

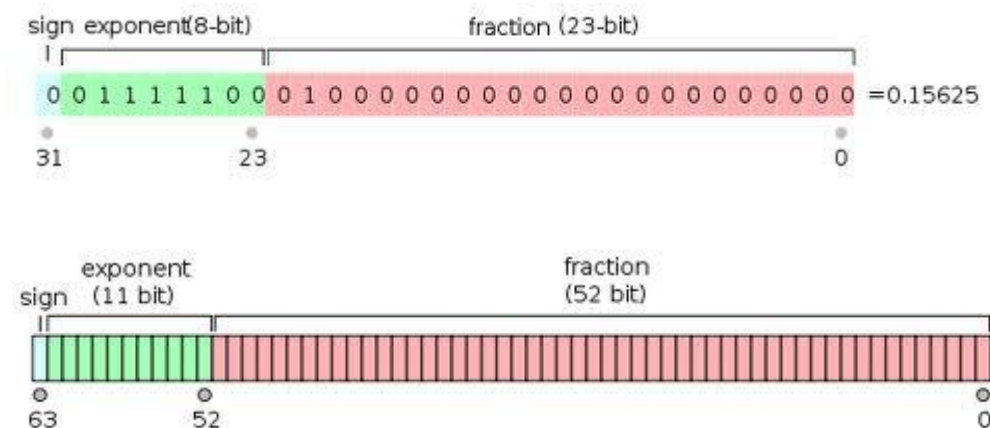


Universidad Politécnica de Valencia

Facultad de Informática

Proyecto final de carrera:

Construcción de un sitio web para la realización de ejercicios IEEE 754



Autor: Rubén Carrión Baeza

Director: Antonio Martí Campoy

# ÍNDICE

<b>1. Introducción</b> .....	5
1.1. Motivación .....	5
1.2. Objetivos y resumen del proyecto .....	5
1.3. Contexto .....	5
1.4. Estructura del documento .....	5
<b>2. Especificación de requisitos</b> .....	6
2.1. Introducción .....	6
2.1.1. Propósito .....	6
2.1.2. Ámbito .....	6
2.1.3. Definición, siglas y abreviaturas .....	6
2.1.4. Referencias .....	7
2.2. Descripción general .....	7
2.2.1. Perspectiva del producto .....	7
2.2.2. Funciones del producto .....	7
2.2.3. Características del usuario .....	8
2.2.4. Restricciones generales .....	8
2.2.5. Supuestos y dependencias .....	8
2.3. Requisitos específicos .....	8
2.3.1. Requerimientos de interfaces externos .....	8
2.3.2. Requisitos funcionales .....	9
2.3.3. Requisitos de eficiencia .....	11
2.3.4. Restricciones de diseño .....	11
2.3.5. Atributos .....	11
2.3.6. Otros requerimientos .....	12
<b>3. Análisis</b> .....	12
3.1. Casos de uso .....	12
3.1.1. Caso de uso: Actor Anónimo .....	12
3.1.2. Caso de uso: Actor Alumno .....	13
<b>4. Diseño</b> .....	13
4.1. Arquitectura de 3 capas .....	13
4.1.1. Capa de presentación .....	13
4.1.2. Capa de lógica de negocio .....	15
4.1.3. Capa de persistencia .....	15
<b>5. Implementación</b> .....	15
5.1. Tecnologías .....	15
5.1.1. Capa de presentación .....	15
5.1.2. Capa de lógica de negocio .....	16
5.1.3. Capa de persistencia .....	16
5.2. Descripción de la implementación .....	17
5.2.1. Parte pública .....	17
5.2.2. Parte privada .....	18
5.2.3. Otras páginas .....	23
<b>6. Evaluación</b> .....	30
6.1. Prueba de validación de CSS .....	30
6.2. Prueba de validación de navegadores web .....	30
6.3. Prueba de seguridad de acceso .....	31
<b>7. Conclusiones</b> .....	32
7.1. Trabajo realizado .....	32
7.2. Valoración personal .....	32

<b>7.3. Futuras mejoras</b> .....	33
<b>8. Bibliografía</b> .....	33
<b>Anexos</b> .....	34
<b>A. Anexo I (Herramientas usadas)</b> .....	34
<b>ArgoUML</b> .....	34
<b>Balsamiq Mockups</b> .....	34

## Ilustraciones

Ilustración 1: Boceto de la interfaz .....	9
Ilustración 2: Diagrama de usuarios.....	12
Ilustración 3: Caso uso Anónimo.....	13
Ilustración 4: Caso uso Alumno.....	13
Ilustración 5: Diseño de la web .....	14
Ilustración 6: Cabecera.....	14
Ilustración 7: Menú .....	14
Ilustración 8: Zona central.....	15
Ilustración 9: login.html .....	17
Ilustración 10: registro.html.....	18
Ilustración 11: index.html .....	18
Ilustración 12: decimal2simple.html.....	19
Ilustración 13: decimal2doble.html .....	20
Ilustración 14: simple2decimal.html.....	21
Ilustración 15: doble2decimal.html .....	22
Ilustración 16: preguntas.html.....	22
Ilustración 17: Validación CSS .....	30
Ilustración 18: Validación navegador web Microsoft Internet Explorer.....	30
Ilustración 19: Validación navegador web Mozilla Firefox .....	31
Ilustración 20: Validación navegador web Google Chrome.....	31
Ilustración 21: Validación de seguridad .....	32
Ilustración 22: Validación de seguridad (error de login).....	32
Ilustración 23: ArgoUML .....	34
Ilustración 24: Balsamiq Mockups .....	35

# 1. Introducción

Este documento es la memoria descriptiva del proceso de desarrollo de una aplicación Web como proyecto final de carrera de Ingeniería Técnica en Informática de Sistemas en la Universidad Politécnica de Valencia.

## 1.1. Motivación

En la actualidad cada vez son más los alumnos que desean resolver ejercicios a través de la Web, pero el software que existe para la realización de estos ejercicios muestra el resultado final y no guía al alumno paso a paso durante la resolución indicándole donde se está equivocando. Por ello la finalidad de este proyecto es la creación de una aplicación Web para resolver los ejercicios de representación IEEE754.

## 1.2. Objetivos y resumen del proyecto

El objetivo principal de este proyecto es realizar una aplicación Web que guíe en la resolución de ejercicios de representación IEEE754 de decimal a simple y doble precisión, y de simple y doble precisión a decimal así como preguntas en relación a la representación de dichos números para que el alumno pueda practicar dichos ejercicios y facilitar el aprendizaje. Además se realizará un seguimiento de los fallos y aciertos del usuario.

Para el uso de la aplicación se requiere una conexión a Internet, navegador web y conocimientos de representación de números decimales en simple y doble precisión.

## 1.3. Contexto

Para lograr estos objetivos es necesaria la existencia de un usuario identificado para registrar sus fallos y aciertos. Desde la aplicación se ofrecerán los pasos a seguir para la resolución de los ejercicios comprobando que el usuario los realiza correctamente para indicarle sus errores. Tras un determinado número de fallos se da por finalizado el ejercicio como fallido.

## 1.4. Estructura del documento

Esta memoria está ordenada de forma que se pueda resumir cada una de las fases del ciclo de vida del proyecto. Se ha seguido como guía el estándar IEEE std 830-1998.

**Introducción:** Capítulo de presentación del proyecto en el que se incluye motivación, objetivos y resumen del proyecto, contexto y estructura del documento.

**Especificación de requisitos:** Capítulo en el que se recogen los requisitos básicos necesarios antes del inicio del proyecto.

**Análisis:** Capítulo en el que se obtiene un modelo conceptual de la aplicación mediante la ayuda de diagramas UML.

**Diseño:** Capítulo en el que se describen los diseños realizados para el proyecto así como la metodología utilizada en el proceso.

**Implementación:** Capítulo en el que se describen las tecnologías y herramientas usadas durante esta fase. Se describirá el funcionamiento de los componentes implementados en la solución.

**Evaluación y pruebas:** Capítulo en el que se exponen las técnicas usadas en la evaluación de la aplicación y el resultado obtenido.

**Conclusiones:** Capítulo que contiene la valoración personal del desarrollo del proyecto y del trabajo realizado.

**Bibliografía:** Capítulo en el que se muestra la documentación utilizada para el desarrollo del proyecto.

## 2. Especificación de requisitos

### 2.1. Introducción

#### 2.1.1. Propósito

El propósito de este apartado es reunir los requerimientos necesarios para el desarrollo correcto de la aplicación.

Esta especificación va dirigida a toda persona interesada en el funcionamiento de la aplicación. Se seguirán las directrices IEEE std 830-1998, uno de los estándares más referenciados para la especificación de requisitos en proyectos Web.

#### 2.1.2. Ámbito

El producto software a describir es una aplicación web de resolución de ejercicios. Con todas las funcionalidades implementadas será posible realizar ejercicios de representación de formato IEEE754 comprobando los errores cometidos durante el proceso sin la necesidad de un profesor.

#### 2.1.3. Definición, siglas y abreviaturas

**Apache:** Tipo de servidor donde se almacena la aplicación, y que permite la gestión de las peticiones sobre la web.

**Autenticación o autenticación:** Proceso de intento de verificar la identidad digital del remitente de una comunicación como una petición para conectarse.

**Decimal:** Número decimal.

**Doble precisión:** Número decimal representado en IEEE754 con 64 bits.

**Formulario (HTML):** Servicio que permite introducir información para enviar a la web y ser procesada.

**Hardware:** conjunto de elementos materiales que conforman una computadora.

**HTML** (HyperText Mark Language): Lenguaje usado para crear documentos en la World Wide Web.

**IEEE** (Institute of Electrical and Electronics Engineers): Asociación de profesionales norteamericanos que aporta criterios de estandarización de dispositivos eléctricos y electrónicos.

**IEEE754:** Estándar de IEEE para aritmética en coma flotante. Define valores especiales como el cero, infinito, NaN y valores desnormalizados. También especifica modos de redondeo.

**JavaScript:** Lenguaje de programación interpretado que se ejecuta en el lado del cliente (navegador).

**Navegador:** Programa que permite ver e interactuar con varios tipos de recursos de Internet disponibles en la World Wide Web.

**Simple precisión:** Número decimal representado en IEEE754 con 32 bits.

**Sistema de Gestión de Bases de Datos (SGBD):** Software dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

**Sistema Operativo (SO):** Un sistema operativo es un software de sistema, es decir, un conjunto de programas de computadora destinado a permitir una administración eficaz de sus recursos.

**Software:** Conjunto de los componentes necesarios para hacer posible la realización de una tarea específica. Soporte lógico de la computadora.

**WAMP:** Aplicación para instalar Apache, PHP y MySQL de forma sencilla

## 2.1.4. Referencias

- IEEE STD 830- IEEE Guide to Software Requirements Specifications.
- <http://es.wikipedia.org>

## 2.2. Descripción general

### 2.2.1. Perspectiva del producto

Este producto software es independiente del sistema operativo utilizado, solo se necesitará un navegador web e Internet para poder usarlo. Para ello es necesario un servidor web para alojar la web y un dominio para su direccionamiento.

### 2.2.2. Funciones del producto

Alumno:

- Realizar ejercicios de conversión de decimal a simple precisión.
- Realizar ejercicios de conversión de decimal a doble precisión.
- Realizar ejercicios de conversión de simple precisión a decimal.

- Realizar ejercicios de conversión de doble precisión a decimal.
- Responder preguntas acerca del estándar IEEE754.

Anónimo:

- Registrarse en la aplicación.

Sistema:

- Abrir sesión de alumno.
- Cerrar sesión de alumno.
- Registrar fallos y aciertos del alumno.

### 2.2.3. Características del usuario

Se identifican como tipos de usuario: alumno y anónimo. Aunque para hacer uso de la aplicación hay que ser un usuario registrado (alumno), el usuario anónimo también existe ya que este puede registrarse para convertirse en un usuario registrado (alumno).

El usuario alumno es el único que puede realizar ejercicios y responder preguntas.

### 2.2.4. Restricciones generales

Para acceder a la Web se deberá disponer del hardware necesario para conectarse a Internet. Cualquier persona podrá visitar la Web mediante un navegador, aunque deberá autenticarse para hacer uso de ella.

### 2.2.5. Supuestos y dependencias

La Web requiere que los usuarios utilicen un navegador web para acceder a ella. No será necesario el uso de un navegador específico, pero sí deberá ser compatible con el protocolo HTTP. Dicho navegador debe permitir la ejecución de código Javascript.

## 2.3. Requisitos específicos

### 2.3.1. Requerimientos de interfaces externos

#### **Interfaces de usuario**

Se pretende que la interfaz de la Web sea simple e intuitiva. Para ello se sigue un patrón general de muchas webs, un logo en la zona superior, un menú lateral y una zona central para resolver los ejercicios. Todo ello acompañado por la interacción de un teclado y un ratón por parte del usuario.

A continuación se muestra el boceto de la interfaz web:



Logo

[Decimal a Simple precisión](#)  
[Decimal a Doble precisión](#)  
[Simple precisión a Decimal](#)  
[Doble precisión a Decimal](#)  
[Preguntas](#)

**Decimal a Simple precisión:**

Decimal:

Parte entera en binario:

Parte decimal en binario:

Decimal en binario (coma fija):

Decimal en binario (normalizado):

Exponente:

Exponente (con exceso):

Signo en binario:

Simple precisión:

Hexadecimal:

Ilustración 1: Boceto de la interfaz

**Interfaces hardware**

Los usuarios deben disponer de acceso Internet y de un navegador web que permita una resolución de al menos 1024x786.

**Interfaces software**

La aplicación web está desarrollada para ejecutarse en cualquier sistema operativo y navegador web estándar.

**Interfaz de comunicación**

La interfaz de comunicación entre cliente y servidor son los estándares TCP/IP a través del protocolo HTTP.

## 2.3.2. Requisitos funcionales

**Alumno**

Conversión de decimal a simple precisión:

- Introducción: Esta función sirve para convertir un número decimal a simple precisión.
- Entrada: Número decimal.
- Proceso: Convierte un número decimal en simple precisión mostrándolo también en hexadecimal.

- Salida: Número decimal en simple precisión en binario y hexadecimal.

Conversión de decimal a doble precisión:

- Introducción: Esta función sirve para convertir un número decimal a doble precisión.
- Entrada: Número decimal.
- Proceso: Convierte un número decimal en doble precisión mostrándolo también en hexadecimal.
- Salida: Número decimal en doble precisión en binario y hexadecimal.

Conversión de simple precisión a decimal:

- Introducción: Esta función sirve para convertir un número en simple precisión a decimal.
- Entrada: Número en simple precisión o hexadecimal.
- Proceso: Convierte un número en simple precisión a decimal.
- Salida: Número decimal.

. Conversión de doble precisión a decimal:

- Introducción: Esta función sirve para convertir un número en doble precisión a decimal.
- Entrada: Número en doble precisión o hexadecimal.
- Proceso: Convierte un número en doble precisión a decimal.
- Salida: Número decimal.

Preguntas estándar IEEE754:

- Introducción: Esta función sirve para que el alumno responda preguntas acerca del estándar IEEE754.
- Entrada: Ninguna.
- Proceso: Se plantea una pregunta y se dan opciones como posibles respuestas para que el alumno selecciona una.
- Salida: Respuesta correcta.

### **Anónimo**

Registrarse como alumno:

- Introducción: Esta función sirve para registrar al usuario anónimo en la aplicación Web.
- Entrada: Datos del formulario de registro (nombre, apellidos, DNI y curso).
- Proceso: Se registra al alumno con los datos indicados. Si ya existe el alumno (existe el DNI) el SGBD impide automáticamente que se registre de nuevo.
- Salida: Reenvía al formulario de identificación para que el usuario acceda a la aplicación.

### **Sistema**

Abrir sesión de alumno:

- Introducción: Esta función sirve para identificar al alumno y permitirle el acceso a la Web.
- Entrada: Datos del formulario de identificación (nombre, apellidos, DNI y curso).

- Proceso: Se comprueba la existencia del DNI introducido por el alumno, en ese caso se comprueba si el resto de datos son correctos. Si los datos coinciden se crea una cookie que mantendrá identificado al alumno.
- Salida: Mensaje de error si la validación es incorrecta o redirección a la página principal en caso contrario.

Cerrar sesión de alumno:

- Introducción: Esta función sirve para que el alumno cierre sesión y deje de estar identificado.
- Entrada: Cookies creada en la identificación.
- Proceso: Se elimina la cookie para dejar de identificar al alumno.
- Salida: Redirección a la página de identificación.

Registrar estadísticas de ejercicios:

- Introducción: Esta función sirve para registrar los aciertos y fallos del alumno.
- Entrada: Ejercicio a resolver y resultado del alumno.
- Proceso: Cuando el alumno inicia un nuevo ejercicio sin terminar el anterior o muestra la solución del ejercicio actual se registra como ejercicio fallado en la base de datos.
- Salida: Mensaje de error si falla el ejercicio o mensaje de acierto en caso contrario.

### 2.3.3. Requisitos de eficiencia

Al estar realizada la Web con Javascript, que es un lenguaje que se ejecuta en el cliente, el servidor se libera del procesamiento quedando reducido a la identificación del alumno y registrar los aciertos y fallos del alumno.

### 2.3.4. Restricciones de diseño

#### Estándares cumplidos

No existen restricciones respecto a estándares.

#### Limitaciones hardware

No existen limitaciones hardware.

### 2.3.5. Atributos

#### Seguridad

Es necesario incluir sistemas de seguridad para cumplir con la Ley Orgánica de Protección de Datos para llevar el proyecto a la realidad.

Las comunicaciones que incluyan datos confidenciales (datos de login) deben ser protegidas mediante conexiones seguras SSL.

#### Mantenimiento

El mantenimiento de la Web es sencillo pero se recomienda seguir este documento para conocer el funcionamiento interno. Es recomendable hacer una copia de seguridad tanto de la Web como de los DNI, claves y estadísticas de ejercicios antes de introducir cambios.

No es necesario el mantenimiento por parte de un administrador ya que no es necesario mantener actualizada la información ni los servicios que ofrece.

### **Portabilidad**

La Web es compatible con cualquier sistema que siga los estándares convencionales de navegación web. Esto significa que la aplicación puede ser soportada por cualquier plataforma y sistema operativo.

## 2.3.6. Otros requerimientos

### **Base de datos**

La Web utiliza MySQL para la gestión de la base de datos. En ella se guardarán los datos de los alumnos (nombre, apellidos, DNI y curso) y las estadísticas de los ejercicios resueltos correcta e incorrectamente.

## 3. Análisis

En la fase de análisis se estudian las funcionalidades de la aplicación. A lo largo de esta fase se describe la estructura y funcionalidad de la aplicación mediante el uso de diagramas UML.

### 3.1. Casos de uso

La fase de análisis de la aplicación describe su funcionalidad utilizando diagramas de casos de uso. Estos diagramas son una representación gráfica de las interacciones entre actores (usuarios o aplicaciones externas que usan la aplicación) y casos de uso (funcionalidad del sistema). Las interacciones entre actores y casos de uso describen el comportamiento del sistema desde el punto de vista del actor.



Ilustración 2: Diagrama de usuarios

#### 3.1.1. Caso de uso: Actor Anónimo

A continuación se muestran los casos de uso para el actor Anónimo. Las acciones que puede realizar este actor son muy limitadas por no ser un usuario identificado por el sistema.

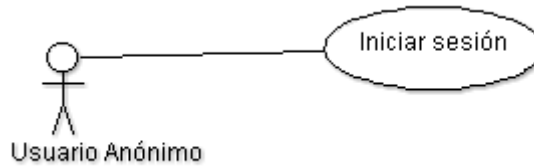


Ilustración 3: Caso uso Anónimo

### 3.1.2. Caso de uso: Actor Alumno

A continuación se muestran los casos de uso para el actor Alumno. Este actor puede realizar todas las acciones de la aplicación.

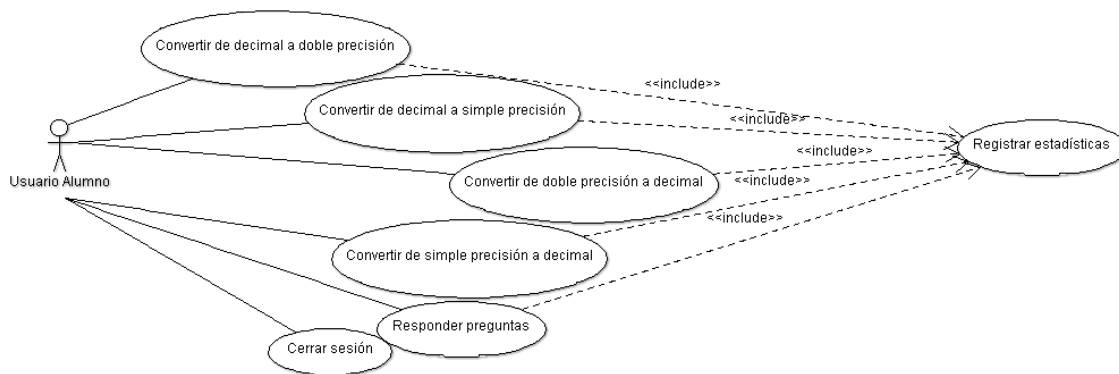


Ilustración 4: Caso uso Alumno

## 4. Diseño

### 4.1. Arquitectura de 3 capas

La arquitectura utilizada para el desarrollo de la aplicación web es una arquitectura de tres capas. En la capa de presentación se encuentra el diseño web (HTML y CSS) y códigos Javascript que resuelven los ejercicios. En la capa de lógica de negocio se encuentra el código PHP para identificar al usuario y guardar sus estadísticas. Y en la capa de persistencia se encuentra el SGBD encargado de guardar los alumnos y registro de ejercicios resueltos por ellos.

#### 4.1.1. Capa de presentación

Es la parte que interactúa con el usuario a través del navegador web. Al estar casi toda la carga de la aplicación en esta capa el servidor podría atender numerosas peticiones de diferentes usuarios. Dispondrá de un diseño sencillo para facilitar la usabilidad de la aplicación.

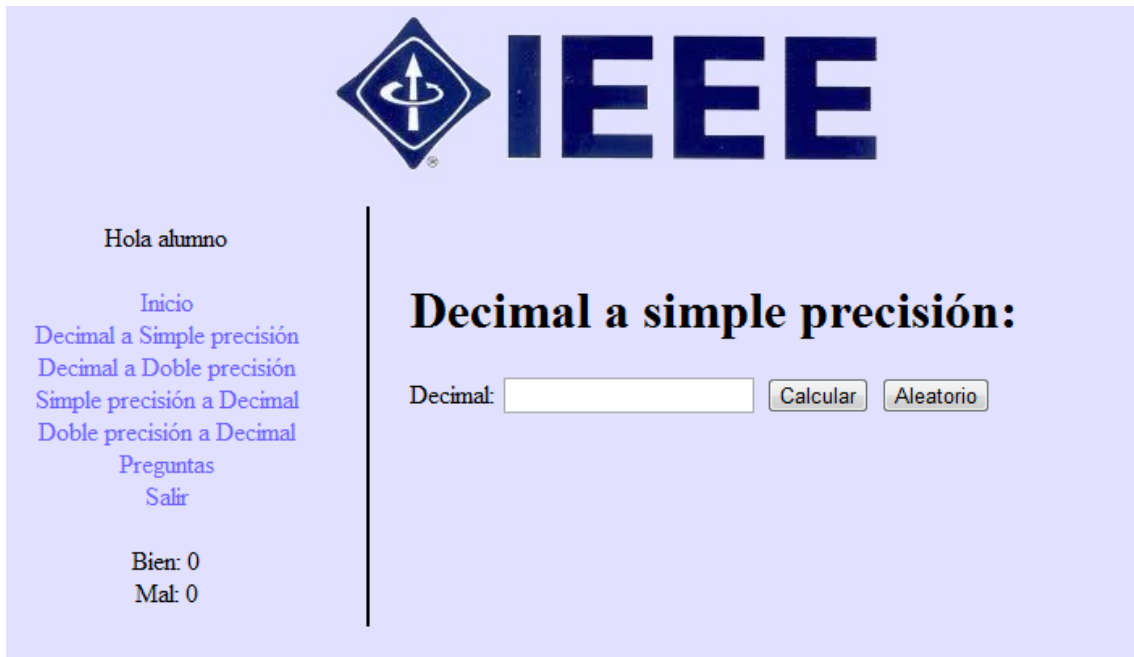


Ilustración 5: Diseño de la web

Como se puede apreciar consta de una cabecera, un menú en el lateral izquierdo y una zona central para información y resolución de ejercicios.

La cabecera está compuesta únicamente por el logo de la página web.



Ilustración 6: Cabecera

El menú lateral contiene los enlaces para la realización de los distintos ejercicios.

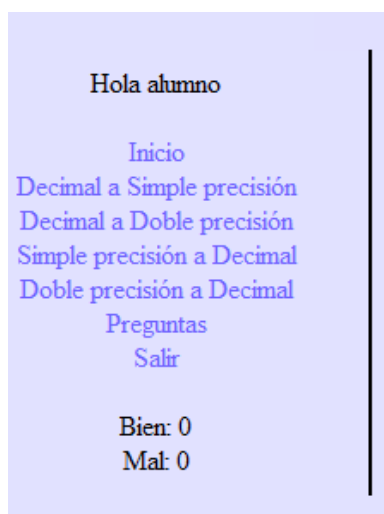
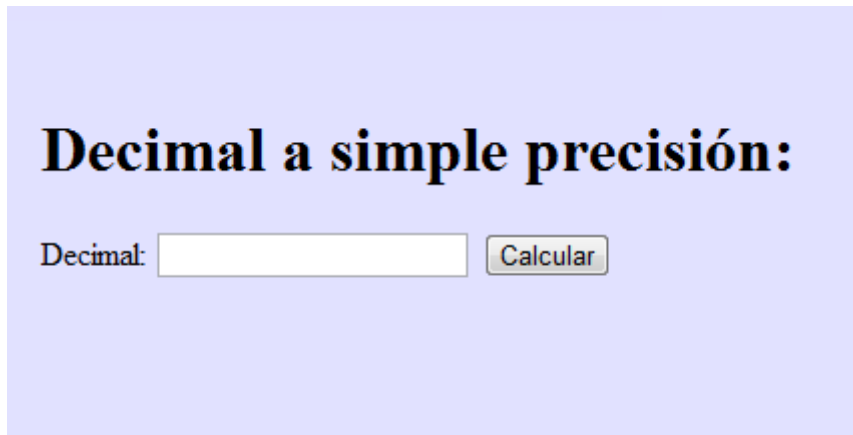


Ilustración 7: Menú

La zona central contiene la información o los pasos para resolver los ejercicios.



**Decimal a simple precisión:**

Decimal:

Ilustración 8: Zona central

### 4.1.2. Capa de lógica de negocio

Es la parte que se ejecuta en el servidor. Se encarga de identificar y mantener identificado al usuario y de escribir las estadísticas de ejercicios resueltos por el usuario en la base de datos.

### 4.1.3. Capa de persistencia

Compuesto únicamente por el SGBD (en este caso MySQL) que contiene los datos necesarios para el funcionamiento de la aplicación web.

## 5. Implementación

Para la elaboración de esta aplicación se desea que se ejecute en el cliente, para ello se desarrollará en HTML y Javascript, y se maquetará con CSS.

### 5.1. Tecnologías

#### 5.1.1. Capa de presentación

**HTML:**

El HTML, siglas de HyperText Markup Language (o lenguaje de marcado de hipertexto), es el lenguaje de marcado utilizado normalmente para el desarrollo de páginas web. Fue creado por el físico nuclear Tim Berners-Lee en 1986, a partir de dos herramientas preexistentes: el concepto de hipertexto y el SGML. El concepto de hipertexto, también conocido como link o ancla, permite conectar dos elementos entre sí. Por otro lado SGML, Standard Generalized Markup Language, es un lenguaje estándar de marcación general que sirve para colocar etiquetas o marcas en un texto que indique como debe mostrarse.

Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de

“etiquetas”, rodeadas por corchetes angulares. HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo Javascript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

El lenguaje HTML puede ser creado y editado con cualquier editor de textos básico, como puede ser el Bloc de Notas de Windows o cualquier otro editor que admita texto sin formato como Microsoft Wordpad, Notepad++, etc. Existen además, otras herramientas más complejas para la realización de sitios Web o edición de código HTML, como por ejemplo Microsoft FrontPage, Macromedia Dreamweaver, etc.

### **CSS:**

El CSS, siglas de Cascading Style Sheets (u hojas de estilo en cascada), es el lenguaje formal usado para definir la presentación de los documentos estructurados escritos en HTML, XML y XHTML. El W3C, World Wide Web Consortium, es el organismo encargado de formular la especificación que servirán como estándar para los navegadores. La idea de CSS es separar la estructura de un documento de su presentación.

Ventajas:

- Aumento de la legibilidad del documento: Es más fácil de leer y editar puesto que no se mezcla contenido con presentación.
- Disminución del tiempo necesario para cambiar el aspecto global del sitio.
- Aumento de la disponibilidad: Una Web puede disponer de varias hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario.

### **Javascript:**

Javascript es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web con una sintaxis semejante a la del lenguaje Java.

A diferencia de Java, Javascript no es orientado a objetos sino que es más bien un lenguaje basado en prototipos. Todos los navegadores interpretan el código Javascript integrado dentro de las páginas web. Se ejecuta en el usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

## **5.1.2. Capa de lógica de negocio**

### **PHP:**

PHP es un lenguaje de programación interpretado ejecutado en el servidor para la creación de contenidos dinámicos en páginas web.

Cuando el usuario hace una petición al servidor para que le envíe una página web generada por un script PHP, el servidor ejecuta el intérprete de PHP el cual procesa el script solicitado que genera el contenido de forma dinámica, regresando el resultado al servidor, el cual lo enviará al cliente.

## **5.1.3. Capa de persistencia**

### **MySQL:**

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario. Su popularidad como aplicación web está muy ligada a PHP.



## 5.2. Descripción de la implementación

Debido a que la aplicación está dirigida a los alumnos, esta es de acceso privado por lo que se debe estar identificado en todo momento. La única parte pública de la web será el login para permitir la identificación de los alumnos.

### 5.2.1. Parte pública

La parte pública de la aplicación está formada por el fichero login.html.

#### **login.html:**

Es la página encargada de loggear a los alumnos, internamente crea una serie de cookies. Es necesario indicar el nombre, apellidos, dni y curso.



Ilustración 9: login.html

#### **registro.html:**

Es la página encargada de registrar a los alumnos. Es necesario indicar el nombre, apellidos, dni y curso.



Hola anónimo

[Inicio](#)  
[Decimal a Simple precisión](#)  
[Decimal a Doble precisión](#)  
[Simple precisión a Decimal](#)  
[Doble precisión a Decimal](#)  
[Preguntas](#)

## Registrarse:

Nombre:

Apellidos:

DNI:

Curso:

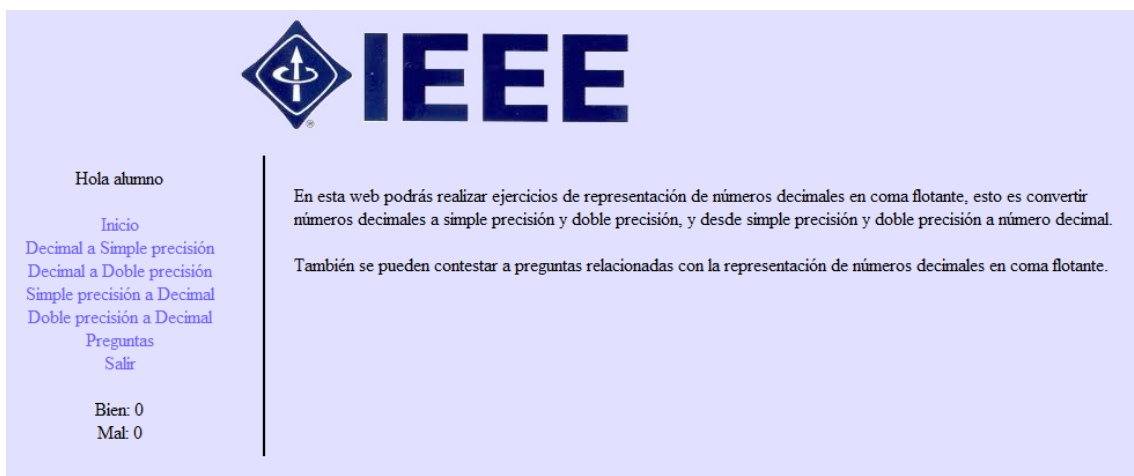
Ilustración 10: registro.html

## 5.2.2. Parte privada

La parte privada de la aplicación está formada por el resto de ficheros.

### index.html:

Es la página inicial de la aplicación donde se describe brevemente el contenido y funciones de la aplicación web.



Hola alumno

[Inicio](#)  
[Decimal a Simple precisión](#)  
[Decimal a Doble precisión](#)  
[Simple precisión a Decimal](#)  
[Doble precisión a Decimal](#)  
[Preguntas](#)  
[Salir](#)

Bien: 0  
Mal: 0

En esta web podrás realizar ejercicios de representación de números decimales en coma flotante, esto es convertir números decimales a simple precisión y doble precisión, y desde simple precisión y doble precisión a número decimal.

También se pueden contestar a preguntas relacionadas con la representación de números decimales en coma flotante.

Ilustración 11: index.html

### decimal2simple.html:

Es la página en la que se realizan ejercicios en los que se convierte un número decimal a simple precisión siguiendo una serie de pasos que se verifican uno a uno.

Estos pasos son:

- Calcular la parte entera en binario.
- Calcular la parte decimal en binario.
- Magnitud sin normalizar.
- Magnitud normalizada.
- Exponente en entero.
- Exponente con exceso en entero.
- Exponente con exceso en binario.
- Signo en binario.
- Simple precisión en binario.
- Simple precisión en hexadecimal.

Pueden generarse números aleatorios para realizar el ejercicio o bien introducir uno manualmente. En cualquier momento durante el proceso puede pulsarse el botón de “Nuevo” para realizar un nuevo ejercicio, y también el botón “Ver solución” para mostrar todos los pasos resueltos.

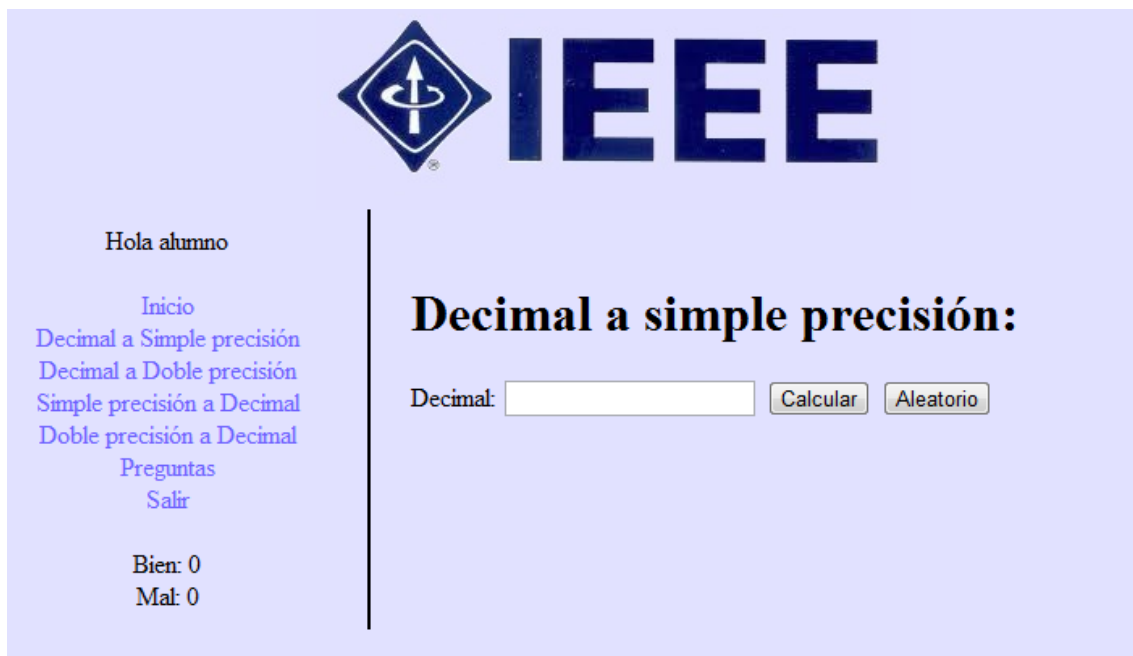


Ilustración 12: decimal2simple.html

### **decimal2doble.html:**

Es la página en la que se realizan ejercicios en los que se convierte un número decimal a doble precisión siguiendo una serie de pasos que se verifican uno a uno.

Estos pasos son:

- Calcular la parte entera en binario.
- Calcular la parte decimal en binario.
- Magnitud sin normalizar.
- Magnitud normalizada.
- Exponente en entero.
- Exponente con exceso en entero.

- Exponente con exceso en binario.
- Signo en binario.
- Doble precisión en binario.
- Doble precisión en hexadecimal.

Pueden generarse números aleatorios para realizar el ejercicio o bien introducir uno manualmente. En cualquier momento durante el proceso puede pulsarse el botón de “Nuevo” para realizar un nuevo ejercicio, y también el botón “Ver solución” para mostrar todos los pasos resueltos.

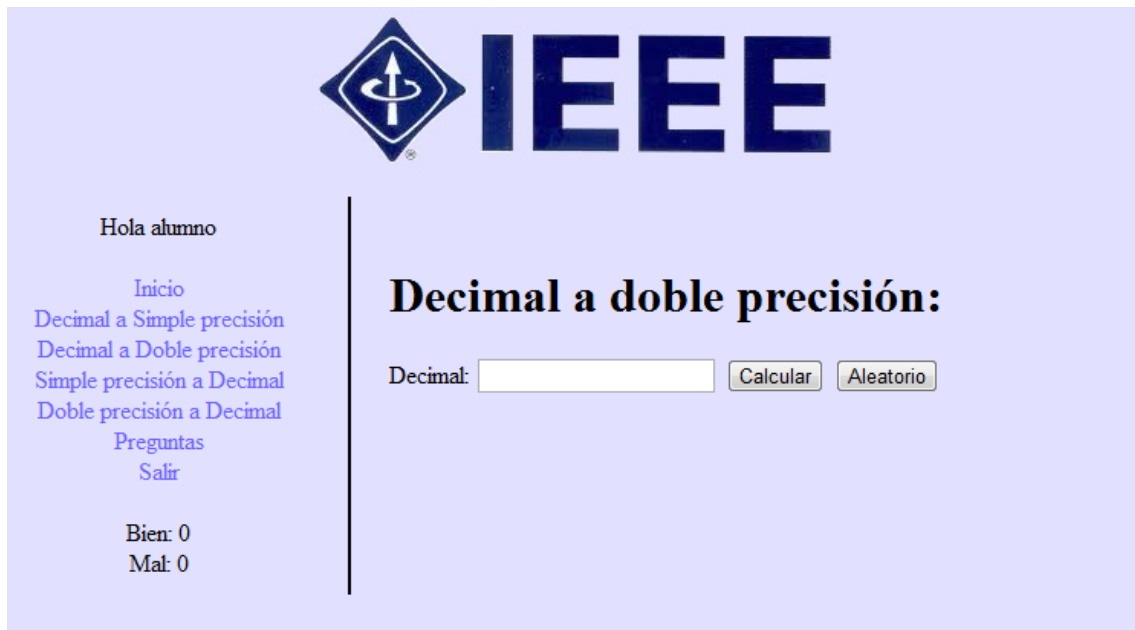


Ilustración 13: decimal2doble.html

### simple2decimal.html:

Es la página en la que se realizan ejercicios en los que se convierte un número decimal representado en simple precisión (en hexadecimal) a decimal siguiendo una serie de pasos que se verifican uno a uno.

Estos pasos son:

- Simple precisión en binario.
- Signo en binario.
- Exponente con exceso en binario.
- Exponente con exceso en entero.
- Exponente en entero.
- Magnitud normalizada.
- Magnitud sin normalizar.
- Decimal.

Pueden generarse números hexadecimales aleatorios para realizar el ejercicio o bien introducir uno manualmente. En cualquier momento durante el proceso puede pulsarse el botón de “Nuevo” para realizar un nuevo ejercicio, y también el botón “Ver solución” para mostrar todos los pasos resueltos.

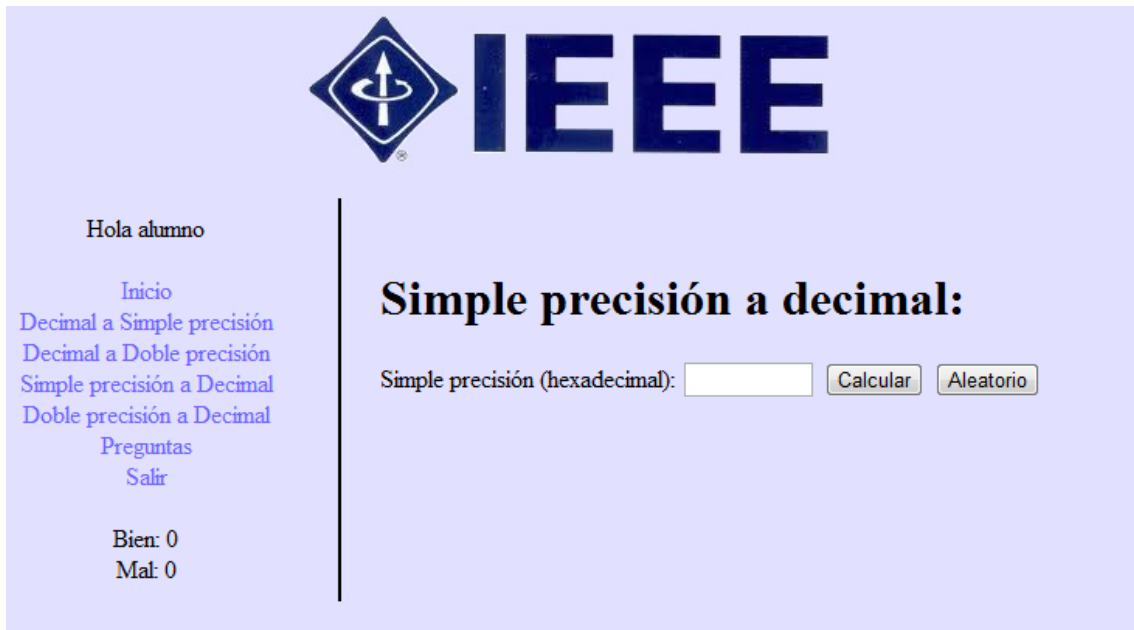


Ilustración 14: simple2decimal.html

**doble2decimal.html:**

Es la página en la que se realizan ejercicios en los que se convierte un número decimal representado en doble precisión (en hexadecimal) a decimal siguiendo una serie de pasos que se verifican uno a uno.

Estos pasos son:

- Doble precisión en binario.
- Signo en binario.
- Exponente con exceso en binario.
- Exponente con exceso en entero.
- Exponente en entero.
- Magnitud normalizada.
- Magnitud sin normalizar.
- Decimal.

Pueden generarse números hexadecimales aleatorios para realizar el ejercicio o bien introducir uno manualmente. En cualquier momento durante el proceso puede pulsarse el botón de “Nuevo” para realizar un nuevo ejercicio, y también el botón “Ver solución” para mostrar todos los pasos resueltos.



Ilustración 15: doble2decimal.html

**preguntas.html:**

Es la página en la que se realizan preguntas sobre el estándar IEEE 754 que deben ser respondidas por el alumno.



Ilustración 16: preguntas.html

**logout.php:**

Es la página que permite dejar de identificar al alumno destruyendo la cookie creada en el proceso de login.

Código:

```
<?php
    setcookie("nombre", "");
    setcookie("apellidos", "");
    setcookie("dni", "");
    setcookie("curso", "");
    header("Location: login.html?error=No%20esta%20autenticado");
?>
```

### 5.2.3. Otras páginas

Se compone por otras páginas web de la aplicación que ejecutan funcionalidad de forma oculta al usuario.

**config.php:**

Contiene los datos de conexión al SGBD MySQL para acceder a la base de datos y leer/escribir información. Se incluye en todas las páginas que requieren una conexión a la base de datos.

Código:

```
<?php
    $host = "host";
    $user = "usuario";
    $password = "contraseña";
    $db_name = "base-de-datos";
?>
```

**login.php:**

Realiza el proceso de login verificando los datos que se le pasan como parámetros (nombre, apellidos, dni y curso). Se ejecuta desde la página web login.html.

Código:

```
<?php
    include_once('config.php');

    function CompararString($string1, $string2) {
        $string1 = strtolower($string1);
        $string1 = str_replace("á", "a", $string1);
        $string1 = str_replace("é", "e", $string1);
        $string1 = str_replace("í", "i", $string1);
        $string1 = str_replace("ó", "o", $string1);
        $string1 = str_replace("ú", "u", $string1);
        $string2 = strtolower($string2);
        $string2 = str_replace("á", "a", $string2);
        $string2 = str_replace("é", "e", $string2);
        $string2 = str_replace("í", "i", $string2);
        $string2 = str_replace("ó", "o", $string2);
        $string2 = str_replace("ú", "u", $string2);
        return $string1==$string2;
    }
```

```

    }

    function EsDni($string) {
        $permitidos = "0123456789";
        if (strlen($string) != 8) return false;
        for ($i=0; $i<strlen($string); $i++) {
            if (strpos($permitidos,
substr($string,$i,1))===false) {
                return false;
            }
        }
        return true;
    }

    $nombre = $_GET['nombre'];
    $apellidos = $_GET['apellidos'];
    $dni = $_GET['dni'];
    $curso = $_GET['curso'];

    if (!EsDni($dni)) {
        header("Location:
login.html?error=E1%20dni%20tiene%20caracteres%20incorrectos");
    }

    $connection = mysql_connect($host, $user, $password) or
die(mysql_error());
    mysql_select_db($db_name, $connection) or die(mysql_error());
    $data = mysql_query("SELECT * FROM alumnos WHERE dni='" . $dni .
"", $connection) or die(mysql_error());
    mysql_close($connection) or die(mysql_error());

    $login = false;
    echo mysql_num_rows($data);
    if (mysql_num_rows($data) == 1) {
        $row = mysql_fetch_array($data);
        if (CompararString($row['nombre'], $nombre) &&
CompararString($row['apellidos'], $apellidos) && ($row['curso'] ==
$curso)) {
            setcookie("nombre", $row['nombre'],
time()+365*24*60*60);
            setcookie("apellidos", $row['apellidos'],
time()+365*24*60*60);
            setcookie("dni", $row['dni'], time()+365*24*60*60);
            setcookie("curso", $row['curso'],
time()+365*24*60*60);
            session_start();
            $login = true;
        }
    }

    if ($login) {
        header("Location: index.html");
    } else {
        header("Location: login.html?error=Datos%20incorrectos");
    }

?>

```



**registrarejercicio.php:**

Es el script encargado de registrar en la base de datos los resultados de los ejercicios realizados y preguntas resueltas por el alumno.

Código:

```
<?php
    include_once('config.php');

    $ejercicio = $_POST['ejercicio'];
    $tipo = $_POST['tipo'];
    $resultado = $_POST['resultado'];

    $connection = mysql_connect($host, $user, $password);
    mysql_select_db($db_name, $connection) or die(mysql_error());
    mysql_query("INSERT INTO
ejercicios(`dni`,`resultado`,`tipo`,`ejercicio`)VALUES('" .
$_COOKIE['dni'] . "','" . $resultado . "','" . $tipo . "','" .
$ejercicio . "')", $connection);
    mysql_close($connection);
?>
```

**registro.php:**

Es el script encargado de registrar a un alumno en la base de datos para que posteriormente pueda identificarse en la aplicación web.

Código:

```
<?php
    include_once('config.php');

    function CompararString($string1, $string2) {
        $string1 = strtolower($string1);
        $string1 = str_replace("á", "a", $string1);
        $string1 = str_replace("é", "e", $string1);
        $string1 = str_replace("í", "i", $string1);
        $string1 = str_replace("ó", "o", $string1);
        $string1 = str_replace("ú", "u", $string1);
        $string2 = strtolower($string2);
        $string2 = str_replace("á", "a", $string2);
        $string2 = str_replace("é", "e", $string2);
        $string2 = str_replace("í", "i", $string2);
        $string2 = str_replace("ó", "o", $string2);
        $string2 = str_replace("ú", "u", $string2);
        return $string1==$string2;
    }

    function EsNombreApellido($string) {
        $permitidos =
"abcdefghijklmñopqrstuvwxyzABCDEFGHIJKLMNÑOPQRSTUVWXYZáéíóúĂĔÍÓÚ ";
        if (strlen($string) != 8) return false;
        for ($i=0; $i<strlen($string); $i++) {
            if (strpos($permitidos,
substr($string,$i,1))===false) {
                return false;
            }
        }
        return true;
    }
```

```

    }

    function EsDni($string) {
        $permitidos = "0123456789";
        if (strlen($string) != 8) return false;
        for ($i=0; $i<strlen($string); $i++) {
            if (strpos($permitidos,
substr($string,$i,1))===false) {
                return false;
            }
        }
        return true;
    }

    $nombre = $_GET['nombre'];
    $apellidos = $_GET['apellidos'];
    $dni = $_GET['dni'];
    $curso = $_GET['curso'];

    if (!EsNombreApellido($nombre)) {
        header("Location:
login.html?error=El%20nombre%20tiene%20caracteres%20incorrectos");
    }
    if (!EsNombreApellido($apellidos)) {
        header("Location:
login.html?error=Los%20apellidos%20tienen%20caracteres%20incorrectos")
;
    }
    if (!EsDni($dni)) {
        header("Location:
login.html?error=El%20dni%20tiene%20caracteres%20incorrectos");
    }
    if (!($curso>=1 && $curso<=3)) {
        header("Location:
login.html?error=El%20curso%20no%20es%20valido");
    }

    $connection = mysql_connect($host, $user, $password) or
die(mysql_error());
    mysql_select_db($db_name, $connection) or die(mysql_error());
    mysql_query("INSERT INTO alumnos
(nombre,apellidos,dni,curso)VALUES('" . $nombre . "','" . $apellidos .
"', '" . $dni . "','" . $curso . "')", $connection) or
die(mysql_error());
    mysql_close($connection) or die(mysql_error());

    header("Location: login.html");
?>

```

**style.css:**

Es la hoja de estilos en cascada que forma el diseño de las páginas de la aplicación. Es común a todas ellas y permite separar el contenido del diseño.

Código:

```

#contenedor {
    margin:0 auto;
    width: 1024px;
}

```

```
#cabecera {
  float: left;
  width: 800px;
  text-align: center;
}
#menu {
  float: left;
  width: 228px;
  border-right-width: 2px;
  border-right-style: solid;
  border-right-color: #000000;
  text-align: center;
  padding: 10px;
}
#contenido {
  float: left;
  width: 724px;
  padding: 25px;
  text-align: left;
}
#menu a {
  color: #6D62FF;
  text-decoration: none;
}
body {
  background-color: #E1E1FF;
  text-align: center;
}
```

### **decimal2ieee.js:**

Contiene las funciones necesarias en Javascript para realizar los ejercicios de conversión de decimal a simple y doble precisión. Estas funciones son:

-InicializarEjercicio: Inicializa un nuevo ejercicio ocultando los pasos para mostrarlos posteriormente uno a uno.

-Nuevo: Inicializa un nuevo ejercicio controlando las estadísticas.

-Aleatorio: Inicializa un nuevo ejercicio con un número decimal aleatorio.

-Check: Se encarga de comprobar cada uno de los pasos realizados en la resolución del ejercicio para desbloquear el siguiente paso.

-ParteDecimal2Binario: Función auxiliar en el proceso de conversión de decimal a simple y doble precisión que se encarga de convertir la parte decimal del número decimal a binario.

-Normalizar: Función auxiliar en el proceso de conversión de decimal a simple y doble precisión que se encarga de normalizar un número binario desnormalizado.

-DecimalASimplePrecision: Función principal que convierte un número decimal a simple precisión.

-DecimalADoblePrecision: Función principal que convierte un número decimal a doble precisión.

-VerSolución: Función que muestra la solución del ejercicio actual.

**ejercicios.js:**

Contiene funciones comunes a todos los tipos de ejercicios, excepto las preguntas sobre el estándar. Estas funciones son:

- QuitarCerosIzquierda: Quita los ceros a la izquierda de un numero entero en binario.
- QuitarCerosDerecha: Quita los ceros a la derecha de la parte decimal de un numero en binario.
- QuitarCeros: Quita los ceros a ambos lados de un numero en binario.
- ExtraerExponente: Extrae los bits de la representación en binario en simple o doble precisión que forman el exponente.
- ExtraerMantisa: Extrae los bits de la representación en binario en simple o doble precisión que forman la mantisa.
- EsHexadecimal: Comprueba si un numero es o no hexadecimal.
- Binario2Signo: Calcula el signo de un número binario.
- Signo2Binario: Calcula la representación binaria del signo de un número.
- Binario2Entero: Convierte a entero un número binario.
- Entero2Binario: Convierte a binario un número entero.
- Binario2Hexadecimal: Convierte a hexadecimal un número binario.
- Hexadecimal2Binario: Convierte a binario un número hexadecimal.
- DecimalAleatorioPrivado: Devuelve un número decimal aleatorio.
- DecimalAleatorio: Crea un decimal para convertir a simple o doble precisión.
- HexadecimalAleatorio: Crea un hexadecimal aleatorio para convertir a decimal.

**ieee2decimal.js:**

Contiene las funciones necesarias en Javascript para realizar los ejercicios de conversión de simple y doble precisión a decimal. Estas funciones son:

- InicializarEjercicio: Inicializa un nuevo ejercicio ocultando los pasos para mostrarlos posteriormente uno a uno.
- Nuevo: Inicializa un nuevo ejercicio controlando las estadísticas.
- Aleatorio: Inicializa un nuevo ejercicio con un número hexadecimal aleatorio.
- Check: Se encarga de comprobar cada uno de los pasos realizados en la resolución del ejercicio para desbloquear el siguiente paso.
- Binario2ParteDecimal: Función auxiliar en el proceso de conversión de simple y doble precisión a decimal que se encarga de obtener la parte decimal de un número binario.
- SimplePrecisionADecimal: Función principal que convierte un número en simple precisión a decimal.

-DoblePrecision2Decimal: Función principal que convierte un número en doble precisión a decimal.

-VerSolución: Función que muestra la solución del ejercicio actual.

#### **preguntas.js:**

Contiene las funciones Javascript para la realización de las preguntas sobre el estándar. Estas funciones son:

-CargarPreguntas: Crea todas las preguntas con sus posibles respuestas indicando cual es la correcta.

-ElegirPregunta: Elige una pregunta al azar entre todas las existentes.

-Check: Comprueba si una pregunta ha sido respondida correcta o incorrectamente.

#### **registrarestadisticas.js:**

Contiene las funciones necesarias para realizar el seguimiento y registro de estadísticas. Estas funciones son:

-MostrarEstadisticas: Muestra las estadísticas del alumno (ejercicios y preguntas resueltos correcta e incorrectamente).

-SumarBien: Aumenta la estadística de ejercicios y preguntas resueltos correctamente.

-SumarMal: Aumenta la estadística de ejercicios y preguntas resueltos incorrectamente.

-EnviarEstadistica: Cada vez que se finaliza un ejercicio o pregunta de forma correcta o incorrecta se ejecuta esta función que envía el resultado al servidor para ser guardado en la base de datos.

#### **sessvars.js:**

Contiene las funciones necesarias para mantener el contador de estadísticas mientras el alumno navega por la aplicación web y resuelve ejercicios y preguntas. Este es un archivo creado por Thomas Frank que permite crear variables de sesión con Javascript. El código puede descargarse y utilizarse gratuitamente desde <http://www.thomasfrank.se/sessionvars.html>

#### **web.js:**

Contiene las funciones Javascript relacionadas con la identificación del alumno. Estas funciones son:

-LeerCookie: Lee la cookie que se indica para obtener su valor.

-Registro: Registra un nuevo alumno con los datos del formulario en la página registro.html.

-Login: Realiza el login de un alumno con los datos del formulario en la página login.html.

-IsLogged: Comprueba si un usuario está identificado como alumno para permitirle el acceso a la aplicación web. Si no lo está le reenvía al formulario de acceso.

-MostrarParametros: Lee los parámetros recibidos por la url y los procesa para mostrar los errores de acceso en la página que contiene el formulario de acceso.

## 6. Evaluación

En esta etapa de evaluación concluye el ciclo de vida del proyecto y lo deja preparado para su uso. Esta etapa medirá el nivel de calidad que ofrece la aplicación, donde se comprobará el código implementado.

### 6.1. Prueba de validación de CSS

Puesto que se ha utilizado una hoja de estilo para la interfaz de la aplicación web, es necesario proceder a la validación del código mediante una prueba de validación desde el siguiente enlace: <http://jigsaw.w3.org/css-validator/>

**¡Enhorabuena! No error encontrado.**

¡Este documento es CSS versión 2.1 válido!

Puede mostrar este icono en cualquier página que valide para que los usuarios vean que se ha preocupado por crear una página Web interoperable. A continuación se encuentra el XHTML que puede usar para añadir el icono a su página Web:

```
<p>
  <a href="http://jigsaw.w3.org/css-validator/check/referer">
    
  </a>
</p>
```

```
<p>
  <a href="http://jigsaw.w3.org/css-validator/check/referer">
    
  </a>
</p>
```

Ilustración 17: Validación CSS

### 6.2. Prueba de validación de navegadores web

Se ha comprobado el correcto de la aplicación en Microsoft Internet Explorer, Mozilla Firefox y Google Chrome. Debido a la simplicidad del diseño CSS no ha habido problemas de compatibilidad entre navegadores web.



Ilustración 18: Validación navegador web Microsoft Internet Explorer



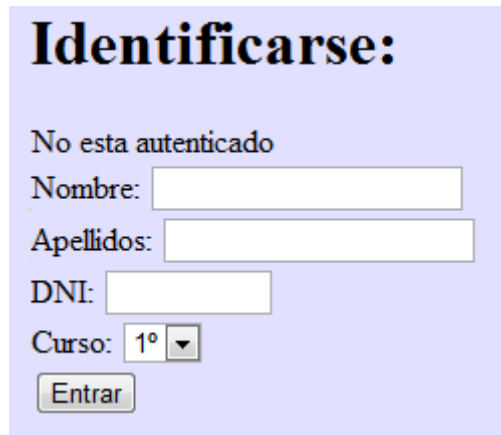
Ilustración 19: Validación navegador web Mozilla Firefox



Ilustración 20: Validación navegador web Google Chrome

### 6.3. Prueba de seguridad de acceso

Para acceder a la aplicación web es necesario estar registrado e identificado como alumno. En caso de identificarse satisfactoriamente, nos reenvía a la página inicial “index.html” con las cookies necesarias para mantenernos identificados en toda la aplicación, en caso contrario, se vuelve a la página de login “login.html” con un mensaje indicando que los datos son incorrectos.



**Identificarse:**

No esta autenticado

Nombre:

Apellidos:

DNI:

Curso: 1° ▼

Ilustración 21: Validación de seguridad



**Identificarse:**

Datos incorrectos

Nombre:

Apellidos:

DNI:

Curso: 1° ▼

Ilustración 22: Validación de seguridad (error de login)

## 7. Conclusiones

### 7.1. Trabajo realizado

La aplicación responde a las expectativas y requerimientos recogidos en la especificación de requisitos. El trabajo realizado ofrece un alto nivel de cumplimiento de objetivos.

La aplicación ofrece al usuario una interfaz intuitiva y de fácil manejo. La sencillez en los pasos durante la realización de los ejercicios facilitan el aprendizaje del alumno.

### 7.2. Valoración personal

Durante el desarrollo del proyecto de final de carrera he podido apreciar cual es el proceso para la creación de una aplicación web: recabar la especificación de requisitos, planificación, implementación y pruebas.

Ha sido un trabajo que me ha llevado más tiempo del que esperaba, pues parecía mucho más sencillo, y aunque no ha quedado perfecto estoy muy satisfecho con el resultado.



Con este proyecto final de carrera he tenido que aprender Javascript a nivel básico, ampliando los lenguajes de programación y tecnologías conocidas previamente para el desarrollo de páginas web.

### 7.3. Futuras mejoras

A continuación se expone un listado con posibles mejoras que podrían hacerse en el futuro:

- Mejorar el aspecto visual de la aplicación.

- Mejorar el proceso de resolución de ejercicios para que sea más eficaz y muestre mejor los errores cometidos por el alumno.

## 8. Bibliografía

- IEEE Std 830 - IEEE Guide to Software Requeriments Specifications. IEEE Standards Board. 345 Eas 47 th Street. New York, NY 10017, USA. 1984.

- Página oficial de WAMP: <http://www.wampserver.com>

- Wikipedia: <http://es.wikipedia.org>

## Anexos

### A. Anexo I (Herramientas usadas)

#### ArgoUML

ArgoUML es una aplicación de diagramado UML escrita en Java, por lo que está disponible en cualquier plataforma soportada por Java. Algunas características:

- Multilenguaje.
- Soporte para el lenguaje de generación de Código: Java, PHP, Python, C++ y C#.
- Generación de ficheros PNG, GIF, JPG, SVG, EPS desde diagramas.

Pero también tiene algunas desventajas: no tiene botón “deshacer” y los modelos a veces no pueden ser re-abiertos, entre otras.

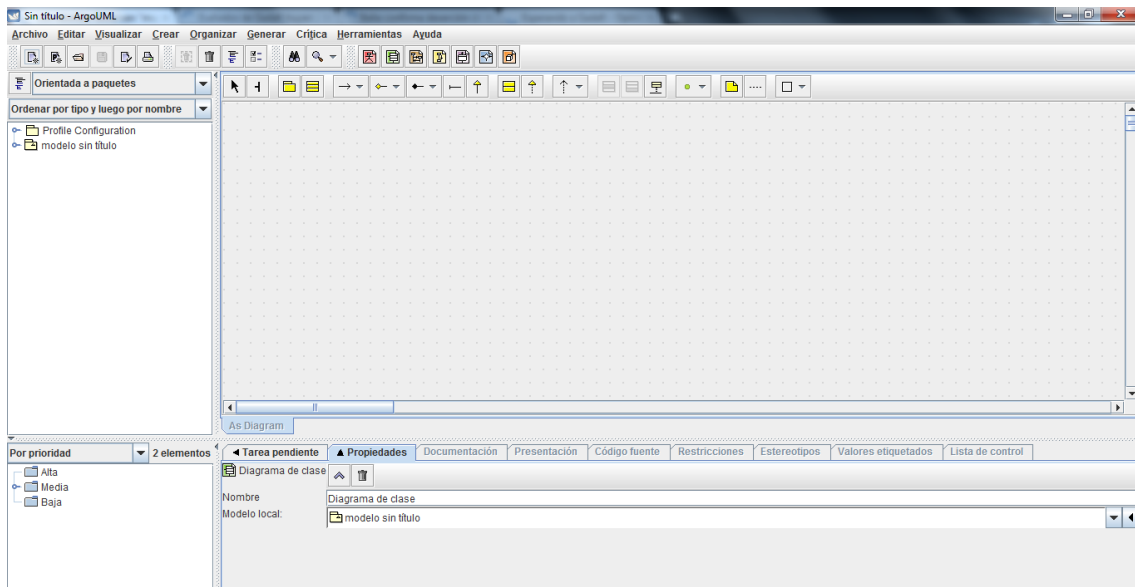


Ilustración 23: ArgoUML

#### Balsamiq Mockups

Balsamiq es una aplicación AIR que permite al usuario arrastrar y soltar elementos de la biblioteca de elementos comunes de diseño a un área de trabajo, permitiendo exportar el producto final en formato PNG.

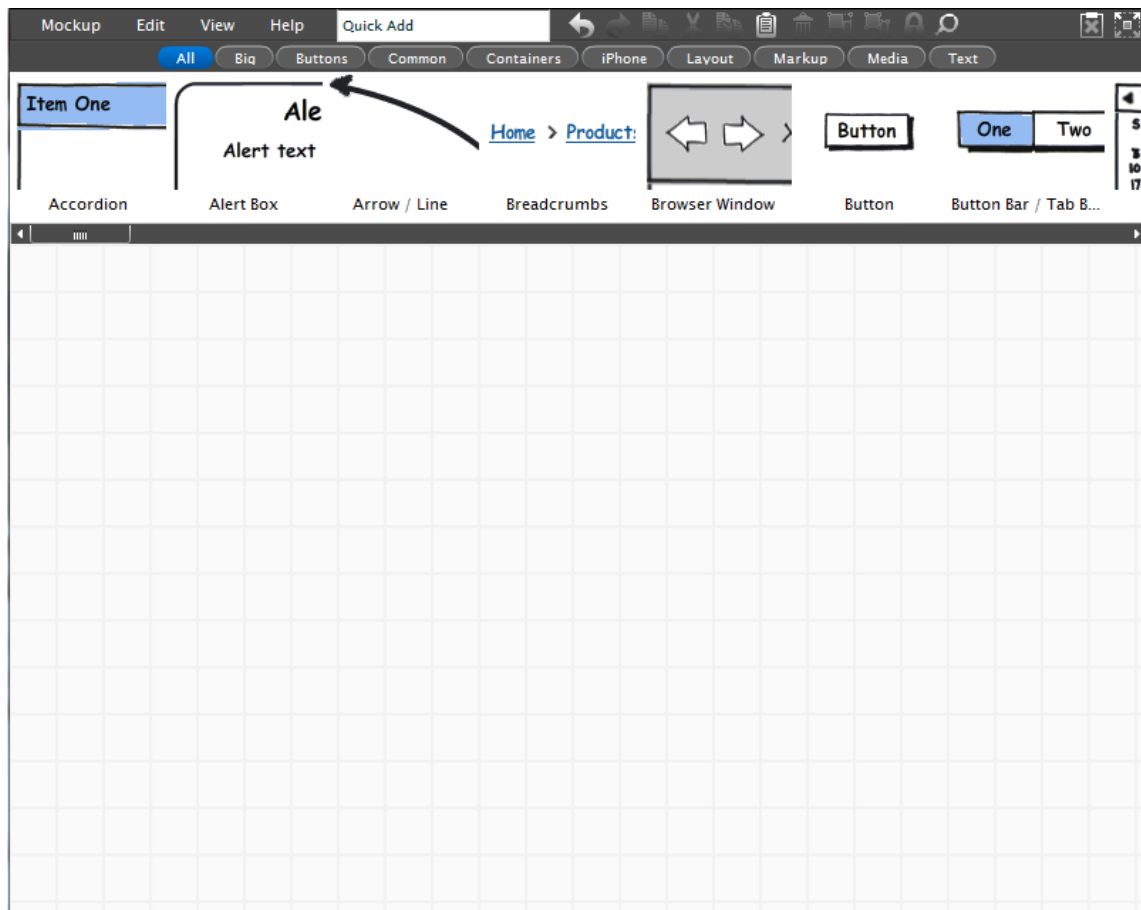


Ilustración 24: Balsamiq Mockups