Se detalla en este documento, la api para descargar datos (Pedidos, Cobranzas, y Motivos de no compra), y para cargar datos a la nube (Artículos, Clientes, etc.)
El modelo de Api Rest, permite la disponibilidad de datos en tiempo real, la experiencia para el usuario se vuelve transparente, el margen de error se reduce a cero y el manejo de datos se simplifica en grandes escalas. La idea es que un servicio, una tarea, o un hilo, haga llamadas periódicas a la api, y se encargue del manejo de datos, evitando la intervención de los usuarios.

Por detalles sobre las tablas y los campos, al final del documento se encuentra toda la información necesaria.

Para todas las llamadas se utiliza un token de prueba, las llamadas retornan datos de prueba, y toda la información que se suba a la nube se almacena en un entorno de pruebas.

Datos de la api

Protocolo: HTTP
IP: 104.131.172.142
Método: POST
Api: chhapi

Un ejemplo para obtener todos los pedidos nuevos del día 13 de mayo del 2016:

La url quedaría compuesta de la siguiente manera:

http://104.131.172.142/chhapi/

los parámetros:

```
{
        action: 'verPedidos',
        token: '73b1f30a5879bc43c18e8f8cb5ab2f78',
        st: 'N',
        timeStart: '2016-05-13',
        timeEnd: '2016-05-13'
}
```

Detallo cada cosa:
La url es la composición de protocolo, la ip, y la api
Los parámetros, están en formato JSON
* action: es un string “verPedidos” y va cambiando según que dato se requiera de la api
* token: es un string “73b1f30a5879bc43c18e8f8cb5ab2f78”, identificador único por cliente que usamos para que la api recupere los datos
* st: es un string “N”, en este caso solicita los pedidos nuevos, para pedidos que ya fueron descargados se pueden solicitar los pedidos con estado “O"
* timeStart: es un string “2016-05-13" fecha “desde” la que se quieren obtener pedidos.
* timeEnd: es un string “2016-05-13" fecha “hasta” la que se quieren obtener pedidos.

Después de esa llamada el servidor da un respuesta, también en formato JSON, paso una respuesta para los parámetros del ejemplo de arriba:


```
{
  "02220160513111442": {
    "cabecera": {
      "mpk": "02220160513111442",
      "PH_CODIGO": "02220160513111442",
      "CL_CODIGO": "01554",
      "PH_FECINI": "2016-05-13",
      "PH_HORINI": "11:14:42",
      "PH_FECFIN": "2016-05-13",
      "PH_HORFIN": "11:14:47",
      "PH_FECMOD": "0000-00-00",
      "PH_HORMOD": "null",
      "PH_FECENT": "2016-05-14",
      "CVE_CODIGO": "001",
      "PH_TOTCOM": "5.05",
      "PH_NETO": "4.17",
      "PH_CANITEM": "1",
```

```
    "VEN_CODIGO": "022",
    "PH_OBSERV": "",
    "PH_BONIFI": "",
    "PH_RECARG": "null",
    "PH_COMREF": "null",
    "PH_ORIGEN": "H",
    "EM_CODIGO": "9999",
    "PH_TIPODOC": "1",
    "PH_CODCOM": "null",
    "PH_TIPCOM": "null",
    "PH_SUCCOM": "null",
    "PH_NUMCOM": "null",
    "PH_CODREF": "null",
    "PH_TIPREF": "null",
    "PH_SUCREF": "null",
    "PH_NUMREF": "null",
    "PH_ANULADO": "null",
    "PH_IDORIGEN": "null",
    "ST": "N",
    "numPedido": "598"
   },
   "detalle": [
    {
     "mpk": "0222016051311144211350-1",
     "PH_CODIGO": "02220160513111442",
     "AR_CODIGO": "1135",
     "UN_CODIGO": "null",
     "PHD_CANTID": "1.00",
     "PHD_IMPUNI": "5.05",
     "PHD_BONIFI": "0",
     "PHD_IMPLIS": "4.17",
     "LP_CODIGO": "01",
     "AR_CODSUS": "null",
     "PHD_COMREF": "null",
     "HH_LINEA": "1040",
     "ST": "N"
    }
   ]
  },
  "02220160513162856": {
   "cabecera": {
    "mpk": "02220160513162856",
    "PH_CODIGO": "02220160513162856",
    "CL_CODIGO": "01348",
    "PH_FECINI": "2016-05-13",
    "PH_HORINI": "16:28:56",
    "PH_FECFIN": "2016-05-13",
    "PH_HORFIN": "16:30:13",
    "PH_FECMOD": "0000-00-00",
    "PH_HORMOD": "null",
    "PH_FECENT": "2016-05-14",
    "CVE_CODIGO": "002",
    "PH_TOTCOM": "260.84",
    "PH_NETO": "215.57",
    "PH_CANITEM": "5",
    "VEN_CODIGO": "022",
    "PH_OBSERV": "",
    "PH_BONIFI": "",
    "PH_RECARG": "null",
    "PH_COMREF": "null",
    "PH_ORIGEN": "H",
    "EM_CODIGO": "0001",
    "PH_TIPODOC": "1",
    "PH_CODCOM": "null",
    "PH_TIPCOM": "null",
    "PH_SUCCOM": "null",
```

```
  "PH_NUMCOM": "null",
  "PH_CODREF": "null",
  "PH_TIPREF": "null",
  "PH_SUCREF": "null",
  "PH_NUMREF": "null",
  "PH_ANULADO": "null",
  "PH_IDORIGEN": "null",
  "ST": "N",
  "numPedido": "599"
},
"detalle": [
 {
  "mpk": "0222016051316285625540-1",
  "PH_CODIGO": "02220160513162856",
  "AR_CODIGO": "2554",
  "UN_CODIGO": "null",
  "PHD_CANTID": "1.00",
  "PHD_IMPUNI": "3.04",
  "PHD_BONIFI": "0",
  "PHD_IMPLIS": "2.51",
  "LP_CODIGO": "01",
  "AR_CODSUS": "null",
  "PHD_COMREF": "null",
  "HH_LINEA": "1043",
  "ST": "N"
 },
 {
  "mpk": "0222016051316285625550-1",
  "PH_CODIGO": "02220160513162856",
  "AR_CODIGO": "2555",
  "UN_CODIGO": "null",
  "PHD_CANTID": "6.00",
  "PHD_IMPUNI": "27.13",
  "PHD_BONIFI": "0",
  "PHD_IMPLIS": "22.42",
  "LP_CODIGO": "01",
  "AR_CODSUS": "null",
  "PHD_COMREF": "null",
  "HH_LINEA": "1044",
  "ST": "N"
 },
 {
  "mpk": "0222016051316285625580-1",
  "PH_CODIGO": "02220160513162856",
  "AR_CODIGO": "2558",
  "UN_CODIGO": "null",
  "PHD_CANTID": "1.00",
  "PHD_IMPUNI": "29.89",
  "PHD_BONIFI": "0",
  "PHD_IMPLIS": "24.70",
  "LP_CODIGO": "01",
  "AR_CODSUS": "null",
  "PHD_COMREF": "null",
  "HH_LINEA": "1045",
  "ST": "N"
 },
 {
  "mpk": "0222016051316285625520-1",
  "PH_CODIGO": "02220160513162856",
  "AR_CODIGO": "2552",
  "UN_CODIGO": "null",
  "PHD_CANTID": "3.00",
  "PHD_IMPUNI": "16.29",
  "PHD_BONIFI": "0",
  "PHD_IMPLIS": "13.46",
  "LP_CODIGO": "01",
```

```
      "AR_CODSUS": "null",
      "PHD_COMREF": "null",
      "HH_LINEA": "1041",
      "ST": "N"
    },
    {
      "mpk": "0222016051316285625530-1",
      "PH_CODIGO": "02220160513162856",
      "AR_CODIGO": "2553",
      "UN_CODIGO": "null",
      "PHD_CANTID": "2.00",
      "PHD_IMPUNI": "8.14",
      "PHD_BONIFI": "0",
      "PHD_IMPLIS": "6.73",
      "LP_CODIGO": "01",
      "AR_CODSUS": "null",
      "PHD_COMREF": "null",
      "HH_LINEA": "1042",
      "ST": "N"
    }
  ]
},
"02220160513163051": {
  "cabecera": {
    "mpk": "02220160513163051",
    "PH_CODIGO": "02220160513163051",
    "CL_CODIGO": "01554",
    "PH_FECINI": "2016-05-13",
    "PH_HORINI": "16:30:51",
    "PH_FECFIN": "2016-05-13",
    "PH_HORFIN": "16:31:40",
    "PH_FECMOD": "0000-00-00",
    "PH_HORMOD": "null",
    "PH_FECENT": "2016-05-14",
    "CVE_CODIGO": "001",
    "PH_TOTCOM": "54.91",
    "PH_NETO": "52.78",
    "PH_CANITEM": "5",
    "VEN_CODIGO": "022",
    "PH_OBSERV": "",
    "PH_BONIFI": "",
    "PH_RECARG": "null",
    "PH_COMREF": "null",
    "PH_ORIGEN": "H",
    "EM_CODIGO": "9999",
    "PH_TIPODOC": "1",
    "PH_CODCOM": "null",
    "PH_TIPCOM": "null",
    "PH_SUCCOM": "null",
    "PH_NUMCOM": "null",
    "PH_CODREF": "null",
    "PH_TIPREF": "null",
    "PH_SUCREF": "null",
    "PH_NUMREF": "null",
    "PH_ANULADO": "null",
    "PH_IDORIGEN": "null",
    "ST": "N",
    "numPedido": "600"
  },
  "detalle": [
    {
      "mpk": "022201605131630512598 20-1",
      "PH_CODIGO": "02220160513163051",
      "AR_CODIGO": "2598",
      "UN_CODIGO": "null",
      "PHD_CANTID": "5.00",
```

```
  "PHD_IMPUNI": "8.95",
  "PHD_BONIFI": "20",
  "PHD_IMPLIS": "7.40",
  "LP_CODIGO": "01",
  "AR_CODSUS": "null",
  "PHD_COMREF": "null",
  "HH_LINEA": "1050",
  "ST": "N"
},
{
  "mpk": "0222016051316305112520-1",
  "PH_CODIGO": "02220160513163051",
  "AR_CODIGO": "1252",
  "UN_CODIGO": "null",
  "PHD_CANTID": "1.00",
  "PHD_IMPUNI": "0.87",
  "PHD_BONIFI": "0",
  "PHD_IMPLIS": "0.72",
  "LP_CODIGO": "01",
  "AR_CODSUS": "null",
  "PHD_COMREF": "null",
  "HH_LINEA": "1048",
  "ST": "N"
},
{
  "mpk": "0222016051316305112530-1",
  "PH_CODIGO": "02220160513163051",
  "AR_CODIGO": "1253",
  "UN_CODIGO": "null",
  "PHD_CANTID": "1.00",
  "PHD_IMPUNI": "1.27",
  "PHD_BONIFI": "0",
  "PHD_IMPLIS": "1.05",
  "LP_CODIGO": "01",
  "AR_CODSUS": "null",
  "PHD_COMREF": "null",
  "HH_LINEA": "1049",
  "ST": "N"
},
{
  "mpk": "0222016051316305112500-1",
  "PH_CODIGO": "02220160513163051",
  "AR_CODIGO": "1250",
  "UN_CODIGO": "null",
  "PHD_CANTID": "1.00",
  "PHD_IMPUNI": "2.37",
  "PHD_BONIFI": "0",
  "PHD_IMPLIS": "1.96",
  "LP_CODIGO": "01",
  "AR_CODSUS": "null",
  "PHD_COMREF": "null",
  "HH_LINEA": "1047",
  "ST": "N"
},
{
  "mpk": "0222016051316305112580-1",
  "PH_CODIGO": "02220160513163051",
  "AR_CODIGO": "1258",
  "UN_CODIGO": "null",
  "PHD_CANTID": "5.00",
  "PHD_IMPUNI": "2.92",
  "PHD_BONIFI": "0",
  "PHD_IMPLIS": "2.41",
  "LP_CODIGO": "01",
  "AR_CODSUS": "null",
  "PHD_COMREF": "null",
```

```
      "HH_LINEA": "1046",
      "ST": "N"
    }
  ]
},
"02220160513163141": {
  "cabecera": {
    "mpk": "02220160513163141",
    "PH_CODIGO": "02220160513163141",
    "CL_CODIGO": "01554",
    "PH_FECINI": "2016-05-13",
    "PH_HORINI": "16:31:41",
    "PH_FECFIN": "2016-05-13",
    "PH_HORFIN": "16:32:17",
    "PH_FECMOD": "0000-00-00",
    "PH_HORMOD": "null",
    "PH_FECENT": "2016-05-14",
    "CVE_CODIGO": "001",
    "PH_TOTCOM": "43.75",
    "PH_NETO": "36.16",
    "PH_CANITEM": "3",
    "VEN_CODIGO": "022",
    "PH_OBSERV": "",
    "PH_BONIFI": "10",
    "PH_RECARG": "null",
    "PH_COMREF": "null",
    "PH_ORIGEN": "H",
    "EM_CODIGO": "9999",
    "PH_TIPODOC": "1",
    "PH_CODCOM": "null",
    "PH_TIPCOM": "null",
    "PH_SUCCOM": "null",
    "PH_NUMCOM": "null",
    "PH_CODREF": "null",
    "PH_TIPREF": "null",
    "PH_SUCREF": "null",
    "PH_NUMREF": "null",
    "PH_ANULADO": "null",
    "PH_IDORIGEN": "null",
    "ST": "N",
    "numPedido": "601"
  },
  "detalle": [
    {
      "mpk": "0222016051316314125540-1",
      "PH_CODIGO": "02220160513163141",
      "AR_CODIGO": "2554",
      "UN_CODIGO": "null",
      "PHD_CANTID": "1.00",
      "PHD_IMPUNI": "3.04",
      "PHD_BONIFI": "0",
      "PHD_IMPLIS": "2.51",
      "LP_CODIGO": "01",
      "AR_CODSUS": "null",
      "PHD_COMREF": "null",
      "HH_LINEA": "1053",
      "ST": "N"
    },
    {
      "mpk": "0222016051316314125530-1",
      "PH_CODIGO": "02220160513163141",
      "AR_CODIGO": "2553",
      "UN_CODIGO": "null",
      "PHD_CANTID": "1.00",
      "PHD_IMPUNI": "8.14",
      "PHD_BONIFI": "0",
```

```
      "PHD_IMPLIS": "6.73",
      "LP_CODIGO": "01",
      "AR_CODSUS": "null",
      "PHD_COMREF": "null",
      "HH_LINEA": "1052",
      "ST": "N"
    },
    {
      "mpk": "0222016051316314125520-1",
      "PH_CODIGO": "02220160513163141",
      "AR_CODIGO": "2552",
      "UN_CODIGO": "null",
      "PHD_CANTID": "2.00",
      "PHD_IMPUNI": "16.29",
      "PHD_BONIFI": "0",
      "PHD_IMPLIS": "13.46",
      "LP_CODIGO": "01",
      "AR_CODSUS": "null",
      "PHD_COMREF": "null",
      "HH_LINEA": "1051",
      "ST": "N"
    }
  ]
 }
}
```


Explico el despliegue:
Los sub elementos del Json son los pedidos, por lo que el elemento cero es el primer pedido
La clave del elemento es el código del pedido, en este caso 02220160513111442
Los sub elementos un pedido, son “cabecera” y “detalle”
En cabecera del pedido están los datos del pedido
En detalle, los sub elementos son las lineas del pedido

——

Un ejemplo para obtener todas las cobranzas nuevas del día 13 de mayo del 2016:

La url quedaría compuesta de la siguiente manera:

http://104.131.172.142/chhapi/

los parámetros:

```
{
        action: ’verCobranzas’,
        token: ’73b1f30a5879bc43c18e8f8cb5ab2f78’,
        st: ’N',
        timeStart: '2016-05-13',
        timeEnd: '2016-05-13'
}
```

Lo que da como resultado


```
{
  "02220160513163039": {
    "cabecera": {
      "mpk": "02220160513163039",
      "ID_COBRANZA": "02220160513163039",
      "CL_CODIGO": "01348",
      "FEC_INI_COBRANZA": "2016-05-13",
      "HORA_INI_COBRANZA": "16:30:39",
      "FEC_FIN_COBRANZA": "2016-05-13",
      "HORA_FIN_COBRANZA": "16:30:39",
```

```
    "MONTOTOTAL": "217.63",
    "NRO_RECIBO": "99999999",
    "VEN_CODIGO": "022",
    "ESTADO": "0",
    "ST": "N"
   },
   "documentosimputados": [
    {
     "mpk": "02220160513163039000061341",
     "ID_COBRANZA": "02220160513163039",
     "CO_CODIGO": "FAC",
     "CO_SUCURS": "0001",
     "CO_NUMERO": "00061341",
     "CO_TIPO": "A",
     "MONTO": "217.63",
     "CUOTA": "01",
     "HH_LINEA": null,
     "ST": "N"
    }
   ],
   "detallepagos": [
    {
     "mpk": "022201605131630390",
     "ID_COBRANZA": "02220160513163039",
     "NUMERO": "0",
     "ID_MEDIOPAGO": "$",
     "CODIGOBANCO": " ",
     "PLAZA": " ",
     "FECHA": "0000-00-00",
     "FECVEN": "0000-00-00",
     "COTIZACION": "1.00",
     "MONTO": "217.63",
     "HH_LINEA": null,
     "ST": "N"
    }
   ]
  },
  "02220160513163037": {
   "cabecera": {
    "mpk": "02220160513163037",
    "ID_COBRANZA": "02220160513163037",
    "CL_CODIGO": "01348",
    "FEC_INI_COBRANZA": "2016-05-13",
    "HORA_INI_COBRANZA": "16:30:37",
    "FEC_FIN_COBRANZA": "2016-05-13",
    "HORA_FIN_COBRANZA": "16:30:37",
    "MONTOTOTAL": "1405.11",
    "NRO_RECIBO": "99999999",
    "VEN_CODIGO": "022",
    "ESTADO": "0",
    "ST": "N"
   },
   "documentosimputados": [
    {
     "mpk": "02220160513163037000060759",
     "ID_COBRANZA": "02220160513163037",
     "CO_CODIGO": "FAC",
     "CO_SUCURS": "0001",
     "CO_NUMERO": "00060759",
     "CO_TIPO": "A",
     "MONTO": "305.08",
     "CUOTA": "01",
     "HH_LINEA": null,
     "ST": "N"
    },
    {
```

```
      "mpk": "02220160513163037000060760",
      "ID_COBRANZA": "02220160513163037",
      "CO_CODIGO": "FAC",
      "CO_SUCURS": "0001",
      "CO_NUMERO": "00060760",
      "CO_TIPO": "A",
      "MONTO": "1100.03",
      "CUOTA": "01",
      "HH_LINEA": null,
      "ST": "N"
    }
  ],
  "detallepagos": [
    {
      "mpk": "022201605131630370",
      "ID_COBRANZA": "02220160513163037",
      "NUMERO": "0",
      "ID_MEDIOPAGO": "$",
      "CODIGOBANCO": " ",
      "PLAZA": " ",
      "FECHA": "0000-00-00",
      "FECVEN": "0000-00-00",
      "COTIZACION": "1.00",
      "MONTO": "1405.11",
      "HH_LINEA": null,
      "ST": "N"
    }
  ]
},
"02220160513163227": {
  "cabecera": {
    "mpk": "02220160513163227",
    "ID_COBRANZA": "02220160513163227",
    "CL_CODIGO": "01831",
    "FEC_INI_COBRANZA": "2016-05-13",
    "HORA_INI_COBRANZA": "16:32:27",
    "FEC_FIN_COBRANZA": "2016-05-13",
    "HORA_FIN_COBRANZA": "16:32:27",
    "MONTOTOTAL": "407.44",
    "NRO_RECIBO": "99999999",
    "VEN_CODIGO": "022",
    "ESTADO": "0",
    "ST": "N"
  },
  "documentosimputados": [
    {
      "mpk": "02220160513163227000127146",
      "ID_COBRANZA": "02220160513163227",
      "CO_CODIGO": "FAC",
      "CO_SUCURS": "9999",
      "CO_NUMERO": "00127146",
      "CO_TIPO": "B",
      "MONTO": "407.44",
      "CUOTA": "01",
      "HH_LINEA": null,
      "ST": "N"
    }
  ],
  "detallepagos": [
    {
      "mpk": "022201605131632270",
      "ID_COBRANZA": "02220160513163227",
      "NUMERO": "0",
      "ID_MEDIOPAGO": "$",
      "CODIGOBANCO": " ",
      "PLAZA": " ",
```

```
      "FECHA": "0000-00-00",
      "FECVEN": "0000-00-00",
      "COTIZACION": "1.00",
      "MONTO": "407.44",
      "HH_LINEA": null,
      "ST": "N"
     }
    ]
   },
   "02220160513163236": {
    "cabecera": {
     "mpk": "02220160513163236",
     "ID_COBRANZA": "02220160513163236",
     "CL_CODIGO": "04461",
     "FEC_INI_COBRANZA": "2016-05-13",
     "HORA_INI_COBRANZA": "16:32:36",
     "FEC_FIN_COBRANZA": "2016-05-13",
     "HORA_FIN_COBRANZA": "16:32:36",
     "MONTOTOTAL": "30.14",
     "NRO_RECIBO": "99999999",
     "VEN_CODIGO": "022",
     "ESTADO": "0",
     "ST": "N"
    },
    "documentosimputados": [
     {
      "mpk": "02220160513163236000060946",
      "ID_COBRANZA": "02220160513163236",
      "CO_CODIGO": "FAC",
      "CO_SUCURS": "0001",
      "CO_NUMERO": "00060946",
      "CO_TIPO": "A",
      "MONTO": "30.14",
      "CUOTA": "01",
      "HH_LINEA": null,
      "ST": "N"
     }
    ],
    "detallepagos": [
     {
      "mpk": "022201605131632360",
      "ID_COBRANZA": "02220160513163236",
      "NUMERO": "0",
      "ID_MEDIOPAGO": "$",
      "CODIGOBANCO": " ",
      "PLAZA": " ",
      "FECHA": "0000-00-00",
      "FECVEN": "0000-00-00",
      "COTIZACION": "1.00",
      "MONTO": "30.14",
      "HH_LINEA": null,
      "ST": "N"
     }
    ]
   }
}
```


Explico el despliegue:
Los sub elementos del Json son las cobranzas, por lo que el elemento cero es la primer cobranza
La clave del elemento es el código de la cobranza, en este caso 02220160513163039
Los sub elementos de una cobranza, son “cabecera”,“documentosimputados” y “detallepagos”
En cabecera de la cobranza están los datos de cobranza, como cliente, vendedor, monto total, etc
En documentosimputados, los sub elementos son los comprobantes que se cobraron en ese movimiento
En detallepagos, los sub elementos son las formas de pago que se utilizaron para cancelar el monto, efectivo, cheques, etc

---

Un ejemplo para obtener todos los motivos de no compra nuevos del día 13 de mayo del 2016:

La url quedaría compuesta de la siguiente manera:

http://104.131.172.142/chhapi/

los parámetros:

```
{
        action: 'verNoCompra',
        token: '73b1f30a5879bc43c18e8f8cb5ab2f78',
        st: 'N',
        timeStart: '2016-05-13',
        timeEnd: '2016-05-13'
}
```

Lo que da como resultado


```
{
  "056022016-05-13022": {
    "cabecera": {
      "mpk": "056022016-05-13022",
      "CL_CODIGO": "05602",
      "CLM_FECHA": "2016-05-13",
      "MNC_CODIGO": "02",
      "VEN_CODIGO": "022",
      "CLM_HORA": "16:33:15",
      "MNC_OBSERV": "",
      "ST": "N"
    }
  },
  "035522016-05-13022": {
    "cabecera": {
      "mpk": "035522016-05-13022",
      "CL_CODIGO": "03552",
      "CLM_FECHA": "2016-05-13",
      "MNC_CODIGO": "01",
      "VEN_CODIGO": "022",
      "CLM_HORA": "16:33:07",
      "MNC_OBSERV": "",
      "ST": "N"
    }
  }
}
```


Explico el despliegue:
Los sub elementos del Json son las no compras, por lo que el elemento cero es la primer no compra
La clave del elemento es el código de la no compra, en este caso 056022016-05-13022
El único sub elemento, es "cabecera"
En cabecera de la no compra están los datos de la no compra, como cliente, vendedor, motivo, etc

---

Un ejemplo de la llamada necesaria para cambiar el estado a un elemento

La url quedaría compuesta de la siguiente manera:

http://104.131.172.142/chhapi/

los parámetros:
{
	action: ’cambiarEstadoRegistro’,
	registro: 'pedidos'
	token: ’73b1f30a5879bc43c18e8f8cb5ab2f78’,
	st: ’O’,
	mpk: ['02220160513111442’,'02220160513162856’]
}

Una vez descargados los pedidos con estado N, es necesario pasarlos a O, para que la llamada a los pedidos nuevos no vuelva a encontrar los mismos pedidos.
En este ejemplo, se cambian los estados de dos pedidos.
en el campo 'mpk’ se envía una lista de claves, se solicita un array para hacer muchos cambios en una sola llamada, ya que la api no permite recursividad de llamadas.
en registro, se envía el tipo de registro que se va a modificar, las opciones son, ‘pedido’, ‘cobranza’ y ‘noCompra’
en st, se envía el estado al que se quiere pasar ese registro, las opciones son ’N’ y ‘O’

———————————————————————————————————————

Un ejemplo para subir datos (Clientes, Articulos, etc.) a la nube a través de la api, en este ejemplo se envían dos registros de la tabla clientes, y dos registros de la tabla artículos.

La url quedaría compuesta de la siguiente manera:

http://104.131.172.142/chhapi/

los parámetros:
{
	action: ’cargarRegistros’,
	token: ’73b1f30a5879bc43c18e8f8cb5ab2f78’,
	datos: {
		"articulos": [{
			"mpk": "9210040101",
			"AR_CODIGO": "2552",
			"AR_DESCRI": "Aceite Natura 6x3000 cc",
			"FA_CODIGO": "001001002",
			"FA_NOMBRE": "aceite girasol",
			"UM_CODIGO": "UN",
			"UM_DESCRI": "UNIDAD",
			"UM_CODCOM": "BU",
			"AR_EQUUM": "6.00",
			"MA_CODIGO": "182",
			"MA_DESCRI": "Aceitera Gral.Deheza",
			"DE_CODIGO": "01",
			"DE_DESCRI": "DEPOSITO ESTECO",
			"ST_STOCK": "1342.00",
			"LP_CODIGO": "01",
			"LP_DESCRI": "LISTA DISTRIBUCION",
			"AP_PRECIO": "13.46",
			"ALI_PORCEN": "21.00",
			"II_PORCEN": "0.00",
			"AR_COSNET": "13.46",
			"ST": "30"
		}, {
			"mpk": "9210050101",
			"AR_CODIGO": "2553",
			"AR_DESCRI": "Aceite Natura 12x1500 cc",
			"FA_CODIGO": "001001002",
			"FA_NOMBRE": "aceite girasol",
			"UM_CODIGO": "UN",
			"UM_DESCRI": "UNIDAD",
			"UM_CODCOM": "BU",

```
                        "AR_EQUUM": "12.00",
                        "MA_CODIGO": "182",
                        "MA_DESCRI": "Aceitera Gral.Deheza",
                        "DE_CODIGO": "01",
                        "DE_DESCRI": "DEPOSITO ESTECO",
                        "ST_STOCK": "0.00",
                        "LP_CODIGO": "01",
                        "LP_DESCRI": "LISTA DISTRIBUCION",
                        "AP_PRECIO": "6.73",
                        "ALI_PORCEN": "21.00",
                        "II_PORCEN": "0.00",
                        "AR_COSNET": "5.31",
                        "ST": "30"
                }],
                "clientes": [{
                        "mpk": "01026",
                        "CL_CODIGO": "01026",
                        "CL_NOMBRE": "Francisco Calle Rodriguez",
                        "CL_DIRECC": "AV. SOTOMAYOR 126",
                        "CL_EMAIL": "franciscocalle@hotmail.com",
                        "CL_CODPOS": "5000",
                        "CL_LOCALI": "CORDOBA",
                        "PO_CODIGO": "05",
                        "VEN_CODIGO": "031",
                        "CL_DOCUME": "16320812",
                        "CL_CUIT": "20163208121",
                        "CL_IVA": "9",
                        "COI_DESCRI": "Sujeto No Categorizado",
                        "LP_CODIGO": "01",
                        "CVE_CODIGO": "001",
                        "ST": "30"
                }, {
                        "mpk": "01029",
                        “CL_CODIGO_X": "01029",
                        "CL_NOMBRE": "Ana Maria Castro Arellano",
                        "CL_DIRECC": "CALLE LAS HORQUIDEAS 145",
                        "CL_EMAIL": "ana mariacastro@hotmail.com",
                        "CL_CODPOS": "5000",
                        "CL_LOCALI": "CORDOBA",
                        "PO_CODIGO": "05",
                        "VEN_CODIGO": "022",
                        "CL_DOCUME": "16329520",
                        "CL_CUIT": "20163295201",
                        "CL_IVA": "9",
                        "COI_DESCRI": "Sujeto No Categorizado",
                        "LP_CODIGO": "01",
                        "CVE_CODIGO": "002",
                        "ST": "30"
                }]
        }
}
```


Lo que da como resultado:


```
{
        “success":[{
                “mpk":"9210040101",
                “tabla":"articulos"
        },{
                “mpk”:"9210050101",
                "tabla":"articulos"
        },{
                “mpk”:"01026",
                "tabla":"clientes"
        },{
                “mpk”:"01029",
```

```
                "tabla":"clientes"
        }],
        “errors":[{
                “mpk":"01029",
                “campos”: [ “mpk”, “CL_CODIGO_X", “CL_NOMBRE", “CL_DIRECC", “CL_EMAIL", “CL_CODPOS", “CL_LOCALI", “PO_CODIGO", “VEN_CODIGO", “CL_DOCUME", “CL_CUIT", “CL_IVA", “COI_DESCRI", “LP_CODIGO", “CVE_CODIGO", “ST" ],
                “valores": [ “01029”, “01029", "Ana Maria Castro Arellano”, "CALLE LAS HORQUIDEAS 145”, "ana mariacastro@hotmail.com”, “5000", “CORDOBA", “05", “022", “16329520", “20163295201", “9", "Sujeto No Categorizado”, “01", “002", “30” ],
                “tabla”:”clientes"
        }]
}
```


En la llamada, los sub elementos del objeto “datos”, son las tablas, y el sub elemento de cada tabla es una lista de jsons, compuesta por los campos y los valores de los registros.

Explico el despliegue:
Los sub elementos del Json son los resultados correctos, y los resultados erróneos, detallados en los sub elementos “success” y “errors”.
Cada uno de estos sub elementos contiene:
success: contiene una lista de objetos resultantes correctos de las operaciones, cada objeto compuesto por los campos “mpk” en donde se encuentra el valor clave del registro enviado, y “tabla” que contiene el nombre de la tabla de origen del registro.
errors: contiene una lista de objetos resultantes erróneos de las operaciones, cada objeto compuesto por los campos “mpk” en donde se encuentra el valor clave del registro enviado, “campos” con una lista conformada por todos los campos enviados, “valores” conformada por todos los valores enviados, y “tabla” que contiene el nombre de la tabla de origen del registro. **En el caso de que la tabla sea incorrecta, en el error solo vuelve un campo indicando cual es la tabla que produjo el error**.
Puede notarse que en los parámetros de la llamada, dentro de la tabla de clientes, hay un campo “CL_CODIGO_X”, que es incorrecto, en la respuesta de la llamada, dentro del sub elemento “errors” están detallados los datos del registro que provocó el error.

——

Un ejemplo para borrar datos de la nube a través de la api, en este ejemplo se envían dos registros de la tabla clientes, y dos registros de la tabla artículos.

La url quedaría compuesta de la siguiente manera:

http://104.131.172.142/chhapi/

los parámetros:

```
{
        action: ’borrarRegistros’,
        token: ’73b1f30a5879bc43c18e8f8cb5ab2f78’,
        datosBorrar: {
                "articulos": ["9210040101","9210050101"],
                "clientes": [“01026”,"01029"]
        }
}
```

Lo que da como resultado:


```
{
        “success":[{
                “mpk":"9210040101",
                “tabla":"articulos"
        },{
                “mpk”:"9210050101",
                "tabla":"articulos"
        },{
                “mpk”:"01026",
                "tabla":"clientes"
```

```
        },{
                “mpk”:"01029",
                "tabla":"clientes"
        }],
        “errors":[{
                “clave":"92100401012","tabla":"articulos"
        }]
}
```


En la llamada, el sub elemento “datosBorrar” contiene los nombres de las tablas, y cada tabla, contiene una lista [] con las claves de los elementos a borrar.

Explico el despliegue:
Los sub elementos del Json son los resultados correctos, y los resultados erróneos, detallados en los sub elementos “success” y “errors”.
Cada uno de estos sub elementos contiene:
success: contiene una lista de objetos resultantes correctos de las operaciones, cada objeto compuesto por los campos “mpk” en donde se encuentra el valor clave del registro enviado, y “tabla” que contiene el nombre de la tabla de origen del registro.
errors: contiene una lista de objetos resultantes erróneos de las operaciones, cada objeto compuesto por los campos “mpk” en donde se encuentra el valor clave del registro enviado, “campos” con una lista conformada por todos los campos enviados, “valores” conformada por todos los valores enviados, y “tabla” que contiene el nombre de la tabla de origen del registro. **En el caso de que la tabla sea incorrecta, en el error solo vuelve un campo indicando cual es la tabla que produjo el error**.

———————————————————————————————————

Y por último, la api tiene una acción destinada a devolver todos los datos de las tablas.

La url quedaría compuesta de la siguiente manera:

http://104.131.172.142/chhapi/

los parámetros:


```
{
        action: ’verDatos’,
        token: ‘73b1f30a5879bc43c18e8f8cb5ab2f78’,
        tabla: ‘articulos’
}
```


El resultado de esa llamada es un json que contiene todos los datos de la tabla solicitada.

———

———

Detalle de los campos de cada tabla:

ARTICULOS

- mpk, clave del registro, resultado de concatenar AR_CODIGO, LP_CODIGO, DE_CODIGO
- AR_CODIGO, código del artículo
- AR_DESCRI, descripción del artículo
- FA_CODIGO, código de familia
- FA_NOMBRE, nombre de familia
- UM_CODIGO, código de unidad de medida
- UM_DESCRI, descripción de unidad de medida
- AR_EQUUM, equivalencia de unidad de medida
- MA_CODIGO, código de marca
- MA_DESCRI, descripción de marca
- DE_CODIGO, código de depósito
- DE_DESCRI, descripción de depósito
- ST_STOCK, cantidad disponible
- LP_CODIGO, código de lista de precio

- LP_DESCRI, descripción de lista de precio
- AP_PRECIO, precio del artículo, sin iva y sin impuestos internos
- ALI_PORCEN, porcentaje de iva
- II_PORCEN, porcentaje de impuesto interno
- AR_COSNET, costo neto del artículo

CLIENTES
- mpk, clave del registro, en el caso de esta tabla tiene el mismo valor que CL_CODIGO
- CL_CODIGO, código de cliente
- CL_NOMBRE, nombre de cliente
- CL_DIRECC, dirección del cliente (una sola)
- CL_EMAIL, email del cliente (uno solo)
- CL_CODPOS, código postal del cliente
- CL_LOCALI, localidad del cliente
- VEN_CODIGO, código de vendedor al que está asignado el cliente
- CL_DOCUME, dni del cliente
- CL_CUIT, cuit del cliente
- CL_IVA, código de condición de iva del cliente
- COI_DESCRI, descripción de condición de iva del cliente
- LP_CODIGO, código de lista de de precios que tiene asignado el cliente
- CVE_CODIGO, código de condición de venta por defecto para el cliente

CLIENTES_TELEFONO (Teléfonos de los clientes) (Si el sistema no maneja los teléfonos de los clientes, podemos prescindir de esta tabla)
- mpk, clave del registro, resultado de concatenar mpk,CL_CODIGO,TT_CODIGO
- CL_CODIGO, código de cliente
- TT_CODIGO, código de tipo de teléfono
- CLT_NUMERO, número de teléfono

COND_VENTA (Condiciones de venta)
- mpk, clave del registro, en el caso de esta tabla es el mismo valor que CVE_CODIGO
- CVE_CODIGO, código de condición de venta
- CVE_DESCRI, descripción de la condición de venta

CTACTE (Cuenta Corriente) (Si el sistema no maneja cuenta corriente, podemos prescindir de esta tabla)
- mpk, clave del registro, resultado de concatenar CO_CODIGO, CO_SUCURS, CO_NUMERO, CO_TIPO, COC_NUMERO
- CL_CODIGO, código de cliente
- CL_NOMBRE, nombre de cliente
- CO_CODIGO, código de comprobante (ej: FAC para factura comprobante)
- CO_SUCURS, código de sucursal (ej: 0001)
- CO_NUMERO, número de comprobante (ej: 00000001)
- CO_TIPO, tipo de comprobante (ej: A, B)
- COC_NUMERO, numero de cuota del comprobante
- VEN_CODIGO, código de vendedor
- CO_FECHA, fecha del comprobante
- CO_FECVEN, fecha de vencimiento del comprobante
- COC_IMPENT, importe entregado en caso de que exista un pago parcial
- COC_IMPORT, importe original del comprobante
- SALDO, resta del importe original menos el importe entregado

MOTIVO_NO_COMPRA (Motivos por los que un cliente puede no comprar, ej: no tiene dinero, no esta el encargado, local cerrado, etc)
- mpk, clave del registro, en el caso de esta tabla, es el mismo valor que MNC_CODIGO
- MNC_CODIGO, código de motivo de no compra
- MNC_DESCRI, descripción de motivo de no compra

RUTAS (Clientes para un vendedor en un día y orden determinado) (Si el sistema no maneja hojas de ruta, podemos prescindir de esta tabla)
- mpk, clave del registro, resultado de concatenar HR_CODIGO, HRD_ORDEN, CL_CODIGO
- HR_CODIGO, código de hoja de ruta
- HRD_ORDEN, órden de visita
- CL_CODIGO, código de cliente
- VEN_CODIGO, código de vendedor
- DIA_CODIGO, día de visita empezando por el 1 para el domingo, 2 lunes … 6 sábado

VENDEDOR

- mpk, en el caso de esta tabla, es el mismo valor que VEN_CODIGO
- PER_CODIGO, código de vendedor
- VEN_CODIGO, código de vendedor
- PER_NOMBRE, nombre de vendedor
- VEN_PASSWORD, es una contraseña numérica para que los vendedores accedan a la aplicación móvil, en caso de no usarla, en este campo poner siempre el número 0

Nuestra sugerencia en cuanto al manejo de datos.

Lo óptimo sería subir las cosas que todavía no se subieron (no todo), las cosas nuevas, y las cosas que cambiaron, ambas en el momento que se producen, por ejemplo, se agrega un artículo y se sube, se modifica un artículo y se sube, eso evitaría tener que subir todos los días todos los artículos, y haría que si hay subidas, sean pocos registros y consuman pocos recursos.
En cuanto al borrado de registros, siguiendo las mismas líneas, deberían borrarse solo los que se dan de baja y en el momento en que se dan de baja, eso evitaría que todos los días se hagan todas las llamadas intentando borrar artículos que probablemente no estén.

Sugerimos tres caminos:

1) En el sistema, en las funciones que dan de alta, modifican, o desactivan registros, agregar la función correspondiente para hacer la llamada a la api, además habría que llevar un registro en alguna tabla para saber que cosas ya se subieron o se borraron, y una función extra que se ejecute manualmente para poder forzar una actualización.

2) El manejo que hacemos nosotros uno de nuestros servicios de sincronización “datasender” resultó bastante eficiente, se generan un tablas espejo que tengan los registros con una columna extra que indica el estado, en donde se definen internamente el significado de esos estados, por ejemplo 0 nuevo, 1 actualizado, 2 eliminado, 3 sincronizado. Entonces el sistema actualiza esa tabla (una vez al dia, cada media hora, o como sea mejor), y un servicio mantiene esa tabla sincronizada con la nube, entonces busca los registros con estado 0 (nuevos), hace la llamada para agregar el registro y si la respuesta es correcta el estado se cambia a 3 (sincronizado), lo mismo para los estados 1 (actualizados), y para los estados 2 (eliminados) hace la llamada de borrar y en vez de cambiar el estado a 3 directamente elimina ese registro. De manera que quedarían esa tabla con estado 0 todos los artículos que fueron correctamente sincronizados con la api.

3) La api cuenta con una llamada que devuelve todos los registros que están en la nube para una tabla solicitada, al obtener los datos se puede comparar con el sistema los 3 estados de los registros por cada registro:

- Un registro que esta en la nube y también esta entre los registros del sistema debería ser comparado para ver si es igual y chequear si esta igual, o algo cambió, en caso de que algo haya cambiado, se hace la actualización de ese registro a través de la api.
- Un registro que esta en la nube, pero no esta entre los registros del sistema, tiene que ser eliminado a través de la api.
- Un registro que esta entre los registros del sistema, pero no esta en la nube, tiene que ser agregado a través de la api.