```
var map1 = function(){
      emit(this.name, this.numProds);
};

var reduce1 = function(name, values){
  const reducer = (accumulator, currentValue) => accumulator + currentValue;
  var n = values.reduce(reducer);
  return n;
};

// > db.clientes.mapReduce(map1, reduce1, {out:"res"});


var map2 = function(){
  this.details.forEach((value)=>{
      emit(value.prod, value.price);
  });
};

var reduce2 = function(prod, prices){
  const reducer = (accumulator, currentValue) => accumulator + currentValue;
  var n = prices.reduce(reducer);
  return n;
};


// Restaurantes por zipCode

var map3 = function(){
  emit(this.address.zipcode, 1);
};

var reduce3 = function(name, values){
  return values.length;
};
// db.restaurantes.mapReduce(map3,reduce3,{out:"resZip"});

// 2. ¿Cuántos grades hay por usuario?
var map3 = function(){
  emit(this.name, this.grades.length);
};

var reduce3 = function(name, values){
  const reducer = (accumulator, currentValue) => accumulator + currentValue;
  var n = values.reduce(reducer);
  return n;
};
// db.restaurantes.mapReduce(map3,reduce3,{out:"resGrades"});


// 3. Cuál es el total de scores por usuario
```

```
var map3 = function(){
  this.grades.forEach((value)=>{
      emit(this.name, value.score);
  });
};

var reduce3 = function(prod, scores){
  const reducer = (accumulator, currentValue) => accumulator + currentValue;
  var n = scores.reduce(reducer);
  return n;
};
// db.restaurantes.mapReduce(map3,reduce3,{out:"resGrades"});

// 4. Cuántos restaurantes obtuvieron el grade A, Grade B y Grade C
var map3 = function(){
  this.grades.forEach((value)=>{
      emit(value.grade, 1);
  });
};
var reduce3 = function(grade, grades){
  const reducer = (accumulator, currentValue) => accumulator + currentValue;
  var n = grades.reduce(reducer);
  return n;
};
//db.restaurantes.mapReduce(map3,reduce3,{out:"resGrades2"});


// 5. Agrupa los restaurantes por “cuisine” y cuenta cuántos restaurantes hay
por cada categoría
var map3 = function(){
      emit(this.cuisine, 1);
};
var reduce3 = function(grade, grades){
  const reducer = (accumulator, currentValue) => accumulator + currentValue;
  var n = grades.reduce(reducer);
  return n;
};
// db.restaurantes.mapReduce(map3,reduce3,{out:"resCuisine"});

// 6. Cuáles son los 10 restaurantes más cercanos a:
// Restaurant_id: 40360045

var map3 = function(){
  var testLat = -73.9829239;
  var testLng = 40.6580753;
      var valueCoordLat = this.address.coord[0];
      var valueCoordLng = this.address.coord[0];
      function  distance(lat1, lon1, lat2, lon2, unit) {
        var radlat1 = Math.PI * lat1/180
        var radlat2 = Math.PI * lat2/180
        var theta = lon1-lon2
        var radtheta = Math.PI * theta/180
        var dist = Math.sin(radlat1) * Math.sin(radlat2) + Math.cos(radlat1)
* Math.cos(radlat2) * Math.cos(radtheta);
        dist = Math.acos(dist)
        dist = dist * 180/Math.PI
        dist = dist * 60 * 1.1515
        if (unit=="K") { dist = dist * 1.609344 }
        if (unit=="N") { dist = dist * 0.8684 }
        return dist
      }
      var distanceFrom = distance(testLat,testLng,valueCoordLat,testLng,'K');
      emit(this.name, distanceFrom);
};
var reduce3 = function(grade, grades){
  const reducer = (accumulator, currentValue) => accumulator + currentValue;
  var n = grades.reduce(reducer);
  return n;
};
//
db.restaurantes.mapReduce(map3,reduce3,{out:"resDistance",order:1,limit:10});
```