



Segundo avance del proyecto



Nombre de los alumnos

Diana Isabel Aguilar Navarro

Daniel Alejandro Juárez Gaona

José Manuel Varela Hernández

Carrera

ISC

Grado y grupo

8-A

Nombre de la asignatura

Arquitectura de servicios

Nombre del Profesor

Roberto Suarez Zinzun

23-3-2023

Diseño del Modelo de la BD

En nuestro módulo de Proyecto optamos por usar el tipo de BD NoSQL.
Los archivos json de la estructura de los datos se encuentran en la carpeta:

ProyectoRest/BD/ModeloEntidades

Scripts de la BD

Los archivos de la creación de la BD se encuentran en la carpeta:

ProyectoRest/BD/CreacionBD.txt

Documentación de los servicios

Revisiones

Nombre:

Revisiones

Tipo de servicio:

Entidad ya que solo hace operaciones CRUD sobre una entidad.

URI:

/proyecto/revisiones

Protocolo:

Https

Plataforma de desarrollo:

- Python 3.11.2
- Flask 2.2.4

Operaciones expuestas:

- Agregar Revisión
 - **Actores:** Docentes
 - **Validaciones:** Que la fecha sea después de la publicación del proyecto.
 - **Tipo:** Post
 - **Entrada:**

```
{
    "_id": "<ObjectId>",
    "IdProyecto": "ref(Proyectos)",
    "FechaIni": "<date>",
    "FechaTer": "<date>",
    "Observaciones": [{
        "Comentario": "<String>",
        "Estatus": "<String>"
    }],
    "IdPersona": "ref(Persona)",
    "Estatus": "<String>"
}
```

- **Salida:**

```

    {
        "Estatus": "El estado de la respuesta",
        "Mensaje": "Mensaje exitoso o rechazado"
    }

```

- Consultar Revisión

- **Actores:** Docentes
- **Validaciones:** Ninguna
- **Tipo:** Get
- **Entrada:**

Agregar la *IDRevisiones* al final de la ruta quedando así:
/proyecto/revisiones/idrevisions

- **Salida:**

```

    {
        "Estatus": "El estado de la respuesta",
        "Mensaje": "Mensaje exitoso o rechazado"
    }

```

- Modificar Revisión

- **Actores:** Docentes
- **Validaciones:** Que los datos correspondan con los tipos necesarios
- **Tipo:** PUT
- **Entrada:**

```

    {
        "_id": "<ObjectId>",
        "IdProyecto": "ref(Proyectos)",
        "FechaIni": "<date>",
        "FechaTer": "<date>",
        "Observaciones": [{
            "Comentario": "<String>",
            "Estatus": "<String>"
        }],
        "IdPersona": "ref(Persona)",
        "Estatus": "<String>"
    }

```

- **Salida:**

```

    {
        "Estatus": "El estado de la respuesta",
        "Mensaje": "Mensaje exitoso o rechazado"
    }

```

- Eliminar Revisión

- **Actores:** Docentes
- **Validaciones:** Ninguna.
- **Tipo:** Delete
- **Entrada:**

```

    {
        "_id": "<ObjectId>",
    }

```

- **Salida:**

```

    {
        "Estatus": "El estado de la respuesta",
        "Mensaje": "Mensaje exitoso o rechazado"
    }

```

Entregables

Nombre:

Entregables

Tipo de servicio:

Entidad ya que solo hace operaciones CRUD sobre una entidad.

URI:

/proyecto/entregables

Protocolo:

Https

Plataforma de desarrollo:

- Python 3.11.2
- Flask 2.2.4

Operaciones expuestas:

- Agregar Entregables
 - **Actores:** Alumnos
 - **Validaciones:** Ninguna.
 - **Tipo:** Post
 - **Entrada:** {

"Fecha_Programada": "<date>",
"Fecha_entregado": "<date>",
"Observaciones": "<string>",
"Archivo": "<binary>"
}
○ **Salida:**
{
"Estatus": "El estado de la respuesta",
"Mensaje": "Mensaje exitoso o rechazado"
}
- Consultar Entregables
 - **Actores:** Alumnos y Docentes
 - **Validaciones:** Ninguna.
 - **Tipo:** Get
 - **Entrada:**
Agregar la *IDEntregables* al final de la ruta quedando así:
/proyecto/entregables/identregables
○ **Salida:**
{
"Estatus": "El estado de la respuesta",
"Mensaje": "Mensaje exitoso o rechazado",
Entregables: {
" _id": "<ObjectId>",
"idProyecto": "ref(Proyecto)",
"Fecha_Programada": "<date>",
"Fecha_entregado": "<date>",
"Observaciones": "<string>",
"Archivo": "<binary>"
}
}
- Modificar Entregables
 - **Actores:** Alumnos
 - **Validaciones:** Que los datos correspondan con los tipos necesarios.

- **Tipo:** Put
- **Entrada:**

```
{
  "_id": "<ObjectId>",
  "Fecha_Programada": "<date>",
  "Fecha_entregado": "<date>",
  "Observaciones": "<string>",
  "Archivo": "<binary>"
}
```
- **Salida:**

```
{
  "Estatus": "El estado de la respuesta",
  "Mensaje": "Mensaje exitoso o rechazado"
}
```
- **Eliminar Entregables**
 - **Actores:** Alumnos
 - **Validaciones:** Ninguna
 - **Tipo:** Delete
 - **Entrada:**

```
{
  "_id": "<ObjectId>",
}
```
 - **Salida:**

```
{
  "Estatus": "El estado de la respuesta",
  "Mensaje": "Mensaje exitoso o rechazado"
}
```

Proyectos

Nombre:

Proyectos

Tipo de servicio:

Entidad ya que solo hace operaciones CRUD sobre una entidad.

URI:

/proyecto/proyectos

Protocolo:

Https

Plataforma de desarrollo:

- Python 3.11.2
- Flask 2.2.4

Operaciones expuestas:

- **Agregar Proyectos**
 - **Actores:** Alumnos y Docentes.
 - **Validaciones:** La fecha de inicio debe ser antes de la fecha de terminación.
 - **Tipo:** Post
 - **Entrada:** Para registrar un proyecto se necesita los siguientes datos:


```
{
            "Nombre": "<string>",
            "Temas": ["<String>"],
            "Fecha_Inicio": "<Date>",
            "Fecha_Termina": "<Date>",
```

```

        "Participantes": [
            {
                "IdPersona": "Ref(Persona)",
                "Nombre": "<String>",
                "Tipo": "<String>"
            }
        ]
    }
}

```

- **Salida:**

```

{
    "Estatus": "El estado de la respuesta",
    "Mensaje": "Mensaje exitoso o rechazado"
}

```

- Consultar Proyectos

- **Actores:** Alumnos y Docentes.
- **Validaciones:** Ninguna,
- **Tipo:** Get
- **Entrada:** Para consultar un proyecto se necesita los siguientes datos:
Agregar la IDProyectos al final de la ruta quedando así:
/proyecto/proyectos/idproyectos
- **Salida:**

```

{

```

```

    "Estatus": "El estado de la respuesta",
    "Mensaje": "Mensaje exitoso o rechazado",
    "Proyectos": {
        "Nombre": "<string>",
        "Temas": ["<String>"],
        "Fecha_Inicio": "<Date>",
        "Fecha_Termina": "<Date>",
        "Participantes": [
            {
                "Nombre": "<String>",
                "Tipo": "<String>"
            }
        ]
    }
}

```

- Modificar Proyectos

- **Actores:** Alumnos y Docentes.
- **Validaciones:** La fecha de inicio debe ser antes de la fecha de terminación.
- **Tipo:** Put
- **Entrada:** Para modificar un proyecto se necesita los siguientes datos:

```

{
    "idProyecto": "<objectID>",
    "Nombre": "<string>",
    "Temas": ["<String>"],
    "Fecha_Inicio": "<Date>",
    "Fecha_Termina": "<Date>",
    "Participantes": [
        {
            "IdPersona": "Ref(Persona)",
            "Nombre": "<String>",

```

```

        "Tipo": "<String>"
      }
    ]
  }

```

```

    }
  ○ Salida:
    {
      "Estatus": "El estado de la respuesta",
      "Mensaje": "Mensaje exitoso o rechazado"
    }

```

- Eliminar Proyectos

- **Actores:** Alumnos y Docentes.
- **Validaciones:** Ninguna,
- **Tipo:** Delete
- **Entrada:** Para modificar un proyecto se necesita los siguientes datos:

```

    {
      "idProyecto": "<objectID>"
    }

```

```

  ○ Salida:
    {
      "Estatus": "El estado de la respuesta",
      "Mensaje": "Mensaje exitoso o rechazado"
    }

```