UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCOLA POLITÈCNICA SUPERIOR DE GANDIA

DESARROLLO DE UNA APLICACIÓN WEB RESPONSIVE HTML5 PARA FORMACIÓN DE EQUIPOS DE TRABAJO EFICIENTES MEDIANTE ALGORITMO ADNe

TRABAJO FIN DE GRADO

GRADO EN SISTEMAS DE TELECOMUNICACIÓN, SONIDO E IMAGEN

AUTOR: Guillem Martí Bellmunt

TUTOR: José Marín-Roig Ramón

2018-2019

Resumen

El algoritmo ADN emocional (ADNe) es el algoritmo desarrollado por Elías Azulay basado en el análisis transaccional. Teniendo como base los trabajos de prestigiosos psiquiatras, científicos y genetistas, este algoritmo ha ayudado a muchas empresas a reclutar a los trabajadores más idóneos ya que se basa en calcular la máxima afinidad entre el comportamiento emocional del candidato, la empresa y las funciones a realizar en ella.

En este proyecto, se pretende usar las virtudes del algoritmo ADNe para realizar el diseño y la implementación de una aplicación web adaptativa a dispositivos móviles, siendo su principal objetivo formar grupos de trabajo óptimos entre los alumnos de cualquier asignatura, dando prioridad a la afinidad de todos los integrantes de los grupos y la usabilidad de la aplicación web.

Palabras clave: aplicación web, web adaptativa, algoritmo de ADNe.

Abstract

The emotional DNA algorithm (eDNA) is the algorithm developed by Elías Azulay based on transactional analysis. Based on the work of prestigious psychiatrists, scientists and geneticists, this algorithm has helped many companies to recruit the most suitable workers since it is based on calculating the maximum affinity between the emotional behavior of the candidate, the company and the functions to perform in.

In this project, we intend to use the virtues of the eDNA algorithm to design and implement an adaptive web application for mobile devices, with the main objective of forming optimal working groups among the students of any subject, giving priority to the affinity of the whole members of the groups and the usability of the web application.

Keywords: web application, web responsive, eDNA algorithm.

Contenido

Tabla de ilustraciones

1. Introducción

La siguiente memoria describe el trabajo final de grado de sistemas de telecomunicación, sonido e imagen, en la escuela politécnica superior de Gandia de la *Universitat Politècnica de València*.

A continuación, se mostrará la evolución que ha tomado el proyecto a lo largo de su desarrollo, desde el planteamiento inicial y análisis de requisitos, hasta su implementación y diseño del *front-end* y *back-end*, es decir, de la capa de presentación y de la capa de datos de la aplicación web, respectivamente.

1.1. Contexto

Este proyecto nace de la necesidad de satisfacer un problema que se repite tanto en empresas como en la docencia. En dichos entornos, la mayoría de los proyectos o trabajos realizados son en grupos de personas donde, en muchos casos, los resultados obtenidos no son los esperados. Esto puede darse por diversos motivos, pero la mayoría de las veces es porque los integrantes del grupo no son afines entre sí.

Para solventar este problema, se ha ideado el desarrollo de esta aplicación web, donde se pretende que se convierta en una herramienta que ayude al profesorado de la Universidad Politécnica de Valencia a formar grupos de trabajo óptimos. De este modo, todos los grupos serán integrados de alumnos con diferentes capacidades y habilidades que harán que el grupo trascienda a un nuevo nivel de rendimiento.

Esta aplicación web no podría llegar a realizarse de no ser por el trabajo de la empresa **Jacobson, Steinberg & Goldman**, ya que se basa en el algoritmo *ADNe*, de Elías Azulay.

1.2. Objetivos del proyecto

El objetivo principal de este proyecto es la creación y desarrollo de una aplicación web adaptativa para formar grupos de trabajo óptimos. La aplicación tiene que ser fácil e intuitiva, por lo que tiene que cumplir con una experiencia de usuario satisfactoria. Además, la aplicación está destinada a ser una herramienta adicional de *PoliformaT*, por lo que sólo sería capaz de usarla el profesorado que inicie sesión con sus credenciales.

Para llevar a cabo con éxito el desarrollo de esta aplicación, se deberán de cumplir los siguientes objetivos específicos:

- Realizar un diseño de la interfaz de usuario que se adapte al portal de *PoliformaT*.
- La aplicación web deberá ser adaptativa a dispositivos móviles, ya sean smartphones o tabletas.
- Crear y editar tablas donde se muestren todos los alumnos de una asignatura específica para, posteriormente, agruparlos en grupos mediante el algoritmo *ADNe*.
- La aplicación será capaz de añadir nuevas restricciones a la hora de formar grupos de trabajo.

1.3. Estructura del documento

La memoria de este proyecto se estructura en el orden cronológico en el que se han realizado cada una de las diferentes fases o etapas del desarrollo de la aplicación web. A continuación, se muestran las diferentes etapas que ha atravesado el trabajo:

- **Algoritmo *ADNe***: en esta fase se va a introducir el algoritmo *ADNe* y cuáles son sus raíces y sobre qué bases científicas se basa. Introduciremos el trabajo de Elías Azulay.
- **Especificación de requisitos**: en esta fase se reúnen los atributos, todos los requisitos funcionales y las restricciones que debe de cumplir la aplicación web final.
- **Descripción de las herramientas de diseño e implementación**: en esta fase se describen los diferentes niveles que componen la arquitectura de la aplicación web, así como las diferentes tecnologías y herramientas utilizadas. Se describe detalladamente la implementación y el código fuente.
- **Análisis**: el análisis describe la estructura y la funcionalidad de la aplicación web mediante diagramas que permiten comprender el sistema. Se incluyen diagramas de casos de uso y de actividades, que describen los principales comportamientos de la web, así como diferentes dibujos conceptuales o bocetos.
- **Diseño gráfico y usabilidad**: aquí se expondrá el diseño del apartado gráfico, de la interfaz de usuario y de la propia usabilidad de la aplicación. Se incluyen capturas de pantalla y conceptos a tener en cuenta acerca de la usabilidad.
- **Conclusiones**: en este apartado se habla de las conclusiones finales obtenidas, de los objetivos cumplidos y de las futuras implementaciones a este trabajo.
- **Bibliografía**: en esta fase se detalla la bibliografía consultada a lo largo del trabajo.
- **Anexos**: por último, se incluyen el código fuente y varios documentos en un documento aparte a este mismo.

2. Algoritmo ADNe

2.1. ¿Qué es ADNe?

El algoritmo ADNe, o ADN emocional, es un algoritmo desarrollado por la compañía valenciana **Jacobson, Steinberg & Goldman**, donde su director ejecutivo es Elías Azulay. Este algoritmo consigue saber cómo es una persona, cómo va a reaccionar y predecir cómo va a ser su conducta, utilizando un método personalizado, preventivo y preciso.

De esta forma, se mide como por primera vez el comportamiento emocional a través de un patrón concreto que abarca más de 36.900 millones de combinaciones emocionales posibles. El algoritmo *ADNe* ha sido desarrollado por un equipo multidisciplinar que agrupa a biotecnólogos, informáticos, genetistas, biólogos moleculares y psicólogos.

2.2. ¿Cómo nació el algoritmo ADNe?

Todo comenzó con el trabajo de Eric Berne* (1910-1970), su nombre real es Eric Leonard Bernstein, psiquiatra fundador de la escuela *Psicológica humanista* y creador del *Análisis Transaccional*, dónde se postula acerca de los *3 estadios del yo*, o estadios *PAN* (Padre, Adulto y Niño).

Ilustración 1. Eric Berne.

La evolución de los estudios de Berne ha llevado a la definición de un patrón numérico del comportamiento emocional sobre los 7 registros básicos enumerados en el Análisis Transaccional*.

Dicho patrón dispone de los atributos requeridos para cotejar sus transacciones entre los 7 registros sobre el estado del yo, que es comparable a la liberación de los principales neurotransmisores que rigen el comportamiento humano, ya sea en su actividad sináptica o inhibidora (los inhibidores son las moléculas que contrarrestan a los neuromoduladores), tanto de forma individual como combinado.

Eric Berne. Wikipedia: https://es.wikipedia.org/wiki/Eric_Berne
Análisis Transaccional. Wikipedia: https://es.wikipedia.org/wiki/An%C3%A1lisis_transaccional

Todo ello nos ilustra sobre las distintas combinaciones al combinar los 7 registros hasta poder caracterizar el espectro emocional de las personas en base a la capacidad e intensidad de liberación de los principales neurotransmisores que rigen el comportamiento humano, teniendo como precursor la expresión génica, estipulada por Kandel.

Eric Kandel* (Premio Nobel 2000 en Medicina y Fisiología) creo la base de la biología molecular, siendo especial importante en el algoritmo *ADNe*. Kandel estableció una conexión directa entre los genes y la sinapsis, siendo la base del *Sistema Neuromodulador*. En este trabajo se postula que la biología molecular era capaz de almacenar memoria a corto y a largo plazo, por lo que nuestro ADN llega a condicionar la memoria y la conducta humana.

Ilustración 2. Eric Kandel.

En las sinapsis, las neuronas se comunican entre ellas mediante los neurotransmisores, que son moléculas encargadas de enviar señales desde una neurona a la siguiente. Otras partículas llamadas neuromoduladores también intervienen sobre la comunicación entre células nerviosas.

En el algoritmo *ADNe* toma como bases *el Sistema Neuromodulador* para entender las claves del comportamiento humano. Los estudios más recientes apuntan a que hay 11 sistemas de neurotransmisores en el ser humano: dopamina, serotonina, endorfina, etc. Estos neurotransmisores se combinan de diferente forma ante un estímulo externo. Cada persona segrega diferentes neurotransmisores combinados de diferentes formas ante dicho mismo impulso. Es por eso, que cada persona reacciona de forma diferente ante un mismo estímulo. Por poner un ejemplo, en el mundo de la ciencia se ha establecido que las personas protectoras tienden a tener niveles altos de oxitocina, todo lo contrario, a las personas que no lo son.

Eric Kandel. Wikipedia: https://es.wikipedia.org/wiki/Eric_Kandel

2.3. Bases

El algoritmo *ADNe* ha conseguido desarrollar un patrón numérico que descifra los umbrales segmentados de los 7 registros del *persotipo* para después combinarlos y obtener el resultado de las personas. Estos siete registros son:

Registro ADNe	Estadio A.T. (E. Berne)	Descripción ADNe
Ob	Padre crítico	Reflexión y objetividad. Visión periférica social.
Pr	Padre protector	Generosidad, disponibilidad y protección afectiva.
Ad	Adulto	Capacidad de análisis y de síntesis de datos.
Nt	Niño natural	Espontaneidad, imaginación y creatividad.
Sm	Niño sumiso	Espectro de atención y aprendizaje. Memoria.
Rb	Niño rebelde	Agresividad, dinamismo y tendencia posesiva.
Mn	Niño pequeño profesor	Influencia, anticipación y astucia.

Ilustración 3. Tabla de los 7 registros del algoritmo ADNe

La combinación de estos 7 estadios o registros da como resultado el *Persotipo*. El *Persotipo* es el resultado del código emocional de cada persona. Lo acuñó el científico genetista Alberto Acedo, doctor en Biología molecular y fundador de la empresa biotecnológica Biome Makers Inc.

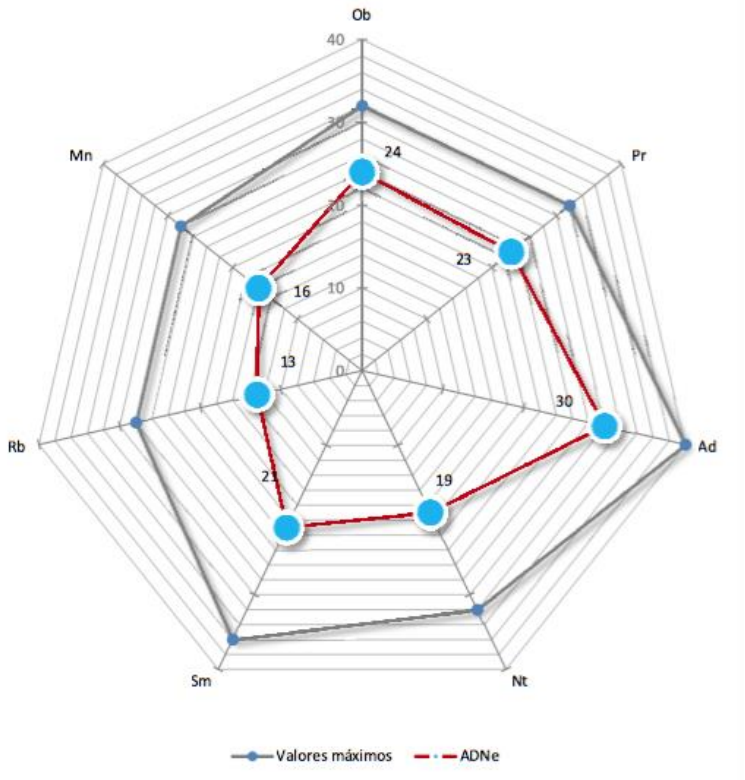

Ilustración 4. Resultado de un persotipo mediante el uso del algoritmo ADNe

Como hemos mencionado anteriormente, es casi imposible que los *persotipos* se repitan, ya que hay 36.900 millones de combinaciones posibles, por lo que podemos sacar un resultado bastante coherente con cada persona.

2.4. El test ADNe

El test Azulay-Bernstein, o más bien conocido como test psicogénico de *ADNe*, está troceado con cada uno de los once neurotransmisores que segrega el cuerpo humano, por lo que podemos abarcar todo el amplio espectro del sistema neuromodulador.

El test cuenta con 59 preguntas, donde cada pregunta abarca a uno o más nuerotransmisores y cuenta con diferentes niveles de ponderación. Posee tanto preguntas activas, como preguntas reactivas y cada una de ellas pondera de forma diferente. También cuenta con dos preguntas de control. El alfa de Cronbach (el indicador de consistencia del test) es muy elevado, teniendo un valor superior al 0,6.

La finalidad de utilizar el algoritmo *ADNe* no es otra que buscar la complementación de dos o más *persotipos* entre sí. De esta forma, obtendremos un grupo totalmente óptimo, donde cada miembro destacara en una o varias facetas de los 7 registros del algoritmo *ADNe*.

Es importante recalcar que las personas pueden ser inhibidores de otras personas que tengan un registro demasiado elevado y pueda desequilibrar los equipos. Por poner un ejemplo, si en un grupo dado hay una persona calificada como manipuladora, el algoritmo podría añadir en el grupo una persona influyente pasiva que amedrente al manipulador.

3. Especificación de requisitos

En este apartado hablaremos sobre los requisitos necesarios que se han tenido en cuenta para que desarrollar la aplicación web.

Esta fase es crucial en el desarrollo de nuestro software ya que se realiza un estudio y captura los requisitos necesarios más necesarios que debe de poseer la aplicación.

Es por eso que se deben de establecer las necesidades para poder proceder a realizar un estudio de las tecnologías antes de realizar la aplicación web.

3.1. Atributos

Al tratarse de un *Trabajo Fin de Grado* ofertado públicamente, los requisitos fueron establecidos por el tutor. A partir de reuniones iniciales, se establecieron las características principales y las pautas que se deben de tomar en este *TFG*.

Esta aplicación web va a ser dedicada exclusivamente para los profesores de la *Universitat Politècnica de València*, por lo que será el profesorado de la UPV quienes sean los **usuarios** principales de nuestra aplicación. Aun así, está aplicación no restringe su uso a fines académicos, también se podría aplicar en empresas para que los encargados, jefes o directivos formen grupos de trabajo con sus trabajadores en nómina.

Las pautas a seguir son las siguientes:

Los profesores se descargan la lista de candidatos desde *PoliformaT*. Esta lista se insertará en la aplicación web a desarrollar y se almacenará en una base de datos en la parte del servidor de la aplicación.

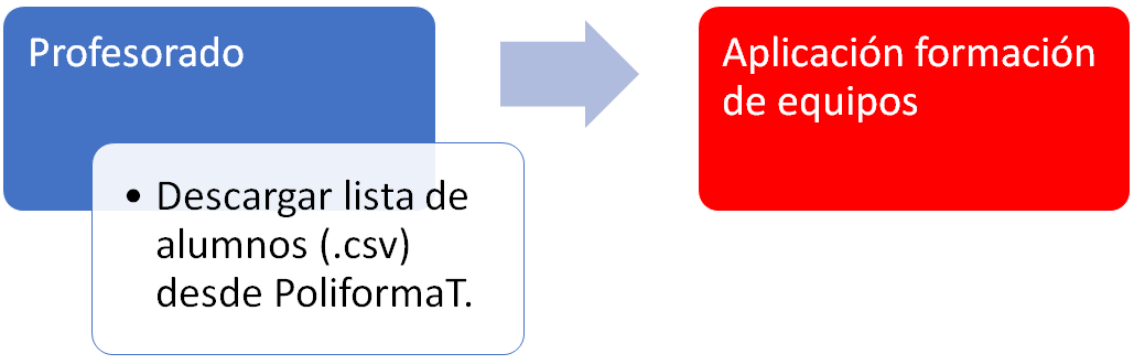

Ilustración 5. Diagrama del proceso que realiza el profesorado de la UPV.

Nuestra aplicación web deberá de alojarse en un servidor web de la *UPV* o, en su defecto, en el servidor de algún colaborador asociado a la *UPV*.

La empresa de Elías Azulay colaborará directamente en este TFG con el desarrollo de una API que interconecte nuestra aplicación con el algoritmo *ADNe*. La API estará alojada en los servidores de la empresa **Jacobson, Steinberg & Goldman** y tendrá unas variables de entrada y unas de salida. El proceso interno de dicha API es totalmente ajeno al desarrollo de nuestra aplicación web.

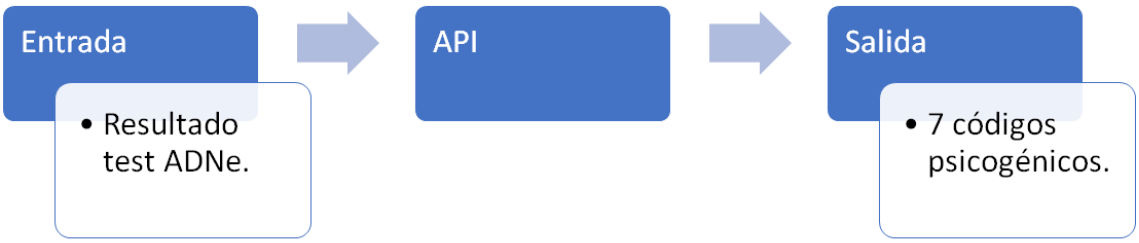

Ilustración 6. Variables de entrada y salida en la API de Elías Azulay.

Los alumnos nunca tendrán acceso a esta aplicación. Ellos recibirán un correo electrónico con un enlace que les redirige a realizar *el test ADNe* en la API de **Jacobson, Steinberg & Goldman** y, una vez finalizado, ya no tienen ningún rol más en este software. Al alumnado se le premiará con un *informe personal ADNe* para que puedan adjuntarlo a su *Currículum Vitae*.

Una vez los alumnos hayan realizado *el test ADNe*, obtendremos los 7 registros psicogénicos de cada alumno recibidos mediante la API de **Jacobson, Steinberg & Goldman** y se almacenarán en la base de datos.

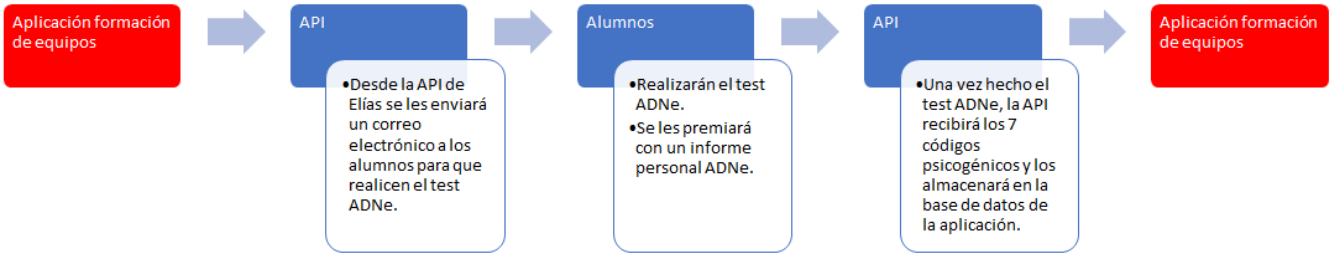

Ilustración 7. Diagrama completo que realiza la API con los resultados de los alumnos.

Con todos los datos almacenados, realizaremos la formación de grupos mediante el algoritmo *ADNe*. El profesor podrá aplicar un número determinado de restricciones, tales como agrupar por diferente género, mezclar bachiller con F.P., separar a los amigos, determinar el número de grupos, etc. Obviamente, estas restricciones afectarán al resultado del algoritmo *ADNe*, por lo que el resultado no será óptimo y se obtendrá un *KPI* (*Key Performance Indicator*) más bajo que si no hay ningún tipo de restricción.

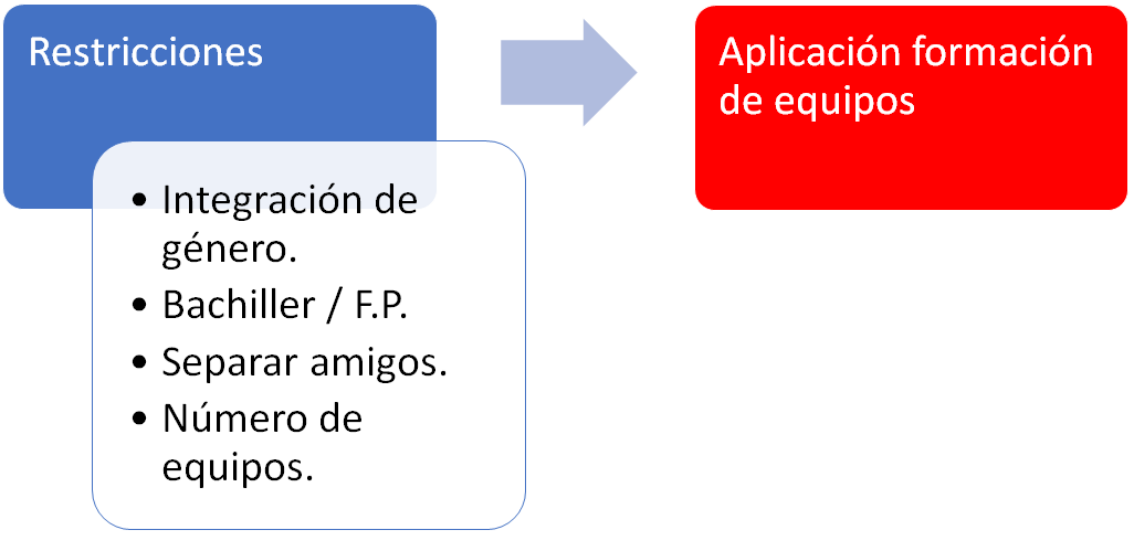

Ilustración 8. Diferentes tipos de restricciones aplicables en la aplicación web.

Una vez se hayan realizado las restricciones pertinentes, se podrá exportar la lista de los equipos formados en diferentes formatos, tales como una tabla de Excel como en formato PDF.

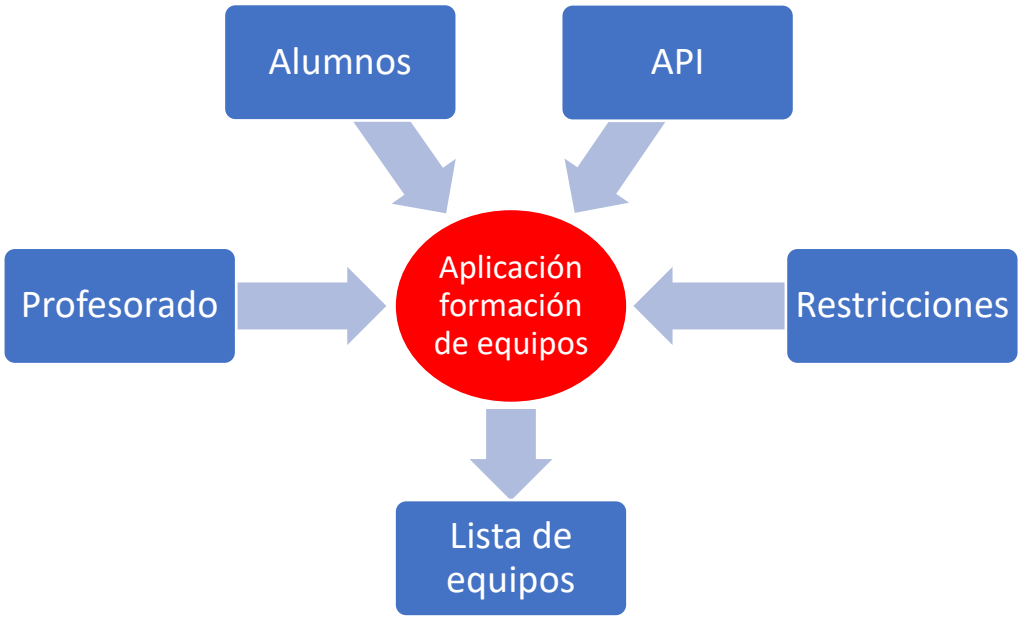

Ilustración 9. Diagrama general del proceso de la aplicación de formación de equipos.

Con estas características establecidas y ya conociendo el procedimiento que queremos que posea nuestra aplicación web, pasamos a detallar tanto los requisitos funcionales como las restricciones de diseño que deben de poseer para llevar a cabo el desarrollo de este software.

3.2. Requisitos funcionales

Los requisitos funcionales describen el comportamiento de la aplicación. Se incluyen las funciones desempeñadas por el sistema, descripciones de los flujos trabajo y cualquier requerimiento adicional. A continuación, se van a documentar las operaciones y actividades que debe de poder desempeñar la aplicación:

Registro de usuario

- **Entradas**: se rellena un formulario para registrar los nuevos usuarios al sistema, es decir, al profesorado de la UPV.
- **Comportamiento**: se debe de comprobar que los campos introducidos son correctos, que no haya ningún usuario repetido en la base de datos, que no se dejen campos en blanco.
- **Salidas**: si todo es correcto, el nuevo usuario se almacena en la base de datos. De no ser así, se notificará cuál ha sido el error para que el usuario vuelva a repetir el proceso.

Inicio de sesión

- **Entradas**: se rellena un formulario de inicio de sesión.
- **Comportamiento**: realizamos una consulta a la base de datos para comprobar si el usuario existe y si la contraseña es correcta.
- **Salidas**: si todo es correcto, el usuario iniciará una nueva sesión en la aplicación, si no es así, se notificará si el usuario no existe o si la contraseña es incorrecta.

Cerrar sesión

- **Entradas**: se pulsará un botón en la interfaz para cerrar la sesión.
- **Comportamiento**: al pulsar el botón, la sesión que este abierta será cerrada y ningún usuario no logueado será capaz de acceder a ninguna parte de nuestra aplicación sin tener una sesión iniciada. Para volver a acceder a los sitios web, se deberá de iniciar la sesión otra vez.
- **Salidas**: cierra la sesión.

Introducir alumnos

- **Entradas**: el profesor introduce en la aplicación web un archivo CSV con el listado de los alumnos de la asignatura. Dicho archivo se lo ha descargado desde *PoliformaT* con anterioridad.
- **Comportamiento**: la aplicación es capaz de *parsear* los alumnos y sus datos personales dentro del archivo para almacenarlos en un objeto de *Javascript*. Es decir, el archivo CSV es una cadena de caracteres gigante. Se deberá de crear una función que sea capaz de dividir y cortar dicha cadena para que se almacene en un objeto de carácter matriz, donde las filas sean los alumnos y las columnas los datos personales de cada alumno. Finalmente, se almacena dicha matriz en la base de datos.
- **Salidas**: si todo es correcto, saldrá una ventana emergente que todo ha salido correcto y redirigirá a la página siguiente. Si no es así, se notificará que hay algún tipo de error y se detallará la forma de proceder.

Editar alumno

- **Entradas**: el usuario pulsa un botón para editar cualquier los datos de cualquier alumno que se encuentre en la base de datos.
- **Comportamiento**: en la interfaz de usuario se mostrará todos los alumnos listados en una tabla y habrá un icono que se podrá cliquear que será para poder editar los datos de los alumnos. Contendrá un formulario que muestre los datos actuales dentro de los campos y puedan borrar y editar. Para ello, al hacer pulsar el botón al icono de cierto alumno, se apuntará a una dirección de la base de datos donde se almacene dicho alumno. En ese momento, se abrirá una nueva página (realmente será la misma) donde estén los datos editables del alumno selecciona. Una vez se pulse el botón *guardar cambios*, se deberán de cambiar en la base de datos donde se almacena el grupo actual.
- **Salidas**: se actualiza los nuevos cambios en la base de datos y se visualizan por pantalla. La página espera la confirmación definitiva para pasar a la siguiente página.

Borrar alumno

- **Entradas**: el usuario pulsa un botón para borrar cualquier alumno que se encuentre en la base de datos.
- **Comportamiento**: en la interfaz de usuario se mostrarán todos los alumnos listados en una tabla y habrá un icono que se podrá cliquear que será para poder borrar cualquier alumno que se encuentre en la base de datos. Al pulsar el botón, aparecerá una ventana de confirmación para evitar eliminaciones no deseadas. Al pulsar el botón sobre cierto alumno, se apuntará a una dirección de la base de datos donde se almacena dicho alumno.
- **Salidas**: el alumno es eliminado de la base de datos y no se mostrará por pantalla. La página espera la confirmación definitiva para pasar a la siguiente página.

Añadir un alumno

- **Entradas**: el usuario rellena un formulario con los datos de un nuevo alumno y pulsa un botón para añadir dicho alumno nuevo a la base de datos.
- **Comportamiento**: debajo de la tabla donde se muestran los alumnos, habrá unos campos de entrada de texto para añadir un nuevo alumno y un botón para guardar el nuevo alumno en la base de datos. Se comprobará que la entrada de datos es correcta.
- **Salidas**: El nuevo usuario añadido se guarda en la base de datos y se muestra por pantalla. La página espera la confirmación definitiva para pasar a la siguiente página.

Enviar correo

- **Entradas**: el usuario pulsa un botón para enviar un correo genérico a los alumnos.
- **Comportamiento**: una vez se tiene confirmado todos los alumnos que tienen que ser agrupados, se debe de enviar un correo electrónico a todos los alumnos que se encuentren en la base de datos, explicando que es para formar grupos de trabajo y que deben de realizar el *test ADNe*. Al pulsar el botón, se abrirá la aplicación por defecto de correo electrónico y estará cumplimentado los destinatarios y el cuerpo del mensaje, donde se alojará el enlace para realizar el test, que se realizará en los servidores de **Jacobson, Steinberg & Goldman**, o vía API.

- **Salidas**: se envían los correos electrónicos a los correos de la universidad de los alumnos para que realicen el *test ADNe*. La página redireccionará a la página del cuadro de mandos.

Formar equipos

- **Entradas**: el usuario introducirá el número total de grupos y podrá aplicar algún tipo de restricción a la hora de agrupar los alumnos.
- **Comportamiento**: una vez todos los alumnos hayan realizado el test, el profesor deberá agrupar los alumnos mediante el algoritmo *ADNe*. Habrá un único campo de entrada donde determinará el número total de grupos. Aquí se le asignará toda la lógica correspondiente para realizar la asignación de grupos. Debe de tener un algoritmo que sea capaz de mostrar al profesorado si los grupos son pares o impares. En el caso de que sean impares, deberá formar grupos irregulares y se advertirá por pantalla de que va a haber un grupo o más que contendrá menos alumnos. Una vez hecho esto, le profesor podrá aplicar restricciones a la hora de formar los grupos, siendo las restricciones tales como mezclar sexos, procedencia académica (*Bachillerato* con *Formación Profesional*), etc. Finalmente se aplica el algoritmo *ADNe* para agrupar los grupos de trabajo.
- **Salidas**: se muestra por pantalla la agrupación de los grupos con todos sus integrantes y un porcentaje KPI de cada grupo. Se contempla que se pueda generar un documento PDF.

Con la formación de equipos, finaliza el comportamiento de la aplicación web.

3.3. Requisitos no funcionales

Los requisitos no funcionales representan las características generales o restricciones de la aplicación. Son los que especifican criterios para evaluar la operación de un servicio de tecnología de la información, en contraste con los requisitos funcionales. Son las siguientes:

Hosting: la aplicación web debe de estar alojada a un servidor para poder utilizarse vía internet.

Diseño adaptable: el software debe de estar diseñado tanto para computadoras como para dispositivos móviles, por lo que la interfaz de usuario debe de adaptarse al dispositivo que se está utilizando al usar nuestra aplicación.

Usabilidad: la interfaz de usuario de nuestra aplicación web debe de ser una interfaz cómoda, fácil e intuitiva para los usuarios. La facilidad de uso que tenga nuestro software tiene un impacto directo a la calidad de la experiencia del usuario, por lo que debe de ser siempre positiva.

Eficiencia: la aplicación web debe de ser capaz de cumplir su función de forma correcta y sin ningún tipo de error.

Escalabilidad: la aplicación debe de tener la capacidad de poder añadirle nuevas funcionalidades que crea el profesorado, es decir, los usuarios, que sean necesarias. El servidor debe de poder aumentar a medida que aumenta la base de datos.

Accesibilidad: la aplicación web debe de seguir los patrones de accesibilidad según la *World Wide Web Consortium* (*W3C*).

Seguridad: el software debe de disponer de un sistema de autenticación, ya que se van a tratar datos sensibles del alumnado. Es por eso que todas las conexiones a la base de datos deben de ser con una conexión segura.

Integración: la aplicación debe de poder usarse en todos los navegadores web actuales o, en su defecto, en los más usados (Google Chrome, Mozilla Firefox y Safari).

Robustez de datos: la aplicación debe de garantizar que hay una entrada correcta de datos y no se duplican ni alumnos ni usuarios. Se debe de respetar la Ley de protección de datos.

4. Descripción de las herramientas de diseño e implementación

4.1. Arquitectura

Nuestra aplicación está basada en el modelo Cliente/Servidor. Los clientes realizan peticiones al Servidor Web que les ofrece una respuesta, de este modo la capacidad de proceso está repartida entre los clientes y los servidores. Mediante esta arquitectura, los accesos, recursos y la base de datos están bajo el amparo del Servidor Web. De este modo, ningún usuario puede acceder a los datos si no están autorizados.

4.2. Tecnologías utilizadas

Para realizar este proyecto se han utilizado varias tecnologías y lenguajes de programación, ya sea para diseñar el aspecto visual, como la lógica y el comportamiento de la aplicación web.

Para el aspecto visual de la aplicación se ha utilizado tanto HTML5 como CSS. El primero está más centrado en la estructura y el segundo más en el estilo. Ambos siguen el estándar de la W3C, por lo que nos aseguramos de que funcionen en todos los navegadores web. De todas formas, he utilizado la librería de Bootstrap para facilitarme el diseño y estilo de la aplicación web. Además, utiliza el sistema de rejilla, que es muy utilizado para el desarrollo de páginas web adaptativas.

Para el comportamiento de la aplicación web se ha utilizado JavaScript en el lado del cliente y PHP en el lado del servidor. El lenguaje de JavaScript es el más extendido en toda la web, ya que es soportado por todos los navegadores web. Es muy fácil de aprender y utilizar y está totalmente integrado en HTML y CSS. Cabe destacar que hemos usado la librería de JQuery para facilitarnos un poco el trabajo a la hora de interactuar con los documentos PHP.

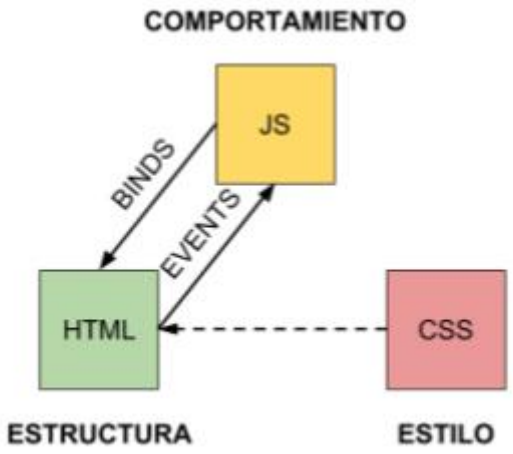

Ilustración 10. Relación HTML, CSS y JavaScript en una página web.

Por lo que respecta a PHP, es de gran ayuda para acceder a la base de datos y enviar/recibir información. He optado por usar este lenguaje de programación porque tiene una gran comunidad de desarrolladores, lo que hace que se pueda encontrar mucha documentación e información en internet. Para la base de datos se ha utilizado MySQL, porque es fácil de utilizar y es la base de datos más utilizada, por lo que es más fácil

encontrar ayuda en internet. La combinación PHP y MySQL hace que nuestra aplicación web sea más robusta y fiable.

4.2.1. HTML5

HTML*, sigla en inglés de *HyperText Markup Language* (lenguaje de marcas de hipertexto), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Es un estándar a cargo del *World Wide Web Consortium* (W3C) o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la *World Wide Web* (WWW). Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado.

Ilustración 11. Logotipo de HTML5.

El lenguaje HTML basa su filosofía de desarrollo en la diferenciación. Para añadir un elemento externo a la página (imagen, vídeo, script, entre otros.), este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. De este modo, la página web contiene solamente texto mientras que recae en el navegador web (interpretador del código) la tarea de unir todos los elementos y visualizar la página final. Al ser un estándar, HTML busca ser un lenguaje que permita que cualquier página web escrita en una determinada versión, pueda ser interpretada de la misma forma (estándar) por cualquier navegador web actualizado.

Actualmente, está estandarizado el uso de HTML5, que es la quinta revisión importante del lenguaje básico de la *World Wide Web*, HTML. HTML5 especifica dos variantes de sintaxis para HTML: una «clásica», HTML (text/html), conocida como HTML5, y una variante XHTML conocida como sintaxis XHTML5 que deberá servirse con sintaxis XML (application/xhtml+xml). Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo. La versión definitiva de la quinta revisión del estándar se publicó en octubre de 2014.

4.2.2. CSS

Hojas de Estilo en Cascada* (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura.

Ilustración 12. Logotipo de CSS3.

CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.

CSS funciona a base de reglas. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML o XML. La regla tiene dos partes: un selector y la declaración. A su vez la declaración está compuesta por una propiedad y el valor que se le asigne.

HTML. Wikipedia: http://es.wikipedia.org/wiki/HTML
HTML5. Wikipedia: http://es.wikipedia.org/wiki/HTML5
Hojas de estilo (CSS). Wc3: http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo

4.2.3. JavaScript

JavaScript* es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

Ilustración 13. Logotipo de JavaScript.

JavaScript se diseñó con una sintaxis similar a C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del *Document Object Model* (DOM).

Actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con ayuda de otras tecnologías como AJAX. JavaScript se interpreta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

4.2.4. PHP

PHP* es un lenguaje de programación interpretado de uso general. Diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página web resultante. PHP ha evolucionado, por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

Ilustración 14. Logotipo de PHP.

PHP se considera uno de los lenguajes más flexibles, potentes y de alto rendimiento conocidos hasta el día de hoy, lo que ha atraído el interés de múltiples sitios con gran demanda de tráfico para optar por él como tecnología de servidor.

Fue creado originalmente por Rasmus Lerdorf en 1995. Actualmente el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP. Este lenguaje forma parte del software libre publicado bajo la licencia PHP, que es incompatible con la Licencia Pública General de GNU debido a las restricciones del uso del término PHP.

4.2.5. MySQL

MySQL* es un sistema de gestión de bases de datos relacional, multihilo y multiusuario. Ha sido desarrollado bajo licencia dual GPL/Licencia comercial por **Oracle Corporation** y está considerada como la base de datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web.

Ilustración 15. Logotipo de MySQL.

JavaScript. Wikipedia: http://es.wikipedia.org/wiki/JavaScript
PHP. Wikipedia: http://es.wikipedia.org/wiki/PHP
MySQL. Wikipedia: http://es.wikipedia.org/wiki/MySQL

4.2.6. Bootstrap

Bootstrap* es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. A diferencia de muchos *frameworks web*, solo se ocupa del desarrollo *front-end*.

Ilustración 16. Logotipo de Bootstrap.

Bootstrap tiene un soporte relativamente incompleto para HTML5 y CSS3, pero es compatible con la mayoría de los navegadores web. La información básica de compatibilidad de sitios web o aplicaciones está disponible para todos los dispositivos y navegadores. Existe un concepto de compatibilidad parcial que hace disponible la información básica de un sitio web para todos los dispositivos y navegadores. Por ejemplo, las propiedades introducidas en CSS3 para las esquinas redondeadas, gradientes y sombras son usadas por Bootstrap a pesar de la falta de soporte de navegadores antiguos. Esto extiende la funcionalidad de la herramienta, pero no es requerida para su uso.

Desde la versión 2.0 también soporta diseños web adaptables. Esto significa que el diseño gráfico de la página se ajusta dinámicamente, tomando en cuenta las características del dispositivo usado (Computadoras, tabletas, teléfonos móviles).

Bootstrap es de código abierto y está disponible en GitHub. Los desarrolladores están motivados a participar en el proyecto y a hacer sus propias contribuciones a la plataforma.

4.2.7. JQuery

jQuery* es una biblioteca o *framework* de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

Ilustración 17. Logotipo de jQuery.

4.2.8. AJAX

AJAX*, acrónimo de *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (**Rich Internet Applications**). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

Ilustración 18. Logotipo de AJAX.

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como JavaScript y *Document Object Model* (DOM).

Bootstrap. Wikipedia: https://es.wikipedia.org/wiki/Bootstrap_(framework)
JQuery. Wikipedia: http://es.wikipedia.org/wiki/JQuery
AJAX. Wikipedia: http://es.wikipedia.org/wiki/AJAX

4.2.9. JSON

JSON*, acrónimo de *JavaScript Object Notation*, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

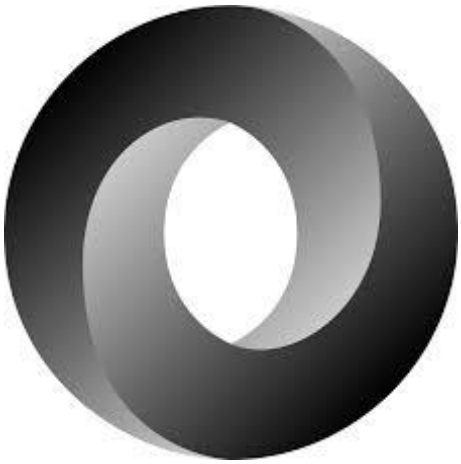

Ilustración 19. Logotipo de JSON.

4.3. Herramientas utilizadas

Para la implementación aplicación web, tanto el *front-end* como el *back-end*, se han utilizado diferentes aplicaciones o herramientas. Tales como:

4.3.1. Editor de texto

Un editor de texto para editar el código fuente de mí aplicación web. Es un programa informático que permite crear y modificar archivos digitales. El programa lee el archivo e interpreta los bytes leídos según el código de caracteres que usa el editor. En mi caso particular, he usado Sublime Text* 3, por todas las características que ofrece, pero se podría haber usado cualquier editor de texto como, por ejemplo, *Visual Studio Code*, *Notepad++*, *Webstorm*, *Atom*, etc.

Ilustración 20. Logotipo de Sublime Text.

JSON. Wikipedia: http://es.wikipedia.org/wiki/JSON
Sublime Text. Wikipedia: https://es.wikipedia.org/wiki/Sublime_Text

4.3.2. XAMPP

Para las pruebas en local se ha utilizado XAMPP*.

Ilustración 21. Logotipo de XAMPP.

XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. Desde la versión "5.6.15", XAMPP cambió la base de datos de MySQL a MariaDB. El cual es un *fork* de MySQL con licencia GPL. El programa se distribuye bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y Mac OS X.

Ilustración 22. Panel de control del XAMPP.

XAMPP. Wikipedia: http://es.wikipedia.org/wiki/XAMPP

4.3.3. phpMyAdmin

Para mantener la base de datos MySQL remotamente hemos utilizado la herramienta phpMyAdmin*.

Ilustración 23. Logotipo de PHPMyAdmin.

phpMyAdmin es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 72 idiomas. Este proyecto se encuentra vigente desde el año 1998, siendo el mejor evaluado en la comunidad de descargas de SourceForge.net como la descarga del mes de diciembre del 2002. Como esta herramienta corre en máquinas con Servidores Webs y Soporte de PHP y MySQL, la tecnología utilizada ha ido variando durante su desarrollo.

phpMyAdmin. Wikipedia: http://es.wikipedia.org/wiki/PhpMyAdmin

5. Análisis

En este apartado vamos a describir la fase de análisis del proyecto. En esta fase se analiza la aplicación que vamos a desarrollar y se describe su estructura y funcionalidad mediante diagramas UML (*Lenguaje unificado de modelado*), que es un lenguaje de modelado de sistemas de software. Usamos este lenguaje porque es el más usado y el más popular en la actualidad. Además, UML está apoyado por el OMG (*Object Management Group*).

Ilustración 24. Logo de UML.

UML es un lenguaje gráfico que sirve para visualizar, especificar, construir y documentar un sistema. Lo vamos a utilizar principalmente para modelar nuestro sistema, es decir, nuestra aplicación web. A continuación, se va a mostrar cómo se ha detallado y se ha construido los métodos y atributos que están presentes en el interior de la aplicación web y dan vida a todo el desarrollo.

Para la realización de dicho desarrollo, hemos diseñado varios diagramas para poder entender mejor la aplicación, que son: diagrama de actividad, diagrama de casos y diagrama de clases.

5.1. Diagrama de actividad

El diagrama de actividad es un diagrama de flujo del proceso que se usa para modelar el comportamiento de nuestro sistema. En este caso, no es un diagrama de actividad puro, ya que normalmente se muestra un proceso paralelo (*parallel processing*). Se ha utilizado principalmente para mostrar, de forma gráfica, el proceso o rumbo que va a tomar la aplicación web.

A continuación, se muestra el proceso que debe de atravesar un usuario desde que entra en nuestro dominio hasta que termina por utilizar la aplicación al realizar la formación de los grupos de trabajo. Para la realización de la misma, se va a dividir el diagrama de actividad en dos partes.

En la primera parte, el diagrama de actividad empieza cuando el usuario entra a la página *index.html*. Como primer punto crítico, el usuario deberá de iniciar la sesión para poder proseguir con la aplicación.

Una vez el usuario ha iniciado sesión, se redireccionará a la página del cuadro de mando (*dashboard.php*) dónde se mostrará todas las asignaturas con listas de usuario creadas (ver en segunda parte), pero como es el primer ciclo o proceso, sólo podremos crear una nueva lista de usuarios, por lo que deberemos de ir a la página *upload.php*.

Aquí, subiremos una lista de los alumnos de la asignatura pertinente a realizar la formación de grupos. De esta página, iremos a *lista.php* donde podremos editar la lista de alumnos, con la capacidad de añadir o borrar alumnos en dicha lista.

Para finalizar, visitaremos la página *correo.php*, que es donde veremos el listado final y como último paso será enviar un correo electrónico a todos los alumnos de dicho listado para que realicen el test de Azulay-Bernstein (test ADNe). Una vez enviado los correos, terminaría la primera parte del diagrama de actividades.

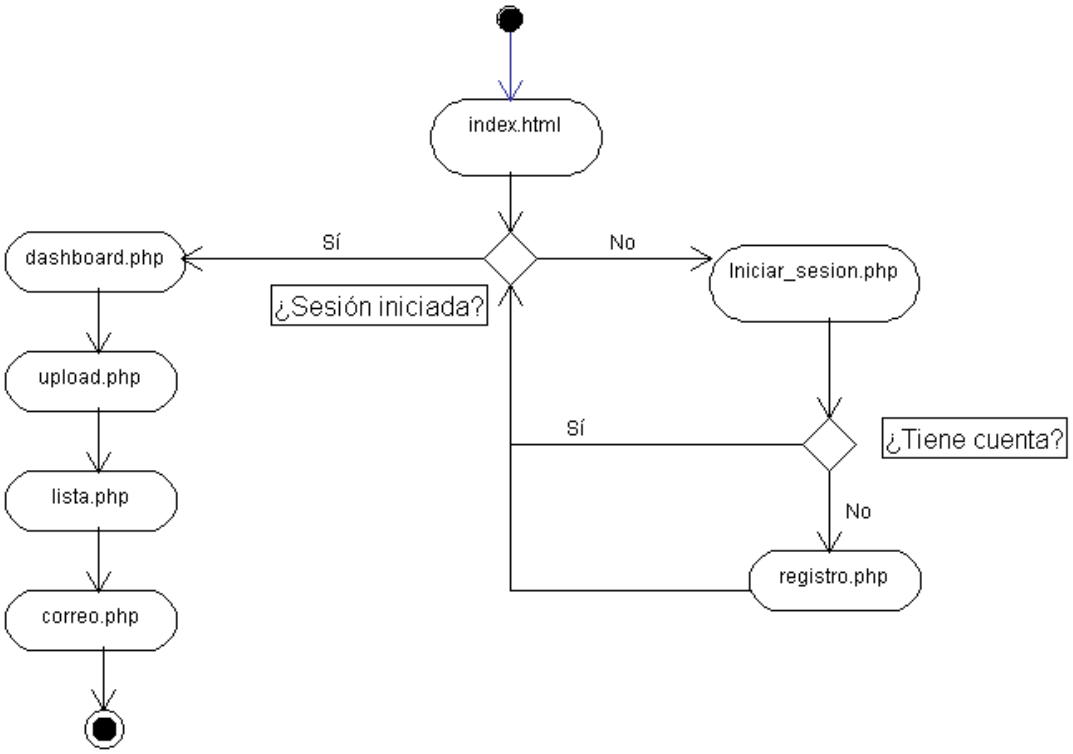

Ilustración 25. Diagrama de actividad inicial.

Nota: hemos creado una página llamada registro.php para poder añadir a la base de datos los profesores con credenciales que van a utilizar la aplicación web. Como en este TFG aún no se ha contemplado el uso de la aplicación en el interior de PoliformaT, es necesario poseer una base de datos con los usuarios y sus contraseñas para poder iniciar sesión.

En la segunda parte, con la **cuenta iniciada**, empezaríamos en el cuadro de mando (*dashboard.php*) y deberíamos de tomar la segunda decisión, si crear un listado nuevo y repetir el primer ciclo explicado con anterioridad, o en vez de eso, terminar el proceso con un listado ya creado.

Si nos decidimos por terminar el ciclo, deberíamos de esperar un tiempo a que todos los alumnos de la asignatura hayan realizado el test ADNe, de no ser así, no se podrían formar los grupos de trabajo por falta de datos (esto es uno de los puntos débiles de la aplicación). En el caso afirmativo de que todos los alumnos hayan realizado el test, deberíamos de tener los datos suficientes para realizar la formación de grupo.

Una vez superado el punto crítico, iríamos a la página *editar_grupo.php*, donde podremos volver a editar los datos de cualquier alumno e, incluso, editar a los alumnos que no han

hecho el test para que podamos proseguir con el uso de la aplicación sin tener a estos alumnos en cuenta.

Una vez ya tenemos todo listo para formar los grupos, nos dirigiremos a la página *formación_grupo.php*, donde el usuario podrá especificar el número de grupos totales y podrá aplicar restricciones a la hora de formar los grupos. El propio algoritmo formará los grupos de trabajo óptimos y otro listado con los grupos si se han aplicado algún tipo de restricción por parte del profesor de la asignatura.

Para finalizar, será el propio criterio del profesor tomar la decisión de cúales serán los grupos finales que se van a formar en su asignatura.

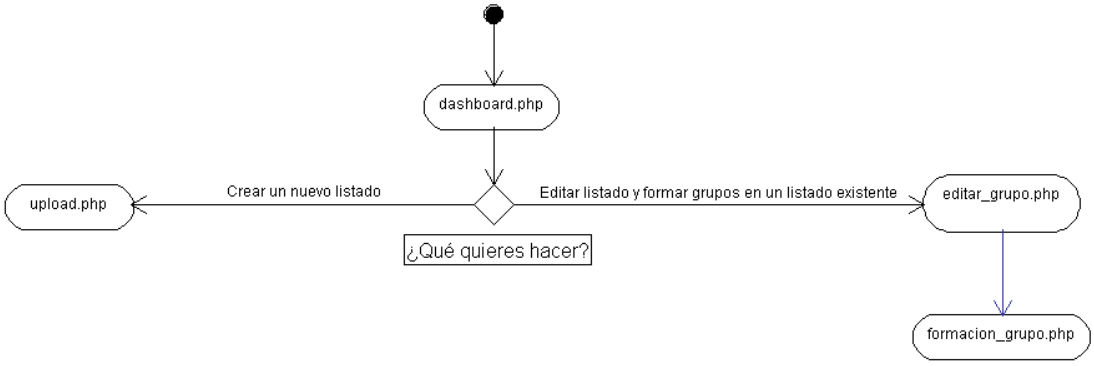

Ilustración 26. Diagrama de actividad una vez tenemos un listado o más creados.

Este sería el proceso total desde que iniciamos la sesión hasta que hacemos la formación de grupos de una asignatura. La aplicación sería capaz de crear infinidad de grupos para todas las asignaturas que el profesor quiera realizar.

5.2. Diagrama de casos de uso

Los diagramas de casos de uso describen el comportamiento de la aplicación web y enfatizan la interacción entre uno o varios actores con el comportamiento del sistema modelado.

En nuestra aplicación web, sólo hay dos diagramas de casos de uso, centrados en los dos únicos actores presentes: el usuario con inicio de sesión y el usuario que no ha iniciado la sesión.

A continuación, vamos a mostrar los casos de uso para cada actor.

5.2.1. Caso de uso: usuario anónimo

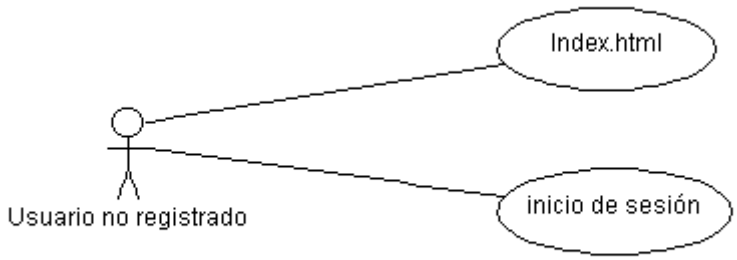

Ilustración 27. Caso de uso: usuario anónimo.

El usuario anónimo es todo aquel que accede a la aplicación web sin identificarse en el sistema. Este actor tendrá el acceso totalmente limitado a casi todos los sitios de nuestra página web. Únicamente podrá iniciar sesión.

5.2.2. Caso de uso: usuario registrado

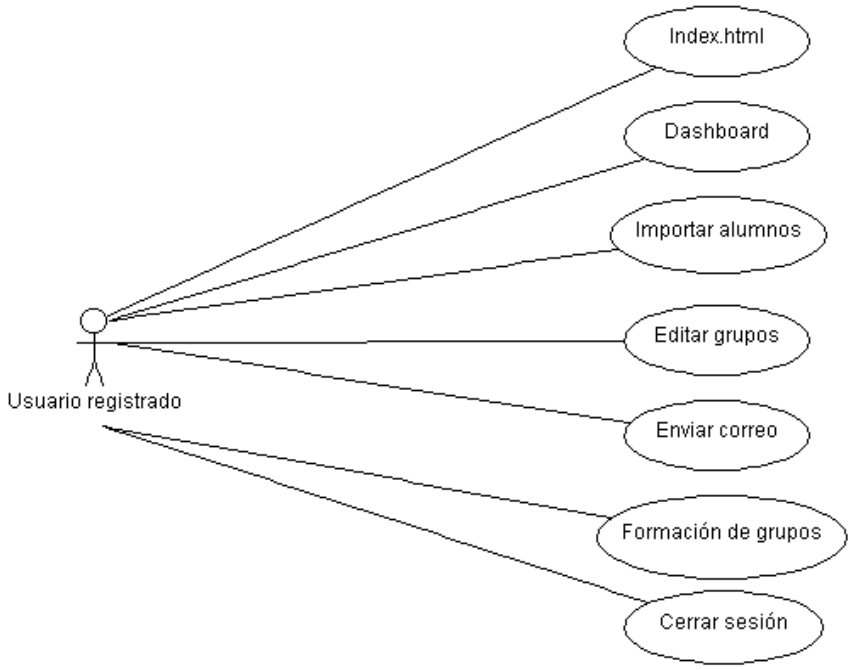

Ilustración 28. Casos de uso: usuario con sesión iniciada como docente.

El usuario registrado es toda aquella persona que forma parte del personal docente o afiliado de la *UPV*, es decir, se trata del profesorado o profesores externos a la universidad con credenciales entregadas por la *UPV*. Este usuario tendrá los privilegios de poder crear nuevas listas de alumnos, editarlas y poder realizar la formación de grupos mediante el algoritmo *ADNe*.

Dicho de otro modo, está aplicación está pensada para que los sólo los profesores de la *Universitat Politècnica de València* puedan usar esta aplicación. Esto no quiere decir que

esté totalmente limitada para su uso exclusivo académico. En un futuro se podría implementar la aplicación web para usarse en entornos laborables, por ejemplo.

5.3. Mockups

Los dibujos conceptuales en papel, bocetos o, del inglés, mockups, son una de las mejores herramientas a la hora de diseñar la interfaz gráfica que se utilizará en la aplicación web. Hemos utilizado este tipo de boceto porque nos permite, en una fase temprana del desarrollo, realizar una primera evaluación de cómo se vería visualmente la aplicación y que características deben de implementarse.

Gracias a esta técnica nos haremos a la idea de cómo va a quedar nuestra aplicación web. Además, siempre se podrán efectuar los cambios oportunos antes de la implementación. A continuación, se muestra un dibujo que representa una pantalla de la interfaz gráfica:

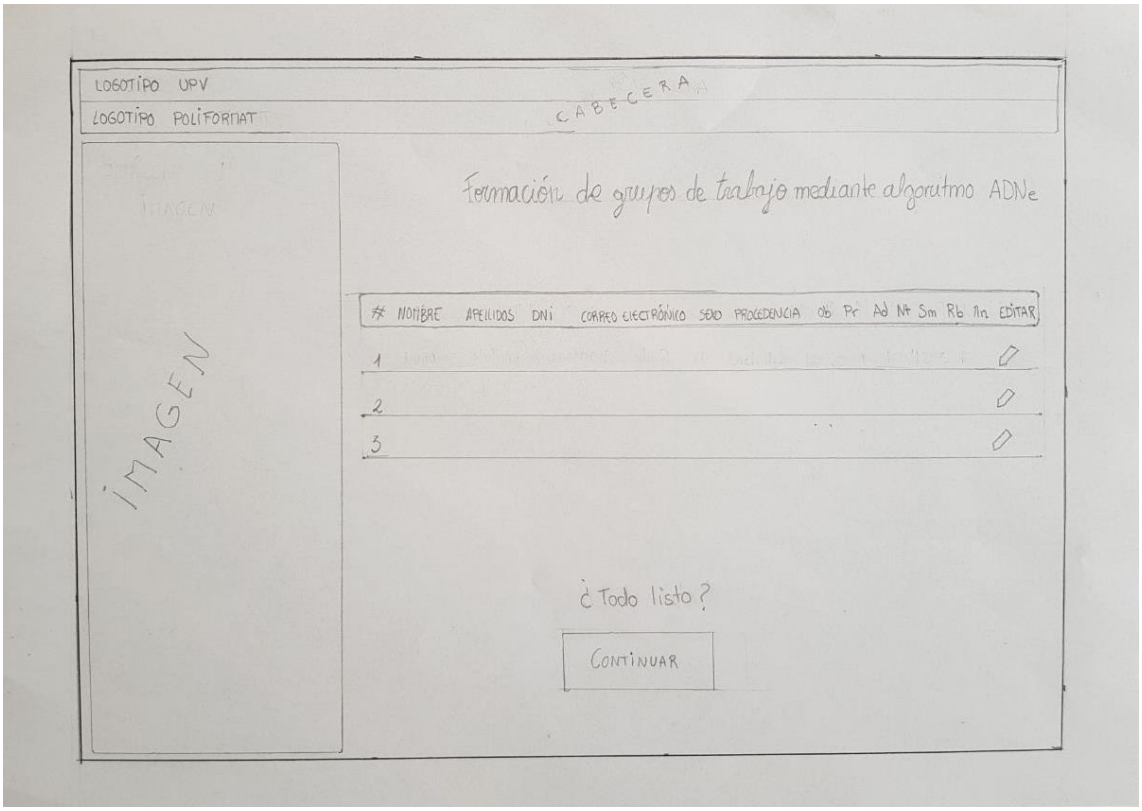

Ilustración 29. Boceto del interfaz gráfico de la aplicación web.

En la figura 29, se muestra un primer diseño de la aplicación web. En ella, podemos observar una cabecera y una imagen lateral parecidas a *PoliformaT*, y con el contenido céntrico, tomando el mayor espacio de la ventana.

En esta ventana en especial, se puede observar que se va a listar cada alumno con todos los datos relevantes para efectuar la formación de grupos mediante el algoritmo *ADNe*.

6. Diseño gráfico y usabilidad

6.1. Diseño gráfico

Cuando hablamos de diseño gráfico y, en especial, al diseño web, no solo hablamos de el aspecto visual que va a tener un sitio o aplicación web. El diseño web engloba tanto el aspecto visual, como el diseño del interfaz y experiencia del usuario, la navegabilidad, interactividad, usabilidad (explicada más adelante) y arquitectura de la información.

A nivel visual, en cualquier página web nos podemos encontrar texto plano, imágenes, audios y videos, como enlaces a otros sitios web. No vale estar dentro de la página web, debe de cumplir unos requisitos de diseño como de usabilidad e interactividad.

Además, la imagen transmitida por el diseño de una página web debe causar una impresión positiva al usuario y debe captar su atención, ya que los usuarios tienden a juzgar la credibilidad y la confianza que les ofrece una página web en base a aspectos como su diseño visual.

Una vez tenemos claros los conceptos para saber qué es importante en un diseño web, he querido enfatizar el diseño visual pareciéndose al mismo diseño que posee *PoliformaT*. Para ello, hemos tenido en cuenta los colores corporativos de la *Universitat Politècnica de València,* como los diferentes elementos que posee su aula virtual. Dichos elementos se han visto reflejados en los bocetos vistos en el punto 5.3. de este documento, donde se han realizado antes de empezar el desarrollo de la aplicación web.

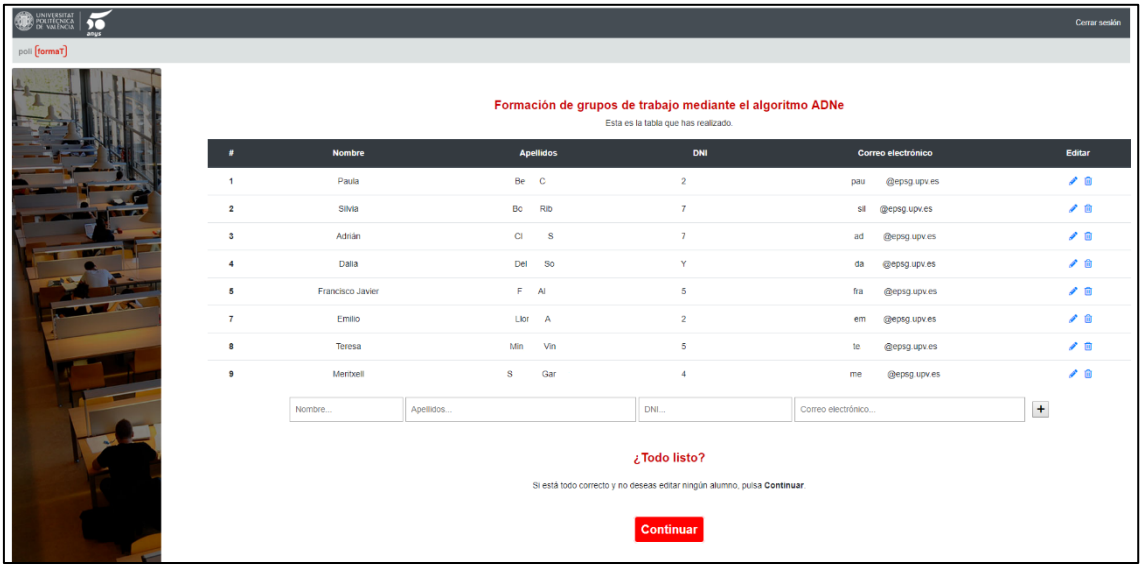

Ilustración 30. Ejemplo del aspecto visual de la aplicación web.

6.2. Diseño adaptable

El diseño web adaptable (del inglés web responsive) es una nueva tendencia de diseño y desarrollo web cuyo objetivo es adaptar la apariencia de las páginas web al dispositivo que se esté utilizando. Hoy en día, las páginas web se ven en multitud de dispositivos como tabletas, teléfonos móviles, libros electrónicos, portátiles, PC, etcétera. Además, aún dentro de cada tipo, cada dispositivo tiene sus características concretas, por lo que esta tecnología pretende que, con un único diseño web, todo se vea correctamente en cualquier dispositivo.

El uso de dispositivos móviles ha aumentado notablemente en esta última década, en particular, los teléfonos móviles inteligentes y las tabletas. La evolución de la navegación en Internet ha ido a la par, y ello ha popularizado la navegación en Internet mediante una creciente variedad de dispositivos y resoluciones de pantalla, lo que a su vez ha creado unas necesidades de adaptar la experiencia de la navegación web a ellos.

La gran ventaja en el uso de esta tecnología es que con una sola versión de HTML y CSS se pueden cubrir todas las resoluciones de pantalla, con lo que el sitio web estará optimizado para distintos dispositivos. Esto mejora la experiencia de usuario a diferencia de lo que ocurre, con sitios web de ancho fijo (como, por ejemplo, la página web de la UPV) cuando se acceden desde dispositivos móviles. Además, desde el punto de vista del posicionamiento en buscadores (o SEO), aparecería una única URL en los resultados de búsqueda, con lo cual se ahorrarían múltiples redirecciones y los fallos que se derivan de estas.

El diseño web adaptable se hace posible gracias a la introducción de las *media queries* en las propiedades de los estilos CSS3. Las *media queries* son una serie de órdenes que se incluyen en la hoja de estilos que indica al documento HTML cómo debe comportarse en diferentes resoluciones de pantalla.

Para entenderlo mejor, los diseños de las páginas web, al igual que los periódicos y las revistas, están basados en columnas, entonces con la filosofía del diseño adaptativo, si una web a resolución de PC (1080x1920 píxeles) tiene 8 columnas, para una tableta (800x600 píxeles) necesitaría sólo 4, y en el caso de un teléfono inteligente cuyo ancho suele ser entre 320 y 480 píxeles las columnas usadas serían 3.

El diseño adaptable debe fluir con una adaptación constante del tamaño de los gráficos y las estructuras compositivas de un sitio web dentro de los diferentes dispositivos y tamaños de pantalla considerando de forma automática la disposición en la que se visualizan los contenidos.

La aplicación web ha sido desarrollada con un diseño adaptable a todo tipo de dispositivos. A continuación, se muestran las figuras 31 y 32 como ejemplo de que se ha llevado a cabo un diseño adaptable.

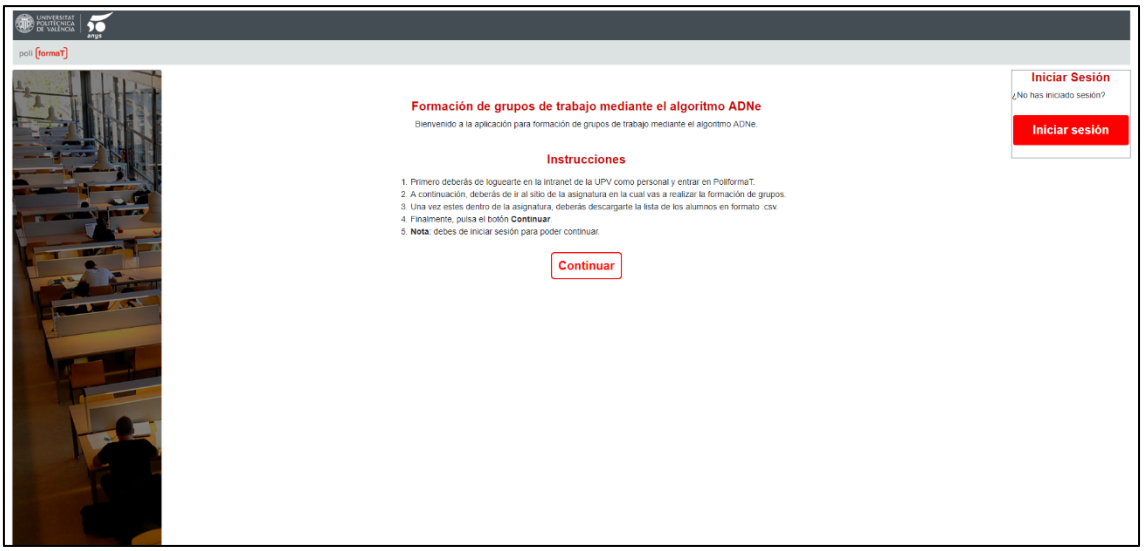

Ilustración 31. Página de inicio desde un PC, con resolución 1080x1920.

Como se puede observar en la figura 31, se esta utilizando todo el ancho de pantalla a una resolución de 1080p, teniendo como elemento principal el texto de introducción a la aplicación web. En la imagen se aprecia como el elemento está totalmente centrado y podemos observar en los laterales una imagen decorativa y un elemento de inicio de sesión flotante a la parte superior derecha.

Ilustración 32. Página de inicio desde un dispositivo móvil, con una resolución de 400x800.

En la figura 32, podemos observar a primera vista que se trata de un dispositivo móvil, por la propia resolución de la imagen. De forma innata, observamos de como la imagen que aparecía en la figura 31 no se encuentra en esta pantalla. Además, el botón flotante que aparecía en anterioridad, ahora se encuentra debajo del texto principal, totalmente adaptado a la nueva resolución.

Esto es sólo un pequeño ejemplo del trabajo realizado para adaptar la aplicación web utilizando las *media queries* de CSS3 para que nuestra aplicación web de formación de grupos eficientes se pueda utilizar en todo tipo de entornos y en todo tipo de dispositivos electrónicos.

6.3. Usabilidad

Definimos como usabilidad a la facilidad con que las personas pueden utilizar una herramienta particular o cualquier otro objeto fabricado por humanos con el fin de alcanzar un objetivo concreto que, en nuestro caso, hablamos de la usabilidad web.

La usabilidad web es la disciplina que estudia la forma de diseñar sitios web para que los usuarios puedan interactuar con ellos de la forma más fácil, cómoda e intuitiva posible.

Parte fundamental de la usabilidad de un sitio web está determinada por los signos gráficos que representan elementos de control y medida para los usuarios y les proporcionan referentes visuales relacionados a elementos que se encuentran en su vida cotidiana y que al interactuar con ellos en el sitio web les permite con facilidad comprenderlo y utilizarlo. Los signos gráficos pueden ser imágenes, botones, menús de navegación, entre otros.

Ilustración 33. Signos gráficos para editar y borrar un alumno en la aplicación web.

La investigación en usabilidad web se centra, principalmente, en la comodidad, navegación, arquitectura de la información y contenido de una aplicación web, y es importante tener en cuenta que la percepción de usabilidad tiene una influencia positiva en el usuario.

Actualmente, las personas tienen mayor acceso a contenidos web desde dispositivos móviles. Sin embargo, cuando el usuario navega en la web desde su teléfono móvil, se enfrenta a problemas de usabilidad debido a que muchos de los contenidos web no están adaptados para visualizarse en dispositivos con pantallas pequeñas.

Para mejorar la experiencia de usuario en servicios móviles es necesario considerar dos aspectos importantes de la usabilidad: la simplicidad y la interactividad. La simplicidad es considerada un concepto crucial para el diseño exitoso de una interfaz de usuario porque la creciente complejidad de la tecnología hace nuestra vida compleja y desordenada. La interactividad ha sido estudiada como una característica esencial de los sitios web que conduce las actividades y actitudes de los usuarios.

A la hora de desarrollar nuestra aplicación web, hemos tenido el concepto de usabilidad presente en todo momento. Para ello, hemos cumplido los siguientes requisitos:

- Estructurar los contenidos.
- Diseño limpio.
- Ceder el control al usuario.
- Facilitar la interacción.
- Simplificar y sintetizar.
- Adaptar la web a todo tipo de dispositivos.

7. Conclusiones

7.1. Valoración personal

Cuando vi este trabajo fin de grado en el listado de ofertas públicas, tuve una especial ilusión por llevarlo a cabo. El propio título ya destacaba sobre los demás y es bastante atractivo como para llamarme la atención de esa forma. Me considero una persona que le gusta la tecnología y programar, por eso no lo dudé y contacté con José Marín para poder hacer este TFG.

Este trabajo fin de grado ha sido un verdadero reto, tanto a nivel personal, como profesional. Yo tenía unas nociones básicas de HTML5, CSS y *Javascript*, pero no tenía ni idea de trabajar en *back-end*, es decir, con código *PHP* y *MySQL*. Es por eso, que el desarrollo de esta aplicación web fue una carrera de obstáculos. Tuve que aprender a desarrollar entornos web con el servidor y trabajar en el TFG por partes iguales. Fue tan ardua la tarea que incluso hubo momentos que quise tirar la toalla.

Al final, persistí y seguí trabajando hasta que el esfuerzo dio sus frutos. Este trabajo es el resultado de este gran esfuerzo y la recompensa no es solo entregar una aplicación web que funciona y puede llegar a ser una herramienta útil para la propia universidad, sino que también he crecido como profesional. Al final, la satisfacción personal de poder haber superado un reto de esta envergadura supera con creces todos los contratiempos y problemas que he sufrido por el camino.

He aprendido a utilizar nuevas herramientas, nuevos lenguajes de programación y métodos de trabajo muy útiles en el sector del desarrollo de software. Es por eso que este TFG me ha preparado para hacer frente al mundo laboral.

Mi conclusión final es que este *Trabajo Final de Grado* ha sido una experiencia satisfactoria y que esta aplicación web puede dar mucho de qué hablar en los próximos años si es que la *Universitat Politècnica de València* está dispuesta a implementarla como herramienta en el propio *PoliformaT*.

7.2. Futuras implementaciones

La verdad es que la propia aplicación web podría ofrecer muchas más mejoras. Aquí se explica cómo funciona el 90% de la app, pero la verdad es que aún le faltarían unas mecánicas para que funcionara al 100%. Las futuras implementaciones serían:

- La compañía ***Jacobson, Steinberg & Goldman*** haya terminado la *API* que interconecta la parte de enviar los correos a los alumnos para que realicen el test de Azulay-Bernstein con la parte de crear los grupos. Una vez implementada la API en esta aplicación web, el resultado sería más rápido y eficaz. El profesorado no tendría que hacer tantas tareas, por lo que sería un proceso más automático y evitaría cualquier tipo de error humano.
- Aumenta en **escalabilidad** y **usabilidad**, haciendo que el profesorado pueda usar la aplicación web en varias asignaturas e, incluso, combinarlas entre sí.
- Aplicar una **capa de seguridad** robusta a la aplicación web para no vulnerar los datos personales de los alumnos ni cualquier dato que se halle en la base de datos. He aplicado las medidas mínimas en la app, como proteger las páginas con información sensible con un *inicio de sesi*ón o, incluso, con aplicar un *hash* en las contraseñas de los usuarios al guardarlas en la base de datos.

- El **diseño gráfico** de la propia página web podría mejorarse y hacerla más atractiva visualmente. He decidido ponerlo en un segundo plano, ya que aquí lo más importante es la lógica de la aplicación web.

Estas serían unas de las futuras mejoras más destacadas. Pero si de verdad la *UPV* acoge este TFG para implementarla dentro de *PoliformaT*, los cambios más destacados serían:

- Alojar la aplicación web dentro de los servidores de la UPV, por lo que se deberían de cambiar todas las rutas internas de la app.
- Se eliminaría el inicio y uso de las sesiones, ya que se usarían las credenciales de *PoliformaT*.
- No sería necesario implementar una capa de seguridad, ya que la aplicación estaría bajo el amparo de *PoliformaT*, teniendo una capa de seguridad muy robusta.
- Se aligeraría el proceso. No deberíamos de descargar ningún archivo *CSV*, sino que directamente, se pulsaría un botón dentro de la asignatura y ya pasaríamos a editar la lista de alumnos, por si se quisieran añadir alumnos de otras asignaturas.

8. Bibliografía

- Algoritmo *ADNe:*
  https://jcbson.com/
  https://riunet.upv.es/bitstream/handle/10251/116593/El%20ADNe%2C%20la%20nueva%20herramienta%20para%20la%20gesti%C3%B3n%20de%20los%20recursos%20humanos.pdf

- Eric Kandel:
  http://www.aperturas.org/articulo.php?articulo=0000546
  https://psicologiaymente.com/neurociencias/tipos-neurotransmisores-funciones
  https://es.wikipedia.org/wiki/Eric_Kandel

- Neuromoduladores y neurotransmisores:
  https://es.wikipedia.org/wiki/Neuromodulador
  https://es.wikipedia.org/wiki/Neurotransmisor
  https://psicologiaymente.com/neurociencias/neurotransmisores-neuromoduladores

- HTML5, CSS, JS, SQL, JQuery:
  https://www.w3schools.com/html/default.asp
  https://www.w3schools.com/css/default.asp
  https://www.w3schools.com/js/default.asp
  https://www.w3schools.com/sql/default.asp
  https://www.w3schools.com/jquery/default.asp

- Documentación JavaScript adicional:
  https://developer.mozilla.org/es/docs/Web/JavaScript

- Documentación PHP:
  https://www.php.net/manual/es/

- Documentación Bootstrap:
  https://getbootstrap.com/docs/

- Documentación PHPMyAdmin:
  https://www.phpmyadmin.net/docs/

- Usabilidad:
  https://es.wikipedia.org/wiki/Usabilidad

- Diseño adaptable:
  https://es.wikipedia.org/wiki/Dise%C3%B1o_web_adaptable

- Wikipedia:
  https://es.wikipedia.org/