**UNIVERSIDAD DE GUADALAJARA**

**CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS**

# “Actividad 2 - 2 Recuperacion”



Seguridad de la información

**Integrantes:**

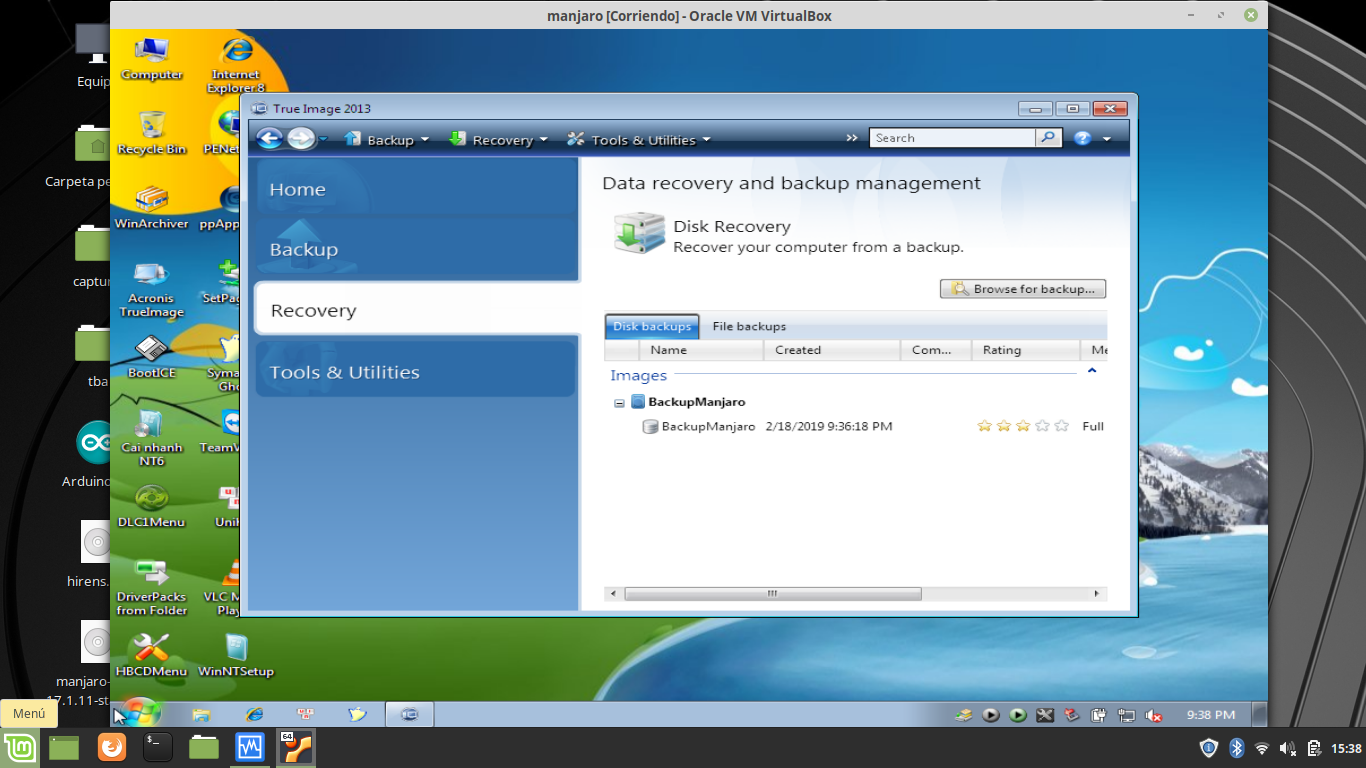
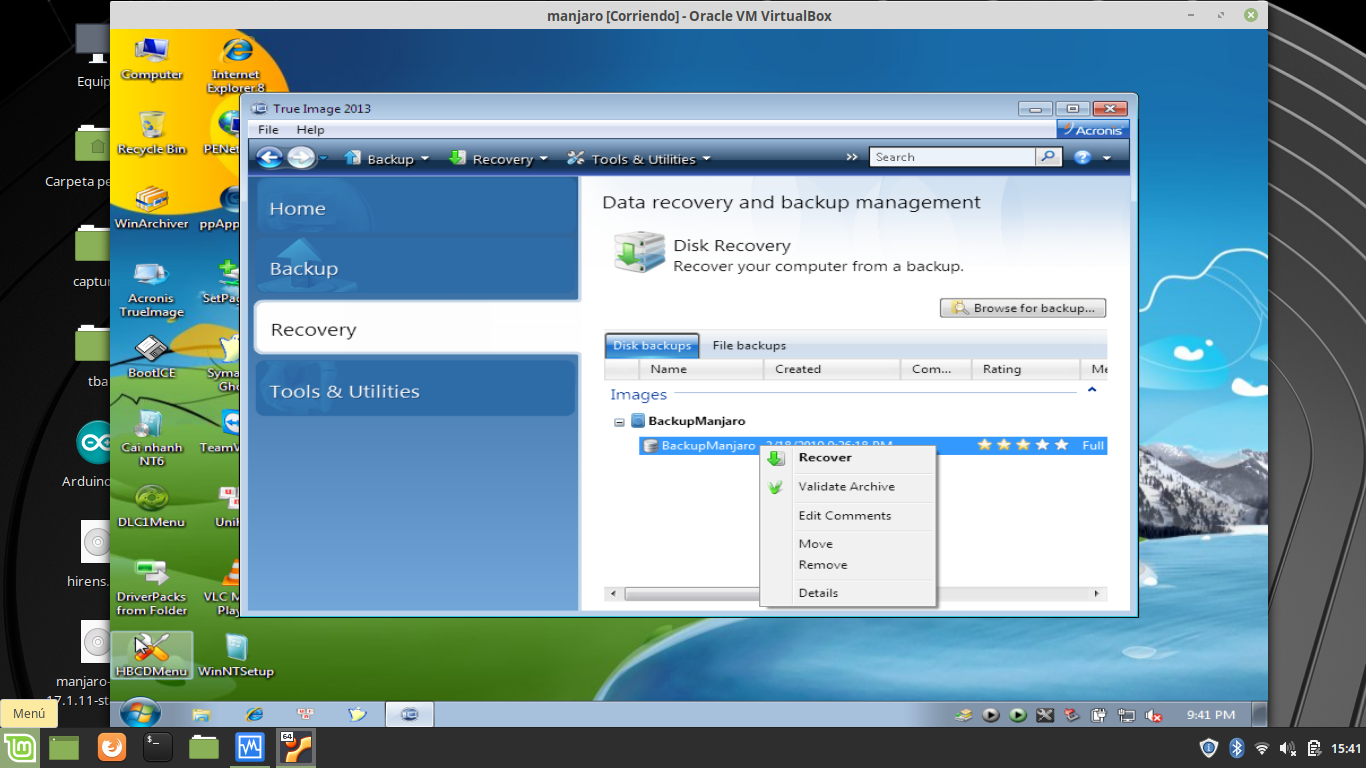
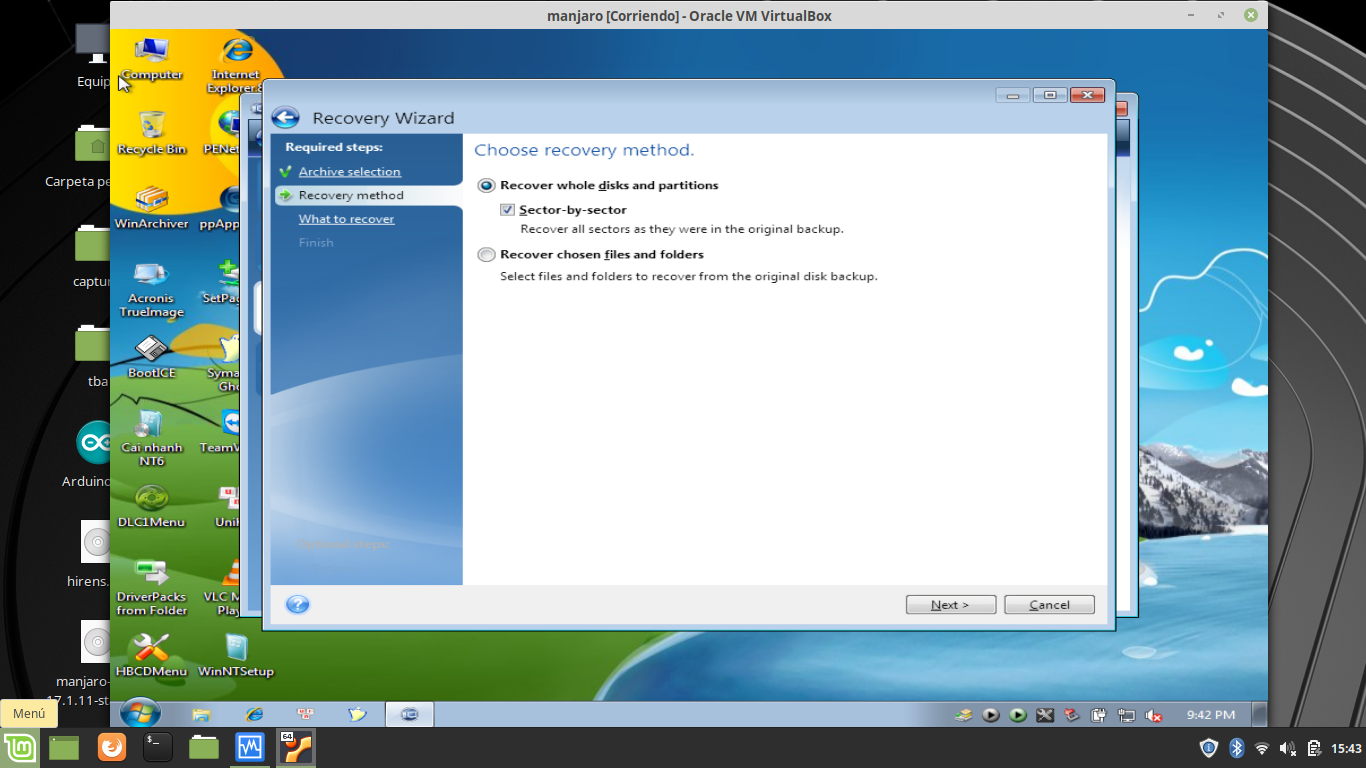
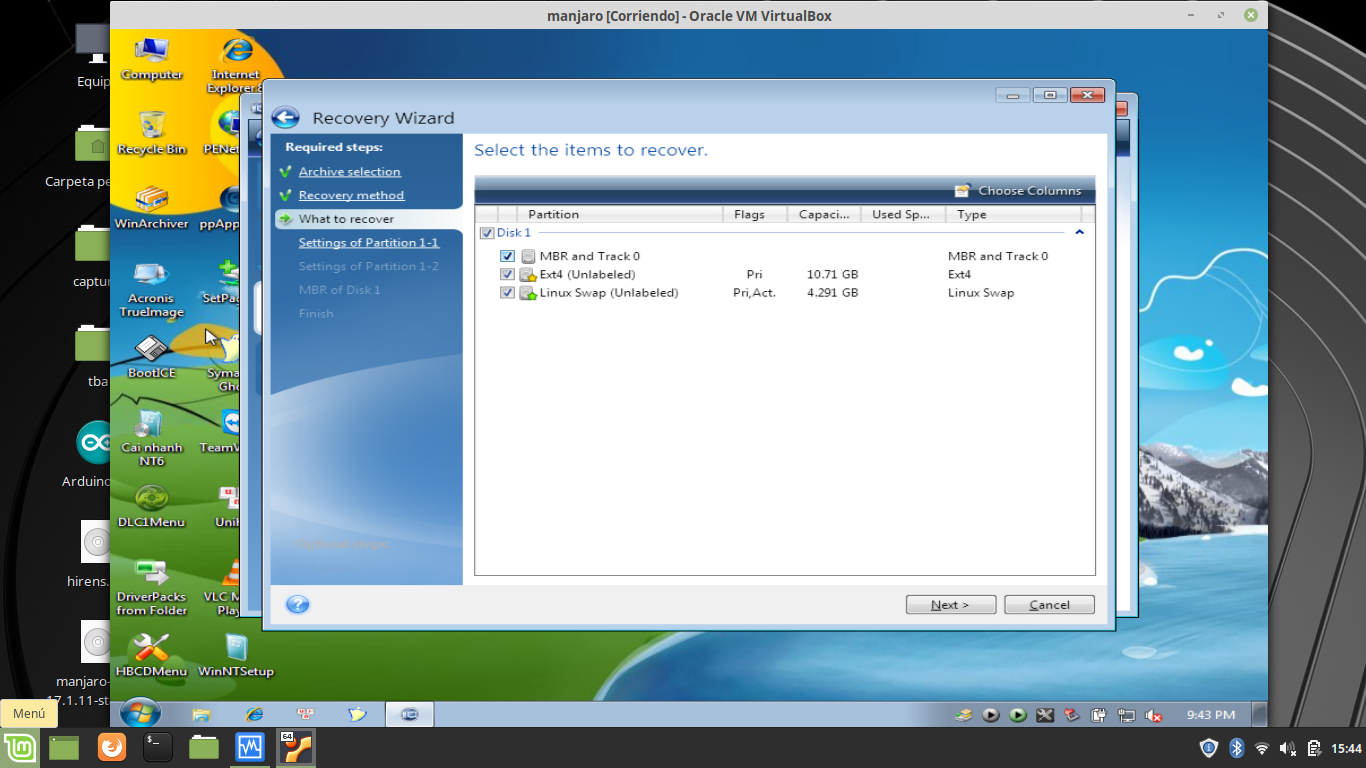
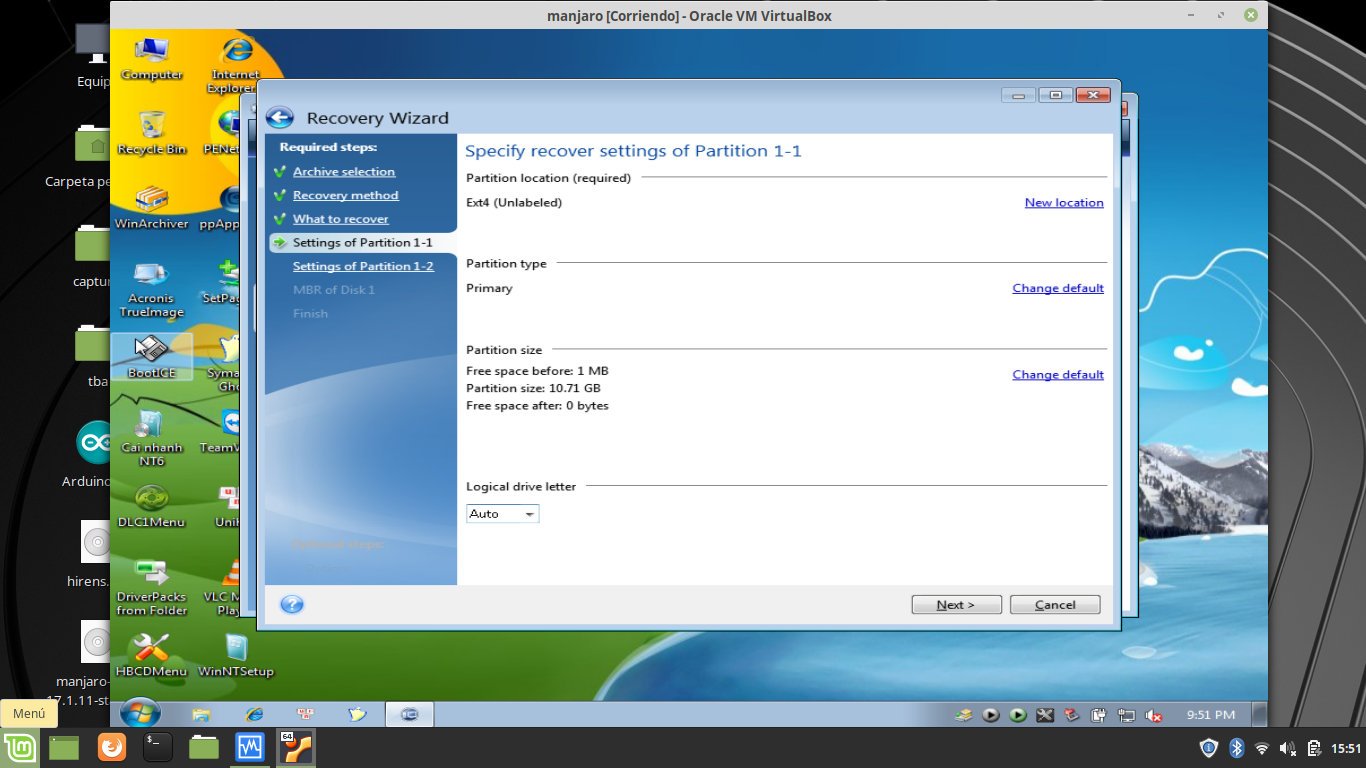
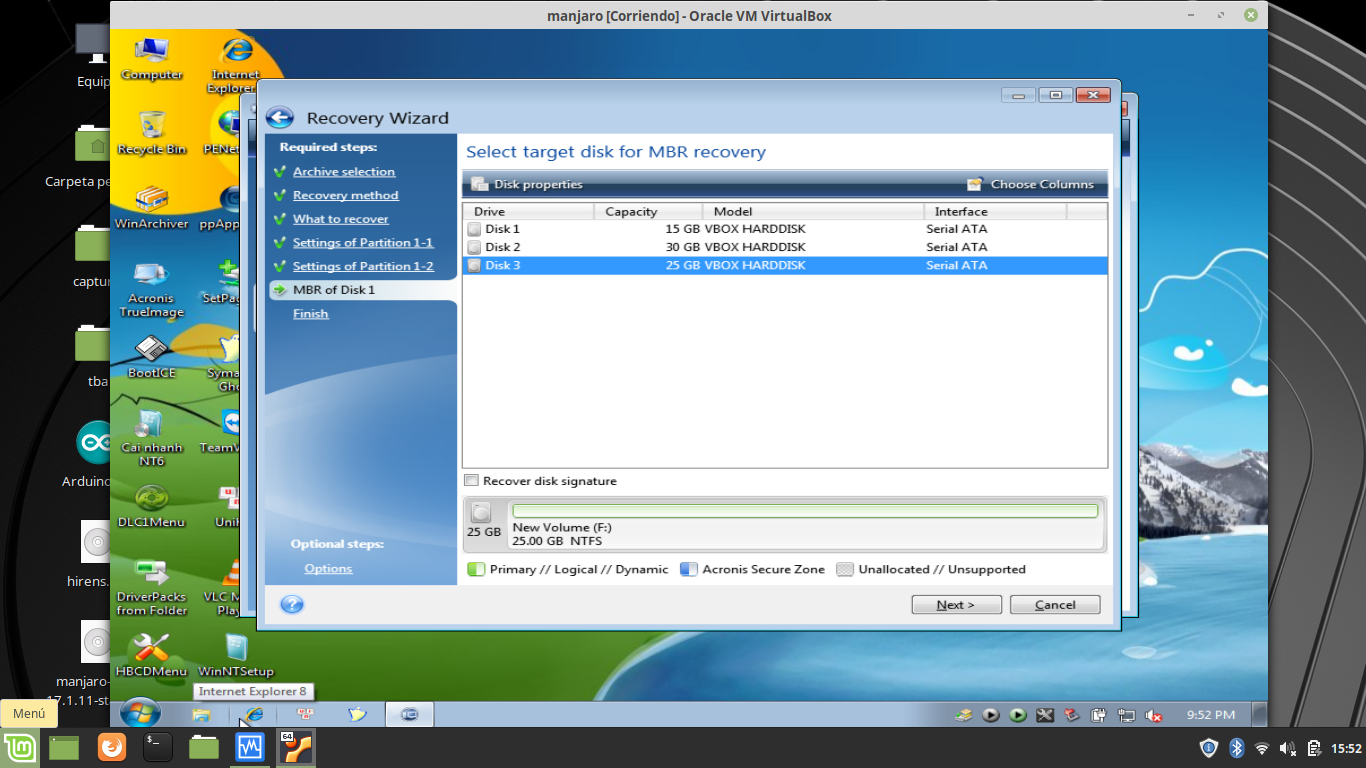
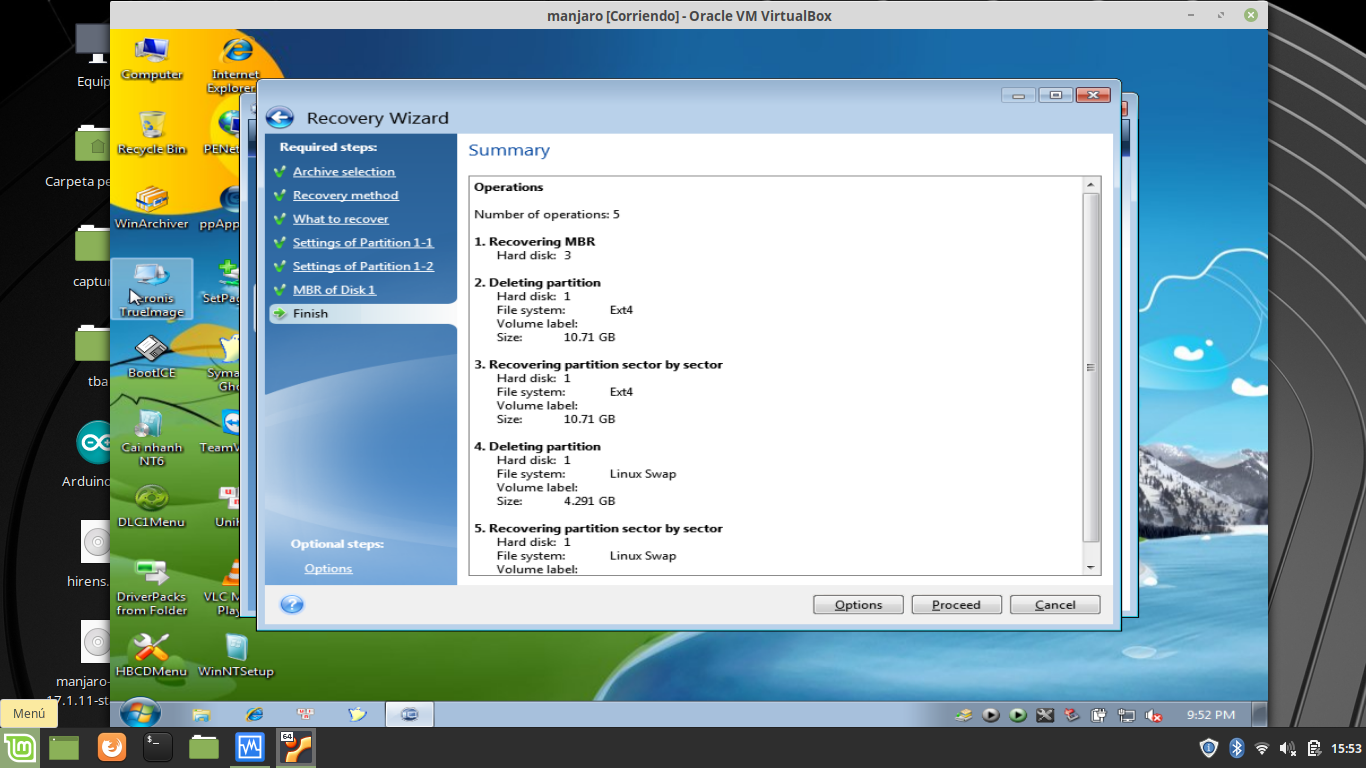
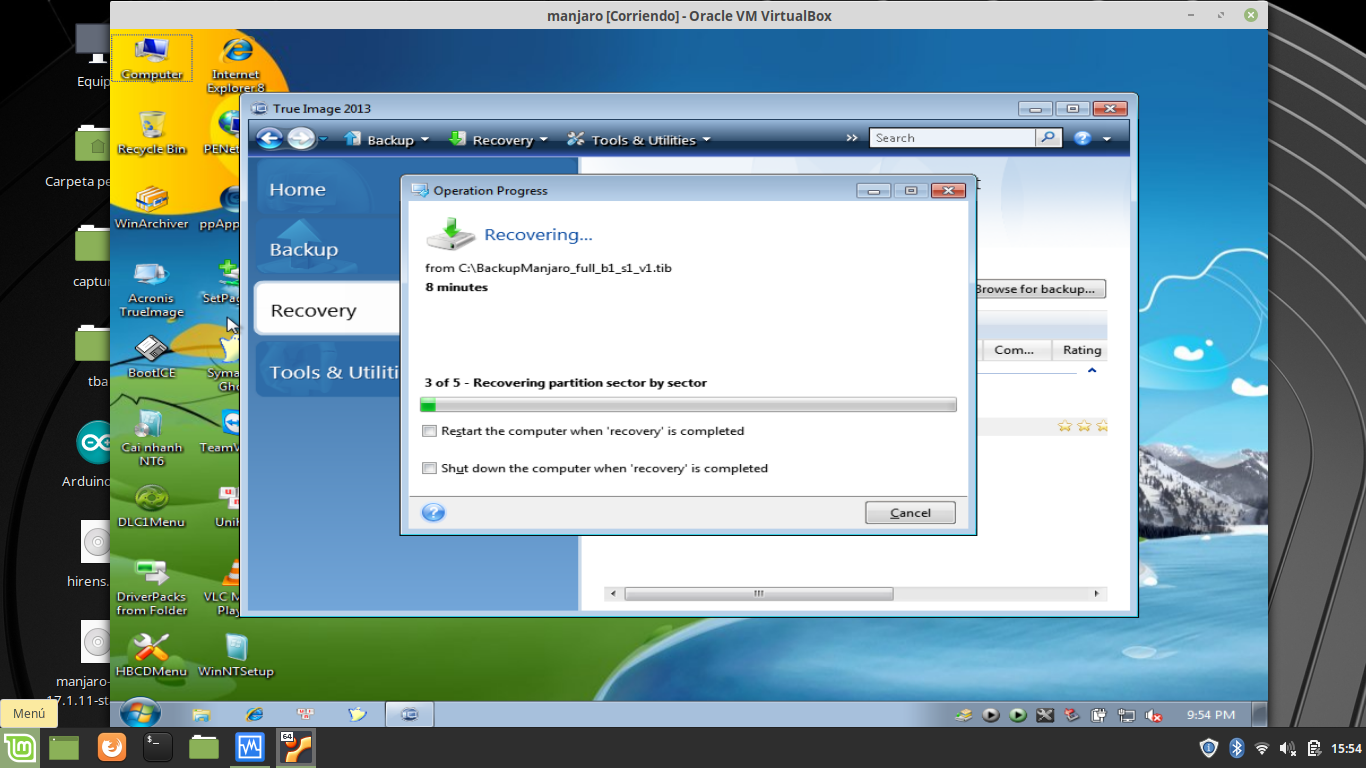
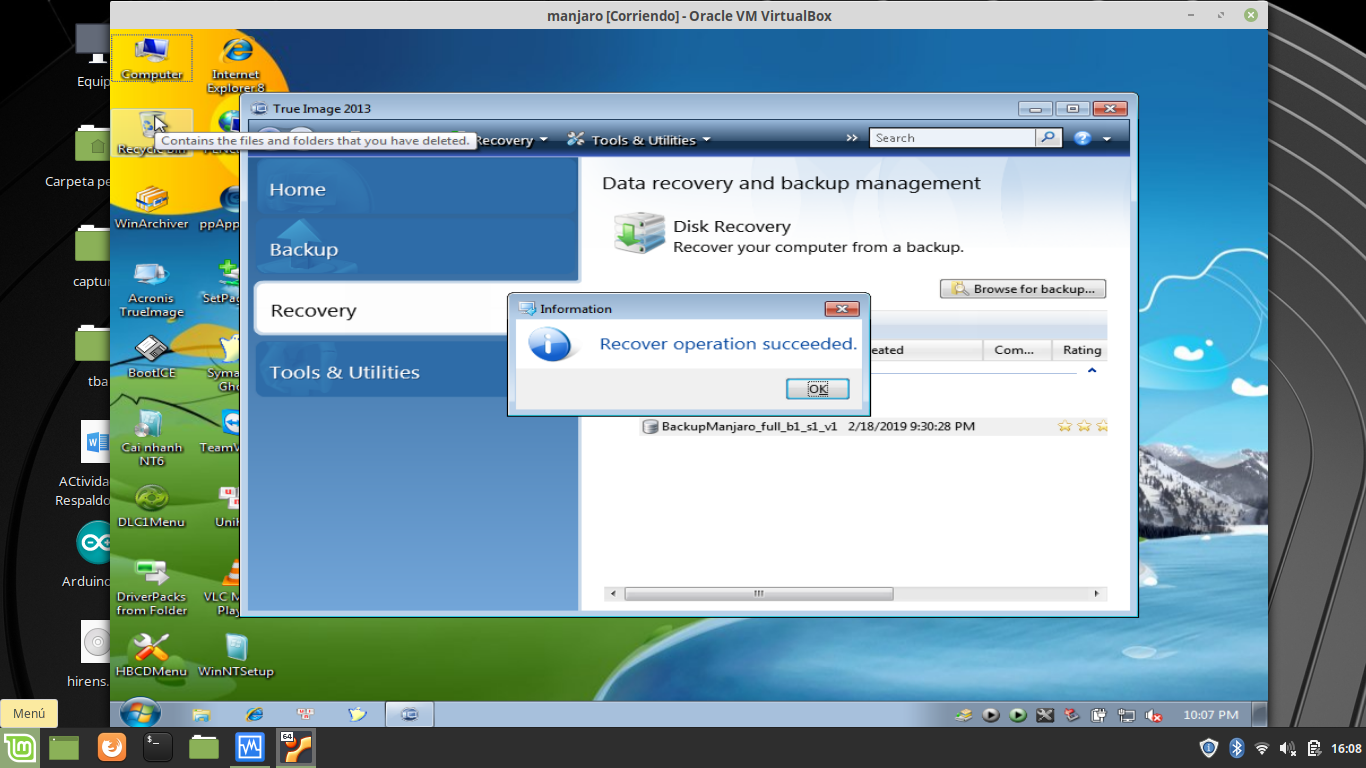
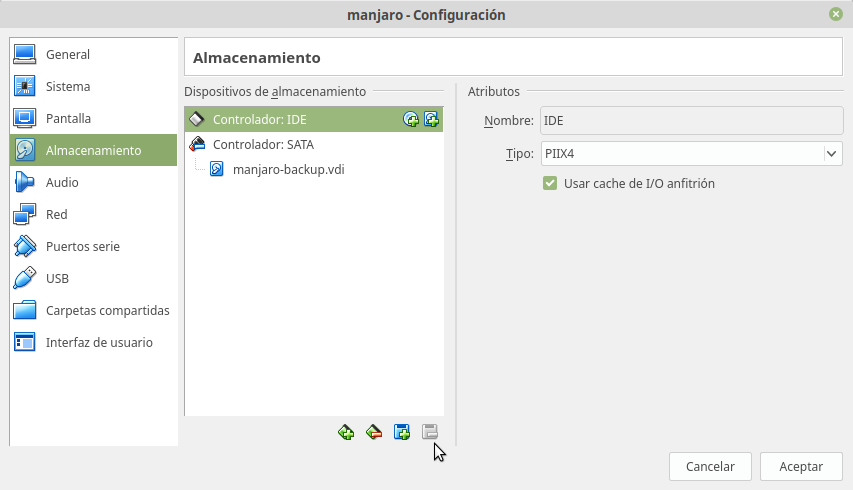
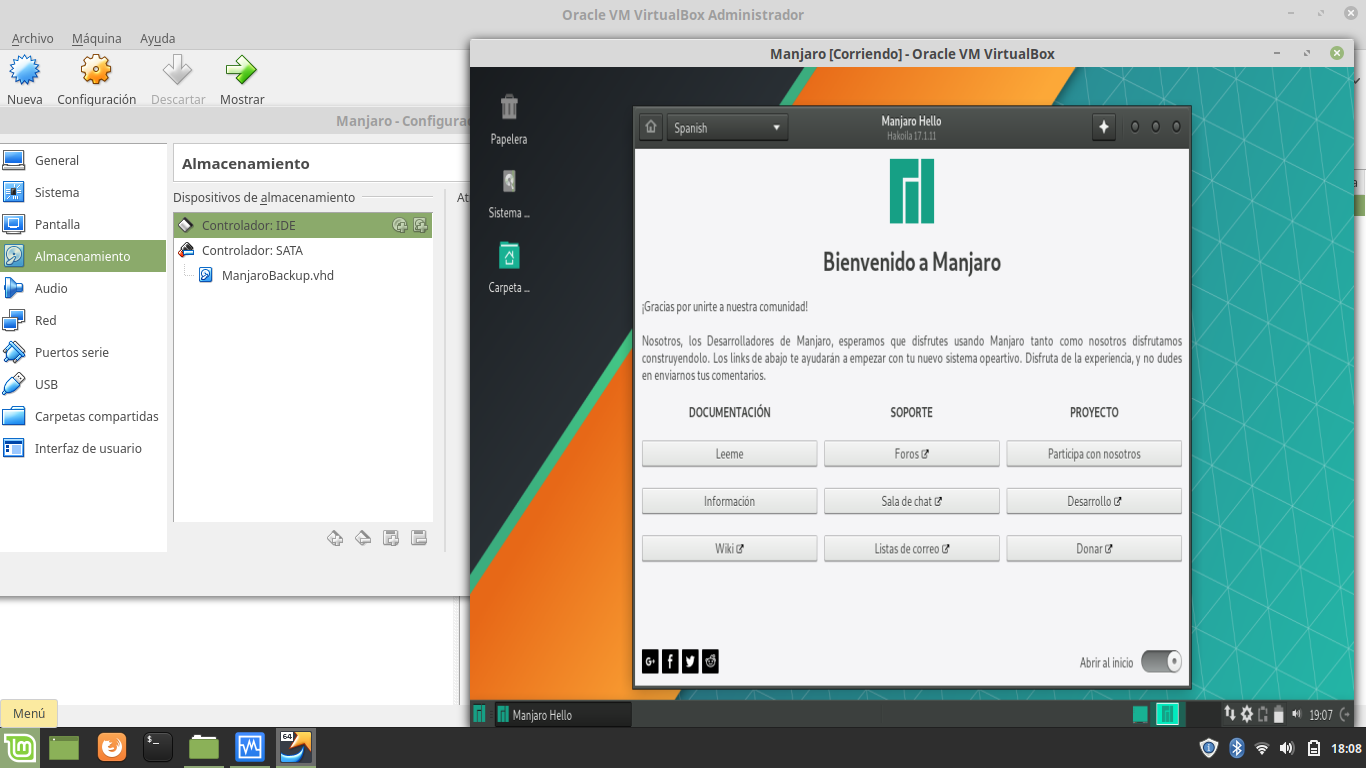
Ayala Márquez Guadalupe Janeth

Morales Rentería Alan André

Silva Ruiz Diego Adán

Díaz Miguel

### Recuperar backup:

1. Ingresando a la misma utileria de hirens (Acronis True Image), Nos iremos al apartado de recovery, en el cual deberiamos ver lo siguiente:
2. 
3. En caso de que no detecte el backup, deberemos dar click en “browse for backups” y buscar el archivo de respaldo. Daremos click derecho en el archivo de respaldo.
4. Daremos click en la opcion “recover”, despues seleccionaremos la casilla “Recover whole disk and partitions” y tambien “sector by sector”.
5. 
6. Seleccionamos el disco (y todas sus particiones) que vamos a recuperar.
7. Para los pasos de “settings of partition 1-1” y “settings of partition 1-2”, dejamos las opciones tal y como estan y damos click en siguiente.
8. 
9. Seleccionamos el disco en el que vamos a recuperar la informacion, este caso el disco de 25 GB.
10. La siguiente ventana nos mostrara lo que realizara, si no hay errores, damos click en “Proceed” para empezar la recuperación.
11. 
12. Esperamos a que termine de cargar…
13. Deberíamos ver lo siguiente:
14. 
15. Para comprobar que funciona, lo que haremos sera desconectar los discos y dejar solamente el de “manjaro-backup”.
16. Iniciamos la maquina, y veremos lo siguiente:
    * 1. 

### Recuperación de la información de la base de datos MySQL:

Continuando con la contraparte de guardar la información de una base de datos, ahora sigue recuperar esa información en dicha base de datos. Cabe declarar que aunque hay varias formas de realizar esta operación, nosotros propusimos la mas sencilla, la cual consiste ingresar el conjunto de registros de una tabla a la vez.

Lo primero que hicimos fue agregar el formulario al area de la vista de configuracion:

<form action="{{asset('/save')}}" method="post" role="form" enctype="multipart/form-data">

@csrf

<center>

<label>Cargar datos en la tabla</label>

<select class="selectpicker" required="" name="tabla">

<option>Insumos</option>

<option>Clientes</option>

<option>Vehiculos</option>

<option>Choferes</option>

</select>

<input type="file" name="file" required="">

<button class="btn btn-default" type="submit">Subir Backup</button>

</center>

</form>

La parte importante, corresponde a la función “asset(/save)” la cual, después de presionar el boton, enviá el formulario, y los datos del mismo, a la ruta correspondiente.

Route::get('/configuracion', 'HomeController@configuracion');

Route::post('/configuracion', 'HomeController@configuracionupdate');

Route::post('/save', 'HomeController@save');

Vemos que la ruta “/save”, la gestiona el controlador “HomeController”, específicamente la función “save” mediante el método de paso de datos “POST”.

Empecemos a ver la función “save”.

public function save(Request $request){

$fileName = 'archivo1-'.time().'.'.request()->file->getClientOriginalExtension();

request()->file->move(public\_path('files'), $fileName);

$linea = 0;

$archivo = fopen(public\_path('files/').$fileName, "r");

Lo primero que hacemos es mover el archivo que cargamos al formulario al servidor, donde le asignamos como nombre, “archivo1-” seguido de la fecha actual y por ultimo la extencion del archivo. Una vez tenemos el archivo en el servidor, procedemos a abrirlo.

$tabla = $request->get('tabla');

$sql="";

switch ($tabla) {

case 'Insumos':

# code...

$sql = "INSERT INTO `insumos`(`idinsumo`, `nombre`, `descripcion`, `existencia`, `preciocompra`, `precioventa`, `stockminimo`, `stockmaximo`, `ubicacion`, `imagen`, `status`) VALUES";

db::select("delete from insumos where 1;");

break;

case 'Clientes':

# code...

$sql = "INSERT INTO `clientes`(`rfc`, `nombre`, `direccion`, `telefono`, `saldo`, `mail`, `status`, `tipotarjeta`, `tarjeta`, `banco`) VALUES";

db::select("delete from clientes where 1;");

break;

case 'Vehiculos':

# code...

$sql = "INSERT INTO `vehiculos`(`placas`, `marca`, `linea`, `modelo`, `kilometraje`, `color`, `capacidad`, `costorenta`, `imagen`, `status`) VALUES";

db::select("delete from clientes where 1;");

break;

case 'Choferes':

# code...

$sql = "INSERT INTO `chofer`(`rfc`, `idusuario`, `nombre`, `direccion`, `telefono`, `vigencialicencia`, `status`, `imagen`) VALUES";

db::select("delete from chofer where 1;");

break;

}

Después lo que hacemos, corroboramos a que tabla cargaremos los registros, evidentemente estas tablas ya deberían tener registros, por ello según la tabla que corresponda, eliminaremos los registros anteriores mediante la funcion “<db::select>(“delete from TABLA” where 1;)”. Aunque esta función esta pensada para servir solo para realizar consultas select en la base de datos, tambien la podemos usar como comodin para mandar sentencias SQL enteras, llamar procedimientos almacenados, funciones, etc.

Como se puede observar según que tabla seleccionamos creamos un cadena con la base de su sentencia SQL para la inserción de datos, pero, sin los datos.

while (($datos = fgetcsv($archivo, ",")) == true)

{

$num = count($datos);

$linea++;

$aux = $sql."(";

foreach ($datos as $dato) {

$aux=$aux."'".$dato."',";

}

$aux=substr($aux, 0, -1).");";

db::select($aux);

}

Ahora si viene la parte de la inserción de los datos, Lo que hacemos es que mientras haya registros en el archivo csv, los cuales están separados por comas, haga lo siguiente. Crea una cadena con la sentencia de inserción precargada y añádele un “(”. después, para cada uno de los datos del registro, añádelo a la cadena encapsulandolo por comillas simples, seguido de una coma. Antes de cerrar el paracentesis, debemos quitar la ultima coma, para ello mandamos la función substring y le decimos que de la variable $aux, la cual es donde vamos construyendo la sentencia, solo lea desde el carácter 0 hasta el ultimo menos el ultimo(-1). Y a lo que nos devuelve la función le concatenamos “);” para finalizar la construcción de la sentencia. Por ultimo realiza la consulta en la base de datos con la función “<db::select>”.

Para finalizar la función save, hacemos lo siguiente.

fclose($archivo);

$iva = DB::table('iva')

->orderBy('fecha','DESC')

->first();

$estado = "Configuracion Guardada";

return view('configuracion')

->with('titulo','Home')

->with('subtitulo','Bienvenido')

->with('iva',$iva)

->with('estado',$estado)

;

Cerramos el archivo, obtenemos los datos que requiere la vista y regresamos la vista, pero con la variable de estado con “Configuración Guardada”, para saber que no hubo errores.

**CONCLUSIONES:**

***General*:**

El respaldo de información es una manera de proteger los datos de importancia, ya que si llegara a pasar algún daño, error o pérdida en la base de datos en la cual guardamos nuestra información o alguna otra fuente, podemos recuperar esos datos y con ello evitar posibles problemas generados por la pérdida de los datos. Siempre es importante realizar respaldos semanalmente, ya que la información se actualiza constante mente, esto nos ayuda a tener nuestro respaldo con datos actualizados.

Existen sofwares muy seguros y sencillos de utilizar, los cuales nos ayudan a realizar esta operación de respaldo y con ello lograr es resguardo de nuestro información de manera exitosa.

***Janeth Ayala:***

Muchas personas no tienen en cuidado de respaldar su información, ya que no le toman importancia a este aspecto, pero es algo que debemos realizar constantemente, no estamos exentos de que nos suceda algún fallo de nuestra celular o computadora y a causa de esto perder nuestra información, por lo que es algo que no podremos recuperar, pero si realizamos respaldos constantemente podremos recuperar estos datos fácilmente en caso de que se nos presente algún problema, y con ello seguir usando esta información.

**Alan Morales:**

Como informático, saber hacer respaldos es de las cosas más indispensables, dado que de los primeros requerimientos de todo sistema de información fiable y seguros es que la misma información pueda ser accesible en cualquier momento, tenerla respaldada y saber cómo recuperar esta información nos proveen de estos aspectos. Luego de realizar la práctica me hizo recordar los cursos del cecatis que tome en la preparatoria, los cuales mayormente eran sobre formatear y clonar discos.

**Diego Silva**

El respaldo de la información actualmente juega un papel sumamente importante para protección de los datos que se necesitan día con día, dentro de las empresas y también los usuarios debes y hacemos respaldos de nuestra información para no perderla en caso de que un dispositivo se dañe y ya no podamos recuperarla nunca jamás, hay muchas medidas para ello actualmente, como utilizar un disco duro externo, servicios de nube, hay gente que aun utiliza CD's aunque no le veo tanta utilidad porque no tienen mucho espacio y tienen un tiempo de vida reducido y son delicados.

**Díaz Miguel**

El realizar el respaldo de la información es bastante útil e importante ya que tener un respaldo de información previene de perder la misma ya que al haber guardado previamente todo si por alguna razón falla o se pierde la información existe un duplicado de dicha