

Gestor de Tareas y Habitos

1. Eleccion del Tema/Dominio de la Aplicacion

Dominio: Productividad y manejo de tareas

Colecciones:

2. Identificacion de Entidades y Atributos

Usuario:

Seria la persona que maneja el perfil, esta persona va a tener acceso a poder ver, sus tareas, habitos, recordatorios y logros

id_usuario, nombre, apellido, email, fecha_registro

Tareas:

Es una tarea que tiene que hacer el usuario, esta tiene un nombre, una prioridad, estado y fecha.

id_tarea, usuario_id, nombre, descripcion, fecha_creacion, fecha_vencimiento, prioridad, estado

Habitos:

El usuario tiene ciertas actividades que desea repetir, ya sea a diario, cada dia de por medio, o cada Lunes.

id_habito, usuario_id, nombre, descripcion, frecuencia, hora_recordatorio, activo

Dia_Habito:

Para saber cuales son los dias en el que usuario hara un habito en especifico

id_dia, habito_id, dia

Seguimiento_Habito:

Vera si un habito fue cumplido en cierto dia

id_seguimiento, habito_id, fecha, completado

Logro:

Cada que un usuario avanza obtiene un logro por haber progresado

id_logro, usuario_id, nombre, descripcion, fecha_obtenido, tipo

Estado:

Es el estado de una tarea, la tarea puede estar completada, en proceso o en espera.

id_estado, estado

Prioridad:

Nos dice la prioridad de una tarea.

id_prioridad, prioridad

3. Analisis de Cardinalidad y Relaciones

- Un usuario puede tener 0 o muchas tareas (0, N)
- Un usuario puede tener 0 o muchos habitos (0, N)
- Un usuario puede tener 0 o muchas logros (0, N)
- Una tarea pertenece a un unico usuario(1, 1)
- Un habito pertenece a un unico usuario(1, 1)
- Un habito tiene 0 o muchos dias asignados (0, N)
- Cada dia es parte de un habito
- Un habito puede tener varios dias (0, N)
- Un logro pertenece a un usuario
- Un usuario puede tener muchos logros (0, N)
- Una tarea puede tener un estado (1, 1)
- Un estado puede tener varias tareas (1, N)
- Una tarea tiene una prioridad (1, 1)
- Una prioridad puede tener una o varias tareas (1, N)

4. Definición Detallada de Colecciones:

Usuario:

{

“_id” : ObjectId

“nombre”: String

“email”: String

“fecha_registro”: String -> “YYYY-MM-DD”

}

Tareas:

{

 “_id”: ObjectId

 “usuario_id”: ObjectId

 “nombre_tarea”: string

 “descripcion”: string

 “fecha_creacion”: String -> “YYYY/MM/DD”

 “fecha_vencimiento”: String -> “YYYY/MM/DD”

}

Prioridad:

{

 “_id”: ObjectId

 “prioridad”: String

}

Estado:

{

 “_id”: ObjectId

 “estado”: String

}

Habitos:

{

 “_id”: ObjectId

 “usuario_id”: ObjectId

 “nombre_habito”: string

```

“descripcion”: string
“id_dias_habito”: ObjectId
    “activo”: bool
}

```

Dias_habito:

```

{
“_id”: ObjectId
“dia”: string
}

```

Logros:

```

{
“_id”: ObjectId
“usuario_id”: ObjectId
“nombre_logro”: string
“descripcion”: string
“fecha_obtenido”: String -> “YYYY/MM/DD”
“tipo”: string
}

```

5. Ejemplos de Endpoints RESTful

Metodo	Endpoint	Descripcion
GET	/api/usuarios	Devuelve la lista de usuarios
GET	/api/usuarios/:id	Devuelve la informacion de un usuario en especifico
POST	/api/usuarios	Crea un nuevo usuario
PUT	/api/usuarios/:id	Actualiza la informacion en un usuario

DELETE	/api/usuarios/:id	Elimina el usuarios (evitar api/usuarios esto eliminaria todos los usuarios)
--------	-------------------	--

Tareas:

Metodo	Endpoint	Descripcion
GET	/api/tareas	Devuelve todas las tareas
GET	/api/tareas/:id	Devuelve la informacion de una tarea
GET	/api/usuarios/:id/tareas	Devuelve las tareas de un usuarios en especifico
POST	/api/tareas	Crea una tarea
PUT	/api/tareas/:id	Cambia/actualiza la informacion de una tarea
DELETE	/api/tareas/:id	Elimina tarea

Habitos:

Metodo	Endpoint	Descripcion
GET	/api/habitos	Devuelve todas las habitos
GET	/api/habitos/:id	Devuelve la informacion de un habito
GET	/api/usuarios/:id/habitos	Devuelve los habitos de un usuarios en especifico
POST	/api/habitos	Crea una habito
PUT	/api/habitos/:id	Cambia/actualiza la informacion de una habito
DELETE	/api/habitos/:id	Elimina habito

Logros:

Metodo	Endpoint	Descripcion
GET	/api/logros	Devuelve todas las logros
GET	/api/logros/:id	Devuelve la informacion de una logros
GET	/api/usuarios/:id/logros	Devuelve las logros de un usuarios en especifico
POST	/api/logros	Crea una logros
PUT	/api/logros/:id	Cambia/actualiza la informacion de una logros
DELETE	/api/logros/:id	Elimina logros

Estado:

Metodo	Endpoint	Descripcion
--------	----------	-------------

GET	/api/estados	Devuelve la lista de estados
GET	/api/estados/:id	Devuelve la información de un estado
POST	/api/estados	Crea un nuevo estado
PUT	/api/estados/:id	Actualiza la información de un estado
DELETE	/api/estados/:id	Elimina un estado
GET	/api/estados	Devuelve la lista de estados
GET	/api/estados/:id	Devuelve la información de un estado

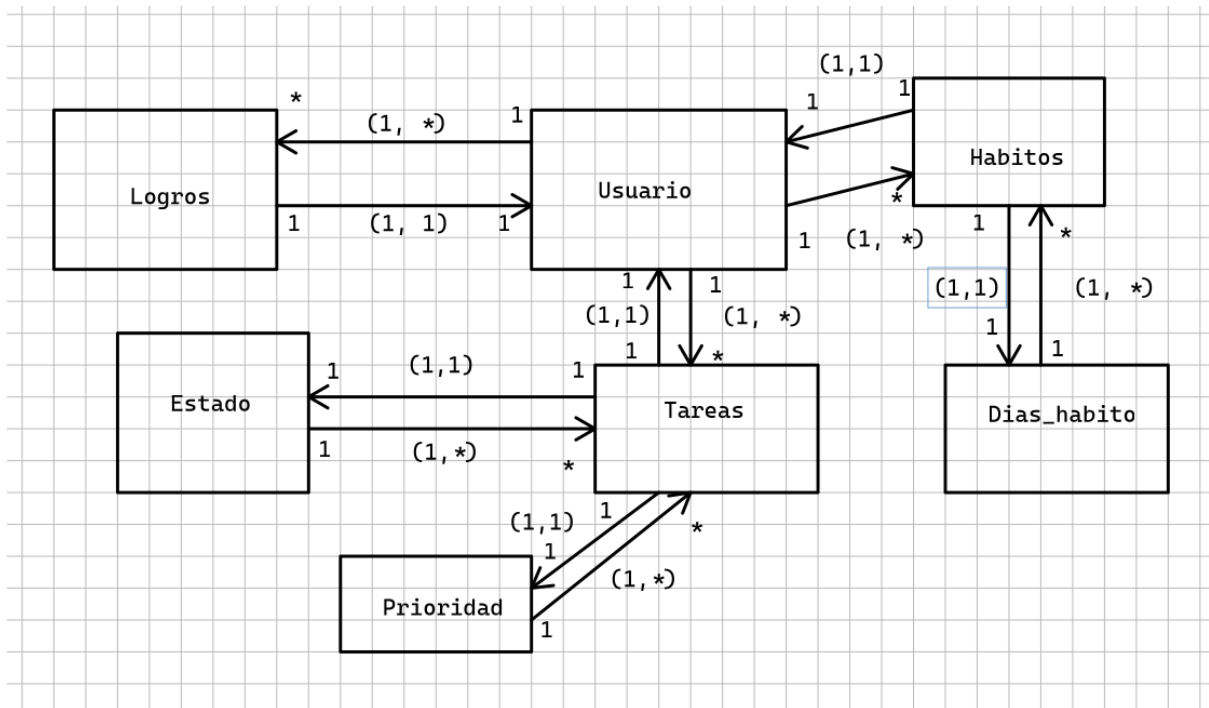
Prioridad:

Metodo	Endpoint	Descripcion
GET	/api/prioridades	Devuelve la lista de prioridades
GET	/api/prioridades/:id	Devuelve la información de una prioridad
POST	/api/prioridades	Crea una nueva prioridad
PUT	/api/prioridades/:id	Actualiza la información de una prioridad
DELETE	/api/prioridades/:id	Elimina una prioridad
GET	/api/prioridades	Devuelve la lista de prioridades

Dias_Habito:

Metodo	Endpoint	Descripcion
GET	/api/días_habito	Devuelve los días en los que hay un habito
GET	/api/días_habito/:id	Devuelve la información de los días de un habito
POST	/api/días_habito	Crea nuevos días para hábitos
PUT	/api/días_habito/:id	Actualiza la información de días_habito

DELETE	/api/dias_habito/:id	Elimina los dias de un habito
--------	----------------------	-------------------------------



Pruebas de como funciona el código, con los endpoints que pudieron ser programados:

Links