

Módulo administración

El módulo de administrador nos permite diferenciar funciones o roles de cada usuario, el administrador se va a encargar de otorgar los derechos de acceso necesarios para cada persona que utilice su cuenta y además podrá realizar un seguimiento del usuario que haya efectuado algún tipo de pregunta. Para poder efectuar todo lo anterior se realizó una lluvia de ideas de las cuales se obtuvo lo siguiente:

Archivar preguntas

1. Administrar el perfil y los derechos de acceso de cada usuario, permitiendo modificar el rol de un usuario cuando sea necesario. Los administradores se pondrán dentro de una estructura de datos para tener más organizado los datos dentro de la aplicación, como porejem:
 - a. Un árbol n-ario en donde las jerarquías se puedan ver en cuanto un administrador coloca un usuario como administrador, así el que coloca al usuario como administrador sería el papá de este.
 - b. Una pila de datos en donde se pueda ver que administrador fue el primero el ser agregado y observar las jerarquías que tienen por más que tengan los mismos derechos de edición.
 - c. Una lista enlazada sería una estructura que ayude a organizar a los administradores conforme se va agregando al módulo administrativo, así como una lista de tiempo.
 - d. Un ArrayList que agregue a los usuarios que accedieron al módulo de administración para tener la información de que usuarios fueron.
 - e. Implementar una base de datos en donde se pueda administrar todos los usuarios, administradores, preguntas, respuestas, etc.
2. Limitar el acceso, de forma que los usuarios solo puedan encargarse de sus funcionalidades
 - a. Se usará un atributo de tipo boolean el cual se encargue de verificar si es un administrador o un usuario.
 - b. Se podrá buscar en una lista que contenga todos los administradores del sistema diferenciados por Studios.
 - c. Se buscará en la contenedora de los administradores de dicho Studio para poder separar la información dependiendo si son administradores o usuarios.
 - d. Se tendrá una interfaz diferente para que los administradores puedan hacer los cambios que requieran, como eliminar preguntas, además de poder ver información que solo está disponible para los administradores.
3. El administrador se debe encargar de la información sobre el entorno, para lograrlo el administrador debe estar en constante monitoreo, para que capte oportunamente cada una de las funcionalidades
 - a. Al administrador se le notificarán las acciones que realicen los usuarios por medio del correo electrónico ingresado a la hora de registrarse.
 - b. Cualquier consulta realizada por el usuario se almacenará en una lista la cual podrá ser controlada desde el perfil del administrador.
 - c. Habrá una sección de reporte de problemas, la cual el administrador estará en capacidad de acceder a él para poder enviar a los programadores en busca de la mejor solución.
4. Archivar preguntas: Se implementará un atributo lógico que permita ocultar la información de la pregunta que permitirá verse como archivada, y el atributo se podrá cambiar con permiso del administrador para poder desarchivar la pregunta.
 - a. Se podrá agregar en una estructura de árbol n-ario en donde la jerarquía será el orden en el que se van archivando las preguntas.
 - b. Se agregaran a una lista enlazada en donde se relacionan por el orden en que se van archivando.

- c. Se agregarán a un ArrayList que permita tener un atributo como un comentario que permita poner por qué fue archivada esa pregunta.
 - d. Se agrega a una cola, la cual va archivando cada una de estas por orden de llegada.
 - e. Se agrega a una pila la cual va archivando cada una de estas por orden de llegada, el cual se puede visualizar de forma que la última que llega es la primera que se visualizará.
 - f. Se implementará un archivador lógico, el cual será un atributo que se le cambiará el valor según el estado de la pregunta.
- 5. Eliminar preguntas: Se implementará un atributo lógico que permita ocultar la información de la pregunta que permitirá verse como borrada.
 - a. Se da la posibilidad de que la pregunta sea eliminada lógicamente, esto realizado por medio de un boolean, el cual se añadirá a una lista donde permanecerán durante un determinado tiempo.
 - b. Se podrá agregar en una estructura de árbol n-ario en donde la jerarquía será el orden en el que se van eliminando las preguntas.
 - c. Se agregarán a una lista enlazada en donde se relacionan por el orden en que se van eliminando.
 - d. Se agregarán a un ArrayList que permita tener un atributo como un comentario que permita poner por qué fue eliminada esa pregunta.
 - e. Se agrega a una cola, la cual va eliminando cada una de estas por orden de llegada.
 - f. Se agrega a una pila, la cual se va eliminando cada una de estas por orden de llegada, el cual se puede visualizar de forma que la última que llega es la primera que se visualizará.
 - g. Eliminar la pregunta del todo y escribir la causa por la que se está borrando la pregunta. La información de la pregunta se elimina por completo del sistema y se le informa a la persona que hizo la pregunta por correo. En el cuerpo del mensaje va la razón de por qué se elimina.
- 6. Generar un reporte de la información que el administrador crea pertinente o con el filtrado que él quiera usar (preguntas, studios, respuestas a una pregunta, etc) y lo pueda ver en un archivo de Excel.
 - a. Utilizar el api de Google sheets que permite enviar los datos a un excel en línea con la cuenta del administrador.
 - b. Utilizar un gridview de la aplicación de visual studio.
- 7. Teniendo un reporte generado previamente, y si el administrador lo requiere, exportar un archivo Excel.
 - a. Usando el api de Google sheets podemos fácilmente dar solución a la exportación de este archivo.
 - b. Utilizar un gridview de la aplicación de visual studio para generar el Excel en el computador que se está utilizando.
- 8. Para crear un studio previamente se necesita validar si el cliente es un administrador, al tener esta información se procederá a crear el studio si el administrador así lo desea
 - a. se creará una nueva base de datos de tipo estudio, el cual tendrá diferentes tipos de atributos, esta base de datos se añadirá a una base de datos de estudio, y por último se adicionará al servidor.
 - b. se creará un nuevo Studio desde la interfaz, lo cual permitirá a un administrador hacer cambios a la implementación de la página.

- c. Se le entregará a cada administrador la implementación de la clase Studio, para que pueda hacer las modificaciones requeridas por cada uno.

Módulo administración

En primer lugar, se ha decidido omitir algunas ideas, como las expresadas a continuación:

- 1.(b , c) se omitieron estos Subíndices, ya que necesitamos utilizar el que mejor cumpla la funcionalidad, en este caso por lo siguiente: va tomar un rumbo muy confuso a la hora de establecer la relación Administrador-Usuario. Estas soluciones toman un poco más de tiempo al compilar, además, al momento de organizar de forma jerárquica la solución va a ser deficiente.
- 2.(b , c) se omitieron estos subíndices, si usamos una contenedora para verificar el tipo de usuario al cual se procede buscar, se tendría que hacer una búsqueda por cada uno de los usuarios registrados en la base de datos, lo cual tomaría un tiempo de procesamiento mayor. Al igual que la anterior idea, la opción de buscar en cada uno de los estudios un usuario en particular, con el objetivo de verificar el tipo que es este, esta opción tomaría un tiempo de compilación mayor al deseado.
- 3. (b) Agregar cada consulta que se haga a una lista de datos ocuparía demasiado espacio, y no habría manera de consultarla de manera óptima, por esta razón la solución es demasiado limitada teniendo en cuenta lo grande que es el proyecto, además de que el programa se basa en que los usuarios puedan hacer y contestar preguntas la funcionalidad que otorga es muy condicionada y el proyecto pide más cosas que almacenar preguntas.
- 4. (a, b, d, e) Todas estas propuestas previas serán descartadas y se implementará una propuesta mejor, en general por el limitado espacio con la que cuentan cada uno, por ejemplo las propuestas *d* y *e* son de espacios muy limitados, ya que son estructuras cola y pila respectivamente, si se desea archivar muchas preguntas estas estructuras podrías generar un overflow y es algo que no se desea. Por otro lado, las propuestas *a* y *b* almacenan una mayor cantidad de datos que las descritas previamente. El árbol n-ario depende de una variable para poder organizar lo que se quiere agregar dentro del árbol y es algo que no se tomará en cuenta. Las listas enlazadas, como lo describe el nombre, se relacionan los datos que la conforman, pero la acción de archivar una pregunta no debería de estar relacionada con el archivar otra pregunta.
- 5(a, b, c, e, f) Estas propuestas no cumplen con lo necesario para poder realizar lo que el cliente solicita para poder implementarlo. La eliminación lógica no sería útil ya que tiene la misma funcionalidad que archivar una pregunta entonces para eso se tiene el requerimiento de archivar las preguntas. Las propuestas b y c son descartadas ya que al igual que el requerimiento pasado no se debe tener algo que relacione la eliminación de una pregunta con otra o tener un factor para clasificar la eliminación de las preguntas en un árbol n-ario, y se descartan las ideas e y f ya que al igual que el requerimiento pasado tienen poca capacidad de almacenamiento y generarían un overflow siendo algo que no se quiere en el desarrollo del proyecto.

- 6. Se elimina la propuesta de gridview por parte de la aplicación de visual ya que la persona que requiera hacer esto no tendrá acceso al código, y por ende esta funcionalidad no servirá en el momento que el administrador quiera hacer el reporte.
- 7. La idea del gridview también será descartada ya que ningún usuario (persona que utilice la aplicación) deberá tener acceso al código de la aplicación.
- 8. la idea de entregarle a cada administrador el permiso de modificar el código hecho por los programadores es muy descabellada, ya que ningún usuario deberá tener acceso a este.

Síntesis Reflexiva

Modulo Administrador

Todo ingeniero es un solucionador de problemas. Su objetivo es satisfacer una necesidad que ha sido expresado en términos coloquiales, y traducirlo a un vocabulario sistemático es una de sus funciones. El método de la ingeniería es esa terminología que se usa. Para llegar a una solución, se debe seguir un algoritmo como lo es; determinar un numero razonable de soluciones posibles del problema en cuestión; posteriormente se debe evaluar cada una de dichas soluciones y finalmente escoger la mas apropiada que satisfaga la necesidad mencionada.

Tras desarrollar este procedimiento nosotros como ingenieros logramos entender el problema y darle manejo a la solución, los resultados que obtuvimos nos permitieron hallar las funcionalidades que nos correspondían como módulo de administrador. Durante el proceso de recopilación de información se encontraron varias alternativas, unas por ideas ya existentes y otras usando el aprendizaje activo, lo importante es que todas estas se enfocaban en dar solución al problema. En algunas situaciones escoger las ideas que satisficieran la cuestión fue un poco complicado, pero se logro llevando un buen análisis entre cada una de ellas, esto se vio reflejado cuando cada funcionalidad tuvo de forma efectiva una solución.

Sin duda el trabajo en equipo es un factor importante al momento de querer desarrollar algún trabajo, la comunicación asertiva es uno de los pilares para poder desarrollar de mejor manera las tareas propuestas, además de la escucha y entendimiento por parte de cada persona del equipo, la buena comprensión en el desarrollo del método de la ingeniería permitió la generación de ideas. Se escogió un líder dentro de nuestro modulo, se delegaron tareas y todo fluyo correctamente, nosotros creemos que esto fue uno de los pilares para llevar acabo todo esto.