

Informe sobre programa de agendamiento de citas médicas

Autores:

Jordy Aguilar

Josué Valencia

Eric Mullo

09 de julio del 2023

Introducción:

La Fundación Salud para todos es una organización sin fines de lucro que se dedica a financiar tratamientos para pacientes con enfermedades raras en Ecuador. Sin embargo, debido a la pandemia del COVID-19, la emergencia sanitaria encontró un sistema de salud con condiciones estructurales de fragmentación y segmentación, menores recursos públicos destinados a la salud y condiciones económicas poco favorables que condicionaron la respuesta frente a la crisis social, sanitaria además las instituciones de salud han experimentado una disminución significativa en sus ingresos, lo que ha dificultado su capacidad para financiar parcial o completamente los tratamientos necesarios. Ante esta situación, las fundaciones de salud han implementado una unidad médica gestionada y administrada por ellos mismos, con el objetivo de optimizar los recursos económicos y garantizar el acceso a tratamientos completos para los pacientes. (Trabajo, 2021)

Sin embargo, la puesta en marcha de esta unidad médica ha presentado varios desafíos, uno de ellos es el agendamiento de citas, que actualmente se realiza de manera manual por el personal de la fundación y ha demostrado ser deficiente. Esta deficiencia ha llevado a la pérdida de información y confusión en las citas agendadas, lo que afecta negativamente la eficiencia y la calidad del servicio brindado.

En este contexto, el presente trabajo tiene como objetivo abordar el problema del agendamiento de citas deficiente en la unidad médica de la Fundación Salud para Todos desde un enfoque técnico y operacional. Se realizará un análisis detallado de las variables de entrada, salida y del proceso asociado al agendamiento de citas. A partir de este análisis, se propondrán soluciones integrales que incluyan métodos y herramientas tecnológicas específicas, teniendo en cuenta las restricciones técnicas y las regulaciones legales vigentes en el ámbito de la salud pública.

Justificación:

La justificación de este trabajo radica en la importancia de mejorar el agendamiento de citas en la unidad médica de la Fundación Salud para todos mediante una solución eficiente y precisa, se busca optimizar los recursos disponibles y garantizar un proceso de agendamiento de citas más ágil y confiable para los pacientes. Esto contribuirá a brindar un mejor servicio de atención médica, asegurando que los pacientes reciban los tratamientos necesarios en el momento oportuno.

Objetivos específicos:

- Evaluar de forma más profunda los problemas que presentan la fundación Salud para Todos, y de esta forma lograr elaborar un programa que se adapte de mejor forma a las necesidades de los pacientes ecuatorianos.
- Verificar el funcionamiento y validez del sistema de agendamiento para poder poner en marcha el sistema, una vez que esto sea comprobado, se podrá presentar el programa de escritorio y estaría listo para que este ya pueda ser ejecutado en las unidades de salud ecuatorianas, ayudando en la simplificación de errores y perdida de información.
- Implementar método para la gestión de citas digitales para que permita la automatización del programa de citas de manera competente e indispensable.

Objetivo general:

Mejorar la eficiencia y calidad de los servicios de salud en Ecuador mediante la evaluación de los problemas existentes en las instituciones de salud pública y privada, y la creación de un programa de escritorio que se adapte a las necesidades de los pacientes ecuatorianos, verificando el funcionamiento y validez del sistema de agendamiento y su implementación en las unidades de salud del país, con el fin de reducir errores y pérdida de información.

Selección y análisis del problema a ser resuelto:

❖ Problema:

Las unidades de salud ecuatorianas presentan una variedad de problemas al momento del agendamiento de citas médicas, debido a que se conserva la forma antigua para realizar este tipo de procesos, que contas de papel, lápiz y una persona encargada de llevar el control de los agendamientos; sin embargo este es un grave problema publico debido a que se llegan a perder gran cantidad de datos importantes de los pacientes ecuatorianos, y además toma mucho tiempo el buscar una cita médica agendada en una gran cantidad de papeles, por esta razón se realizó el programa de escritorio que ayudará a la fundación Salud para Todos a contabilizar y manejar de una forma más eficaz y eficiente las citas médicas agendadas.

❖ Análisis técnico y operacional:

Variables de entrada:

- Opción (Opción que el usuario escoja para realizar una acción dentro del programa).
- Fecha
- Hora
- Nombre del paciente o descripcion
- Medico
- Numero de cedula o identificación
- Horarios de atención

Se solicita al usuario ingresar la fecha y hora deseada para agendar una cita. El programa mostrará por pantalla los horarios de atención disponibles, permitiendo al usuario seleccionar la hora que prefiera. A continuación, se recopilan los datos del paciente, solicitando su nombre y número de identificación o cédula. Esta información se utilizará para mostrar por pantalla las citas agendadas junto con los datos correspondientes del paciente. Luego, se muestra por pantalla la lista de médicos disponibles, donde el usuario deberá seleccionar al médico con el que desea agendar su cita.

Variables de salida:

- Cita (Esta variable indica el numero de la cita que agendo un usuario, para de esta forma poder llevar la contabilización de las mimas)
- Fecha
- Hora
- Nombre del paciente o descripción
- Medico
- Estado (En esta variable se muestra la disponibilidad del medio que puede ser ocupado cuando la cita se haya realizado con éxito).

Se vuelven a mostrar las variables de entrada ya que al finalizar la agendación de una cita con éxito, luego el usuario podrá visualizar las citas agendadas y al momento de incorporar esta información de las variables ingresadas por el usuario anteriormente, la información de las citas será mucho más clara y estructurada.

Proceso asociado:

- Registro y almacenamiento de la información de los pacientes
- Asignación de citas
- Seguimiento de la disponibilidad de los médicos
- Seguimiento de la disponibilidad de los horarios disponibles

El programa realiza un seguimiento de la información ingresada por el usuario para evitar duplicación de citas entre diferentes usuarios. Durante el proceso de solicitud de información al usuario, el programa mostrará por pantalla las opciones disponibles para que el usuario pueda seleccionar las deseadas. Por ejemplo, se mostrará una lista de horarios y médicos disponibles para agendar una cita.

Propuestas de solución:

1) Implementación de un sistema de gestión de citas en línea:

- Método:

Desarrollo de una plataforma web para computadores y dispositivos móviles que permita a los pacientes programar sus citas de manera autónoma en la para fundación Salud para Todos, esta propuesta de solución permitirá a los usuarios poder agendar una cita medica por medio de una página web que podría ingresar por medio de una URL o del nombre del sitio web que se establezca, es importante recalcar que para que este método funcione de manera efectiva y rápida, el usuario o paciente debe contar con un buen acceso a internet y con un dispositivo de gama media como mínimo para que la pagina no se congele o se demore mucho en cargar.

Para la implementación de esta propuesta de solución se usarán muchas estructuras de programación entre ellas; la estructura de programación secuencial, condicional, bucles, etc. Ya que se busca realizar un código estructurado para la fácil comprensión del programador. En esta propuesta de solución una vez que se haya realizado, las personas podrán acceder al programa por medio de una URL o con el nombre del sitio web.

- Herramientas tecnológicas:

Lenguaje de programación (El lenguaje utilizado es el lenguaje C), para codificar el programa se usa visual studio code versión 17.6; además se usa archivos ejecutables para el almacenamiento de la información, algo que es importante recalcar en esta propuesta de soluciones que para poder implementarla se debe aprender sobre HTML ya que la implementación será sobre una página o sitio web, además se necesitara una computadora con buenas características para poder realizar la codificación o implementación correctamente.

- Restricciones técnicas:

La restricción técnica mas notoria en esta propuesta es la falta de conocimiento en HTML, ya que se debe usar este lenguaje para poder crear la página web en la que se va a implementar el software.

Además de usar HTML se necesitará usar CSS para que la página web tenga una mejor estructura y se vea mucho mas agradable visualmente, por lo que es una clara desventaja ya que no se tiene el conocimiento necesario sobre estos lenguajes de programación, sin embargo, si se aplica esta propuesta, será mucho más fácil y sencillo acceder a usar el programa ya que bastará con conocer el nombre del sitio web o usar una URL.

2) Uso de un sistema de gestión de citas automatizado:

- Método:

En esta propuesta de solución se propone crear un software que permita a los usuarios agendar citas médicas por medio de un programa de escritorio, el cual se lo realizara en el editor de código Visual Studio Code usando el lenguaje de programación C, en el programa el usuario podrá ingresar sus datos, fecha y hora además del doctor que desee ser atendido según la institución de salud que use el programa, es importante hacer énfasis en este tema ya que cada fundación puede modificar el horario de atención y los doctores disponibles, para hacer esto simplemente se debe acceder al código y modificar los datos mencionados anteriormente, además el programa contara con una interfaz sencilla para que cualquier persona se le haga un trabajo simple poder agendar sus citas según sus necesidades.

En la implementación de esta propuesta se usarán las estructuras de programación secuencial, condicional, bucles, estructuras de control de flujo, es importante el uso de estas estructuras de control para que el código sea claro y comprensible para cualquier persona que este encargado del mantenimiento o actualización del programa. Además, para poder tener el programa se lo podrá conseguir de distintas formas, ya sea por dispositivos USB, discos duros o unidades de estado sólido (SSD), esto debido a que es un programa que ocupará poca memoria, y no presentará ningún problema en su instalación.

- Herramientas tecnológicas:

Se necesitará un editor de código el cual será Visual Studio Code versión 17.6, en el cual se escribirá el código en el lenguaje C, además se necesitará una computadora de buenas características para poder codificar y probar la funcionalidad del código sin ningún tipo de problemas, también podríamos usar algoritmos estructurados para realizar el código de forma estructurada y legible.

- Restricciones técnicas:

En esta propuesta de solución no existen restricciones técnicas ya que se usará lenguaje y editor de código que nosotros ya conocemos por lo cual implementar esta solución podría ser lo mas óptimo para solventar el problema planteado acerca de la agendación de citas en la fundación Salud para Todos.

Esta propuesta de solución tiene una clara ventaja a comparación de las demás, ya que todo lo que se usará para implementarla, es de nuestro conocimiento como estudiantes, y todo lo que se usará en la codificación y desarrollo son cuestiones que ya conocemos, sin embargo una desventaja que puede tener esta propuesta es que puede ser limitada en algunos aspectos, y que si se la desea modificarla para que sea mucho mas amplia, se necesitará algo de tiempo para agregarle mas funciones y datos adicionales que la fundación Salud para todos desee.

3) Mejora del proceso manual de agendamiento de citas:

- Método:

Se desea crear un software de agendación de citas en el cual se optimice los procedimientos manuales mediante la implementación de formularios electrónicos, plantillas y sistemas de seguimiento de citas, en el cual se usará el editor de código Visual Studio Code y se programa en lenguaje C, además se necesitará adquirir información sobre cómo funcionan las bases de datos y como implementarlas para evitar errores en la codificación o implementación del programa.

Para la implementación de esta propuesta se usarán distintos tipos de estructuras de programación como, por ejemplo: Secuenciales, condicionales, estructuras de control de flujo, entre otras. Estas estructuras de control son las más comunes en un programa, ya que ayudan a conseguir un código mejor estructurado y legible. Los usuarios que deseen usar este programa podrán hacerlo mediante la nube, ya que se podría subir el programa a Dropbox o Google Drive, y así seria una manera sencilla y practica para que cualquier persona pueda acceder a la aplicación.

- Herramientas tecnológicas:

Se necesitará Visual Studio Code como editor de código, además necesitaremos plantillas de formularios electrónicos, hojas de cálculo para el seguimiento de citas, y base de datos para almacenar toda la información que el usuario ingrese por teclado.

- Restricciones técnicas:

Las restricciones técnicas de esta propuesta de solución es el tiempo y la magnitud del código que se desea implementare, ya que se usara base de datos, hojas de cálculo, plantillas de formularios, etc.

La documentación será extensa, y no se conoce a profundidad como es el uso de base de datos en un programa, además podría ser un poco más complicado la comprensión del funcionamiento del programa para algunas personas.

Existe una clara ventaja en esta propuesta de solución y es que será un programa muy completo, ya que se usará base de datos para poder almacenar datos en la nube, sin embargo al mismo tiempo esto es una desventaja ya que esto haría que el programa sea un poco más complicado de entender, y para desarrollarlo perfectamente se necesitara mucho mas tiempo, además algunos conceptos que se usarán aun no son de nuestro conocimiento, por lo tanto deberemos investigar y aprender sobre algunos temas que necesitemos, lo cual no será recomendable ya que no se tendrá experiencia.

Método seleccionado:

La metodología más efectiva para un deficiente agendamiento de citas es la propuesta numero dos ya que da a conocer información clara y concisa para la solución de cualquier tipo de problema. Es importante abarcar varios temas importantes como la libertad que tiene los pacientes ya que, tienen la suficiente capacidad para que puedan agendar una cita cuando ellos quieran por medio de una aplicación de escritorio, este programa podrá correr sin ningún problema en versiones de Windows 10 versión 1909 y versiones posteriores. (Build, 2023)

Al escoger esta propuesta de solución los usuarios no tendrán problemas en el entendimiento del programa ya que se presentará una interfaz sencilla con las especificaciones necesarias, Además los pacientes podrán usar el programa a cualquier hora del día, algo que nos ofrece el programa es la lista de horarios y doctores que están disponibles, y de esta forma se evitará citas duplicadas para personas distintas.

En muchas ocasiones las personas no cuentan con el tiempo necesario para poder ir a una institución de salud a agendar su cita, debido a este inconveniente, se elabora el programa computacional para asegurar la eficiencia y ahorro del tiempo por parte de los pacientes como de los trabajadores porque ya no habría la necesidad para agendar una cita por vía telefónica ni visitas presenciales. Tienen la facilidad de acceder al programa a cualquier hora del día para poder realizar la reservación de la cita al horario que más se acomode el paciente. La implementación de la tecnología ha ido mejorando mucho para el agendamiento de citas, lo más relevante es la utilización del lenguaje en C para la realización del sistema en línea. Cuando haces un programa en C se asegura el almacenamiento de la información completa de las fichas medicas para tener una mayor organización de todas las citas médicas

Definición de trabajo y metodología a utilizar:

- **DEFINIR OBJETIVOS Y ALCANCES:**

Buscamos establecer de mejor manera y con una mejor planificación la mejora de agendamientos de citas, determinamos el alcance de la solución siempre y cuando consideremos que está a nuestras manos poder desarrollarlo.

- **IDENTIFICAR ACTIVIDADES Y RESPONSABLES:**

En esta sección identificaremos las actividades necesarias para diseñar, desarrollar e implementar la solución, escogemos y asignamos responsables para cada actividad distribuyéndonos de mejor manera el proyecto designado.

- **ESTABLECER TIEMPOS DE EJECUCION:**

Definimos plazos para cada actividad, estableciendo así mismo fechas de inicio y de finalización, en esta parte empezamos a considerar la disponibilidad de tiempo y recursos.

- **DISEÑO Y DESARROLLO DE LA SOLUCION:**

En este caso optamos por un sistema de citas en línea, implementamos el diseño y desarrollo de la plataforma de escritorio, elegimos implementar un sistema automatizado, desarrollaremos el software y lo implementaremos en el gestor de citas.

- **PRUEBAS Y AJUSTES:**

Realizamos pruebas buscando el mejor funcionamiento del software y a su vez detectar posibles errores y mejoras, realizar ajustes y mejoras necesarias a la base de datos

- **IMPLEMENTACION Y SEGUIMIENTO:**

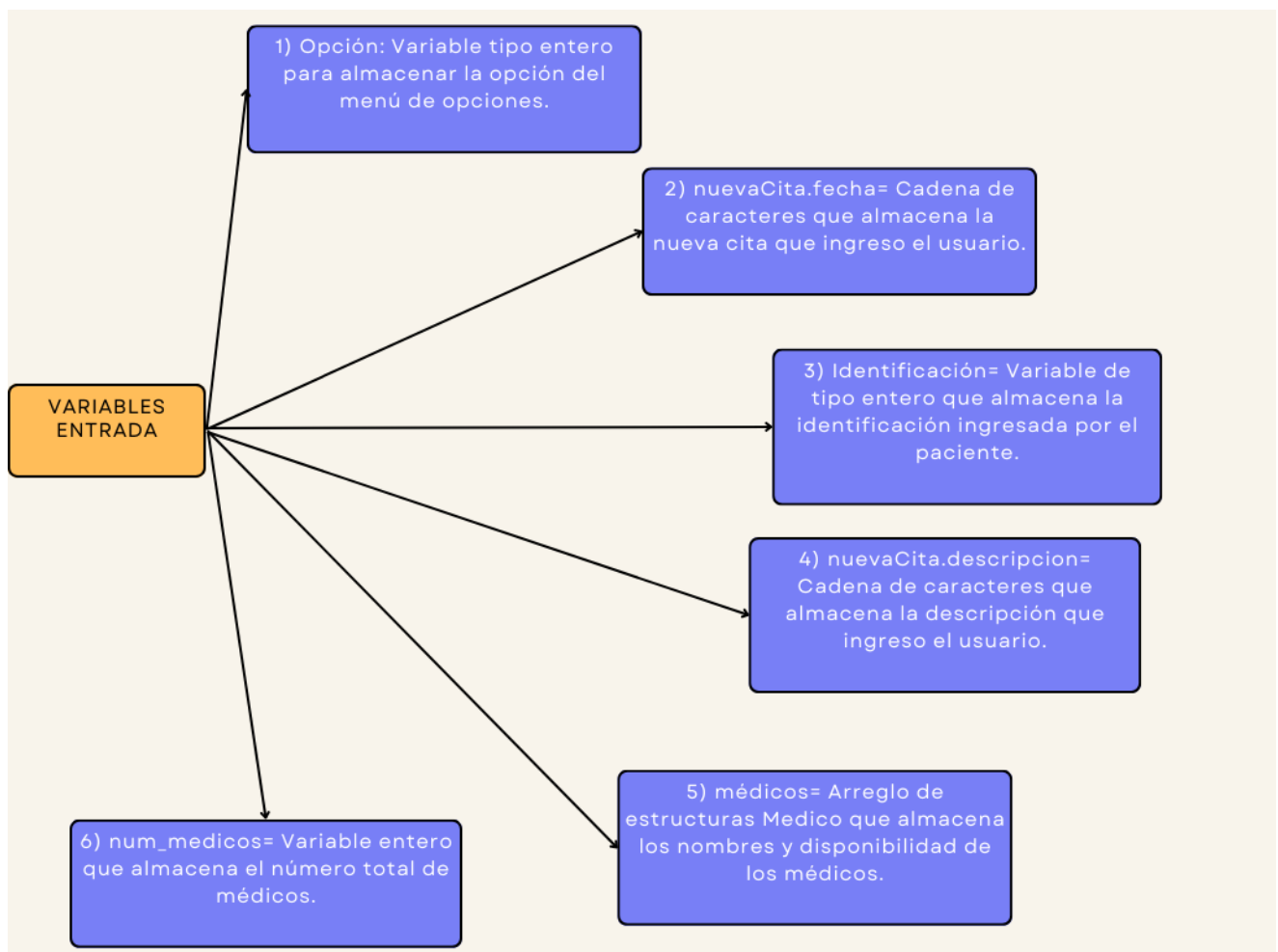
Implementar el software en los diferentes sitios médicos y ver el funcionamiento para garantizar un software de calidad, garantizando que este implementado con las mejoras necesarias para no tener problemas en algún agendamiento médico.

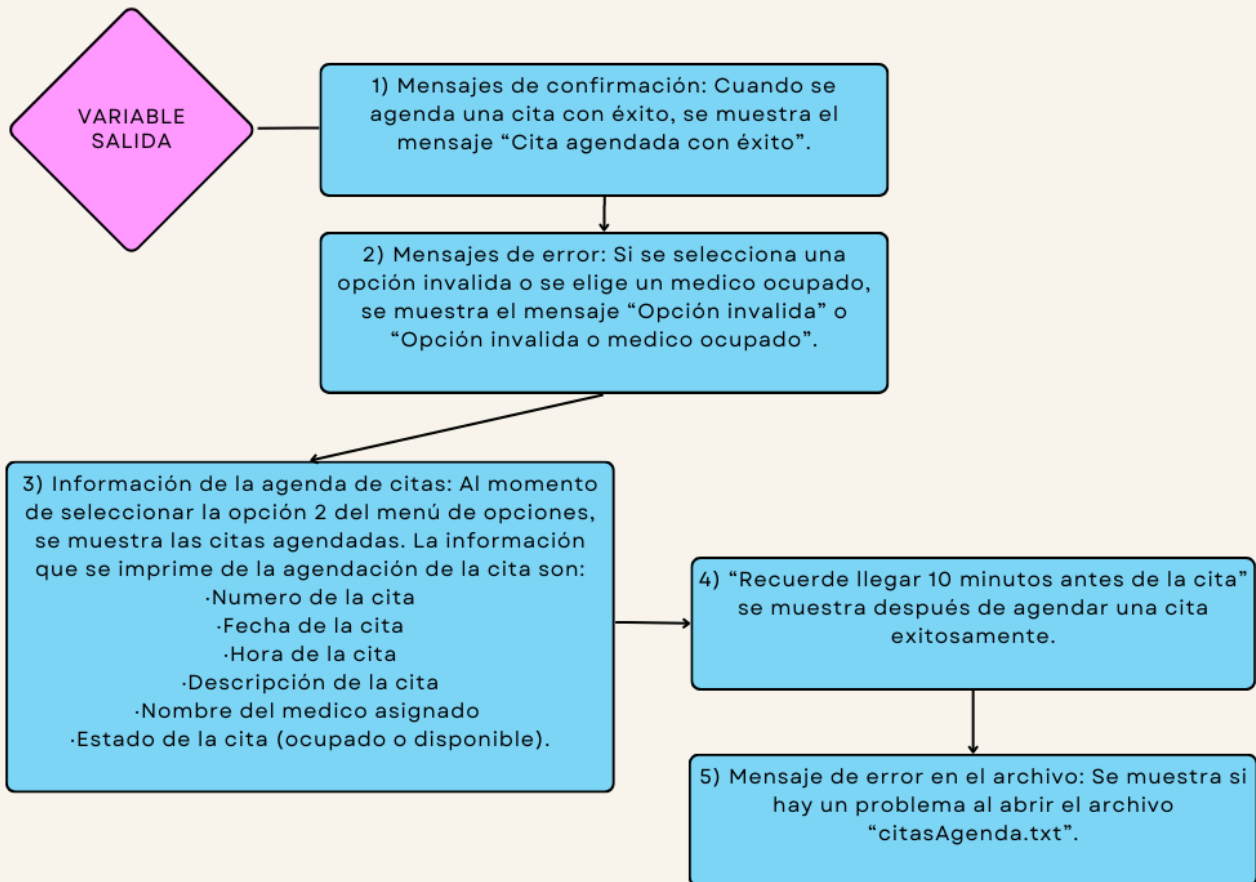
- **EVALUACION Y MEJORA:**

Evaluar constantemente el funcionamiento y efectividad del software implementado y así mismo seguir mejorando cada que se requiera; verificar siempre la base de datos buscando optimizar y garantizar el funcionamiento de mejor manera.

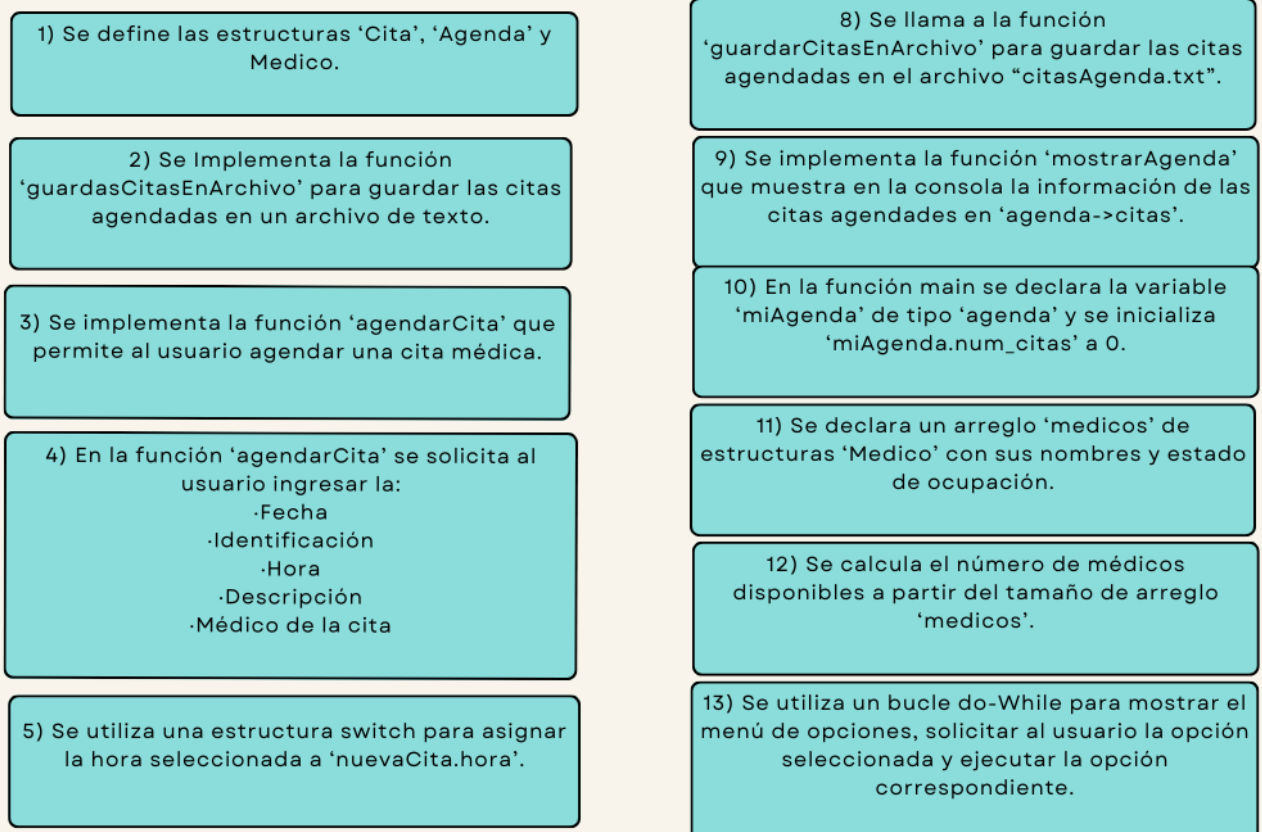
| Actividad | Responsable | Tiempo de ejecución |
|--|----------------|---------------------|
| Definir objetivos y alcance | Josue Valencia | 1 día |
| Identificar actividades y responsables | Eric Mullo | 1 día |
| Establecer tiempos de ejecución | Jordy Aguilar | 2 días |
| Diseño y desarrollo de la solución | Josue Valencia | 3 días |
| Pruebas y ajustes | Jordy Aguilar | 1 día |
| Implementación y seguimiento | Josue Valencia | 1 mes |
| Evaluación y mejora | Jordy Aguilar | Cada mes |

Diseño de la propuesta de solución:





PROCESO ASOCIADO



6) Se muestra al usuario la lista de médicos disponibles y se valida la opción seleccionada.

7) Si la opción y el médico son válidos, se asignan los datos de la cita, se actualiza el estado del médico, se guarda la cita en 'agenda -> citas' y se muestra un mensaje de confirmación.

14) El bucle se repite hasta que el usuario digite la opción salir.

Ejecución de escenarios de prueba:

El programa pedirá al usuario los siguientes datos para poder ejecutar el programa de forma correcta: nombre del paciente, fecha, hora y el doctor específico al cual desee reservar una cita; al momento de ejecutar el programa, se desplegará un menú con distintas opciones, entre ellas:

- 1) Agendar cita.
- 2) Revisar citas.
- 3) Salir.

Al momento de la ejecución del programa y obtención de datos, se pueden presentar diferentes tipos de casos especiales:

- En algún caso si el usuario ingresa una opción que no conste con las opciones presentadas por el programa, se le presentara un mensaje por pantalla con el texto "Usted ingreso una opción no valida, por favor vuelva a intentarlo".
- Si al momento que el programa le pida al usuario ingresar la hora para agendar la cita y se ingresa un número mayor a 24, se presentara un mensaje por pantalla que diga "La hora ingresada no es correcta, por favor intentar más tarde".
- Si al momento que el programa le pida al usuario ingresar la hora para agendar la cita y se ingresa un numero negativo, se mostrara por pantalla un mensaje con el texto "La hora ingresada no es correcta, por favor intentar más tarde".
- Si al momento que el programa le pida al usuario ingresar la fecha para agendar la cita y se ingresa una fecha mayor a 31, se presentara un mensaje por pantalla que diga "La fecha ingresada no es correcta, por favor intentar más tarde".

- Si al momento que el programa le pida al usuario ingresar la fecha para agendar la cita y se ingresa una fecha anticipada con un signo menos, se presentara un mensaje por pantalla que diga “La fecha ingresada no es correcta, por favor intentar más tarde”.
- Si al momento de obtención de datos el usuario ingresa un carácter especial que el programa no se lo pida, el programa se congelará y el usuario deberá intentarlo de nuevo.
- Si al momento que el programa le pida al usuario digitar su nombre por teclado, y no se digita ningún carácter y se envía la respuesta, el programa se congelará, por lo tanto, el usuario deberá iniciar nuevamente el programa.

Conclusiones:

Con el desarrollo de este programa de escritorio se mejorará la eficacia al momento de agendar citas, ya que se lo realizara de forma mucho más efectiva y rápida, debido a que para agendar cualquier tipo de cita, bastará con ingresar el nombre del paciente, número de cedula, nombre del doctor, hora y fecha, una vez obtenido estos datos el programa evaluará todos los datos ingresados desde el teclado y los guardará para luego mostrarlos por pantalla, además permitirá visualizar al usuario las citas ya agendadas de una manera fácil y sencilla, simplemente deberá ingresar la opción que diga “Consultar citas agendadas”, esta opción se podrá visualizar en el menú de opciones que se desplegará al ejecutar el programa, y de esta forma se podrá conocer con exactitud cualquier dato que así se desee.

Este sistema ayudará a reducir el tiempo de espera de los pacientes, ya que, si se realiza el agendado de citas de la manera tradicional, a papel y lápiz, la persona encargada de brindar atención a los clientes se tomará su tiempo buscando el nombre del paciente, además en algunos casos este tipo de datos se llegan a perder, ya que existen gran cantidad de personas esperando poder agendar o consultar sus citas médicas en la fundación Salud para Todos.

Según una investigación y estadísticas del IESS manifiesta que una persona debe esperar de 6 a 10 días laborables para poder asistir a una cita con su médico y que en diferentes temporadas del año estos números van en aumento, hasta podría llegar a 20 días laborables. (IESS, 2022)

Sin embargo, estos datos estarán en constante cambio ya que dependerá mucho de la ciudad en la que se encuentra y si la agendación de citas es en una institución de salud pública o privada.

Como se menciono anteriormente los sistemas de salud llegan a estallar en diferentes épocas del año o como fue el caso de la pandemia que asecho a todo el mundo, el coronavirus, donde todos los hospitales del mundo llegaron a estallar por la cantidad de pacientes. (Culebras, 2020)

Además, es importante recordar que los sistemas llegan a colapsar en algunos periodos de tiempo ya que no se cuenta con suficiente personal médico, por esta razón implementar este programa ayudaría a mejorar en una parte la organización y seguridad de datos de los pacientes ecuatorianos.

Finalmente, la aplicación permitirá al personal y especialistas médicos obtener datos precisos al instante, y de esta forma brindar una mejor atención a sus pacientes. Al implementar este programa se cumpliría con los objetivos previstos que ayudarían a solventar los problemas de agendación de citas en la fundación Salud para Todos

Recomendaciones:

Se recomienda usar el programa de escritorio de forma consiente y responsable, para evitar problemas con la codificación de esta, además es recomendable hacer uso constante de la misma, para mantener los datos actualizados y de forma mucho más segura.

Se recomienda un buen acceso a internet de parte de la institución pública que ejecutará el programa, para evitar cualquier tipo de congelamiento o demora al momento que el programa presente los resultados por pantalla. Se recomienda contratar un personal capacitado para que pueda hacer el respectivo mantenimiento de la aplicación de escritorio para la corrección de errores, mejoras de las capacidades, eliminación de funciones obsoletas y optimización. (B, 2015)

Además, es recomendable hacer modificaciones en un futuro, las cuales permitan al programa de escritorio ofrecer mayor número de soluciones a problemas que se puedan presentar en un lapso; además recordar que es importante que cualquier tipo de mantenimiento o modificación al programa, se debe realizar la respectiva documentación para que cualquier otro personal que trabaje dentro del programa pueda entender con claridad lo que realiza cada sección. (Kevin Carvajal, 2020)

Funcionamiento del código:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  #define MAX_CITAS 100
6  #define MAX_MEDICOS 10
7
8  struct Cita {
9      char fecha[20];
10     char hora[10];
11     char descripcion[100];
12     char medico[50];
13     int identificacion;
14     int ocupada;
15 };
16
17 struct Agenda {
18     struct Cita citas[MAX_CITAS];
19     int num_citas;
20 };
21
22 struct Medico {
23     char nombre[50];
24     int ocupado;
25 };
26
```

Se empieza incluyendo las librerías que se usarán a lo largo de la codificación del programa, las cuales serán: `stdio.h`, `stdlib.h` y `string.h`; luego se define macros para el número máximo de citas que en este caso será de 100 y también el número máximo de medico que será de 10. Esto se hace ya que se desea que estos valores sean globales dentro del programa y no se puedan modificar. Luego lo que se hace es crear tres structs llamadas: Cita, Agenda, Medico; esto se hace para agrupar diferentes tipos de datos relacionados bajo un solo nombre y de esta forma mantener todos los datos de manera organizada y ordenada.

```
27 void guardarCitasEnArchivo(struct Agenda* agenda) {
28     FILE* archivo = fopen("citasAgenda.txt", "w");
29     if (archivo == NULL) {
30         printf("Error al abrir el archivo.\n");
31         return;
32     }
33
34     for (int i = 0; i < agenda->num_citas; i++) {
35         fprintf(archivo, "Cita %d:\n", i + 1);
36         fprintf(archivo, "Fecha: %s\n", agenda->citas[i].fecha);
37         fprintf(archivo, "Hora: %s\n", agenda->citas[i].hora);
38         fprintf(archivo, "Nombre del paciente: %s\n", agenda->citas[i].descripcion);
39         fprintf(archivo, "Medico: %s\n", agenda->citas[i].medico);
40         fprintf(archivo, "Estado: %s\n", agenda->citas[i].ocupada ? "Ocupada" : "Disponible");
41         fprintf(archivo, "-----\n");
42     }
43
44     fclose(archivo);
45
46 }
47
```

Luego se procede a crear una función de tipo void llamada “guardarCitasEnArchivos”, en la que tomara como argumento un puntero a una struct agenda; luego se crea un archivo file para poder ir almacenando todo lo que ingrese el usuario en el archivo llamado “citasAgenda.txt”. Para abrir el archivo txt se usa la función `fopen`, seguido del nombre que nosotros deseamos y entre comillas la letra (w) ya que se desea abrir el archivo txt en modo de escritura. Si el archivo es igual a NULL, quiere decir que ocurrió un error al abrir el archivo txt, por lo tanto, se presentara por pantalla un mensaje que diga “Error al abrir el archivo”.

Luego creo un bucle for para que se vaya guardando toda la información que el usuario ingrese, el bucle se repetirá hasta que agenda sea menor que el numero de citas; se usa la función `fprintf` ya que se desea escribir dentro del archivo que creamos anteriormente llamado “citasAgenda.txt”. En la línea 41 encontramos el signo (?) seguido de “Ocupada” : “Disponible”, esto es un operador ternario en C, el cual consiste en que si (`agenda -> citas[i].ocupada`) es verdadero o en otras palabras es distinto de cero, se selecciona la cadena “Ocupada”, en cambio si es falso o en otras palabras es cero, se seleccionará la cadena “Disponible”.

Y al final simplemente uso la función `fclose` para cerrar el archivo.

```
47
48 void agendarCita(struct Agenda* agenda, struct Medico* medicos, int num_medicos) {
49     int identificacion;
50     if (agenda->num_citas < MAX_CITAS) {
51         struct Cita nuevaCita;
52         printf("Fecha: ");
53         scanf("%s", nuevaCita.fecha);
54
55         printf("Ingrese su numero de cedula: ");
56         scanf("%d", &identificacion);
57
58         printf("Horario disponible para la agendacion de citas:\n");
59         printf("1) 9:00\n");
60         printf("2) 10:00\n");
61         printf("3) 11:00\n");
62         printf("4) 15:00\n");
63         printf("5) 19:00\n");
64         printf("6) 21:00\n");
65         printf("7) 16:00\n");
66         printf("8) 17:00\n");
67         printf("9) 23:00\n");
68         // Agrega más opciones de hora según sea necesario
69     }
```

En esta parte vuelvo a crear otra función de tipo void llamada “agendarCita”, en la cual tomara distintos argumentos; dentro de esta función creo un if que se ejecute siempre y cuando numero de citas sea menor que el número máximo de citas; luego lo que se hace es pedir al usuario que ingrese la fecha, numero de cedula, y luego le muestro por pantalla el horario de atención disponible para que el usuario pueda ingresar el que mas accesible se le haga.

```
70     int opcion;
71     printf("Seleccione el numero de la hora: ");
72     scanf("%d", &opcion);
73     switch (opcion) {
74         case 1:
75             strcpy(nuevaCita.hora, "9:00");
76             break;
77         case 2:
78             strcpy(nuevaCita.hora, "10:00");
79             break;
80         case 3:
81             strcpy(nuevaCita.hora, "11:00");
82             break;
83         case 4:
84             strcpy(nuevaCita.hora, "15:00");
85             break;
86         case 5:
87             strcpy(nuevaCita.hora, "19:00");
88             break;
89         case 6:
90             strcpy(nuevaCita.hora, "21:00");
91             break;
92         case 7:
93             strcpy(nuevaCita.hora, "16:00");
94             break;
95         case 8:
96             strcpy(nuevaCita.hora, "17:00");
97             break;
98         case 9:
99             strcpy(nuevaCita.hora, "23:00");
100            break;
101        default:
102            printf("Usted ingreso una hora no establecida!!\nPor favor vuelva a intentarlo!!\n");
103            return;
104    }
```

Una vez que se muestra el horario de atención, el usuario ingresará su opción, luego según la opción del usuario se reemplazara la hora elegida en la variable (nuevaCita.hora) esto se hace con la ayuda de la función strcpy; de esta forma ya tendremos la hora que el usuario eligió guardada en la variable (nuevaCita.hora).

```
105 |
106 |     printf("Ingrese su nombre: ");
107 |     scanf("%s", nuevaCita.descripcion);
108 |
109 |     printf("Medicos disponibles:\n");
110 |     for (int i = 0; i < num_medicos; i++) {
111 |         if (medicos[i].ocupado == 0) {
112 |             printf("%d %s\n", i + 1, medicos[i].nombre);
113 |         }
114 |     }
115 |     printf("Seleccione el numero del medico: ");
116 |     scanf("%d", &opcion);
117 |     if (opcion >= 1 && opcion <= num_medicos && medicos[opcion - 1].ocupado == 0) {
118 |         strcpy(nuevaCita.medico, medicos[opcion - 1].nombre);
119 |         nuevaCita.ocupada = 1;
120 |         medicos[opcion - 1].ocupado = 1;
121 |
122 |         agenda->citas[agenda->num_citas] = nuevaCita;
123 |         agenda->num_citas++;
124 |
125 |         guardarCitasEnArchivo(agenda);
126 |
127 |         printf("\n!!Cita agendada con exito!!\n");
128 |         printf("IMPORTANTE: \nRecuerde llegar 10 minutos antes a su cita.");
129 |     } else {
130 |         printf("Opcion invalida o medico ocupado.\n");
131 |     }
132 | } else {
133 |     printf("La agenda esta llena!! \nNo es posible agendar mas citas.\n");
134 | }
135 | }
136 |
```

En esta parte seguiremos pidiendo al usuario que ingrese el resto de los datos como lo es el nombre y luego el medico con el que desee reservar la consulta; en la línea 107 se usa "%s", esto es un especificador de formato para leer una cadena de caracteres hasta encontrar el carácter de nueva línea (\n).

Después de eso se muestran la lista de médicos disponibles, el cual pasa por un bucle for el cual cumple la función de evaluar y conocer si un medico esta disponible o no, para luego mostrarlos por pantalla.

Luego se le solicita al usuario ingresar el numero del medico que desee, la opción elegida por el usuario se la evaluara en un if, la cual debe ser >= 1, <=numero de médicos y si ocupado es igual a cero; si se cumple con todas estas condiciones se actualizaran los valores para mas tarde ser presentados por pantalla, y si no se cumple toda la condición se presentará un mensaje que diga "Opción invalida o medico ocupado". O en otro caso se mostrará el mensaje "La agenda está llena!! No es posible agendar más citas".


```
136
137 void mostrarAgenda(struct Agenda* agenda) {
138     printf("-----\n");
139     printf("Agenda de citas:\n");
140     for (int i = 0; i < agenda->num_citas; i++) {
141         printf("Cita %d:\n", i + 1);
142         printf("Fecha: %s\n", agenda->citas[i].fecha);
143         printf("Hora: %s\n", agenda->citas[i].hora);
144         printf("Nombre del paciente: %s\n", agenda->citas[i].descripcion);
145         printf("Medico: %s\n", agenda->citas[i].medico);
146         printf("Estado: %s\n", agenda->citas[i].ocupada ? "Ocupada" : "Disponible");
147         printf("-----\n");
148     }
149 }
150
```

Luego se crea otra función de tipo void llamada “mostrarAgenda”, esto se hace para tener mejor organizado el código y después en el main.c simplemente llamar a la función; entonces esta función lo único que hace es mostrar por pantalla todos los datos de la cita.

```
169     int opcion;
170     do {
171         printf("--BIENVENIDO A LA FUNDACION SALUD PARA TODOS--\n");
172         printf("\n");
173         printf("-----Menu-----\n");
174         printf("1) Agendar cita\n");
175         printf("2) Mostrar agenda\n");
176         printf("3) Salir\n");
177         printf("Ingrese su opcion: ");
178         scanf("%d", &opcion);
179
180         switch (opcion) {
181             case 1:
182                 agendarCita(&miAgenda, medicos, num_medicos);
183                 break;
184             case 2:
185                 mostrarAgenda(&miAgenda);
186                 break;
187             case 3:
188                 printf("Tenga un bonito dia.\n");
189                 break;
190             default:
191                 printf("Usted ingreso una opcion no valida!! Vuelva a intentarlo\n");
192                 break;
193         }
194
195         printf("\n");
196     } while (opcion != 3);
197
198     return 0;
199 }
```

En este apartado simplemente se muestra por pantalla las opciones que se puede realizar en el programa, para luego pedirle al usuario que desee cual quiere ejecutar, y dependiendo de su elección se llamará a las funciones que realicen cada acción que así se desee, se evaluará la opción ingresada por el usuario por medio de la función switch.

Ejecución del programa:

```
--BIENVENIDO A LA FUNDACION SALUD PARA TODOS--

-----Menu-----
1) Agendar cita
2) Mostrar agenda
3) Salir
Ingrese su opcion: █
```

El programa pide ingresar una opción, en este caso se repetirá este menú hasta que se ingrese la opción de salir (3).

```
-----Menu-----
1) Agendar cita
2) Mostrar agenda
3) Salir
Ingrese su opcion: 1
Fecha: 10/07/2023
Ingrese su numero de cedula: 0750251539
Horario disponible para la agendacion de citas:
1) 9:00
2) 10:00
3) 11:00
4) 15:00
5) 19:00
6) 21:00
7) 16:00
8) 17:00
9) 23:00
Seleccione el numero de la hora: 8
Ingrese su nombre: Jordy Aguilar
Medicos disponibles:
1) Dr. Perez
2) Dra. Gomez
3) Dr. Lopez
4) Dra. Rodriguez
5) Dr. Martinez
6) Dr. Fernandez
7) Dra. Sanchez
8) Dra. Sarmiento
9) Dr. Mullo
10) Dr. Valencia
Seleccione el numero del medico: 10

!!Cita agendada con exito!!
IMPORTANTE:
Recuerde llegar 10 minutos antes a su cita.
```

Si se elige la primera opción la cual es (Agendar Cita) el programa pedirá distintos datos al usuario, como se muestra en pantalla. Después de terminar de pedir los datos el programa volverá a mostrar el menú, en este caso ingresaremos la segunda opción (Mostrar Agenda).

```
--BIENVENIDO A LA FUNDACION SALUD PARA TODOS--

-----Menu-----
1) Agendar cita
2) Mostrar agenda
3) Salir
Ingrese su opcion: 2
-----

Agenda de citas:
Cita 1:
Fecha: 10/07/2023
Hora: 17:00
Nombre del paciente: Jordy Aguilar
Medico: Dr. Valencia
Estado: Ocupada
-----

--BIENVENIDO A LA FUNDACION SALUD PARA TODOS--

-----Menu-----
1) Agendar cita
2) Mostrar agenda
3) Salir
Ingrese su opcion: █
```

Y por último seleccionaremos la opción 3 que es (Salir). Además, es importante recordar de que todo se guardará en el archivo txt que creamos con el nombre de (citasAgenda.txt):

```
≡ citasAgenda.txt
1  Cita 1:
2  Fecha: 10/07/2023
3  Hora: 17:00
4  Nombre del paciente: Jordy Aguilar
5  Medico: Dr. Valencia
6  Estado: Ocupada
7  -----
8  █
```

Referencias

- Build, M. (2 de abril de 2023). *Esta es la era de la inteligencia artificial; construya el futuro más rápido con IA de Microsoft*. Obtenido de Esta es la era de la inteligencia artificial; construya el futuro más rápido con IA de Microsoft: <https://learn.microsoft.com/es-es/cpp/build/walkthrough-compile-a-c-program-on-the-command-line?view=msvc-170>
- Kevin Carvajal, C. S. (noviembre de 2020). *Desarrollo de un aplicacion web para el control de citas y manejo de historial medico en la unidad medica Family Care de la ciudad de Guayaquil*. Obtenido de Desarrollo de un aplicacion web para el control de citas y manejo de historial medico en la unidad medica Family Care de la ciudad de Guayaquil: <https://dspace.ups.edu.ec/bitstream/123456789/20905/1/UPS-GT003362.pdf>
- Trabajo, O. I. (07 de junio de 2021). *El sistema de salud ecuatoriano y la covid 19*. Obtenido de El sistema de salud ecuatoriano y la covid 19: https://www.ilo.org/wcmsp5/groups/public/---americas/---ro-lima/---sro-lima/documents/publication/wcms_799790.pdf
- B, L. (19 de marzo de 2015). *Software Maintenance Management*. Obtenido de Software Maintenance Management: <https://web.archive.org/web/20150319075401/http://www.compaid.com/caiinterne/t/ezone/capersjones-maintenance.pdf>
- Culebras, J. (7 de abril de 2020). En la pandemia de COVID-19 no hay camas para todos, ¿a quién tratamos? Obtenido de En la pandemia de COVID-19 no hay camas para todos, ¿a quién tratamos?: https://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S2529-850X2020000600589
- IESS. (junio de 2022). IESS. Obtenido de IESS: https://www.iless.gob.ec/noticias/-/asset_publisher/4DHq/content/iess-reduce-tiempo-de-asignacion-de-citas-medicas/10174?redirect=https%3A%2F%2Fwww.iless.gob.ec%2Fnoticias%3Fp_p_id%3D101_INSTANCE_4DHq%26p_p_lifecycle%3D0%26p_p_state%3Dnormal%26p_p_mode%3Dvi