

1. Crear y leer registros desde php

```
<?php
$redis = new Redis();
$conn = $redis->connect('127.0.0.1', 6379);

if ($conn) {
    echo "Conexion exitosa: estas conectado a Redis...";
    $command = true;
}
else {
    echo "No se pudo conectar con la base de datos de Redis :(";
}

if ($command) {
    //Asignamos los parametros: clave - valor
    $redis->set("llave", "Hola mundo");
    echo "llave=" . $redis->get("llave");
}
```

2. Crear y leer listas desde php

```
<?php
$redis = new Redis();
$conn = $redis->connect('127.0.0.1', 6379);

if ($conn) {
    echo "Conexion exitosa: estas conectado a Redis...</br>";
    $command = true;
}
else {
    echo "No se pudo conectar con la base de datos de Redis :(";
}

if ($command) {
    //Creamos un arreglo
    $redis->lpush("lista-bd", "Redis");
    $redis->lpush("lista-bd", "Oracle");
    $redis->lpush("lista-bd", "PostgreSQL");
    $redis->lpush("lista-bd", "Mysql");
    $redis->lpush("lista-bd", "Mongo");
    //Asignamos el arreglo a una variable
    $listbd = $redis->lrange("lista-bd", 0, 3);
    //Ciclo que muestra cada valor del arreglo
    foreach ($listbd as $nombre) {
        echo "<br>" . $nombre;
    }
}
```

3. Incrementar y decrementar valores

```
<?php
/**
```

incr y decr incrementan o disminuyen en 1 el valor de la clave especificada.
incrby y decrby incrementan o disminuyen el valor de la clave especificada en el argumento proporcionado.

```
*/  
//Inicializa la instancia de redis  
$redis = new Redis();  
//Crea la conexion a redis  
$conn = $redis->connect('127.0.0.1', 6379);  
//Si se conecto muestra un mensaje de conexion exitosa  
if ($conn) {  
    echo "Conexion exitosa: estas conectado a Redis...</br>";  
}  
//Si no se conecto muestra un mensaje de error  
else {  
    echo "No se pudo conectar con la base de datos de Redis :(";  
    exit();  
}  
//Inicializa la variable en 100  
$redis->set("counter", "100");  
//Le suma 1 osea 100+1=101  
$redis->incr("counter");  
//Le suma 9 osea 101+9=110  
$redis->incrBy("counter", 9);  
//Le resta 4 osea 110-4=106  
$redis->decrBy("counter", 4);  
//Le resta 1 osea 106-1=105  
$redis->decr("counter");  
//Muestra el resultado  
echo $redis->get("counter");
```

4. Leer y crear hash desde php

```
<?php  
//Inicializa la instancia de redis  
$redis = new Redis();  
//Crea la conexion a redis  
$conn = $redis->connect('127.0.0.1', 6379);  
//Si se conecto muestra un mensaje de conexion exitosa  
if ($conn) {  
    echo "Conexion exitosa: estas conectado a Redis...<br/>";  
}  
//Si no se conecto muestra un mensaje de error  
else {  
    echo "No se pudo conectar con la base de datos de Redis :(";  
    exit();  
}  
echo "*** Hashes <br/>";  
//Al hash empleado:soporte le asignamos un arreglo de datos  
$redis->hset("empleado:soporte", "nombre", "Eugenio");  
$redis->hset("empleado:soporte", "apellidos", "Chaparro Maya");
```

```

$redis->hset("empleado:soprote", "email", "eucm2g@gmail.com");
$redis->hset("empleado:soprote", "edad", "32");
//Sacamos el array de llaves que contiene el hash empleado:soprote
$keys = $redis->hkeys("empleado:soprote");
//Hacemos un ciclo de todas la llaves del hash empleado:soprote
foreach ($keys as $key) {
    //Obtenemos el valor de cada llave contenida en el hash
    $valor=$redis->hget("empleado:soprote", $key);
    //Mostramos la llave y su valor
    echo $key . ": " . $valor . "<br/>";
}

```

5. manejo de String append, substr y strlen

```

<?php
/*
    append añade la cadena propocionada al final del valor de la clave especificada.
    substr devuelve una cadena de la clave especificada comprendida entre dos
    posiciones.
    strlen devuelve el número de caracteres del valor de la clave especificada.
*/
$redis = new Redis();
$conn = $redis->connect('127.0.0.1', 6379);
//Si la conexion fue exitosa manda un mensaje
if ($conn) {
    echo "Conexion exitosa: estas conectado a Redis...<br/>";
}
//Si la conexion no fue exitosa manda un mensaje y sale del sistema
else {
    echo "No se pudo conectar con la base de datos de Redis :(";
    exit();
}
//Creamos la llave con un valor
$redis->set("saludo", "Hola ");
echo "<strong>Valor incial</strong><br/>".$redis->get("saludo") . "<br/><br/>";

//Le agregamos un string al final
$redis->append("saludo", "Mundo!");
echo "<strong>Le agregamos un string al final</strong><br/>".$redis->get("saludo").
"<br/><br/>";

//Sacamos de la letra 5 a la ultima
$res=$redis->substr("saludo", 5, -1) . "<br/>";
echo "<strong>Sacamos de la letra 5 a la ultima</strong><br/>".$res. "<br/><br/>";

//Sacamos de la 0 a la 2
$res=$redis->substr("saludo", 0, 2) . "<br/>";
echo "<strong>Sacamos de la 0 a la 2</strong><br/>".$res. "<br/><br/>";

//Muestra el tamaño de la llave

```

```
echo "<strong>Muestra el tamaño de la llave </strong><br/>" . $redis->strlen("saludo")  
 . "<br/><br/>";
```

```
//Borra la llave  
$redis->del("saludo");
```

6. manejo de conjuntos

```
<?php
```

```
/*
```

```
union: une 2 o mas conjuntos
```

```
sinter: Extra los elementos en comun de 2 conjuntos
```

```
sdiff: Los elementos de un conjunto que no esten en otro
```

```
*/
```

```
//Creamos la instancia de redis
```

```
$redis = new Redis();
```

```
//Nos conectamos a redis
```

```
$conn = $redis->connect('127.0.0.1', 6379);
```

```
//Si la conexion fue exitosa mostramos este texto
```

```
if ($conn) {
```

```
    echo "Conexion exitosa: estas conectado a Redis...<br/>";
```

```
}
```

```
//Si la conexion no fue exitosa mostramos este texto y nos salimos
```

```
else {
```

```
    echo "No se pudo conectar con la base de datos de Redis :(";
```

```
    exit();
```

```
}
```

```
//Borramos la key (para evitar futuros conflictos)
```

```
$redis->del("empleados:desarrollo");
```

```
$redis->del("empleados:soporte");
```

```
//Agregamos un arreglo de todos los empleados de desarrollo
```

```
$redis->sadd("empleados:desarrollo", "empleado1", "empleado2", "empleado3");
```

```
//Agregamos un arreglo de todos los empleados de soporte
```

```
$redis->sadd("empleados:soporte", "empleado3", "empleado4", "empleado5");
```

```
//Unimos todos los empleados
```

```
$totalEmpleados = $redis->union("empleados:soporte", "empleados:desarrollo");
```

```
//Sacamos solo los empleados que esten en soporte y tambien en desarrollo
```

```
$comunesEmpleados = $redis->sinter("empleados:soporte", "empleados:desarrollo");
```

```
//Empleados exclusivamente de desarrollo
```

```
$empleadosSoloDesarrollo = $redis->sdiff("empleados:desarrollo",  
"empleados:soporte");
```

```
//Mostramos los resultados de union
```

```
echo "<br><strong>Unimos todos los empleados:</strong> " . "<br/>";
```

```
foreach ($totalEmpleados as $empleado) {
```

```
    echo $empleado . "<br/>";
```

```
}
```

```
//Mostramos los resultados de sdiff
echo "</br><strong>Sacamos solo los empleados que esten en soporte y tambien en
desarrollo:</strong> " . "<br/>";
foreach ($comunesEmpleados as $empleado) {
    echo $empleado . "<br/>";
}
```

```
//Mostramos los resultados de sunion
echo "</br><strong>Empleados exclusivamente de desarrollo:</strong> " . "<br/>";
foreach ($empleadosSoloDesarrollo as $empleado) {
    echo $empleado . "<br/>";
}
```

```
//Borramos las llaves
$redis->del("empleados:desarrollo");
$redis->del("empleados:soporte");
```

7. manejo de transacciones

```
<?php
/*
```

TRANSACTIONS

Las transacciones se inician con multi(). En Redis hay dos modos transaccionales:

REDIS:MULTI: El bloque de instrucciones se ejecuta como una única transacción. Este es el valor predeterminado.

REDIS:PIPELINE: El bloque de instrucciones se transmite más rápido al servidor, pero sin que quede garantizada la atomicidad.

Para finalizar la transacción usaremos:

exec(): ejecuta las instrucciones de una transacción.
discard(): cancela las instrucciones de una transacción.

```
*/
```

```
//Creamos la instancia de redis
```

```
$redis = new Redis();
```

```
//Nos conectamos a redis
```

```
$conn = $redis->connect('127.0.0.1', 6379);
```

```
//Si la conexion fue exitosa mostramos este texto
```

```
if ($conn) {
```

```
    echo "Conexion exitosa: estas conectado a Redis...<br/>";
```

```
}
```

```
//Si la conexion no fue exitosa mostramos este texto y nos salimos
```

```
else {
```

```
    echo "No se pudo conectar con la base de datos de Redis :(";
```

```
    exit();
```

```
}
```

```
//Inicializamos la transaccion
```

```
$tran1 = $redis->multi();
```

```
//Agregamos a la transaccion 2 llaves con sus respectivos valores
```

```
$tran1->set("a", "1");
```

```
$tran1->set("b", "2");  
//Ejecutamos la transaccion  
$tran1->exec();  
//Mostramos los resultados despues de la transaccion  
echo "<br/><strong>Despues de hacer exec()</strong><br/>";  
echo "a:" . $redis->get("a") . " b:" . $redis->get("b") . "<br/><br/>";
```

```
$tran2 = $redis->multi();  
$tran2->set("a", "3");  
$tran2->set("b", "4");  
//Desacarta la actual transaccion y deja los datos previos a la transaccion actual  
$tran2->discard();  
//Mostramos los resultados despues de discard  
echo "<strong>Despues de hacer discard()</strong><br/>";  
echo "a:" . $redis->get("a") . " b:" . $redis->get("b") . "<br/>";
```