

**Tipo** : Guía de Enunciado  
**Capítulo** : Single Page Application (SPA) con React  
**Duración** : 90 minutos

---

## I. OBJETIVO

Implementación de Redux para el módulo de Login.

## 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

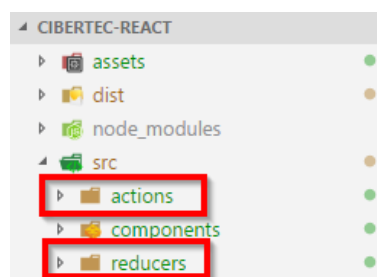
1. Abrir el proyecto “**Cibertec**” en Visual Studio 2017.
2. Nos ubicamos en el proyecto “**Cibertec.WebApi**”.
3. Procedemos a instalar el siguiente paquete:
  - Install-Package Microsoft.Owin.Cors
4. En el fichero “**Startup.cs**” agregamos la siguiente línea:  
`app.UseCors(CorsOptions.AllowAll);` (Guiarse por la imagen)

```
0 references | Cesar Velarde, 2 days ago | 2 authors, 6 changes | 0 exceptions
public void Configuration(IApplicationBuilder app)
{
    log4net.Config.XmlConfigurator.Configure();
    var log = log4net.LogManager.GetLogger(typeof(Startup));
    log.Debug("Logging is enabled");

    var config = new HttpConfiguration();
    config.Services.Replace(typeof(IExceptionHandler), new GlobalExceptionHandler());

    DIConfig.ConfigureInjector(config);
    app.UseCors(CorsOptions.AllowAll);
    TokenConfig.ConfigureOAuth(app, config);
    RouteConfig.Register(config);
    WebApiConfig.Configure(config);
    app.UseWebApi(config);
}
```

5. Volver a publicar la WebApi.
6. Abrir Visual Studio Code y ubicarnos en nuestra carpeta de “**Cibertec-React**”.
7. Procedemos a crear las carpetas “**actions**” y “**reducers**” como se muestra en la imagen:



8. En la carpeta “**assets**” creamos nuestro el fichero “**styles.css**” con el siguiente código:

```
.container-background {
    background: url('imgs/login.jpg') no-repeat center center
    fixed;
    background-size: cover;
}

.full-height-row {
    height: 100vh;
}

.panel-login {
    transform: translateY(35%);
    display: inline-block;
    width: 400px;
}

.panel-list {
    margin-top: 3rem;
    display: inline-block;
    width: 600px;
}

.panel-list--empty {
    margin-top: 20rem;
}

.login-header-img {
    height: 80px;
}

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

.page-not-found {
    color: #80d7ff;
    padding-top: 35vh;
    font-size: 7vw;
}

.page-not-found-small {
    color: transparent;
}
```

```
.list-icon {
  padding-right: .5rem;
  padding-left: .5rem;
  cursor: pointer;
}
```

9. En la carpeta “**actions**” creamos el fichero “**creators.tsx**” con el siguiente código:

```
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) });
  }
}
```

10. En la carpeta “**actions**” creamos el fichero “**types.tsx**” con el siguiente código:

```
export const GOT_TOKEN = 'GOT_TOKEN'
```

11. En la carpeta “**components**” creamos el fichero “**404.tsx**” con el siguiente código:

```
import * as React from 'react';

export default function NotFound () {
  return (
    <div className="container-fluid container-login">
      <div className="row full-height-row">
        <div className="col-sm-12">
          <div className="text-center">
            <h1 className="page-not-found">Page not
found :(
              <br/>
              <small className="page-not-found-
small">You need to go back</small>
            </h1>
          </div>
        </div>
      </div>
    </div>
  )
}
```

```

        </div>
      </div>
    </div>
  </div>
);
}

```

12. En la carpeta “**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.loginReducer.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);
    }

    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></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>
                                    <p className="login-token-
preview">{this.props.token}</p>
                                </div>
                            </div>
                        </div>
                    </div>
                </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);

```

13. En la carpeta “**reducers**” creamos el fichero “**authentication.tsx**” con el siguiente código:

```

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

export function loginReducer(state = INITIAL_STATE, action: any) {
    switch (action.type) {
        case types.GOT_TOKEN:
            return {...state, token: action.token}
    }

    return state;
}

```

14. En la carpeta “**reducers**” creamos el fichero “**rootreducer.tsx**” con el siguiente código:

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

const rootReducer = combineReducers({
  loginReducer
});

export default rootReducer;
```

15. En la carpeta “**src**” creamos el fichero “**defaultParams.tsx**” con el siguiente código:

```
export default function DefaultParams() {
  return {};
}
```

16. En la carpeta “**src**” creamos el fichero “**routes.tsx**” con el siguiente código:

```
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';

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="*" component={NotFound} />
        </Router>
      </Provider>
    );
  }
}
```

17. En el fichero “index.html” agregaremos la referencia a nuestra hoja de estilos como se muestra en la imagen:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7     <title>Cibertec React</title>
8
9     <link rel="stylesheet"
10    href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
11    integrity="sha384-BVYiiSIFeK1dGmJRAKycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
12    crossorigin="anonymous" />
13    <link rel="stylesheet" href="assets/styles.css" />
14   </head>
15   <body>
16     <div id="cibertec-app"></div>
17     <script src="./node_modules/react/dist/react.js"></script>
18     <script src="./node_modules/react-dom/dist/react-dom.js"></script>
19     <script src="./dist/bundle.js"></script>
20   </body>
21 </html>
```

18. 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\We
bDeveloper\Cibertec-React/tsconfig.json.

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

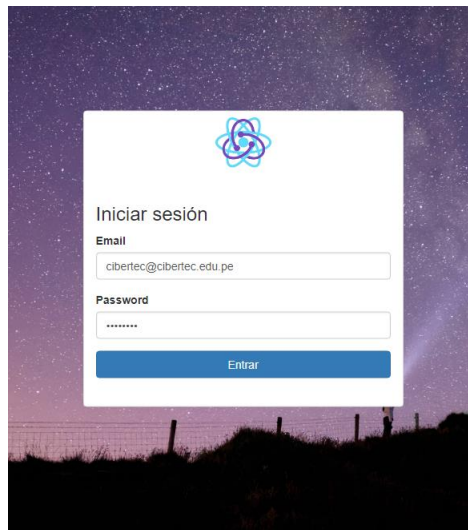
[at-loader] Ok, 0.129 sec.
Hash: 2d9a78428d417b2e180c
Version: webpack 3.8.1
Time: 1873ms

   Asset      Size  Chunks             Chunk Names
bundle.js    322 kB      0  [emitted] [big]  main
bundle.js.map 353 kB      0  [emitted]         main
[11] (webpack)/buildin/global.js 488 bytes {0} [built]
[47] ./src/actions/types.tsx 49 bytes {0} [built]
[53] ./src/index.tsx 270 bytes {0} [built]
[56] ./src/routes.tsx 1.59 kB {0} [built]
[98] (webpack)/buildin/module.js 495 bytes {0} [built]
[108] ./src/reducers/rootReducer.tsx 296 bytes {0} [built]
[109] ./src/reducers/authentication.tsx 655 bytes {0} [built]
[110] ./src/defaultParams.tsx 160 bytes {0} [built]
[112] ./src/actions/creators.tsx 789 bytes {0} [built]
+ 124 hidden modules

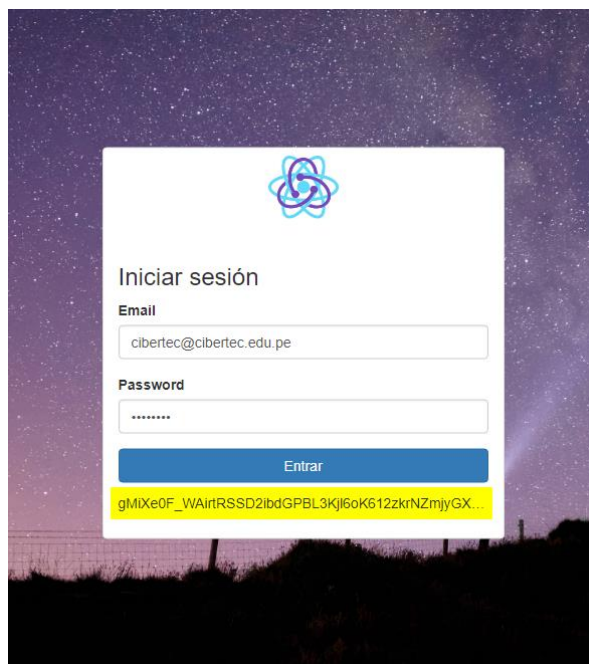
C:\Repos\WebDeveloper\Cibertec-React>
```



19. Abrimos el “index.html” desde nuestro navegador y visualizaremos lo siguiente:



20. Hacemos clic en entrar y visualizaremos que nuestro token aparece en la parte inferior de la misma ventana, eso confirmará que todo está funcionando correctamente:



#### 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?