**Documentación Proyecto**

**Plataforma de Gestión de Pacientes**

**Hospital General de Barranquilla**

**NRC:** 2126

**Equipo No.** 7

**Integrantes:**

*Juan Andrés Ríos Montealegre*

*Dainer José Gutiérrez Vergara*

*Claudia Emilia Bedoya Madrid*

*Sandra Liliana Nomesque Suárez*

*Mónica Carolina Delgado Molina*

**Descripción de roles del equipo** (*Sprint 1*)

|  |  |  |  |
| --- | --- | --- | --- |
| Rol | Integrante | Descripción | Tareas |
| Product owner | *Juan Andrés Ríos Montealegre* | Entidad que solicita el desarrollo. | Decidir qué trabajo necesita hacerse y maximizar el valor del proyecto. |
| Scrum Master | *Helmer Segundo Fuentes Alvarado* | Supervisión del cumplimiento de las actividades. | Supervisión semanal de tareas, ayudando al equipo y a la organización. Traslada la visión del proyecto al equipo. |
| Development team | *Juan Andrés Ríos Montealegre* | Desarrollador. | En cada Sprint construye y entrega un incremento del producto. |
| Development team | *Dainer José Gutiérrez Vergara* | Desarrollador. | En cada Sprint, construye y entrega un incremento del producto |
| Development team | *Claudia Emilia Bedoya Madrid* | Desarrollador. | En cada Sprint, construye y entrega un incremento del producto |
| Development team | *Sandra Liliana Nomesque Suárez* | Desarrollador. | En cada Sprint, construye y entrega un incremento del producto |
| Development team | *Mónica Carolina Delgado Molina* | Desarrollador. | En cada Sprint, construye y entrega un incremento del producto |

**Definición de artefactos** (*Sprint 1*)

**Backlog Sprint 1 Planificación y diseño de la aplicación**

|  |  |  |  |
| --- | --- | --- | --- |
| User story | Descripción | Estimación (Horas) | Responsable |
| Descripción de roles | Asignación de tareas a los desarrolladores. | 2 | Todos |
| Definición de artefactos | Definir los módulos a generar según solicitud del Hospital General de Barranquilla. | 10 | Todos |
| Diagrama de clases | Crear diagrama de clases. | 5 | Todos |
| Definición de cronograma | Estructurar cronograma de proyecto. | 5 | Todos |
| Entrega del sprint | Entrega de la información en los formatos solicitados. | 0.5 | Todos |

**Backlog Sprint 2 Desarrollo Front-End de la aplicación**

|  |  |  |  |
| --- | --- | --- | --- |
| User story | Descripción | Estimación (Horas) | Responsable |
| Mapa de navegabilidad | Documento descriptivo del mapa de navegabilidad. | 10 | Todos |
| Diseño y estilo de vista de inicio | Diseño e implementación preliminar de esta vista. | 6 | *Dainer José Gutiérrez Vergara* |
| Diseño y estilo de vista de paciente | Diseño e implementación preliminar de esta vista. | 6 | *Mónica Carolina Delgado Molina* |
| Diseño y estilo de vista de login | Diseño e implementación preliminar de esta vista. | 6 | *Claudia Emilia Bedoya Madrid* |
| Diseño y estilo de vista de médico | Diseño e implementación preliminar de esta vista. | 6 | *Sandra Liliana Nomesque Suárez* |
| Diseño y estilo de vista de administrador | Diseño e implementación preliminar de esta vista. | 6 | *Claudia Emilia Bedoya Madrid* |
| Documento de vistas | Documento con la descripción y las imágenes de las vistas preliminares. | 4 | *Mónica Carolina Delgado Molina* |
| Maquetas en Balsamiq | Crear maquetas de las vistas en Balsamiq | 6 | *Juan Andrés Ríos Montealegre* |
| Aprender GIT | Aprendizaje autónomo de la herramienta GIT. | 10 | *Sandra Liliana Nomesque Suárez* |
| Creación del proyecto en GIT | Trabajo colaborativo del mapa de navegabilidad, vista y estilo de la aplicación. | 5 | *Sandra Liliana Nomesque Suárez* |

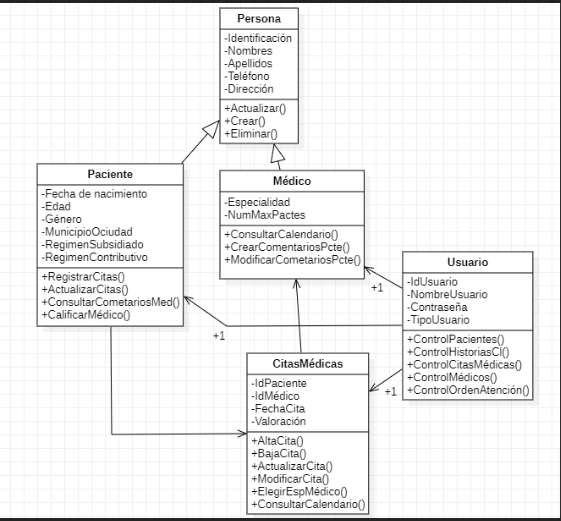
**Backlog Sprint 3 Desarrollo Back-End de la aplicación**

|  |  |  |  |
| --- | --- | --- | --- |
| User story | Descripción | Estimación (Horas) | Responsable |
| Diseño e implementación de los controladores para vista de ingreso/registro | Especificación de la ruta.  Definición de métodos HTTP permitidos.  Lógica algorítmica.  Documento descriptivo del diseño y la especificación de los controladores definidos. | 10 | *Dainer José Gutiérrez Vergara* |
| Diseño e implementación de los controladores para vista de login | Especificación de la ruta.  Definición de métodos HTTP permitidos.  Lógica algorítmica.  Documento descriptivo del diseño y la especificación de los controladores definidos. | 10 | *Sandra Liliana Nomesque Suárez* |
| Diseño e implementación de los controladores para vista de forgot password | Especificación de la ruta.  Definición de métodos HTTP permitidos.  Lógica algorítmica.  Documento descriptivo del diseño y la especificación de los controladores definidos. | 10 | *Mónica Carolina Delgado Molina* |
| Diseño e implementación de los controladores para vista de send message | Especificación de la ruta.  Definición de métodos HTTP permitidos.  Lógica algorítmica.  Documento descriptivo del diseño y la especificación de los controladores definidos. | 10 | *Claudia Emilia Bedoya Madrid* |
| Diseño e implementación de base de datos | Diseño de diagrama relacional.  Diseño e implementación de las tablas de las bases de datos con SQLite.  Documento descriptivo de las tablas y las relaciones más importantes. | 10 | *Sandra Liliana Nomesque Suárez* |
| Diseño e implementación de portal de acceso usando método de autenticación basado en usuario y contraseña | Creación de sesiones.  Uso de funciones hash criptográficas para almacenar contraseñas usando salts. | 5 | *Juan Andrés Ríos Montealegre* |

**Backlog Sprint 4 Despliegue de la aplicación**

|  |  |  |  |
| --- | --- | --- | --- |
| User story | Descripción | Estimación (Horas) | Responsable |
| Definición de requerimientos para el despliegue de la aplicación | Documento descriptivo de los requerimientos y el proceso de despliegue de la aplicación. | 5 | *Claudia Emilia Bedoya Madrid y Mónica Carolina Delgado Molina* |
| Configuración, despliegue y verificación del funcionamiento de la aplicación en una plataforma como servicio (PaaS) | Despliegue de la aplicación en una PaaS (PythonAnywhere o Heroku).  Verificación de la conectividad al dominio web <https://mi_dominio>.  Verificación del correcto funcionamiento de la aplicación. | 10 | Todos |

**Diagrama de clases** (*Sprint 1*)



Fuente: Elaboración propia (2022).

*Clase Persona:* donde se realizarán los procedimientos de creación, eliminación y actualización de datos generales de los usuarios (paciente, médico).

*Clase Paciente:* donde se realizarán los procedimientos de registro y actualización de citasdel paciente y también el de visualización de los comentarios del médico y que permitirá al paciente dejar la calificación sobre la atención médica*.*

*Clase Médico:* donde se realizarán los procedimientos de consulta de calendario de citas y consulta, crear o modificar los comentarios indicados a cada paciente.

*Clase CitasMedicas:* donde se realizarán los procedimientos de creación, modificación y eliminación de citas, a la vez que se manejarán los procedimientos para la selección del médico de la cita correspondiente y la consulta del calendario.

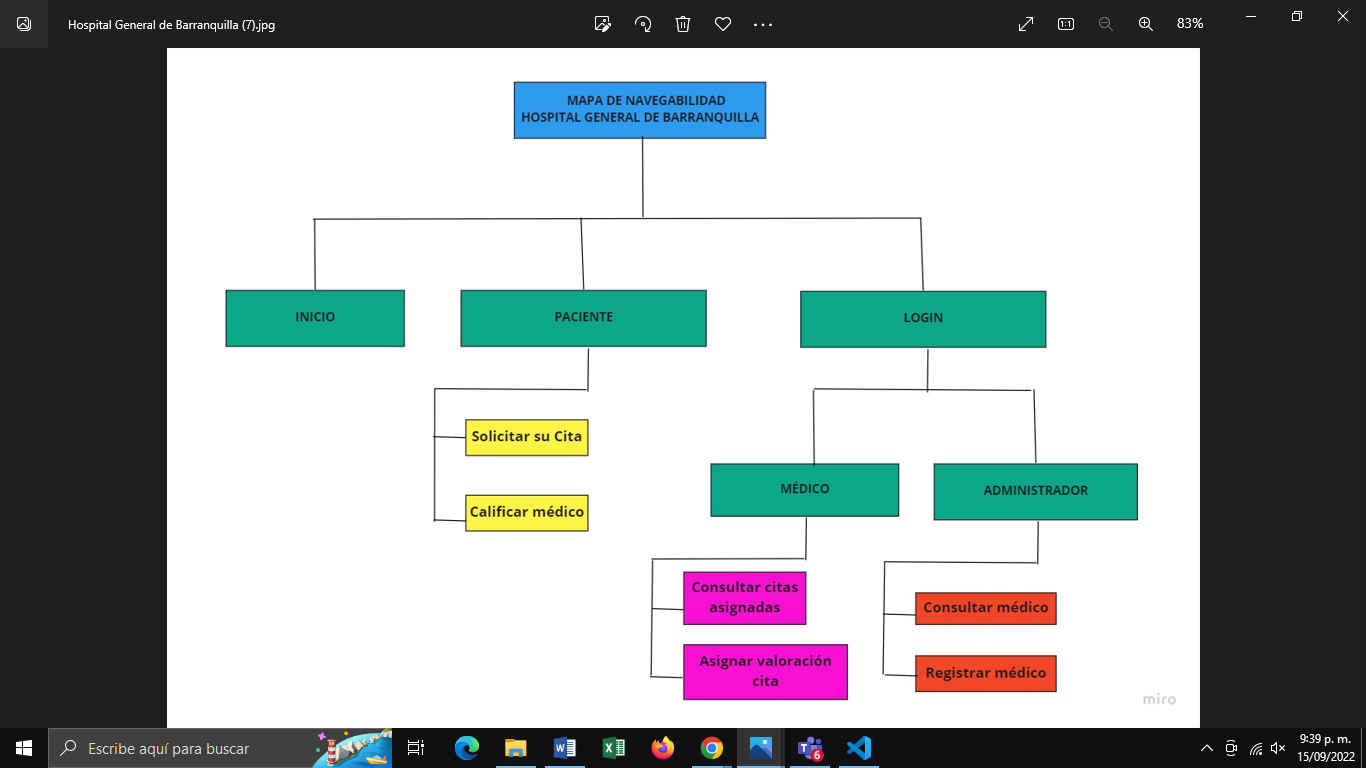
*Clase Usuarios:* donde se realizarán los procedimientos para llamar a las otras clases según corresponda el nivel asignado al usuario que ingresa.

**Cronograma de tareas** (*Sprint 1*)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CRONOGRAMA DE TAREAS** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SPRINT 1 | SEPTIEMBRE | | | | | | | | | | | | | | | | | | | | | | | | | | OCTUBRE | | | | | | | | | | | | |
| SEM1 | | | | | | SEM2 | | | | | | SEM3 | | | | | | SEM4 | | | | | | | | SEM1 | | | | | | SEM2 | | | | | | SEM3 |
| ACTIVIDADES | 1 | 2 | 3 | 4 | 6 | 7 | 8 | 9 | 10 | 12 | 13 | 14 | 15 | 16 | 17 | 19 | 20 | 21 | 22 | 23 | 24 | 26 | 27 | 28 | 29 | 30 | 1 | 3 | 4 | 5 | 6 | 7 | 8 | 10 | 11 | 12 | 13 | 14 | 15 |
| **Descripción de roles** | x | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Definición de artefactos** | x | x | x | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Diagrama de clases** |  | x | x | x | x | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Definición de cronograma** |  | x | x | x | x | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Entrega del sprint** |  |  |  |  |  |  | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Mapa de navegabilidad** |  |  |  |  |  |  |  | x | x | x | x | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Diseño y estilo de vista de inicio** |  |  |  |  |  |  |  |  | x | x | x | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Diseño y estilo de vista de médico** |  |  |  |  |  |  |  |  | x | x | x | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Diseño y estilo de vista de paciente** |  |  |  |  |  |  |  |  | x | x | x | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Diseño y estilo de vista de administrador** |  |  |  |  |  |  |  |  |  | x | x | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Documento de vistas** |  |  |  |  |  |  |  |  |  | x | x | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Aprender GIT** |  |  |  |  |  |  |  |  | x | x | x | x | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Creación del proyecto en GIT** |  |  |  |  |  |  |  |  |  |  | x | x | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Entrega del sprint** |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Diseño e implementación de los controladores para vista de ingreso** |  |  |  |  |  |  |  |  |  |  |  |  |  | x | x | x | x | x | x | x | x | x | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Diseño e implementación de los controladores para vista de médico** |  |  |  |  |  |  |  |  |  |  |  |  |  | x | x | x | x | x | x | x | x | x | x | x | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Diseño e implementación de los controladores para vista de paciente** |  |  |  |  |  |  |  |  |  |  |  |  |  | x | x | x | x | x | x | x | x | x | x | x | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Diseño e implementación de los controladores para vista de administrador** |  |  |  |  |  |  |  |  |  |  |  |  |  | x | x | x | x | x | x | x | x | x | x | x | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Diseño e implementación de base de datos** |  |  |  |  |  |  |  |  |  |  |  |  |  | x | x | x | x | x | x | x | x | x | x | x | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Diseño e implementación de portal de acceso usando método de autenticación basado en usuario y contraseña** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | x | x | x | x | x | x | x | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Entrega de sprint** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Definición de requerimientos para el despliegue de la aplicación** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | x | x | x | x | x |  |  |  |  |  |  |  |  |  |
| **Configuración, despliegue y verificación del funcionamiento de la aplicación en una plataforma como servicio (PaaS)** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | x | x | x | x | x | x | x | x | x | x |  |  |  |  |
| **Entrega de sprint** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | x |  |  |  |  |

Fuente: Elaboración propia (2022).

**Mapa de navegabilidad** (*Sprint 2*)



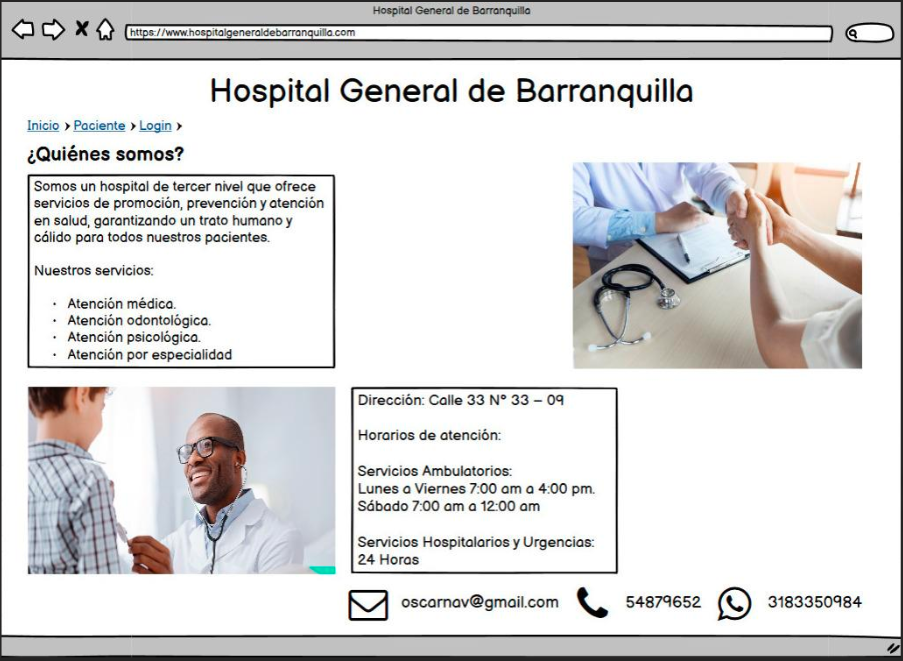
Fuente: Elaboración propia (2022).

En el Mapa de navegabilidad se observan las cuatro vistas que tendrá la página del Hospital General de Barranquilla. La primera corresponde a la ventana de Inicio, la segunda a Paciente, la tercera un Login que despliega a Médico y a Administrador. En el Mapa de navegabilidad se pueden observar las opciones que aparecerán en cada ventana.

**Vistas de la aplicación** (*Sprint 2*)

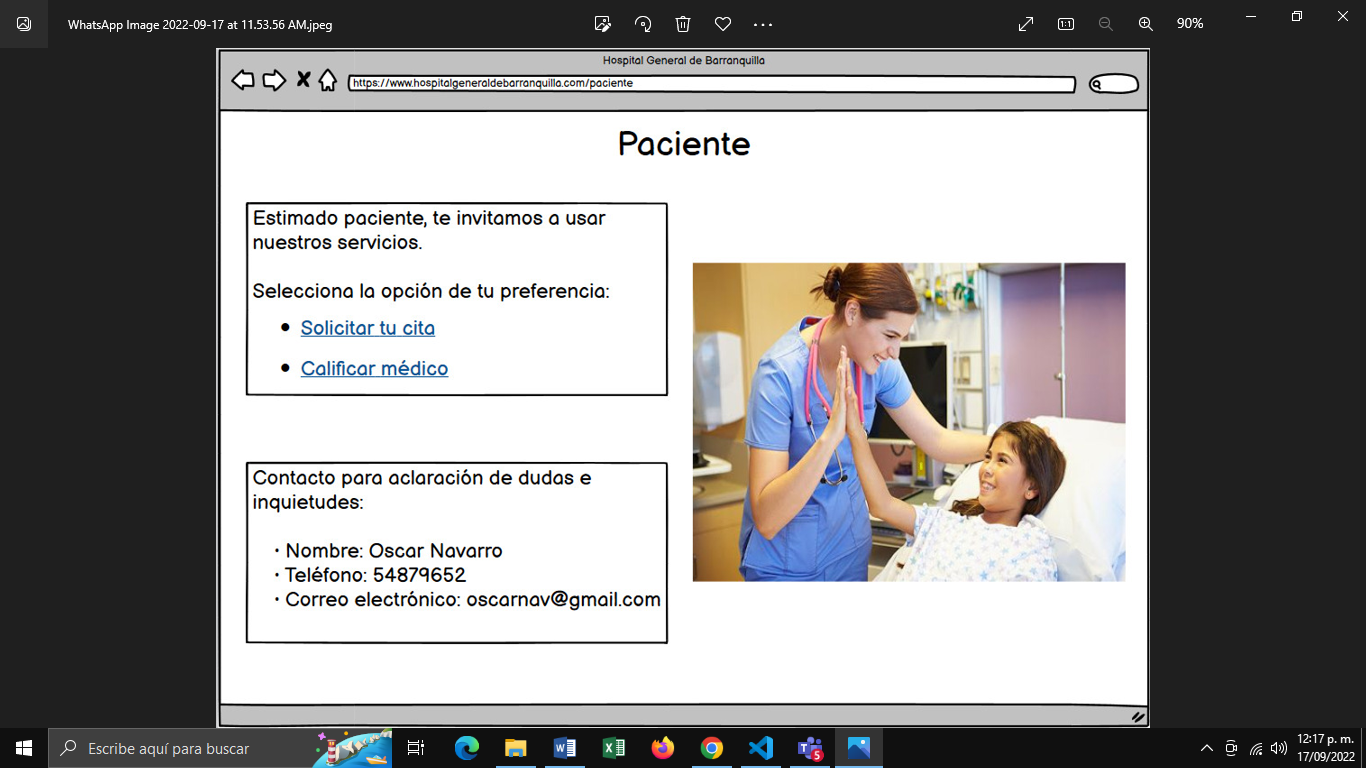
Con el uso del aplicativo Balsamiq se procedió a maquetar cada una de las pantallas que hacen parte de la página web del Hospital General de Barranquilla. Posteriormente, se crearon las pantallas en html y css. A continuación, se pueden observar las maquetas:

* Pantalla de Inicio:



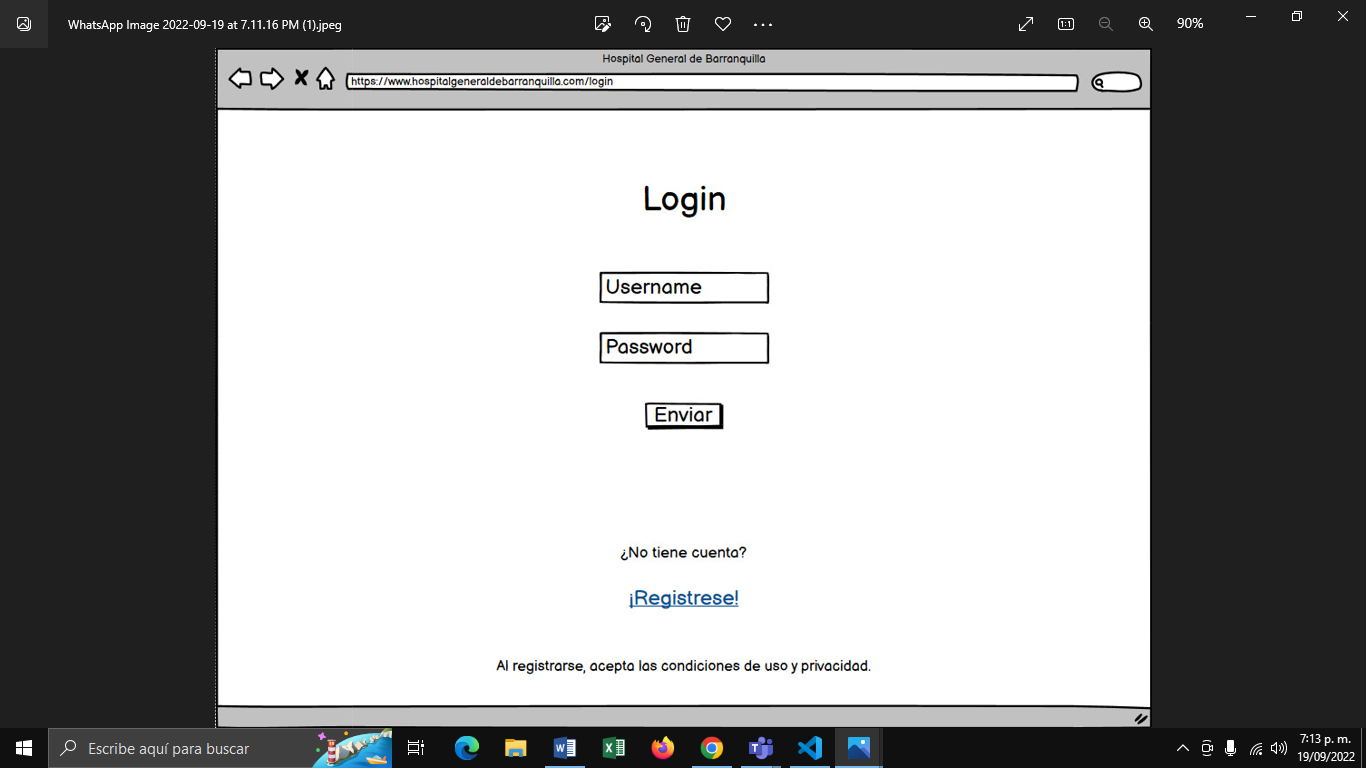
Fuente: Elaboración propia (2022).

* Pantalla de Paciente:



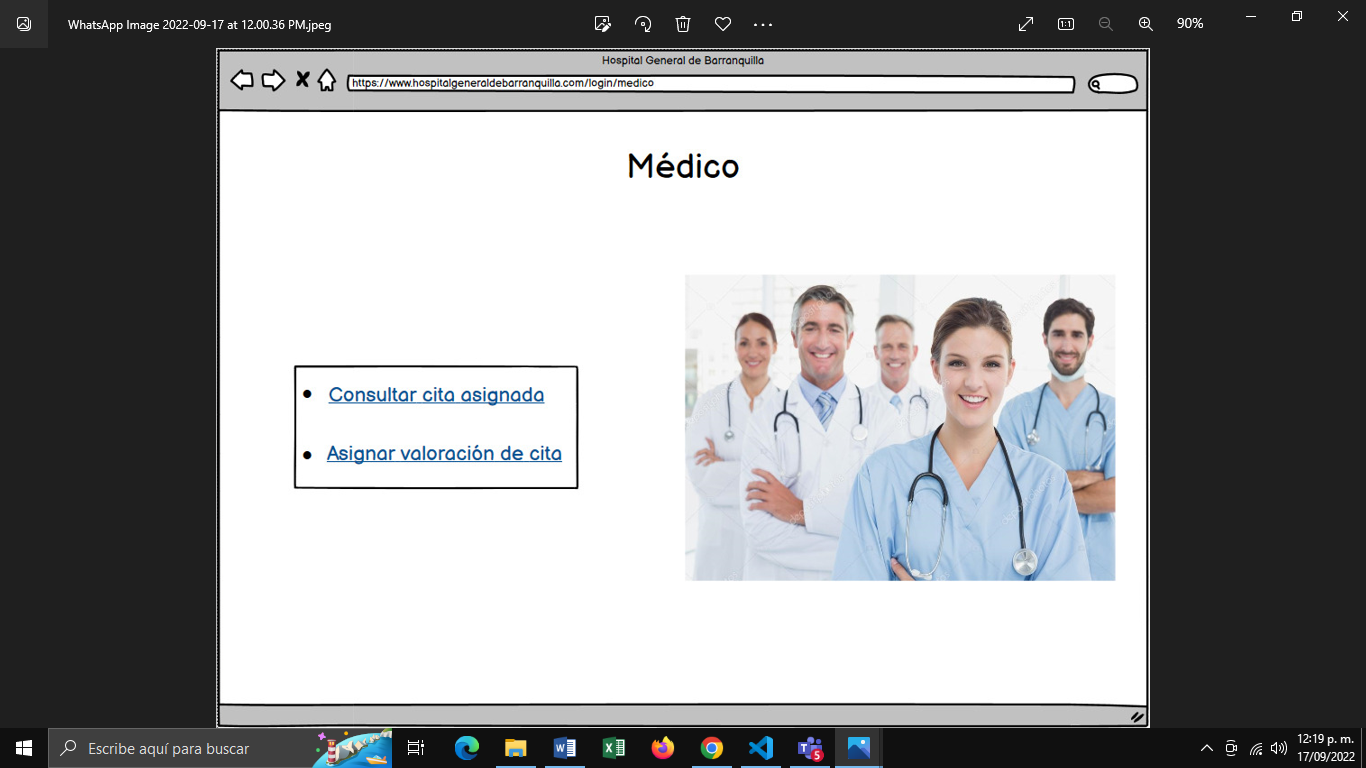
Fuente: Elaboración propia (2022).

* Pantalla Login:



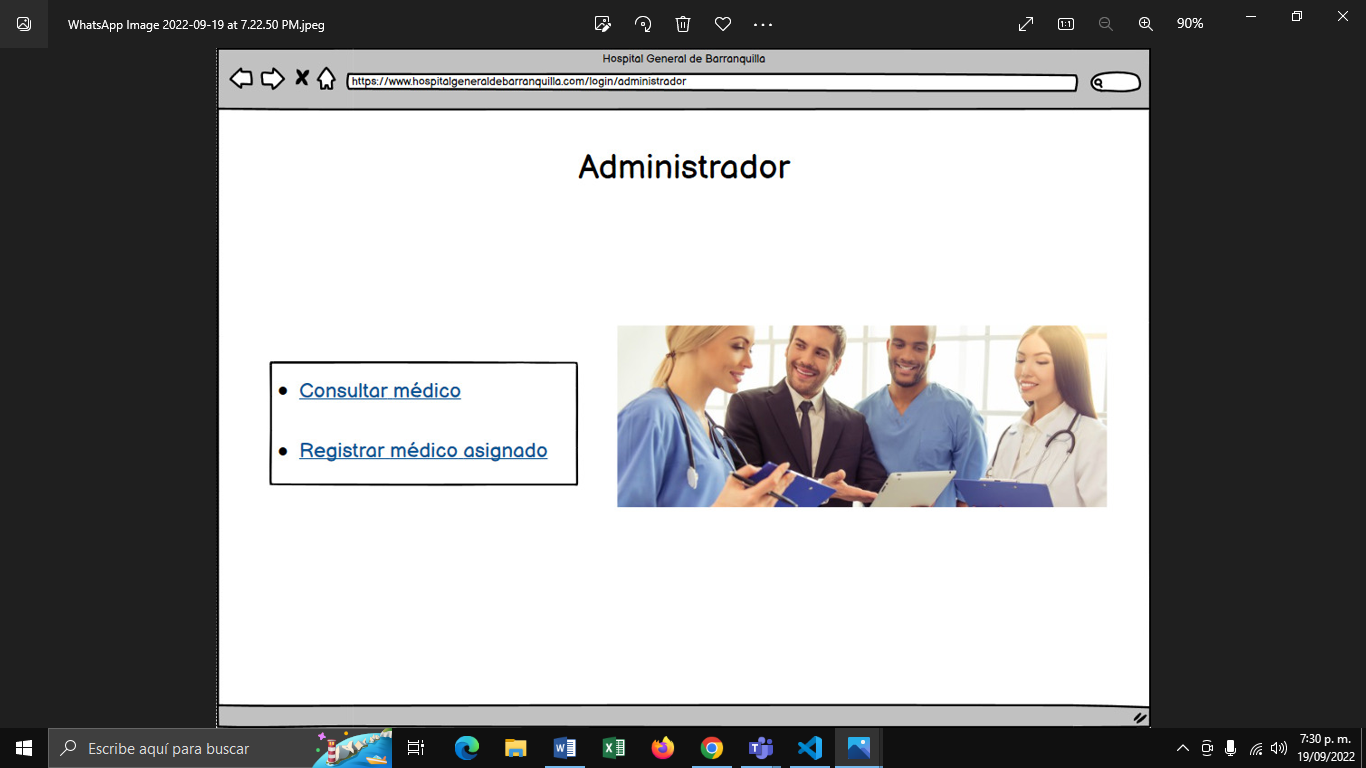
Fuente: Elaboración propia (2022).

* Pantalla de Médico:



Fuente: Elaboración propia (2022).

* Pantalla de administrador:



Fuente: Elaboración propia (2022).

**Vistas de las páginas que hacen parte de la página web (sprint 2)**

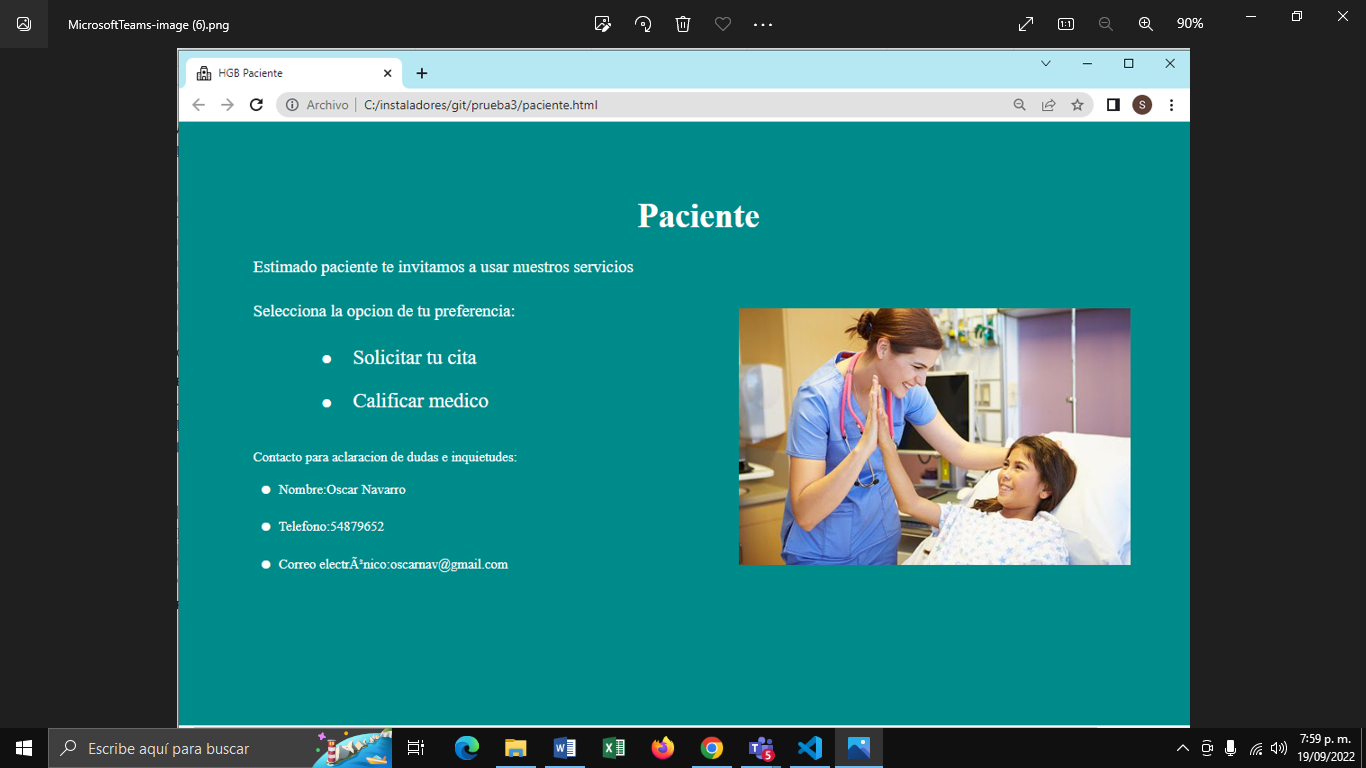
A continuación, se muestran cómo quedaron las pantallas de la página web de acuerdo con el mapa de navegabilidad.

* Pantalla de inicio:



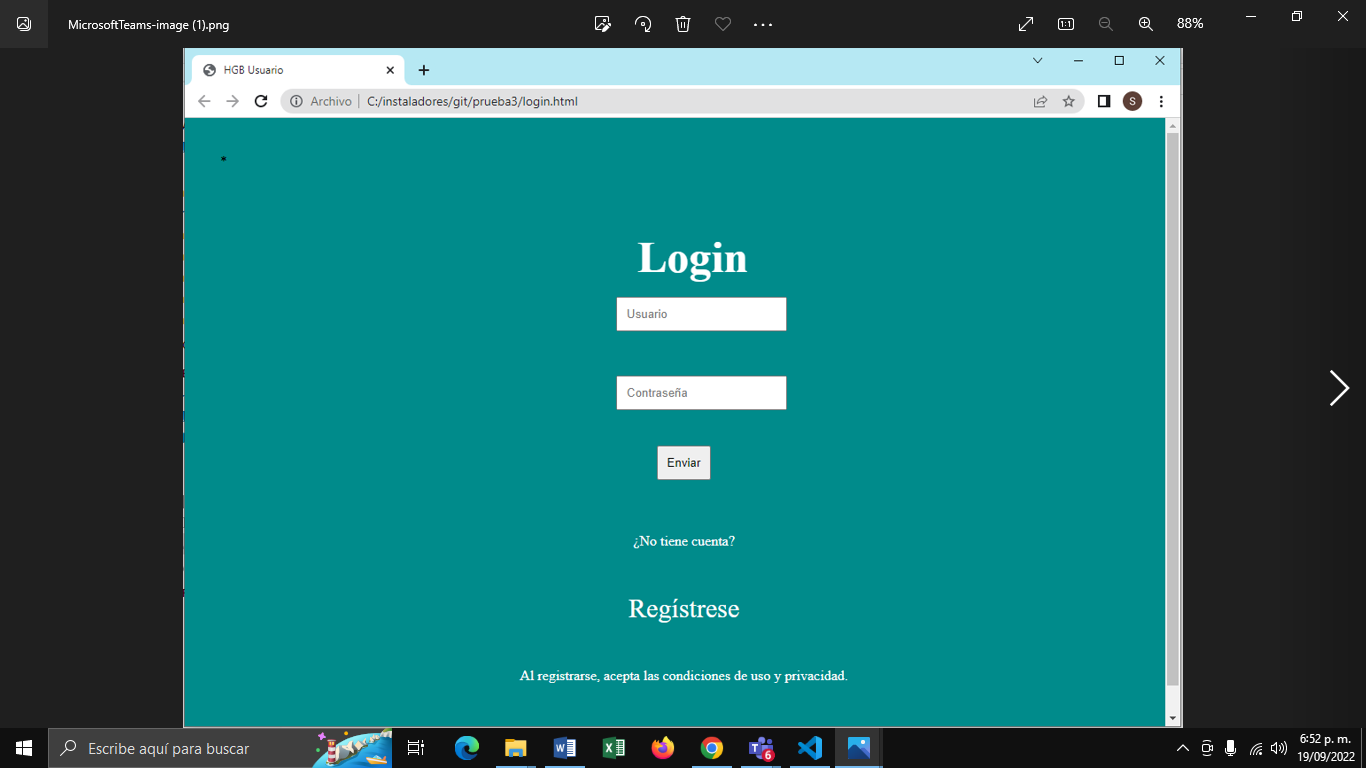
Fuente: Elaboración propia (2022).

* Pantalla de Paciente:



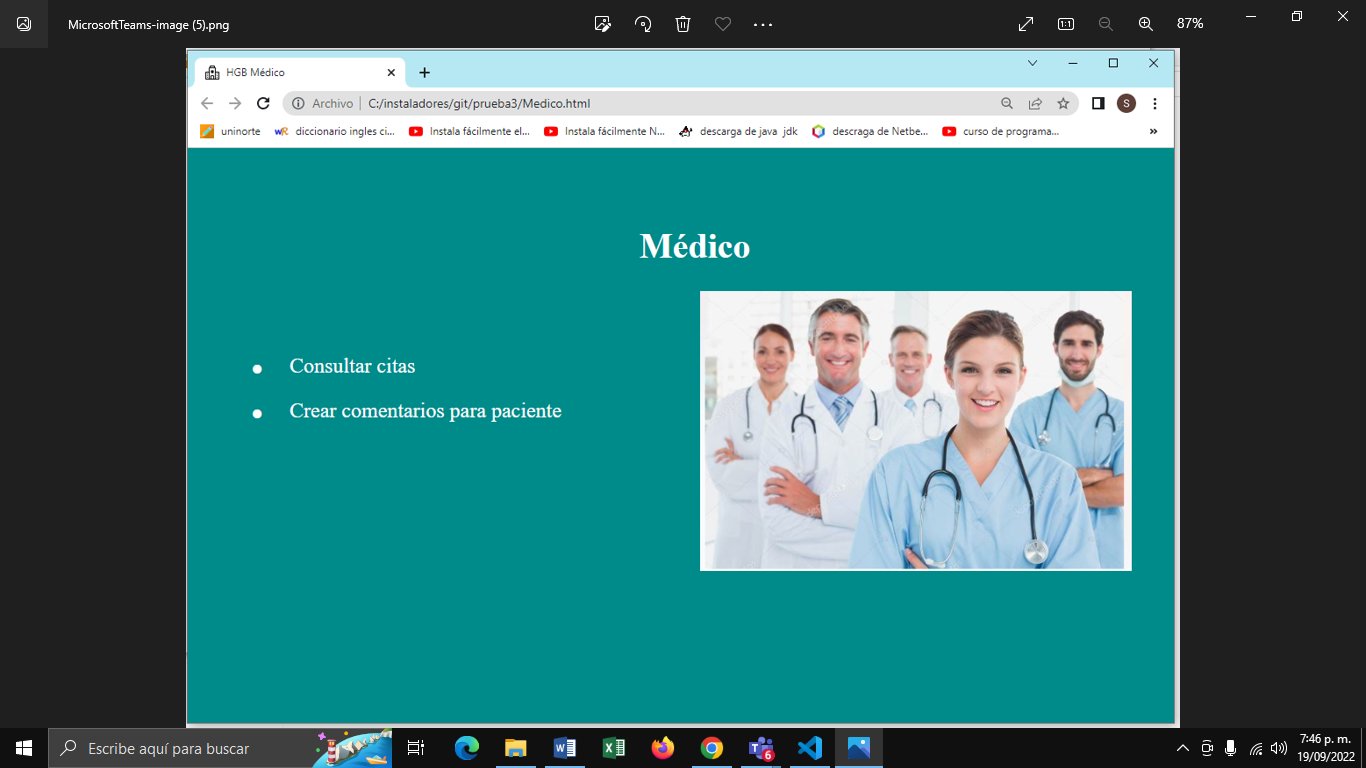
Fuente: Elaboración propia (2022).

* Pantalla de Login:



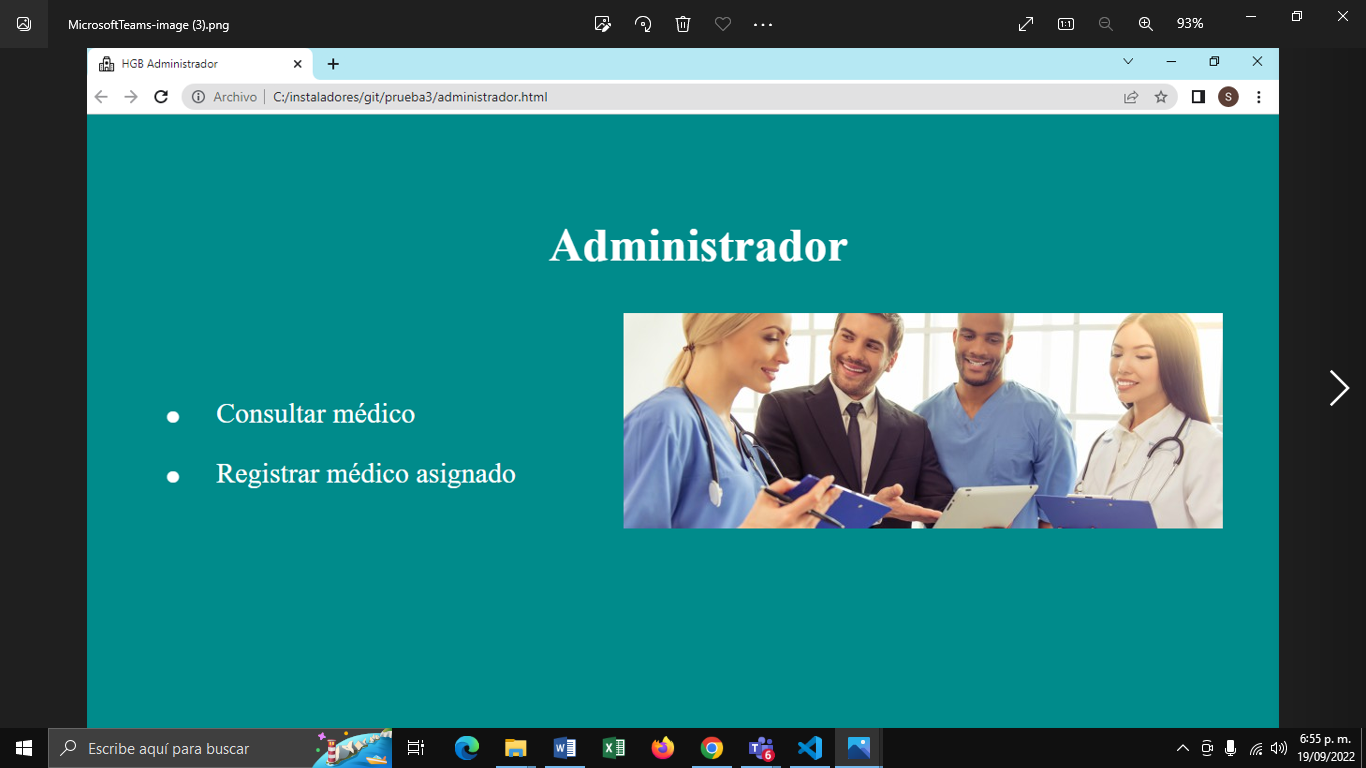
Fuente: Elaboración propia (2022).

* Pantalla de Médico:



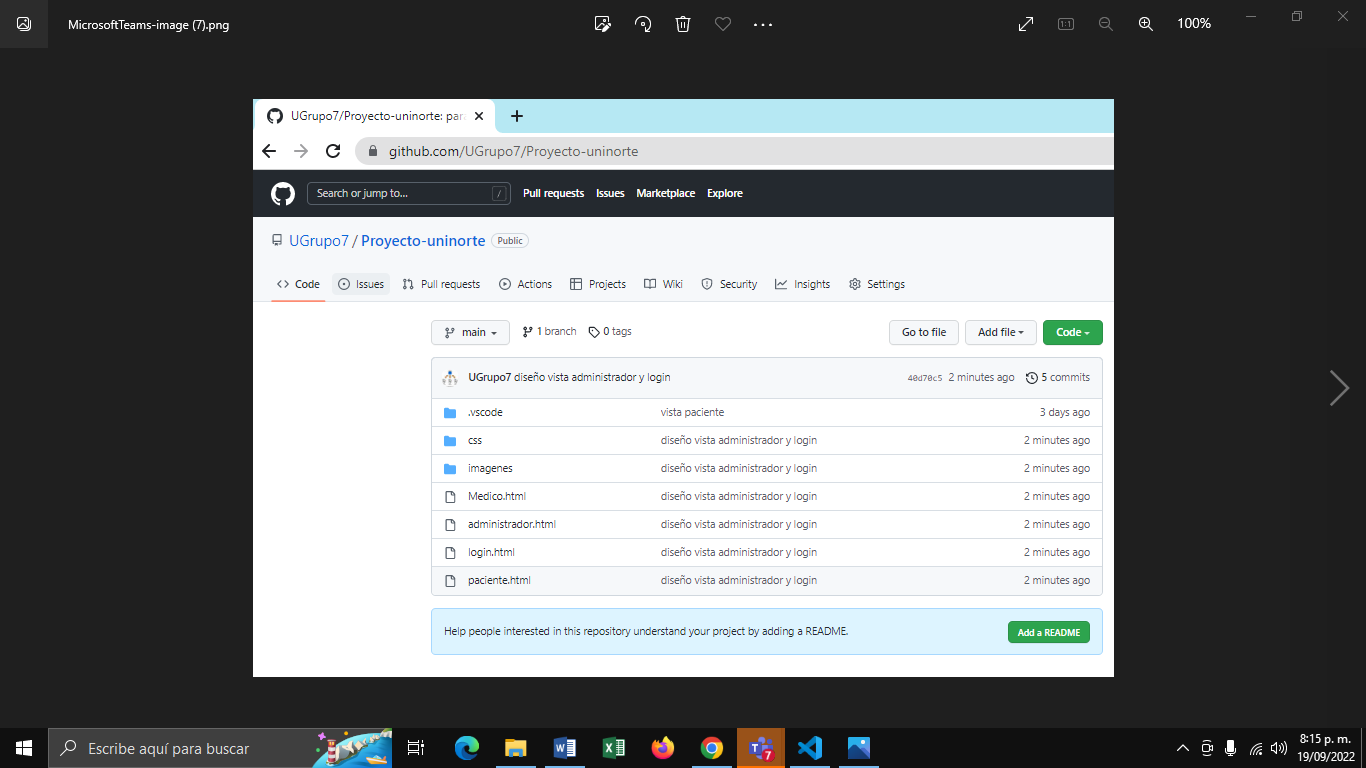
Fuente: Elaboración propia (2022).

* Pantalla de administrador:

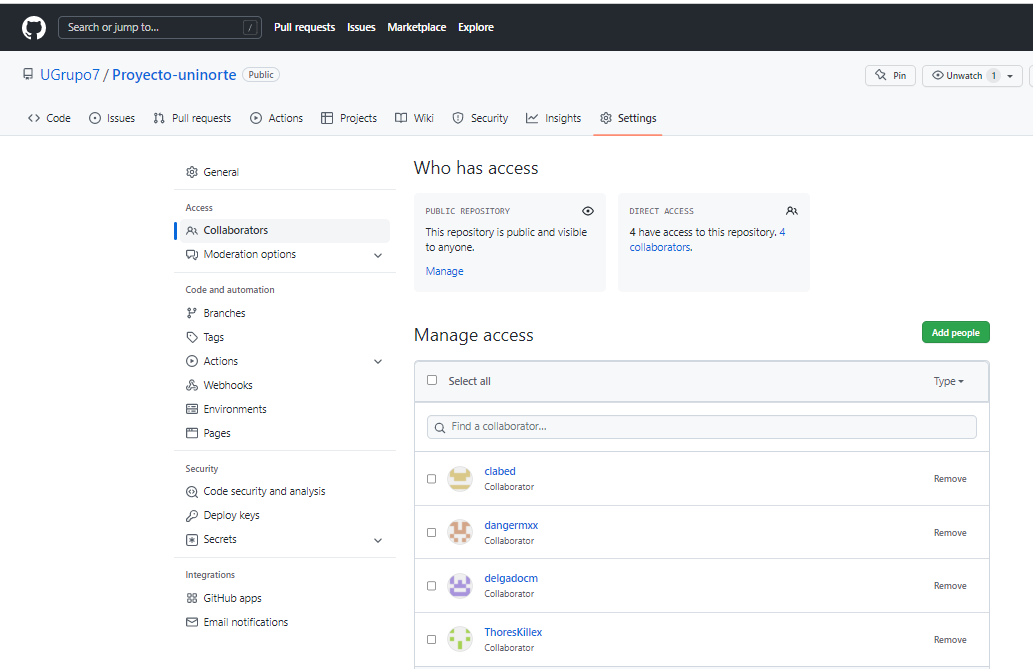


Fuente: Elaboración propia (2022).

* Pantallazo GitHub:



Fuente: Elaboración propia (2022).

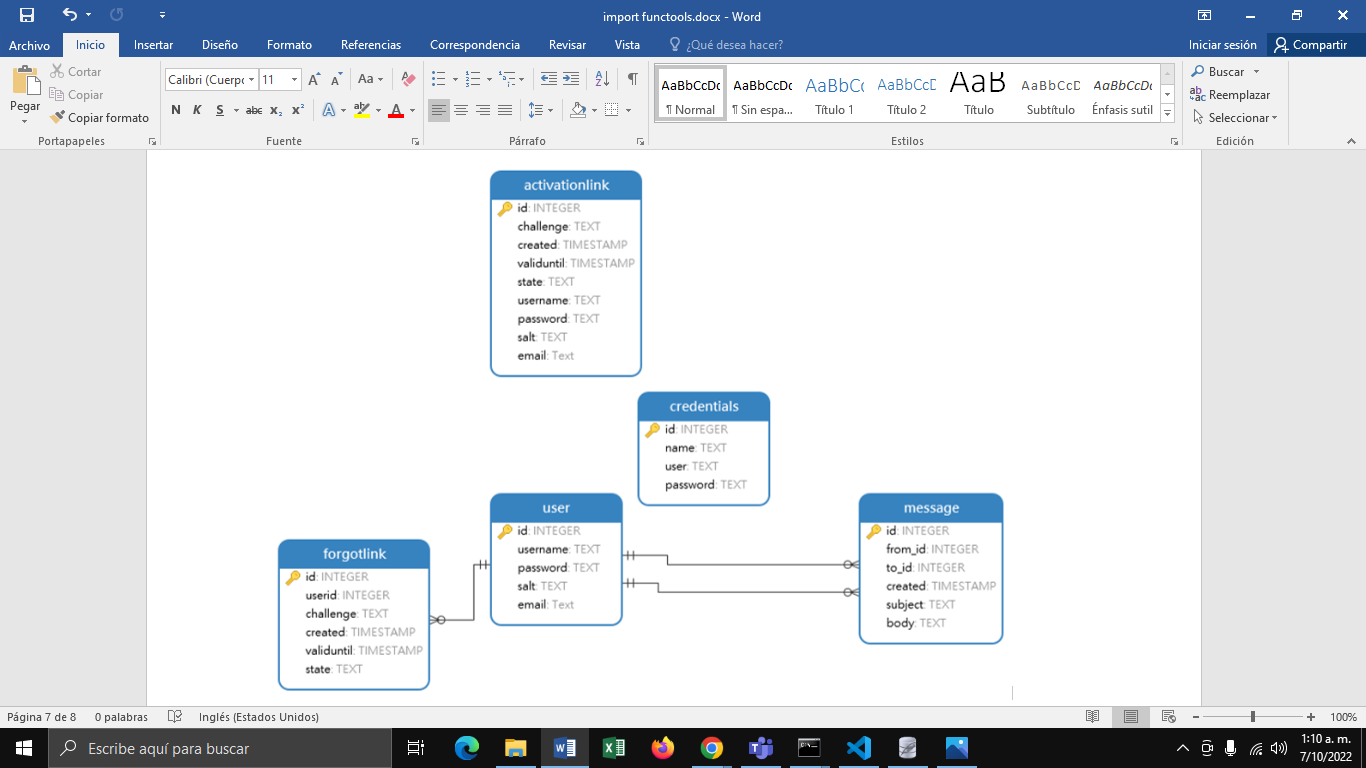


Fuente: Elaboración propia (2022).

**Métodos controladores del Back-End** (*Sprint 3*)

* Nombre método controlador: auth.py
* Ruta: /activate
* Métodos HTTP permitidos: GET y POST
* Lógica algorítmica: Permite activar correctamente el usuario que se ha registrado previamente. Se validan los datos para que la información ingrese en forma adecuada y segura.
* Ruta: /register
* Métodos HTTP permitidos: GET y POST
* Lógica algorítmica: Permite ingresar información del usuario: username, password y email. Si los datos anteriores no cumplen con los parámetros solicitados, se envía un mensaje de “error” para que el usuario proceda a hacer un nuevo registro de los datos. Si los datos son correctos, se enviará un mensaje al email para la posterior activación de la cuenta.
* Ruta: /confirm
* Métodos HTTP permitidos: GET y POST
* Lógica algorítmica: para confirmar la contraseña, se realizan dos registros, los cuales deben ser iguales. La contraseña debe cumplir con unos parámetros específicos relacionados con la seguridad y el uso de caracteres para que sea única.
* Ruta: /change
* Métodos HTTP permitidos: GET y POST
* Lógica algorítmica: el usuario tiene la posibilidad de cambiar la contraseña cuando lo desee y también por razones de seguridad.
* Ruta: /forgot
* Métodos HTTP permitidos: GET y POST
* Lógica algorítmica: Si el usuario olvida su contraseña, puede registrar el correo para restablecerla. El usuario recibirá un enlace en su correo que le permitirá cambiar la contraseña. En la pantalla, se solicitará una nueva contraseña y su confirmación.
* Ruta: /login
* Métodos HTTP permitidos: GET y POST
* Lógica algorítmica: en el login, los campos de registro requeridos son usuario y contraseña para tener acceso. En caso de que el registro no sea el adecuado, el usuario recibirá un mensaje con la alerta para que rectifique los datos.
* Ruta: /logout
* Métodos HTTP permitidos: GET y POST
* Lógica algorítmica: el logout le permite al usuario cerrar la sesión.
* Nombre método controlador: inbox.py
  + Ruta: /getDB
  + Métodos HTTP permitidos: no aplica.
  + Lógica algorítmica: permite la conexión con la base de datos.
  + Ruta: /show
  + Métodos HTTP permitidos: no aplica.
  + Lógica algorítmica: se encarga de buscar el mensaje del usuario en la base datos.
  + Ruta: /send
  + Métodos HTTP permitidos: GET y POST.
  + Lógica algorítmica: muestra la pantalla para que el usuario escriba el mensaje. Los campos de registro son: to, subject y body. Si el usuario no escribe un mensaje en los campos requeridos, se genera un error.

**Base de datos** (*Sprint 3*)



Fuente: elaboración propia (2022)

En el diagrama de la base de datos se pueden observar cinco tablas: activationlink, credentials, forgotlink, user y message. En ellas aparece una relación directa (llaves foráneas) entre user y message. Esta relación está retroalimentada porque el usuario puede enviar un mensaje con la posibilidad de recibir una respuesta.

En este mismo sentido, el usuario tiene una relación directa (llave foránea) con forgotlink, es decir, que el usuario puede registrar el olvido de su contraseña y generar una nueva.

**Prácticas de programación segura** (*Sprint 3*)

Entre las buenas prácticas de seguridad informática, tanto las librerías instaladas como la sintáxis del código, deben garantizar el registro seguro de los datos de los usuarios, las claves de acceso deben ser robustas y difíciles de descifrar como en el programa desarrollado. El control de versiones también constituye una práctica segura que se realiza, en este caso, con la herramienta GIT.

Respecto a los prepared statements como parte del uso de buenas prácticas, estas son plantillas utilizadas para consultar la base de datos (Sqlite). Estas plantillas de declaración funcionan a partir de variables que solo se reemplazan con los valores reales dentro del sistema. De esta manera, los datos se almacenan en forma segura.

Otra práctica segura es el manejo de las funciones hash para proteger la confidencialidad de una clave. Esto significa, que una contraseña debe quedar almacenada en la base de datos después de pasar por un algoritmo y transformarse en un índice o dirección.

**Enlace GitHub**

El trabajo colaborativo se ha ejecutado desde el aplicativo GitHub, el cual permite “automatizar y mejorar el flujo de trabajo” (Lima, A., et al., 2014) entre los integrantes del equipo. A continuación, se encuentra el enlace desde donde se podrá realizar un mejor seguimiento al trabajo desarrollado por el equipo 7.

<https://github.com/UGrupo7/template_sprint_3_nc.git>

**Bibliografía**

Crovi Druetta, D., Aguirre, D., Apodaca, J., & Camacho, O. (2002). Página Web: una propuesta para su análisis. Revista Mexicana de Ciencias Políticas y Sociales, 45(185), 167-185.

Faranello, S. (2012). Balsamiq wireframes quickstart guide. Packt Publishing.

García, J. J., Guerrini, G., & Catania, B. (1996). Dimensiones en el diseño de un modelo de vistas orientadas a objetos. In *Actas de las Primeras Jornadas de Investigación y Docencia en Bases de Datos* (pp. 119-129). Universidade da Coruña.

Jaramillo, C. M. Z., Álvarez, C. A., & Isaza, F. A. (2007). Refinamiento del diagrama de clases UML a Oracle® 9I en ATOM³. *DYNA: revista de la Facultad de Minas. Universidad Nacional de Colombia. Sede Medellín*, *74*(151), 147-159.

Lima, A., Rossi, L., & Musolesi, M. (2014, May). Coding together at scale: GitHub as a collaborative social network. In Eighth international AAAI conference on weblogs and social media.

Leonardi, M. C., Mauco, V., & Leoni, H. (2005). CIMTool: Una herramienta para la definición de un diagrama de clases UML. In *XI Congreso Argentino de Ciencias de la Computación*.

Sánchez, S., Domínguez, P., & Velásquez, L. (2018). Hashing: Técnicas y Hash para la Protección de Datos. *Universidad Tecnológica de Panamá, Grupo de Investigación*.

Thomas, S., Williams, L., & Xie, T. (2009). On automated prepared statement generation to remove SQL injection vulnerabilities. *Information and Software Technology*, *51*(3), 589-598.