

**Índice**

**Contenido Pág.**

[ÍNDICE DE TABLAS III](#_Toc58337623)

[ÍNDICE DE FIGURAS III](#_Toc58337624)

[INTRODUCCIÓN IV](#_Toc58337625)

[JUSTIFICACIÓN IV](#_Toc58337626)

[OBJETIVOS IV](#_Toc58337627)

[ALCANCES V](#_Toc58337628)

[LIMITACIONES V](#_Toc58337629)

[ESTADO DEL ARTE V](#_Toc58337630)

[FUNDAMENTO TEORICO 1](#_Toc58337631)

[Aplicaciones web 1](#_Toc58337632)

[Asp net framework 1](#_Toc58337633)

[Modelo Vista Controlador (MVC) 2](#_Toc58337634)

[C# 3](#_Toc58337635)

[SQL SERVER 3](#_Toc58337636)

[JavaScript 3](#_Toc58337637)

[CSS 3](#_Toc58337638)

[AXIOS 4](#_Toc58337639)

[Vue.JS 4](#_Toc58337640)

[Bootstrap 4](#_Toc58337641)

[PROCEDIMIENTO Y DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS 5](#_Toc58337642)

[Características 5](#_Toc58337643)

[Fases 5](#_Toc58337644)

[Pre-juego 5](#_Toc58337645)

[Juego 6](#_Toc58337646)

[Post-juego 6](#_Toc58337647)

[Pasos de la planificación 6](#_Toc58337648)

[Pasos de la arquitectura 7](#_Toc58337649)

[Cierre 9](#_Toc58337650)

[RESULTADOS, PLANOS, GRÁFICAS, PROTOTIPOS, MAQUETAS, PROGRAMAS 10](#_Toc58337651)

[Problemática 10](#_Toc58337652)

[Diseño de la base de datos 10](#_Toc58337653)

[Diccionario de datos 10](#_Toc58337654)

[Tabla AspNetRoles 11](#_Toc58337655)

[Tabla Appointments 11](#_Toc58337656)

[Tabla Dentists 12](#_Toc58337657)

[Tabla Patients 13](#_Toc58337658)

[Tabla RequestAppointments 14](#_Toc58337659)

[Tabla PatientMonitoring 15](#_Toc58337660)

[Tabla AsNetUserRoles 16](#_Toc58337661)

[Tabla AspNetUsers 16](#_Toc58337662)

[Interfaz general del sistema 16](#_Toc58337663)

[REFERENCIAS BIBLIOGRAFICAS 23](#_Toc58337664)

[REFERENCIAS WEB 23](#_Toc58337665)

# **ÍNDICE DE TABLAS**

**Contenido Pág.**

[Tabla 1. Estado del arte. VI](#_Toc53844877)

[Tabla 2. Tabla AspNetRoles. 7](#_Toc53844878)

[Tabla 3. Tabla Appointments. 8](#_Toc53844879)

[Tabla 4. Tabla dentists. 9](#_Toc53844880)

[Tabla 5. Tabla Patients. 10](#_Toc53844881)

[Tabla 6. Tabla RequestAppointments. 11](#_Toc53844882)

[Tabla 7. Tabla PatientMonitoring. 11](#_Toc53844883)

[Tabla 8. Tabla AspNetUserRoles. 12](#_Toc53844884)

[Tabla 9. Tabla AspNetUsers. 12](#_Toc53844885)

# **ÍNDICE DE FIGURAS**

**Contenido Pág.**

[Figura 1. MVC 2](#_Toc58337592)

[Figura 2. Modelo E-R. 10](#_Toc58337593)

[Figura 3. Login 17](#_Toc58337594)

[Figura 4. Recuperar contraseña 17](#_Toc58337595)

[Figura 5. Solicitar cuenta. 18](#_Toc58337596)

[Figura 6. Home super usuario 18](#_Toc58337597)

[Figura 7. odontólogos modulo administrador 19](#_Toc58337598)

[Figura 8. Paciente modulo administrador 19](#_Toc58337599)

[Figura 9. configuración de cuenta módulo administrador 20](#_Toc58337600)

[Figura 10. Paciente módulo odontólogo. 20](#_Toc58337601)

[Figura 11. Consultas 21](#_Toc58337602)

[Figura 12. Seguimiento de consultas 21](#_Toc58337603)

[Figura 13. Configuración de cuenta modulo odontólogo 22](#_Toc58337604)

[Figura 14. Historial 22](#_Toc58337605)

[Figura 15. Consultas modulo pacientes 23](#_Toc58337606)

[Figura 16. configuración de cuenta modulo paciente 23](#_Toc58337607)

# **INTRODUCCIÓN**

# **JUSTIFICACIÓN**

Observando la necesidad que tiene el cirujano dentista de practica general y el especialista en áreas odontológicas de contar con un documento personalizado, ordenado y objetivo de cada paciente que cursa un tratamiento. Al tratarse de un expediente los registros se llevan a cabo en papel, en la actualidad los procesos realizados en cualquier campo están cambiando, la industria se está modernizando en todos los ámbitos, buscando siempre la automatización y tratando constantemente de hacer más eficientes sus procesos para mejorar sus resultados, el uso de documentos digitales juega un papel muy importante en estas mejoras ya que permite distribuir tareas, llevando un control de estas de forma eficaz.

Finalmente, se espera que al implementar el proyecto web sirva como apoyo para la gestión y control de citas, a su vez como una herramienta para que los clientes de los odontólogos puedan consultar de una manera más estructurada el seguimiento de su tratamiento.

# **OBJETIVOS**

GENERAL

Desarrollar una plataforma web utilizando ASP.NET Core MVC que facilite a los odontólogos la gestión y control de consultas realizadas a pacientes.

ESPECIFICOS

Diseñar una base de datos utilizando SQL server para proporcionar consultas oportunas y almacenar datos.

Crear una plataforma web utilizando el framework ASP.NET Core MVC para facilitar al usuario el registro de datos.

Programar un módulo para altas, bajas, modificaciones y eliminaciones por parte del administrador y odontólogos.

# **ALCANCES**

El sistema estará disponible para los profesionales de la salud bucodental.

El administrador podrá iniciar sesión para el registro de clientes (odontólogos).

El odontólogo podrá iniciar sesión para registrar pacientes, asignar y visualizar consultas.

# **LIMITACIONES**

La plataforma estará únicamente para navegadores web.

# **ESTADO DEL ARTE**

El estado del arte es una modalidad de la investigación documental que permite el estudio del conocimiento acumulado (escrito en textos) dentro de un área específica. (Montoya, 2005)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Hallazgo | Año | Características | Referencia | Aporte |
| Dindoc | 2014 | Sistema de agenda en línea, la cual permite que el mismo paciente registre el día y la hora de su cita como mejor prefiera, puede ingresar información básica del paciente, motivo de su visita, seguro de gastos médicos, datos de facturación entre otros. | (Solutions, 2014)  Solutions, M. (2014). *dindoc.mx*. Obtenido de http://dindoc.mx/ | Ahorro de tiempo  Información confiable por provenir directamente del paciente  Accesible desde cualquier sitio con conexión a internet |
| Edensys V2 | 2020 | eDENSys© V2 es un software técnico nuevo y mejorado de apoyo administrativo que facilitará la gestión y control sistematizado del expediente clínico odontológico en todos aquellos consultorios médicos en donde se brinda el proceso de atención médica bucal de manera personalizada | (aplicaciones, 2020)  aplicaciones, D. d.  (2020). grupodesisa.mx.  Obtenido de https://grupodesisa.mx/software-edensys | Reportes analíticos por Paciente y por Ingresos de consulta.  Sistema multiusuario, con módulo de seguridad de accesos por opción.  Tecnología Cliente/Servidor (registro en Red) |
| Medisel | 2020 | Es una plataforma web la cual permite atender a sus pacientes vía remota y segura utilizando el sistema de Telemedicina que MediSel ha integrado a la plataforma. | (MediSel, 2020)  MediSel. (2020). *expedienteclinico.mx*. Obtenido de https://expedienteclinico.mx/ | Acceso al expediente desde cualquier dispositivo con acceso a Internet (computadora, tableta o teléfono celular).  App móvil para el médico (iOs y Android). |

Tabla 1. Estado del arte.

# **FUNDAMENTO TEORICO**

# **Aplicaciones web**

Las aplicaciones web están compuestas por tres partes:

Una base de datos: en la base de datos se almacenan precisamente eso, todos los datos, permisos, usuarios, contenidos, información de interés.

El código de la aplicación: la propia aplicación se aloja en un servidor en la nube de aplicaciones, en algunos casos puede almacenarse en un servidor local.

El acceso a través del navegador: puedes acceder utilizando un ordenador o un dispositivo móvil por medio de un navegador. Se incluye el administrador o el gestor que es el usuario final. Podrá acceder a distintas secciones dependiendo de los permisos con los que cuente. (einatec, 2019)

# **Asp net framework**

.NET es una plataforma de desarrollo compuesta por herramientas, lenguajes de programación y bibliotecas para crear muchos tipos diferentes de aplicaciones.

ASP.NET amplía la [plataforma de desarrollo .NET](https://translate.googleusercontent.com/translate_c?depth=1&pto=aue&rurl=translate.google.com&sl=auto&sp=nmt4&tl=es&u=https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet&usg=ALkJrhh8f7AdPxIWEqrEzhCcEbDakeERfw) con herramientas y bibliotecas específicamente para crear aplicaciones web. ASP.NET funciona más rápido que cualquier marco web popular en los [puntos de referencia](https://translate.googleusercontent.com/translate_c?depth=1&pto=aue&rurl=translate.google.com&sl=auto&sp=nmt4&tl=es&u=https://www.techempower.com/benchmarks/&usg=ALkJrhiGb2sPs2BOehXW_bT29VCH_pypfg#hw=ph&test=plaintext) independientes de [TechEmpower](https://translate.googleusercontent.com/translate_c?depth=1&pto=aue&rurl=translate.google.com&sl=auto&sp=nmt4&tl=es&u=https://www.techempower.com/benchmarks/&usg=ALkJrhiGb2sPs2BOehXW_bT29VCH_pypfg#hw=ph&test=plaintext).

ASP.NET es compatible con los protocolos de autenticación estándar de la industria. Las funciones integradas ayudan a proteger sus aplicaciones contra la creación de scripts entre sitios (XSS) y la falsificación de solicitudes entre sitios (CSRF).

ASP.NET proporciona una base de datos de usuarios incorporada con soporte para autenticación multifactor y autenticación externa con Google, Twitter y más. (Microsoft, 2020)

# **Modelo Vista Controlador (MVC)**

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.

El Modelo que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.

La Vista, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.

El Controlador, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno. (Alicante, 1996-2020)

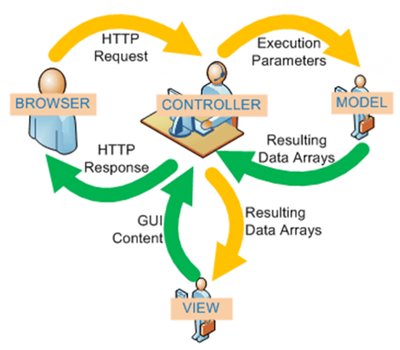


Figura 1. MVC

# **C#**

La sintaxis viene derivada de C y C+ + y utiliza el modelo de objetos de la plataforma .NET, muy parecido al de Java, aunque incluye mejoras propias de otros lenguajes. Como curiosidad, el nombre de este lenguaje fue inspirado por la escala musical. En ella, la letra C equivale a la nota musical do y el símbolo # significa sostenido, lo que indica que es un semitono más alto. Así, C# sugiere que es superior a C y C++. (S.L, 2018-2020)

# **SQL SERVER**

El lenguaje de desarrollo utilizado (por línea de comandos o mediante la interfaz gráfica de Management Studio) es [Transact-SQL](https://es.wikipedia.org/wiki/Transact-SQL) (TSQL), una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos ([DML](https://es.wikipedia.org/wiki/Lenguaje_de_manipulaci%C3%B3n_de_datos)), crear tablas y definir relaciones entre ellas ([DDL](https://es.wikipedia.org/wiki/Lenguaje_de_definici%C3%B3n_de_datos)). (Microsoft, 2020)

# **JavaScript**

JavaScript (JS) es un lenguaje de programación ligero, interpretado, o compilado [justo-a-tiempo](https://en.wikipedia.org/wiki/Just-in-time_compilation) (just-in-time) con [funciones de primera clase](https://developer.mozilla.org/es/docs/Glossary/Funcion_de_primera_clase). Si bien es más conocido como un lenguaje de scripting (secuencias de comandos) para páginas web, y es usado en [muchos entornos fuera del navegador](https://en.wikipedia.org/wiki/JavaScript#Uses_outside_Web_pages), tal como [Node.js](https://developer.mozilla.org/es/docs/Glossary/Node.js), [Apache CouchDB](https://couchdb.apache.org/) and [Adobe Acrobat](http://www.adobe.com/devnet/acrobat/javascript.html). JavaScript es un lenguaje de [programación basada en prototipos](https://developer.mozilla.org/es/docs/Glossary/Prototype-based_programming), multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa (por ejemplo programación funcional). Lee más en [acerca de JavaScript](https://developer.mozilla.org/es/docs/Web/JavaScript/About_JavaScript). (docs, 2005-2020)

# **CSS**

Hojas de Estilo en Cascada (del inglés Cascading Style Sheets) o CSS es el lenguaje de [estilos](https://wiki.developer.mozilla.org/es/docs/Web/API/StyleSheet) utilizado para describir la presentación de documentos [HTML](https://developer.mozilla.org/es/docs/HTML) o [XML](https://developer.mozilla.org/es/docs/XML) (incluyendo varios languages basados en XML como [SVG](https://wiki.developer.mozilla.org/es/docs/Web/SVG), [MathML](https://wiki.developer.mozilla.org/es/docs/Web/MathML) o [XHTML](https://developer.mozilla.org/es/docs/XHTML)). CSS describe como debe ser renderizado el elemento estructurado en la pantalla, en papel, en el habla o en otros medios.

CSS es uno de los lenguajes base de la Open Web y posee una [especificación estandarizada](http://www.w3.org/Style/CSS/#specs) por parte del W3C. Anteriormente , el desarrollo de varias partes de las especificaciones de CSS era realizado de manera sincrónica, lo que permiía el versionado de las recomendaciones. Probablemente habrás escuchado acerca de CSS1, CSS2.1, CSS3. Sin embargo, CSS4 nunca se ha lanzado como una versión oficial. (docs, 2005-2020)

# **AXIOS**

[Axios](https://github.com/axios/axios) es una librería JavaScript que puede ejecutarse en el navegador y que nos permite hacer sencillas las operaciones como cliente HTTP, por lo que podremos configurar y realizar solicitudes a un servidor y recibiremos respuestas fáciles de procesar. Aunque podría resultarnos muy útil en infinidad de situaciones, deberíamos antes analizarla las casuísticas con cuidado para determinar si es la mejor solución. En este artículo, analizamos las principales características de Axios y qué debemos tener en cuenta antes de empezar a trabajar con esta librería. (ARSYS, 2020)

# **Vue.JS**

Vue es un framework open source de JavaScript, el cual nos permite construir interfaces de usuarios de una forma muy sencilla. Vue fue creado por [Evan You](https://twitter.com/youyuxi) ex trabajador de Google, quien, es importante mencionar, fue desarrollador Angular. Vue fue lanzado en el año 2014. Aunque inicialmente fue pensado para ser una biblioteca personal, la comunidad hizo que el proyecto creciera a un ritmo impresionante, posicionándolo hoy en día como uno de los Frameworks web más populares, junto con Angular y React. (codigofacilito, 2020)

# **Bootstrap**

Bootstrap es un [framework](https://rockcontent.com/es/blog/framework/) CSS desarrollado por Twitter en 2010, para estandarizar las herramientas de la compañía.

Inicialmente, se llamó Twitter Blueprint y, un poco más tarde, en 2011, se transformó en código abierto y su nombre cambió para Bootstrap. Desde entonces fue actualizado varias veces y ya se encuentra en la versión 4.4.

El framework combina [CSS](https://rockcontent.com/es/blog/que-es-css/) y JavaScript para estilizar los elementos de una página HTML. Permite mucho más que, simplemente, cambiar el color de los botones y los enlaces.

Esta es una herramienta que proporciona interactividad en la página, por lo que ofrece una serie de componentes que facilitan la comunicación con el usuario, como menús de navegación, controles de página, barras de progreso y más. (team, s.f.)

# **PROCEDIMIENTO Y DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS**

Scrum es el método de desarrollo de nuevos productos realizado con equipos reducidos, multidisciplinares, que trabajan con comunicación directa y empleando ingeniería concurrente, en lugar de ciclos o fases secuenciales. (ScrumManager, 2013)

## **Características**

Los primeros en observar diferentes implementaciones de SCRUM para el desarrollo de nuevos productos con alto rendimiento fueron Takeuchi y Nonaka en Fuji-Xerox, Canon, Honda, NEC, Epson, Brother, 3M, Xerox, y Hewlett-Packard. Coplien ha seguido y documentado un enfoque similar para el desarrollo de software en Borland, logrando la mayor productividad con C++. Recientemente, Jeff Sutherland ha aplicado un enfoque más refinado de SCRUM en Smaltalk, y Schwaber en el desarrollo de Delphi. Llamamos a este enfoque metodología SCRUM (véase Takeuchi y Nonaka, 1986) por el uso del término SCRUM en rugby –la formación cerrada que forman los delanteros para realizar el avance. Scrum es una metodología para mejora y mantenimiento de un sistema existente o la producción de un prototipo. Se asume la existencia de diseño y código en su mayor parte orientado a objetos, basado en librerías y clases

## **Fases**

## **Pre-juego**

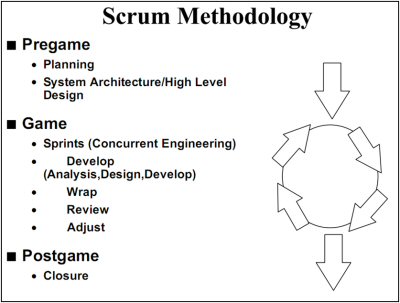
Planificación: Definición de una nueva versión basada en la pila actual, junto con una estimación de coste y agenda. Si se trata de un nuevo sistema, esta fase abarca tanto la visión como el análisis. Si se trata de la mejora de un sistema existente comprende un análisis de alcance más limitado. Arquitectura: Diseño de la implementación de las funcionalidades de la pila. Esta fase incluye la modificación de la arquitectura y diseño generales. (ScrumManager, 2013)

## **Juego**

Desarrollo de sprints: Desarrollo de la funcionalidad de la nueva versión con respeto continuo a las variables de tiempo, requisitos, costo y competencia. La interacción con estas variables define el final de esta fase. El sistema va evolucionando a través de múltiples iteraciones de desarrollo o sprints. (ScrumManager, 2013)

## **Post-juego**

Preparación para el lanzamiento de la versión, incluyendo la documentación final y pruebas antes del lanzamiento de la versión. (ScrumManager, 2013)



## **Pasos de la planificación**

* Desarrollo de un backlog completo.
* Determinación de la fecha de entrega y la funcionalidad de una o más versiones.
* Selección de la versión más adecuada para desarrollo inmediato.
* Trazado de los “paquetes del producto” (objetos) sobre los elementos del backlog de la versión elegida.
* Selección del equipo o equipos para desarrollar la nueva versión.
* Evaluación y control adecuado de los riesgos.
* Estimación del coste de la versión, incluyendo desarrollo, material, marketing, formación y despliegue.
* Conformidad de la dirección y financiación del proyecto.

## **Pasos de la arquitectura**

* Revisión de los elementos del backlog incluidos en la versión.
* Identificación de los cambios necesarios para implementar el backlog.
* Análisis del dominio para incluir los requisitos que incluye el desarrollo mejora o actualización.
* Acotar la arquitectura del sistema para apoyar el nuevo contexto y necesidades.
* Identificar problemas del desarrollo o modificaciones.
* Reunión de revisión de diseño. Cada equipo presenta los cambios para implementar los elementos del backlog, e identificar posibles reasignaciones.

##### **Pasos del desarrollo (Sprint)**

La fase de desarrollo es un ciclo de trabajo repetitivo. La gestión determina el cumplimiento de los tiempos, funcionalidad y calidad. Este enfoque es conocido también como ingeniería concurrente. (ScrumManager, 2013)

El desarrollo consiste en los siguientes macro-procesos:

* Reunión con los equipos para revisar los planes de lanzamiento de versión.
* Distribución, revisión y ajuste de los estándares de conformidad para el producto.
* Sprints iterativos hasta que el producto se considera listo para su distribución.

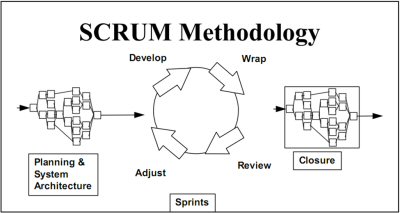
Un sprint es un conjunto de actividades de desarrollo llevado a cabo durante un periodo predefinido, por lo general entre una y cuatro semanas. Duración basada en la complejidad del producto, evaluación de riesgos y grado de supervisión deseado. El tiempo determinado para el sprint establece su velocidad e intensidad. El riesgo se evalúa de forma continua a través de las respuestas a los controles adecuados establecidos. (ScrumManager, 2013)

Cada sprint consiste en uno o varios equipos realizando:

* Desarrollo: Definición de los cambios necesarios para la implementación de los requisitos del backlog en módulos, la apertura de los módulos, análisis del dominio, diseño, desarrollo, implementación, pruebas y documentación de los cambios. El Desarrollo consiste en el micro proceso de descubrimiento, invención e implementación.
* Envoltura: Cierre de los módulos, creación de una versión ejecutable con los cambios que implementas los requisitos del backlog.
* Revisión: Reunión de todos los equipos para presentar el trabajo y revisar el progreso, identificando y resolviendo posibles cuestiones y añadiendo nuevos elementos al backlog. Se revisan los riesgos y las respuestas apropiadas.
* Ajuste: Consolidación de la información de la revisión de los módulos afectados.  
  Cada sprint es seguido de una revisión cuyas características son:

Está presente y participa el equipo al completo.

* La revisión puede incluir a clientes, personal de ventas y otros.
* La revisión cubre los sistemas funcionales y ejecutables abarcados por el equipo e incluye los cambios que se han realizado para implementar los elementos del backlog.
* En la revisión se pueden evidenciar cambios en la forma en la que se han implementado los elementos del backlog.
* La revisión también puede introducir elementos nuevos en el backlog, cambiando de esta forma los contenidos y dirección de las versiones previstas.
* Se determina la fecha de la siguiente revisión en base al progreso y complejidad. La duración normal de los sprints es de 1 a 4 semanas.



## **Cierre**

Cuando el equipo de gestión siente que las variables de tiempo, parte completada, requisitos, coste y calidad están alineadas para producir una nueva versión, declaran cerrada la versión, dando paso a esta fase. En esta fase se prepara el producto generado para producir una nueva versión. Entre las tareas de cierre se encuentran: integración, pruebas del sistema, documentación de usuario, preparación del material de formación y marketing. (ScrumManager, 2013)

# **RESULTADOS, PLANOS, GRÁFICAS, PROTOTIPOS, MAQUETAS, PROGRAMAS**

## **Problemática**

## **Diseño de la base de datos**

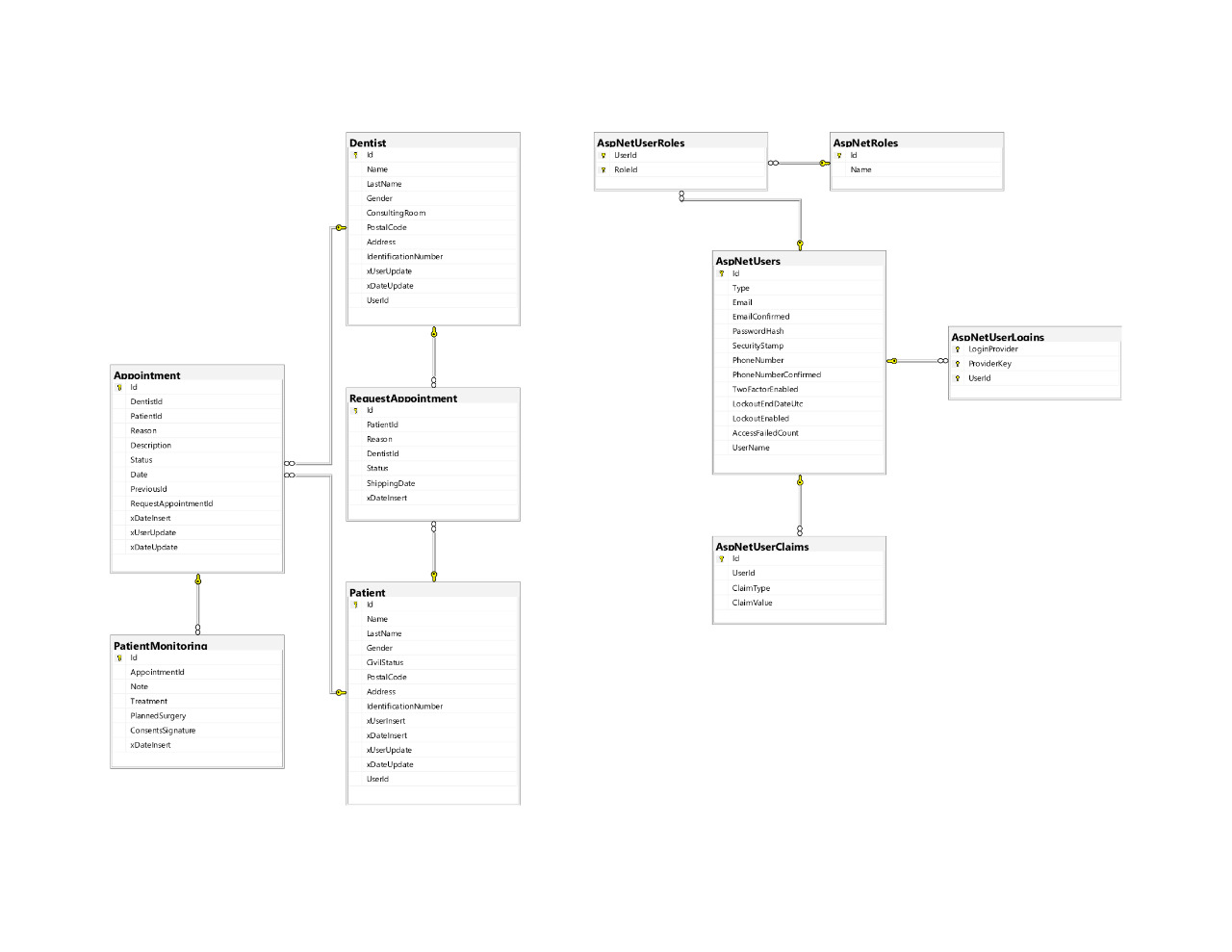


Figura 2. Modelo E-R.

## **Diccionario de datos**

Un diccionario de datos es un conjunto de definiciones  que contiene las características lógicas y puntuales de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización.  
Identifica los procesos donde se emplean los datos y los sitios donde se necesita el acceso inmediato a la información, se desarrolla durante el análisis de flujo de datos y auxilia a los analistas que participan en la determinación de los requerimientos del sistema, su contenido también se emplea durante el diseño.  
En un diccionario de datos se encuentra la lista de todos los elementos que forman parte del flujo de datos de todo el sistema. Los elementos más importantes son flujos de datos, almacenes de datos y procesos.

El diccionario de datos guarda los detalles y descripción de todos estos elementos.  
El diccionario se desarrolla durante el análisis de flujo de datos y auxilia a los analistas que participan en la determinación de los requerimientos de sistemas. (weebly, 2020)

## **Tabla AspNetRoles**

Almacenara los roles que estarán dentro de la plataforma.

|  |  |  |
| --- | --- | --- |
| Tabla: AspNetRoles | | |
| CAMPO | **DESCRIPCIÓN** | **TIPO DE DATO** |
| Id | Identificador principal de la tabla AspNetRoles. | nvarchar(128) |
| Name | Nombre de los diferentes roles que tendrán los usuarios | nvarchar(256) |

Tabla 2. Tabla AspNetRoles.

## **Tabla Appointments**

Almacenara las citas que realizara el dentista.

|  |  |  |
| --- | --- | --- |
| Tabla: Appointments | | |
| CAMPO | **DESCRIPCIÓN** | **TIPO DE DATO** |
| Id | Identificador principal de la tabla Appointments. | int |
| DentistId | Indicador foráneo del dentista que está realizando la cita. | int |
| PatientId | Indicador foráneo del paciente al cual se le asigna la cita. | int |
| Reason | Motivo por la cual se asignó la cita | nvarchar(100) |
| Date | Fecha que se asigna para la consulta | datetime |
| Description | Descripción detallada del motivo de la cita | nvarchar(200) |
| Status | Estado de la cita | nvarchar(15) |
| PreviousId | Indica si la cita es hija de otra cita anterior. | int |
| RequestAppointmentId | Indicador foráneo para indicar si la cita es una solicitada por un paciente. | int |
| xDateInsert | Fecha y hora en la cual se realizó la inserción a la tabla | datetime |
| xUserUpdate | Indicador foráneo para tener un control de los usuarios que realizan cambios o inserciones. | nvarchar(128) |
| xDateUpdate | Fecha y hora en la cual se realizó la actualización a la tabla | datetime |

Tabla 3. Tabla Appointments.

## **Tabla Dentists**

Almacena la información de los dentistas.

|  |  |  |
| --- | --- | --- |
| Tabla: Dentists | | |
| CAMPO | **DESCRIPCIÓN** | **TIPO DE DATO** |
| Id | Identificador principal de la tabla Dentists. | int |
| Name | Guarda el nombre de los/las dentistas. | nvarchar(20) |
| LastName | Guarda los apellidos de los/las dentistas. | nvarchar(30) |
| Gender | Almacena el género de los/las dentistas. | nvarchar(10) |
| ConsultingRoom | Almacena el consultorio. | nvarchar(100) |
| PostalCode | Almacena el código postal de los/las dentistas. | int |
| Address | Almacena la dirección de los/las dentistas. | nvarchar(100) |
| IdentificationNumber | Almacena el número de identificación de los/las dentistas. | nvarchar(13) |
| xDateInsert | Fecha y hora en la cual se realizó la inserción a la tabla | datetime |
| xUserUpdate | Indicador foráneo para tener un control de los usuarios que realizan cambios o inserciones. | nvarchar(128) |
| xDateUpdate | Fecha y hora en la cual se realizó la actualización a la tabla | datetime |
| UserId | Indicador foráneo para tener un usuario para el inicio de sesión. | Nvarchar(128) |

Tabla 4. Tabla dentists.

## **Tabla Patients**

Almacena la información de los pacientes de los dentistas.

|  |  |  |
| --- | --- | --- |
| Tabla: Patients | | |
| CAMPO | **DESCRIPCIÓN** | **TIPO DE DATO** |
| Id | Identificador principal de la tabla Patients. | int |
| Name | Guarda el nombre de los/las dentistas. | nvarchar(20) |
| LastName | Guarda los apellidos de los/las dentistas. | nvarchar(30) |
| Gender | Almacena el género de los/las dentistas. | nvarchar(10) |
| CivilStatus | Almacena el estado civil de los/las pacientes. | nvarchar(15) |
| PostalCode | Almacena el código postal de los/las dentistas. | int |
| Address | Almacena la dirección de los/las dentistas. | nvarchar(100) |
| IdentificationNumber | Almacena el número de identificación de los/las dentistas. | nvarchar(13) |
| xUserInsert | Indicador foráneo para tener un control de los usuarios que realizan cambios o inserciones. | nvarchar(128) |
| xDateInsert | Fecha y hora en la cual se realizó la inserción a la tabla | datetime |
| xUserUpdate | Indicador foráneo para tener un control de los usuarios que realizan cambios o inserciones. | nvarchar(128) |
| xDateUpdate | Fecha y hora en la cual se realizó la actualización a la tabla | datetime |
| UserId | Indicador foráneo para tener un usuario para el inicio de sesión. | nvarchar(128) |
| ConsentSignature | Firma digital para comprobar la aprobación del paciente al registrarse en la plataforma. | varbinary(max) |

Tabla 5. Tabla Patients.

## **Tabla RequestAppointments**

Almacena las citas solicitadas por los pacientes.

|  |  |  |
| --- | --- | --- |
| Tabla: RequestAppointments | | |
| CAMPO | **DESCRIPCIÓN** | **TIPO DE DATO** |
| Id | Identificador principal de la tabla RequestAppointments. | int |
| PatientId | Indicador foráneo del paciente que está solicitando la cita. | int |
| Reason | Motivo por el cual se solicita la cita. | nvarchar(100) |
| DentistId | Indicador foráneo del dentista al cual se le está solicitando la cita. | nvarchar(15) |
| Status | Almacena el estado de la cita si fue aprobada o rechazada. | nvarchar(15) |
| ShippingDate | Fecha en la que se envió la solicitud. | nvarchar(100) |
| xDateInsert | Fecha y hora en la cual se realizó la inserción a la tabla | datetime |

Tabla 6. Tabla RequestAppointments.

## **Tabla PatientMonitoring**

Almacena el seguimiento que se le da los pacientes de acuerdo con las citas que haya tenido.

|  |  |  |
| --- | --- | --- |
| Tabla: PatientMonitoring | | |
| CAMPO | **DESCRIPCIÓN** | **TIPO DE DATO** |
| Id | Identificador principal de la tabla PatientMonitoring. | int |
| AppointmentId | Indicador foráneo de la cita atendida. | int |
| Note | Almacena la nota que el dentista asigna en relación con la cita. | nvarchar(300) |
| Treatment | Almacena el tratamiento asignado por el dentista en relación con la cita. | nvarchar(300) |
| PlannedSurgery | Almacena si se tiene una cirugía planeada para poder realizar la cita. | nvarchar(300) |
| ConsentsSignature | Almacena la firma del paciente en la cual está de acuerdo con las notas de su tratamiento. | nvarchar(100) |
| xDateInsert | Fecha y hora en la cual se realizó la inserción a la tabla | datetime |

Tabla 7. Tabla PatientMonitoring.

## **Tabla AsNetUserRoles**

Almacena a los usuarios con sus roles.

|  |  |  |
| --- | --- | --- |
| Tabla: AspNetUserRoles | | |
| CAMPO | **DESCRIPCIÓN** | **TIPO DE DATO** |
| UserId | Identificador foráneo del usuario que se le asignara algún rol. | nvarchar(128) |
| RoleId | Identificador foráneo del rol que se le asignara a el usuario. | nvarchar(128) |

Tabla 8. Tabla AspNetUserRoles.

## **Tabla AspNetUsers**

Almacena a los usuarios con sus roles.

|  |  |  |
| --- | --- | --- |
| Tabla: AspNetUsers | | |
| CAMPO | **DESCRIPCIÓN** | **TIPO DE DATO** |
| Id | Identificador principal de la tabla AspNetUsers. | nvarchar(128) |
| Email | Almacena el correo de los usuarios. | nvarchar(256) |
| PasswordHash | Almacena la contraseña cifrada del usuario. | nvarchar(max) |
| PhoneNumber | Almacena el número de teléfono del usuario. | nvarchar(max) |
| Type | Indica el tipo de usuario. | nvarchar(15) |

Tabla 9. Tabla AspNetUsers.

## **Interfaz general del sistema**

Una vez desarrollada la plataforma web se establecieron las interfaces que constituyen la plataforma en general.

Pantalla inicial de la plataforma será el login que estará disponible para usuarios anónimos con las opciones de ingresar, solicitar una cuenta o recuperar contraseña.

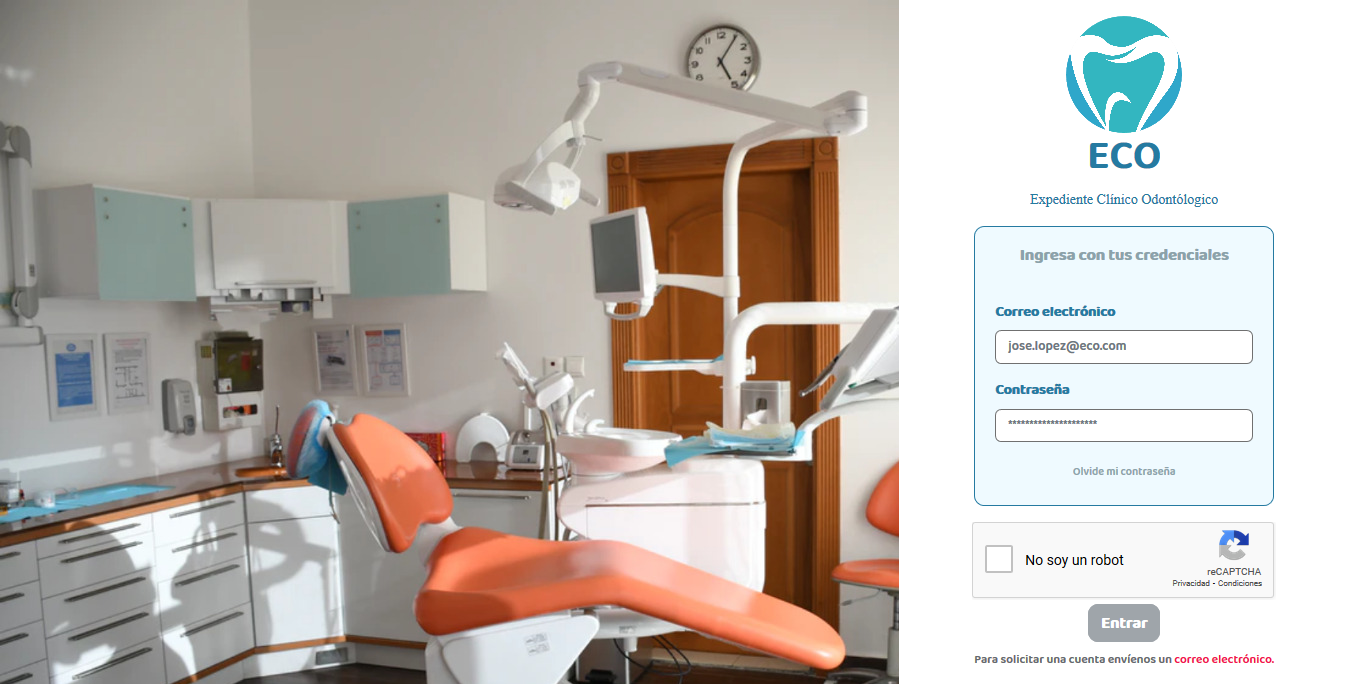


Figura 3. Login

Pantalla de recuperación de contraseña, estará disponible para usuarios anónimos en la cual se podrá solicitar el restablecimiento de contraseña.

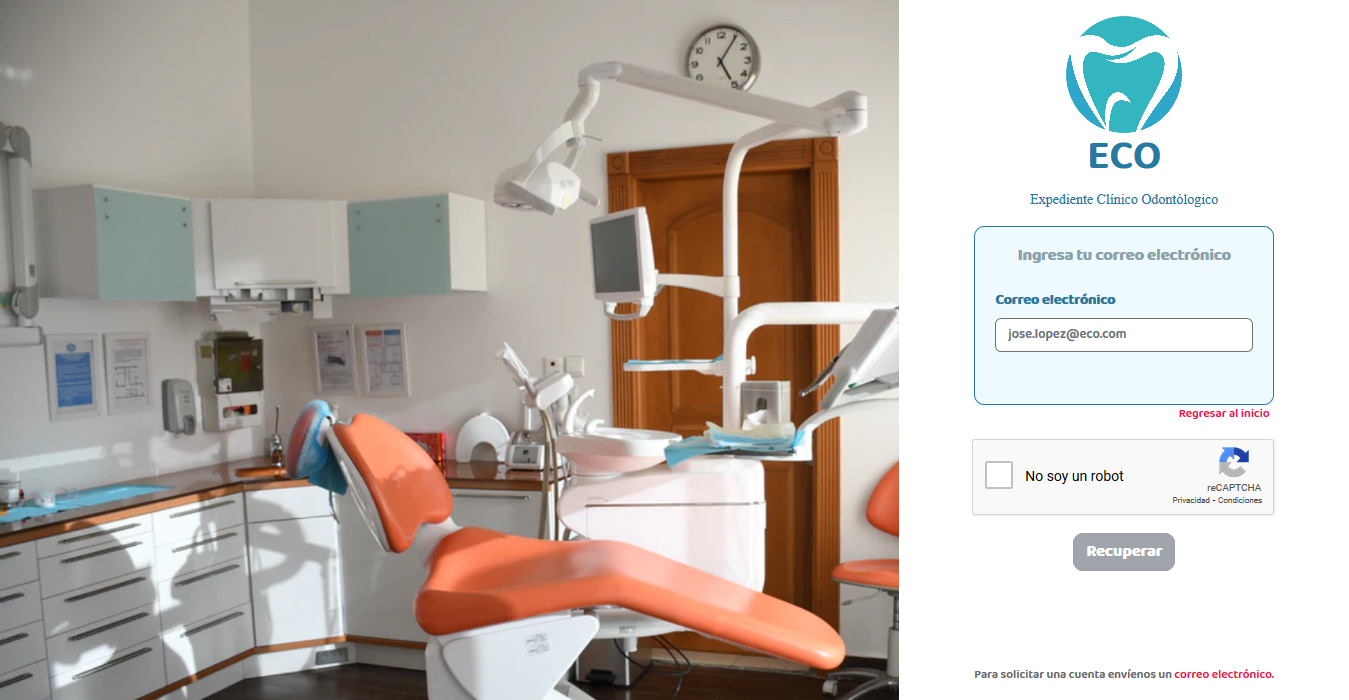


Figura 4. Recuperar contraseña

Pantalla para solicitar una cuenta, estará disponible para usuarios anónimos en la cual se podrá solicitar una cuenta proporcionando la información requerida dentro del formulario.



Figura 5. Solicitar cuenta.

Pantalla inicial para usuarios con rol de super usuario, en la cual se podrá registrar a usuarios con rol de administrador.



Figura 6. Home super usuario

Pantalla inicial para usuarios con rol de administrador, en la cual se podrá registrar a odontólogos y pacientes (indicando a que odontólogo pertenece).

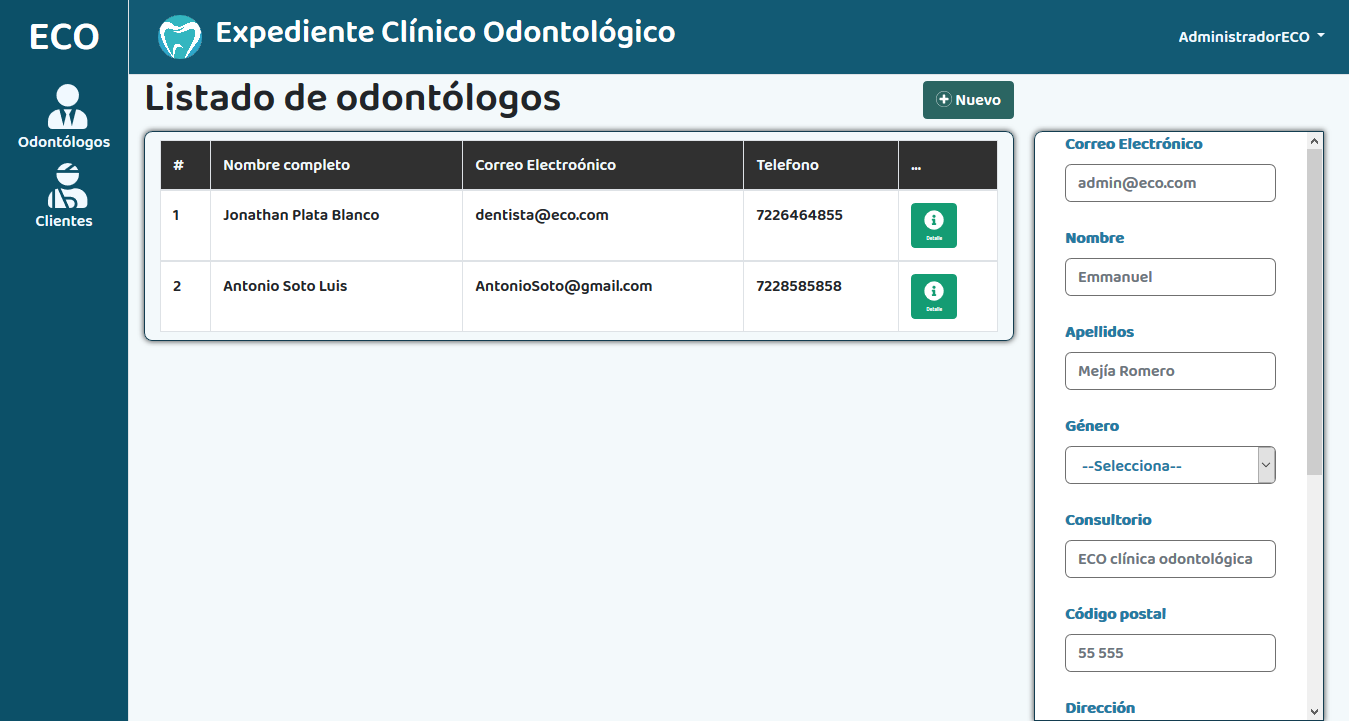


Figura 7. odontólogos modulo administrador

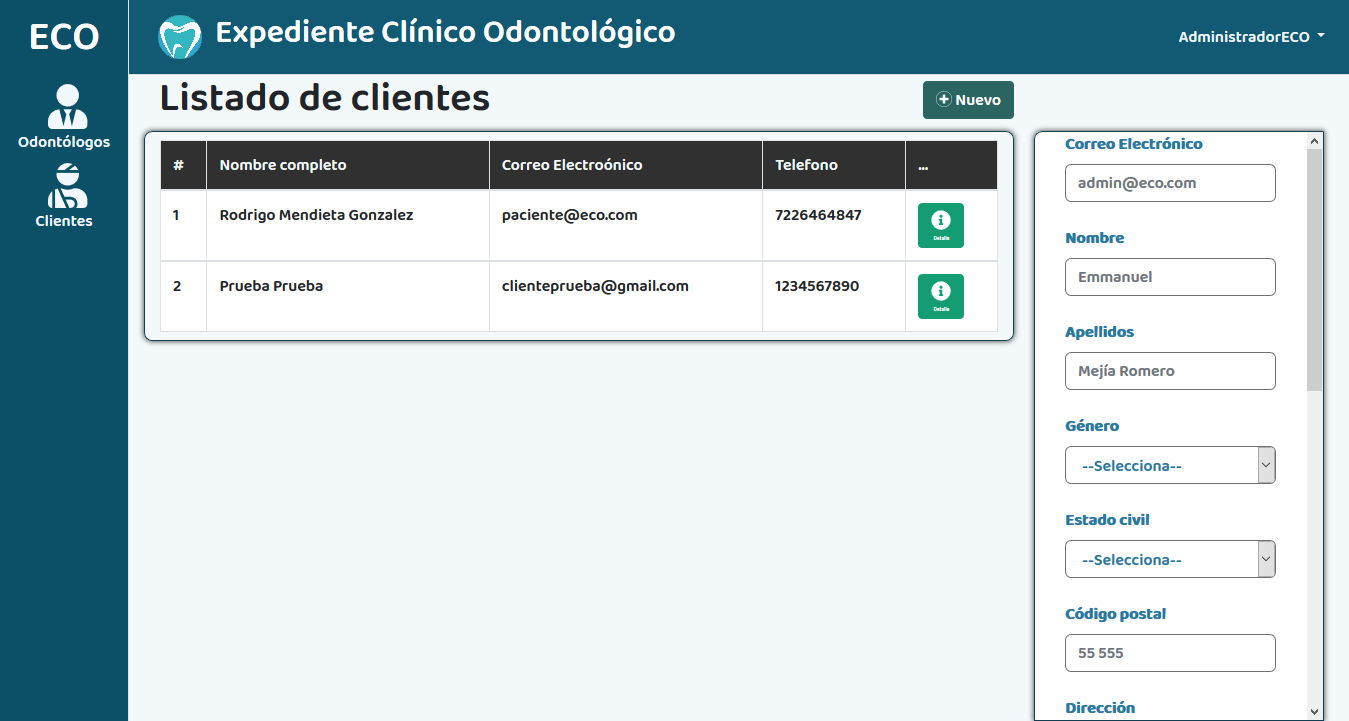


Figura 8. Paciente modulo administrador

Pantalla para configurar datos de usuario con rol administrador.

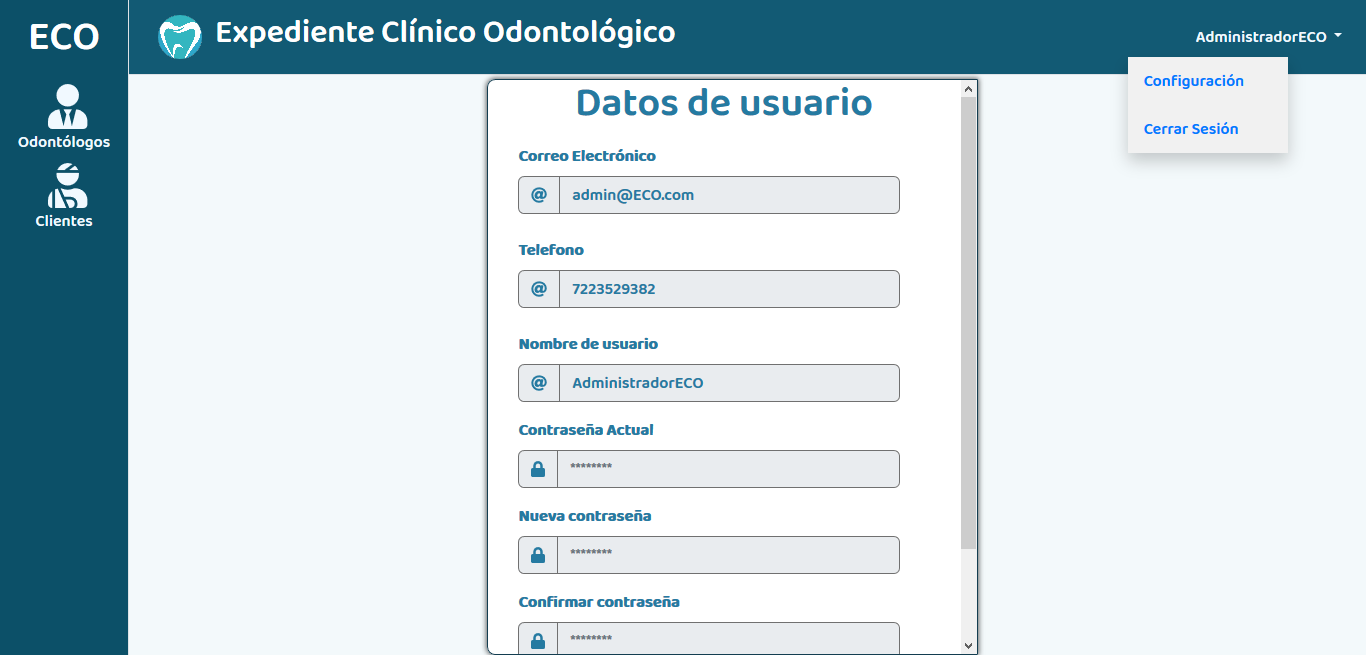


Figura 9. configuración de cuenta módulo administrador

Pantalla inicial para usuarios con rol de odontólogo, en la cual se podrá registrar, modificar y listar pacientes.



Figura 10. Paciente módulo odontólogo.

Pantalla para agendar citas, el odontólogo podrá registrar, modificar (se podrá modificar siempre y cuando no haya sido atendida la consulta) y listar las consultas agendadas para sus pacientes.



Figura 11. Consultas

Pantalla para dar seguimiento a las consultas.



Figura 12. Seguimiento de consultas

Pantalla para configurar los datos de usuario con rol de odontólogo.



Figura 13. Configuración de cuenta modulo odontólogo

Pantalla inicial de usuario con rol paciente, en la cual podrá ver las consultas y visualizar el detalle de esta.



Figura 14. Historial

Pantalla de consultas, se podrá solicitar una consulta y ver las citas que tiene agendadas.



Figura 15. Consultas modulo pacientes

Pantalla para configurar datos de usuario con rol de paciente.



Figura 16. configuración de cuenta modulo paciente

# **REFERENCIAS BIBLIOGRAFICAS**

# **REFERENCIAS WEB**

Alicante, U. d. (1996-2020). *si.ua.es*. Obtenido de https://si.ua.es/es

aplicaciones, D. d. (2020). *grupodesisa.mx*. Obtenido de https://grupodesisa.mx/software-edensys

ARSYS. (2020). *arsys.es*. Obtenido de https://www.arsys.es

codigofacilito. (2020). *codigofacilito.com*. Obtenido de https://codigofacilito.com/articulos/que-es-vue

docs, M. w. (2005-2020). *developer.mozilla.org*. Obtenido de https://developer.mozilla.org

einatec. (2019). *www.einatec.com.mx*. Obtenido de https://einatec.com/

MediSel. (2020). *expedienteclinico.mx*. Obtenido de https://expedienteclinico.mx/

Mercadeo. (2020). *Mercadeo.com*. Obtenido de http://www.mercadeo.com

Microsoft. (2020). *dotnet.microsoft.com*. Obtenido de https://dotnet.microsoft.com

Montoya, N. P. (10 de Octubre de 2005). *Dialnet.* Obtenido de https://dialnet.unirioja.es/servlet/articulo?codigo=5599263

S.L, O. (2018-2020). *openwebinars.net*. Obtenido de https://openwebinars.net

ScrumManager. (5 de Marzo de 2013). *scrummanager.net.* Obtenido de https://www.scrummanager.net/bok/index.php?title=Modelo\_original\_de\_Scrum\_para\_desarrollo\_de\_software

Solutions, M. (2014). *dindoc.mx*. Obtenido de http://dindoc.mx/

team, B. (s.f.). *getbootstrap.com*. Obtenido de https://getbootstrap.com/

weebly. (2020). *ingenieriadesoftwaretdea.weebly.com*. Obtenido de https://ingenieriadesoftwaretdea.weebly.com