

# BRAINHUB

---

Por: Ismael Moreno, Santiago Daza y Wei Xu

CURSO 2023/2024

2º DAW

# Índice

1. Introducción .....	3
2. Palabras Clave y Tabla de Acrónimos y Conceptos Clave .....	4
3. Módulos y Herramientas aplicadas .....	5
4. Componentes del equipo y aportación realizada por cada estudiante .....	6
5. Fases del Diseño .....	7
5. 1. Estudio y Diseño UI de la Aplicación Web .....	7
5. 1. 1. Sprints 1 y 2 DIW - Investigación y Arquitectura de la Web .....	7
5. 1. 2. Sprint 3 DIW - Wireframes .....	10
5. 1. 3. Sprint 4 de DIW - Sistema de Diseño .....	12
5. 1. 4. Sprint 5 DIW - Mockups e IxD .....	16
5. 2. Desarrollo de BrainHub en Cliente (Front-end) y Servidor (PHP) .....	19
5. 2. 1. Sprint 1 Entorno Cliente y Servidor .....	19
5. 2. 2. Sprint 2 Entorno Cliente y Servidor .....	21
5. 2. 3. Sprint 3 Entorno Cliente y Servidor .....	24
5. 2. 4. Sprint 4 Entorno Cliente y Servidor .....	27
6. Conclusiones .....	35
7. Bibliografía .....	36
8. Anexos .....	37

## **1. Introducción**

En la actualidad, el incremento de problemas de ansiedad relacionados con el uso de las redes sociales, experiencias traumáticas, y otras situaciones cotidianas, se ha convertido en una preocupación creciente para la sociedad. Frente a este escenario, nace BrainHub, una solución innovadora en forma de aplicación web enfocada en el ámbito de la psicología. Su principal objetivo es ofrecer un espacio seguro y accesible donde los usuarios puedan encontrar apoyo y herramientas para manejar y tratar los diversos tipos de ansiedad.

Concebida con la premisa de la facilidad de uso, BrainHub se presenta como una plataforma intuitiva que dispone de una serie de funcionalidades diseñadas para maximizar su utilidad. Entre ellas, destacan dos modalidades de acceso: una versión gratuita que permite el uso de la mayoría de las herramientas y recursos, y una opción premium, que amplía el acceso a contenidos exclusivos y funciones adicionales.

Una característica fundamental de BrainHub es su enfoque comunitario y de soporte personalizado. Los usuarios pueden contactar directamente con terapeutas para agendar citas online, filtrando por diversas características para encontrar la opción que mejor se ajuste a sus necesidades. Además, la plataforma fomenta la creación de un entorno comunitario mediante foros, donde los usuarios pueden compartir experiencias, consejos y ofrecer feedback mutuo.

La motivación detrás de la creación de BrainHub surge de la necesidad de abordar de manera efectiva los desafíos de la salud mental en la era digital, proporcionando un recurso accesible que empodere a los individuos en su búsqueda de bienestar psicológico. Con este proyecto, aspiramos a alcanzar objetivos generales tales como mejorar la conciencia y comprensión sobre la ansiedad, facilitar el acceso a intervenciones psicológicas y fortalecer la comunidad de apoyo mutuo entre personas que enfrentan desafíos similares.

## **2. Palabras Clave y Tabla de Acrónimos y Conceptos Clave**

- Ansiedad
- Salud Mental
- Terapia Online
- Redes Sociales
- Bienestar Psicológico
- Comunidad de Apoyo
- Intervención Psicológica
- Aplicación Web

Acrónimo/Concepto	Descripción
UI/UX	User Interface (Interfaz de Usuario) / User Experience (Experiencia de Usuario), referente al diseño y experiencia general del usuario al interactuar con la aplicación web.
API	Application Programming Interface, conjunto de rutinas, protocolos y herramientas para construir software y aplicaciones.
SQL	Structured Query Language, lenguaje de programación utilizado para la gestión y manipulación de bases de datos.
PHP	Hypertext Preprocessor, lenguaje de script del lado del servidor utilizado para el desarrollo web.
JS	JavaScript, lenguaje de programación utilizado para crear contenido interactivo en sitios web.
HTML	HyperText Markup Language, estándar que se utiliza para crear y presentar contenidos en la World Wide Web.
CSS	Cascading Style Sheets, tecnología de hojas de estilo utilizada para describir la presentación de un documento escrito en HTML o XML.
MVP	Minimum Viable Product, producto con las características mínimas necesarias para satisfacer a los primeros usuarios y proporcionar retroalimentación para futuros desarrollos de producto.

### **3. Módulos y Herramientas aplicadas**

Para este proyecto se usaron diferentes tecnologías a modo de integración entre los distintos módulos del curso.

Por parte del módulo de Diseño de Interfaces Web, se usó Figma para todo el diseño de los Wireframes, los componentes, la paleta de colores y los Mockups. En esta plataforma web se pueden realizar diseños interactivos intuitivos, como una herramienta de diseño y al mismo tiempo una herramienta para ofrecer funcionalidad sobre esos diseños.

Por parte del módulo de Entorno Cliente, se programó en HTML, CSS y JS. El uso de HTML para diseñar la arquitectura de las páginas, y el CSS para ofrecer un estilo con colores y animaciones que reflejan la imagen de la empresa. Con JS se han programado las validaciones de formularios, fetch para actualizar la información de la página sin necesidad de refrescar la página, y manejo de eventos para recoger datos.

Por parte del módulo de Entorno Servidor, se trabajó con PHP y SQL, sobre todo para ofrecer una conexión con la base de datos e imprimir los datos en las distintas páginas. Con PHP se pudieron realizar validaciones también, comprobando que no se duplicasen datos en la base de datos, y haciendo funcionar la web para pedir citas, borrarlas, o editar información. Todas las sentencias para realizar esto fueron programadas en SQL para comunicarnos con la base de datos.

## **4. Componentes del equipo y aportación realizada por cada estudiante**

El proyecto BrainHub fue desarrollado por un equipo compuesto por Ismael Moreno, Wei Xu y Santiago Daza. La organización y asignación de tareas se gestionó a través de Notion, lo que permitió un reparto equitativo del trabajo y una colaboración eficiente. A continuación, se presenta un resumen de las contribuciones principales de cada miembro:

### **Ismael Moreno:**

- **Backend:** Desarrollo en PHP, especialmente en funciones relacionadas con la base de datos.
- **Frontend:** Contribución en JavaScript para funcionalidades interactivas.

### **Wei Xu:**

- **Diseño:** Liderazgo en la estilización CSS, enfocado en la interfaz de usuario y experiencia visual.
- **Frontend:** Apoyo en JavaScript junto con el equipo.

### **Santiago Daza:**

- **Backend:** Implementación en PHP para gestión de sesiones y seguridad.
- **Frontend:** Desarrollo en JavaScript, asegurando la interactividad de la aplicación.

### **Herramienta de Gestión de Tareas:**

**Notion:** Utilizado para la organización del proyecto, permitiendo una asignación clara de tareas y seguimiento del progreso de manera eficiente.

## 5. Fases del Proyecto

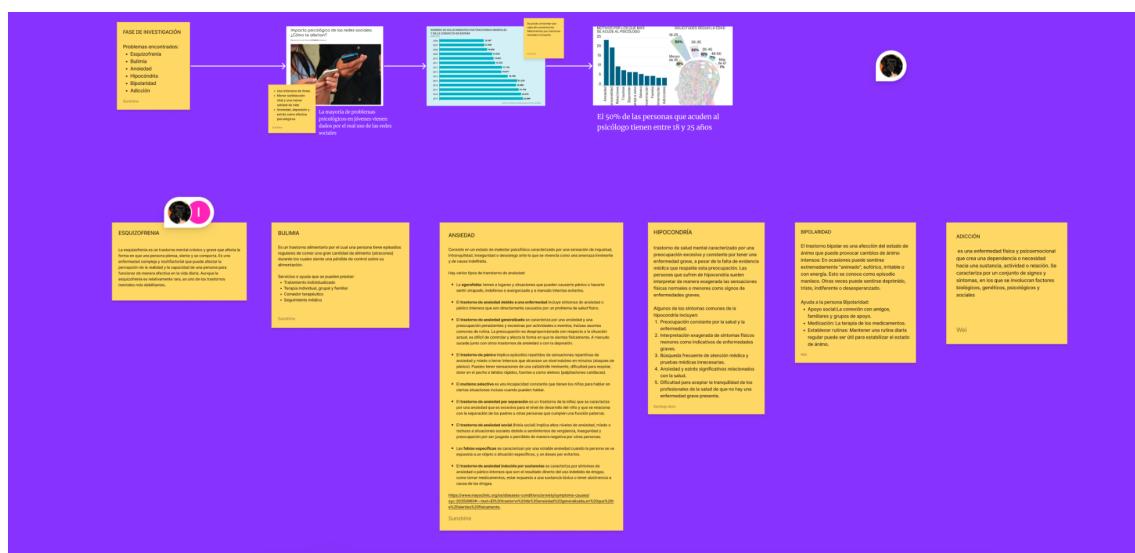
### 5. 1. Estudio y Diseño UI de la Aplicación Web

#### 5. 1. 1. Sprints 1 y 2 DIW - Investigación y Arquitectura de la Web:

Llegamos al primer sprint del proyecto, sólo en la parte de diseño de interfaces de momento. Nos expusieron el proyecto que tendríamos que hacer y estuvimos preparando los assets en Figma.

Dado que la temática del proyecto es una aplicación web de psicología, El equipo se centró en buscar los trastornos más comunes entre los jóvenes, para poder dirigirnos a ese público que hoy en día son quienes están más conectados a internet.

Después de la búsqueda, se recopilaron los datos en el Figma.



Nos hemos centrado más en la ansiedad, ya que es un tema muy generalizado y que da mucho juego. A Partir de ahí, ampliamos más el post-it de la ansiedad, buscando todos sus tipos, los síntomas, etc.

Además, hemos buscado páginas web relacionadas con el tratamiento psicológico online, y hemos analizado puntos fuertes y cosas a mejorar, para más tarde poder

aplicarlo a nuestro propio proyecto. Entre estos puntos fuertes se encuentran las citas online, que hemos aplicado de cara a nuestra aplicación.

Una vez terminada la comparación con la competencia, hemos diseñado los perfiles de usuario. En total hicimos tres, uno por cada integrante del equipo. Estos tres perfiles ocupan diferentes tipos de personas. El primero, el que podría ser hacia un perfil más de niño, como por ejemplo el mutismo selectivo que desarrollan algunos niños y que consiste en no poder hablar en determinadas situaciones. Este perfil podría ir bien en nuestra página web, con sesiones con los profesionales en el tema de todo el mundo y con ejercicios y recursos disponibles para ayudar. Además hay foros de la comunidad donde se pueden acostumbrar a expresarse y cada vez atenuar la enfermedad.

El segundo perfil ocupa a un adulto de mediana edad que podría tener trastorno del pánico. Este trastorno se caracteriza por ataques de pánico que dejan inmóvil a la persona e incluso con síntomas físicos como temblores, convulsiones y sudores fríos. Este perfil de usuarios podrían encajar en nuestra web al tener disponibles recursos para calmarse frente a un ataque, así mismo como foros comunitarios para hablar con personas que tengan los mismos patrones y llegar a una posible solución.

El tercer perfil ocupa a un joven con ansiedad social. Este es el trastorno de ansiedad más generalizado, y es que hoy en día los jóvenes tienen ansiedad de ser juzgados en situaciones sociales, como en las redes o con sus amigos.

En nuestra web también tienen su sitio, y es que pueden contactar con terapeutas especializados en la materia y hablar con la comunidad. También tienen grupos de apoyo con personas que tienen el mismo trastorno, y de esta manera intentan calmar la ansiedad.

Una vez definidos los perfiles de usuario, empezamos a hacer los User Persona, y la verdad que tuvimos bastantes problemas en esta parte. Como siempre, cada uno ha hecho su User Persona, pero por malentendidos con la tarea, no lo enfocamos del todo a la psicología sino que fuimos a más ámbitos. Por lo tanto, estábamos cubriendo problemas que no iba a resolver nuestra web de psicología.

Después de la primera entrega, tuvimos que corregirlo ya que estaba todo mal. Así que esta vez sí nos centramos en la psicología y dimos con ciertos puntos que nuestra web podría cubrir. Por ejemplo, podríamos ofrecer ejercicios de apoyo, para conocer procedimientos y respuestas a los síntomas. También podríamos ofrecer recursos online para que los usuarios pudiesen avanzar aunque no contratasen los servicios del terapeuta.

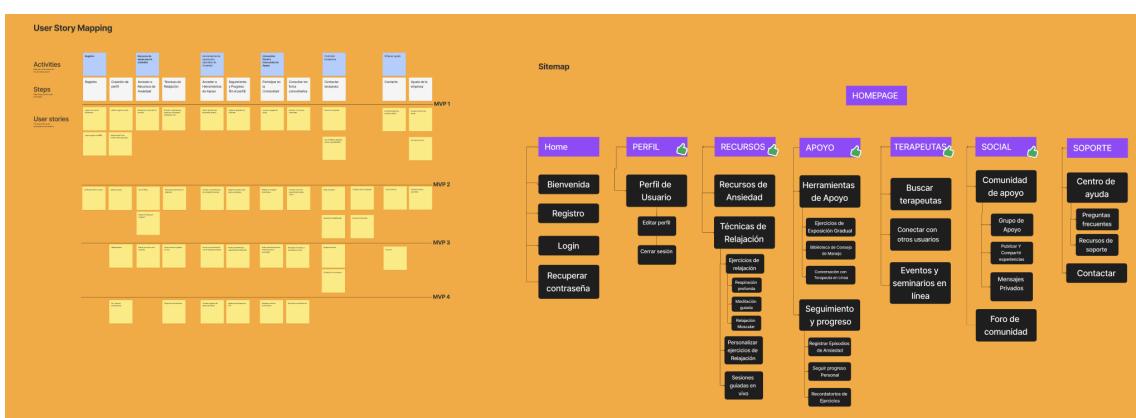
Con esto, se empezó a trabajar los MVPs de la web. Al principio no se sabía muy bien qué hacer pero con las ideas de los User Persona, se pudo empezar a planificar mejor la web. Probablemente habría que cambiar más cosas en el futuro, pero se sacó una web con buenas ideas.

La principal era que la web está dividida en dos perfiles: terapeuta y cliente. Así, los terapeutas se podrían registrar y ofrecer sus servicios, mientras que los clientes verían esos servicios y los podrían contratar.

Planificando la web así, salieron varios problemas. Habría que hacer dos webs diferentes, podría no haber regulación para los terapeutas y además no habría mucho tiempo.

Los MVPs se hicieron sin pensar en eso, pero más tarde se cambiarían.

El sitemap fue un poco más de lo mismo. Como no estaba muy clara la idea de hacer el perfil para terapeutas o no, se dejó como que sí pero en un futuro se podría editar.

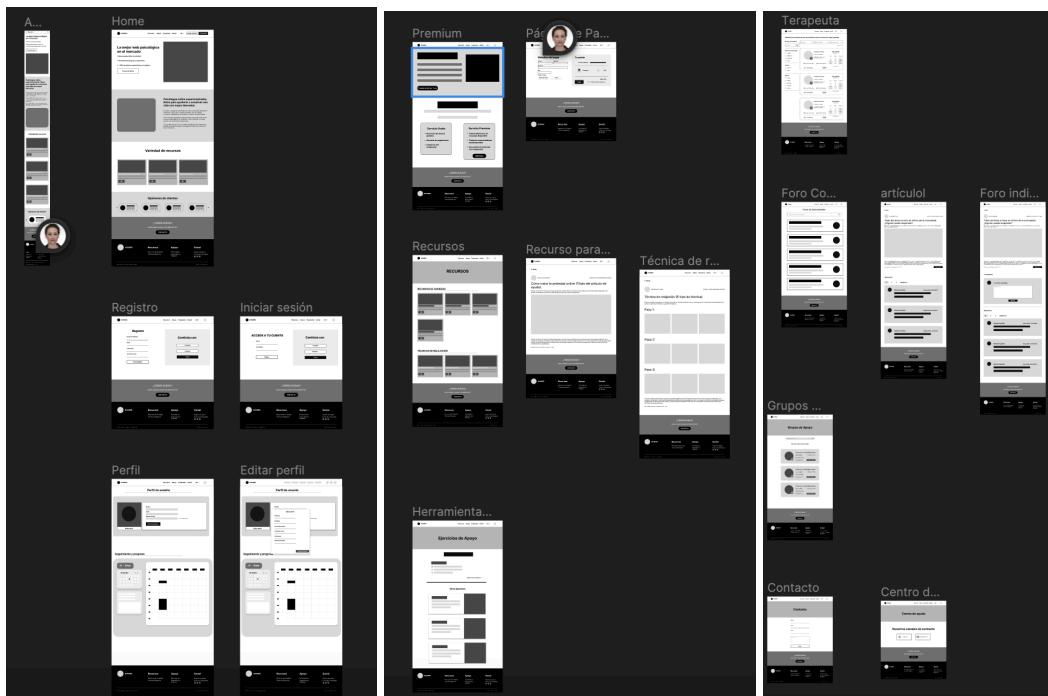


## **5.1.2. Sprint 3 DIW - Wireframes**

En este sprint se realizaron los wireframes. La idea de la web y su diseño estaban claros, así que enseguida se empezó a trabajar en ello. Se pudo hacer una web bastante amplia. El único problema fue el de los terapeutas. Primero se sugirió hacer los dos perfiles para hacer un mejor proyecto. El equipo estaba de acuerdo pero cuando llegó el momento de los Wireframes se vio que era muy difícil por tiempo y magnitud hacer ambos perfiles.

Se llegó a una idea en común que sugería que los terapeutas eran contratados por la compañía y ofrecían sus servicios en la web. Pero todo gestionado por los propios creadores de la web. De esta manera, los terapeutas no perdían su visibilidad y sus ingresos, pero sí era un producto capaz de desarrollarse a tiempo.

Con esto claro, los Wireframes eran más fáciles de hacer y más rápido. El reparto fue equitativo. Con la investigación que se había hecho de la competencia, también se cogieron algunas ideas para los diseños.



Hubo problemas en este sprint, ya que había un plan para la web que luego se tuvo que ampliar a medida que se hacían más páginas de los Wireframes. Para empezar, la Home estaba bastante clara, pero todo lo que había en ella era interactivo, lo que resultó en tener que hacer más webs para todos aquellos posibles links.

Además, con los recursos y foros, había que hacer unas dos o tres pantallas más, para las respuestas en los foros, para los distintos tipos de recursos, para cada artículo en específico, etc.

No supuso mucho problema, pero la verdad es que el mismo día de la entrega hubo que arreglar pequeños detalles y hacer alguna pantalla extra no planificada. Pero esto de hecho fue un punto a favor, ya que la entrega fueron unos Wireframes bastante detallados que se calificaron como un 10.

El motivo de estos detalles era para en el futuro tener que aplicar sólo color a los Wireframes e ir más ligeros, ya que estos sprints anteriores había muchos contratiempos que daban algún que otro susto y al final había que ir con prisas.

Resumiendo los Wireframes, optamos por un estilo formal pero sencillo, que atrajese a los jóvenes y fuese atractivo e intuitivo a la vista. No se sabían aún los colores principales, pero se sabía que serían colores pastel o claros, y jugando con la claridad de la web se podían hacer buenos contrastes.

### **5. 1. 3. Sprint 4 de DIW - Sistema de Diseño**

En este Sprint se buscaba crear la Guía de Estilos y el Kit de Componentes. Esto quiere decir que habría que hacer el logo, definir los colores principales, definir la fuente de letra, definir el tamaño de letra, definir las imágenes (o el tipo) que se van a usar y por la parte de componentes, definir inputs, botones y tarjetas.

En este punto, los miembros del equipo propusieron varios colores. Santiago propuso azul y verde, mientras que Ismael proponía el morado como color principal y el verde para el color de contraste. A Wei le daba igual así que se decidió entre esos colores. Por una parte, Ismael argumentaba que el azul emite tristeza, mientras que el morado suele estar asociado a la inteligencia, la formalidad y en otras páginas, a temas relacionados con el cerebro y psicología. Por otra parte, Santiago sugería que el morado queda apagado respecto a un azul brillante, y que el morado no está asociado con la psicología. Finalmente se hizo un mix y se usaría el morado como color principal y el verde para resaltar.

Usando proporciones de 60% (blanco), 30% (morado) y 10% (verde) se harían los botones e inputs. Por una parte, era sencillo hacer los botones con estos colores, pero al final no quedaban bien en la web, ya que eran dos colores muy distintos y no sintonizaba bien.

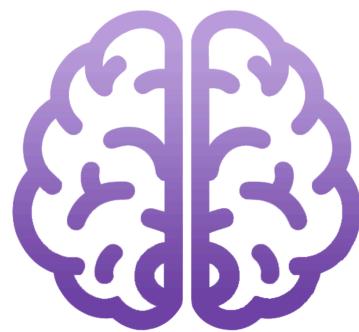


El nombre de la web se decidió en este momento. Aunque se estuvo mirando nombres relacionados con la terapia, como Therapy, así como Seraphy (de Serenidad y

Therapy) o algo relacionado, la idea era hacer un cerebro de imagen, algo minimalista y fácilmente reconocible. Por lo tanto, la primera palabra sugerida era Brain. Como el público objetivo sobre todo son los jóvenes, la otra parte del nombre sería Hub, ya que hay muchas redes sociales que usan este nombre al final, como un lugar amistoso para estar. El nombre final sería BrainHub.

Siguiendo con la idea del cerebro, se optó por un logotipo minimalista ya que inspira profesionalidad y estilo. Además, a los jóvenes les suele atraer más un logo minimalista y es más reconocible. También se optó por usar el lila en todo el logo, ya que el cerebro generalmente está representado en un color lila/rosado, y que además coincide con nuestra paleta de colores e inspira elegancia y conexión emocional (perfecto para una web de psicología).

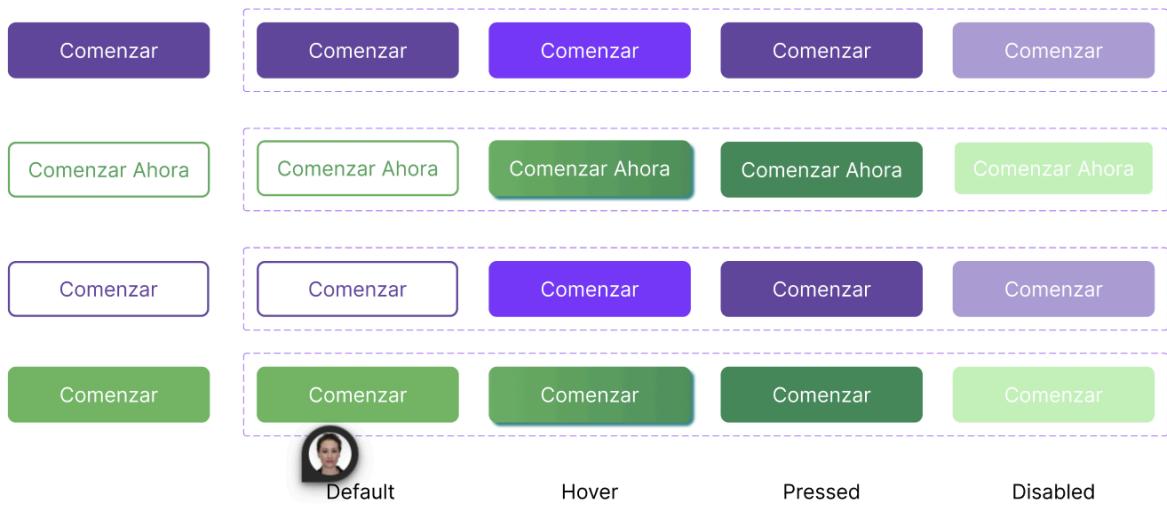
Aquí está el logo acabado:



Se usó el degradado de tonos morados al final, para que no fuese tan simple el logo, ya que la mente humana no lo es.

En la parte de kit de componentes se optó por un botón ni muy redondo ni muy recto. Los botones serían rectangulares con un border radius de 12px. De todas formas dependiendo del diseño cambiaría o no. El botón principal, para registrarse o comenzar, sería verde, el color de contraste, para llamar la atención. El resto de botones, morados, pero los secundarios, sin relleno, sólo el outline.

## Botón



Aquí hubo un problema porque la primera entrega iba sin botón disabled y con los botones con distintas formas. No había una sintonía entre ellos. Finalmente optamos por este border de 12 px en todos.

Los inputs seguirían este estilo. La barra de búsqueda también tiene 12px de border radius. Se optó por hacerlos todos negros, ya que realmente no se sabía qué variante de color se usaría en cada página.

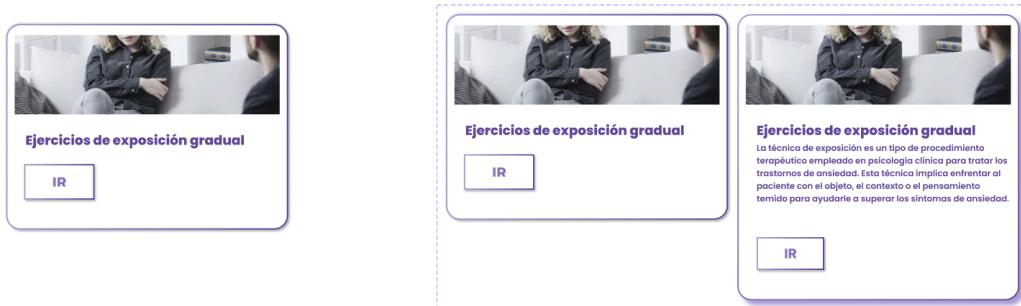
Un aspecto que sí estaba muy claro de los inputs es que cuando estaban focus, seleccionados, el texto se iría arriba a la izquierda. Haciendo una investigación se vió que casi todos los inputs de las webs más concurridas, como Netflix, Google o Amazon, lo hacían así. Este es el primer diseño de los inputs:

The image displays a series of input field prototypes. It includes a simple text input labeled 'Input', a 'Nombre y apellidos' field with a placeholder 'Nombre y apellidos' and a green border, a 'Contraseña' field with a placeholder 'Contraseña' and a red border, a 'Contraseña' field with a placeholder 'Contraseña' and a red border, a 'Email' field with a placeholder 'Email' and a green border, and a 'Buscar' field with a placeholder 'Buscar' and a green border.

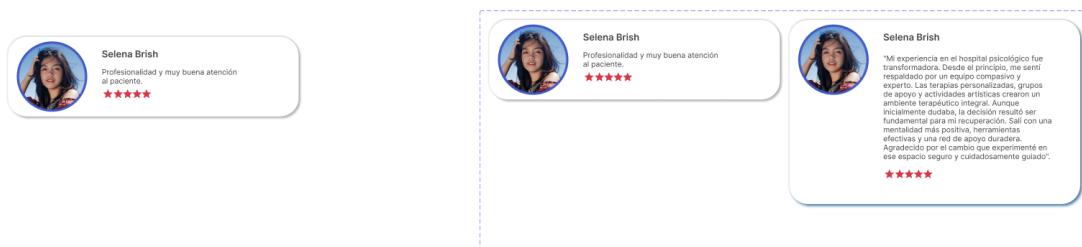
La variante de color que sí estaba clara era la de éxito o error, ya definidos en la paleta de colores.

Por último, las tarjetas también quedaron definidas como rectángulos con border radius. En este caso, tendrían más border radius que los botones o los inputs. Se usarán principalmente para las opiniones y para los recursos.

#### Tarjeta de Recursos



#### Opiniones de Clientes



En el tema de imágenes, se decidió usar imágenes en vez de ilustraciones ya que son más reales y queremos que los usuarios tengan esa impresión de que la web es profesional y que es un servicio con profesionales y técnicos de la salud mental. Muestra situaciones fácilmente reconocibles y que el usuario recordará.

### **5.1.4. Sprint 5 DIW - Mockups e IxD**

En este sprint se pedía hacer los mockups partiendo de los Wireframes y el Sistema de Diseño. Esto quiere decir que había que usar la paleta de colores y los componentes para dar estilo a los Wireframes hechos previamente en el Sprint 4. Además, había que detallar más, casi como si fuera un reflejo de la web final.

En este aspecto los Wireframes de BrainHub estaban bastante completos. Sólo había que cambiar los colores y los componentes.

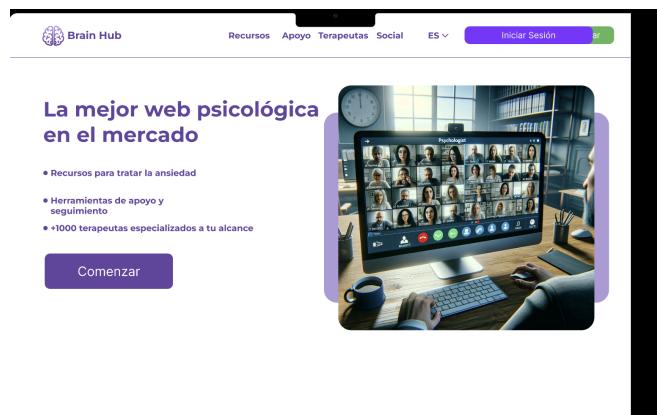
También, se pedía el Prototipo Interactivo. Esto quiere decir, que los enlaces y componentes del Mockup, tenían que comportarse tal y como se comportarían en la web. En el caso de la web de BrainHub, tenía muchos enlaces y elementos interactivos, pero todo se podía automatizar bien así que no se esperaban muchos problemas.

Los problemas llegaron en el color. Los Wireframes estaban muy bien hechos, pero a la hora de darle color, había mucho lío con el verde, el morado, el negro... Al final, decidimos poner mucho morado, y dejar el verde para casos muy especiales.

De todas formas, la entrega no quedó muy bien. Durante las dos semanas para hacerlo, el reparto fue el siguiente:

- Todos harían sus Wireframes a Mockup
- En común, haríamos el prototipo interactivo

Parece que el Sprint 4 dejó ciertos problemas en los componentes. A la hora de meterlos en el mockup, perdían su forma original.

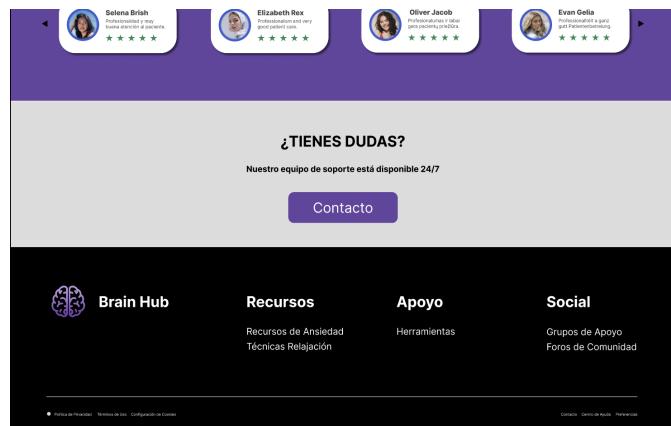


Se fue el tiempo intentando arreglar estos errores y al final no se pudieron arreglar todos.

En cuanto al color, como comentaba antes, se nos hizo un lío. Se quería usar mucho morado pero sin dejar de lado el blanco y el verde. Aún así, en páginas como esta, seguía quedando mucho morado, pero se dejó porque parecía que quedaba bien.

Se decidió usar los colores de una forma variable. En algunas páginas se usará mucho y en otras no tanto. De todas formas se seguiría un mismo estilo, pero vimos webs como estas y nos inspiró a no preocuparnos tanto por pequeños porcentajes:

Se decidió usar también un footer negro, como simbolizando el final de la web y quedando bien en contraste con el propio fondo de esta.



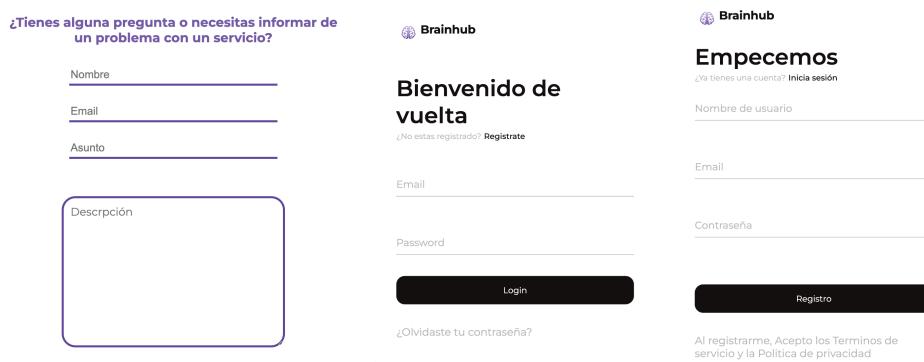
Algunos apartados como se ve en la imagen de arriba, tenían el fondo gris, otros blanco y otros morado. De esta manera se quiere dar un juego de colores, atractivo para los jóvenes.

## 5. 2. Desarrollo de BrainHub en Cliente (Front-end) y Servidor (PHP)

### 5. 2. 1. Sprint 1 Entorno Cliente y Servidor

En este Sprint se pedía realizar a partir de los Mockups, al menos tres formularios. En este punto, el trabajo se dividió con Notion. Principalmente Wei Xu se encargaría del css, y Santiago Daza e Ismael Moreno de php y js. Aún así, todos harían de todo, sin desentenderse completamente del resto de tareas.

Para este primer sprint, el equipo hizo más de tres páginas, aunque incluyendo los tres formularios: registro, login y contacto.



En este caso, Ismael hizo el contacto y Santiago Daza hizo la página de registrar.php con la animación incluida.

También, aunque no se pedía, se hizo la validación del formulario, así podíamos trabajar sin tener que esperarnos al siguiente split.

Por la parte de servidor, se hicieron todas las tablas correspondientes en crear\_tablas.php, así como conecta.php para realizar la conexión a la DB. Esta última tenía dos funciones: una para obtener la conexión sin la DB y la otra con la DB.

Al principio se hicieron 8 tablas, que fueron repartidas mitad y mitad entre Wei Xu e Ismael Moreno. Lo que no se hizo fueron los inserts, que el propio día de entrega se comentó con la profesora y se hizo para aprobar el sprint.

Además de los formularios y las páginas que los contenían, se avanzó y se hizo la página home. En esta tendría un link al registro y login, y un apartado de opiniones que conectaría con la base de datos para mostrarlo con php.



Mientras tanto, se iban haciendo todas las webs responsive para móvil.

## **5. 2. 2. Sprint 2 Entorno Cliente y Servidor**

En este sprint se pedía la validación de formulario con js y el tratamiento de datos con php para hacer inserts en las tablas de la base de datos.

Desde el primer sprint ya estaba hecho la validación de formulario. Para entrar más en detalle se hizo lo siguiente:

En el formulario de contacto hicimos validación para que el nombre tuviera el formato adecuado. No se podría escribir ni números ni mayúsculas si no era la primera letra. Además, todo validaba que tuviese la longitud establecida en la base de datos.

Esto se hace con javascript, con el siguiente fragmento de código que quiere decir una letra mayúscula y minúscula, y el resto minúsculas:

```
$(nombre).on("input", function () {
    comprobar(nombre,errorNombre,/^[a-zA-Z][a-z]*$/,false);
});
```

En el register, se tuvo que hacer validaciones extra, ya que se podían introducir datos en la base de datos y esto hacía necesarias más validaciones.

Estas serían validar que el usuario no esté registrado ya, o que el correo o el nombre de usuario no estén en uso. Además de las propias validaciones de formato del nombre de usuario, el correo y la contraseña.

Por una parte, el nombre de usuario no puede incluir números, y no puede incluir una mayúscula que no sea la inicial. El correo es el formato típico, con el @ y el .com. La contraseña tiene que ser entre 8 y 14 caracteres.

```
//comprobación automática de nombre
$(nombre2).on("input", function () {
    comprobar(nombre2, errorNombre2,/^[a-zA-Z][a-z]*$/);
});
//validación de contraseña
$(pwd2).on("input", function () {
    comprobar(pwd2,errorPwd2,/^\S{8,14}$/);
});
//validación de correo
$(correo2).on("input", function () {
    comprobar(correo2,errorCorreo2,/^\w+@[a-z]+\.[a-z]{2,3}$/);
});
```

El resto de validaciones se realizan en procesarLogin y procesarRegister.php. Se comprueba mediante un select de la base de datos que el usuario o el correo no estén ya escogidos. En cualquier caso, se avisará al usuario.

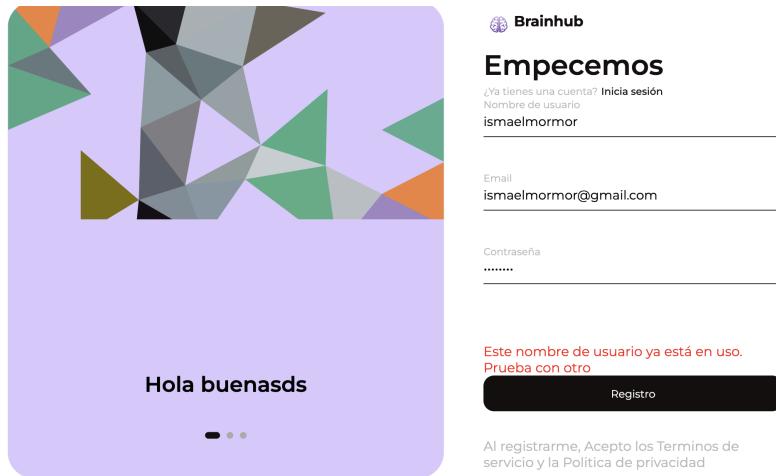
```
// Consulta SQL para verificar si el nombre de usuario ya está en uso
$sql = "SELECT * FROM usuario WHERE nombreUser = ?";
$stmt = $conexion->prepare($sql);
$stmt->bind_param("s", $user);
$stmt->execute();
$resultado = $stmt->get_result();

if ($resultado->num_rows > 0) {
    echo json_encode(['success' => false, 'mensaje' => "Este nombre de usuario ya está en uso. Prueba con otro"]);
} else {
    // Consulta SQL para verificar si el correo electrónico ya está registrado
    $sql = "SELECT * FROM usuario WHERE email = ?";
    $stmt = $conexion->prepare($sql);
    $stmt->bind_param("s", $email);
    $stmt->execute();
    $resultado = $stmt->get_result();

    if ($resultado->num_rows > 0) {
        echo json_encode(['success' => false, 'mensaje' => "Este correo ya está registrado. Prueba a iniciar sesión"]);
    } else {
        // Consulta SQL para insertar el nuevo usuario con contraseña hasheada
        $sql = "INSERT INTO usuario (nombreUser, email, password) VALUES (?, ?, ?)";
        $stmt = $conexion->prepare($sql);
        $stmt->bind_param("sss", $user, $email, $password);

        if ($stmt->execute()) {
            echo json_encode(['success' => true, 'mensaje' => "Se ha registrado exitosamente. Inicie sesión."]);
        } else {
            echo json_encode(['success' => false, 'mensaje' => "Error al registrar tu cuenta. Por favor, inténtalo de nuevo más tarde."]);
        }
    }
}
```

Y vemos el aviso:



Todo esto en un principio se hizo sin tener en cuenta la inyección de sql. Tampoco se hizo el hash de la password, que más tarde se hará.

En el formulario de login hay una validación para el formato del correo electrónico, ya que hay que comprobar que realmente escribe un correo. Esta es igual a la de register.

A parte de eso, hay validaciones para comprobar el usuario y la contraseña. Primero se comprueba que el usuario existe, y si existe, se comprueba la contraseña. Si el usuario

no existe, se avisará, pero en cualquier caso, el error que le aparece al usuario es “el usuario o la contraseña están mal”. Esto es porque hay que mantener la seguridad por si hay individuos intentando acertar el correo electrónico de una persona.

```
$conexion = getConexion();
// Consulta el usuario por su email
$sql = "SELECT * FROM usuario WHERE email = ?";
$stmt = $conexion->prepare($sql);
$stmt->bind_param("s", $email);
$stmt->execute();      You, hace 5 días • commit login funcional

$resultado = $stmt->get_result();
if ($resultado->num_rows > 0) {
    // Recorrer cada fila del resultado
    while ($fila = $resultado->fetch_assoc()) {
        $hashed_password = $fila['password'];
        $id_user = $fila['id'];
    }

    // Verifica si la contraseña proporcionada coincide con el hash almacenado
    if (password_verify($password, $hashed_password)) {
        // Si coincide, establece la sesión y redirige
        session_start();
        $_SESSION['id'] = $id_user;
        echo json_encode(['success' => true, 'mensaje' => "Inicio de sesión exitoso.", 'redirect' => 'index.php']);
    } else {
        // Si no coincide, muestra un mensaje de error
        echo json_encode(['success' => false, 'mensaje' => "El correo o la contraseña no son correctos"]);
    }
} else {
    // Si el usuario no existe, muestra un mensaje de error
    echo json_encode(['success' => false, 'mensaje' => "No se encuentra al usuario"]);
}
```

Como el hash no se introdujo en este sprint, se haría para el siguiente, junto al resto de requisitos.

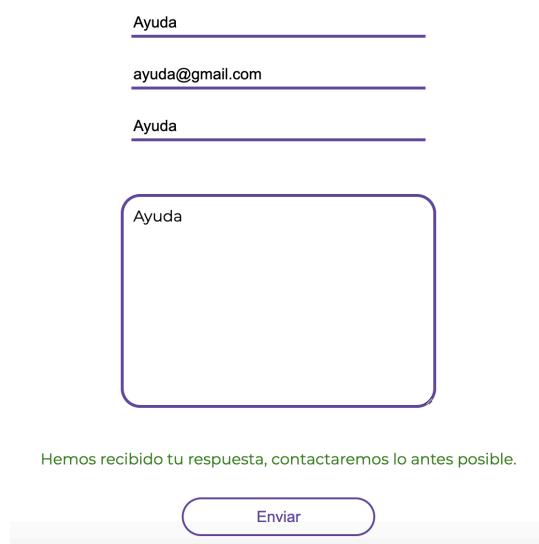
### **5. 2. 3. Sprint 3 Entorno Cliente y Servidor**

En este sprint se pedía el uso de fetch para actualizar los datos de las páginas, hacer más páginas con datos extraídos de la base de datos y el envío de objetos json para el fetch.

Este sprint fue más cargado de lo que parecía, ya que hubo que añadir muchos archivos nuevos, preparar muchas páginas para añadir contenido directo de la base de datos y preparar las consultas con statements para prevenir la inyección de sql.

Se empezó añadiendo fetch a todos los formularios que hacían consultas a la base de datos. Para empezar, el formulario de contacto.

Si se inserta correctamente la fila en la tabla de la base de datos, sale un mensaje de color verde avisando de que se realizó con éxito.



Esto se realizó de la siguiente manera:

Se aplica un fetch en el js del formulario, para que cuando se tendría que ir al evento default, se haga lo que indique dentro de la función que le ponemos. En esa función hacemos el fetch que hace un post del cuerpo del form a una página, en este caso procesarContacto.

```
var formData = new FormData(form);

// Usa fetch para enviar los datos al servidor
fetch('procesarContacto.php', {
  method: 'POST',
  body: formData
})
```

Una vez se envía a ese archivo, se procesa la solicitud dentro. Se extraen los datos del POST y se realiza la sentencia del insert. Si se realiza con éxito, lo que se mandará será un objeto json con un mensaje y un boolean, success. Si el boolean es true, significa que se realizó con éxito y por lo tanto se escribirá el texto en verde.

```
if ($stmt->execute()) {
    echo json_encode(['mensaje' => "Hemos recibido tu respuesta, contactaremos lo antes posible."]);
} else {
    echo json_encode(['mensaje' => "Ha habido un problema al enviar tu respuesta. Inténtalo más tarde."]);
}
$stmt->close();
```

Esto se recoge de vuelta en el fetch, como una response y se decodifica el json para ofrecer una actualización del contenido sin necesidad de refrescar la página.

En el resto de páginas es igual, excepto en la de login. Cuando se hace login, si se realiza con éxito, se hace un redirect al index. Así el usuario se sale del login y se cambia el nav.

```
// Verifica si la contraseña proporcionada coincide con el hash almacenado
if (password_verify($password, $hashed_password)) {
    // Si coincide, establece la sesión y redirige
    session_start();
    $_SESSION['id'] = $id_user;
    echo json_encode(['success' => true, 'mensaje' => "Inicio de sesión exitoso.", 'redirect' => 'index.php']);
} else {
    // Si no coincide, muestra un mensaje de error
    echo json_encode(['success' => false, 'mensaje' => "El correo o la contraseña no son correctos"]);
}
else {
    // Si el usuario no existe, muestra un mensaje de error
    echo json_encode(['success' => false, 'mensaje' => "No se encuentra al usuario"]);
```

Esto se hace con un redirect, que luego en el fetch se procesa y se redirige:

```
.then(data => {
    if (data.success) {
        window.location.href = data.redirect;
    } else {
```

Otro punto del sprint era prevenir la inyección del sql. Esto se hace quitando las consultas simples con las variables dentro del string de la consulta, y haciendo un statement preparado. Luego con bind params se vincula el parámetro que queremos. Al igual que con los fetch, esto es prácticamente lo mismo en todos:

```
$sql = "SELECT * FROM usuario WHERE nombreUser = ?";
$stmt = $conexion->prepare($sql);
$stmt->bind_param("s", $user);
$stmt->execute();
$resultado = $stmt->get_result();
```

Al final se había pedido también manejar la session del usuario pero no se aplicó en este sprint por parte del equipo. De igual manera en cuanto se acabó este sprint se empezó a trabajar en la sesión para poder enseñarlo al momento.

## **5. 2. 4. Sprint 4 Entorno Cliente y Servidor**

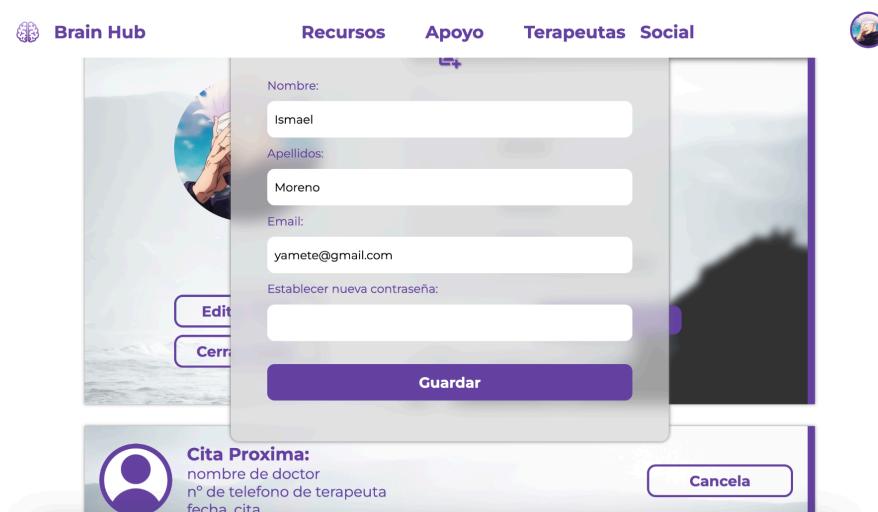
Este sprint es sin duda el que está más cargado de trabajo. En este Sprint se pide finalizar el proyecto y resolver todos aquellos errores de compilación posibles. Con lo cual, habría que terminar de agregar todas aquellas páginas que queden y montar la lógica de la web entera.

BrainHub en este punto estaba a medias. La página de login y register estaba completa. La de contacto también. Había que corregir errores de Home y alguna otra página. Pero había que hacer casi toda la web con Session ya que no la teníamos.

Creando la session con el siguiente fragmento de código, podemos mantenerla entre página y página sin necesidad de pasar la variable por POST. Así, se puede conseguir apartado como el de perfil, respuestas en foros, etc.

```
session_start();
if (isset($_SESSION["id"])) {
    $id=$_SESSION["id"];
}
```

Como en este sprint se ha hecho casi toda la web, iremos punto por punto. Empezando por el apartado de perfil, que tiene la posibilidad de ver tu cuenta, cambiar datos del perfil, como la imagen, el nombre, apellidos, email y la contraseña.



Además, se puede ver y cancelar tus citas pedidas a través de la web. Esto ayuda a organizarse aunque las citas se tengan que llevar a cabo externamente, generalmente por número de teléfono.

También tenemos el botón de hacerse premium, que te lleva a la página de premium.php, donde seleccionas el plan y te lleva al método de pago.

The screenshot shows a payment interface. On the left, a purple sidebar displays the title "Servicio Premium" and a list of benefits:

- Toda la biblioteca de recursos disponible
- Todas las características desbloqueadas
- Descuento en sesiones con terapeutas

A green button at the bottom of the sidebar says "10€/mes".

The main area is titled "Detalles del Pago" and contains fields for Name, Surname, Card Number, Expiry Date, and CVV. A green "Pagar" button is at the bottom. At the top right, it says "Premium 0 10€". Below the payment form, a section titled "¿TIENES DUDAS?" offers support information.

Como bien dice en el resumen del plan, desbloqueas el contenido premium. Y es que si no eres premium, en la página de recursos te salen bloqueados.



#### RECURSOS DE ANSIEDAD



Pero cuando compras el premium, el botón del perfil cambia a “Ya eres premium” y se desbloquea el acceso al contenido.

Aquí lo podemos ver:

## RECURSOS DE ANSIEDAD

---

**Cómo relajarse**

[ver detalle →](#)



**Cómo detectar un ataque**

[ver detalle →](#)



Lo siguiente será la página de recursos. En esta página, se muestran los diferentes artículos que son extraídos de la base de datos. La manera de mirar si son premium o no, es por la base de datos, donde hay un campo llamado “premium” que es un 0 o un 1. De igual manera está en usuario. Así que se comprueba si el usuario es premium y el artículo también y si el usuario no lo es, se bloquea.

```
$sql = "SELECT * FROM recursos";
$result=mysqli_query($conexion,$sql);
if ($result->num_rows>0) {
    while ($recurso=$result->fetch_assoc()) {
        if($recurso['premium']==1 && $premium==1){
            $link = '<a href="recursosContenido.php?id_recurso='.$recurso['id'].'">
                    ver detalle<i class="bx bx-right-arrow-alt"></i>
                </a>';
        }else if($recurso['premium']==1 && $premium==0){
            $link = '<a href="premium.php">
                    Hazte premium<i class="bx bx-lock-alt"></i>
                </a>';
        }else{
            $link = '<a href="recursosContenido.php?id_recurso='.$recurso['id'].'">
                    ver detalle<i class="bx bx-right-arrow-alt"></i>
                </a>';
        }
        echo '
            <div class="tema'.$recurso['id'].'">
                <div>
                    <h3>' . $recurso['titular'] . '</h3>'.
                    $link
                '</div>
                
            </div>
        ';
    }
}
```

También dentro del recursoContenido hay una validación donde si el usuario no es premium y está accediendo a un artículo premium (mediante escribir la url) se le redirige a premium.php para comprar el premium.

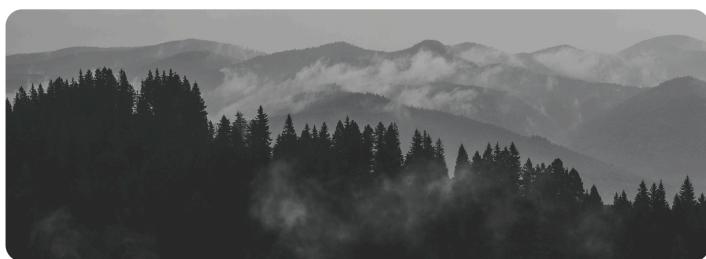
```
while ($contenido = $resulta->fetch_assoc()) {  
    // Si el usuario no es premium y el recurso es sólo para premiums, le mandamos al premium.php  
    if($premium==0 && $contenido['premium']==1){  
        header("Location: premium.php");  
    }  
}
```

La página de cada recurso tiene un banner y el titular del artículo. También tiene el cuerpo del mismo. Esto se consigue mediante los links, poniendo ?id\_recurso=x y luego en el archivo recursoContenido se obtienen los datos con esa id.



## Cómo detectar un ataque

2024-03-03 21:55:21



Si vemos que la persona empieza a sudar y tiritar, podríamos estar ante un ataque de pánico

Vamos con Grupos de Apoyo, sección donde se realizan reuniones de la comunidad (previamente habladas con los administradores de la web) y los usuarios se pueden unir a ellas.

El diseño es simple, la imagen del usuario que la organiza, la fecha de la reunión y los participantes.



Buscar un grupo

**Organizado por:** Gabriel  
Tema: Ansiedad general  
Fecha: 2024-03-23 00:00:00  
Participantes: 0

**Organizado por:** Ismael  
Tema: Ayuda en el uso de la web

Si se le da al botón de apuntarse ahora, se vuelve verde y se apunta el usuario a la reunión. Se le enviaría un correo con los datos detallados, pero no funciona eso así que no se pueden ver los detalles realmente.

La siguiente sección se trata del foro de la comunidad. Aquí, se pueden ver distintos artículos hechos por la comunidad, donde se pueden poner respuestas. Hay una barra de búsqueda para filtrar por título, y se puede añadir tu propio foro.

**TEMAS**

Título	Fecha	Usuario
¿Cómo podemos saber cuando tendremos un ataque de pánico?	2024-03-03 21:55:21	
¿Alguien sabe si la terapia con masajes ayuda al mutismo?	2024-03-03 21:55:21	
¿Merece la pena pagar BrainHub?	2024-03-03 21:55:21	
¿Cómo se usan los ejercicios de apoyo?	2024-03-03 21:55:21	
¿Por qué los gobiernos no dan ayudas?	2024-03-03 21:55:21	
Ayuda	2024-03-03 22:14:26	
<b>Agrega tu título</b>	También apunte tu artículo	

Si entramos al foro, veremos la imagen del usuario, el banner y la descripción del foro. Podemos añadir respuestas si queremos.

**¿Cómo podemos saber cuando tendremos un ataque de pánico?**

2024-03-03 21:55:21

Buenas, me llamo Ismael y me gustaría saber cuando podría darme un ataque de pánico. Desafortunadamente sufrí de Trastorno del Pánico y eso me provoca que en ocasiones me quede parado en un lugar público.

**Repuesta**

**Santiago Daza** Yo encontré un patrón cuando me daban ataques, y aprendí a controlarlo con ejercicios de relajación.

Casi acabando, la página de terapéutas. En esta página se verán todos los terapéutas registrados por los administradores de la web. Se puede ver el precio junto a las diferentes características del terapéuta. A la derecha está el calendario mediante el cual se le puede pedir cita.

**María Paveda Martínez**  
Nº Identificación: 247334  
Especialización: Fobia Social  
Idiomas: Español, English  
Nacionalidad: Española  
Cita - 50 minutos  
50.00€

Hoy	Mañana	Pasado
14:00	14:00	14:00
15:00	15:00	15:00
16:00		
18:00	18:00	18:00
19:00	19:00	19:00
20:00	20:00	20:00

**Nikolas Müller Weber**  
Nº Identificación: 99834  
Especialización: Mutismo Selectivo

Hoy	Mañana	Pasado

Arriba, se disponen varios filtros. Empezando por el menú de hamburguesa, se despliega un aside donde se puede filtrar por nacionalidad e idiomas.

Arriba a la derecha hay un select para seleccionar la especialidad del terapéuta. Y en el medio encontramos la barra de búsqueda para buscarlos directamente.

Cuando se hace click en una hora del calendario, salta una ventana de confirmación, un alert en el navegador. Si cancelamos, cancelamos el pedido de la cita, pero si confirmamos, se pide la cita y la hora se marca con una línea del calendario y no se puede volver a pedir cita.

Aquí se usan eventos de js para detectar cuando se hace click en la hora, y en vez de alert() se usa confirm() para verificar.

En los filtros de arriba y los de la izquierda, se usan también eventos de js para ver qué se selecciona, pero luego se hace un fetch a procesarTerapeuta para los filtros. Con bind params y el prepare statement, se aplican los distintos filtros.

Se pueden ver las citas en el perfil de usuario, donde se mostró anteriormente.

Por último, está la web de añadir opiniones, donde los usuarios pueden dejar su opinión de nuestra web. Estas opiniones aparecerán en el index.php de la web, con estrellas y la descripción que ponga el usuario.

En la web podemos ver el resto de opiniones también, así como la media de estrellas.

A esta página se puede acceder al final de las opiniones de todos los clientes:



**Brain Hub**

Recursos Apoyo Terapeutas Social

**Brain Hub**  
5 opiniones  
4.00 ⭐⭐⭐⭐  
EMPRESA VERIFICADA

**¡Comparte tu experiencia!**  
5 opiniones →

**Opiniones 4.00 ⭐⭐⭐⭐**  
5 opiniones en total

1 ⭐	2 ⭐	3 ⭐	4 ⭐	5 ⭐
██████████	██████████	██████████	██████████	██████████

**Ismael Moreno**  
**El mejor servicio**  
Me encantó el servicio de todos los terapeutas.  
⭐⭐⭐⭐

**Gabriel Rodríguez**  
**No sirve el premium**  
No hay mucho más contenido aparte del gratuito, por lo que se queda corto el premium. Aún así le doy 4 estrellas  
⭐⭐⭐

**Ibai Llanos**  
**Best Web**

**Wei Xu**  
**Je n'ai pas aimé les articles**

## **6. Conclusiones**

El proyecto ha sido más grande de lo esperado. A los integrantes del equipo nos hubiese gustado contar con más tiempo. Como conclusión, hemos aprendido bastante, sobre todo de sesiones, inyección sql y de fetch.

Siguiendo con el tema del tiempo, creemos que podría haber sido más fácil si cuando se realizan los Wireframes se empieza a trabajar en HTML, ya que se agiliza más la toma de decisiones y se empieza antes a maquetar.

De todas maneras, hemos disfrutado de todo el proceso de hacer el proyecto, del trimestre dedicado a Figma y a diseño, y de los problemas que se tenían que solucionar con php.

Evidentemente la página web es mejorable, y hay muchas características que no se han podido incluir o perfeccionar por falta de tiempo. Se ha intentado acabar y ofrecer un producto funcional aunque algo más pequeño.

## **7. Bibliografía**

- Manual de PHP, Explode:

<https://www.php.net/manual/en/control-structures.foreach.php>

- Manual de PHP, Foreach:

<https://www.php.net/manual/en/control-structures.foreach.php>

- Manual de PHP, Funciones de hashing de contraseñas:

<https://www.php.net/manual/es/ref.password.php>

- Stackoverflow, Varias consultas:

<https://stackoverflow.com>

- Chat GPT, Ayuda con el código:

<https://chat.openai.com>

- Figma, Diseño de Interfaz:

<https://www.figma.com>

- Notion, Aplicación para el reparto de tareas:

<https://www.notion.so/>

## **8. Anexos**

### **8. 1. Pruebas en PHP:**

#### **Objetivo de las Pruebas:**

El objetivo de estas pruebas era garantizar la funcionalidad y fiabilidad de la aplicación web BrainHub, abarcando desde la conexión con la base de datos hasta la interacción del usuario con la interfaz de la aplicación.

#### **Metodología de las Pruebas:**

Las pruebas se llevaron a cabo manualmente durante el desarrollo del proyecto. Para ello, se utilizaron instrucciones `echo` en PHP para validar la conexión y operaciones en la base de datos.

#### **1. Conexión a la Base de Datos (conecta.php):**

El objetivo de estas pruebas es asegurar que la aplicación maneja adecuadamente los casos de error al intentar conectarse a la base de datos, tanto cuando se especifica el nombre de la base de datos como cuando no se hace.

##### **• Prueba de Conexión con Base de Datos Especificada:**

Se intenta establecer una conexión con la base de datos brainhub utilizando credenciales incorrectas o un nombre de base de datos inexistente para verificar que el sistema maneja correctamente el error y muestra el mensaje esperado. Para ello, modificamos temporalmente los valores de \$username, \$password o \$dbname en la función getConexion() para utilizar credenciales inválidas o un nombre de base de datos incorrecto.

```
// Crear conexión
$conexion = mysqli_connect($servername, $username, $password, $dbname) or die("Error de conexión");
```

Al ejecutar la aplicación con estos parámetros alterados, esperamos que el proceso se detenga y se muestre el mensaje "Error de conexión" sin revelar detalles específicos del error a los usuarios finales.

Al realizar la prueba, se confirmó que el mensaje "Error de conexión" se mostraba correctamente, indicando que la aplicación gestiona adecuadamente los casos de error al intentar conectarse a la base de datos especificada.

- **Prueba de Conexión Sin Especificar Base de Datos:**

Similar a la prueba anterior, pero esta vez omitiendo el nombre de la base de datos para verificar el manejo de errores en la función getConexionsindb().

Modificamos los valores de \$username o \$password en la función getConexionsindb() para provocar un fallo en la conexión.

```
// Crear conexión
$conexion = mysqli_connect($servername, $username, $password) or die("Error de conexión");
```

Al ejecutar la aplicación con los parámetros alterados, se espera que se muestre el mismo mensaje de error "Error de conexión", asegurando que los usuarios no reciban información detallada sobre el fallo.

La prueba demostró que, al igual que en el primer caso, el mensaje de error se mostraba como se esperaba, confirmando que la gestión de errores funciona correctamente incluso cuando no se especifica una base de datos.

Estas pruebas manuales confirmaron que la aplicación maneja adecuadamente los errores de conexión con la base de datos, mostrando un mensaje genérico "Error de conexión" en cualquier situación de fallo. Este comportamiento previene la divulgación de información sensible y proporciona una indicación clara de que la conexión ha fallado.

## **2. Inserción de Datos (crear\_tablas.php):**

El objetivo de estas pruebas es validar que la inserción inicial de usuarios en la base de datos se realiza correctamente y solo cuando la tabla usuario está vacía, y asegurar que los errores durante el proceso de inserción se manejan adecuadamente.

Se realizaron pruebas manuales ejecutando el script crear\_tablas.php, el cual incluye una comprobación previa de registros en la tabla usuario y posteriormente inserta datos de prueba si la tabla está vacía.

### **Procedimiento y resultados:**

#### **Comprobación de la Existencia de Registros:**

Verificar que el script no intenta insertar datos si ya existen registros en la tabla usuario.

Antes de ejecutar el script, se añaden manualmente registros a la tabla usuario utilizando herramientas de gestión de bases de datos como phpMyAdmin.

```
$select = "SELECT * FROM usuario";
$result = $conexion->query($select);
if ($result->num_rows == 0) {
```

Resultado Esperado: El script no realiza ninguna inserción de datos debido a la presencia de registros existentes en la tabla.

#### **Inserción de datos:**

Comprobar que los datos de los usuarios se insertan correctamente cuando la tabla usuario está vacía.

Asegurándose de que la tabla usuario está vacía, se ejecuta el script crear\_tablas.php. Después de la ejecución, se utiliza una herramienta de gestión de bases de datos para verificar que los registros se han insertado.

```
mysqli_query($conexion, $insert1) or die("Error insert usuario");
```

Resultado Esperado: Se observa que los registros de usuario especificados en el script se han añadido correctamente a la tabla usuario sin duplicaciones, y no se muestra el mensaje "Error insert usuario", indicando que la inserción fue exitosa.

#### Manejo de Errores:

Confirmar que el script maneja adecuadamente los errores durante la inserción de datos.

Se modifica intencionadamente el script para incluir errores en la consulta de inserción (por ejemplo, especificando nombres de columnas incorrectos) y se ejecuta.

Resultado Esperado: Al intentar ejecutar la consulta modificada, el script detiene su ejecución y muestra el mensaje "Error insert usuario", lo que indica que el manejo de errores funciona como se espera.

#### Conclusiones de las Pruebas:

Las pruebas manuales demostraron que la inserción inicial de datos de usuario en la aplicación BrainHub se realiza de manera eficaz y segura, respetando la condición de que la tabla debe estar vacía para proceder. Además, el sistema de gestión de errores garantiza que cualquier fallo durante el proceso de inserción se informa claramente, evitando así la inserción parcial o incorrecta de datos.

### **3. Conclusiones Generales de las Pruebas de PHP**

Es importante destacar que los casos de prueba presentados aquí para la conexión a la base de datos y la inserción de datos, aunque detallados específicamente, son solo una muestra representativa de un conjunto más amplio de pruebas realizadas a lo largo del desarrollo de BrainHub. Este enfoque de prueba, que incluye tanto la validación del funcionamiento esperado como el manejo de errores, se ha aplicado sistemáticamente en varias otras áreas del proyecto. Cada componente, funcionalidad y módulo ha sido sometido a pruebas similares para asegurar la integridad, seguridad y usabilidad de nuestra aplicación. La metodología y los principios subyacentes detrás de estos

ejemplos se replicaron en todo el proyecto, reflejando nuestro compromiso con la calidad y la fiabilidad del sistema en su conjunto.