Tipo : Guía de Enunciado
Capítulo : Single Page Application (SPA) con React
Duración : 40 minutos

I. OBJETIVO

Listado y paginación con ReactJS y Bootstrap.

II. REQUISITOS

Los siguientes elementos de software son necesarios para la realización del laboratorio:

- Windows 10 (como mínimo Windows 8)
- NodeJs
- Visual Studio Code
- Visual Studio 2017 (mínimo 2015)

III. EJECUCIÓN DEL LABORATORIO

1. Abrir la solución “**Cibertec**” en Visual Studio 2017 y nos ubicamos en el proyecto “**Cibertec.WebApi**”.

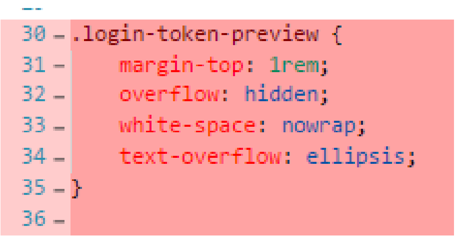
2. Procedemos a modificar nuestro web api, para lo cual nos ubicamos en el proyecto “**Cibertec.WebApi**” y abrimos el controlador “**ProductController**” y agregamos estos nuevos endpoints:

```
[HttpGet]
[Route("count")]
public IHttpActionResult GetCount()
{
    return Ok(_unit.Customers.Count());
}

[HttpGet]
[Route("list/{page}/{rows}")]
public IHttpActionResult GetList(int page, int rows)
{
    var startRecord = ((page - 1) * rows) + 1;
    var endRecord = page * rows;
    return Ok(_unit.Customers.PagedList(startRecord, endRecord));
}
```

3. Ahora procedemos a abrir Visual Studio Code y nos ubicamos en nuestra carpeta “**Cibertec-React**”.

4. Procedemos a editar en la carpeta “**assets**” el fichero “**styles.css**”, eliminamos la siguiente clase “**login-token-preview**” que fue creada temporalmente.



```
.login-token-preview {
    margin-top: 1rem;
    overflow: hidden;
    white-space: nowrap;
    text-overflow: ellipsis;
}
```

5. De la carpeta “**src**” editamos el archivo “**defaultsParams.tsx**” de la siguiente manera:

```
export default function DefaultParams() {
    return {
        customerReducer: {
            token: "",
            customers: Array<any>()
        }
    };
}
```

6. De la carpeta “**src**” editamos el archivo “**routes.tsx**” de la siguiente manera:

```
import * as React from 'react';
import {Route, Router, IndexRoute, hashHistory} from 'react-router';
import { Provider } from 'react-redux';
import { createStore, applyMiddleware, compose } from 'redux';
import reduxThunk from 'redux-thunk';
import reducers from './reducers/rootReducer';
import DefaultParams from './defaultParams';

import Login from "./components/Login";
import NotFound from "./components/404";
import CustomerList from "./components/CustomerList";

const createStoreWithMiddleware =
applyMiddleware(reduxThunk)(createStore);
const store = createStoreWithMiddleware(reducers, DefaultParams());

export class CustomersApp extends React.Component<any, any> {
    render() {
        return (
            <Provider store={store}>
                <Router history={hashHistory}>
                    <Route path="/" component={Login}/>
                    <Route path="/customers" component={CustomerList}/>
                    <Route path="*" component={NotFound} />
                </Router>
            </Provider>
        );
    }
}
```

7. De la carpeta “**src\actions**” editamos el archivo “**creators.tsx**” de la siguiente manera:

```
import * as types from './types';
import axios from 'axios';
import { Dispatch } from "redux";
import * as URLSearchParams from 'url-search-params'

let apiUrl = 'http://localhost/Cibertec.WebApi';

export function loginUser(email: string, password: string) {
    return function (dispatch: any) {
        let params = new URLSearchParams();
        params.append('grant_type', 'password');
        params.append('username', email);
        params.append('password', password);
        axios.post(`${apiUrl}/token` , params)
            .then(response => {
                dispatch({
                    type: types.GOT_TOKEN,
                    token: response.data.access_token
                });
            })
            .catch((error) => { console.log(error) });
    }
}


export function getCustomerList(token: string, page: number,
pageSize: number) {
    return function (dispatch: any) {
        let config = { headers: { Authorization: `Bearer ${token}`
}};
        axios.get(`${apiUrl}/customer/list/${page}/${pageSize}`,
config)
            .then(response => {
                dispatch({
                    type: types.GOT_CUSTOMERS,
                    customers: response.data
                });
            })
            .catch((error) => { console.log(error) });
    }
}

export function getCustomerCount(token: string) {
    return function (dispatch: any) {
        let config = { headers: { Authorization: `Bearer ${token}`
}};
        axios.get(`${apiUrl}/customer/count`, config)
```

```
            .then(response => {
                dispatch({
                    type: types.GOT_CUSTOMERS_COUNT,
                    customerCount: response.data
                });
            })
            .catch((error) => { console.log(error) });
    }
}
```

8. De la carpeta “**src\actions**” editamos el archivo “**types.tsx**” de la siguiente manera:

```
export const GOT_TOKEN = 'GOT_TOKEN'
export const GOT_CUSTOMERS = 'GOT_CUSTOMERS';
export const GOT_CUSTOMERS_COUNT = 'GOT_CUSTOMERS_COUNT';
```

9. En la carpeta “**src\components**” creamos el fichero “**CustomerList.tsx**” con el siguiente código:

```
import * as React from 'react';
import { bindActionCreators } from 'redux';
import { connect } from 'react-redux';
import { getCustomerList } from '../actions/creators';
import { getCustomerCount } from '../actions/creators';
import { hashHistory } from 'react-router';
import { Link } from 'react-router';

interface ICustomerListProps extends React.Props<any> {
    token?: string,
    customers?: Array<any>,
    localGetCustomerList?: Function,
    localGetCustomerCount?: Function
}

interface ICustomerListState {
    page: number,
    pageSize: number,
    customerCount: number,
    maxPageNumber: number
};

function mapStateToProps(state: any, ownProps: any) {
    return {
        token: state.loginReducer.token,
        customers: state.customerReducer.customers,
        customerCount: state.customerReducer.customerCount
    };
}

function mapDispatchToProps(dispatch: any) {
    return {
        localGetCustomerList: bindActionCreators(getCustomerList, 
dispatch),
        localGetCustomerCount: bindActionCreators(getCustomerCount, 
dispatch)
    };
}

class CustomerList extends React.Component<any, ICustomerListState> 
{
    constructor(props: ICustomerListProps) {
        super(props);

        this.state = {
            page: 1,
            pageSize: 20,
            customerCount: 20,
            maxPageNumber: 1
        };

        this.handlePagePrev = this.handlePagePrev.bind(this);
        this.handlePageNext = this.handlePageNext.bind(this);
    }

    componentDidMount() {
        let { token, customerCount } = this.props;
        let { page, pageSize } = this.state;
        this.props.localGetCustomerList(token, page, pageSize);
        this.props.localGetCustomerCount(token);
    }

    render() {
        let { customers } = this.props;
        let panelClassName = this.props.customers.length > 0 ? 
"panel-list" : "panel-list panel-list--empty";
        return <div className="container-fluid container-
background">
            <div className="row full-height-row">
                <div className="col-md-3 hidden-sm"></div>
                <div className="col-sm-12 col-md-6">
                    <div className="text-center">
                        <div className={`panel panel-default text-
left ${panelClassName}`}>
                            <div className="panel-body">
                                <h3>Lista de clientes</h3>
                                <table className="table table-
bordered table-hover table-condensed">
                                    <thead>
                                        <tr>
```

```
                                            <th>ID</th>
                                            <th>Nombre
Completo</th>
                                            <th>Ciudad</th>
                                            <th>País</th>
                                            <th>Teléfono</th>
                                            <th>Acciones</th>
                                        </tr>
                                    </thead>
                                    <tbody>
                                        {customers.map((item) => {
                                            return <tr
key={item.id}>
                                                <td>{item.id}</td>

<td>{`${item.firstName} ${item.lastName}`}</td>

<td>{item.city}</td>

<td>{item.country}</td>

<td>{item.phone}</td>
                                                <td>
                                                    <Link
to={`/customerview/${item.id}`}><span className="glyphicon
glyphicon-eye-open list-icon text-primary"></span></Link>
                                                    <Link
to={`/customeredit/${item.id}`}><span className="glyphicon
glyphicon-pencil list-icon text-primary"></span></Link>
                                                </td>
                                            </tr>
                                        })}
                                    </tbody>
                                </table>

                                <div className="text-center">
                                    <div className="btn-group"
role="group">
                                        <button type="button"
className="btn btn-success" onClick={this.handlePagePrev}>&laquo;
Anterior</button>
                                        <button type="button"
className="btn btn-success" onClick={this.handlePageNext}>Siguiente
&raquo;</button>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
```

```
                </div>
                <div className="col-md-3 hidden-sm"></div>
            </div>
        </div>;
    };

    handlePagePrev(event: any) {
        event.preventDefault();
        let { token } = this.props
        let { page, pageSize } = this.state
        let newPage = page - 1 > 0 ? page - 1 : page;

        this.setState({ page: newPage }, () => {
            this.props.localGetCustomerList(token, this.state.page,
pageSize);
        });

    }

    handlePageNext(event: any) {
        event.preventDefault();
        let { token, customerCount } = this.props
        let { page, pageSize } = this.state

        let maxPage = (customerCount % pageSize > 0 ?
(customerCount / pageSize) + 1 : (customerCount / pageSize));
        let newPage = page + 1;
        if (newPage <= maxPage) {
            this.setState({ page: newPage }, () => {
                this.props.localGetCustomerList(token,
this.state.page, pageSize);
            });
        }
    }

}

export default connect(mapStateToProps,
mapDispatchToProps)(CustomerList);
```

10. En la carpeta “**src\components**” creamos el fichero “**Login.tsx**” con el siguiente código:

```
import * as React from "react";
import { bindActionCreators } from 'redux';
import { connect } from 'react-redux';
import { hashHistory } from 'react-router';
import { loginUser } from '../actions/creators';
```

```
interface ILoginState {
    email?: string;
    password?: string;
}

interface ILoginProps extends React.Props<any>{
    token?: string,
    customers?: any[];
    localLoginUser?: Function,
}

function mapStateToProps(state: any, ownProps: any) {
    return {
        token: state.loginReducer.token,
        customers: state.customerReducer.customers
    };
}

function mapDispatchToProps(dispatch: any) {
    return {
        localLoginUser: bindActionCreators(loginUser, dispatch),
    };
}

class Login extends React.Component<any, ILoginState> {
    constructor(props: ILoginProps) {
        super(props);
        this.state = {
            email: 'cibertec@cibertec.edu.pe',
            password: '12345678'
        }

        this.handleEmailChange = this.handleEmailChange.bind(this);
        this.handlePasswordChange = 
this.handlePasswordChange.bind(this);
        this.handleLoginUser = this.handleLoginUser.bind(this);
    }

    componentWillReceiveProps(nextProps: any) {
        if (nextProps.token !== '')
            hashHistory.push('/customers');
    }

    render() {
        let {
            email,
            password
        } = this.state;
        return <div className="container-fluid container-
background">
            <div className="row full-height-row">
                <div className="col-md-3 hidden-sm"></div>
                <div className="col-sm-12 col-md-6">
                    <div className="text-center">
                        <div className="panel panel-default panel-
login text-left">
                            <div className="text-center">
                                <div><img
src="assets/imgs/reactdux.png" className="login-header-img"
/></div>
                            </div>
                            <div className="panel-body">
                                <h3>Iniciar sesión</h3>
                                <form>
                                    <div className="form-group">
                                        <label >Email</label>
                                        <input type="email"
className="form-control"  placeholder="Email"

onChange={this.handleEmailChange} value={email}/>
                                    </div>
                                    <div className="form-group">
                                        <label >Password</label>
                                        <input type="password"
className="form-control" placeholder="Password"

onChange={this.handlePasswordChange} value={password}/>
                                    </div>
                                    <button className="btn btn-
primary btn-block" onClick={this.handleLoginUser}>Entrar</button>
                                </form>
                            </div>
                        </div>
                    </div>
                </div>
                <div className="col-md-3 hidden-sm"></div>
            </div>
        </div>
    }

    handleEmailChange (event: any) {
        let newVal = event.currentTarget.value;
        this.setState({email: newVal});
    }

    handlePasswordChange (event: any) {
        let newVal = event.currentTarget.value;
        this.setState({password: newVal});
    }

    handleLoginUser(event: any) {
        event.preventDefault();
        let {
            email,
            password
        } = this.state;

        this.props.localLoginUser(email, password);
    }

}

export default connect(mapStateToProps, mapDispatchToProps)(Login);
```

11. En la carpeta “**src\reducers**” creamos el fichero “**customers.tsx**” con el siguiente código:

```
import * as types from '../actions/types';
import { hashHistory } from 'react-router';
const INITIAL_STATE = {};

export function customerReducer(state = INITIAL_STATE, action: any)
{
    switch (action.type) {
        case types.GOT_CUSTOMERS:
            return { ...state, customers: action.customers };
        case types.GOT_CUSTOMERS_COUNT:
            return { ...state, customerCount: action.customerCount
};
    }

    return state;
}
```

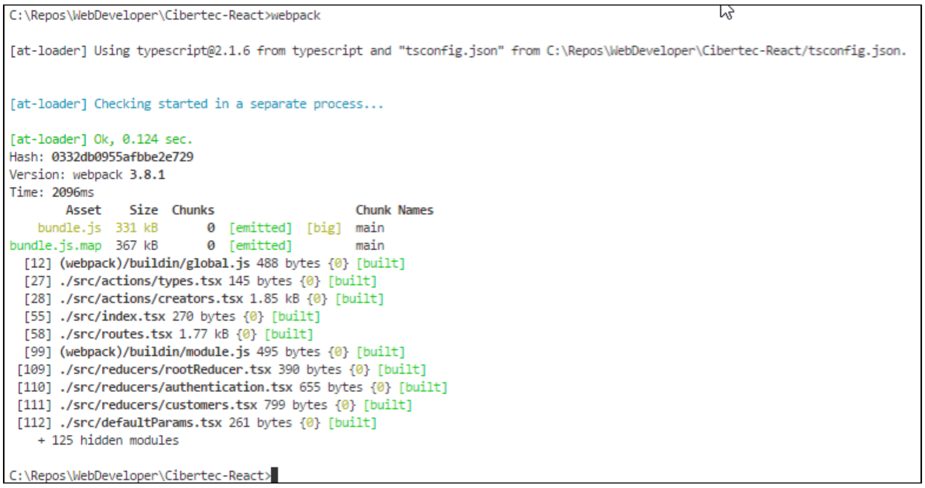
12. Y ahora en la misma ubicación editamos el fichero “**rootReducer.tsx**” de la siguiente manera:

```
import { combineReducers } from 'redux';
import { loginReducer } from './authentication';
import { customerReducer } from './customers';
```

```
const rootReducer = combineReducers({
    loginReducer,
    customerReducer
});

export default rootReducer;
```

13. Una vez finalizado procedemos a ejecutar el comando “**webpack**” desde el terminal integrado de Visual Studio Code.



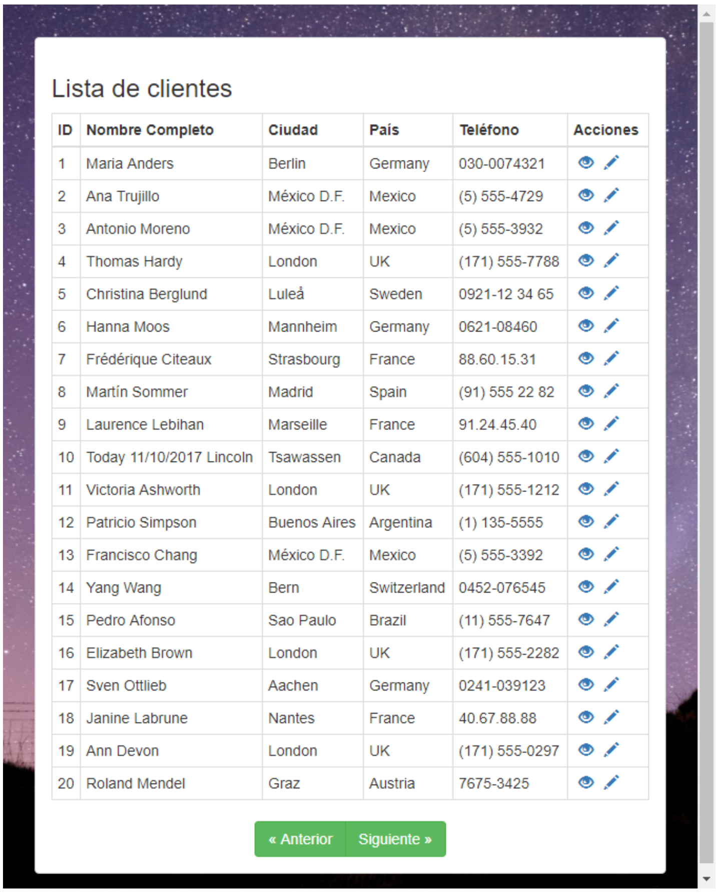
```
C:\Repos\WebDeveloper\Cibertec-React>webpack

[at-loader] Using typescript@2.1.6 from typescript and "tsconfig.json" from C:\Repos\WebDeveloper\Cibertec-React/tsconfig.json.


[at-loader] Checking started in a separate process...

[at-loader] Ok, 0.124 sec.
Hash: 0332db0955afbbe2e729
Version: webpack 3.8.1
Time: 2096ms
        Asset    Size  Chunks                    Chunk Names
    bundle.js  331 kB       0  [emitted]  [big]  main
bundle.js.map  367 kB       0  [emitted]         main
  [12] (webpack)/buildin/global.js 488 bytes {0} [built]
  [27] ./src/actions/types.tsx 145 bytes {0} [built]
  [28] ./src/actions/creators.tsx 1.85 kB {0} [built]
  [55] ./src/index.tsx 270 bytes {0} [built]
  [58] ./src/routes.tsx 1.77 kB {0} [built]
  [99] (webpack)/buildin/module.js 495 bytes {0} [built]
 [109] ./src/reducers/rootReducer.tsx 390 bytes {0} [built]
 [110] ./src/reducers/authentication.tsx 655 bytes {0} [built]
 [111] ./src/reducers/customers.tsx 799 bytes {0} [built]
 [112] ./src/defaultParams.tsx 261 bytes {0} [built]
    + 125 hidden modules

C:\Repos\WebDeveloper\Cibertec-React>
```



Lista de clientes

ID	Nombre Completo	Ciudad	País	Teléfono	Acciones
1	Maria Anders	Berlin	Germany	030-0074321	👁 ✎
2	Ana Trujillo	México D.F.	Mexico	(5) 555-4729	👁 ✎
3	Antonio Moreno	México D.F.	Mexico	(5) 555-3932	👁 ✎
4	Thomas Hardy	London	UK	(171) 555-7788	👁 ✎
5	Christina Berglund	Luleå	Sweden	0921-12 34 65	👁 ✎
6	Hanna Moos	Mannheim	Germany	0621-08460	👁 ✎
7	Frédérique Citeaux	Strasbourg	France	88.60.15.31	👁 ✎
8	Martín Sommer	Madrid	Spain	(91) 555 22 82	👁 ✎
9	Laurence Lebihan	Marseille	France	91.24.45.40	👁 ✎
10	Today 11/10/2017 Lincoln	Tsawassen	Canada	(604) 555-1010	👁 ✎
11	Victoria Ashworth	London	UK	(171) 555-1212	👁 ✎
12	Patricio Simpson	Buenos Aires	Argentina	(1) 135-5555	👁 ✎
13	Francisco Chang	México D.F.	Mexico	(5) 555-3392	👁 ✎
14	Yang Wang	Bern	Switzerland	0452-076545	👁 ✎
15	Pedro Afonso	Sao Paulo	Brazil	(11) 555-7647	👁 ✎
16	Elizabeth Brown	London	UK	(171) 555-2282	👁 ✎
17	Sven Ottlieb	Aachen	Germany	0241-039123	👁 ✎
18	Janine Labrune	Nantes	France	40.67.88.88	👁 ✎
19	Ann Devon	London	UK	(171) 555-0297	👁 ✎
20	Roland Mendel	Graz	Austria	7675-3425	👁 ✎

IV. EVALUACIÓN

1. ¿Por qué es importante el uso de Nodejs para el desarrollo frontend?
2. ¿Qué es npm?
3. ¿Qué es webpack?
4. ¿Qué significan para npm el -g y –-save-dev?