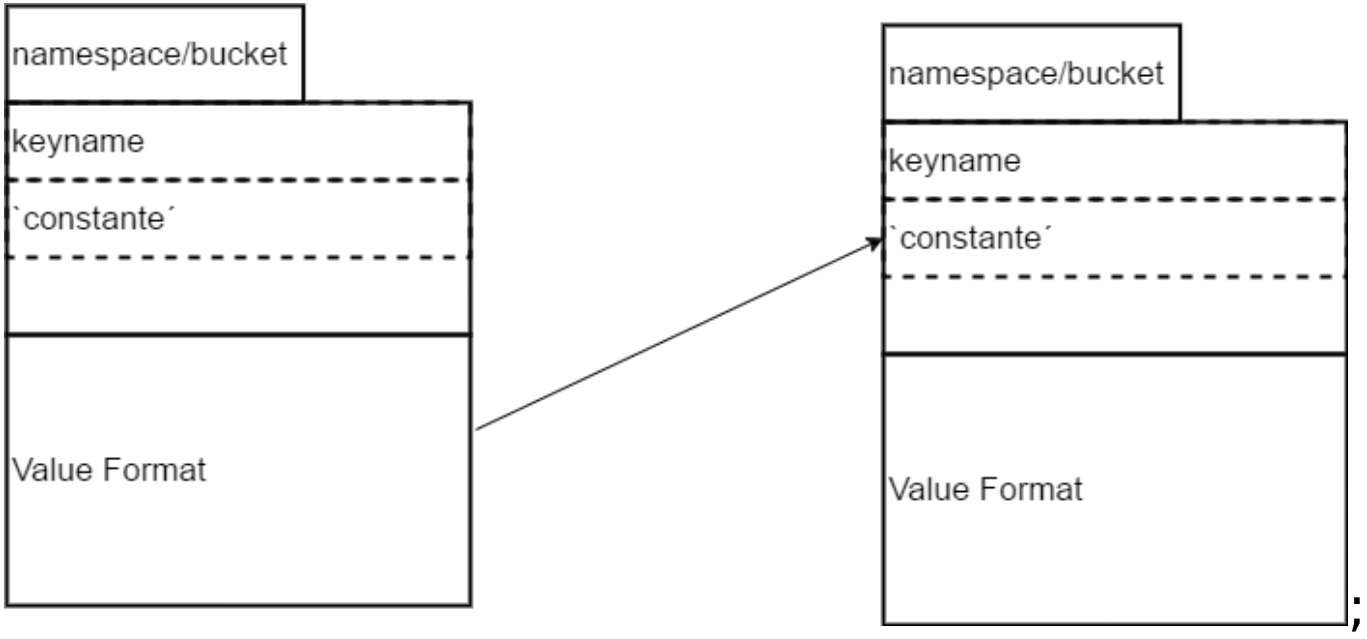
Key-Value

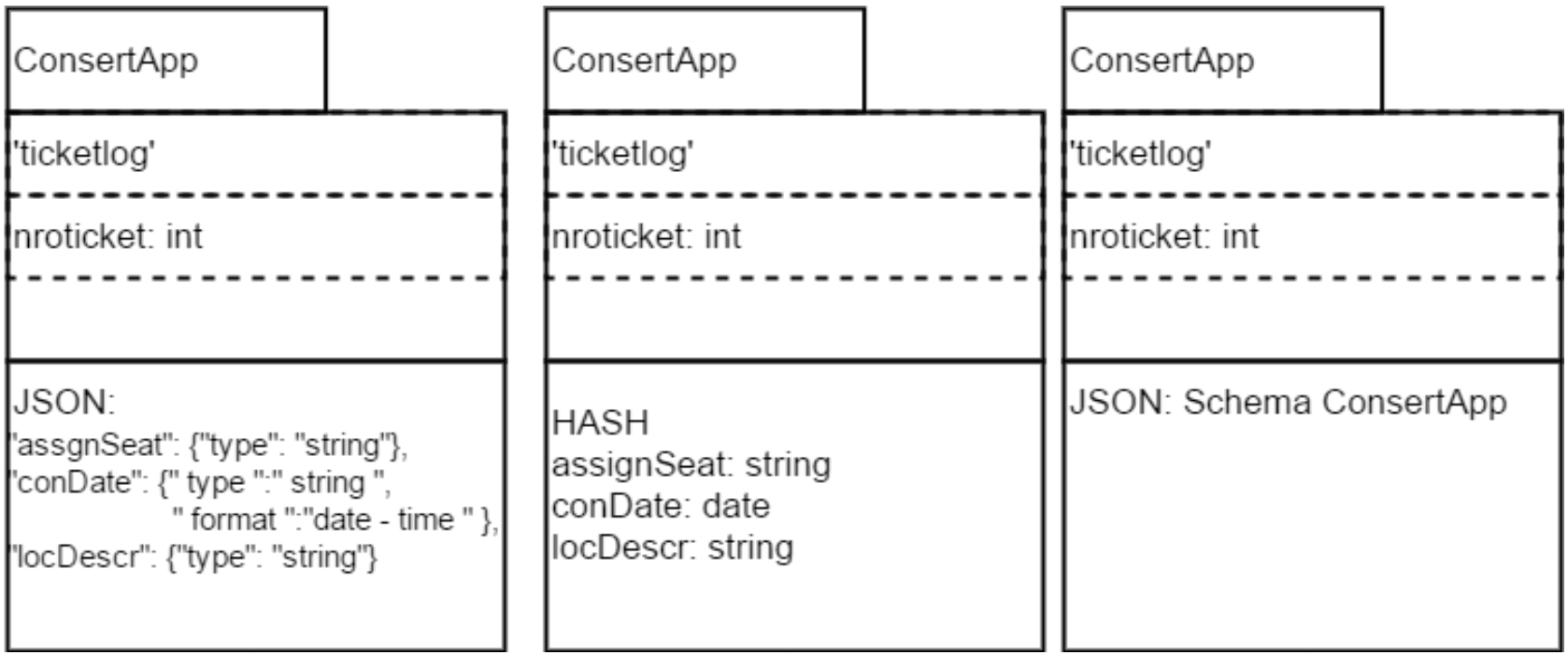
Notación Key-Value

- Espacio de nombres optativo
- Partes de la clave
- Valor en formatos JSON, Lista, Hash, string, etc.
- Relación entre un valor y una clave de otro dato.

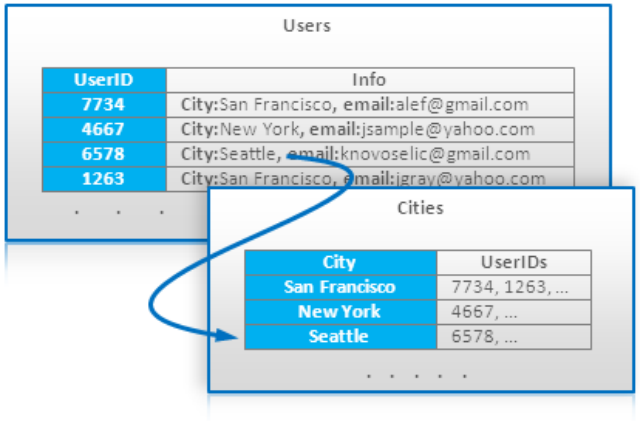


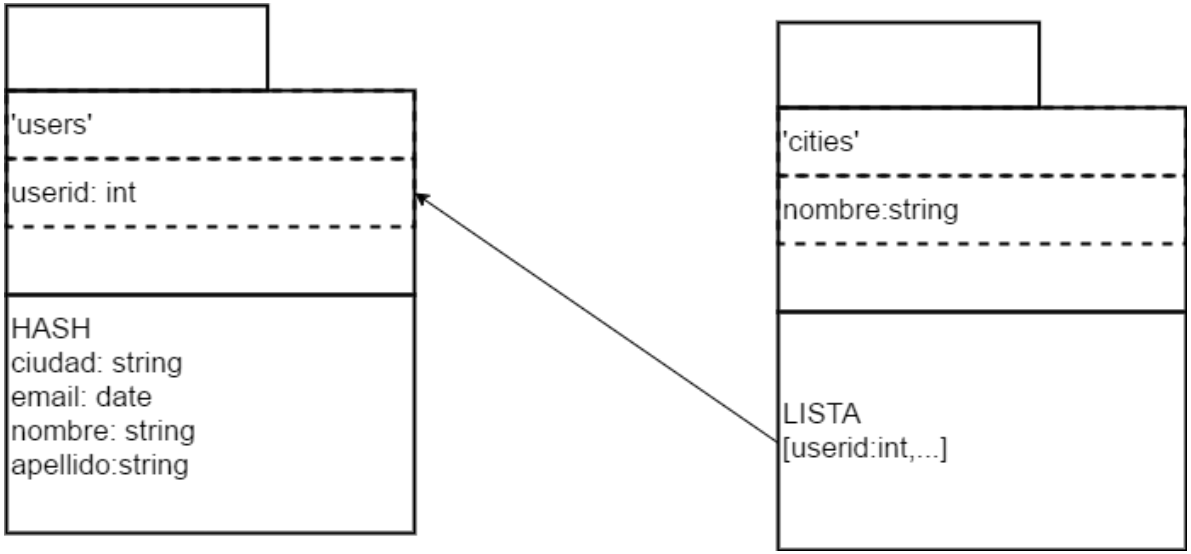
Notación Key-Value Ejemplo

```
ConcertApp[ticketLog:9888] = {"conDate":15-Mar-2015, "locDescr": "Springfield Civic Center",
      "assgnSeat": "J38"}
```



Notación Key-Value Ejemplo referencias





Ejercicio

Indices

Diseñar la base de datos para un twitter

- Se pueden usar como tipos de datos: Json, Hash, Sets
  - **HSET** usuarios nombre Jhon apellido Doe
  - **HGET** usuarios nombre $\Longrightarrow$ *Jhon*
- Se puede usar una operación: **INCR key**. (Clave tipo INCR en el diagrama)
  - **INCR** prox_id $\Longrightarrow$ 10

Familia de Columnas

Diseño Aspectos Generales

Cómo diseñar

Establecer los patrones de escritura y lectura.

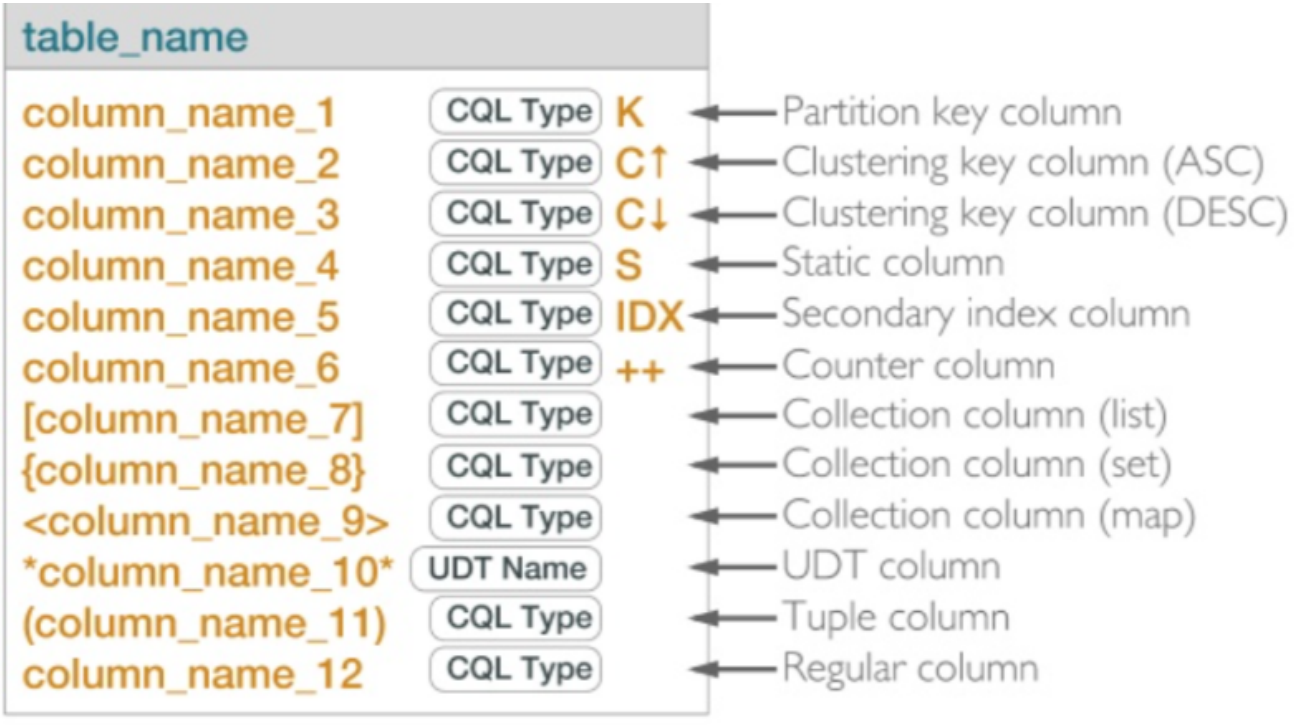
- Desnormalizar!
- Usar columnas sin valor
- Usar nombres y valores de columna para almacenar datos
- Modelar una entidad como una fila simple
- Mantener un número adecuado de versiones en los valores de las columnas.
  - HBase permite especificar el mínimo y máximo número de versiones.
- Evitar estructuras complejas en los valores de las columnas.

Reglas de Mapeo

Basados en los DMP (Data Modeling Principles), las reglas de mapeo ayudan a realizar la transición desde el modelo conceptual al modelo lógico.

1. **MR1** (*Entities and Relationships*): Los tipos de entidades y relaciones mapean a tablas mientras que los datos se asignan a filas. Los atributos de las entidades y las relaciones se mapean a columnas
2. **MR2** (*Equality Search Attributes*): Si se utilizan en una consulta por igualdad de atributos, entonces, éstos forman la clave primaria y tales columnas se incluirán en las particiones por clave
3. **MR3** (*Inequality Search Attributes*): Si se utilizan en consultas por desigualdad, estos atributos mapean como columnas dinámicas, conteniendo los valores
4. **MR4** (*Ordering Attributes*): Mapea a columna dinámica con orden ascendente o descendente según se especifique en la consulta
5. **MR5**(*Key Attributes*): Mapea a clave primaria. Una tabla que almacena datos de entidades o relaciones como filas debe incluir atributos claves que identifique estos datos univocamente

Cassandra - Notación/Método Chebotko



Ejemplo: DER

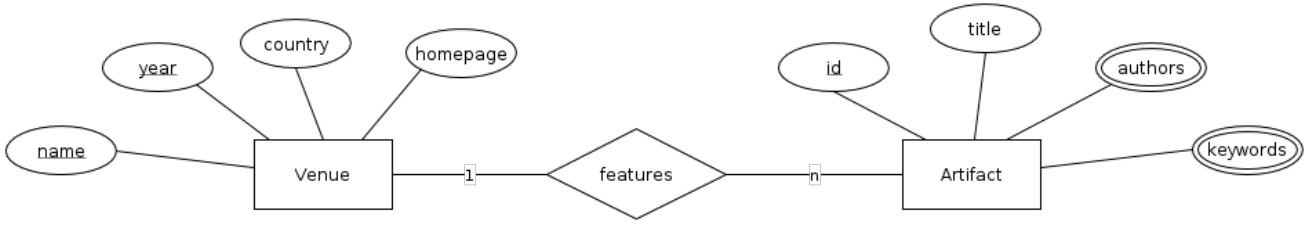
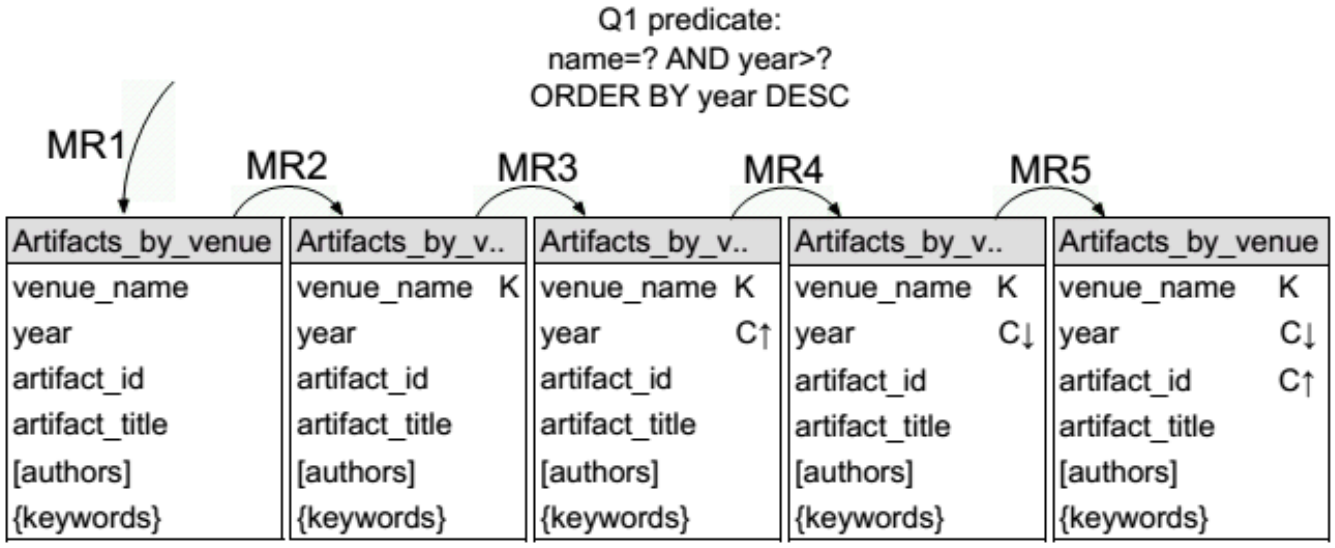


Diagrama de Entidad-Relación

- Q1: Obtener los artefactos publicados en *venue* con un nombre dado después de un año dado. El resultado debe estar ordenado en forma descendente



Map-Reduce

1. Entrada: un conjunto de pares clave-valor. De esta manera se logra permitir la composición de procesos
2. El desarrollador especifica dos funciones:
   - $Map(k, v) \rightarrow < k', v' >^*$
     - Toma un par *key-value* y devuelve un conjunto de pares *key-value*
     - Hay un llamada a Map por cada par (k,v)
   - $Reduce(k', < v' >^*) \rightarrow < k'', v'' >^*$
     - Todos los valores de v' para la misma k' se reducen y procesan en conjunto

Ejemplo - Contar Palabras

```
map(String input_key, String input_value):
// input_key: document name
// input_value: document contents
for each word w in input_value:
EmitIntermediate(w, 1);
```

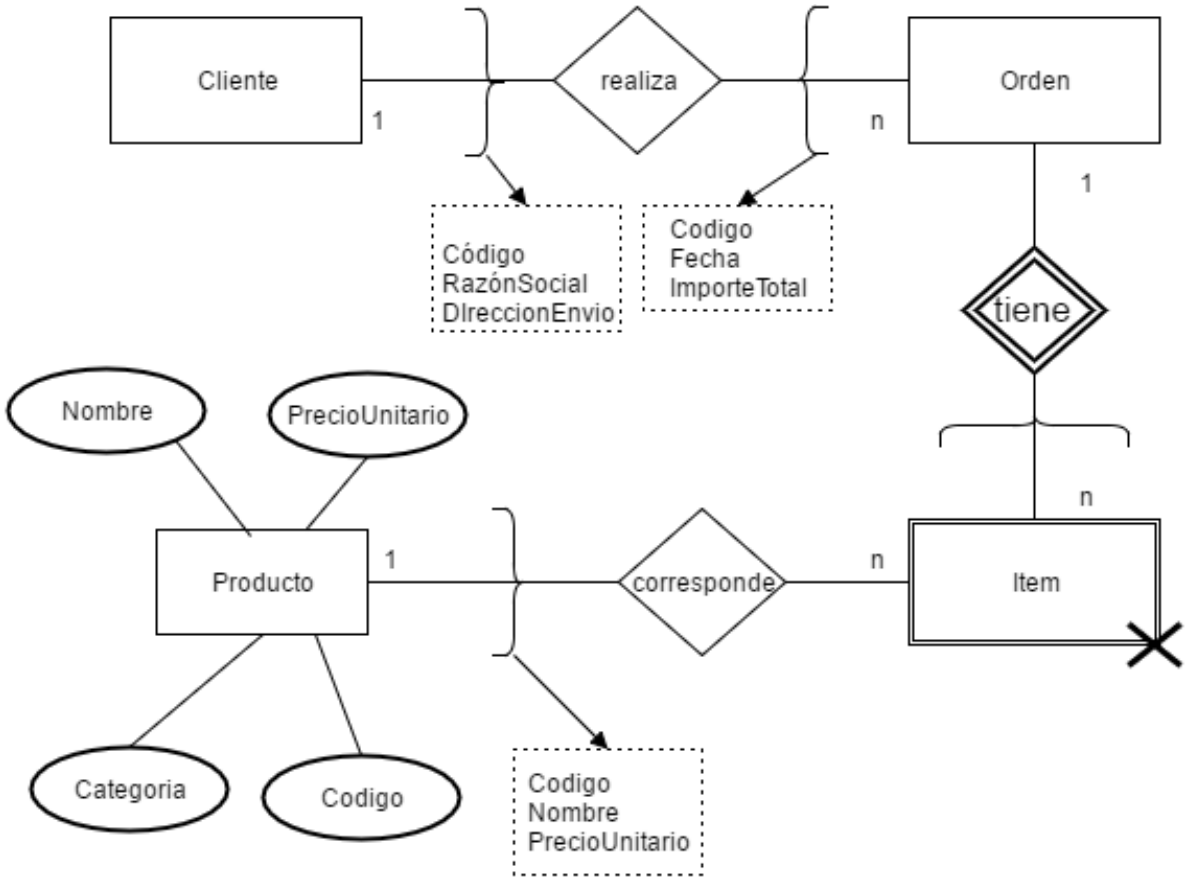
Map

```
reduce(String output_key, Iterator intermediate_values):
// output_key: word
// output_values: ????
int result = 0;
for each v in intermediate_values:
result += v;
Emit(result);
```

Reduce

Document Database

Desnormalización parcial

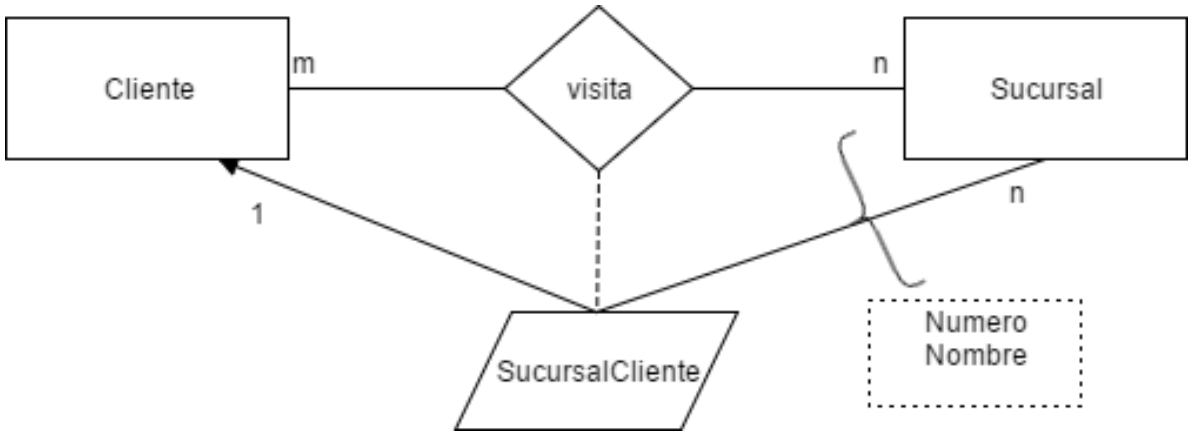


Desnormalización

JSON Scheme para Documento Orden

```
"Orden": {"type": "object",
   "properties": {
     "OrdenID": {"type":"integer" },
     "Fecha":{"type":"string","format":"date-time"},
     "ImporteTotal":{"type":"integer"},
     "Cliente": {
           "type":"object",
           "properties":{
               "Codigo":{"type":"integer"},
               "RazonSocial":{"type":"string"},
               "DireccionEnvio":{"type":"string"}
       }
     },
     "ItemsOrden":{
          "type":"Array" ,
          "items":
            {"type":"object",
             "properties":{
                 "Cantidad":{"type":"integer"},
                 "Codigo":{"type":"integer" },
                 "Nombre":{"type":"string" },
                 "PrecioUnitario":{" type ":" string " }}
            }
      }
   }
 }
```

Uso de Documentos Auxiliares




```
"SucursalCliente": {"type": "object",
          "properties": {
             "ClienteID": {"type":"integer" },
             "Sucursales":{
               "type":"Array",
               "items":{"type":"object",
                   "properties":{
                     "Codigo":{"type":"integer" },
                     "Nombre":{"type":"string" },
                 }
              }
            }
   }
```