|  |
| --- |
| UNIVERSIDAD TECNOLOGICA NACIONAL |
| Formulario de Arquitectura |
| Atención Odontológica |
|  |
| **Frattin Juan 71939**  **Grober Luciana**  **Marin Agustin 71940**  **Ulbrich Danilo 71933** |
|  |

|  |
| --- |
|  |

# Objetivo

Nuestro principal objetivo es el desarrollo de un sistema que facilite el manejo de datos de pacientes en un consultorio odontológico, ya sea uno en el que cuente con secretaria y más de un odontólogo o una en la que atienda solo uno.

La idea vino desde un integrante del grupo que conocía la necesidad debido a un familiar cercano que trabaja en esta área. Debido a esto abordamos la problemática teniendo en cuenta las necesidades reales de estos, y queriendo brindar la mejor solución posible.

No fue la única idea que se propuso, ya que a nivel conceptual era bastante más complicada que realizar un simple programa de facturación. Debido a esto tuvimos que interiorizarnos en la forma que los odontólogos llevan a cabo el seguimiento de sus pacientes y la utilización de formularios especiales (odontograma). También costo adaptarse a la utilización de términos que no eran precisamente claros (elemento, prestación), pero que en definitiva hacían al proyecto.

Una vez definidos los conceptos que hacían al proyecto el desarrollo del mismo se dio de forma normal, salvo por casos aislados.

# Datos a Validar

En el proceso de desarrollo del proyecto nos vimos ante la necesidad de corroborar información sensible del usuario y del paciente. Estos datos debían ser correctos y de cierto tipo, por lo que se definieron todos los tipos de datos específicos a cada una de las tablas, como también el largo de los mismos. Esto se lleva a cabo del lado de la base.

En el inicio del proyecto nos vimos ante la necesidad de plasmar la estructura de la base de datos, pero al no tener una imagen práctica del sistema o una visión clara de la envergadura de este contábamos con una tabla Pacientes y otra de Usuarios, que básicamente contaban con casi los mismos elementos en ambas. En definitiva se fusionaron ambas tablas dejando la tabla Usuarios.

Del lado del frontend nos vimos con la dificultad de desarrollar una validación por el número de matrícula del Odontólogo, pero no así para el caso del Administrador o la Secretaria. Por lo que se modificó este dato en la tabla Usuarios para que acepte valores null en estos casos. Por otro lado en los formularios de carga de datos (ABM) se realiza una validación del tipo de dato y que no estén vacíos algunos campos obligatorios, esto se hace mediante el método validar\_campos().

En un principio se pensó en utilizar una tabla Arreglos para llevar un control de las prestaciones realizadas a cada elemento y cada cara, pero luego de comenzar a desarrollar el sistema nos dimos cuenta que esta no era necesaria, ya que estas tablas se relacionaban entre si directamente en la tabla DetalleHistorial. Por lo que Arreglos quedo obsoleta y se la dio de baja.

# Capas

En el desarrollo de este proyecto se nos presentó la modalidad de trabajar con capas, en un principio algo tedioso, ya que no estábamos acostumbrados a programar de esta forma. Pero ante la posibilidad de aprender a manejar esta nueva técnica de desarrollar decidimos hacerle frente y modificar el sistema ya existente para llevarlo a capas.

Este proceso de pasar el programa a capas se dificulto en algunos momentos ya que no todos los integrantes estaban al tanto de como concebir las capas. Es por esto que tuvimos varias demoras e inconvenientes, hasta determinar la causa de los errores y poder solucionarlos.

Luego de que todos los integrantes del grupo se integraron con la modalidad, el desarrollo se hizo mucho más ameno y más rápido.

# Estética

Conceptualmente el sistema consistió en un programa de fácil utilización para personas con pocos conocimientos en informática o en el manejo de ordenadores. Por lo que decidimos que el sistema tenga una apariencia agradable y simple, de forma que sin mucha experiencia pudieses utilizar el programa.

Entonces el diseño se utilizó únicamente con botones a la vista y con ayuda de imágenes sobre estos para facilitar la conceptualización de la función detrás de estos. Así también los formularios se pensaron de tal forma que la distribución de estos sea agradable para el usuario.

El sistema cuanta con un menú principal desde el cual se puede acceder a las demás funciones, esto fue pensado desde un inicio y no hubo mayores inconvenientes en el desarrollo del mismo.

# Mensajes

Los mensajes con los que cuenta el sistema son los esenciales para la comprensión del funcionamiento del mismo, que devuelva al usuario un feedback si este está realizando alguna acción incorrecta o si realiza una carga exitosa de datos.

El contenido de los mensajes en si no es extenso sino más bien resumido y simple, para que el usuario entienda rápidamente y tome acciones ante tales casos. Los mensajes de carga de datos exitosa solamente son a modo informativo para que el usuario sepa que se realizó de manera correcta la carga.

El programa también cuenta con mensajes de confirmación en el caso del cierre del mismo por si el usuario accidentalmente presiono para salir del mismo. Estos son una medida puramente preventiva para el usuario.

Durante el proceso de programación del sistema utilizamos los mensajes para visualizar las salidas o las sentencias de SQL, de manera que nos servían como banderas de ejecución. Estas por supuesto deben ser quitadas del programa en su versión final ya que altera la experiencia del usuario en el sistema.

# Testeo

Las pruebas realizas al programa se fueron realizando durante el proceso de programación, algunos errores fueron forzados a propósito y otros no tanto. Pero siempre buscando que el sistema funcionara de forma estable.

Se realizó una prueba de funcionamiento con un usuario inexperto para verificar que el sistema fuese de simple manejo y que no hubiera mayores complicaciones. El usuario pudo acceder al sistema sin ninguna dificultad y tras recorrer algunas funciones pudo comprender la utilización del mismo él solo.

Creemos que las pruebas realizadas hasta la fecha no son todas las posibles que se podrían realizar pero que son las necesarias para que el programa funcione correctamente y pueda ser utilizado.

# Distribución del trabajo

La conformación del equipo en un principio fue de tres integrantes, pero a pedido de una compañera que no tenía grupo accedimos a ser cuatro. Esto implicó una mayor organización ya que coordinar los avances del grupo era complicado, debido a la imposibilidad de hacer funcionar git en la mayoría de las maquinas.

Más allá de estas complicaciones el equipo realizo múltiples reuniones para debatir temas claves del sistema (base de datos, funcionamiento, etc) y posteriormente la división de tareas correspondiente a cada miembro.

Algunos miembros se atrasaron en el plazo de entrega de sus respectivas tareas debido a distintos motivos, pero el proyecto avanzo igual gracias al esfuerzo conjunto del equipo.

Creemos que el desarrollo de este programa de forma conjunta nos enseño como se debe manejar un grupo en un proyecto, dividiendo tareas y cumpliendo plazos. Algunos del grupo contamos con experiencia en metodologías agiles y creemos que eso pudo ser bastante útil a la hora de organizar el trabajo.