

INF-343 SISTEMAS DISTRIBUIDOS

TAREA N°1: “Desarrollo de una aplicación distribuida simple de reserva aérea usando servicios web”

27 de marzo de 2023

ANTECEDENTES GENERALES

Objetivos:

- Aprender a programar con el lenguaje de programación concurrente desarrollado por Google (<https://go.dev/>)
- Desarrollar una aplicación con servicios Web basados en RESTful APIs, programando con lenguaje Go y usando Gin Web Framework. Se usará como base de datos (NoSQL) MongoDB para la persistencia de los datos.

Fechas importantes:

Ayudantía por Zoom: jueves 30 de marzo de 2023

Fecha de entrega de la tarea: miércoles 19 de abril de 2023

ESPECIFICACIÓN DE LA APLICACIÓN

Una aerolínea acaba de salir al mercado y ya dispone de diversas rutas entre distintas ciudades del mundo. Debido a que está en un proceso de digitalización, necesita apoyo para desarrollar los procesos básicos para que comience a operar. En particular, necesita desarrollar tecnología para los siguientes casos:

1. Gestión de reservas.
2. Generación de estadísticas generales.

Por simplicidad, supondremos que la tecnología solo es manejada por el personal de la aerolínea, por lo que cualquier requerimiento por parte de algún pasajero, tiene que ser el personal quien intervenga. Los pasajeros generalmente necesitan:

1. Hacer reservas.
2. Modificar reservas.

Debido a que esta aerolínea es de bajo costo (*low cost*), los siguientes servicio adicionales (*ancillaries*) se venden para un pasaje:

Servicios Adicionales (<i>ancillaries</i>)	Costo
Equipaje de mano	10.000
Equipaje de bodega	30.000
Asiento	5.000
Embarque y <i>Check In</i> prioritario	2.000

Mascota en cabina	40.000
Mascota en bodega	40.000
Equipaje especial	35.000
Acceso a Salón VIP	15.000
Wi-Fi a bordo	20.000

Para realizar una reserva, la aerolínea necesita los siguientes datos de parte del pasajero:

1. Fecha de ida
2. Fecha de regreso (solo si aplica)
3. Ciudad de origen
4. Ciudad de destino
5. Cantidad de pasajeros
6. Nombre y apellido de cada pasajero
7. Edad de cada pasajero
8. Servicios adicionales para cada pasajero

Inicialmente la aerolínea solo necesita los primeros 4 puntos, ya que luego de eso le menciona al pasajero los vuelos que hay a disposición y su precio. Dependiendo del vuelo que haya seleccionado, se le indicará los servicios adicionales (*ancillaries*) que hay disponibles, ya que depende del stock que tenga vuelo hasta ese momento.

Una vez que el pasajero confirme sus datos, una reserva será generada y contendrá todos los datos mencionados previamente. Además, también se añadirá un PNR (Passenger Name Record), el cual corresponde a una especie de ID. El PNR estará formado por letras del alfabeto inglés y dígitos entre 0 y 9, el largo es de 6 caracteres.

Luego que la reserva se encuentre creada, alguno de los pasajeros podrían querer modificarla. Para esto, solo se necesitará el PNR y el apellido. El pasajero no puede quitar los *ancillaries* añadidos en la venta inicial, pero sí puede añadir más en caso de que el stock del vuelo lo permita. Otra opción que tiene es cambiar las fechas, ya sea para la ida o la vuelta (si es que aplica). El costo de cambiar una fecha es equivalente a la diferencia con respecto al vuelo anterior más 20.000 por cada pasajero. Si el tipo de reserva es ida y vuelta, ninguno de los dos vuelos que se quieran cambiar tienen que ser incompatibles con las fechas, por ejemplo.

Los directivos de la aerolínea necesitan recopilar información valiosa que les permita mejorar el negocio, ya sea añadiendo o quitando rutas, comprando más aviones, contratando más personal, etc. Para esto, de acuerdo a las reservas hechas de manera histórica, necesitan responder las siguientes preguntas:

1. ¿Cuál es la ruta que ha significado mayor ganancia?
2. ¿Cuál es la ruta que ha significado menor ganancia?
3. ¿Cuál es el ranking de *ancillaries* en cuánto a ganancia total por cada uno?
4. ¿Cuál es el promedio de pasajeros por mes?

DESARROLLO

El desarrollo debe realizarse a través del lenguaje de programación Go y utilizando como base de datos MongoDB.

Los modelos son los siguientes:

Vuelos

Atributos	Tipo de dato
numero_vuelo	string
origen	string
destino	string
hora_salida	string
hora_llegada	string
avion	Objeto
ancillaries	Array

Ejemplo (para un vuelo):

```
{
  "numero_vuelo": "323",
  "origen": "IQQ",
  "destino": "SCL",
  "hora_salida": "20:38",
  "hora_llegada": "22:50",
  "fecha": "14/04/2023",
  "avion": {
    "modelo": "A320neo",
    "numero_de_serie": "12345",
    "stock_de_pasajeros": 90
  },
  "ancillaries": [
    {
      "nombre": "Equipaje de mano",
      "stock": 68,
      "ssr": "BGH"
    },
    {
      "nombre": "Equipaje de bodega",
      "stock": 92,
      "ssr": "BGR"
    }
  ]
}
```

```

    {
      "nombre": "Asiento",
      "stock": 40,
      "ssr": "STDF"
    },
    {
      "nombre": "Embarque y Check In prioritario",
      "stock": 79,
      "ssr": "PAXS"
    },
    {
      "nombre": "Mascota en cabina",
      "stock": 4,
      "ssr": "PTCR"
    },
    {
      "nombre": "Mascota en bodega",
      "stock": 12,
      "ssr": "AVIH"
    },
    {
      "nombre": "Equipaje especial",
      "stock": 71,
      "ssr": "SPML"
    },
    {
      "nombre": "Acceso a Salón VIP",
      "stock": 36,
      "ssr": "LNGE"
    },
    {
      "nombre": "Wi-Fi a bordo",
      "stock": 57,
      "ssr": "WIFI"
    }
  ]
}

```

Reservas

vuelos	array
pasajeros	array

Ejemplo:

```

{
  "vuelos": [
    {
      "numero_vuelo": "323",
      "origen": "IQQ",

```

```

        "destino": "SCL",
        "hora_salida": "20:38",
        "hora_llegada": "22:50",
        "fecha": "14/04/2023"
    },
    {
        "numero_vuelo": "460",
        "origen": "SCL",
        "destino": "IQQ",
        "hora_salida": "14:54",
        "hora_llegada": "17:24",
        "fecha": "21/04/2023"
    }
],
"pasajeros": [
    {
        "nombre": "Juan",
        "apellido": "Rojas",
        "edad": 25,
        "ancillaries": [
            {
                "ida": [
                    {
                        "ssr": "BGH",
                        "cantidad": 2
                    },
                    {
                        "ssr": "PAXS",
                        "cantidad": 1
                    }
                ],
                "vuelta": [
                    {
                        "ssr": "STDF",
                        "cantidad": 1
                    }
                ]
            }
        ],
        "balances": {
            "ancillaries_ida": 22000,
            "vuelo_ida": 77880,
            "ancillaries_vuelta": 5000,
            "vuelo_vuelta": 88500,
        }
    }
]
}

```

Interfaz de los servicios Web (*endpoints*)

Para la persistencia y recuperación de los datos, se deberá desarrollar una API REST, los *endpoints* son los siguientes:

Endpoint	127.0.0.1:5000/api/reserva
Descripción	Obtener una reserva
Método	GET
Parámetros	?pnr=AABB12&apellido=rojas
Respuesta	Idéntico al ejemplo
Observación	Si la reserva no se encuentra, retornar mensaje descriptivo.

Endpoint	127.0.0.1:5000/api/reserva
Descripción	Crear una reserva
Método	POST
Cuerpo	Idéntico al ejemplo
Respuesta	{ "PNR": "AABB12" }

Endpoint	127.0.0.1:5000/api/reserva
Descripción	Modificar una reserva
Método	PUT
Parámetros	?pnr=AABB12&apellido=rojas
Cuerpo	Idéntico al ejemplo
Respuesta	{ "PNR": "AABB12" }

Endpoint	127.0.0.1:5000/api/vuelo
Descripción	Obtener vuelos
Método	GET
Parámetros	?origen=IQQ&destino=SCL&fecha=14/04/23
Respuesta	[{ }

	<pre> "numero_vuelo": "323", "origen": "IQQ", "destino": "SCL", "hora_salida": "20:38", "hora_llegada": "22:50", "fecha": "14/04/2023", "avion": { "modelo": "A320neo", "numero_de_serie": "12345", "stock_de_pasajeros": 90 }] </pre>
Observación	Si no se encuentran vuelos, retornar un array vacío.

Endpoint	127.0.0.1:5000/api/vuelo
Descripción	Modificar el stock de pasajeros para un vuelo
Método	PUT
Parámetros	?numero_vuelo=323&origen=IQQ&destino=SCL&fecha=14/04/23
Cuerpo	<pre> { "stock_de_pasajeros": 88 } </pre>
Respuesta	<pre> { "numero_vuelo": "323", "origen": "IQQ", "destino": "SCL", "hora_salida": "20:38", "hora_llegada": "22:50" } </pre>

Endpoint	127.0.0.1:5000/api/estadisticas
Descripción	Obtener estadísticas
Método	GET
Parámetros	Sin parámetros
Respuesta	<pre> { "ruta_mayor_ganancia": "SCL-IQQ", "ruta_menor_ganancia": "PUC-PMC", "ranking_ancillaries": [{ "nombre": "Equipaje de mano", "ganancia": 86314660, "ssr": "BGH" }, </pre>

	<pre> { "nombre": "Equipaje de bodega", "ganancia": 47471110, "ssr": "BGR" }, { "nombre": "Asiento", "ganancia": 87437468, "ssr": "STDF" }, { "nombre": "Embarque y Check In prioritario", "ganancia": 63250477, "ssr": "PAXS" }, { "nombre": "Mascota en cabina", "ganancia": 77324078, "ssr": "PTCR" }, { "nombre": "Mascota en bodega", "ganancia": 45496322, "ssr": "AVIH" }, { "nombre": "Equipaje especial", "ganancia": 79477730, "ssr": "SPML" }, { "nombre": "Acceso a Salón VIP", "ganancia": 26140914, "ssr": "LNGE" }, { "nombre": "Wi-Fi a bordo", "ganancia": 68095436, "ssr": "WIFI" } }, "promedio_pasajeros": { "enero": 1212, "febrero": 1385, "marzo": 726, "abril": 1050, "mayo": 890, "junio": 1295, "julio": 554, "agosto": 942, "septiembre": 1357, "octubre": 690, "noviembre": 1117, "diciembre": 821 </pre>
--	---

	}
Observación	Si no hay suficientes datos, establecer los valores numéricos en 0.

Programa

Dentro del directorio de la tarea, tiene que existir un archivo `main.go` que incluya la ejecución inicial del programa, cuando se ejecute, debe desplegarse un menú y en base a este menú interactuar con la API REST. Otro que se requerirá, es un archivo `.env` que incluya todas las variables de entorno; como mínimo, tiene que contener las siguientes:

1. SERVER: 127.0.0.1
2. PORT: 5000
3. CONNECTION_STRING: mongodb://admin:admin@127.0.0.1:27017/aerolinea

El primero y el segundo corresponden a los datos del servicio que debe desplegar con Go. El tercero, es el string de conexión para MongoDB. Ninguno de los valores anteriores tienen que cambiarse, tiene que usar exactamente los mismos para la tarea.

El tercer archivo requerido es `server.go`, que incluye toda la lógica de la API REST y que se ejecuta al levantar un servicio local en el puerto 5000, tomando los valores desde el archivo `.env`.

Ejemplos de ejecución (las tildes se omitieron)

```
Menu
1. Gestionar reserva
2. Obtener estadísticas
3. Salir
Ingrese una opcion: 1
Submenu:
1. Crear reserva
2. Obtener reserva
3. Modificar reserva
4. Salir
Ingrese una opcion: 1
Ingrese fecha de ida: 14/04/2023
Ingrese fecha de regreso: 21/04/23
Ingrese origen: IQQ
Ingrese destino: SCL
Ingrese cantidad de pasajeros: 1
Vuelos disponibles:
Ida:
1. 323 20:38 - 22:50 $77880
2. 481 14:27 - 16:42 $79650
Ingrese una opción: 1
```

Vuelta:

1. 320 04:24 - 06:49 \$85550

Ingrese una opción: **1**

Pasajero 1:

Ingrese nombre: **Juan**

Ingrese apellido: **Perez**

Ingrese edad: **25**

Ancillaries Ida:

1. Equipaje de mano: 10000

2. Equipaje de bodega: 30000

3. Asiento: 5000

4. Embarque y Check In prioritario: 2000

5. Mascota en cabina: 40000

6. Mascota en bodeg: 40000

7. Equipaje especial: 35000

8. Acceso a Salón VIP: 15000

9. Wi-Fi a bordo: 20000

Ingrese los ancillaries (separados por coma): **1,2**

Total ancillaries: 40000

Ancillaries Vuelta:

1. Equipaje de mano: 10000

2. Equipaje de bodega: 30000

3. Asiento: 5000

4. Embarque y Check In prioritario: 2000

5. Mascota en cabina: 40000

6. Mascota en bodeg: 40000

7. Equipaje especial: 35000

8. Acceso a Salón VIP: 15000

9. Wi-Fi a bordo: 20000

Ingrese los ancillaries (separados por coma): **5**

Total ancillaries: 40000

La reserva fue generada el PNR es: ABCD12

El costo total de la reserva fue de \$243430

Submenu:

1. Crear reserva

2. Obtener reserva

3. Modificar reserva

4. Salir

Ingrese una opcion: **4**

Menu

1. Gestionar reserva

2. Obtener estadisticas

3. Salir

Ingrese una opcion: **3**

Programa finalizado!

```
Menu
1. Gestionar reserva
2. Obtener estadisticas
3. Salir
Ingrese una opcion: 1
Submenu:
1. Crear reserva
2. Obtener reserva
3. Modificar reserva
4. Salir
Ingrese una opcion: 2
Ingrese el PNR: AAB12
Ingrese el apellido: Perez
Ida:
323 20:38 - 22:50
Vuelta:
320 04:24 - 06:49
Pasajeros:
Juan Perez 25
Ancillaries Ida: BGH,PAXS
Ancillaries vuelta: WIFI
Submenu:
1. Crear reserva
2. Obtener reserva
3. Modificar reserva
4. Salir
Ingrese una opcion: 4
Menu
1. Gestionar reserva
2. Obtener estadisticas
3. Salir
Ingrese una opcion: 3
Programa finalizado!
```

```
Menu
1. Gestionar reserva
2. Obtener estadisticas
3. Salir
Ingrese una opcion: 1
Submenu:
1. Crear reserva
2. Obtener reserva
3. Modificar reserva
4. Salir
Ingrese una opcion: 3
Ingrese el PNR: AAB12
Ingrese el apellido: Perez
Opciones:
1. Cambiar fecha de vuelo
2. Adicionar ancillaries
3. Salir
```

```
Ingrese una opcion: 1
Vuelos:
1. ida: 323 20:38 - 22:50
2. vuelta: 320 04:24 - 06:49
Ingrese una opcion: 2
Ingrese nueva fecha: 22/04/2023
Vuelos disponibles:
1. 325 20:39 - 22:51 $97880
2. 482 14:27 - 16:42 $99650
Ingrese una opcion: 2
La reserva fue modificada exitosamente!
Opciones:
1. Cambiar fecha de vuelo
2. Adicionar ancillaries
Ingrese una opcion: 2
Ingrese el PNR: AAB12
Ingrese el apellido: Perez
Vuelos:
1. ida: 325 20:39 - 22:51
2. vuelta: 482 14:27 - 16:42
Ingrese una opcion: 1
Ancillaries disponibles:
1. Equipaje de mano: 10000
2. Equipaje de bodega: 30000
3. Asiento: 5000
4. Embarque y Check In prioritario: 2000
5. Mascota en cabina: 40000
6. Mascota en bodeg: 40000
7. Equipaje especial: 35000
8. Acceso a Salón VIP: 15000
9. Wi-Fi a bordo: 20000
Ingrese los ancillaries (separados por coma): 1,2
Total ancillaries: 40000
Los ancillaries fueron agregados correctamente!
Opciones:
1. Cambiar fecha de vuelo
2. Adicionar ancillaries
3. Salir
Ingrese una opcion: 3
Submenu:
1. Crear reserva
2. Obtener reserva
3. Modificar reserva
4. Salir
Ingrese una opcion: 4
Menu
1. Gestionar reserva
2. Obtener estadisticas
3. Salir
Ingrese una opcion: 3
Programa finalizado!
```

Observaciones:

1. Tanto el origen y destino se ingresan en el formato de IATA: https://es.wikipedia.org/wiki/C%C3%B3digo_de_aeropuertos_de_IATA
2. El costo de un pasaje es la cantidad de minutos de duración del vuelo multiplicado por 590.
3. Si no se desea un vuelo de regreso, ingresar el string "no", luego eso interpretará para omitir las opciones del vuelo de regreso.
4. No es necesario realizar validaciones. Puede asumir que todo lo que ingresará está en el formato correcto; sin embargo, si hay alguna ambigüedad, ¡pregunte!
5. Considere que el apellido ingresado podría contener mayúsculas o minúsculas, pero para la búsqueda eso es indiferente. Por ejemplo, "Rojas" y "ROJAS" es idéntico a "rojas".

REGLAS DE ENTREGA Y EVALUACIÓN

1. Respetar los valores para las variables de entorno.
2. Solo se realizarán dos ejecuciones para revisar la tarea:
 - a) `go run main.go`
 - b) `go run server.go`
3. Utilice los mismos modelos entregados y no modifique los atributos.
4. Todo el contenido del desarrollo tiene que ser subido a la máquina virtual que se le asignará a su grupo.
5. La tarea tiene que ser subida en un repositorio a <https://gitlab.inf.utfsm.cl>
6. Para dudas, escribir a hugo.leyton@sansano.usm.cl

RÚBICA

1. Aspectos generales:
 - Aprendizaje y dominio del lenguaje Go.
 - Diseño de una arquitectura orientada a servicios usando API RESTful (*endpoints*).
 - Documentación y calidad del código.
2. Configuración de la base de datos en Mongo DB
 - Correcta definición de las colecciones.
3. Desarrollo de un cliente de gestión de reserva
 - Calidad de la interfaz de presentación para el usuario
 - Correcta invocación de los servicios de la API RESTful
4. Desarrollo del cliente de estadísticas del negocio
 - Calidad de la interfaz de presentación para el usuario
 - Correcta invocación de los servicios de la API RESTful

