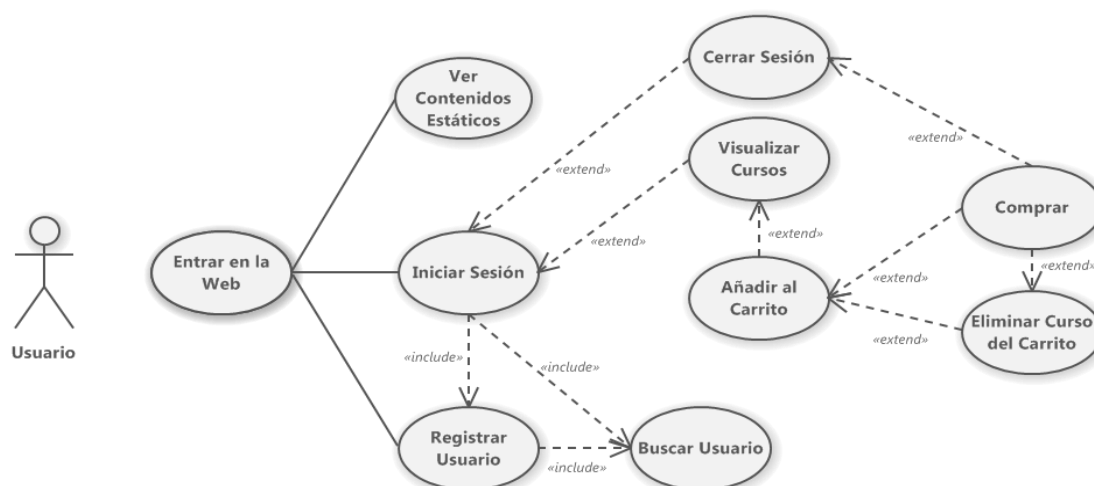
Su función es narrar la historia con pocas imágenes secuenciadas y acompañadas de textos, y sirve para el detalle de los personajes y un breve recuento de lo que está sucediendo, sirviendo de guía para todos los que participan.



5.4.- Casos de Uso

Una vez el alumno ha valorado los diferentes tipos de usuario a los que va dirigido su página, es momento de empezar a crearla. Antes de nada, se valora hacer el caso de uso de aquellos que utilicen la aplicación web.

Un caso de uso es la descripción de una acción o actividad. Un diagrama de caso de uso es una descripción de las actividades que deberá realizar alguien o algo para llevar a cabo algún proceso. Los personajes o entidades que participarán en un diagrama de caso de uso se denominan actores. En el contexto de ingeniería del software, un diagrama de caso de uso representa a un sistema o subsistema como un conjunto de interacciones que se desarrollarán entre casos de uso y entre estos y sus actores en respuesta a un evento que inicia un actor principal. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas.



Herramienta realizada con Software Ideas Modeler

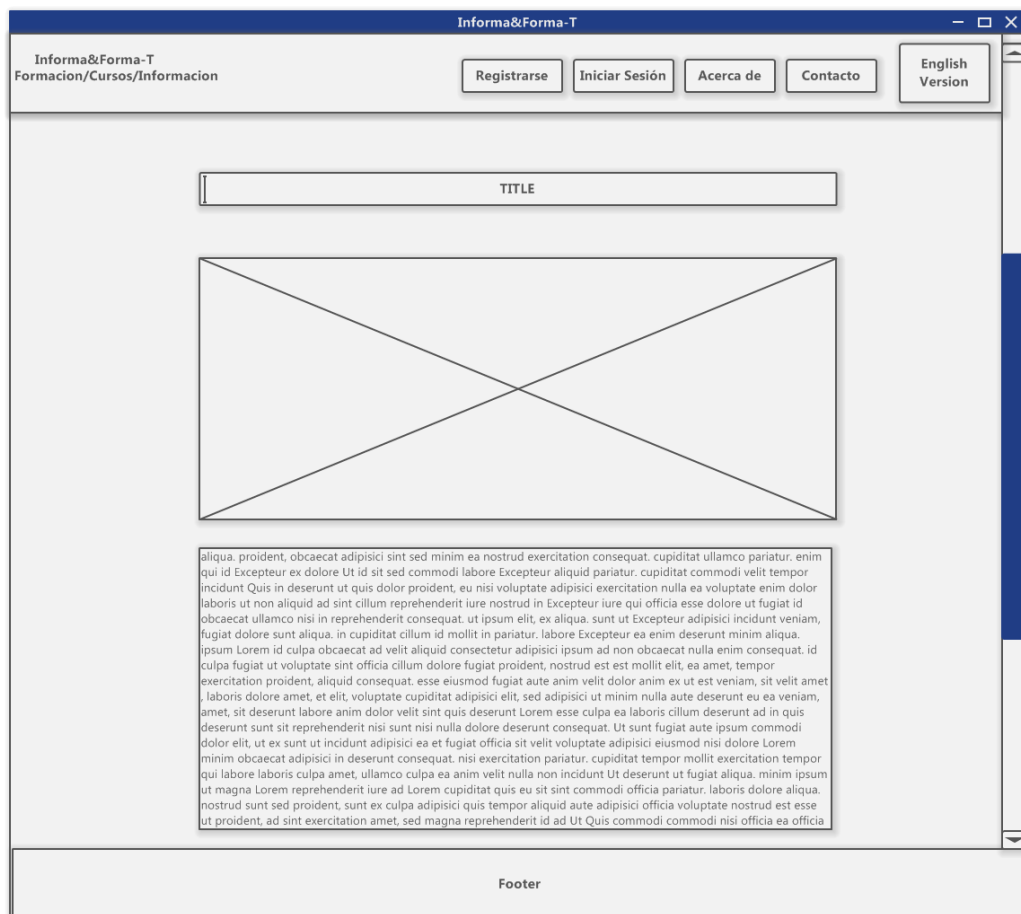
- 1.- Dicho esto, en el caso de uso que vemos, muestra como el usuario lo primero que tiene que hacer es acceder a la web.
- 2.- El usuario una vez acceda, tiene dos posibilidades. Primero ver los contenidos estáticos, que simplemente muestran la información de la página, cuáles son sus objetivos, qué puede encontrar en ella, quiénes la forman, cómo contactar con ellos, etc. Por otro lado, el usuario podrá iniciar sesión. Eso sí, para poder iniciar sesión, el usuario deberá estar registrado y acceder con su usuario y contraseña.
- 3.- Cuando haya iniciado sesión, el usuario podrá ver los cursos disponibles que se ofrecen, podrá ver sus contenidos, a quién va dirigido, sus salidas profesionales, las horas lectivas, precio, etc.
- 4.- Podrá añadir a su carro de compra aquellos cursos que le interesen, así como eliminarlos y finalmente, comprarlos.
- 5.- Una vez realizada la compra, podrá seguir navegando en la web o bien, cerrar la sesión.

5.5.- WireFrame

Ahora que el alumno tiene claro que puede hacer sus actores en la página web, tenemos que volver al módulo de Diseño de Interfaces Web para saber cómo va a crear el contenido de su página, qué aspecto le quiere dar, cuál va a ser su tipografía, estilos, etc.

Un website wireframe, también es conocido como un esquema de página o plano de pantalla, es una guía visual que representa el esqueleto o estructura visual de un sitio web. El wireframe esquematiza el diseño de página u ordenamiento del contenido del sitio web, incluyendo elementos de la interfaz y sistemas de navegación, y cómo funcionan en conjunto. Usualmente este esquema carece de estilo tipográfico, color o aplicaciones gráficas, ya que su principal objetivo reside en la funcionalidad, comportamiento y jerarquía de contenidos. En otras palabras, se enfoca en “qué hace la pantalla, no cómo se ve. Los wireframes se enfocan en:

- 1.- Los tipos de información que será mostrada.
- 2.- La cantidad de las funciones disponibles.
- 3.- Las prioridades relativas de la información y las funciones.
- 4.- Las reglas para mostrar ciertos tipos de información.
- 5.- El efecto de los distintos escenarios en la pantalla.



WireFrame página inicio (index) creada con Software Ideas Modeler

Informa&Forma-T
Formacion/Cursos/Informacion

Registrarse Iniciar Sesión Acerca de Contacto English Version

TITLE

LOGO

Email

Password

Nombre

Apellidos

Dirección

Provincia

Teléfono

Fecha Nac.

Registrarse

Footer

WireFrame página de registro de usuario

Informa&Forma-T
Formacion/Cursos/Informacion

Registrarse Iniciar Sesión Acerca de Contacto English Version

TITLE

Photo

Nombre

Corre Electrónico

Mensaje

Enviar

Footer

WireFrame página de contacto

5.6.- Guía de Estilos

Una guía de estilo, está compuesta por un conjunto de criterios preceptuados, por normas para el diseño y la redacción de documentos, ya sea para el uso general, o para ser utilizados por los redactores de periódicos u otras organizaciones que también publican textos. La implementación de un manual de estilo permite publicar siguiendo una uniformidad de estilo. Los manuales de estilo son frecuentes en el uso general y especializado, en medios escritos, orales y gráficos. El manual de estilo se compone tanto de normas lingüísticas, como de estilo, para que el mensaje sea más coherente, eficaz y correcto. [...]

https://es.wikipedia.org/wiki/Manual_de_estilo



Logo

Texto: Century Gothic
Color: #FE049C
Tamaño Principal: 510 * 95 px



Fondo de página
#666
Line-height: 1.5em



Icono Tamaño: 54 * 57 px



Barra de Menú

Texto: Century Gothic
Background color: #333
Texto-Color: #FFFFFF
Header: hover: #FE049C



Buttom Background color: #333

Guía de Estilos realizada con www.canva.com

Guía completa: https://www.canva.com/design/DAEfBTqolb4/Jcxj78zJr5vcHWBHD_ftdA/edit

5.7.- Bases de Datos

Los usuarios introducen datos, las aplicaciones analizan, procesan y realizan cálculos con esos datos. Y por último se produce la salida de resultados.

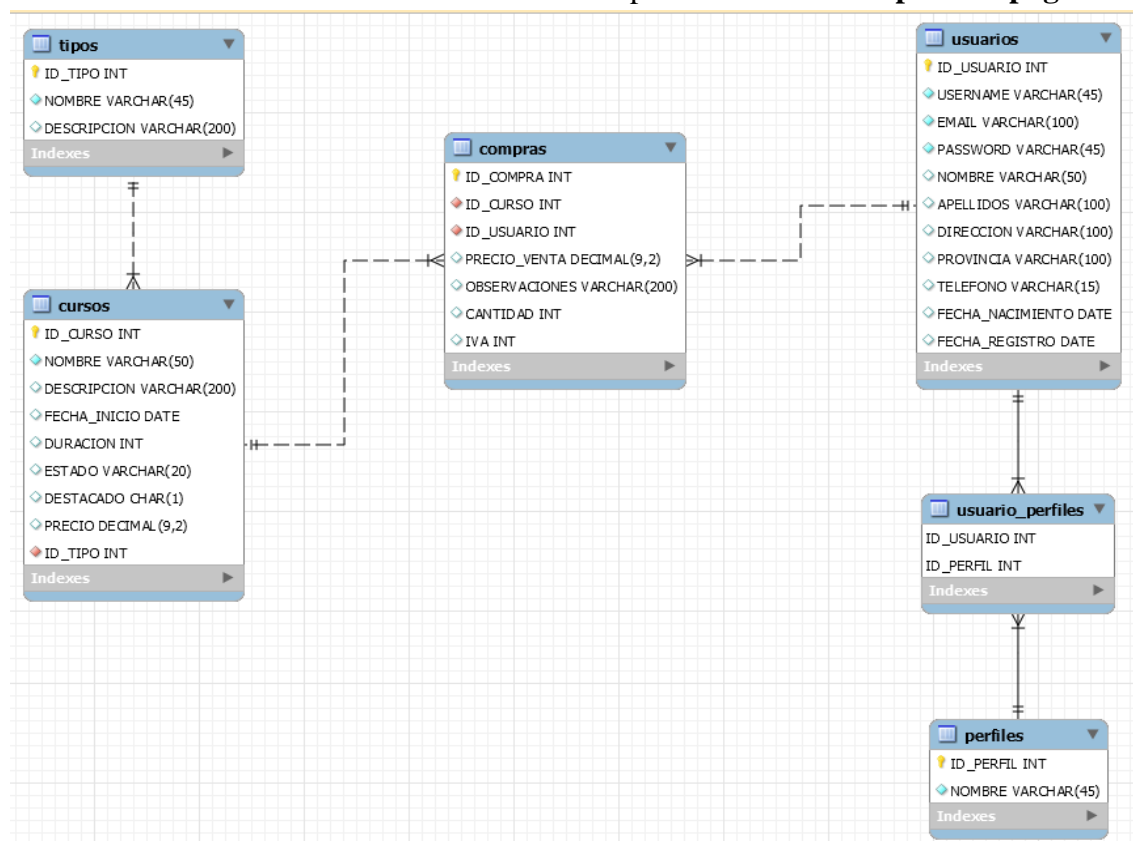
Toda esta información se debe guardar en algún lugar. Las técnicas empleadas para almacenar datos son sumamente importantes para la velocidad de acceso y recuperación de los mismos. El método más utilizado es el almacenamiento en bases de datos.

Las bases de datos proporcionan las siguientes funciones:

- Almacenamiento
- Organización
- Acceso
- Seguridad
- Manipulación
- Salvaguarda

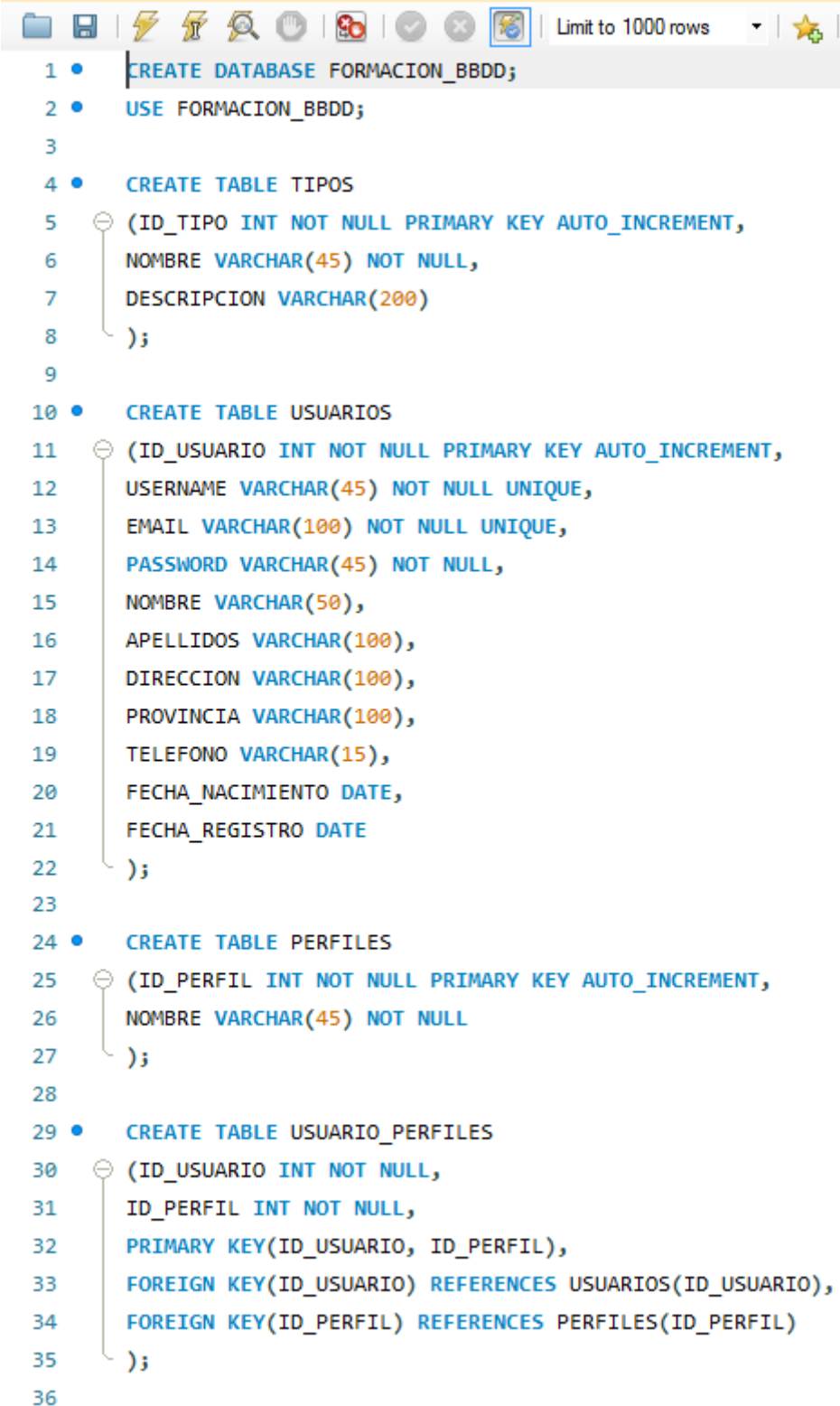
Instituto Tecnológico Edix <https://institutotecnologico.edix.com/>

Por todo ello, es buena práctica tener bien estructurada una base de datos capaz de realizar estas funciones. Es preciso saber, antes de poner en marcha la aplicación, qué y cuántas bases de datos se van a necesitar en una primera estancia. **Explica en pág. 09**



Esquema realizado mediante la herramienta de MySQL WorkBench

Para la creación de esta base de datos relacional, el alumno ha utilizado la herramienta de MySQL WorkBench. A continuación, se muestra una parte de cómo realiza el script para su creación.



```
1 • CREATE DATABASE FORMACION_BBDD;
2 • USE FORMACION_BBDD;
3
4 • CREATE TABLE TIPOS
5   (ID_TIPO INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
6    NOMBRE VARCHAR(45) NOT NULL,
7    DESCRIPCION VARCHAR(200)
8   );
9
10 • CREATE TABLE USUARIOS
11   (ID_USUARIO INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
12    USERNAME VARCHAR(45) NOT NULL UNIQUE,
13    EMAIL VARCHAR(100) NOT NULL UNIQUE,
14    PASSWORD VARCHAR(45) NOT NULL,
15    NOMBRE VARCHAR(50),
16    APELLIDOS VARCHAR(100),
17    DIRECCION VARCHAR(100),
18    PROVINCIA VARCHAR(100),
19    TELEFONO VARCHAR(15),
20    FECHA_NACIMIENTO DATE,
21    FECHA_REGISTRO DATE
22   );
23
24 • CREATE TABLE PERFILES
25   (ID_PERFIL INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
26    NOMBRE VARCHAR(45) NOT NULL
27   );
28
29 • CREATE TABLE USUARIO_PERFILES
30   (ID_USUARIO INT NOT NULL,
31    ID_PERFIL INT NOT NULL,
32    PRIMARY KEY(ID_USUARIO, ID_PERFIL),
33    FOREIGN KEY(ID_USUARIO) REFERENCES USUARIOS(ID_USUARIO),
34    FOREIGN KEY(ID_PERFIL) REFERENCES PERFILES(ID_PERFIL)
35   );
36
```

El código completo en el reposito de GitHub del alumno https://github.com/FranDeVicente/TFG_2-DAW_ITEdix

5.8.- HTML

HTML (HyperText Markup Language) es un lenguaje de etiquetas usado por los diseñadores y programadores para crear páginas web.

Se ha convertido en el estándar de visualización de la información, siendo la vista de muchas aplicaciones, tanto de las páginas web más tradicionales como de aplicaciones de servidor para servicios empresariales.

Actualmente, los lenguajes de programación más modernos como Java, .NET, Python o PHP disponen de librerías y tienen interiorizado este sistema de representación para mostrar la capa de vista de los programas.

Un documento HTML está formado por su contenido y etiquetas que delimitan sus partes para dar formato.

Por todo ello, el alumno comienza a realizar el código HTML para su página principal (index) y dar contenido a su aplicación web. Para la realización de esta inicialización, utiliza la herramienta Visual Studio Code.

A continuación, se muestra parte del primer boceto del código HTML respecto a la página principal de la aplicación web (index), pudiéndose ver el código completo en el repositorio git del alumno.

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset='utf-8'>
5      <meta http-equiv='X-UA-Compatible' content='IE=edge'>
6      <title>Informa&Forma-T</title>
7      <link rel="icon" type="image/x-icon" href="IMAGENES\lcdn.ico" href="IMAGENES\LogoSimb.png">
8      <meta name='viewport' content='width=device-width, initial-scale=1'>
9      <link rel="stylesheet" type="text/css" href="CSS/estilo.css">
10     <script type="text/javascript" src="JS/javaScript.js"></script>
11     <script src='main.js'></script>
12 </head>
13
14 <body>
15     <header id="main-header">
16         <a id="logo-header" href="index.html">
17             <span class="site-name">Informa&Forma-T</span>
18             <span class="site-desc">Formacion / Cursos / Informacion</span>
19         </a>
20         <nav>
21             <ul>
22                 <li><a href="iniciaSesion.html">Iniciar Sesión</a></li>
23                 <li><a href="acercaDe.html">Acerca de</a></li>
24                 <li><a href="contacto.html">Contacto</a></li>
25             </ul>
26         </nav>
27     </header>
```

Maquetación creada con la herramienta Visual Studio Code

5.9.- CSS

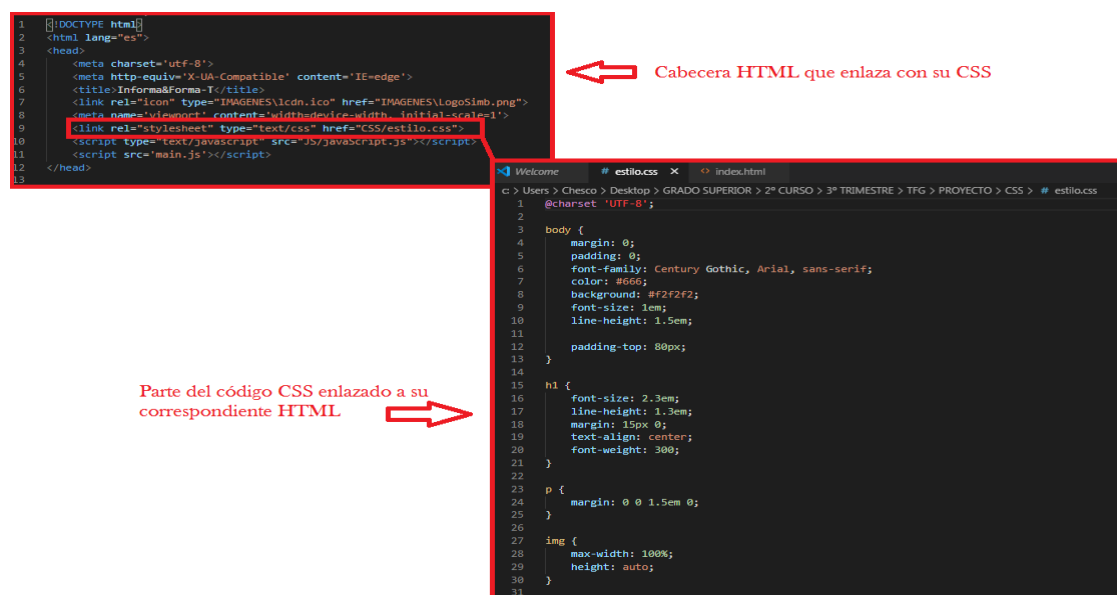
Con las hojas de estilo no solo podremos hacer que nuestros documentos se vean del modo que deseemos, también definiremos la estructura de nuestra página, menús, columnas, etc.

Además, hoy en día, debemos hacer que nuestras páginas se adapten a múltiples dispositivos, y CSS será un gran protagonista en esta tarea.

Los elementos son configurados según unas reglas no cambiantes (estáticas), sin la posibilidad de condicionar su estado.

Características de un CSS ESTÁTICO	CSS DINÁMICO
<p>Cada elemento del estilo tiene sus atributos estáticos, por lo que no pueden modificarse en tiempo de carga o de ejecución.</p> <p>Los atributos no pueden ser calculados en tiempo de carga o ejecución.</p> <p>Si varios elementos tienen atributos iguales, se definen para cada uno de ellos.</p> <p>Para cambiar los estilos se accede al fichero CSS directamente y se cambia el valor del atributo.</p> <p>Aunque CSS estático, podremos utilizar algunas propiedades como :hover, :focus, etc., que aunque generan algo de dinamismo al documento, se sigue considerando estático.</p> <p>https://institutotecnologico.edix.com/</p>	<p>Para facilitar la creación de los estilos CSS existen una serie de herramientas que permiten manejar los estilos mediante un pre-procesamiento o en tiempo de ejecución. Con esto se consigue mejorar la estructura y eficiencia de la hoja de estilos.</p> <p>Para crear este tipo de CSS es necesario usar:</p> <p>Herramientas como Stylus, LESS o SASS. Estas herramientas se constituyen como un lenguaje de programación que realiza un pre-procesamiento de las hojas de estilos, de manera que antes de ser cargadas en la página se han calculado y definido.</p>

A continuación, se muestra parte de la estructura inicial del CSS creado, pudiéndose visualizar el código completo en el repositorio del alumno.



5.10.- JavaScript

JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas y JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

Para la aplicación web, el alumno ha utilizado este lenguaje para la validación del formulario de registro de usuario. Se muestra un fragmento del código utilizado en primera estancia, pudiéndose visualizar por completo en el repositorio GitHub.

```

1 //generamos la función mediante jquery para que ejecute JS una vez esté cargada toda la página
2 $(document).ready(function(){
3     $('#order').click(validationForm); //llamamos a la función de validar formulario
4 });
5
6 //Hacemos la función para validar el formulario (del input tipo texto)
7 function validationForm(){
8
9     //validamos el campo email.
10    if( $("#email").val().trim().length == 0){
11        alert("[ERROR] Debes introducir un correo electrónico");
12        //Evitamos que se ejecute el evento de mandar el formulario
13        return false
14    }
15    //expresión regular para el formato email
16    if ($("#email").val().match(/^([a-zA-Z0-9\._-])+@[a-zA-Z0-9-]{2,}[.][a-zA-Z]{2,4}$/)){//metemos la expresion regular
17    } else {
18        alert("[ERROR] El correo electrónico no es válido.");
19        return false;
20    }
21
22    //decimos que si el campo nombre está vacío..
23    if( $("#name").val().trim().length == 0){
24        alert("[ERROR] Debes introducir tu nombre"); //..entonces muestre una alerta de error
25        //Evitamos que se ejecute el evento de mandar el formulario
26        return false
27    }
28
29    //decimos que si el campo nombre está vacío..
30    if( $("#apellidos").val().trim().length == 0){
31        alert("[ERROR] Debes introducir tus apellidos"); //..entonces muestre una alerta de error
32        //Evitamos que se ejecute el evento de mandar el formulario
33        return false
34    }
35 }

```

Para ello, ha sido necesario instalar la librería JS <https://code.jquery.com/jquery-3.5.1.min.js>

The screenshot shows a web application interface. At the top left, there is a header with the text "Informa&Forma-T" and "Formación / Cursos / Información". At the top right, there are links for "Iniciar Sesión", "Acerca de", and "Contacto". In the center, there is a registration form with the following fields: "Email:", "Password:", "Nombre:", "Apellidos:", "Dirección:", "Provincia:" (with a dropdown menu showing "Seleccione Provincia"), "Teléfono:", "Fecha de Nacimiento:" (with a date picker set to "dd/mm/aaaa"), and an "Enviar" button. Above the form, there is a modal box with the title "Esta página dice" and the message "[ERROR] Debes introducir un correo electrónico". There is an "Aceptar" button next to the error message.

5.11.- Modelo, Vista, Controlador (MVC)

Modelo-vista-controlador es un patrón de arquitectura de software, que separa los datos y principalmente lo que es la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

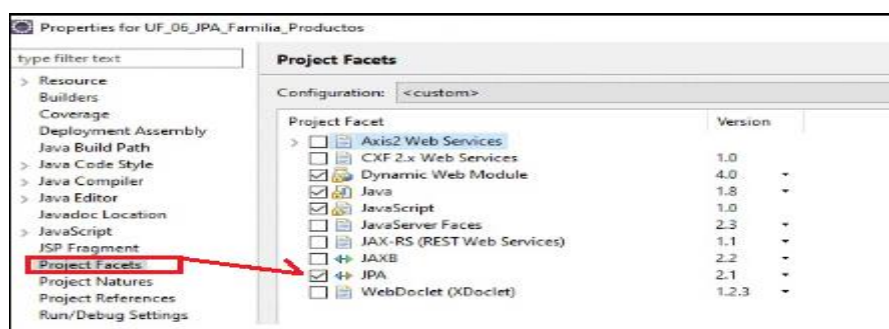
<https://es.wikipedia.org/wiki/Modelo%20%80%93vista%20%80%93controlador>

Para la presentación de la aplicación, el alumno realiza un proyecto “**Maven**” creando consigo una extensión del proyecto como “war”. El proyecto en Maven es muy útil para que al ejecutarse desde cualquier Eclipse, pueda ejecutar todo el código del mismo ya que actualiza automáticamente sus librerías que, en caso de no disponer de ellas, las buscará vía internet para actualizarlas.

El proyecto será creado como “**Dinamic Web Project**”. Utiliza JPA como framework en su lenguaje de programación, el cual internamente ejecuta JDBC que maneja datos relacionales en aplicaciones usando la Plataforma Java en sus ediciones Standard (Java SE) y Enterprise (Java EE). JPA origina la persistencia en este contexto que cubre tres áreas:

- 1.- La API en sí misma, definida en el paquete javax.persistence
- 2.- El lenguaje de consulta Java Persistence Query Language (JPQL).
- 3.- Metadatos objeto/relacional.

El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos (siguiendo el patrón de mapeo objeto-relacional), como sí pasaba con EJB2, y permitir usar objetos regulares (conocidos como POJO).



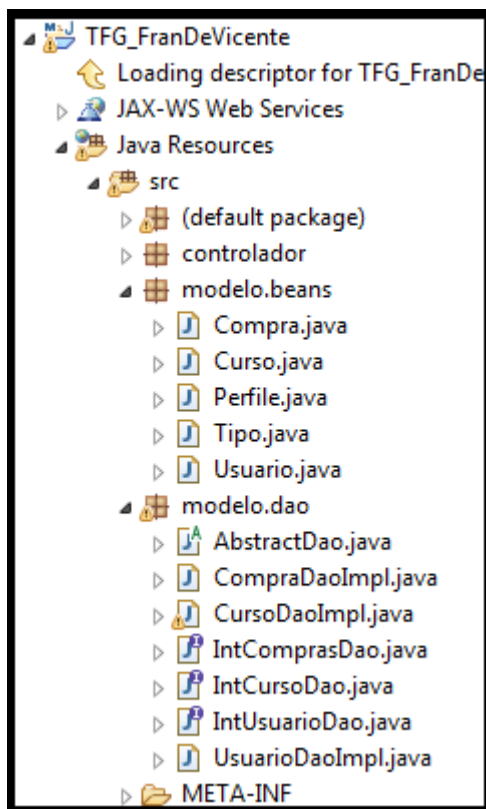
Por otro lado, se ha creado la conexión a la BBDD la cual será utilizada MySQL en la que incluye su driver conector correspondiente, descargado de la página oficial <https://dev.mysql.com/downloads/connector/j/>

Una vez está el driver descargado, sólo hay que poner la BBDD que se va a utilizar, junto con el nombre de usuario y contraseña asociado a MySQL Workbench. Por otro lado, se modifica el documento “Persistence.xml” del proyecto así como el “pom.xml” para su correcto funcionamiento, según establece sus dependencias <https://mvnrepository.com/artifact/org.eclipse.persistence/org.eclipse.persistence.jpa/2.7.7>.

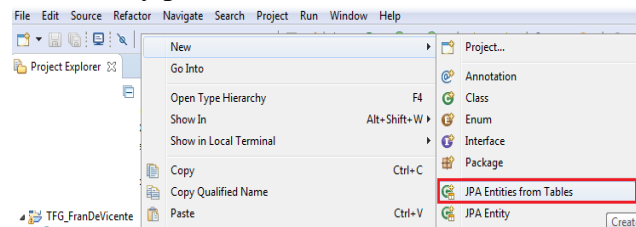
Puede verse la instalación completa y detallada, en la clase impartida en la asignatura “Desarrollo Web Entorno Servidor” en el siguiente enlace: <https://institutotecnologicotelefonica.adobeconnect.com/pcnkyvo4pbcn/>

A continuación se explica el modelo vista controlador realizado en el proyecto.

5.11.1.- Modelo^{Beans}



Para la parte del modelo, el alumno separa en clases con extensión “.beans” y “.dao” en las que los beans serán las clases utilizadas y asociadas a la Base de Datos. Cada tabla, lleva consigo una clase, las cuales están conectadas a dicha BBDD. Para ello, se realiza un **mapeo** de las clases y pueda funcionar.



En las migraciones de las claves primarias, en relación N:M, migramos siempre la clave primaria (PK) de uno, como la clave ajena del otro (FK). En las clases, migramos el objeto entero de manera unidireccional, pudiéndose hacer bidireccional si fuera preciso.

Todas las clases tendrán serializable, también la colección de propiedades del java.util. Maneja “identity” para los auto-incrementos de las PK.

Hace las anotaciones correspondientes, tales como:

@**Entity** para indicar que la clase está en JPA.

@**Table**, indica a la tabla que está asociada dicha clase.

@**NamedQuery** para la query que usará en el llamamiento a la tabla.

@**Id**, el cual sólo una columna podrá tener este atributo. Corresponderá a la PK normalmente.

@**GeneratedValue** corresponde al auto-increment de la base de datos.

@**Column** indicará a la columna que hace referencia.

@**ManyToOne** la relación, en éste caso uno a muchos, que lleva las tablas entre sí.

@**JoinColumn** como anotación de los joins que une las tablas.

Cuando se realiza el mapeo de las clases, genera automáticamente los métodos “**Getters and Setters**” por lo que sólo deberemos incluir los métodos “**hashCode()** y **equals()**” y el método “**toString()**” que nos viene bien para ver todo mediante las pruebas que realicemos en los test a modo prueba. No podemos olvidar que el mapeo genera un **constructor** por defecto, por lo que el alumno genera el constructor con todo a su interés.

5.11.2.- Modelo^{Dao}

Prácticamente todas las aplicaciones de hoy en día, requiere acceso al menos a una fuente de datos, dichas fuentes son por lo general base de datos relacionales, por lo que muchas veces no tenemos problema en acceder a los datos, sin embargo, hay ocasiones en las que necesitamos tener más de una fuente de datos o la fuente de datos que tenemos puede variar, lo que nos obligaría a refactorizar gran parte del código. Para esto, tenemos el patrón Arquitectónico Data Access Object (DAO), el cual permite separar la lógica de acceso a datos de los Bussines Objects u Objetos de negocios, de tal forma que el DAO encapsula toda la lógica de acceso de datos al resto de la aplicación.

Una de las grandes problemáticas al momento de acceder a los datos, es que la implementación y formato de la información puede variar según la fuente de los datos. Implementar la lógica de acceso a datos en la capa de lógica de negocio puedes ser un gran problema, pues tendríamos que lidiar con la lógica de negocio en sí, más la implementación para acceder a los datos, adicional, si tenemos múltiples fuentes de datos o estas pueden variar, tendríamos que implementar las diferentes lógicas para acceder las diferentes fuentes de datos, como podrían ser: bases de datos relacionales, No SQL, XML, archivos planos, Webservices, etc).

Dado lo anterior, el patrón DAO propone separar por completo la lógica de negocio de la lógica para acceder a los datos, de esta forma, el DAO proporcionará los métodos necesarios para insertar, actualizar, borrar y consultar la información; por otra parte, la capa de negocio solo se preocupa por lógica de negocio y utiliza el DAO para interactuar con la fuente de datos.

Mediante el patrón Abstract Factory podemos definir una serie de familias de clases que permitan conectarnos a las diferentes fuentes de datos. Para esto, examinaremos un sistema de autenticación de usuarios, el cual puede leer los usuarios en una base de datos o sobre un XML, adicional, el sistema generara registros de login que podrán ser utilizados para auditorias.

En este punto, los DAO ya implementan una interfaz común, lo que permite intercambiar la implementación sin afectar al Business Object. Sin embargo, ahora solo falta resolver la forma en que el Business Service obtendrá la familiar de interfaces, es por ello que deberemos crear un Factory para cada familia de interfaces.

Libro: Introducción a los patrones de diseño , por Óscar Blancante. <https://reactiveprogramming.io/books/design-patterns/es>

```
1 package modelo.dao;
2
3 import javax.persistence.EntityManager;
4
5 //Hacemos la clase abstracta, de ésta manera no hace falta que sus métodos los sean.
6 public abstract class AbstractDao {
7
8     protected EntityManagerFactory emf; //de la librería persistence
9     protected EntityManager em; //el EntityManager que llevará los métodos del persiste, el find, queries y demás
10    protected EntityTransaction tx; //hereda también del java persistence para las transacciones
11    protected Query query; //para las queries
12    protected String sql; //para las jpql el string, que llamaremos sql
13
14    public AbstractDao() {
15        //llamamos a la clase persistence y su método, al que le pasaremos un String con la unidad de persistencia.
16        emf = Persistence.createEntityManagerFactory("TFG_FranDeVicente"); //le pasamos el nombre del proyecto para que coja la conexión
17        em = emf.createEntityManager(); //crea el entityManager y lo devuelve
18        tx = em.getTransaction(); //Devuelve la transacción.
19    }
20 }
```

Clase abstracta a la que hace llamamiento las implementaciones del modelo DAO para su conexión a la BBDD

```
IntUsuarioDao.java
1 package modelo.dao;
2
3 import java.util.List;
4
5
6
7 public interface IntUsuarioDao {
8
9     int altaUsuario (Usuario usuario);
10    List<Usuario> findByUser(String usuario, String password);
11
12 }
13
```

Interfaz Clase Usuario

```

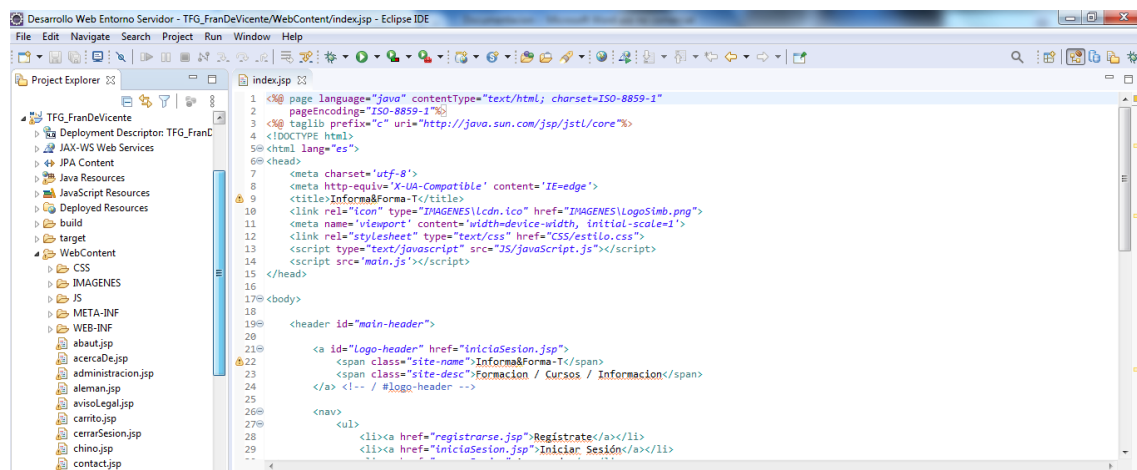
1 UsuarioDaoImpl.java
4 import java.util.List;
7
8 public class UsuarioDaoImpl extends AbstractDao implements IntUsuarioDao{
9
10 @Override
11 public int altaUsuario(Usuario usuario) {
12     int filas = 0;
13     try {
14         tx.begin();
15         em.persist(usuario);
16         tx.commit();
17         filas = 1;
18     } catch (Exception e) {
19         e.printStackTrace();
20     }
21     return filas;
22 }
23
24 @SuppressWarnings("unchecked")
25 @Override
26 public List<Usuario> findByUser(String usuario, String password) {
27     sql = "SELECT u FROM Usuario u WHERE u.username =:cod1 AND u.password =:cod2";
28     query = em.createQuery(sql);
29     query.setParameter("cod1", usuario);
30     query.setParameter("cod2", password);
31     return query.getResultList();
32 }
33
34 }

```

Parte de la Clase de Implementación de la Interfaz. Extiende de la clase abstracta e implementa la interfaz.

5.11.3.- Vista

La Vista, presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario), por tanto requiere de dicho 'modelo' la información que debe representar como salida.



Modelo Vista TFG del alumno

Si bien en un primer modelo de Front-End, la vista estática la creábamos mediante HTML y las hojas de estilo en cascada CSS, en la parte Back-End la vista forma a pasar a ser dinámica y para ello las realizamos mediante JSP y CSS.

Como hemos visto, se les llama dinámicas porque su aspecto cambia en función de los datos de la aplicación. Este tipo de vistas se deben generar mediante páginas JSP, ya que han de incluir cierta lógica que construya dinámicamente la parte de la respuesta que depende de los datos de aplicación.

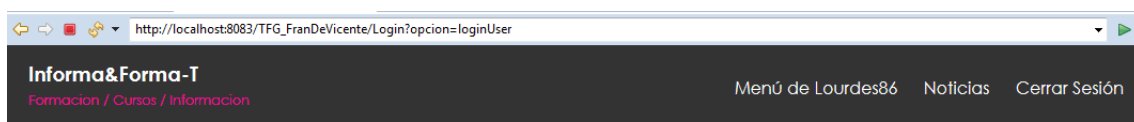
Las páginas JSP de una vista pueden ser construidas, tanto con scriptlets Java, como con acciones JSTL y expresiones EL.

Al igual que en el caso de las vistas estáticas, todos los enlaces y formularios incluidos en la página JSP apuntarán al servlet controler, enviando en la URL el correspondiente valor del parámetro operación.

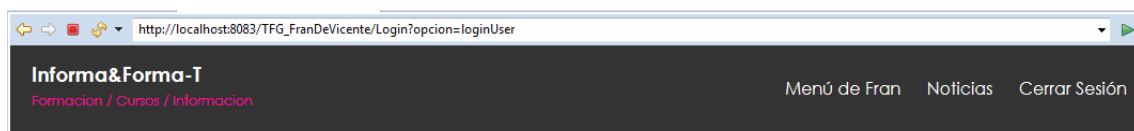
```
26<
27<
28<
29<
30<
31<
32<
33<
34<
<nav>
  <ul>
    <li><a href="carrito.jsp"><p>Menú de <%= (String)session.getAttribute("userName") %></p></a></li>
    <li><a href="noticias.jsp">Noticias</a></li>
    <li><form action="Login?opcion=cerrarSesion" method="post">
      <a id="iniSesion" href="cerrarSesion.jsp">Cerrar Sesión</a>
    </form></li>
  </ul>
</nav><!-- / nav -->
```

Vemos como una vez el usuario se loguea, mediante una acción JSTL, cogemos el atributo de la tabla que nos interese y dicha acción, mostrará el resultado dependiendo de quién inicie la sesión.

Inicio de sesión de un usuario aleatorio:



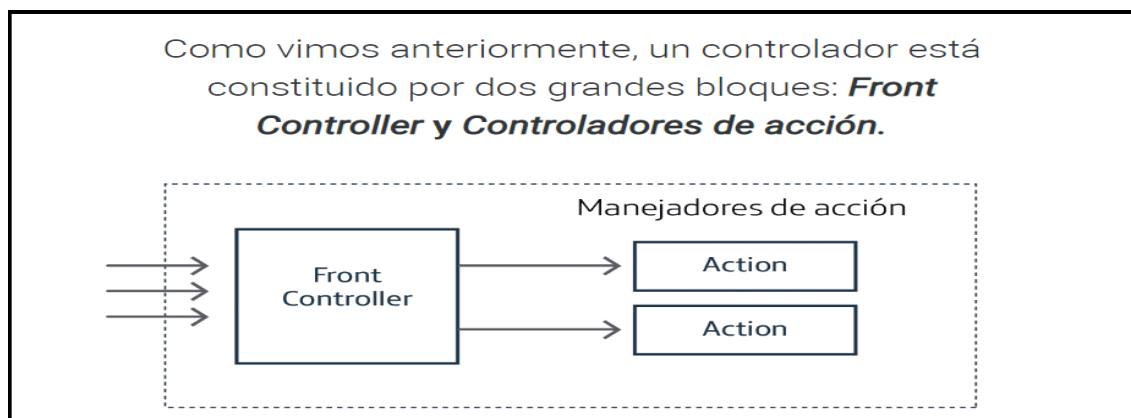
Inicio de sesión de un segundo usuario:



Vemos como gracias a las acciones JSTL, podemos personalizar cada registro de entrada en la aplicación web.

5.11.4.- Controlador

El controlador es el punto central de la aplicación, ya que gobierna el funcionamiento del resto de los bloques.



Front Controller, es la parte del controlador a la que se dirigen todas las peticiones desde la capa cliente. Una vez determinada la operación a realizar, transfiere el control al controlador de acción correspondiente.

El front controller se implementa mediante un servlet. Existen varias formas en las que este servlet puede determinar qué petición ha provocado su ejecución, que es lo mismo que determinar qué operación debe realizar.

Uno de estos mecanismos para determinar la operación a realizar, es enviar un parámetro en la petición que tenga diferentes valores para cada enlace o un formulario que apunte al servlet. Evaluando el parámetro mediante una instrucción alternativa, el servlet decide a dónde transferir la petición.

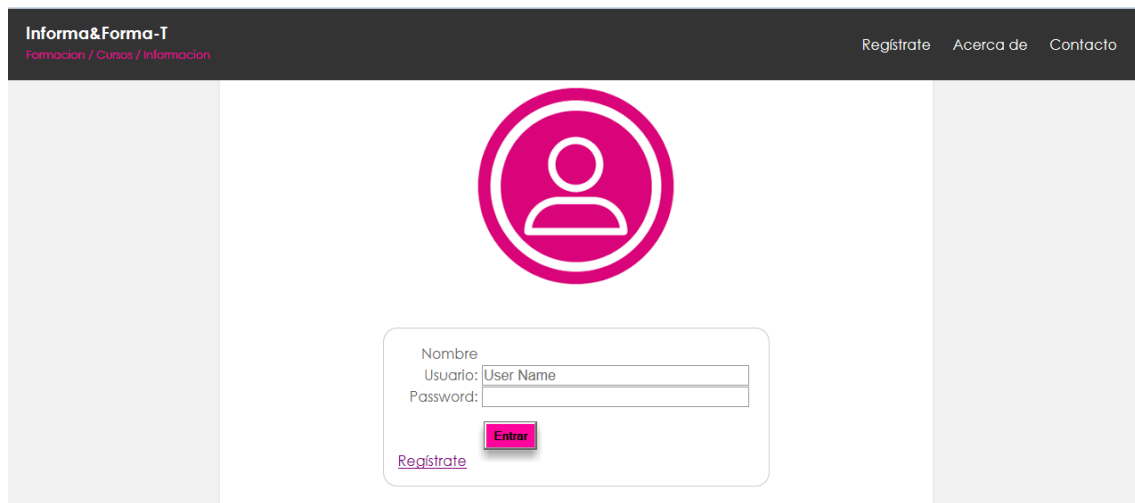
Controladores de acción. Cuando un servlet front controller determina la operación a realizar, transfiere el control al controlador de acción encargado de esta tarea. Estos controladores de acción son también servlets, que reciben los parámetros de la petición y realizan las correspondientes llamadas a los métodos del modelo.

Si alguno de estos métodos devuelve un resultado que debe ser visualizado en una página, el controlador de acción almacenará este dato en un atributo de petición, sesión o aplicación, según corresponda, para después transferir el control de la petición a la página JSP encargada de recoger esta información y volcarla en la página de respuesta.

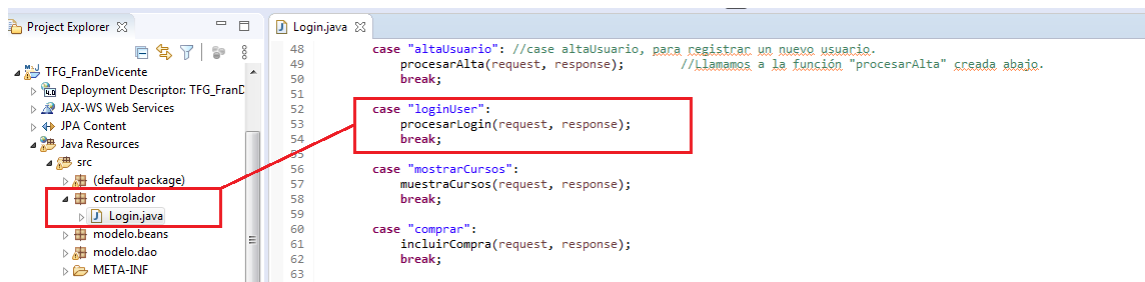
Es posible que alguna de las peticiones que lleguen al front controller no requieran la ejecución de un controlador de acción, pues simplemente se trata de acceder a una determinada página HTML de la aplicación.

Instituto Tecnológico Edix

A continuación se muestra el formulario de logueo para iniciar sesión del usuario.



Este formulario está enviando una petición al servidor el cual, Para estos casos, el front controller transferirá directamente la petición a la página correspondiente.



Una vez llamada a la petición, esta ejecuta su función correspondiente:

```
136 protected void procesarLogin(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
137     HttpSession miSesion = request.getSession();
138     String mensaje = null;
139     String usuario = request.getParameter("userName");
140     String password = request.getParameter("password");
141     UsuarioDaoImpl udao = new UsuarioDaoImpl();
142     List<Usuario> listaUsuarios = new ArrayList<Usuario>();
143     listaUsuarios = udao.findByUser(usuario, password);
144
145     if(listaUsuarios.size() != 0) {
146         miSesion.setAttribute("userName", request.getParameter("userName"));
147         miSesion.setAttribute("listaUsuarios", listaUsuarios);
148         request.getRequestDispatcher("menuBienvenida.jsp").forward(request, response);
149     } else {
150         mensaje = "Usuario y/o contraseña incorrectos";
151         request.setAttribute("mensaje", mensaje);
152         request.getRequestDispatcher("errorLogin.jsp").forward(request, response);
153     }
154 }
155 }
```

Y como vemos, si encuentra el usuario registrado en la BBDD, el controlador redirige al usuario a la página principal que, en caso contrario, le redirigirá a otra página como en este caso es la página de error.

5.11.5.- Apache Tomcat



Apache Tomcat, funciona como un contenedor de servlets. Implementa las especificaciones de los servlets y de JavaServer Pages (JSP). Por lo que es necesario tener un servidor de aplicaciones como es el caso de Apache Tomcat el cual se ha utilizado para este proyecto

su versión 9.0

Entorno

Tomcat es un contenedor web con soporte de servlets y JSPs. Incluye el compilador Jasper, que compila JSPs convirtiéndolos en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

Estructura de directorios

La jerarquía de directorios de instalación de Tomcat incluye:

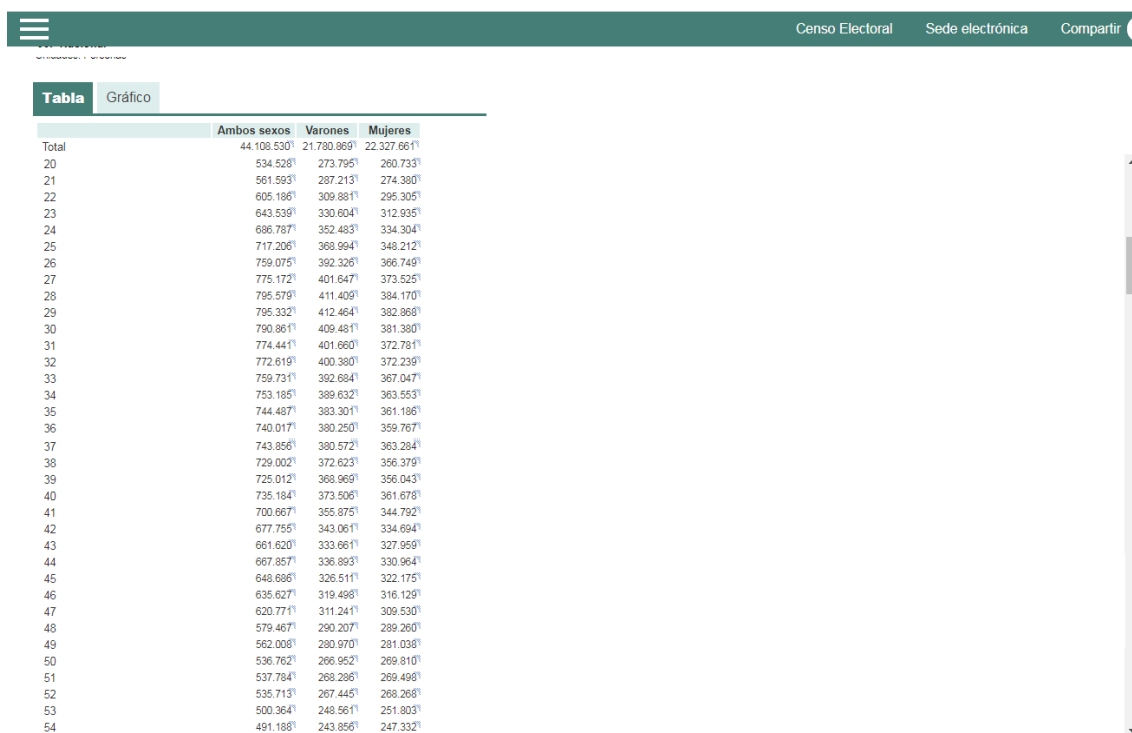
- **bin** - arranque, cierre, y otros scripts y ejecutables.
- **common** - clases comunes que pueden utilizar Catalina y las aplicaciones web.
- **conf** - ficheros XML y los correspondientes DTD para la configuración de Tomcat.
- **logs** - logs de Catalina y de las aplicaciones.
- **server** - clases utilizadas solamente por Catalina.
- **shared** - clases compartidas por todas las aplicaciones web.
- **webapps** - directorio que contiene las aplicaciones web.
- **work** - almacenamiento temporal de ficheros y directorios.

1 3 1



5.12.3.- Micro-Entorno

Al ser venta de cursos vía online, el alumno no se centra en una comunidad concreta sino a nivel nacional. El principal público que se pretende alcanzar, es el comprendido entre 20 y 50 años. Según publica el INE (Instituto Nacional de Estadísticas), en 2021 se registra más de 20 millones de personas en este rango de edad.



The screenshot shows the INE website interface. At the top, there is a navigation bar with 'Censo Electoral', 'Sede electrónica', and 'Compartir'. Below this, there are tabs for 'Tabla' and 'Gráfico'. The 'Tabla' tab is selected, displaying a table with population data. The table has four columns: 'Total', 'Ambos sexos', 'Varones', and 'Mujeres'. The rows represent age groups from 20 to 54. The data is presented in a clean, professional layout with a light blue header and a white body. A vertical scrollbar is visible on the right side of the table.

	Ambos sexos	Varones	Mujeres
Total	44.106.530 ¹	21.790.869 ¹	22.327.661 ¹
20	534.528 ¹	273.795 ¹	260.733 ¹
21	561.593 ¹	287.213 ¹	274.380 ¹
22	605.186 ¹	309.881 ¹	295.305 ¹
23	643.539 ¹	330.604 ¹	312.935 ¹
24	686.787 ¹	352.483 ¹	334.304 ¹
25	717.206 ¹	368.994 ¹	348.212 ¹
26	759.075 ¹	392.326 ¹	366.749 ¹
27	775.172 ¹	401.647 ¹	373.525 ¹
28	795.579 ¹	411.409 ¹	384.170 ¹
29	795.332 ¹	412.464 ¹	382.868 ¹
30	790.861 ¹	409.481 ¹	381.380 ¹
31	774.441 ¹	401.660 ¹	372.781 ¹
32	772.619 ¹	400.380 ¹	372.239 ¹
33	759.731 ¹	392.684 ¹	367.047 ¹
34	753.185 ¹	389.632 ¹	363.553 ¹
35	744.487 ¹	383.301 ¹	361.186 ¹
36	740.017 ¹	380.250 ¹	359.767 ¹
37	743.856 ¹	380.572 ¹	363.284 ¹
38	729.002 ¹	372.623 ¹	356.379 ¹
39	725.012 ¹	368.969 ¹	356.043 ¹
40	735.184 ¹	373.506 ¹	361.678 ¹
41	700.667 ¹	355.875 ¹	344.792 ¹
42	677.755 ¹	343.061 ¹	334.694 ¹
43	661.620 ¹	333.661 ¹	327.959 ¹
44	667.857 ¹	336.893 ¹	330.964 ¹
45	648.696 ¹	326.511 ¹	322.175 ¹
46	635.627 ¹	319.498 ¹	316.129 ¹
47	620.771 ¹	311.241 ¹	309.530 ¹
48	579.467 ¹	290.207 ¹	289.260 ¹
49	562.008 ¹	280.970 ¹	281.038 ¹
50	536.762 ¹	266.952 ¹	269.810 ¹
51	537.784 ¹	268.286 ¹	269.498 ¹
52	535.713 ¹	267.445 ¹	268.268 ¹
53	500.364 ¹	248.561 ¹	251.803 ¹
54	491.188 ¹	243.856 ¹	247.332 ¹

<https://www.ine.es/jaxi/Datos.htm?path=/t20/e245/p04/a2005/&file=00000002.px#!tabs-tabla>

5.12.4.- Factor económico

Según un artículo publicado en www.eleconomista.es, los españoles gastan una media de 327€/año en estudios privados, situándonos en uno de los países de Europa con menos inversión en formación. Sin embargo, aquellos que complementan su formación pública con una formación adicional, conllevan a la larga mejores puestos de trabajo y por consiguiente, más ingresos.

Por otro lado, la formación digital ha ido en aumento a causa de la crisis sanitaria sufrida desde 2020. Este hecho, ha sido un factor importantísimo para la formación online, ya que más del 60% no la realizaba con anterioridad por miedo a la dejadez, no sólo por parte del alumnado sino también por parte de los centros formativos, y a raíz de esta crisis sanitaria, ha posicionado la comodidad y el compromiso por parte de ambos, hasta el punto de que actualmente se prefiere este tipo de formación, creciendo hasta un 85% en sus ingresos.

5.12.5.- Macro-Entorno





Cuantificando la cantidad de clientes potenciales, podremos asegurar que estimando a la baja, al menos un 15% de la población (3mill.) comprendida entre 20 y 50 años sería la fuente de ingresos principal de la aplicación web que, suponiendo que compran una media de 150€ por curso, el beneficio obtenido será de 450 millones.

5.12.6.- Competencia

La competencia en éste sector ha aumentado desde el Covid-19 existiendo actualmente un total de 250 plataformas entre Apps, centros digitales, etc. No obstante, como se demostró en puntos anteriores, lo que se ofrece distinto a todas ellas, es que en este caso la parte interna de marketing se desvincula por completo, dejando al alumno-cliente su espacio en la toma de decisiones. Según un estudio de mercado publicado en 2019 por el diario digital www.lainformacion.com , más de 20% de los clientes desestiman realizar su formación por la presión que reciben en su búsqueda de información.

5.12.7.- Análisis DAFO

El análisis DAFO muestra las Debilidades, Amenazas, Fortalezas y Oportunidades como herramienta útil para el análisis del entorno y formulación de objetivos y estrategias de la empresa.

 Debilidades	 Amenazas
<ul style="list-style-type: none">- Conocimientos Desarrollo Aplicación- Marketing para darse a conocer sin llamadas emitidas a futuros alumnos/clientes.	<ul style="list-style-type: none">- Bastante competencia formativa- Proveedores y precios competentes- Desinterés del alumnado
 Fortalezas	 Oportunidades
<ul style="list-style-type: none">- No existe competencia en la forma de trabajar los clientes.- Crecimiento potencial, cumpliendo estadísticas- Sólo publicidad mediática (RRSS)	<ul style="list-style-type: none">- Cada vez más alumnos optan por la formación digital- Cero contacto 100% cercanía- Análisis de datos de consumo

5.12.8.- Plan de Marketing

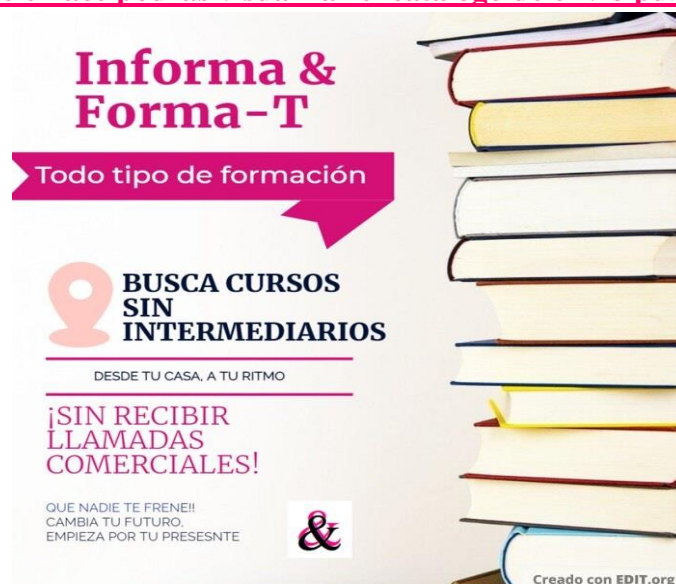
Como bien se ha llevado remarcando desde el inicio, el objetivo del alumno con la aplicación web, es evadir al usuario final del recibo masivo de llamadas comerciales, por lo que es de suma importancia llevar un buen plan de marketing y captar la atención de los usuarios.

El foco principal en el que el alumno pretende captar a sus principales clientes, será mediante las Redes Sociales (Instagram, Facebook, Twitter...) dada la importancia de la utilización de estas plataformas.

En Instagram y Facebook, de promedio la publicidad cuesta poco más de 4 € por cada mil impresiones que se hagan (banner entre historias y entre fotos) en los que, en el momento que un usuario haga clic sobre éste, te cobrarán a parte una media de 0,15€ por cada clic. Todo dependerá también del público al que se quiera abarcar, pues estas dos plataformas pertenecientes a la misma empresa, hace filtros de búsqueda y cobra según las estadísticas de venta y el alcance de su mercado como por ejemplo, si es ropa femenina cuesta mucho más que masculina ya que las mujeres, gastan una media de casi 600€/año frente a 350€/año de los hombres. Por tanto, se basa en estas estadísticas para sus costes publicitarios.

El segundo foco de marketing va destinado a un público más maduro, con edades comprendidas entre los 35 y 50 años. En los que el alumno pretende publicitar con una estrategia de marketing diferente a los anteriores ya que, además de spots publicitarios en redes sociales, pretende hacerlo mediante cartas comerciales, pues considera que hoy en día a penas se recibe este tipo de mensajería y lograría llamar la atención de sus receptores para que al menos, éstos se molesten en ver qué les ha llegado a casa.

En este enlace podrás visualizar el catálogo de envío publicitario



5.12.9.- Proveedores

Para proveer la página web de formación, es necesario tener un buen material didáctico. El alumno realiza una búsqueda exhaustiva de los diversos proveedores que existen en el mercado y encuentra una página donde precisamente se reúne este tipo de sector para ofrecer sus productos a precios muy competitivos.



Ofrece material para todos los cursos que se ofrecen en la aplicación del alumno y dicho proveedor, ofrece los materiales por un coste medio de 300€ cada uno. Por lo que, si el alumno ofrece cursos con un coste medio de 150€, este tipo de material sale muy rentable ya que una vez adquirido el producto, sólo habrá que preocuparse por sus actualizaciones a un mínimo coste. Esto conlleva a poder ampliar el mercado formativo a corto plazo.

5.12.10.- Plan Financiero

Un buen plan financiero debe de constar de tres puntos muy importantes, los cuales son:

1.- Plan de Tesorería: Consiste en prever si algún falta dinero y en ese caso, cómo se va a financiar el negocio.

[Financiación] Para la financiación económica del negocio, el alumno tiene previsto una solicitud de préstamo para nuevos emprendedores el cuál se puede solicitar a través de las entidades bancarias de hasta un máximo de 50 mil. € cantidad más que suficiente para iniciar la actividad comercial.

2.- Cuenta de Resultados: Prever si se obtendrá beneficios o pérdidas.

[Resultados] Se realiza un la cuenta de resultados en el siguiente punto, donde el alumno hará el estudio de los ingresos, gastos e inversiones para el desarrollo de la aplicación.

3.- Balance Previsional: Obtener toda la información sobre el patrimonio de la empresa en cuanto a lo que debe y lo que le deben.

[Balance] En este punto, el alumno contará con los gastos adicionales que supone poner en marcha el negocio tales como impuestos, facturas, ingresos, etc.

5.12.11.- Plan Económico

Haciendo balance de la situación anterior, podemos observar como en principio solo es necesario invertir en nuevos equipos informáticos y de primera calidad para tener la aplicación en su máxima resolución. A este punto, sí que se ha valorado adquirir material intangible para almacenar los datos en un servidor externo proporcionado por un tercero y haciendo una estimación de costes muy por encima de su precio en el mercado, de cara a solicitar el préstamo inicial para arrancar con la actividad.

Cuenta también con el pago de tributos administrativos, tales como impuesto de autónomos y sociedades, impuestos de patentes en marca y logo, etc.

Se le suma un préstamo estimado de 30mil € para adquirir el material formativo y los sistemas informáticos con sus software y almacenamiento oportuno.

Durante el primer año, no se cuenta con incorporar plantilla de empleados haciendo una previsión durante este primer año para ver la evolución del negocio y en caso de beneficios, poder ampliar el mercado y contar con profesionales que se incorporen en la plataforma.

PLAN DE INVERSIONES				
INMOVILIZADO	Inicial	1º año	2º año	3º año
Inmovilizado Intangible:				
‡ Patentes y Marcas	150,00	150,00	0,00	0,00
‡ Derechos Traspaso	0,00	0,00	0,00	0,00
‡ Aplicaciones Informáticas	9.480,00	18.960,00	9.480,00	9.480,00
TOTAL	9.630,00	19.110,00	9.480,00	9.480,00
Inmovilizado Material:				
‡ Terrenos	0,00	0,00	0,00	0,00
‡ Construcciones	0,00	0,00	0,00	0,00
‡ Maquinaria	0,00	0,00	0,00	0,00
‡ Herramientas y Utiles	0,00	0,00	0,00	0,00
‡ Mobiliario	0,00	0,00	0,00	0,00
‡ Equipos informáticos	12.540,46	12.540,46	0,00	0,00
‡ Elementos de transporte	0,00	0,00	0,00	0,00
‡ Otros	1.561,83	1.561,83	0,00	0,00
TOTAL	14.102,29	14.102,29	0,00	0,00
Inversiones Financieras:				
‡ Depósitos y Fianzas (largo plazo)	0,00	0,00	0,00	0,00
‡ Otros	0,00	0,00	0,00	0,00
TOTAL ACTIVO NO CORRIENTE	23.732,29	33.212,29	42.692,29	52.172,29

PLAN DE FINANCIACIÓN				
CONCEPTO	Inicial	1º año	2º año	3º año
RECURSOS PROPIOS:				
↳ Capital	3.000,00	3.000,00	3.000,00	3.000,00
↳ Aportación dineraria de los socios	3.000,00	3.000,00	3.000,00	3.000,00
↳ Otras aportaciones de los socios	0,00	0,00	0,00	0,00
TOTAL RECURSOS PROPIOS	3.000,00	3.000,00	3.000,00	3.000,00
ACREEDORES A LARGO PLAZO:				
↳ Deudas a largo plazo entidades de crédito	30.000,00	28.362,64	26.679,51	24.949,34
↳ Acreedores a largo plazo				
↳ Otras deudas a largo plazo				
TOTAL ACREEDORES A LARGO PLAZO	30.000,00	28.362,64	26.679,51	24.949,34
ACREEDORES A CORTO PLAZO:				
↳ Deudas a corto plazo entidades de crédito	0,00	0,00	0,00	0,00
↳ Proveedores		0,00	0,00	0,00
↳ Administraciones Públicas		6.494,94	7.339,05	88.053,56
TOTAL ACREEDORES CORTO PLAZO	0,00	6.494,94	7.339,05	88.053,56
TOTAL	33.000,00	37.857,58	37.018,56	116.002,89

PRESUPUESTO DE TESORERÍA (OPERATIVO)			
CONCEPTO	1º año	2º año	3º año
SALDO INICIAL	4.283,93	33.709,43	58.115,29
COBROS:			
↳ Por ventas anteriores			
↳ Por ventas del período	144.000,00	144.000,00	540.000,00
TOTAL SALDO INICIAL + COBROS	*****	*****	598.115,29
PAGOS:			
POR GASTOS VARIABLES:			
↳ Compras período anterior			
↳ Compras del período	0,00	0,00	0,00
POR OTROS GASTOS VARIABLES:			
DEL PERÍODO ANTERIOR:			
↳ Hacienda Pública IRPF			
↳ Seguridad Social			
DEL PERÍODO:			
↳ Suministros	2.511,96	2.763,16	2.763,16
↳ Otros gastos variables + Inversiones	35.670,80	27.604,13	29.217,47
↳ Sueldos y salarios			
↳ Seguridad Social empresa			
↳ Hacienda Pública IRPF			
↳ Seguridad Social			
POR GASTOS FIJOS:			
DEL PERÍODO ANTERIOR:			
↳ Hacienda Pública IRPF			
↳ Seguridad Social			
DEL PERÍODO:			
↳ Gastos Fijos (Varios)	0,00	0,00	0,00
↳ Amortización de deudas con ent. créd.	1.637,36	1.683,13	1.730,17
↳ Total Gastos Financ. Exentos IVA	807,39	761,62	714,58
↳ Sueldos y salarios líquidos	42.302,98	42.302,98	42.302,98
↳ Seguridad Social (Empresa o autónomo)	9.994,39	9.994,39	9.994,39
↳ Hacienda Pública IRPF	5.057,14	5.057,14	5.057,14
↳ Seguridad Social (trabajador)	3.211,28	3.211,28	3.211,28
↳ IVA período	13.381,19	19.721,38	88.168,65
↳ Impuesto de Sociedades	0,00	6.494,94	7.339,05
TOTAL PAGOS	*****	119.594,15	190.498,86

PREVISIÓN DE CUENTA DE PÉRDIDAS Y GANANCIAS			
CONCEPTO	1** año	2* año	3** año
↳ Importe Neto por Cifra de Negocios	119.008,26	119.008,26	446.280,99
↳ Aprovisionamientos	0,00	0,00	0,00
↳ Otros ingresos de explotación	0,00	0,00	0,00
↳ Gastos de personal	-60.565,79	-60.565,79	-60.565,79
↳ Otros gastos de explotación	-22.076,00	-15.616,93	-16.950,27
↳ Dotac. Amortización Inmovilizado intangible	-6.256,80	-9.385,20	-12.513,60
↳ Dotac. Amortización Inmovilizado material	-3.322,53	-3.322,53	-3.322,53
RESULTADOS DE EXPLOTACIÓN	26.787,14	30.117,81	*****
↳ Ingresos Financieros	0,00	0,00	0,00
↳ Gastos Financieros	807,39	761,62	714,58
RESULTADOS FINANCIEROS	-807,39	-761,62	-714,58
RESULTADOS ANTES DE IMPUESTOS:	25.979,75	29.356,18	352.214,22
↳ Impuestos sociedades	6.494,94	7.339,05	88.053,56
RESULTADO DEL EJERCICIO	19.484,81	22.017,14	264.160,67
Impuesto de Sociedades	25%	25%	25%

BALANCE DE SITUACIÓN FINAL			
ACTIVO	1** año	2* año	3** año
ACTIVO NO CORRIENTE	23.632,96	20.405,22	14.049,09
INMOVILIZADO INTANGIBLE			
↳ Programas informáticos	18.960,00	28.440,00	37.920,00
↳ Patentes y Marcas	150,00	150,00	150,00
↳ Derechos (Traspaso y Otros)/ Gtos. anticipados	0,00	0,00	0,00
↳ Amortización acumulada intangible	-6.256,80	-15.642,00	-28.155,60
INMOVILIZADO MATERIAL			
↳ Terrenos	0,00	0,00	0,00
↳ Construcciones	0,00	0,00	0,00
↳ Maquinaria	0,00	0,00	0,00
↳ Herramientas y útiles	0,00	0,00	0,00
↳ Mobiliario	0,00	0,00	0,00
↳ Equipo informático	12.540,46	12.540,46	12.540,46
↳ Elementos de transporte	0,00	0,00	0,00
↳ Otros	1.561,83	1.561,83	1.561,83
↳ Amortización acumulada material	-3.322,53	-6.645,07	-9.967,60
INVERSIONES FINANCIERAS			
↳ Depósitos y Fianzas	0,00	0,00	0,00
ACTIVO CORRIENTE	33.709,43	58.115,29	407.616,43
REALIZABLE			
↳ Existencias Finales	0,00	0,00	0,00
↳ Materias primas			
↳ Productos terminados			
DEUDORES COMERCIALES Y OTRAS CUENTAS POR COBRAR			
↳ Clientes y Deudores + HP x IVA	0,00	0,00	0,00
EFFECTIVO Y OTROS ACTIVOS LIQUIDOS			
↳ Tesorería	33.709,43	58.115,29	407.616,43
TOTAL ACTIVO	57.342,39	78.520,51	421.665,51

PATRIMONIO NETO	1º año	2º año	3º año
RECURSOS PROPIOS	22.484,81	44.501,95	308.662,62
↳ Capital	3.000,00	3.000,00	3.000,00
↳ Reservas		19.484,81	41.501,95
↳ Resultados negativos de ejerc. anteriores		0,00	0,00
↳ Resultados del ejercicio	19.484,81	22.017,14	264.160,67
PASIVO NO CORRIENTE	28.362,64	26.679,51	24.949,34
↳ Deudas a largo plazo entidades de crédito	28.362,64	26.679,51	24.949,34
↳ Acreedores a largo plazo	0,00	0,00	0,00
PASIVO CORRIENTE	6.494,94	7.339,05	88.053,56
↳ Proveedores	0,00	0,00	0,00
↳ Hacienda Pública IRPF			
↳ Org. Seguridad Social acreedores			
↳ Hacienda Pública por IVA	0,00	0,00	0,00
↳ Hacienda Pública Impuesto Sociedades	6.494,94	7.339,05	88.053,56
TOTAL PATRIMONIO NETO Y PASIVO	57.342,39	78.520,51	421.665,51

LIQUIDACIÓN DE IVA - AÑO (1)

IVA SOPORTADO	
TIPO APLICABLE	10%

IVA REPERCUTIDO	
TIPO APLICABLE	10%

SALDO HACIENDA PÚBLICA POR IVA

6.372,00

PARTIDAS CON IVA SOPORTADO

Inversiones y Gastos	33.212,29
Gastos de Explotac.	22.076,00
Aprovisionamientos	0,00
	55.288,29
	5.528,83

PARTIDAS CON IVA REPERCUTIDO

Facturación x Ventas	119.008,26
Otros Ingresos	
	119.008,26
	11.900,83

AÑO 1 (AÑO DE ADQUISICIÓN)				
ELEMENTOS	Precio Adquisición	% AMORTIZACIÓN	Nº AÑOS	CUOTA ANUAL AMORTIZABLE
Intangible				
SOFTWARE INFORMÁTICO	18.960,00	33	3,03	6.256,80
TOTALES				6.256,80
Material				
CONSTRUCCIONES	0,00	2	50,00	0,00
MAQUINARIA	0,00	12	8,33	0,00
UTILLAJE Y HERRAMIENTAS	0,00	12	8,33	0,00
MOBILIARIO	0,00	10	10,00	0,00
EQUIPOS INFORMÁTICOS	12.540,46	25	4,00	3.135,12
ELEMENTOS DE TRANSPORTE	0,00	16	6,25	0,00
OTROS (INSTALAC. NECESARIAS)	1.561,83	12	8,33	187,42
TOTALES				3.322,53

Llegados a éste punto, observamos que la rentabilidad del negocio es de 57.000€ netos anuales como beneficio de la activad, lo que resulta más que rentable para inicializar la aplicación de venta de formación online sin intermediarios.

6.- Conclusiones y Mejoras del Proyecto

Como conclusión final del proyecto, se observa que el uso de la aplicación es más que intuitiva según se ha experimentado el uso de la misma con el entorno más cercano del alumno, los cuales han podido utilizarla en un entorno libre y sin indicaciones y han podido probar que no es necesario indagar mucho para su manejo.

En cuanto a los resultados esperados, hubo pruebas que no tenían funcionalidad y se eliminaron y otras que a pesar de su funcionalidad, no tenían ninguna relevancia para el uso de la misma tales como “filtrar cursos” ya que en la fase de experimentación con usuarios ajenos, estos reflejaban que no les resultaba necesario viendo como están clasificados por diferentes ámbitos.

Hay que tener en cuenta la reacción de aquellos que han interactuado con la aplicación y es de mencionar que a diferencia de otras páginas de venta de cursos, ésta no da lugar a ningún tipo de presión comercial, lo que invita y cumple con el objetivo de que el usuario permanezca en la misma para investigar sobre los diferentes módulos formativos, consiguiendo a su vez que incentive la compra final del producto.

Respecto a las mejoras de la aplicación, habrá que ampliar más contenido de formación así como otra interfaz de usuario para el perfil del profesorado, teniendo que aumentar las bases de datos para que estos últimos puedan llevar un control de quiénes realizan su formación, cuánto tiempo le dedican a la misma, qué clasificaciones obtienen en sus resultados, etc.

También está la mejora por parte del usuario final, en la que habría que mejorar la forma de ver sus progresos, sus evaluaciones, poner en marcha el ámbito de comunidad donde cada usuario pueda subir comentarios de su experiencia en la formación, etc.

Finalmente, a lo que a nivel personal se refiere, el alumno se da cuenta de todo lo que ha ido aprendiendo durante estos dos años de formación profesional, sin dejar de lado que aún queda mucho camino por recorrer y más aprendizaje en cuanto a nuevos lenguajes de programación. También ha sido consciente de la forma evolutiva que este TFG ha generado, observando de como en un principio hace mucho código sin forma productiva y viendo cómo puede hacer lo mismo una vez ha ido finalizando el trabajo, pero con mucho menos código de que ha ejecutado. Por ejemplo, hacía una página CSS por cada página HTML y JSP que ejecutaba, dándose cuenta de que era innecesario y podía reutilizar las mismas.

También aprendió a tratar con diferentes tipos de perfiles, llegando a la conclusión que al final por mucho que sea su aplicación, lo que cuenta es la decisión del usuario y la opinión que éste tenga sobre el manejo de la misma con sus beneficios e inconvenientes, sabiendo aceptar las críticas y encajarlas como mejoras en la usabilidad del mismo.

7.- Bibliografía

[Textos]

En la descripción de los textos, el alumno se ha apoyado en un 80% sobre los contenidos impartidos durante su formación, accediendo a su material proporcionado por el instituto tecnológico <https://institutotecnologico.edix.com/>

Para el resto de la información, ha realizado búsquedas a través de páginas como www.wikipedia.com <https://es.stackoverflow.com/> www.github.com así como libros de gran ayuda en material de desarrollo web tales como “*Introducción a los patrones de diseño*” de Óscar Blancante. “*No me hagas pensar*” de Steve Krug para la parte de experiencia de usuario.

[Imágenes]

Las imágenes mostradas del proyecto, han sido gestionadas con herramientas de trabajo indicadas en cada apartado y haciendo capturas de los resultados obtenidos en el trabajo del alumno. Para la herramienta en el proyecto de empresa, se ha utilizado las impartidas por el profesor de dicha asignatura mediante plantillas que distribuyó en el tiempo de impartir su formación (tales como plantillas excell para los cálculos financieros).

8.- Anexos

Tanto el código fuente de la aplicación, como los scripts realizados para la creación de las bases de datos pueden visualizarse en el repositorio git del alumno, siendo este https://github.com/FranDeVicente/TFG_2-DAW_ITEdix