

Proyecto final 3º trimestre



En el proyecto se emplea la base de datos del centro de investigación Marmol Center, subido a mongo Atlas. La empresa se divide en departamentos especializados (Mecánica, Electromagnetismo, Teología, Mecánica Cuántica, Medicina, Magia y Energías Renovables) que contratan a sus propios empleados. Dichos trabajadores realizan proyectos que son financiados por sus departamentos y por inversores externos.

La base de datos está compuesta por 4 colecciones: departamentos, empleados, proyectos e inversores

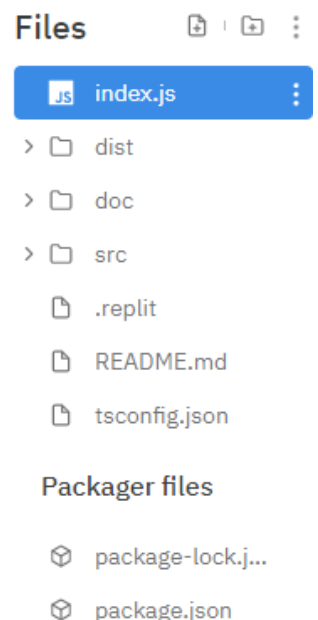
El proyecto tiene dos criterios definidos: la correcta conexión de una base de datos en mongo Atlas al entorno online Replit y la manipulación de los datos de la db en formato json mediante TypeScript.

Estructura de datos

Está formado por tres directorios: dist, doc y src.

La carpeta doc contiene los pdf con toda la información necesaria para el proyecto:

- El primer pdf “Estructura de replit” explica los lenguajes de programación, entornos de desarrollo, los formatos de datos y los paquetes usados
- El segundo pdf “Conexiones con la BD” explica cómo se conecta replit con la base de datos de Mongo Atlas.
- Este último pdf explica en qué consiste la base de datos y las distintas consultas que se realizan, así como su lógica y sintaxis



Schemas: los esquemas utilizados, análogos a los de mongo, son:

Empleados:

```
const empleadoSchema = new Schema({
  _id:String,
  nombre:String,
  fechaEntrada:Date,
  jornadaSemanal:Number,
  departamento:String,
  proyecto:Number,
  faltas:Number
})
```

```
export const Empleados = model('empleados',empleadoSchema)
```

Departamentos:

```
const departamentoSchema = new Schema({
  _id: String,
  ingresoMensual:Number,
  sueldoHora:Number,
  proyectos:Array,
})
```

```
export const Departamentos = model('departamentos',departamentoSchema)
```

Inversores :

```
const inversorSchema = new Schema({
  _id: Number,
  nombre: String,
  abono: Number,
})
```

```
export const Inversores = model('inversores',inversorSchema)
```

Proyectos :

```
const proyectoSchema = new Schema({
  _id:Number,
  pseudonimo: String,
  descripcion: String,
  presupuestoMensual: Number,
  tiempo: {
    fechaInicio: Date,
    fechaMeta: Date,
    terminado: Boolean,
  },
  financiacion: Number
})
```

```
export const Proyectos = model('proyectos',proyectoSchema)
```

Consultas

Hay un total de 4 consultas que, en conjunto, emplean los fundamentos básicos y avanzados de typescript: tipos de constantes y variables, bucles for y do while, iteradores, etc...

Función 0 : Una cálida bienvenida que se inicia de forma automática al conectarse con la base de datos.

```
const fun0 = async (req: Request, res: Response) => {  
  await db.conectarBD()  
  res.send("Bienvenido a la base de datos de Marmol Center")  
  db.desconectarBD()  
}
```

Función 1 : Cada departamento recibe ciertas cantidades de dinero para financiar sus proyectos, que es el resultado del pago que le da la empresa y las inversiones de distintas entidades. Hay departamentos que tienen excedentes en sus ingresos, y otros que tienen pérdidas. Alejandro Mármol decide implantar una medida socialista en la empresa, con caracter retroactivo, y todos los departamentos donarán sus excedentes a quienes estén en números rojos, y el sobrante se repartirá en partes iguales.

```
{  
  "_id" : "Electromagnetismo",  
  "ingresoMensual" : 80500,  
  "financiación" : 6900,  
  "salarioEmpleados" : 4356,  
  "costeProyectos" : 70650,  
  "ganancias" : 87400,  
  "pagos" : 75006,  
  "fondos" : 12394  
}
```

Primero, se conecta a la base de datos de mongo y realiza una consulta. Esta devuelve los datos de cada departamento, siendo “fondos” el pilar de la función, pues de él se quitará el dinero o, de lo contrario, se donará.

La lógica empleada es la siguiente:

Para cada departamento (elem) se observa si sus fondos son positivos. De ser así se le quita el excedente (restandoselo a su ingreso mensual) y se guarda en una bolsa llamada exceso.

Si los fondos son negativos, se le suma al ingreso mensual el déficit quitar su deuda, y se guarda el dinero cedido en una bolsa llamada perdida.

Luego, se restan las bolsas y se obtiene el dinero sobrante de los donativos, y se repartirá en partes iguales a cada departamento, de modos que todos tienen los mismos fondos, independientemente del coste de sus proyectos

Luego se devuelve el array en formato json como resultado de la consulta para, en un futuro, insertarlo de nuevo en la base de datos.

```
for (elem of array){  
  if ( elem.fondos > 0){  
    elem.ingresoMensual -= elem.fondos  
    exceso += elem.fondos  
    elem.fondos = 0  
  }  
  
  if ( elem.fondos < 0){  
    elem.ingresoMensual -= elem.fondos  
    perdida -= elem.fondos  
    elem.fondos = 0  
  }  
}  
  
reparto = exceso - perdida  
reparto = reparto/7  
reparto = Math.round(reparto)  
  
for (elem of array){  
  elem.ingresoMensual+= reparto  
  elem.fondos+= reparto  
}
```

Función 2 : La empresa decide realizar donativos trimestrales al mejor trabajador de cada departamento, que será aquel que tenga más antigüedad y no haya cometido ninguna falta.

```
"_id" : "Energías Renovables",
"trabajadores" : [
  {
    "nombre" : "Diego Rósame Caricia",
    "faltas" : 0,
    "diasTrabajados" : 745,
    "sueldo" : 1575
  },
  {
    "nombre" : "Alibabá Alahuak Bárcenas",
    "faltas" : 0,
    "diasTrabajados" : 745,
    "sueldo" : 1575
  }
]
```

La consulta devuelve la información de cada trabajador en la empresa. Luego se le da el resultado a la variable array. Se realizan a hora dos bucles for dentro de otro:

```
for (element of array){
  for (elem of element.trabajadores){ //Se comparan los trabajadores
    if(elem.faltas==0){
      if(elem.diasTrabajados>tiempo)
      {tiempo = elem.diasTrabajados
      trabajadorElegido = elem.nombre}
    }
  }
}

for (elem of element.trabajadores){//Se localiza al mejor y recompensa
  if(elem.nombre == trabajadorElegido)
  { trabajadoresNombre.push(elem.nombre)
    trabajadoresSueldo.push(Math.round(elem.sueldo))
    conjuntoDinero+=Math.round(elem.sueldo)
  }
}
tiempo = 0
}
```

Para cada departamento se van a hacer dos inspecciones: Primero se busca a quien sea el mejor: si no ha cometido ninguna falta se apunta el tiempo trabajado, y se irá comparando con los del resto.

Luego se busca quien ha sido el elegido y se introducen sus datos en dos arrays (nombre y sueldo), formando una matriz doble y, más adelante, triple.

Ahora se reparte el dinero. El dinero que se dona cada trimestre se realiza de manera proporcional a cada trabajador. Para ello se sumó los sueldos de cada uno (conjuntoDinero) y se calcula el porcentaje que le corresponde del premio.

```
for (var i = 0; i < trabajadoresSueldo.length; i++) {
  porcentaje = trabajadoresSueldo[i]/conjuntoDinero*100
  premio = porcentaje/100*dinero
  trabajadoresPremio.push(Math.round(premio))
}
```

El premio se introduce en un tercer array, formando así una matriz entre el nombre, el sueldo y el premio de cada trabajador. Luego se introducen en un array resultado que tiene una interfaz respuesta apropiada para la matriz, y por ultimo se envia como respuesta en formato json.

```
for (var i = 0; i < trabajadoresNombre.length; i++){
  resultado.push({
    nombre:trabajadoresNombre[i],
    salario:trabajadoresSueldo[i],
    premio:trabajadoresPremio[i]
  })
}
```

Función 3 : La empresa decide realizar un donativo a Caritas, y ruega a los inversores que donen una porción de su financiación mensual hasta alcanzar la recaudación requerida

```
"_id" : "GMA", "abono" : 6300 }
"_id" : "NASA", "abono" : 10300 }
"_id" : "Venture", "abono" : 1350 }
"_id" : "Vodafone", "abono" : 600 }
"_id" : "EOLIC", "abono" : 5000 }
"_id" : "Santa Sede", "abono" : 50000 }
"_id" : "Howarts", "abono" : 13000 }
"_id" : "Sun Gold", "abono" : 1680 }
"_id" : "Health Heart", "abono" : 12500 }
```

La consulta devuelve cada inversor junto con la suma de sus abonos mensuales a sus proyectos financiados.

Luego, se crea un array con los inversores que han aceptado donar una parte de sus pagos (abonantes).

```
[
  { _id: 'GMA', abono: 6300 },
  { _id: 'Sun Gold', abono: 1680 },
  { _id: 'Howarts', abono: 13000 },
  { _id: 'Health Heart', abono: 12500 },
  { _id: 'Venture', abono: 1350 },
  { _id: 'EOLIC', abono: 5000 },
  { _id: 'Santa Sede', abono: 50000 },
  { _id: 'Vodafone', abono: 600 },
  { _id: 'NASA', abono: 10300 }
]
[ 'EOLIC', 'NASA', 'Howarts' ]
3
Ronda de donaciones numero 1
GMA:
no abona nada
Sun Gold:
no abona nada
Howarts:
abona 650
Recaudado : 650
Health Heart:
no abona nada
Venture:
no abona nada
EOLIC:
abona 250
Recaudado : 900
```

Se crea un bucle doWhile, que realizará rondas de donativos del 5% de las inversiones hasta que se haya alcanzado la recaudación completa. Si el inversor está en los abonantes, se calcula su aporte, se le añade al lote y se verifica con un boolean que es donante .

Una vez alcanzado la cantidad requerida, se sale del bucle y se envia el total como respuesta.

Bibliografía

- Replit del profesor Adolfo Salto : <https://replit.com/@AdolfoSalto>
- Mi perspicacia e inteligencia lógica.