



**CIFP CARLOS III CARTAGENA
PROYECTO **RECUÉRDAME****

José Antonio Borrero Fernández
Joaquín Vidal Guzmán



Contenido

1)	Fundamentos legales.....	2
2)	Breve descripción.....	2
3)	Lista de programas y tecnologías:.....	2
4)	Nuestros objetivos	3
5)	Estructura del proyecto	3
6)	Diseño e imagen corporativa	4
7)	Mongodb.....	5
8)	Heroku.....	6
9)	Extras.....	7
	Reproductor	7
	Recuperación de contraseña	7
	Excel con progresos de los jugadores	8
10)	Webs y referencias.....	9

1) Fundamentos legales.

Este proyecto de investigación se circunscribe al apartado 4.c.3 de la resolución de 1 de marzo de 2010 de la Dirección General de Formación Profesional, por la que se establecen las instrucciones para el adecuado desarrollo docente del módulo de proyectos en los nuevos ciclos formativos de formación profesional de grado superior derivados de la Ley Orgánica de Educación 2/2006 de 3 de mayo, en el ámbito de la Comunidad Autónoma de la Región de Murcia.

Tal y como expone la citada resolución de 1 de marzo de 2010 de la Consejería de Educación en el que se establece el currículo del ciclo formativo de grado superior de Desarrollo de Aplicaciones Web (DAW), la duración de este proyecto será de 30 horas.

2) Breve descripción

El proyecto “Recuérdame” pretende ser una herramienta informática que ayude a los enfermos neurológicos en el ejercicio de las actividades cognitivas y motoras finas, a través de sencillos juegos pretendemos entretener, divertir y ejercitar.

El proyecto consta de dos grandes bloques, el primer bloque que podemos denominar “el gestor” que está orientado a que lo usen los cuidadores y es el encargado de gestionar los jugadores y los juegos, el segundo bloque son los juegos, que para nosotros son las herramientas para ejercitar capacidades como la memoria, la lógica o las matemáticas.

3) Lista de programas y tecnologías:

- Lenguajes y librerías:
 - Html.
 - Css.
 - Javascript.
 - JQuery.
 - JQuery ui.
 - Parallax js
 - AJAX.
 - PHP.
 - MongoDB.
 - Bootstrap 3.
 - Canvas.
- Programas y tecnologías:
 - Git.
 - GitHub.
 - SourceTree.

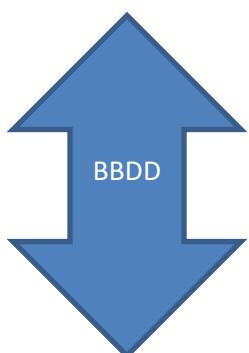
- PhpStorm.
- Brackets.
- Xamp.
- MongoDB Community Server.

4) Nuestros objetivos

Desde un principio nos planteamos hacer un proyecto que pudiera servir para algo y para alguien, es por lo que elegimos una temática social.

Desde el punto de vista académico decidimos intentar usar todos los conocimientos adquiridos, pero enfocado a tecnologías que no se viesen en el ciclo, es por lo que elegimos hacer un proyecto por capas, una base de datos no relacional, como es **Mongodb** y un servidor muy alejado de los convencionales como es **Heroku**.

5) Estructura del proyecto

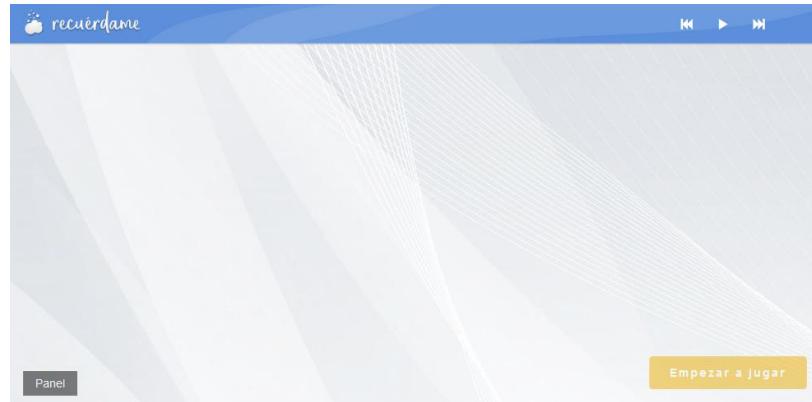


El proyecto está programado o dividido por capas, el usuario ve la que hemos llamado “capa cliente” donde se encuentran todo el diseño y los ficheros html y jQuery, estos últimos llaman a los servicios de la “capa servidor” que está programada en PHP 5.6 que a su vez llama a un “manejador de BBDD” que es el que ataca a la base de datos, de forma que el usuario usa directamente los archivos php del servidor y es totalmente opaco el acceso de a base de datos, de hecho la parte servidora está programada de tal forma que únicamente cambiando el manejador de base de datos “BBDD.php” por otro basado en mysql, u Oracle, funcionaría totalmente sin cambiar nada.



Siguiendo esta idea de modularidad, hemos programado “un marco” que va llamando a los juegos, los ejecuta “dentro”, recoge las puntuaciones y las envía a la base de datos, de forma que para introducir un nuevo juego en el sistema sólo hay que subirlo al servidor y “darlo de alta”

en la base de datos, de esta forma sin tocar una línea de código el juego ya estaría disponible tanto para los usuarios como para los jugadores, esto está pensado para que en el caso de usarse en entornos cerrados o privados, cualquiera con unas mínimas indicaciones que se darán en el apartado de juegos, podría subir más (Información de los juegos en Anexo I).



6) Diseño e imagen corporativa



Tanto la imagen corporativa como el diseño general del sitio web, se han desarrollado siguiendo una línea sencilla y fluida.

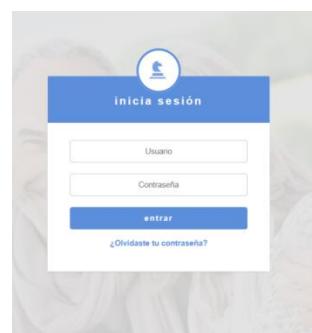
El logotipo simboliza la perdida de la memoria, los recuerdos que se van diluyendo a lo largo del tiempo, a la misma vez que la silueta principal tiene una forma basada en la de un cerebro de una manera bastante sutil.

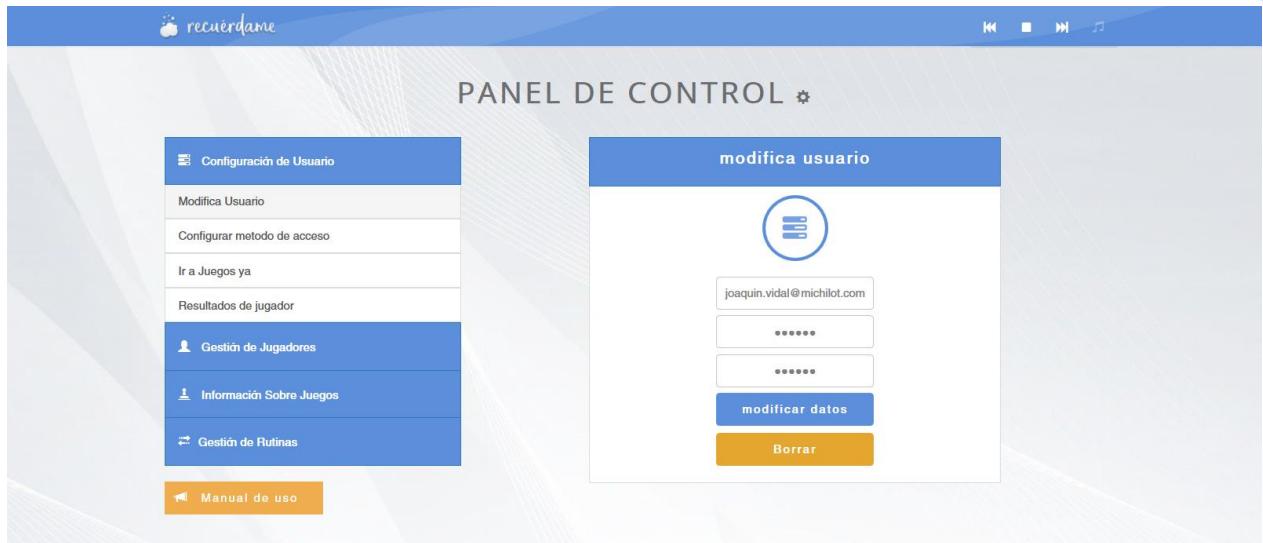
Centrándonos en el estilo de la web propiamente dicha, hemos pretendido un fácil manejo y usabilidad tanto desde escritorio como desde tablet. Con botones y campos grandes y bien

diferenciados, usando una gama de colores que haga la experiencia de usuario lo más agradable y relajada posible, pensando principalmente en los jugadores que más inconvenientes puedan tener a la hora de concentrarse en el entrenamiento con los juegos. En base a esto, pese a que el diseño lleve trabajo detrás, no tendrá una imagen “recargada” ni a priori impactante, pero si lo más funcional y acorde posible, lo más limpia posible para evitar las posibles distracciones que antes comentábamos.

Lo primero que veremos en el index, serán una página de inicio corporativa, en la que podremos ver unos backgrounds cambiar con el efecto “fade in” cada pocos segundos. Mediante AJAX cargaremos los formularios de login, registro y recuperación de contraseña.

Una vez el cuidador ha iniciado sesión, se encontrará en el panel de control, en el cual tendrá acceso a un manual con indicaciones mínimas y un video tutorial explicando más detalladamente el funcionamiento de las diferentes opciones de dicho panel.





Como se puede ver, las opciones del menú son bastante intuitivas, igualmente, una imagen vale más que mil palabras, así que la mejor manera de explicarlo será [dirigiéndonos al manual](#).

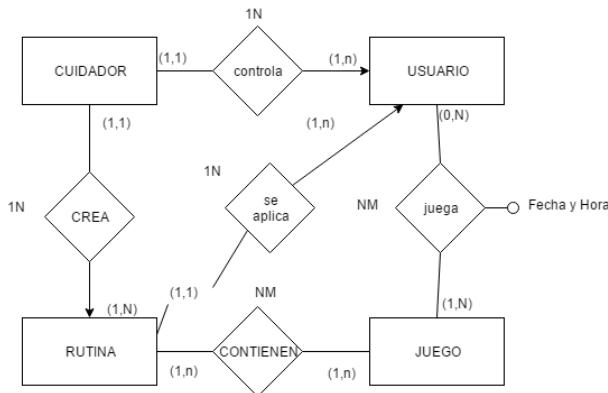
Se ha recurrido a funciones de librerías de JavaScript como **jQuery**, **jQuery UI** y **js Parallax** para conseguir todos los efectos deseados, como flip-flop, listas ordenables arrastrando, parallax, etc...

7) Mongodb.

Una de las primeas decisiones que tomamos en nuestro afán de hacer las cosas de forma distinta a lo que habíamos visto en clase fue la elección de una base de datos no relacional (si las usan Google y Facebook algo bueno tendrán) después de estudiar varias, seleccionamos Mongodb como nuestra base de datos.

Como ya hemos dicho Mongodb es una base datos no relacional (mal llamada NOSQL) orientada a documentos. A grandes rasgos son bases de datos para manejar grandes cantidades de datos con una mínima relación entre ellos. La principal característica es la libertad en el diseño. Imaginemos un restaurante con 1000 clientes, si sólo 1 de ellos tiene una alergia ¿Por qué tengo que tener 999 registros con un campo nulo o vacío? ¿Cuánta red y memoria desperdiciamos mandando datos nulos o vacíos? Partiendo de esta idea Mongodb fabrica documentos (los registros en mysql) a medida, si un cliente tiene 5 teléfonos ese cliente tendrá 5 campos teléfono sin que eso implique que el resto tenga que tenerlos. Esta libertad es a costa de la pérdida de relaciones, si no todos los documentos tienen el mismo número de campos y ni siquiera sabemos que los campos "clave" van a estar en todos, no se puede exigir que uno en particular se relacione con otra colección (tabla en mysql) y como consecuencia la integridad referencial se debe mantener por código.

Esta decisión supuso el primer gran cambio en el diseño de la aplicación, todo el diseño que teníamos pensado para una base de datos relacional de la noche a la mañana se convertía en un conjunto de colecciones que no se podían relacionar.



En la página oficial (véase webs y referencias), se puede encontrar software para montar muy fácilmente un servidor local para poder hacer las pruebas. La librería necesaria para php se instala con el composer y en la página oficial de php explica los pasos.

Esta elección desembocaría más adelante en el mayor problema con el que nos hemos enfrentado que con un 90% del proyecto hecho nos hizo plantearnos cambiarlo a mysql (no se puede entender si la parte del servidor, por lo que lo explicaremos en el siguiente apartado). No encontramos ningún servidor gratuito que integrará apache + php + mongodb, con lo que tuvimos que separar mongo de la web, tampoco fue fácil encontrar un servidor mongodb gratuito, al final encontramos [mlab](#), que es donde se aloja nuestra base de datos. La mayoría de los servidores tradicionales, al tener maríadb integrado no permite trabajar con otra clase de bbdd y algunos ni siquiera con una base de datos que esté alojada fuera, es por lo que al final nos decantamos por:

8) Heroku

Este servidor cumplía con las condiciones antes hemos descrito, trabaja con apache, php y permite conexiones a base de datos externas, a parte cumplía otra de nuestras condiciones, se sale de lo habitual.

La primera “rareza” que tiene Heroku es el método de subida, mientras que los servidores convencionales usan ftp para subir las páginas, e incluso desde sus paneles principales tienen herramientas para este fin, al crear una “aplicación” en Heroku este nos asigna un repositorio git, donde hay que subir nuestra página por medio de cualquier herramienta (nosotros usamos [Git Bash](#) y [SourceTree](#)). Al ser un servidor que puede ejecutar aplicaciones en un gran número lenguajes de programación, hay que indicarle desde el panel de control que vamos a utilizar, en nuestro caso php y mlab, y aquí viene la segunda parte de nuestro problema.



Los servidores “convencionales” el php.ini contiene la información sobre las extensiones (en forma de archivos .dll), de forma que por ejemplo en nuestras pruebas locales, aparte de descargarnos la librería Mongodb, nos descargamos y enlazamos el archivo php_mongodb.dll y funcionaba todo. Cual fue nuestra sorpresa cuando al subir todo a Heroku y Mlab daba error de que no encontraba unas clases del controlador de mongo y por tanto no se podía comunicar con la base de datos, con lo que no funcionaba nada. Como ya hemos dicho anteriormente este fue el mayor reto al que nos hemos enfrentado. Tardamos 15 días, 2 sistemas operativos, y cerca de 50 mails para encontrar la solución, el problema está en que al no tener un php.ini Heroku no carga las extensiones, y aunque tengamos instalada la librería falta “la mitad” del controlador. Si recordamos para instalar mongodb teníamos que usar el Composer, pues bien, al instalar algo con Composer, se crean un par de archivos en formato json que indican las extensiones que necesitan, para que heroku instale esas extensiones hay que poner ese archivo creado en el directorio principal, junto al index y copiando estos dos archivos al directorio principal, todo funcionó perfectamente.



Tenemos que decir que a pesar de las peculiaridades de Heroku y Mlab, las versiones gratuitas funcionan perfectamente y no hemos notado ni cortes de servicios ni latencias excesivas.

Finalmente, este es el dominio donde tenemos alojado Recuérdame: recuerdos.herokuapp.com

9) Extras

Reproductor

Tanto como para el cuidador mientras se mueve por el panel de control, como para el usuario mientras entrena con los juegos, en la barra superior de la pantalla tendremos acceso en todo momento a un sencillo reproductor. Dispondremos de una selección de piezas de corte clásico que se han creído convenientes para la ocasión, auto-reproduciéndose una tras otra. Pudiendo pasar de canción cuando apetezca. Mientras esté sonando la música, aparecerá un indicador parpadeante a la derecha de este mismo reproductor: un icono de una doble corchea.



Recuperación de contraseña



Se ha configurado un método fácil y muy seguro de recuperar la contraseña, escribiendo tu usuario cuidador y el e-mail con el que te registraste, enviará un correo a dicha cuenta de correo con tu contraseña. Para esto hemos usado la librería phpmailer que como su nombre indica

facilita el envío de mails desde php. Su uso es muy sencillo se crea un objeto y vamos dándole valores a las propiedades (`mimail->setfrom("")`, `mimail->addCC("", etc)`) y finalmente se lanza la función enviar. Como ya explicamos arriba es compatible con la api de Gmail o como en nuestro caso servidores “propios”.

Excel con progresos de los jugadores

Como ya se ha comentado, uno de los cometidos principales de Recuérdame es poder ver y analizar los progresos que hace un jugador día a día, ver si mejora tiempos, consigue avanzar en juegos que antes no podía, alcanzar mayores niveles, etc... Para esto hemos considerado que una manera fácil de filtrar y manejar a tu antojo dicha información, es poder contar con ella en un fichero excel, para ello se ha recurrido a la librería PhpExcel, la cual nos permite imprimir excel tanto con una tabla html, o modificando las celdas con métodos de la misma librería. En este caso, al tratarse de una tabla sencilla que ya tenemos hecha para la misma web, hemos optado por crear un Html Reader en vez de trabajar directamente con el objeto PhpExcel.

 [descargar excel](#)

10) Conclusiones

Ahora que hemos echado la vista atrás y repasado el trabajo de estos últimos meses, nos sentimos tan orgullosos como gratificado por el trabajo realizado y la recompensa obtenida. Cuando empezamos a plantear el proyecto, teníamos muy claro que queríamos hacer algo que sirviese para algo o para alguien, nuestra idea no era un despliegue tecnológico y visual que al terminar la presentación cayese en el olvido, pensamos en proyectos de usos sociales o académicos y finalmente nos decantamos por este, entre otras cosas porque hay muy poco software dedicado a las personas mayores y no digamos para personas con alguna enfermedad o discapacidad mental. ¡Lo hemos conseguido!, “**Recuérdame**” es una herramienta válida y útil tanto para las personas mayores como para cuidadores o incluso terapeutas.

La parte gratificante del proyecto es que nos ha hecho crecer como programadores, y no nos referimos a que hayamos aprendido mongo o a usar una librería, eso se puede aprender en cualquier momento de un manual o vídeo. Lo que más nos ha enseñado ha sido la parte sin ordenador, la planificación del proyecto (para quien era, que se incluía, que tipos de colores y sonidos, etc), el trabajar en equipo todos los domingos teníamos reunión por Skype para informarnos cómo íbamos, que problemas encontrábamos, tomar decisiones... Pero sobre todo el enfrentarnos a problemas que hemos tenido que investigar y solucionar porque para que el proyecto avanzara sin tener que cambiar de objetivos (no podemos evitar acordarnos de nuestro amigo Abdul del servicio técnico del server MLab).

Se nos ha quedado clavada la espinita del juego de la familia, pero las fuentes legales consultadas y las latencias de los servidores no han permitido que el juego esté a la altura del resto del proyecto (versión 2.0 is coming).

11) Webs y referencias.

- [Php.net](#) : web de referencia para el lenguaje y drivers.
- [Mongodb.com](#): web de referencia para el servidor y lenguaje para las bases de datos no referenciales MongoDB.
- [Heroku.com](#): Servidor de aplicaciones y webs no convencionales, con manuales para el despliegue y uso.
- [Mlab](#): servidor de MongoDB con manuales y herramientas para el uso de la base de datos.
- [Stackoverflow.com](#): web de consultas concretas.
- Canal [codigofacilito](#) curso de mongodb
- [Pixelcog.github.io](#): librería y tutoriales para el parallax.
- [MDN, developer mozilla](#): tutoriales de canvas, funciones de jQuery.
- [Jqueryui.com](#): funciones de la librería jQuery UI.

