

Trabajo Práctico Integrador

Desarrollo de Aplicaciones Cliente-Servidor

Entrega: Primera entrega

Profesores:

- QUEVEDO, Fabricio
- VILLAVERDE, Jorge

Grupo N° 8 - Integrantes:

- ACEVEDO, Fernando
- GETZEL, Martín
- LUCAS, Dania
- LÓPEZ, María Fernanda
- MONETA, Federico
- OJEDA, María Jose
- VALDES, Manuel

Ciclo lectivo: 2021

Cuatrimestre: Primero



Lenguaje seleccionado:

Tomamos la decisión de hacer el trabajo principalmente en JavaScript, debido a que este lenguaje nos permite trabajar en clientes livianos y existen una gran cantidad de frameworks disponibles para facilitar nuestro trabajo.

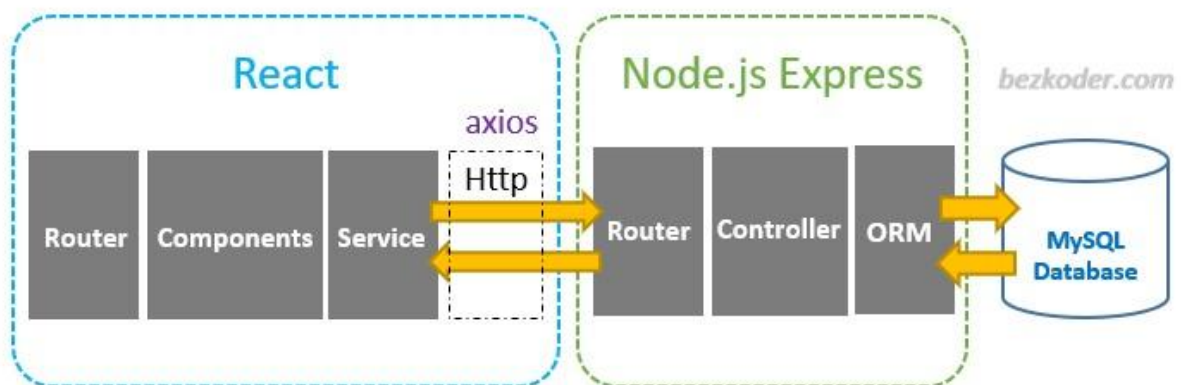
Patrones utilizados:

Para este proyecto, utilizamos el patrón Modelo-Vista-Controlador, ya que el mismo nos permite separar el trabajo en varias capas (que nos es muy útil a la hora de dividir el trabajo en sub-equipos de trabajo) y obtener un código con una mejor legibilidad.

Tecnologías utilizadas:

- **Node.js - Express** para el backend, exporta REST APIs e interactúa con una BDD. El entorno de ejecución de node permite a los desarrolladores crear toda clase de herramientas de lado servidor y aplicaciones en JavaScript.
- **MySQL** como motor de base de datos, utilizando servidor **Azure**.
- **Sequelize** como ORM para conectar a la base de datos en MySQL.
- **React + HTML + CSS + Js** para el frontend, envía HTTP Requests y retorna HTTP Responses usando Axios, consume data de sus componentes. React Route es utilizado para navegar entre páginas.

Arquitectura seleccionada:



Decidimos utilizar esta arquitectura ya que contamos con una amplia documentación de la misma, y entendemos fácilmente su funcionamiento.

Librería:

Se desarrolló una librería de JavaScript llamada **grupo8**, donde se definieron los métodos necesarios para comunicarse con las APIs del Ministerio de Desarrollo Productivo (con la finalidad de enviar regímenes mensuales en formato JSON) y la Secretaría de Comercio Interior. Para esto se definieron los métodos en un archivo .js y luego se lo exportó como módulo.



```
module.exports = {  
  getNotificacion: getNotificacion,  
  postRegimen: postRegimen,  
  loginMinisterio: loginMinisterio,  
  generarRegimen: generarRegimen,  
  generarRegistro: generarRegistro,  
  cargarRegistro: cargarRegistro  
}
```

Si se la quiere usar a la librería se la puede importar de la siguiente manera:

```
const empresas = require('./lib/grupo8.js')
```

Así luego se podrá llamar a cada método de la librería con **empresas.[metodo]**

Funciones definidas:

Comunicaciones con *Secretaria*:

getNotificacion

```
empresas.getNotificacion(api_notificacion,token)
```

Retorna la notificación que da la secretaria con en el siguiente formato json



```
{
  "id":3,
  "descripcion": "Maiores quod excepturi provident",
  "fecha": "2051-10-10T05:01:09.0727",
  "token": "1ff48y7yr-bf8228-42d3d-2e2e-d3e3e3e3ee"
}
```

Comunicación con *Ministerio*:

loginMinisterio

```
var token = empresas.loginMinisterio( [URL_api], [user], [password])
```

Función devuelve el token que retorna el ministerio al autenticarse, se le pasara como parámetro la url de la api de ministerio, usuario y contraseña.

Regímenes a postear en *Ministerio*:

generarRegistro → *generarRegimen* → *postRegimen*

generarRegistro

```
const registro1 = empresas.generarRegistro("Ripio",1234567895825,3800,"m^3",3500,2698)
```

Formato JSON de registro a generar

```
sal = {
  "denominacion": denom,
  "codigo_ean": ean,
  "precio_unidad": precioUn,
  "unidad_medida": unidadMed,
  "cantidad_prod": cantProd,
  "cantidad_vend": cantVend
}
```



cargarRegistro

```
var lista_reg = empresas.cargarRegistro(registro1, lista_reg);
```

Tomamos cada uno de los registros generados previamente y los cargamos en una lista, la cuál nos permitirá agregarlos posteriormente al régimen.

generarRegimen

```
var regimen = empresas.generarRegimen(2020, 10, lista_reg, fecha, "15362585915", "Ricardo Ramirez e Hijos S.R.L");
```

A partir de info obtenida de la BDD de la empresa, pasada como parámetros a la función, se genera el régimen con un cierto formato deseado para que luego pueda ser aceptado por la api del Ministerio

```
sal = {  
  "periodo": {  
    "year": anio,  
    "month": mes  
  },  
  "listaRegistro": registros,  
  "date_upload": fecha_carga,  
  "day_limit": 10,  
  "infoEmpresa": {  
    "cuit": inputCuit,  
    "razon_social": nombre_empresa  
  }  
}
```

postRegimen



```
empresas.postRegimen(api_regimen, regimen);
```

Una vez generados los regímenes se podrá realizar un *POST a API del ministerio*.

Web de Empresas:

Se desarrolló un proyecto node-express-react que será la plataforma utilizada por la empresa para utilizar los métodos definidos en la librería de empresas. Esta tendrá en la BDD los datos a cargar en los regímenes a enviar a ministerio, y por medio de su front veremos las notificaciones recibidas por la secretaria.

Estructura del proyecto:

Backend y Frontend:

