UNIVERSIDAD DE SANTIAGO DE COMPOSTELA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA

Emozio

Autor:
Raquel Gil Martínez

Director:
José Manuel Cotos Yáñez

Grado en Ingeniería Informática

Febrero 2018

Trabajo de Fin de Grado presentado en la Escuela Técnica Superior de Ingeniería de la Universidad de Santiago de Compostela para la obtención del Grao en Ingeniería Informática

D. José Manuel Cotos Yáñez, Profesor del Departamento de Electrónica y Computación de la Universidad de Santiago de Compostela,

INFORMA:

Que la presente memoria, titulada *Emozio*, presentada por **D.ª Raquel Gil Martínez** para superar los créditos correspondientes al Trabajo de Fin de Grado de la titulación de Grado en Ingeniería Informática, se realizó bajo nuestra dirección en el Departamento de Electrónica y Computación de la Universidad de Santiago de Compostela.

Y para que así conste a los efectos oportunos, expide el presente informe en Santiago de Compostela, a 5 de febrero de 2018:

El director, El alumno,

José Manuel Cotos Yáñez Raquel Gil Martínez

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introdución

El curso pasado (2016-2017) participé en el programa para emprendedores Yuzz "Jóvenes con ideas", donde aprendí a convertir una idea empresarial en un modelo de negocio.

Mi equipo promotor estaba compuesto por Bruno Rodríguez, psicólogo, y yo misma. Nuestra vocación hacia la aportación de soluciones en el campo de la salud mental, se vio impulsada por la adecuada combinación entre nuestras experiencias y estudios, dando lugar a nuestro proyecto empresarial Emozio.

1.1. Contexto

Emozio será una plataforma web de búsqueda de psicólogos, donde el paciente cubre un test validado científicamente que permitirá a la plataforma asignar al profesional más adecuado para tratar su problema. Se trata pues de una plataforma de servicios de emparejamiento entre un paciente y el mejor profesional disponible.

La finalidad es ofrecer el mejor especialista para solucionar el problema que tiene un determinado usuario, solventando la tediosa y difícil tarea de encontrar un psicólogo competente cuando surge la necesidad, y evitando terapias inefectivas.

Por otro lado, la mayoría de los psicólogos que ejercen su profesión tienen la necesidad de encontrar un flujo constante de pacientes para mantener a flote su consulta/despacho, por lo que pertenecer a nuestro catálogo supondrá un plús añadido a sus servicios ya que éstos serán respaldados por un sello de calidad (Emozio) que hará posible el aumento y fidelización de clientes/pacientes. Así como, aportarles visualización y publicidad con costes más reducidos que los existentes.

1.2. Objetivos

El objetivo del proyecto es desarrollar una plataforma web que permita la búsqueda de psicólogos, donde, por un lado, los pacientes cubren un test que permitirá caracterizarlos, de forma que la plataforma pueda asignarles el profesional más adecuado para tratar su problema. Por otro lado, la plataforma ha de permitir a los psicólogos darse de alta como profesionales caracterizados por su *expertise*. Dicha plataforma, ha de estar provista de diferentes servicios comunes como la gestión los usuarios (pacientes y psicólogos), posibilitar el filtrado de los psicólogos que aparecen como resultado del test en función de diferentes parámetros y poner en contacto el paciente con el psicólogo en cuestión...

Se pretende que la plataforma desarrollada en el marco de trabajo del TFG, sirva como producto mínimo viable (PMV) del modelo de negocio, por lo que se persigue poder mostrar el producto como un prototipo a los usuarios finales.

En concreto, se persigue conseguir los siguientes objetivos transversales:

- Desarrollar una plataforma web.
- Permitir que los pacientes cubran un test que los pueda caracterizar.
- Emparejar a los pacientes con los psicólogos que puedan tratarles.
- Realizar la gestión de usuarios.
- Permitir el filtrado en los resultados del test.
- Permitir la comunicación entre pacientes y psicólogos.

1.3. Estructura de la memoria

Capítulo 2

Gestión del proyecto

En este capítulo se detallan las metodologías de la gestión de proyectos que se han seguido en este trabajo presentes en el PMBOK[**?**].

Las siguientes secciones describen el alcance del proyecto, la metodología de desarrollo, la planificación temporal, la gestión de la configuración, la gestión de costes y la gestión de riesgos.

2.1. Alcance del proyecto

Según el PMBOK[**?**] se trata del trabajo realizado para entregar un producto, servicio o resultado con las funciones y características especificadas. La línea base del proyecto, consistirá en el enunciado del alcance, la estructura de desglose de trabajo (EDT/WBS) y el diccionario EDT/WBS asociado.

2.1.1. Enunciado del alcance

Como promotora de la *startup* Emozio, necesito una plataforma web que de viabilidad a mi modelo de negocio, el cual consiste en la asignación del psicólogo más adecuado para tratar la patología de un paciente.

Como somos una *startup* que ha surgido de forma reciente, todavía preciso validar y afianzar mi propuesta de valor para lanzarla al mercado. Por tanto, el requisito primordial es disponer de un producto mínimo viable que me permita mostrar esta propuesta a mi público objetivo.

Para desempeñar el producto mínimo viable necesito desarrollar una plataforma *web* en la que pueda identificar una o varias posibles patologías en un determinado paciente. Una vez conocida la patología y el porcentaje de la misma en la caracterización del paciente, la plataforma le emparejará a un listado de psicólogos que podrían tratar una o varias de sus patologías. Además, estos psicólogos podrán ser filtrados por distintos campos, como el precio, el seguro de salud, si se trata de una cita presencial u *online*, entre otros.

A su vez, debe permitir al paciente ponerse en contacto con el psicólogo del listado por el que desea ser atendido. Este mensaje de contacto, podrá visualizarlo el psicólogo a través de su bandeja de entrada.

Cabe decir, que tanto psicólogos como pacientes podrán darse de alta en la plataforma y hacer uso de sus servicios, por lo que se necesitará una mínima gestión de usuarios.

Para que el paciente pueda evaluar la cita recibida por el psicólogo, éste podrá dejar un comentario con una valoración en su perfil para que pueda ser compartido al resto de usuarios. De esta forma, se dará la transparencia necesaria al servicio de recomendación lo que nos valdrá de sello de calidad.

2.1.2. Criterios de aceptación

Las condiciones que deben cumplirse para que el proyecto sea aceptado son:

- Entrega en plazo de todos los entregables.
- Realización de los requisitos esperados que tengan mayor prioridad.

2.1.3. Entregables del proyecto

Se deberá hacer entrega de:

- Tres copias en papel de la memoria, junto con el manual de instalación y los manuales de usuario
- Una copia en soporte digital de:
  - La memoria, junto con el manual de instalación y los manuales de usuario
  - El código fuente
  - El ejecutable

2.1.4. Exclusiones del proyecto

Las exclusiones aplicadas al proyecto son:

- El mantenimiento y soporte de la plataforma una vez realizada la entrega.
- Análisis de campo e investigación asociada al desarrollo del cuestionario científico de asignación.

2.1.5. Restricciones del proyecto

El alcance de este proyecto sólo abarcará el producto mínimo viable de lo que será la plataforma real de Emozio, y que servirá como punto de partida para la implementación de la misma. Su pretensión no se trata de realizar el desarrollo completo de la aplicación con todas sus características totalmente funcionales, sino realizar una primera aproximación a la misma.

2.1.6. Supuestos del proyecto

Asumimos que al tratarse de un producto mínimo viable, no se podrá estimar el crecimiento del sistema tanto a nivel de funcionalidades como en el contenido de la base de datos, por lo que no se podrá preveer cómo será la escalabilidad real del sistema.

2.1.7. Estructura de descomposición de tareas (EDT/WBS) del proyecto

La estructura de descomposición de tareas viene representada en la figura 2.1.

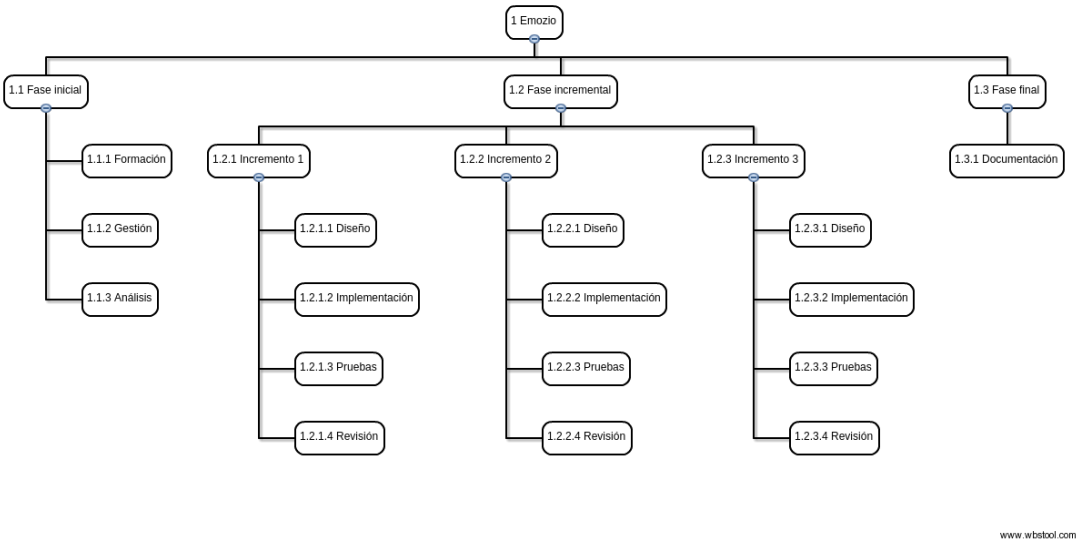


Figura 2.1: Estructura de descomposición de tareas del proyecto

Diccionario EDT/WBS

El proyecto está dividido en tres fases: Inicial, incremental y final. Éstas se encuentran descritas en mayor detalle en la sección donde se detalla la metodología de desarrollo utilizada.

Fase inicial Compuesta por la formación en las principales tecnologías utilizadas (*Stack MEAN framework*), la elaboración de la gestión del proyecto y un primer análisis inicial del producto.

Fase incremental Se encuentra dividida entre los tres incrementos que se van a realizar, cada uno con sus respectivas tareas: Diseño, implementación, pruebas y revisión.

Fase final En ella se elaborará toda la documentación final del proyecto que se va a presentar: La memoria, el manual de instalación y el manual de usuario.

2.2. Metodología de desarrollo

Una metodología de desarrollo de software en ingeniería de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información.

Según Pressman[**?**], cualquier proceso del software ágil se caracteriza por la forma en la que aborda cierto número de suposiciones clave acerca de la mayoría de proyectos de software:

- Es difícil predecir que requerimientos de software persistirán y cuales cambiarán o pronosticar como cambiaron las prioridades del cliente a medida que avanza el proyecto.
- En muchos tipos de software el diseño y la construcción de ejecutarse en forma simultánea de modo que los modelos de diseño se prueban a medida que se crean.
- El análisis, el diseño, la construcción y las pruebas no son tan predecibles como nos gustaría.

Las principales motivaciones para utilizar este tipo de metodología se trataban de:

- Inicialmente, se desconocía mucha información: Como el algoritmo que se iba a utilizar para el emparejamiento, como la caracterización de los pacientes y psicólogos, como la información que era necesario almacenar en la base datos...
- Existía la posibilidad de que el psicólogo experto, mi principal *stakeholder* el cuál me iba a proporcionar el *feedback* e información necesaria del sistema, me pidiese cambios repentinos al descubrir nueva información relevante a cerca del emparejamiento.

- Por otra parte, sí se tenían claro los subsistemas que se requerían dentro de la plataforma.
- Como se trata de un proyecto que surge de un estado del arte, necesitábamos gran margen para introducir cambios y poder integrarlos.
- La inexperiencia en las tecnologías utilizadas podrían suponer que hubiese continuos cambios en el diseño y el código de la plataforma.

Por tanto, se requería un proceso adaptativo e incremental, donde en cada incremento se desarrollase uno de los subsistemas identificados en nuestra plataforma.

En este tipo de procesos, deben entregarse incrementos de software (prototipos ejecutables o porciones de un sistema) en periodos cortos de tiempo. Este enfoque iterativo permite que el cliente evalúe de forma regular el incremento de software, dé la retroalimentación necesaria e influya en las adaptaciones del proceso que se realicen para aprovechar esa retroalimentación.

Las fases que se van a realizar en este proyecto son:

1. **Fase Inicial:** Conlleva la gestión de proyecto, la formación y un primer análisis inicial de lo que sería la plataforma.
2. **Fase incremental:** En esta fase se realizarán los tres incrementos.
3. **Fase final:** Se realizará la documentación del proyecto.

Fase incremental

En esta sección se describirán con mayor detalle en qué consisten los tres incrementos del proceso. Donde cada uno de ellos tendrá sus propias fases de diseño, implementación, pruebas y revisión.

Incremento 1

El primer incremento se realizará justo después de la fase inicial, y se corresponde con el subsistema del cuestionario de emparejamiento paciente-psicólogo.

Según la planificación temporal descrita en la gestión de tiempos, se prevé que el incremento 1 comience el día 9 de noviembre y finalice el 6 de diciembre. Esta previsión tiene en cuenta que es el primer contacto directo con las tecnologías utilizadas por lo que es predecible que se trabaje de forma más pautada y se produzcan muchos ensayo-error. Además de que en él se implementa la complejidad algorítmica relacionada con el emparejamiento.

Incremento 2

Tras finalizar el primer incremento, se procede a realizar este segundo que se corresponde con el subsistema de gestión de usuarios.

Según la planificación prevista, estará comprendido entre el 7 de diciembre y el 20 de diciembre.

Incremento 3

El último incremento es el subsistema de comunicación.

Siguiendo la planificación temporal realizada, durará desde el 21 de diciembre hasta el 3 de enero.

A lo largo de esta memoria cada tarea realizada, estará definida dentro del contexto del incremento en el cual se realiza.

2.3. Planificación temporal

La previsión temporal del presente proyecto estaba comprendido entre el 16 de octubre de 2017 y el 21 de enero de 2018, con un trabajo de seis horas diarias. La estimación temporal en tareas, y la asignación de las mismas a los recursos, se puede observar en el diagrama de Gantt 2.2.

Fase inicial	**16d**
Formación	10d
Gestión	3d
Análisis	3d
Fin "Fase Inicial"	
Fase incremental	**40d**
Incremento 1: Cuestionario	**20d**
Diseño	2d
Implementación	13d
Pruebas	3d
Revisión	2d
Fin "Incremento 1: Cuestionario"	
Incremento 2: Gestión usuarios	**10d**
Diseño	2d
Implementación	5d
Pruebas	2d
Revisión	1d
Fin "Incremento 2: Gestión usuarios"	
Incremento 3: Comunicación	**10d**
Diseño	2d
Implementación	5d
Pruebas	2d
Revisión	1d
Fin "Incremento 3: Comunicación"	
Fin "Fase incremental"	
Fase final	**10d**
Documentación	10d
Fin "Fase final"	

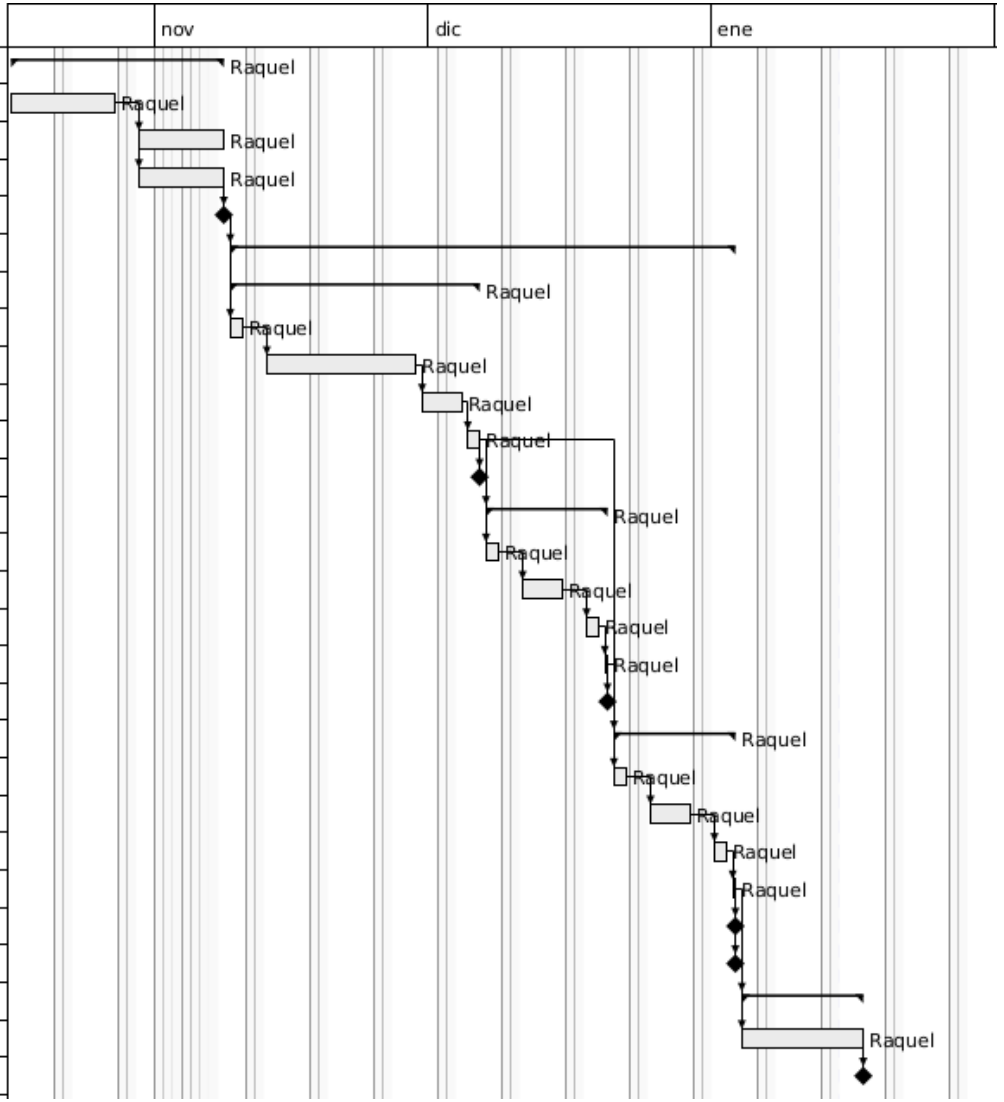


Figura 2.2: Diagrama de Gantt

2.4. Gestión de la configuración

La gestión de la configuración del software es un conjunto de actividades diseñadas para administrar el cambio mediante la identificación de los productos de trabajo con potencial de cambio, las relaciones entre ellos, la definición de mecanismos para administrar diferentes versiones de los mismos y el control de los cambios impuestos, así como la auditoría y reporte de los cambios realizados[**?**].

2.4.1. Línea base

El IEEE (IEEE Std. No. 610.12-1990) indica que una especificación o producto que se revisó formalmente y con el que se estuvo de acuerdo, servirá como base para un mayor desarrollo y que cambia sólo a través de procedimientos de control de cambio formal.

Las líneas base de nuestro proyecto son fundamentalmente el enunciado del alcance del proyecto, los requisitos *software*, el diseño y el código del anterior incremento (si existe).

Creación de líneas base

Como nuestro proyecto se desarrolla de manera incremental, la línea base del segundo incremento, será tanto el enunciado del alcance del proyecto, los requisitos software asociados a dicho incremento y finalmente, el código resultado del incremento 1. De forma análoga sería para el caso del incremento 3, que tendría como línea base adicional el código resultado del incremento 2.

2.4.2. Repositorio para la gestión de la configuración

Un repositorio es el conjunto de mecanismos y estructuras de datos que permiten administrar el cambio de forma efectiva, asegurando la integridad, posibilidad de compartir e integrar datos. Para lograr estas capacidades, el repositorio se define como un metamodelo que determina cómo se almacena la información en el repositorio, cómo pueden acceder las herramientas a los datos, cuán bien puede mantenerse la seguridad e integridad de los datos y cuán fácilmente puede extenderse el modelo existente para alojar nuevas necesidades[**?**].

Repositorio escogido

Este proyecto se encuentra almacenado en un repositorio GitHub que es una plataforma de desarrollo colaborativo de software para alojar proyectos usando el sistema de control de versiones Git[17]. Git nos permitirá tener una copia del repositorio del proyecto en local y otra en remoto. El proyecto en local sufrirá constantes modificaciones, que una vez validadas, se guardarán en el remoto.

Metamodelo

La estructura de información que se encuentra en el repositorio viene definida en la tabla 2.1.

Carpeta		Descripción del contenido
em_diagramas		Contiene el archivo de StarUML que contiene todos los diagramas del proyecto: Casos de uso, modelo de datos, patrón MVC...
em_memoria	em_analisis	Contiene todos los archivos referentes al análisis.
	em_anexos	Contiene todos los anexos de la memoria del proyecto.
	em_diseño	Contiene todos los archivos referentes al diseño.
	em_gest_proy	Contiene todos los archivos referentes a la gestión del proyecto.
	em_introducción	Contiene la introducción de la memoria del proyecto.
	em_memoria_final	Se trata del documento en LaTeX que contiene la memoria a entregar.
	em_plan_pruebas	Contiene todos los archivos referentes al plan de pruebas.
em_mockup		Contiene el mockup hecho con Pencil de la plataforma web.
em_web		Contiene todos los archivos que constituyen el código fuente de la plataforma web.

Tabla 2.1: Metamodelo

Dentro de cada carpeta o subcarpeta, los archivos aparecen con un nombre descriptivo. Por ejemplo, en la subcarpeta de `em_memoria` llamada `em_gest_proy` se encuentra `gest_costes.odt` que es el archivo correspondiente a la subsección de costes de la sección de gestión del proyecto que habrá en la memoria final.

2.4.3. Sistema de gestión de la configuración

La estructura del repositorio está distribuida del siguiente modo:

1. **Master** Contendrá la última versión validada del código fuente, es decir, tras pasar las pruebas del incremento 1, contendrá el código fuente del incremento 1, y así, sucesivamente.

2. **Branches**

   - **memoria_branch** Contendrá los commits de las distintas versiones de la memoria del proyecto.
   - **diagrama_branch** Contendrá los commits de las distintas versiones de los diagramas de la plataforma.
   - **mockup_branch** Contendrá los commits de las distintas versiones del mockup de la plataforma.
   - **cuestionario_branch, usuarios_branch y contacto_branch** Contendrán respectivamente, los *commits* del código fuente asociado a los incrementos 1, 2 y 3. Antes de comenzar un incremento, se crea una *branch* de *master* y se implementan las funcionalidades pertenecientes a ese incremento. Una vez realizada su fase de pruebas, se hará un *merge* de esa *branch* con el *master*, y posteriormente, se elimina.

2.5. Gestión de costes

2.6. Gestión de riesgos

2.6.1. Plan de gestión de riesgos

El riesgo de un proyecto es un evento o condición incierta que, de producirse, tiene un efecto positivo o negativo en uno o más de los objetivos del proyecto. Éste, puede tener una o más causas, y de materializarse, uno o más impactos[**?**].

Para poder prevenir estas consecuencias, se debe llevar un registro de los posibles riesgos que pueden acontecer. Así, una vez identificados y analizados, podemos planificar las respuestas para los mismos.

Para poder definir la probabilidad e impacto de la materialización de un riesgo, se deben definir las escalas correspondientes a los mismos. Para poder medir la probabilidad e impacto que puede tener un riesgo se han creado las tablas 2.3 y 2.2.

La **probabilidad** representa la expectativa de ocurrencia real del riesgo, y el **impacto**, representa el efecto que la ocurrencia del riesgo tendría en el desarrollo del proyecto, en términos de coste, esfuerzo o duración total del mismo.

Si vinculamos la probabilidad de ocurrencia de cada riesgo con su impacto sobre los objetivos del proyecto en caso de que ocurra dicho riesgo, podemos obtener la matriz que representa el nivel de exposición al riesgo (dado por el producto del impacto por la probabilidad). Esta matriz 2.4 determinará la gestión de los riesgos del proyecto.

Valoración del impacto	
Repercusión en Plazo / Esfuerzo / Costes	**Impacto**
>= 20 %	Alto
Entre 10 % y 20 %	Medio
<= 10 %	Bajo

Tabla 2.2: Valoración del impacto

Valoración de la probabilidad	
Ocurrencia del Riesgo	**Probabilidad**
>= 80 % (casi segura)	Alta
Entre 30 % y 80 % (muy probable)	Media
<= 30 % (poco probable)	Baja

Tabla 2.3: Valoración de la probabilidad

Nivel de exposición al riesgo				
		Probabilidad		
		Alta	Media	Baja
Impacto	Alto	Alto	Alto	Medio
	Medio	Alto	Medio	Bajo
	Bajo	Medio	Bajo	Bajo

Tabla 2.4: Nivel de exposición al riesgo

2.6.2. Identificación de riesgos

Tras identificar los riesgos por medio de revisión de la documentación, tormenta de ideas y análisis de supuestos, se procede a la elaboración del siguiente registro de riesgos 2.5.

2.6.3. Análisis cualitativo de riesgos

Mediante el análisis cualitativo de los riesgos podemos evaluar la prioridad de los riesgos identificados a través de la probabilidad relativa de ocurrencia, del impacto correspondiente sobre los objetivos del proyecto si los riesgos llegasen a presentarse. Esta evaluación 2.6 refleja la actitud que existe frente a los riesgos.

La matriz de exposición de los riesgos 2.7 da a conocer las prioridades de acción en el caso de que se materializasen los riesgos.

2.6.4. Plan de respuesta a los riesgos

Con la planificación de respuesta a los riesgos se busca desarrollar opciones y acciones para mejorar las oportunidades y reducir las amenazas a los objetivos del proyecto. Las respuestas a los riesgos deben adecuarse a la importancia del riesgo, ser rentables con relación al desafía a cumplir, realistas dentro del contexto del proyecto y a cargo de una persona responsable [**?**].

Debido a que sólo existe una persona a cargo del proyecto, ésta será la responsable de acatar las medidas oportunas.

Identificador	Nombre	Descripción
RSG-001	Práctica deficiente de la gestión de proyectos	Se produce una práctica deficiente en la gestión de proyectos debido a la inexperiencia.
RSG-002	Dependencia de participantes externos	Existe dependencia de participantes externos fuera del ámbito de control directo del proyecto como el tutor que guía el proyecto o el experto psicólogo.
RSG-003	Falta de experiencia en las tecnologías utilizadas	La inexperiencia en las tecnologías utilizadas puede repercutir en la planificación prevista del proyecto.
RSG-004	Pérdida de información	Se puede producir la pérdida de información debido a un mal guardado de los datos en el respositorio.
RSG-005	Retraso en la planificación temporal	Acontece un retraso en la planificación temporal prevista debido a causas externas como enfermedad o asuntos personales.
RSG-006	Caída de la red proveedora de Internet	Una caída de la red proveedora de Internet puede causar retrasos en la planificación o pérdida deinformación.
RSG-007	Fallo de suministro eléctrico	Un fallo del suministro eléctrico puede causar retrasos en la planificación o pérdida deinformación.
RSG-008	Ataques maliciosos en el equipo de trabajo	Un ataque malicioso en el equipo de trabajo puede significar que existe una brecha de seguridad en el sistema por lo que existe un posible robado de datos o incluso pérdidas de información.

Tabla 2.5: Registro de riesgos

Identificador	Nombre	Probabilidad	Impacto	Exposición
RSG-001	Práctica deficiente de la gestión de proyectos	Media	Alto	Alto
RSG-002	Dependencia de participantes externos	Alta	Medio	Alto
RSG-003	Falta de experiencia en las tecnologías utilizadas	Alta	Alto	Alto
RSG-004	Pérdida de información	Baja	Alto	Medio
RSG-005	Retraso en la planificación temporal	Baja	Medio	Bajo
RSG-006	Caída de la red proveedora de Internet	Baja	Alto	Medio
RSG-007	Fallo de suministro eléctrico	Baja	Alto	Medio
RSG-008	Ataques maliciosos en el equipo de trabajo	Baja	Alto	Medio

Tabla 2.6: Evaluación de los riesgos

		Probabilidad		
		Alta	Media	Baja
Impacto	Alto	RSG-003	RSG-001	RSG-004 RSG-006 RSG-007 RSG-008
	Medio	RSG-002		RSG-005
	Bajo			

Tabla 2.7: Matriz de exposición de los riesgos

RSG-001	**Práctica deficiente de la gestión de proyectos**
Acción de prevención	Mitigar. Se asegurará que la persona encargada de la gestión de proyectos se forme específicamente en dicho ámbito.
Indicador	La persona encargada de la gestión no efectúa adecuadamente los procesos a seguir.
Acción de corrección	Se dedicarán más horas de trabajo a la gestión de proyectos.

Tabla 2.8: Plan de respuesta RSG-001

RSG-002	**Dependencia de participantes externos**
Estrategia	Minimizar. Debido a la imprevisión que supone estar a expensas de otro *stakeholder* del proyecto se plantea mover las reuniones agendadas.
Indicador	Ausencia o falta de respuesta por parte de un *stakeholder*.
Acción de corrección	Se procederá a abordar las distintas tareas de forma ficticia o en último recurso se obviarán.

Tabla 2.9: Plan de respuesta RSG-002

RSG-003	**Falta de experiencia en las tecnologías utilizadas**
Estrategia	Mitigar. Se formará al trabajador en las tecnologías utilizadas antes de comenzar con el desarrollo del proyecto.
Indicador	El trabajador desconoce cómo implementar determinada funcionalidad
Acción de corrección	Se pausará por un determinado tiempo el desarrollo y se estudiará lo necesario para poder continuar.

Tabla 2.10: Plan de respuesta RSG-003

RSG-004	**Pérdida de información**
Estrategia	Acción preventiva. Se harán volcados del trabajo realizado en el repositorio de forma periódica de al menos dos días de trabajo.
Indicador	Los últimos cambios efectuados no están reflejados en los archivos de trabajo ni en la última versión del repositorio.
Acción de corrección	Se repartirá el número de horas de trabajo correspondiente a esa parte durante el próximo periodo de la planificación.

Tabla 2.11: Plan de respuesta RSG-004

RSG-005	**Retraso en la planificación temporal**
Estrategia	Minimizar. Debido a que se puede producir un retraso en la planificación temporal por causas ajenas al proyecto, se recuperarán las horas perdidas reubicándolas en la planificación temporal.
Indicador	Surge una enfermedad o un asunto personal.
Acción de corrección	Se reubicará la carga de trabajo prevista y que no haya sido realizada a lo largo del periodo que vaya a continuación.

Tabla 2.12: Plan de respuesta RSG-005

RSG-006	**Caída de la red proveedora de Internet**
Estrategia	Mitigar. Se dispondrá de una red alternativa a la que usamos de forma habitual.
Indicador	Falla la conexión a Internet o no se encuentra la red.
Acción de corrección	Se procederá a buscar una red alternativa. Por ejemplo, en el caso de estar trabajando con la conexión WiFi y que ésta sufra una caída, proceder a conectarnos a la red móvil del *smartphone* personal.

Tabla 2.13: Plan de respuesta RSG-006

RSG-007	**Fallo de suministro eléctrico**
Estrategia	Minimizar. Se procede a hacer copias de seguridad de forma periódica. Si el fallo de suministro eléctrico causa una pérdida del trabajo realizado, se retoma el trabajo a partir de la última copia de seguridad existente.
Indicador	El ordenador se queda sin suministro eléctrico de forma repentina tras una subida de tensión.
Acción de corrección	Se rehará el trabajo a partir de la última versión guardada.

Tabla 2.14: Plan de respuesta RSG-007

RSG-008	Ataques maliciosos en el equipo de trabajo
Estrategia	Evitar. No descargar recursos de fuentes no fiables, no utilizar el ordenador de trabajo como
Indicador	Fallos desconocidos hasta el momento en el sistema, pérdida de información, comportamiento
Acción de corrección	Se hará un formateo del sistema y se recuperarán los últimos cambios, de ser afectados, de la

Tabla 2.15: Plan de respuesta RSG-008

2.7. Análisis

La ingeniería de requisitos proporciona el mecanismo apropiado para entender lo que dice el cliente, analizar las necesidades, evaluar la factibilidad, negociar una solución, especificar la solución sin ambigüedades, validar la especificación y administrar los requisitos a medida que se transforman en un sistema funcional[**?**].

2.7.1. Casos de uso

Para poder entender cómo los usuarios emplearán finalmente las funciones y características del software, se debe crear un conjunto de escenarios que identifiquen la naturaleza de los usos para el sistema que se va a construir. La descripción de la manera en la que se utilizará el sistema se la conoce como caso de uso.

Diagrama de casos de uso

Las relaciones entre los actores y los casos de uso viene dada por el diagrama de casos de uso2.3.

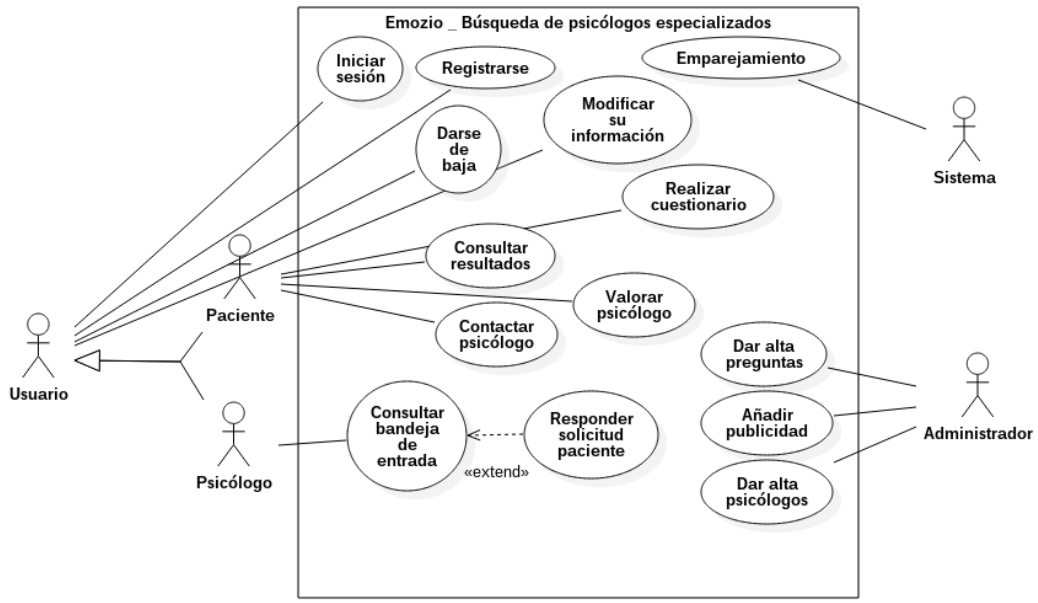


Figura 2.3: Diagrama de casos de uso

Actores del sistema

Un actor2.16 es cualquier cosa que se comunica con el sistema o producto y que sea externo a este.

Identificador	Nombre	Descripción
ACT-001	Usuario	Cualquier persona que acceda a nuestra plataforma web.
ACT-002	Paciente	Cualquier persona que desee ser evaluada por nuestra aplicación.
ACT-003	Psicólogo	Cualquier persona que ejerza profesionalmente como psicólogo.
ACT-004	Sistema	Encargado de ejecutar el algoritmo de emparejamiento.
ACT-005	Administrador	Persona encargada de gestionar los recursos de la aplicación.

Tabla 2.16: Actores del sistema

Especificación de casos de uso

Para poder determinar y evaluar correctamente cada caso de uso, tenemos una escala de la frecuencia de uso2.17 y otra de la prioridad2.18.

Frecuencia de uso	
Ocasionalmente	Se utilizará en algunas ocasiones, no de forma habitual o por costumbre.
Puntualmente	Se utilizará en raras ocasiones.
Limitada	Sólo podrá utilizarse en una única ocasión.

Tabla 2.17: Frecuencias de uso

Prioridad	
Alta	El caso de uso es imprescindible para el funcionamiento normal de la aplicación.
Media	El caso de uso aporta funcionalidad necesaria en el sistema pero no aporta un valor fundamental.
Baja	El caso de uso aporta funcionalidades extra pero es totalmente prescindible.

Tabla 2.18: Prioridades

CU-001	**Iniciar sesión**
Actor principal	Usuario
Objetivo en contexto	El actor desea acceder a una cuenta de usuario ya existente mediante sus claves de acceso.
Precondiciones	Las claves de usuario deben ser válidas.
Disparador	El actor desea acceder a la plataforma para disfrutar de sus servicios.
Escenario	1. El actor accede a Emozio. 2. El actor introduce sus claves de acceso. 3. El actor pulsa el botón de acceso. 4. El sistema muestra el perfil del actor donde podrá acceder al resto de servicios.
Excepciones	1. Las claves de usuario son incorrectas. 2. El actor no estaba registrado en la plataforma en el momento que se realizó el acceso.
Prioridad	Alta. El actor debe iniciar sesión en la aplicación para poder utilizar cualquiera de sus servicios.
Disponibilidad	En el segundo incremento
Frecuencia de uso	Ocasionalmente
Canal del actor	A través de un navegador con acceso a Internet.
Dependencias (con los requisitos)	RI-001 RI-003 RF-001

Tabla 2.19: CU-001 Iniciar sesión

CU-002	**Registrarse**
Actor principal	Usuario
Objetivo en contexto	El actor desea registrarse en la plataforma cediendo sus datos personales para poder acceder a los servicios de la plataforma.
Precondiciones	Los datos introducidos en el formulario deben ser válidos.
Disparador	El actor desea registrarse en la plataforma para disfrutar de sus servicios.
Escenario	1. El actor accede a Emozio. 2. El actor introduce sus datos en un formulario de registro. 3. El actor pulsa el botón de registro. 4. Se completa el registro. 4.1. Si es paciente, el sistema muestra el perfil del usuario donde podrá acceder al resto de servicios. 4.2. Si es psicólogo, sus datos son enviados por correo para que el administrador pueda validarlos.
Excepciones	1. Los datos introducidos en el formulario son incorrectos. 2. Ya existe el usuario dentro de la plataforma.
Prioridad	Alta. El actor debe estar registrado en la aplicación para poder utilizar cualquiera de sus servicios.
Disponibilidad	En el segundo incremento
Frecuencia de uso	Limitada
Canal del actor	A través de un navegador con acceso a internet.
Dependencias (con los requisitos)	RI-001 RI-003 RF-001 RF-002

Tabla 2.20: CU-002 Registrarse

CU-003	**Realizar cuestionario**
Actor principal	Paciente
Objetivo en contexto	El paciente desea conocer qué profesional es el más adecuado para tratar su dolencia realizando el cuestionario de emparejamiento.
Precondiciones	El cuestionario debe estar cubierto.
Disparador	El paciente debe haber pulsado el botón de "hacer el cuestionario".
Escenario	1. El paciente accede a Emozio. 2. El paciente entra en su perfil de usuario ya sea por registro o acceso. 3. El sistema le muestra su perfil de usuario. 4. El paciente pulsa el botón de hacer el cuestionario. 5. El sistema le muestra el formulario que debe cubrir. 6. El paciente cubre las respuestas del formulario. 7. El paciente pulsa el botón de conocerlos resultados. 8. El sistema le mostrará su perfil en el que se encuentran los resultados.
Excepciones	Los datos introducidos en el formulario son incorrectos.
Prioridad	Alta. Es la característica central de la plataforma.
Disponibilidad	En el primer incremento
Frecuencia de uso	Puntualmente
Canal del actor	A través de un navegador con acceso a Internet.
Dependencias (con los requisitos)	RI-001 RI-002 RI-003 RF-001 RF-002 RF-005

Tabla 2.21: CU-003 Realizar cuestionario

CU-004	**Consultar resultados**
Actor principal	Paciente
Objetivo en contexto	El paciente desea consultar el resultado del cuestionario de asignación.
Precondiciones	El cuestionario debe estar cubierto.
Disparador	El paciente debe acceder a su perfil.
Escenario	1. El paciente accede a Emozio. 2. El paciente entra en su perfil de usuario ya sea por registro o acceso. 3. El sistema le muestra su perfil de usuario donde se encuentran los resultados del cuestionario de asignación.
Excepciones	El paciente no ha cubierto el test en ninguna ocasión.
Prioridad	Alta. Es la característica central de la plataforma.
Disponibilidad	En el primer incremento
Frecuencia de uso	Ocasionalmente
Canal del actor	A través de un navegador con acceso a Internet.
Dependencias (con los requisitos)	RI-001 RI-003 RF-001 RF-002 RF-005 RF-006

Tabla 2.22: CU-004 Consultar resultados

CU-005	**Contactar psicólogo**
Actor principal	Paciente
Objetivo en contexto	El paciente desea ponerse en contacto con uno de los psicólogos que le han sido asignados.
Precondiciones	1. El cuestionario debe estar cubierto. 2. Las respuestas del cuestionario no dieron resultados imprecisos.
Disparador	El paciente accede a su perfil y se decide a contactar con el psicólogo.
Escenario	1. El paciente accede a Emozio. 2. El paciente entra en su perfil de usuario ya sea por registro o acceso. 3. El sistema le muestra su perfil de usuario donde se encuentran los resultados del cuestionario de asignación. 4. El paciente pulsa el botón de contacto del psicólogo en cuestión. 5. El sistema le muestra un formulario de contacto que debe cubrir. 6. El paciente cubre el formulario. 7. El paciente pulsa el botón de enviar. 8. El sistema muestra un mensaje con el estado de la operación.
Excepciones	1. El paciente no ha cubierto el test en ninguna ocasión. 2. El test dió un resultado impreciso para las respuestas dadas en el cuestionario.

Tabla 2.23: CU-005 Contactar psicólogo - Parte 1

CU-005	**Contactar psicólogo**
Prioridad	Moderada. Puede implementarse después de las funciones básicas.
Disponibilidad	En el tercer incremento
Frecuencia de uso	Puntualmente
Canal del actor	A través de un navegador con acceso a Internet.
Dependencias (con los requisitos)	RI-001 RI-003 RF-001 RF-002 RF-005

Tabla 2.24: CU-005 Contactar psicólogo - Parte 2

CU-006	**Modificar información**
Actor principal	Usuario
Objetivo en contexto	El usuario desea cambiar sus datos porque en ese instante su información de usuario es incorrecta.
Precondiciones	El usuario tiene acceso a la plataforma.
Disparador	El usuario debe acceder a su página principal.
Escenario	1. El usuario accede a Emozio. 2. El usuario entra en su página principal ya sea por registro o acceso. 3. El sistema le muestra su página principal. 4. El usuario pulsa el botón de modificación de datos. 5. El sistema le muestra un formulario con sus datos actuales. 6. El usuario cubre el formulario con los datos correctos. 7. El usuario pulsa el botón de enviar. 8. El sistema muestra un mensaje con el estado de la operación.
Excepciones	El usuario cancela la operación en curso.
Prioridad	Moderada. Puede implementarse después de las funciones básicas.
Disponibilidad	En el segundo incremento
Frecuencia de uso	Puntualmente
Canal del actor	A través de un navegador con acceso a Internet.
Dependencias (con los requisitos)	RI-001 RI-003 RF-001 RF-002

Tabla 2.25: CU-006 Modificar información

CU-007	**Valorar psicólogo**
Actor principal	Paciente
Objetivo en contexto	El paciente desea valorar al psicólogo con el que se ha puesto en contacto.
Precondiciones	El paciente se ha puesto en contacto previamente con el psicólogo en cuestión.
Disparador	El paciente debe acceder al perfil del psicólogo.
Escenario	1. El paciente accede a Emozio. 2. El paciente entra en su perfil de usuario ya sea por registro o acceso. 3. El sistema le muestra su perfil de usuario donde se encuentran los resultados del cuestionario de asignación. 4. El paciente pulsa el botón de mostrar más información del psicólogo. 5. El sistema le muestra el perfil del psicólogo seleccionado. 6. El paciente pulsa el botón de realizar valoración. 7. El sistema muestra un breve formulario que le permite valorar y escribir un mensaje. 8. El paciente pulsa el botón de enviar. 9. El sistema muestra un mensaje con el estado de la operación.
Excepciones	1. El paciente no se había puesto en contacto con el psicólogo previamente. 2. El paciente ya había dejado una valoración en el perfil del psicólogo.

Tabla 2.26: CU-007 Valorar psicólogo - Parte 1

CU-007	**Valorar psicólogo**
Prioridad	Moderada. Puede implementarse después de las funciones básicas.
Disponibilidad	En el tercer incremento
Frecuencia de uso	Puntualmente
Canal del actor	A través de un navegador con acceso a Internet.
Dependencias (con los requisitos)	RI-001 RI-002 RF-001 RF-002 RF-005 RF-007

Tabla 2.27: CU-007 Valorar psicólogo - Parte 2

CU-008	**Filtrar resultados cuestionario**
Actor principal	Paciente
Objetivo en contexto	El paciente desea filtrar su lista de psicólogos resultado en función de unos parámetros.
Precondiciones	1. El cuestionario debe estar cubierto. 2. Las respuestas del cuestionario no dieron resultados imprecisos.
Disparador	El paciente accede a su perfil y desea filtrar los resultados del cuestionario.
Escenario	1. El paciente accede a Emozio. 2. El paciente entra en su perfil de usuario ya sea por registro o acceso. 3. El sistema le muestra su perfil de usuario donde se encuentran los resultados del cuestionario de asignación. 4. El paciente cubre el formulario de filtros que se muestra en la página. 5. El paciente pulsa el botón de enviar. 6. El sistema muestra el listado de resultados con los filtros que seleccionó el paciente.
Excepciones	1. El paciente no ha cubierto el test en ninguna ocasión. 2. El test dió un resultado impreciso para las respuestas dadas en el cuestionario.
Prioridad	Moderada. Puede implementarse después de las funciones básicas.
Disponibilidad	En el primer incremento
Frecuencia de uso	Ocasionalmente
Canal del actor	A través de un navegador con acceso a Internet.
Dependencias (con los requisitos)	RI-001 RI-003 RF-001 RF-002 RF-005

Tabla 2.28: CU-008 Filtrar resultados cuestionario

CU-009	**Consultar perfil psicólogo**
Actor principal	Paciente
Objetivo en contexto	El paciente desea consultar la información de perfil de uno de los psicólogos resultado.
Precondiciones	1. El cuestionario debe estar cubierto. 2. Las respuestas del cuestionario no dieron resultados imprecisos.
Disparador	El paciente debe acceder al perfil del psicólogo.
Escenario	1. El paciente accede a Emozio. 2. El paciente entra en su perfil de usuario ya sea por registro o acceso. 3. El sistema le muestra su perfil de usuario donde se encuentran los resultados del cuestionario de asignación. 4. El paciente pulsa el botón de mostrar más información del psicólogo. 5. El sistema le muestra el perfil del psicólogo seleccionado.
Excepciones	El paciente cancela la operación en curso.
Prioridad	Moderada. Puede implementarse después de las funciones básicas.
Disponibilidad	En el primer incremento
Frecuencia de uso	Ocasionalmente
Canal del actor	A través de un navegador con acceso a internet.
Dependencias (con los requisitos)	RI-001 RI-003 RF-001 RF-002 RF-005

Tabla 2.29: CU-009 Consultar perfil psicólogo

CU-010	**Darse de baja**
Actor principal	Usuario
Objetivo en contexto	El usuario desea darse debaja en la aplicación.
Precondiciones	El usuario tiene acceso a la plataforma.
Disparador	El usuario debe acceder a su página principal.
Escenario	1. El usuario accede a Emozio. 2. El usuario entra en su página principal ya sea por registro o acceso. 3. El sistema le muestra su página principal. 4. El usuario pulsa el botón de modificación de datos. 5. El sistema le muestra un formulario con sus datos actuales, y un botón para darse de baja. 6. El usuario pulsa el botón de darse de baja. 7. El sistema muestra un pop-up pidiendo confirmación para ejecutar la operación. 8. El sistema elimina al usuario de la base de datos. 9. El sistema muestra al usuario la página de inicio.
Excepciones	El usuario cancela la operación en curso.
Prioridad	Moderada. Puede implementarse después de las funciones básicas.
Disponibilidad	En el segundo incremento
Frecuencia de uso	Limitada
Canal del actor	A través de un navegador con acceso a Internet.
Dependencias (con los requisitos)	RI-001 RI-003 RF-001 RF-002

Tabla 2.30: CU-010 Darse de baja

CU-011	**Consultar bandeja de entrada**
Actor principal	Psicólogo
Objetivo en contexto	El psicólogo desea consultar las peticiones que le han llegado a su bandeja de entrada.
Precondiciones	Estar registrado.
Disparador	El psicólogo accede a su perfil y quiere consultar su bandeja de entrada.
Escenario	1. El psicólogo accede a Emozio. 2. El psicólogo accede a su perfil de usuario. 3. El sistema le muestra los mensajes pendientes de su bandeja de entrada.
Excepciones	-
Prioridad	Moderada. Puede implementarse después de las funciones básicas.
Disponibilidad	En el tercer incremento
Frecuencia de uso	Ocasionalmente
Canal del actor	A través de un navegador con acceso a Internet.
Dependencias (con los requisitos)	RI-001 RI-003 RF-001 RF-002

Tabla 2.31: CU-011 Consultar bandeja de entrada

CU-012	**Responder solicitud paciente**
Actor principal	Psicólogo
Objetivo en contexto	El psicólogo desea responder a una de las peticiones que le han llegado a su bandeja de entrada.
Precondiciones	Tener peticiones pendientes en la bandeja de entrada.
Disparador	El psicólogo accede a su perfil, consulta su bandeja de entrada y selecciona una de las peticiones.
Escenario	1. El psicólogo accede a Emozio. 2. El psicólogo accede a su perfil de usuario. 3. El sistema le muestra los mensajes pendientes de su bandeja de entrada. 4. El psicólogo selecciona una de las peticiones pendientes y la responde. 5. El sistema muestra un mensaje de éxito, y pasa la petición pendiente a respondida.
Excepciones	-
Prioridad	Moderada. Puede implementarse después de las funciones básicas.
Disponibilidad	En el tercer incremento
Frecuencia de uso	Ocasionalmente
Canal del actor	A través de un navegador con acceso a Internet.
Dependencias (con los requisitos)	RI-001 RI-003 RF-001 RF-002

Tabla 2.32: CU-012 Responder solicitud paciente

CU-013	**Emparejamiento**
Actor principal	Sistema
Objetivo en contexto	El sistema realiza el emparejamiento paciente-psicólogo.
Precondiciones	El paciente en cuestión se disponga a realizar el test.
Disparador	El paciente selecciona “Hacer el test”.
Escenario	Cuando se envíen los resultados del test de un paciente, el sistema hará la correspondiente asignación paciente-psicólogo.
Excepciones	-
Prioridad	Alta. Es la característica central de la plataforma.
Disponibilidad	En el primer incremento
Frecuencia de uso	Ocasionalmente
Canal del actor	A través de un navegador con acceso a Internet.
Dependencias (con los requisitos)	RI-001 RI-003 RF-001 RF-002 RF-005

Tabla 2.33: CU-013 Emparejamiento

CU-014	**Dar alta a las preguntas**
Actor principal	Administrador
Objetivo en contexto	El administrador dará de alta a las preguntas asociadas a las patologías.
Precondiciones	Tener nuevas preguntas para añadir al cuestionario.
Disparador	Aparece una nueva patología o se modifican y/o actualizan las preguntas de una patología.
Escenario	El administrador añade las preguntas a la base de datos.
Excepciones	-
Prioridad	Alta. Es la característica central de la plataforma.
Disponibilidad	En el primer incremento
Frecuencia de uso	Puntualmente
Canal del actor	A través de una herramienta de gestión de bases de datos.
Dependencias (con los requisitos)	RI-002

Tabla 2.34: CU-014 Dar alta a las preguntas

CU-015	**Añadir la publicidad**
Actor principal	Administrador
Objetivo en contexto	El administrador añadirá la publicidad necesaria a la aplicación en el momento que se precise.
Precondiciones	-
Disparador	El administrador quiere actualizar los contenidos publicitarios.
Escenario	El administrador añade las preguntas a la base de datos.
Excepciones	-
Prioridad	Moderada. Puede implementarse después de las funciones básicas.
Disponibilidad	En el tercer incremento
Frecuencia de uso	Puntualmente
Canal del actor	A través de una herramienta de edición de código.
Dependencias (con los requisitos)	-

Tabla 2.35: CU-015 Añadir la publicidad

CU-016	**Dar alta psicólogos**
Actor principal	Administrador
Objetivo en contexto	El administrador dará de alta a los psicólogos que hayan enviado el formulario de registro.
Precondiciones	El psicólogo que se va a dar de alta, ha de haber cubierto el formulario de registro.
Disparador	El administrador desea actualizar la colección de psicólogos de la base de datos.
Escenario	El administrador añade la información del psicólogo a la base de datos.
Excepciones	-
Prioridad	Moderada. Puede implementarse después de las funciones básicas.
Disponibilidad	En el segundo incremento
Frecuencia de uso	Puntualmente
Canal del actor	A través de un navegador con acceso a Internet y una herramienta de gestión de bases de datos.
Dependencias (con los requisitos)	RI-003 RI-002

Tabla 2.36: CU-016 Dar alta psicólogos

2.7.2. Modelado de datos

Si los requisitos del *software* incluyen la necesidad de crear, ampliar o hacer interfaz con una base de datos, o si deben construirse y manipularse estructuras de datos complejas, se debe crear un modelo de datos como parte del modelado general de los requisitos.[**?**]

Los modelos semánticos de datos son la definición de la forma lógica de los datos procesados por el sistema. [**?**]

¿Por qué MongoDB?

En nuestro proyecto, los datos se encuentran almacenados en una MongoDB que consiste en una base de datos NoSQL[1] de código abierto, orientada a docu-

[1]Las bases de datos NoSQL son aquellas que generalmente no son relacionales y no tienen un lenguaje de consulta como SQL.

mentos y a mantener datos desestructurados, sobretodo cuando esta crece exponencialmente. [**?**]

Comparativa MongoDB con SQL

Teniendo de referencia una base de datos SQL podemos comprender cómo MongoDB almacena la información con una pequeña analogía. En la gestión de sistemas de bases de datos relacionales, los datos se guardan en filas contenidas dentro de tablas. En cambio, en MongoDB los datos son guardados como documentos contenidos en colecciones.

Una colección es un conjunto de documentos, como éstos son independientes, pueden contener distintos campos, lo que indica que su esquema es dinámico. A su vez, estos documentos son muy similares a los *JSON Objects*, son conocidos como *BSON objects* y la principal diferencia es que contienen a mayores el tipo *ObjectID* que sirve de identificador del documento. Aquí es donde resaltamos una de las ventajas de utilizar MongoDB dentro de MEAN Stack, puesto que la información manejada en el lado cliente se obtiene y representa del mismo modo en el lado servidor.

En nuestro proyecto, las bases de datos son NoSQL, las cuales no tienen una representación estandarizada. Por tanto, y por entendimiento, se ha realizado a mayores una aproximación del modelo no relacional2.4 al de un modelo entidad relación2.5.

Modelo de datos NoSQL

En nuestro modelo de datos NoSQL la información está organizada en colecciones. Se debe resaltar que en algunas ocasiones, los documentos se encuentran embebidos dentro de otros. Como por ejemplo, en el caso de los comentarios de los psicólogos. Los comentarios son documentos embebidos (no tienen *ObjectID*) dentro de los documentos de la colección de psicólogos.

Por otra parte, están los documentos referenciados, que son independientes. Tomando el mismo ejemplo anterior, dentro de un documento embebido de comentarios hayamos una referencia a un documento de la colección de pacientes (_idPaciente).

La estructura se ha creado de este modo porque embebiendo documentos sólo se necesitaría una única consulta, el documento en cuestión se accedería con su documento padre y las escrituras serían atómicas (no se tendrían que escribir en múltiples documentos). Pero en el caso de que el documento sea de gran relevancia y obtenga gran cantidad de información, se hará referencia a él.

Modelo entidad-relación (MER)

El modelo entidad-relación-atributo2.5 muestra las entidades de datos sus atributos asociados y las relaciones entre estas entidades.

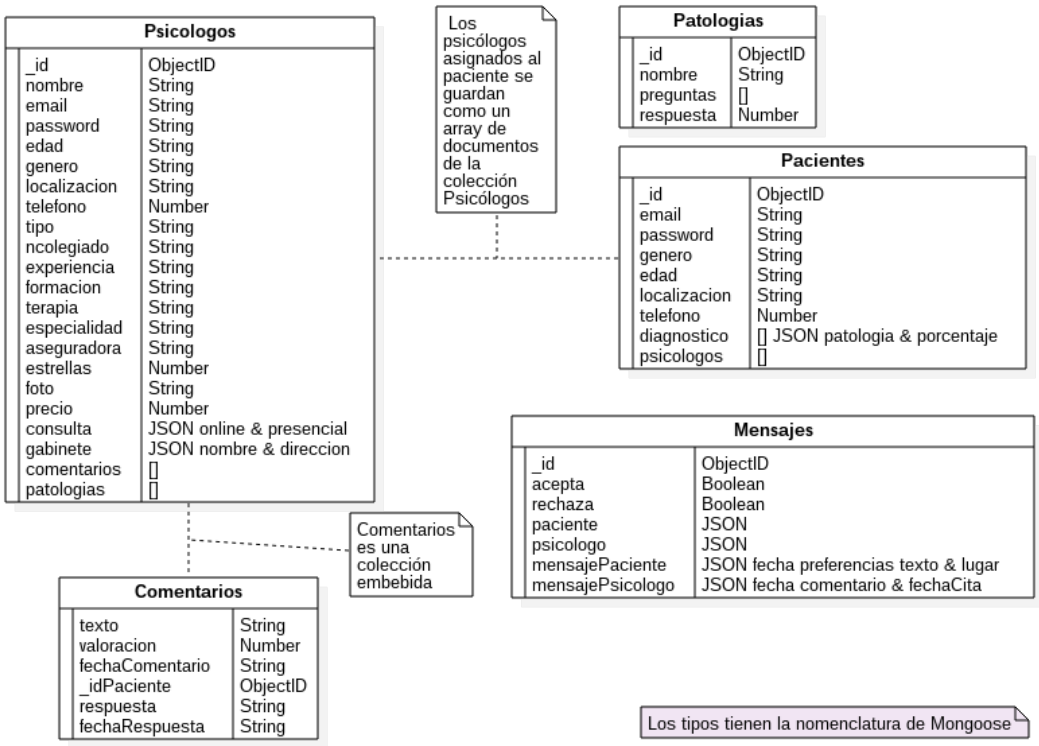


Figura 2.4: Modelo de datos MongoDB

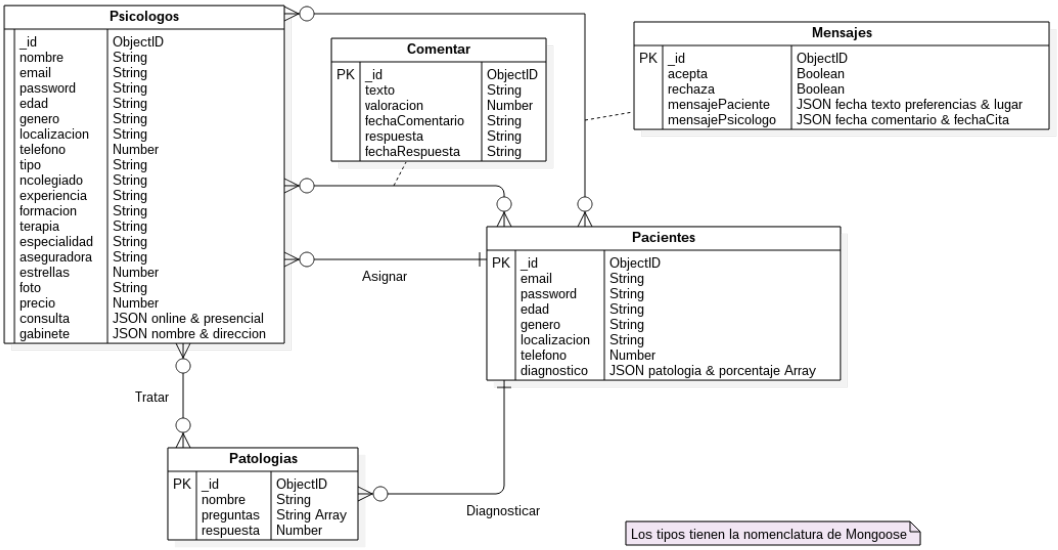


Figura 2.5: Modelo entidad-relación

Diccionario de datos

Para mantener descripciones más detalladas de las entidades incluidas en el modelo se utilizan diccionarios de datos para gestionar toda la información.

Por tanto, un diccionario de datos es una lista de nombres ordenada alfabéticamente incluido en los modelos del sistema. Este nos proporciona un mecanismo de gestión e inventariado de nombres y sirve como un almacén de información.

Colección	Descripción
Mensajes	Almacena toda la información referente a las comunicaciones entre psicólogos y pacientes.
Pacientes	Almacena toda la información sobre los pacientes del sistema.
Patologías	Almacena todas las patologías registradas hasta el momento en el sistema.
Psicólogos	Almacena toda la información sobre los psicólogos del sistema.

Tabla 2.37: Diccionario de datos: Colecciones

Colecciones2.37

BSON de los documentos2.382.392.402.42

2.7.3. Análisis de requisitos

Los requisitos para un sistema son la descripción de los servicios proporcionados por el sistema sus restricciones operativas y reflejan las necesidades de los clientes [**?**].

Para poder evaluar corréctamente los requisitos, se debe definir su escala de estabilidad2.43 [**?**].

Requisitos de información

Los requisitos de información guardan toda la información que debe almacenar el sistema para poder cumplir con sus objetivos. Identifican el concepto relevante sobre el que guardar información así como qué datos específicos del concepto son importantes.

Requisitos funcionales

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema. Los requisitos funcionales se encuentran divididos en función de los incrementos donde vayan a implementarse[**?**].

Documento	**Esquema BSON**		
	Campo	**Tipo de dato**	**Descripción**
Mensaje	_id	ObjectID	Identificador del documento
	acepta	Boolean	El psicólogo acepta la solicitud de un paciente.
	mensajePaciente - fecha - preferencias - texto	JSON - String - String - String	Es la información del mensaje que le envía un paciente al psicólogo. Se guarda la fecha donde fue enviado el mensaje, las preferencias horarias que tendría el paciente y el comentario que quiera transmitirle el paciente.
	paciente	JSON	Guarda información necesaria del documento del paciente que envió el mensaje.
	psicólogo	JSON	Guarda información necesaria del documento del psicólogo que recibió el mensaje.
	rechaza	Boolean	El psicólogo rechaza la solicitud de un paciente.
	mensajePsicologo - comentario - fecha - fechaCita	JSON - String - String - String	Mensaje que envía un psicólogo como respuesta a la solicitud de un paciente. Se guarda el comentario que quiera transmitirle el psicólogo, la fecha en que fué enviado el mensaje y la fecha de la cita en caso de que se la dé.

Tabla 2.38: Diccionario de datos: BSON Documento Mensaje

Documento	**Esquema BSON**		
	Campo	**Tipo de dato**	**Descripción**
Paciente	_id	ObjectID	Identificador del documento.
	edad	String	Fecha de nacimiento del psicólogo.
	email	String	E-mail del paciente.
	Genero	String	Género del paciente.
	diagnostico - porcentaje - patologia	Array JSON - Number - BSON	Diagnóstico resultado del cuestionario. Cada documento embebido en el array está definido por el porcentaje que tiene en una patología e información necesaria de esa patología.
	localizacion	String	Ciudad donde reside el paciente.
	password	String	Contraseña de la cuenta de usuario del paciente.
	psicologos	Array BSON	Conjunto de información de los psicólogos que le son asignados al paciente.
	telefono	Number	Teléfono del paciente.

Tabla 2.39: Diccionario de datos: BSON Documento Paciente

Documento	**Esquema BSON**		
	Campo	**Tipo de dato**	**Descripción**
Patología	_id	ObjectID	Identificador del documento.
	nombre	String	Nombre de la patología.
	preguntas	Array String	Conjunto de preguntas sobre los síntomas de la patología.
	respuesta	Number	Valor de una pregunta.

Tabla 2.40: Diccionario de datos: BSON Documento Patología

Documento	**Esquema BSON**		
	Campo	**Tipo de dato**	**Descripción**
Psicólogo	_id	ObjectID	Identificador del documento.
	aseguradora	String	Aseguradora para la que trabaja el psicólogo.
	comentarios - _idPaciente - fechaComentario - fechaRespuesta - respuesta - texto - valoracion	Array JSON - ObjectID - String - String - String - String - Number	Comentario del paciente y respuesta del psicólogo al comentario. Se guarda el identificador del paciente, la fecha en la que fue enviado, la fecha en la que fue respondido, la respuesta del psicólogo, el comentario del paciente y la valoración que le dá al psicólogo.
	consulta - online - presencial	JSON - Boolean - Boolean	Indica si el psicólogo hace consultas presenciales, a distancia o ambas.
	edad	String	Fecha de nacimiento del psicólogo.
	email	String	E-mail del psicólogo.
	estrellas	Number	Valoración media de los pacientes del psicólogo.
	experiencia	String	Experiencia laboral del psicólogo.
	formacion	String	Formación del psicólogo.
	foto	String	Foto del psicólogo.
	localizacion	String	Ciudad donde reside el psicólogo.

Tabla 2.41: Diccionario de datos: BSON Documento Psicólogo - Parte 1

Documento	**Esquema BSON**		
	Campo	**Tipo de dato**	**Descripción**
Psicólogo	patologias	Array String	Conjunto de nombres de las patologías que trata el psicólogo.
	ncolegiado	String	Número de colegiado del psicólogo.
	nombre	String	Nombre del psicólogo.
	password	String	Contraseña de la cuenta de usuario del psicólogo.
	precio	Number	Precio de la primera consulta que dé el psicólogo.
	telefono	Number	Teléfono del psicólogo.
	terapia	String	Tipo de terapia que realiza el psicólogo.
	tipo	String	Tipo del psicólogo.

Tabla 2.42: Diccionario de datos: BSON Documento Psicólogo - Parte 2

Escala	**Descripción**
Baja	El requisito puede sufrir cambios con facilidad
Media	Existe cierto grado de ocurrencia al cambio en el requisito
Alta	La probabilidad de cambios en el requisito es muy baja
Muy Alta	La probabilidad de cambios es casi nula

Tabla 2.43: Escala de estabilidad de los requisitos

RI-001	**Paciente**
Descripción	El sistema deberá almacenar la información correspondiente a los pacientes.
Datos específicos	E-mail Contraseña Género Edad Localización Teléfono Síntomas Diagnóstico Psicólogos que pueden tratar su caso
Estabilidad	Media

Tabla 2.44: RI-001 Paciente

RI-002	**Patología**
Descripción	El sistema deberá almacenar la información correspondiente a las patologías.
Datos específicos	Nombre Síntomas
Estabilidad	Alta

Tabla 2.45: RI-002 Patología

RI-003	**Psicólogo**
Descripción	El sistema deberá almacenar la información correspondiente a los psicólogos.
Datos específicos	Nombre
	E-mail Contraseña Género Edad Localización Teléfono Tipo Número Experiencia Formación Terapia Especialidad Aseguradora Patologías Valoración Precio Tipo de consulta: *Online* o presencial Gabinete
Estabilidad	Media

Tabla 2.46: RI-003 Psicólogo

RF - 001	**Acceso usuarios**
Descripción	Cualquier usuario registrado en la plataforma podrá acceder a la plataforma a través de la página de acceso.
Estabilidad	Alta
Dependencias	RI-001 RI-003 RF-002

Tabla 2.47: RF-001 Acceso usuarios

RF - 002	**Registro**
Descripción	Cualquier usuario podrá acceder a los servicios de la plataforma tras haber cubierto el formulario de registro.
Estabilidad	Alta
Dependencias	RI-001 RI-003

Tabla 2.48: RF-002 Registro

RF - 003	**Baja**
Descripción	Cualquier usuario registrado en la plataforma podrá darse de baja de la plataforma.
Estabilidad	Alta
Dependencias	RI-001 RI-003 RF-002

Tabla 2.49: RF-003 Baja

RF - 004	**Modificación de los datos**
Descripción	Cualquier usuario registrado en la plataforma podrá modificar sus datos de usuario.
Estabilidad	Alta
Dependencias	RI-001 RI-003 RF-001 RF-002

Tabla 2.50: RF-004 Modificación de los datos

RF - 005	**Emparejamiento**
Descripción	Se garantizará al paciente el emparejamiento con el psicólogo más adecuado para tratar su caso. Cualquier paciente registrado en la plataforma podrá especificar sus síntomas a través de un cuestionario asignándole al menos un psicólogo al paciente tras conocer su patología.
Estabilidad	Alta
Dependencias	RI-001 RI-002 RI-003 RF-001 RF-002

Tabla 2.51: RF-005 Emparejamiento

RF - 006	**Filtrado de los resultados en base a distintos criterios**
Descripción	Cualquier paciente, que haya obtenido resultados concluyentes en el cuestionario, podrá filtrarlos en base a distintos criterios como las características geográficas, el precio o su seguro médico.
Estabilidad	Alta
Dependencias	RI-003 RF-001 RF-002 RF-005

Tabla 2.52: RF-006 Filtrado de los resultados en base a distintos criterios

RF - 007	**Contacto del paciente con el psicólogo**
Descripción	El paciente podrá contactar con cualquiera de los psicólogos que le fueron asignados tras realizar el cuestionario.
Estabilidad	Alta
Dependencias	RI-001 RI-003 RF-001 RF-002 RF-005

Tabla 2.53: RF-007 Contacto del paciente con el psicólogo

RF - 008	**Valoración del psicólogo por parte del paciente**
Descripción	El paciente podrá valorar al psicólogo que le haya atendido.
Estabilidad	Alta
Dependencias	RI-001 RI-003 RF-001 RF-002 RF-005

Tabla 2.54: RF-008 Valoración del psicólogo por parte del paciente

Requisitos no funcionales

Los requisitos no funcionales son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restrucciones de tiempo, sobre el proceso de desarrollo y estándares[**?**].

RNF - 001	**Encriptado de datos**
Descripción	Los datos de los pacientes almacenados en la base de datos deberán estar cifrados con un algoritmo de cifrado.
Estabilidad	Alta
Dependencias	(Es transversal)

Tabla 2.55: RNF-001 Encriptado de datos

RNF - 002	**Tiempo de respuesta de asignación**
Descripción	El paciente tras enviar las respuestas del cuestionario, los resultados de asignación de psicólogo del mismo deberán tener un tiempo de respuesta de como máximo 5 segundos.
Estabilidad	Alta
Dependencias	RF-005

Tabla 2.56: RNF-002 Tiempo de respuesta de asignación

Matrices de rastreabilidad

El rastreo es una propiedad de la especificación de requisitos que refleja la facilidad de encontrar requisitos relacionados. Es importante ya que nos da a conocer las relaciones que hay entre éstos y el diseño del sistema, y así, si se proponen cambios, se podrá rastrear cuál es el impacto de esos cambios en los otros requisitos y en el diseño del sistema.

		Requisitos de información		
		RI-001	RI-002	RI-003
Requisitos funcionales	RF-001	X		X
	RF-002	X		X
	RF-003	X		X
	RF-004	X		X
	RF-005	X	X	X
	RF-006			X
	RF-007	X		X
	RF-008	X		X

Tabla 2.57: Matriz de rastreabilidad: RF / RI

Matriz de rastreabilidad: RF / RI2.57

Matriz de rastreabilidad: CU / RF2.58

2.7.4. Algoritmo de *matching*

Para poder emparejar al psicólogo que puede tratar la patología de un paciente, se ha diseñado el algoritmo descrito a continuación. Como el estudio del test validado científicamente que determina cómo se van a evaluar las patologías, pacientes y psicólogos en cuestión no está diseñado todavía, se ha tenido que realizar una breve y pequeña aproximación de cómo sería el funcionamiento aparente del cuestionario.

		Requisitos funcionales						
		RF-001	RF-002	RF-003	RF-004	RF-005	RF-006	RF-007
Casos de uso	CU-001	X						
	CU-002	X	X					
	CU-003	X	X			X		
	CU-004	X	X			X	X	
	CU-005	X	X			X		
	CU-006	X	X					
	CU-007	X	X			X		X
	CU-008	X	X			X		
	CU-009	X	X			X		
	CU-010	X	X					
	CU-011	X						
	CU-012	X						
	CU-013	X	X			X		
	CU-014							
	CU-015							
	CU-016							

Tabla 2.58: Matriz de rastreabilidad: CU / RF

En nuestro proyecto, cada patología tendrá asociadas una serie de preguntas. La respuesta a todas las preguntas nos dará una ponderación de valor 1. En el conjunto de las posibles preguntas de una patología, existirá una pregunta principal cuyo valor es 0,5, y el valor de cada una del resto, viene dado por la siguiente fórmula:

$$valor_pregunta = \frac{0,5}{n_{preguntas} - 1} \tag{2.1}$$

Teniendo en cuenta esto, el test de la plataforma está dividido en dos partes:

La primera, muestra la pregunta más relevante de cada patología estimada por nuestro experto psicólogo. En el momento que el paciente indica que tiene el síntoma especificado en la pregunta A (correspondiente a la patología A), se le diagnosticará que posee la patología A con una ponderación del 0,5.

La segunda, muestra las preguntas restantes a las patologías marcadas en la primera parte, es decir, aquellas patologías que le han sido diagnosticadas con una ponderación del 0,5. Por cada pregunta marcada en la segunda parte, se sumará el valor de esa pregunta al porcentaje actual.

Una vez enviado el formulario, sólo si el porcentaje es mayor o igual al 0,7, se determina que el paciente padece esa patología.

Un ejemplo práctico (figura 2.6) Si el paciente marcase las preguntas que aparecen en color gris, en la segunda parte del test aparecerían el resto de preguntas de la patología en cuestión.

Tras finalizar el test, los resultados serían:

- Trastorno adaptativo 0,625: Se descartaría.
- Síntomas psicóticos 0,75

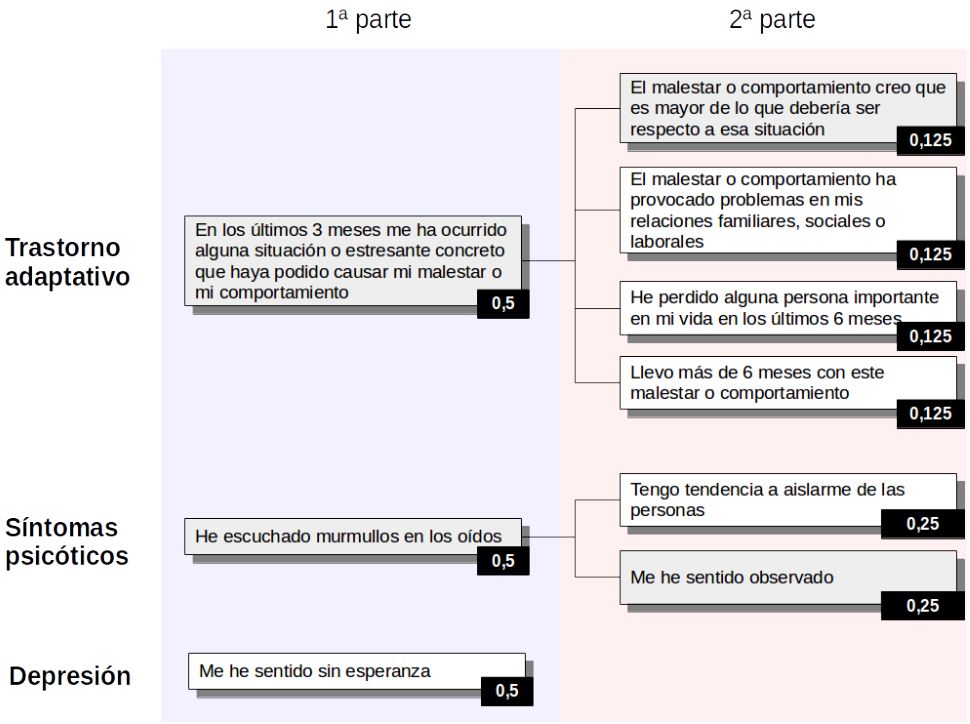


Figura 2.6: Ejemplo algoritmo de *matching*

Capítulo 3

Diseño

3.1. Diseño de la interfaz

El diseño de la interfaz de usuario crea un medio eficaz de comunicación entre los seres humanos y la computadora.

3.1.1. Principios de diseño de la experiencia de usuario

Para poder realizar un diseño de interfaz centrado en el usuario, primero hemos de conocer quiénes y cómo son nuestros usuarios. Durante el desarrollo del modelo de negocio pudimos entrevistar a nuestros posibles clientes (futuros usuarios) y pudimos hacernos una idea de cuál es nuestro nicho de mercado.

En nuestra plataforma, tenemos dos tipos de perfiles: Los pacientes3.1 y los psicólogos3.2.

Hacer un diseño centrado en el usuario implica tenerlo presente a lo largo de todo el proceso de diseño y tratar de entender cuáles son sus necesidades, intereses y limitaciones.

3.1.2. Principios de diseño de la interfaz de usuario

Layout

Nuestra interfaz seguirá la regla de los tercios para ayudar a concretar el enfoque del usuario. La regla de los tercios es una forma de composición para ordenar objetos dentro de la imagen al dividirla en nueve partes iguales utilizando dos líneas imaginarias paralelas y equiespaciadas de forma horizontal y otras dos con la mismas características de forma vertical. Los puntos donde se cortan las líneas son los puntos de intersección y sirven para distribuir los elementos de la página. Entre los puntos de intersección se ha de ubicar el centro de atención para crear una imagen estéticamente agradable y equilibrada[37].

Pacientes	
Demografía	España
Edad	Mayores de 16 años
Experiencia laboral	Estudios mínimos
Contexto profesional	Cualquiera
Necesidades e intereses	Buscar al psicólogo más adecuado para tratar su problemática.
	Reducir las preocupaciones de aquellas poblaciones que no puedan acudir físicamente a una consulta por motivos de desplazamiento, urgencias, estigmas sociales...
¿Cuándo y dónde utilizaran este servicio?	Cuando surja la necesidad de contactar con un psicólogo y a través de un ordenador.

Tabla 3.1: Perfil paciente

Psicólogos	
Demografía	España
Edad	Mayoría de edad
Experiencia laboral	Estudios superiores
Contexto profesional	Psicólogos con especialidad clínica o sanitaria que estén colegiados para el ejercicio de la actividad profesional.
Necesidades e intereses	Conseguir un flujo constante de pacientes.
¿Cuándo y dónde utilizaran este servicio?	Periódicamente y a través de un ordenador.

Tabla 3.2: Perfil psicólogo

La librería Bootstrap, que es una de las librerías escogidas para el diseño de la página, posee un sistema de cuadrículas[29] que permite dividir la página en filas y columnas. La cuadrícula siempre divide la página en 12 columnas. Como 12 es múltiplo 3, se puede aplicar fácilmente la regla de los tercios a este sistema de cuadrículas.

Por otra parte, también se debe tener en cuenta el público que va a tener nuestra página a la hora de disponer los objetos en ella. La jerarquía visual es importante porque dirige la atención de los ojos del usuario. En nuestro caso, nuestra población es occidental, por lo que el usuario comienza a leer la página desde la esquina superior izquierda. Por este motivo, es interesante situar el logo de nuestra plataforma en esta esquina y la navegación a su lado de forma horizontal.

Colores

En la página predominan los colores que, para nuestros futuros usuarios pertenecientes a la población occidental, transmiten:

- Verde: Seguridad, salud
- Azul: Seguridad, confianza, estabilidad, veracidad, lealtad

El verde y azul son colores análogos, puesto que se encuentran uno pegado a otro en la rueda de color.

Es importante tener en cuenta la cultura de nuestros usuarios, puesto que entre culturas pueden existir connotaciones de los colores totalmente opuestas.

Para el color de las letras se han utilizado el blanco y el negro (colores neutros) y hacen contraste con el color predominante del background y de los elementos de la página.

3.1.3. Fuentes y tipografía

La tipografía predominante en toda la página es Lato[32], pero para aquellos mensajes que queramos resaltar en algún momento puntual se utilizará Merriweather[33].

La elección de las fuentes se debe a que la tipografía contextualiza el tipo de contenido de la página, no sólo lo hace a nivel verbal sino que también de forma visual. El lector, primero identifica los patrones gráficos de la página, y después, analiza el lenguaje y lee.

- Lato Es de tipo *sans serif*[1] transicional: Los trazos son fuertes y los caracteres son derechos y uniformes. Transmite sencillez y modernez. Se suele utilizar en tecnología y aplicaciones portables.

[1] ***Sans serif*:** Fuentes con ausencia de *serif*.

- Merriweather Es de tipo egipcio (*slab serif*): Existe muy poco contraste entre los trazos, y la *serif* es gruesa. Transmite autoridad pero en tono amistoso. Se suele utilizar en *marketing* y aplicaciones promocionales.

3.1.4. Interacción persona-ordenador

Sociedad de la información

La tendencia que existe hoy en día es digitalizar toda clase de servicios. Los servicios TIC permiten acceder a la salud, el ocio, el bienestar, la formación... derechos básicos que pertenecen a toda la ciudadanía.

A pesar de ello, existen personas que no tienen fácil acceso a este tipo de servicios ya sea por razones de limitaciones geográficas como es el caso del rural, aspectos de género culturales o religiosos, la edad, aspectos socieconómicos (personas bajo el umbral de difícil acceso a las TIC) y la discapacidad. Aunque el proyecto no pueda abarcar a toda clase de colectivos por limitaciones de tiempo, sí es importante tenerlos en cuenta para el futuro.

Diversidad funcional

La diversidad funcional es inherente al ser humano: Una persona experimenta variaciones en su capacidad de dependencia a lo largo de su vida, aunque sea de manera temporal. Por ejemplo, a veces nos sentimos limitados por no poder utilizar el ordenador portátil al no tener batería; o con la edad, la gente obtiene discapacidades que antes no tenía como la pérdida de vista.

Accesibilidad

La accesibilidad la interacción persona-ordenador es el conjunto de propiedades que debe incorporar un producto, servicio o sistema, de forma que el mayor número posible de personas, en el mayor número posible de circunstancias, que sea comercialmente práctico tener en cuenta, puede acceder a él y usarlo.[36]

Un producto es accesible si, por mucho tiempo o esfuerzo que requiera, la tarea puede realizarse.

Usabilidad

La usabilidad es la efectividad, eficiencia y satisfacción con la que usuarios específicos pueden abarcar unos objetivos determinados en un entorno particular[7].

Por tanto, los sistemas deben ser usables y accesibles para que la inmensa mayoría de las personas puedan utilizarlo con calidad.

Diseño universal

El diseño universal es la estrategia que tiene como objetivo diseñar productos y servicios que puedan se utilizados por el mayor número de personas, considerando que existe una amplia variedad de habilidades humanas y no una habilidad media, sin necesidad de llevar a cabo una adaptación o diseñi especializado, simplificando la vida de todas las personas con independencia de su edad, talla o capacidad[38].

En la práctica, esto supone un auténtico reto, por lo que el diseño para todos acaba equivaliendo a diseño para "la mayoría". Aquí es donde se entra a valorar la posibilidad de añadir productos de apoyo para que pueda ser utilizado por las minorías.

En España hay leyes que tratan de combatir con este tipo de discriminación como la Ley 51/2005, de 2 de diciembre, de igualdad de oportunidades, no discriminación y accesibilidad universal de las personas con discapacidad. Por otra parte, AENOR posee la norma UNE 139803:2012. Requisitos de Accesibilidad para contenidos en la web[27].

Para lograr que nuestra plataforma pueda ser utilizada por el mayor número de personas posible se ha decidido seguir los principios heurísticos de Nielsen y Molich.

Principios heurísticos

La Interacción Persona Ordenador (IPO) presenta a la Evaluación Heurística (EH) como un método de evaluación de la usabilidad por inspección que debe ser llevado a cabo a través de unos principios heurísticos previamente establecidos. Por ser un método de evaluación de la usabilidad, tiene como objetivo medir la calidad de la interfaz de cualquier sistema interactivo en relación a su facilidad para ser aprendido y usado por un determinado grupo de usuarios en un determinado contexto de uso.

Aplicar los principios heurísticos nos sirve de guía para el proceso de diseño y nos permite identificar problemas de usabilidad en las interfaces de usuario. En nuestra aplicación, tendremos en cuenta los principios heurísticos de Nielsen y Molich:

1. Visibilidad del estado del sistema El sistema debe siempre mantener a los usuarios informados del estado del sistema, con una realimentación apropiada y en un tiempo razonable.

2. Lenguaje de los usuarios El sistema debe hablar el lenguaje de los usuarios, utilizando convenciones del mundo real, disponiendo la información en un orden natural y lógico.

3. Control y libertad para el usuario Los usuarios eligen a veces funciones del sistema por error y necesitan una salida del estado indeseado sin tener que pasar por un diálogo extendido.

4. Consistencia y estándares Se deben seguir las normas y convenios de la plataforma para las que se implementa el sistema.

5. Ayuda a los usuarios para reconocimiento, diagnóstico y recuperación de errores Los mensajes de error deben expresarse en un lenguaje claro y explicativo.

6. Prevención de errores Se debe prevenir la aparición de errores que mejor que generar buenos mensajes de error.

7. Reconocimiento antes de cancelación El usuario no debería tener que recordar la información de una parte del diálogo a la otra.

8. Flexibilidad y eficiencia de uso Las instrucciones para el uso del sistema deben ser visibles o fácilmente accesibles siempre que se necesiten.

9. Estética de diálogos y diseño minimalista No deben contener la información que sea inaplicable o se necesite raramente. Cada unidad adicional de la información en un diálogo compite con las unidades relevantes de la información y disminuye su visibilidad.

10. Ayuda general y documentación Aunque es mejor si el sistema se pueda usar sin documentación, puede ser necesario disponer de ayuda y documentación. Ésta ha de ser fácil de buscar, centrada en las tareas del usuario, tener información de las etapas a realizar y que no sea muy extensa. [43]

3.1.5. *Mockup*

3.2. Diseño de la navegación

Para definir las rutas de navegación que permiten a los usuarios acceder al contenido y a las funciones de la plataforma web, debemos identificar la semántica de la navegación para los distintos usuarios del sitio y definir la sintaxis para efectuar la navegación.

3.2.1. Semántica de la navegación

Se define a partir del rol que toma un actor dentro de un caso de uso, ya que cada actor tiene distintos requisitos de navegación. A medida que un usuario interactúa con la web, encuentra una serie de unidades semánticas de navegación (USN) que son un conjunto de estructuras de información y navegación relacionadas que colaboran para el cumplimiento de un subconjunto de requisitos de dicho usuario.

Una USN está compuesta por un conjunto de elementos de navegación llamados formas de navegar (FdN) que representan la mejor ruta de navegación para

lograr una meta específica. Está formada por un conjunto de nodos de navegación conectados por vínculos, que en algún pueden tratarse de otra USN.

Para cada caso de uso, se procede a diseñar su USN. Las correspondencias se pueden apreciar en la tabla3.3.

Caso de uso	Unidad semántica de navegación
CU-001	USN-001 (Paciente)
CU-002	USN-002
CU-003	USN-003
CU-004	USN-004
CU-005	USN-005
CU-006	USN-006
CU-007	USN-007
CU-008	USN-008
CU-009	USN-009
CU-010	USN-010
CU-011	USN-011 (Paciente)
	USN-011 (Psicólogo)
CU-012	USN-012

Tabla 3.3: Correspondencia entre CU y USN

3.2.2. Sintaxis de navegación

La sintaxis de navegación refleja la mecánica de la navegación para cada USN. Para nuestra plataforma web, se ha diseñado un mapa del sitio que describe todas las posibles navegaciones existentes. Para cada tipo de usuario (perfil o psicólogo), existe un mapa3.14 diferente.

3.3. Tecnologías utilizadas

3.3.1. Frameworks

Un *framework* es una arquitectura de *software* que modela las relaciones generales de las entidades del dominio, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.

Los *frameworks* involucrados en nuestro proyecto son MEAN Stack, Bootstrap y Semantic UI.

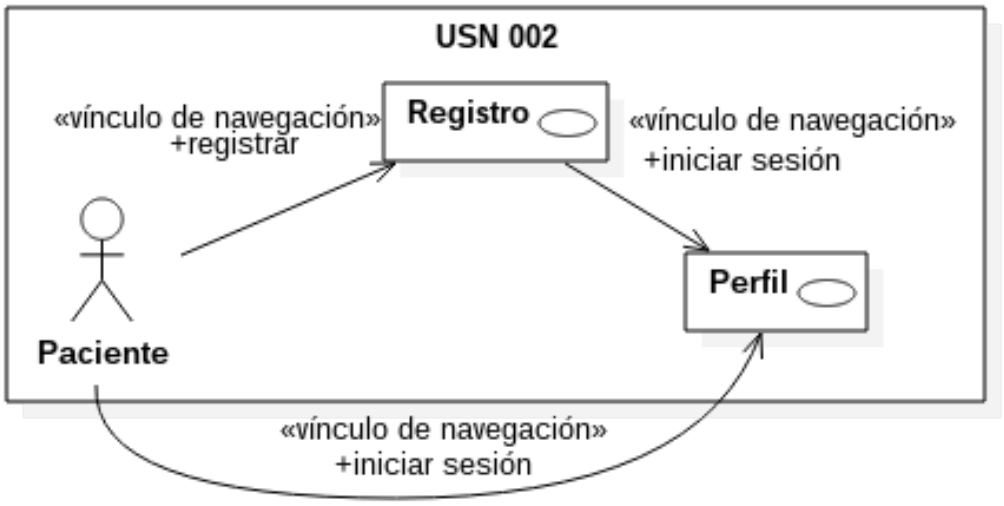


Figura 3.1: USN-001

Figura 3.2: USN-002

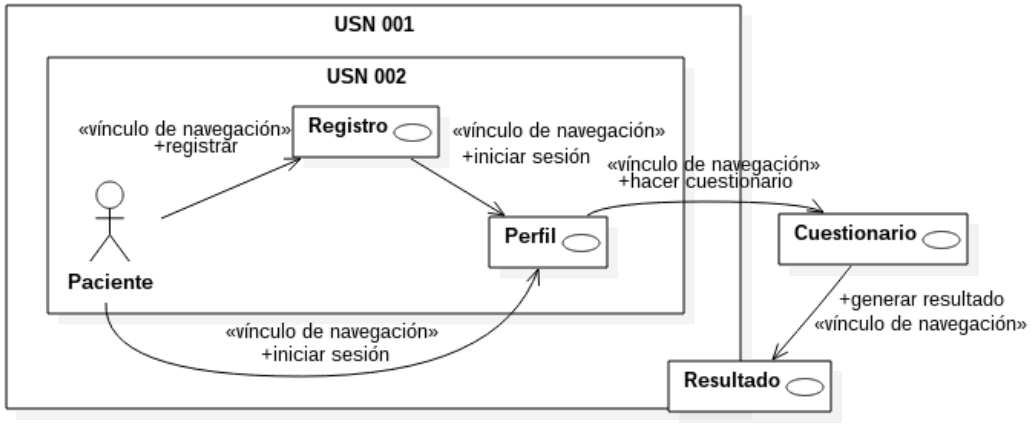


Figura 3.3: USN-003

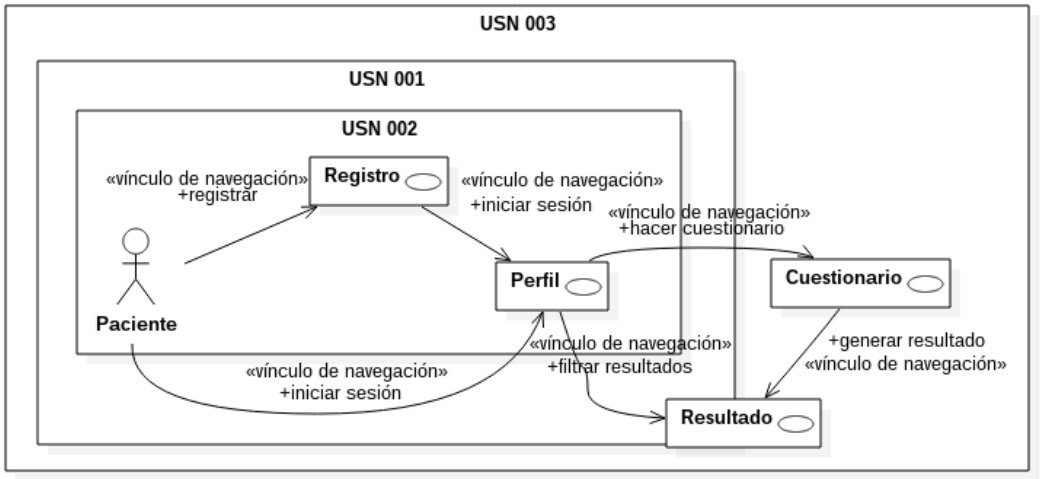


Figura 3.4: USN-004

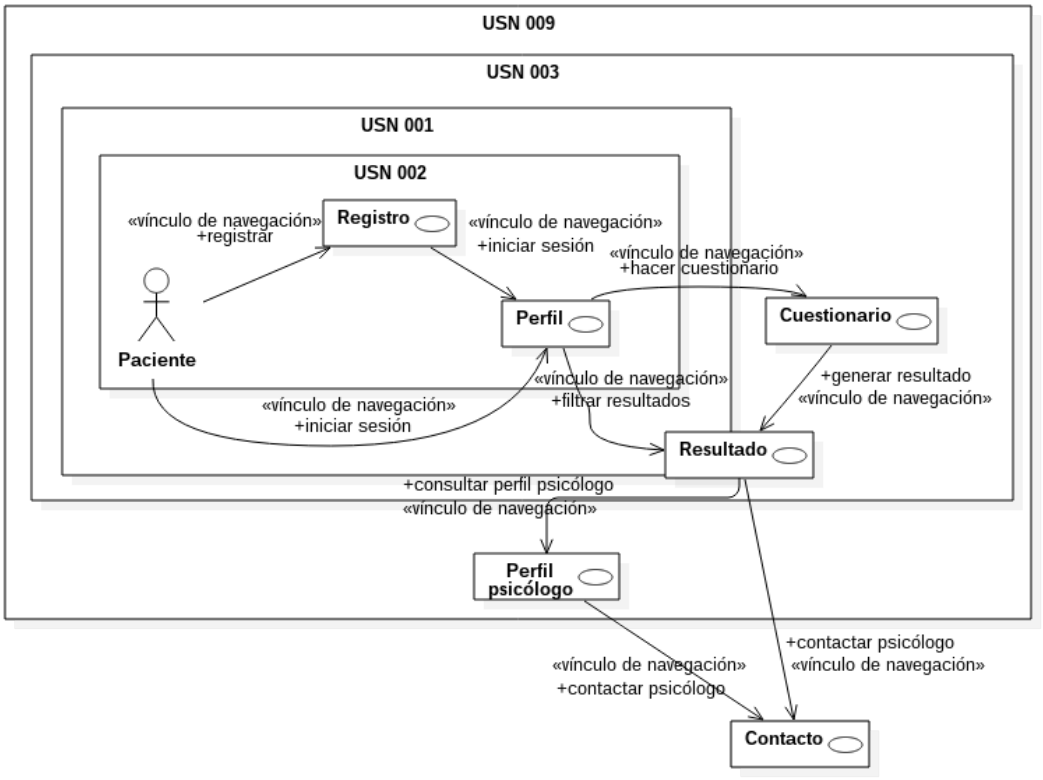


Figura 3.5: USN-005

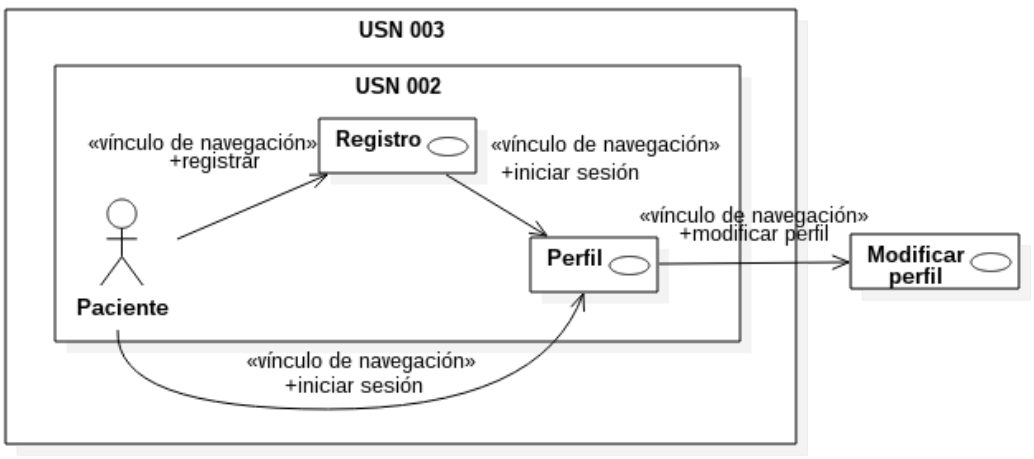


Figura 3.6: USN-006

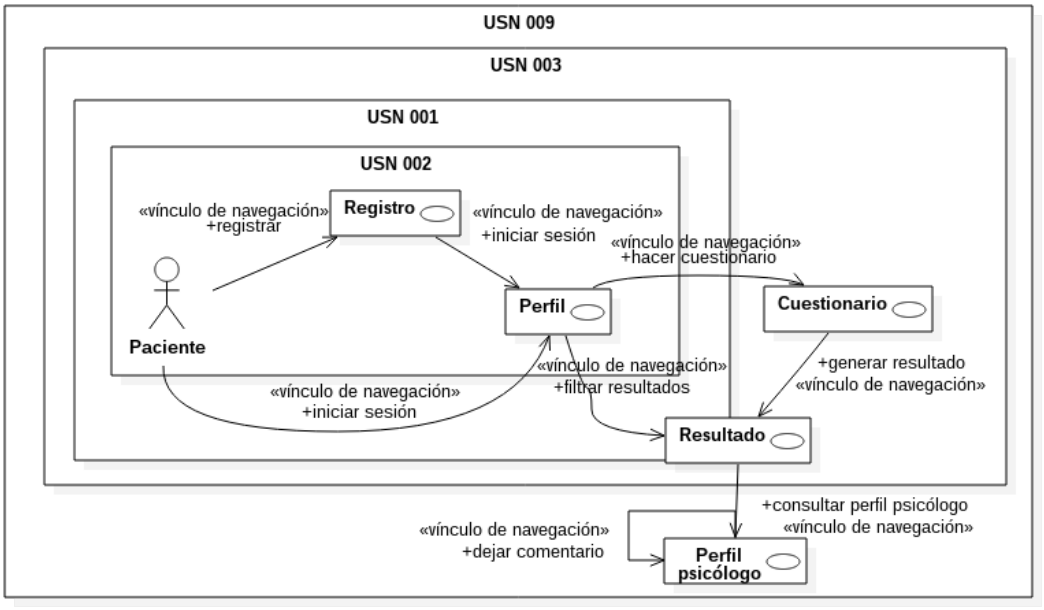


Figura 3.7: USN-007

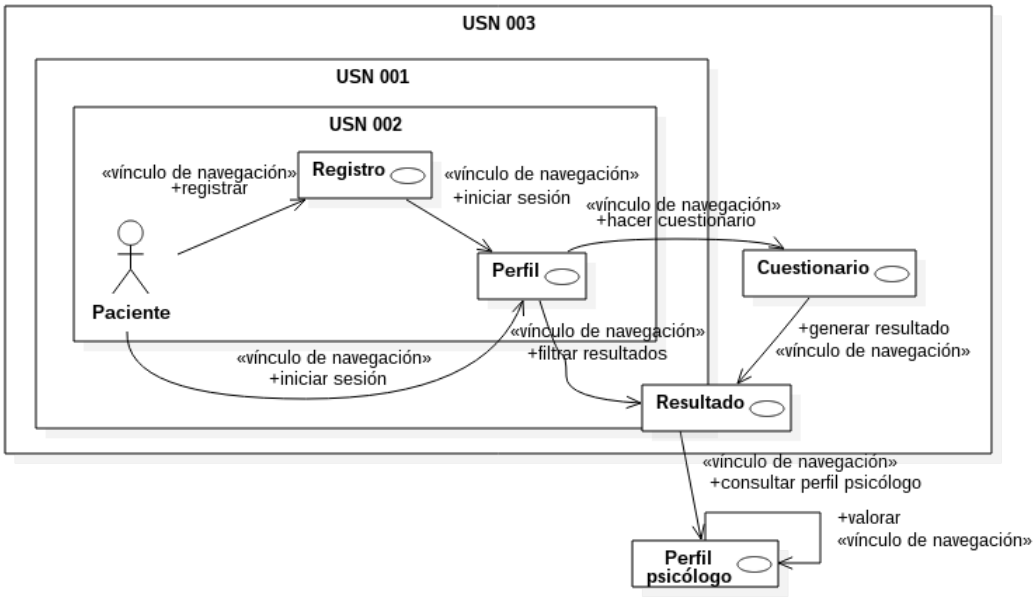


Figura 3.8: USN-008

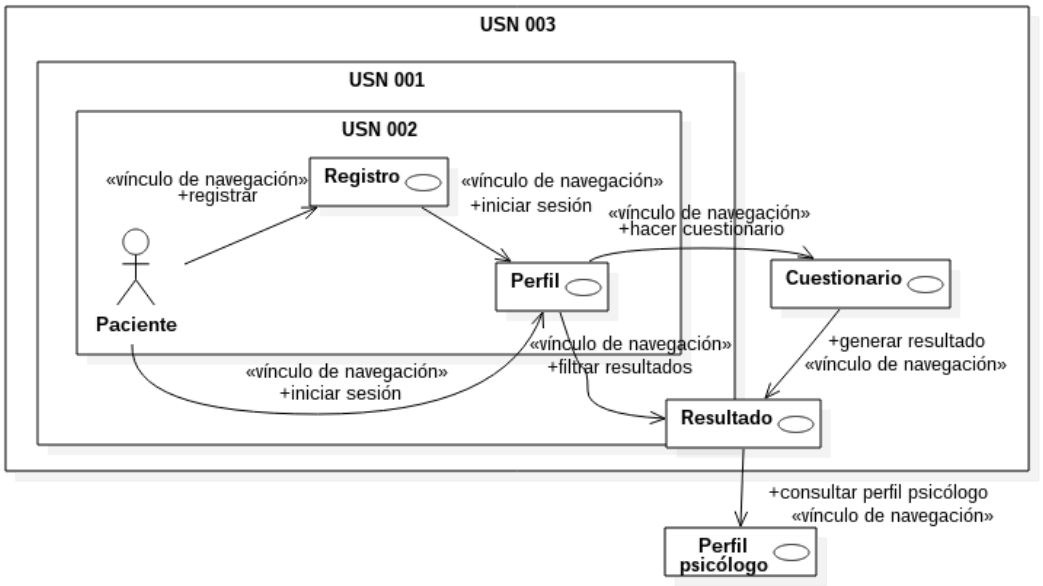


Figura 3.9: USN-009

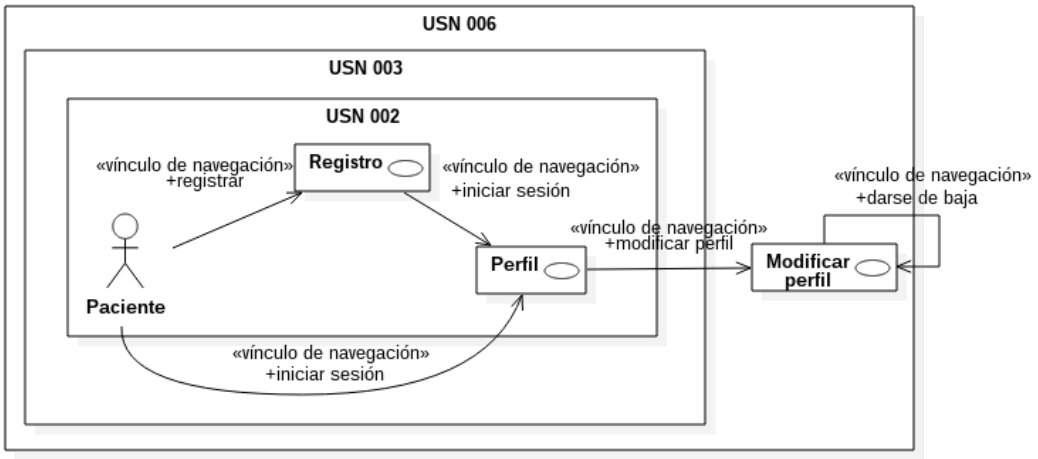


Figura 3.10: USN-010

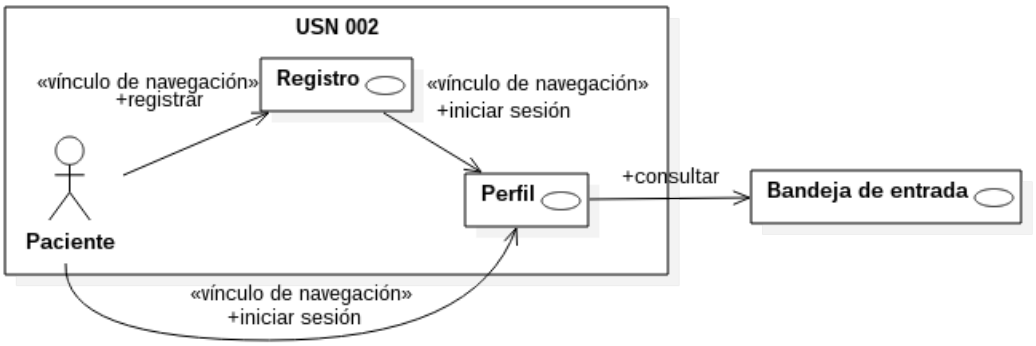


Figura 3.11: USN-011 (Paciente)

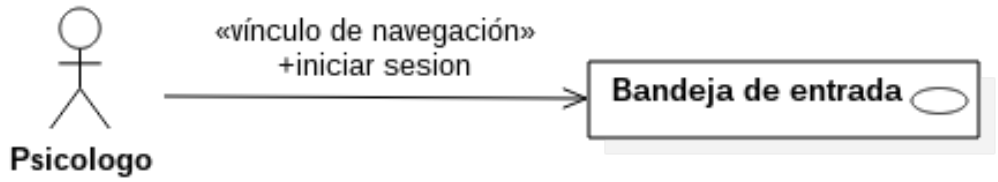


Figura 3.12: USN-011 (Psicólogo)

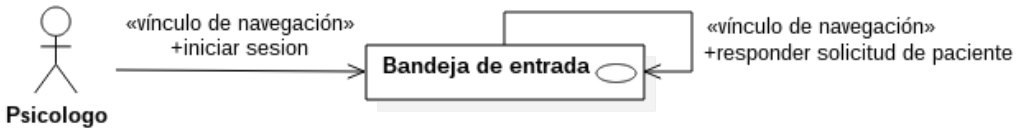


Figura 3.13: USN-012

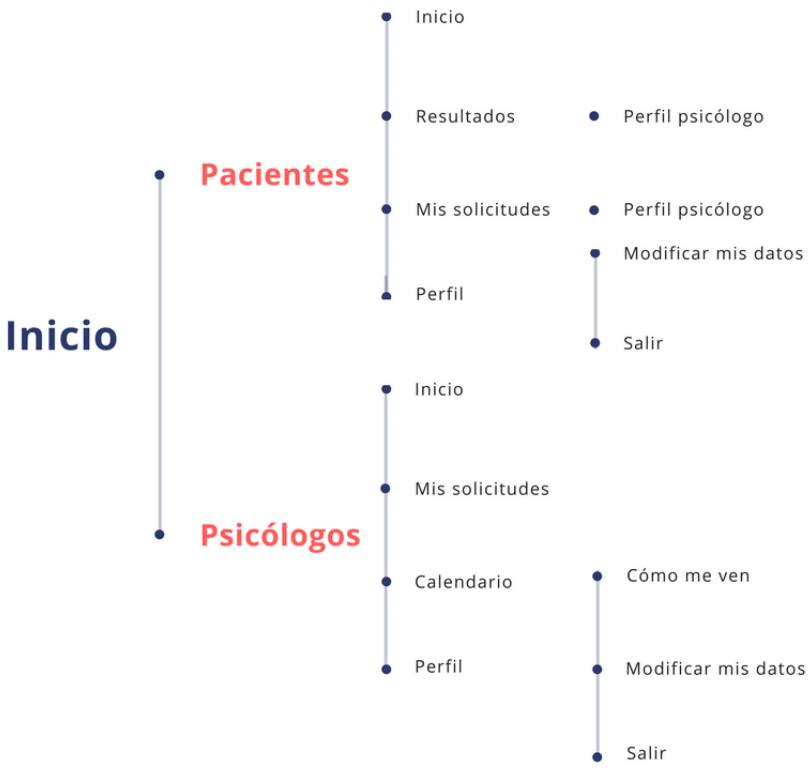


Figura 3.14: Mapa de navegación

MEAN Stack

MEAN stack es un *framework* para el desarrollo de aplicaciones, y páginas web dinámicas, basadas en JavaScript: MongoDB, ExpressJS, AngularJS y NodeJS, lo cual permite que se integren entre ellas eficazmente.

Con MongoDB podemos almacenar nuestros documentos en formato JSON, se pueden escribir consultas en nuestro servidor ExpressJS y NodeJS, y del mismo modo, pasar esos documentos JSON a nuestro *frontend* hecho con AngularJS.

El *debugging* y la administración de la base de datos se vuelven más sencillas cuando el objeto almacenado en la base de datos es idéntico al objeto que tu cliente JavaScript puede ver[40].

Algunos de los motivos por los que escogí este *framework* es por su escalabilidad, rapidez y flexibilidad, ya que permite que en un futuro sea sencillo poder adaptar la aplicación a plataformas móviles, o añadir cambios con facilidad.

Los componentes que forman el *framework* son:

- **MongoDB** MongoDB es una base de datos ágil NoSQL orientada a documentos que permite que los esquemas cambien rápidamente a medida que las aplicaciones evolucionan, proporcionando siempre la funcionalidad que los desarrolladores esperan de las bases de datos tradicionales, tales como índices secundarios, un lenguaje completo de búsquedas y consistencia. En resumen, MongoDB brinda escalabilidad, rendimiento y gran disponibilidad[41].

- **ExpressJS** ExpressJS es una middleware de aplicaciones web Node.js minimalista y flexible que proporciona un conjunto sólido de características para aplicaciones web y móviles[14]. Se trata de una API REST que utiliza métodos HTTP para obtener datos o generar operaciones sobre esos datos.

- **AngularJS** AngularJS es un framework para construir client applications en HTML y Javascript, aunque también puede ser otro lenguaje como TypeScript que sea compilado a JavaScript. El framework posee un conjunto de librerías funcionales, y otras opcionales[4].

  Angular permite fácilmente construir aplicaciones web, ya que combina declarative templates, dependency injection y end to end tooling, e integra buenas prácticas de programación. Las aplicaciones desarrolladas con Angular funcionan tanto para web, móvil o escritorio[2].

- **NodeJS** NodeJS es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. Node.js usa un modelo de operaciones entrada/salida sin bloqueo (asíncrono) y orientado a eventos, que lo hace liviano y eficiente, ya que nunca se bloquea.

  Node está diseñado sin hilos, este presenta un bucle de eventos como un entorno en vez de una librería. Node simplemente ingresa el bucle de eventos

después de ejecutar el *script* de entrada. Node sale del bucle de eventos cuando no hay más callbacks que ejecutar[1].

Bootstrap

Bootstrap es un *framework* de código abierto para desarrollar con HTML, CSS y JS. Contiene características como un grid responsive, docenas de componentes, JavaScript plugins, tipografía, control de formularios y capacidad de personalización[12].

Semantic UI

Semantic UI es un *framework* que permite a los desarrolladores contruir rápidamente sitios web, con HTML conciso, JavaScript intuitivo, y simplicando el *debugging*. Semantic está diseñado de manera *responsive* permitiendo que la aplicación sea escalable en múltiples dispositivos. Semantic está preparado para poder asociarse con otros frameworks como React, Angular, Meteor y Ember[5].

3.3.2. Lenguajes de programación

Los principales lenguajes de programación utilizados son:

CSS3

Hojas de Estilo en Cascada (Cascading Style Sheets) es el lenguaje utilizado para describir la presentación de documentos HTML o XML CSS describe como debe ser renderizado el elemento estructurado en pantalla, en papel, hablado o en otros medios[3].

HTML

HTML, que significa Lenguaje de Marcado para Hipertextos (HyperText Markup Language) es el elemento de construcción más básico de una página web y se usa para crear y representar visualmente una página web. Determina el contenido de la página web, pero no su funcionalidad[6].

JavaScript

Es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de *script* para páginas web, pero también usado en muchos entornos sin navegador, tales como node.js.

Es un lenguaje script multi-paradigma, basado en prototipos[2], dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa[8].

3.3.3. *Modules*

Un *module* es cualquier fichero o directorio que puede ser cargado por NodeJS.

Mongoose

Mongoose proporciona una sencilla, solución basa en esquemas para el modelo de los datos de la aplicación[21].

Bcrypt

Bcrypt permite *hashear* y comparar contraseñas en Node.

NodeMailer

NodeMailer permite a las aplicaciones el envío de mensajes.

FullCalendar

FullCalendar es un calendario de eventos JavaScript personalizable y de código abierto[15].

3.3.4. Librerías

Una librería es un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca.

Places de Google Maps JavaScript API

Las funciones de la biblioteca JavaScript de Google Places permite que una aplicación busque sitios (definidos en esta API como establecimientos, ubicaciones geográficas o puntos de interés destacados) dentro de un área definida, como los límites de un mapa o alrededor de un punto fijo. También, ofrece una función de autocompletado que puedes usar para dar a tus aplicaciones el comportamiento de escritura anticipada del campo de búsqueda de Google Maps. Cuando un usuario comienza a escribir una dirección, la función de autocompletado termina la tarea[10].

[2]La programación basada en prototipos es un estilo de programación orientada a objetos que reutiliza comportamientos de objetos existentes.

3.3.5. *Middleware*

Un *middleware* proporciona la lógica de intercambio entre aplicaciones. El *middleware* abstrae de la complejidad y heterogeneidad de las redes de comunicaciones subyacentes, así como de los sistemas operativos y lenguajes de programación, proporcionando una API para la fácil programación y manejo de aplicaciones distribuidas.

Passport

Passport es un *middleware* de autenticación para NodeJS. Extremadamente flexible y modular. También puede ser utilizado en cualquier aplicación web basada en Express. Tiene múltiples estrategias que soportan autenticación utilizando nombre y contraseña, Facebook, Twitter, y más[23].

3.4. Herramientas utilizadas

Las herramientas utilizadas en este trabajo fueron:

- **Brackets**
  - *Descripción:* Brackets es un editor de texto de código abierto[13].
  - *Uso:* Desarrollo software de la aplicación.
- **Git**
  - *Descripción:* Git es un sistema de código abierto de control de versiones distribuído diseñado para manipular cualquier tipo de proyectos con rapidez y eficacia[16].
  - *Uso:* Gestión de la configuración del proyecto.
- **GitHub**
  - *Descripción:* Es una plataforma de desarrollo colaborativo que permite alojar proyectos utilizando el sistema de control de versiones de Git[17].
  - *Uso:* Gestión de la configuración del proyecto.
- **Grunt**
  - *Descripción:* Es una herramienta que permite simplificar el proceso de construcción (build) de proyectos en JavaScript. Sirven para automatizar tareas repetitivas como minificación, compilación, testeo unitario...[18]
  - *Uso:* Durante el desarrollo de la aplicación.

- **LaTeX**
  - *Descripción:* Es un sistema tipográfico que incluye características diseñadas para la producción de documentación técnica y científica[19].
  - *Uso:* Elaboración de la memoria final.
- **LibreOffice**
  - *Descripción:* Es un conjunto de aplicaciones de oficina: Writer, el procesador de textos, Calc, la hoja de cálculos, Impress, el editor de presentaciones, Draw, nuestra aplicación de dibujo y diagramas de flujo; entre otros[20].
  - *Uso:* Borradores y algunos diagramas del proyecto.
- **Pencil**
  - *Descripción:* Pencil es una GUI de prototipado de código abierlo para creación de *mockups*[24].
  - *Uso:* Elaboración del *mockup*.
- **NPM**
  - *Descripción:* NPM es un gestor de paquetes JavaScript. Es el repositorio más grande de librerías de código abierto en el mundo[22].
  - *Uso:* Instalación de paquetes y librerías software.
- **Robomongo**
  - *Descripción:* Robomongo es una GUI que maneja la shell de MongoDB[25].
  - *Uso:* Gestión de la base de datos.
- **StarUML**
  - *Descripción:* StarUML es una herramienta de modelado UML que permite hacer multitud de tipos de diagramas como de clase, de objeto, de casos de uso, de componente, entre otros[26].
  - *Uso:* Diseño de los diagramas del proyecto.

3.5. Seguridad

3.5.1. ¿Qué es Bcrypt?

Niels Provos y David Maxieres diseñaron Bcrypt, una función de *hashing* de contraseñas basado en el tipo de cifrado Blowfish. Es empleado en algunas distribuciones de Linux por defecto.

Cuando se genera un *hash* asociado a la contraseña por lo general los algoritmos comunes (md5, sha-1,...) incorporan un valor de *salt*, este fragmento se emplea para generar dicho *hash*, de esta forma se consigue que dos contraseñas iguales que generarían el mismo *hash* no lo hagan, algo muy importante para luchar contra ataques de fuerza bruta, así como para dificultar los ataques de Rainbow table.

Los ataque de Rainbow table son efectivos cuando las contraseñas son "hasheadas" de la misma manera, de esta forma para dos contraseñas iguales el hash sería el mismo. Así es como surge el *salt*, añadir un *hash* a cada contraseña nos devuelve dos *hashes* distintos para la misma contraseña. Es importante evitar la *reutilización* de *salt*. Sin embargo no es suficiente el uso de *salt* para el almacenamiento de contraseñas debido a que *salt* pasa a ser inútil para ataques de fuerza bruta o de diccionario.

3.5.2. Ventajas de Bcrypt

En este punto es donde entra Bcrypt, para solucionar el problema en el que nos hallamos debemos hablar del número de iteraciones. Esto mejora a los *hashes* comunes gracias a su lentitud. ¿Qué quiere decir esto? Empleando una variante de cifrado Blowfish se introduce un factor de trabajo que permite controlar el coste de la función *hash*. Mediante este factor Bcrypt se actualiza de acuerdo a la ley de Moore con la evolución de la tecnología.

Para verlo más concretamente podemos ver la tabla de comparación siguiente, y recalcar que aunque las contraseñas no requieran de una protección tan elevada, gracias al factor de trabajo la optimización de bcrypt se consigue con un buen equilibrio entre velocidad y seguridad. El sacrificio de un poco de rendimiento se traduce en un aumento de la seguridad. Esta tabla muestra una prueba de la fuerza de bcrypt, que ha sido lanzado en un clúster de 25 GPU para la rotura de contraseñas en *hash*, han resultado los siguientes datos:

- md5($password)
  - 180 billones resultados/s
  - 9.4 Horas
- sha1($password)
  - 61 billones resultados/s
  - 27 Horas
- md5crypt
  - 77 millones resultados/s
  - 2.5 Años

- bcrypt con un factor de 5
  - 71 mil resultados/s
  - 2700 Años

3.5.3. Funcionamiento

La librería bcrypt nos permite crear el *hash* de una contraseña mediante *saltRounds*, este valor nos da control sobre el factor de coste de procesado de datos. A mayor valor de *saltRound* mayor coste de cálculo del *hash* asociado a una contraseña. Por defecto este valor viene situado en 10.

En el caso del registro de usuario se envía la contraseña y el valor del parámetro *saltRounds* a la librería para que la cifre, esta nos devuelve el *hash* asociado a la contraseña en el usuario de la base de datos. Cuando el usuario quiere entrar la contraseña introducida se cifra, para ello se busca el *salt* asociado al usuario de la base de datos. Una vez cifrada con el *salt* se comparan los *hashes*. La librería nos devuelve un booleano que indica si las contraseñas coinciden o no, con lo cual podemos dejar entrar el usuario o devolver un error.

3.5.4. Cifrado Blowfish

Bruce Schneier en 1993 diseña un codificador de bloques simétricos de 64 bits, con claves de hasta 448 bits, es empleado en un abundante número de productos de cifrado. No tiene técnicas de criptoanálisis que hayan resultado efectivas contra este algoritmo. Su licencia es totalmente libre.

3.6. Diseño de la arquitectura

3.6.1. SPA

Single page aplication, de ahora en adelante SPA, es una aplicación o sitio web que cabe en una sóla página web con el objeto de proveer una experiencia de usuario más fluida y una interfaz más enriquecida. Una de las ventajas de este tipo de aplicaciones es que son capaces de actualizar una parte de la interfaz, sin necesidad de enviar o recibir una petición de *full-page*.

3.6.2. Particularidades de JavaScript

En JavaSCript no existen las clases

JavaScript es un lenguaje orientado a objetos basado en prototipos en lugar de clases. A diferencia de los lenguajes orientados a objetos basados en clases, un

lenguaje basado en prototipos no hace distinción entre clases e instancias: Simplemente tiene objetos. Estos objetos son conocidos como objetos prototípicos, que son objetos que se utilizan como una plantilla a partir de la cual se obtiene el conjunto inicial de propiedades de un nuevo objeto[42].

Para poder entender cuáles son las diferencias entre un lenguaje orientado a objetos basados en clases (Java) y basados en prototipos (JavaScript) se han listado sus diferencias en la tabla 3.4.

Basados en clases (Java)	**Basados en prototipos (JavaScript)**
La clase y su instancia son entidades distintas	Todos los objetos pueden heredar de otro objeto
Define una clase en la definición de clase; se instancia una clase con los métodos constructores.	Define y crea un conjunto de objetos con funciones constructoras.
Se crea un objeto con el operador new.	Igual.
Se construye una jerarquía de objetos utilizando la definición de las clases para definir subclases de clases existentes.	Se construye una jerarquía de objetos mediante la asignación de un objeto como el prototipo asociado a una función constructor.
Se heredan propiedades siguiendo la cadena de clases.	Se heredan propiedades siguiendo la cadena de prototipos.
La definición de una clase especifica todas las propiedades de todas las instancias de esa clase. No se pueden añadir propiedades dinámicamente en tiempo de ejecución.	El conjunto inicial de propiedades lo determina la función constructor o el prototipo. Se pueden añadir y quitar propiedades dinámicamente a objetos específicos o a un conjunto de objetos.

Tabla 3.4: Comparativa Java y JavaScript

Por estos motivos, los patrones que se han diseñado con UML consisten en una aproximación de cómo se representaría el diseño de la plataforma.

3.6.3. *Callbacks* en JavaScript

Las *callbacks* JavaScript son utilizadas para gestionar eventos de manera responsable en el lado cliente ejecutando funciones asíncronas, y en Node, las *callbacks* también son utilizadas en el lado servidor para dar servicio a múltiples peticiones simultáneas de clientes.

Una *callback* es una función que es pasada como argumento a otra función, que espera ser invocada tanto inmediátamente como en algún momento futuro.

Las *callbacks* pueden ser vistas como una forma de "*continuation-passing style*" (CPS), cuyo control es pasado explícitamente de forma continuada, en el caso de las *callback* es pasado como argumento representando una continuación.

JavaScript utiliza un modelo *event-driven* con un único hilo de ejecución. Programar con *callbacks* es especialmente útil cuando el llamador no quiere esperar hasta que la llamada se complete. Para conseguirlo, la operación no bloqueante (*non-blocking operation*) es agendada (*scheduled*) como una *callback* y el hilo principal continua su síncrona ejecución. Cuando la operación se completa, un mensaje es encolado en una cola de tareas según la *callback* que lo provee. El bucle de eventos (*event loop*) en JavaScript prioriza que el hilo ejecute la pila de llamadas primero; cuando la pila está vacía, el bucle de eventos desencola un mensaje para la cola de tareas y ejecuta la correspondiente función *callback*. En JavaScript, las *callbacks* pueden llamarse "funciones" o "funciones anónimas". En el proyecto, para gestionar las *callbacks* se han utilizado promises. Las *promise* son una extensión del lenguaje JavaScript [39].

Una *promise* es un *proxy* para un valor no necesariamente conocido en el momento que es creada. Permite asociar manejadores que actuarán asincrónicamente sobre un eventual valor en caso de éxito, o la razón del fallo. Esto permite que métodos asíncronos devuelvan valores como si fueran síncronos: en vez de inmediatamente retornar el valor final, el método asíncrono devuelve una *promise* y suministra su valor en algún momento en el futuro[9]. Al objeto le enganchas las funciones *callback*, en vez de pasar funciones *callback* a una función[11]. Nos permiten mejorar la legibilidad de nuestro código y evitar tener que pasar el contenido de las funciones directamente como argumentos a nuestra llamada. Las *promise* en JavaScript sirven para evitar el "*callback-hell*" que surge de llamar a una función asíncrona en JavaScript.

Implementando patrones de diseño con JavaScript

En la mayoría de lenguajes tradicionales orientados a objetos existen las clases, las interfaces, la herencia, la encapsulación y el polimorfismo. Pero JavaScript, no, puesto que es muy sencillo: Trabaja por medio de callbacks (funciones).

Un objeto es la unidad responsable de proporcionar estados y comportamiento; y la declaración de las funciones en JavaScript proporcionan ambas[**?**].

3.6.4. Patrones de diseño

MVC y MVVM

El patrón Modelo Vista Controlador, de ahora en adelante MVC, es un patrón de arquitectura software que permite separar la representación visual de la información, de la interacción del usuario.

MVC es un patrón *composite*. Los patrones *composite* son aquellos patrones capaces de trabajar juntos para darle solución a un problema en común[**?**].

El patrón de diseño MVC está compuesto por:

- Modelo: Almacena toda la información, estados y lógica de la aplicación.

- Vista: Se trata de a interfaz de usuario que muestra una representación del modelo. La vista informa al controlador qué trata de hacer el usuario.

- Controlador: Toma la entrada del usuario, la gestiona y le transmite lo que ha interpretado al modelo. También, manipula cómo se debe ver la vista.

Actualmente, se han desarrollado nuevos tipos de modelo MV*, como el Modelo Vista VistaModelo, de ahora en adelante MVVM, que está basada en el patrón MVC y el Modelo Vista Presentador, que trata de separar con más claridad el desarrollo de la interfaz de usuario con el de la lógica de negocio y el comportamiento de la aplicación.

Los componentes de los patrones MVVM son:

- Modelo: Representa los datos específicos del dominio o información con la que nuestra aplicación debe trabajar. Almacena información, pero comúnmente no posee ningún tipo de comportamiento. No formatean la información ni influencian en cómo los datos aparecen en el navegador, ya que no es su responsabilidad.

- Vista: Es la única parte de la aplicación con la que el usuario interactúa. Contiene la información obtenida de la sincronización entre la vista y el modelo (*data binding*[30]), los eventos y comportamientos. Es una interfaz de usuario interactiva que representa el estado de la VistaModelo. La vista no es la encargada de gestionar su estado, sólo se mantiene sincronizada con la VistaModelo.

- VistaModelo: Se puede considerar como un Controlador especializado que actúa como un conversor de datos. Transforma la información del Modelo en la información de la Vista[42].

En MEAN STACK, estos dos patrones se pueden aparecen combinados como se muestra en la figura3.15.

Page Controller

Page Controller tiene un único *controller* por cada página lógica de la página web[**?**].

Las responsabilidades del *Page Controller* son:

- Analizar la URL y extraer cualquier dato de formulario necesario para el comportamiento.

- Crear e invocar cualquier objeto del modelo para procesar datos. Cualquier dato relevante de una petición HTML ha de ser pasada al modelo, de esta forma el modelo no necesita ningún tipo de conexión a las peticiones HTML.

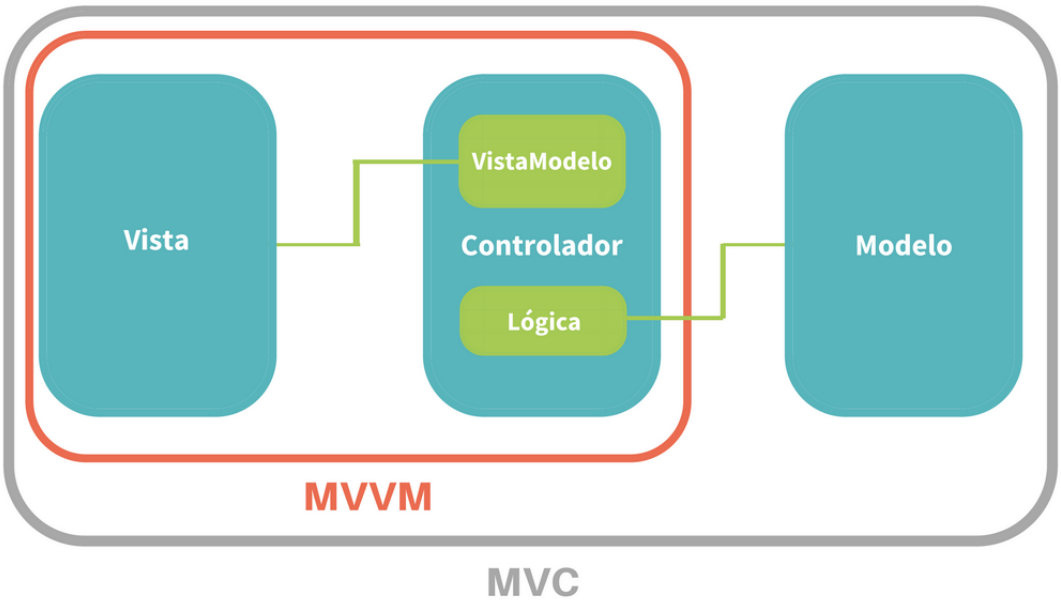


Figura 3.15: Mean STACK - MVC y MVVM

- Determina qué vista ha de ser mostrada a continuación como resultado de la página y proporcionar al modelo información sobre ello.

El patrón *Page Controller* acepta entradas desde la petición de una página, invoca las acciones pedidas en el modelo, y determina la vista que se ha de usar como página resultante[34].

En AngularJS, tenemos *controllers* los cuales tienen unas responsabilidades limitadas: Ellos no aceptan las peticiones del usuario porque es la responsabilidad de los *services* $route o $state y la renderización de la página es responsabilidad de la directiva ng-view. Para determinar qué vista ha de ser mostrada utiliza el *service* $location.

Al igual que *page controllers*, los *controllers* de AngularJS gestionan las interacciones de usuario, proveen y actualizan los modelos. El modelo está unido a la vista a través de $scope[28].

3.6.5. *Observer*

El patrón *Observer* es un patrón de diseño *software* en el cual un objeto, llamado *subject*, mantiene una lista de sus dependencias, llamadas *observers*, y las notifica automáticamente en cualquier cambio de estado, normalmente utilizando uno de sus métodos.

Data-binding

Data binding en Angular es la sincronización entre el modelo y la vista.

El *controller* proporciona comportamiento a través de $scope, mientras que se pueden utilizar en el código HTML para mostrar contenido del modelo en la vista. O la *directive* ng-model para asociar el modelo con la vista.

Cuando la información en el modelo cambia, la vista refleja dicho cambio, y cuando la vista es modificada, el modelo se actualiza.

Debida a la inmediata sincronización entre el modelo y la vista, el controlador puede estar completamente separado de la vista, y simplemente concentrarse en el modelo de datos[30].

El mostrado en la figura 3.16 representa el funcionamiento descrito a continuación. El *module* es el objeto de mayor nivel en Angular. Una aplicación Angular es especificada por la directiva ng-app siempre correspondiente a un módulo. Config es un objeto que configura la aplicación Angular, permite configurar las rutas (*routes*) necesarias en la SPA porque gestionan las *templates*. En Angular se empareja cada *template* con un *controller*. La ¡view¿representa la parte *front-end* y el *controller* y la *factory* representan el *back-end*. $scope vive entre ambos, y Angular permite asegurar que ambos tengan la última versión de la aplicación. El bucle de eventos y $scope permiten a Angular *two way data binding*, y como resultado es que ya no es necesario acceder al DOM desde JavaScript[**?**].

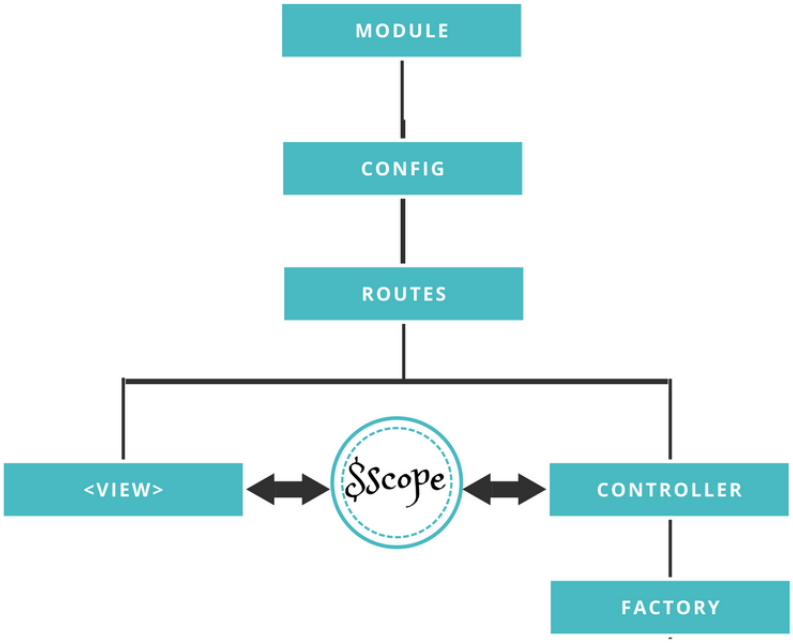


Figura 3.16: Esquema *data-binding*

Dependency Injection

Dependency injection permite obtener una instancia de las clases que se requieren y que una *factory* o *injector* te las provea tanto en tiempo de compilación como en el de ejecución[31].

Dependency injection suele ser utilizado para evitar al patrón *Singleton*[28].

3.6.6. *Factory Method*

Los propósitos del patrón *Factory* son:

- Crear objetos
- Realiza operaciones repetitivas cuando se preparan objetos similares.
- Ofrece que los consumidores de la *Factory* puedan crear objetos sin necesidad de conocer el tipo específico (de la clase) en tiempo de compilación[45].

El patrón *factory* provee una interfaz genérica para crear objetos, donde hay que especificar el tipo de objeto *factory* que deseamos obtener[44].

En AngularJS se hace por medio del *Factory Method*[28].

3.6.7. *Proxy*

En el patrón *proxy*, un objeto actúa como interfaz de otro objeto. El *proxy* se encuentra entre el cliente de un objeto y el objeto en sí, protegiendo el acceso a dicho objeto.

Al tratarse de un *proxy* virtual solventa la problemática que existe si inicializar el sujeto real es costoso, y puede ocurrir que el cliente lo inicialice pero nunca llegue a usarlo. En este caso, el *proxy* puede ayudar siendo la interfaz del sujeto real. El *proxy* recibe la petición de inicialización, pero nunca lo pasa a no ser que realmente el sujeto real sea utilizado.

La imagen 3.17, muestra el posible escenario de cuando un cliente realiza una petición de inicialización y el *proxy* responde que todo está bien, pero realmente no pasa el mensaje, a no ser que sea obvio que el cliente necesite hacer algo con el sujeto. Sólo en ese caso, el *proxy* le pasará ambos mensajes juntos[45].

3.6.8. *Data Mapper*

Un *Data Mapper* es una capa de acceso a datos (*Data Access Layer*) que permite la transferencia bidireccional de datos entre el lugar de almacenamiento de los datos persistentes y la representación en memoria de esos datos, manteniéndolos independientes[28].

En Angular a través del módulo ngResource, nos permite realizar conexiones REST a nuestra API para enviar y recibir información mediante el servicio $http.

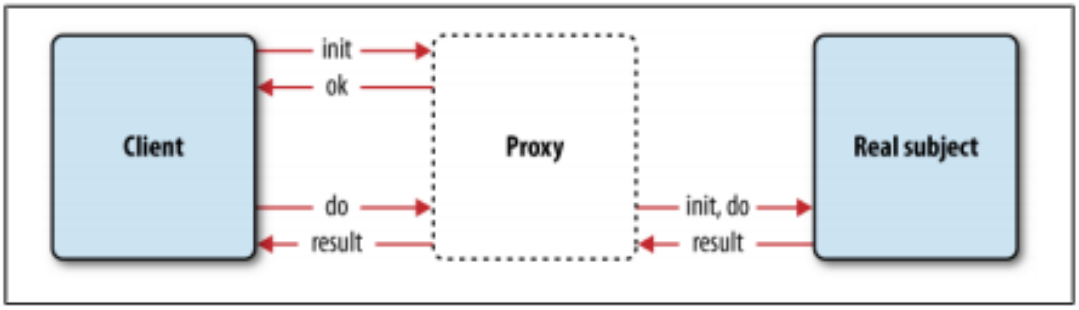


Figura 3.17: *Proxy pattern*[45]

Sin embargo, los datos pasados desde el servidor se encuentran en un formato apropiado gracias a la librería Mongoose.

Mongoose proporciona por un lado el esquema que da formato a los datos obtenidos de la base de datos, y por otro lado, el acceso a los mismos.

3.6.9. API REST

REST (*Representational State Transfer*) es un estilo de arquitectura diseñado para sistemas distribuídos. No está estandarizado pero posee una serie de directivas, como permanecer sin estado, tener relaciones cliente-servidor y una interfaz uniforme. Suele estar relacionado con HTTP.

Sus principios son:

- Expone recursos fácilmente con un directorio de URLs estructurado.
- Representa los data objects y atributos en formato JSON o XML.
- Los mensajes utilizan métodos HTTP explícitamente: GET, POST, PUT y DELETE.
- Protocolo cliente/servidor sin estado: Cada petición HTTP contiene toda la información necesaria para ejecutarla, lo que permite que ni cliente ni servidor necesiten recordar ningún estado previo para satisfacerla[35].

3.6.10. Estructura de directorios

Es importante mantener una estructura de directorios3.18 y de código organizada, pensando en el mantenimiento y en la escalabilidad de la aplicación.

A continuación, se describen (por orden de aparición) los distintos directorios y ficheros que pertenecen a la estructura básica del proyecto:

- app Contiene la parte *frontend* de la aplicación.

- assets Contiene algunos de los recursos de la aplicación.
- images Contiene las imágenes utilizadas en la aplicación.
- javascript Contiene todos los archivos JavaScript de la parte *frontend* de la aplicación.
- app.js Es nuestro fichero de inicio.
- controllers Contiene los *controllers* de los *templates*.
- routes.js Archivo que determina el enrutado de nuestra aplicación.
- factories Contiene los *factories*.
- vendor Contiene algunas librerías JavaScript utilizadas en la aplicación.
- semantic Contiene todos los archivos del paquete semantic-ui.
- styles Contiene algunas librerías CSS utilizadas en la aplicación.
- templates Contiene los *templates* de la aplicación.
- views Contiene la view de nuestra aplicación.
- app.js Archivo de configuración del arranque del servidor.
- node_modules Contiene todos los paquetes Node.js utilizados en la aplicación.
- npm-shrinkwrap.json Archivo que mantiene un registro de versiones de los paquetes.
- package.json Archivo que mantiene una lista de los paquetes de los que depende la aplicación.
- README.md Archivo que describe brevemente la aplicación.
- semantic.json Archivo que contiene configuraciones de compilación para Gulp.
- server Contiene la parte *backend* de la aplicación.
- config Contiene ficheros de configuración de la aplicación.
- expressConfig.js Archivo que contiene configuración de Express.js.
- models Contiene los models de la aplicación.
- routes Contiene las rutas API REST de la aplicación.
- routes.js Archivo que configura los módulos y enrutamiento de la aplicación.

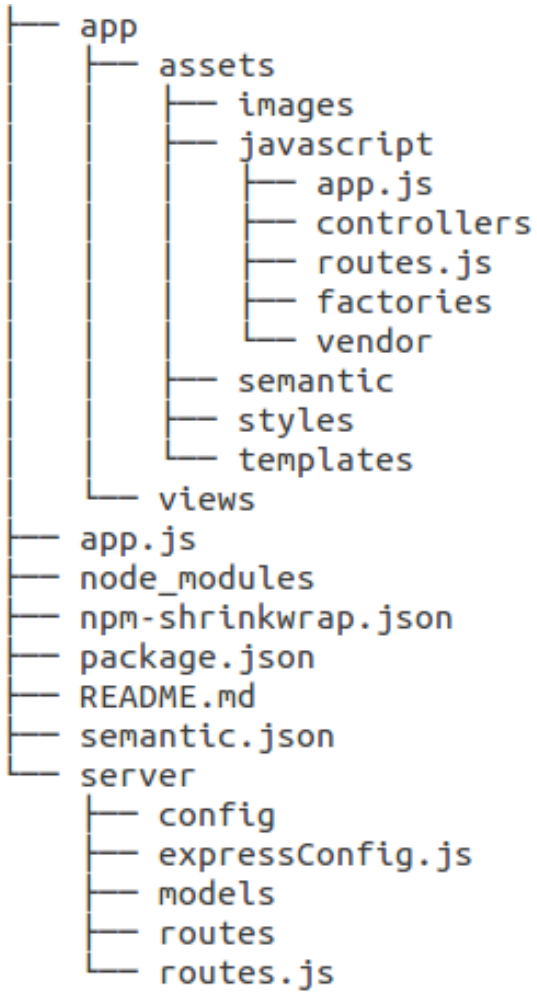


Figura 3.18: Estructura de directorios

3.7. Diagramas

Para una mayor comprensión de la composición del sistema, se han desarrollado diferentes tipos de diagramas.

En la figura 3.19 se muestra cómo interactúa el sistema a grandes rasgos a través del patrón MVC-MVVM. No se trata de un diagrama de clases, pero sirve para tener visión general de los distintos elementos que componen el sistema. En amarillo, aparecen elementos de Angular utilizados y se podrían interpretar como los patrones (en azul) que se indican a su lado.

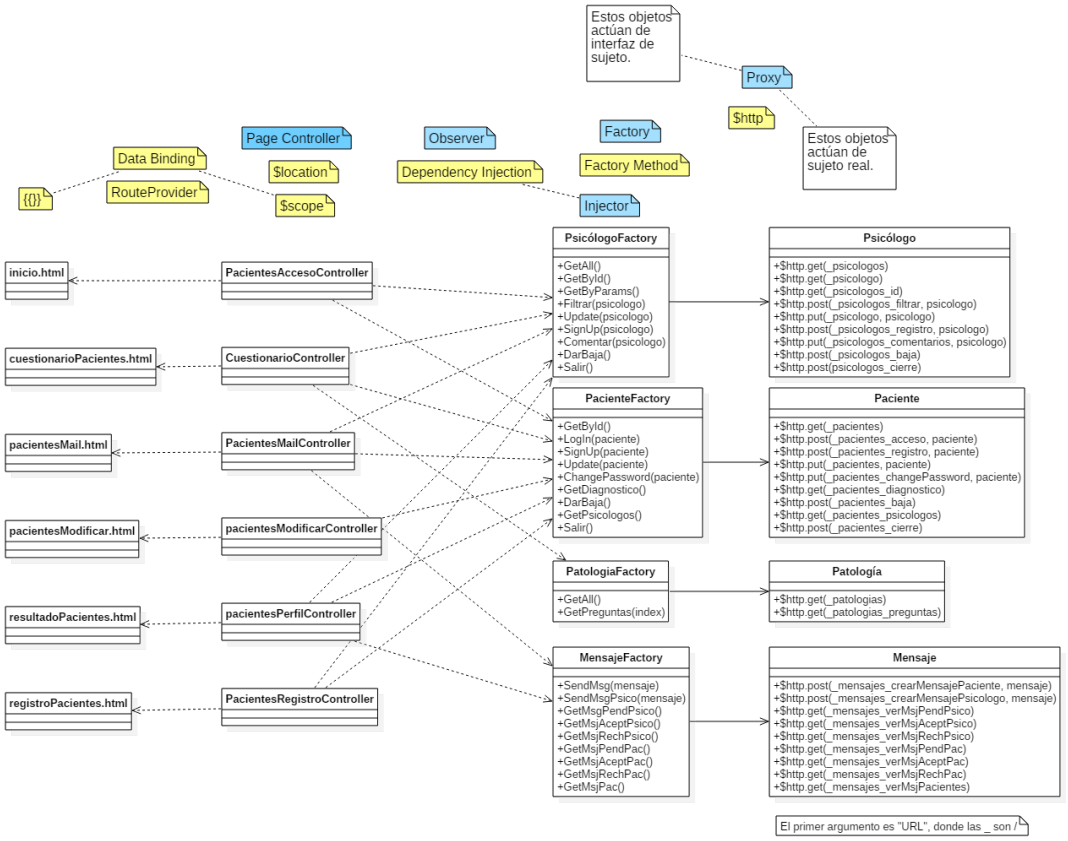


Figura 3.19: Patrón MVC-MVVM

En la figura 3.20, se muestra cómo se implementa la capa de datos *Data mapper* con ayuda de los *schemas* y funciones propias de la librería *Moongoose*.

Para ejemplificar las interacciones entre los distintos componentes del sistema, se han creado dos diagramas de secuencia: Uno referente al caso de uso CU-003 Realizar cuestionario3.21 y otro del CU-005 Contactar psicólogo3.22.

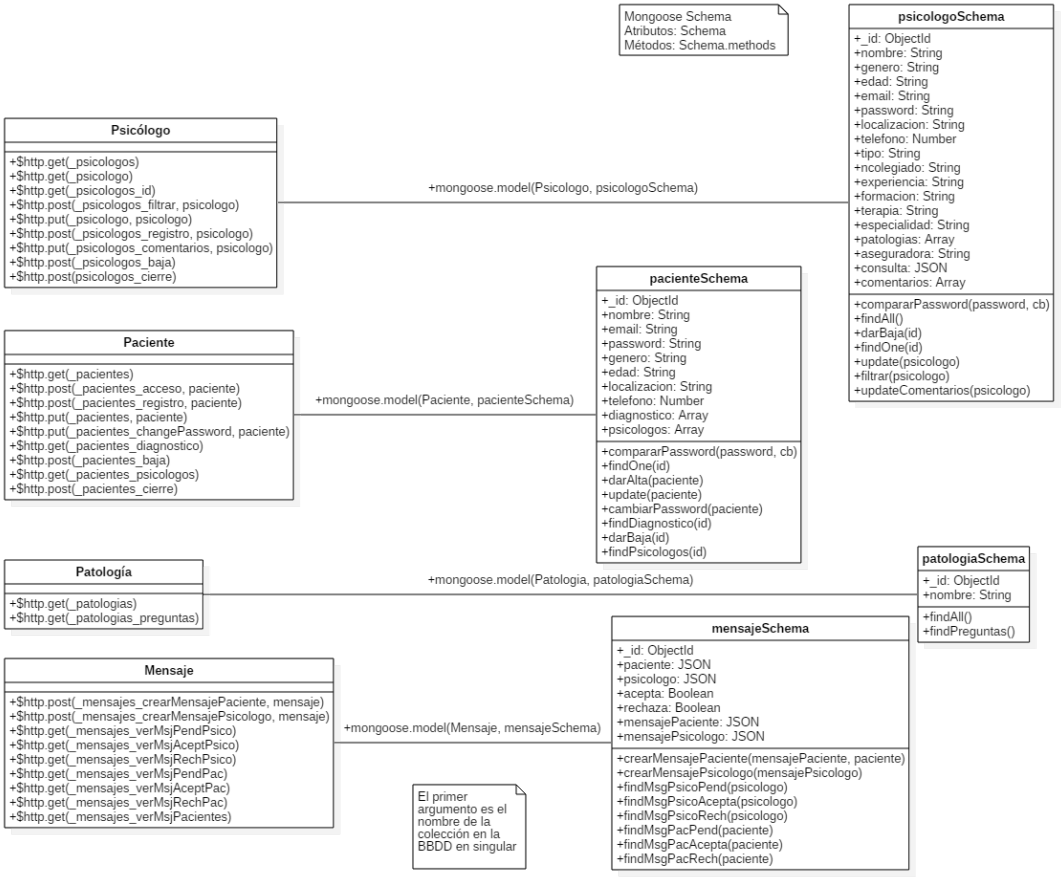


Figura 3.20: Data mapper

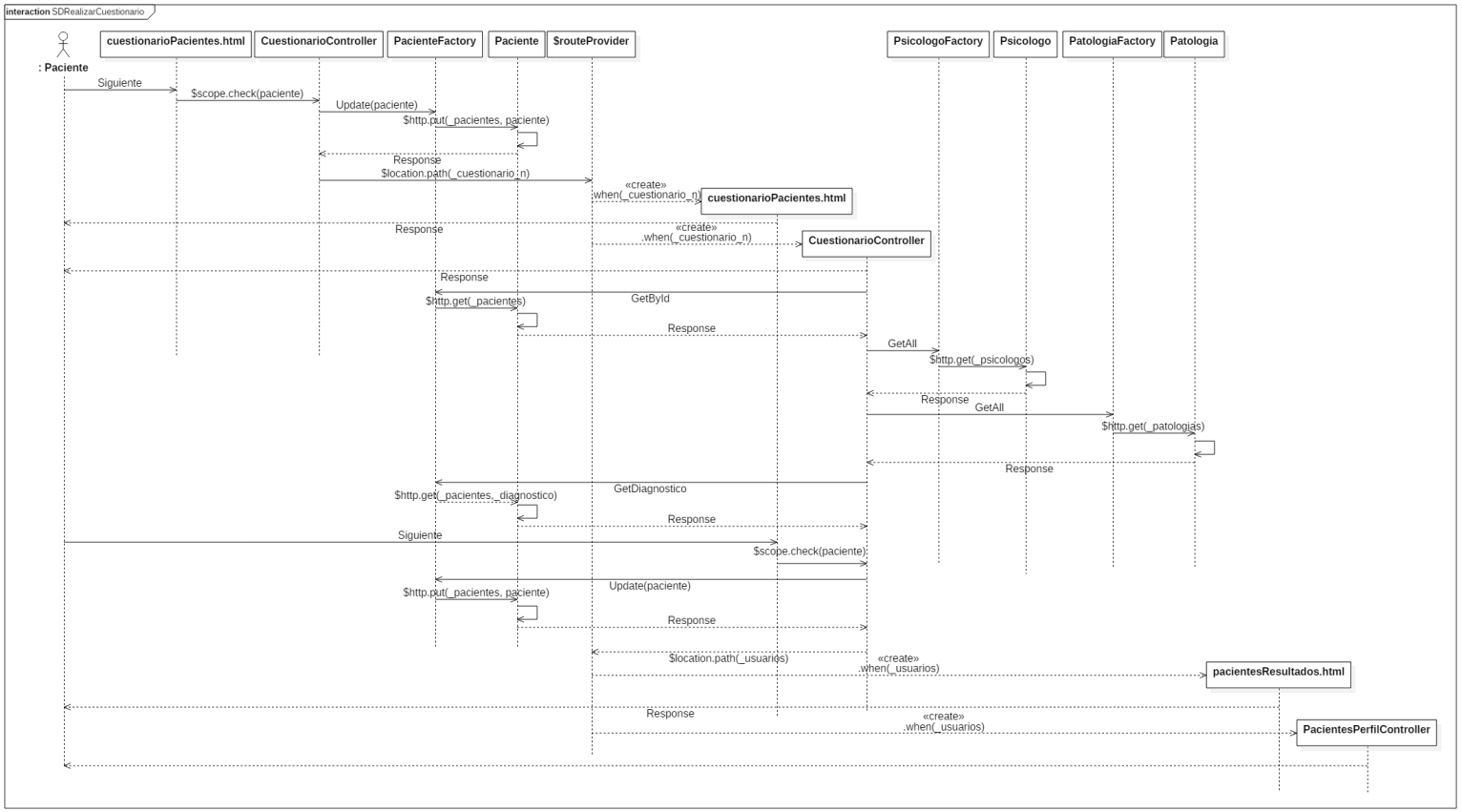


Figura 3.21: Diagrama de secuencia del CU-003

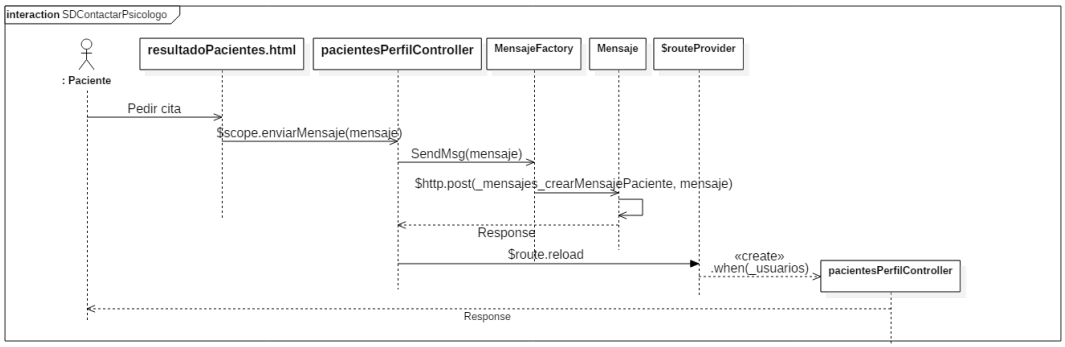


Figura 3.22: Diagrama de secuencia del CU-005

Capítulo 4

Implementación

En las secciones siguientes se mencionarán los aspectos más relevantes de la implementación.

4.1. Implementación del primer incremento

El primer incremento fue la primera toma de contacto con la tecnología utilizada tras la formación.

Los requisitos implementados en el primer incremento son los asociados a la parte del cuestionario de emparejamiento:

- RF-005: Emparejamiento
- RF-006: Filtrado de los resultados en base a distintos criterios
- RNF-002: Tiempo de respuesta de asignación

Las tareas más trascendentales realizadas en este incremento son las descritas a continuación:

Configuración Angular Todos los archivos Angular utilizados deben importarse en la *root view* (app/views/index.html en el proyecto): Librerías JavaScript, app.js, route.js, controllers y factories.

Para indicar a la aplicación que se trata de una aplicación Angular se debe especificar en la etiqueta ¡html¿de la *root view* (app/views/index.html) de la siguiente forma:

```
1 <html ng-app="Emozio"> </html>
```

Se define el módulo principal de la aplicación en el archivo app/assets/javascript/app.js:

```
1 angular.module('Emozio', ['ngRoute', 'ngResource']);
```

Entre corchetes se encuentran las dependencias que tiene nuestro módulo Emozio:

- `ngRoute`: Permite a la aplicación convertirse en una SPA, permitiendo la navegación ente distintas páginas sin necesidad de recargar. W3 angular-routing
- `ngResource`: Permite crear objetos para interactuar con los datos RESTful del lado servidor. El objeto devuelto tiene métodos de acción que proporcionan comportamiento sin necesidad de interactuar con el servicio $http a bajo nivel. Las acciones por defecto son:

```
1 { 'get':    {method:'GET'},
2   'save':   {method:'POST'},
3   'query':  {method:'GET', isArray:true},
4   'remove': {method:'DELETE'},
5   'delete': {method:'DELETE'} };
```

Llamar a estos métodos invoca a $http con el método http especificado, destino y parámetros.

Para indicar al $routeProvider dónde mostrar las templates se utiliza la siguiente directiva: `<div ui-view></div>`

Template	***Controller***
pacientes/cuestionarioPacientes.html	pacientes/cuestionarioController.js
pacientes/pacientesResultados.html	pacientes/perfilController.js

Implementación de las *templates* y *controllers* de los requisitos. Éstos se encuentran dentro del directorio app/assets.

Creación de las *factories*, *routes* y *models* de: Paciente, Psicologo y Patologia.

```
1 var MONGO_URL = 'mongodb://localhost:27017/emozio';
```

Se define la URL de la base de datos a la que se conecta la app.

```
1   var options = {
2     useMongoClient: true,
3     autoIndex: false, // No crear index
4     reconnectTries: Number.MAX_VALUE, // Nunca para de reintentar
          conectarse
5     reconnectInterval: 500, // Reconexion cada 500ms
6     poolSize: 10, // Mantener una conexion de 10 sockets
7     // Si no se conecta, devuelve un error inmediatamente antes
         de tratar de reconectarse
8     bufferMaxEntries: 0
9   };
```

Se establecen las opciones de conexión.

```
1 mongoose.Promise = global.Promise;
```

Se declara que las *promise* que va a utilizar Mongoose son las globales proporcionadas por Bluebird. En la base de datos utilizamos las *promise* definidas e implementadas en la librería Bluebird.

```
1 mongoose.connect(MONGO_URL, options, function(err, res) {
2   if(err) {
3     console.log('ERROR: Reconectando a la BBDD. ' + err);
4   }else{
5     console.log("Conectado a la BBDD");
6   }
7 });
```

Función de conexión de Mongoose a la base de datos con las opciones especificadas.

```
1   var db = mongoose.connection;
2   /* Si sucede un error, mostrarlo */
3   db.on('error', console.error.bind(console, 'Error de conexion:'
        ));
4   db.once('open', function() {
5     console.log("Con exito");
6   });
```

Se abren las conexiones a la base de datos.

```
1 app.use("/", express.static("app/"));
```

Con express.static se especifican los directorios donde se encuentran los archivos que Express va a leer. Facilita el acceso a los *assets* (bienes) de la carpeta app desde el servidor. Lo que nos permite tener la parte cliente separada del servidor.

```
1 app.set('views', __dirname + '/../app/views');
```

Define que las routes a las templates se rendericen con el render method dentro del directorio views.

```
app.use(bodyParser.urlencoded({ extended: true }));
```

Especifica que los datos recogidos de un formulario se pasen a través del método *post*.

```
app.use(bodyParser.json());
```

Mediante el paquete Body-parser, podemos tratar los objetos en formato JSON sin necesidad de manipularlos, o cambiar su tipo.

```
var app = express();
```

Se inicializa el servidor express.

```
  app.get('/', function(req, res){
    res.sendfile('index.html', {root: app.settings.views});
  });
```

Express establece cuál va a ser la *view* raíz (*root*) enviada tras el inicio del servidor.

Además, se importan los archivos de server/routes que sirven como *endpoint* (punto medio) entre la parte cliente y servidor, de esta forma durante la implementación, logramos mantener la parte del *frontend* independiente del *backend* haciéndola funcional.

4.2. Implementación del segundo incremento

Especificación del arranque del servidor y se establece cuál es la vista raíz en el archivo routes.js. También, se vinculan todos los archivos que va utilizar el servidor: Archivos de configuración, de conexiones al modelo... Los requisitos implementados en el segundo incremento son los asociados a la parte de la gestión de usuarios:

- RF-001: Acceso usuarios
- RF-002: Registro
- RF-003: Baja
- RF-004: Modificación de los datos
- RNF-001: Encriptado de datos

Template	*Controller*
inicio.html	pacientes/pacientesAccesoController.js
pacientes/registroPacientes.html	pacientes/pacientesRegistroController.js
pacientes/pacientesModificar.html	pacientes/pacientesModificarController.js
psicologos/psicologosModificar.html	psicologos/psicologosModificarController.js
psicologos/registroPsicologo.html	psicologos/psicologosRegistroController.js

Gestión de sesiones de usuario con PassportJS y Express-session. *La configuración de PassportJS se realizó en el directorio server/config/passport.js*

Las credenciales utilizadas para autenticar a un usuario sólo son transmitidas durante la petición de acceso (*login request*). Si la autenticación se realiza con éxito, la sesión será establecida y mantenida vía una cookie en el navegador del usuario. Cualquier petición posterior no contendrá las credenciales, únicamente la *cookie* que identifica la sesión. Para poder gestionar las sesiones de acceso (*login sessions*), Passport *serialize* y *deserialize* instancias de usuario.

```
1 passport.serializeUser(function(usuarios, done){
2   done(null, usuarios._id);
3 })
```

Sólo el ID de usuario es "creado" en la sesión, manteniendo mínima la cantidad de datos guardados. Cuando las siguientes peticiones sean recibidas, este ID será el utilizado para encontrar al usuario, el cual fue guardado en `req.user`.

```
1  passport.deserializeUser(function(id, done){
2    Paciente.findById(id, function(error, usuario){
3      if(usuario!=null){
4        done(null, usuario);
5      } else {
6        Psicologo.findById(id, function(error, usuario){
7          done(null, usuario);
8        });
9      }
10   });
11 })
```

`deserializeUser()` es invocado en cada petición por `passport.session`. Permite cargar información adicional a la información de usuario en cada petición; este objeto está asociado a la petición como `req.user` haciéndolo accesible en la gestión de peticiones.

```
1 passport.use(new LocalStrategy(
2   {
3     usernameField: 'email',
4     passwordField: 'password'
5   },
6   function (username, password, done) {
```

```
7     Paciente.findOne({email: username}, function(error, paciente)
          {
8       if(!paciente){
9         //                return done(null, false, {message: '
              Este email: '+email+'no esta registrado'});
10        Psicologo.findOne({email: username}, function(error,
              psicologo){
11          if(!psicologo) {
12            return done(null, false, {message: 'Este email: '+
                  username+'no esta registrado'});
13          } else {
14            psicologo.compararPassword(password, function(error,
                  sonIguales){
15              if(sonIguales){
16                return done(null, psicologo);
17              } else {
18                return done(null, false, {message: 'La contrasena
                      no es valida'});
19              }
20            });
21          }
22        });
23      } else {
24        paciente.compararPassword(password, function(error,
              sonIguales){
25          if(sonIguales){
26            return done(null, paciente);
27          } else {
28            return done(null, false, {message: 'La contrasena no
                  es valida'});
29          }
30        });
31      }
32    });
33  }
34 ));
```

Para poder utilizar la autenticación por `username` y `password`, Passport utiliza el mecanismo proporcionado por su módulo passport-local.

- `UsernameField` y `PassworfField` son los obtenidos del cuerpo de la petición (`req.body`) recibida cuando un usuario quiere acceder.
- Se busca al paciente que posea esos datos; y se pueden dar dos casos:
  - Si no existe, se busca al psicólogo que posea esos datos:
    - Si no existe: El usuario no está registrado.
    - Si existe: Se comprueba que la contraseña sea la misma a la introducida. Si son iguales, el acceso es correcto. Si no, la contraseña no es válida.

- Si existe: Se comprueba que la contraseña sea la misma a la introducida. Si son iguales, el acceso es correcto. Si no, la contraseña no es válida.

```
exports.estaAutenticado = function (req, res, next){
  if(req.isAuthenticated()){
    return next();
  }
  req.session.error = 'Please sign in!';
  res.redirect('/');
}
```

Función que comprueba si el usuario que está realizando una petición, está autenticado.

En la route de Pacientes del directorio server/routes/paciente.js:

```
app.route('/pacientes/acceso')
    .post(function(req, res, next){
    setTimeout(function(){
      passport.authenticate('local', function(error, paciente,
          info){
        if(error){
          return next(error);
        }
        if(!paciente) {
          return null;
        }else{
          req.login(paciente, {}, function(err) {
            if (err) {
              return null;
            };
            return res.json(paciente);
          });
        }
      })(req, res, next); //Funcion que devuelve passport y que
          debe ser invocada de esta forma
    }, 50);
  });
```

El formulario de acceso es enviado por el servidor a través del método POST. Utilizando `authenticate()` es como gestionamos la petición de acceso: En el caso de que el paciente exista, hacemos el `login`.

```
  app.use(session({
    /* Se utiliza para firmar el ID de sesion de la cookie*/
    secret: 'ESTO ES SECRETO',
    /* Por cada llamada realizada al servidor, la sesion se
       guardara en la BBDD */
    resave: true,
```

```
6     /* Cuando se realiza la llamada por primera vez, guarda un
          objeto vacio con informacion de esa session */
7     saveUninitialized: true,
8     store: new MongoStore({
9       url: MONGO_URL,
10      /* Si sucede un error, trata de volver a conectarse */
11      autoReconnect: true
12    })
13  }));
```

Se establecen las opciones de la sesión que será utilizada en la aplicación. Cada sesión es guardada en la base de datos a modo de registro, aunque por el momento no es un funcionamiento relevante para nuestra aplicación.

```
1   app.use(passport.initialize());
```

Se inicializa PassportJS.

```
1   app.use(passport.session());
```

Se determina que passport utilice las sesiones.

Encriptación bcrypt de las contraseñas En la aplicación, el paquete bcrypt es utilizado para encriptar las contraseñas de usuario.

Uno de los casos donde es necesario utilizar bcrypt es cuando es necesaria la comparación de contraseñas en el inicio de sesión. Se hace por medio de la siguiente función:

```
1   /* Comprobar una contrasena con su hash */
2   bcrypt.compare(password, this.password, function(error,
        sonIguales){
3     if(error){
4       return cb(error);
5     }
6     cb(null, sonIguales);
7   })
```

Compara la contraseña con la que está tratando de acceder el usuario después de cifrarla con el *hash* que existe actualmente en la base de datos. El *hash* que se encuentra en la base de datos tiene almacenados el coste, la sal y la contraseña, por lo que bcrypt sabe cómo cifrar la nueva contraseña introducida.

Un ejemplo práctico: El *hash* introducido en la base de datos podría ser el siguiente: `$2a$10$vI8aWBnW3fID.ZQ4/zo1G.q1lRps.9cGLcZEiGDMVr5yUP1KUOYTa`

Este *hash* contiene tres cadenas concatenadas con el símbolo "$":

- `2a` identifica la versión del algoritmo bcrypt utilizado.
- `10` es el factor de coste: Se han utilizado 2^{10} iteraciones de la función de derivación de la key.

- `vI8aWBnW3fID.ZQ4/zo1G.q1lRps.9cGLcZEiGDMVr5yUP1KUOYTOa` son la sal y la contraseña cifrados, concatenados y codificados en Base64. Los 22 primeros caracteres codificados en un valor para la sal de 16-byte. El resto de caracteres son la contraseña cifrada que va a ser comparada durante la autenticación.

Otro de los casos donde es necesario utilizar bcrypt es al dar de alta a un usuario.

```
1  bcrypt.genSalt(10, function(error, salt){
2    if(error) {
3      console.log("Error en la sal");
4    }
5    bcrypt.hash(paciente.password, salt, null, function(error,
        hash){
6      if(error) {
7      console.log("Error en el hash");
8      }
9    });
10 });
```

Primero se genera la sal con un factor de coste 10, y después, se crea el *hash* con la contraseña que ha introducido el paciente y la sal. Este *hash* será almacenado en la base de datos.

Google Maps JavaScript API para el autocompletado de la localización en los formularios de registro y modificación. Para autocompletar las localizaciones introducidas por el usuario tanto en el formulario de registro como en el de modificación, se ha utilizado el autocompletado para direcciones y términos búsqueda de Google Maps JavaScript API.

Por ejemplo, en el formulario de registro de usuarios de la *template* pacientes/registroPacientes.html, fue incorporado al código HTML de la siguiente forma:

```
1  <div class="field" id="locationField">
2    <div class="ui left icon input">
3      <i class="marker icon"></i>
4      <input id="autocomplete" name="localizacion" placeholder="
          Localizacion" ng-focus="geolocate()" type="text"
          required>
5    </div>
6  </div>
```

Donde ng-focus ng-focus es una directiva que indica que cuando el input esté señalado, se ejecute la función `geolocate()`.

La función `geolocate` se encuentra en pacientes/pacientesRegistroController.js y viene definida por Google de la siguiente forma:

```
1  $scope.geolocate = function() {
2    if (navigator.geolocation) {
3      navigator.geolocation.getCurrentPosition(function(position)
           {
4        var geolocation = {
5          lat: position.coords.latitude,
6          lng: position.coords.longitude
7        };
8      });
9    }
10 }
```

Esta función, simplemente toma la ubicación donde se sitúa el usuario en ese momento. $scope se utiliza para pasar la función del *controller* a la *template*.

Por otra parte, la página se encuentra contenida bajo la siguiente etiqueta:

```
1 <div class='container' ng-init="initAutocomplete()">...</div>
```

La directiva ng-init ng-init evalúa `initAutocomplete()` al inicializar la aplicación.

La función `initAutocomplete` se encuentra en el *controller*.

```
1  $scope.initAutocomplete = function() {
2    autocomplete = new google.maps.places.Autocomplete(
3      (document.getElementById('autocomplete')),
4      { types: ['(cities)'], /* Se buscaran ciudades */
5       componentRestrictions: {country: "es"}}); /* Restringidas
           dentro de Espana */
6
7    /* Cuando el usuario selecciona una opcion del desplegable,
        se ejecuta la funcion indicada */
8    autocomplete.addListener('place_changed', fillInAddress);
9
10   /* Funcion que recupera el lugar escogido en el
        autocompletado */
11   function fillInAddress() {
12     /* Toma los detalles del lugar del objeto de autocompletado
          */
13     place = autocomplete.getPlace();
14   }
15 }
```

`google.maps.places.Autocomplete` crea un objeto tomando como argumentos el `input` donde se quiere autocompletar y una serie de opciones. Para la aplicación, se han restringido las opciones a ciudades españolas.

Nodemailer.js Cuando un psicólogo cubre el formulario de registro, sus datos son mandados al correo electrónico de Emozio para poder valorarlos antes de formar parte de nuestra base de datos. Para enviar los e-mails se ha utilizado Nodemailer. Algunas particularidades son descritas a continuación.

```
var transporter = nodemailer.createTransport({
  service: 'gmail',
  auth: {
    user: 'emozio.info@gmail.com',
    pass: '********************'
  }
});
```

Se crea un objeto transporter utilizado en el trasporte por defecto SMTP. SMTP es el transporte utilizado en Nodemailer para el envío de mensajes, pero también, es el protocolo utilizado por diferentes *host* de *email*.

```
var mailOptions = {
  from: 'emozio.info@gmail.com',
  to: 'emozio.info@gmail.com',
  subject: 'Emozio Web - Nuevo psicologo',
  generateTextFromHTML: true,
  html: /*Aqui iria el codigo HTML */
};
```

Se especifican las opciones de *email* que se quieren enviar.

```
transporter.sendMail(mailOptions, (error, info) => {
  if (error) {
    return console.log(error);
  }
  console.log('Message sent: %s', info.messageId);
});
```

Se envía el *email* con la función `sendMail` a través del `transporter`.

Validación de formularios con SemanticUI Para la validación de formularios se utilizó SemanticUI que funciona de la siguiente forma:

En la *template*, se han de tener todos los campos de formulario corréctamente nombrados y añadir un identificador a la etiqueta `<form>`.

En el *controller*, existe un objeto especial al que se le pueden pasar la lista de elementos del formulario a validar, las reglas que ha de cumplir cada campo y los mensajes de error en cada caso.

En el código de la aplicación, un ejemplo podría ser:

```
$('#access_form').form({
  on : 'blur', /* Cada elemento se evalua por separado */
  inline: 'false', /* Los mensajes de validacion no se disponen
      en linea */
  /* Campos a validar: Identificador y reglas a evaluar */
  fields : {
    email : {
      identifier : 'email',
      rules : [
        {
```

```
            type : 'empty',
            prompt : 'Por favor, introduzca un e-mail.'
          },
          {
            type: 'email',
            prompt: 'El formato del e-mail es incorrecto.'
          },
          {
            type: 'maxLength[50]',
            prompt: 'Demasiados caracteres.'
          }
        ]
      },
      password: {
        identifier: 'password',
        rules: [
          {
            type: 'empty',
            prompt: 'Por favor, introduzca una contrasena.'
          },
          {
            type: 'maxLength[50]',
            prompt: 'Demasiados caracteres.'
          }
        ]
      }
    }
  });
```

Los campos a validar en el formulario de acceso a la plataforma son `email` y `password`.

Para `email`, las reglas de validación son:

- El campo no debe estar vacío.
- El formato debe ser de tipo `email`.
- La longitud máxima de carácteres permitidos es 50.

Para `password`, las reglas de validación son:

- El campo no debe estar vacío.
- La longitud máxima de carácteres permitidos es 50.

Para evaluar que se cumplen las condiciones especificadas, se utiliza la siguiente función:

```
$('#access_form').form('is valid');
```

4.3. Implementación del tercer incremento

Los requisitos implementados en el segundo incremento son los asociados a la parte de la comunicación:

- RF-007: Contacto del paciente con el psicólogo
- RF-008: Valoración del psicólogo por parte del paciente

Además, aunque en primera instancia no estuviese contemplado, surgió la idea de gestionar la comunicación como un sistema de citas. Por ello, se incorporó a posteriori un calendario mensual donde el psicólogo pudiese ver las citas que tiene agendadas con una vista por mes, por semana y por día.

Las tareas más trascendentales realizadas en este incremento son las descritas a continuación:

Template	***Controller***
pacientes/pacientesMail.html	pacientes/pacientesMailController.js
psicologos/psicologosMail.html	psicologos/psicologosMailController.js
psicologos/psicologoCalendario.html	psicologos/psicologosCalendarioController.js

Implementación de las *templates* y *controllers* de los requisitos. Éstos se encuentran dentro del directorio app/assets.

Creación de la *factories*, *routes* y *models* de Mensaje.

Visualización de las citas agendadas por medio de FullCalendar En la *template* psicologoCalendario.html, el calendario es dispuesto de la siguiente forma:

```
1 <div id="calendar" ui-calendar ng-model="eventSources"></div>
```

Este elemento es gestionado por el psicologosCalendarioController.js.

```
1  $('#calendar').fullCalendar({
2    header: { /* Opciones de la barra de herramientas */
3        left: 'prev,next today', /* A la izquierda: Anterior,
             siguiente y hoy */
4        center: 'title', /* En el centro: Nombre del mes */
5        right: 'month,basicWeek,basicDay' /* A la derecha: Mes,
             semana y dia */
6      },
7      buttonText : { /* Nombre que aparece en las opciones de la
          barra de herramientas */
8        today:    'hoy',
```

```
9         month:     'mes',
10        week:      'semana',
11        day:       'dia',
12        prev:    '<',
13        next:    '>'
14      },
15      firstDay : 1, /* Empieza el lunes */
16      dayNames : ['Domingo', 'Lunes', 'Martes', 'Miercoles', '
            Jueves', 'Viernes', 'Sabado'], /* Nombres de los dias */
17      dayNamesShort : ['Dom.', 'Lun.', 'Mart.', 'Mierc.', 'Juev.'
            , 'Vier.', 'Sab.'], /* Nombres abreviados de los dias */
18      monthNames: ['Enero','Febrero','Marzo','Abril','Mayo','
            Junio','Julio','Agosto','Septiembre','Octubre','
            Noviembre','Diciembre'], /* Nombres de los meses */
19      monthNamesShort: ['Ene','Feb','Mar','Abr','May','Jun','Jul'
            ,'Ago','Sep','Oct','Nov','Dic'], /* Nombres abreviados
            de los meses */
20      navLinks: true, /* Permite navegar entre las distintas
            vistas: Mes, semana y dia */
21      editable: true, /* Los eventos pueden ser modificados */
22      eventLimit: true, /* Numero de eventos mostrados en un dia
            limitado */
23      theme: true, /* Activa el tema */
24      themeSystem:'bootstrap3', /* Establece el tipo de tema */
25      eventColor: '#ffffff', /* Establece el color del fondo de
            los eventos */
26      eventTextColor: '#008080', /* Establece el color del texto
            de los eventos */
27      eventClick: function (calEvent) { . . . },
28
29      height: 600 /* Establece la altura del calendario */
30    });
```

Para meter los eventos en el calendario se utiliza la función `fullcalendar` con el argumento `renderEvent`.

```
1 $('#calendar').fullCalendar('renderEvent', cita, true);
```

- Se inicializa el calendario.
- Opciones mostradas en los botones del calendario.
- Traducción de los mensajes de los botones (que traen por defecto) al castellano.
- Indica el día donde comienza la semana.
- Traducción de los días, abreviaturas de los días, meses y abreviaturas de los meses al castellano.
- Función que se ejecuta cuando se hace *click* en un evento y lo muestra.

Capítulo 5

Pruebas

5.1. Pruebas de contenido

Se evalúa el diseño del contenido buscando errores de tipo:

- **Sintáctico**

  En este grupo se encuentran errores de estructura de las oraciones, faltas de ortografía, vocabulario, puntuación y gramática. Estos se pueden tratar con ayuda de un corrector ortográfico[**?**].

- **Semántico**

  Los errores se producen por falta de sentido en las oraciones, inconsistencias, incorrecciones e ambigüedades.

  La manera de tratar este tipo de errores podría ser mediante la formulación de las siguientes preguntas:

  - ¿La información es realmente precisa?
  - ¿La información es concisa y puntual?
  - ¿La disposición del contenido es fácil de comprender para el usuario?
  - ¿La información situada dentro de un objeto de contenido puede encontrarse con facilidad?
  - ¿Se proporcionaron referencias adecuadas para toda la información derivada de otras fuentes?
  - ¿La información presentada es consistente con la información presentada en otros objetos de contenido?
  - ¿El contenido no es ofensivo, confuso o abre la puerta a demandas?
  - ¿El contenido no infringe derechos de autor o nombres comerciales existentes?

- ¿El contenido incluye vínculos internos que complementan el contenido existente? ¿Los vínculos son correctos?
- ¿El estilo estético del contenido no entra en conflicto con el estilo estético de la interfaz?

- **De organización o estructura de contenido**

  El error consiste en que los objetos existentes dentro del contenido no se corresponden con los que dicen ser. Por ejemplo, el objeto x muestra la foto del objeto y. Para identificarlos se debe revisar detenidamente la página.

  En nuestro caso, nuestra plataforma web se encuentra conectada a una base de datos generando objetos de contenido dinámico creados en tiempo real, por lo que se deben hacer pruebas ejecutables para descubrir errores de contenido rastreables derivados de las consultas realizadas. Por tanto, se hará una prueba de cada consulta cliente-servidor a la base de datos que deberán cumplir que:

  - La información pasada entre cliente y servidor es válida.
  - Se formatean correctamente los datos recogidos del usuario.
  - Se formatean correctamente los datos mostrados al usuario.

5.1.1. Incremento 1

Se evaluará cada tipo de error por página. Las pruebas de contenido del incremento 1 son PC-001, PC-002, PC-003, PC-004, PC-005 y PC-006.

PC-001	**Se comprueba que no existen errores sintácticos en los textos de la página con ayuda de un corrector ortográfico.**	
Tipo de error	Sintáctico	
Evaluación por página	inicio.html	Correcta
	pacientesResultados.html	Correcta
	cuestionarioPacientes.html	Correcta

Tabla 5.1: PC-001

5.1.2. Incremento 2

Se evaluará cada tipo de error por página. Las pruebas de contenido del incremento 2 son PC-007, PC-008, PC-009, PC-010, PC-011 y PC-012.

Se evaluará cada tipo de error por página:

PC-002	**Se comprueba que no existen errores semánticos en los textos de la página mediante la revisión y respuesta de las preguntas en cuestión.**	
Tipo de error	Semántico	
Evaluación por página	inicio.html	Correcta
	pacientesResultados.html	Correcta
	cuestionarioPacientes.html	Correcta

Tabla 5.2: PC-002

PC-003	**Se comprueba que la estructura y contenido de los objetos existentes en la página es correcta mediante revisión manual.**	
Tipo de error	De organización o estructura de contenido	
Evaluación por página	inicio.html	Correcta
	pacientesResultados.html	Correcta
	cuestionarioPacientes.html	Correcta

Tabla 5.3: PC-003

PC-004	**Se comprueba que el contenido obtenido de la base de datos cumple que la información pasada entre cliente y servidor es válida.**	
Tipo de error	Del contenido obtenido de la base de datos	
Evaluación por página	pacientesResultados.html	Correcta
	cuestionarioPacientes.html	Correcta

Tabla 5.4: PC-004

PC-005	**Se comprueba que el contenido obtenido de la base de datos cumple que se formatean correctamente los datos recogidos del usuario.**	
Tipo de error	Del contenido obtenido de la base de datos	
Evaluación por página	pacientesResultados.html	Correcta
	cuestionarioPacientes.html	Correcta

Tabla 5.5: PC-005

PC-006	**Se comprueba que el contenido obtenido de la base de datos cumple que se formatean correctamente los datos mostrados al usuario.**	
Tipo de error	Del contenido obtenido de la base de datos	
Evaluación por página	pacientesResultados.html	Correcta
	cuestionarioPacientes.html	Correcta

Tabla 5.6: PC-006

PC-007	**Se comprueba que no existen errores sintácticos en los textos de la página con ayuda de un corrector ortográfico.**	
Tipo de error	Sintáctico	
Evaluación por página	pacientesModificar.html	Correcto
	registroPacientes.html	Correcto
	psicologoPerfil.html	Correcto
	psicologosModificar.html	Correcto
	registroPsicologo.html	Correcto

Tabla 5.7: PC-007

PC-008	**Se comprueba que no existen errores semánticos en los textos de la página mediante la revisión y respuesta de las preguntas en cuestión.**	
Tipo de error	Semántico	
Evaluación por página	pacientesModificar.html	Correcto
	registroPacientes.html	Correcto
	psicologoPerfil.html	Correcto
	psicologosModificar.html	Correcto
	registroPsicologo.html	Correcto

Tabla 5.8: PC-008

PC-009	**Se comprueba que la estructura y contenido de los objetos existentes en la página es correcta mediante revisión manual.**	
Tipo de error	De organización o estructura de contenido	
Evaluación por página	pacientesModificar.html	Correcto
	registroPacientes.html	Correcto
	psicologoPerfil.html	Correcto
	psicologosModificar.html	Correcto
	registroPsicologo.html	Correcto

Tabla 5.9: PC-009

PC-010	**Se comprueba que el contenido obtenido de la base de datos cumple que la información pasada entre cliente y servidor es válida.**	
Tipo de error	Del contenido obtenido de la base de datos	
Evaluación por página	pacientesModificar.html	Correcto
	registroPacientes.html	Correcto
	psicologoPerfil.html	Correcto
	psicologosModificar.html	Correcto
	registroPsicologo.html	Correcto

Tabla 5.10: PC-010

PC-011	**Se comprueba que el contenido obtenido de la base de datos cumple que se formatean correctamente los datos recogidos del usuario.**	
Tipo de error	Del contenido obtenido de la base de datos	
Evaluación por página	pacientesModificar.html	Correcto
	registroPacientes.html	Correcto
	psicologoPerfil.html	Correcto
	psicologosModificar.html	Correcto
	registroPsicologo.html	Correcto

Tabla 5.11: PC-011

PC-012	**Se comprueba que el contenido obtenido de la base de datos cumple que se formatean correctamente los datos mostrados al usuario.**	
Tipo de error	Del contenido obtenido de la base de datos	
Evaluación por página	pacientesModificar.html	Correcto
	registroPacientes.html	Correcto
	psicologoPerfil.html	Correcto
	psicologosModificar.html	Correcto
	registroPsicologo.html	Correcto

Tabla 5.12: PC-012

5.1.3. Incremento 3

Se evaluará cada tipo de error por página. Las pruebas de contenido del incremento 3 son PC-013, PC-014, PC-015, PC-016, PC-017 y PC-018.

PC-013	**Se comprueba que no existen errores sintácticos en los textos de la página con ayuda de un corrector ortográfico.**	
Tipo de error	Sintáctico	
Evaluación por página	pacientesMail.html	Correcto
	psicologosMail.html	Correcto
	psicologoCalendario.html	Correcto
	psicologoPerfil.html	Correcto

Tabla 5.13: PC-013

5.2. Pruebas de interfaz

La estrategia global consiste en encontrar errores relacionados con mecanismos de interfaz específicos, con la semántica de navegación de la interfaz, con la funcionalidad de la plataforma web o con el despliegue de su contenido. Los pasos que se van a seguir son:

- Se prueban las características de la interfaz para garantizar que las reglas de diseño, estética y contenido visual estén disponibles sin error para el usuario mediante una revisión visual.

- Los componentes de la interfaz se prueban de manera análoga a una prueba de unidad (verificación).

PC-014	**Se comprueba que no existen errores semánticos en los textos de la página mediante la revisión y respuesta de las preguntas en cuestión.**	
Tipo de error	Semántico	
Evaluación por página	pacientesMail.html	Correcto
	psicologosMail.html	Correcto
	psicologoCalendario.html	Correcto
	psicologoPerfil.html	Correcto

Tabla 5.14: PC-014

PC-015	**Se comprueba que la estructura y contenido de los objetos existentes en la página es correcta mediante revisión manual.**	
Tipo de error	De organización o estructura de contenido	
Evaluación por página	pacientesMail.html	Correcto
	psicologosMail.html	Correcto
	psicologoCalendario.html	Correcto
	psicologoPerfil.html	Correcto

Tabla 5.15: PC-015

PC-016	**Se comprueba que el contenido obtenido de la base de datos cumple que la información pasada entre cliente y servidor es válida.**	
Tipo de error	Del contenido obtenido de la base de datos	
Evaluación por página	pacientesMail.html	Correcto
	psicologosMail.html	Correcto
	psicologoCalendario.html	Correcto
	psicologoPerfil.html	Correcto

Tabla 5.16: PC-016

PC-017	**Se comprueba que el contenido obtenido de la base de datos cumple que se formatean correctamente los datos recogidos del usuario.**	
Tipo de error	Del contenido obtenido de la base de datos	
Evaluación por página	pacientesMail.html	Correcto
	psicologosMail.html	Correcto
	psicologoCalendario.html	Correcto
	psicologoPerfil.html	Correcto

Tabla 5.17: PC-017

PC-018	**Se comprueba que el contenido obtenido de la base de datos cumple que se formatean correctamente los datos mostrados al usuario.**	
Tipo de error	Del contenido obtenido de la base de datos	
Evaluación por página	pacientesMail.html	Correcto
	psicologosMail.html	Correcto
	psicologoCalendario.html	Correcto
	psicologoPerfil.html	Correcto

Tabla 5.18: PC-018

Para encontrar los errores, se realizarán las siguientes pruebas en función del componente de la interfaz:

- Vínculos

  Se construye una lista con todos los vínculos y se ejecuta cada uno individualmente.

- Formularios

  Se comprueban las siguientes preguntas:

  - Las etiquetas identifican correctamente los campos dentro del formulario.
  - Los campos obligatorios se identifican visualmente para el usuario.
  - El servidor recibe toda la información contenida dentro del formulario, y ningún dato se pierde en la transmisión entre cliente y servidor.
  - Se usan valores por defecto adecuados cuando el usuario no selecciona de un menú desplegable o conjunto de botones.
  - Las funciones del navegador, como el retroceso, no corrompen la entrada de datos en el formulario.
  - Los guiones que realizan la comprobación de errores en los datos ingresados funcionan de manera adecuada y proporcionan mensajes de error significativos.
  - Los campos del formulario tienen ancho y tipos de datos adecuados.
  - El formulario establece salvaguardas adecuadas que prohíben que el usuario ingrese cadenas de texto más largas que cierto máximo predefinido.
  - Todas las opciones para menús desplegables se ordenan y especifican en forma significativa para el usuario final.
  - Las características de auto completado del navegador no conducen a errores en la entrada de datos.

- Ventanas pop-up

  Se plantean las siguientes cuestiones:

  - El pop-up tiene el tamaño y posición adecuadas.
  - El pop-up no cubre la ventana de la webapp original
  - El diseño estético del pop-up es consistente con el diseño estético de la interfaz.
  - Las barras de desplazamiento y otros elementos similares se ubican y funcionan de manera adecuada.

- Se realizan pruebas de usabilidad.

  Se evalúa el grado en el cual los usuarios pueden interactuar efectivamente con la webapp y el grado en que la webapp guía las acciones de usuario, proporciona retroalimentación significativa y refuerza un enfoque de interacción consistente, haciéndola amigable y facilitando la vida del usuario.

  En este caso, a pesar de que el ingeniero es quién contribuye al diseño de las pruebas de usabilidad, será una tercera persona quién las ejecute.

  Para realizar este tipo de pruebas, se fijan distintas categorías y objetivos de prueba, y se plantean preguntas en relación a las mismas. En las pruebas se ve reflejado la media de los resultados del test de usabilidad.

- Se realizan pruebas de accesibilidad

  Finalmente, se realiza una prueba de accesibilidad para garantizar que la plataforma es accesible para la mayoría de su público, aunque sea a través de productos de apoyo. La prueba de usabilidad se realizó a través de la herramienta para desarrolladores de Google Chrome Audits que analiza varios aspectos referentes a la accesibilidad:

  - Si el contraste de color es satisfactorio.
  - Los atributos HTML son utilizados corréctamente.
  - Los atributos ARIA siguen las mejores prácticas para mejorar la experiencia de usuario que usan productos de apoyo, como un lector de pantalla.
  - Los elementos tienen nombres discernibles:
  - Los botones tienen nombre accesible.
  - Los links tienen nombre accesible.
  - Los elementos describen bien el contenido.
  - Los elementos están bien estructurados.
  - El lenguaje de la página está corréctamente especificado.
  - Las etiquetas meta están usadas de manera apropiada.

  Para cada página analizada, se documentará las acciones a realizar para corregir los matices producidos.

  En cada uno de los incrementos se ha ido afianzando la accesibilidad de la aplicación. Los resultados mostrados son los referentes al último incremento.

5.2.1. Incremento 1

Las pruebas de interfaz realizadas en el incremento 1 son PI-001, PI-002, PI-003 y PI-004.

PI-001	**Se prueban las características de la interfaz para garantizar que las reglas de diseño, estética y contenido visual estén disponibles sin error para el usuario mediante una revisión visual.**	
Evaluación por página	inicio.html	Correcta
	pacientesResultados.html	Correcta
	cuestionarioPacientes.html	Correcta

Tabla 5.19: PI-001

PI-002	**Se realiza una prueba de unidad concreta (verificación) para cada componente de la interfaz.**	
Evaluación por página	pacientesResultados.html	Correcta
	cuestionarioPacientes.html	Correcta

Tabla 5.20: PI-002

5.2.2. Incremento 2

Las pruebas de interfaz realizadas en el incremento 2 son PI-005, PI-006, PI-007 y PI-008.

PI-005	**Se prueban las características de la interfaz para garantizar que las reglas de diseño, estética y contenido visual estén disponibles sin error para el usuario mediante una revisión visual.**	
Evaluación por página	pacientesModificar.html	Correcto
	registroPacientes.html	Correcto
	psicologoPerfil.html	Correcto
	psicologosModificar.html	Correcto
	registroPsicologo.html	Correcto

Tabla 5.21: PI-005

5.2.3. Incremento 3

Las pruebas de interfaz realizadas en el incremento 3 son PI-009, PI-010, PI-011, PI-012 y PI-013.

PI-006	**Se realiza una prueba de unidad concreta (verificación) para cada componente de la interfaz.**	
Evaluación por página	pacientesModificar.html	Correcta
	registroPacientes.html	Correcta
	psicologoPerfil.html	Correcta
	psicologosModificar.html	Correcta
	registroPsicologo.html	Correcta
	pacientesResultados.html	Correcta
	inicio.html	Correcta

Tabla 5.22: PI-006

PI-009	**Se prueban las características de la interfaz para garantizar que las reglas de diseño, estética y contenido visual estén disponibles sin error para el usuario mediante una revisión visual.**	
Evaluación por página	pacientesMail.html	Correcto
	psicologosMail.html	Correcto
	psicologoPerfil.html	Correcto
	psicologoCalendario.html	Correcto

Tabla 5.23: PI-009

PI-010	**Se realiza una prueba de unidad concreta (verificación) para cada componente de la interfaz. Anexo**	
Evaluación por página	pacientesMail.html	Correcto
	psicologosMail.html	Correcto
	psicologoPerfil.html	Correcto
	psicologoCalendario.html	Correcto

Tabla 5.24: PI-010

PI-011	**Prueba de usabilidad de los pacientes**	
Categoría	**Pregunta**	**Evaluación**
Interactividad	¿El sitio web es fácil de utilizar?	4,33 (De acuerdo)
	¿La navegación en la página web es fácil e intuitiva?	4 (De acuerdo)
	¿Ha sido fácil encontrar las características en los menús?	4 (De acuerdo)
	¿Los mecanismos de interacción (por ejemplo, menús desplegables, botones, punteros) son fáciles de entender y usar?	3,67 (De acuerdo)
	¿Las características, funciones y contenido importantes pueden usarse de forma oportuna?	4,67 (De acuerdo)
	Utilizar el programa ha sido...	1 (Difícil)
Plantilla	¿Encuentras lo que buscas de forma rápida en la página web?	4,67 (De acuerdo)
	¿Necesitarías consultar un manual para el uso de la página?	1,33 (De acuerdo)
	¿Te has sentido seguro al usar el sitio web?	3,33 (De acuerdo)
	¿Los contenidos de la página web están bien estructurados?	4 (De acuerdo)
Legibilidad	¿El texto está bien escrito y es comprensible?	3 (De acuerdo)
	Comprender los mensajes ha sido...	1,33 (Difícil)
	¿La información en la página está dispuesta de forma clara?	4,33 (De acuerdo)
	La recuperación de errores ha sido...	1,67 (Difícil)
Estética	¿La plantilla, color, fuente y características relacionadas facilitan el uso?	4,67 (De acuerdo)
	¿Los usuarios "se sienten cómodos" con la apariencia y el sentimiento que transmite la webapp?	4,67 (De acuerdo)
Personalización	Contestar el test ha sido...	1 (Difícil)
	Filtrar los resultados del test ha sido...	2 (Difícil)
	Pedir una cita al psicólogo es...	2 (Difícil)
	Registrarse en la plataforma es...	1 (Difícil)
	¿La información mostrada sobre el psicólogo me resulta de utilidad?	3,33 (De acuerdo)

Tabla 5.25: PI-011

PI-012	**Prueba de usabilidad psicólogos**	
Categoría	**Pregunta**	**Evaluación**
Interactividad	¿El sitio web es fácil de utilizar?	4,5 (De acuerdo)
	¿La navegación en la página web es fácil e intuitiva?	4,5 (De acuerdo)
	¿Ha sido fácil encontrar las características en los menús?	5 (De acuerdo)
	¿Los mecanismos de interacción (por ejemplo, menús desplegables, botones, punteros) son fáciles de entender y usar?	5 (De acuerdo)
	¿Las características, funciones y contenido importantes pueden usarse de forma oportuna?	4 (De acuerdo)
	Utilizar el programa ha sido...	1 (Difícil)
Plantilla	¿Encuentras lo que buscas de forma rápida en la página web?	4 (De acuerdo)
	¿Necesitarías consultar un manual para el uso de la página?	1 (De acuerdo)
	¿Te has sentido seguro al usar el sitio web?	4 (De acuerdo)
	¿Los contenidos de la página web están bien estructurados?	3,5 (De acuerdo)
Legibilidad	¿El texto está bien escrito y es comprensible?	4 (De acuerdo)
	Comprender los mensajes ha sido...	1,5 (Difícil)
	¿La información en la página está dispuesta de forma clara?	3,5 (De acuerdo)
	La recuperación de errores ha sido...	2,5 (Difícil)
Estética	¿La plantilla, color, fuente y características relacionadas facilitan el uso?	5 (De acuerdo)
	¿Los usuarios "se sienten cómodos" con la apariencia y el sentimiento que transmite la webapp?	3,5 (De acuerdo)
Personalización	Dar respuesta a las peticiones de los pacientes ha sido...	1 (Difícil)
	Consultar el calendario de citas ha sido...	1 (Difícil)

Tabla 5.26: PI-012

PI-013	**Prueba de accesibilidad**
Página	**Evaluación**
inicio.html	97 %
pacientesResultados.html	94 %
cuestionarioPacientes.html	97 %
pacientesModificar.html	100 %
pacientesMail.html	94 %
registroPacientes.html	97 %
registroPsicologos.html	97 %
psicologosMail.html	94 %
perfilPsicologo.html	94 %
psicologoCalendario.html	100 %

Tabla 5.27: PI-013

5.3. Pruebas de navegación

Existen dos tipos de prueba de navegación:

- Prueba de la sintaxis de navegación

  Se prueba cada uno de los mecanismos de navegación de la siguiente forma:

  - Redirecciones

    Gestionan que el usuario introduzca una URL inexistente. Por tanto, se prueban URLs incorrectas y todas redirigen a la página de inicio.
  - Marcas de página

    La webapp debe asegurarse de que el título de página al extraerse sea significativo al crear la marca.
  - Mapa de sitio

    Proporciona una tabla de contenido completa para todas las páginas web. Se debe probar cada vínculo de entrada del mapa de sitio.

    En nuestra plataforma no va a existir mapa del sitio, puesto que todos los servicios se pueden acceder desde cualquier página del sitio web.

- Prueba de la semántica de navegación.

  Para cada USN se deben contestar a las siguientes preguntas:

  - ¿La USN se logra en su totalidad sin error?
  - ¿Todo nodo de navegación (definido por una USN) se alcanza dentro del contexto de las rutas de navegación definidas por la USN?

- ¿Existe un mecanismo (distinto a la flecha "retroceso" del navegador) para regresar al nodo de navegación anterior y al comienzo de la ruta de navegación?
- ¿Los mecanismos de navegación dentro de un gran nodo de navegación (es decir, una página web grande) funcionan de manera adecuada?
- Si una función debe ejecutarse en un nodo y el usuario elige no proporcionar entrada, ¿el resto de la USN puede completarse?
- Si una función debe ejecutarse en un nodo y ocurre un error en el procesamiento de la función, ¿la USN puede completarse?
- ¿Los nombres de nodo son significativos para los usuarios finales?
- ¿El usuario entiende su ubicación dentro de la arquitectura de contenido conforme se ejecuta la USN?

5.3.1. Incremento 1

Como es necesaria la correcta implementación de la gestión de usuarios del incremento 2 para que las USN sean ejecutadas en su totalidad, se posponen la USN-003, USN-004, USN-008 y USN-009 correspondientes a los casos de uso del incremento 1. La ejecución de las pruebas correspondientes se realizarán durante las pruebas del incremento 2.

5.3.2. Incremento 2

Las pruebas de navegación realizadas en el incremento 1 son PN-001 y PN-002

PN-001	**Se evalúa la correcta ejecución de la navegación para cada una de las USN del incremento 1.**	
Evaluación	USN-003	Correcto
	USN-004	Correcto
	USN-008	Correcto
	USN-009	Correcto

Tabla 5.28: PN-001

5.3.3. Incremento 3

Las pruebas de navegación realizadas en el incremento 3 son PN-005 y PN-006.

PN-002	**Se evalúa la correcta ejecución de la navegación para cada una de las USN del incremento 2.**	
Evaluación	USN-001	Correcto
	USN-002	Correcto
	USN-006	Correcto
	USN-010	Correcto

Tabla 5.29: PN-002

PN-003	**Se prueban los mecanismos de navegación.**	
Evaluación	Redirecciones	Correcto
	Marcas de página	Aceptable – Se identifica correctamente la página pero de forma no significativa.
	Mapa de sitio	No aplica
	Motores de búsqueda internos	No aplica

Tabla 5.30: PN-003

PN-004	**Se evalúa la correcta ejecución de la navegación para cada una de las USN del incremento 3.**	
Evaluación	USN-005	Correcta
	USN-007	Correcta
	USN-011	Correcta
	USN-012	Correcta

Tabla 5.31: PN-004

5.4. Prueba de componente

Cada componente se prueba dentro del contexto de un caso de uso de forma análoga a la prueba de validación,.es decir, para cada requisito. Para algunas pruebas, se probará una opción que pueda conducir a error para estudiar cómo es la gestión del mismo (prueba del error forzado), y otra opción, que produzca un éxito.

5.4.1. Incremento 1

Las pruebas de componente realizadas en el incremento 1 son PCo-001 y PCo-002.

PCo-001	**Prueba del requisito RF-005**
Descripción del requisito	Se garantizará al paciente el emparejamiento con el psicólogo más adecuado para tratar su caso. Cualquier paciente registrado en la plataforma podrá especificar sus síntomas a través de un cuestionario asignándole al menos un psicólogo al paciente tras conocer su patología.
Descripción de la prueba	1. El paciente selecciona“Hacer el test” en su página de resultados.
	2. El paciente cubre sí o no a las respuestas del test.
	3. El paciente envía el cuestionario al pulsar el botón “Enviar”.
Criterios de paso/fallo	El sistema guardará las patologías identificadas en el paciente, así como los psicólogos que pueden tratarlo.
	La aplicación muestra en la página de resultados del paciente al listado de psicólogos que le pueden tratar.
Estado	Correcto

Tabla 5.32: PCo-001

5.4.2. Incremento 2

Las pruebas de componente realizadas en el incremento 2 son PCo-003, PCo-004, PCo-005 y PCo-006.

PCo-002	**Prueba del requisito RF-006**
Descripción del requisito	Cualquier paciente, que haya obtenido resultados concluyentes en el cuestionario, podrá filtrarlos en base a distintos criterios como las características geográficas, el precio o su seguro médico.
Descripción de la prueba	1. El paciente ha resuelto el test previamente y tiene resultados válidos.
	2. Se cubren los campos del cuestionario.
	3. Se selecciona el botón "Filtrar"
Criterios de paso/fallo	Se mostrará al usuario los psicólogos que pueden tratarle filtrados por las preferencias de búsqueda indicadas.
Estado	Correcto

Tabla 5.33: PCo-002

PCo-003	**Prueba del requisito RF-001**
Descripción del requisito	Cualquier usuario registrado en la plataforma podrá acceder a la plataforma a través de la página de acceso.
Descripción de la prueba	1: Se pueden dar los siguientes casos:
	1.1. El usuario cubre el formulario de la página de inicio y le da a "Acceder"
	1.2. ídem desde la página de registro de pacientes
	1.3. Ídem desde la página de registro de psicólogos
Criterios de paso/fallo	Paso: El sistema autenticará al usuario con éxito y le mostrará su página principal (los resultados para el paciente y la bandeja de entrada para el psicólogo).
	Fallo: El sistema no podrá autenticar al usuario, permanecerá en la página actual y mostrará un mensaje de error.
Estado	Correcto

Tabla 5.34: PCo-003

PCo-004	**Prueba del requisito RF-002**
Descripción del requisito	Cualquier usuario podrá acceder a los servicios de la plataforma tras haber cubierto el formulario de registro.
Descripción de la prueba	El usuario cubre el formulario de registro (dependiendo de su perfil paciente o psicólogo) y lo enviará.
Criterios de paso/fallo	Si el usuario es un paciente. Paso: El sistema guardará los datos del paciente en la BBDD, lo autenticará y le mostrará su página de resultados. Fallo: El sistema no registrará los datos del formulario, permanecerá en la página actual y mostrará un mensaje de error.
	Si el usuario es un psicólogo. Paso: El sistema enviará un correo electrónico a la cuenta de correo,encargada de validar todas las peticiones de alta de psicólogo. Fallo: El sistema no registrará los datos del formulario, permanecerá en la página actual y mostrará un mensaje de error.
Estado	Correcto

Tabla 5.35: PCo-004

PCo-005	**Prueba del requisito RF-003**
Descripción del requisito	Cualquier usuario registrado en la plataforma podrá darse de baja de la plataforma.
Descripción de la prueba	El usuario pulsa el botón “Darme de Baja” y posteriormente también, el de “Continuar” la baja.
Criterios de paso/fallo	El sistema eliminará al usuario de la base de datos, y le mostrará a éste la página principal con la sesión cerrada.
Estado	Correcto

Tabla 5.36: PCo-005

PCo-006	**Prueba del requisito RF-004**
Descripción del requisito	Cualquier usuario registrado en la plataforma podrá modificar sus datos de usuario.
Descripción de la prueba	1.El usuario cubrirá el formulario de modificación de datos correspondiente.
	2. El usuario seleccionará el botón "Enviar" .
Criterios de paso/fallo	Si el usuario es un paciente: Paso: El sistema modificará los datos del paciente en la BBDD y le mostrará un mensaje de éxito. Fallo: El sistema no registrará los datos del formulario, permanecerá en la página actual y mostrará un mensaje de error.
	Si el usuario es un psicólogo: Paso: El sistema enviará un correo electrónico a la cuenta de correo encargada de validar todas las peticiones de alta de psicólogo. Fallo: El sistema no registrará los datos del formulario, permanecerá en la página actual y mostrará un mensaje de error.
Estado	Correcto

Tabla 5.37: PCo-006

5.4.3. Incremento 3

Las pruebas de componente realizadas en el incremento 3 son PCo-007 y PCo-008.

PCo-007	**Prueba del requisito RF-007**
Descripción del requisito	El paciente podrá contactar con cualquiera de los psicólogos que le fueron asignados tras realizar el cuestionario.
Descripción de la prueba	1. El paciente cubrirá el formulario de contacto del psicólogo que haya seleccionado.
	2. El paciente seleccionará el botón "Enviar".
Criterios de paso/fallo	El sistema enviará un mensaje al psicólogo en cuestión, y mostrará un mensaje de éxito.
Estado	Correcto

Tabla 5.38: PCo-007

PCo-008	**Prueba del requisito RF-008**
Descripción del requisito	El paciente podrá valorar al psicólogo que le haya atendido.
Descripción de la prueba	1. El paciente enviará su valoración a través de un formulario.
	2. El paciente seleccionará el botón "Enviar".
Criterios de paso/fallo	El sistema publicará la valoración en el perfil del psicólogo, y mostrará un mensaje de éxito.
Estado	Correcto

Tabla 5.39: PCo-008

5.5. Pruebas seguridad y rendimiento

5.5.1. Prueba de seguridad

Las *webapps* y los entornos en los lados cliente y servidor donde se albergan representan un blanco atractivo para *hackers* externos, empleados internos, competidores deshonestos y para quien quiera robar información sensible, modificar contenido maliciosamente, degradar el rendimiento, deshabilitar la funcionalidad o avergonzar a una persona, organización o negocio.

Para proteger las posibles vulnerabilidades que puedan producirse en nuestra plataforma se han implantado los siguientes elementos de seguridad:

- Autenticación: La autenticación es la capacidad de demostrar que un usuario es realmente quién dicha persona asegura ser.

  La autenticación en la plataforma web se asegura por medio de e-mail y contraseña de usuario.

- Encriptado: Limitado es un mecanismo de codificación que protege los datos sensibles de forma que hace imposible leerlos por quienes tienen intenciones maliciosas.

  El requisito no funcional RNF-001: Encriptado de datos del proyecto pretende asegurar la seguridad y el cifrado de datos. Las contraseñas en la aplicación se protegen mediante el cifrado Blowfish gracias a la librería Bcrypt. La prueba de este requisito PS-001 buscará acreditar de que se está realizando el encriptado de forma correcta, pero no tratará de comprobar la resistencia del cifrado en sí.

PS-001	**Prueba del requisito RF-008**
Descripción del requisito	El paciente podrá valorar al psicólogo que le haya atendido.
Descripción de la prueba	1. El paciente enviará su valoración a través de un formulario.
	2. El paciente seleccionará el botón "Enviar".
Criterios de paso/fallo	El sistema publicará la valoración en el perfil del psicólogo, y mostrará un mensaje de éxito.
Estado	Correcto

Tabla 5.40: PS-001

5.5.2. Prueba de rendimiento

Las pruebas de rendimiento se usan para descubrir problemas de rendimiento que pueden ser resultado de: Falta de recursos en el lado servidor, red con ancho de banda inadecuada, capacidades de base de datos, capacidades de sistema operativo deficientes o débiles, funcionalidad de *webapp* pobremente diseñada y otros conflictos de *hardware* o software que pueden conducir al rendimiento cliente-servidor degradado. La intención es:

- Comprender como responde el sistema conforme aumenta la carga (usuarios número de transacciones o volumen de datos global)

- Recopilar mediciones que conducirán a modificaciones de diseño para mejorar el rendimiento.

  En el proyecto el requisito no funcional RNF-002: Tiempo de respuesta de asignación perseguía dar una respuesta rápida al usuario cuando se dieran

los resultados del algoritmo de asignación. La prueba para cerciorar que se cumple dicho requisito es la PR-001.

PR-001	**Prueba del requisito RNF-002**
Descripción del requisito	El paciente tras enviar las respuestas del cuestionario los resultados de asignación de psicólogo del mismo deberán tener un tiempo de respuesta de como máximo 5 segundos.
Descripción de la prueba	Se medirá con una herramienta el tiempo de respuesta de la página.
Criterios de paso/fallo	La prueba se superará si el tiempo de respuesta ha sido inferior a o igual a 5 segundos. En caso contrario fallará.
Estado	Correcto

Tabla 5.41: PR-001

Capítulo 6

Conclusiones

En el presente proyecto se ha conseguido desarrollar una plataforma web que sirve como producto mínimo viable a Emozio.

La plataforma permite que los pacientes que estén registrados puedan ser caracterizados cubriendo un test cuyas preguntas se corresponden con distintos síntomas que están presentes en las patologías de nuestra base de datos.

Una vez que el sistema determina el diagnóstico del paciente, es decir, qué patologías padece y en qué porcentaje, se le asocian aquellos psicólogos que tratan esas patologías. En cualquier momento, el psicólogo puede consultar el perfil público de estos psicólogos. En el perfil del psicólogo, el paciente puede dejarle un comentario con valoración.

El paciente puede decidir a cuál de los psicólogos (uno o varios) les envía una solicitud de cita. Una vez enviada, queda registrada en la página donde quedan listadas sus solicitudes pendientes de respuesta, aceptadas y rechazadas.

Cabe añadir, que el paciente puede registrarse, modificar sus datos o darse de baja.

Por otra parte, los psicólogos registrados en la plataforma pueden consultar las solicitudes que le han sido enviadas y aceptarlas o rechazarlas, dependiendo de su criterio. En estas solicitudes, se muestra el diagnóstico del paciente, por lo que puede entrar a valorar si puede o no tratarlo.

Los psicólogos, al aceptar una solicitud de cita, deben especificar cuándo se va a dar lugar. Esta cita, queda guardada en un calendario de citas al que puede consultar en cualquier momento.

Los psicólogos, además, pueden responder a los comentarios realizados por pacientes que aparezcan en su perfil.

Para poder pertenecer a la plataforma, puede rellenar un formulario que es enviado al correo electrónico de mi equipo Emozio. Nuestro equipo, entrará a valorar si el psicólogo es adecuado y cumple las certificaciones y garantías necesarias para pertenecer a nuestro catálogo de psicólogos. De la misma forma, cuando un psicólogo quiera modificar sus datos, también llegarán a nuestro correo para ser evaluados. Finalmente, un psicólogo puede darse de baja en cualquier momento.

6.1. Posibles ampliaciones

A raíz de haber realizado este proyecto, surgen nuevas oportunidades de mejora, arreglos y ampliaciones:

- Conseguir que la plataforma web sea completamente *responsive*, es decir, que funcione corréctamente en cualquier tipo de dispositivo.
- Conseguir garantizar que la plataforma cumpla todas las leyes vinculantes a la misma, como la Ley Orgánica 15/1999, del 13 de diciembre, sobre la Protección de Datos de Carácter Personal (LOPD).
- Mejorar el algoritmo de emparejamiento mediante técnicas de inteligencia artificial tras el estudio e investigación de psicología que se realice sobre cómo caracterizar los pacientes, los psicólogos, las patologías y cómo se relacionan entre ambos.
- Crear un perfil de administrador para que realice la evaluación de los candidatos psicólogos, haga de mediador entre pacientes y psicólogos si existe algún problema entre ambos; desde la plataforma.
- Sistema de notificaciones cuando llegue una nueva solicitud (en el caso del perfil de psicólogo), o una respuesta a una solicitud (en el caso del perfil de paciente).
- Permitir la cancelación de una cita por parte de psicólogos y pacientes.
- Mejorar el nombre de los archivos y de las rutas de la aplicación para que sea más representativo; y a la larga, sea más fácil darle soporte.
- Obtener los psicólogos que se han dado como resultado por orden de cercanía.
- Añadir nuevos filtros en el filtrado de resultados.
- Reflexionar y analizar una posible migración a una base de datos MySQL, ya que a pesar de que MongoDB de mucha flexibilidad, tiene demasiadas dependencias entre los documentos por lo que a veces los cambios no se consiguen reflejar en todo. Especialmente, a medida que crece el tamaño de la base de datos.
- Crear un historial de las citas que ha tenido un psicólogo con un determinado paciente, reflejando sus observaciones. De esta forma, podría tener un seguimiento del paciente siempre que quisiese a través de la plataforma.

Apéndice A

Anexo Pruebas

A.1. Pruebas de contenido

A.1.1. Incremento 1

Anexos de la PC-002

A.1.2. Incremento 2

Anexo de la PC-008.

A.1.3. Incremento 3

Anexos de la PC-014.

A.2. Pruebas de interfaz

A.2.1. Incremento 1

Anexos de la PI-002.

A.2.2. Incremento 2

Anexos de la PI-006.

PC-002	Anexo	
Página evaluada	inicio.html	
Preguntas	¿La información es realmente precisa?	Sí
	¿La información es concisa y puntual?	Sí
	¿La disposición del contenido es fácil de comprender para el usuario?	Sí
	¿La información situada dentro de un objeto de contenido puede encontrarse con facilidad?	Sí
	¿Se proporcionaron referencias adecuadas para toda la información derivada de otras fuentes?	No aplica
	¿La información presentada es consistente con la información presentada en otros objetos de contenido?	Sí
	¿El contenido no es ofensivo, confuso o abre la puerta a demandas?	Sí
	¿El contenido no infringe derechos de autor o nombres comerciales existentes?	Sí
	¿El contenido incluye vínculos internos que complementan el contenido existente? ¿Los vínculos son correctos?	No aplica
	¿El estilo estético del contenido no entra en conflicto con el estilo estético de la interfaz?	Sí

Tabla A.1: Anexo PC-002

PC-002	Anexo	
Página evaluada	pacientesResultados.html	
Preguntas	¿La información es realmente precisa?	Sí
	¿La información es concisa y puntual?	Sí
	¿La disposición del contenido es fácil de comprender para el usuario?	Sí
	¿La información situada dentro de un objeto de contenido puede encontrarse con facilidad?	Sí
	¿Se proporcionaron referencias adecuadas para toda la información derivada de otras fuentes?	No aplica
	¿La información presentada es consistente con la información presentada en otros objetos de contenido?	Sí
	¿El contenido no es ofensivo, confuso o abre la puerta a demandas?	Sí
	¿El contenido no infringe derechos de autor o nombres comerciales existentes?	Sí
	¿El contenido incluye vínculos internos que complementan el contenido existente? ¿Los vínculos son correctos?	Sí
	¿El estilo estético del contenido no entra en conflicto con el estilo estético de la interfaz?	Sí

Tabla A.2: Anexo PC-002

PC-008	Anexo	
Página evaluada	pacientesModificar.html	
Preguntas	¿La información es realmente precisa?	Sí
	¿La información es concisa y puntual?	Sí
	¿La disposición del contenido es fácil de comprender para el usuario?	Sí
	¿La información situada dentro de un objeto de contenido puede encontrarse con facilidad?	Sí
	¿Se proporcionaron referencias adecuadas para toda la información derivada de otras fuentes?	No aplica
	¿La información presentada es consistente con la información presentada en otros objetos de contenido?	Sí
	¿El contenido no es ofensivo, confuso o abre la puerta a demandas?	Sí
	¿El contenido no infringe derechos de autor o nombres comerciales existentes?	Sí
	¿El contenido incluye vínculos internos que complementan el contenido existente? ¿Los vínculos son correctos?	No aplica
	¿El estilo estético del contenido no entra en conflicto con el estilo estético de la interfaz?	Sí

Tabla A.3: Anexo PC-008

PC-012	Anexo	
Página evaluada	registroPacientes.html	
Preguntas	¿La información es realmente precisa?	Sí
	¿La información es concisa y puntual?	Sí
	¿La disposición del contenido es fácil de comprender para el usuario?	Sí
	¿La información situada dentro de un objeto de contenido puede encontrarse con facilidad?	Sí
	¿Se proporcionaron referencias adecuadas para toda la información derivada de otras fuentes?	No aplica
	¿La información presentada es consistente con la información presentada en otros objetos de contenido?	Sí
	¿El contenido no es ofensivo, confuso o abre la puerta a demandas?	Sí
	¿El contenido no infringe derechos de autor o nombres comerciales existentes?	Sí
	¿El contenido incluye vínculos internos que complementan el contenido existente? ¿Los vínculos son correctos?	Sí
	¿El estilo estético del contenido no entra en conflicto con el estilo estético de la interfaz?	Sí

Tabla A.4: Anexo PC-012

PC-008	Anexo	
Página evaluada	psicologoPerfil.html	
Preguntas	¿La información es realmente precisa?	Sí
	¿La información es concisa y puntual?	Sí
	¿La disposición del contenido es fácil de comprender para el usuario?	Sí
	¿La información situada dentro de un objeto de contenido puede encontrarse con facilidad?	Sí
	¿Se proporcionaron referencias adecuadas para toda la información derivada de otras fuentes?	No aplica
	¿La información presentada es consistente con la información presentada en otros objetos de contenido?	Sí
	¿El contenido no es ofensivo, confuso o abre la puerta a demandas?	Sí
	¿El contenido no infringe derechos de autor o nombres comerciales existentes?	Sí
	¿El contenido incluye vínculos internos que complementan el contenido existente? ¿Los vínculos son correctos?	Sí
	¿El estilo estético del contenido no entra en conflicto con el estilo estético de la interfaz?	Sí

Tabla A.5: Anexo PC-008

PC-008	Anexo	
Página evaluada	psicologosModificar.html ó registroPsicologo.html	
Preguntas	¿La información es realmente precisa?	Sí
	¿La información es concisa y puntual?	Sí
	¿La disposición del contenido es fácil de comprender para el usuario?	Sí
	¿La información situada dentro de un objeto de contenido puede encontrarse con facilidad?	Sí
	¿Se proporcionaron referencias adecuadas para toda la información derivada de otras fuentes?	No aplica
	¿La información presentada es consistente con la información presentada en otros objetos de contenido?	Sí
	¿El contenido no es ofensivo, confuso o abre la puerta a demandas?	Sí
	¿El contenido no infringe derechos de autor o nombres comerciales existentes?	Sí
	¿El contenido incluye vínculos internos que complementan el contenido existente? ¿Los vínculos son correctos?	Sí
	¿El estilo estético del contenido no entra en conflicto con el estilo estético de la interfaz?	Sí

Tabla A.6: Anexo PC-008

PC-014	Anexo	
Páginas evaluadas	pacientesMail.html, psicologosMail.html, psicologoPerfil.html, psicologoCalendario.html	
Preguntas	¿La información es realmente precisa?	Sí
	¿La información es concisa y puntual?	Sí
	¿La disposición del contenido es fácil de comprender para el usuario?	Sí
	¿La información situada dentro de un objeto de contenido puede encontrarse con facilidad?	Sí
	¿Se proporcionaron referencias adecuadas para toda la información derivada de otras fuentes?	No aplica
	¿La información presentada es consistente con la información presentada en otros objetos de contenido?	Sí
	¿El contenido no es ofensivo, confuso o abre la puerta a demandas?	Sí
	¿El contenido no infringe derechos de autor o nombres comerciales existentes?	Sí
	¿El contenido incluye vínculos internos que complementan el contenido existente? ¿Los vínculos son correctos?	No aplica
	¿El estilo estético del contenido no entra en conflicto con el estilo estético de la interfaz?	Sí

Tabla A.7: Anexo PC-014

Anexo PI-002		
Página	**Vínculos**	**Evaluación**
pacientesResultados.html	Hacer el test	Correcta
	Filtrar	Correcta
	Ver información psicólogo	Correcta
	Pedir cita	Correcta
cuestionarioPacientes.html	Siguiente	Correcta
	Enviar	Correcta

Tabla A.8: Anexo PI-002

Anexo PI-002		
Página	**Formulario**	**Evaluación**
pacientesResultados.html	Las etiquetas identifican correctamente los campos dentro del formulario.	Correcta
	Los campos obligatorios se identifican visualmente para el usuario.	No aplica
	El servidor recibe toda la información contenida dentro del formulario, y ningún dato se pierde en la transmisión entre cliente y servidor.	Correcta
	Se usan valores por defecto adecuados cuando el usuario no selecciona de un menú desplegable o conjunto de botones.	Correcta
	Las funciones del navegador, como el retroceso, no corrompen la entrada de datos en el formulario.	Correcta
	Los guiones que realizan la comprobación de errores en los datos ingresados funcionan de manera adecuada y proporcionan mensajes de error significativos.	No aplica
	Los campos del formulario tienen ancho y tipos de datos adecuados.	Correcta
	El formulario establece salvaguardas adecuadas que prohíben que el usuario ingrese cadenas de texto más largas que cierto máximo predefinido.	No aplica
	Todas las opciones para menús desplegables se ordenan y especifican en forma significativa para el usuario final.	No aplica
	Las características de autocompletado del navegador no conducen a errores en la entrada de datos.	No aplica

Tabla A.9: Anexo PI-002

Anexo PI-002		
Página	**Pop-up: Confirmación Hacer el test**	**Evaluación**
pacientesResultados.html	El pop-up tiene el tamańo y posición adecuadas.	Correcta
	El pop-up no cubre la ventana de la webapp original	Correcta
	El diseńo estético del pop-up es consistente con el diseńo estético de la interfaz.	Correcta
	Las barras de desplazamiento y otros elementos similares se ubican y funcionan de manera adecuada.	No aplica

Tabla A.10: Anexo PI-002

Anexo PI-006		
Página	**Vínculos**	**Evaluación**
inicio.html	Registrar	Correcto
	Inicio	Correcto
	Resultados	Correcto
	Perfil >Modificar mis datos (Paciente)	Correcto
	Perfil >Salir (Paciente)	Correcto
	Perfil >Cómo me ven	Correcto
	Perfil >Modificar mis datos (Psicólogo)	Correcto
	Perfil >Salir (Psicólogo)	Correcto
registroPacientes.html	Enviar	Correcto
	Pedir cita	Correcto
	Términos y condiciones de uso	Correcto
registroPsicologo.html	Siguiente 1	Correcto
	Términos y condiciones de uso	Correcto
	Siguiente 2	Correcto
	Enviar	Correcto
	Registrar	Correcto
pacientesResultados.html	Inicio	Correcto
	Resultados	Correcto
	Perfil >Modificar mis datos	Correcto
	Perfil >Salir	Correcto

Tabla A.11: Anexo PI-006

Anexo PI-006		
Página	**Vínculos**	**Evaluación**
pacientesModificar.html	Inicio	Correcto
	Resultados	Correcto
	Perfil >Modificar mis datos	Correcto
	Perfil >Salir	Correcto
	Darme de baja	Correcto
psicologosModificar.html	Inicio	Correcto
	Perfil >Modificar mis datos	Correcto
	Perfil >Cómo me ven	Correcto
	Perfil >Salir	Correcto
	Siguiente 1	Correcto
	Siguiente 2	Correcto
	Enviar	Correcto
	Darme de baja	Correcto
psicologosPerfil.html	Inicio	Correcto
	Perfil >Modificar mis datos (Psicólogo)	Correcto
	Perfil >Cómo me ven	Correcto
	Perfil >Salir (Psicólogo)	Correcto
	Inicio	Correcto
	Resultados	Correcto
	Perfil >Modificar mis datos (Paciente)	Correcto
	Perfil >Salir (Paciente)	Correcto
	Pedir cita	Correcto

Tabla A.12: Anexo PI-006

Anexo PI-006		
Página	**Formulario de acceso**	**Evaluación**
inicio.html	Las etiquetas identifican correctamente los campos dentro del formulario.	Correcta
registroPacientes.html	Los campos obligatorios se identifican visualmente para el usuario.	No aplica
registroPsicologo.html	El servidor recibe toda la información contenida dentro del formulario, y ningún dato se pierde en la transmisión entre cliente y servidor.	Correcta
	Se usan valores por defecto adecuados cuando el usuario no selecciona de un menú desplegable o conjunto de botones.	No aplica
	Las funciones del navegador, como el retroceso, no corrompen la entrada de datos en el formulario.	No aplica
	Los guiones que realizan la comprobación de errores en los datos ingresados funcionan de manera adecuada y proporcionan mensajes de error significativos.	Correcta
	Los campos del formulario tienen ancho y tipos de datos adecuados.	Correcta
	El formulario establece salvaguardas adecuadas que prohíben que el usuario ingrese cadenas de texto más largas que cierto máximo predefinido.	Correcta
	Todas las opciones para menús desplegables se ordenan y especifican en forma significativa para el usuario final.	No aplica
	Las características de autocompletado del navegador no conducen a errores en la entrada de datos.	No aplica

Tabla A.13: Anexo PI-006

Anexo PI-006		
Página	**Formulario de registro**	**Evaluación**
registroPacientes.html	Las etiquetas identifican correctamente los campos dentro del formulario.	Correcta
	Los campos obligatorios se identifican visualmente para el usuario.	No aplica
	El servidor recibe toda la información contenida dentro del formulario, y ningún dato se pierde en la transmisión entre cliente y servidor.	Correcta
	Se usan valores por defecto adecuados cuando el usuario no selecciona de un menú desplegable o conjunto de botones.	No aplica
	Las funciones del navegador, como el retroceso, no corrompen la entrada de datos en el formulario.	No aplica
	Los guiones que realizan la comprobación de errores en los datos ingresados funcionan de manera adecuada y proporcionan mensajes de error significativos.	Correcta
	Los campos del formulario tienen ancho y tipos de datos adecuados.	Correcta
	El formulario establece salvaguardas adecuadas que prohíben que el usuario ingrese cadenas de texto más largas que cierto máximo predefinido.	Correcta
	Todas las opciones para menús desplegables se ordenan y especifican en forma significativa para el usuario final.	Correcta
	Las características de autocompletado del navegador no conducen a errores en la entrada de datos.	Correcta

Tabla A.14: Anexo PI-006

Anexo PI-006		
Página	**Formularios de acceso y modificación de datos**	**Evaluación**
registroPsicologo.html psicologosModificar.html	Las etiquetas identifican correctamente los campos dentro del formulario.	Correcta
	Los campos obligatorios se identifican visualmente para el usuario.	Correcta
	El servidor recibe toda la información contenida dentro del formulario, y ningún dato se pierde en la transmisión entre cliente y servidor.	Correcta
	Se usan valores por defecto adecuados cuando el usuario no selecciona de un menú desplegable o conjunto de botones.	No aplica
	Las funciones del navegador, como el retroceso, no corrompen la entrada de datos en el formulario.	Correcta
	Los guiones que realizan la comprobación de errores en los datos ingresados funcionan de manera adecuada y proporcionan mensajes de error significativos.	Correcta
	Los campos del formulario tienen ancho y tipos de datos adecuados.	Correcta
	El formulario establece salvaguardas adecuadas que prohíben que el usuario ingrese cadenas de texto más largas que cierto máximo predefinido.	Correcta
	Todas las opciones para menús desplegables se ordenan y especifican en forma significativa para el usuario final.	Correcta
	Las características de autocompletado del navegador no conducen a errores en la entrada de datos.	Correcta

Tabla A.15: Anexo PI-006

Anexo PI-006		
Página	**Formularios de modificar datos y modificar contraseña**	**Evaluación**
pacientesModificar.html	Las etiquetas identifican correctamente los campos dentro del formulario.	Correcta
	Los campos obligatorios se identifican visualmente para el usuario.	No aplica
	El servidor recibe toda la información contenida dentro del formulario, y ningún dato se pierde en la transmisión entre cliente y servidor.	Correcta
	Se usan valores por defecto adecuados cuando el usuario no selecciona de un menú desplegable o conjunto de botones.	No aplica
	Las funciones del navegador, como el retroceso, no corrompen la entrada de datos en el formulario.	No aplica
	Los guiones que realizan la comprobación de errores en los datos ingresados funcionan de manera adecuada y proporcionan mensajes de error significativos.	Correcta
	Los campos del formulario tienen ancho y tipos de datos adecuados.	Correcta
	El formulario establece salvaguardas adecuadas que prohíben que el usuario ingrese cadenas de texto más largas que cierto máximo predefinido.	Correcta
	Todas las opciones para menús desplegables se ordenan y especifican en forma significativa para el usuario final.	Correcta
	Las características de autocompletado del navegador no conducen a errores en la entrada de datos.	Correcta

Tabla A.16: Anexo PI-006

Anexo PI-006		
Página	**Pop-up: Términos y condiciones de uso**	**Evaluación**
registroPacientes.html	El pop-up tiene el tamańo y posición adecuadas.	Correcta
registroPsicologo.html	El pop-up no cubre la ventana de la webapp original	Correcta
	El diseńo estético del pop-up es consistente con el diseńo estético de la interfaz.	Correcta
	Las barras de desplazamiento y otros elementos similares se ubican y funcionan de manera adecuada.	Correcta

Tabla A.17: Anexo PI-006

Anexo PI-006		
Página	**Pop-up: Confirmación Darme de baja**	**Evaluación**
pacientesModificar.html	El pop-up tiene el tamańo y posición adecuadas.	Correcta
psicologosModificar.html	El pop-up no cubre la ventana de la webapp original	Correcta
	El diseńo estético del pop-up es consistente con el diseńo estético de la interfaz.	Correcta
	Las barras de desplazamiento y otros elementos similares se ubican y funcionan de manera adecuada.	No aplica

Tabla A.18: Anexo PI-006

A.2.3. Incremento 3

Anexos de la PI-010.

Anexo PI-010		
Página	**Vínculos**	**Evaluación**
pacientesMail.html	Inicio	Correcta
	Resultados	Correcta
	Perfil >Modificar mis datos	Correcta
	Pendientes	Correcta
	Aceptadas	Correcta
	Rechazadas	Correcta
	Perfil psicólogo	Correcta
psicologoPerfil.html	Mis solicitudes	Correcta
	Pedir cita	Correcta
	Comentar	Correcta
	Perfil >Modificar mis datos	Correcta
psicologosMail.html	Inicio	Correcta
	Calendario	Correcta
	Perfil >¿Cómo me ven?	Correcta
	Perfil >Modificar mis datos	Correcta
	Aceptar	Correcta
	Rechazar	Correcta
	Pendientes	Correcta
	Aceptadas	Correcta
	Rechazadas	Correcta
psicologoCalendario.html	Inicio	Correcta
	Calendario	Correcta
	Mis solicitudes	Correcta
	Perfil >¿Cómo me ven?	Correcta
	Perfil >Modificar mis datos	Correcta

Tabla A.19: Anexo PI-010

Anexo PI-010		
Página	**Pop-up: Mi petición**	**Evaluación**
pacientesMail.html	El pop-up tiene el tamańo y posición adecuadas.	Correcta
	El pop-up no cubre la ventana de la webapp original	Correcta
	El diseńo estético del pop-up es consistente con el diseńo estético de la interfaz.	Correcta
	Las barras de desplazamiento y otros elementos similares se ubican y funcionan de manera adecuada.	No aplica

Tabla A.20: Anexo PI-010

Anexo PI-010		
Página	**Pop-up: Pedir cita (Paciente)**	**Evaluación**
perfilPsicologo.html	El pop-up tiene el tamańo y posición adecuadas.	Correcta
	El pop-up no cubre la ventana de la webapp original	Correcta
	El diseńo estético del pop-up es consistente con el diseńo estético de la interfaz.	Correcta
	Las barras de desplazamiento y otros elementos similares se ubican y funcionan de manera adecuada.	No aplica

Tabla A.21: Anexo PI-010

Anexo PI-010		
Página	**Pop-up: Comentar (Paciente)**	**Evaluación**
perfilPsicologo.html	El pop-up tiene el tamańo y posición adecuadas.	Correcta
	El pop-up no cubre la ventana de la webapp original	Correcta
	El diseńo estético del pop-up es consistente con el diseńo estético de la interfaz.	Correcta
	Las barras de desplazamiento y otros elementos similares se ubican y funcionan de manera adecuada.	No aplica

Tabla A.22: Anexo PI-010

Anexo PI-010		
Página	**Pop-up: Responder (Psicólogo)**	**Evaluación**
perfilPsicologo.html	El pop-up tiene el tamańo y posición adecuadas.	Correcta
	El pop-up no cubre la ventana de la webapp original	Correcta
	El diseńo estético del pop-up es consistente con el diseńo estético de la interfaz.	Correcta
	Las barras de desplazamiento y otros elementos similares se ubican y funcionan de manera adecuada.	No aplica

Tabla A.23: Anexo PI-010

Anexo PI-010		
Página	**Formulario:Comentar(Paciente)**	**Evaluación**
perfilPsicologo.html	Las etiquetas identifican correctamente los campos dentro del formulario.	Correcta
	Los campos obligatorios se identifican visualmente para el usuario.	No aplica
	El servidor recibe toda la información contenida dentro del formulario, y ningún dato se pierde en la transmisión entre cliente y servidor.	Pendiente
	Se usan valores por defecto adecuados cuando el usuario no selecciona de un menú desplegable o conjunto de botones.	Correcta
	Las funciones del navegador, como el retroceso, no corrompen la entrada de datos en el formulario.	Correcta
	Los guiones que realizan la comprobación de errores en los datos ingresados funcionan de manera adecuada y proporcionan mensajes de error significativos.	No aplica
	Los campos del formulario tienen ancho y tipos de datos adecuados.	Correcta
	El formulario establece salvaguardas adecuadas que prohíben que el usuario ingrese cadenas de texto más largas que cierto máximo predefinido.	Correcta
	Todas las opciones para menús desplegables se ordenan y especifican en forma significativa para el usuario final.	No aplica
	Las características de autocompletado del navegador no conducen a errores en la entrada de datos.	No aplica

Tabla A.24: Anexo PI-010

Anexo PI-010		
Página	**Formulario: Pedir cita (Paciente)**	**Evaluación**
perfilPsicologo.html	Las etiquetas identifican correctamente los campos dentro del formulario.	Correcta
	Los campos obligatorios se identifican visualmente para el usuario.	No aplica
	El servidor recibe toda la información contenida dentro del formulario, y ningún dato se pierde en la transmisión entre cliente y servidor.	Correcta
	Se usan valores por defecto adecuados cuando el usuario no selecciona de un menú desplegable o conjunto de botones.	Correcta
	Las funciones del navegador, como el retroceso, no corrompen la entrada de datos en el formulario.	Correcta
	Los guiones que realizan la comprobación de errores en los datos ingresados funcionan de manera adecuada y proporcionan mensajes de error significativos.	No aplica
	Los campos del formulario tienen ancho y tipos de datos adecuados.	Correcta
	El formulario establece salvaguardas adecuadas que prohíben que el usuario ingrese cadenas de texto más largas que cierto máximo predefinido.	Correcta
	Todas las opciones para menús desplegables se ordenan y especifican en forma significativa para el usuario final.	No aplica
	Las características de autocompletado del navegador no conducen a errores en la entrada de datos.	No aplica

Tabla A.25: Anexo PI-010

Anexo PI-010		
Página	**Formulario: Responder (Paciente)**	**Evaluación**
perfilPsicologo.html	Las etiquetas identifican correctamente los campos dentro del formulario.	Correcta
	Los campos obligatorios se identifican visualmente para el usuario.	No aplica
	El servidor recibe toda la información contenida dentro del formulario, y ningún dato se pierde en la transmisión entre cliente y servidor.	Correcta
	Se usan valores por defecto adecuados cuando el usuario no selecciona de un menú desplegable o conjunto de botones.	Correcta
	Las funciones del navegador, como el retroceso, no corrompen la entrada de datos en el formulario.	Correcta
	Los guiones que realizan la comprobación de errores en los datos ingresados funcionan de manera adecuada y proporcionan mensajes de error significativos.	No aplica
	Los campos del formulario tienen ancho y tipos de datos adecuados.	Correcta
	El formulario establece salvaguardas adecuadas que prohíben que el usuario ingrese cadenas de texto más largas que cierto máximo predefinido.	Correcta
	Todas las opciones para menús desplegables se ordenan y especifican en forma significativa para el usuario final.	No aplica
	Las características de autocompletado del navegador no conducen a errores en la entrada de datos.	No aplica

Tabla A.26: Anexo PI-010

Anexo PI-010		
Página	**Formulario: Aceptar**	**Evaluación**
psicologosMail.html	Las etiquetas identifican correctamente los campos dentro del formulario.	Correcta
	Los campos obligatorios se identifican visualmente para el usuario.	Correcta
	El servidor recibe toda la información contenida dentro del formulario, y ningún dato se pierde en la transmisión entre cliente y servidor.	Correcta
	Se usan valores por defecto adecuados cuando el usuario no selecciona de un menú desplegable o conjunto de botones.	No aplica
	Las funciones del navegador, como el retroceso, no corrompen la entrada de datos en el formulario.	Correcta
	Los guiones que realizan la comprobación de errores en los datos ingresados funcionan de manera adecuada y proporcionan mensajes de error significativos.	Correcta
	Los campos del formulario tienen ancho y tipos de datos adecuados.	Correcta
	El formulario establece salvaguardas adecuadas que prohíben que el usuario ingrese cadenas de texto más largas que cierto máximo predefinido.	Correcta
	Todas las opciones para menús desplegables se ordenan y especifican en forma significativa para el usuario final.	No aplica
	Las características de autocompletado del navegador no conducen a errores en la entrada de datos.	No aplica

Tabla A.27: Anexo PI-010

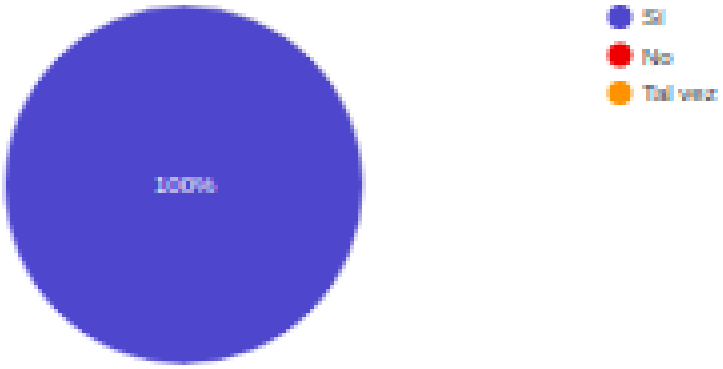
Anexo PI-010		
Página	**Formulario: Rechazar**	**Evaluación**
psicologosMail.html	Las etiquetas identifican correctamente los campos dentro del formulario.	Correcta
	Los campos obligatorios se identifican visualmente para el usuario.	Correcta
	El servidor recibe toda la información contenida dentro del formulario, y ningún dato se pierde en la transmisión entre cliente y servidor.	Correcta
	Se usan valores por defecto adecuados cuando el usuario no selecciona de un menú desplegable o conjunto de botones.	No aplica
	Las funciones del navegador, como el retroceso, no corrompen la entrada de datos en el formulario.	Correcta
	Los guiones que realizan la comprobación de errores en los datos ingresados funcionan de manera adecuada y proporcionan mensajes de error significativos.	Correcta
	Los campos del formulario tienen ancho y tipos de datos adecuados.	Correcta
	El formulario establece salvaguardas adecuadas que prohíben que el usuario ingrese cadenas de texto más largas que cierto máximo predefinido.	Correcta
	Todas las opciones para menús desplegables se ordenan y especifican en forma significativa para el usuario final.	No aplica
	Las características de autocompletado del navegador no conducen a errores en la entrada de datos.	No aplica

Tabla A.28: Anexo PI-010

Test de usabilidad de los pacientes

¿Volverías a visitar la página web?

2 responses



¿Cuáles son los motivos por los que volverías o no?

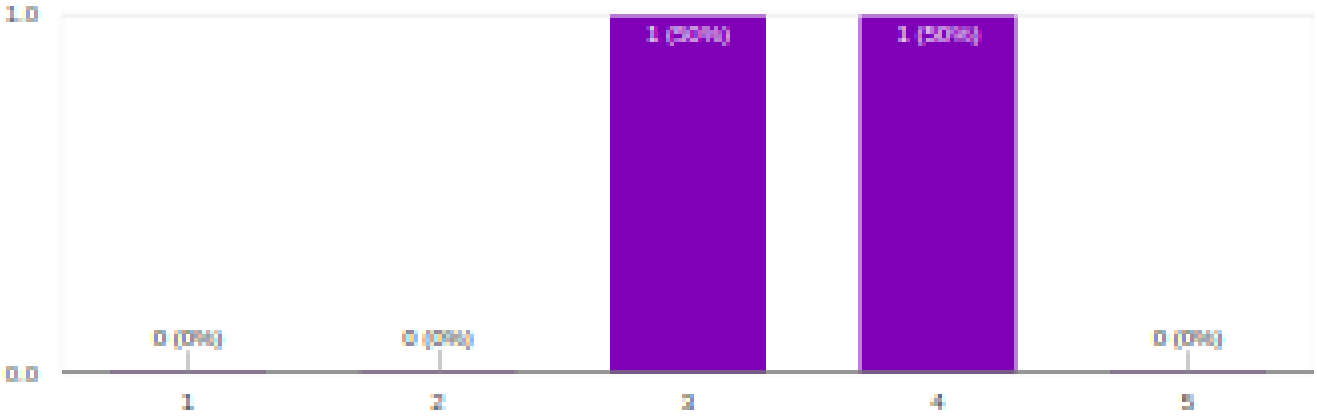
2 responses

Me resulta muy interesante el concepto, permitirá que personas con problemas pero intimidados por la sociedad actual puedan comunicarse con un psicólogo sin ningún tipo de problema.

Resulta interesante

¿La información en la página web está dispuesta de forma clara?

2 responses

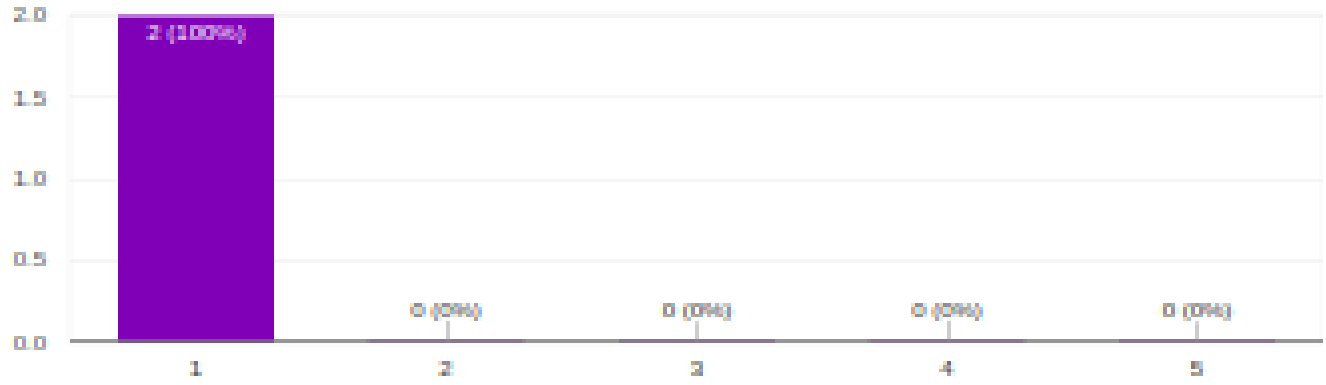


¿La navegación en la página web es fácil e intuitiva?

2 responses

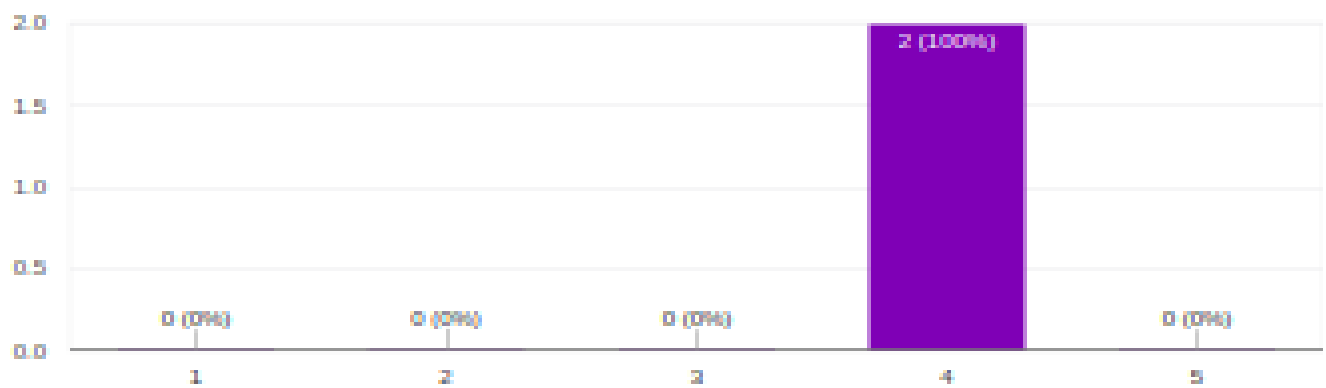
¿Necesitarias consultar un manual para el uso de la página?

2 responses



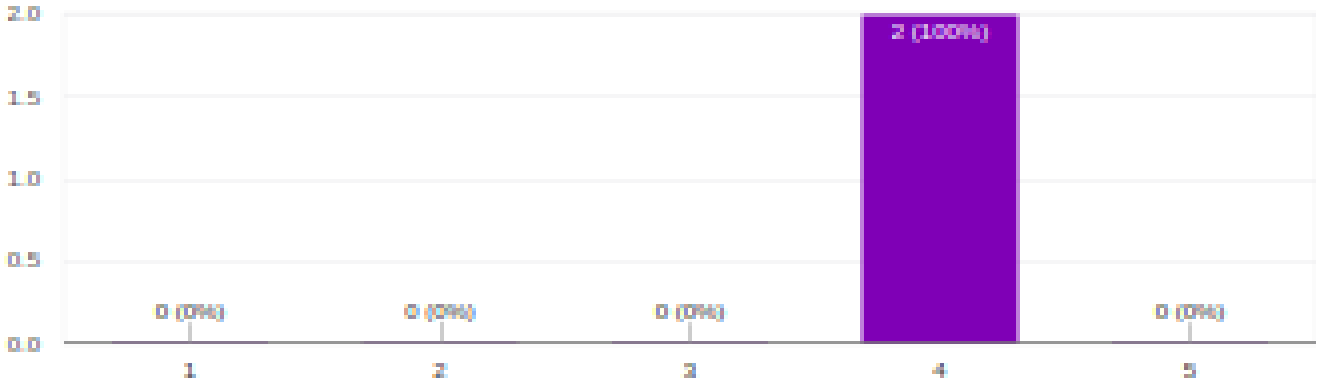
¿Encuentras lo que buscas de forma rápida en la página web?

2 responses



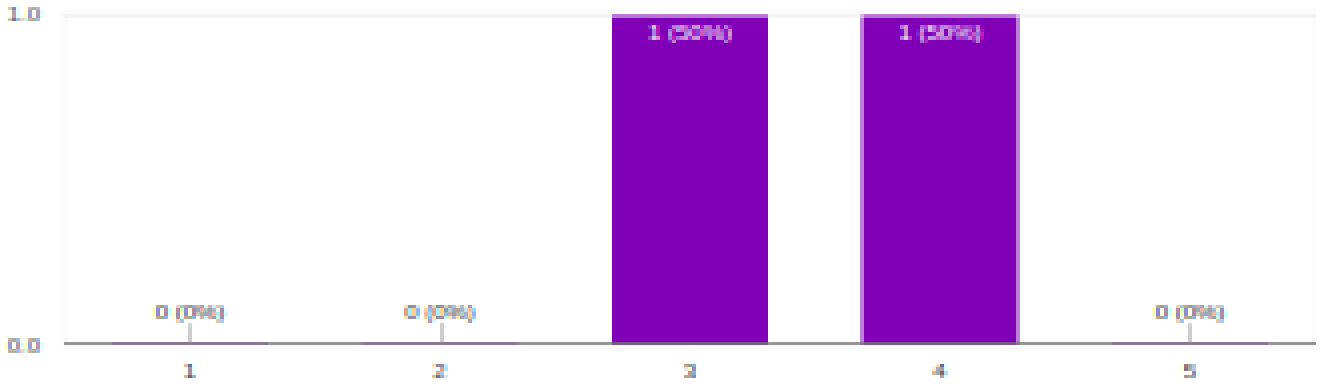
¿Te has sentido seguro al usar el sitio web?

2 responses



¿Los contenidos de la página web están bien estructurados?

2 responses

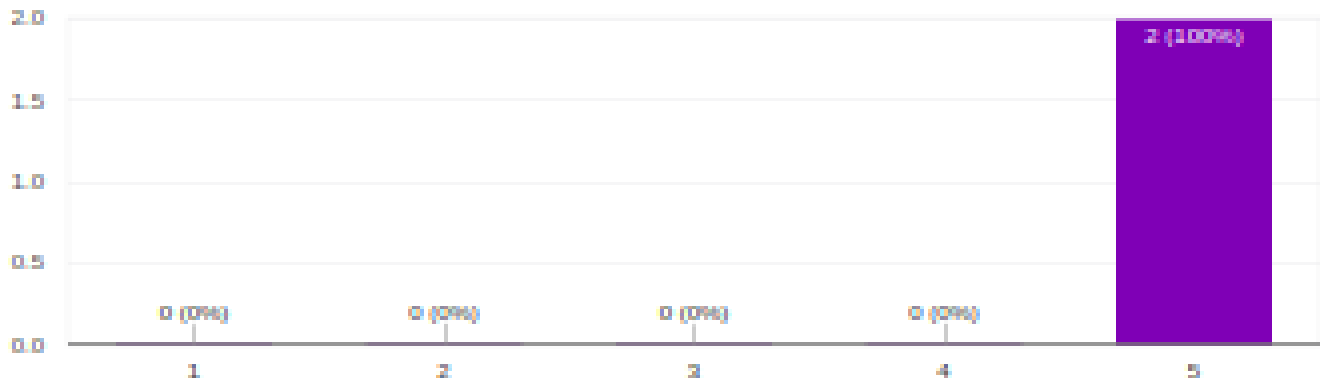


¿El sitio web es fácil de utilizar?

2 responses

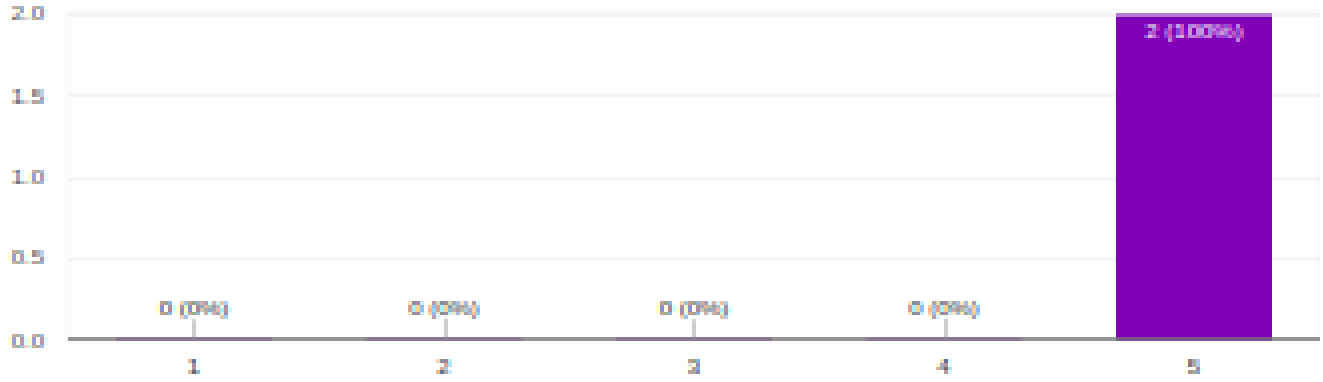
¿Ha sido fácil encontrar las características que querías en los menús?

2 responses



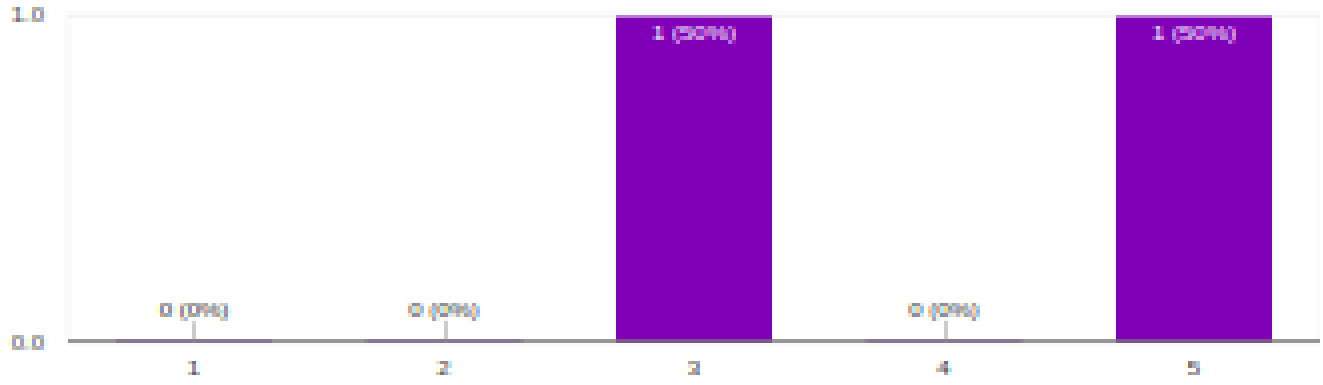
¿Los mecanismos de interacción (por ejemplo, menús desplegables, botones, punteros) son fáciles de entender y usar?

2 responses



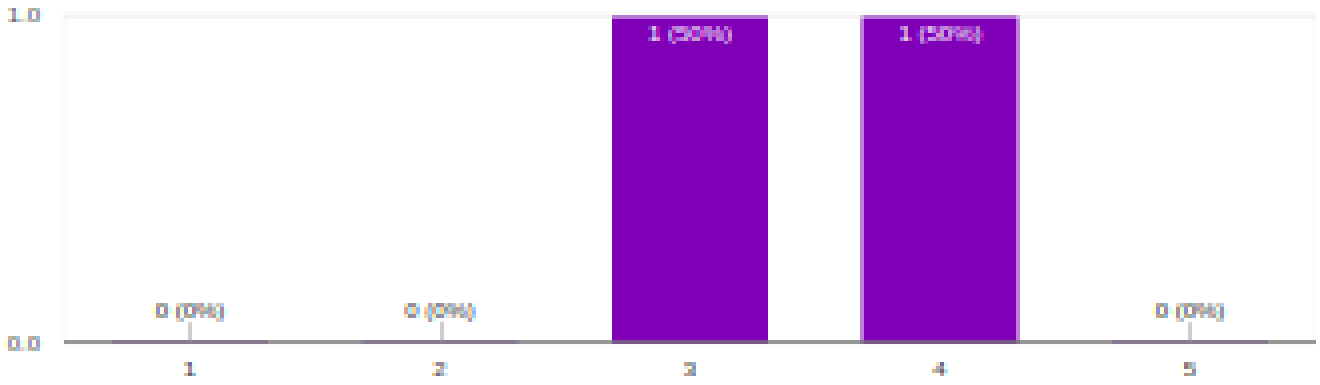
¿El texto está bien escrito y es comprensible?

2 responses



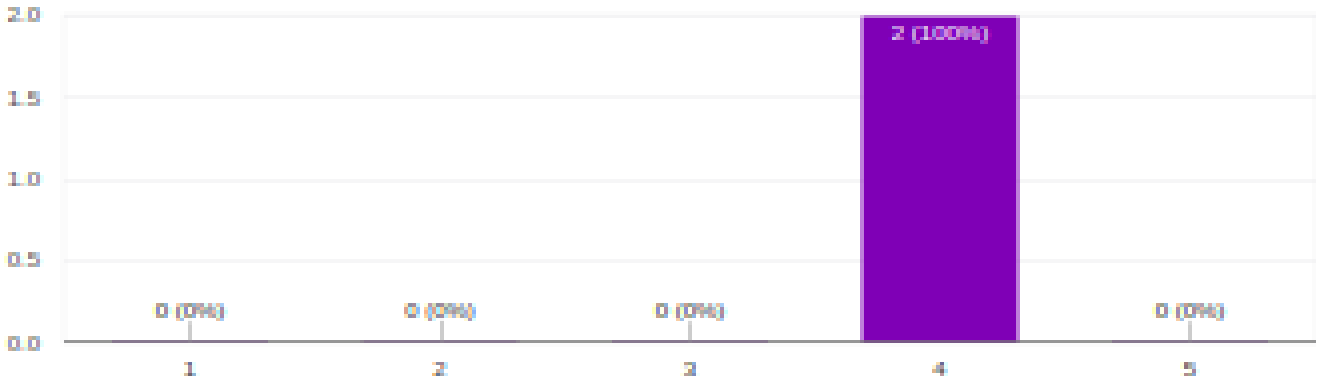
¿Te sientes cómodo con la apariencia y el sentimiento que transmite la plataforma?

2 responses



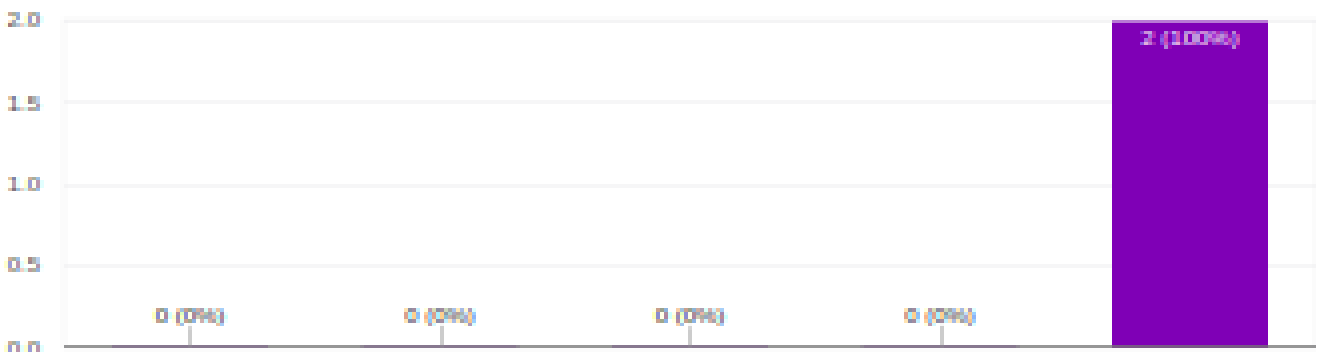
¿Las características, funciones y contenido importantes pueden usarse de forma oportuna y sin esperas?

2 responses



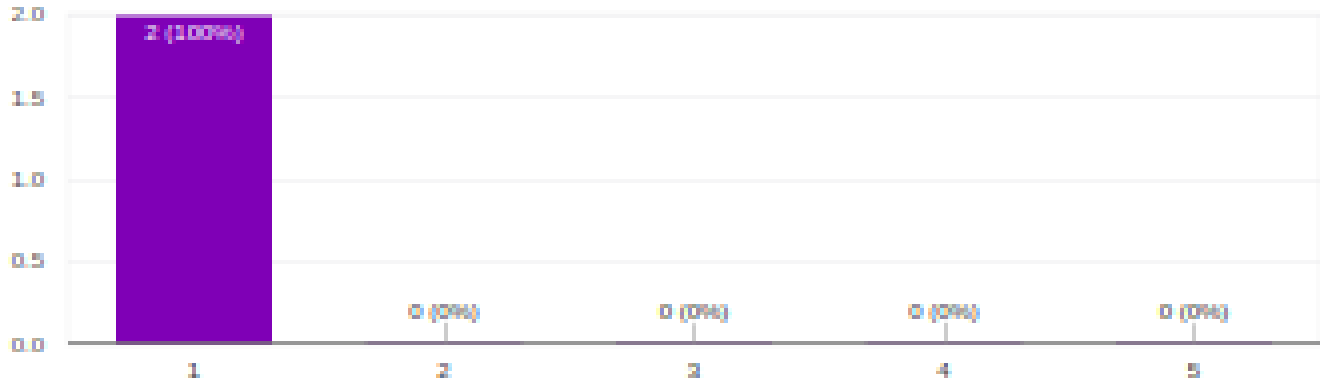
¿La plantilla, color, fuente y características relacionadas facilitan el uso?

2 responses



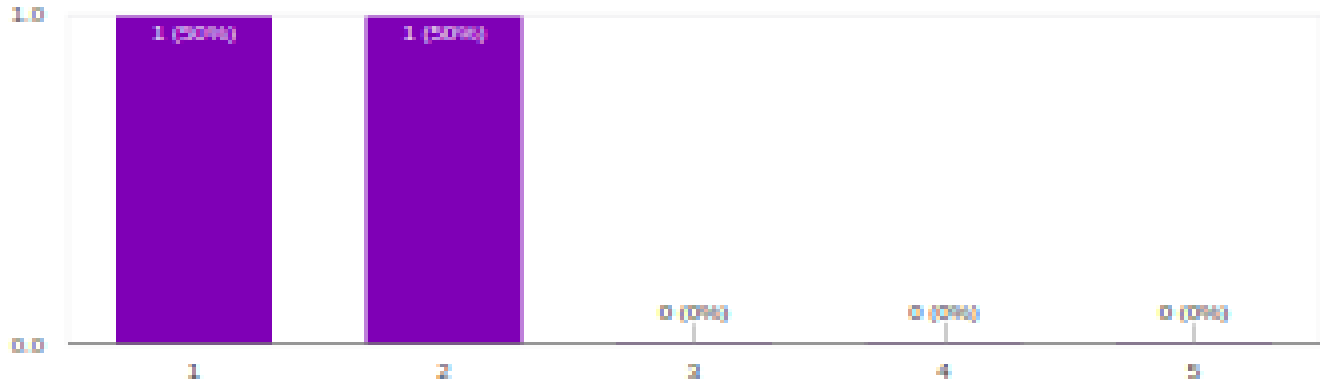
Utilizar el programa ha sido...

2 responses



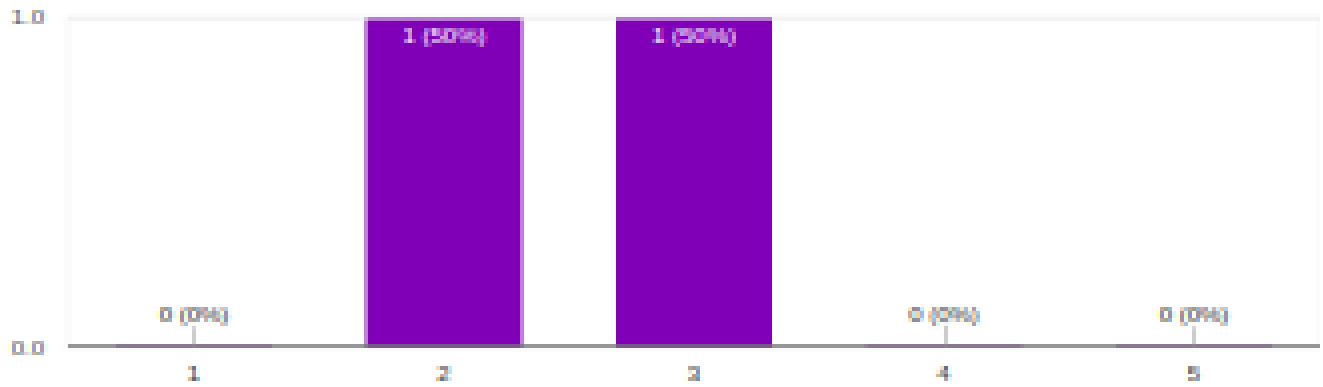
Comprender los mensajes ha sido...

2 responses



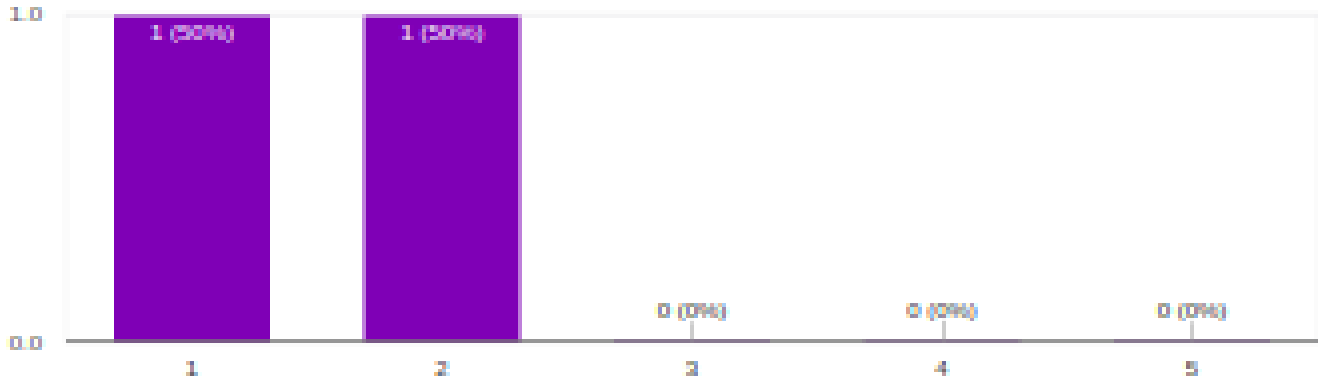
La recuperación de errores ha sido...

2 responses



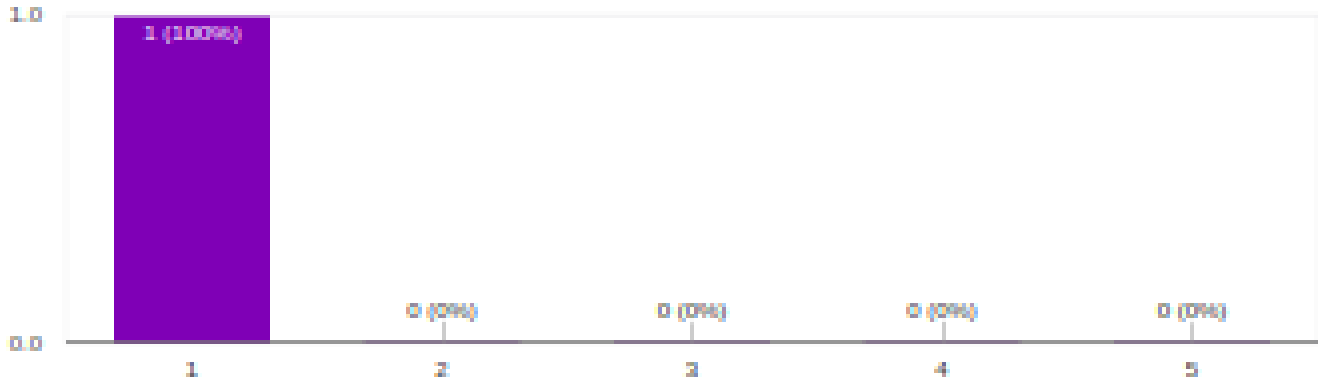
¿Es el programa es cómodo de utilizar?

2 responses



Dar respuesta a las peticiones de los pacientes ha sido...

1 response



Consultar el calendario de citas ha sido...

1 response

¿Alguna sugerencia?

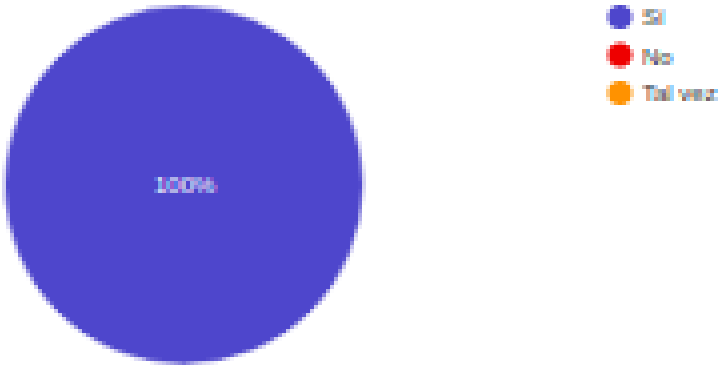
0 responses

No responses yet for this question.

Test de usabilidad de los psicólogos

¿Volverías a visitar la página web?

3 responses



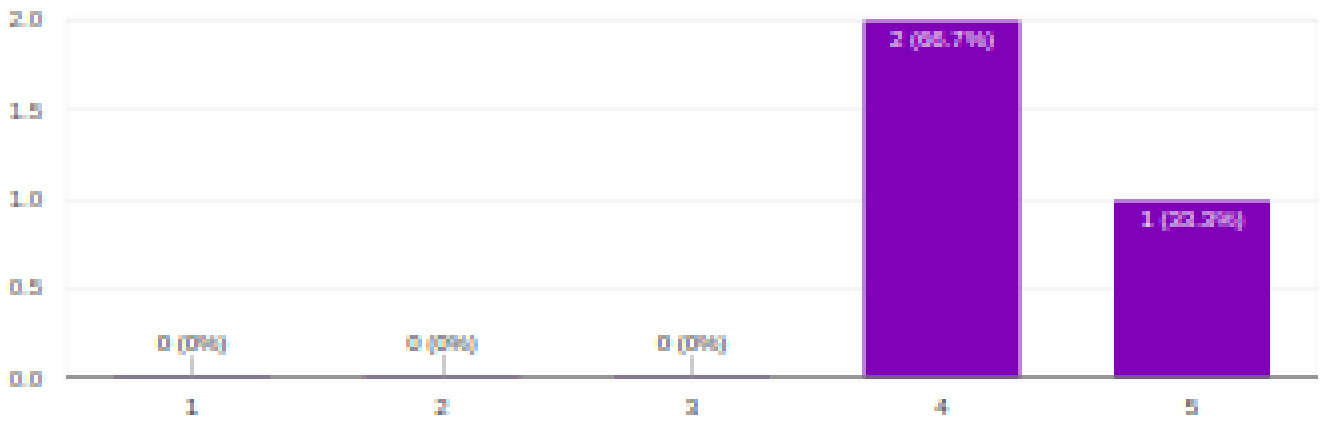
¿Cuáles son los motivos por los que volverías o no?

3 responses

Es fácil de usar y muy intuitiva
Me gusta la ayuda que presta a la hora de encontrar un psicólogo, su uso es sencillo y no me ha costado adaptarme a ella.
La idea me parece interesante

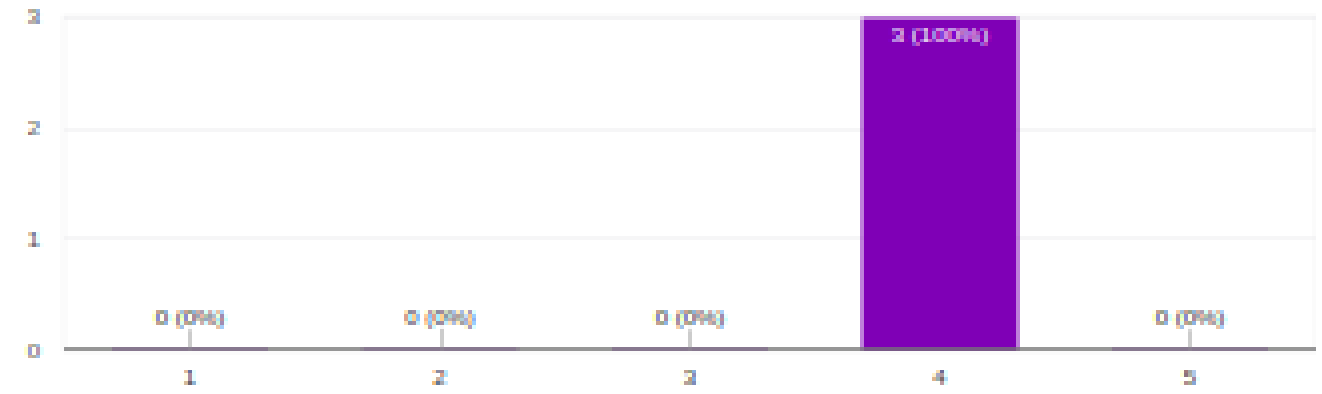
¿La información en la página web está dispuesta de forma clara?

3 responses



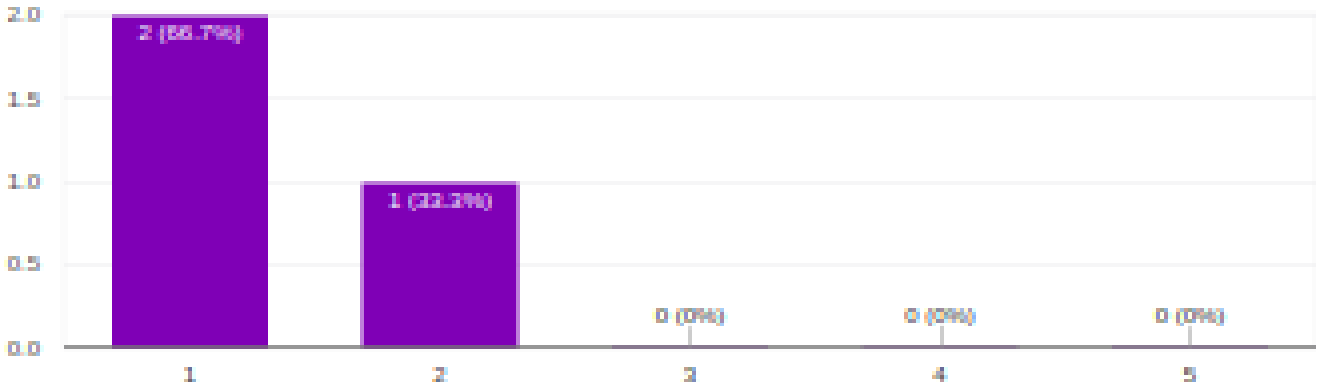
¿La navegación en la página web es fácil e intuitiva?

3 responses



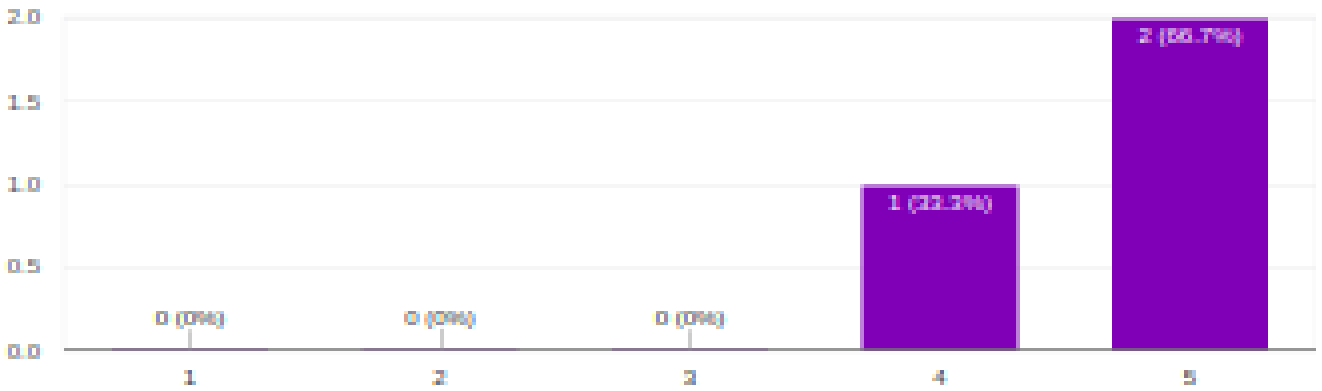
¿Necesitarías consultar un manual para el uso de la página?

3 responses



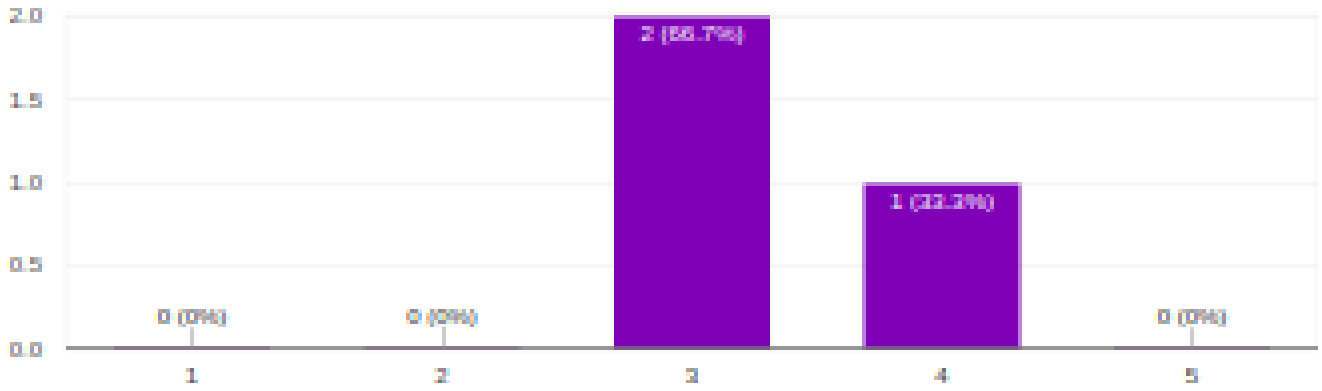
¿Encuentras lo que buscas de forma rápida en la página web?

3 responses



¿Te has sentido seguro al usar el sitio web?

3 responses

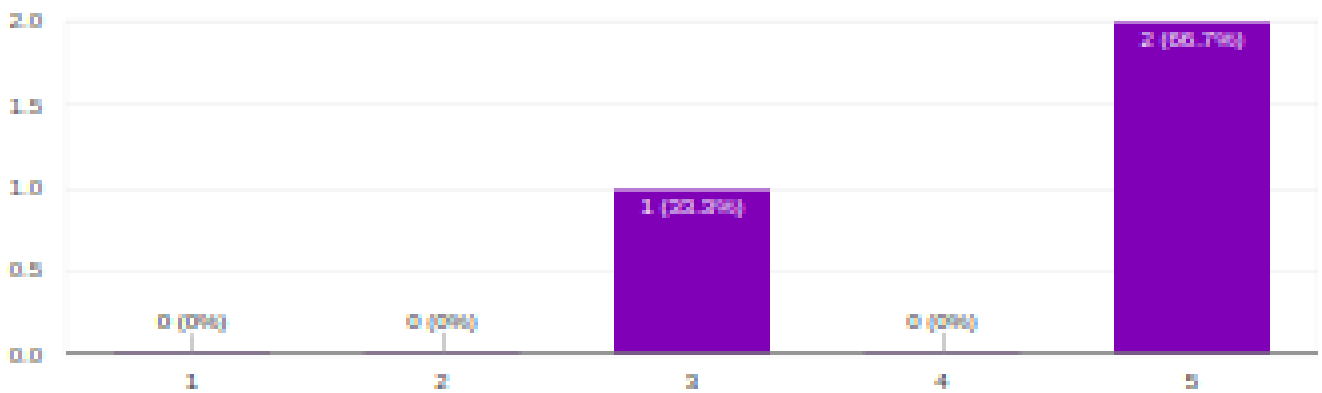


¿Los contenidos de la página web están bien estructurados?

3 responses

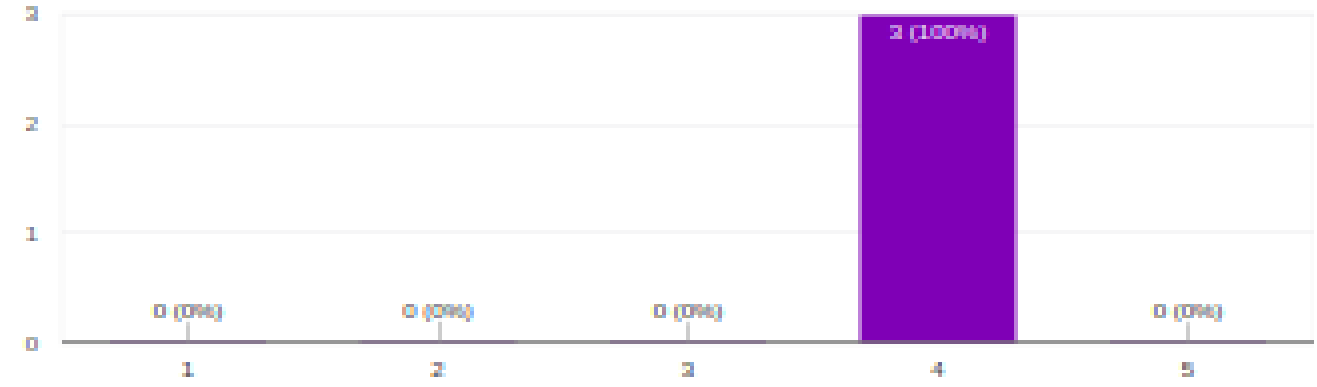
¿El sitio web es fácil de utilizar?

3 responses



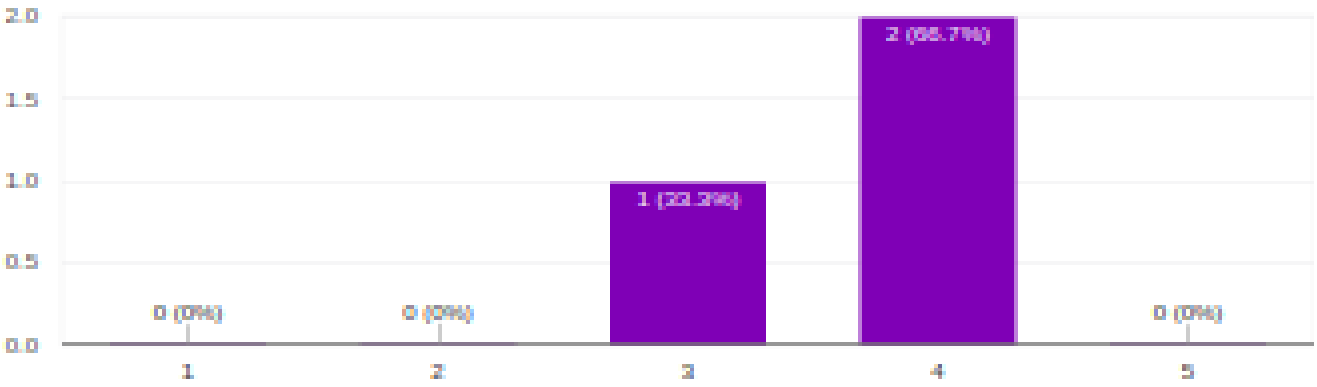
¿Ha sido fácil encontrar las características que querías en los menús?

3 responses



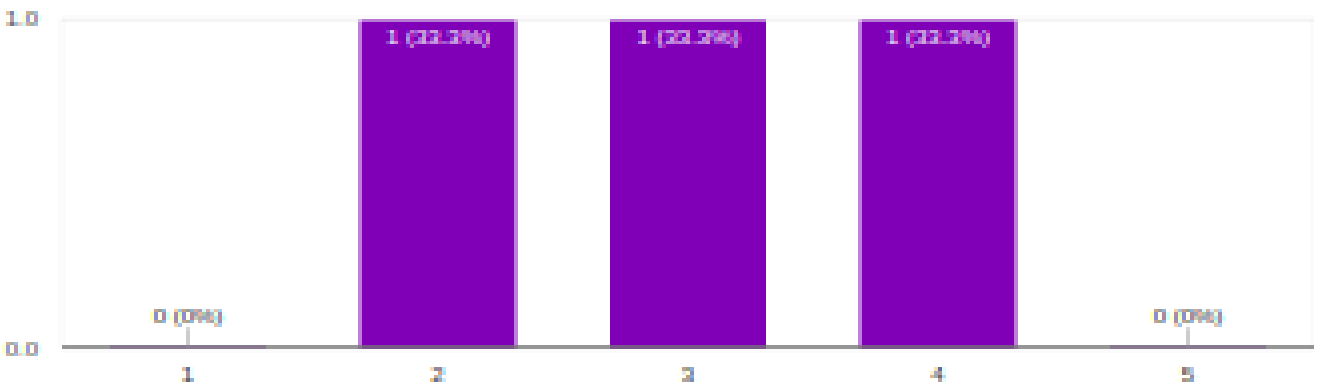
¿Los mecanismos de interacción (por ejemplo, menús desplegables, botones, punteros) son fáciles de entender y usar?

3 responses



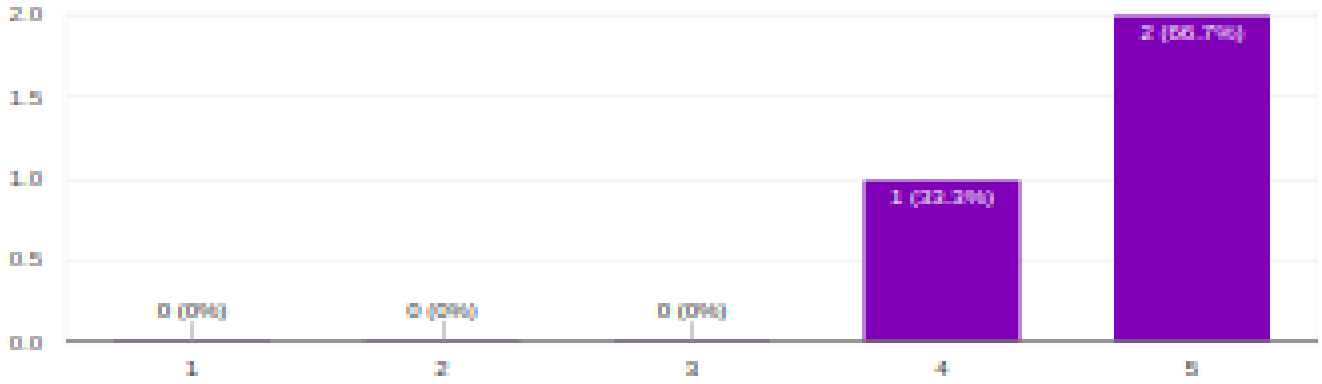
¿El texto está bien escrito y es comprensible?

3 responses



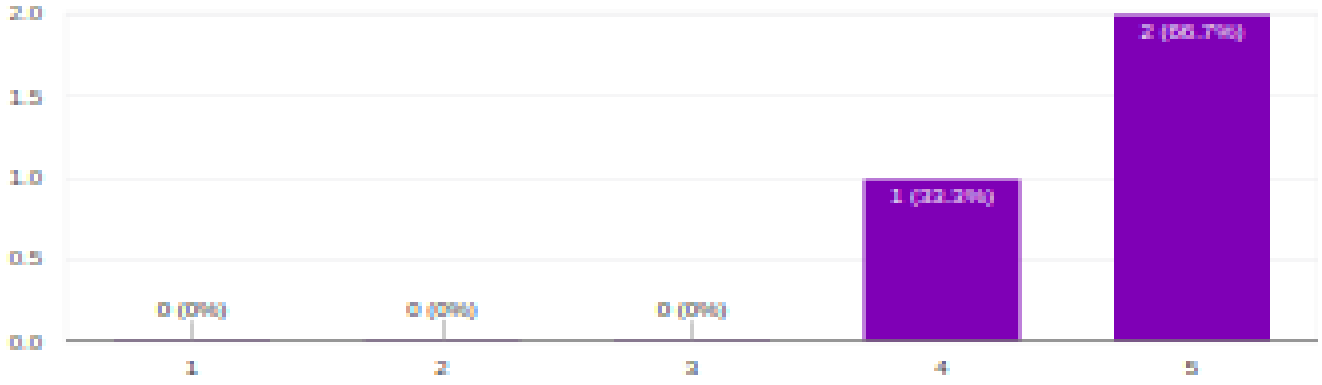
¿Te sientes cómodo con la apariencia y el sentimiento que transmite la plataforma?

3 responses



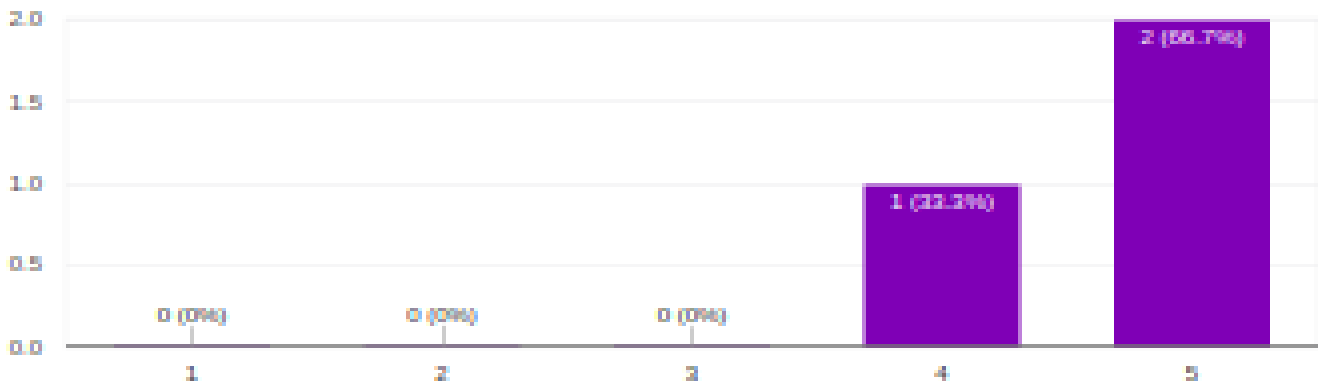
¿Las características, funciones y contenido importantes pueden usarse de forma oportuna?

3 responses



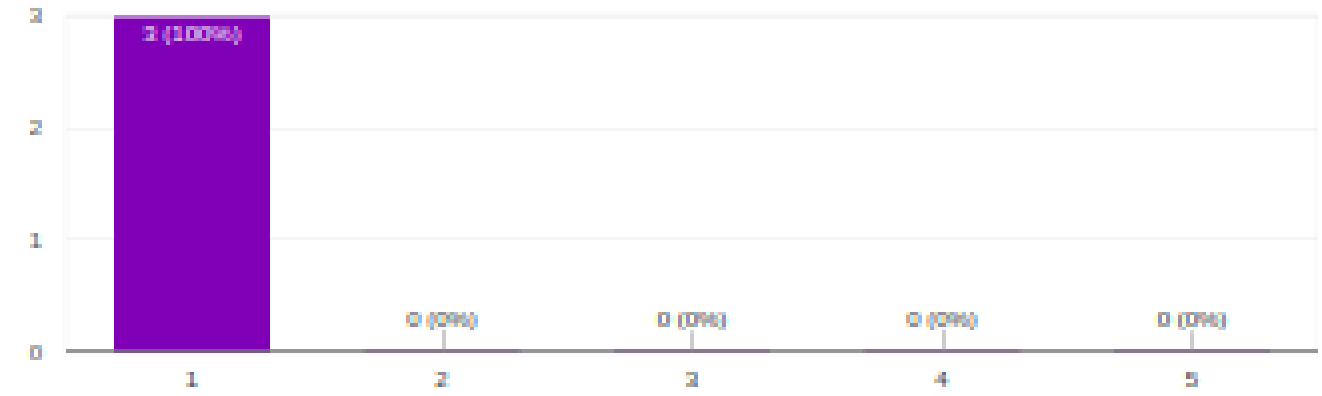
¿La plantilla, color, fuente y características relacionadas facilitan el uso?

3 responses



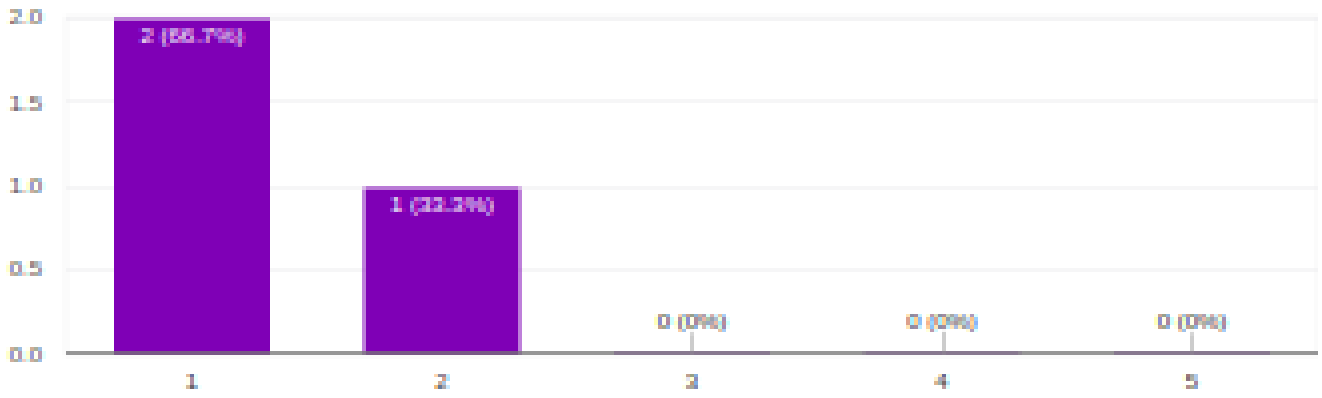
Utilizar el programa ha sido...

3 responses



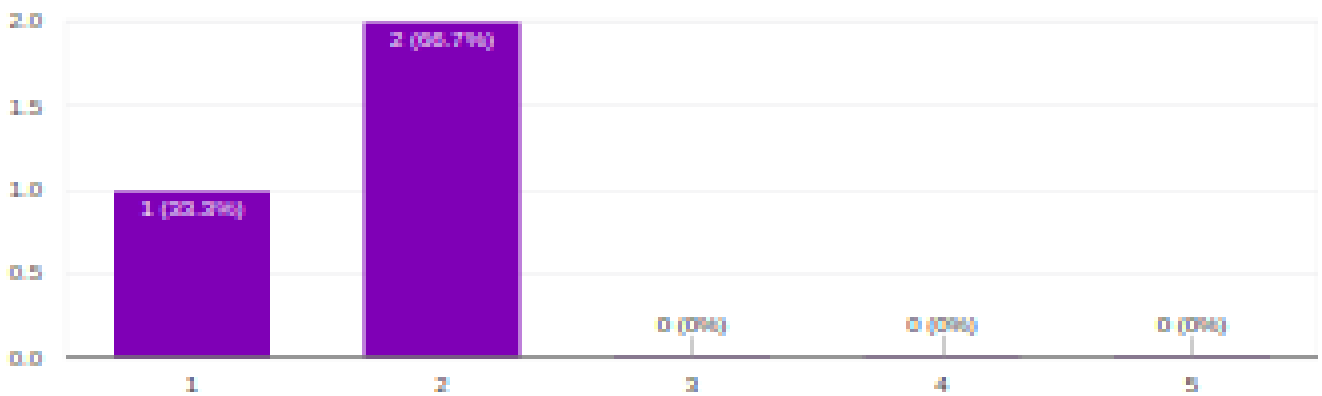
Comprender los mensajes ha sido...

3 responses



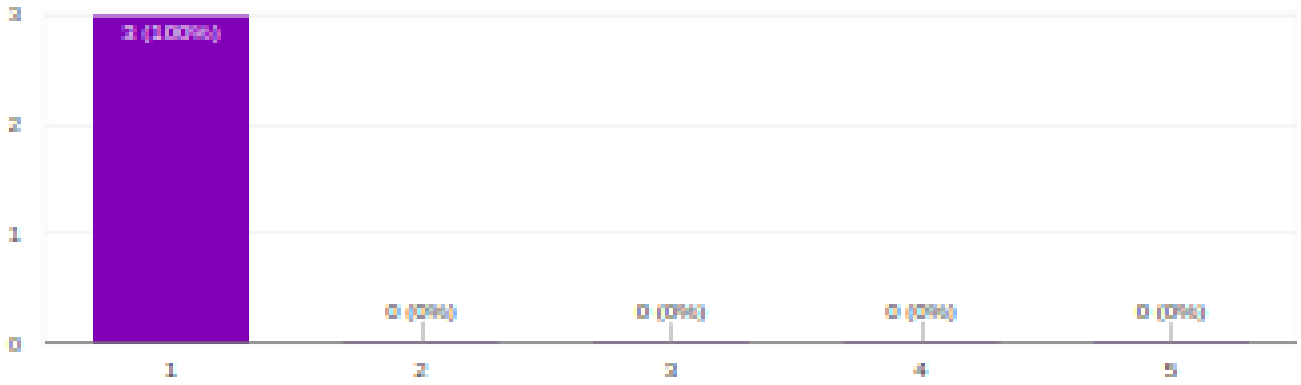
La recuperación de errores ha sido...

3 responses



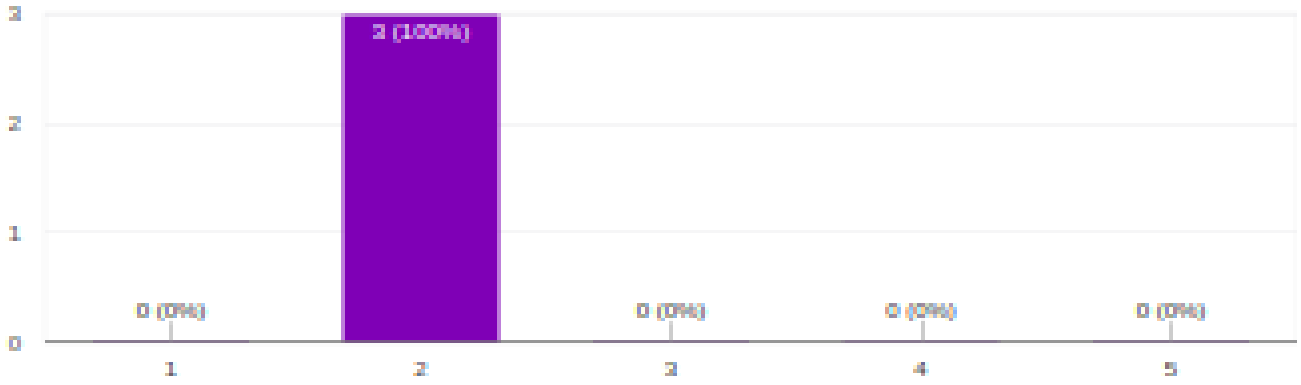
Contestar el test ha sido...

3 responses



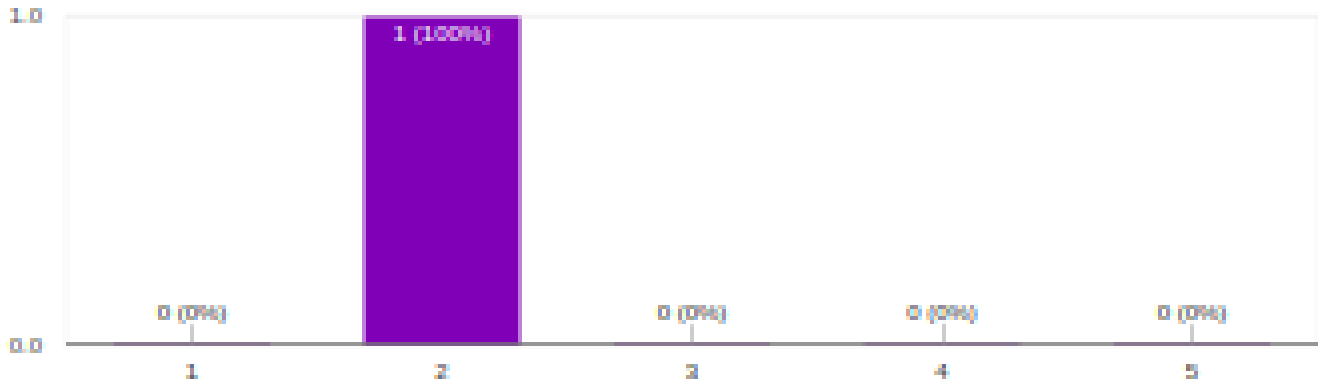
Filtrar los resultados del test ha sido...

3 responses



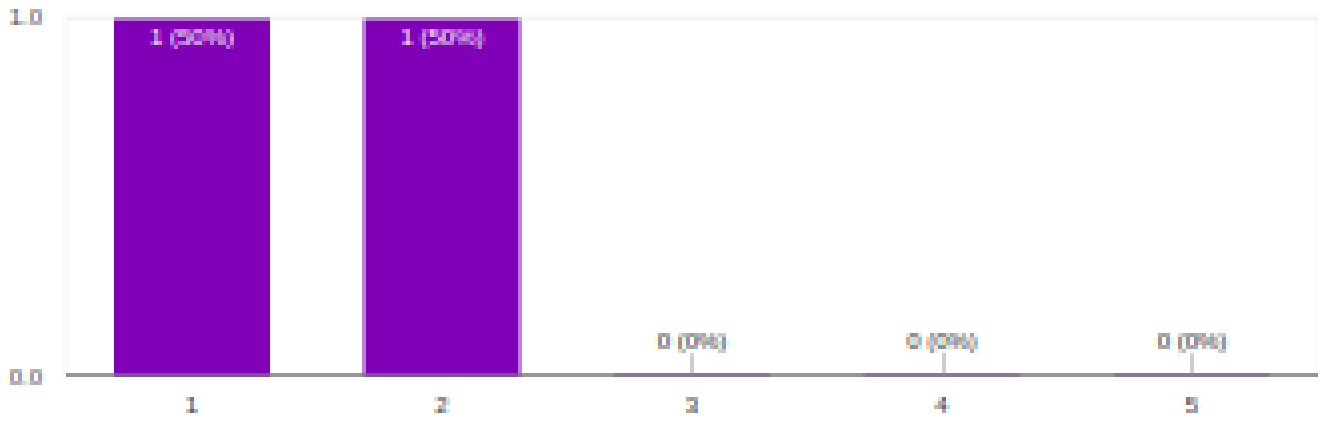
Pedir una cita al psicólogo es...

1 response



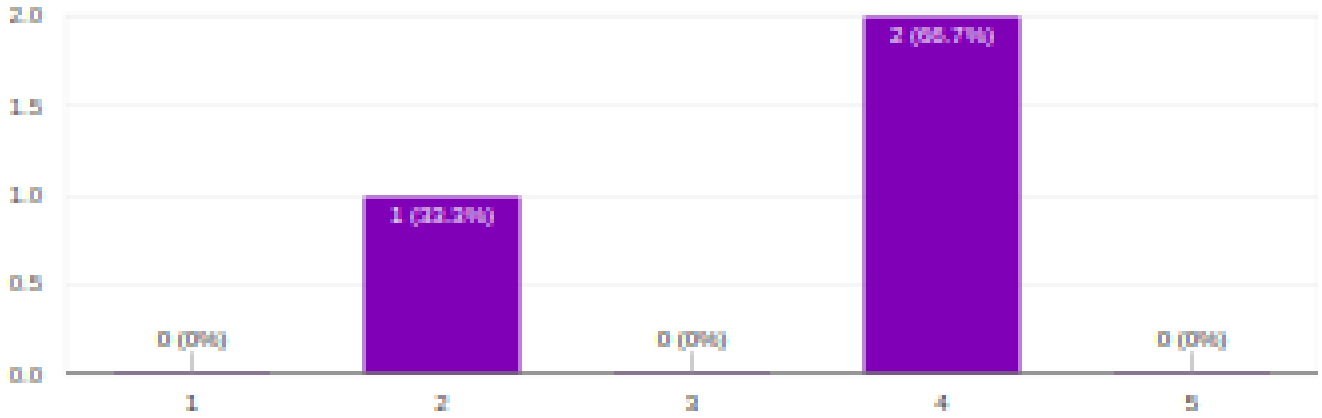
Registrarse en la plataforma es...

2 responses



¿La información mostrada sobre el psicólogo me resulta de utilidad?

3 responses



¿Alguna sugerencia?

1 response

Se ve todo muy fácil, espero que no cambie la facilidad con la que se encuentran las cosas y cumplen su función.

Anexo PI-013	**Prueba de accesibilidad**	
Página	**Errores**	**Solución**
inicio.html	El contraste entre el fondo y el elemento del contenido no es suficiente: Botón "Acceder".	Se considerará la elección de un color diferente.

Tabla A.29: Anexo PI-013

Anexo PI-013	**Prueba de accesibilidad**	
Página	**Errores**	**Solución**
pacientesResultados.html	El contraste entre el fondo y el elemento del contenido no es suficiente: Botón "Filtrar". Los atributos "id" de la página no son únicos.	Se considerará la elección de un color diferente. Se trata de los ids asociados a los containers de los psicólogos y que contienen los mismos atributos. Es el mismo elemento de contenido dispuesto de manera recursiva, por lo que no se modificará.

Tabla A.30: Anexo PI-013

A.3. Pruebas de navegación

A.3.1. Incrementos 1 y 2

Anexos PN-001 y PN-002.

A.3.2. Incremento 3

Anexo de la PN-006.

Anexo PI-013	**Prueba de accesibilidad**	
Página	**Errores**	**Solución**
cuestionarioPacientes.html	El contraste entre el fondo y el elemento del contenido no es suficiente: Botón "Sí" activado o "No" activado.	Se considerará la elección de un color diferente.

Tabla A.31: Anexo PI-013

Anexo PI-013	**Prueba de accesibilidad**	
Página	**Errores**	**Solución**
pacientesMail.html	El contraste entre el fondo y el elemento del contenido no es suficiente: Botones "Pendientes", "Aceptadas" y "Rechazadas, etiqueta con el número de peticiones de cada tipo, etiqueta de estado de la petición "Pendiente de respuesta". Texto del mensaje.	Se considerará la elección de un color diferente.
	Los atributos "id" de la página no son únicos.	Se trata de los ids asociados a cada uno de los mensajes enviados y al psicólogo remitente de los mismos. Es el mismo elemento de contenido dispuesto de manera recursiva, por lo que no se modificará.

Tabla A.32: Anexo PI-013

Anexo PI-013	**Prueba de accesibilidad**	
Página	**Errores**	**Solución**
registroPacientes.html	El contraste entre el fondo y el elemento del contenido no es suficiente: Botón "Acceder", enlace "Términos y condiciones de uso", botón "Registrar" y enlace "Registro de psicólogos"	Se considerará la elección de un color diferente para los botones, mientras que los enlaces se mantendrán con su color por defecto.

Tabla A.33: Anexo PI-013

Anexo PI-013	**Prueba de accesibilidad**	
Página	**Errores**	**Solución**
registroPsicologos.html	El contraste entre el fondo y el elemento del contenido no es suficiente: Botón "Acceder", título de cada paso y botón "Siguiente"	Se considerará la elección de un color diferente.

Tabla A.34: Anexo PI-013

Anexo PI-013	**Prueba de accesibilidad**	
Página	**Errores**	**Solución**
psicologosMail.html	El contraste entre el fondo y el elemento del contenido no es suficiente: Mismos botones y etiquetas que en PacientesMail.html, botón "Aceptar" y título "Diagnóstico"	Se considerará la elección de un color diferente.
	Los atributos "id" de la página no son únicos.	Se trata de los ids asociados a cada uno de los mensajes enviados y al diagnóstico del paciente remitente de los mismos. Es el mismo elemento de contenido dispuesto de manera recursiva, por lo que no se modificará.

Tabla A.35: Anexo PI-013

Anexo PI-013	**Prueba de accesibilidad**	
Página	**Errores**	**Solución**
perfilPsicologo.html	El contraste entre el fondo y el elemento del contenido no es suficiente: Títulos "Formación", "Experiencia" y "Especialidad" y; fecha y botón "Responder" de cada comentario	Se considerará la elección de un color diferente.
	Los atributos "id" de la página no son únicos.	Se trata de los ids asociados a cada uno de los comentarios. Es el mismo elemento de contenido dispuesto de manera recursiva, por lo que no se modificará.

Tabla A.36: Anexo PI-013

PN-001: USN-003, USN-004, USN-008, USN-009	**Evaluación**
¿La USN se logra en su totalidad sin error?	Sí
¿Todo nodo de navegación (definido por una USN) se alcanza dentro del contexto de las rutas de navegación definidas por la USN?	Sí
¿Existe un mecanismo (distinto a la flecha "retroceso" del navegador) para regresar al nodo de navegación anterior y al comienzo de la ruta de navegación?	No aplica - La profundidad de la navegación es de 1
¿Los mecanismos de navegación dentro de un gran nodo de navegación (es decir, una página web grande) funcionan de manera adecuada?	Sí
Si una función debe ejecutarse en un nodo y el usuario elige no proporcionar entrada, ¿el resto de la USN puede completarse?	Sí
Si una función debe ejecutarse en un nodo y ocurre un error en el procesamiento de la función, ¿la USN puede completarse?	Sí
¿Los nombres de nodo son significativos para los usuarios finales?	Sí
¿El usuario entiende su ubicación dentro de la arquitectura de contenido conforme se ejecuta la USN?	Sí

Tabla A.37: PN-001

PN-002: USN-001, USN-002, USN-006, USN-010	**Evaluación**
¿La USN se logra en su totalidad sin error?	Sí
¿Todo nodo de navegación (definido por una USN) se alcanza dentro del contexto de las rutas de navegación definidas por la USN?	Sí
¿Existe un mecanismo (distinto a la flecha "retroceso" del navegador) para regresar al nodo de navegación anterior y al comienzo de la ruta de navegación?	No aplica - La profundidad de navegación es de 1
¿Los mecanismos de navegación dentro de un gran nodo de navegación (es decir, una página web grande) funcionan de manera adecuada?	Sí
Si una función debe ejecutarse en un nodo y el usuario elige no proporcionar entrada, ¿el resto de la USN puede completarse?	Sí
Si una función debe ejecutarse en un nodo y ocurre un error en el procesamiento de la función, ¿la USN puede completarse?	Sí
¿Los nombres de nodo son significativos para los usuarios finales?	Sí
¿El usuario entiende su ubicación dentro de la arquitectura de contenido conforme se ejecuta la USN?	Sí

Tabla A.38: Anexo PN-002

PN-004: USN-005, USN-007, USN-011, USN-012	**Evaluación**
¿La USN se logra en su totalidad sin error?	Sí
¿Todo nodo de navegación (definido por una USN) se alcanza dentro del contexto de las rutas de navegación definidas por la USN?	Sí
¿Existe un mecanismo (distinto a la flecha "retroceso" del navegador) para regresar al nodo de navegación anterior y al comienzo de la ruta de navegación?	No aplica – La profundidad de la navegación es de 1
¿Los mecanismos de navegación dentro de un gran nodo de navegación (es decir, una página web grande) funcionan de manera adecuada?	Sí
Si una función debe ejecutarse en un nodo y el usuario elige no proporcionar entrada, ¿el resto de la USN puede completarse?	Sí
Si una función debe ejecutarse en un nodo y ocurre un error en el procesamiento de la función, ¿la USN puede completarse?	Sí
¿Los nombres de nodo son significativos para los usuarios finales?	Sí
¿El usuario entiende su ubicación dentro de la arquitectura de contenido conforme se ejecuta la USN?	Sí

Tabla A.39: Anexo PN-004

Apéndice B

Manual de instalación

El proyecto ha sido desarrollado en Linux por lo que la instalación deberá hacerse a través de ese sistema operativo.

Se deben seguir los siguientes pasos:

1. Se abre la terminal.
2. Sitúate sobre el directorio del proyecto `em_web` con el comando `cd`.
3. **Instalación de la aplicación**

   a) Se instala el gestor de paquetes NPM con el siguiente comando:

   ```
   $ sudo apt-get install npm
   ```

   b) Se instalan todas las dependencias de la aplicación que se encuentran en la carpeta local `node_modules`, y que están listadas como dependencias en el archivo `package.json`:

   ```
   $ sudo npm install
   ```

4. **Instalación del gestor de bases de datos**

   a) Se instala MongoDB con:

   ```
   $ sudo apt-get install -y mongodb-org
   ```

   b) Para correr MongoDB se utiliza:

   ```
   $ mongo
   ```

   c) Se crea la base de datos de Emozio con el comando que aparece a continuación. Una vez creada la base de datos, se puede acceder a ella con el mismo comando.

   ```
   $ use emozio
   ```

   d) Se cargan los datos en la base de datos ejecutando:

```
$ mongoimport -dab emozio -collection pacientes -type json -file
    server/sports/seed.json -jsonArray --drop

$ mongoimport -dab emozio -collection patologias -type json -file
    server/sports/seed.json -jsonArray --drop

$ mongoimport -dab emozio -collection psicologos -type json -file
    server/sports/seed.json -jsonArray -drop

$ mongoimport -dab emozio -collection mensajes -type json -file
    server/sports/seed.json -jsonArray --drop
```

Modificación del proyecto Para modificar el proyecto, se tendrá que abrir con un editor de texto. Si se desea utilizar el gestor de bases de datos Robomongo se deberá descargar a través de su página web: `https://robomongo.org/download`

Correr la aplicación

1. Para correr la aplicación, se deberá ejecutar el comando:

   ```
   $ npm start
   ```

2. Como nuestro servidor se encuentra escuchando el puerto 8000, para poder visualizar la aplicación en el navegador, se tendrá que indicar la dirección `localhost:8000`.

Bibliografía

[1] Acerca de Node.js. https://nodejs.org/es/about/, Consultado de 29 de enero del 2018.

[2] Arquitectura AngularJS. https://angular.io/guide/architecture, Consultado de 29 de enero del 2018.

[3] CSS. https://developer.mozilla.org/es/docs/Web/CSS, Consultado de 29 de enero del 2018.

[4] Documentación de AngularJS. https://angular.io/docs, Consultado de 29 de enero del 2018.

[5] Github Semantic UI. https://github.com/Semantic-Org/Semantic-UI, Consultado de 29 de enero del 2018.

[6] HTML. https://developer.mozilla.org/es/docs/Web/HTML, Consultado de 29 de enero del 2018.

[7] ISO 9241-171:2008 Ergonomics of human-system interaction - Part 171: Guidance on software accessibility. https://www.iso.org/standard/39080.html, Consultado de 29 de enero del 2018.

[8] JavaScript. https://developer.mozilla.org/es/docs/Web/JavaScript, Consultado de 29 de enero del 2018.

[9] Objetos globales: Promise. https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Promise, Consultado de 29 de enero del 2018.

[10] Places de Google Maps JavaScript API. https://developers.google.com/maps/documentation/javascript/places-autocomplete?hl=es-419, Consultado de 29 de enero del 2018.

[11] Promise. https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Usar_promesas, Consultado de 29 de enero del 2018.

[12] Página oficial de Bootstrap. https://nodejs.org/es/about/, Consultado de 29 de enero del 2018.

[13] Página oficial de Brackets. http://brackets.io/, Consultado de 29 de enero del 2018.

[14] Página oficial de ExpressJS. http://expressjs.com/es/, Consultado de 29 de enero del 2018.

[15] Página oficial de FullCalendar. https://fullcalendar.io/, Consultado de 29 de enero del 2018.

[16] Página oficial de Git. https://git-scm.com/, Consultado de 29 de enero del 2018.

[17] Página oficial de GitHub. https://github.com/, Consultado de 29 de enero del 2018.

[18] Página oficial de GruntJS. https://gruntjs.com/, Consultado de 29 de enero del 2018.

[19] Página oficial de LaTeX. https://www.latex-project.org/, Consultado de 29 de enero del 2018.

[20] Página oficial de LibreOffice. https://es.libreoffice.org/, Consultado de 29 de enero del 2018.

[21] Página oficial de Mongoose. http://mongoosejs.com/, Consultado de 29 de enero del 2018.

[22] Página oficial de NPMJS. https://www.npmjs.com/, Consultado de 29 de enero del 2018.

[23] Página oficial de PassporJS. http://www.passportjs.org/, Consultado de 29 de enero del 2018.

[24] Página oficial de Pencil. https://pencil.evolus.vn/, Consultado de 29 de enero del 2018.

[25] Página oficial de Robomongo. https://robomongo.org/, Consultado de 29 de enero del 2018.

[26] Página oficial de StarUML. http://staruml.io/, Consultado de 29 de enero del 2018.

[27] UNE 139803:2012 Requisitos de accesibilidad para contenidos en la Web. http://administracionelectronica.gob.es/pae_Home/pae_Estrategias/pae_Accesibilidad/pae_normativa/pae_eInclusion_Normas_Accesibilidad.html#.WBbsEYVwZZ0, Consultado de 29 de enero del 2018.

[28] AngularJS in Patterns. https://github.com/mgechev/angularjs-in-patterns, Consultado el: 29 de Enero del 2018.

[29] Bootstrap Grids. w3schools. https://www.w3schools.com/bootstrap/bootstrap_grid_basic.asp, Consultado el: 29 de Enero del 2018.

[30] Data binding. w3schools. https://www.w3schools.com/angular/angular_databinding.asp, Consultado el: 29 de Enero del 2018.

[31] Dependency Injection in JavaScript. https://dzone.com/articles/dependency-injection-0, Consultado el: 29 de Enero del 2018.

[32] Lato. Google Fonts. https://fonts.google.com/specimen/Lato, Consultado el: 29 de Enero del 2018.

[33] Merriweather. Google Fonts. https://fonts.google.com/specimen/Merriweather, Consultado el: 29 de Enero del 2018.

[34] Page Controller. Microsoft. https://msdn.microsoft.com/en-us/library/ff649595.aspx, Consultado el: 29 de Enero del 2018.

[35] Understanding REST. https://spring.io/understanding/REST, Consultado el: 29 de Enero del 2018.

[36] Thorén C. *Nordic Guidelines for Computer Accessibility*. Nordic Cooperation on Disability, second edition, 1998.

[37] George Field. *Chromatics: Or, the Analogy, Harmony, and Philosophy of Colours*. David Bogue, Fleet Street, 1845.

[38] Ekberg J. *Un paso adelante: Diseño para todos. Proyecto INCLUDE.* CEAPAT-IMSERSO, Madrid., 2000.

[39] Ivan Beschastnikh Keheliya Gallaba, Ali Mesbah. *Don't Call Us, We'll Call You: Characterizing Callbacks in Javascript*. Empirical Software Engineering and Measurement (ESEM), 2015 ACM/IEEE International Symposium on, 2015.

[40] MongoDB. The MEAN Stack: MongoDB, ExpressJS, AngularJS and Node.js. https://www.mongodb.com/blog/post/the-mean-stack-mongodb-expressjs-angularjs-and, Consultado de 29 de enero del 2018.

[41] MongoDB. Reinventando la gestión de datos. `https://www.mongodb.com/es`, Consultado de 29 de enero del 2018.

[42] Fernando Monteiro. *Learning Single-page Web Application Development.* PACKT PUBLISHING, 2014.

[43] Afra Pascual y Jesús Lorés Mª Paula González. Evaluación heurística. Universitat de Lleida. `http://w.aipo.es/libro/pdf/15-Evaluacion-Heuristica.pdf`, Consultado de 29 de enero del 2018.

[44] Addy Osmani. *Learning JavaScript Design Patterns.* O'Reilly, 2012.

[45] Stoyan Stefanov. *JavaScript Patterns.* O'Reilly, 2010.