**Grupo 11: Andrea Juliana Paredes Valderrama 201510631**

**Cristian Camilo Novoa Avellaneda 201513389**

**EXPERIMENTO 3**

**Mantenibilidad**

La mantenibilidad en la arquitectura de software es un atributo de calidad indispensable que permite a futuro el fácil manejo y mantenimiento de un proyecto. En este proyecto en particular, los patrones de diseño utilizados como mecanismo para el mejoramiento de la mantenibilidad de nuestro fueron Observer y State. Dichos patrones fueron escogidos debido a que clasifican como patrones de comportamiento los cuales no solo describen estructuras de relación entre las clases, sino también permiten la comunicación entre ellas cuando hay un cambio de comportamiento. Los patrones escogidos fueron implementados como se describe a continuación:

* **Observer:**

Este patrón fue implementado de modo que ayudase en la asignación de responsabilidades relacionadas con la notificación de emergencias de un paciente, de tal forma que si el paciente cambia su estado se realice las acciones necesarias en el sistema de forma independiente. Específicamente, fueron creados dos clases de observadores que extienden de una clase padre (Observador), y que se asignaron a cada uno de los pacientes del sistema. El primero, MedicoObserver se encarga de notificar a cada uno de los médicos del paciente cuando el estado cambia a rojo o amarillo, y dependiendo del caso envía una notificación con mensajes diferentes a los médicos. El segundo es EmergenciaObserver, este observador se encarga únicamente de crear una emergencia en caso de que el estado del paciente cambie a rojo y guardarla en el historial médico del paciente. De esta forma, cada observador tiene una responsabilidad diferente cuando se produce un cambio relevante en el paciente y así cumplir con los requerimientos funcionales del sistema independientemente.

* **State:**

Este patrón fue elegido ya que es de gran utilidad para definir de forma independiente los diferentes estados que un objeto puede tener. En nuestro caso, se implementó para definir los diferentes estados de salud de un paciente: verde, amarillo o rojo. Por esta razón, cada uno de los estados fue creado como clases diferentes las cuales implementan una interfaz común llamada State. En dicha interfaz se definió la signatura de los métodos requeridos por los estados entre los cuales se destaca *definirEstado*. Este es el método encargado de analizar los valores de las mediciones hechas por los sensores y, de acuerdo a los rangos de valores establecidos para cada nivel de salud, modificar el estado del paciente de ser necesario y almacenar el motivo por el cual se realizó el cambio. De esta forma, este patrón facilita la modificabilidad del proyecto en caso de que se requiera modificar la política de estados de un paciente sin tener que alterar más código que el de las clases State.

En cuanto a la comparación entre la modificabilidad del proyecto con y sin la implementación de patrones de diseño, se estima que con los patrones de diseño la modificabilidad aumenta. Esto gracias a que se reduce el acoplamiento entre las clases y se aumenta la especificidad en las responsabilidades de las mismas.

Acoplamiento = #relaciones totales excepto herencia / # de relaciones totales

A1: Acoplamiento sin patrones de diseño

A2: Acoplamiento con patrones de diseño

A1 = 9/9 = 1

A2 = 9/12 =0,75

Comentarios críticos de los usuarios:

Persona 1:

* La primera vista de la página no es muy cómoda debido al contraste.
* Texto más detallado
* El login comete el mismo error que la vista anterior

Persona 2:

* Hacer funcionalidad en los botones de redes si se van a dejar
* Corregir el color del logo para adecuarlo a la estética de toda la página

Persona 3:

* Corregir la pantalla de login para hacerlo parecer más de medicina.
* Pequeña corrección del logo.

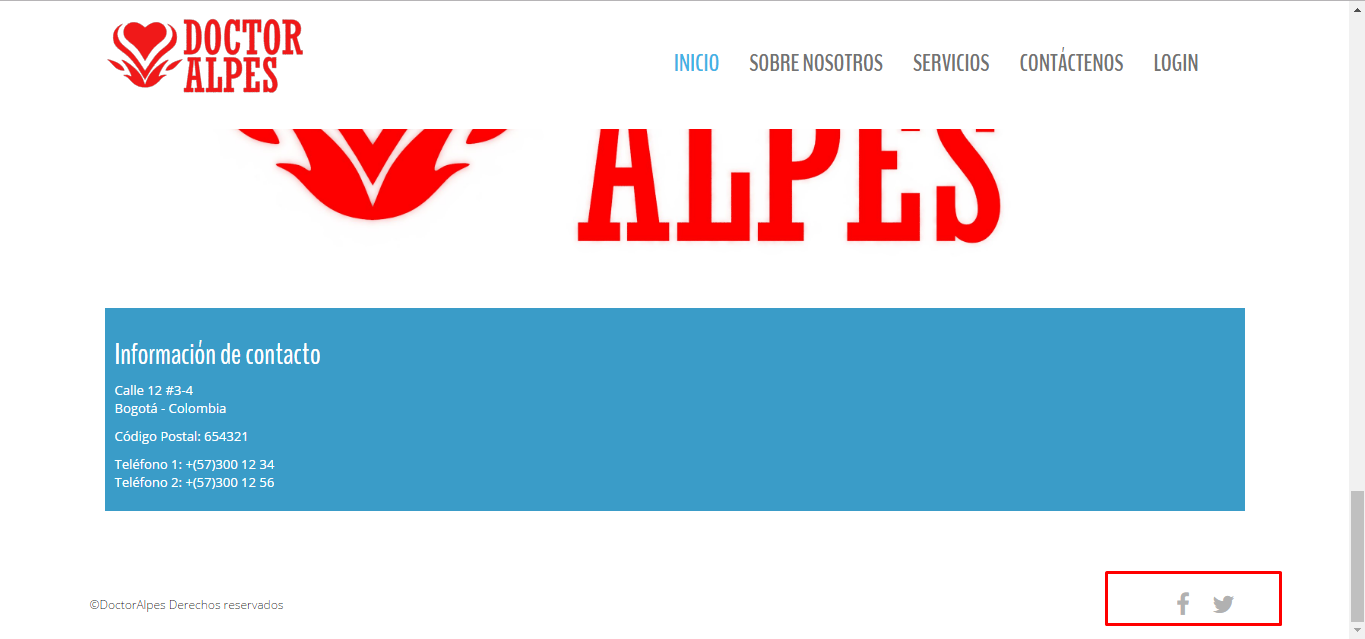
Persona 4:

* Usar imágenes propias.
* La primera imagen que se ve está muy cargada y no contrastan las letras con el fondo.
* Agrandar un poco la imagen del login para llenar espacio.

Persona 5:

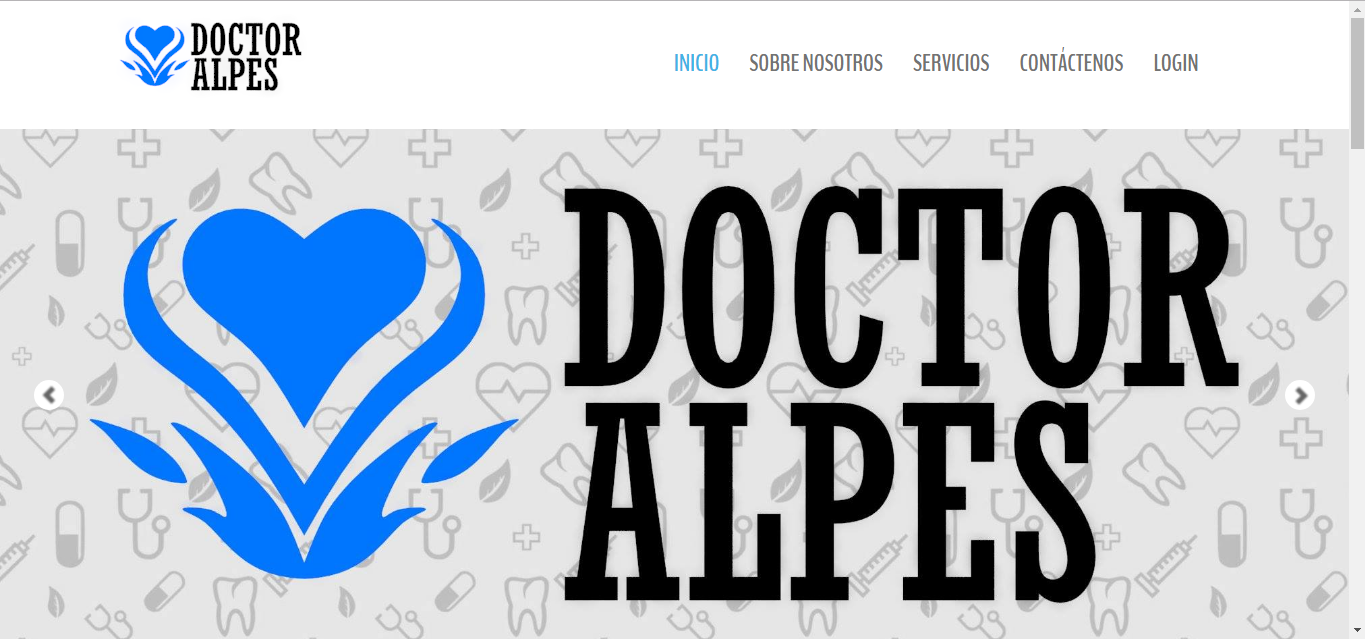
* Cambiar la pantalla de login por una con mejor contraste.
* Funcionalidad de botones de las redes.
* Primera imagen del inicio.

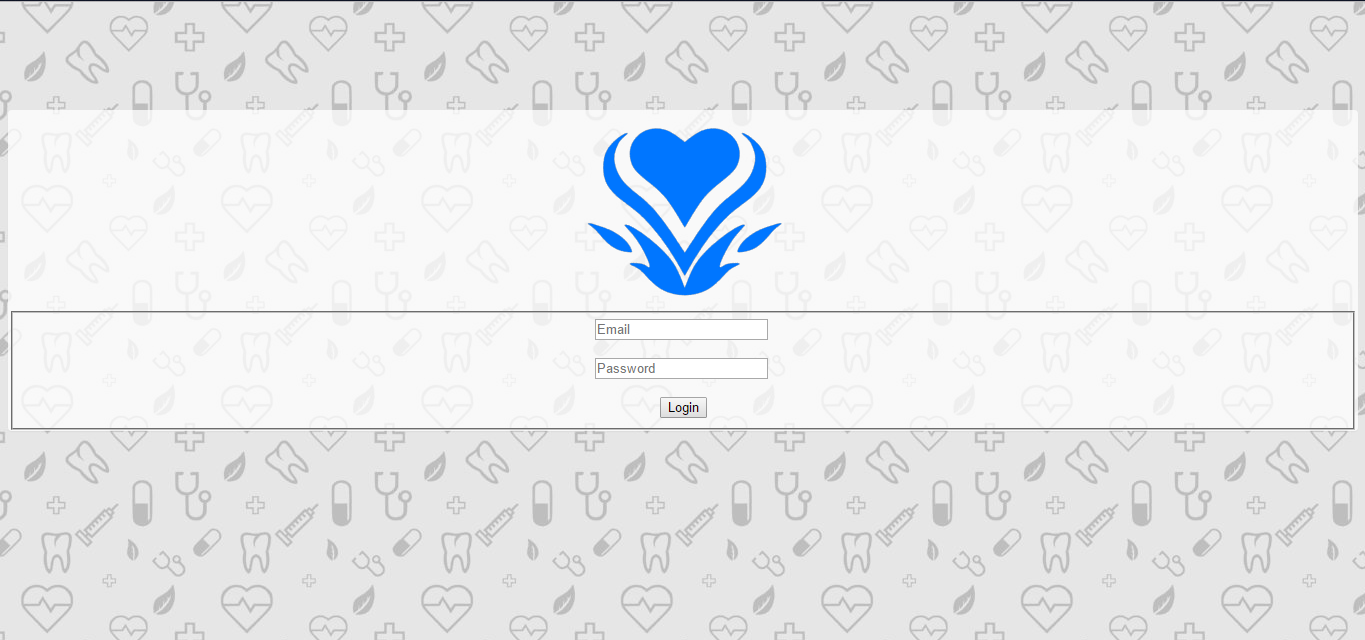
Antes:





Después:





Se usa correctamente la ley de fits, las opciones importantes se dejaron en los bordes de la web y con un tamaño grande para hacer click de manera más fácil.

Se usó la grilla en algunas vistas, con el fin de separar de mejor la manera la información.

El logo es presentado en cada una de las vistas, con un diseño acorde a la página.

Tiene enlaces significativos para acceder a sus redes sociales, de manera rápida y cómoda.