

Universidad Complutense de Madrid.

DG ADE + INGENIERÍA INFORMÁTICA

Asignatura: Aplicaciones Web



PRÁCTICA 3 FLEXAM: Arquitectura y prototipo funcional del proyecto

Miembros:

Daniel García Miguel

Martin Veselinov Georgiev

Alejandro Paniagua López

Alejandro de Mateo Bodegas

17/04/2024

ÍNDICE

1. Introducción.....	3
2. Listado de scripts para las vistas (MODIFICAR).....	5
2.1. Funcionalidad común.....	5
2.2. Página principal.....	6
2.3. Información y sobre nosotros.....	7
2.4. Buscar y realizar tests.....	8
2.5. Estadísticas.....	10
2.6. Diagrama de scripts de vistas.....	11
3. Listado de scripts adicionales (MODIFICAR).....	12
- sign_up_logic.php:.....	12
- login_logic.php:.....	12
- get_test.php:.....	12
Este script verifica si el usuario ha iniciado sesión y, de ser así, obtiene los tests disponibles para dicho usuario. Ejecuta una consulta SQL que selecciona el ID del test, su título, el nombre de la asignatura asociada y el número de intentos realizados por el usuario en cada test.....	12
- obtener_datos_intentos.php:.....	12
4. Estructura de la BBDD.....	13
4.1. Usuarios y Relaciones Académicas:.....	13
4.2. Evaluaciones y Preguntas:.....	13
4.3. Interacción Usuario-Test:.....	14
5. Prototipo funcional del proyecto.....	14
5.1. Página principal e información.....	15
5.2. Login o registro.....	16
5.3. Tests.....	17
5.4. Estadísticas.....	18
5.5. Menú de usuarios (Nueva implementación).....	19
6. Datos de acceso de los usuarios y sección VPS.....	21
7. Actividades.....	21

1. Introducción

En esta tercera práctica hemos abordado una refactorización del código que planteamos en la Práctica 2, tratando de adaptarnos al modelo de 3 Capas en el que básicamente estaremos empleando los siguiente tipos de archivos:

- SA: servicio de aplicación: objetos con las operaciones alto nivel(controlador)
- DAO: Data Access Object: patrón de diseño especializado en persistir los datos: acceso a la BBDD
- TO: "Transfer Object" para objetos básicos: atributos, getters y setters

Tratando de aplicar la programación web orientada a objetos empleando Objetos para la funcionalidad fundamental de nuestra aplicación en lugar de la división en comun/logic/views que seguíamos antiguamente.

El patrón Singleton aparece también implementado mediante Aplicacion.php, que hace esta labor en esta entrega.

Además, hemos añadido nueva funcionalidad disponible en el punto 5. Prototipo funcional del proyecto.

Esta nueva funcionalidad se relaciona con los scripts *resultados_test.php* y *user_menu.php* que hemos añadido.

- **includes**
 - **BBDD**
 - Flexam.sql
 - **comun:**
 - config.php
 - footer.php
 - header.php
 - **DAO: Faltaría completar la implementación completa de todos los scripts y relacionarlos con la funcionalidad que debería ofrecer la aplicación de cara a la entrega final.**
 - DAOAsignatura
 - DAOGrado
 - DAOIntento
 - DAOOpcion
 - DAOPregunta
 - DAOTest
 - DAOUniversidad
 - DAOUsuario
- **TO: De igual manera, la implementación completa queda sujeta a la entrega final.**

- AsignaturaTO
- GradoTO
- IntentoTO
- OpcionTO
- PreguntaTO
- TestTO
- UniversidadTO
- UsuarioTO
- **SA:**
 - Aplicacion.php
- **logic: (después de la refactorización completa esta carpeta quedaría desaparecida, viéndose sustituida por las Clases y Objetos necesarios en la carpeta DAO, TO y SA.).**

La funcionalidad no quedaría excluida, sino que quedaría dividida entre las nuevas clases que creamos con el modelo de 3 capas.

 - fetch_grados.php
 - get_tests.php
 - login_logic.php
 - logout.php
 - obtener_datos_intentos.php
 - procesar_test.php
 - sign_up_logic.php
 - signupHandler.php

Fuera del antiguo directorio view de la segunda entrega hemos dejado las vistas de la aplicación:

- aboutus.php
- index.php
- info.php
- login.php
- user_menu.php
- menu_tests.php
- sign_up.php
- realize_test.php
- resultados_test.php
- resultados_totales.php
- **resources:**
 - **imagenes:**
 - alex.png
 - dani.png
 - estudiantes.jpg
 - flexam_sinfondo.png

- logo_letras.png
- logo_sombra.png
- martin.png
- pani_cut.png
- **styles:**
 - styles.css

En donde, en **negrita** señalamos los directorios principales, los cuales son BBDD, code, resources y styles.

Dentro del primero encuadramos nuestra base de datos SQL, que posteriormente describiremos, para garantizar el funcionamiento correcto de nuestro prototipo funcional.

En “code”, se encuentran todos los scripts que hacen que la aplicación funcione correctamente, encontrando tres subdirectorios en **negrita y cursiva**: comun, logic y views.

El directorio “comun” está dedicado a tres scripts que son empleados tanto por los llamados en esta Práctica “scripts adicionales” como por los scripts para las vistas, como pudieran ser el header, el footer o config.php, que es el encargado de poner a funcionar nuestra base de datos.

Entrega 2:

En el directorio “logic” tenemos encapsulados los scripts adicionales encargados de encapsular el back-end de nuestra aplicación y en el directorio “views” tenemos aquellos scripts que son más exclusivos y centrados en las vistas que tendrá la aplicación. Si bien es cierto que estos scripts también harán uso de los “scripts adicionales” para conectar con la base de datos cuando es necesario y hace falta mostrar cierta información por pantalla, también son los scripts principales que encapsulan los html necesarios para organizar la información mostrada en las diferentes páginas. Posteriormente describimos la funcionalidad e implementación de cada uno.

Entrega 3:

Tenemos un directorio includes que hace la encapsulación del resto de carpetas importantes que tiene nuestro proyecto, sobre todo teniendo en cuenta el modelo de 3 capas.

En cuanto al directorio “resources”, tenemos los archivos de recurso, que en nuestro caso son imágenes, como pudieran ser diversos logos, fotos de los miembros, etc.

Por último, tal y como está indicado, empleamos CSS para gestionar los estilos de nuestra aplicación y es por ello que incluimos el directorio “styles”

que contendrá el archivo `styles.css` con los estilos y la descripción de cada uno de ellos para las diferentes estructuras que estamos empleando en los scripts de “code” anteriormente mencionados, especialmente los capturados dentro de “views”.

Como mencionaremos en el apartado dedicado al prototipo funcional actual, de momento, al tratarse del 25% del Proyecto final, nuestra funcionalidad está bastante limitada, y es que el usuario que se registre en nuestra página podrá resolver test de SU universidad y de SU grado que ya hayamos subido nosotros previamente, mediante nuestra base de datos, de modo que no podrá crearlos.

Del mismo modo, tampoco tendrá estadísticas muy detalladas a la hora de resolverlos y todavía no se contempla el rol de “Creador de tests”, pues todos los usuarios que puedan interactuar con FLEXAM solo pueden resolver los tests de su universidad y titulación por el momento.

2. Listado de scripts para las vistas

En este capítulo de la memoria, se detallarán y explicarán todos los scripts de vistas que se implementarán, clasificados por funcionalidades, con el propósito de respaldar la interacción en la web. Se proporcionará una breve descripción de cada script, indicando su utilidad y a qué funcionalidad pertenece. Hemos realizado la división en cuanto a funcionalidades comunes, página principal, información Sobre Nosotros, Realización de los tests, y por último, Estadísticas, que son las mostradas al finalizar un test.

2.1. Funcionalidad común

En este apartado se detallan los scripts comunes a todos los demás (Son usados por el resto). Estos scripts incluyen funciones y herramientas compartidas que son esenciales para el funcionamiento general de la aplicación.

- **header.php:**

Encapsula toda la información referente al menú mostrado en la parte superior de todas nuestras páginas, siendo una vista común a todas ellas. En este caso hemos decidido incluir un logo que hemos hecho, acompañado de un delimitador, para posteriormente introducir de manera centrada los botones del menú de navegación. Este menú está fijo en su posición relativa, apareciendo siempre en la parte superior dando igual el scroll del usuario. Se puede consultar el contenido en `header.php` y los estilos en `styles.css`.



Captura de pantalla del header

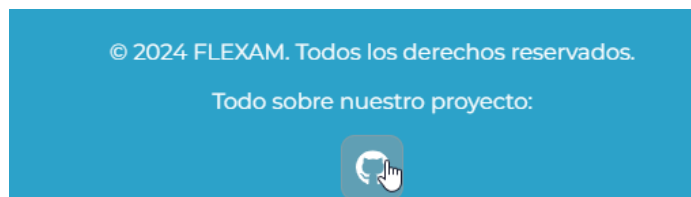
Además, tendremos un header distinto, o una visualización distinta si el usuario se ha registrado y ha iniciado sesión. Dándole la opción de salir desde el menú saludando con su nombre, lo cual hemos hecho con un bloque php dentro del script.

- **footer.php:**

Del mismo modo que el header, común a todas las vistas y páginas. En este caso incluimos información básica y un botón a nuestro Proyecto de la Práctica en github. Este botón lo hemos incluido extrayendo el HTML y CSS de una página online de recursos. Cuya URL es la siguiente:

<https://uiverse.io/vinodjangid07/wonderful-yak-52>

El bloque de estilos que se da lo hemos incluido en nuestro styles.css y el bloque de html viene reflejado en el código de footer.php.



Captura de pantalla del footer

2.2. Página principal

La página principal es la puerta de entrada a la aplicación y proporciona una visión general de las funcionalidades disponibles.

- **index.php:**

Aquí es donde viene contenido todo el código referente al HOME de nuestra aplicación. En este caso, se trata de una página de inicio sencilla, en la que fundamentalmente queremos ser ilustrativos con el usuario para que pueda acceder a las diversas áreas funcionales disponibles que tenemos, que son Tests, Sobre Nosotros, Info y Login, para lo cual hemos añadido una serie de botones en la parte inferior. Además, hemos añadido unos efectos de "hover" en la imagen y el cuadro de texto descritos en styles.css.



Captura de pantalla de index.php

2.3. Información y sobre nosotros

Esta funcionalidad ofrece a los usuarios información adicional sobre la aplicación y el equipo detrás de ella. Los scripts correspondientes se encargan de presentar esta información de manera clara y accesible, proporcionando detalles sobre el propósito y el funcionamiento de la plataforma, así como sobre el equipo responsable de su desarrollo.

- **info.php:**

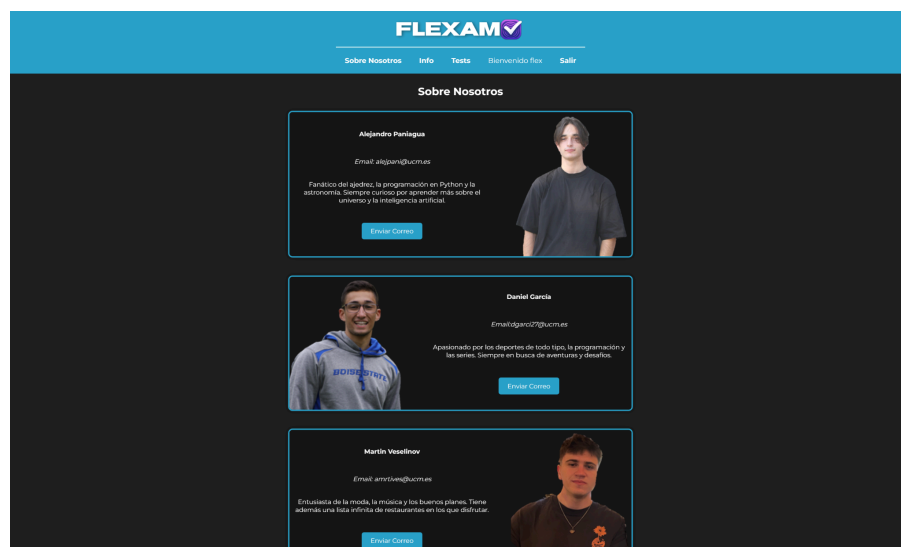
Esta vista es la que describe más detalladamente el funcionamiento actual de nuestro prototipo de aplicación. Empleando el mismo estilo de cuadro de texto translúcido que empleamos en index.php:



Captura de pantalla de info.php

- **aboutus.php:**

Se trata de una vista heredada en parte de nuestra anterior práctica, en donde mostramos a los miembros del equipo con una breve descripción y una fotografía.



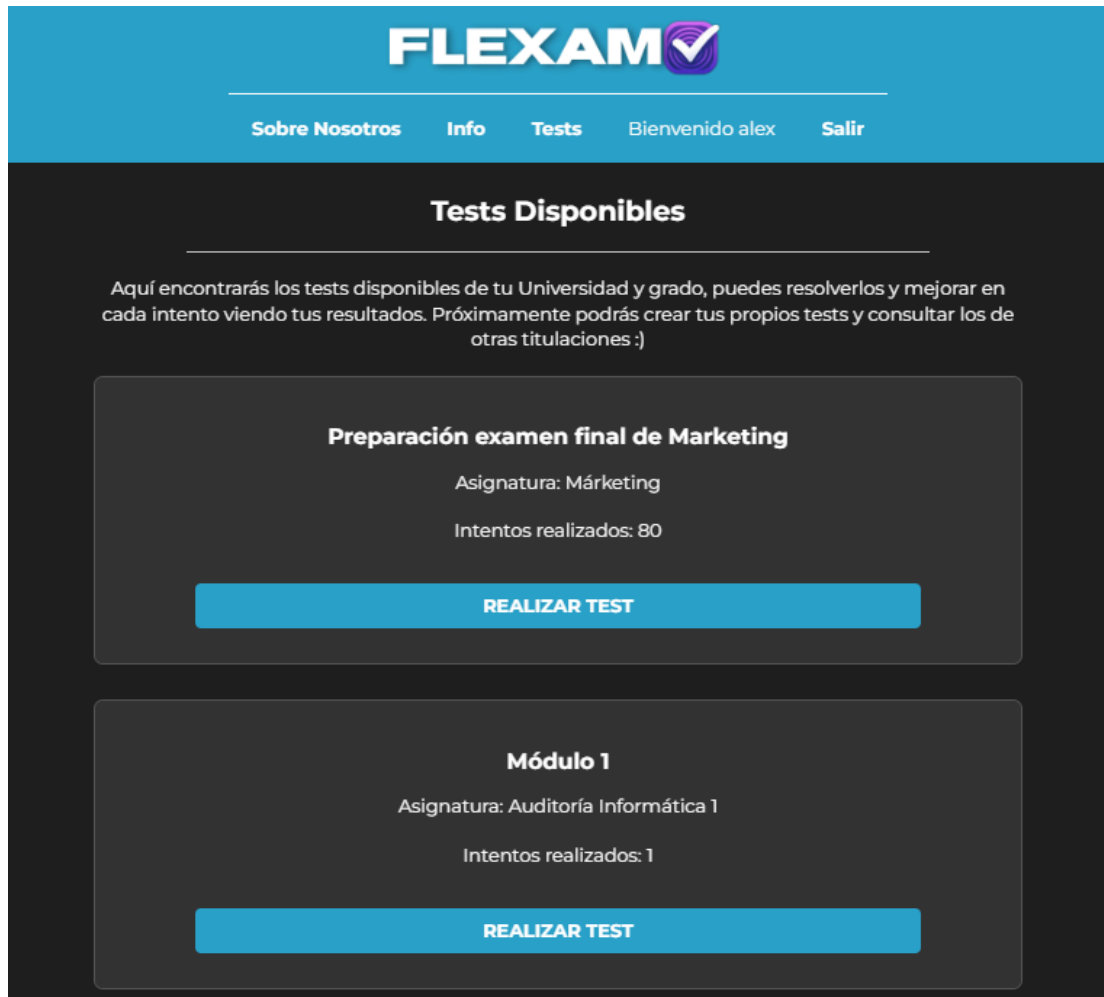
Captura de pantalla de aboutus.php

2.4. Buscar y realizar tests

Permite a los usuarios buscar y realizar pruebas relacionadas con las asignaturas que tienen en su grado. Los scripts asociados facilitan la búsqueda y presentación de pruebas disponibles, así como la interacción del usuario durante la realización de las mismas.

- **menu_tests.php:**

Este es el menú de tests que aparece tras haber realizado el login. En esta vista, aparecerán todos los tests disponibles en Flexam de la Universidad y grado del usuario que ha iniciado sesión.



Captura de pantalla de menu_tests.php

- **realize_test.php:**

Este es el formato de la vista que tendrá nuestra página de realización de los tests.



Captura de pantalla de realize_test.php

2.5. Estadísticas

Esta funcionalidad proporciona a los usuarios información estadística relevante sobre su actividad en la plataforma, como resultados de pruebas anteriores o progreso en ciertos temas, además de los resultados obtenidos en el nuevo test realizado.. Los scripts correspondientes se encargan de recopilar, procesar y presentar esta información de manera clara y comprensible para el usuario.

- **resultados_test.php:**

Muestra las estadísticas del test realizado. El número de preguntas, de aciertos, de fallos, de preguntas no respondidas y la nota sobre 10.

Además se muestran las estadísticas de todos los intentos que se ha realizado el test y un gráfico de la evolución de las notas en el tiempo.



Captura de pantalla de resultados_tests.php

2.6. Diagrama de scripts de vistas

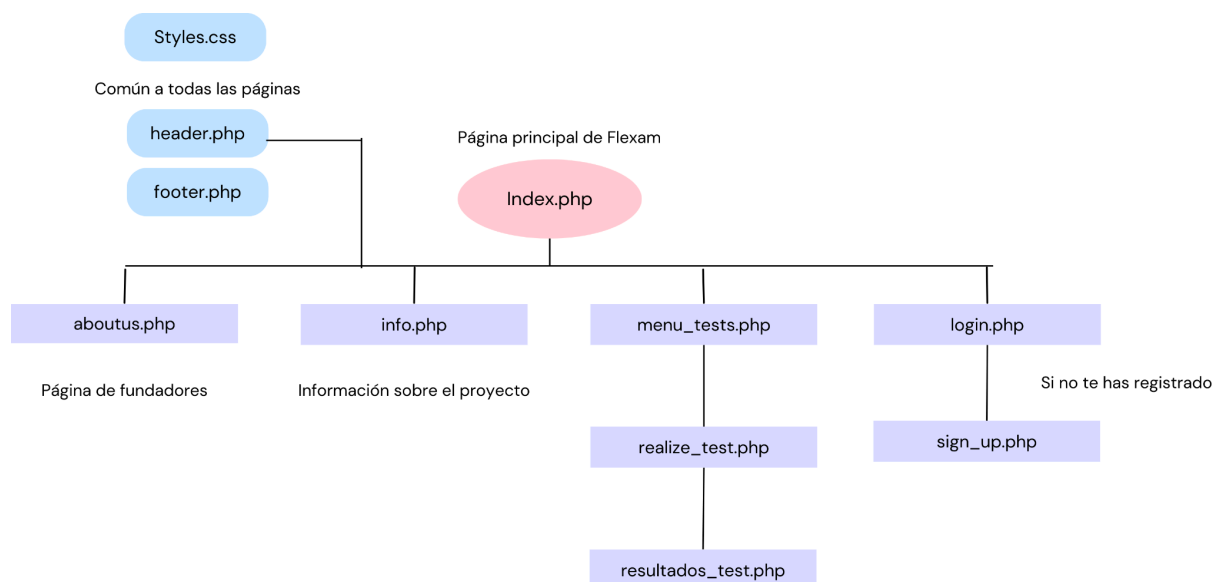


Imagen de la ruta de datos de la aplicación

3. Listado de scripts adicionales (**MODIFICAR**):

De cara a la entrega final faltaría ajustar esta parte y descartar este listado de scripts adicionales relacionándolo todo con el modelo en 3 capas que hemos planteado en la introducción del documento

- **sign_up_logic.php:**

Aquí se procesa el formulario de registro de usuario. Si todos los criterios se cumplen, registra al usuario en la base de datos.

- **login_logic.php:**

Este script PHP define una función authenticate para autenticar usuarios. Inicia asegurándose de que una sesión está activa y luego intenta verificar las credenciales del usuario contra registros en la base de datos. Utiliza consultas preparadas para buscar un usuario por su nombre, y si encuentra una coincidencia, verifica que la contraseña proporcionada coincida con la versión hasheada almacenada en la base de datos. Si la autenticación es exitosa, establece variables de sesión y redirige al usuario a una página de inicio. Si las credenciales no son válidas, prepara un mensaje de error.

- **logout.php:**

Este script finaliza la sesión del usuario y lo redirige a la página de inicio.

- **get_test.php:**

Este script verifica si el usuario ha iniciado sesión y, de ser así, obtiene los tests disponibles para dicho usuario. Ejecuta una consulta SQL que selecciona el ID del test, su título, el nombre de la asignatura asociada y el número de intentos realizados por el usuario en cada test.

- **procesar_test.php:**

Aquí se procesa la información resultante del envío del test guardando esta en la base de datos, concretamente en la tabla [respuesta_usuario](#).

- **obtener_datos_intentos.php:**

Este script recupera los datos de los intentos realizados por un usuario de un test específico.

- **signupHandler.php:**

Este script recibe el ID de una universidad a través del formulario de registro, ejecuta una consulta en la base de datos para recuperar todos los grados asociados a esa universidad.

- **config.php:**

Común para todos los scripts con acceso a la base de datos, este script está configurado para establecer la conexión a la base de datos "flexam".

4. Estructura de la BBDD

La estructura de la base de datos de Flexam se organiza en tres grandes bloques, cada uno diseñado para cumplir con diferentes requerimientos funcionales del sistema. A continuación, se detalla la organización actualizada de cada bloque con base en los últimos cambios realizados:

4.1. Usuarios y Relaciones Académicas:

La tabla *usuarios* registra información vital de los usuarios, incluyendo *ID_usuario*, *user*, *psw* (contraseña), *nombre*, *apellidos*, *email* (único por usuario), *ID_universidad*, *ID_grado*, y *rol*, siendo *ID_usuario* la clave primaria. Esta tabla se relaciona con las tablas de *universidades* y *grados* a través de *ID_universidad* y *ID_grado* respectivamente, facilitando la asociación de usuarios con sus respectivas instituciones educativas y programas académicos permitiendo así ofrecer los tests desarrollados específicamente por y para estos estudiantes.

Las tablas *universidades*, *grados*, y *asignaturas* forman la base de la estructura académica. *universidades* contiene *ID_universidad*, *nombre*, *abreviatura*, y *ciudad*, con *ID_universidad* como clave primaria. *grados* enlaza cada programa académico con una universidad mediante *ID_grado*, *ID_universidad*, y *nombre*, donde *ID_grado* es la clave primaria. *asignaturas* lista las materias ofrecidas, vinculadas a universidades a través de *ID_asignatura*, *ID_universidad*, *nombre*, y *abreviatura*, siendo *ID_asignatura* la clave primaria. La tabla *grado_asignatura* establece la relación entre los grados y las asignaturas que ofrecen.

Estas relaciones nos permiten ofrecer tests a alumnos de las mismas universidades que, aunque pertenezcan a grados diferentes, cursen la misma asignatura, algo que sucede muy a menudo.

4.2. Evaluaciones y Preguntas:

La tabla *tests* almacena los tests o evaluaciones creados por los usuarios, con campos para *ID_test* (clave primaria), *titulo*, *ID_usuario* (vinculación con la tabla de usuarios), *num_preguntas*, *es_publico*, y *es_anonimo*. Se introduce la tabla *test_asignatura* con *ID_test*, *ID_universidad*, *ID_asignatura* como claves primarias para relacionar tests específicos con

asignaturas y universidades, permitiendo su uso compartido entre programas académicos similares.

Las tablas *preguntas* y *opciones* facilitan la creación de preguntas de opción múltiple para los tests, donde *preguntas* incluye *ID_pregunta* (clave primaria), *ID_test*, y *pregunta*, y *opciones* especifica las posibles respuestas con *ID_test*, *ID_pregunta*, *ID_opcion* (todas claves primarias), *opcion*, y *correcta* (indicador de la respuesta correcta).

De esta manera, podemos tener preguntas sin un número definido de opciones permitiendo así todo tipo de tests (verdadero falso, una sola respuesta, multi-respuesta...).

4.3. Interacción Usuario-Test:

Finalmente, la tabla *respuesta_usuario* captura las interacciones de los usuarios con los tests, documentando cada intento con *ID_test*, *ID_usuario*, *ID_intento* (claves primarias), *nota*, *aciertos*, *fecha*, y *restriccion*, lo que permite el seguimiento del rendimiento del usuario mediante multitud de estadísticas en los intentos de los tests. Esta tabla y la información que contiene está pensada para el desarrollo de las nuevas funcionalidades en futuras entregas.

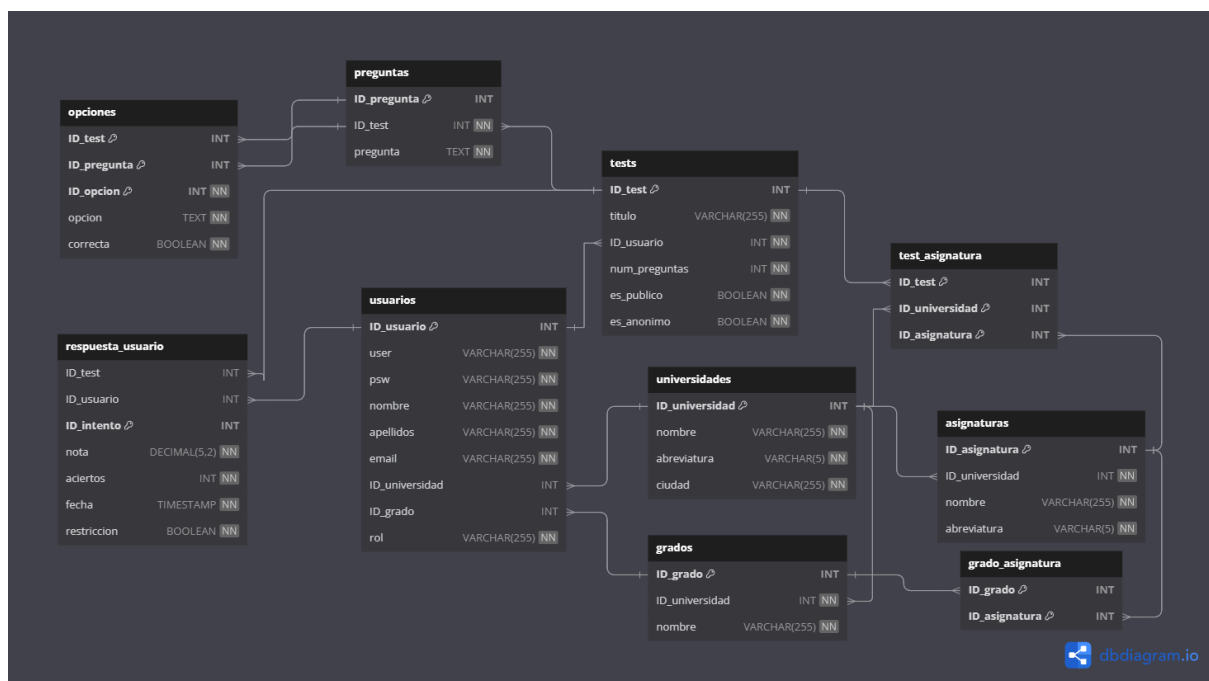


Diagrama de las relaciones entre tablas creado con dbdiagram.io

5. Prototipo funcional del proyecto

A continuación, se describe el funcionamiento general de nuestro prototipo de FLEXAM. Este prototipo representa una versión inicial de la aplicación web, que abarca las funcionalidades básicas y proporciona una idea clara de la experiencia del usuario:

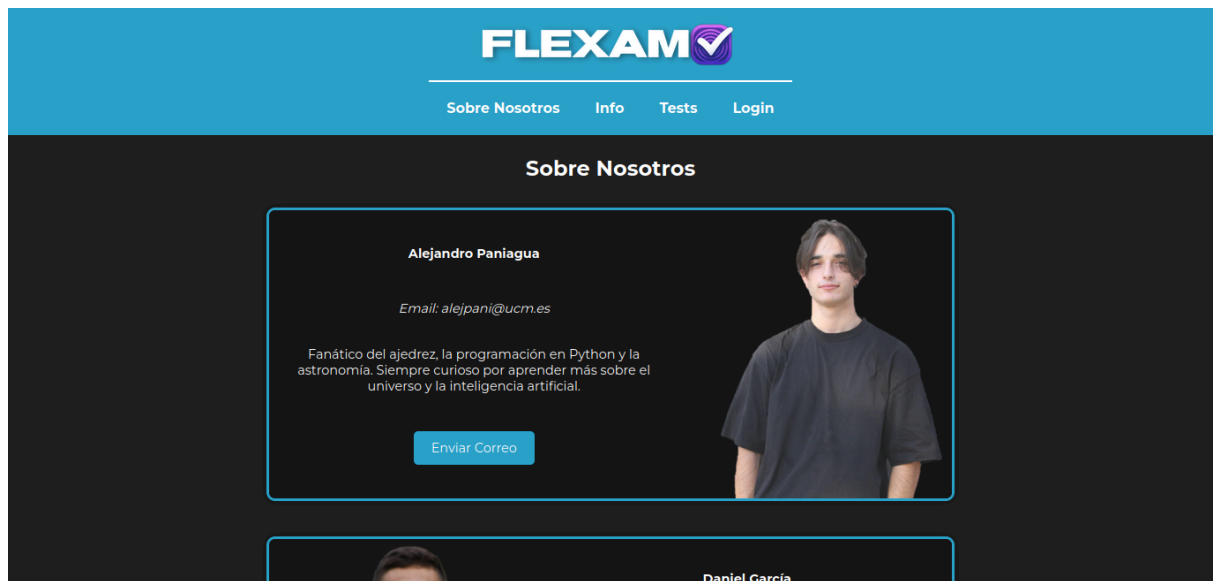
5.1. Página principal e información

Nada más buscar la página de Flexam, los usuarios llegan a la página principal, la puerta de entrada de la aplicación. En ella se explica un poco para qué sirve y cuál es nuestra función principal. Tanto en el header como en el index.php hay enlaces a todas las funcionalidades de nuestra página.



Página principal de Flexam

La página de [Sobre Nosotros](#) es otra extensión de la página principal. En ella se detalla en unas tarjetas los miembros fundadores de Flexam. También se detalla información de cada uno y un enlace de contacto al correo electrónico de cada miembro.



Página de about us

Por último, referente también a la página principal, la página de información donde simplemente se explica a más detalle para que sirve o cuál es la funcionalidad de Flexam.



Página que enseña más información de Flexam

5.2. Login o registro

Si continuamos explorando Flexam, encontramos un enlace a la página de login. Esta tiene un formulario para usuarios registrados donde puedes loguearte con tu usuario y contraseña. En cambio, si eres un nuevo usuario, también existe un enlace a la página [Sign up](#) donde podrás registrarte sin problema a nuestra aplicación.

Regístrate ahora.'"/>

Página de login de la aplicación

La página para registrarte en nuestra app goza de un registro donde deberás escribir ciertos datos personales que ayudarán a nuestra web a proporcionarte tests de asignaturas referentes a tu carrera. El formulario de registro se compone de preguntas personales y académicas donde deberás seleccionar tanto tu universidad como el grado que estás cursando.

Página de registro de Flexam

Tanto la página de registro como la página de login te devolverán a la página principal donde ahora, en el header, en lugar del enlace a login aparecerá un mensaje de bienvenida y un enlace que deberás pulsar en el caso de que quieras cerrar tu sesión de usuario.

5.3. Tests

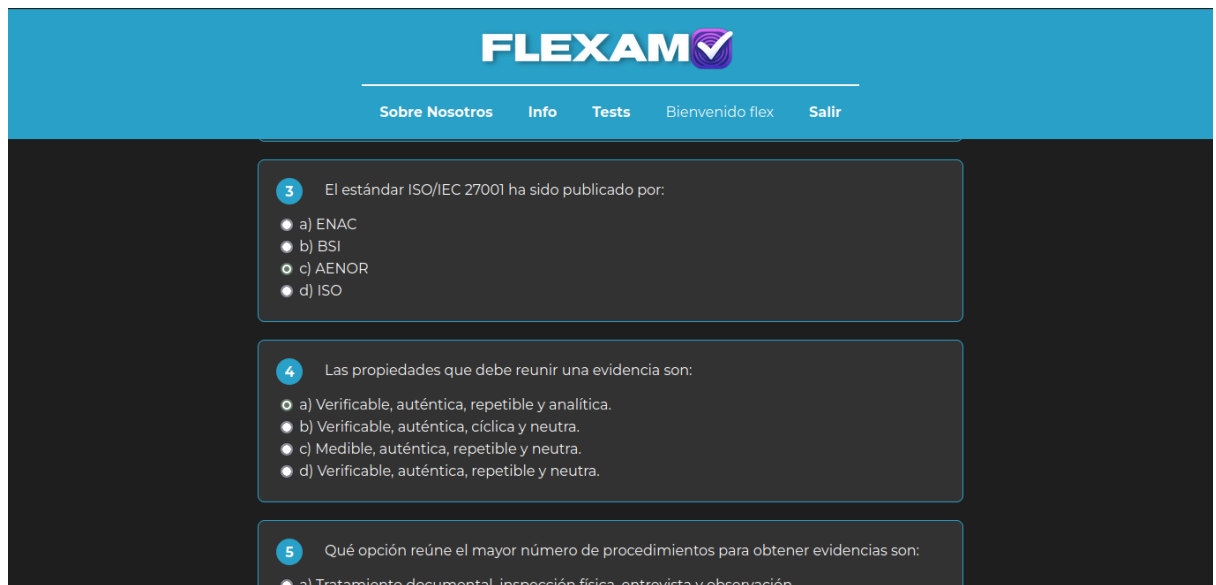
En la página principal y en el header hay un enlace [Tests](#) que te llevará a la página donde reside la funcionalidad principal de Flexam. Es la página de búsqueda de tests. Para esta entrega, la página te mostrará todos los tests disponibles para tu grado específico. En futuras entregas la página contará con un dispositivo de búsqueda que te permitirá filtrar los tests en función de la universidad y el grado que elijas.

Podemos ver diferentes tarjetas, cada una, es un test distinto donde se muestra su nombre, así como la asignatura a la que pertenece y el número de intentos que has realizado.



Página de selección de tests disponibles

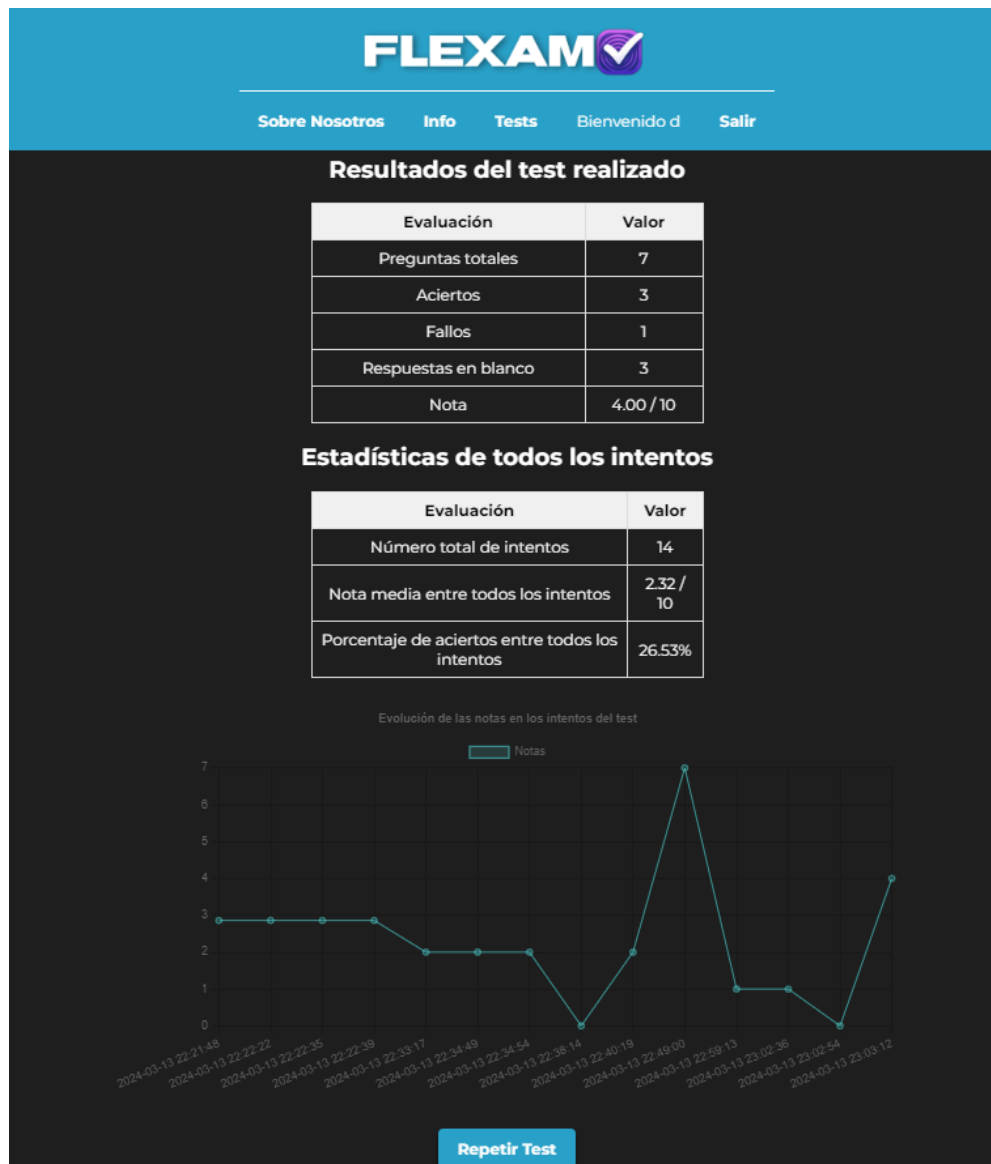
Si seleccionamos un test, este nos llevará a la página donde se realiza el test. Para esta entrega, esta página es un formulario sencillo compuesto por las preguntas test y sus diferentes opciones. Una vez contestadas todas las preguntas se puede enviar el test para que se compruebe el resultado.



Página donde se resuelve un test

5.4. Estadísticas

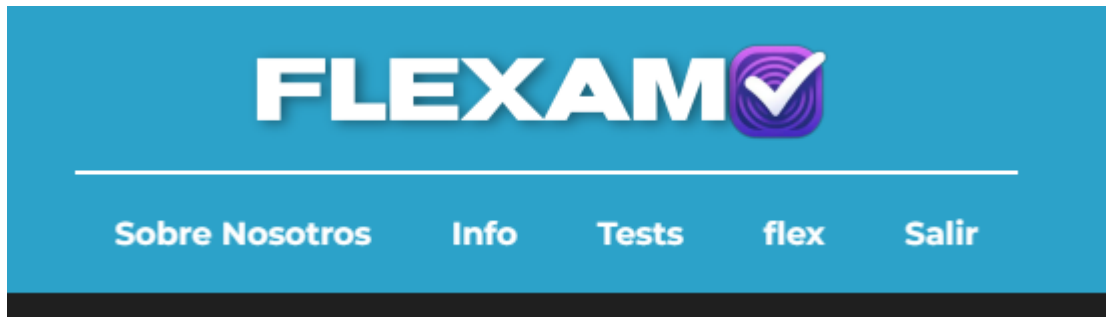
Una vez que se realiza un test y se envía, la página te enviará automáticamente a la página de resultados y estadísticas. En un futuro, a esta se podrá acceder en cualquier momento para ver las estadísticas de tus tests.



Página de resultados y estadísticas de tus tests (*resultados_test.php*)

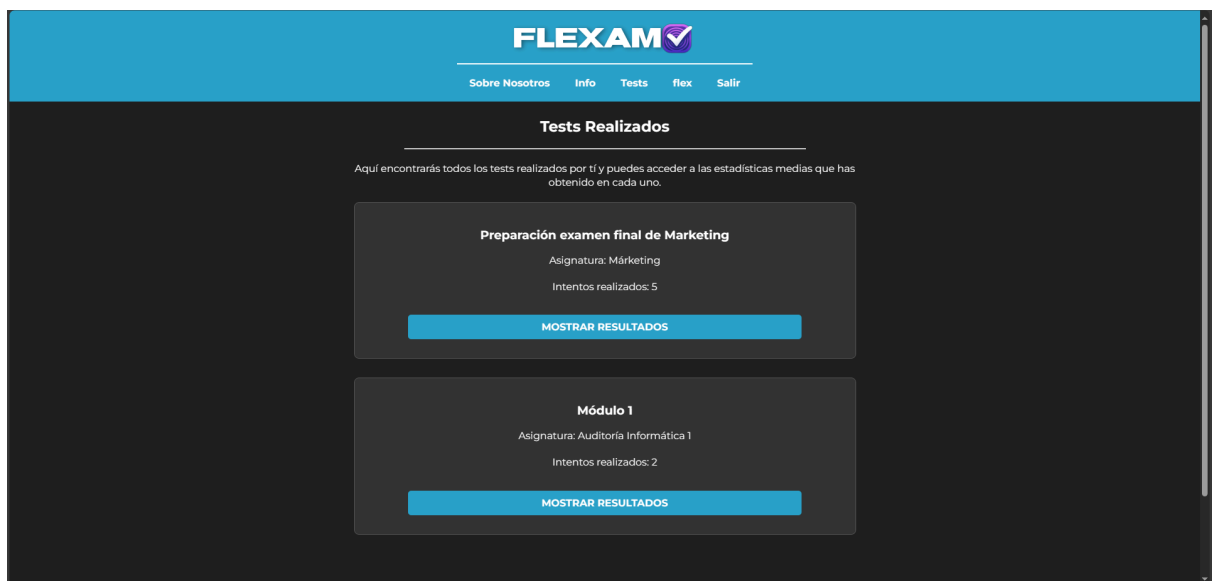
5.5. Menú de usuarios (Nueva implementación)

Hemos añadido una nueva funcionalidad relacionada con los usuarios. Ahora aparecerá el nombre del usuario en cuestión en la cabecera, y haciendo click nos aparecerán sus tests realizados hasta ese momento



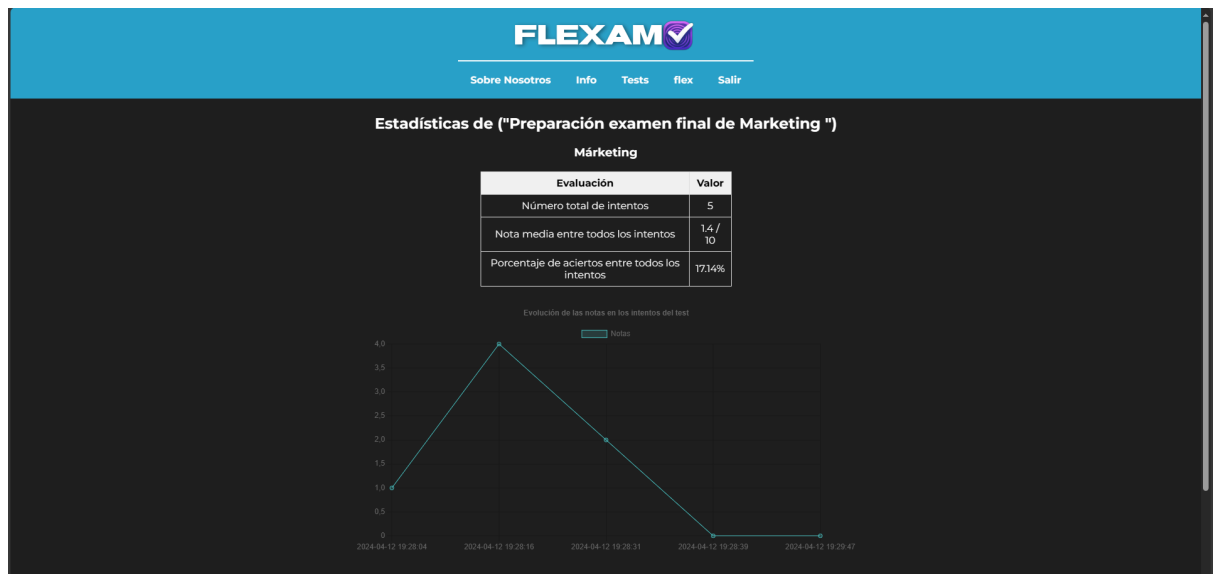
Nueva cabecera de la aplicación

Este nuevo botón te dará acceso al nuevo menú de usuario implementado.



user_menu.php

En este nuevo menú podrás ver todos los tests que hayas hecho alguna vez y el número de intentos realizados en cada uno. Si presionas el botón de “Mostrar resultados” accederás a la página de los resultados globales de ese test.



resultados_totales.php

Esta nueva página te permite ver los resultados globales de un test sin tener que hacerlo nuevamente como ocurría con el prototipo anterior.

6. Datos de acceso de los usuarios y sección VPS

Usuario para pruebas (insertado con el .sql):

user: **flex**

password: **flexam**

Funcionalidad añadida con 3ª entrega:

- Url que da acceso a la aplicación:
<https://vm024.containers.fdi.ucm.es/>
- Url github:
<https://github.com/Alex7pl/PracticasAW.git>

7. Actividades

Nombre	Actividad realizada
Martin Veselinov	Refactorización de la estructura funcional del proyecto, modulando el modelo de 3 capas junto con patrón Singleton

Alejandro Paniagua	Implementación de la página y menú de usuarios y la lógica de esta
Daniel García	Lógica referente al nuevo servidor, implementación de este en la aplicación web
Alejandro de Mateo	Implementación del patrón Singleton a la aplicación