

Manuel Navajas Segovia¹ y Sonia Pamplona Roche²

Evaluación de la facilidad de aprendizaje de *frameworks* JavaScript: Backbone, Angular y Ember

Sumario

1. Introducción
2. Metodología
3. Resultados
4. Conclusiones
5. Bibliografía

Apéndices

Extracto:

JavaScript es uno de los lenguajes de programación más utilizados en el desarrollo web, lo que ha provocado la aparición de una gran variedad de *frameworks* JavaScript y la consiguiente dificultad para seleccionar uno de ellos. El presente estudio considera la usabilidad como un factor de adopción de un *framework* JavaScript y tiene como objetivo evaluar la usabilidad de tres conocidos *frameworks* JavaScript: Backbone, Angular y Ember. La evaluación de la usabilidad se realizó mediante la técnica del test de usabilidad y se centró en los siguientes atributos: facilidad de aprendizaje, tasa de errores y satisfacción. Los resultados mostraron que Angular fue el *framework* que los participantes del estudio percibieron con mayor usabilidad, seguido por Ember y por Backbone en último lugar.

Palabras clave: facilidad de aprendizaje, usabilidad, *framework* JavaScript, Backbone, Angular, Ember.

Fecha de entrada: 01-08-2018

Fecha de aceptación: 30-08-2018

Fecha de revisión: 04-09-2018

¹ M. Navajas Segovia, egresado del Máster Universitario en Arquitectura del Software de la Universidad a Distancia de Madrid (UDIMA) y analista programador en la empresa ALTEN.

² S. Pamplona Roche, profesora ayudante doctora de la Universidad a Distancia de Madrid (UDIMA).

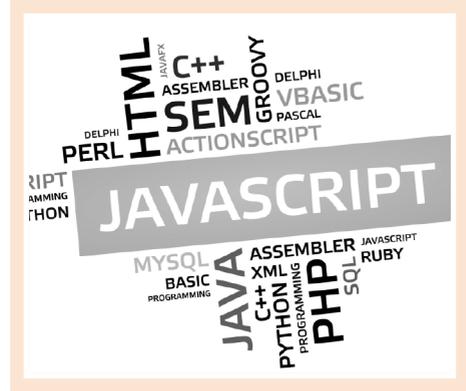
Evaluation of the learnability of JavaScript frameworks: Backbone, Angular and Ember

Abstract:

JavaScript is one of the most used programming languages in web development, which has led to the emergence of a variety of JavaScript frameworks and the consequent difficulty in selecting one of them. The present study focuses on usability as a factor of adoption of a JavaScript framework and aims to evaluate the usability of three well-known JavaScript frameworks: Backbone, Angular and Ember. The evaluation of usability was carried out using the usability test technique and focused on the following attributes: learnability, error rate and satisfaction. The results showed that Angular was the framework that the participants of the study perceived with greater usability followed by Ember and Backbone in last place.

Keywords: learnability, usability, JavaScript framework, Backbone, Angular, Ember.

La [...] proliferación de *frameworks* hace difícil la selección de uno de ellos. [...] son necesarios estudios que ayuden en esta tarea, puesto que la elección de un *framework* u otro puede incidir de forma considerable en el rendimiento de un proyecto



1. INTRODUCCIÓN

JavaScript es uno de los lenguajes de programación más populares de la web y también unos de los más usados (Mariano, 2017). Este hecho ha provocado la aparición de una gran variedad de *frameworks* JavaScript. Los *frameworks* son estructuras de código genérico y reusable que ayudan a los desarrolladores en las tareas de programación con el fin de aumentar la productividad (Scholten, Tromp y Nijhuis, 2017). La mencionada proliferación de *frameworks* hace difícil la selección de uno de ellos. Así pues, son necesarios estudios que ayuden en esta tarea, puesto que la elección de un *framework* u otro puede incidir de forma considerable en el rendimiento de un proyecto (Pano, Graziotin y Abrahamsson, 2018). Sin embargo, la investigación en esta área ha sido escasa y existen muy pocos estudios que aporten información para seleccionar un *framework* JavaScript. Uno de los primeros trabajos que prestó atención sobre este problema evaluó la calidad y el rendimiento de seis *frameworks* JavaScript teniendo en cuenta factores de calidad del *software* y el test de rendimiento (Gizas, Christodoulou y Papatheodorou, 2012). Por otra parte, Graziotin y Abrahamsson (2013) presentaron un diseño de investigación para comparar *frameworks* JavaScript.

Recientemente se ha culminado una tesis de máster que realiza una comparativa de tres *frameworks* JavaScript. Para ello se desarrolló una aplicación de *benchmarking* que simula eventos de usuario y utiliza métricas de complejidad de *software* y experimentos para medir estas métricas (Mariano, 2017).

Otra metodología que se ha utilizado en este tipo de estudios es la cualitativa interpretativa (Pano *et al.*, 2018). En dicho estudio se realiza una investigación sobre los factores que conducen a la adopción de un *framework* JavaScript haciendo entrevistas a 18 participantes expertos en desarrollo web.

La originalidad de nuestra investigación reside en considerar el grado de usabilidad de un *framework* como un factor de adopción para el mismo y en realizar un análisis de la usabilidad de tres conocidos *frameworks* JavaScript. La usabilidad es una medida de lo bien que un sistema *software* facilita a usuarios específicos la realización de determinadas tareas (Ferré, 2015). En particular, nuestro estudio analiza los siguientes aspectos de la usabilidad:

- Facilidad de aprendizaje.
- Tasa de errores.
- Satisfacción.

El objetivo de este estudio es evaluar la usabilidad de los *frameworks* JavaScript Backbone, Angular y Ember. Para ello, se ha usado la técnica principal a la hora de realizar este tipo de evaluaciones: el test de usabilidad. Las siguientes preguntas de investigación han guiado el estudio:

- ¿Qué resultados se han obtenido en el test de usabilidad llevado a cabo con los tres *frameworks* seleccionados?
- ¿Cómo valoran los participantes del test cada uno de los *frameworks* con respecto a la facilidad de aprendizaje?
- ¿Cómo valoran los participantes del test cada uno de los *frameworks* con respecto a la tasa de errores?
- ¿Cómo valoran los participantes del test cada uno de los *frameworks* con respecto a la satisfacción?

2. METODOLOGÍA

En este trabajo se ha realizado una evaluación de la usabilidad de tres *frameworks* de JavaScript, usando la técnica del test de usabilidad (Ferré, 2015). A continuación se describe el perfil de los participantes, el diseño y la configuración del entorno informático utilizado, así como los diferentes test llevados a cabo.

La originalidad de nuestra investigación reside en considerar el grado de usabilidad de un *framework* como un factor de adopción para el mismo y en realizar un análisis de la usabilidad de tres conocidos *frameworks* JavaScript

2.1. Perfil de los participantes del test

El perfil de los participantes de este estudio fue un perfil técnico, con conocimientos avanzados del mundo del desarrollo y la programación. Este perfil precisaba de estudios superiores relacionados con la informática, con unos cinco o seis años de experiencia en el mundo del desarrollo web, y familiarización con tecnologías .NET. También era necesario que los participantes no tuvieran conocimientos o experiencia previa con *frameworks* de JavaScript.

Cada participante realizó tres test de usabilidad diferentes, uno para cada uno de los *frameworks* estudiados. Se permitió a los participantes realizar los test de forma remota desde su hogar con el fin de adaptarse a los horarios y circunstancias personales de cada uno de ellos.

Dado que cada test requería el empleo de varias horas para su realización, también se permitió que los participantes completaran los test en varias sesiones. Se estableció que cada participante debía finalizar los tres test en un plazo no superior a cuatro semanas.

2.2. Diseño y configuración del entorno

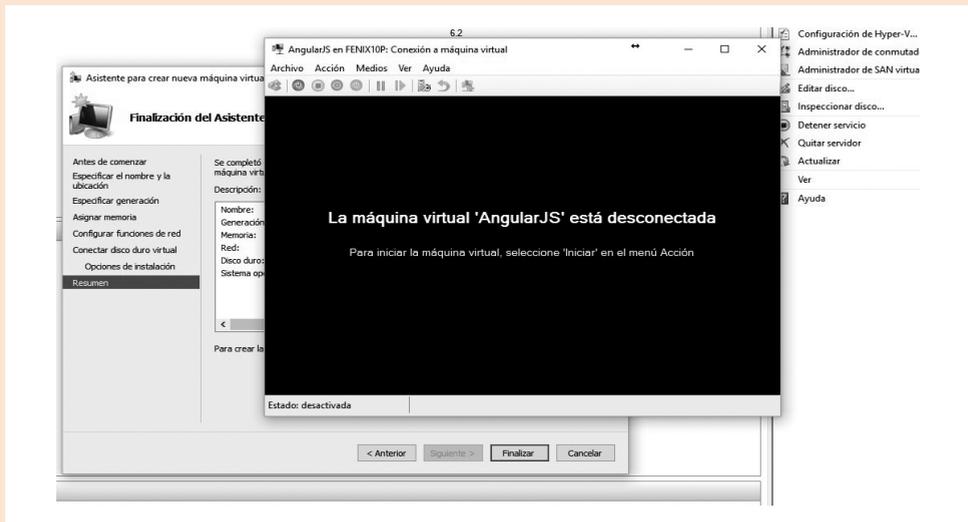
El sistema en el que se desarrollaron los test se basó en un conjunto de tres máquinas virtuales, una por cada *framework* que se planteó estudiar. Una máquina virtual es un *software* que simula un ordenador y que puede ejecutar programas como si fuese uno real (Smith y Nair, 2005).

Para crear las máquinas virtuales se utilizó el asistente de creación de máquinas virtuales de Hyper, programa

de virtualización que permite crear y administrar máquinas virtuales, disponible e integrado con Windows 10 Professional (véase figura 1). Después, se cargó una imagen ISO de Windows 10 Professional en cada una (se podría haber utilizado también cualquier otro sistema

operativo de Microsoft). La instalación se podría haber realizado también con un servidor remoto de instalación en red (Shah, 2013). Una vez creada la máquina, e instalado el sistema operativo, se arrancó el administrador de Hyper-V y se instaló el resto del *software* necesario.

Figura 1. Conexión a una máquina virtual desde el administrador Hyper-V



Fuente: elaboración propia.

También se utilizó la aplicación Team Viewer (*software* para conexiones remotas), que permite tener varias conexiones simultáneas con máquinas reales o virtuales situadas en localizaciones diferentes. Gracias a este *software*, tanto los participantes del test como el responsable de dirigirlo pudieron conectarse a las máquinas.

Por último, hay que indicar que cada máquina tenía instalado el *software* necesario básico para poder desarrollar el test (Visual Studio y *frameworks*). De esta forma, los participantes solo tuvieron que entrar en la máquina virtual para la realización del test, sin instalar ni configurar ningún *software* de desarrollo.

Visual Studio es el entorno de desarrollo integrado (*integrated development environment* [IDE]) para sistemas operativos de la compañía Microsoft. Tiene soporte para diferentes lenguajes de programación, siendo los más importantes actualmente C# y VB. Tiene varias versiones

tanto comerciales como gratuitas, disponibles desde el sitio web oficial de la aplicación <<https://www.visualstudio.com/es/>>. La instalación de este IDE es sencilla y, mediante un asistente guiado, se puede completar de forma correcta y sin dificultades.

2.2.1. Características del hardware empleado

Para la construcción del entorno de los test, se empleó un ordenador con un procesador Intel Core i5-2320 y 16 Gb de memoria RAM, además de un disco duro de 1Tb. La memoria RAM tiene especial importancia en el entorno desarrollado. Se debe a que todos los procesos y el propio funcionamiento de la máquina virtual están siempre limitados por los recursos que dispone la máquina física. Por este motivo se utilizó un equipo con 16 Gb de RAM, con el fin de que varias máquinas virtuales funcionasen a la vez de forma eficiente y fluida.

Por último, hay que indicar que el sistema operativo que se utilizó fue Windows 10 Professional, tanto para la máquina principal como para las máquinas virtuales.

2.2.2. Creación de los proyectos de prueba

Se usó una máquina virtual por cada uno de los *frameworks* estudiados por motivos organizativos. Esta decisión proporcionó las siguientes ventajas:

- Separar las pruebas de los participantes.
- Agrupar los test por tipo de *framework*.
- Simultaneidad en la ejecución de varios test.

Dada la extensa duración de las sesiones, se contó con la posibilidad de que dos participantes tuvieran que realizar el test en el mismo intervalo de tiempo. El hecho de que las máquinas estuviesen separadas permitía la posibilidad de que varios participantes ejecutaran el test de forma simultánea en máquinas diferentes.

El objetivo de los test fue evaluar la usabilidad de los *frameworks*. Por este motivo, además de los entornos y de las máquinas virtuales, se crearon los proyectos base que utilizarían los participantes, con los *frameworks* ya instalados. De esta forma, el participante, a la hora de realizar el test, solo tenía que centrarse en el uso del *framework*.

Como en cada máquina los participantes tenían un tipo de *framework* diferente para trabajar, se creó un proyecto base de cada tipo que se utilizó de plantilla para los tres participantes. Estos proyectos se crearon con Visual Studio y fueron proyectos genéricos (solución en blanco), a los cuales se les añadió la carpeta con los archivos del *framework* necesarios. Los *frameworks* incluidos en este estudio se instalaron mediante el mismo procedimiento, que consistió en descargar un archivo comprimido del sitio web del fabricante y extraerlo en la carpeta del proyecto. Tras descomprimir los archivos, se pudo empezar a trabajar con el *framework* haciendo llamadas a las librerías y archivos en el código del proyecto.

2.3. Planificación del test

Una vez que el entorno estuvo preparado para la realización de los test, se procedió a planificar el desarrollo de los mismos. Se planificaron diferentes etapas que

los participantes tuvieron que completar, las cuales se organizaron de la siguiente manera:

- Presentación del test.
- Introducción a los *frameworks*.
- Desarrollo del tutorial de Backbone.
- Cuestionario posttest de Backbone.
- Desarrollo del tutorial de Angular.
- Cuestionario posttest de Angular.
- Desarrollo del tutorial de Ember.
- Cuestionario posttest de Ember.
- Cuestionario final.

2.3.1. Presentación e introducción del test a los participantes

En el inicio del test, a cada participante se le entregó por escrito una breve presentación del propósito del trabajo que iban a realizar. Esto se hizo con la finalidad de informarles y para resolver sus dudas acerca del objetivo o finalidad de los test. También para aclarar a dichos participantes que los errores o equivocaciones no debían preocuparles, ya que la evaluación era sobre los *frameworks* y no sobre sus conocimientos o habilidades. Junto con las instrucciones, se les entregó una breve introducción del tipo de *frameworks* que se iban a encontrar en el test. En esta página de información se aportaban algunos detalles para que estuvieran ya algo familiarizados con los conceptos e ideas más básicas antes de comenzar el primer test.

También se entregó una breve descripción escrita de cada *framework*. El objetivo fue que los participantes dispusieran de una breve introducción antes de comenzar un tutorial concreto que les permitiese tener unas nociones básicas del *framework* en cuestión. También se entregaron datos de acceso a las máquinas virtuales donde tendrían que hacer el test de usabilidad, junto con el enlace para descargar Team Viewer. Sobre la instalación de Team Viewer no se hizo ningún comentario, porque la instalación del programa era sencilla y básica para un participante del perfil seleccionado.

2.3.2. Videotutoriales

El test de los participantes se realizó con la ayuda de tres videotutoriales que se buscaron, probaron y verifi-

caron con la documentación oficial de cada *framework*. El motivo de usar esos videotutoriales en lugar de un videotutorial oficial se debió al exceso de tiempo que habría requerido seguir este último. En un videotutorial oficial se describe todo el *framework* de forma extensa y detallada, lo cual habría requerido mucho tiempo a la hora de filtrar qué partes podrían realizar los participantes y cuáles no. Hay que tener en cuenta que algunas características de estos *frameworks* son complejas y es necesario profundizar mucho en su conocimiento para poder manejarlas bien. Aunque en este trabajo se han descartado los videotutoriales oficiales para controlar la duración de los test de usabilidad, es interesante tener en cuenta este punto para posibles test de usabilidad más extensos o que abarquen todas las características disponibles en un *framework*.

Los videotutoriales tuvieron la siguiente duración:

- Backbone (2 h 30 min).
- Ember (2 h 30 min).
- Angular (2 h 40 min).

Se estimó que cada participante debía terminar en unas 3 horas y 30 minutos como máximo, dependiendo de la habilidad y del entusiasmo que el *framework* provocara en él.

Cada tutorial se analizó previamente y fue probado de forma individual por parte del organizador. También se verificó, en la medida de lo posible, que los contenidos se ajustaran con los ofrecidos en el sitio oficial de cada *framework*. Fue necesario garantizar que dichos contenidos eran verdaderos y no contenían errores graves ni fallos. También se probaron los videotutoriales para analizar la duración y los posibles problemas o erratas que pudieran contener.

2.3.3. Cuestionarios posttest de cada *framework* y final

Al concluir cada videotutorial, los participantes completaron un breve cuestionario acerca del *framework* (véase apéndice 1). Las preguntas fueron las mismas para los tres *frameworks*. El objetivo de dicho cuestionario fue poder ver la opinión de los participantes nada más concluir el test de cada tutorial. Con este cuestionario se realizó posteriormente parte del análisis individual de cada *framework* y de los resultados obtenidos.

Una vez terminados todos los test por parte de cada participante, estos tuvieron que rellenar otro cuestionario general de 19 preguntas acerca de los tres *frameworks* (véase apéndice 2). Esto permitiría después, junto con las anotaciones del entrevistador y los cuestionarios de cada *framework*, poder analizar los resultados de los tres *frameworks* y extraer conclusiones. En dicho cuestionario posttest, se buscaba ampliar y recoger información acerca de las características seleccionadas para el estudio, que se exponen a continuación:

- Sintaxis y organización del *framework*.
- Estructura y división de la lógica de negocio.
- Facilidad en el uso de URL amigables.
- Facilidad para propagar los cambios de una variable a otra y viceversa.
- Utilidades extras para manejar/diseñar códigos HTML.
- Soporte por parte de la comunidad e información oficial disponible.

2.3.4. Realización y ejecución de los test

Los test se realizaron en el mismo orden por parte de los tres participantes. Es decir, todos los participantes hicieron primero el test de Backbone, después continuaron con el test de Angular y, finalmente, terminaron con el de Ember. Cada participante indicó su disponibilidad horaria y el tiempo estimado de dedicación para poder organizar las sesiones, los turnos y las máquinas virtuales.

El ritmo de cada participante fue diferente y dependió de la disponibilidad horaria y de las circunstancias personales. No obstante, los tiempos totales invertidos por los participantes fueron similares en los tres *frameworks*. La única excepción fue el primer participante, que tuvo más problemas por erratas no detectadas y problemas de configuración ajenos al tutorial. Estos problemas se corrigieron para los siguientes participantes.

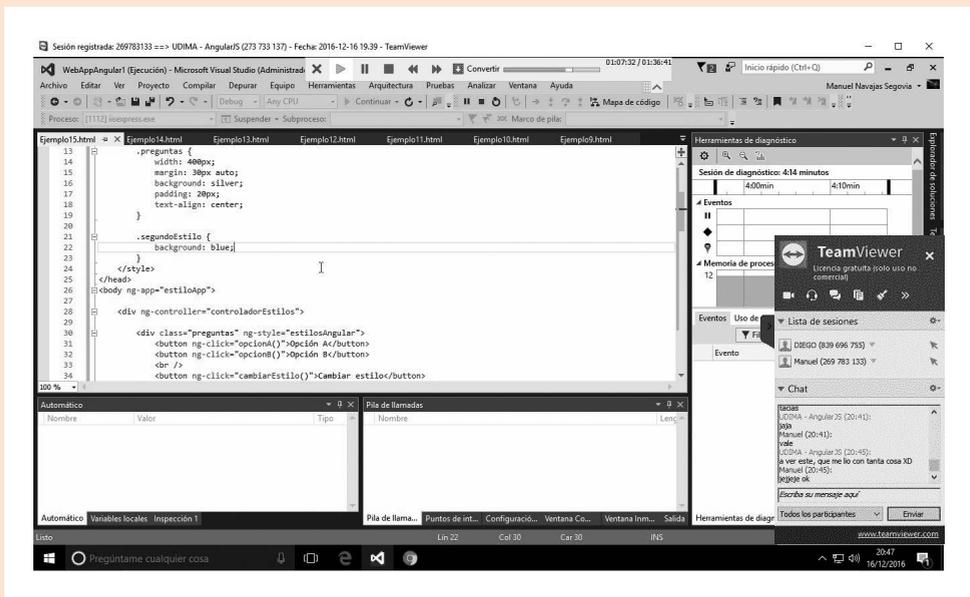
El desarrollo del test se realizó de la siguiente manera con cada sesión. El organizador se conectaba a la máquina virtual mediante Team Viewer. A continuación, se dejaba abierto el proyecto concreto del participante en el IDE para realizar los test. Por último, se revisaba la conexión y la configuración del proyecto en la máquina virtual para comprobar que funcionaba todo correctamente. A continuación, se le

comunicaba al participante que podía proceder a conectarse. Durante la sesión, el participante de turno se comunicaba con el organizador mediante el chat integrado con la herramienta Team Viewer. De esta forma, el participante podía comunicarse con el responsable en todo momento y en directo, para cualquier circunstancia y problema que se le plantease durante la realización del test. En el chat visible de la figura 2 se pueden identificar los dos participantes conectados en un momento concreto.

El organizador utilizó el chat para ayudar al participante y para realizar consultas acerca de la opinión que este tenía sobre los *frameworks*. También, para ir recabando toda la información posible junto con las anotaciones, las cuales permitirían realizar un análisis de resultados posterior mucho más completo.

Una vez finalizado cada uno de los test, el participante tuvo que contestar el cuestionario correspondiente. Concluidos los tres test, el participante cumplimentó un cuestionario general.

Figura 2. Team Viewer (sesión con ordenador remoto)



Fuente: elaboración propia.

2.3.5. Problemas técnicos surgidos durante los test

No fueron muchos los problemas técnicos que aparecieron durante la realización de los test. Los más importantes estuvieron relacionados con la configuración de los proyectos de prueba en Visual Studio (librerías, errores del IDE, etc.). En algún momento llegaron a suponer un retraso considerable, ya que algunos problemas resultaron complejos incluso para

el organizador. Los principales problemas que ocurrieron durante la ejecución de los test fueron los que se enumeran a continuación:

- Desconexión de la máquina virtual pasadas dos horas.
- Pérdida de permisos de escritura en las carpetas de proyecto.
- Erratas en los videotutoriales.
- Problemas de versionado en los *frameworks*.

2.3.6. Duración de los test

Según la planificación inicial, el tiempo de realización de los test por cada participante se estimó en unas 3 horas y 30 minutos, pero esta medición fue un tanto optimista. El motivo principal de dicha estimación se basó en que durante el ensayo de los test (realizado previamente a la ejecución de los participantes), el autor y organizador de este trabajo dispuso de dos monitores. Este detalle en concreto resultó ser de gran utilidad, ya que facilitó mucho la tarea para seguir los vídeos a la vez que se escribía el código en el IDE. Además, evitó tener que cambiar de aplicación constantemente, así como los despistes y las pérdidas de tiempo resultantes de este constante cambio de ventanas de aplicaciones diferentes. Pero en caso de no disponer de estas dos pantallas físicas, como así ocurrió con los participantes, penalizaba en el tiempo empleado debido a la necesidad constante de cambio entre una ventana y otra de las aplicaciones.

2.4. Datos recogidos durante la realización de los test

Los datos que se recogieron durante la realización de los test fueron los siguientes:

- Anotaciones realizadas por el organizador durante la realización de los test de cada participante.
- Registro del chat de cada una de las sesiones de los test de usabilidad.
- Cuestionarios cumplimentados por cada participante. En particular, cada participante cumplimentó cuatro cuestionarios, uno por cada uno de los *frameworks* estudiados y un último cuestionario general.

3. RESULTADOS

A continuación se presentan las respuestas a las preguntas de investigación del estudio, obtenidas a partir del análisis de los datos recogidos en los test de usabilidad.

3.1. Resultados del test de usabilidad

En este apartado se expone el resultado de los análisis de los cuestionarios y de las observaciones realizadas durante el desarrollo de cada test para cada participante en cada *framework*. Además, se muestran gráficos en los que se pueden visualizar los resultados.

Los gráficos presentados (véanse figuras 3, 4 y 5) están realizados de acuerdo con el análisis de las respuestas de los cuestionarios de los participantes del test. La construcción de estos gráficos, que se presentan por cada *framework*, se ha realizado tomando las respuestas y acotándolas en dos posibles valores (afirmativo o negativo). Es decir, por ejemplo, si una pregunta tenía una respuesta positiva, indicando que el participante estaba de acuerdo con el aspecto preguntado, en el gráfico se ha interpretado como un participante a favor de esta característica y se ha sumado una unidad a esa barra.

Cada barra de cada gráfico representa una característica estudiada y analizada en los test. De esta forma, y pese a que el estudio no es de tipo cuantitativo, se han creado gráficos que permiten sintetizar los resultados del test de ese *framework*. Como se podrá observar, hay nueve características, una por cada pregunta del cuestionario individual.

3.1.1. Resultados del test para Backbone

De los tres *frameworks* con los que se trabajó, este es el que resultó ser el menos satisfactorio para los participantes del test. En la figura 3 se puede observar el resultado de forma global.

Pese a ser el primer *framework* que los participantes comenzaron a utilizar, no les resultó complicado en exceso, pero tampoco causó una impresión muy positiva a ninguno. Un posible motivo de este resultado fue que la sintaxis y las instrucciones de Backbone no resultaron intuitivas a los participantes.

Backbone es un *framework* bastante pequeño en comparación con Angular y Ember. Por ello, en los otros dos test se seleccionaron funcionalidades comunes a los tres *frameworks*. De esta forma, se podían emplear tiempos similares en los test. No obstante, los tiempos empleados en los test de Backbone fueron algo elevados con respecto a los otros dos *frameworks*, pese a su supuesta simplicidad y pese a no disponer de tantas características adicionales como Angular o Ember.

Del análisis de las observaciones que tuvieron lugar durante la ejecución de los test, cabe destacar que los participantes no encontraron grandes dificultades en el desarrollo de los tutoriales. Aun así, se registraron algunas cuestiones interesantes que se comentan a continuación:

- **Primer participante.** Es el que más tiempo empleó para completar el test 1. En total tardó 4 horas. Esta

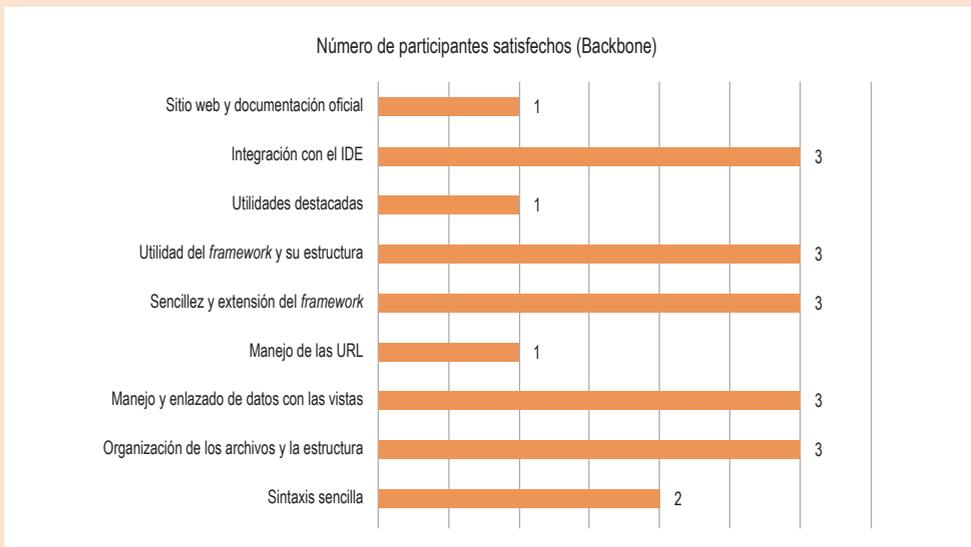
duración se pudo deber, principalmente, a que, al ser el primer participante, encontró alguna dificultad en el test no prevista por parte del organizador. También, se observó, durante la ejecución, que el participante iba muy despacio. El principal motivo fue la aparición de errores de JavaScript y la falta de utilidad de la consola del navegador para resolverlos. Estos errores provocaron que el participante interrumpiese el test en algunos momentos y que necesitase ayuda en varias ocasiones para continuar. Por último, en cuanto a la web, el participante indicó que esta le parecía completa de información acerca del *framework*, pero de difícil manejo para buscar una ayuda concreta.

- **Segundo participante.** Fue el que más rápido realizó este test (lo completó en 2 h 23 min) y no tuvo grandes dificultades. No obstante, detuvo su test varias veces durante algunos minutos por errores de sintaxis. Los errores de JavaScript, y el tratamiento de errores que realiza Backbone, de nuevo no ayudaron nada al participante. En cuanto a la web oficial del *framework*, el participante comentó que su diseño y presentación eran pobres, pero la web era completa en contenido e información. El participante, una vez finalizado el tutorial, indicó que el *framework* le gustó

bastante. En cuanto al cuestionario, el *framework* le resultó sencillo y fácil de utilizar, pese a las dificultades iniciales. Se deduce que le gustó y que le resultó sencillo, pero la impresión general, analizando las respuestas del cuestionario y las anotaciones durante la sesión del test, es que no le pareció lo suficientemente interesante como para profundizar en su conocimiento o seguir utilizándolo en un futuro.

- **Tercer participante.** También empleó bastante tiempo en finalizar el test de usabilidad de este *framework* (3 h 30 min). De los tres participantes, este fue el que necesitó más tiempo para empezar a entender en qué consistía el *framework* y qué tenía que hacer, llegando a perderse en diferentes momentos del comienzo. En el cuestionario, el participante indicó que le pareció ligeramente complicada la sintaxis y que necesitó familiarizarse más con el lenguaje. Este participante se detuvo con frecuencia por los errores de JavaScript de la consola del navegador. En este caso concreto, le retrasó bastante, y el *framework* no le aportó ninguna ayuda con los errores. El mensaje de error no fue muy indicativo; de ahí surgieron los retrasos y las pérdidas de tiempo considerables durante la ejecución del test.

Figura 3. Resultados para Backbone



Fuente: elaboración propia.

3.1.2. Resultados del test para Angular

En este apartado se analizan los resultados obtenidos en el test de usabilidad para los tres participantes con el *framework* de Angular, que ha sido el *framework* mejor valorado en este estudio y el que mayor satisfacción ha causado a los participantes.

Este *framework* ha sido el que más interés ha suscitado entre los participantes y el que ha generado menos dificultades y problemas. También su web resultó ser mucho más del gusto de los participantes que la de Backbone. Además, en este *framework* se dieron problemas con los mensajes de error por la consola, aunque en menor número e incidencia. El resultado se puede observar de forma global en la figura 4.

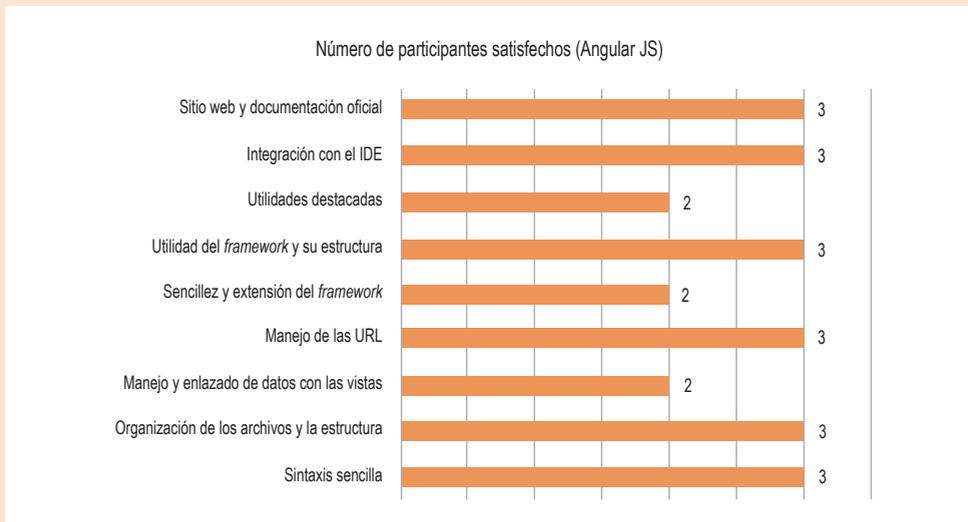
A continuación, se expondrán los resultados y los puntos destacados del test de cada participante de forma individual:

- **Primer participante.** Utilizó en total 4 horas para completar el tutorial y se mostró más interesado y activo que en el test anterior. Desde el comienzo del test, con los vídeos de Angular, el participante observó diferencias sustanciales en comparación con el primer *framework*. Además, indicó que le

estaba gustando mucho más. No obstante, también se detuvo en algunos momentos, como en la parte referente al controlador, donde necesitó alguna ayuda para poder continuar. Tras indicarle cómo funcionaban los errores de la consola con Angular en el navegador, continuó sin mayores problemas. Entre los aspectos que claramente le causaron una impresión positiva destacaron los eventos y la gestión y el tratamiento de los modelos con SCOPE. Por último, el participante indicó que la web le pareció mucho mejor que la de Backbone, más completa y útil.

- **Segundo participante.** Realizó el test de Angular con menos problemas. No obstante, fue el participante que más tiempo empleó (en total 4 h 30 min). De este participante hay que destacar que ejecutó el test con bastante ánimo y mayor interés que el de Backbone. Durante el test indicó varias veces que le estaba agradando mucho y se mostró bastante más activo que en el primer test. Esto indica que el *framework* le resultó mucho más interesante y atractivo para trabajar en un proyecto. Este hecho también se observó en los datos del cuestionario. El participante se mostró bastante satisfecho con el *framework* y lo encontró más completo que

Figura 4. Resultados para Angular



Fuente: elaboración propia.

Backbone. En el cuestionario comentó que «simplificaba muchas tareas complejas» con una buena organización. Le gustó el concepto de agrupar las instrucciones con «ng-» y el uso de las llaves ({{}}) para los *bindings*, junto con la variable «\$scope». En cuanto al sitio web, el participante se mostró más satisfecho con esta página. Le resultó muy bien organizada y completa.

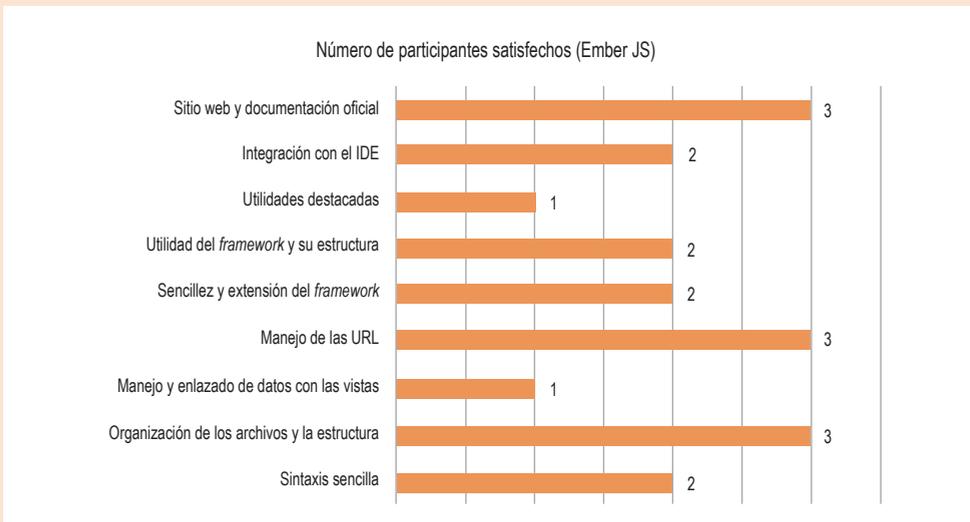
- **Tercer participante.** Realizó el test de Angular y lo completó en menos tiempo que los otros dos participantes (3 h 30 min). Con este participante costó un poco obtener opiniones durante el test, pero se mostró mucho más satisfecho que con Backbone, teniendo apenas pequeños problemas de sintaxis que consiguió resolver rápidamente. El mismo participante indicó varias veces que estaba teniendo menos problemas en comparación con Backbone. De hecho, el participante no tuvo problemas en el tema de las *cascading style sheets* (CSS), al contrario que los otros dos. Tampoco con las rutas, que fue la parte más compleja. En cuanto a la web, al igual que los otros dos participantes, opinó que tenía mucha información, mejor organizada y más manejable.

3.1.3. Resultados del test para Ember

El último *framework* sobre el que se realizó el test de usabilidad fue Ember. Este *framework* fue mejor valorado por los participantes en comparación con Backbone, pero no llegó a ser tan satisfactorio como Angular. Sucedió lo mismo con los problemas detectados; estos fueron algo más complejos para los participantes que los encontrados en el test de Angular. Se observaron los problemas comunes que habían presentado los otros *frameworks* con respecto a los mensajes de error por la consola. También hay que destacar que los tres participantes tuvieron problemas con la librería «Ember Data». Este test destacó por lo poco intuitiva que resultó ser la contrabarra (Ember utiliza el símbolo «/» para cerrar instrucciones de formar similar a HTML), generando mucha confusión en los participantes.

Por último, en cuanto a la web, resultó ser mucho más del gusto de los participantes que la de Backbone, pero no llegó a convencer tanto como la de Angular, excepto a un usuario, a quien le gustó más la de Ember. La satisfacción con este *framework* fue inferior a la de Angular. El resultado se puede observar de forma global en la figura 5.

Figura 5. Resultados para Ember



Fuente: elaboración propia.

A continuación, explicamos brevemente los resultados obtenidos con cada participante:

- **Primer participante.** Fue el que encontró más problemas durante la ejecución del test. En comparación con los otros dos participantes, tardó 3 horas y 30 minutos en completar el test. Algunos problemas que encontró fueron la falta de la librería «Ember Data», que se pretendía dar en el proyecto de partida. Otros fueron fruto de errores en sintaxis o errores de compresión de la librería ya mencionada antes (Ember Data), lo cual le ocasionó muchos problemas. También se encontró con los típicos errores de sintaxis que se mostraban por la consola del navegador. En cuanto al cuestionario, el participante dejó constancia de que le parecía más compleja la sintaxis que la de Angular. La organización del proyecto y de los archivos no resultó ser especialmente buena para el participante, aunque sí le pareció sencilla la forma de enlazar los datos. Por último, en relación a la web, indicó que le resultó más complicada que la de Angular. En concreto no le gustó la forma en que se gestionaban las versiones de *framework* y sus archivos y librerías.
- **Segundo participante.** Resolvió el test de Ember en un tiempo bastante bueno (3 h 15 min) y lo completó sin muchas dificultades. Algún problema destacable fue el complemento de «Ember Data» con el que el participante sufrió algún retraso, al igual que algunos errores de sintaxis. El test resultó ser de su agrado, aunque no llegó al nivel de satisfacción de Angular. Un comentario interesante que hizo el participante fue el hecho de no poder acceder a las propiedades de los elementos en la consola como en los *frameworks* anteriores. Y en cuanto al cuestionario hay que destacar que valoró de forma positiva el *framework*, pero no llegó al nivel de satisfacción de Angular. Lo más interesante para este participante fueron la gestión del modelo y el uso de plantillas «Handlebars». En cuanto a la web, la consideró muy elaborada y le causó una impresión muy positiva, en especial el ejemplo que usa para ir construyendo funcionalidades, aunque no especificó si le gustó más o menos que los otros *frameworks*.
- **Tercer participante.** Con el último participante del test de Ember, sucedió lo mismo que con el segundo. Este participante completó el test en 4 horas, sin muchas complicaciones. Las dificulta-

des más notorias que tuvo fueron referentes al manejo de las URL, donde nuevamente los errores que mostró la consola del navegador no fueron de gran ayuda para terminar la tarea. De hecho, el manejo de las URL no terminó de entenderlo bien. En este caso, el participante necesitó ayuda para completar este punto del tutorial. El participante tuvo también alguna dificultad con los controladores y con la librería «Ember Data». En cuanto al cuestionario de este *framework*, el participante no se mostró satisfecho con la sintaxis en comparación con los otros *frameworks*, aunque sí le agradó la organización de los archivos y el proyecto. El enlazado de datos tampoco le resultó fácil. Por último, indicó que la página web le pareció la más completa, refiriéndose al buen diseño y a la practicidad del sitio. Es la web que más le gustó de los tres *frameworks*.

3.1.4. Resumen de los resultados de los test de usabilidad

A continuación se sintetizan las conclusiones más importantes de los test de usabilidad realizados:

- **Backbone.** Este *framework* fue el que menos éxito obtuvo entre los participantes. Los tiempos empleados, los problemas detectados, así como los comentarios recogidos durante el desarrollo de los test, indican claramente que es el *framework* con peores resultados. Además, pese a ser el *framework* más ligero y pequeño, resultó difícil de entender a los participantes. Por último, la documentación de la web oficial tampoco fue satisfactoria para ninguno de los tres participantes en comparación con los otros dos *frameworks*. Esto hace que, junto con lo comentado anteriormente, Backbone haya sido el *framework* que menor facilidad de aprendizaje ha tenido en este estudio, en comparación con Angular y Ember.
- **Angular.** Este *framework* es el que más satisfacción aportó a los participantes en todos los aspectos. Los tiempos empleados, y el número de problemas detectados, así como los comentarios durante el desarrollo de este test, indican claramente que este *framework* fue el mejor valorado. También fue en el que se observó la mayor facilidad de aprendizaje. De los cuestionarios se pue-

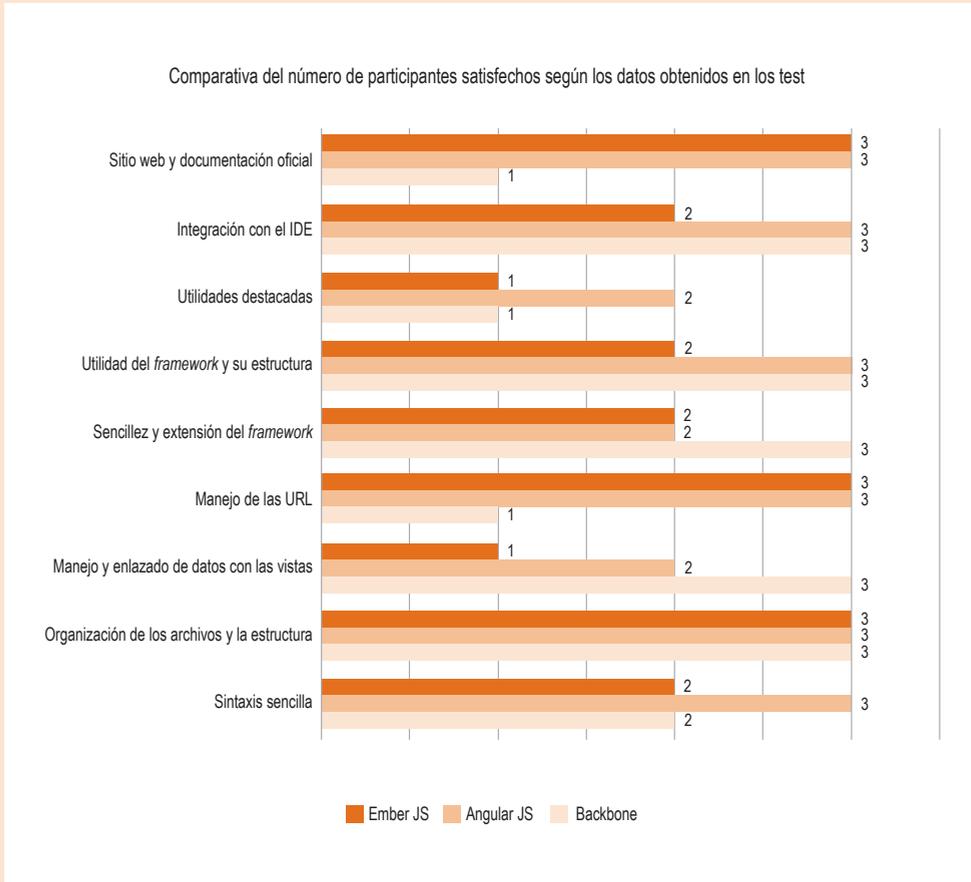
de extraer como conclusión que resultó ser el más interesante, tanto por sus características y funcionalidades como por sus sintaxis y organización. En cuanto a la documentación oficial del sitio web, los tres participantes coincidieron en que era muy completa y organizada, y mucho más manejable respecto a los otros *frameworks* analizados.

- **Ember.** Este es el segundo *framework* que más ha gustado a los participantes, aunque estos tuvieron algunos inconvenientes de consideración en la ejecución de los test. No obstante, las funcionalidades y características gustaron, y resultó el test más fácil de completar, comparado con Backbone. En ca-

racterísticas y funcionalidades fue algo más sencillo que Backbone, pero no llegó al grado de satisfacción de Angular. A pesar de ello, a los participantes del estudio les resultó agradable, con una facilidad de aprendizaje buena. También el sitio web y la información disponible en esta resultaron ser satisfactorios. Incluso uno de los participantes consideró la web de Ember como la mejor de los tres *frameworks* estudiados.

Para finalizar, en la figura 6 se puede observar un gráfico que recoge de forma global los resultados obtenidos en los cuestionarios individuales de cada *framework*.

Figura 6. Comparativa de los resultados individuales de cada *framework*



Fuente: elaboración propia.

3.1.5. Conclusiones comunes observadas en los tres frameworks

De los test realizados, se detectaron una serie de características referentes a la usabilidad que son comunes para los tres *frameworks*:

- La gestión de errores por consola en los tres *frameworks* no fue muy satisfactoria. Para los participantes, en la mayoría de los casos no resultó ser muy útil a la hora de encontrar una solución. Esto provocó que interrumpiesen el test en algunos momentos y que necesitaran ayuda en varias ocasiones para continuar. Es un problema que se ha observado en los tres participantes. Está más relacionado con JavaScript y su gestión de errores, pero los *frameworks* pueden ayudar a controlar este punto y hacerlo más sencillo y útil para los desarrolladores. Angular y Ember tienen ya características desarrolladas al respecto, pero se pueden mejorar más. Por tanto, de acuerdo con el resultado de este estudio, es necesario prestar más atención a la gestión de errores por consola en los *frameworks* JavaScript.
- Los tres participantes se mostraron de acuerdo con que la integración en la herramienta de desarrollo (en este caso Visual Studio) fuera satisfactoria en los tres casos. No hubo mucha discrepancia en este punto con los *frameworks* estudiados. Ember es el que resultó algo menos satisfactorio en este punto.
- La sintaxis de los tres *frameworks* también fue sencilla, sin que supusiera un esfuerzo muy grande de un *framework* respecto a los demás para entenderlo, o más compleja respecto a los otros dos casos analizados. Backbone resultó ser el que tuvo una peor valoración en este aspecto, pero sin llegar a ser demasiado problemático.

3.2. Evaluación de la facilidad de aprendizaje, de la tasa de errores y de la satisfacción

3.2.1. Facilidad de aprendizaje

Este atributo indica cómo de fácil es aprender la funcionalidad básica de un sistema (Ferré, 2015). En este trabajo, la facilidad de aprendizaje incluía los puntos analizados en el apartado de metodología. Estos puntos son:

- Análisis de la sintaxis y organización del *framework*.
- Análisis de la estructura y división de la lógica de negocio del diseño en la aplicación.
- Facilidad en el uso de las URL amigables.
- Facilidad para propagar los cambios de una variable a otra y viceversa.
- Utilidades extras para manejar y diseñar código HTML (directivas).
- Soporte por parte de la comunidad e información disponible para los desarrolladores.

Con el análisis de estos puntos se puede concluir que el *framework* que mejor facilidad de aprendizaje demostró tener fue Angular. El *framework* que ocupa el segundo lugar no está tan claro, ya que tanto Backbone como Ember parecían estar en este aspecto igualados, y quizá Ember podría estar por delante de Backbone en este punto.

3.2.2. Tasa de errores

Este atributo se refiere al número de errores cometidos por el participante mientras realizaba una tarea (Ferré, 2015). El estudio de este atributo también se trató en la mayoría de los puntos anteriores, en particular:

- Análisis de la sintaxis y organización del *framework*.
- Análisis de la estructura y división de la lógica de negocio del diseño en la aplicación.
- Facilidad en el uso de las URL amigables.
- Facilidad para propagar los cambios de una variable a otra y viceversa.

En el análisis de estos cuatro puntos es donde se pueden encontrar la mayoría de los errores cometidos por los participantes al realizar los test. Angular es el *framework* con el que los participantes cometieron menos errores, tanto de sintaxis como de estructura general y otros conceptos. Nuevamente es difícil determinar qué *framework* ocupaba el segundo lugar en este aspecto. Posiblemente fue Backbone, ya que, si bien obtuvo peores resultados por la dificultad de sus características y conceptos, Ember tuvo mayor tasa de fallos en sintaxis y en el manejo de sus características. Se podría decir que Backbone fue más fuerte en los puntos débiles que Ember y viceversa.

3.2.3. Satisfacción

El último atributo que ha sido objeto de estudio en este trabajo se refiere a la impresión subjetiva que el participante obtuvo del sistema (Ferré, 2015). Los análisis realizados en los puntos anteriores adelantaron información de este aspecto y fueron una parte importante para medir la satisfacción. No obstante, el análisis en concreto de este aspecto se basó en los resultados obtenidos tras analizar los siguientes puntos:

- Visión general y preferencias de uso según los participantes.
- Soporte por parte de la comunidad e información disponible para los desarrolladores.

Una vez más los datos indican que Angular fue el que mejor satisfacción aportó a los participantes. Durante el desarrollo de los test, así como por los comentarios de los participantes y por las respuestas de los cuestionarios, este *framework* fue el mejor valorado. En segundo lugar, volvimos a tener esa disparidad de resultados entre Backbone y Ember, estando un poco igualados, pero Ember se podría decir que fue el segundo *framework* que más satisfacción aportó a los participantes según los datos analizados.

Observando el resultado de los atributos de usabilidad (facilidad de aprendizaje, tasa de errores y satisfacción) analizados en el estudio, se pudo concluir que el *framework* con mejor usabilidad de los tres estudiados fue Angular, que estuvo por encima de Backbone y Ember en casi todos los aspectos. Como se ha podido observar en el desarrollo del estudio, la usabilidad de un *framework* depende de muchos detalles, desde la sintaxis, hasta la documentación y la ayuda disponibles para los desarrolladores. Todos estos factores influyen y aportan facilidades que mejoran la usabilidad.

3.2.4. Conclusiones acerca de la facilidad de aprendizaje, de la tasa de errores y de la satisfacción

La usabilidad puede ser un factor que influya en la adopción de un *framework* JavaScript. En este estudio los tres participantes seleccionaron Angular como el mejor *framework* en diferentes aspectos de la usabilidad. Este hecho coincide con que Angular es un referente en la actualidad. Es difícil que un desarrollador



de aplicaciones web no haya escuchado hablar de este *framework*, que además dispone de una comunidad bastante amplia. En cambio, los otros dos *frameworks* son menos conocidos, aunque ambos tienen su comunidad y apoyos. Por ejemplo, cuando se contactó con los participantes para el estudio, los tres habían escuchado o leído algo acerca de Angular, uno de ellos tenía alguna noción de Backbone, pero ninguno de los participantes tenía ningún conocimiento de Ember. Estos datos parecen sugerir que la usabilidad tiene un papel importante en la adopción de un *framework* JavaScript y en su difusión en el mundo del desarrollo.

La usabilidad de un *framework* puede influir en el ánimo, en el interés y en la velocidad de trabajo de los desarrolladores, haciendo posible que un proyecto pueda ser desarrollado en mayor o menor tiempo y con mayor o menor éxito. Todos estos factores pueden influir en la calidad del código generado, así como en la estructura y en la organización del producto final.

Por último, y para concluir, no podemos olvidar la importancia que la usabilidad tiene para el coste en formación de los desarrolladores. Hay que tener en cuenta la repercusión tanto económica como de tiempo que puede tener para la formación de los empleados de una empresa u organización el hecho de elegir un *framework* con grado de usabilidad mayor o menor.

La usabilidad puede ser un factor que influya en la adopción de un *framework* JavaScript. En este estudio los tres participantes seleccionaron Angular como el mejor *framework* en diferentes aspectos de la usabilidad

4. CONCLUSIONES

En este trabajo se ha analizado la usabilidad de tres *frameworks* JavaScript, conocidos y utilizados en el mundo del desarrollo web, donde el que mejor resultado ha obtenido ha sido Angular, seguido de Ember y de Backbone en último lugar. Se ha mostrado que la usabilidad, un factor muchas veces infravalorado en el área de desarrollo, podría tener un papel importante en la adopción de un *framework* JavaScript.

Nuestro estudio contribuye al conocimiento acerca de la adopción de la tecnología por parte de los profesionales informáticos y es útil en particular para desarrolladores web.

Además, los resultados obtenidos pueden ayudar en la creación de nuevos *frameworks* adaptados a las necesidades de los desarrolladores.

5. BIBLIOGRAFÍA

Ferré Grau, X. (2015). *Interacción persona-ordenador*. Madrid: Ediciones CEF.

Gizas, A., Christodoulou, S. y Papatheodorou, T. (2012). Comparative evaluation of JavaScript frameworks. *Proceedings of the 21st International Conference on World Wide Web* (pp. 513-514). New York, EE. UU: ACM. doi: 10.1145/2187980.2188103.

Graziotin, D. y Abrahamsson, P. (2013). Making sense out of a jungle of JavaScript frameworks. (Vol. 7.983). En J. Heidrich, M. Oivo, A. Jedlitschka y M. T. Baldassarre (Eds.), *Product-Focused Software Process Improvement. PROFES 2013. Lecture Notes in Compu-*

ter Science. Springer, Berlin, Heidelberg.

Mariano, C. L. (2017). *Benchmarking JavaScript Frameworks. Masters dissertation*.

Pano, A., Graziotin, D. y Abrahamsson, P. (2018). Factors and actors leading to the adoption of a JavaScript framework. *Empirical Software Engineering Journal* (in press).

Scholten, T., Tromp, N. y Nijhuis, M. (2017). *JavaScript Frameworks*. Woodcliff Lake.

Shah, Z. H. (2013). *Windows Server 2012 Hyper-V: Deploying Hyper-V Enterprise Server Virtualization Platform*. Birmingham: Packt Publishing.

Smith, J. E. y Nair, R. (2005). *Virtual Machines*. San Francisco: Elsevier.

APÉNDICES

1. Cuestionario individual para cada *framework*

1. ¿Qué te ha parecido el <i>framework</i> ? ¿Su sintaxis te ha resultado sencilla?
2. ¿Crees que la organización de los archivos y la estructura del proyecto es buena?
3. ¿El manejo y el enlazado de los datos a la vista te ha parecido sencillo o complejo?
4. ¿Qué tal te ha resultado el manejo de las URL? ¿Lo considerarías difícil de gestionar?
5. ¿Te parece que es un <i>framework</i> complejo y extenso o es sencillo y ligero?
6. ¿Ves útil este <i>framework</i> y la estructura que propone?
7. ¿Has visto alguna utilidad/característica que te haya llamado especialmente la atención?
.../...
.../...
8. ¿Te ha gustado la integración de la herramienta con el entorno de desarrollo?
9. ¿Qué te ha parecido el sitio web oficial del <i>framework</i> ? ¿Lo encuentras amigable y útil o crees que presenta algún problema?
10. ¿Quieres comentar alguna cuestión más sobre el <i>framework</i> ?

2. Cuestionario general sobre los tres *frameworks*

1. ¿Cuál de los <i>frameworks</i> vistos te ha gustado más y por qué?
2. ¿Cuál de los tres <i>frameworks</i> te ha parecido más completo?
3. ¿Cuál de los tres <i>frameworks</i> te ha parecido más fácil de entender o manejar?
4. ¿Cuál de los tres <i>frameworks</i> te ha resultado más útil?
5. ¿Lo utilizarías en tus proyectos laborales y personales si te dieran la oportunidad?
6. ¿Cuál de los tres <i>frameworks</i> te ha resultado más complicado de entender y por qué? ¿Qué es lo que no te ha gustado más comparándolo con los otros dos <i>frameworks</i> ?
7. ¿Qué <i>framework</i> crees que tiene el proceso de instalación más sencillo?
8. ¿Cuál de los tres <i>frameworks</i> crees que aporta más a un proyecto profesional teniendo en cuenta las características vistas en los tutoriales que has realizado?
9. ¿Cuál de los tres <i>frameworks</i> te parece que aporta funcionalidades más interesantes?
10. En cuanto al código, ¿qué <i>framework</i> te ha parecido que permite tener un código mejor estructurado y organizado?

Magisterio de Educación Infantil

Hoy en día los centros educativos tienen una imperiosa necesidad de disponer de personal especializado, capaz de hacer frente a las necesidades educativas de la etapa infantil, de acuerdo con los conocimientos ya logrados por las diversas ciencias que hoy se ocupan de los niños en los primeros años de su vida, así como de los logros relativos al desarrollo de la inteligencia, la emocionalidad y la formación de la personalidad temprana, resultantes de estudios recientes sobre el desenvolvimiento de la mente infantil.

PLAN DE ESTUDIOS (la obtención del grado conlleva la realización de 240 créditos)

CURSO	ASIGNATURAS	TIPO	CRÉD.
1	Didáctica e Innovación Curricular en Educación Infantil	T	6
1	Psicología de la Educación	T	6
1	Historia y Teoría de la Educación	T	6
1	Tecnologías de la Información y Gestión del Conocimiento	T	6
1	Psicología del Desarrollo	T	6
1	Desarrollo de Habilidades Lingüísticas y Lectoescritoras	B	6
1	Psicomotricidad Infantil	B	6
1	Sociología de la Familia y de la Infancia	T	6
1	Organización y Gestión del Aula	T	6
1	Métodos, Recursos y Nuevas Tecnologías para el Aprendizaje ..	T	6
2	Conocimiento del Medio Social y Cultural y su Didáctica	B	6
2	Conocimiento del Medio Natural y su Didáctica	B	6
2	Lengua Extranjera para Maestros: Inglés	B	6
2	Psicología del Aprendizaje	T	6
2	Estimulación y Atención Temprana	T	6
2	Sociología de la Educación	T	6
2	Expresión Musical y su Didáctica	B	6
2	Desarrollo del Pensamiento Lógico Matemático y su Didáctica ..	B	6
2	PRÁCTICAS EXTERNAS I	B	12
3	Lengua y Literatura y su Didáctica	B	6
3	Atención a la Diversidad en Educación Infantil	T	6
3	Didáctica de la Lengua Inglesa en Educación Infantil	B	6
3	Metodología de Investigación en Educación	T	6
3	Optativa 1 (*)	O	6
3	Organización y Liderazgo de Centros Escolares	T	6
3	Orientación e Intervención Tutorial	T	6
3	Optativa 2 (*)	O	6
3	PRÁCTICAS EXTERNAS II	B	12
4	El Juego en Educación Infantil	T	6
4	Expresión Plástica y Visual y su Didáctica	B	6
4	Optativa 3 (*)	O	6
4	Optativa 4 (*)	O	6
4	Optativa 5 (*)	O	6
4	Salud, Infancia y Alimentación	T	4
4	PRÁCTICAS EXTERNAS III	B	12
4	TRABAJO FIN DE GRADO	B	14

Magisterio de Educación Primaria

Son objetivos de la Educación Primaria, entre otros: conocer y apreciar los valores y las normas de convivencia, aprender a obrar de acuerdo con ellas, prepararse para el ejercicio activo de la ciudadanía y respetar los derechos humanos, así como el pluralismo propio de una sociedad democrática. También, desarrollar hábitos de trabajo individual y de equipo, de esfuerzo y responsabilidad en el estudio, así como actividades de confianza en uno mismo, sentido crítico, iniciativa personal, curiosidad, interés y creatividad en el aprendizaje.

PLAN DE ESTUDIOS (la obtención del grado conlleva la realización de 240 créditos)

CURSO	ASIGNATURAS	TIPO	CRÉD.
1	Didáctica e Innovación Curricular en Educación Primaria	T	6
1	Psicología de la Educación	T	6
1	Historia y Teoría de la Educación	T	6
1	Tecnologías de la Información y Gestión del Conocimiento	B	6
1	Psicología del Desarrollo	T	6
1	Lengua Española	B	6
1	Fundamentos de Matemáticas	B	6
1	Conocimiento del Medio Natural	B	6
1	Sociología de la Educación	T	6
1	Métodos, Recursos y Nuevas Tecnologías para el Aprendizaje ..	B	6
2	Atención a la Diversidad en Educación Primaria	T	6
2	Educación Física y su Didáctica	B	6
2	Lengua Extranjera para Maestros: Inglés	B	6
2	Educación Musical y su Didáctica	B	6
2	Metodología de Investigación en Educación	T	6
2	Organización y Liderazgo de Centros Escolares	T	6
2	Orientación e Intervención Tutorial	T	6
2	Psicología del Aprendizaje	T	6
2	PRÁCTICAS EXTERNAS I	B	12
3	Literatura Infantil y Juvenil	B	6
3	Didáctica de las Matemáticas	B	6
3	Didáctica de la Lengua Inglesa en Educación Primaria	B	6
3	Conocimiento del Medio Social y Cultural	B	6
3	Optativa 1 (*)	O	6
3	Didáctica de las Ciencias Sociales	B	6
3	Educación Plástica y Visual y su Didáctica	B	6
3	Optativa 2 (*)	O	6
3	PRÁCTICAS EXTERNAS II	B	12
4	Didáctica de las Ciencias Experimentales	B	6
4	Didáctica de la Lengua y la Literatura	B	6
4	Optativa 3 (*)	O	6
4	Optativa 4 (*)	O	6
4	Optativa 5 (*)	O	6
4	Educación en Valores	B	4
4	PRÁCTICAS EXTERNAS III	B	12
4	TRABAJO FIN DE GRADO	B	14

(*) La lista de asignaturas optativas se puede consultar en www.udima.es. T = Formación básica; B = Formación obligatoria; O = Asignatura optativa

Menciones en los grados de Magisterio



Menciones en los grados de Magisterio de Educación Infantil y Primaria

Los grados en Magisterio de Educación Infantil y Primaria tienen cinco menciones. Cada una de ellas se compone de 30 créditos ECTS, pudiendo los estudiantes optar por una de las menciones para poder finalizar el grado o bien obtener el título sin mención cursando 30 créditos optativos a su libre elección de los ofertados. También podrían cursar más de una mención. Si el alumno deseara obtener varias menciones, deberá cursar los créditos asociados a cada una de ellas.

Mención en Lengua Inglesa	
Asignatura	ECTS
Lengua Inglesa I	6
Lengua Inglesa II	6
Lengua Inglesa III	6
Educación para el Bilingüismo: CLIL	6
Didáctica Avanzada de la Lengua Inglesa	6

Mención en Pedagogía Terapéutica	
Asignatura	ECTS
Principios y Estrategias de la Educación Inclusiva	6
Evaluación y Orientación Psicopedagógica	6
Necesidades Específicas de Apoyo Educativo y Orientación Familiar	6
Intervención Psicopedagógica en Dificultades de Aprendizaje	6
Programas de Intervención para Alumnos con Necesidades Específicas de Apoyo Educativo	6

Mención en Audición y Lenguaje	
Asignatura	ECTS
Principios y Estrategias de la Educación Inclusiva	6
Trastornos del Habla y del Lenguaje	6
Sistemas Alternativos y Aumentativos de Comunicación.....	6
Evaluación e Intervención en Lenguaje Oral.....	6
Evaluación e Intervención en Lenguaje Escrito	6

Mención en Tecnología Educativa	
Asignatura	ECTS
Recursos Tecnológicos e Innovación Docente.....	6
Desarrollo Tecnológico y Educación Intercultural	6
Diseño de Materiales Didácticos con TIC	6
Integración de las TIC en la Enseñanza de las Artes y las Humanidades	6
Integración de las TIC en la Enseñanza de las Ciencias y las Matemáticas	6

Mención en Enseñanza de la Religión Católica*	
Asignatura	ECTS
Historia de la Iglesia	6
Religión, Cultura y Valores	6
El Mensaje Cristiano	6
La Iglesia, los Sacramentos y la Moral	6
Pedagogía y Didáctica de la Religión Católica	6

* Cubre los requisitos de formación universitaria para poder solicitar la DECA a la Conferencia Episcopal (ver plan de estudios).

Curso de adaptación al grado

Este curso de adaptación al grado ofrece a los maestros diplomados en la Especialidad de Educación Infantil o Primaria la posibilidad de obtener formación en campos determinados dentro del ejercicio profesional docente en estas etapas, a través de las menciones cualificadoras mencionadas anteriormente.

El objetivo principal del plan de estudios de este curso de adaptación al grado es contribuir a la actualización de la formación de los maestros diplomados. La aplicación de las TIC a la educación y de líneas pedagógicas innovadoras fruto de la investigación en educación hacen necesaria la actualización de los conocimientos didácticos de los diplomados y la formación de los maestros en investigación e innovación.

Al finalizar el curso de adaptación se obtiene el título de grado en Magisterio de Educación Infantil o de Educación Primaria.

La docencia en la etapa de educación infantil o primaria es una profesión regulada. Los graduados en Magisterio de Educación Infantil o Primaria tienen como principal salida profesional el trabajo como profesores en estas etapas, tanto en centros públicos como concertados y privados.

Si bien otras salidas profesionales para estos títulos pueden ser:

- Participación en proyectos educativos de organismos e instituciones (centros culturales, museos, asociaciones, ONG, etc.).
- Centros de educación para adultos.
- Centros de ocio y tiempo libre.
- Participación en programas de extensión educativa (actividades extraescolares, actividades de apoyo, etc.).
- Diseño y elaboración de materiales didácticos.
- Participación en proyectos de atención a la infancia y familiar.

Psicología (Rama CC. de la Salud)

Siguiendo el modelo científico-profesional de psicólogo (o *scientist-practitioner*), se trata de aportar a los alumnos los conocimientos científicos necesarios para comprender, interpretar, analizar y explicar el comportamiento humano, así como para evaluar e intervenir en el ámbito individual y social, con el fin de que los psicólogos y la psicología promuevan y mejoren la salud y la calidad de vida de las personas.

PLAN DE ESTUDIOS (la obtención del grado conlleva la realización de 240 créditos)

CURSO	ASIGNATURAS	TIPO	CRÉD.
1	Historia de la Psicología	T	6
1	Psicología del Aprendizaje	T	6
1	Bases Biológicas de la Conducta	T	6
1	Tecnologías de la Información y de la Comunicación	T	6
1	Sociología General	T	6
1	Psicología del Desarrollo I.....	T	6
1	Introducción a la Antropología	T	6
1	Métodos, Diseños y Técnicas de Investigación Psicológica	T	6
1	Psicología Social	T	6
1	Psicología de la Motivación y Emoción	T	6
2	Neurociencia	B	6
2	Psicología de la Percepción y la Atención.....	B	6
2	Estadística Descriptiva e Inferencial	B	6
2	Psicología del Desarrollo II	B	6
2	Psicología de la Personalidad y las Diferencias Individuales	B	6
2	Análisis de Datos y Diseños en Psicología	B	6
2	Psicología de la Memoria	B	6
2	Evaluación Psicológica	B	6
2	Psicología de los Grupos	B	6
2	Optativa 1 (*)	O	6
3	Psicopatología I	B	6
3	Intervención y Tratamiento Psicológico	B	6
3	Psicometría	B	6
3	Evaluación en Clínica y Salud	B	6
3	Optativa 2 (*)	O	6
3	Psicopatología II	B	6
3	Psicofisiología	B	6
3	Psicología del Pensamiento y del Lenguaje	B	6
3	Psicología Clínica	B	6
3	Optativa 3 (*)	O	6
4	Psicología de la Salud	B	6
4	Psicología de la Educación	B	6
4	Psicología del Trabajo y de las Organizaciones	B	6
4	Ética y Deontología Profesional	B	6
4	Optativa 4 (*)	O	6
4	Intervención Psicosocial	B	6
4	Optativa 5 (*)	O	6
4	PRÁCTICUM	B	9
4	TRABAJO FIN DE GRADO	B	9

Historia

Se conjugan los conocimientos humanísticos básicos y generalistas con el aprendizaje de las herramientas y técnicas de las nuevas TIC. Los estudiantes adquirirán la formación, los conocimientos y las habilidades necesarias para permitirles el pleno desarrollo de las funciones relacionadas con la investigación y la enseñanza de la historia, para que comprendan y hagan comprensibles a los demás los acontecimientos del pasado.

PLAN DE ESTUDIOS (la obtención del grado conlleva la realización de 240 créditos)

CURSO	ASIGNATURAS	TIPO	CRÉD.
1	Ciencia Histórica. Conceptos y Etapas de la Historia Universal ..	T	6
1	Geografía General	T	6
1	Historia Social y Política Contemporáneas	T	6
1	Tecnología y Gestión de la Información y del Conocimiento	T	6
1	Historia del Arte. Conceptos Fundamentales	T	6
1	Antropología General	T	6
1	España Actual	T	6
1	Historia Económica Española y Mundial	T	6
1	Historia de América	T	6
1	Prehistoria Universal	T	6
2	Prehistoria de la Península Ibérica	B	6
2	Arte Prehistórico	B	6
2	Historia del Mundo Grecoromano	B	6
2	Historia Medieval Universal	B	6
2	Historia de España Antigua	B	6
2	Historia de la Cultura Escrita	B	6
2	Historia Antigua del Próximo Oriente	B	6
2	Pensamiento Antiguo y Medieval	B	6
2	Historia de las Sociedades Peninsulares en la Edad Media	B	6
2	Optativa 1 (*)	O	6
3	Historia del Arte Antiguo y Medieval	B	6
3	Historia Cultural de la Edad Media	B	6
3	Historia Universal Moderna	B	6
3	Historia de España Moderna	B	6
3	Optativa 2 (*)	O	6
3	Pensamiento Moderno	B	6
3	Historia Universal Contemporánea	B	6
3	Fundamentos de Arqueología	B	6
3	Paleografía y Diplomática. Epigrafía y Numismática	B	6
3	Optativa 3 (*)	O	6
4	Historia Política y Social de la Edad Moderna	B	6
4	Historia Contemporánea de España. El Siglo XIX	B	6
4	Pensamiento Contemporáneo.....	B	6
4	Historia del Arte Moderno y Contemporáneo	B	6
4	Optativa 4 (*)	O	6
4	Historia Cultural de la Edad Moderna	B	6
4	El Mundo Actual	B	6
4	Métodos y Técnicas de Investigación Histórica. Tendencias Historiográficas Actuales	B	6
4	TRABAJO FIN DE GRADO	B	12

(*) La lista de asignaturas optativas se puede consultar en www.udima.es. T = Formación básica; B = Formación obligatoria; O = Asignatura optativa