

Manual de instalación y configuración del sistema de información

Presentado por:

Mayra Alexandra Rojas Peña

Dayana María Murillo Valderrama

Leonardo Augusto Salinas Sánchez

Ficha: 2175913

Análisis y desarrollo de sistemas de información

Servicio nacional de aprendizaje

Febrero 22 de 2022

Contenido

Introducción.....	3
Requisitos mínimos.	4
Instalación del programa y componentes.....	5
Proyecto BackEnd.....	6
Ejecución del proyecto BackEnd	9
Ejecución proyecto Front	10
Consideraciones adicionales.....	12
Referencias.....	14

Introducción.

El objetivo de este documento es servir de guía orientada al equipo técnico para ejecutar la instalación y correcto despliegue del aplicativo desarrollado llamado “Oak Trees”.

Por medio de esta guía se explicará la instalación de los aplicativos necesarios, la ejecución de las aplicaciones desarrolladas y cada uno de sus componentes, así como otro tipo de información que puede ser útil para un futuro uso.

Este manual es de suma importancia ya que contiene la descripción del producto final y el seguir cada uno de los pasos tal cual como están plasmados puede evitar futuros errores de ejecución.

Requisitos mínimos.

Hardware:

- Computador con mínimo 4GB de memoria RAM
- Mínimo 100GB de almacenamiento disponible en disco

Software:

- Sistema operativo Windows 10 en adelante
- MySQL Workbench o XAMP
- IntelliJ
- Visual Studio Code
- NodeJS
- Angular CLI
- OpenJDK 11
- Git
- Heroku

Instalación del programa y componentes.

Instalación proyecto para desarrollo

Nuestro proyecto, el sitio web Oak Trees lo desarrollamos con ayuda con GitHub y se encuentra en el siguiente repositorio: <https://github.com/sacnix/oaktrees>

El proyecto fue desarrollado separando el BackEnd y el FrontEnd, es por esto que en el comprimido podemos encontrar 2 carpetas:

- **oakFront: proyecto Front**
- **oaktrees: proyecto Back**

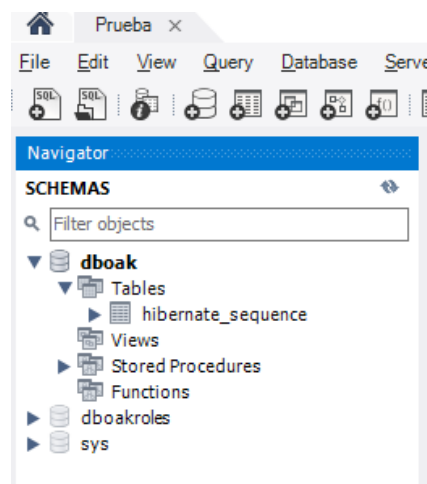
Para el BackEnd utilizamos la tecnología de Java Sprint Boot, para el FrontEnd utilizamos Angular con HTML y Bootstrap y para la base de datos utilizamos MySQL pero utilizamos Hibernate con JPA de SpringBoot para la creación de la gestión de la BD.

El proyecto utiliza una base de datos creada en JawsDB MySQL el cual es utilizado como plugin en Heroku para su despliegue. Pero para ambiente de desarrollo podemos crear la base de datos en local.

Para la ejecución del proyecto en local se requiere inicialmente crear la Base de datos en un gestor que soporte bases de datos MySQL, recomendamos gestores como MySQL Workbench o XAMPP.

1. Copiar, pegar y ejecutar los scripts que se encuentran en el archivo llamado **scripts.sql** en el gestor de BD.
2. Comprobar que la base de datos se encuentre correctamente creada

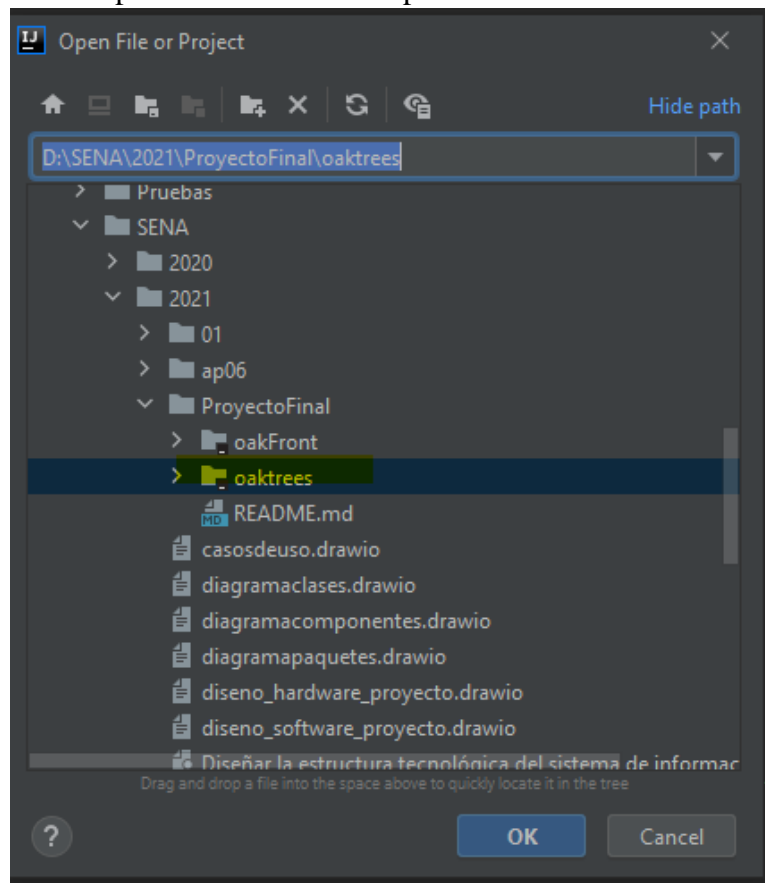
Ejemplo en MySQL WorkBench:



Una vez instalada la BD procedemos con la ejecución del proyecto. Iniciamos ejecutando el proyecto Back y luego el Front.

Proyecto BackEnd

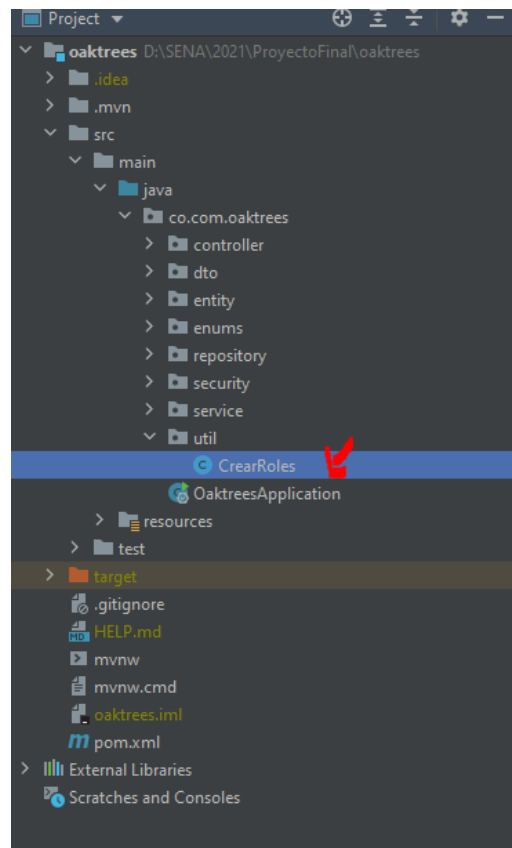
1. Debemos abrir la carpeta llamada **oaktrees** en un IDE que soporte Java y Java Spring Boot, recomendamos utilizar IntelliJ versión community
2. File – Open – Buscamos la carpeta oaktrees



IMPORTANTE: los siguientes pasos serán necesarios ejecutarlos SOLO la primera vez que se ejecutará el proyecto ya que este paso se encarga de crear los roles, usuario admin y demás datos necesarios en la BD.

Una vez cargado el proyecto Back en el IDE buscaremos el archivo en la siguiente ruta:
oaktrees\src\main\java\co\com\oaktrees\util

A. Abrimos el archivo llamado CrearRoles.java



El archivo se verá de la siguiente manera:

```
CatalogoController.java  CrearRoles.java  CategoriaController.java  Producto.java  ProductoService.java  ProductoRepository.java
//oaktrees.com
EstadoPedidoService estadoPedidoService;

@Autowired
PersonaService personaService;

@Override
public void run(String... args) throws Exception {
    /** Rol rolAdmin = new Rol(RolNombre.ROL_ADMIN);
    Rol rolUser = new Rol(RolNombre.ROL_USER);
    Rol rolVendedor = new Rol(RolNombre.ROL_VENDEDOR);
    rolService.save(rolAdmin);
    rolService.save(rolUser);
    rolService.save(rolVendedor);

    TipoEntrega tienda = new TipoEntrega(TipoEntregaNombre.TIENDA);
    TipoEntrega domicilio = new TipoEntrega(TipoEntregaNombre.DOMICILIO);
    tipoEntregaService.save(tienda);
    tipoEntregaService.save(domicilio);

    EstadoPedido estadoEntregado = new EstadoPedido(EstadoPedidoNombre.ENTREGADO);
    EstadoPedido estadoEntrega = new EstadoPedido(EstadoPedidoNombre.EN_ENTREGA);
    EstadoPedido estadoRecibido = new EstadoPedido(EstadoPedidoNombre.RECIBIDO);
    EstadoPedido estadoCancelado = new EstadoPedido(EstadoPedidoNombre.CANCELADO);
    EstadoPedido estadoListo = new EstadoPedido(EstadoPedidoNombre.LISTO);
    estadoPedidoService.save(estadoEntregado);
    estadoPedidoService.save(estadoEntrega);
    estadoPedidoService.save(estadoRecibido);
    estadoPedidoService.save(estadoCancelado);
    estadoPedidoService.save(estadoListo);

    Persona personaAdmin = new Persona("admin", "12345", "admin@gmail.com", "$2a$10$5hIDtX8JhK9Ame6.laFc.IQgvhK1VZU7h55Hozh2Q88xx.fPEJ2e");
    Set<Rol> roles = new HashSet<>();
    roles.add(rolService.getByName(RolNombre.ROL_ADMIN).get());
    personaAdmin.setRoles(roles);
    personaService.save(personaAdmin);**/
}
```

B. Es necesario que quitemos el comentario de todo este bloque, esto basta con eliminar el “/**” al inicio de la línea 38 y el “**/” al final de la línea 65.

Quedaría así:

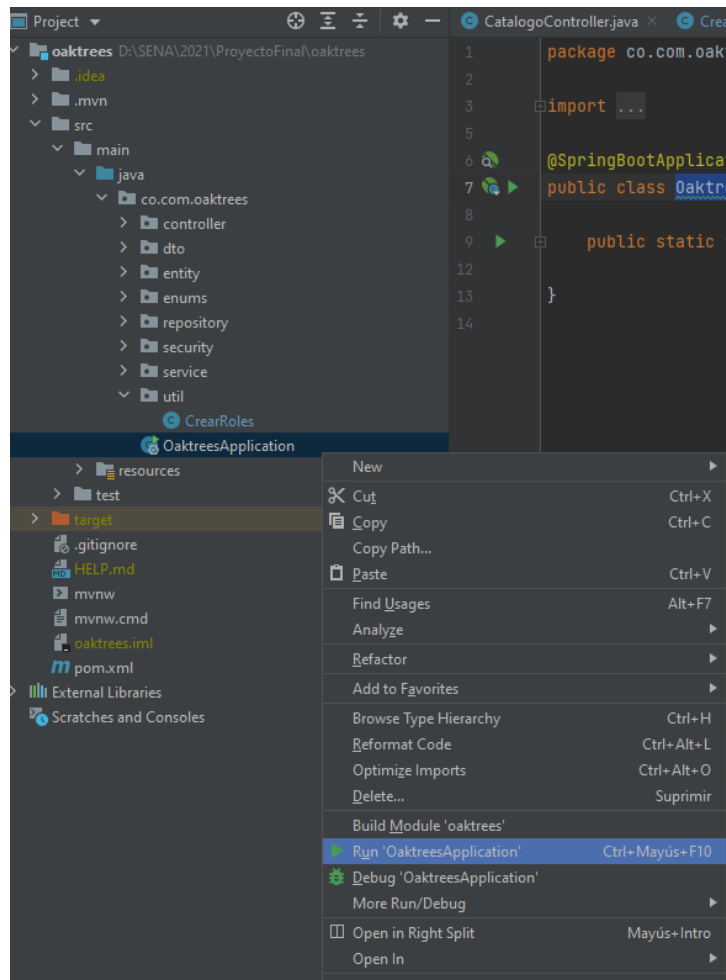
```
36      @Override
37      public void run(String... args) throws Exception {
38          Rol rolAdmin = new Rol(RolNombre.ROL_ADMIN);
39          Rol rolUser = new Rol(RolNombre.ROL_USER);
40          Rol rolVendedor = new Rol(RolNombre.ROL_VENDEDOR);
41          rolService.save(rolAdmin);
42          rolService.save(rolUser);
43          rolService.save(rolVendedor);
44
45          TipoEntrega tienda = new TipoEntrega(TipoEntregaNombre.TIENDA);
46          TipoEntrega domicilio = new TipoEntrega(TipoEntregaNombre.DOMICILIO);
47          tipoEntregaService.save(tienda);
48          tipoEntregaService.save(domicilio);
49
50          EstadoPedido estadoEntregado = new EstadoPedido(EstadoPedidoNombre.ENTREGADO);
51          EstadoPedido estadoEntrega = new EstadoPedido(EstadoPedidoNombre.EN_ENTREGA);
52          EstadoPedido estadoRecibido = new EstadoPedido(EstadoPedidoNombre.RECIBIDO);
53          EstadoPedido estadoCancelado = new EstadoPedido(EstadoPedidoNombre.CANCELADO);
54          EstadoPedido estadoListo = new EstadoPedido(EstadoPedidoNombre.LISTO);
55          estadoPedidoService.save(estadoEntregado);
56          estadoPedidoService.save(estadoEntrega);
57          estadoPedidoService.save(estadoRecibido);
58          estadoPedidoService.save(estadoCancelado);
59          estadoPedidoService.save(estadoListo);
60
61          Persona personaAdmin = new Persona( nombre: "admin", telefono: "12345", correo: "admin@gmail.com", clave: "$");
62          Set<Rol> roles = new HashSet<>();
63          roles.add(rolService.getByNombre(RolNombre.ROL_ADMIN).get());
64          personaAdmin.setRoles(roles);
65          personaService.save(personaAdmin);
66
67      }
```

- C. Procedemos con el paso de la ejecución del proyecto
- D. Después de ejecutar el proyecto debemos finalizar la ejecución y devolver los cambios que hicimos en la clase CrearRoles.

IMPORTANTE: al haberse ejecutado el proyecto por primera vez es necesario volver a esta clase y volver a poner el código de comentario que eliminamos para que el bloque quede comentado nuevamente.

Ejecución del proyecto BackEnd

1. Para esto desde el IDE en la vista de proyecto daremos clic derecho sobre la clase llamada *OaktreesApplication*
2. Seleccionamos la opción Run '*OaktreesApplication*'

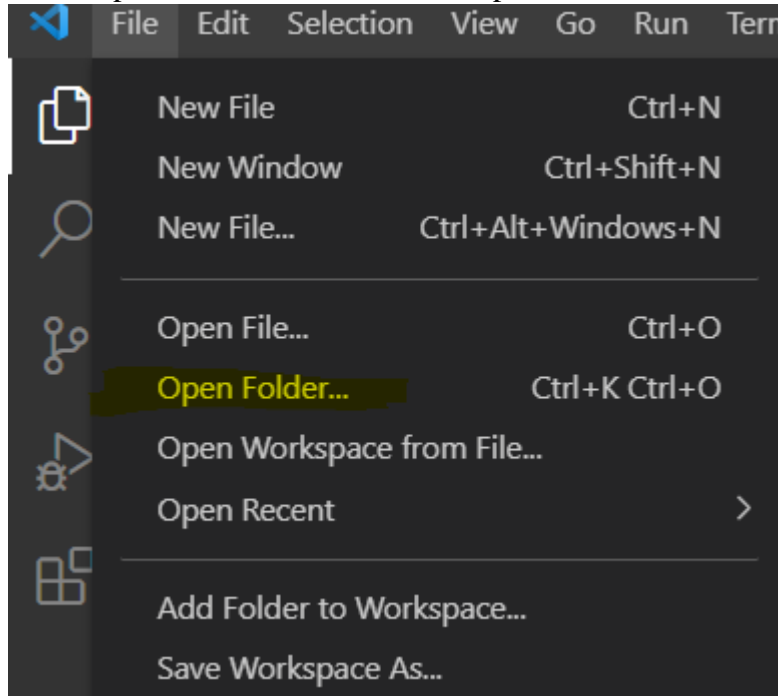


La consola mostrará la ejecución, cuando se muestre el siguiente mensaje resaltado en amarillo estará iniciado nuestro Back de manera exitosa:

```
2021-11-22 20:27:59.899 INFO 7672 --- [ restartedMain] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction
2021-11-22 20:27:59.908 INFO 7672 --- [ restartedMain] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2021-11-22 20:28:00.849 INFO 7672 --- [ restartedMain] o.s.s.web.DefaultSecurityFilterChain : Will secure any request with [org.springframework.security.web.context.request
2021-11-22 20:28:01.040 INFO 7672 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2021-11-22 20:28:01.092 INFO 7672 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2021-11-22 20:28:01.102 INFO 7672 --- [ restartedMain] co.com.oaktrees.OaktreesApplication : Started OaktreesApplication in 27.173 seconds (JVM running for 28.046)
```

Ejecución proyecto Front

1. Debemos abrir la carpeta llamada **oakFront** en un IDE que soporte TypeScript, Angular y HTML. Recomendamos Visual Studio Code.
2. File – Open Folder – Buscamos la carpeta **oakFront**



3. Debemos ejecutar el comando de instalación de los módulos, para esto abrimos la terminal de Visual Studio que está en el menú superior y ejecutamos el comando “npm install”

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS D:\SENA\2021\ProyectoFinal\oakFront> npm install
```

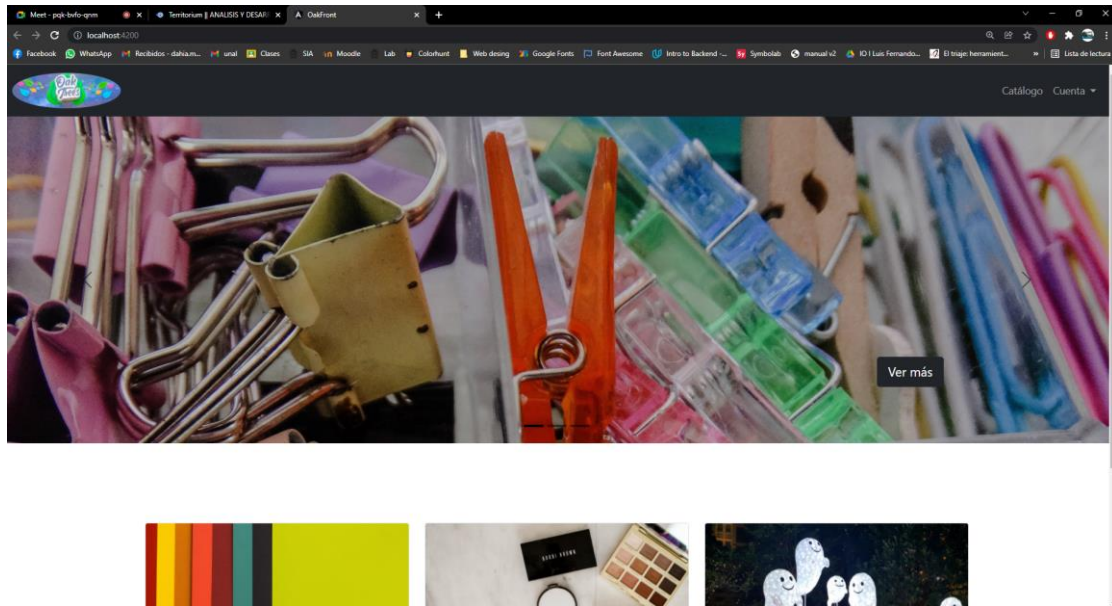
4. Una vez finalizada la instalación de los módulos, procedemos a iniciar el proyecto Front, ejecutando el comando “ng serve -o” en la terminal.

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS D:\SENA\2021\ProyectoFinal\oakFront> ng serve -o
```

5. Con esto el navegador se lanzará y veremos el sitio web de Oak Trees



Consideraciones adicionales.

Los pasos anteriormente mencionados son para la instalación y ejecución del proyecto en Local. Para realizar el despliegue del proyecto seguiremos estos pasos:

Despliegue de aplicación web

El despliegue se realizará en Heroku por lo que requerimos de la instalación de GIT en el computador y adicional de Heroku:

[Enlace para descargar Heroku](#)

Podemos verificar que tenemos Heroku correctamente instalado con el comando *heroku version* en CMD:

```
S:\SENA\2021\OakTreesFinal\oakFront>heroku version
» Warning: heroku update available from 7.53.0 to 7.59.2.
heroku/7.53.0 win32-x64 node-v12.21.0

S:\SENA\2021\OakTreesFinal\oakFront>
```

IMPORTANTE: este proceso debe repetirse tanto para el proyecto back como front ya que son 2 aplicaciones diferentes.

Procedemos a acceder a la carpeta donde se encuentra ubicado el proyecto a desplegar e iniciamos una terminal allí

- Ejecutar el comando *heroku login*
- Iniciar sesión en Heroku
- Ejecutar el siguiente comando para el proyecto **back** *heroku git:clone -a oak-trees-spring*.
- Ejecutar el siguiente comando para el proyecto **back** *heroku git:clone -a oak-trees-angular*

IMPORTANTE: este solo se ejecuta la primera vez que se realizará despliegue en la máquina para clonar el repositorio.

- Ejecutar el comando *cd oak-trees-spring* para el proyecto back o *cd oak-trees-angular* para el proyecto front

En este punto tenemos el repositorio de Heroku clonado. Si se requieren realizar cambios al proyecto, realizarlos y ejecutar los siguientes comandos para desplegar con los nuevos cambios:

- `git add .`
- `git commit -am "comentario del commit"`
- `git push heroku master`

Verificar que el despliegue se haya completado y el correcto funcionamiento de la aplicación en el siguiente enlace: <https://oak-trees-angular.herokuapp.com/>

Referencias.

https://sena.territorio.la/content/index.php/institucion/Titulada/institution/SENA/Tecnologia/228106/Contenido/DocArtic/AP9_OA_ManTec.pdf

<https://www.heroku.com/platform>