

Entidades	Operaciones	Information Needed	Tipo	Rate
Tipo de Habitación	Registrar/Actualizar/Borrar	Nombre, dotación, precio por noche	Escritura	1/año
Tipo de habitación	Consultar detalles	Nombre, dotación, precio por noche	Lectura	100/hora
Habitaciones	Registrar/Actualizar/Borrar	# de Hab, tipo de habitación	Escritura	1/mes
Habitaciones	Consultar detalles	# de Hab, tipo de habitación	Lectura	100/hora
Servicios	Registrar/Actualizar/Borrar	Id, tipo, nombre	Escritura	2/mes
Servicios	Consultar detalles	Id, tipo, nombre	Lectura	100/hora
Reservas Habitación	Registrar/Actualizar/Borrar	Id, usuario, costo, plan, fechas	Escritura	12/día
Reservas Habitación	Consultar detalles	Id, usuario, costo, plan, fechas	Lectura	12/día
Check in	Registrar/Actualizar/Borrar	Usuario, reserva, habitación, fecha, costo	Escritura	12/día
Check in	Consultar detalles	Usuario, reserva, habitación, fecha, costo	Lectura	12/día
Reservas Servicio	Registrar/Actualizar/Borrar	Habitación, usuario, fecha, costo	Escritura	1/hora
Reservas Servicio	Consultar detalles	Habitación, usuario, fecha, costo	Lectura	1/hora
Check Out	Registrar/Actualizar/Borrar	Usuario, reserva, habitación, fecha, costo, cuenta	Escritura	12/día
Check Out	Consultar detalles	Usuario, reserva, habitación, fecha, costo, cuenta	Lectura	12/día
Reservas de servicios y servicios	Dinero recolectado por servicios en el último año	Fechas, costo, nombre de servicio	Lectura	1/mes
Reservas de habitación	Mostrar índice de ocupación dentro del último año	Índice de ocupación, fecha, # de habitaciones	Lectura	1/mes
Cliente y reservas servicios	Consumo de un cliente en un rango de fechas	Cliente, fecha, reservas de servicios	Lectura	1/mes
Cientes, reservas servicios	Consumo de un servicio	Cientes, fechas, nombre del servicio, atributo de agrupamiento	Lectura	1/mes

Selección de Esquema de Asociación:

Justificación habitaciones con tipos de habitación:

	Pregunta	Embeber	Referenciar
Simplicity	Would keeping the pieces of information together lead to a simpler data model and code?	1	0
Go together	Do the pieces of information have a “has-a”, “cointains”, or a similar relationship?	2	0
Query Atomicity	Does the application query the pieces of information together?	3	0
Update Complexity	Are the pieces of informatioon updated together?	4	0
Archival	Should the pieces of information be archived at the same time?	0	1
Cardinality	Is there a high cardinality (current or growing) in the child side of the relationship?	0	2
Data Duplication	Would data duplication bee too complicated to manage and undesired?	5	0
Document Size	Would the combined size of the pieces of information take too much memory or transfer bandwidth for the application?	0	3
Document Growth	Would the embedded piece grow without bound?	0	4
Workload	Are the pieces of information written at different times in a write-heavy workload?	0	5
Individuality	For the children side of the reelationship, can the pieces exist by themselves without a parent?	6	0

Justificación reservas habitaciones con planes de consumo:

	Pregunta	Embeber	Referenciar
Simplicity	Would keeping the pieces of information together lead to a simpler data model and code?	1	0
Go together	Do the pieces of information have a “has-a”, “cointains”, or a similar relationship?	2	0
Query Atomicity	Does the application query the pieces of information together?	3	0
Update Complexity	Are the pieces of informatioon updated together?	4	0
Archival	Should the pieces of information be archived at the same time?	0	1
Cardinality	Is there a high cardinality (current or growing) in the child side of the relationship?	0	2
Data Duplication	Would data duplication bee too complicated to manage and undesired?	5	0

Document Size	Would the combined size of the pieces of information take too much memory or transfer bandwidth for the application?	0	3
Document Growth	Would the embedded piece grow without bound?	0	4
Workload	Are the pieces of information written at different times in a write-heavy workload?	0	5
Individuality	For the children side of the relationship, can the pieces exist by themselves without a parent?	6	0

Justificación de asociación entre servicios con reservas servicios:

	Pregunta	Embeber	Referenciar
Simplicity	Would keeping the pieces of information together lead to a simpler data model and code?	1	0
Go together	Do the pieces of information have a “has-a”, “contains”, or a similar relationship?	2	0
Query Atomicity	Does the application query the pieces of information together?	3	0
Update Complexity	Are the pieces of information updated together?	0	1
Archival	Should the pieces of information be archived at the same time?	0	2
Cardinality	Is there a high cardinality (current or growing) in the child side of the relationship?	0	3
Data Duplication	Would data duplication be too complicated to manage and undesired?	4	0
Document Size	Would the combined size of the pieces of information take too much memory or transfer bandwidth for the application?	0	4
Document Growth	Would the embedded piece grow without bound?	0	5
Workload	Are the pieces of information written at different times in a write-heavy workload?	0	6
Individuality	For the children side of the relationship, can the pieces exist by themselves without a parent?	5	0

habitaciones		
_id: string tipo: <table> <tr> <th>tipos_habitacion</th></tr> <tr> <td> tipo: enum(string) costo_por_noche: int capacidad: int dotaciones: [string] </td></tr> </table> reservas: [reservas_habitaciones._id]	tipos_habitacion	tipo: enum(string) costo_por_noche: int capacidad: int dotaciones: [string]
tipos_habitacion		
tipo: enum(string) costo_por_noche: int capacidad: int dotaciones: [string]		

servicios
_id: objectId nombre: string : : atributos dependientes del tipo de servicio : : reservas: [reservas_servicios._id]

reservas_habitaciones		
_id: objectId id_habitacion: habitaciones._id usuario: { tipo_doc: usuarios.tipo_doc, num_doc: usuarios.num_doc } fecha_inicio: ISODate fecha_fin: ISODate num_personas: int plan_consumo: <table> <tr> <th>planes_consumo</th></tr> <tr> <td> nombre: enum(string) tasa: float min_noches: int max_noches: int </td></tr> </table> costo_reserva: float costo_consumos: float cuenta_abierta: bool fecha_check_in: ISODate fecha_check_out: ISODate	planes_consumo	nombre: enum(string) tasa: float min_noches: int max_noches: int
planes_consumo		
nombre: enum(string) tasa: float min_noches: int max_noches: int		

reservas_servicios
_id: objectId id_servicio: servicios._id id_reserva_habitacion: reservas_habitaciones._id usuario: { tipo_doc: usuarios.tipo_doc, num_doc: usuarios.num_doc } fecha: ISODate costo: float

usuarios
_id: objectId tipo_doc: string num_doc: int tipo: enum(string) nombre: string email: string password: string reservas_habitaciones: [reservas_habitaciones._id] reservas_servicios: [reservas_servicios._id]

Scripts de validación:

habitaciones:

```

1  {
2    $jsonSchema: {
3      bsonType: 'object',
4      required: [
5        '_id',
6        'tipo',
7        'reservas'
8      ],
9      properties: {
10       _id: {
11         bsonType: 'string'
12       },
13       tipo: {
14         bsonType: 'object',
15         required: [
16           'tipo',
17           'costo_por_noche',
18           'capacidad',
19           'dotaciones'
20         ],
21         properties: {
22           tipo: {
23             bsonType: 'string',
24             enum: [
25               'doble',
26               'familiar',
27               'sencilla',
28               'suite',
29               'suite presidencial'
30             ]
31           },
32           capacidad: {
33             bsonType: 'number',
34             minimum: 0
35           },
36           dotaciones: {
37             bsonType: 'array',
38             items: {
39               bsonType: 'string'
40             }
41           },
42           costo_por_noche: {
43             bsonType: 'number',
44             minimum: 0
45           }
46         }
47       },
48       reservas: {
49         bsonType: 'array',
50         items: {
51           bsonType: 'objectId'
52         }
53       }
54     }
55   }
56 }

```

reservas_habitaciones:

```

1  {
2    $jsonSchema: {
3      bsonType: 'object',
4      required: [
5        'id_habitacion',
6        'usuario',
7        'fecha_inicio',
8        'fecha_fin',
9        'num_personas',
10       'plan_consumo',
11       'costo_reserva',
12       'costo_consumos',
13       'cuenta_abierta'
14     ],
15     properties: {
16       id_habitacion: {
17         bsonType: 'string'
18       },
19       usuario: {
20         bsonType: 'object',
21         required: [
22           'tipo_doc',
23           'num_doc'
24         ],
25         properties: {
26           tipo_doc: {
27             bsonType: 'string'
28           },
29           num_doc: {
30             bsonType: 'number'
31           }
32         },
33       },
34       fecha_inicio: {
35         bsonType: 'date'
36       },
37       fecha_fin: {
38         bsonType: 'date'
39       },
40       num_personas: {
41         bsonType: 'number',
42         minimum: 0
43       },
44     },
45     plan_consumo: {
46       bsonType: 'object',
47       required: [
48         'nombre',
49         'tasa',
50         'min_noches',
51         'max_noches'
52       ],
53       properties: {
54         nombre: {
55           bsonType: 'string'
56         },
57         tasa: {
58           bsonType: 'double'
59         },
60         min_noches: {
61           bsonType: 'number',
62           minimum: 0
63         },
64         max_noches: {
65           bsonType: 'number'
66         }
67       },
68       costo_reserva: {
69         bsonType: 'number',
70         minimum: 0
71       },
72       costo_consumos: {
73         bsonType: 'number',
74         minimum: 0
75       },
76       cuenta_abierta: {
77         bsonType: 'bool'
78       },
79       fecha_check_in: {
80         bsonType: 'date'
81       },
82       fecha_check_out: {
83         bsonType: 'date'
84       }
85     }
86   }
87 }

```

reservas_servicios:

```

1 ▾ {
2 ▾   $jsonSchema: {
3     bsonType: 'object',
4 ▾   required: [
5     'id_servicio',
6     'id_reserva_habitacion',
7     'usuario',
8     'fecha',
9     'costo'
10   ],
11 ▾   properties: {
12 ▾     id_servicio: {
13       bsonType: 'objectId'
14     },
15 ▾     id_reserva_habitacion: {
16       bsonType: 'objectId'
17     },
18 ▾     usuario: {
19       bsonType: 'object',
20 ▾     required: [
21       'tipo_doc',
22       'num_doc'
23     ],
24 ▾     properties: {
25 ▾       tipo_doc: {
26         bsonType: 'string'
27       },
28 ▾       num_doc: {
29         bsonType: 'number'
30       }
31     }
32   },
33 ▾   fecha: {
34     bsonType: 'date'
35   },
36 ▾   costo: {
37     bsonType: 'number'
38   }
39 }
40 }
41 }

```

servicios:

```
1  {
2  $jsonSchema: {
3    bsonType: 'object',
4    required: [
5      'nombre',
6      'reservas'
7    ],
8    properties: {
9      nombre: {
10        bsonType: 'string'
11      },
12      reservas: {
13        bsonType: 'array',
14        items: {
15          bsonType: 'objectId'
16        }
17      }
18    }
19  }
20 }
```

usuarios:


```

1  ▾ {
2  ▾   $jsonSchema: {
3      bsonType: 'object',
4  ▾   required: [
5       'tipo_doc',
6       'num_doc',
7       'tipo',
8       'nombre',
9       'email',
10      'password',
11      'reservas_habitaciones',
12      'reservas_servicios'
13  ],
14  ▾   properties: {
15  ▾     tipo_doc: {
16         bsonType: 'string'
17     },
18  ▾     num_doc: {
19         bsonType: 'number'
20     },
21  ▾     tipo: {
22         bsonType: 'string',
23  ▾     'enum': [
24         'admin',
25         'cliente',
26         'empleado',
27         'gerente',
28         'recepcionista'
29     ]
30     },
31  ▾     nombre: {
32         bsonType: 'string'
33     },
34  ▾     email: {
35         bsonType: 'string'
36     },
37  ▾     password: {
38         bsonType: 'string'
39     },
40  ▾     reservas_habitaciones: {
41         bsonType: 'array',
42  ▾     items: {
43         bsonType: 'objectId'
44     }
45     },
46  ▾     reservas_servicios: {
47         bsonType: 'array',
48  ▾     items: {
49         bsonType: 'objectId'
50     }
51     }
52   }
53 }
54 }

```