

<div style="text-align: center;">  <p><b>Universidad Tecnológica Nacional</b> <b>Facultad Regional Avellaneda</b></p> </div>									
<b>Materia: PROGRAMACIÓN III</b>									
<b>Apellido:</b>					<b>Fecha:</b>		<b>2C 2023</b>		
<b>Nombre:</b>					<b>Docente<sup>(a)</sup>:</b>		<b>F. Lippi / A. Friadenrich</b>		
<b>División:</b>					<b>Nota<sup>(a)</sup>:</b>				
<b>Legajo:</b>					<b>Firma<sup>(a)</sup>:</b>				
<b>Instancia<sup>(a)</sup>:</b>	<b>PP</b>	<b>x</b>	<b>RPP</b>		<b>SP</b>		<b>RSP</b>		<b>FIN</b>

Se debe realizar una aplicación para dar de ingreso con **foto del usuario/cliente**.

Los datos se persistirán en archivos (ej. txt, json, csv, etc.)

Se deben respetar los nombres de los archivos y de las clases.

Se debe crear **una clase en PHP** por cada entidad y los archivos PHP solo deben llamar a métodos de las clases.

#### 1-

A- **index.php**: Recibe todas las peticiones que realiza el cliente (utilizaremos Postman), y administra a qué archivo se debe incluir.

B- **ClienteAlta.php**: (por POST) se ingresa Nombre y Apellido, Tipo Documento, Nro. Documento, Email, Tipo de Cliente (individual o corporativo), País, Ciudad y Teléfono.

Se guardan los datos en el archivo **hoteles.json**, tomando un id autoincremental de 6 dígitos como Nro. de Cliente (emulado). Si el nombre y tipo ya existen, se actualiza la información y se agrega al registro existente.

completar el alta con imagen/foto del cliente, guardando la imagen con Número y Tipo de Cliente (ej.: NNNNNNTT) como identificación en la carpeta:

**/ImágenesDeClientes/2023.**

#### 2-

**ConsultarCliente.php**: (por POST) Se ingresa Tipo y Nro. de Cliente, si coincide con algún registro del archivo hoteles.json, retornar el país, ciudad y teléfono del cliente/s. De lo contrario informar si no existe la combinación de nro y tipo de cliente o, si existe el número y no el tipo para dicho número, el mensaje: "tipo de cliente incorrecto".

### 3-

a- **ReservaHabitacion.php**: (por POST) se recibe el Tipo de Cliente, Nro de Cliente, Fecha de Entrada, Fecha de Salida, Tipo de Habitación (Simple, Doble, Suite), y el importe total de la reserva. Si el cliente existe en **hoteles.json**, se registra la reserva en el archivo **reservas.json** con un id autoincremental). Si el cliente no existe, informar el error.

b- Completar la reserva con imagen de confirmación de reserva con el nombre: Tipo de Cliente, Nro. de Cliente e Id de Reserva, guardando la imagen en la carpeta **/ImagenesDeReservas2023**.

### 4- ConsultaReservas.php: (por GET)

Datos a consultar:

a- El total de reservas (importe) por tipo de habitación y fecha en un día en particular (se envía por parámetro), si no se pasa fecha, se muestran las del día anterior.

b- El listado de reservas para un cliente **en particular**.

c- El listado de reservas entre dos fechas **ordenado por fecha**.

d- El listado de reservas por **tipo de habitación**.

### 5- ModificarCliente.php (por PUT)

Debe recibir todos los datos propios de un cliente; si dicho cliente existe (comparar por Tipo y Nro. de Cliente) se modifica, de lo contrario informar que no existe ese cliente.

6- **CancelarReserva.php**: (por POST) se recibe el Tipo de Cliente, Nro de Cliente, y el Id de Reserva a cancelar. Si el cliente existe en **hoteles.json** y la reserva en **reservas.json**, se marca como cancelada en el registro de reservas. Si el cliente o la reserva no existen, informar el tipo de error.

### 7- AjusteReserva.php (por POST),

Se ingresa el número de reserva afectada al ajuste y el motivo del mismo. El número de reserva debe existir.

Guardar en el archivo **ajustes.json**

Actualiza en el estado de la reserva en el archivo **reservas.json**

## Agregados y modificaciones a ejecutar durante el parcial:

### 8- ClienteAlta.php:

- a- La validación del Cliente existente deberá hacerse considerando los atributos: **número de cliente** y **DNI** que deben ser únicos en el sistema.
- b- Se debe adecuar la lógica desarrollada para permitir el ingreso por parámetro opcional de **Modalidad de Pago**, que permita definir un modalidad de pago inicial como **Efectivo**.
- c- Se identificó una mejora funcional por la que el atributo "tipo de cliente" contendrá el Tipo Cliente siendo **INDI** para Individual, **CORPO** para Corporativo y el tipo Documento **DNI, LE, LC, PASAPORTE**.

Ejemplo: **INDI-DNI** ó **CORPO-PASAPORTE**

**IMPORTANTE:** verificar y adecuar todos los puntos funcionales donde estos cambios pueden impactar.

**9- BorrarCliente.php** (por DELETE), debe recibir un número el tipo y número de cliente y debe realizar la baja de la cuenta (soft-delete, no físicamente) y la foto relacionada a ese cliente debe moverse a la carpeta **/ImagenesBackupClientes/2023**.

### 10- ConsultaReservas.php: (por GET)

A la consulta de reservas ya existente, incorporar:

- a- El total cancelado (importe) por tipo de **cliente** y **fecha** en un día en particular (se envía por parámetro), si no se pasa fecha, se muestran las del día anterior.
- b- El listado de cancelaciones para un cliente en **particular**.
- c- El listado de cancelaciones entre **dos fechas ordenado por fecha**.
- d- El listado de cancelaciones **por tipo de cliente**.
- e- El listado de todas las operaciones (reservas y cancelaciones) por usuario.
- f- El listado de Reservas **por tipo de modalidad**.

## Agregados y modificaciones a ejecutar en 2do parcial:

- A.** Se deben realizar todas las persistencias en BD utilizando PDO y sus métodos para dicho fin (**mandatorio**)
- B.** Se debe realizar el enrutamiento de peticiones en el index utilizando Slim Framework 4 (**mandatorio**)
- C.** Llevar el diseño de la solución al patrón MVC (Model View Controller)

**Código obsoleto, copiado y pegado que no tenga utilidad (-1 punto).**

**Se pueden bajar plantillas de internet o traer código hecho, pero en ningún caso se debe incluir código obsoleto o que no cumpla ninguna función dentro del parcial.**

**Se deberá incluir una colección de Postman que contenga todas las peticiones para cada punto.**

**Las calificaciones deberían ser en función de los cambios solicitados sobre este enunciado.**