

Consultorio Odontológico del Valle

Documento de la arquitectura

Programación de aplicaciones visuales I

Bermejo, Dagotto, Lalomía y Parrucci
13-11-2020

Índice

Compartición de la base de datos	1
Compartición de código	1
Interfaz.....	2
Acuerdos de programación	3
Conocimientos adquiridos	3
Problemas y soluciones	4
Roles	4

Compartición de la base de datos

Con respecto a la base de datos se intentó trabajar en conjunto utilizando SQL Server Management Studio, que es una aplicación de software lanzada por primera vez con Microsoft SQL Server 2005 que se usa para configurar, administrar y administrar todos los componentes dentro de Microsoft SQL Server. Es el sucesor del Enterprise Manager en SQL 2000 o antes., ya que, en grupo formulamos las tablas necesarias con sus atributos porque decidimos no hacer un dominio ya dado sino formular el propio que se basó en un consultorio odontológico integral. Entonces constantemente teníamos cambios. La manera en la que nos manejamos fue la siguiente: decidimos que alguna integrante del grupo compartiera su pantalla por Discord, que es una aplicación freeware de VoIP y chat por texto, para que entre todas podamos ir construyendo la base de datos relacional. La persona que compartía la pantalla en ese momento era la encargada de escribir el código, una vez que estaba finalizado, guardamos el script de la base de datos (secuencia de comandos o guión es un término informal que se usa para designar a un programa relativamente simple), y lo compartimos entre todas las integrantes.

Luego, en caso de que alguna realizara algún tipo de modificación de manera individual debía avisar a las demás integrantes y así cada una lo agregaba si eran modificaciones pequeñas y si no, se volvió a compartir el script.

Compartición de código

El código se compartió mediante GitHub, que es una forma de alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador. GitHub nos permitía trabajar de manera individual en horarios en donde cada una pudiera trabajar, sin necesidad de que todas estemos conectadas para avanzar en el proyecto. Utilizamos la compartición de un repositorio para ello, llamado "Consultoriotp". El cual tiene un link de acceso que es:

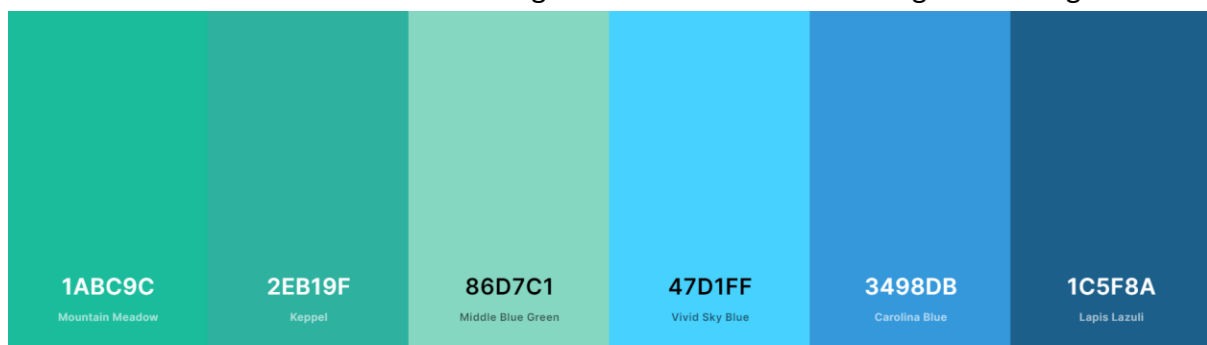
<https://github.com/LaliParrucci/Consultoriotp.git>

GitHub tiene un problema en particular cuando dos personas intentan acceder a las mismas líneas de código y hacer commit de las mismas. En estos casos, Git no puede determinar automáticamente qué es correcto. Los conflictos sólo afectan al desarrollador que realiza la fusión, el resto del equipo no se entera del conflicto. Git marcará el archivo como que tiene un conflicto y detendrá el proceso de fusión. Entonces el desarrollador es el responsable de resolver el conflicto. El control de versiones nos ayuda a manejar estos conflictos al darnos herramientas para resolver cambios que se hayan solapado. Si tenemos uno, lo resolvemos analizando el código que comparten líneas y decidir cuál es el que debería quedarse o si se pondrían en líneas diferentes. También para tratar de evitar que el conflicto se genere, tratamos de dividirnos las tareas, para que cada una tuviera sus horas al día para realizar el

trabajo, y una vez finalizado se hiciera commit (se refiere, en el contexto de la ciencia de la computación a la idea de confirmar un conjunto de cambios provisionales de forma permanente) y otra compañera pudiera hacer el push (se usa para cargar contenido del repositorio local a un repositorio remoto. El envío es la forma de transferir commits desde tu repositorio local a un repositorio remoto). También trabajamos en conjunto en diversas ocasiones para resolver problemas más complejos, y esto lo llevamos adelante mediante llamadas grupales, una integrante compartía su pantalla y entre todas resolvimos los problemas que se nos presentaban.

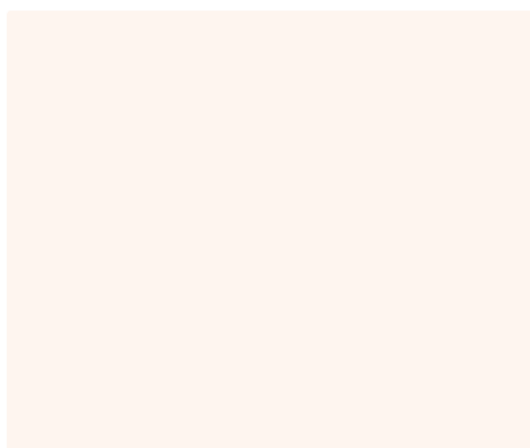
Interfaz

Con respecto a las interfaces, intentamos elegir una interfaz amigable, que sea sencilla de manejar y de visualizar para el odontólogo o para la secretaria, con colores que se adapten al ambiente de un Consultorio Odontológico como se muestra en la siguiente imagen.



También, usamos el siguiente tono para las letras, también para el menú, el logo y otros detalles que se visualizarán de mejor forma en el manual de usuario:

Acerca de #FEF5EF



HEX	#FEF5EF
NOMBRE	-
TONO	naranja pálido
RGB	254 245 239
HSV	24° 5% 99%
CMYK	0% 3% 5% 0%

Acuerdos de programación

Los acuerdos de programación se basaron en clases teóricas, en donde nos enseñaron a dividir las responsabilidades en diferentes capas; la capa de datos, la de lógica de negocio y la capa de presentación en este trabajo se vieron notablemente diferenciadas, de hecho lo implementamos de una manera mucho más modularizada, respetando estándares utilizados en la lógica de programación, para efectivizar el código y que sea legible para otras personas que quieran modificarlo o necesiten verlo:

Lógica (BusinessLayer): La capa de negocio expone la lógica necesaria a la capa de presentación para que el usuario, a través de la interfaz, interactúe con las funcionalidades de la aplicación.

Acceso a datos (DataAccessLayer): La necesidad de vincular los datos guardados en una base de datos relacional, con los objetos de una aplicación orientada a objetos, determinó la aparición del concepto de persistencia de objetos. Siguiendo el estilo de desarrollo en tres capas, la persistencia queda recogida en su propia capa, separada de la lógica de negocio y de la interfaz de usuario.

Entidades (Entities): Aunque aparentemente es una cuarta capa, realmente no lo es. Se encarga de contener todos aquellos objetos (clases) que representan al negocio, y esta es la única que puede instanciarse en las 3 capas anteriores, es decir, solo ella puede tener comunicación con el resto pero su función se limita a únicamente ser un puente de transporte de datos.

Esta capa complementa a la Capa de Negocio **Presentación (GUILayer):** Mediante la capa de presentación se separa la interacción del usuario respecto a la lógica de negocio. Todo esto también se realizó para evitar duplicación de código respetando los dos principios básicos fundamentales de programación, alta cohesión y bajo acoplamiento.

Conocimientos adquiridos

En general, todo lo visto en la materia fue aprendizaje, ya sea desde aprender a usar GitHub, un programa para compartición de código, en donde pudimos trabajar cada una en particular, y poder compartirlo en un repositorio, hasta aprender a generar estadísticas en un programa de escritorio, que fue el último tema visto en la materia. Y además, aprender a trabajar en equipo, ya que es la forma en la que deberíamos trabajar si nos dedicamos a desarrollar software.

Logramos aprender a realizar ABMC, que sería seleccionar las clases o las tablas que necesitan tener un alta, una baja, una modificación o algún tipo de consulta.

Aprendimos a manejar transacciones. Una transacción es un proceso en el cual se involucran dos tablas o más y se modifican al mismo tiempo, quedando realizada si se hace por completo, ya que si la misma no se completa si queda a mitad de esta y debería quedarse en la situación en donde estaba cuando empezó.

Por último, aprendimos a generar reportes y estadísticas como antes hemos mencionados, para lo cual nos comunicamos con un odontólogo en particular, consultando al mismo, que

estadísticas y reportes les sería útil para llevar un control sobre su consultorio, y en base a su respuesta y a la aceptación de nuestra docente a cargo se realizaron.

Problemas y soluciones

Los inconvenientes que tuvimos fueron al principio con respecto al funcionamiento del programa usado para compartir código, ya que no entendíamos su funcionamiento, ni cómo manejar conflictos. Los conflictos suceden cuando dos o más personas intentan acceder al código en la misma línea y realizan un push sin haber hecho un pull anteriormente. **git pull** es uno de los muchos comandos que se encargan de "sincronizar" el contenido remoto. El comando **git remote** se utiliza para especificar los extremos remotos sobre los que operarán los comandos de sincronización. El comando **git push** sirve para cargar contenido en un repositorio remoto. Entonces, tuvimos que aprender a resolverlos también y a entender su funcionamiento.

Hubo complicaciones con el entendimiento de las transacciones ya que al ser un programa hecho por nosotras mismas tuvimos la responsabilidad de averiguar relaciones, tablas, y atributos que necesitamos y la forma en la que se desarrolla. Luego de solucionar el problema de compartir el código, los problemas que se nos presentaron fueron los que se suelen presentar al aprender a programar cosas novedosas, como por ejemplo aprender a buscar opciones de codificación para algunas funciones adicionales a las vistas en clases, las mismas no fueron difíciles de solucionar, pero si alguna tenía más dificultad que otra, nos reunimos a ver el problema entre todas y ayudarnos a solucionarlo compartiendo pantalla, o bien, pactamos una reunión de zoom con la docente para evacuarlas además de las clases virtuales dadas.

Roles

La distribución de roles fue en general bastante compartida ya que todas deberíamos abarcar la misma cantidad de temas y conocimientos, por lo general en un equipo hay varios roles pero tratamos de que sea lo más unificado posible. Una persona se encargaba del diseño visual y la presentación del producto, otra persona en la codificación o especialización del código, otra persona en la comunicación con la tutora para reuniones, seguimiento y consultas sobre el desarrollo del mismo, y otra sobre la investigación del funcionamiento del programa en general, pero básicamente todas intentamos abarcar todas las funciones, o hacer reuniones en grupo para dialogar sobre el funcionamiento de las transacciones, u operaciones necesarias para el producto, como informes y estadísticas. Para luego quizás, dividir responsabilidades, una vez que todas logramos comprender cómo realizar por ejemplo un listado o un ABMC, nos repartimos las tareas y lo hacíamos de forma individual para toparnos con problemas y forzarnos a aprender a solucionarlos de manera particular.