

BARTECA

Andrea Castilla Cocera
2º DAM



ÍNDICE

- INTRODCUCCIÓN
- TECNOLOGÍAS
 - BACKEND
 - FRONTEND
- DESPLIEGUE
- CONCLUSIONES



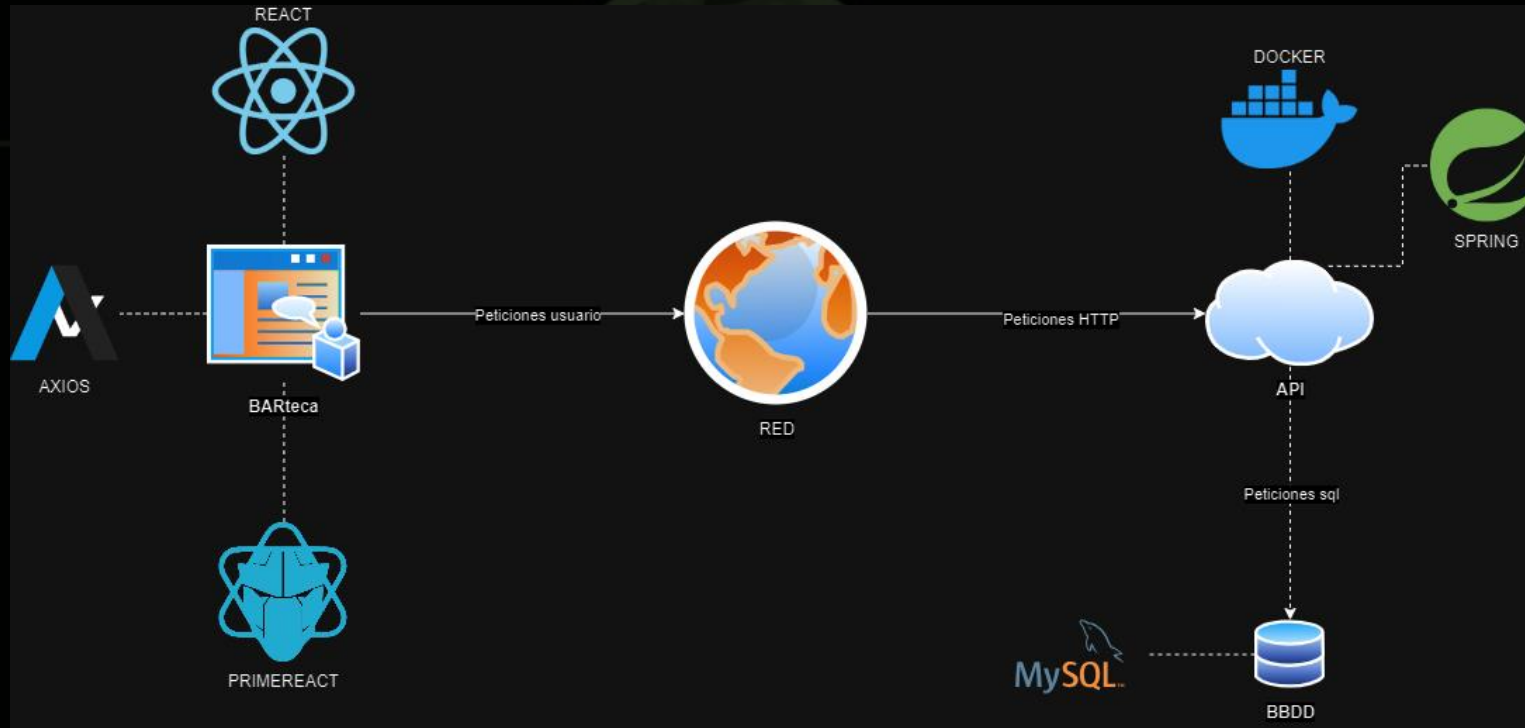
INTRODUCCIÓN

BARteca surge de la idea de una biblioteca de bares (BAR – [biblio]teca), la idea principal es parecida, ya que se basa en una aplicación dónde el usuario busca bares y puede hacer una reserva, así como los libros.

Se basa además en los foros, donde los usuarios pueden hacer entradas y diferentes acciones entre estas, pero no deja de ser algo dedicado al usuario.

La única ‘regla’ que se cumple para acceder a este foro es que el usuario tenga una cuenta, si no solo verá un logueo, dándole la opción de registrarse.

TECNOLOGÍAS



TECNOLOGÍAS → BACKEND

- SPRING

¿Qué es?: Spring es un framework de código abierto que da soporte al desarrollo de aplicaciones basadas en Java mediante el uso de objetos sencillos.

Características:

- ☐ Funciona sobre JVM (traduce bytes de Java a instrucciones nativas del host)
- ☐ Estructura modular (estructura flexible)
- ☐ Permite hacer todo tipo de aplicaciones



TECNOLOGÍAS → BACKEND

SPRING INITIALIZR

The screenshot shows the Spring Initializr web application. It has a dark theme with a sidebar on the left containing a hamburger menu and a gear icon. The main content area is divided into three sections: 'Project', 'Language', and 'Dependencies'. The 'Project' section has radio buttons for 'Gradle - Groovy' (selected), 'Gradle - Kotlin', and 'Maven'. The 'Language' section has radio buttons for 'Java' (selected), 'Kotlin', and 'Groovy'. The 'Dependencies' section has a text input field and a button 'ADD ... CTRL + B'. Below these sections is the 'Spring Boot' section with radio buttons for versions '3.3.1 (SNAPSHOT)', '3.3.0' (selected), '3.2.7 (SNAPSHOT)', and '3.2.6'. The 'Project Metadata' section contains form fields for 'Group' (com.example), 'Artifact' (demo), 'Name' (demo), 'Description' (Demo project for Spring Boot), and 'Package name' (com.example.demo). At the bottom, there are three buttons: 'GENERATE CTRL + G', 'EXPLORE CTRL + SPACE', and 'SHARE...'. The bottom right corner has a small icon of a person.

```
@Entity
@Data
@NoArgsConstructor
public class Usuario {
```

```
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
```

```
    @Column(length = 40, nullable = false)
    private String nombre;
```

```
    @Column(length = 20, nullable = false)
    private String username;
```

```
    @Column(length = 75, nullable = false)
    private String password;
```

```
    @Column(unique = true, length = 70, nullable = false)
    private String correo;
```

```
    @Column(length = 9, nullable = false)
    private String telefono;
```

```
}
```

```
@Repository
public interface BarRepo extends JpaRepository<Bar, Long>{
}
```

MODELO

REPOSITORIO

TECNOLOGÍAS → BACKEND

CONTROLADOR

```
@RestController
@RequestMapping("/barteca")
public class BarController {
    @Autowired
    BarRepo repositorioBar;

    @GetMapping("bar")
    List<Bar> findAll(){
        return repositorioBar.findAll();
    }

    @GetMapping("bar/{id}")
    Bar findById(@PathVariable(value = "id") Long id){
        return repositorioBar.findById(id).get();
    }
    ...
}
```

WEBCONFIG

```
@EnableWebSecurity
@Configuration
public class WebConfig implements WebMvcConfigurer {

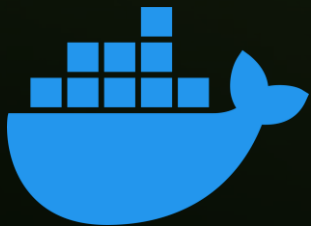
    @Autowired
    DataSource dataSource;

    @Bean
    SecurityFilterChain filter(HttpSecurity http) throws Exception {
        return http
            .authorizeHttpRequests((requests) -> requests
                .requestMatchers("/barteca/**", "/usuario/**", "/bar/**",
                    "/reserva/**", "/login/**").permitAll()
            )
            .exceptionHandling((exception) -> exception
                .accessDeniedPage("/denegado"))
            .formLogin((formLogin) -> formLogin
                .permitAll())
            .rememberMe(Customizer.withDefaults())
            .logout((logout) -> logout
                .invalidateHttpSession(true)
                .logoutSuccessUrl("/")
                .permitAll())
            .csrf((csrf) -> csrf.disable())
            .cors(Customizer.withDefaults())
            .build();
    }
}
```

TECNOLOGÍAS → BACKEND

DOCKER

¿Qué es?: Plataforma con la que podemos automatizar la implementación dentro de contenedores (entornos ligeros, portátiles y autosuficientes) con todo lo necesario para ejecutar la aplicación.



```
docker-compose up -d
```

```
docker build -t
```

```
docker run -d
```

```
version: '3.1'
```

```
services:
```

```
db:
```

```
image: mysql
```

```
restart: "no"
```

```
environment:
```

```
MYSQL_ROOT_USERNAME: ${MYSQL_ROOT_USERNAME}
```

```
MYSQL_ROOT_PASSWORD: ${MYSQL_ROOT_PASSWORD}
```

```
networks:
```

```
- mysql_network
```

```
ports:
```

```
- 3306:3306
```

```
volumes:
```

```
- mysql_data:/var/lib/mysql
```

```
- ./init.sql:/docker-entrypoint-initdb.d/init.sql
```

```
adminer:
```

```
image: adminer
```

```
restart: "no"
```

```
networks:
```

```
- mysql_network
```

```
ports:
```

```
- 8181:8080
```

```
networks:
```

```
mysql_network:
```

```
volumes:
```

```
mysql_data:
```


TECNOLOGÍAS → FRONTEND

- REACT

¿Qué es?: Es una biblioteca de JavaScript desarrollada por Facebook que se utiliza para construir interfaces de usuario interactivas y reutilizables.

Características:

- ☐ Componentes reutilizables
- ☐ Virtual DOM (cambios compara este con el real)
- ☐ JSX (HTML + JS, estructura y apariencia componentes)



TECNOLOGÍAS → FRONTEND

- AXIOS

¿Qué es?: Es una biblioteca de *JavaScript* que permite realizar solicitudes *HTTP* desde una aplicación web.

Características:

- ☐ Sintaxis sencilla
- ☐ Admite promesas (basada en Promise JS)
- ☐ Compatibilidad con navegadores



TECNOLOGÍAS



FRONTEND

```
const URL = 'http://localhost:8080/barteca';
```

```
const getBares = async (state) => {  
  const token = localStorage.getItem('token'); // Obtener el token  
  if (!token) {  
    console.error('No token found, redirecting to login');  
    window.location.href = "/";  
    return;  
  }  
}
```

```
try {  
  const req = await axios.get(URL + '/bar', {  
    headers: {  
      'Authorization': `Bearer ${token}`  
    },  
    withCredentials: true  
  });  
  console.log(req);  
  state(req.data);  
} catch (error) {  
  console.error('Error fetching data:', error);  
  if (error.response && error.response.status === 401) {  
    window.location.href = "/";  
  }  
}  
};
```

```
const nuevoBar = async (data) => {  
  try {  
    const response = await axios.post(`${URL}/bar`, data);  
    console.log('Registro exitoso:', response.data);  
  } catch (error) {  
    console.error('Error registrando usuario:', error);  
  }  
};
```



TECNOLOGÍAS



FRONTEND

- EMAIL JS

¿Qué es?: Es una biblioteca que facilita el envío de correos electrónicos desde aplicaciones JavaScript sin necesidad de un servidor backend.

Características:

- ☐ Facilidad de uso
- ☐ Compatibilidad (Gmail, Outlook, Yahoo,...)
- ☐ Plantillas
- ☐ Soporte archivos adjuntos
- ☐ Compatibilidad con diferentes lenguajes
- ☐ Eventos y callbacks



TECNOLOGÍAS



FRONTEND

```
import React, { useRef } from 'react';
import emailjs from '@emailjs/browser';

export const ContactUs = () => {
  const form = useRef();

  const sendEmail = (e) => {
    e.preventDefault();

    emailjs
      .sendForm('YOUR_SERVICE_ID', 'YOUR_TEMPLATE_ID', form.current, {
        publicKey: 'YOUR_PUBLIC_KEY',
      })
      .then(
        () => {
          console.log('SUCCESS!');
        },
        (error) => {
          console.log('FAILED...', error.text);
        }
      );
  };
};
```

```
return (
  <form ref={form} onSubmit={sendEmail}>
    <label>Name</label>
    <input type="text" name="user_name" />
    <label>Email</label>
    <input type="email" name="user_email" />
    <label>Message</label>
    <textarea name="message" />
    <input type="submit" value="Send" />
  </form>
);
```



TECNOLOGÍAS → FRONTEND

- PRIMEREACT

¿Qué es?: Es una librería de componentes de React, esta nos facilita el uso de componentes generales y su diseño, como son botones, popups, inputs, etc.

Con esta también podemos crear formularios con validación, esto los encontramos en el registro de un usuario, de un bar y de una reserva.



TECNOLOGÍAS



FRONTEND

```
const items = [
  {
    icon: 'pi pi-shop',
    command: () => {
      navigate('/bares');
    }
  }, {
    icon: 'pi pi-book',
    command: () => {
      navigate('/reservas');
    }
  }, {
    icon: 'pi pi-question-circle',
    command: () => {
      navigate('/about');
    }
  }, {
    icon: 'pi pi-sign-out',
    command: () => {
      logout();
      navigate('/');
    }
  }
];
```

```
<Route path="/bares" element={<div style={{ position: 'relative'}}>
  <BarList/>
  <Tooltip target=".speeddial-top-rigth .p-speeddial-action" />
  <SpeedDial model={([...itemsadd, ...items]} direction="down" style={{ right: 0, bottom: 0 }}
    className="speeddial-top-rigth rigth-0 top-0"
    buttonClassName="p-button-help" />
  <ScrollTop threshold={100} behavior="smooth" />
</div>
```



DESPLIEGUE

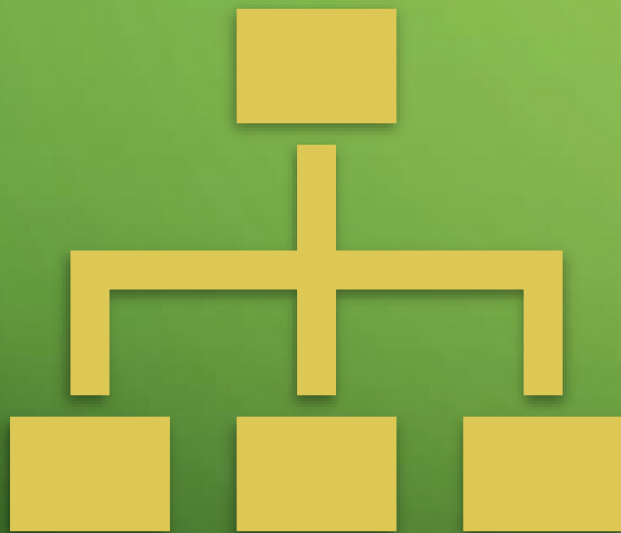
Para el despliegue podría haber utilizado Vercel, plataforma popular que fue creada por el mismo equipo de Next.js y ofrece: integración continua, despliegue automatizado y una CDN global.

Otra opción sería Netlify, esta ofrece despliegues continuos, integración con git y funciones serverless.

Estas opciones son las más recomendadas por su fácil uso, integración continua y características avanzadas.

Además, también contamos con GitHub Pages, una opción para proyectos más pequeños o personales, también es gratuita y permite desplegar aplicaciones estáticas directamente desde un repositorio de GitHub.

CONCLUSIONES



- Creación de roles (Usuario, Administrador, Dueño)
 - Creación de diferentes páginas según rol
 - Creación de estado de reservas
- Refactorización de código y mejora

The background is a solid green color with a subtle gradient. In the four corners, there are white line art illustrations of circuit boards or neural networks, featuring lines and small circles.

FIN

Proyecto realizado por:
Andrea Castilla Cocera