Informe de la aplicación

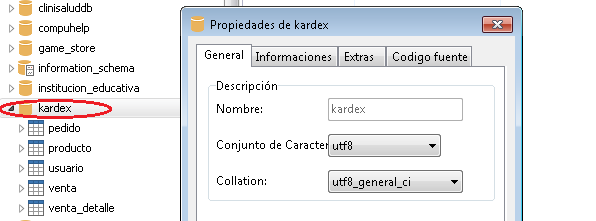
El objetivo del documento es describir lo más preciso posible y de una manera detallada la forma en que se fue construida la aplicación y las herramientas que se utilizó.

# Herramienta y tecnología de desarrollo

* Netbeans versión 8.2 con plugin typescript.
* JDK – JRE 1.8
* Maven versión 3.5.0 (apache-maven-3.5.0)
* WidFly Application Server (jboss-eap-7.0)
* MySQL Server 5.0.2 , MySQL Front
* Node JS v8.11.3, Angular IO, Angular CLI v6.0.8, npm v6.4.1
* Capa Front End: HTML5, Angular 6, PrimeNg 6.
* Capa Back End: Rest JAX-RS, EJB 3.0, Hibernate JPA 2.1, Orika Java Bean Mapping, log4j 1.2.17

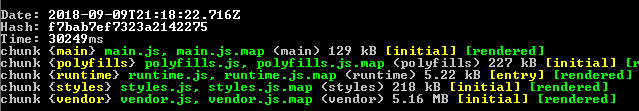
# Instalación y despliegue de la aplicación

* Aplicación se encuentra en repositorio publico github en la siguiente ruta [**https://github.com/wilmercado87/** **kardex**](https://github.com/wilmercado87/hotels-app)
* Descargar kardex-master.zip, descomprimir y abrirlo en el IDE Netbeans 8.2
* El IDE de estar configurado con Maven 3.0.5, WildFly Application Server (Jboss EAP 7) y JDK 1.8
* Configurar Jboss EAP 7 en la ruta raíz del pc recomendado C:\
* Configurar las variables de entorno :
  + EAP\_HOME C:\JBoss\jboss-eap-7.0
  + JAVA\_HOME C:\Program Files\Java\jdk1.8.0\_144
  + JRE\_HOME C:\Program Files\Java\jre1.8.0\_144
  + M2\_HOME C:\apache-maven-3.5.0
  + MAVEN\_HOME C:\apache-maven-3.5.0
* Configurar el driver mysql copiando en el servidor el jar **mysql-connector-java-5.1.17.jar** en la ruta C:\JBoss\jboss-eap-7.0\modules\system\layers\base\com\mysql\main y el archivo module.xml **ruta de entrega: README\driver mysql**
* Configurar el Data Source del servidor modificando el archivo C:\JBoss\jboss-eap-7.0\standalone\configuration\standalone-full.xml, copiar lo que está en el archivo **README\data-source\data-source.xml** y agregarlo en los tags <datasources><datasources > del standalone-full.xml.
* Crear la base de datos **kardex**

****

* Crear las tablas importando el script sql **ruta de entrega README\sql\kardex.sql**
* Iniciar el servidor jboss-eap-7.0 desde IDE NetBeans “recomendado”.
* Ejecutar consola cmd de comando **npm install npm@latest** última versión.
* Ejecutar consola cmd de comando **npm install @angular/cli@latest** última versión.
* Ejecutar el consola cmd el siguiente comando en la ruta de aplicación kardex **C:\kardex\kardex-web\src\main\webapp\kardexfront**  **npm --save install**
* Dentro de la misma ruta anterior ejecutar el comando **npm run build**.

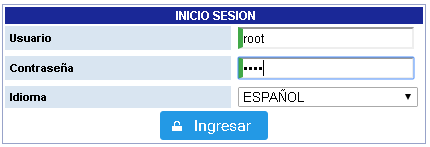
****

****

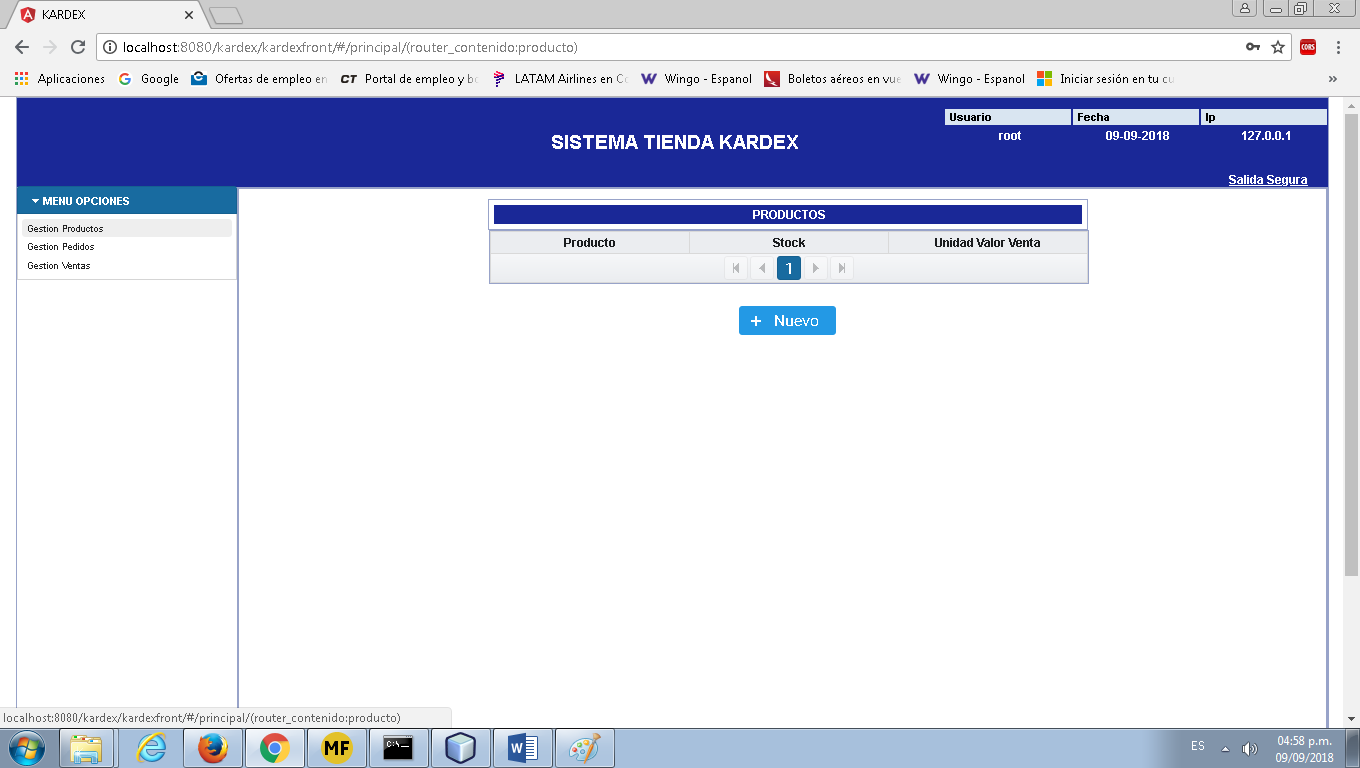
* Desde IDE NetBeans clic derecho **clean and build** en el módulo **kardex** para iniciar con el compilado y empaquetado del war, este automáticamente se copiara en la ruta **C:\JBoss\jboss-eap-7.0\standalone\deployments\ kardex-web-1.0-SNAPSHOT.war**, esperar que el archivo despliegue **kardex-web-1.0-SNAPSHOT.war.deployed**.
* En caso de fallar por alguna configuración hago entrega del war listo para ser desplegado en el servidor jboss-eap-7.0 en la ruta de entrega **README\war\kardex-web-1.0-SNAPSHOT.war**
* Desplegada la aplicación en el servidor iniciar en el navegador con la ruta: [**http://localhost:8080/kardex/kardexfront**](http://localhost:8080/kardex/kardexfront)
* **Inicio Sesión: usuario defecto: root – Contraseña: 8704**

****

****

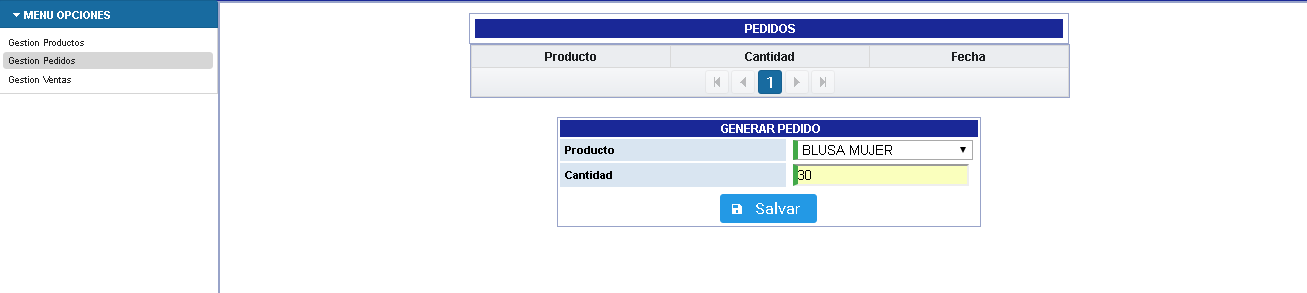
****

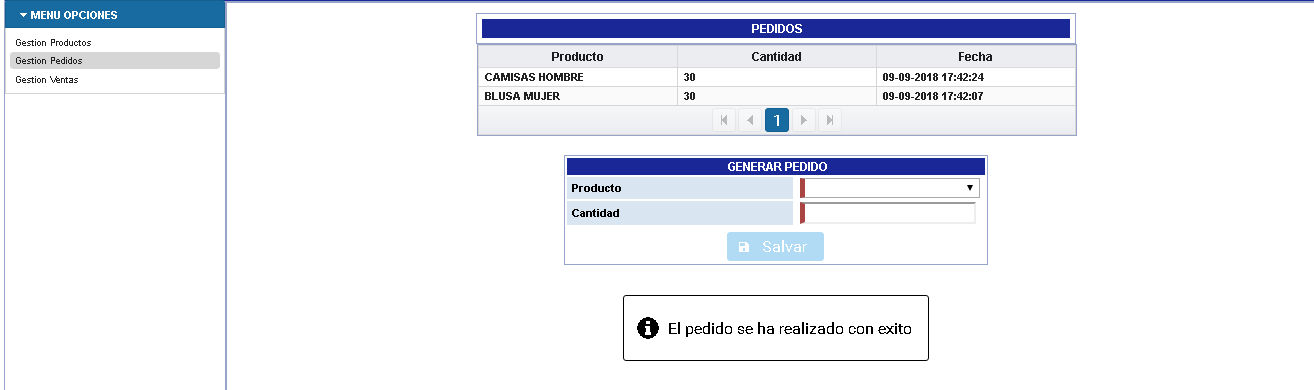
****



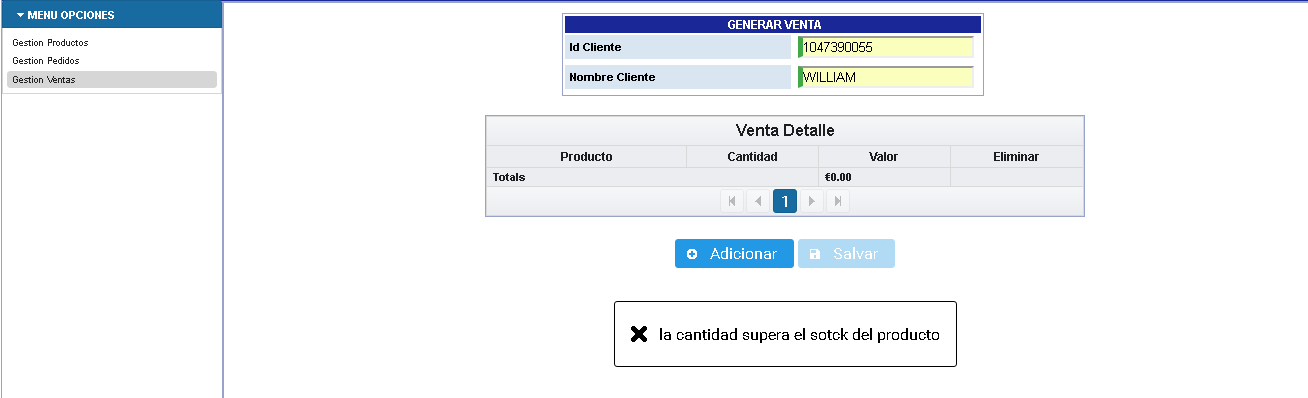
****

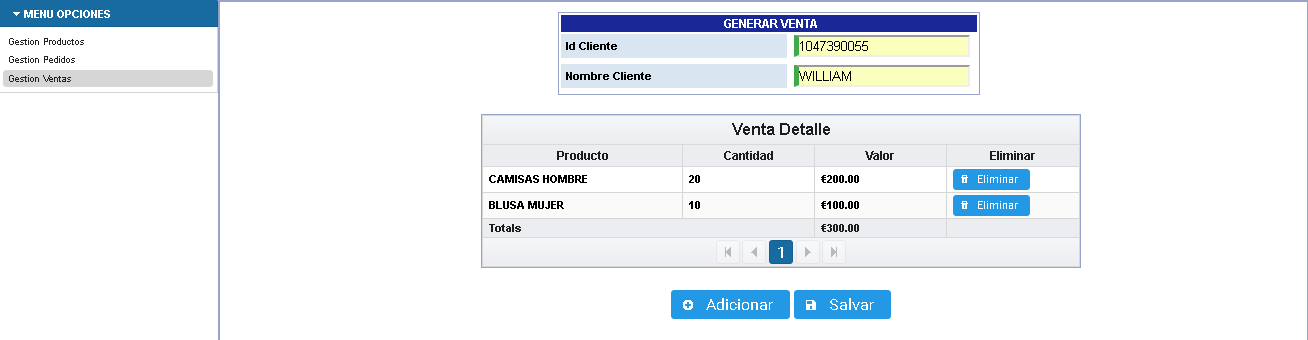
****

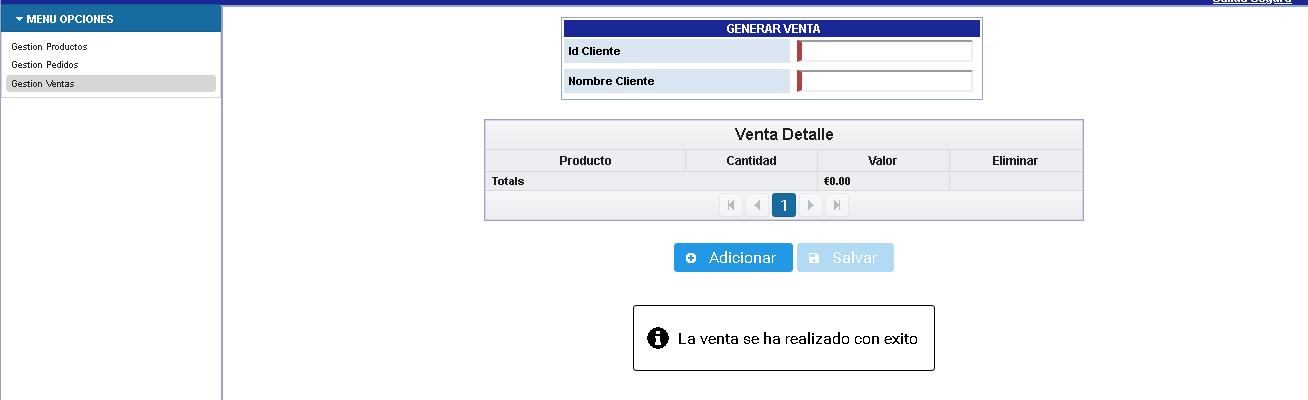
****

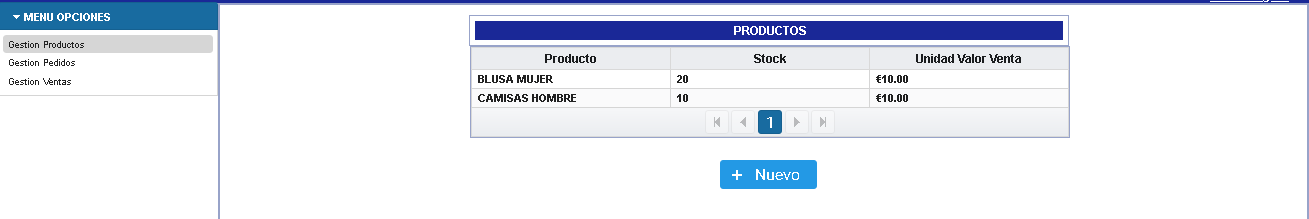
****

****

****

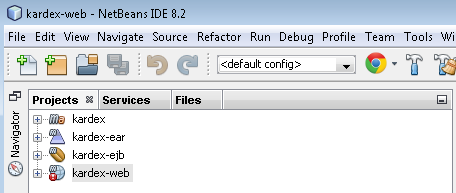
****

****

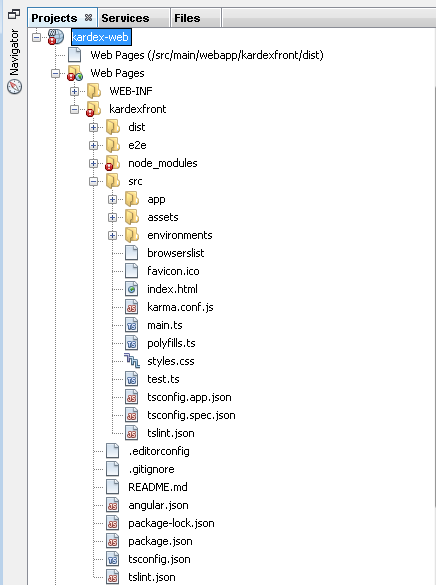
****

# arquitectura de la aplicación

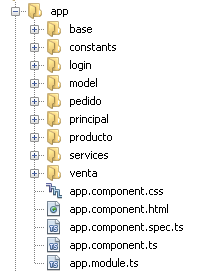
* La estructura de aplicación tiene los siguientes modulos.



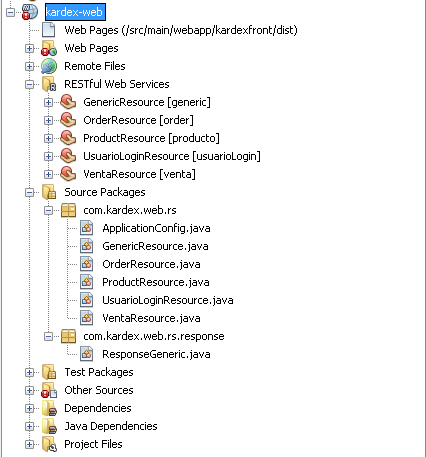
* La arquitectura está compuesta por dos componentes principales una front end con tecnologías (Angular 6, PrimeNg, HTML5) y otra back end con tecnología Java Empresarial (JEE7).
* **Módulo kardex-web:**

****

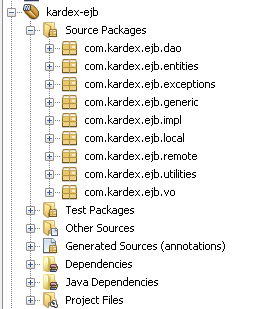
* **kardex-front**: Esta estructura se creó con la herramienta **Angular IO** que permite crear el arquetipo de angular, además facilita crear componentes, clases, servicios, contantes etc, contiene toda la lógica de presentación, compuesta de esta forma

****

* Además en este mismo contiene los servicios rest en JAX-RS servicios que son expuesto para ser consumidos desde los servicios de la capa kardexfront de presentación y estos a su vez son inyectados en los componentes.

****

* **Módulo kardex-ejb:** Encapsula toda la lógica de negocios mediante EJB Session Bean Stateless y EJB Singleton, estos mediante la inyección de dependencia son llamados desde los servicios rest.



**DAO:** Patrón que contiene los Objetos de acceso de datos a la base datos**.**

**Entities:** Contiene las entidades que relacionan las tablas mediante el objeto mapeo relacional ORM.

**Exceptions:** Contiene la excepción general entre capas.

**Generic:** Contiene un lógica abstracta genérica de persistencia con las operaciones a las base de datos.

**Impl:** Contiene la implementación de los EJB Session Bean Stateless

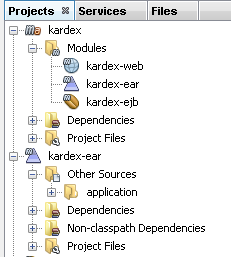
**Local:** Contiene Interfaces ejb locales.

**Remote:** Contiene Interfaces ejb remotas.

**Utilities:** Contiene clases utilitarias para validaciones y trasformación de objetos mediante el api Orika de entities a Vos y viceversa.

**Vo:** Contiene Vos Value Objetc objetos que son operados en la capa superior

* **Módulo kardex-ear:** Modulo ear de la aplicación para este caso no se utiliza.
* **Módulo kardex:** Modulo root que se utiliza para compilar y empaquetar todos el proyecto.

****