



Programa educativo

INGENIERÍA EN SISTEMAS
COMPUTACIONALES

Grupo

Especialidad DFS

Nombre de la materia

ARQUITECTURA DE SERVICIOS

Nombre del alumno

ANA PAULA BARAJAS CAMPOS
JAZMÍN BERENICE VEGA GÓMEZ

Nombre del trabajo

Segundo Avance del Proyecto

Nombre del Profesor

ROBERTO SUÁREZ ZINZUN

Fecha

23 de Marzo de 2023

SEGUNDO AVANCE DEL PROYECTO. (Documentación de Servicios).

- SERVICIOS DE EDIFICIOS:

Nombre del servicio: Edificios REST

Tipo de servicio: Entidad

URI: /Edificios

Plataforma de desarrollo: Python 3.7, Flask 2.2.3

Operaciones:

1. Registro de un nuevo edificio.
2. Consulta de todos los edificios.
3. Consulta de edificio por su id.
4. Actualización de datos de un edificio.
5. Eliminación de un edificio.

Operación: Registrar Edificio

Elemento	Valor
Método de Acceso	POST
Actor(es)	Administrador
Proceso	Registro de un edificio en la base de datos para poder asociar las salas que hay en él.
Entrada	{ "nombreEdificio": String, "descripción": String }
Salida	{ "estatus": String, "mensaje": String }

Operación: Consultar Edificios

Elemento	Valor
Método de Acceso	GET
Actor(es)	Alumno, docente, coordinador, administrador
Proceso	Consulta de todos los edificios existentes en la base de datos.
Entrada	N/A
Salida	<pre>{ "estatus": String, "mensaje": String, "edificios": [{ "id": int, "nombreEdificio": String, "descripción": String }] }</pre>

Operación: Consultar Edificio individual

Elemento	Valor
Método de Acceso	GET
Actor(es)	Alumno, docente, coordinador, administrador
Proceso	Consulta de un edificio con base a su id.
Entrada	idEdificio: int
Salida	<pre>{ "estatus": String, "mensaje": String, "edificio": { "id": int, "nombreEdificio": String, "descripción": String } }</pre>

Operación: Modificar Edificio

Elemento	Valor
Método de Acceso	PUT
Actor(es)	Administrador
Proceso	Modificación de los datos de un edificio. Validaciones: Verificar que el edificio que se desea modificar exista en la base de datos.
Entrada	idEdificio: int nombreEdificio: String (en caso de querer modificarlo) descripción: String (en caso de querer modificarlo)
Salida	{ "estatus": String, "mensaje": String, }

Operación: Eliminar Edificio

Elemento	Valor
Método de Acceso	DELETE
Actor(es)	Administrador
Proceso	Eliminación de un edificio. Validaciones: Verificar que el edificio que se desea eliminar exista en la base de datos.
Entrada	idEdificio: int
Salida	{ "estatus": String, "mensaje": String, }

- **SERVICIOS DE SALAS:**

Nombre del servicio: Salas REST

Tipo de servicio: Entidad

URI: /Salas

Plataforma de desarrollo: Python 3.7, Flask 2.2.3

Operaciones:

1. Registro de una nueva sala.
2. Consulta de todas las salas.
3. Consulta de sala por su id.
4. Consulta de disponibilidad de sala.
5. Actualización de datos de una sala.
6. Eliminación de una sala.

Operación: Registrar Sala

Elemento	Valor
Método de Acceso	POST
Actor(es)	Administrador
Proceso	Registro de una nueva sala en la base de datos.
Entrada	<pre>{ "nombreSala": String, "descripción": String, "capacidad": int "edificio": int }</pre>
Salida	<pre>{ "estatus": String, "mensaje": String }</pre>

Operación: Consultar Salas

Elemento	Valor
Método de Acceso	GET
Actor(es)	Alumno, docente, coordinador, administrador
Proceso	Consulta de todas las salas existentes en la base de datos.
Entrada	N/A
Salida	<pre>{ "estatus": String, "mensaje": String, "salas": [{ "id": int, "nombreSala": String, "descripción": String, "estatus": String, "capacidad": int, "edificio": int }] }</pre>

Operación: Consultar Sala individual

Elemento	Valor
Método de Acceso	GET
Actor(es)	Alumno, docente, coordinador, administrador
Proceso	Consulta de una sala con base a su id.
Entrada	idSala: int
Salida	<pre>{ "estatus": String, "mensaje": String, "sala": { "id": int, "nombreSala": String, "descripción": String, </pre>

	<pre> "estatus": String, "capacidad": int, "edificio": int } } }</pre>
--	--

Operación: Consulta de disponibilidad de Sala.

Elemento	Valor
Método de Acceso	GET
Actor(es)	Alumno, docente, coordinador, administrador
Proceso	Mostrar el estatus de la sala,
Entrada	idSala: int
Salida	<pre> { "estatus": String, "mensaje": String, "sala": { "id": int, "estatus": String } }</pre>

Operación: Modificar Sala

Elemento	Valor
Método de Acceso	PUT
Actor(es)	Administrador
Proceso	Modificación de los datos de una sala. Validaciones: Verificar que la sala que se desea modificar exista en la base de datos.
Entrada	idSala: int nombreSala: String (en caso de querer modificarlo) descripción: String (en caso de querer modificarlo) estatus: String (en caso de querer modificarlo) capacidad: int (en caso de querer modificarlo) edificio: int (en caso de querer modificarlo)

Salida	{ “estatus”: String, “mensaje”: String, }
---------------	--

Operación: Eliminar Sala

Elemento	Valor
Método de Acceso	DELETE
Actor(es)	Administrador
Proceso	Eliminación de una sala. Validaciones: Verificar que la sala que se desea eliminar exista en la base de datos.
Entrada	idSala: int
Salida	{ “estatus”: String, “mensaje”: String, }

- **SERVICIOS DEL MOBILIARIO:**

Nombre del servicio: Mobiliario REST

Tipo de servicio: Entidad

URI: /Mobiliario

Plataforma de desarrollo: Python 3.7, Flask 2.2.3

Operaciones:

1. Registro de mobiliario.
2. Consulta de mobiliario por sala
3. Actualizar mobiliario
4. Eliminar Mobiliario

Operación: Registrar Mobiliario

Elemento	Valor
Método de Acceso	POST
Actor(es)	Administrador
Proceso	Registro de mobiliario.
Entrada	<pre>{ "id": int, "nombre": String, "descripcion": String, "cantidad": int, "sala": int }</pre>
Salida	<pre>{ "estatus": String, "mensaje": String }</pre>

Operación: Consulta de mobiliario por sala

Elemento	Valor
Método de Acceso	GET
Actor(es)	Alumno, docente, coordinador, administrador
Proceso	Consultar/mostrar el mobiliario que existe en determinada sala.
Entrada	idSala: int
Salida	<pre>{ "estatus": String, "mensaje": String, "mobiliario":[{ "id":int, "nombre":String, "descripcion":String, "cantidad":int, "sala":int }] }</pre>

Operación: Actualizar Mobiliario

Elemento	Valor
Método de Acceso	PUT
Actor(es)	Administrador
Proceso	<p>Modificación de o los datos del mobiliario de la sala que ya se seleccionó previamente.</p> <p>Se validará que el mobiliario a modificar exista en la sala, en la Base de Datos.</p>
Entrada	<p>idMobiliario: int</p> <p>nombre: String (en caso de querer modificarlo)</p> <p>descripción: String (en caso de querer modificarlo)</p> <p>cantidad: int (en caso de querer modificarlo)</p>
Salida	<pre>{ "estatus": String, "mensaje": String,</pre>

	}
--	---

Operación: Eliminar Mobiliario

Elemento	Valor
Método de Acceso	DELETE
Actor(es)	Administrador
Proceso	Se eliminará el mobiliario (de la sala que se seleccionó). Se validará que el mobiliario a eliminar exista en la sala, en la Base de Datos.
Entrada	idMobiliario: int
Salida	{ "estatus": String, "mensaje": String, }

- **SERVICIOS DE RESERVAS:**

Nombre del servicio: Reservas REST

Tipo de servicio: Entidad

URI: /Reservas

Plataforma de desarrollo: Python 3.7, Flask 2.2.3

Operaciones:

1. Agregar una Reserva.
2. Consulta de las reservaciones existentes.
3. Consulta de reserva por su id.
4. Actualización de la reserva realizada.
5. Eliminar Reserva.

Operación: Agregar Reserva

Elemento	Valor
Método de Acceso	POST
Actor(es)	Alumno, docente, coordinador, administrador
Proceso	Registrar una reserva de sala en la base de datos. Validar que la sala donde se desea hacer la reserva esté disponible.
Entrada	{ "idReserva": String, "horaInicio": String, "horaFin": String, "fecha": Date, "estatus": String, "sala": int }
Salida	{ "estatus": String, "mensaje": String }

Operación: Consulta de las reservaciones existentes.

Elemento	Valor
Método de Acceso	GET
Actor(es)	Alumno, docente, coordinador, administrador
Proceso	Mostar los datos de las Reservas que se han hecho en la Base de Datos.
Entrada	N/A
Salida	<pre>{ "estatus": String, "mensaje": String, "reserva": [{ "idReserva":int, "horaInicio": String, "horaFin": String, "fecha": Date, "estatus": String, "sala": int }] }</pre>

Operación: Consulta de reserva por su id.

Elemento	Valor
Método de Acceso	GET
Actor(es)	Alumno, docente, coordinador, administrador
Proceso	Mostar la reserva filtrándola por su id. Validar que exista la reserva en la Base de Datos.
Entrada	idReserva: int
Salida	<pre>{ "estatus": String, "mensaje": String, "reserva": [{ "idReserva":int, "horaInicio": String,</pre>

	<pre> "horaFin": String, "fecha": Date, "estatus": String, "sala": int }] } </pre>
--	---

Operación: Actualización de la reserva realizada.

Elemento	Valor
Método de Acceso	PUT
Actor(es)	Alumno, docente, coordinador, administrador
Proceso	Modificación de o los datos de la reserva de la sala que ya se seleccionó previamente.
Entrada	idReserva: int horaInicio: String (en caso de querer modificarlo) horaFin: String (en caso de querer modificarlo) fecha: Date (en caso de querer modificarlo) estatus: String (en caso de querer modificarlo) sala: int (en caso de querer modificarlo)
Salida	<pre> { "estatus": String, "mensaje": String, } </pre>

Operación: Eliminar Reserva.

Elemento	Valor
Método de Acceso	DELETE
Actor(es)	Alumno, docente, coordinador, administrador
Proceso	Eliminar la reserva seleccionada (por su id). Validar que la reserva exista en la Base de Datos
Entrada	idReserva: int
Salida	<pre> { "estatus": String, "mensaje": String, } </pre>