

Xseed

Curso de React _ Technisys

Clase 15 -

Repaso y Ejercicio Final

by Diego Cáceres

Repaso

Repaso

- Clase 1 - Introduccion a React, JSX y Componentes
- Clase 2 - Tipos de Componentes, más sobre Props, manejo de State, keyword this
- Clase 3 - ES6 (Ecmascript 2015) - Conditional Rendering
- Clase 4 - Webpack, Babel, Proyecto desde cero, Lists, Props.children
- Clase 5 - PropTypes, DefaultProps, Forms en React
- Clase 6 - Lifecycle Methods, External Data Access, Presentational vs Container
- Clase 7 - Routing en React, High Order Components

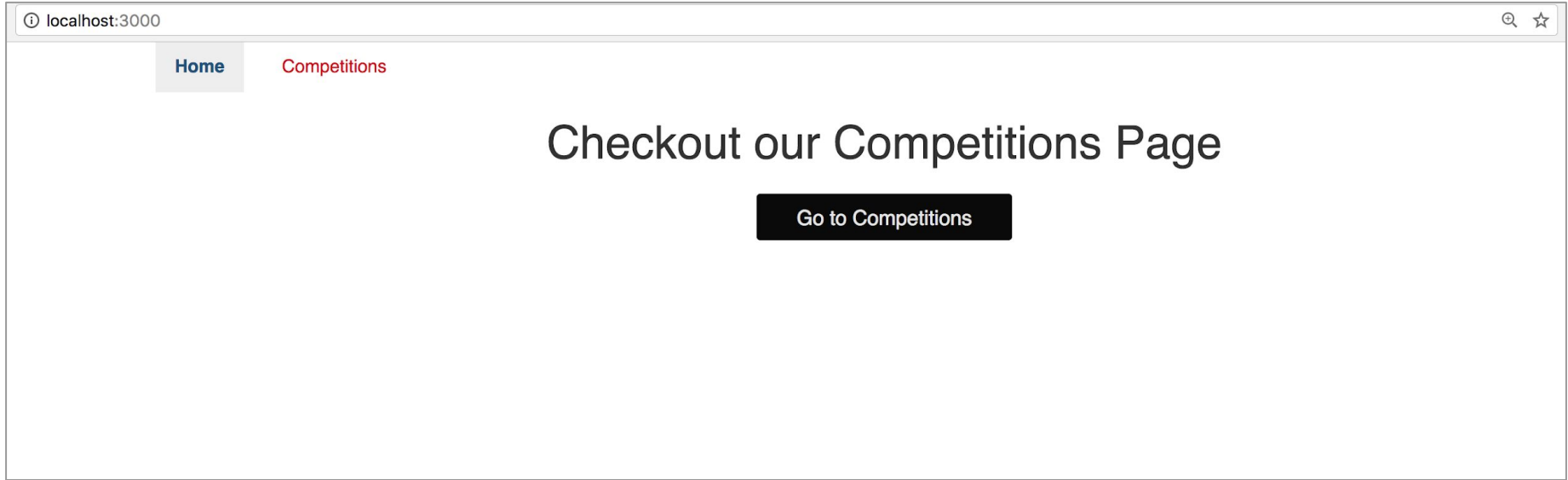
Repaso

- Clase 8 - Redux
- Clase 9 - Ejercicios Redux
- Clase 10 - Redux Middleware
- Clase 11 - Redux Thunk
- Clase 12 - Generators - Redux Sagas
- Clase 13 - Formik
- Clase 14 - React Context y Testing

Ejercicio Final: Soccer Center

Ejercicio: SoccerCenter

- Replicar el sitio web para consultar datos sobre competiciones de futbol SoccerCenter ([link](#)).
- Partir de un **proyecto base específico** para este ejercicio que ya trae las imágenes, estilos, y algunos componentes.
- El proyecto tiene instalado **React Bootstrap** ([documentación](#)).
- **Cuidar la prolijidad** (indentación, espaciado, nombre de variables, etc).
- El proyecto utiliza datos de una [api pública](#), por lo que lo mejor es que cada uno solicite una key para cambiar en el archivo src/services/config.js



localhost:3000/Competitions

Home Competitions

Available Competitions

Campeonato Brasileiro da Série A - BSA - 2017

Teams: 20
Progress in matches: 24 of 38
Last updated: 2017-09-19T01:10:08Z

[See positions table](#)

Premier League 2017/18 - PL - 2017

Teams: 20
Progress in matches: 6 of 38
Last updated: 2017-09-20T04:00:01Z

[See positions table](#)

Championship 2017/18 - ELC - 2017

Teams: 24
Progress in matches: 9 of 46
Last updated: 2017-09-19T07:00:02Z

[See positions table](#)

League One 2017/18 - EL1 - 2017




Teams: 24
Progress in matches: 5 of 46
Last updated: 2017-09-19T20:50:11Z

[See positions table](#)

[Home](#)[Competitions](#)

Tabla de Campeonato Brasileiro da Série A

Position	Name	Wins	Draws	Losses	Goals	Goals Agains	Goals Diff	Played Matches	Points	See Squad
1	Corinthians	16	5	3	34	13	21	24	53	+
2	Grêmio	13	4	7	40	21	19	24	43	+
3	Santos FC	11	8	5	25	16	9	24	41	+
4	Palmeiras	12	4	8	35	26	9	24	40	+
5	EC Flamengo	10	8	6	33	23	10	24	38	+
6	Cruzeiro	10	7	7	29	21	8	24	37	+
7	Botaogo	10	7	7	29	23	6	24	37	+
8	Atlético PR	9	7	8	29	27	2	24	34	+
9	Fluminense FC	7	10	7	34	34	0	24	31	+
10	Atlético Mineiro	8	7	9	26	28	-2	24	31	+
11	Vasco da Gama	9	4	11	24	35	-11	24	31	+
12	Sport Recife	8	5	11	30	36	-6	24	29	+
13	Avaí SC	7	8	9	15	27	-12	24	29	+
14	Ponte Preta	7	7	10	26	32	-6	24	28	+
15	Chapecoense	8	4	12	28	37	-9	24	28	+
16	Bahia	7	6	11	30	32	-2	24	27	+

18	Everton FC	1	1	3	2	10	-8	5	4	
19	AFC Bournemouth	1	0	4	3	9	-6	5	3	
20	Crystal Palace FC	0	0	5	0	8	-8	5	0	

Team Manchester United FC



Romelu Lukaku
Position: Centre-Forward
Nationality: Belgium



David de Gea
Position: Keeper
Nationality: Spain



Sergio Romero
Position: Keeper
Nationality: Argentina



Eric Bailly
Position: Centre-Back
Nationality: Cote d'Ivoire



Chris Smalling
Position: Centre-Back
Nationality: England



Marcos Rojo
Position: Centre-Back
Nationality: Argentina



Phil Jones
Position: Centre-Back
Nationality: England



Daley Blind
Position: Left-Back
Nationality: Netherlands



Luke Shaw
Position: Left-Back
Nationality: England



Matteo Darmian
Position: Right-Back
Nationality: Italy



Antonio Valencia
Position: Right-Back
Nationality: Ecuador



Michael Carrick
Position: Defensive Midfield
Nationality: England



Paul Pogba
Position: Central Midfield



Ander Herrera
Position: Central Midfield



Marouane Fellaini
Position: Central Midfield



Ashley Young
Position: Left Midfield



Henrikh Mkhitaryan
Position: Attacking Midfield



Juan Mata
Position: Attacking Midfield

Ejercicio

- Completar el componente `Nav` para que muestre los dos links.
- Implementar en `App.js` el ruteo de la App. Tiene que dibujar el componente `Nav` y luego las rutas para `Home`, `ListOfCompetitions` y `NotFound`.
- Completar luego el componente `ListOfCompetitions`.
- Para que se cargue la lista de competencias, van a tener que implementar las sagas correspondientes.
- Si hacen click en el botón "See positions table" están navegando a otra ruta, por lo que en `App.js` tienen que definir otra ruta que en el path reciba el id de la competición y dibuje el componente `Competition`.

Ejercicio

- Completar el componente `Competition`, que muestre el `Loader` si esta cargando y si no la tabla de posiciones. En el `componentDidMount` es necesario que dispatchee la acción `loadLeagueTableById` e implementar la saga correspondiente. También implementar la llamada a la api en `services.js`. También tienen que completar los cases en el reducer.
- Luego completar los datos que se le pasan a `PosTableRow` para poder ver la tabla completa.
- Luego completar `PosTableRow` para que se conecte con `Redux` y pueda emitir la accion `selectTeamId` cuando se seleccione un equipo para ver los jugadores.

Ejercicio

- Cuando se esté seleccionando el equipo y esto llegue a modificar el State del Store de Redux, agregar en el componente `Competition` que muestre `TeamSquad` cuando haya un equipo seleccionado.
- Tendrán que crear una saga para que cargue los jugadores de un equipo, cuando se seleccione uno (`selectTeamId`).
- Luego hacer que `TeamSquad` lea del state los players para mostrar para cada uno de ellos un componente `Player`.
- Completar el componente `Player` para que muestre los datos de cada jugador, y agregarle `PropTypes`.

Ejercicio – Opcionales

La API también brinda información sobre los fixtures de las competiciones (<http://api.football-data.org/v1/competitions/446/fixtures>). Se podría agregar una nueva sección o botón que muestre el fixture de una competición, tanto los partidos completados como los pendientes.

- Necesitarán una nueva saga para ir a buscar los Fixtures.
- Agregar el case en el reducer.
- Agregar el método en `services/index.js`.
- Crear al menos un nuevo componente conectado con esta info del Store y que la muestre.

Ejercicio – Opcionales

Además de mostrar todos los partidos, se podría filtrar la lista para ver los terminados o los pendientes. En el componente nuevo de Fixture pueden mostrar dos radio-buttons para elegir, u otra forma similar:

```
<div>
  <label style={{ width: "100%" }}> Finished Games
    <input style={{ marginLeft: "10px", marginRight: "10px" }}
      type="radio" name="fixture-type" checked={this.state.seeFinished}
      onChange={() => this.handleClick(true)} />
  </label>
  <label style={{ width: "100%" }}> Pending Games
    <input style={{ marginLeft: "10px", marginRight: "10px" }}
      type="radio" name="fixture-type" checked={!this.state.seeFinished}
      onChange={() => this.handleClick(false)} />
  </label>
</div>
```