
TFG: Aplicación Foto-clic

Área: Desarrollo de aplicaciones interactivas

Titulación: Grado Multimedia



Autor: Alberto Marín Mendoza

Consultor: Kenneth Capseta Nieto

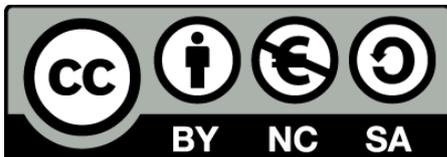
Profesor: Carlos Casado Martínez

Fecha: 25/09/2017

Copyright

Licencia del Proyecto TFG:

- Creative Commons BY-SA-NC
- <https://creativecommons.org/licenses/by-nc-sa/3.0/es/>



Licencia de uso de Spectrum JQuery ColorPicker

- Copyright (c) Brian Grinstead
- Se concede permiso, de forma gratuita, a cualquier persona que obtenga una copia de este software y los archivos de documentación asociados (del "Software"), para tratar el Software sin restricciones, incluyendo COMERCIALIZACIÓN, APTITUD PARA UN PROPÓSITO PARTICULAR Y NO INFRACCIÓN. EN NINGÚN CASO SERÁN LOS AUTORES O LOS TITULARES DE LOS DERECHOS DE AUTOR RESPONSABLES POR CUALQUIER RECLAMO, DAÑOS U OTRA RESPONSABILIDAD, YA SEA EN ACCIÓN DEL CONTRATO, EXTRA CONTRACTUAL O DE OTRO TIPO, DERIVADOS, FUERA DE O EN CONEXIÓN CON EL USO DEL SOFTWARE.

Licencia de JQuery Mobile:

- Copyright jQuery Foundation and other contributors,
- <https://jquery.org/>
- Mas información sobre la licencia: <https://github.com/jquery/jquery-mobile/blob/master/LICENSE.txt>

Licencia de JQuery

- MIT License: <https://opensource.org/licenses/MIT>
- Mas información sobre la licencia: <https://jquery.org/license/>

Licencia sobre API Facebook

- Licencia Apache
- Más información: <http://www.apache.org/licenses/LICENSE-2.0.html>

Licencia de uso de FontAwesome.io

- **Licencia de Fuentes:** SIL OFL 1.1
- **Licencia de código:** MIT License
- **Licencia de documentación:** CC BY 3.0
- Información completa sobre su licencia: <http://fontawesome.io/license/>

Licencias de p5.js y p5.dom

- La biblioteca p5.js es software libre; puedes redistribuirla y/o modificarla según los términos de la [GNU Lesser General Public License](#) publicada por la Free Software Foundation, versión 2.1.
- La Referencia del lenguaje se encuentra bajo una licencia [Creative Commons](#) que permite reusar este contenido para propósitos no-comerciales si se otorga el crédito correspondiente.

HTML5-Paint-Bucket-Tool

- Copyright: William Malone (www.williammalone.com)
- Licencia de uso: La porción de código se encuentra bajo licencia de uso de Apache License, Version 2.0. Más información sobre esta licencia: <http://www.apache.org/licenses/LICENSE-2.0>

Función drawImageProp

- Copyright: Ken Fyrstenberg Nilsen
- Licencia de uso: [Creative commons](#), CC BY-SA 3.0
- Fuente: <https://stackoverflow.com/questions/21961839/simulation-background-size-cover-in-canvas/21961894>

CodeIgniter

- Copyright © 2014 - 2017: British Columbia Institute of Technology
- Licencia de uso: The MIT License (MIT)
- Fuente: https://codeigniter.com/user_guide/license.html

Abstract

De entre todas las propuestas hechas por el consultor para elegir un proyecto, la más atractiva es la creación de una herramienta de dibujo, tipo Paint, pensada para trabajar de forma online. La aplicación es una web app que copia las herramientas de dibujo básico incorporados en el clásico Paint de Windows incluyendo nuevas características que lo hagan más avanzado que el programa original.

Aunque la aplicación incluye las típicas herramientas de dibujo básico como las formas poligonales, relleno, borrado, etc., también incorpora dos características nuevas: la posibilidad de trabajar online y la incorporación de nuevas herramientas de retoque fotográfico que permitan al usuario aplicar filtros básicos, al estilo Instagram, sobre sus imágenes.

El carácter online de la aplicación abre un mundo de posibilidades que permitirá a los usuarios compartir sus imágenes de forma inmediata en redes sociales, disponer de un espacio virtual y poder utilizar la aplicación desde cualquier lugar y dispositivo. Esta última característica es posible gracias al diseño Responsive que sigue la aplicación y al sistema de registro y autenticación de usuarios, necesario para controlar los datos de los mismos.

En conclusión, el objetivo principal del proyecto es ofrecer un programa de dibujo que se ejecute en cualquier dispositivo y para ello, el proyecto se estructurará en dos partes. Por un lado, está el FrontEnd de la aplicación que será el responsable de la parte visual y se programará usando HTML y Javascript, que son lenguajes de programación que trabajan del lado del cliente. Por otra parte, también habrá que implementar el BackEnd de la aplicación, que será desarrollado usando lenguajes de programación que gestionen el lado del servidor y se programan usando lenguajes como PHP y MySQL.

Notación

Tipo de letra usado en todo el texto:	Calibri	11 Ptos.
	Ejemplo: Texto	
Título de portada:	Calibri Negrita	22 Ptos. Gris claro y sombreado.
	Ejemplo: Texto	
Título 1:	Calibri Light Azul Oscuro 25%	16 Ptos.
	Ejemplo: Texto	
Título 2:	Calibri Light Azul Oscuro 25%	13 Ptos.
	Ejemplo: Texto	
Título 3:	Calibri Light Azul Oscuro 50%	12 Ptos.
	Ejemplo: Texto	
Citas:	Calibri Cursiva	11 Ptos. Negro claro 50%
	Ejemplo: <i>Texto</i>	
Leyendas y pie de imágenes:	Calibri Light Cursiva	8 Ptos. Negro claro 50%
	Ejemplo: <i>Texto</i>	
Fragmentos de código:	Consola Sombreado Blanco 15%	11 Ptos.
	Texto: Texto	
Resaltar texto	Calibri	11 Ptos
	Estilo: Normal	

Contenido

Copyright	2
Licencia del Proyecto TFG:	2
Licencia de uso de Spectrum JQuery ColorPicker	2
Licencia de JQuery Mobile:	2
Licencia de JQuery.....	2
Licencia sobre API Facebook	2
Licencia de uso de FontAwesome.io	3
Licencias de p5.js y p5.dom.....	3
HTML5-Paint-Bucket-Tool	3
Función drawImageProp	3
CodeIgniter.....	3
Abstract	4
Notación	5
Introducción	10
Descripción.....	11
Objetivos	12
Objetivos personales:.....	12
Objetivos generales:.....	12
Objetivos específicos:.....	12
Metodología.....	13
Planificación	14
Registro de Hitos	14
Planificación	15
Representación temporal.....	15
Proceso de trabajo	17
Búsqueda de recursos e información.....	17
Diseño de la aplicación y Prototipado.....	18
Diseño de imágenes	18
Diseño de Prototipos.....	19
Diseño de la base de datos.....	20
Desarrollo del FrontEnd	21
Diseño de la aplicación a bajo Nivel (JavaScript)	21
Creación e inicialización de los elementos del DOM	22
Creación del objeto Forma.....	22
Comportamiento de elementos incluidos en la aplicación.....	23

Aplicación de efectos	23
Diseño del BackEnd (CodeIgniter)	24
APIS utilizadas	27
jQuery API	27
jQuery Mobile API	27
Facebook API (SDK de Facebook para JavaScript)	27
Icomoon y Fontawsome	28
Spectrum	28
P5.js y P5.dom.js	28
jQuery UI	30
CodeIgniter	30
Diagramas UML	31
Caso de uso: Crear un dibujo	31
Prototipos	32
De bajo nivel:	32
Prototipos de Alto nivel	33
Guion	35
Perfil de usuario	37
Usabilidad	38
Seguridad	39
Tests	40
Bugs	42
Versiones de la aplicación	43
Requisitos de instalación	44
Instrucciones de instalación	45
Instrucciones de uso	47
Bugs	50
Proyección de futuro	51
Presupuesto	52
Análisis de mercado	53
Análisis de la competencia	53
Público objetivo	53
Tipos de consumidores	53
Competencia indirecta	53
Análisis DAFO	53
Viabilidad	55

Alcance del proyecto.....	55
Análisis de situación	55
Conclusión:.....	57
Anexos.....	58
Anexo 1. Entregables del proyecto	58
Anexo 2: Código fuente.....	58
Código HTML para definir el aspecto dependiendo del dispositivo	58
Código HTML que configura las hojas de estilo de Bootstrap.....	58
Creación de un círculo.....	58
Creación de una forma libre.....	60
Código HTML de la barra vertical.....	62
Control de la resolución de pantalla mediante JavaScript.....	62
Disposición de las librerías usadas en cabecera HTML. [Extracto de código].....	63
Método mostrarForma()	64
Web prueba para aplicación de efectos.....	68
Código de sckets.js	69
Uso de la API de Facebook.....	70
Implementación HTML de Spectrum.	71
Código del archivo filtros.js.....	72
Código que muestra un cuadro de dialogo modal.....	74
Código para crear un Slider.....	74
Código que selecciona el filtro a través del cuadro desplegable Aplicar Filtro.....	75
Código Mysql para generar la base de datos	76
Código del lado del servidor.....	76
Anexo 3. Librerías / Código externo utilizado	84
Anexo 4. Capturas de pantalla	85
Imagen del logo creada con Illustrator	85
Imagen del pincel	85
Goma de borrar.....	86
Esquema que muestra la distribución de las capas HTML.....	86
Control SELECT en forma de SPAN	87
Estructura de archivos de CodeIgniter	87
Estructura de directorios del servidor local	88
Encriptación de iPaint	88
Archivo de configuración de base de datos en CodeIgniter	89
Modificaciones del archivo Autoload.php	89

Uso de la API de Facebook en modo escritorio.....	90
Buscador de iconos de la web fontawesome.io.....	90
Creación de iconos a través de Icomoon	91
Cuadro desplegable para aplicar filtros	91
Cuadro de dialogo Nuevo.....	91
Aplicar efecto Blur.....	92
Ventana de identificación o registro	92
Cabecera de la aplicación web	92
Forma del ratón al pasar por el lienzo	93
Aspecto del pie de página de la aplicación	93
Anexo 6. Libro de estilo.....	94
Título de la aplicación.....	94
Marca o logotipo	94
Tipografía	94
Botones	95
Anexo 7. Resumen ejecutivo.....	96
Anexo 9. Bibliografía	97
Anexo 10. Nota biográfica.....	98

Introducción

Como graduado multimedia, me he especializado en el desarrollo de aplicaciones interactivas. La creación de cualquier aplicación conlleva el análisis y el estudio de tres elementos fundamentales:

- Estudio de la lógica de negocio y cómo afecta el uso de los sistemas de información al modelo de negocio asociado.
- Gestión del proyecto, considerando las tres variables fundamentales: Alcance, Coste y Tiempo.
- Estudio y búsqueda de recursos.

En general, el desarrollo de cualquier aplicación, constituye una gran oportunidad para poner en práctica mis conocimientos informáticos, mis habilidades como gestor de proyectos y tener una visión más realista sobre el tiempo que se tarda en la realización de aplicaciones y posibles inconvenientes que se puedan ir encontrando. De manera más concreta, la elección de desarrollar una aplicación web de dibujo, me permite adquirir experiencia en los siguientes puntos:

Planificación realista: La planificación del proyecto se ha llevado a cabo mediante la creación de un diagrama de Gantt. Cuando no tienes experiencia, esta parte supone ser una de las más difíciles porque se desconoce el coste temporal de cada tarea y, además, el diagrama de Gantt pretende ser lo más realista posible.

Coste económico: El coste y viabilidad del proyecto también es un punto importante a tener en cuenta. Nadie quiere un proyecto que no sea económicamente rentable. Definir una línea de negocio alrededor del proyecto que lo amortice en un tiempo límite y además que controle el coste de realización del mismo son puntos importantes y que hay que tener en cuenta en la realización de cualquier proyecto.

Metodología: A la hora de emprender un proyecto siempre conviene seguir una metodología, ya que, de lo contrario, el proyecto puede volverse caótico y no terminar bien. La importancia de los Blueprints y guiones interactivos, radica precisamente en este hecho, es decir, no debemos empezar a trabajar sobre un proyecto sin tener clara cuál es la línea gráfica que va a seguir el mismo ni tampoco cuales son las diferentes pantallas que van a ir apareciendo hasta resolver una necesidad el usuario.

Nuevas disciplinas adquiridas: Actualmente tengo conocimientos en JavaScript, jQuery, MySQL, PHP, etc..., pero lo que creo que es realmente importante en la realización de cualquier proyecto es que no es necesario que lo sepas todo para poder abarcarlo. ¿Acaso Steve Jobs conocía como funcionaban las antenas cuando invento el Iphone?. Nos encontramos ante una aplicación web donde aprenderé a usar el objeto canvas, pero ante ciertas dificultades, tendré que acudir a ciertos portales webs especializados a pedir ayuda y tratar de resolver los problemas por mí mismo.

En definitiva, la realización de este proyecto supone todo un reto que me permitirá adquirir nuevas habilidades, mayor seguridad, mayor capacidad organizativa, etc... Por tanto, todo esto son oportunidades que me permitirán tener una visión más realista de lo que conlleva la realización de cualquier proyecto.

Descripción

Foto-clic, es una aplicación de dibujo pensada para trabajar en modo online. Su funcionamiento es similar a Paint ya que ofrece las mismas características que este último, pero incorporando además funciones avanzadas de tratamiento de imágenes que apliquen transformaciones puntuales, espaciales y geométricas. El desarrollo de estos filtros, se ha decidido implementarlo mediante una biblioteca de JavaScript llamada p5.js, que permite interpretar el código implementado en Processing y llevarlo al contexto web. Gracias a esta nueva característica se podrá aclarar u oscurecer imágenes, posterizar, suavizar, dilatar una imagen o incluso rotarla.

Su carácter online independiza a la aplicación de la plataforma y del dispositivo donde se ejecuta, y por este motivo se hace necesario controlar el acceso a los datos mediante un sistema de gestión de usuarios. Para poder guardar o recuperar datos de la nube, el programa siempre pedirá la autenticación del usuario, no obstante, para no interferir en el uso de la aplicación se prioriza su uso frente al registro de nuevos usuarios, es decir, los usuarios anónimos podrán usar la aplicación en su totalidad siempre que no tengan que guardar o recuperar algún dato, en cuyo caso sí tendrán que pasar por el proceso de registro/autenticación, el cual es imprescindible para asegurar el control de los datos guardados. Además, los usuarios también podrán compartir sus trabajos en las redes sociales o incluso promocionar el uso de la herramienta.

Por último, para incorporar todas estas mejoras, la aplicación se desarrolla siguiendo el patrón de diseño MVC (Modelo-Vista-Controlador). **Para el diseño del Backend se utiliza PHP y MySQL**; la combinación de ambas tecnologías permitirá controlar el almacenamiento de los datos y la gestión de usuarios. Con ayuda del **framework CodeIgniter**, la **tecnología de servidor se implementará de manera más sencilla y comprensible**. Por otra parte, **para el diseño del FrontEnd se necesita HTML, CSS3 y Javascript**, siendo este último el que más peso tiene en el desarrollo del programa Fotoclic. La mayoría de **las operaciones en JavaScript se llevan a cabo realizando operaciones sobre un objeto canvas**. El uso de frameworks como JQuery y Konva, facilitarán la programación del mismo. En lo relativo a la parte visual, la disposición de los elementos web y el diseño responsive se desarrolla usando Bootstrap.

Foto-clic incluye las herramientas clásicas de cualquier programa de dibujo: selección rectangular, dibujo de líneas, dibujo de formas poligonales, zoom, coloreado e introducción de textos. Para diferenciarse de la aplicación de Paint, Foto-clic incluye las siguientes mejoras:

- **Multiplataforma:** La aplicación se puede ejecutar en diferentes dispositivos sin importar el sistema operativo en uso.
- **Multidispositivo:** Su diseño Responsive, permite su ejecución tanto en ordenadores como dispositivos móviles o tabletas.
- **Funciones avanzadas de tratamiento de imágenes:** Foto-clic incorpora filtros avanzados para la transformación de imágenes.
- **Portabilidad:** Se permite el almacenamiento de las imágenes tanto en la nube como en local. Esta característica aumenta el control y la disponibilidad de los trabajos realizados.
- **Autenticación de usuarios:** Permite una mayor gestión de usuarios y un mayor control sobre los datos.
- **Última tecnología en el desarrollo:** A nivel técnico, la aplicación se desarrolla utilizando el patrón de diseño MVC para permitir mayor modularidad y escalabilidad.

- **Promoción:** Permite a los usuarios compartir sus trabajos fácilmente a través de las redes sociales.

Objetivos

Objetivos personales:

A nivel personal, la realización de este proyecto permitirá consolidar mis conocimientos en programación avanzada, y además permitirá explorar nuevas tecnologías y patrones de diseño que hasta ahora no he usado. En consecuencia, la realización de este proyecto debe constituir una buena oportunidad para acreditar mis conocimientos a nivel profesional y personal.

Objetivos generales:

Como se ha comentado en apartados anteriores, para dotar a la aplicación de un carácter online, es necesario que esta se ejecute en un navegador web. Por tanto, su diseño y desarrollo estará basado en el uso de tecnología web que por un lado permita la ejecución del programa en el ordenador cliente mediante código JavaScript y por otro, permita el almacenamiento y/o gestión de usuarios y datos en remoto, usando para tal fin tecnología PHP y MySQL. La combinación de ambas tecnologías se llevará a cabo de manera asíncrona para no demorar con las cargas de datos y ofrecer la mayor similitud posible a los programas de escritorio. Esta tecnología se implementará mediante AJAX.

Por último, para facilitar el desarrollo de la aplicación tanto del lado del servidor como del lado del cliente, se usarán frameworks que permitan mayor escalabilidad y entendimiento del mismo, siempre, acompañado de una buena documentación que permita futuras versiones del programa en caso de recibir alguna actualización. Por otra parte, la aplicación debe promocionar su uso mediante el uso de redes sociales, donde, además, los usuarios podrán compartir sus trabajos y de manera indirecta promocionarán el uso de la aplicación.

Objetivos específicos:

La aplicación tiene como objetivo proporcionar una herramienta de dibujo online parecida a Paint que permita a los usuarios realizar dibujos sobre un lienzo y con el ratón mediante un conjunto de herramientas básicas. Estos dibujos podrán ser compartidos, guardados en la nube o descargados al sistema local.

La aplicación debería ejecutarse en un entorno web, siendo compatible con cualquier navegador y plataforma, es decir, siguiendo un diseño responsive. La adaptación del programa en diferentes pantallas siempre debe seguir el diseño y resolución más adecuados para mejorar la experiencia de usuario.

Para hacer que la interfaz de la aplicación se fácil de usar y los comandos sean claramente reconocibles, esta debería seguir una línea gráfica parecida a Paint. Con esta estrategia se persigue aumentar la usabilidad de la aplicación y hacerla fácilmente reconocible como programa de dibujo.

La aplicación requiere un sistema de autenticación de usuarios que ofrezca seguridad y control sobre los datos de los usuarios. Para ello, los usuarios nuevos podrán registrarse a través de la propia página.

Metodología

La metodología elegida consiste en primer lugar crear una serie de blueprints de pantallas que se pasarán a HTML y CSS. Luego definimos el comportamiento y apariencia con JavaScript y por último definimos el comportamiento en el lado del servidor con PHP y MySQL.

Por supuesto, puede ocurrir que me quede atascado en muchos puntos, amenazando el cumplimiento de la planificación. Para evitar esto, cuando llegue a un punto donde no pueda avanzar, empezaré a trabajar sobre otro tema dejando apartado temporalmente el problema en curso.

Además, cuando me enfrente a un problema que por su complejidad no sea capaz de resolver, optaré por algunas de las siguientes opciones:

- **Simplificar.** Así conseguiré dividir el problema en fragmentos más fáciles de comprender y será más fácil buscar una solución final. Por ejemplo, al implementar la herramienta circulo, en vez de hacerlo sobre el proyecto original, empiezo uno nuevo donde incluya los elementos necesarios para que funcione. Empiezo haciendo el circulo, luego lo rehago añadiendo interactividad con el ratón y por último añado lo aprendido al proyecto final.
- **Consultar.** Existen multitud de repositorios webs especializados donde se puede preguntar cualquier duda. La mayoría de las veces dan una solución, no obstante, cuando no lo den, buscaré códigos que se parezcan a lo que trate de implementar y siguiendo con lo explicado en el apartado anterior adapto el código a mi proyecto.
- **Estudiar.** A menudo, si se estudia y se aclaran conceptos, puede servir de estrategia para conseguir resolver un problema. En mi caso no conozco el objeto Canvas ni tampoco Codelgniter, entre otras muchas tecnologías. Para ello acudiré a videos de youtube, blogs, o documentación, que me permita seguir una línea en el proceso de trabajo.

En conclusión, la metodología que pretendo aplicar sigue el clásico “Divide y vencerás” junto con una buena dosis de documentación y estudio de la materia. No obstante, habrá momentos en que me atasque y no pueda continuar con el proceso de trabajo. En estos casos, continuaré con la planificación del proyecto previamente definida, y retomaré el problema cuando encuentre la mejor solución.

Planificación

Desglosando el proyecto en fases de trabajo, se distinguen 6 etapas claramente diferenciadas:

- **Diseño e implementación de la base de datos.** Base de datos que albergará la información relativa a los usuarios y la ruta de acceso a las imágenes guardadas en el servidor.
- **Desarrollo de la aplicación cliente.** Se refiere al desarrollo lógico de la aplicación foto-clic, es decir, todas y cada una de las funciones implicadas para el funcionamiento de la aplicación. El lenguaje utilizado en esta fase es javascript.
- **Programación del lado del servidor.** Se refiere al desarrollo lógico relativo a la gestión de usuarios (creación, acceso) de los usuarios y la posibilidad de guardar y recuperar imágenes del servidor.
- **Diseño gráfico de la interfaz.** En esta fase se decide el aspecto visual de la aplicación y se desarrollan todos los componentes gráficos que van a modelar la apariencia de la aplicación. En esta fase están también implicados aspectos de arquitectura de información, diseño de interfaces y usabilidad.
- **Integración de contenidos.** Esta fase consiste en la integración de todos los componentes realizados en etapas anteriores. En esta etapa se trabajará con HTML 5 y CSS3 para poder combinar todos los componentes teniendo en cuenta las diferentes plataformas en las que se va a ejecutar la aplicación. Es en este punto donde se desarrollará el diseño Responsive de la aplicación.
- **Aspectos legales.** Se desarrollará toda la documentación y apartados necesarios que informen al usuario sobre el uso de cookies y los derechos de propiedad intelectual, así como los mecanismos para solicitar la baja del servicio.

Dado que el proyecto da mucho juego, y con el propósito de no hacerlo demasiado extenso, este se desarrollará siguiendo un proceso evolutivo iterativo que consta de tres etapas: Diseño, Producción y Difusión. Se implementarán primero las operaciones básicas de dibujo Paint, y luego, las operaciones de autenticación y gestión de usuarios. En última instancia se implementará las operaciones de filtros de imágenes y compartición de datos por redes sociales.

También se desarrolla una planificación previa al proyecto, que permita desglosar en fases las diferentes tareas implicadas en su ejecución y proporcione una idea del coste temporal del mismo. La planificación no entra en contradicción con la característica evolutiva del proceso de trabajo, ya que es una guía que permite ir cumpliendo con la consecución de hitos, además de servir como herramienta para tener una mejor idea de la viabilidad del proyecto.

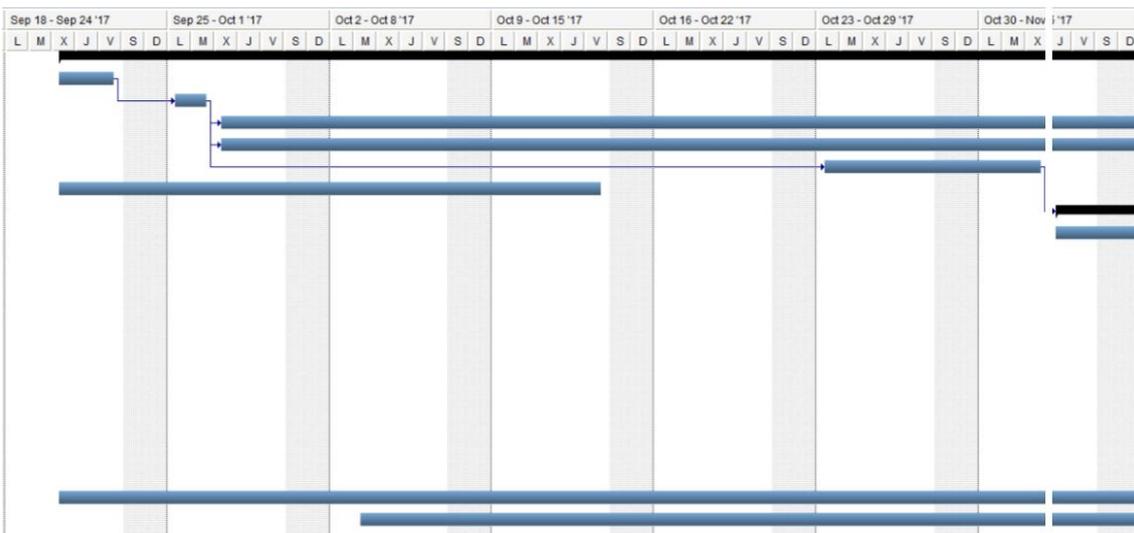
Registro de Hitos

NOMBRE	DURACIÓN	INICIO	FIN
PAC1	13 días	20/09/2017	03/10/2017
PAC2	28 días	04/10/2017	01/11/2017
PAC3	31 días	02/11/2017	03/12/2017
Entrega Final	39 días	06/12/2017	14/01/2018

Planificaci3n

Representaci3n temporal

Project Name TFG						
	i	Nombre	Duraci3n	Inicio	Fin	Predecesoras
1		[-] Dise1o	49d1as?	20/09/2017	27/11/2017	
2		Desarrollo de blueprints	3d1as?	20/09/2017	22/09/2017	
3		Dise1o de interfaz web	2d1as?	25/09/2017	26/09/2017	2
4		Programaci3n HTML	44d1as?	27/09/2017	27/11/2017	3
5		Programaci3n CSS3	43.5d1as?	27/09/2017	27/11/2017	3
6		Dise1o de la base de datos	8d1as?	23/10/2017	01/11/2017	3
7		An1lisis de la aplicaci3n y b1squeda de primeros recursos	18d1as?	20/09/2017	13/10/2017	
8		[-] Producci3n	42d1as?	02/11/2017	29/12/2017	6
9		Desarrollo de la aplicaci3n cliente Javascript	42d1as?	02/11/2017	29/12/2017	
10		Inclusi3n de filtros de imagenes	10d1as?	20/11/2017	01/12/2017	
11		Configuraci3n de servidor local	4.88d1as?	04/12/2017	08/12/2017	10
12		Programaci3n backend CodeIgniter	15d1as?	11/12/2017	29/12/2017	11
13		Desarrollo de la base de datos	3d1as?	06/12/2017	08/12/2017	
14		Inclusi3n de redes sociales	9d1as?	11/12/2017	21/12/2017	11
15		Ultimos retoques y dise1o de herramientas	5d1as?	25/12/2017	29/12/2017	
16		Estudio de la plataforma CodeIgniter	19.38d1as?	04/12/2017	29/12/2017	
17		[-] Difusi3n	9d1as?	01/01/2018	11/01/2018	8
18		Redacci3n de textos legales y licencias	3.13d1as?	01/01/2018	04/01/2018	
19		Integraci3n de contenidos	5d1as?	01/01/2018	05/01/2018	
20		Detecci3n y Correcci3n de errores	9d1as?	01/01/2018	11/01/2018	
21		Documentaci3n Memoria TFG	78.38d1as?	20/09/2017	08/01/2018	
22		Busqueda de recursos y fuentes de informaci3n	72.63d1as?	03/10/2017	11/01/2018	



Proceso de trabajo

El proceso de trabajo se divide en varias etapas:

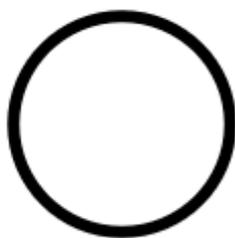
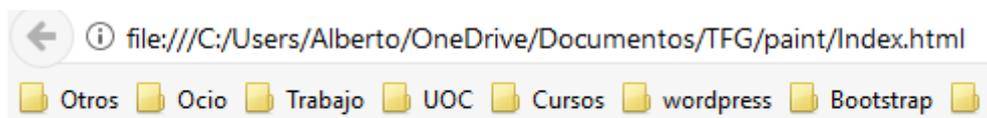
1. Búsqueda de recursos e información
2. Diseño de la aplicación (Prototipado)
3. Diseño de la base de datos
4. Desarrollo del Frontend
5. Diseño de la aplicación a bajo nivel (Javascript)
6. Diseño del Backend (CodeIgniter)

Búsqueda de recursos e información.

El eje central para el funcionamiento de la aplicación es el objeto Canvas incluido en el estándar HTML5. Este objeto, permite la representación de imágenes y su manipulación a través de código JavaScript y es compatible con la mayoría de los navegadores webs, aunque en versiones anteriores a la versión 9 de Internet Explorer presenta problemas de compatibilidad que pueden ser resueltos mediante plugins desarrollados por Google y Mozilla.

Como punto de partida, la web de [desarrollo Mozilla](#) ofrece un tutorial muy amplio donde se explica su uso. A partir de aquí, y con la ayuda de [Google](#) y foros (como [stackoverflow](#)), ha sido posible implementar las operaciones básicas de dibujo necesarias para la aplicación. Estas operaciones son las siguientes:

Creación de un círculo [[ver código fuente](#)].



Creación de una forma libre [[ver código fuente](#)].

file:///C:/Users/Alberto/OneDrive/Documentos
Otros Ocio Trabajo UOC Cursos wc



Para conseguir implementar las formas anteriores ha sido necesario consultar varias fuentes de información en Internet, pero la más influyente ha sido: <https://blog.openalfa.com/como-dibujar-con-el-raton-en-un-canvas-html5>

Llegados a este punto, y para no demorarme mucho en este apartado, el procedimiento para crear el resto de formas es el mismo, pudiendo cambiar en algunos detalles.

Diseño de la aplicación y Prototipado

Diseño de imágenes

La aplicación necesita tener un símbolo que la identifique, además de hacer uso de recursos gráficos que se incluyen como parte de las herramientas.

En el caso del logotipo, he creado la siguiente imagen a través de Illustrator:

*Creación del **logotipo** [ver imagen].*

La imagen del pincel incluida en el logotipo es de dominio público y ha sido descargada de la web:

<https://publicdomainvectors.org/es/vectoriales-gratuitas/Pincel-y-colorante-marr%C3%B3n/60439.html>

*Ver Imagen del **pincel** [ver imagen]*

En cuanto a los iconos de las herramientas, todos han sido creados por mi o extraídos de repositorios públicos como FontAwesome.io o Icomoon.io. Igual que antes, se crea en Illustrator la forma vectorial para luego exportarla a formato “.svg”:

*Proceso de creación de la **goma de borrar** [ver imagen]*

La librería Icomoon.io, se encargará de incluirla como icono integrante de la barra de herramientas.

Diseño de Prototipos

Siguiendo con la planificación propuesta en la PAC01, procedo a crear los prototipos de la aplicación. Para la creación de los prototipos, he tenido en cuenta las medidas recomendadas en la web [librosweb](#):

Medidas recomendadas para el diseño Responsive son:

- Dispositivos móviles en modo vertical:
 - Mínimo horizontal: 320px
 - Máximo horizontal: 599px
- Dispositivos móviles en modo horizontal:
 - Mínimo: 480px
 - Máximo: 599px
- Tabletas pequeñas en modo Portrait:
 - Mínimo: 600px
 - Máximo: 799px
- Tabletas pequeñas en modo Landscape:
 - Mínimo: 800px
 - Máximo: 767px
- Tabletas en modo Portrait:
 - Mínimo: 768px
 - Máximo 1023px
- Tabletas en modo Landscape:
 - Mínimo: 1024px
 - Máximo: indeterminado

La plantilla [CSS Bootstrap](#), me permite diseñar una web en modo responsive. Su web, presenta una extensa documentación que indica paso a paso como desarrollar los estilos pensando en modo responsive primero. Esto se hace añadiendo una etiqueta <META>, que configure el viewport de la página y añadiendo los enlaces a Bootstrap:

```
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
```

La creación de los blueprints en alto nivel, los desarrollo directamente en HTML y JavaScript.

A nivel HTML:

Se ha indicado que use una hoja de estilo diferente para cada rango de resolución:

Código HTML que define las hojas de estilo para cada tipo de dispositivo [\[Ver código\]](#)

Además, se han incluido las hojas de estilo de Bootstrap:

Código HTML que configura las hojas de estilo Bootstrap [\[Ver código\]](#)

La idea consiste en crear la siguiente plantilla HTML:

Esquema que muestra la distribución de las capas HTML. [\[Ver imagen\]](#)

La Barra vertical (zona que contiene las utilidades de la aplicación) cambiará de aspecto dependiendo de la resolución, es decir, en modo escritorio aparece como una barra vertical de dos columnas, pero en modo Responsive-Landscape aparece como una única fila horizontal. Es

por ello, que, con una resolución para escritorio, la barra vertical se configura como un DIV que contiene una lista no numerada de elementos:

Código HTML de la barra vertical. [\[ver código fuente\]](#).

En el diseño del modo Responsive, mediante código JavaScript, se elimina esta estructura HTML y se crea de forma dinámica una nueva estructura más acorde con la nueva resolución. Por tanto, la comprobación de la resolución, no solo se lleva a cabo a nivel de estilos mediante el uso de media queries, sino que también se hace a través de código JavaScript, que permita modificar elementos HTML en la página para adaptarla a las nuevas necesidades:

Control de la resolución de pantalla mediante JavaScript. [\[ver código fuente\]](#).

Las funciones “showLandscapeScreen()” y “showPortraitScreen()” crean elementos HTML y actualizan el contenido HTML de la página principal. La función “commandSetupControllers()” añade comportamientos a estos elementos, para controlar su funcionamiento.

Para poder utilizar el evento “orientationchange” he incluido la librería jquery-mobile, tal y como muestra la siguiente línea HTML:

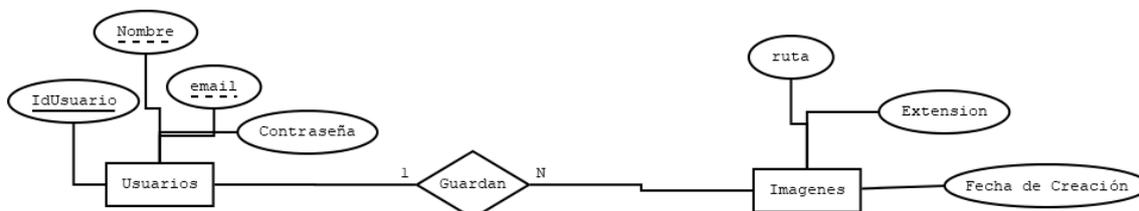
```
<script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>
```

El código completo se encuentra adjunto a este documento [\[Anexo 2\]](#).

Diseño de la base de datos

Modelo entidad/Relación

Dado que la aplicación requiere autenticación por parte de usuarios, el diseño Entidad / Relación de la base de datos quedaría como sigue:



Modelo Relacional

Relación Usuarios, con los siguientes atributos:

- **IdUsuario:** De tipo cadena “12548215”, “124125”, “145847568”. Lo defino de tipo cadena porque aunque contenga datos numéricos, cuando utilice el identificador de Facebook, como clave primaria, me almacenará números demasiado grandes para valores enteros. Es clave principal y no contiene datos repetidos
- **Nombre:** Tipo Cadena, con valores del tipo “Alberto”, “Antonio”, “Manu”, etc.... Es clave alternativa y no contiene datos repetidos.
- **Email:** Tipo cadena, con valores como “almt@gmail.com” , “perico@hotmail.com”, etc.... Es clave alternativa y no contiene datos repetidos
- **Contraseña:** Tipo cadena, “asdf46er”, “fsf89733”, “asd848ee”, etc...

Relación Imágenes, con los siguientes atributos:

- *Ruta*: Tipo cadena, con información del nombre y la ruta relativa de la imagen “./alberto/imagen1”, “./user2/imagen34”, etc...
- *Extensión*: “png”, “bmp”, “jpg”,
- *idUsuario*: es una clave foránea que apunta a la clave principal de la relación Usuarios.
- *Fecha de Creación*: Tipo fecha/hora. Almacena información sobre la fecha y la hora en que la imagen fue creada.

Código MySql de generación de las tablas y la base de datos [[Ver código](#)]

Desarrollo del FrontEnd

Habiendo completado las tres fases anteriores, la apariencia de la aplicación empieza a tener forma, no obstante, el proceso de creación del FrontEnd no termina en la etapa de Diseño de aplicación y Prototipado, porque a medida que voy incluyendo librerías auxiliares, estas van modificando la apariencia y comportamiento de los elementos incrustados. Por ejemplo, me llevó varios días descubrir que el control SELECT, que utilizo para elegir los filtros de imágenes, se transformaban en etiquetas de tipo SPAN, tal como muestra la siguiente imagen:

Imagen de un control SELECT que se construye con SPAN [[Ver imagen](#)]

Además, a medida que iba creando cuadros de diálogos que no había previsto en etapas anteriores, sucedía que tenía que volver a escribir código HTML y CSS, con sumo cuidado de no afectar a otros componentes ya creados. Por ejemplo, cuando incorporé la librería JQuery UI, para el uso de cuadros de diálogos modales, tuve que modificar el orden de inserción de las llamadas a las librerías ya que unas se pisan a otras, siendo el orden definitivo el siguiente:

Disposición de las librerías usadas en cabecera HTML. [[Ver código fuente](#)].

Además de lo anterior, en esta etapa he ido creando los diferentes cuadros de diálogos que me faltaban:

- Cuadro de dialogo Abrir
- Cuadro de dialogo Nuevo
- Cuadro de dialogo Guardar
- Cuadro de dialogo Iniciar sesión
- Cuadro de dialogo Efectos

Aún falta por crear la fase del registro, que se llevará a cabo cuando proceda a desarrollar la fase del BackEnd.

Diseño de la aplicación a bajo Nivel (JavaScript)

Esta etapa es la más compleja, porque implica toda la programación del lado del cliente e implementado en JavaScript. Esta etapa se subdivide a su vez en las siguientes fases:

- **Proceso de inicialización de los elementos del DOM.** Toda la inicialización de los elementos del DOM se lleva a cabo en el fichero ‘inicio.js’.
- **Creación del objeto forma.** La forma representa todo lo que se dibuja en el lienzo.
- **Comportamiento de elementos incluidos en la aplicación.** Algunos elementos, requieren configuraciones más complejas. Para no ensuciar de código el fichero ‘inicio.js’, utilizo el fichero ‘buttonActionSetup.js’, que me sirve para definir comportamientos avanzados de los elementos incluidos en el DOM.
- **Aplicación de efectos.** Esta es la parte responsable de aplicar los filtros de imágenes.

Creación e inicialización de los elementos del DOM

Primero se asegura la carga completa del DOM mediante la función:

```
$(document).ready(function() {  
    [.....]  
});
```

Dentro de esta función se definen el comportamiento de los objetos **screen** y **window**. Por un lado, si la resolución horizontal está entre 710px y 799px, se carga la función “showLandScapeScreen()”, en cambio para resoluciones inferiores a 709px se ejecuta la función ‘showPortraitScreen()’ y para resoluciones superiores a 800px realizamos unos pequeños ajustes.

Control de la resolución de pantalla [\[ver código\]](#)

En cuanto a la inicialización del objeto window, la verdad es que me ha traído de cabeza porque me llevó varios días darme cuenta de su funcionamiento. Dado que la aplicación está pensada para ejecutarse en entornos móviles, hay que tener en cuenta los diferentes giros de pantalla adaptando la ventana al ancho de pantalla en cada momento. El objeto window dispone del evento “orientationChange”, que es el responsable de definir el comportamiento en cada momento. Lo que hago es, por tanto, llamar a la función ‘showLandscapeScreen()’ o ‘showPortraitScreen()’ según esté la pantalla en horizontal o vertical.

Lo que me dio problemas fue, que haciendo pruebas con el visor Responsive de Google y Firefox, la función no se ejecutaba. Tarde en darme cuenta de que la función no se ejecutaba, porque el emulador de ambos navegadores no ejecuta el evento “orientationChange”, de hecho, en la web JQuery Mobile, hay un ejemplo de uso del evento orientationChange, donde se puede comprobar que los navegadores no ejecutan dicho evento.

Por último, defino el comportamiento de cada uno de los botones al hacer clic en ellos. Dado que la ejecución de la aplicación muestra diferentes botones, según la plataforma en que se ejecute, defino el comportamiento del evento clic para botones en modo escritorio y en modo Responsive.

Creación del objeto Forma

En el archivo forma.js he creado una clase Forma, que es llamada desde los botones del DOM al hacer clic. La vida útil del objeto forma empieza en la etapa anterior “Inicialización del DOM”, al llamar a una función ‘init()’, que es la responsable de crear una instancia forma de la clase Forma.

En la función ‘init()’ también se inicializan ciertos eventos importantes para el tratamiento de la forma y asociados al objeto canvas. Estos eventos son:

- Mousedown
- Touchstart
- Mousemove
- Touchmove
- Mouseup
- Touchend

Cada vez que ocurre uno de estos eventos, y con el tipo de forma previamente seleccionada al haber hecho clic en el botón correspondiente, ocurre lo siguiente:

- Se define una forma nueva
- Se muestra la nueva forma
- Se guarda en el historial de formas

Al definir una nueva forma se ejecuta el método 'forma.definirForma()', y cuyo código es el siguiente:

```
this.definirForma= function(x, y, estado){
    lista_de_puntos.push([x, y, estado]);
}
```

Una vez se ha definido la forma al levantar el dedo del ratón (y durante el proceso de creación) la forma se muestra:

Método mostrarForma(). [\[Ver código fuente\]](#).

Hay que tener en cuenta, que una forma es algo abstracto y que por tanto puede representar un círculo, una línea, un rectángulo o incluso un color de relleno. En el futuro, por el modo en que he desarrollado el objeto Forma, me gustaría ampliarlo con nuevas herramientas como líneas rectas, estrellas o elipses, que aún no han sido implementadas por falta de tiempo.

Comportamiento de elementos incluidos en la aplicación

La interacción del selector del grosor de línea o el paso del foco entre botones para indicar que botón ha sido pulsado se llevan a cabo en el archivo 'buttonActionSetup.js'. Además, existen ciertos elementos que se ejecutan de forma dinámica, como los cuadros de diálogo abrir, nuevo, guardar...

Aplicación de efectos

Como ya se ha comentado en puntos anteriores, esta es la fase en la que he programado la aplicación de efectos a las imágenes mediante la inclusión de las librerías p5.js y p5.dom.js. El fichero donde programo los efectos es filtro.js. Pero antes de llegar a este punto, tuve que simplificar el trabajo, creándome una web pequeña con las herramientas necesarias para comprender el funcionamiento de estas librerías.

Para ello me creé una web llamada prueba con el siguiente código HTML:

Web prueba para aplicación de efectos. [\[Ver código fuente\]](#)

A continuación, me creé un archivo JavaScript llamado sketch.js, con el siguiente código de prueba:

Código de sketch.js. [\[Extracto de código\]](#)

El código anterior me ha servido de código de estudio para más tarde empezar a trabajar sobre mi proyecto.

Como se puede ver, el proceso de programación en Javascript ha sido el más complejo hasta el momento y el trabajo, no ha sido del todo satisfactorio. Por ejemplo, aún existen herramientas que se me resisten y no sé cómo programarlas. Por ejemplo:

- Uso de la webcam.

- Uso de la herramienta selección
- Uso de la herramienta zoom
- Abrir una imagen

El uso de la webcam requiere de un servidor seguro para que funcione. Actualmente, dispongo de un servidor seguro, pero aun así, sigo teniendo problemas para hacer funcionar el componente VIDEO. En el siguiente post: <https://stackoverflow.com/questions/47569400/how-to-make-the-camera-work-with-javascript/47586084#47586084>, pido ayuda para configurar el uso de la cámara.

Por otra parte, la herramienta selección representa un verdadero quebradero de cabeza, porque su función es la de seleccionar un área para operar con ella. El problema es que generalmente esta herramienta está pensada para aplicaciones de escritorio, donde se usan herramientas estandarizadas como copiar, cortar y pegar. Para aplicaciones responsive, no veo mucha utilidad a la herramienta selección, si bien es verdad que una forma de solucionar el problema consistiría en tratar a los objetos que incorporo en el canvas como elementos seleccionables y movibles, es decir, que el usuario pueda arrastrar e interactuar con ellos. Este implicaría un cambio profundo en la implementación de mi código y no sé si es viable.

La herramienta zoom, no he sido capaz de implementarla aún. He buscado ayuda en la web y he encontrado información que me puede ser útil. Hay un ejemplo de uso de la herramienta zoom en la web: http://phrogz.net/tmp/canvas_zoom_to_cursor.html, pero al tratar de adaptarlo para mi trabajo no he podido, no sé si por cansancio o por falta de conocimientos.

Por último, la incorporación de imágenes, aún está en desarrollo y aunque he logrado abrir imágenes locales, tengo el problema de que en el modo responsive la imagen no se adapta al tamaño del canvas. Problema que espero resolver pronto.

Diseño del BackEnd (CodeIgniter)

Habiendo terminado todas las etapas anteriores: Prototipado, diseño de base de datos, FrontEnd y programación del objeto Forma. Procedo a programar el proceso de trabajo relativo al servidor.

La programación del lado del servidor la llevo a cabo usando como lenguaje PHP, pero para no partir de cero y beneficiarme de un estilo de programación más organizado y estandarizado, me descargo CodeIgniter y lo instalo en mi servidor local.

CodeIgniter es un software *Open Source*, de desarrollo web pensado para construir páginas webs dinámicas usando PHP. La ventaja de este software es que está basado en el popular Modelo vista controlador (MVC), que es un patrón de diseño que consiste en dividir la solución de un problema en tres capas independientes. Por una parte, está la capa visual que es la que se encarga de la presentación de los resultados a los usuarios. Esta capa (Vista), contiene todas las páginas escritas en HTML y JavaScript, que muestran los resultados obtenidos de las otras dos capas. La capa encargada de las operaciones llevadas a cabo en el servidor (capa Controlador) se encarga de realizar las operaciones necesarias hasta dar con una solución que se muestre a través de la capa Vista. Dicho de otro modo, la lógica de la aplicación se implementa en la capa Controlador. Por último, la capa Modelo es la que se encarga de realizar las operaciones de entrada/salida con la base de datos. Cualquier dato que manipule la capa Controlador lo presenta a través de la capa vista y lo extrae a través de la capa Modelo.

CodeIgniter utiliza el patrón de diseño MVC y por tanto distribuye sus archivos de la siguiente forma:

Estructura de archivos de CodeIgniter [\[ver imagen\]](#)

El uso de CodeIgniter es muy sencillo y consiste en copiar toda la estructura de directorios en mi servidor local:

Estructura de directorios del servidor local [\[ver imagen\]](#)

Una vez he descargado y copiado la estructura de directorios, me creo una carpeta “Static” en la carpeta raíz del sitio local, donde incluiré todos los archivos de JavaScript, Imágenes, Fuentes, y archivos CSS, para ser llamados desde la Vista.

También creo en la carpeta raíz una carpeta llamada “images”, que contendrá las imágenes de que los usuarios guardan.

Una vez tengo todos los archivos copiados, modifico las páginas para que las plantillas, estilos e imágenes sean localizadas dentro de la carpeta *static*. Además, edito algunos archivos de configuración de CodeIgniter para un funcionamiento correcto. En primer lugar, modifico el archivo *config.php*, concretamente la línea que hace referencia a la llave de encriptación:

```
$config['encryption_key'] = 'e86b3395fa2b298cc0554644582f3340';
```

La línea anterior me permite usar sesiones, así que le indico el valor de la palabra ‘iPaint’ encriptada en MD5.

Encriptación de iPaint [\[ver Imagen\]](#)

Dado que también voy a usar bases de datos, modifico los parámetros del archivo *database.php*.

Archivo de configuración de base de datos. [\[ver Imagen\]](#)

Por último, dado que CodeIgniter también incluye un conjunto de utilidades (librerías) que facilitan su uso, modifico el archivo *autoload.php*:

Modificaciones del archivo autoload.php [\[ver imagen\]](#)

Una vez tengo CodeIgniter configurado y listo para funcionar me creo en la carpeta “views” tres archivos, *footer.php*, *header.php* y *index.php*. Estos archivos contienen la vista de la aplicación y son una copia del archivo *index.html* de la versión local de la aplicación. No obstante, se han realizado algunos cambios para que se comuniquen con el controlador.

En la carpeta “controller”, incluyo un archivo *Welcome.php* y un archivo “*Login.php*”. Cuando un usuario teclea el nombre de dominio, lo primero que hace CodeIgniter es llamar al controlador *Welcome.php*.

En *Welcome.php* se ejecuta la función *index* que comprueba si hay una sesión iniciada. En caso afirmativo, carga las vistas directamente pasando los valores de sesión. En caso negativo, carga las vistas sin pasar valores de sesión.

Una vez la página “*index.php*” se ha cargado, el usuario puede iniciar sesión y para ello debe ejecutar el controlador “*Login.php*”. Este controlador valida al usuario y contiene resto de métodos relativos al guardado, listado y borrado de imágenes. En realidad, las operaciones

relativas a las imágenes, debería haberlas incluido en un controlador diferente, pero por inexperiencia y falta de tiempo, las implementé todas en el mismo controlador.

Por último, el controlador Login.php, carga el modelo "Auth_model" incluido en la carpeta Model, que incluye las operaciones de escritura y lectura de la base de datos.

APIS utilizadas

- JQuery Api
- JQuery Mobile Api
- Facebook API
- IcoMoon y FontAwesome
- Spectrum
- P5.js y P5.dom.js
- JQuery UI
- CodeIgniter

JQuery API

Es una librería de JavaScript que proporciona mayor rapidez, claridad y sencillez en el código. Permite manejar eventos, crear animaciones, implementar AJAX y manipular elementos HTML de manera mucho más simple.

Para su uso, solo basta con indicarlo en una etiqueta HTML al comienzo del archivo, en el apartado <HEAD>

```
<script src="http://code.jquery.com/jquery-1.11.1.min.js"></script>  
https://api.jquery.com/
```

JQuery Mobile API

Es una extensión de JQuery API que proporciona nuevos manejadores de eventos para la adaptación web a dispositivos móviles. Por ejemplo, cuando un usuario rota el móvil se produce un evento de rotación de pantalla cuyo comportamiento podemos controlarlo gracias a esta API.

Para su uso, solo basta con indicarlo en una etiqueta HTML al comienzo del archivo, en el apartado <HEAD>

```
<script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-  
1.4.5.min.js"></script>  
http://api.jquerymobile.com/
```

Facebook API (SDK de Facebook para JavaScript)

Es una librería que permite interactuar con Facebook de diferentes maneras. Para el caso que nos ocupa, la librería se usa para iniciar sesión de manera automática, evitando al usuario perder tiempo en el proceso de registro. El inicio de sesión, nos devuelve información del usuario a través de Facebook que posteriormente se almacena en nuestra base de datos. El control de esta librería se realiza mediante JavaScript.

<https://developers.facebook.com/docs/javascript/quickstart>

La siguiente imagen muestra el uso de la librería en modo escritorio. Se puede observar como gracias a la API, nos aparece un botón de Facebook con el nombre de usuario, invitando a iniciar sesión con su usuario de Facebook.

Uso de la API de Facebook en modo escritorio [ver imagen]

La función “getInfo()”, nos devuelve información sobre el usuario en forma de objeto:

```
Object { error: Object } web:74:3
>> getInfo()
← undefined
Object { first_name: "Alberto", last_name: "Marin", name: "Alberto Marin", id: web:74:3
, picture: Object, email: "a@...com" }
```

Para usar la API, simplemente hay que seguir el tutorial que hay en la web de Facebook y escribir el siguiente código al principio de la página web, justo después de la etiqueta <BODY>.

Uso de la API de Facebook. [\[Ver código fuente\]](#).

Icomoon y Fontawsome

Estas librerías no son API's, pero permiten mostrar cualquier imagen en formato textual. En el caso de Fontawesome, existen un conjunto de iconos predefinidos que permiten usarse incluyendo la etiqueta siguiente en la cabecera de la página web:

```
<link rel="stylesheet" href="./font-awesome-4.7.0/css/font-awesome.min.css">
```

El sitio web <http://fontawesome.io/icons/> ofrece un buscador de iconos bastante completo que además indica como incluir en formato HTML el icono que se busca:

Buscador de iconos de fontawesome.io [\[ver Imagen\]](#)

Por otra parte, si el icono que buscamos no se encuentra disponible, podemos acudir a la web <http://icomoon.io>, que provee una aplicación que permite crear una lista de iconos personalizados, tal y como muestra la siguiente imagen:

Creación de iconos personalizados a través de Icomoon [\[Ver imagen\]](#)

Al igual que con Fontawesome, para poder usar este recurso, hay que incluir en la cabecera de la página el siguiente enlace:

```
<link rel="stylesheet" type="text/css" href="./css/style-icomoon.css">
```

La creación de los iconos se lleva a cabo creándolos a través de un programa de dibujo vectorial como Illustrator y posteriormente guardándolos en formato ".svg".

Spectrum

Es un plugin desarrollado por Brian Grinstead, que está disponible en la web <https://bgrins.github.io/spectrum/>. Este plugin permite incluir un selector de color personalizable en tu sitio web. En la web aparecen muchas opciones de configuración y además asegura que es responsive. Para poder usarlo, he incluido la siguiente etiqueta en la cabecera de la página:

```
<link rel="stylesheet" type="text/css" href="./bgrins-spectrum-98454b5/spectrum.css">
```

Además, su configuración se lleva a cabo a través del siguiente código:

Implementación HTML de Spectrum. [\[Ver código fuente\]](#).

P5.js y P5.dom.js

P5.js permite usar JavaScript al estilo de Processing, es decir, proporciona una función Setup que configura e inicializa el canvas y una función iterativa Draw, que permite manipular el

canvas creando animaciones, efectos e interactividad. Su uso es ideal para artistas y programadores de videojuegos, o como en mi caso, manipulación de imágenes.

Esta librería crea un canvas en tiempo de ejecución sobre el que se efectúan todas las operaciones. En mi caso, este canvas dinámico, lo mantengo siempre oculto porque lo utilizo como un canvas auxiliar en el que realizo transformaciones para posteriormente pasar el resultado de mis operaciones al canvas principal.

Desafortunadamente, las operaciones y transformaciones que P5.js realiza, siempre las hace sobre el mismo objeto canvas, es decir, no permite realizar operaciones sobre un canvas que haya sido creado previamente o sobre un elemento del DOM. Por esta razón, debo incluir también la librería P5.dom.js, que es una librería que permite interactuar con objetos HTML5 como videos, audios, webcam, input, etc.

En mi caso, dado que el uso de estas librerías las limito a la aplicación de filtros fotográficos, el flujo de trabajo es el siguiente:

- Incluyo las librerías P5.js y P5.dom.js en el documento HTML principal y en el siguiente orden:

```
<script
src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.5.16/p5.js"></scri
pt>
<script language="javascript" type="text/javascript"
src="./javascript/p5.dom.js"></script>
```

- Creo e incluyo un archivo JavaScript, donde realizo las operaciones sobre el canvas y que se llama filtros.js. El código de este archivo es el siguiente:

Código del archivo filtros.js. [\[Ver código fuente\]](#).

Obsérvese que, en el código anterior, la función *'setup'*, sirve para inicializar el canvas virtual. Por otra parte, la función *'draw'*, no se utiliza porque en realidad no necesitamos realizar ningún proceso iterativo, como por ejemplo una animación.

Pero la función más importante es *'mySelectEvent'*, ya que es la responsable de aplicar el filtro. Esta función recibe tres parámetros:

- **Parámetro val:** es el tipo de filtro que vamos a aplicar y lo recibe de una función que se ejecuta al hacer clic en el cuadro desplegable de filtros:

Cuadro desplegable para aplicar filtros. [\[ver imagen\]](#)

- **Parámetro img:** Representa la imagen que está representada actualmente en el canvas. Esta, es la imagen a la que se le aplicará el filtro y se pasa a la función en forma de array.
- **Parámetro n:** Este parámetro, solo tiene sentido para los filtros "POSTERIZE" y "BLUR", ya que estos son los únicos filtros que necesitan un parámetro que indiquen que cantidad de efecto se quiere aplicar. Dado que no todos los efectos requieren de este parámetro, es necesario preguntar a la función por el número de argumentos que se le pasan y así determinar las posibilidades.

La función *mySelectEvent()*, es llamada, cada vez que se elige un efecto del *'select efectos'*. La función *mostrarVentana()* definida dentro del archivo *'buttonActionSetup.js'*, es la responsable de llamar a *mySelectEvent()*, cada vez que se interactúe con el cuadro desplegable:

Código de selección del filtro [\[ver código\]](#)

jQuery UI

Esta librería, permite usar controles ya definidos y listos para ser usados, ahorrando así horas de trabajo. En mi caso, he usado un cuadro de dialogo modal y un control Slider.

- El cuadro de dialogo modal, se crea cada vez que hacemos click en el botón nuevo.

Cuadro de dialogo Nuevo... [\[ver imagen\]](#)

- Aunque este cuadro de dialogo que lo podría haber hecho yo, al igual que el resto de ventanas, decidí usar JQuery UI, por el hecho de que el cuadro era modal y ofrecía compatibilidad en otros dispositivos. El código responsable de mostrar el cuadro de dialogo aparece en el fichero 'buttonActionSetup.js' y es como sigue:

Código que muestra un cuadro de dialogo modal. [\[Ver código fuente\]](#).

En realidad, lo que se hace en el código anterior es inicializar un objeto y definir el comportamiento de los botones.

- El Slider o barra de deslizamiento horizontal, es otro control que viene predefinido y que es útil para ofrecer al usuario una forma fácil e intuitiva de aplicar una cantidad de efecto. Este Slider aparece en la ventana de aplicación de efectos:

Cuadro de dialgo Aplicar efecto... Blur [\[Ver imagen\]](#)

El código responsable de crear el Slider es el siguiente:

Código para crear un Slider. [\[Ver código fuente\]](#).

El código anterior aparece también en el fichero 'buttonActionSetup.js', que es el fichero que define el comportamiento de los botones.

CodeIgniter

Este framework está implementado en PHP y su uso consiste en descargar de la web <https://www.codeigniter.com/> y copiar en el servidor la estructura de carpetas y archivos que ya viene definida por defecto. Hecho esto se configuran los archivos que se localizan en el directorio 'application/config', para poder conectarlo con el servidor de base de datos y usar librerías que ya trae implementadas por defecto.

A partir de entonces, todas las páginas que se llamen desde el servidor llamen a un archivo controlador que dispondrá de métodos. Por defecto, los controladores cargarán el método Index (<http://localhost/controlador>), en cambio si se desea ejecutar un método diferente se teclea el nombre del controlador junto con el nombre del método. (<http://localhost/controlador/metodo>).

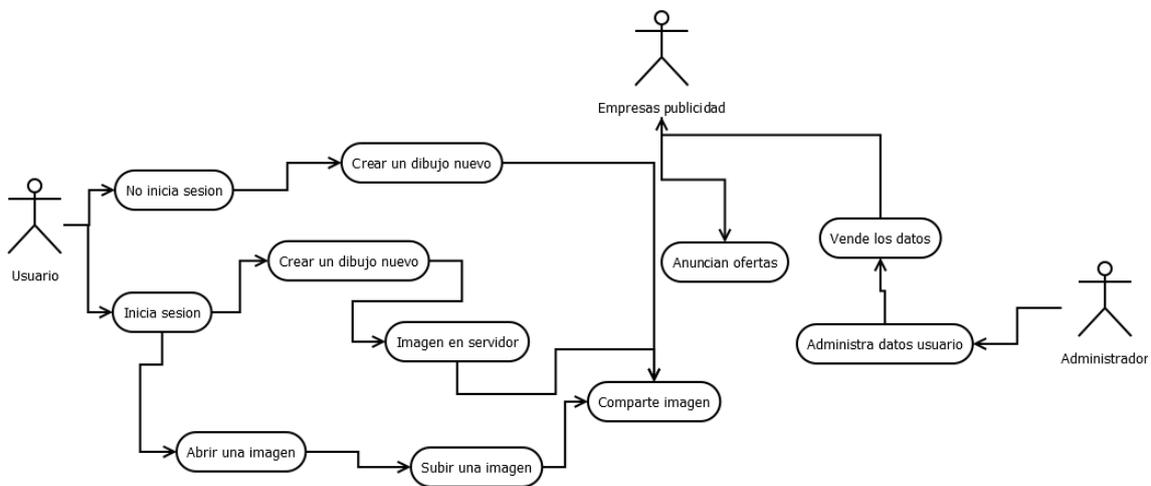
Los controladores cargan las vistas, que están localizadas dentro de la carpeta views. Las vistas están escritas en HTML y Javascript, aunque también pueden contener algo de código en PHP. No obstante, el código PHP que se incluye en las vistas no está pensado para calcular, sino más bien para renderiza el resultado obtenido por el controlador.

Por último, los modelos (localizados en la carpeta Model), son los responsables de la gestión de la base de datos. Ahí, realizamos las operaciones de lectura y escritura de datos en la BBDD.

Diagramas UML

Caso de uso: Crear un dibujo

Un usuario entra en la aplicación y le aparece un lienzo en blanco. El usuario puede dibujar formas que selecciona de la barra de herramientas. También puede abrir una imagen para su edición básica. Al abrir una imagen, si el lienzo está usado preguntará si quiere guardar los cambios. El usuario puede deshacer / rehacer cambios en todo momento, hasta un máximo de 20 cambios. Cuando el usuario termine de editar el lienzo, puede guardar o compartir la imagen creada. Para compartir la imagen en redes sociales, no es necesario que el usuario se haya identificado en el sistema, pero si necesita identificarse en la red social elegida. En el caso de que un usuario abra/guarde una imagen es necesario que el usuario se identifique. Si es un usuario nuevo, necesita pasar por un proceso de registro que puede hacer rellenando un formulario o logueando a través de su red social.



Prototipos

De bajo nivel:

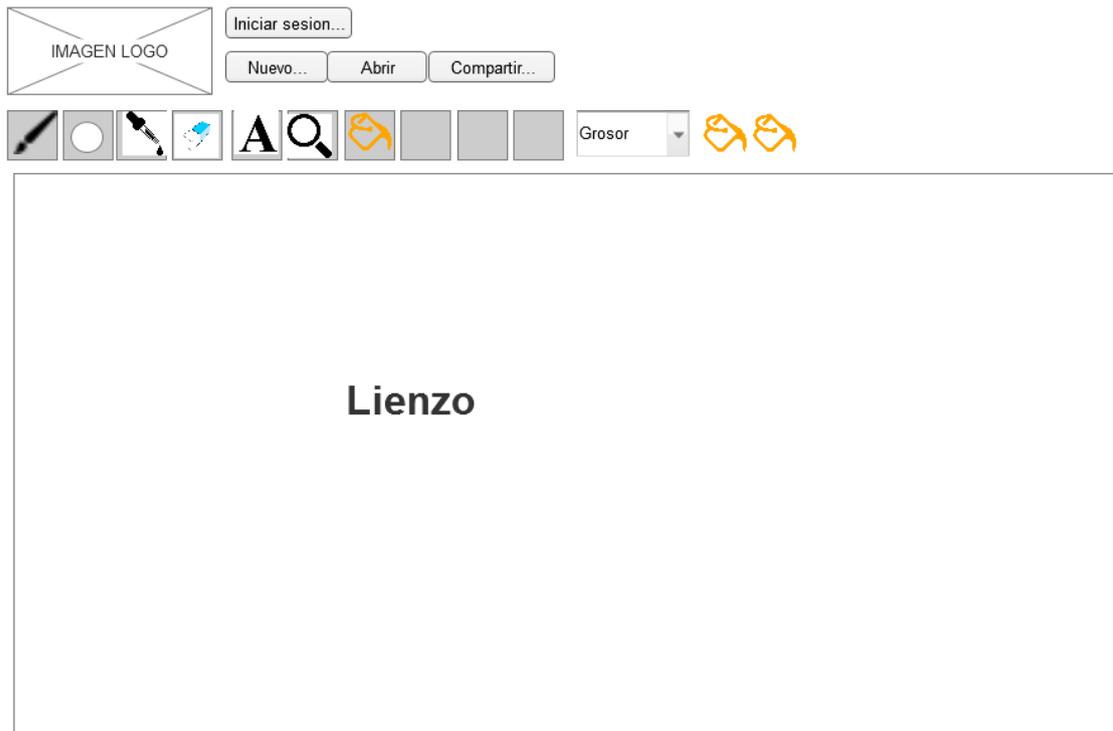
Modo escritorio (1024px de ancho)



Modo Responsive-portrait (320px de ancho)

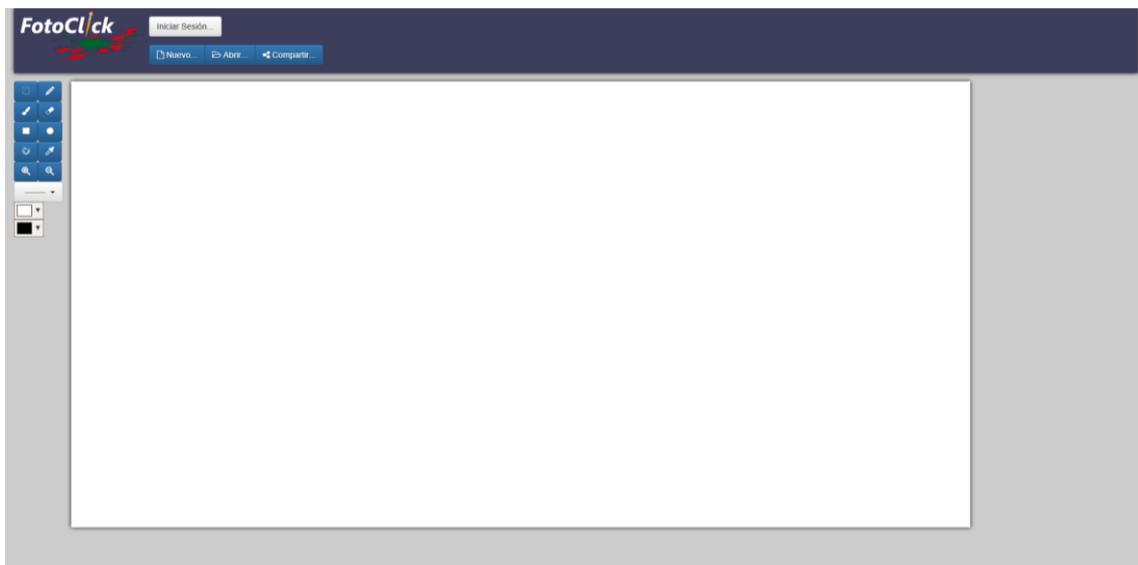


Modo Responsive-landscape (600px de ancho)

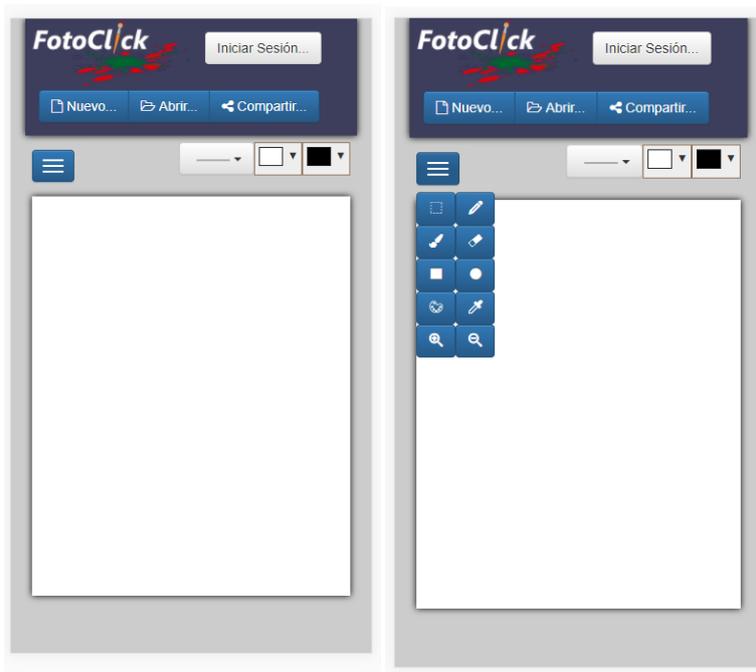


Prototipos de Alto nivel

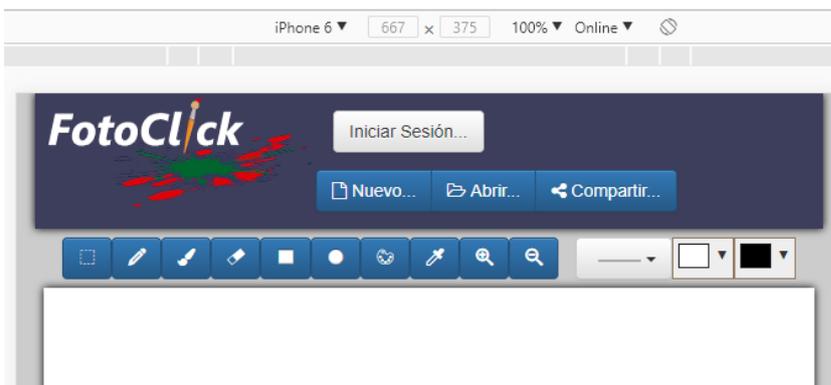
Modo Escritorio



Modo Responsive (Portrait)



Modo Responsive (Landscape)



Guion

Un usuario accede a la aplicación abriendo el navegador web y tecleando la dirección siguiente: “www.galeonweb.es/web”. El navegador que use el usuario es indiferente.

La aplicación se abre presentando un lienzo central donde de forma predeterminada aparece activa la herramienta selección. El usuario puede elegir la forma que desea pintar seleccionándola desde la barra de herramientas localizada en la zona izquierda de la ventana. Si el usuario se conecta desde móvil o Tablet, tiene que pinchar en el botón con forma de sándwich para que se desplieguen las herramientas disponibles. Cuando el usuario selecciona una herramienta, automáticamente puede colocarla en el lienzo con solo arrastrar el ratón mientras lo mantiene pulsado. Si el usuario se conecta desde móvil o Tablet, entonces tiene que mantener presionado el dedo sobre la pantalla para poder representar la forma que haya seleccionado.

Las herramientas que utiliza el usuario son personalizables en grosor y color de línea y color de relleno. Estas herramientas aparecen en la parte inferior de la barra de herramientas (o a la izquierda de la pantalla en dispositivos móviles) y el usuario puede seleccionarlas en cualquier momento personalizando las características del trazo y relleno de sus formas.

Las formas libres que el usuario dibuje que no queden completamente cerradas, podrán cerrarse con un doble click o con una doble pulsación.

El usuario puede avanzar o retroceder en el tiempo sobre las modificaciones que vaya realizando sobre el lienzo. Las acciones de deshacer y rehacer son las únicas acciones que permiten al usuario eliminar elementos de la escena, ya que este no dispone de un botón de borrado. El botón de eliminar objeto (o forma), no está disponible, porque el usuario no trabaja con elementos vectoriales y tampoco trabaja en un lienzo compuesto por capas. Por tanto, la aplicación provee al usuario de una única capa que sirve de interfaz de trabajo para el usuario, y los elementos que este vaya incorporando serán tratados como imágenes binarias y no como imágenes vectoriales.

La aplicación no provee de herramientas de uso tradicional como Cortar, Copiar y pegar. En cambio, el usuario dispone de una herramienta de selección, que le permite seleccionar un área rectangular del lienzo para arrastrarla (cortar) o duplicarla (copiar), cuando este arrastra dicha área con el ratón o con el dedo (en modo Responsive). Además, la herramienta selección permite al usuario limitar las modificaciones que el usuario haga sobre dicha área.

El usuario tampoco dispone de herramienta zoom, pero en cambio si puede ampliar o reducir la imagen usando la rueda del ratón cuando está en el modo escritorio o el movimiento de los dedos cuando está en modo Responsive.

Las operaciones de entrada y salida se encuentran en la parte superior de la ventana, concretamente en la cabecera de la página. Si el usuario abre una imagen con el lienzo pintado, le pedirá iniciar sesión y luego le abrirá la imagen colocándola en el lienzo. Las imágenes que el usuario abre, se adaptarán en tamaño a las proporciones del lienzo, siendo imposible abrir una imagen en las proporciones de resolución originales.

Si el usuario ya había iniciado sesión y había pintado el lienzo, entonces le pedirá guardar los cambios antes de reemplazar el dibujo por la imagen que desea abrir. Si el usuario desea guardar los cambios, entonces aparecerá un cuadro de dialogo donde el usuario podrá el nombre y el formato de la imagen. Al usuario se le permite guardar los cambios en el servidor,

o descargar la imagen a ordenador local, no obstante, siempre se guarda una copia en el servidor.

Una vez el usuario haya guardado la imagen, o si el usuario no tiene que guardar ninguna imagen (o modificación hecha en el lienzo), este puede abrir una imagen nueva que rellenará el lienzo con la nueva imagen seleccionada del servidor o del sistema local. Al abrir la imagen, se le pregunta al usuario si la imagen está ubicada en el servidor o hay que subirla del ordenador. La apertura de una nueva imagen implica la creación de un documento nuevo con la nueva imagen en pantalla sobrescribiendo cualquier cambio o forma incorporada con anterioridad.

En cualquier caso, cuando un usuario abre una imagen, esta se coloca en el lienzo ajustándose a las proporciones horizontales del lienzo y ajustando la altura verticalmente a la proporción horizontal, para no distorsionar la imagen.

En los dispositivos móviles el usuario encontrará la opción de abrir la imagen desde la cámara de fotos. Cuando el usuario seleccione esta opción, el móvil incluirá en el lienzo una foto del usuario.

Cuando un usuario toma una foto, puede seleccionar de la barra de herramientas el botón de filtro de imagen, que le abrirá un cuadro con una lista de filtros que puede aplicar.

El usuario siempre puede seleccionar la opción “Nuevo...” que aparece en la parte superior, el cual le permite a este empezar un nuevo trabajo preguntando siempre que desea hacer con los cambios realizados en el trabajo anterior.

El usuario que inicia sesión para utilizar la aplicación debe estar previamente registrado. El cuadro de dialogo que aparece al iniciar sesión muestra tres opciones: Iniciar sesión por Facebook, iniciar sesión introduciendo correo electrónico más contraseña, y por último el botón de registro.

Cuando un usuario se registra, recibirá en su correo un email informando sobre el registro y las opciones de cancelación de recepción de publicidad e información sobre los derechos que puede ejercer sobre sus datos. El usuario también puede modificar su información de registro a través del cuadro de configuración del perfil, donde puede actualizar su contraseña.

Para compartir información, el usuario solo tiene que pulsar el botón de compartir, que le mostrará un cuadro de dialogo con opciones relativas a ello.

Perfil de usuario

Para el uso de la aplicación se considera que, de manera general que los usuarios disponen de conocimientos básicos en informática y saben navegar por Internet. Este tipo de usuarios sabe manejar un navegador web, y hacer uso de los servicios básicos que se ofrecen en la red, como buscar recursos y/o tener acceso a redes sociales. No se requieren mayores conocimientos, aunque sí unas características mínimas tecnológicas, donde el usuario tenga disposición de un ordenador de sobremesa o portátil y/o un móvil o Tablet. Los usuarios que se conecten a través de móvil, necesitarán que este sea táctil y que dispongan de una resolución horizontal mínima de pantalla de 320px.

Además, se da por hecho que los usuarios ya están familiarizados con el uso de la herramienta Paint o cualquier programa de dibujo básico y, por tanto, conocen las operaciones básicas de entrada y salida, inserción de formas de dibujo y compartir en redes sociales.

Los usuarios son en su mayoría de nacionalidad española y desean retocar fotos para compartirlas por las redes sociales. Por otra parte, la desaparición de Paint en Windows, puede provocar que ciertos usuarios nostálgicos quieran continuar usando la herramienta y buscar un sustituto.

Usabilidad

La interfaz de la aplicación, sigue la línea gráfica basada en otros programas del mismo género para facilitar el aprendizaje. Aunque la aplicación requiera autenticación por los usuarios, se prioriza el uso de la misma; los usuarios pueden usar la aplicación sin necesidad de identificarse y la autenticación será necesaria en las situaciones en que el usuario quiera abrir o guardar una imagen.

Las formas de interacción dependerán del dispositivo desde el que el usuario se conecte. Cuando un usuario se conecta a través de un ordenador, usará el ratón como medio de interacción. En cambio, cuando los usuarios accedan a través de pantallas móviles o tabletas, se usará el dedo. Cuando un usuario necesite introducir texto, necesitará usar un teclado que, en el caso de los dispositivos móviles, será virtual.

Los usuarios siempre se encuentran ante la misma ventana, salvo cuando se identifiquen o compartan contenido en las redes sociales, en cuyo caso, aparecerá un popup embebido en la aplicación que permitirá la identificación o registro.

Ventana de identificación o registro [\[Ver imagen\]](#)

La interfaz está agrupada en 4 áreas, que son: cabecera, barra de herramientas, lienzo y pie de página. La cabecera muestra herramientas de uso estándar (abrir, guardar, compartir, ...), e información relativa al usuario del programa y de la propia aplicación, es decir, muestra el logotipo que identifica la aplicación y el nombre del usuario que haya iniciado sesión.

Cabecera de la aplicación [\[ver Imagen\]](#)

Además, el usuario podrá acceder a la configuración de su cuenta y también permitirá identificar si se ha iniciado o no sesión. La barra de herramientas, agrupa todas las herramientas que el usuario necesita para interactuar con el lienzo. El lienzo se hace reconocible por situarse en la parte central y por mantener semejanza con otros programas. En los ordenadores de escritorio, el usuario percibirá como el puntero del ratón cambia su forma para indicar que parte de la ventana se corresponde con el área de trabajo.

Forma del ratón al pasar por el lienzo [\[ver Imagen\]](#)

Por último, en el pie de página aparecerá información actualizada sobre la posición del puntero sobre el lienzo, herramienta seleccionada y otras características de información general.

Aspecto del pie página de la aplicación [\[ver imagen\]](#)

A lo largo del documento, se mantiene una coherencia visual donde se juega siempre con los colores violeta, azul y blanco. Los enlaces se muestran en blanco y sin subrayado, pero el usuario percibirá que puede interactuar con ellos al pasar el ratón sobre la etiqueta.

Seguridad

La aplicación permite guardar archivos en la nube y compartirlos. Esto se hace a través de un sistema de autenticación que permita separar los datos por usuarios usando su nombre de usuario y contraseña. Para implementar este sistema de autenticación se necesita desarrollar una base de datos que almacene información de forma segura. La aplicación implementa políticas de seguridad a dos niveles: evitar la usurpación de identidad y el acceso a datos sin autorización.

Las contraseñas se almacenan en la base de datos usando un mecanismo de encriptación MD5. Por otra parte, la aplicación asegurará en el formulario de registro, que los usuarios escriban una dirección email real.

A nivel técnico, la aplicación se desarrolla por versiones que se testean para comprobar su fiabilidad en cuanto a fallos y rendimiento.

Tests

Crear la aplicación en modo responsive entraña mucha dificultad, por el número de plataformas que existen en el mercado actualmente. Hay que tener en cuenta que la aplicación tiene que verse bien tanto en monitores de escritorio como en dispositivos móviles y tabletas. Además, cada uno de estos dispositivos trabajan a diferentes resoluciones y con diferentes navegadores.

Según la web [librosweb](#), cualquier persona que desarrolle un sitio web debe tener en cuenta que:

- Para tabletas en vertical o móviles en horizontal, la resolución mínima horizontal es de 768px.
- Para Tablet as en horizontal y escritorios normales, la resolución mínima horizontal es de 1024px
- Para escritorios muy anchos, la resolución horizontal mínima es de 1200px.

Teniendo en cuenta la información anterior y utilizando las herramientas responsive de los navegadores Chrome y Firefox, al final he ajustado la visualización de la siguiente manera:

```
<meta name="viewport" content="width=device-width, initial-scale=1" user-scalable="no">
<link rel="stylesheet" media="screen and (min-device-width: 800px)" type="text/css" href="./css/main-theme.css">
<link rel="stylesheet" media="screen and (min-device-width: 710px) and (max-device-width: 799px)" type="text/css" href="./css/tablet-landscape-theme.css">
<link rel="stylesheet" media="screen and (min-device-width: 320px) and (max-device-width: 709px)" type="text/css" href="./css/mobile-portrait-theme.css">
```

No obstante, la adaptación a los dispositivos móviles no se limita únicamente a cambiar las hojas de estilos CSS, sino que dependiendo del dispositivo hay que discriminar información no relevante, por motivos de espacio, y reubicar los elementos para que la experiencia del usuario se centre en el uso del lienzo. Demasiados botones, en una pantalla de tamaño reducido, pueden entorpecer el uso de la aplicación y por tanto, favorecer al abandono de la misma.

Así pues, tras haber creado los prototipos y haberlos desarrollado visualizándolos en los navegadores, de diferentes dispositivos, se ha ido depurando la aplicación a base de ir probando de forma iterativa. Esto ha sido así, porque además hay navegadores que no interpretan por igual el lenguaje HTML y CSS, y por tanto, con el fin de buscar la mayor compatibilidad posible con todos los sistemas, he tenido que ejecutar el código usando tres navegadores diferentes y tres plataformas diferentes. Los dispositivos usados han sido:

- Ordenador de sobremesa:
 - Resolución de 1920x1080
 - Navegadores: Edge, Chrome y Firefox
- Ordenador portátil:
 - Resolución: 1320x720
 - Resolución: 800x600
 - Navegadores: Edge, Chrome y Firefox

- M3vil Xiaomi (Portrait y Landscape)
 - Resolucio: 1920x1080
 - Navegadores: Edge, Chrome, Firefox y Xiaomi Navigator

Para realizar las pruebas, he subido el trabajo v3a ftp, a un espacio web personal:
www.galeonweb.es/web

Bugs

Dado que la aplicación aún está en desarrollo todavía dispone de errores. Los errores que nos encontramos al usar la aplicación son:

- La herramienta de color está mal representada. Actualmente, esta herramienta está representada por un cuentagotas cuando debería mostrar un icono de bote de pintura.
- La cámara web aún no funciona.
- La autenticación de usuarios aún no está implementada
- Cuando el usuario abre una imagen en modo responsive, esta no respeta las proporciones del lienzo y se abre de forma desproporcionada siendo imposible su correcta visualización.
- Hay herramientas que se ejecutan varias veces y provoca que el historial de acciones de 'Hacer y Deshacer' no funcionen correctamente.
- Al usar el modo responsive, si el usuario gira la pantalla, el lienzo no mantiene sus proporciones originales provocando un estiramiento de la imagen.

Versiones de la aplicación

La aplicación consta de dos versiones.

- **Versión Alpha:** Una primera versión pensada para trabajar en local y que solo implementa las operaciones de dibujo. Esta versión se desarrolló en la PAC2 y comprendía el desarrollo HTML y JavaScript de la aplicación. No dispone de backend, por lo que las operaciones de autenticación de usuario y guardado remoto de imágenes no están implementadas.
- **Versión Beta:** La segunda y última versión permite realizar dibujos y dispone de BackEnd. En esta versión los usuarios pueden autenticarse y guardar imágenes en remoto.
- **Versión 1.0.:** Versión totalmente operativa del programa, pero aún no ha sido implementada y se propone como mejora. En esta versión el usuario puede loguearse usando las redes sociales, y dispone de la posibilidad de recuperar la contraseña en caso de olvido. Además, las herramientas que faltaron por implementar en etapas anteriores (zoom, selección) ya quedan implementadas.

Requisitos de instalación

Fotoclick es una aplicación web Cliente / Servidor y por tanto necesita estar instalada en un servidor web. Las características del servidor son las siguientes:

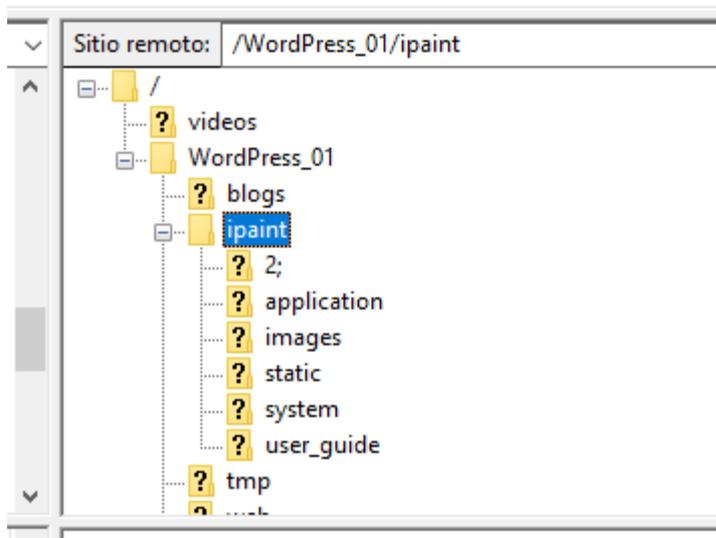
- **Servidor web seguro.** Es decir, para que la cámara funcione en el modo Responsive, el servidor debe tener acceso vía HTTPS y por tanto debe estar configurado para el acceso a conexiones seguras mediante certificado digital.
- **Base de datos (MySQL) + PHP.** El servidor debe implementar un servidor de base de datos en MySQL e incluir el módulo PHP para la interpretación de este lenguaje.
- **Servidor web.** En este sentido se puede usar cualquier servidor web (Apache, IIS, ...) aunque recomiendo Apache, ya que es el servidor donde se ha implementado.
- **CodeIgniter.** El servidor debe tener instalado el módulo CodeIgniter para el funcionamiento del Modelo-Vista-Controlador.
- **Servidor de correo saliente (SMTP) y entrante (POP3).** En realidad, el más importante es el servicio de correo saliente para que los usuarios puedan recibir el email de confirmación cuando se registren. El servidor de correo POP3 no es realmente importante, aunque si recomendable. El servidor de correo entrante permitiría a los usuarios poder comunicarse con la compañía.

Una vez disponemos de todos los servicios mencionados arriba, instalamos la aplicación copiándola en el servidor y re-ajustando algunos valores. Hecho esto, la aplicación debería funcionar en cualquier dispositivo móvil o sobremesa, pero siempre a través de un navegador web. No se precisan grandes requisitos técnicos en los dispositivos donde se ejecutan, salvo que, por supuesto dispongan de acceso a Internet y en el caso de los dispositivos móviles, dispongan de cámaras y dimensión mínima de pantalla 320 px horizontales.

En lo relativo a la formación y/o conocimientos necesarios para el uso de la aplicación, no se precisan conocimientos avanzados de informática, aunque si haber interactuado previamente con redes sociales y otros programas de dibujo. Básicamente, cualquier persona con conocimientos básicos en el uso del móvil y/o ordenador, no debería tener problemas para usar una herramienta de dibujo que sigue, por convenio, la misma disposición de herramientas que otras herramientas de dibujo básicas.

Instrucciones de instalación

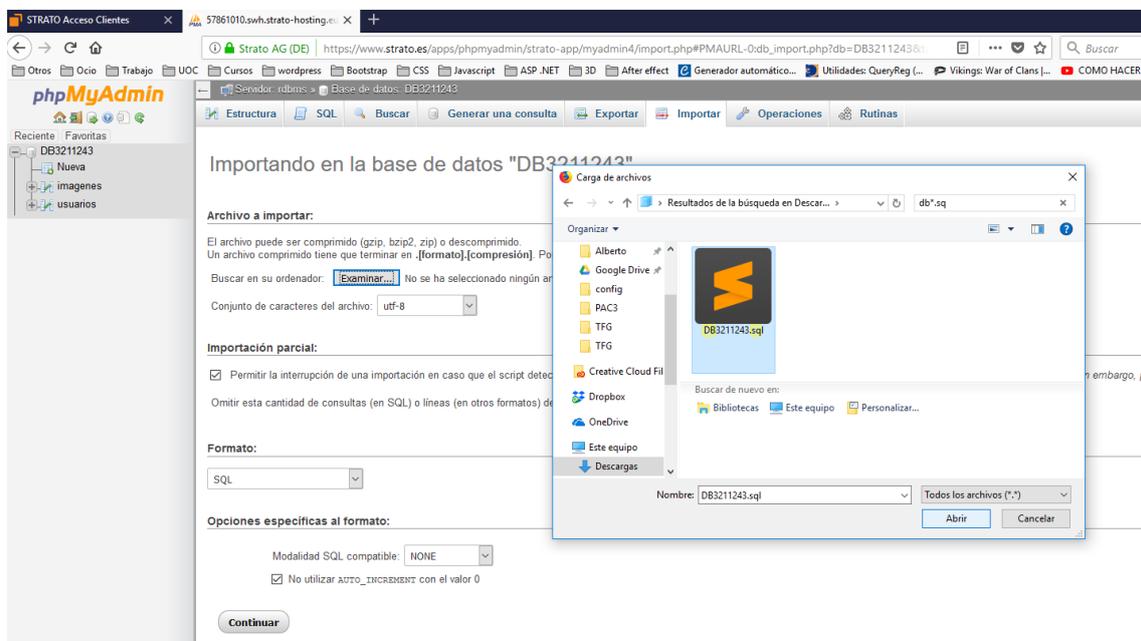
Se copia el directorio ipaint al servidor, tal y como muestra la siguiente imagen:



En los archivos index.php, header.php, y footer.php, se reemplazan las referencias a archivos actualizando el nombre de dominio. Ejemplo:

```
<script src="https://galeonweb.es/ipaint/static/javascript/filtros.js"></script>
<script src="https://galeonweb.es/ipaint/static/javascript/resolucion.js"></script>
<script src="https://galeonweb.es/ipaint/static/javascript/buttonActionsSetup.js"></script>
<script src="https://galeonweb.es/ipaint/static/javascript/formas.js"></script>
<script src="https://galeonweb.es/ipaint/static/javascript/transformaciones.js"></script>
<script src="https://galeonweb.es/ipaint/static/javascript/inicio.js"></script>
```

Se importa el código SQL al servidor de base de datos:



Si al importar la base de datos genera errores, entonces hay que copiar el código MySQL en generar una consulta y ejecutarlo, editando las opciones del código que entran en conflicto:

```

29 CREATE TABLE IF NOT EXISTS `imagenes` (
30   `ruta` varchar(255) NOT NULL,
31   `extension` varchar(3) NOT NULL,
32   `fecha` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
33   `usuario_fk` bigint(20) NOT NULL,
34   KEY `usuario_fk` (`usuario_fk`)
35 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
36
37 --
38 -- Estructura de tabla para la tabla `usuarios`
39 --
40
41 CREATE TABLE IF NOT EXISTS `usuarios` (
42   `idusuario` bigint(20) NOT NULL AUTO_INCREMENT,
43   `nombre` varchar(255) NOT NULL,
44   `email` varchar(255) NOT NULL,
45   `password` varchar(255) NOT NULL,
46   PRIMARY KEY (`idusuario`),
47   UNIQUE KEY `nombre` (`nombre`),
48   UNIQUE KEY `email` (`email`)
49 ) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
50
51 --
52 -- Filtros para la tabla `imagenes`
53 --
54 ALTER TABLE `imagenes`
55   ADD CONSTRAINT `imagenes_ibfk_1` FOREIGN KEY (`usuario_fk`) REFERENCES `usuarios` (`idusuario`);
56

```

Por último, debemos indicar a CodeIgniter, sobre que base de datos se va a trabajar. Para ello modificamos las siguientes líneas en el archivo config/database.php

```

76 $db['default'] = array(
77   'dsn' => '',
78   'hostname' => 'localhost',
79   'username' => 'root',
80   'password' => ' ',
81   'database' => 'ipaint',
82   'dbdriver' => 'mysql',
83   'dbprefix' => '',
84   'pconnect' => FALSE,
85   'db_debug' => (ENVIRONMENT !== 'production'),
86   'cache_on' => FALSE,
87   'cachedir' => '',
88   'char_set' => 'utf8',
89   'dbcollat' => 'utf8_general_ci',
90   'swap_pre' => '',
91   'encrypt' => FALSE,
92   'compress' => FALSE,
93   'stricton' => FALSE,
94   'failover' => array(),
95   'save_queries' => TRUE
96 );

```

Si todo ha ido bien, la aplicación debería quedar perfectamente instalada y lista para su funcionamiento.

Instrucciones de uso

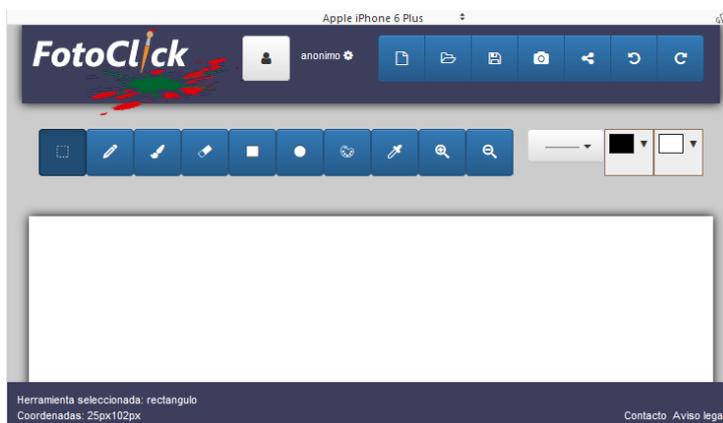
La aplicación se abre desde el navegador. Para que funcione el usuario debe teclear en la barra de direcciones del navegador la siguiente URL:

<http://galeonweb.es/ipaint>

El usuario puede pintar usando la barra de herramientas de dibujo disponible en horizontal o vertical, dependiendo del tipo de dispositivo desde el que lo visualiza.



Modo escritorio



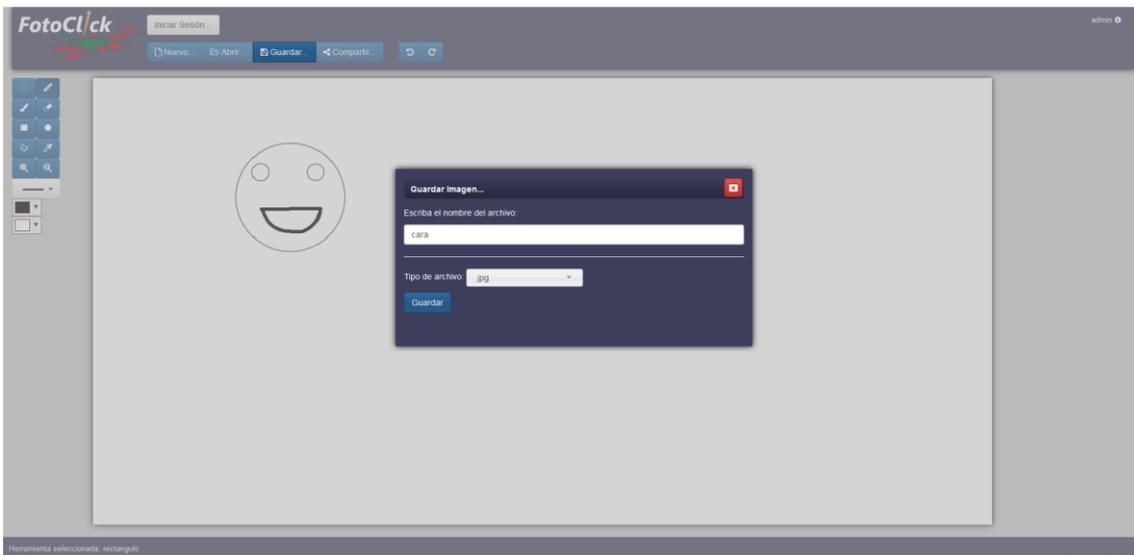
Modo Tablet

Para que un usuario pueda guardar imágenes, necesita haber iniciado sesión. Esto lo realiza a través del botón de inicio de sesión o :



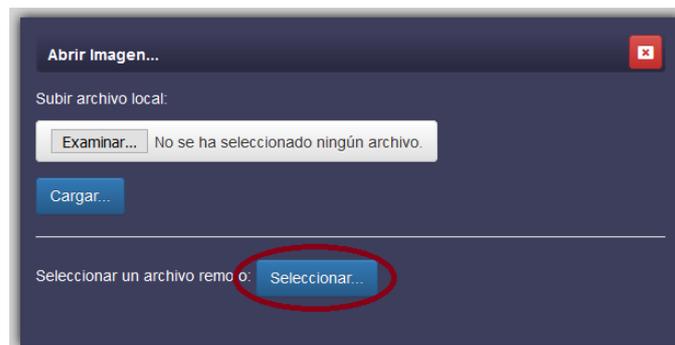
Inicio de sesión

En el cuadro de diálogo de inicio de sesión, los usuarios nuevos pueden registrarse facilitando su email, nombre de usuario y contraseña. Hecho esto, cuando el usuario haya realizado un dibujo, puede guardarlo en el servidor remoto:



Guardado de imagen

Cuando el usuario quiere recuperar la imagen usará el botón abrir. Este cuadro permite abrir archivos locales o remotos. Para abrir un archivo local, pulsa en examinar y luego en cargar. Para abrir un archivo remoto, el usuario pulsará sobre el botón seleccionar y se le mostrarán las imágenes guardadas en el servidor y ordenadas por orden cronológico, listas para que el usuario seleccione la que desea abrir:



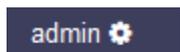
Cuadro de dialogo abrir



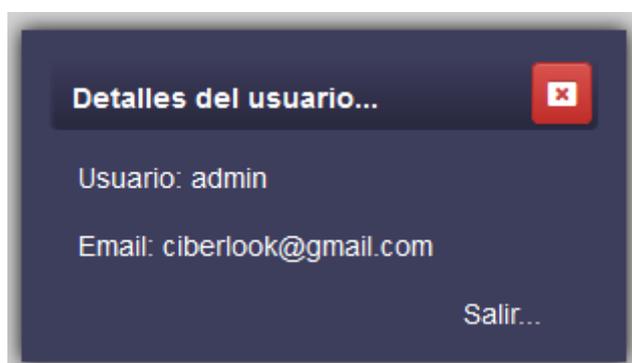
Lista de imágenes

Además, el usuario también puede descargarse la imagen al dispositivo o eliminarla de forma definitiva del servidor local.

Una vez el usuario ha terminado de usar la aplicación, puede salir de ella cerrando sesión, pulsando en el enlace ubicado en la esquina superior derecha:



Este enlace muestra una ventana que permite al usuario cerrar sesión:



Cerrar sesión

Bugs

A continuación, detallo los errores detectados en la aplicación:

- Cuando un usuario crea una imagen que resulta ser muy pesada, al guardarla no aparece un preloader que indique al usuario el progreso del guardado. La aplicación parece que se cuelga o no hace nada. En realidad, está trabajando, lo que ocurre es que al no disponer de un preloader se desorienta al usuario.
- Ciertas imágenes en PNG, se guardan con el fondo negro.
- Cuando se guarda una imagen con el mismo nombre, el programa no avisa de que la imagen ya existe. Se desconoce si la imagen queda sobrescrita o simplemente la aplicación no hace nada.
- Las herramientas de zoom no funcionan.
- La herramienta selección no hace nada.
- La herramienta de cámara funciona, pero no captura la imagen para que el usuario pueda retocarla.
- La herramienta deshacer y rehacer funcionan bien, pero en algunos casos hay que pulsar más veces que el número de operaciones realizadas.
- El usuario no puede iniciar sesión usando su cuenta de Facebook
- El usuario no puede compartir sus imágenes a través de Facebook

Proyección de futuro

La aplicación ha sido terminada en cuanto a sus características básicas. Permite pintar y autenticar usuarios. Además, las imágenes pueden guardarse en remoto y ser descargadas o borradas bajo demanda de los usuarios. No obstante, hay ciertas operaciones que necesitan mejorarse y se pueden proponer como mejoras de futuro.

- Los usuarios que olviden su contraseña, no pueden iniciar sesión y no disponen de mecanismos de recuperación de contraseña. Sería interesante proponer esta funcionalidad como mejora de futuro.
- Sería interesante incluir un preloader que permita al usuario cuantificar la duración de las operaciones llevadas a cabo. Ciertas operaciones, en especial las de entrada y salida pueden requerir tiempo y es por ello que es importante que el usuario tenga una idea del progreso de las mismas.
- Terminar las herramientas de dibujo que han faltado por desarrollar. Las herramientas zoom y selección son indispensables para el tratamiento básico de las imágenes. Además, hay herramientas que se echan de menos como la herramienta texto o herramientas poligonales como la elipse, estrella o trazos poligonales personalizados.

Presupuesto

A continuación, se presenta un presupuesto no definitivo sobre el proyecto a realizar, teniendo en cuenta los siguientes aspectos:

- Coste de alojamiento web: 50€/anuales
- Coste de mano de obra: 12€ hora
- Tiempo dedicado al proyecto: 80 horas / mes
- Tiempo dedicado a la gestión: 10 horas / mes
- Personal involucrado en el proyecto: 1 Persona

De todo lo anterior, estimo que el coste del proyecto es de: $(4.16€ + 1080€) * 1 = 1084.16€$

Para decidir el precio por hora se han consultado varias webs, y cada una de ellas da una cantidad muy diferente. Al final, la que más fiable y sensata me ha parecido es la siguiente: <https://www.upwork.com/hiring/for-clients/cost-to-hire-front-end-web-developer/>, donde se muestra una tabla que ofrece diferentes precios según el tipo de proyecto. Además, teniendo en cuenta de que mi condición de programador está en la fase de “Junior”, me parece justo empezar con el precio mínimo.

Dado que por falta de tiempo y conocimientos no he sido capaz de finalizar totalmente el proyecto, considero que habría sido necesario recurrir a personal especializado para terminarlo. Teniendo en cuenta esto, el proyecto incrementaría el precio:

Coste de una persona: 15€/hora * 20 horas= **300€/Persona**

Coste total del proyecto: $1084.16€ + 300€ = 1384.16€$

Análisis de mercado

Análisis de la competencia

Haciendo una búsqueda en Google sobre variantes del programa Paint online, existen las siguientes páginas que ofrecen este servicio:

- **Yuidraw:** Youidraw.com/apps/painter
 - Es una aplicación, que parece estar implementada en flash, pero que no está pensada para trabajar en modo responsive. No utiliza las ventajas de la movilidad.
- **Galactic.ink:** Galactic.ink/sketchpad
 - También es una aplicación de dibujo similar a Paint, pero tampoco está pensada para su uso en modo responsive.
- **iPaint:** Jswidget.com/index-ipaint.html
 - Esta aplicación se ofrece de forma gratuita en su versión escritorio. En su versión responsive, hay que pagar.

Público objetivo

Se diferencian dos tipos de clientes:

- Clientes que usan una plataforma de sobremesa con sistema operativo diferente a Windows y desean tener un programa de dibujo básico.
- Clientes que desean hacer una foto desde el móvil y hacer algún retoque fotográfico, además de incluir algún dibujo y compartirlo en las redes sociales.

Tipos de consumidores

- Clientes conservadores que son reacios a usar una aplicación que desconocen. La aplicación debe ser fácilmente identificable como una aplicación de dibujo, y los consumidores deben identificarla fácilmente como tal.
- Clientes que usan las redes sociales constantemente y continuamente hacen fotos y las cuelgan en Facebook o Instagram. Además, están acostumbrados a poner fotos que lleven filtros fotográficos.
- Clientes que se sienten atraídos por lo que comparten amigos en las redes sociales. Desconocen la aplicación, pero se mueven por el impulso social de hacer y compartir lo mismo que sus contactos.
- Clientes que quieren usar una aplicación de dibujo básica y de fácil manejo. No disponen de una aplicación instalada en su ordenador para tal fin.

Competencia indirecta

Existen multitud de programas que, sin ser programas específicos de dibujo, constituyen una amenaza para la salida al mercado de la aplicación. Por ejemplo, Instagram permite realizar fotos y aplicar instantáneamente filtros. Facebook, permite hacer fotos y aplicarle filtros e incluir formas básicas.

Análisis DAFO

	Internos	Externos
Negativos	<i>Debilidades</i>	<i>Amenazas</i>
	<ul style="list-style-type: none">• Falta de experiencia en el	<ul style="list-style-type: none">• Alta competitividad

- | | |
|---|---|
| <ul style="list-style-type: none"> desarrollo de la plataforma • Problemas relativos a la protección de datos y seguridad. • Necesidad de conocimientos en SEO. • La app no ofrece herramientas que seduzcan al público objetivo. • No es una aplicación nativa • Alojamiento en servidor remoto conlleva coste | <ul style="list-style-type: none"> indirecta. • Aplicación no genera confianza. • Actualizaciones en los navegadores pueden afectar al rendimiento de la app. • Malos comentarios por parte de los clientes • Problemas legales en cuanto a la gestión de datos y la privacidad. |
|---|---|

Positivos

Fortalezas

Oportunidades

- | | |
|--|---|
| <ul style="list-style-type: none"> • No necesita instalación • Aplicación es gratuita • No existen muchas aplicaciones iguales. • La aplicación es responsive • El coste en servidor web es barato y no requiere mucho mantenimiento. | <ul style="list-style-type: none"> • Permite llegar a un gran número de usuarios a través de redes sociales • Se puede crear una versión extendida que permita su venta |
|--|---|

Viabilidad

Alcance del proyecto

La aplicación se define como una aplicación de dibujo básico tipo Paint, pero en formato web. Paint es una herramienta que ha evolucionado con el tiempo y que en su versión más moderna ofrece herramientas más sofisticadas. La aplicación se desarrolla imitando la versión antigua de Paint, donde se insertan formas básicas con un color y estilo de líneas determinado.



Imagen protegida por derechos de autor.

Autor: Jonathan Eccles

Fuente: <http://newscult.com/microsoft-might-abandon-ms-paint-and-we-are-all-heartbroken/>

La aplicación se realiza en forma de web app, con código HTML5 y javascript y se ofrecerá en modo responsive y escritorio. La resolución mínima horizontal a la que funcionará es de 320px. Cualquier resolución inferior a esa, no se controla.

En el modo responsive, la aplicación ofrecerá la opción de hacer fotos que puedan retocarse y compartir. Pero en el modo escritorio, únicamente se ofrece la forma habitual de trabajo al estilo Paint. La aplicación no ofrecerá opciones de geoposicionamiento ni acceso a la agenda del teléfono, pero si acceso a la cámara para poder realizar fotos y enviarlas a los contactos.

Fotoclick se ofrece de forma gratuita y sus ingresos se obtienen a partir de la publicidad incluida en la misma y de la venta de datos a terceros.

Análisis de situación

El desarrollo de la aplicación es complejo por el número de dispositivos para el que se diseña y la alta competencia indirecta existente en el mercado. La falta de experiencia en el sector tanto a nivel tecnológico como a nivel comercial, puede representar un problema que se solucionaría incorporando personal cualificado.

Además, el almacenamiento de información privada y el tratamiento de los mismos, requiere que la aplicación informe a los usuarios de las políticas de protección de datos y del uso de estos.

Actualmente la aplicación se encuentra en una fase temprana de su desarrollo. La aplicación aún no es funcional y solo dispone de la interfaz de trabajo que da una idea de cómo va a ser su funcionamiento final. Falta completar los elementos de la interfaz gráfica, tratando de cumplir la planificación propuesta en puntos anteriores e incluir el resto de elementos integrantes relativos al Backend.

Cuando la aplicación finalice, se deberá publicar en un servidor web con soporte PHP y MySQL. El coste del servicio web y el alojamiento es de 47,88€/año, según la web de www.strato.es. Esto incluye el dominio, el alojamiento y resto de herramientas webs que entre otras cosas

permiten medir las estadísticas del sitio web y ayudan con la gestión del mismo. Concretamente los datos estadísticos, permitirá hacer un seguimiento del éxito del proyecto y favorecerá en el empuje financiero de este.

Por otra parte, existen diferentes modelos de fijación de precios relativos a la publicidad incluida en la web:

- **Costo por acción:** Se suele pagar alrededor de unos 50€ cada vez que un cliente indaga sobre un producto anunciado en la web.
- **Costo por clic:** El anunciante paga por clic. No existe un precio fijado, pero el mínimo está en 0.01€
- **Tarifa plana:** El anunciante paga un precio por publicarse en la web. Esto dependerá del número de visitas que se tengan, porque los anunciantes pagarán más o menos dependiendo de la popularidad del sitio web.

Teniendo en cuenta lo anterior, si elegimos el costo por clic, como fuente de financiación:

Se necesitaría 5000 usuarios (13 usuarios al día) que visitaran la web anualmente e hicieran clic en un anuncio para generar ingresos. Dado que representa un número elevado, para conseguir mayor probabilidad de éxito, se puede utilizar las cuentas de correo de los usuarios para enviar publicidad de empresas anunciantes, las cuales pagarían una tarifa plana 15€/mes por el envío de publicidad.

Además de lo anterior, si fuera necesario, se puede realizar una versión profesional de la aplicación en el futuro que conlleve un coste de 30€.

Según lo anterior, suponiendo que disponemos de 5000 visitas/anuales de los que obtenemos 1000 clicks y una cartera de 10 empresas:

Proyecto web fotoclick		
	Ingresos	Costes
Año 0		47,88€
Año 1	10 + 1800=1810€	228,88€
Año 2	20 + 3600=3620€	409,88€
Año 3	50 + 5000=5050€	552.88

- Se estima un coste de gestión del 10%

$$VAN = -47.88 - \frac{(1581,12)}{(1+0.1)^1} + \frac{(3210.12)}{(1+0.1)^2} + \frac{(4497.12)}{(1+0.1)^3} = 1437,38 + 2652,99 + 3378,75 = \mathbf{7469,12}$$

El proyecto es viable si presenta un VAN mayor que 0.

Conclusión:

Creo que la creación de este proyecto me ha ayudado a comprender que el no conocer alguna tecnología de programación no es una excusa para no implementar cualquier proyecto que me proponga. Es cierto, que no he conseguido terminar del todo el TFG debido a mi inexperiencia, falta de conocimientos y disponibilidad, no obstante, considero positiva mi participación en esta empresa, por varios motivos:

- La experiencia es un grado, y este proyecto me sirve de precedente para saber cuantificar con mayor realismo cuánto cuesta hacer algo, tanto a nivel económico como a nivel de recursos humanos. Además, me sitúa en una posición más realista y me permite planificar mejor futuros proyectos que me plantee realizar.
- A lo largo del proyecto he ido usando diferentes disciplinas que he estudiado en la carrera. Arquitectura de la información, programación y programación web avanzada, seguridad en servidores, tratamiento de imágenes, guionización, gestión de empresas, etc... Creo que este ha sido un proyecto perfecto para poner en práctica la mayor parte de las asignaturas vistas en el grado y supone una fantástica oportunidad para medir mis conocimientos.
- A nivel curricular, creo que el TFG me ha proporcionado nuevos conocimientos y habilidades que me hacen sentir más seguro de mí mismo. Creo que me ayudará positivamente para la mejora de mis méritos profesionales.
- He disfrutado con cada minuto invertido en el proyecto. Creo que la aplicación elegida era la adecuada y perfecta para disfrutar de la asignatura ya que daba mucho juego. Lamento mucho no haberlo terminado por falta de tiempo.

Tras encontrar el TFG una experiencia tan gratificante, estoy deseando poner en práctica mis habilidades como graduado multimedia (y como gestor de proyectos multimedia), para la creación de proyectos webs personales.

Anexos

Anexo 1. Entregables del proyecto

El proyecto consta de un archivo comprimido que consta de los siguientes documentos:

- DOCUMENTACIÓN
 - **Memoria TFG.** Es este documento y aparece como un documento Word con el nombre de "memoria_TFG.docx".
 - **Autoinforme de evaluación**
- PROYECTO
 - Contiene la **carpeta ipaint** que es el programa listo para su instalación en servidor web. En esta carpeta aparecen los archivos fuentes, imágenes, base de datos, etc...
- PRESENTACIONES
 - **Presentación del proyecto escrita.** Documento en formato Word dirigido a un público no especializado y orientada a la difusión del proyecto.
 - **Presentación del proyecto en formato video.** Video dirigido a al tribunal de evaluación y que cubre los diferentes temas pedidos en la práctica final.

Anexo 2: Código fuente

Código HTML para definir el aspecto dependiendo del dispositivo

```
<link rel="stylesheet" media="screen and (min-device-width: 1024px)" type="text/css" href="./css/main-theme.css">
<link rel="stylesheet" media="screen and (max-device-width: 799px)" type="text/css" href="./css/small-tablet-portrait-theme.css">
<link rel="stylesheet" media="screen and (min-device-width: 320px) and (max-device-width: 599px)" type="text/css" href="./css/mobile-portrait-theme.css">
```

Código HTML que configura las hojas de estilo de Bootstrap

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="sha384-BVYiSiFeK1dGmJRAkycuHAHRg320mUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
crossorigin="anonymous">
<!-- Optional theme -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-theme.min.css" integrity="sha384-rHyoN1iRsVXV4nD0JutlnGaslCJuC7uwjduW9SVrLvRYooPp2bWYgmgJQIXw1/Sp"
crossorigin="anonymous">
```

Creación de un círculo

```
//Comenzamos aqui
function init(){
    //Inicializamos el canvas y su contexto
    canvas=document.getElementById("lienzo");
    canvas.setAttribute('width',800);
    canvas.setAttribute('height',600);
    if(typeof G_vmlCanvasManager != 'undefined') {
        canvas =
G_vmlCanvasManager.initElement(canvas);
```

```

    }
    ctx=canvas.getContext("2d");

    //Creamos un circulo indicando 3 parametros: el tipo de
    linea, el grosor y el color
    var circulo=new Circulo("round", 5,"#000000");

    //Presionamos el botón izquierdo del ratón
    $('#lienzo').mousedown(function(e){
        //Cuando pulse en el lienzo actualizamos la variable
        pulsado y obtendremos las coordenadas del raton
        pulsado=true;
        var posX=e.pageX-this.offsetLeft;
        var posY=e.pageY-this.offsetTop;

        //Defino el origen y comienzo el circulo
        var origen=new Punto();
        origen.definirPunto(posX, posY, "round", "thick",
        "#000000");
        circulo.setOrigen(origen);

        //El segundo parametro indica que se represente el circulo
        que hemos definido
        circulo.definirCirculo(origen,true);

    });

    //Movemos el ratón
    $('#lienzo').mousemove(function(e){
        var posX=e.pageX-this.offsetLeft;
        var posY=e.pageY-this.offsetTop;
        var vertice=new Punto();
        vertice.definirPunto(posX, posY, "round", "thick",
        "#000000");

        //Evito que se muestren circulos concentricos, es decir,
        refresco pantalla
        canvas.width=canvas.width;
        //Si mientras movemos el ratón, el botón se mantiene
        pulsado, definimos el circulo a partir de su vértice

        if (pulsado){
            ctx.beginPath();
            var radio=Math.sqrt(Math.pow((vertice.getX()-
            circulo.getOrigen().getX()),2)+Math.pow(
            (vertice.getY()-circulo.getOrigen().getY()),2));

            ctx.arc(circulo.getOrigen().getX(),
            circulo.getOrigen().getY(),radio,0,2*Math.PI);

            ctx.lineWidth=6;
            ctx.closePath();
            ctx.stroke();
        }
    });

```

```

    });

    $('#lienzo').mouseup(function(e){
        pulsado=false; //Indicamos que hemos soltado el ratón
    });

    $('#lienzo').mouseleave(function(e){
        pulsado=false; //Indicamos que hemos soltado el
ratón
    });
}

```

Creación de una forma libre

```

//Una forma se compone de puntos
function Forma(tipo, grosor, color){
    var lista_de_puntos = new Array();
    var tipo=tipo;
    var grosor=grosor;
    var color=color;

    //La forma se define por las posiciones y el estado. Este
último determina si la forma está acabada
    this.definirForma= function(x, y, estado){
        lista_de_puntos.push([x, y, estado]);
    }

    this.limpiarForma = function (){
        lista_de_puntos.length=0;
    }

    this.mostrarForma= function(){
        for(var i=0; i < lista_de_puntos.length; i++)
        {
            ctx.strokeStyle=color;
            ctx.lineJoin=tipo;
            ctx.lineWidth=grosor;
            ctx.beginPath();
            //Si no estamos al principio del array y se ha
            pulsado el ratón pintamos el punto
            if(lista_de_puntos[i][2] && i){
                //Entramos aquí si pulsamos y arrastramos
                ctx.moveTo(lista_de_puntos[i-1][0],
                    lista_de_puntos[i-1][1]);
            }else{
                //Entramos aquí si pulsamos y soltamos
                ctx.moveTo(lista_de_puntos[i][0],
                    lista_de_puntos[i][1]);
            }
            ctx.lineTo(lista_de_puntos[i][0],
                lista_de_puntos[i][1]);
            ctx.closePath();
            ctx.stroke();
        }
    }
}

```

```

    }
}

//Comenzamos aqui
function init(){
    //Inicializamos el canvas y su contexto
    canvas=document.getElementById("lienzo");
    canvas.setAttribute('width',800);
    canvas.setAttribute('height',600);
    if(typeof G_vmlCanvasManager != 'undefined') {
        canvas = G_vmlCanvasManager.initElement(canvas);
    }

    ctx=canvas.getContext("2d");

    //Creamos una forma
    var forma=new Forma("round", 5,"#000000");

    //Presionamos el botón izquierdo del ratón
    $('#lienzo').mousedown(function(e){
        //Cuando pulse en el lienzo actualizamos la variable
        pulsado y obtendremos las coordenadas del raton
        pulsado=true;
        var posX=e.pageX-this.offsetLeft;
        var posY=e.pageY-this.offsetTop;

        //Defino el origen y comienzo el circulo

        forma.definirForma(posX,posY,false);
        forma.mostrarForma();

    });

    //Movemos el ratón
    $('#lienzo').mousemove(function(e){
        var posX=e.pageX-this.offsetLeft;
        var posY=e.pageY-this.offsetTop;

        //Evito que se muestren circulos concentricos, es
        decir, refresco pantalla
        //canvas.width=canvas.width;
        //Si mientras movemos el ratón, el botón se mantiene
        pulsado, definimos el ciruculo a partir de su
        vertice
        if (pulsado){
            forma.definirForma(posX,posY,false);
            forma.mostrarForma();
        }
    });

    $('#lienzo').mouseup(function(e){
        var posX=e.pageX-this.offsetLeft;
        var posY=e.pageY-this.offsetTop;
        pulsado=false; //Indicamos que hemos soltado el
        ratón
    });
}

```

```

        forma.definirForma(posX,posY,true);

    });

    $('#lienzo').mouseleave(function(e){
        pulsado=false; //Indicamos que hemos soltado el
                        ratón
    });
    forma.mostrarForma();
}

```

Código HTML de la barra vertical

```

<div id="barra_vertical">
    <ul id="barra_herramientas">
        <li>
            <button type="button" class="btn btn-primary" data-
toggle="button"><span class="icon-seleccionar"></span></button>
            <button type="button" class="btn btn-primary" data-
toggle="button"><span class="icon-pencil"></span></button>
        </li>
        <li>
            <button type="button" class="btn btn-primary" data-
toggle="button"><span class="icon-paint-brush"></span></button>
            <button type="button" class="btn btn-primary" data-
toggle="button"><span class="icon-goma"></span></button>
        </li>
        <li>
            <button type="button" class="btn btn-primary" data-
toggle="button"><span class="icon-cuadrado"><span
class="path1"></span><span class="path2"></span></span></button>
            <button type="button" class="btn btn-primary" data-
toggle="button"><span class="icon-circulo"><span
class="path1"></span><span class="path2"></span></span></button>
        </li>
        <li>
            <button type="button" class="btn btn-primary"
data-toggle="button"><span class="icon-colores"></span></button>
            <button type="button" class="btn btn-primary"
data-toggle="button"><span class="icon-eyedropper"></span></button>
        </li>
        <li>
            <button type="button" class="btn btn-primary"
data-toggle="button"><span class="icon-zoom-in"></span></button>
            <button type="button" class="btn btn-primary"
data-toggle="button"><span class="icon-zoom-out"></span></button>
        </li>
    </li>
    </ul>

```

Control de la resolución de pantalla mediante JavaScript

```

$(document).ready(function() {
    if (screen.width<=799 && screen.width>599) { //Entro en modo
responsive

```

```

        showLandscapeScreen();
    } else if (screen.width<=599) {
        showPortraitScreen();
    } else if (screen>=800) {
        if ($("#mask").hasClass("oculto"))
            $("#mask").removeClass("oculto");
    }

    //Controlamos la orientación del dispositivo
    $( window ).on( "orientationchange", function( event ) {
        if (event.orientation=="landscape" &&
(screen.width<=799 && screen.width>599) )
            showLandscapeScreen();

        if (event.orientation=="portrait" &&
(screen.width<=599))
            showPortraitScreen();
        commandSetupControllers();
    });

    //Controlamos el comportamiento de las herramientas
    commandSetupControllers();

    //Desabilito el mensaje "loading"
    $( document ).on( "mobileinit", function() {
$.mobile.loader.prototype.options.disabled = true;
    });
    // As submitted by @Aras
$.mobile.loading( "hide" );
    // (or presumably as submitted by @Pnct)
$.mobile.loading().hide();
});

```

Disposición de las librerías usadas en cabecera HTML. [Extracto de código].

```

<link rel="stylesheet" media="screen and (min-device-width: 800px)"
type="text/css" href="./css/main-theme.css">
    <link rel="stylesheet" media="screen and (min-device-width: 710px)
and (max-device-width: 799px)" type="text/css" href="./css/tablet-
landscape-theme.css">
    <link rel="stylesheet" media="screen and (min-device-width: 320px)
and (max-device-width: 709px)" type="text/css" href="./css/mobile-
portrait-theme.css">

    <link rel="stylesheet" href="./font-awesome-4.7.0/css/font-
awesome.min.css">
    <link rel="stylesheet" type="text/css" href="./css/style-
icomoon.css">
    <link rel="stylesheet" type="text/css" href="./bgrins-spectrum-
98454b5/spectrum.css">

```

```

    <!-- Latest compiled and minified CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.mi
n.css" integrity="sha384-
BVYiISIFeK1dGmJRAkycuHAHRg320mUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
crossorigin="anonymous">

    <!-- Optional theme -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-
theme.min.css" integrity="sha384-
rHyoN1iRsVXV4nD0JutlNGaslCJuC7uwjduW9SVrLvRYooPp2bWYgmgJQIXw1/Sp"
crossorigin="anonymous">

    <link rel="stylesheet"
href="https://code.jquery.com/ui/1.12.1/themes/smoothness/jquery-
ui.css">

<script src="https://code.jquery.com/jquery-1.12.4.min.js"
    integrity="sha256-
ZosEbRLbNQzLpnKIkEdrPv7l0y9C27hHQ+Xp8a4MxAQ="
    crossorigin="anonymous"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.
js" integrity="sha384-
Tc5Iqib027qvyjSMfHjOMaLkfuWvxZxUPnCJA712mCWNIpG9mGCD8wGNicPD7Txa"
crossorigin="anonymous"></script>
    <script src="./bgrins-spectrum-98454b5/spectrum.js"></script>
<script src="https://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.min.js"
integrity="sha256-VazP97ZCwtekAsvgPBSUwPFKdrwD3unUfSGVYrahUqU="
crossorigin="anonymous"></script>
    <!-- Latest compiled and minified JavaScript -->
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.5.16/p5.js"></scri
pt>
    <script language="javascript" type="text/javascript"
src="./javascript/p5.dom.js"></script>
    <script src="./javascript/filtros.js"></script>
    <script src="./javascript/resolucion.js"></script>
    <script src="./javascript/buttonActionsSetup.js"></script>
    <script src="./javascript/formas.js"></script>
    <script src="./javascript/transformaciones.js"></script>
    <script src="./javascript/inicio.js"></script>

```

Método mostrarForma()

```

this.mostrarForma= function(){
    // Clear the entire canvas
    if (herramienta!="lapiz" && herramienta!="goma")
        lista_de_puntos=[];

```

```

    ctx.clearRect(0, 0, canvas.width, canvas.height); //limpio la
pantalla

    //ctx.drawImage(new
Image(forma.getImage()),canvas.width,canvas.height);

    switch (herramienta) {
        case "seleccion":
            //imgData=imageAux; //Actualizo la propiedad
imgData, que siempre contiene un pantallazo inmediatamente anterior
            if (imgData!==undefined)
                ctx.putImageData(imgData,0, 0); // vuelvo a
colocar la pantalla
            else {

                imgData=ctx.getImageData(0,0,canvas.width,canvas.height);
                ctx.putImageData(imgData,0, 0);
            }
            ctx.clearRect(0, 0, canvas.width, canvas.height);
            ctx.putImageData(imgData,0, 0);
            ctx.strokeStyle="rgba(200,0,0,0.3)";
            ctx.fillStyle="rgba(200,0,0,0.3)";
            ctx.lineJoin="miter";
            ctx.lineWidth=1;
            ctx.beginPath();
            //hago el rectangulo de seleccion
            ctx.fillRect(x,y,destinoX-x,destinoY-y);
            ctx.closePath();
            ctx.stroke();

            break;
        case "lapis":
        case "goma":
            //Compruebo si hay alguna seleccion y la borro
            if (destinoX-x!=0 || destinoY-y!=0){
                ctx.putImageData(imgData,0, 0);
                x=0;
                y=0;
                destinoX=0;
                destinoY=0;
            }
            if (imgData!==undefined)
                ctx.putImageData(imgData,0, 0); // vuelvo a
colocar la pantalla
            else {

                imgData=ctx.getImageData(0,0,canvas.width,canvas.height);
                ctx.putImageData(imgData,0, 0);
            }
            for(var i=0; i < lista_de_puntos.length; i++)
            {
                if (herramienta==="lapis")
                    ctx.strokeStyle='rgba('+color.r+',
'+color.g+', '+color.b+', '+color.a+')';

```

```

        else if (herramienta === "goma")
ctx.strokeStyle='rgba('+colorRelleno.r+', '+colorRelleno.g+',
'+colorRelleno.b+', '+colorRelleno.a+')';
        ctx.lineJoin=tipo;
        ctx.lineWidth=grosor;
        ctx.beginPath();
        //Si no estamos al principio del array y se ha pulsado el ratón
pintamos el punto
        if(lista_de_puntos[i][2] && i){
        //Entramos aqui si pulsamos y arrastramos
        ctx.moveTo(lista_de_puntos[i-1][0], lista_de_puntos[i-1][1]);
        }else{
        //Entramos aqui si pulsamos y soltamos
ctx.moveTo(lista_de_puntos[i][0], lista_de_puntos[i][1]);}
ctx.lineTo(lista_de_puntos[i][0], lista_de_puntos[i][1]);
ctx.closePath();
ctx.stroke();
        }

        break;
        case "circulo":
        if (imgData!==undefined)
        ctx.putImageData(imgData,0, 0); // vuelvo a colocar la
pantalla
        else {
        imgData=ctx.getImageData(0,0,canvas.width,canvas.height);
        ctx.putImageData(imgData,0, 0);
        }
        ctx.strokeStyle='rgba('+color.r+', '+color.g+',
'+color.b+', '+color.a+')';
        ctx.lineJoin=tipo;
        ctx.lineWidth=grosor;
        ctx.fillStyle='rgba('+colorRelleno.r+',
'+colorRelleno.g+', '+colorRelleno.b+', '+colorRelleno.a+')';
        ctx.beginPath();
        var radio=Math.sqrt(Math.pow((x-destinoX),2)+Math.pow((y-
destinoY),2));
        ctx.arc(x,y,radio,0,2*Math.PI);
        ctx.fill();
        ctx.closePath();
        ctx.stroke();
        break;
        case "rectangulo":
        if (imgData!==undefined)
        ctx.putImageData(imgData,0, 0); // vuelvo a colocar
la pantalla
        else {

        imgData=ctx.getImageData(0,0,canvas.width,canvas.height);
        ctx.putImageData(imgData,0, 0);
        }
        ctx.strokeStyle='rgba('+color.r+', '+color.g+',
'+color.b+', '+color.a+')';
        ctx.lineJoin=tipo;

```

```

        ctx.lineWidth=grosor;
        ctx.fillStyle='rgba('+colorRelleno.r+',
'+colorRelleno.g+', '+colorRelleno.b+', '+colorRelleno.a+')';
        ctx.beginPath();
        ctx.rect(x,y,destinoX-x,destinoY-y);
        ctx.fill();
        ctx.closePath();
        ctx.stroke();
        break;

        case "relleno":
            if (imgData!==undefined)
                ctx.putImageData(imgData,0, 0); // vuelvo a colocar
la pantalla
            else {

                imgData=ctx.getImageData(0,0,canvas.width,canvas.height);
                ctx.putImageData(imgData,0, 0);
            }
            //Selecciono el color del pixel con el ratón sobre el que
voy a pintar y lo descompongo en los colores RGB
            var cont=0; //variable para asegurar que el bucle no se
haga infinito
            var pixelStack=[[x,y]];
            var startR=ctx.getImageData(x,y,1,1).data[0];
            var startG=ctx.getImageData(x,y,1,1).data[1];
            var startB=ctx.getImageData(x,y,1,1).data[2];

            if (colorRelleno.r===startR && colorRelleno.g===startG &&
colorRelleno.b===startB){
                return; //Está tratando de rellenar con el mismo color
            }else {

                while (pixelStack.length) {
                    cont++;
                    var newPos, posX, posY ,pixelPos, reachLeft,
reachRight;
                    //El pixel elegido lo borro del array y comienzo a
operar con él obteniendo sus coordenadas x,y dentro del canvas
                    newPos=pixelStack.pop();
                    posX=newPos[0];
                    posY=newPos[1];

                    //Posicion del pixel en el array

                    pixelPos=(posY*imgData.width+posX)*4;

                    //Voy subiendo pixeles y comparando el color hasta
dar con uno que no coincida o llegar al tope del lienzo
                    //Al mismo tiempo voy
                    //traduciendo las coordenadas del pixel horizontal en el array
                    while (posY-- >= 0 &&
matchStartColor(pixelPos, startR, startG, startB)) {
                        pixelPos -= imgData.width*4;

```

```

    }
    pixelPos += imgData.width*4;
    ++posY;
    reachLeft=false;
    reachRight=false;
    //Voy bajando pixeles y comparando colores
    while (posY++ < imgData.height-1 && matchStartColor(pixelPos,
startR, startG, startB)) {
    //Coloreo el pixel
    colorPixel(pixelPos);

    //Desde el pixel vertical, compruebo dentro de los limites del
canvas
    //si horizontalmente el pixel de la izquierda coincide con el
color seleccionado. Además realizo esta operacion a medida que voy
bajando y coloreando verticalmente
    if (posX>0) {
    if (matchStartColor(pixelPos -4, startR, startG, startB)) {
        if (!reachLeft) {
            pixelStack.push([posX-1,posY]);
            reachLeft = true;
        }
    } else if (reachLeft) {
        reachLeft = false;
    }
    }
    //Si estamos horizontalmente dentro del lienzo (por la derecha)
    if (posX< imgData.width-1) {
        if (matchStartColor(pixelPos + 4, startR, startG, startB))
{
        if (!reachRight) {
            pixelStack.push([posX+1,posY]);
            reachRight=true;
        }
    } else if (reachRight) {
        reachRight=false;
    }
    }
    pixelPos += imgData.width*4;
    }}
    }
    ctx.putImageData(imgData,0, 0);

    break;
    }
}

```

Web prueba para aplicación de efectos.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

```

```

<title>Documento sin título</title>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.5.16/p5.js"></scri
pt>

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha256-
k2WSCIexGz0j3Euiig+TlR8gA0EmpJuc790EeY5L45g="
crossorigin="anonymous"></script>
<script language="javascript" type="text/javascript"
src="./javascript/p5.dom.js"></script> <script
src="./javascript/sketch.js"></script>

</head>

<body>
<div id="lienzo">
<canvas id="poo"></canvas>
</div>
<label for="efectosId" >Elija un efecto</label>
<select id="efectosId" >
<option value="THRESHOLD" selected>Limite</option>
<option value="GRAY">Escala de grises</option>
<option value="OPAQUE">Opacidad</option>
<option value="INVERT">Invertir</option>
<option value="POSTERIZE">Posterizar</option>
<option value="DILATE">Desenfoque Dilatar</option>
<option value="BLUR">Desenfoque blur</option>
<option value="ERODE">Desenfoque Erosionar</option>
</select>
</body>
</html>

```

Código de sckets.js

```

var imgData, $select;

$(document).ready(function(e) {
  $('#efectosId').on('change',function(){
    alert("hola mundo");
  });
});
function setup(){
  var $canvas=$("#poo");
  var ctx=$canvas[0].getContext("2d");
  var canvasProcessing=createCanvas($canvas.width(),
  $canvas.height()); //Creo el canvas de processing
  var ctxp=canvasProcessing.elt.getContext("2d");
  canvasProcessing.parent('lienzo'); //Coloco el canvas como hijo
del lienzo

  //Cojo el valor del select
  $select=$("#select");
  //Cargo una imagen

```

```

var img=new Image();
img.src="./imagenes/logo.png";
img.onload=function(){
    ctx.drawImage(img,0,0); //Coloco la imagen en el canvas
    //ctx.drawImage($canvas[0],0,0); //Esta linea copia la
imagen del canvas al canvas del processing
    imgData=ctx.getImageData(0,0,canvas.width,canvas.height);
//Extraigo su array
    ctx.putImageData(imgData,0,0); //Copio el array en otro
canvas,
        filter($select.val());
    };
    $select.on('change',mySelectEvent);
}

function mySelectEvent() {
    var $canvas=$("#poo");
    var ctx=$canvas[0].getContext("2d");
    var canvasProcessing=createCanvas($canvas.width(),
$canvas.height()); //Creo el canvas de processing
    var ctxp=canvasProcessing.elt.getContext("2d");
    canvasProcessing.parent('lienzo'); //Coloco el canvas como hijo
del lienzo

    //Cojo el valor del select
    $select=$("#select");
    //Cargo una imagen
    var img=new Image();
    img.src="./imagenes/logo.png";
    img.onload=function(){
        ctx.drawImage(img,0,0); //Coloco la imagen en el canvas
        //ctx.drawImage($canvas[0],0,0); //Esta linea copia la
imagen del canvas al canvas del processing
        imgData=ctx.getImageData(0,0,canvas.width,canvas.height);
//Extraigo su array
        ctx.putImageData(imgData,0,0); //Copio el array en otro
canvas,
            filter($select.val());
        };
        filter($select.val());
    }

function draw(){
    ellipse(50,50,80,80);
}

```

Uso de la API de Facebook

```

<script>
window.fbAsyncInit = function() {
    FB.init({
        appId      : '499307157094768',
        cookie     : true,
        xfbml      : true,
        version    : 'v2.10'
    });
}

```

```

});

FB.getLoginStatus(function(response) {
  if (response.status === 'connected') {
    // the user is logged in and has authenticated your
    // app, and response.authResponse supplies
    // the user's ID, a valid access token, a signed
    // request, and the time the access token
    // and signed request each expire
    var uid = response.authResponse.userID;
    var accessToken = response.authResponse.accessToken;
  } else if (response.status === 'not_authorized') {
    // the user is logged in to Facebook,
    // but has not authenticated your app

  } else {
    // the user isn't logged in to Facebook.
  }
});

FB.AppEvents.logPageView();
});

(function(d, s, id) {
  var js, fjs = d.getElementsByTagName(s)[0];
  if (d.getElementById(id)) return;
  js = d.createElement(s); js.id = id;
  js.src = 'https://connect.facebook.net/es_ES/sdk.js';
  fjs.parentNode.insertBefore(js, fjs);
})(document, 'script', 'facebook-jssdk');

//Obtengo la información del usuario, para luego usarla a conveniencia
function getInfo(){
  FB.api('/me','GET', {fields: 'first_name, last_name, name, id,
picture.width(150).height(150), email'}, function(response) {
    //Hacemos algo con la información.
    console.log(response);
  });
}
</script>

```

Implementación HTML de Spectrum.

```

<li>
  <input type="color" id="frontColor"/>
</li>
</ul>
<script type="text/javascript">
  $("#frontColor").spectrum({
    color:"#000",
    showPaletteOnly: true,
    togglePaletteOnly: true,
    togglePaletteMoreText: 'more',

```

```

togglePaletteLessText: 'less',
allowEmpty: true,

palette:[
["#000", "#444", "#666", "#999", "#ccc", "#eee", "#f3f3f3", "#fff"],
["#f00", "#f90", "#ff0", "#0f0", "#0ff", "#00f", "#90f", "#f0f"],
["#f4cccc", "#fce5cd", "#fff2cc", "#d9ead3", "#d0e0e3", "#cfe2f3", "#d
9d2e9", "#ead1dc"],

["#ea9999", "#f9cb9c", "#ffe599", "#b6d7a8", "#a2c4c9", "#9fc5e8", "#b
4a7d6", "#d5a6bd"],
["#e06666", "#f6b26b", "#ffd966", "#93c47d", "#76a5af", "#6fa8dc", "#8e7cc3"
, "#c27ba0"],
["#c00", "#e69138", "#f1c232", "#6aa84f", "#45818e", "#3d85c6", "#674ea7", "#
a64d79"],
["#900", "#b45f06", "#bf9000", "#38761d", "#134f5c", "#0b5394", "#351c75", "#
741b47"],
["#600", "#783f04", "#7f6000", "#274e13", "#0c343d", "#073763", "#20124d", "#
4c1130"]
]

});
$("#backColor").spectrum({
color: "#fff",
showPaletteOnly: true,
togglePaletteOnly: true,
togglePaletteMoreText: 'more',
togglePaletteLessText: 'less',
allowEmpty: true,
palette:[
["#000", "#444", "#666", "#999", "#ccc", "#eee", "#f3f3f3", "#fff"],

["#f00", "#f90", "#ff0", "#0f0", "#0ff", "#00f", "#90f", "#f0f"],

["#f4cccc", "#fce5cd", "#fff2cc", "#d9ead3", "#d0e0e3", "#cfe2f3", "#d9d2e9"
, "#ead1dc"],
["#ea9999", "#f9cb9c", "#ffe599", "#b6d7a8", "#a2c4c9", "#9fc5e8", "#b4a7d6"
, "#d5a6bd"],
["#e06666", "#f6b26b", "#ffd966", "#93c47d", "#76a5af", "#6fa8dc", "#8e7cc3"
, "#c27ba0"],
["#c00", "#e69138", "#f1c232", "#6aa84f", "#45818e", "#3d85c6", "#674ea7", "#
a64d79"],
["#900", "#b45f06", "#bf9000", "#38761d", "#134f5c", "#0b5394", "#351c75", "#
741b47"],
["#600", "#783f04", "#7f6000", "#274e13", "#0c343d", "#073763", "#20124d", "#
4c1130"]
]

});
</script>

```

Código del archivo filtros.js

```

function setup(){
//Selecciona el canvas principal y su contexto
var $canvas=$("#poo");
var ctx=$canvas[0].getContext("2d");

```

```

        //Creo el canvas donde aplicare el filtro y su contexto. Este
        canvas es auxiliar
        var canvasProcessing=createCanvas($canvas.width(),
        $canvas.height()); //Creo el canvas de processing
        var ctxp=canvasProcessing.elt.getContext("2d");
        canvasProcessing.parent('lienzo'); //Coloco el canvas como hijo
        del lienzo
        canvasProcessing.addClass("oculto"); //Oculto el lienzo auxiliar
    }

function mySelectEvent(val,img,n) {
    var imgDataAux,imgOrig;
    var $canvas=$("#poo");
    var ctx=$canvas[0].getContext("2d");
    //Creo el canvas de processing
    var canvasProcessing=createCanvas($canvas.width(),
    $canvas.height());
    var ctxp=canvasProcessing.elt.getContext("2d");
    var num;

    //Compruebo el número de parametros de la funcion
    if (arguments.length==1)
        imgOrig=forma.getImage();
    else { //Si Abrimos la ventana filtro
        num=1;

        if (arguments==2)
            num=1;
        else
            num=n;

        imgOrig=img;
    }
    if (val==="NORMAL"){ //En este caso no se aplica ningun filtro y
    la forma se deja como estaba originalmente
        ctx.putImageData(imgOrig,0,0);
        forma.setImage(imgOrig);
        return;
    }
    canvasProcessing.parent('lienzo'); //Coloco el canvas como hijo
    del lienzo

    //Coloco la imagen en el canvas auxiliar
    ctxp.putImageData(forma.getImage(),0,0);
    if (val==="BLUR" || val==="POSTERIZE"){

        filter(val,num);
    } else
        filter(val);

    //Extraigo el array del contexto auxiliar

```

```

imgDataAux=ctx.getImageData(0,0,canvas.width,canvas.height)
ctx.putImageData(imgDataAux,0, 0);
}
function draw(){
}

```

Código que muestra un cuadro de dialogo modal.

```

//Muestra un cuadro de dialogo personalizado
function mostrarDialogo(titulo,texto){
    var d=$( '<div class="dialog"
title='+titulo+'>'+texto+'</div>' ).dialog({
        modal:true,
        buttons: [
            {
                text:"Yes",
                "class": 'DialogButton',
                click: function() {alert('you chose
yes');}
            },
            {
                text:"No",
                "class": 'DialogButton',
                click: function() {
                    forma.limpiarForma();

                    forma.setImage(ctx.getImageData(0,0,canvas.width,canvas.height))
; //Inicializo la imagen
                    d.dialog('close');
                }
            },
            {
                text:"Cancel",
                "class": 'DialogButton',
                click: function()
{d.dialog('close');}
            }
        ]
    });
    $( function() {
        d.dialog();
    } );
}

```

Código para crear un Slider.

```

function mostrarSlider(){
    var mn,mx;
    var seleccion=$("#efectosId-button .ui-selectmenu-text").text();

    if (seleccion==="Posterizar"){
        mn=2;
        mx=250;
    }
    if (seleccion==="Desenfoco blur") {
        mn=1;
        mx=10;
    }

    $( "#slider" ).slider({

```

```

range:"min",
value:1,
min:mn,
max:mx,
slide:function(event,ui){
    var $canvas=$("#poo");
    var ctx=$canvas[0].getContext("2d");
    var img,op;
    img=ctx.getImageData(0,0,canvas.width,canvas.height);
    op=$("#efectosId-button .ui-selectmenu-text").text();
    switch (op) {
        case "Desenfoque blur":
            mySelectEvent("BLUR",img,ui.value);
            break;
        case "Posterizar":
            mySelectEvent("POSTERIZE",img,ui.value);
            break;
    }
    $("#valor").text(ui.value);
}
});
$("#valor").text($("#slider").slider("value"));
$("#valor").parent().removeClass("oculto");
$("#slider").removeClass("oculto");
}

```

Código que selecciona el filtro a través del cuadro desplegable Aplicar Filtro

```

op=$("#efectosId-button .ui-selectmenu-text").text();
var imgOrig=forma.getImage();
switch (op) {
    case "Normal":
        mySelectEvent("NORMAL",imgOrig);
        break;
    case "Limite":
        mySelectEvent("THRESHOLD",imgOrig);
        break;
    case "Escala de grises":
        mySelectEvent("GRAY",imgOrig);
        break;
    case "Opacidad":
        mySelectEvent("OPAQUE",imgOrig);
        break;
    case "Invertir":
        mySelectEvent("INVERT",imgOrig);
        break;
    case "Posterizar":
        mySelectEvent("POSTERIZE",imgOrig);
        break;
    case "Desenfoque Dilatar":
        mySelectEvent("DILATE",imgOrig);
        break;
    case "Desenfoque blur":
        mySelectEvent("BLUR",imgOrig);
        break;
}

```

```

        case "Desenfoque Erosionar":
            mySelectEvent("BLUR",imgOrig);
            break;
    }

```

Código Mysql para generar la base de datos

Creación de tabla imagenes

```

CREATE TABLE IF NOT EXISTS `imagenes` (
  `ruta` varchar(255) NOT NULL,
  `extension` varchar(3) NOT NULL,
  `fecha` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `usuario_fk` bigint(20) NOT NULL,
  KEY `usuario_fk` (`usuario_fk`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Creación de tabla usuarios

```

CREATE TABLE IF NOT EXISTS `usuarios` (
  `idusuario` bigint(20) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  PRIMARY KEY (`idusuario`),
  UNIQUE KEY `nombre` (`nombre`),
  UNIQUE KEY `email` (`email`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

```

Creación de relación entre Usuarios e Imagenes

```

ALTER TABLE `imagenes`
  ADD CONSTRAINT `imagenes_ibfk_1` FOREIGN KEY (`usuario_fk`)
REFERENCES `usuarios` (`idusuario`);

```

Código del lado del servidor

Código del controlador Welcome.php

Este fragmento de código es el que inicializa la ejecución de la aplicación. Cuando un usuario teclea el nombre de dominio, el controlador Welcome.php verifica si hay una sesión iniciada y muestra información sobre la sesión en caso de existir:

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Welcome extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
    }

    //Método que se ejecuta por defecto al arrancar la aplicación
    public function index()
    {
        //Comprobamos si existe una sesión iniciada.
    }
}

```

```

        $is_logged_in = ($this->session->has_userdata('logged_in'))?$this->session->userdata('logged_in'):NULL;

        if (!empty($is_logged_in) && $is_logged_in ) //Si la sesión ya ha sido iniciada
        {
            $datos['username']=$this->session->userdata('username');
            $datos['email']=$this->session->userdata('email');

            $this->load->view('header',$datos);
            $this->load->view('index');
            $this->load->view('footer');
        } else
        {
            $this->load->view('header');
            $this->load->view('index');
            $this->load->view('footer');
        }
    }
}

```

Quando la aplicación se está ejecutando, si el usuario quiere iniciar o cerrar sesión, o desea guardar o abrir una imagen del servidor, se llama al controlador Login.php. Este controlador contiene el siguiente código:

Código del controlador Login.php

```

<?php
header('Access-Control-Allow-Origin: *');
header("Access-Control-Allow-Methods: GET, OPTIONS");
if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Login extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
        //Cargamos el modelo Auth_model que me permite operar con
BBDD
        $this->load->Model('Auth_model');
        //Cargamos la libreria file para operar con archivos, util
para guardar las imagenes
        $this->load->helper("file");
    }

    public function index()
    {

```

```

}

//Método para cerrar sesión
public function logout()
{
    header('Content-Type: application/json');
    $this->session->sess_destroy();
    $respuesta="ok";
    echo json_encode($respuesta);
}

//Método para iniciar sesión
public function auth()
{
    $username = $this->input->post('usuario');
    $password = $this->input->post('password');

    //call the model for auth
    if ($this->Auth_model->login($username, $password))
    {
        header('Content-Type: application/json');
        $respuesta=$this->session->userdata();
        echo json_encode($respuesta);
    }
}

//Método para guardar una imagen en el servidor
public function guardarImagen(){

    header('Content-Type: application/json');
    $respuesta=array (
        'logged_in' => $this->session-
>userdata('logged_in'),
        'estado' => TRUE
    );

    if (!empty($respuesta['logged_in'])){
        //Obtengo la imagen
        $img = $this->input->post('imgBase64');
        $name = $this->input->post('name');
        $format = $this->input->post('format');

        $folder = 'images/'.$this->session->userdata('id');
        if (!file_exists($folder)) {
            mkdir($folder, 0777, true);
        }

        $location = $folder.'/'.$name.$format;
        //formateo la imagen y la convierto en un
string
        $img = substr($img,strpos($img,",")+1);

```

```

        $img = base64_decode($img);

        switch ($format) {
            case '.png':

$respuesta['estado']=file_put_contents($location, $img);

                break;
            case '.jpg':

$source=imagecreatefromstring($img);

$respuesta['estado']=imagejpeg($source,$location,100);

                break;

            default:
                # code...
                break;
        }
    }

    if ($respuesta['estado']!==FALSE){
        $this->Auth_model->addImageDB($location,$format);
    }

    echo json_encode($respuesta);
}

//Método para obtener datos de sesión
public function obtenerDatosSession()
{
    header('Content-Type: application/json');
    echo json_encode($this->session->userdata());
}

//Método para registrar un usuario
public function registro()
{
    //$this->session->sess_destroy();
    $userData = array(
        'nombre' => $this->input->post('insertUser'),
        'email' => $this->input->post('insertEmail'),
        'password' => $this->input->post('insertPassword'),
    );

    $operacion = $this->Auth_model->insertar($userData);

    if ($operacion)
    {
        //Inicio sesión con los datos del nuevo usuario

```

```

        if ($this->Auth_model->login($userData['nombre'],
$userData['password']))
        {
            header('Content-Type: application/json');
            $respuesta=$this->session->userdata();
            echo json_encode($respuesta);
        }

        //Envio un email de confirmación de registro
        $config = Array(
            'protocol' => 'sendmail',
            'mailtype' => 'html',
            'smtp_host' => 'smtp.strato.com', //your SMTP
host
            'smtp_port' => 465,
            'smtp_user' => 'ipaint@galeonweb.es', //your
SMTP email address
            'smtp_pass' => '2m3ixbsX//', //your SMTP email
password
            'charset' => 'iso-8859-1',
            'wordwrap' => TRUE
        );
        $this->load->library('email', $config);
        $this->email->set_newline("\r\n");
        $this->email->set_header('MIME-Version', '1.0;
charset=utf-8'); //must add this line
        $this->email->set_header('Content-type',
'text/html'); //must add this line

        $this->email->from('ipaint@galeonweb.es', 'alberto
marin'); //from/sender email [with name (optional)]
        $this->email->to($userData['email']); //to/receiver
email

        $this->email->subject('Confirmación de su registro
en IPaint'); //email subject

        $data = array('userName'=> 'Anil Kumar Panigrahi');
        $message = $this->load->view('message',$data,TRUE);
//html template/body with dynamic data
        $this->email->message($message);
        $this->email->send();

    } else
        echo "Fucked";
    }

    //Método para listar imagenes del servidor
    public function listarImagenes(){
        header('Content-Type: application/json');
        $id=$this->session->userdata('id');
        $listado=$this->Auth_model->listarImagenes($id);
        echo json_encode($listado);
    }
}

```

```

//Método para borrar imagenes
public function borrarImagen(){
    header('Content-Type: application/json');
    $id=$this->session->userdata('id');
    $img = $this->input->post('image');
    $path= "images/".$id."/".$img;
    if (unlink("./".$path)){
        $this->Auth_model->borrarImagen($id,$path);
        $respuesta="ok";
    } else
        $respuesta="error";

    echo json_encode($respuesta);
}
}

```

Código del Modelo Auth_Model.php

Este es el archivo responsable de las operaciones con la base de datos:

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access
allowed');
class Auth_model extends CI_Model {

    public function __construct()
    {
        parent::__construct();
    }

    //Método que comprueba si el usuario existe en la tabla usuarios
    public function login($name, $password){
        $password = md5($password);
        $this->db->where('nombre',$name);
        $this->db->where('password',$password);
        $query = $this->db->get('usuarios');
        //Si hay algún resultado significa que lo ha encontrado
        if($query->num_rows()==1){
            foreach ($query->result() as $row){
                $data = array(
                    'id' => $row->idusuario,
                    'username'=> $row->nombre,
                    'email' => $row->email,
                    'logged_in'=>TRUE
                );
            }
            $this->session->set_userdata($data);
            return TRUE;
        }
        else{
            return FALSE;
        }
    }
}

```

```

//Método que registra un nuevo usuario
public function insertar($data){

    $insertData=array(
        'nombre' => $data['nombre'],
        'email' => $data['email'],
        'password' => md5($data['password']),
    );

    $this->db->where('nombre',$data['nombre']);
    $query1 = $this->db->get('usuarios');

    $this->db->where('email',$data['email']);

    $query2 = $this->db->get('usuarios');
    //Si hay algún resultado significa que el usuario ya existe
    if($query1->num_rows()>=1 || $query2->num_rows()>=1){
        return FALSE;
    } else {
        //insert user data to users table
        return $this->db->insert('usuarios', $insertData);
    }
}

//Método que inserta una nueva imagen en la base de datos
public function addImageDB($location, $format){
    $timestamp = date('Y-m-d G:i:s');
    $insertData=array(
        'ruta' => $location,
        'extension' => $format,
        'fecha' => $timestamp,
        'usuario_fk' => $this->session->userdata('id'),
    );

    //Si guardamos con el mismo nombre evitamos registros
    duplicados
    $this->db->where('ruta',$location);
    $this->db->where('usuario_fk',$this->session->userdata('id'));
    $query = $this->db->get('imagenes');
    if ($query->num_rows()==0)
        return $this->db->insert('imagenes', $insertData);

}

//Muestra un listado de las imagenes del usuario según su id
public function listarImagenes($id){
    $this->db->where('usuario_fk',$id);
    $this->db->order_by("fecha", "asc");
    $query = $this->db->get('imagenes');
    return $query->result();
}

//Borra la imagen que el usuario haya seleccionado de la base de
datos

```

```

public function borrarImagen($id, $path){
    $this->db->delete('imagenes', array('usuario_fk' => $id,
'ruta' => $path));
    }
}

```

Como se puede comprobar, el servidor devuelve los datos siempre en formato json, que mediante JavaScript (Ajax) es leído por la página cliente. El mecanismo es siempre el mismo y a modo de ejemplo incluyo el siguiente fragmento:

Borrado de imagen (FrontEnd)

```

function borrarImagen(e) {
    var link=e.target.closest('.thumbnail').firstChild.href;
    var pattern=/\w+(?:\.\w+)*$/g;
    var filename=pattern.exec(link)[0];

    $.ajax({
        url:'index.php/Login/borrarImagen',
        type: "POST",
        data: {
            'image':filename,
        }
    }).done(function(result){
        console.log(result);
        if (result=="ok"){
            e.target.closest('.col-sm-1').remove();
        }
    })
}

```

Obsérvese como cuando la operación se haya completado, se elimina de la página el objeto de clase `'col-sm-1'`

Anexo 3. Librerías / Código externo utilizado

Spectrum JQuery ColorPicker Sitio Web: <https://bgrins.github.io/spectrum/>

Para el uso del control selector del color

jQuery Mobile Sitio Web: <https://jquerymobile.com/>

Para el uso de la gestión de eventos y adaptación visual a móviles.

jQuery Sitio Web: <https://jquery.com/>

Para el uso de selectores CSS a través de JavaScript y facilidad de programación

jQuery UI: Sitio Web: <https://jqueryui.com/>

Para el uso de controles ya definidos.

Facebook Javascript SDK Sitio Web: <https://developers.facebook.com/docs/javascript/>

Para el uso de la aplicacion de facebook en el inicio de sesión

FontAwesome.io Sitio web: <http://fontawesome.io/>

Para el uso de iconos vectoriales y escalables

P5.js y P5.dom.js Sitio web: <https://p5js.org/>

Para la aplicación de filtros de imágenes

HTML5-Paint-Bucket-Tool Sitio Web: www.williammalone.com

Para poder usar la herramienta cubo de pintura. El código rellena una zona delimitada de un color seleccionado previamente

Función drawImageProp Sitio Web: <https://stackoverflow.com/>

Para poder mantener las proporciones de las imágenes cuando se abren en modo responsive.

CodeIgniter Sitio Web: <https://codeigniter.com/>

Responsable del control de la base de datos y la lógica llevada a cabo en el lado del servidor.

Anexo 4. Capturas de pantalla Imagen del logo creada con Illustrator

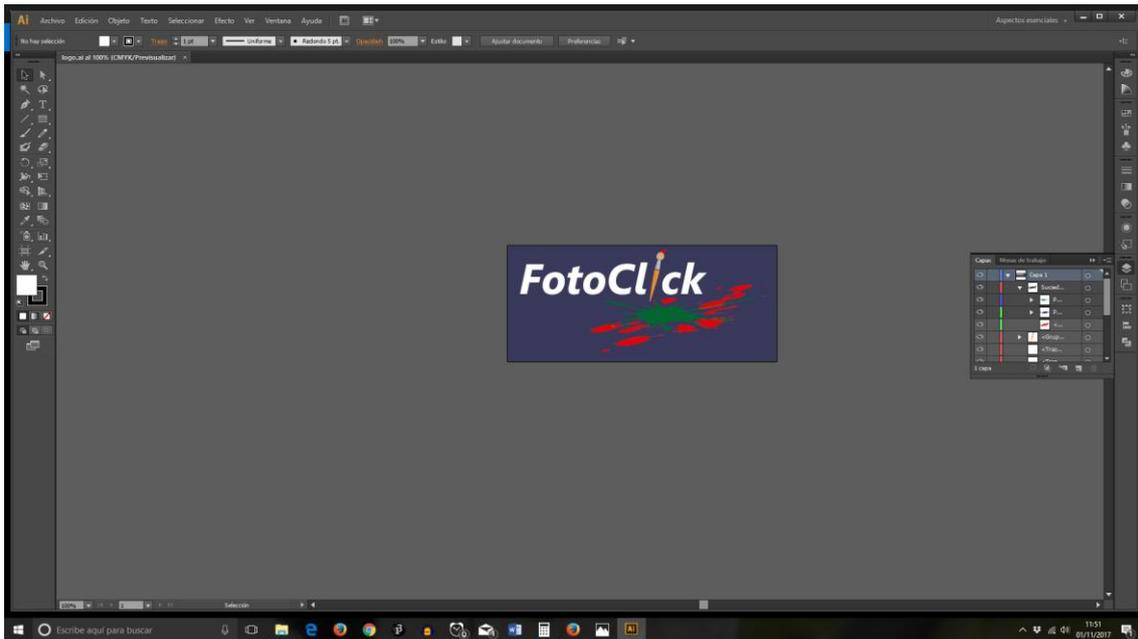
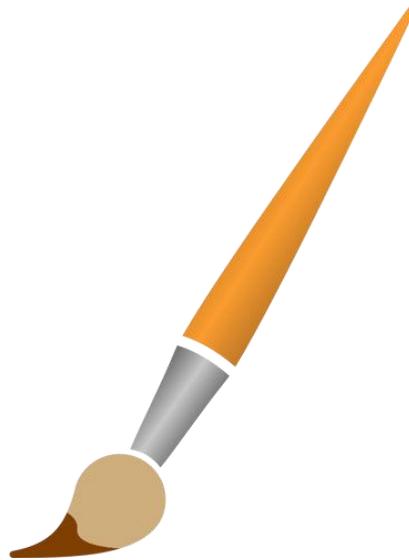
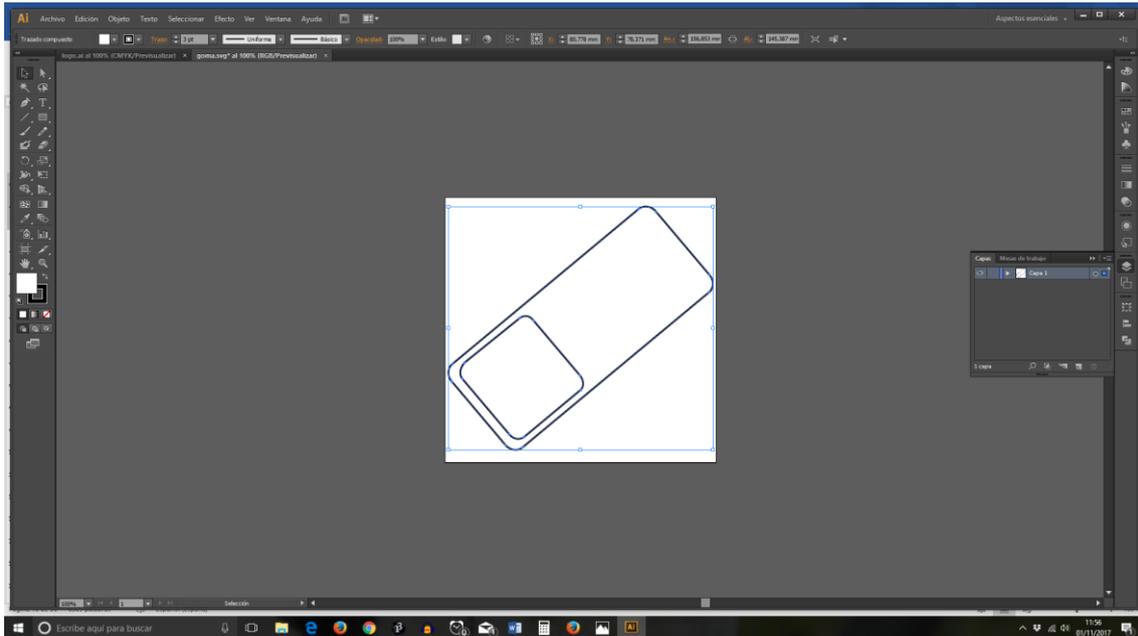


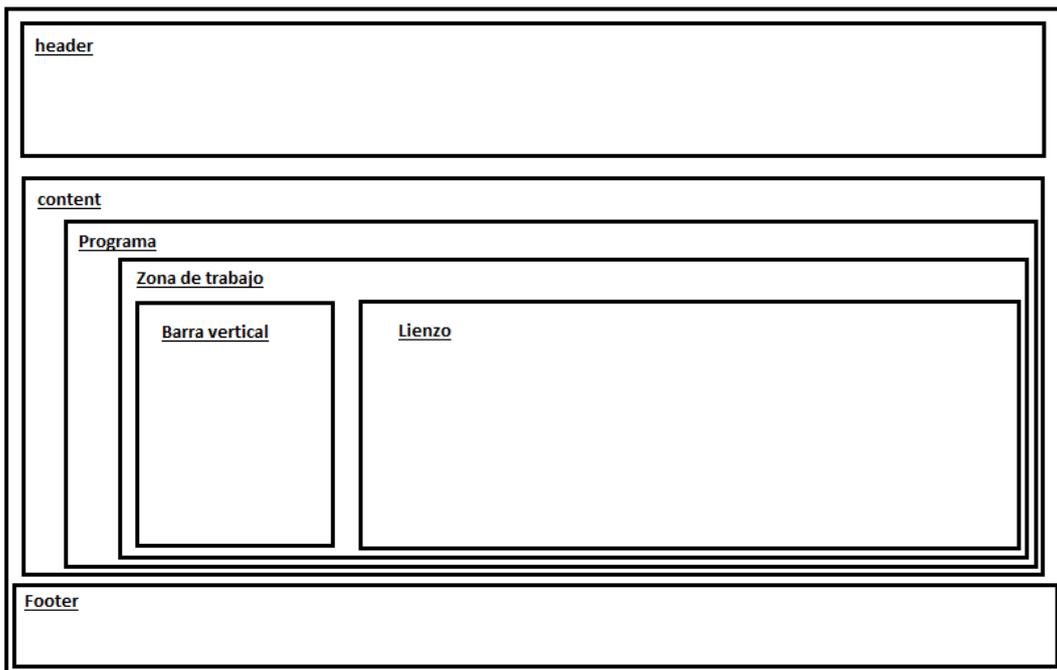
Imagen del pincel



Goma de borrar



Esquema que muestra la distribución de las capas HTML



Control SELECT en forma de SPAN

seleccionada: rectangulo
s: 0 px, 90 px

pector [] Consola [] Depurador [] Editor de estilos [] Rendimiento [] Memoria [] Red [] Almacenamiento

Buscar en HTML

```
<div id="efectos"></div>  
<div id="efectos" style="display: block;">  
  <div class="ui-dialog-titlebar ui-corner-all ui-widget-header ui-helper-clearfix ui-draggable-handle"></div>  
  <label for="efectosId-button" style="margin:1em;">Elija un efecto</label>  
  <select id="efectosId" style="display: none;"></select>  
  <span id="efectosId-button" class="ui-selectmenu-button ui-button ui-widget ui-selectmenu-button-open ui-corner-top tabindex="0" role="combobox" aria-expanded="true" aria-autocomplete="list" aria-owns="efectosId-menu" aria-haspopup="true" aria-activedescendant="ui-id-5" aria-labelledby="ui-id-2" aria-disabled="false">  
    <span class="ui-selectmenu-icon ui-icon ui-icon-triangle-1-s"></span>  
    <span class="ui-selectmenu-text">Normal</span>  
  </span>
```

Estructura de archivos de CodeIgniter

Nombre	Fecha de modifica...
cache	21/11/2017 18:55
config	21/11/2017 18:55
controllers	23/12/2017 13:32
core	21/11/2017 18:55
helpers	08/12/2017 12:12
hooks	21/11/2017 18:55
language	21/11/2017 18:55
libraries	24/12/2017 1:30
logs	11/01/2018 11:16
models	18/12/2017 11:29
third_party	21/11/2017 18:55
views	23/12/2017 10:11
.htaccess	28/12/2017 14:39
index.html	25/09/2017 9:43

Estructura de directorios del servidor local

Este equipo > Nuevo vol (D:) > wamp > www > iPaint > Bus

Nombre	Fecha de modifica...	Tipo	Tamaño
application	21/11/2017 18:55	Carpeta de archivos	
images	27/12/2017 11:45	Carpeta de archivos	
static	13/12/2017 11:43	Carpeta de archivos	
system	21/11/2017 18:55	Carpeta de archivos	
user_guide	21/11/2017 18:55	Carpeta de archivos	
.editorconfig	25/09/2017 9:43	Archivo EDITORC...	1 KB
.gitignore	25/09/2017 9:43	Archivo GITIGNORE	1 KB
composer.json	25/09/2017 9:43	Archivo JSON	1 KB
contributing.md	25/09/2017 9:43	Archivo MD	7 KB
index.php	25/09/2017 9:43	Archivo PHP	11 KB
license.txt	25/09/2017 9:43	Documento de tex...	2 KB
readme.rst	25/09/2017 9:43	Archivo RST	3 KB

Encryptación de iPaint

function md5()

Online generator [md5 hash](#) of a string

md5 ()

hash darling, hash!

md5 checksum:

e86b3395fa2b298cc0554644582f3340



Archivo de configuración de base de datos en CodeIgniter

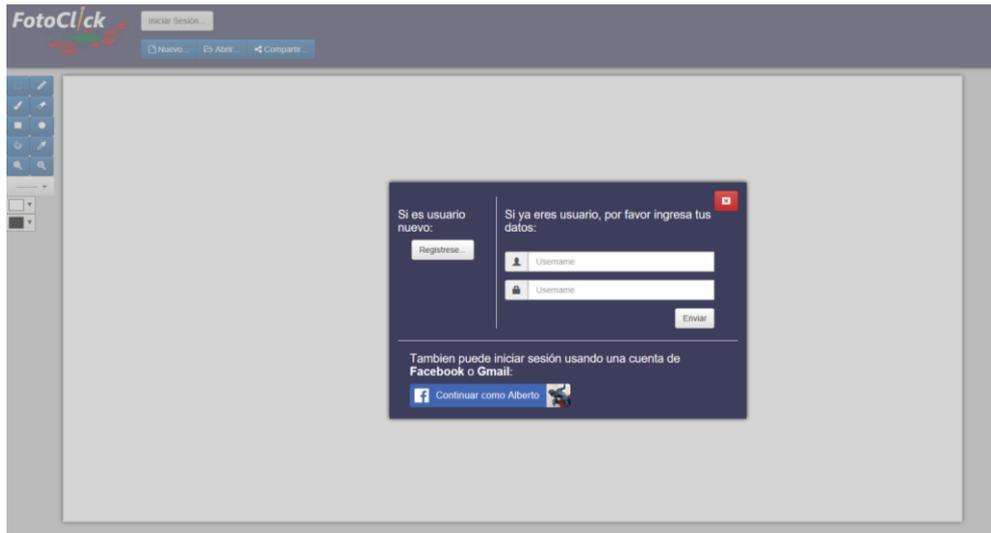
```
76 $db['default'] = array(  
77     'dsn' => '',  
78     'hostname' => 'localhost',  
79     'username' => 'root',  
80     'password' => ' ',  
81     'database' => 'ipaint',  
82     'dbdriver' => 'mysql',  
83     'dbprefix' => '',  
84     'pconnect' => FALSE,  
85     'db_debug' => (ENVIRONMENT !== 'production'),  
86     'cache_on' => FALSE,  
87     'cachedir' => '',  
88     'char_set' => 'utf8',  
89     'dbcollat' => 'utf8_general_ci',  
90     'swap_pre' => '',  
91     'encrypt' => FALSE,  
92     'compress' => FALSE,  
93     'stricton' => FALSE,  
94     'failover' => array(),  
95     'save_queries' => TRUE  
96 );  
97
```

Modificaciones del archivo Autoload.php

```
61 $autoload['libraries'] = array('session','database','  
    form_validation');
```

```
92 $autoload['helper'] = array('url');
```

Uso de la API de Facebook en modo escritorio



Buscador de iconos de la web fontawesome.io



You asked, Font Awesome delivers with 41 shiny new icons in version 4.7. Want to request new icons? [Here's how](#). Need vectors or want to use on the desktop? Check the [cheatsheet](#).



Hire the top 3% of vetted, senior-level developers through Toptal. No risk. ads via Carbon

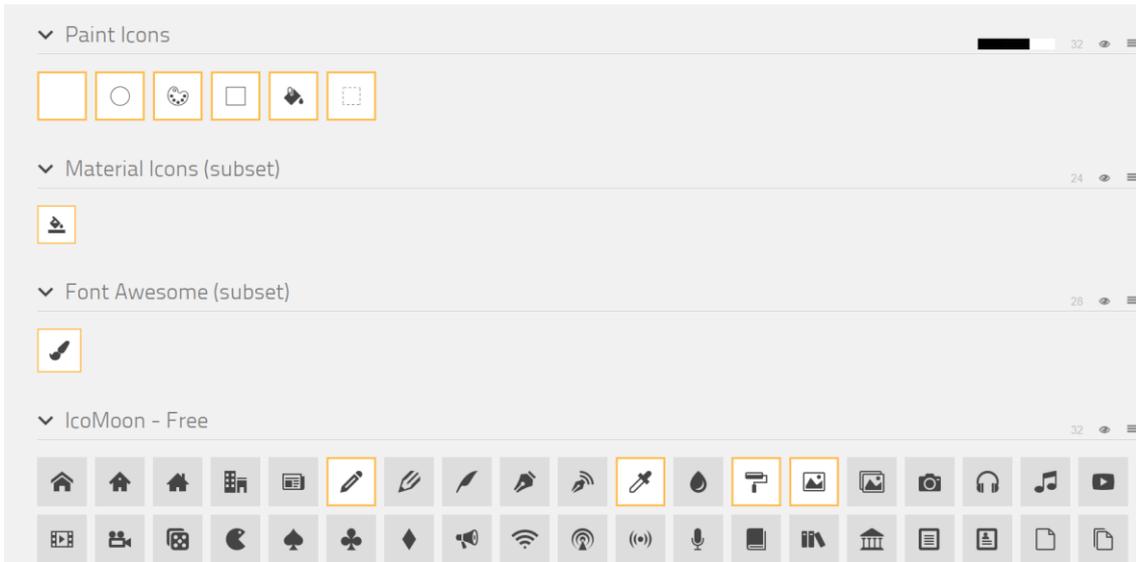


by Algolia

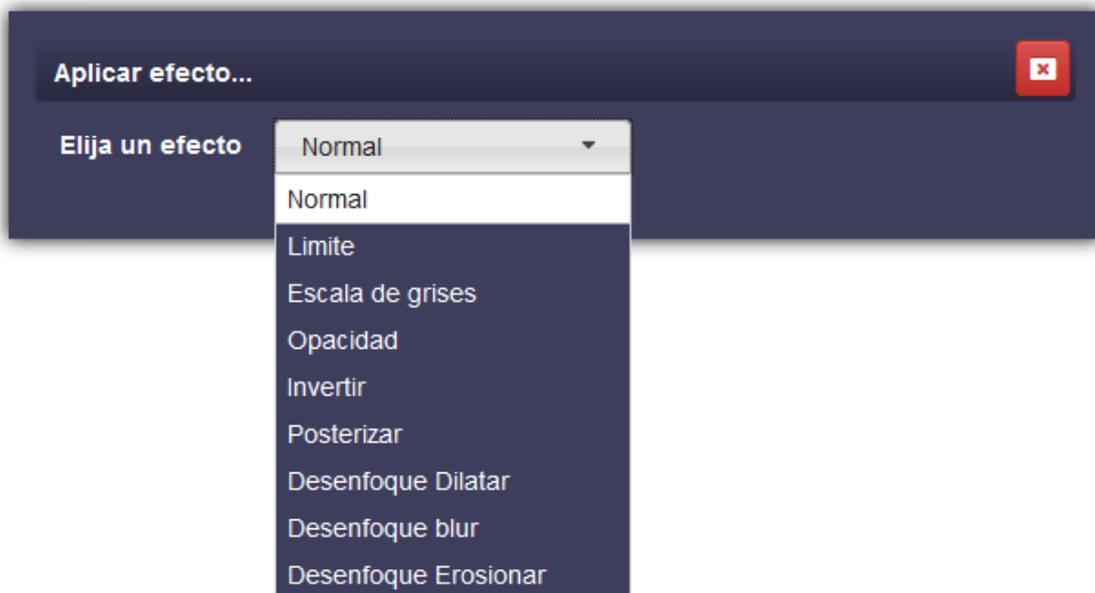
41 New Icons in 4.7

- | | | | |
|---------------------------|----------------|-----------------|-------------------------|
| address-book | address-book-o | address-card | address-card-o |
| bandcamp | bath | bathtub (alias) | drivers-license (alias) |
| drivers-license-o (alias) | eercast | envelope-open | envelope-open-o |
| etsy | free-code-camp | grav | handshake-o |
| id-badge | id-card | id-card-o | imdb |
| linode | meetup | microchip | podcast |
| quora | ravelry | s15 (alias) | shower |
| snowflake-o | superpowers | telegram | thermometer (alias) |

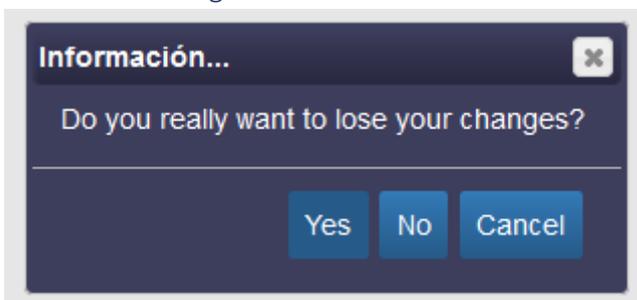
Creación de iconos a través de Icomoon



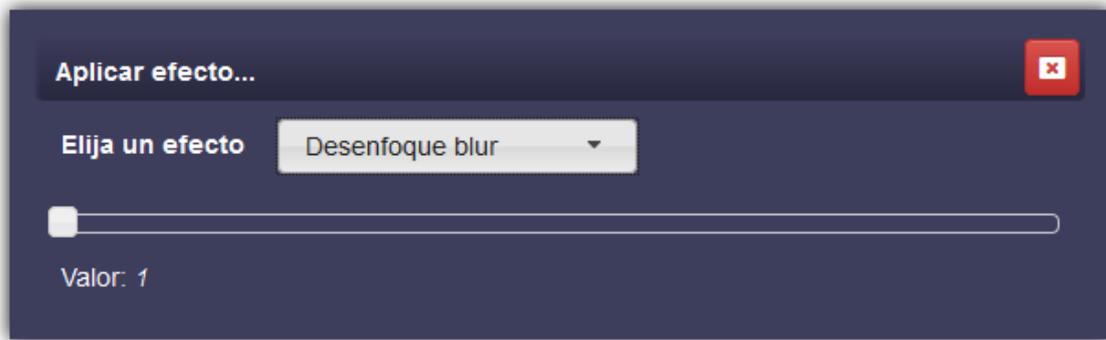
Cuadro desplegable para aplicar filtros



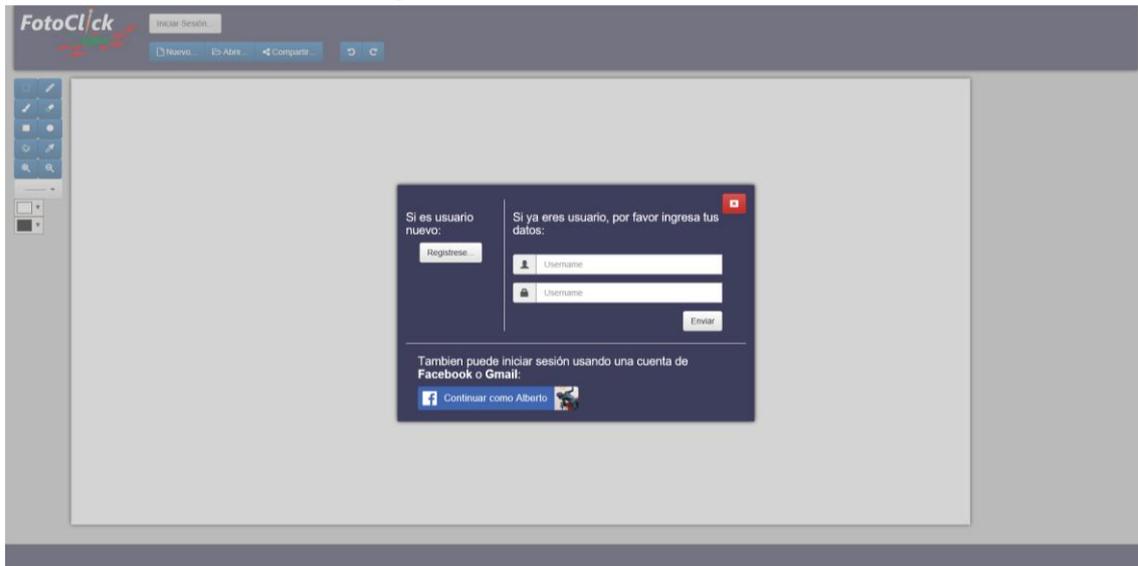
Cuadro de dialogo Nuevo...



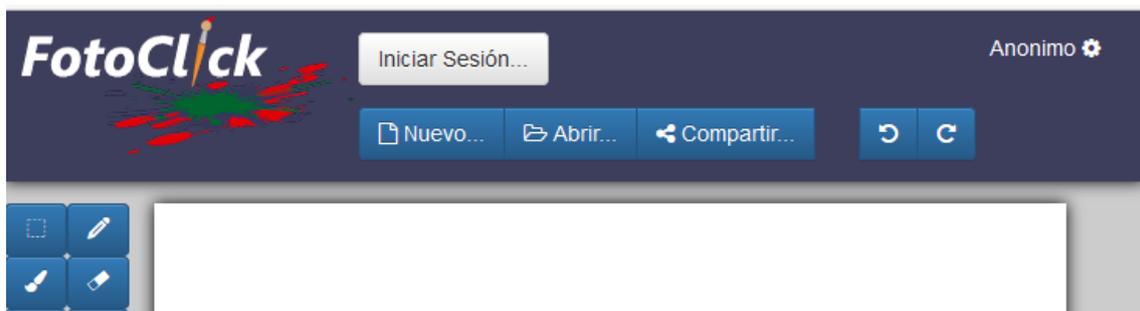
Aplicar efecto Blur



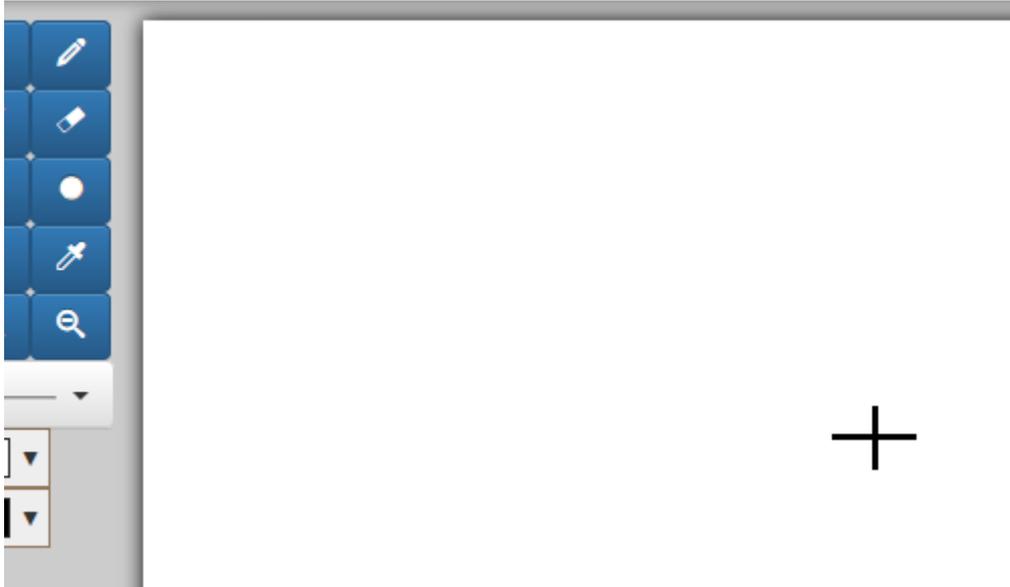
Ventana de identificación o registro



Cabecera de la aplicación web



Forma del ratón al pasar por el lienzo



Aspecto del pie de página de la aplicación



Anexo 6. Libro de estilo

Título de la aplicación

Fotoclick

Marca o logotipo



Color de fondo del logotipo:

#3D3D5C



Tipografía

Texto normal

Helvetica Neue ,Helvetica, Arial, sans-serif

Tamaño: 14px

Estilo Normal

 Nuevo...

Helvetica Neue ,Helvetica, Arial, sans-serif

Tamaño: 0.9 em

Estilo Normal

Herramienta seleccionada: rectangulo
Coordenadas: 813 px, 753 px

Título1

Helvetica Neue ,Helvetica, Arial, sans-serif

Tamaño:14px

Estilo: Negrita

Información...

Color del texto

Blanco sobre fondo violeta

Herramienta seleccionada: rectangulo
Coordenadas: 813 px, 753 px

Negro sobre fondo blanco

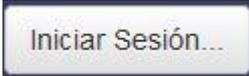
No se ha seleccionado ningún archivo.

Botones

Apariencia de botones



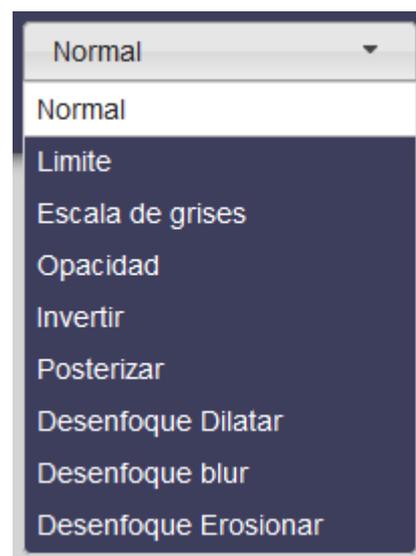
Botones resaltados



Botón de cierre



Cuadros desplegables



Anexo 7. Resumen ejecutivo

Nombre comercial. Fotoclick

Resumen comercial.

Es una aplicación web de dibujo que funciona en cualquier plataforma (móvil o escritorio) y que ofrece ventajas de movilidad respecto de otras aplicaciones de dibujo básico. La aplicación permite compartir el contenido en redes sociales, sacar fotos y modificarlas incluyendo nuevas formas creadas por el usuario de manera intuitiva.

Modelo de negocio.

La aplicación se ofrece de forma gratuita, no obstante, se requiere el registro de usuario para que se pueda beneficiar de todas las ventajas y pueda usar la aplicación en su totalidad. Una vez el usuario se haya registrado podrá beneficiarse de las opciones de movilidad implementadas en la aplicación. A cambio, el usuario acepta ceder sus datos a la empresa con fines comerciales. La publicidad será la fuente de financiación principal del proyecto. No obstante, el proyecto puede crecer y dar lugar a diferentes versiones más avanzadas, que si que pueden tener costes.

Productos y servicios.

Se ofrece una aplicación online gratuita de dibujo básico, donde los usuarios podrán disfrutar gratuitamente de la aplicación y se beneficiaran de recibir ofertas en su email, que estará identificada de manera reglamentaria, con la palabra PUBLI en el asunto.

Mercado y competencia.

Esta información ya aparece estudiada en el apartado Análisis de mercado [[ver apartado](#)]

Presupuesto.

Esta información ya aparece estudiada y analizada en el apartado Presupuesto [[ver apartado](#)]

Viabilidad.

Esta información ya aparece estudiada y analizada en el apartado Presupuesto [[ver apartado](#)]

Análisis DAFO.

[[ver análisis DAFO](#)]

Anexo 9. Bibliografía

Codeigniter.com. (2018). *CodeIgniter User Guide — CodeIgniter 3.1.7 documentation*. [online] Available at: https://codeigniter.com/user_guide/ [Accessed 14 Jan. 2018].

Corcoy Costalago, J. and Ariztoy Ajuria, I. (2018). *Bit 41*. 1st ed. Barcelona: UOC.

Dev.mysql.com. (2018). *MySQL :: MySQL 5.7 Reference Manual*. [online] Available at: <https://dev.mysql.com/doc/refman/5.7/en/> [Accessed 14 Jan. 2018].

Mark Otto, a. (2018). *Introduction*. [online] Getbootstrap.com. Available at: <https://getbootstrap.com/docs/4.0/getting-started/introduction/> [Accessed 14 Jan. 2018].

Mozilla Developer Network. (2018). *JavaScript*. [online] Available at: <https://developer.mozilla.org/es/docs/Web/JavaScript> [Accessed 14 Jan. 2018].

P5js.org. (2018). *p5.js | learn*. [online] Available at: <https://p5js.org/learn/> [Accessed 14 Jan. 2018].

Php.net. (2018). *PHP: Manual de PHP - Manual*. [online] Available at: <http://php.net/manual/es/> [Accessed 14 Jan. 2018].

W3schools.com. (2018). *jQuery Tutorial*. [online] Available at: <https://www.w3schools.com/Jquery/default.asp> [Accessed 14 Jan. 2018].

CSS-Tricks. (2018). *CSS-Tricks*. [online] Available at: <https://css-tricks.com/> [Accessed 14 Jan. 2018].

Anexo 10. Nota biográfica

Mi nombre es Alberto Marín, soy de Cádiz y tengo 42 años. A nivel académico tengo el Bachiller/COU y el título de Técnico Superior en Informática. A nivel profesional he trabajado principalmente como formador ocupacional de cursos para la Junta de Andalucía. Además, también he gestionado un negocio propio durante 15 años. Actualmente, trabajo dando soporte técnico a clientes de Orange.

Hace 8 años decidí reciclarme y adquirir conocimientos actualizados en informática. Los medios digitales y la informática son mi pasión y durante los últimos 4 años del grado me he dedicado en profundidad al estudio de la multimedia. Antes de empezar el grado ya disponía del título de informático, así que ya partía de algunos conocimientos previos. Además, por propia inquietud, también conocía algunos lenguajes de programación que estudié por mi cuenta, no obstante, me faltaban disciplinas como la gestión de proyectos, o terminologías comúnmente empleadas en esta área.

El grado Multimedia me ha proporcionado seguridad, conocimientos y procedimientos de trabajo, que me ayudan a conseguir mis objetivos gracias a la realización de numerosos proyectos que siguen procesos de trabajos específicos. Además de los proyectos que he realizado en el grado también he realizado proyectos personales tanto para la web como para sistemas de escritorios. Los lenguajes de programación que más he usado son Javascript, HTML, CSS y PHP. En el grado he usado otros lenguajes como Processing, JQuery y ActionScript.

Me considero una persona autodidacta y en continuo aprendizaje, y por ello también me he registrado en portales webs como freeCodeCamp, donde se dan cursos gratuitos de programación. No obstante, a veces tengo que poner límites a mi pasión por los medios digitales y hacer algo de vida social. Me encanta disfrutar de mis perros, sacándolos a pasear al bosque o a la playa. También disfruto de cosas sencillas como una buena película o un buen libro, y por supuesto de la compañía de familia y amigos.

Estando a las puertas de terminar el grado, me gustaría dedicarme a la docencia ya que, por motivos de experiencia y aptitudes, creo que soy perfecto para dicho puesto. Realizar un master en educación sería fantástico para cumplir mis expectativas, no obstante, tampoco descarto la opción de realizar un master de especialización en Multimedia, pues me gustaría perfeccionar lo aprendido en el grado con nuevas disciplinas y herramientas que me permitan alcanzar mayor grado de conocimiento.