

# Ejercicio Oracle Database

José Julián Barbosa Ayala  
Michael Hernández Vera  
Camilo Vega Ramírez

## Contenido

<b>Introducción.</b>	<b>2</b>
<b>Punto A</b>	<b>2</b>
Query . . . . .	2
Resultado JSON . . . . .	4
Resultado Tabular . . . . .	10
<b>Punto B</b>	<b>12</b>
Query . . . . .	12
Resultado JSON . . . . .	15
Resultado Tabular . . . . .	17
<b>Punto C</b>	<b>17</b>
Query . . . . .	17
Resultado JSON . . . . .	21
Resultado Tabular . . . . .	21

## Introducción.

A continuación, presentamos el desarrollo de la actividad de MongoDB. Para cada uno de los puntos, se muestra el código de consulta en MongoDB, el resultado en formato JSON, que se utiliza para la comunicación entre sistemas, y también se muestra el resultado en formato tabular, el cual ha sido transformado utilizando R para facilitar su lectura por parte de los humanos.

---

## Punto A

*Obtener la lista de los cursos ofrecidos en la carrera de **Ingeniería de Sistemas**. Se deben obtener los siguientes datos: nombre del curso, nombre completo del profesor que dicta el curso, el salón en que se dicta el curso y la hora en la que se dicta el curso.*

### Query

**NOTA:** Se entiende que la solicitud de “*cursos ofrecidos*” incluye todos los cursos listados por la carrera, independientemente de si tienen o no profesores u horarios asignados. Consideramos que el propósito de la consulta es revisar el estado de los cursos y, en caso de que alguno no tenga asignado un profesor o un horario, solucionar esta situación. Si se desea filtrar únicamente aquellos cursos con profesores y horarios asignados, se puede agregar al final de la consulta una instrucción \$match de la siguiente forma \&and: [{ profesor: { \&ne: null } }, { salon: { \&ne: null } }]. Igualmente, es importante tener en cuenta que hay dos cursos que aparecen duplicados debido a que tienen diferentes profesores y horarios: “Ética” y “Optimización Matemática”. Se decide no agruparlos con un \$group para mostrarlos como cursos individuales, ya que hay más cursos que cuentan con un solo profesor que aquellos que cuentan con dos. Por último, se puede observar que para todos los cursos de la colección “cursos” que tienen un calendario de dos o más días, tanto los edificios como los salones y horarios son los mismos. Por lo tanto, en la salida se decidió mostrar un solo salón con un horario único por curso, y se guardan los días en un array, con el fin de ahorrar espacio en la salida. Si en el futuro esta situación cambia, se puede modificar el “Stage 6 \$project” para eliminar esta unificación y mostrar el calendario tal como se recibe del “Stage 5”.

```
use university

var PuntoA = db.carreras.aggregate([
  {
    $match:
      /**
       * Stage 1:
       * Se filtra por la carrera de "Ingeniería de Sistemas"
       */
      {
        nombre: "Ingeniería de Sistemas",
      },
  },
  {
    $lookup:
      /**
       * Stage 2:
       * Unimos con la colección cursos por el id del curso.
       */
```

```

    {
      from: "cursos",
      localField: "cursos.curso",
      foreignField: "_id",
      as: "cursos",
    },
  ],
  {
    $unwind:
      /**
       * Stage 3:
       * Pasamos el array cursos a objeto.
       */
      {
        path: "$cursos",
        preserveNullAndEmptyArrays: true,
      },
  },
  {
    $lookup:
      /**
       * Stage 4:
       * Unimos con colección profesores por el id del profesor.
       */
      {
        from: "profesores",
        localField: "cursos.profesor",
        foreignField: "_id",
        as: "cursos.profesor",
      },
  },
  {
    $unwind:
      /**
       * Stage 5:
       * Pasamos el array cursos.profesor a objeto.
       */
      {
        path: "$cursos.profesor",
        preserveNullAndEmptyArrays: true,
      },
  },
  {
    $project:
      /**
       * Stage 6:
       * Se genera el formato de salida por curso, profesor, salon, horario,
       * y array de los dias en los cuales se dicta el curso.
       */
      {
        _id: 0,
        curso: "$cursos.nombre",
        profesor: "$cursos.profesor.nombre",
      },
  },
}

```

```

salon: {
  $concat: [
    {
      $arrayElemAt: [
        "$cursos.calendario.salon",
        0,
      ],
    },
    " - Edificio: ",
    {
      $arrayElemAt: [
        "$cursos.calendario.edificio",
        0,
      ],
    },
  ],
},
horario: {
  $concat: [
    {
      $arrayElemAt: [
        "$cursos.calendario.inicio",
        0,
      ],
    },
    " - ",
    {
      $arrayElemAt: [
        "$cursos.calendario.fin",
        0,
      ],
    },
  ],
},
dias: "$cursos.calendario.dia",
},
],
).toArray();

cat("PuntoA.json", toJson(PuntoA));
print(PuntoA);

```

## Resultado JSON

```

## [
##   {
##     "curso": "Introducción a la Programación",
##     "profesor": "Ben Stone",
##     "salon": "1.4 - Edificio: El Lago",
##     "horario": "08:00 AM - 10:00 AM",
##     "dias": [
##       "Martes",
##       "Jueves"

```

```

## ]
## },
## {
##   "curso": "Introducción al Modelado de Sistemas",
##   "salon": null,
##   "horario": null
## },
## {
##   "curso": "Estructuras de Datos",
##   "profesor": "Ben Stone",
##   "salon": "3.4 - Edificio: Almendros",
##   "horario": "10:00 AM - 12:00 PM",
##   "dias": [
##     "Martes",
##     "Jueves"
##   ]
## },
## {
##   "curso": "Técnicas y Prácticas de Programación",
##   "profesor": "Ben Stone",
##   "salon": "3.4 - Edificio: Almendros",
##   "horario": "04:00 PM - 06:00 PM",
##   "dias": [
##     "Lunes",
##     "Miércoles"
##   ]
## },
## {
##   "curso": "Programación Orientada a Objetos",
##   "salon": null,
##   "horario": null
## },
## {
##   "curso": "Lógica Digital y Lenguaje Máquina",
##   "profesor": "Grace Stone",
##   "salon": "1.2 - Edificio: El Lago",
##   "horario": "02:00 PM - 04:00 PM",
##   "dias": [
##     "Miércoles",
##     "Viernes"
##   ]
## },
## {
##   "curso": "Programación Funcional",
##   "salon": null,
##   "horario": null
## },
## {
##   "curso": "Lógica para Ciencias de la Computación",
##   "salon": null,
##   "horario": null
## },
## {
##   "curso": "Árboles y Grafos",

```

```

##  "salon": null,
##  "horario": null
## },
## {
##  "curso": "Arquitectura del Computador",
##  "profesor": "Grace Stone",
##  "salon": "2.6 - Edificio: El Lago",
##  "horario": "04:00 PM - 06:00 PM",
##  "dias": [
##    "Lunes",
##    "Viernes"
##  ]
## },
## {
##  "curso": "Diseño de Interfaces Humano-Computador",
##  "salon": null,
##  "horario": null
## },
## {
##  "curso": "Computabilidad y Complejidad",
##  "salon": null,
##  "horario": null
## },
## {
##  "curso": "Computación Gráfica",
##  "salon": null,
##  "horario": null
## },
## {
##  "curso": "Comunicación de Datos",
##  "salon": null,
##  "horario": null
## },
## {
##  "curso": "Desarrollo Formal de Sistemas",
##  "salon": null,
##  "horario": null
## },
## {
##  "curso": "Análisis y Diseño de Algoritmos",
##  "salon": null,
##  "horario": null
## },
## {
##  "curso": "Gestión y Modelación de Datos",
##  "salon": null,
##  "horario": null
## },
## {
##  "curso": "Sistemas Operativos",
##  "salon": null,
##  "horario": null
## },
## {

```

```

## "curso": "Programación Paralela",
## "profesor": "Ulrich Nielsen",
## "salon": "3.3 - Edificio: Palmas",
## "horario": "08:00 AM - 10:00 AM",
## "dias": [
##     "Martes",
##     "Jueves"
## ]
## },
## {
##     "curso": "Aprendizaje Automático",
##     "profesor": "Robert Vance",
##     "salon": "1.1 - Edificio: El Samán",
##     "horario": "04:00 PM - 06:00 PM",
##     "dias": [
##         "Martes",
##         "Jueves"
##     ]
## },
## {
##     "curso": "Animación y Simulación",
##     "salon": null,
##     "horario": null
## },
## {
##     "curso": "Introducción a la Seguridad Informática",
##     "profesor": "Robert Vance",
##     "salon": "3.5 - Edificio: Palmas",
##     "horario": "04:00 PM - 06:00 PM",
##     "dias": [
##         "Miércoles",
##         "Viernes"
##     ]
## },
## {
##     "curso": "Construcción de Software y Pruebas",
##     "salon": null,
##     "horario": null
## },
## {
##     "curso": "Sistemas Inteligentes",
##     "profesor": "Robert Vance",
##     "salon": "2.2 - Edificio: Cedro Rosado",
##     "horario": "10:00 AM - 12:00 PM",
##     "dias": [
##         "Lunes",
##         "Miércoles"
##     ]
## },
## {
##     "curso": "Computación en la Nube",
##     "salon": null,
##     "horario": null
## },

```

```

## {
##   "curso": "Desarrollo de Videojuegos",
##   "profesor": "Ben Stone",
##   "salon": "3.2 - Edificio: Palmas",
##   "horario": "02:00 PM - 05:00 PM",
##   "dias": [
##     "Martes",
##     "Viernes"
##   ]
## },
## {
##   "curso": "Cinemática y Dinámica",
##   "profesor": "Saanvi Bahl",
##   "salon": "2.2 - Edificio: El Lago",
##   "horario": "02:00 PM - 04:00 PM",
##   "dias": [
##     "Lunes",
##     "Miércoles"
##   ]
## },
## {
##   "curso": "Electricidad y Magnetismo",
##   "profesor": "Saanvi Bahl",
##   "salon": "1.5 - Edificio: El Lago",
##   "horario": "04:00 PM - 06:00 PM",
##   "dias": [
##     "Lunes",
##     "Miércoles",
##     "Viernes"
##   ]
## },
## {
##   "curso": "Cálculo Diferencial",
##   "profesor": "Grace Stone",
##   "salon": "2.5 - Edificio: El Lago",
##   "horario": "08:00 AM - 10:00 AM",
##   "dias": [
##     "Martes",
##     "Jueves"
##   ]
## },
## {
##   "curso": "Álgebra Lineal",
##   "profesor": "Grace Stone",
##   "salon": "3.1 - Edificio: El Lago",
##   "horario": "10:00AM - 12:00 PM",
##   "dias": [
##     "Martes",
##     "Jueves"
##   ]
## },
## {
##   "curso": "Cálculo Integral",
##   "profesor": "Jared Vásquez",

```



```

## "salon": "3.3 - Edificio: El Lago",
## "horario": "10:00 AM - 12:00 PM",
## "dias": [
##   "Martes",
##   "Jueves"
## ]
## },
## {
##   "curso": "Cálculo Multivariado",
##   "profesor": "Jared Vásquez",
##   "salon": "2.1 - Edificio: El Lago",
##   "horario": "02:00 PM - 04:00 PM",
##   "dias": [
##     "Martes",
##     "Jueves"
##   ]
## },
## {
##   "curso": "Optimización Matemática",
##   "profesor": "Jared Vásquez",
##   "salon": "2.2 - Edificio: El Lago",
##   "horario": "02:00 PM - 04:00 PM",
##   "dias": [
##     "Lunes",
##     "Miércoles"
##   ]
## },
## {
##   "curso": "Optimización Matemática",
##   "profesor": "Ulrich Nielsen",
##   "salon": null,
##   "horario": null
## },
## {
##   "curso": "Teología",
##   "profesor": "Zeke Landon",
##   "salon": "1.8 - Edificio: El Samán",
##   "horario": "10:00 AM - 12:00 PM",
##   "dias": [
##     "Lunes"
##   ]
## },
## {
##   "curso": "Ética",
##   "profesor": "Jared Vásquez",
##   "salon": "3.7 - Edificio: Almendros",
##   "horario": "08:00 AM - 10:00 AM",
##   "dias": [
##     "Viernes"
##   ]
## },
## {
##   "curso": "Ética",
##   "profesor": "Mikaela Stone",

```

```
## "salon": "3.1 - Edificio: Almendros",
## "horario": "08:00 AM - 10:00 AM",
## "dias": [
##   "Miércoles"
## ]
## },
## {
##   "curso": "Trabajo de Grado",
##   "salon": null,
##   "horario": null
## },
## {
##   "curso": "Práctica Estudiantil",
##   "profesor": "Jared Vásquez",
##   "salon": "1.6 - Edificio: El Samán",
##   "horario": "02:00 PM - 04:00 PM",
##   "dias": [
##     "Viernes"
##   ]
## }
## ]
##
```

## Resultado Tabular

```
fromJSON('PuntoA.json', flatten = T) |>
  mutate(dias = map_chr(dias, ~str_c(., collapse = "// "))) |>
  arrange(curso, profesor) |>
  select(curso, profesor, salon, horario, dias) |>
  kable("latex", booktabs=T, linesep = "", longtable = TRUE) |>
  kable_styling(position = "center",
                latex_options = c("HOLD_position", "repeat_header",
                                "striped")) |>

  column_spec(1, width = "10em") |>
  column_spec(6, width = "6em") |>
  column_spec(4, width = "6em") |>
  column_spec(5, width = "6em")
```

curso	profesor	salon	horario	dias
Animación y Simulación	NA	NA	NA	
Análisis y Diseño de Algoritmos	NA	NA	NA	
Aprendizaje Automático	Robert Vance	1.1 - Edificio: El Samán	04:00 PM - 06:00 PM	Martes// Jueves
Arquitectura del Computador	Grace Stone	2.6 - Edificio: El Lago	04:00 PM - 06:00 PM	Lunes// Viernes
Cinematografía y Dinámica	Saanvi Bahl	2.2 - Edificio: El Lago	02:00 PM - 04:00 PM	Lunes// Miércoles
Computabilidad y Complejidad	NA	NA	NA	

(continued)

curso	profesor	salon	horario	dias
Computación Gráfica	NA	NA	NA	
Computación en la Nube	NA	NA	NA	
Comunicación de Datos	NA	NA	NA	
Construcción de Software y Pruebas	NA	NA	NA	
Cálculo Diferencial	Grace Stone	2.5 - Edificio: El Lago	08:00 AM - 10:00 AM	Martes// Jueves
Cálculo Integral	Jared Vásquez	3.3 - Edificio: El Lago	10:00 AM - 12:00 PM	Martes// Jueves
Cálculo Multivariado	Jared Vásquez	2.1 - Edificio: El Lago	02:00 PM - 04:00 PM	Martes// Jueves
Desarrollo Formal de Sistemas	NA	NA	NA	
Desarrollo de Videojuegos	Ben Stone	3.2 - Edificio: Palmas	02:00 PM - 05:00 PM	Martes// Viernes
Diseño de Interfaces Humano-Computador	NA	NA	NA	
Electricidad y Magnetismo	Saanvi Bahl	1.5 - Edificio: El Lago	04:00 PM - 06:00 PM	Lunes// Miércoles// Viernes
Estructuras de Datos	Ben Stone	3.4 - Edificio: Almendros	10:00 AM - 12:00 PM	Martes// Jueves
Gestión y Modelación de Datos	NA	NA	NA	
Introducción a la Programación	Ben Stone	1.4 - Edificio: El Lago	08:00 AM - 10:00 AM	Martes// Jueves
Introducción a la Seguridad Informática	Robert Vance	3.5 - Edificio: Palmas	04:00 PM - 06:00 PM	Miércoles// Viernes
Introducción al Modelado de Sistemas	NA	NA	NA	
Lógica Digital y Lenguaje Máquina	Grace Stone	1.2 - Edificio: El Lago	02:00 PM - 04:00 PM	Miércoles// Viernes
Lógica para Ciencias de la Computación	NA	NA	NA	
Optimización Matemática	Jared Vásquez	2.2 - Edificio: El Lago	02:00 PM - 04:00 PM	Lunes// Miércoles
Optimización Matemática	Ulrich Nielsen	NA	NA	
Programación Funcional	NA	NA	NA	
Programación Orientada a Objetos	NA	NA	NA	
Programación Paralela	Ulrich Nielsen	3.3 - Edificio: Palmas	08:00 AM - 10:00 AM	Martes// Jueves
Práctica Estudiantil	Jared Vásquez	1.6 - Edificio: El Samán	02:00 PM - 04:00 PM	Viernes
Sistemas Inteligentes	Robert Vance	2.2 - Edificio: Cedro Rosado	10:00 AM - 12:00 PM	Lunes// Miércoles
Sistemas Operativos	NA	NA	NA	

(continued)

curso	profesor	salon	horario	dias
Teología	Zeke Landon	1.8 - Edificio: El Samán	10:00 AM - 12:00 PM	Lunes
Trabajo de Grado	NA	NA	NA	
Técnicas y Prácticas de Programación	Ben Stone	3.4 - Edificio: Almendros	04:00 PM - 06:00 PM	Lunes// Miércoles
Álgebra Lineal	Grace Stone	3.1 - Edificio: El Lago	10:00AM - 12:00 PM	Martes// Jueves
Árboles y Grafos	NA	NA	NA	
Ética	Jared Vásquez	3.7 - Edificio: Almendros	08:00 AM - 10:00 AM	Viernes
Ética	Mikaela Stone	3.1 - Edificio: Almendros	08:00 AM - 10:00 AM	Miércoles

## Punto B

*Obtener la lista de profesores que dictan cursos en **dos** carreras diferentes. Se deben obtener los siguientes datos: nombre completo del profesor, nombre del curso que dicta y nombre de la carrera a la que pertenece el curso.*

## Query

**NOTA:** Se entiende que la solicitud consiste en obtener aquellos profesores que imparten cursos exclusivamente en dos carreras. Como todos los profesores que aparecen en el resultado dictan cursos en dos carreras y tienen múltiples cursos por carrera, se han dejado los datos anidados en formato JSON para aprovechar esta funcionalidad propia de MongoDB que facilita la comunicación entre sistemas. No obstante, para facilitar la lectura humana en el resultado en formato tabular se muestran los datos desanidados.

```
use university

var PuntoB = db.carreras.aggregate([
  {
    $unwind:
      /**
       * Stage 1:
       * Se saca la lista completa de cursos por carrera.
       */
      {
        path: "$cursos",
        preserveNullAndEmptyArrays: true,
      },
  },
  {
    $project:
      /**
       * Stage 2:
       * Se dejan solo los campos de nombre de carrera e id del curso.
       */
  }
```

```

    */
    {
      _id: 0,
      carrera: "$nombre",
      id_curso: "$cursos.curso",
    },
  },
  {
    $lookup:
      /**
       * Stage 3:
       * Unimos con la colección curso por id del curso.
       */
      {
        from: "cursos",
        localField: "id_curso",
        foreignField: "_id",
        as: "curso",
      },
  },
  {
    $unwind:
      /**
       * Stage 4:
       * Pasamos el array curso a objeto.
       */
      {
        path: "$curso",
        preserveNullAndEmptyArrays: true,
      },
  },
  {
    $project:
      /**
       * Stage 5:
       * Dejamos los campos de interés carrera, nombre del curso e id del
       * profesor.
       */
      {
        _id: 0,
        carrera: "$carrera",
        curso: "$curso.nombre",
        id_profesor: "$curso.profesor",
      },
  },
  {
    $lookup:
      /**
       * Stage 6:
       * Unimos con la colección profesores por id del profesor.
       */
      {
        from: "profesores",

```

```

        localField: "id_profesor",
        foreignField: "_id",
        as: "profesor",
    },
},
{
    $unwind:
        /**
         * Stage 7:
         * Pasamos e array profesor a objeto
         */
        {
            path: "$profesor",
            preserveNullAndEmptyArrays: false,
        },
},
{
    $project:
        /**
         * Stage 8: Dejamos lo campos de interes nombre del profesor, carrera y
         * curso
         */
        {
            profesor: "$profesor.nombre",
            carrera: "$carrera",
            curso: "$curso",
        },
},
{
    $group:
        /**
         * Stage 9:
         * Agrupamos por profesor y carrera obteniendo un array de cursos de esa
         * combinación
         */
        {
            _id: {
                profesor: "$profesor",
                carrera: "$carrera",
            },
            cursos: {
                $addToSet: "$curso",
            },
        },
},
{
    $group:
        /**
         * Stage 10:
         * Agrupamos por solo el profesor, colocando en un array la lista de
         * carreras en las que dicta el profesor con sus respectivos cursos
         */
        {

```

```

        _id: "$_id.profesor",
        carreras_cursos: {
            $addToSet: {
                carrera: "$_id.carrera",
                cursos: "$cursos",
            },
        },
    },
},
{
    $project:
        /**
         * Stage 11:
         * cambiamos visualmente el _id a profesor
         */
        {
            _id: 0,
            profesor: "$_id",
            carreras_cursos: "$carreras_cursos",
        },
},
{
    $match:
        /**
         * Stage 12
         * Filtramos por aquellos profesores cuyos conteos de carreras sean
         * igual a 2
         */
        {
            $expr: {
                $eq: [
                    {
                        $size: "$carreras_cursos",
                    },
                    2,
                ],
            },
        },
},
]).toArray();

cat("PuntoB.json", toJson(PuntoB));
print(PuntoB);

```

## Resultado JSON

```

## [
## {
##   "profesor": "Ben Stone",
##   "carreras_cursos": [
##     {
##       "carrera": "Matemáticas Aplicadas",
##       "cursos": [

```

```

##      "Estructuras de Datos"
##    ]
##  },
##  {
##    "carrera": "Ingeniería de Sistemas",
##    "cursos": [
##      "Desarrollo de Videojuegos",
##      "Introducción a la Programación",
##      "Estructuras de Datos",
##      "Técnicas y Prácticas de Programación"
##    ]
##  }
## ]
## },
## {
##   "profesor": "Claudia Tiedemann",
##   "careras_cursos": [
##     {
##       "carrera": "Matemáticas Aplicadas",
##       "cursos": [
##         "Álgebra Moderna"
##       ]
##     },
##     {
##       "carrera": "Ingeniería Mecánica",
##       "cursos": [
##         "Propiedades de los Materiales",
##         "Diseño Mecánico",
##         "Máquinas Térmicas e Hidráulicas"
##       ]
##     }
##   ]
## },
## {
##   "profesor": "Ulrich Nielsen",
##   "careras_cursos": [
##     {
##       "carrera": "Ingeniería de Sistemas",
##       "cursos": [
##         "Programación Paralela",
##         "Optimización Matemática"
##       ]
##     },
##     {
##       "carrera": "Matemáticas Aplicadas",
##       "cursos": [
##         "Optimización Matemática"
##       ]
##     }
##   ]
## }
## ]
##

```



## Resultado Tabular

```
fromJSON('PuntoB.json', flatten = T) |>
  unnest(careras_cursos, keep_empty = T) |>
  unnest(cursos, keep_empty = T) |>
  arrange(profesor, carrera, cursos) |>
  kable("latex", booktabs = T, linesep = "", longtable = TRUE) |>
  kable_styling(position = "center",
                latex_options = c("HOLD_position", "repeat_header",
                                  "striped"))
```

profesor	carrera	cursos
Ben Stone	Ingeniería de Sistemas	Desarrollo de Videojuegos
Ben Stone	Ingeniería de Sistemas	Estructuras de Datos
Ben Stone	Ingeniería de Sistemas	Introducción a la Programación
Ben Stone	Ingeniería de Sistemas	Técnicas y Prácticas de Programación
Ben Stone	Matemáticas Aplicadas	Estructuras de Datos
Claudia Tiedemann	Ingeniería Mecánica	Diseño Mecánico
Claudia Tiedemann	Ingeniería Mecánica	Máquinas Térmicas e Hidráulicas
Claudia Tiedemann	Ingeniería Mecánica	Propiedades de los Materiales
Claudia Tiedemann	Matemáticas Aplicadas	Álgebra Moderna
Ulrich Nielsen	Ingeniería de Sistemas	Optimización Matemática
Ulrich Nielsen	Ingeniería de Sistemas	Programación Paralela
Ulrich Nielsen	Matemáticas Aplicadas	Optimización Matemática

## Punto C

### Query

**NOTA:** Este query parte del supuesto de que un estudiante únicamente puede inscribirse en cursos de su carrera correspondientes a un semestre igual o anterior al que actualmente está cursando. Además, tras realizar el ejercicio de la documentación, se ha detectado que la colección de **cursos** incluye el campo **cupo**. Asumimos que si un curso no cuenta con este campo, significa que ya se ha completado su cupo y no es posible inscribirse en él.

```
use university

var PuntoC = db.estudiantes.aggregate([
  {
    $match:
      /**
       * Stage 1:
       * Filtramos por un estudiante.
       */
    {
      nombre: "Olive Stone",
    },
  },
])
```

```

},
{
  $unwind:
    /**
     * Stage 2:
     * Pasamos el array carreras un objeto.
     */
    {
      path: "$carreras",
      preserveNullAndEmptyArrays: true,
    },
},
{
  $lookup:
    /**
     * Stage 3:
     * Unimos con la colección carreras por el id de la carrera
     */
    {
      from: "carreras",
      localField: "carreras",
      foreignField: "_id",
      as: "carreras",
    },
},
{
  $unwind:
    /**
     * Stage 4:
     * Pasamos el array carrrras a un objeto.
     */
    {
      path: "$carreras",
      preserveNullAndEmptyArrays: true,
    },
},
{
  $project:
    /**
     * Stage 5:
     * Seleccionamos los campos de interes estudiante, carrera, semestre,
     * creamos el array cursos_carrera que contiene todos los cursos que ofrece
     * la carrera y cramos el array cursos_inscritos que contiene los curso ya
     * que el estudiante ya ha inscrito.
     */
    {
      _id: 0,
      estudiante: "$nombre",
      carrera: "$carreras.nombre",
      semestre: "$semestre",
      cursos_carrera: "$carreras.cursos",
      cursos_inscritos: "$cursos",
    },
},

```

```

},
{
  $project:
    /**
     * Stage 6:
     * Seleccionamos la carrera y aquellos registros de curso_carrera cuyo
     * semestre es menor o igual al semestre del estudiante y que no se
     * encuentren contenidos en la lista de cursos que ya matriculó este.
     */
    {
      carrera: "$carrera",
      cursos_carrera: {
        $filter: {
          input: "$cursos_carrera",
          as: "curso",
          cond: {
            $and: [
              {
                $lte: [
                  "$$curso.semestre",
                  "$semestre",
                ],
              },
              {
                $not: {
                  $in: [
                    "$$curso.curso",
                    "$cursos_inscritos",
                  ],
                },
              },
            ],
          },
        },
      },
    },
  },
},
},
{
  $unwind:
    /**
     * Stage 7:
     * Listamos individualmente los cursos contenidos en cursos_carrera.
     */
    {
      path: "$cursos_carrera",
      preserveNullAndEmptyArrays: true,
    },
},
{
  $project:
    /**
     * Stage 8:
     * Seleccionamos los campos de interes carreram semestre e id del curso.

```

```

    */
    {
        carrera: "$carrera",
        semestre: "$cursos_carrera.semestre",
        id_curso: "$cursos_carrera.curso",
    },
},
{
    $lookup:
    /**
     * Stage 9:
     * Unimos con la colección cursos por el id del curso.
     */
    {
        from: "cursos",
        localField: "id_curso",
        foreignField: "_id",
        as: "curso",
    },
},
{
    $match:
    /**
     * Stage 10:
     * Dejamos solo aquellos cursos que cuentan con cupos para ser matriculados.
     */
    {
        "curso.cupo": {
            $gt: 0,
        },
    },
},
{
    $unwind:
    /**
     * Stage 11:
     * Pasamos el array curso a un objeto.
     */
    {
        path: "$curso",
        preserveNullAndEmptyArrays: true,
    },
},
{
    $project:
    /**
     * Stage 12:
     * Seleccionamos los campos de interes curso, carrera y semestre.
     */
    {
        curso: "$curso.nombre",
        carrera: "$carrera",
        semestre: "$semestre",
    },
},

```

```

    },
  },
]).toArray();

cat("PuntoC.json", toJSON(PuntoC));
print(PuntoC);

```

## Resultado JSON

```

## [
##  {
##    "curso": "Estructuras de Datos",
##    "carrera": "Matemáticas Aplicadas",
##    "semestre": 2
##  },
##  {
##    "curso": "Álgebra Moderna",
##    "carrera": "Matemáticas Aplicadas",
##    "semestre": 4
##  },
##  {
##    "curso": "Estadística Aplicada",
##    "carrera": "Matemáticas Aplicadas",
##    "semestre": 5
##  }
## ]
##

```

## Resultado Tabular

```

fromJSON('PuntoC.json', flatten = T) |>
  kable("latex", booktabs = T, linesep = "", longtable = TRUE) |>
  kable_styling(position = "center",
    latex_options = c("HOLD_position", "repeat_header",
      "striped"))

```

curso	carrera	semestre
Estructuras de Datos	Matemáticas Aplicadas	2
Álgebra Moderna	Matemáticas Aplicadas	4
Estadística Aplicada	Matemáticas Aplicadas	5