

## REGISTRO POBLACIONAL CÁNCER CALI

### Actividades Realizadas

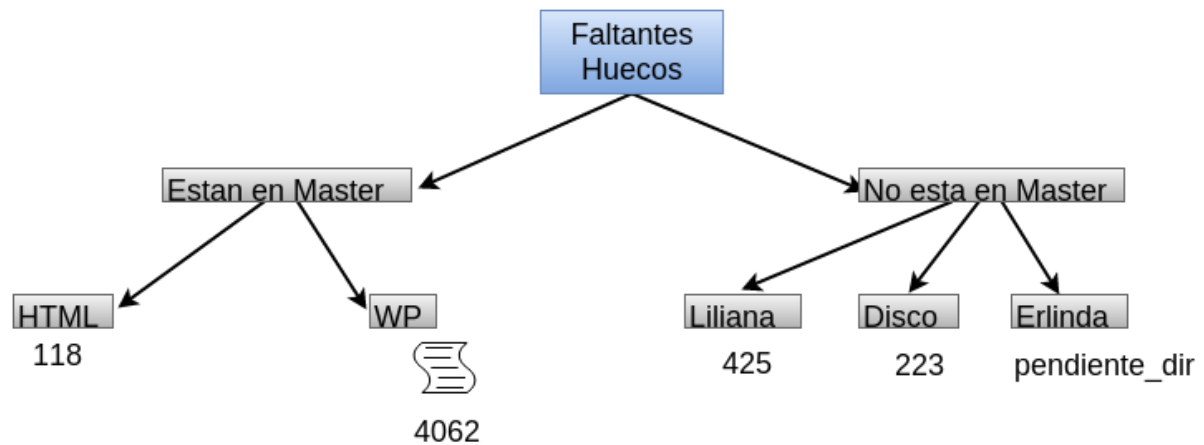
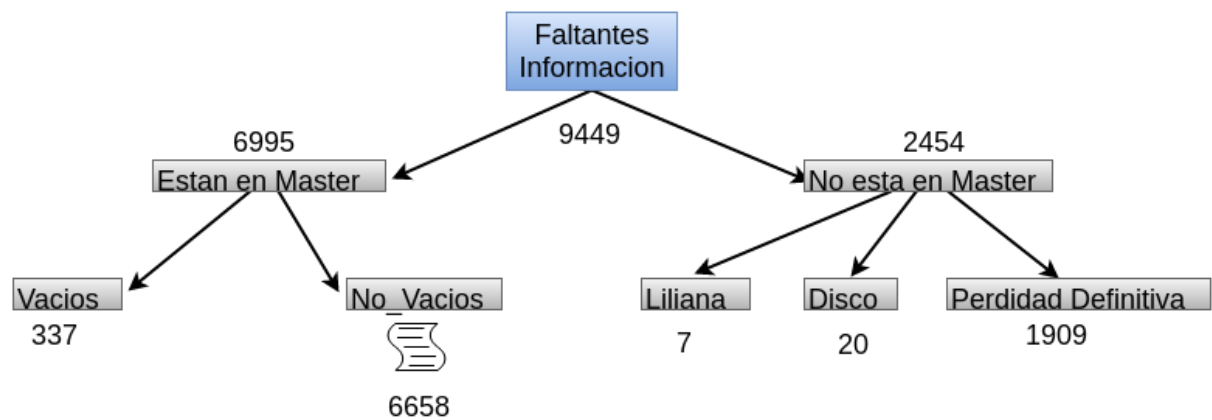
Nelson Portilla

Julio - Agosto

2016

1. Se revisó la herramienta RedCap y se solucionaron los problemas de configuración.
2. Se instaló y se configuró la conexión ssh con el servidor.
  - a. El modo de conectarse desde la terminal linux: `~$ ssh fox@192.168.216.21`
3. Se requiere instalar un servidor de correos, se instala Postfix.
  - a. Error al descargar paquetes: se debe borrar la carpeta `/var/lib/apt/lists/lock`
4. Se instala y configura POSTFIX:
  - a. `~$ sudo apt-get install postfix`
  - b. seleccionar sitio-internet
  - c. colocar un nombre al servidor de correo
  - d. se debe configurar Postfix con un servicio de correo tercero, como gmail de registro.cancercali
  - e. Se configura el main.cf
  - f. INFO:  
<https://miraspi.wordpress.com/2013/02/14/postfix-y-gmail-para-enviar-correo-del-sistema/>
5. Se prueba que se pueden enviar correos a cualquier cuenta desde el servidor.
6. Desde RedCap se crean usuarios y se envía automáticamente al correo de usuario el enlace para registrar la contraseña.
7. Se configura RedCap para que el acceso ya no sea público, sino por Login
8. Se crean SuperUsuarios a paolacollazos, lebravo y nelsonph88.
9. Se eliminan los privilegios del primer usuario por defecto. site admin
10. para problemas con el *Cron* se otorgan permisos de escritura.
11. para solucionar problemas con mycrypt, se instala un paquete de php5
12. Se descargan y actualiza lista de repositorios. ir a: [repogen.simplylinux.ch](http://repogen.simplylinux.ch)
13. Se requiere subir a la base de datos los archivos de informes-patología-html, específicamente las descripción macro micro y diagnóstico.
14. Debido a que no se encontró el script en Java que se había realizado anteriormente, se realiza un nuevo proceso en PYTHON.
15. **PatologiaInsertion:** scripts en Python para la extracción automática de información e inserción en la base de datos.
16. Enlace del repositorio: <https://github.com/nelson-portilla/PatologiaInsertion>
17. En el folder *PatologiaInsertion* se ubican los siguientes subdirectorios:
  - a. **scripts:**
    - i. **creaciontabla:** instrucciones en SQL para la creación de la tabla muestra\_html
    - ii. **registro.csv:** Datos que se extraen de cada folder y luego se cargan a la base de datos

- iii. **insertarFROMcsv.sql**: instrucción en sql para cargar datos del registro.csv a la tabla
  - iv. **extraccion.py**: Script en python para la extraccion de informacion
  - v. **insertar.py**: Script en python para la insercion de los datos anteriormente extraidos.
  - b. **informes-patologia-html**: Folder con todos los archivos en subcarpetas.
  - c. **informesEnTXT**: Folder con todos los archivos convertidos en TXT, en una sola línea.
18. Para las pruebas se descarga postgres en el PC-cliente linux
19. Se hace una copia de la base de datos: PatologiaHUV
20. Se hace una copia del esquema public, donde se encuentra la tabla muestra.
- a. Exportar:
  - b. `pg_dump -h 192.168.216.21 -d PatologiaHUV -U postgres -n public > file.sql`
  - c. Importar:
  - d. `psql -h localhost -d PatologiaHUV -f file.sql`
21. Se realiza una copia de los informes del servidor al PC-cliente:
- a. `rsync -avz -e 'ssh' fox@192.168.216.21:/var/www/informes-patologia-html /home/registro/Documentos/PatologiaInsertion/`
22. Primera Version: se copia de un archivo a la base de datos
23. Segunda Version: se copia varios archivos ubicados en subfolders
24. Prueba1: 1.935 archivos cargados: errores con algunos vacios.
25. prueba2: 2.010 archivos cargados: errores por caracteres
26. prueba3: 60.206 archivos cargados: errores con folder "otros"
27. prueba4: 84.312 archivos cargados: HC se debe cambiar a BigInt
28. prueba5: 183.537 archivos cargados en 1636 segundos (27 minutos)
29. RELEASE 1.0
30. Antes de hacer las pruebas en el servidor, se agrega una nueva columna "cedula"
- ETAPA DE ANÁLISIS: Información faltante**
31. Se obtiene una lista de los archivos en la base de datos que les falte la información de macro, micro y diagnóstico.
32. Se saca la lista de archivos en los siguientes discos:
- a. **Master**
  - b. **Liliana**
  - c. **Erlinda**
  - d. **Disco Dañado**
33. Se obtiene una lista de archivos que no se subieron a la base de datos (huecos)
34. Dos Contenedores:
- a. Info(archivos en BD que falta información)
  - b. Huecos (archivos no cargados a la bd)
- ETAPA DE BÚSQUEDA**
35. Se realiza una búsqueda por todos los discos para encontrar la ubicación de los archivos faltantes.
36. Se crean listas con los nombres de los archivos que se encuentren en cada disco.



Año	Cantidad
00	15
03	10
07	2
08	4
95	263
96	921
97	544
98	19
99	131

### ETAPA DE RECUPERACIÓN

37. Se copian al computador los archivos de todos los discos.
38. Con las listas, se sacan los archivos de cada disco.
39. Los archivos que están en WP se convierten a HTML

40. NOTA: Durante este proceso, se perdieron algunos archivos por estas razones:
- a. Archivos vacíos
  - b. Los nombres tenían algún carácter (- " \_ ; ' ".TXTT", "@", etc)
  - c. La conversión de wp a html no se realizó con éxito, debido al formato TXT

Tablas

41. Huecos

- a. Disco

<b>Año</b>	<b>Cantidad</b>
00	1
14	4
15	173
17	1
88	1
91	1
<i>TOTAL</i>	<i>181</i>

- b. Master HTML

<b>Año</b>	<b>Cantidad</b>
15	1
16	117
<i>TOTAL</i>	<i>118</i>

- c. Master WP

<b>Año</b>	<b>Cantidad</b>
16	146
<i>TOTAL</i>	<i>146</i>

- d. Miriam

<b>Año</b>	<b>Cantidad</b>
16	420
<i>TOTAL</i>	<i>420</i>

42. Info

- a. disco

<b>Año</b>	<b>Cantidad</b>
------------	-----------------

6	2
96	1
97	15
<i>TOTAL</i>	<i>18</i>

b. liliana

<b>Año</b>	<b>Cantidad</b>
0	1
3	1
<i>TOTAL</i>	<i>2</i>

c. master

<b>Año</b>	<b>Cantidad</b>
0	201
1	1403
2	564
3	240
4	216
5	166
6	240
7	206
8	144
9	176
10	92
11	49
12	121
13	451
14	80

15	511
16	341
98	428
99	1015
<i>TOTAL</i>	<i>6644</i>