



INSTITUTO TECNOLÓGICO DE ESTUDIOS SUPERIORES DE ZAMORA

2do Avance del proyecto -

MIS Roberto Suarez Zinzun - Arquitectura de servicios

Integrantes

- | | |
|-----------------------------|----------|
| - Abraham Pardo Orozco | 19010384 |
| - Ana Aurelia Galvan Cortes | 19010365 |
| - Luis Miguel Ríos Mendoza | 19010390 |
| - Reyes Sanchez Luis Angel | 19010388 |

Nombre del Servicio	Servicio de ventas de dulceria
Tipo de servicio	Este servicio es de tipo Entidad
URI	/dulceria
Protocolo	HTTP
Plataforma de desarrollo	Se desarrollara con el lenguaje de programación: Python v 3.11.2, también se trabajará con Flask v 2.2.3
Operaciones expuestas	Agregar nuevos productos a la lista Modificar la información de un producto existente Eliminar un producto Consultar la lista de productos

OPERACIÓN: Agregar nuevos productos a la lista(dulcería)

Elemento	Valor
Actor	Empleado encargado de administración de dulcería
Método de acceso	post
Validación	Validar la información del nuevo producto y agregarlo a la lista de productos disponibles.

Entrada	<pre> { "_id": "ObjectId", "fecha_venta": "Date", "id_empleado": "String", "nombre_empleado": "String", "productos_vendidos": [{ "producto_id": "ObjectId", "cantidad": "Number", "precio_unitario": "Number" }, { "producto_id": "ObjectId", "cantidad": "Number", "precio_unitario": "Number" }], "total_venta": "Number" } </pre>
Salida	<pre> { "message": "Se ingreso correctamente" } </pre>

OPERACIÓN: Modificar la información de un producto existente(dulceria)

Elemento	Valor
Actor	Empleado encargado de administración de dulcería
Método de acceso	put
Validación	Para modificar la información de un producto existente: validar la información del producto a modificar y actualizarla en la lista de productos disponibles.

Entrada	<pre>{ "_id": "ObjectId", "fecha_venta": "Date", "id_empleado": "String", "nombre_empleado": "String", "productos_vendidos": [{ "producto_id": "ObjectId", "cantidad": "Number", "precio_unitario": "Number" }, { "producto_id": "ObjectId", "cantidad": "Number", "precio_unitario": "Number" }], "total_venta": "Number" }</pre>
Salida	<pre>{ "message": "Se modificó correctamente" }</pre>

OPERACIÓN: Eliminar un producto (dulcería)

Elemento	Valor
Actor	Empleado encargado de administración de dulcería
Método de acceso	delete
Validación	Para eliminar un producto: validar la información del producto a eliminar y eliminarlo de la lista de productos disponibles.
Entrada	IdSolicitudes(int)-(/dulceria/3)

Salida	<pre>{ "message": "Se eliminó correctamente" }</pre>
--------	--

OPERACIÓN: Consultar la lista de productos(dulcería)

Elemento	Valor
Actor	Empleado encargado de administración de dulcería
Método de acceso	get
Validación	Para consultar la lista de productos: retornar la lista de productos disponibles.
Entrada	Idinventario(int)-(/dulceria/7)
Salida	<pre>{ "_id": "ObjectId", "fecha_venta": "Date", "id_empleado": "String", "nombre_empleado": "String", "productos_vendidos": [{ "producto_id": "ObjectId", "cantidad": "Number", </pre>

	<pre> "precio_unitario": "Number" },{ "producto_id": "ObjectId", "cantidad": "Number", "precio_unitario": "Number" }}, "total_venta": "Number" } </pre>
--	---

Nombre del Servicio	Servicio de ventas de boletos
Tipo de servicio	Este servicio es de tipo Entidad
URI	/boletos
Protocolo	HTTP
Plataforma de desarrollo	Se desarrollara con el lenguaje de programación: Python v 3.11.2, también se trabajará con Flask v 2.2.3
Operaciones expuestas	<p>Vender un nuevo boleto para una función específica.</p> <p>Modificar la información de un boleto existente.</p> <p>Cancelar la venta de un boleto.</p> <p>Consultar la lista de boletos vendidos para una función específica.</p>

OPERACIÓN: Agregar nuevos productos a la lista(boletos)

Elemento	Valor
Actor	Empleado encargado de administración de boletos
Método de acceso	post
Validación	Validar la información del nuevo producto y agregarlo a la lista de productos disponibles.

Entrada

```
{  
  "_id": ObjectId(),  
  "idVenta": Number,  
  "id_empleado": String,  
  "nombre_empleado": String,  
  "numBoletosVendidos": Number,  
  "totalVenta": Double,  
  "fechaVenta": String,  
  "funcion": {  
    "idFuncion": Number,  
    "horaInicio": String,  
    "horaFin": String,  
    "fecha": String,  
    "pelicula_id": ObjectId,  
    "pelicula_nombre": String,  
    "sala": {  
      "idSala": Number,  
      "numSala": Number,  
      "capacidad": Number,  
      "tipo": String  
    }  
  },  
}
```


Salida	{ "message": "Se ingreso correctamente" }
--------	---

OPERACIÓN: Modificar la información de un producto existente(boletos)

Elemento	Valor
Actor	Empleado encargado de administración de boletos
Método de acceso	put
Validación	Para modificar la información de un producto existente: validar la información del producto a modificar y actualizarla en la lista de boletos disponibles.

Entrada

```
{  
  
  "_id": ObjectId(),  
  "idVenta": Number,  
  "id_empleado": String,  
  "nombre_empleado": String,  
  "numBoletosVendidos": Number,  
  "totalVenta": Double,  
  "fechaVenta": String,  
  "funcion": {  
    "idFuncion": Number,  
    "horaInicio": String,  
    "horaFin": String,  
    "fecha": String,  
    "pelicula_id": ObjectId,  
    "pelicula_nombre": String,  
    "sala": {  
      "idSala": Number,  
      "numSala": Number,  
      "capacidad": Number,  
      "tipo": String  
    }  
  },  
}
```

Salida	<pre>{ "message": "Se modifique correctamente" }</pre>
--------	--

OPERACIÓN: Eliminar un producto (boletos)

Elemento	Valor
Actor	Empleado encargado de administración de boletos
Método de acceso	delete
Validación	Para eliminar un producto: validar la información del boleto a eliminar y eliminarlo de la lista de boletos disponibles.

Entrada

```
{  
  
  "_id": ObjectId(),  
  "idVenta": Number,  
  "id_empleado": String,  
  "nombre_empleado": String,  
  "numBoletosVendidos": Number,  
  "totalVenta": Double,  
  "fechaVenta": String,  
  "funcion": {  
    "idFuncion": Number,  
    "horaInicio": String,  
    "horaFin": String,  
    "fecha": String,  
    "pelicula_id": ObjectId,  
    "pelicula_nombre": String,  
    "sala": {  
      "idSala": Number,  
      "numSala": Number,  
      "capacidad": Number,  
      "tipo": String  
    }  
  },  
}
```

Salida	<pre>{ "message": "Se eliminó correctamente" }</pre>
--------	--

OPERACIÓN: Consultar la lista de productos(boletos)

Elemento	Valor
Actor	Empleado encargado de administración de boletos
Método de acceso	get
Validación	Para consultar la lista de productos: retornar la lista de boletos disponibles.
Entrada	Idinventario(int)-(/boletos/7)
Salida	<pre>{ "_id": ObjectId(), "idVenta": Number, "id_empleado": String, "nombre_empleado": String, "numBoletosVendidos": Number, "totalVenta": Double, "fechaVenta": String, "funcion": {</pre>

	<pre>"idFuncion": Number, "horaInicio": String, "horaFin": String, "fecha": String, "pelicula_id": ObjectId, "pelicula_nombre": String, "sala": { "idSala": Number, "numSala": Number, "capacidad": Number, "tipo": String }}, "Lugares": [{ "\$ref": "String", "\$id": "ObjectId" }, { "\$ref": "String", "\$id": "ObjectId" }]</pre>
--	--

Nombre del Servicio	Servicio de Inventario
Tipo de servicio	Este servicio es de tipo Tarea
URI	/inventario
Plataforma de desarrollo	Se desarrollara con el lenguaje de programación: Python v 3.11.2, también se trabajará con Flask v 2.2.3
Operaciones expuestas	Consultar Inventario Actualizar inventario Ingresar Eliminación

Operación: Consultar (Inventario)

Elemento	Valor
Método de acceso	Get
Ruta	/Inventario
Protocolo	http
Actor(es)	Empleado Contabilidad
Validaciones	- Validar que no haya productos repetidos
Entrada	Idinventario(int)-(/Inventario/7)
Proceso	A base del número de ID mostrará la información requerida o si no se especifica mostrará toda la información del inventario
Salida	{ "_id": "ObjectId", "nombre": "String", "descripcion": "String", "categoria": "String", "presentacion": "Number", "precio": "Number", "fecha_vencimiento": "Date", "imagen_url": "String",

	"disponibilidad": "Number" }
--	---------------------------------

Operación: Actualizar (Inventario)

Elemento	Valor
Método de acceso	Put
Ruta	/Inventario
Protocolo	HTTP
Actores(es)	Contabilidad
validaciones	-Validar que el formato de la información sea el correcto, si se pide numero validar que realmente sea un número. - Validar que no haya productos repetidos -Validar que la fecha en que se ingreso producto no sea la misma para evitar la caducidad
Entrada	<pre>{ "_id": "ObjectId", "nombre": "String", "descripcion": "String", "categoria": "String", "presentacion": "Number", "precio": "Number", "fecha_vencimiento": "Date", "imagen_url": "String", "disponibilidad": "Number" }</pre>
Proceso	A base del número de ID, actualiza la información requerida
Salida	<pre>{ "_id": "ObjectId", "nombre": "String",</pre>

	<pre>"descripcion": "String", "categoria": "String", "presentacion": "Number", "precio": "Number", "fecha_vencimiento": "Date", "imagen_url": "String", "disponibilidad": "Number" }</pre>
--	--

Operación: Ingresar (inventario)

Elemento	Valor
Método de acceso	Post
Ruta	/Inventario
Protocolo	HTTP
Actor(es)	Contabilidad
validaciones	<p>-Validar que el formato de la información sea el correcto, si se pide numero validar que realmente sea un número.</p> <p>- Validar que no haya productos repetidos</p> <p>-Validar que la fecha en que se ingreso producto no sea la misma para evitar la caducidad</p>
Entrada	<pre>{ "_id": "ObjectId", "nombre": "String", "descripcion": "String", "categoria": "String", "presentacion": "Number", "precio": "Number", "fecha_vencimiento": "Date", "imagen_url": "String", "disponibilidad": "Number" }</pre>

Proceso	Añadir el elemento al inventario con valores proporcionado en la entrada
Salida	<pre>{ "message": "Se ingreso correctamente" }</pre>

Operación: Eliminar (inventario)

Elemento	Valor
Método de acceso	Delete
Ruta	/Inventario
Protocolo	HTTP
Actor(es)	Contabilidad
validación	-Pedir una confirmación de la eliminación -Verificar que el elemento que se desea eliminar
Entrada	IdSolicitudes(int)-(/Inventario/3)
Proceso	Elimina el elemento con el número de ID especificado
Salida	<pre>{ "message": "Se eliminó correctamente" }</pre>

Nombre del Servicio	Servicio de Películas
Tipo de servicio	Este servicio es de tipo Tarea
URI	/Películas
Plataforma de desarrollo	Se desarrollara con el lenguaje de programación: Python v 3.11.2, también se trabajará con Flask v 2.2.3
Operaciones expuestas	Consultar Películas Actualizar Películas Ingresar Eliminación

Operación: Consultar (Películas)

Elemento	Valor
Método de acceso	Get
Ruta	/Películas
Protocolo	http
Actor(es)	Empleado
Validaciones	- Validar que no haya películas repetidas
Entrada	IdPelícula(int)-(/Películas/7)
Proceso	A base del número de ID mostrará la información requerida o si no se especifica mostrará toda la información del inventario
Salida	<pre>{ "_id": ObjectId(), "idPelícula": Number, "nombre": String, "genero": String, "clasificacion": String, "duracion": Number, "nacionalidad": String, "reparto": String, "sinopsis": String }</pre>

--	--

Operación: Actualizar (Películas)

Elemento	Valor
Método de acceso	Put
Ruta	/Peliculas
Protocolo	HTTP
Actores(es)	Gerente
validaciones	-Validar que el formato de la información sea el correcto, si se pide número validar que realmente sea un número.
Entrada	<pre>{ "_id": ObjectId(), "idPelicula": Number, "nombre": String, "genero": String, "clasificacion": String, "duracion": Number, "nacionalidad": String, "reparto": String, "sinopsis": String }</pre>
Proceso	A base del número de ID, actualiza la información requerida
Salida	<pre>{ "_id": ObjectId(), "idPelicula": Number, "nombre": String, "genero": String, "clasificacion": String, "duracion": Number, "nacionalidad": String, "reparto": String, "sinopsis": String }</pre>

Operación: Ingresar (Películas)

Elemento	Valor
Método de acceso	Post
Ruta	/Peliculas
Protocolo	HTTP
Actor(es)	Gerente
validaciones	-Validar que el formato de la información sea el correcto, si se pide numero validar que realmente sea un número. - Validar que no haya Películas repetidas
Entrada	{ "_id": ObjectId(), "idPelicula": Number, "nombre": String, "genero": String, "clasificacion": String, "duracion": Number, "nacionalidad": String, "reparto": String, "sinopsis": String }
Proceso	Añadir el elemento al inventario con valores proporcionado en la entrada
Salida	{ "message": "Se ingreso correctamente" }

Operación: Eliminar (Peliculas)

Elemento	Valor
Método de acceso	Delete
Ruta	/Inventario
Protocolo	HTTP
Actor(es)	Gerente
validación	-Pedir una confirmación de la eliminación -Verificar que el elemento que se desea eliminar
Entrada	IdSolicitudes(int)-(/Peliculas/3)
Proceso	Elimina el elemento con el número de ID especificado
Salida	{ "message": "Se eliminó correctamente" }

Nombre del Servicio	Servicio de planilla de empleados
Tipo del Servicio	Entidad
URL	/empleados
Protocolo	HTTP
Plataforma de Desarrollo	Se desarrollara con el lenguaje de programación: Python v 3.11.2, también se trabajará con Flask v 2.2.3
Operaciones expuestas	Agregar Empleado Eliminar Empleado Consultar Empleado Modificar Empleado
Validaciones:	

OPERACION AGREGAR EMPLEADO

Elemento	Valor
Actor	Empleado
Método de acceso	Post
Validación	Una confirmación de qué si se agregó
Entrada	<pre>{ "idEmpleado": "String" , "nombre": "String", "direccion": "String", "telefono": "String", "rfc": "String", "nss": "String", "puesto": "String", "fechaNacimiento": "String", "fechaIngreso": "String" , "sexo": "String" }</pre>

Salida	{ "message": "Se ingreso correctamente" }
--------	---

OPERACION Eliminar EMPLEADO

Elemento	Valor
Actor	Empleado
Método de acceso	Delete
Validación	Una confirmación de qué si se eliminó
Entrada	IdEmpleado(int)-(/Empleado/3)
Salida	{ "message": "Se eliminó correctamente" }

OPERACION CONSULTAR EMPLEADO

Elemento	Valor
Actor	Empleado
Método de acceso	Get
Validación	- Validar que no haya datos repetidos
Entrada	IdEmpleado(int)-(/Empeado/7)
Salida	{ "idEmpleado":"String", "nombre":"String", "direccion":"String", "telefono":"String", }

	<pre>"rfc":"String", "nss":"String", "puesto":"String", "fechaNacimiento":"String", "fechaIngreso":"String", "sexo":"String" }</pre>
--	--

OPERACION Modificar EMPLEADO

Elemento	Valor
Actor	Empleado
Método de acceso	Put
Validación	Una confirmación de qué si se modificó
Entrada	<pre>{ "idEmpleado":"String", "nombre":"String", "direccion":"String", "telefono":"String", "rfc":"String", "nss":"String", "puesto":"String", "fechaNacimiento":"String", "fechaIngreso":"String", "sexo":"String" }</pre>
Salida	<pre>{ "message": "Se actualizo correctamente" }</pre>

Nombre del Servicio	Servicio de Salas
Tipo del Servicio	Entidad
URL	/Lugares
Protocolo	HTTP
Plataforma de Desarrollo	Se desarrollara con el lenguaje de programación: Python v 3.11.2, también se trabajará con Flask v 2.2.3
Operaciones expuestas	Agregar Sala Consultar Sala Eliminar Sala Modificar Sala
Validaciones:	

Operación Agregar Lugares

Elemento	Valor
Actor	Sala
Método de acceso	Post
Validación	Una confirmación que si se agregó
Entrada	<pre>{ "_id": "ObjectId", "sala": { "idSala": "Number", "numSala": "Number", "capacidad": "Number", "tipo": "String", "lugares": [{ "fila": "String", "numeroAsiento": "Number", "tipoAsiento": "String", "estaDisponible": "Boolean" }, {</pre>

	<pre> "fila": "String", "numeroAsiento": "Number", "tipoAsiento": "String", "estaDisponible": "Boolean" }] } } </pre>
Salida	<pre> { "message": "Se agregó correctamente" } </pre>

Operación Consultar Sala

Elemento	Valor
Actor	Sala
Método de acceso	Get
Validación	- Validar que no haya datos repetidos
Entrada	_id(int)-(/_id/7)
Salida	<pre> { "_id": "ObjectId", "sala": { "idSala": "Number", "numSala": "Number", "capacidad": "Number", "tipo": "String", "lugares": [{ "fila": "String", "numeroAsiento": "Number", "tipoAsiento": "String", "estaDisponible": "Boolean" }, { "fila": "String", </pre>

	<pre> "numeroAsiento": "Number", "tipoAsiento": "String", "estaDisponible": "Boolean" }] } } </pre>
--	--

Operacion Eliminar Sala

Elemento	Valor
Actor	Sala
Método de acceso	Delete
Validación	Una confirmación qué si se eliminó
Entrada	_Id(int)-(/_id/3)
Salida	<pre> { "message": "Se Eliminó correctamente" } </pre>

Operación Modificar Sala

Elemento	Valor
Actor	Sala
Método de acceso	Put
Validación	Validar que el formato de la información sea el correcto, si se pide numero validar que realmente sea un número.

Entrada	<pre> { "_id": "ObjectId", "sala": { "idSala": "Number", "numSala": "Number", "capacidad": "Number", "tipo": "String", "lugares": [{ "fila": "String", "numeroAsiento": "Number", "tipoAsiento": "String", "estaDisponible": "Boolean" }, { "fila": "String", "numeroAsiento": "Number", "tipoAsiento": "String", "estaDisponible": "Boolean" }] } } </pre>
Salida	<pre> { "message": "Se actualizo correctamente" } </pre>