

# **Instituto Tecnológico Colonia**

## **CTC**

### **Obligatorio: Software para empresa de viajes**

Entregado como requisito para la obtención del título de Analista Programador

Federico Bonjour

Agustin Maciel

Tutores: Carlos Rodríguez

Año: 2022

## **ÍNDICE.**

1.	<b>Declaración de autoría</b>	<b>2</b>
2.	<b>Letra del problema</b>	<b>3</b> <b>4</b>
3.	<b>Descripción del análisis y la solución</b>	<b>5</b> <b>6</b>
4.	<b>Herramientas utilizadas</b>	<b>7</b> <b>8</b>
5.	<b>Alcances obtenidos a consideración del equipo.</b>	<b>9</b> <b>10</b>
6.	<b>Diagrama de clases</b>	<b>11</b>

## 1. Declaración de autoría.

Nosotros, los estudiantes Federico Bonjour y Agustin Maciel, declaramos que la obra presentada en su totalidad está desarrollada por nosotros. Podemos afirmar que:

El proyecto se realizó mientras cursábamos Ing. de Software;

Aplicamos los conocimientos proporcionados por nuestro tutor;

Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la obra es enteramente nuestra;

Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.



---

Federico Bonjour

06/10/2022



---

Agustin Maciel

06/10/2022

## 2. Letra del problema.

### Obligatorio 2 DDA

*Fecha: desde el 03 de noviembre al 30 de noviembre de 2022, a las 23:59.*

Una empresa de turismo quiere manejar los planes para ofertar a sus clientes y luego poder asociar esos planes con los clientes que los adquieran.

Hay clientes estándar y clientes premium. Los clientes premium son aquellos que ya han adquirido 3 planes de viajes o más. Para ellos, el costo del cuarto plan en adelante tiene un 20% de descuento.

#### **EL SISTEMA DEBE PERMITIR:**

- 1) Mostrar un dashboard administrativo en ambiente Web o Mobile (sencillo) que permita trabajar con planes de viaje y clientes.
- 2) Dar de alta, eliminar y modificar planes de viaje. Cada uno de ellos tiene un solo destino (máx. 20 dígitos alfanuméricos), fecha, modalidad (solamente pueden ser aérea, marítima o terrestre), precio en USD, carrusel de fotos (opcional).
- 3) dar de alta, eliminar y modificar clientes. Los clientes tienen CI (máx. 8 dígitos, mín. 7 dígitos, sin puntos ni guiones), nombre (máx. 30 dígitos alfanuméricos), apellido (máx. 30 dígitos alfanuméricos), email (máx. 30 dígitos alfanuméricos), planes comprados (si no tiene ninguno se deberá indicar con un mensaje que "no tiene planes comprados"). 3) contratar o borrarse de un plan (o más) de viaje.
- 4) Listar los planes (viajes) de un cliente mostrando todos los atributos de cada viaje.
- 5) Listar el primer viaje que tendrá un cliente después de una fecha especificada.
- 6) Controlar que no se permita ingresar viajes (planes) con fechas anteriores a la fecha actual.
- 7) Controles de errores para ingreso de datos.

**PASOS A SEGUIR:**

- 1) Analizar la realidad planteada y presentar una solución en Java.
- 2) Realizar el diagrama de clases con sus métodos y atributos.
- 3) Realizar el código que una vez ejecutado permita implementar la solución al problema.

**A ENTREGAR:**

1. Documentación en formato PDF que contenga:
  - a. Portada del trabajo con los nombres de los integrantes del equipo.
  - b. Letra del problema.
  - c. Descripción del análisis y la solución.
  - d. Alcances obtenidos a consideración del equipo.
  - e. Diagrama de clases.
  - f. Código impreso de las clases.
2. El código de las clases implementadas:
  - a. en el backend, en un archivo rar llamado backendObligatorio2-ApellidoDeAlumno1- ApellidoDeAlumno2
  - b. en el frontend, en un archivo rar llamado frontendObligatorio2-ApellidoDeAlumno1- ApellidoDeAlumno2
3. La base de datos (preferentemente en MySQL v 5.7.40 o SQL Server v 2012).

### 3. Descripción del análisis y la solución.

La solución surge a partir de los requerimientos planteados por una empresa enfocada en la venta de planes de viaje, quienes tienen el interés en automatizar los procesos y sobre todo almacenar la información generada por todos los integrantes del proceso:

- Clientes:
  - Los clientes son aquellas personas que realizan compras de planes de viaje a los distintos destinos.
  - Existen clientes standard y premium, siendo estos últimos, aquellos que cuentan con descuentos en las compras.
- Planes de viaje:
  - Son las entidades principales del sistema y son compradas por la entidad mencionada anteriormente.
  - Tienen distintos atributos como destino, costo, modalidad, etc.
  - Las modalidades pueden ser terrestre, marítima o aérea.
- Compras:
  - Las compras son aquellas relaciones entre los clientes y los planes de viaje, las mismas se diferencian por un identificador único.
  - Las compras tienen un precio propio, para contemplar los distintos tipos de usuario.

El sistema debe ofrecer la posibilidad de administrar cada uno de los puntos mencionados anteriormente, por lo tanto se lleva a cabo el desarrollo de un sistema que se adapta y permita realizar esta gestión de forma informatizada y de manera sencilla.

La orientación del sistema está principalmente enfocada en los empleados de la empresa. Los mismos deben entrar al sistema y desde la interfaz principal tienen acceso a todos los módulos, permitiendo realizar lo mencionado anteriormente centralizado en un solo sistema.

Existieron algunas decisiones en cuanto a diseño que se investigaron, plantearon y decidieron en el equipo para lograr el sistema requerido:

- Identificadores:
  - Los identificadores de los distintos 'objetos' son importantes para poder filtrar por los mismos y realizar operaciones individualmente.
  - Para las compras, existe un identificador propio (id\_compra)
  - Para los clientes, basta con tomar el documento de identidad como forma de identificación (CI).
  - Para los viajes, existe un identificador numérico que lo define.

- Base de datos:
  - La base de datos es el centro de toda la información del sistema, la decisión de la misma se basó principalmente tomando en cuenta el rendimiento y el tiempo de respuesta.
  - También se tomaron en cuenta los conocimientos previos de los integrantes para optar por MySQL.
- Front-End:
  - El front end es el conjunto de código que define la interacción con el usuario, tanto gráfica como lógicamente.
  - La justificación que impulsó que el equipo opte por ReactJS, es el conocimiento previo de la herramienta, y la sencillez a la hora de comunicarse a través de REST APIs.
- Estructura del proyecto y extensiones:
  - Para facilitar la creación de APIs desde Spring, existen infinitas herramientas que ayudan a acelerar el proceso.
  - En el presente proyecto se utilizan distintas técnicas (anotaciones, entidades, repositorios, servicios) para mejorar el proceso de desarrollo.

## 4. Herramientas utilizadas.

Durante el desarrollo del proyecto, se utilizaron distintas tecnologías y marcos de trabajo para llevar adelante el desarrollo:

### **Backend:**

Para el desarrollo del backend, se tomó ventaja del reconocido lenguaje de programación **JAVA**, incluyendo algunas extensiones y frameworks que permiten agilizar el trabajo. JAVA es una herramienta muy utilizada en ambientes de back-end debido a su gran capacidad para interactuar con bases de datos y sus diversas formas de acoplarse con otros sistemas.

Por otra parte, JAVA, por sí mismo no es capaz de trabajar con sistemas desarrollados en otros lenguajes, por lo tanto se toma la decisión de implementar el marco de trabajo Spring para manejar las peticiones desde el front-end y resguardar la información.

Spring es un framework de uso recurrente por los desarrolladores principalmente por la capacidad del mismo de trabajar con REST APIs, permitiendo esto que el sistema desarrollado en Java pueda actuar como back end de cualquier interfaz server-side.

### **Frontend:**

Adicionalmente, el frontend significó la utilización de una tecnología hermana de la mencionada anteriormente, JavaScript. Este lenguaje es sumamente reconocido a nivel mundial y su uso por parte de desarrolladores se encuentra en sumo crecimiento. Actúa muy bien del lado del servidor y permite desarrollar sistemas que trabajen con la lógica del software de manera cercana al usuario.

Para complementar el lenguaje JavaScript, se acordó incorporar la utilización del marco de trabajo ReactJS para poder trabajar con interfaces gráficas de una forma más modular y separando en componentes aquellas secciones repetidas.

React permite desarrollar utilizando el lenguaje anteriormente mencionado, además de añadir la posibilidad de gestionar hojas de estilos y componentes adicionales.

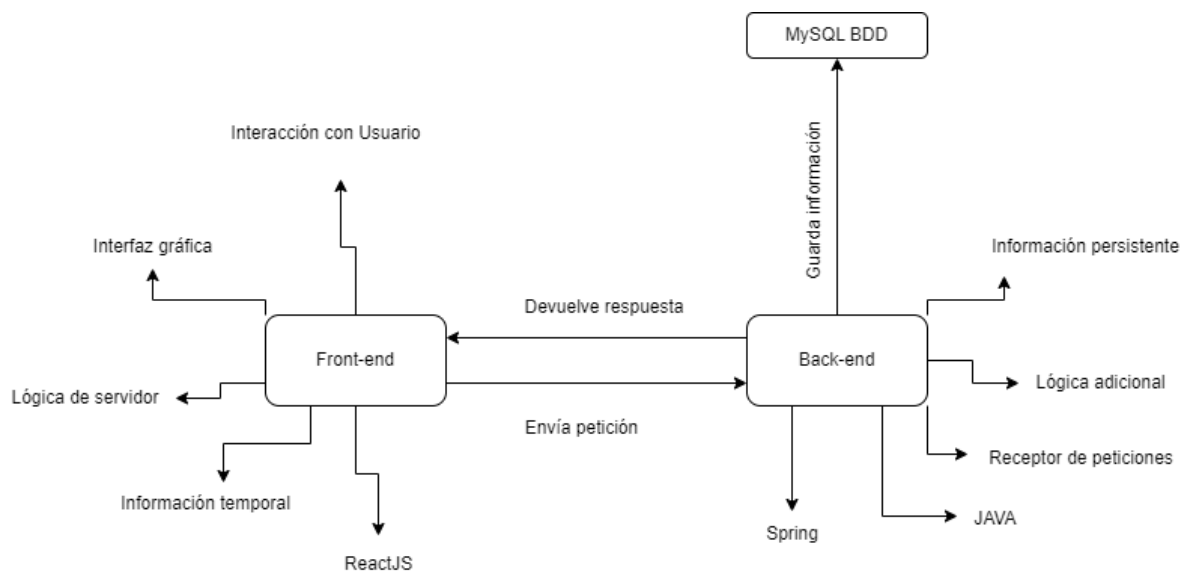


Entre estos componentes adicionales, podemos resaltar aquellos provenientes de la librería MaterialUI y React-Bootstrap que funcionaron como gran ayuda a la hora de desarrollar la interfaz.

### API Rest:

Las llamadas de API son fundamentales en el funcionamiento del sistema, ya que estas permiten la comunicación entre los dos grande módulos mencionados anteriormente (BACKEND y FRONTEND). Las mismas no dependen de ningún lenguaje de programación común y se comunican a través de objetos convertidos en cadenas de texto, permitiendo así manejar atributos de las entidades tanto desde un lado como del otro.

A continuación se agrega un diagrama con la estructura del sistema y el sector que ocupa cada herramienta mencionada.



## 5. Alcances obtenidos a consideración del equipo.

### Rúbrica con puntos a evaluar.

- Mostrar un dashboard administrativo en ambiente Web o Mobile (sencillo) que permita trabajar con planes de viaje y clientes:
  - Consideramos que cumplimos este punto, ya que la aplicación se desarrolló en un ambiente web responsive, donde se puede manipular tablas de planes y clientes para agregar, modificar o eliminar cualquiera de estos.
- Dar de alta, eliminar y modificar planes de viaje. Cada uno de ellos tiene un solo destino (máx. 20 dígitos alfanuméricos), fecha, modalidad (solamente pueden ser aérea, marítima o terrestre), precio en USD, carrusel de fotos (opcional).
  - El programa permite manipular planes de viaje con sus respectivas restricciones como que cada viaje tiene un solo destino o la modalidad que solo puede ser aérea, marítima o terrestre.
  - En cuanto a administración de imágenes (cargas y descargas), no se llegó a cumplir por completo, debido a su criticidad baja y su opcionalidad, no se tomó como una funcionalidad de las más importantes y surgieron dificultades a la hora de su implementación. Como alternativa, se implementó un carrousel de imágenes en cada uno de los planes, con las mismas imágenes.
- Dar de alta, eliminar y modificar clientes. Los clientes tienen CI (máx. 8 dígitos, mín. 7 dígitos, sin puntos ni guiones), nombre (máx. 30 dígitos alfanuméricos), apellido (máx. 30 dígitos alfanuméricos), email (máx. 30 dígitos alfanuméricos), planes comprados (si no tiene ninguno se deberá indicar con un mensaje que "no tiene planes comprados").
  - Consideramos que este punto se cumplió en su totalidad. El sistema permite manipular clientes con las restricciones planteadas.
- Contratar o borrarse de un plan (o más) de viaje.
  - Cada cliente puede contratar o darse de baja de cualquier plan de viaje en cualquier momento por lo que consideramos que se cumplió con este punto.
- Listar los planes (viajes) de un cliente mostrando todos los atributos de cada viaje.
  - Seleccionando un cliente en el apartado de listados adicionales podemos ver todos los planes que ha contratado por lo que consideramos que este punto se cumplió.

- Listar el primer viaje que tendrá un cliente después de una fecha especificada.
  - Al igual que en el punto anterior, seleccionando un cliente en el apartado de listados adicionales y seleccionando una fecha se nos van a mostrar el plan más cercano después de esa fecha para ese cliente. Consideramos que cumplimos con este punto.
- Controlar que no se permite ingresar viajes (planes) con fechas anteriores a la fecha actual.
  - Si en el apartado de planes ingresamos una fecha anterior a la de ese día nos va a mostrar un error informando que no se pueden ingresar fechas anteriores a la actual. Consideramos que cumplimos con este punto.
- Controles de errores para ingreso de datos.
  - Teniendo en cuenta la letra, se hicieron todos los controles pertinentes para que los datos sean correctos por lo que consideramos que cumplimos con este punto.

## 6. Diagrama de clases.

