

# ÍNDICE



## Aplicación JHipster y generador DevOps

*Genera todas las pilas más modernas a partir de solo un modelo de entidad (y configuraciones predeterminadas)*

### VISIÓN GENERAL

Generar código usando JHipster

Código de generador JHipster

Descargar desde GitHub

Construir código de aplicación

Instala NodeJs

Instala Ruby y Yeoman

Instalar el gestor de paquetes Yarn utilizado por JHipster

Instalación de hilados (gen. Lockfile)

.gitignore

JHipster info

Inicio maven

Procesos de inicio de hilo.

Inicio de sesión de administrador

Métricas Admin (Monitoreo)

Registros (Logging)

Hacer un pequeño cambio

BrowserSync

Reconstruir la aplicación

Cambiar imagenes

Cambiar el nombre de la aplicación

Configuración de interfaz de usuario inmersión profunda

Configuración de administrador

Ver código fuente

JavaScript

Requiere navegador moderno

Polyfill hacia atrás

Depuración angular

CSS adaptable a Bootstrap

Cambiar CSS

Administración de usuarios admin

Configurar servidor de correo electrónico

Registrar usuario

Inicio de sesión del usuario

Back-end

Base de datos de administración

Liquibase db schema tracking

Procesos del servidor local

Entidades administrativas

Añadir Entidades

Aplicaciones personalizadas construidas sobre JHipster

JHipster libro para construir

21-points.com

Gran gran ejemplo

Fuente principal de Java

Fuente de prueba de Java

DevOps Back-end technologies

Netflix OSS

Instalación de generador local

Construir el entorno de Devbox

Instalar JHipster usando hilo

Instalar JHipster

Instalar JHipster Design Language Studio

Tirón de Maven

Eclipse

Ejecutar generador

Subgeneradores

Archivos de configuración

Examinar código de aplicación de muestra

Prueba

Dockerize

Generador para kubernetes

Ejecutar en modo de producción localmente

Modificaciones

Módulos adicionales

Aún queda por hacer

Implementación en la nube

Implementar a la nube Heroku

Construir e implementar en CloudFoundry

Medios de comunicación social




Desarrolladores de JHipster

Márketing

Videos

Más sobre DevOps

Este artículo proporciona una introducción práctica para generar una aplicación (con pruebas) utilizando las herramientas más modernas (las más recientes y productivas) para las herramientas de front-end, back-end y DevOps, ejecutándose primero localmente, luego públicamente en una nube escalable.

 <h3>Front-end</h3> <ul style="list-style-type: none"> <li>• HTML5 Boilerplate</li> <li>• Twitter Bootstrap</li> <li>• Angular / AngularJS</li> <li>• Responsive Web Design</li> <li>• Websockets</li> <li>• Angular Translate</li> </ul>	 <h3>Back-end</h3> <ul style="list-style-type: none"> <li>• Spring Boot</li> <li>• Spring Security</li> <li>• JPA, MongoDB, Cassandra</li> <li>• Elasticsearch</li> <li>• Ehcache, HazelCast</li> <li>• Metrics</li> </ul>	 <h3>Tooling</h3> <ul style="list-style-type: none"> <li>• Maven, Gradle</li> <li>• Webpack, BrowserSync</li> <li>• JUnit, Gatling, Karma, Protractor</li> <li>• Docker, Docker Compose, Cloud Foundry, Heroku, AWS, Kubernetes</li> </ul>
--	---	---

La generación de código de programación no es nueva. Ahora es más útil, completo y, por lo tanto, importante. Tradicionalmente, la elaboración de todo a mano parecía más económica. Crear un generador fue una inversión adicional que requiere un pensamiento a largo plazo y algo de gimnasia mental.

Pero ahora muchos se dan cuenta de que mantener **interfaces** entre las muchas aplicaciones y componentes de DevOps que parecen necesitar una actualización constante requiere mucho más tiempo de lo previsto. La complejidad de todo el código necesario requiere que una **comunidad** de desarrolladores trabaje en integraciones para todos los demás. Todos se benefician del trabajo de cada persona, no solo para una sola aplicación dentro de una sola compañía. Esto hace que las actualizaciones sean mucho más fáciles que hackearlas cada vez. Esto reduce la "deuda técnica" que causa retrasos y errores de fuerza.

Esta visita guiada práctica incluye:

1. Genere el código utilizando JHipster para crear un conjunto básico de ajustes de configuración basados en archivos.
2. Descarga desde GitHub las diferentes bibliotecas de módulos de nodo .
3. Ejecutar el servidor web localmente .
4. Funcionalidad de back-end de administrador (Monitoreo, Registro, etc.).
5. Haz un pequeño cambio para ver la sincronización del navegador y
6. Re-compilación de la aplicación para reconocer los cambios.
7. Cambiar imágenes

8. Cambiar el nombre de la aplicación
9. Configuración de interfaz de usuario inmersión profunda .
10. Personalizar el servidor de correo electrónico para el registro de nuevos usuarios .
11. Gestionar bases de datos y otras tecnologías de back-end .
12. Aplicaciones construidas sobre JHipster
13. Añadir entidades en aplicaciones de muestra.
14. Añadir módulos adicionales .
15. Swagger API .
16. Sube a la rama Master en GitHub / GitLab.
17. Ejecutar servidores públicamente en una nube .

## Generar código usando JHipster

<https://jhipster.tech> es la página de inicio de JHipster. Se basa en la fuente en <https://github.com/jhipster/jhipster.github.io>



Haga clic para ver una imagen más grande.

1. Instale el último Kit de desarrollo de Java (JDK).

La "J" en JHipster se refiere al lenguaje de programación Java. Así, el "java" en la cuenta de Twitter [@java\\_hipster](https://twitter.com/java_hipster) .

## Código de generador JHipster

## 2. Vaya al generador de JHipster en

<https://github.com/jhipster/generator-jhipster>

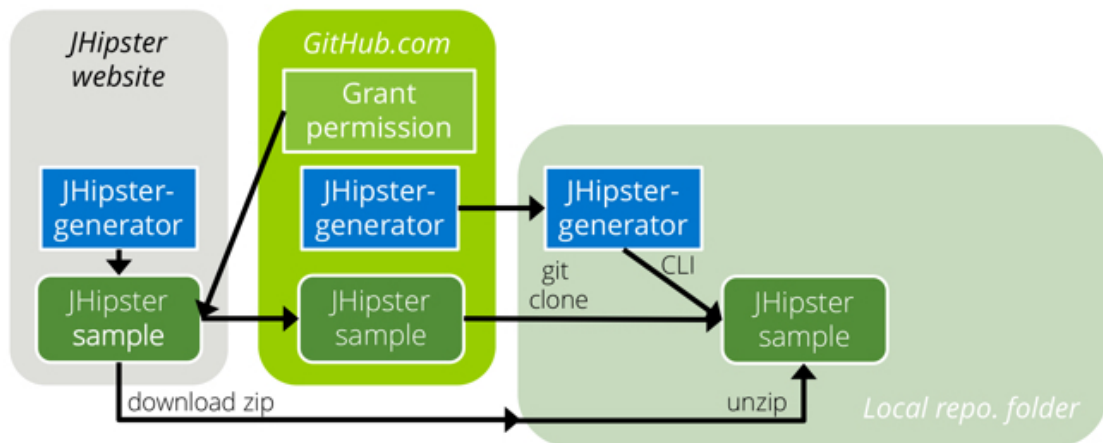
El primer compromiso con el repo fue por Julien Dubois el 21 de octubre de 2013. Trabaja en Ippon en Francia , Estados Unidos y Australia. A través de la colaboración de 19 desarrolladores y cerca de 400 colaboradores en GitHub , el generador JHipster está siendo utilizado por más de 20 compañías .

Los tatuajes en el cuello del personaje de dibujos animados de JHipster son logotipos para el back-end de Java Spring Boot y la biblioteca de Angular front-end .

<https://gitter.im/jhipster/generator-jhipster> es donde los colaboradores se reúnen.



Hay dos formas de utilizar el generador JHipster:



A) Bifurque, luego clone en su carpeta local el repositorio de una aplicación JHipster personalizada que haya generado un tercero. Instrucciones aquí .

B) Clone o descargue (luego descomprima) el repositorio del generador en su carpeta local para ejecutar su programa de línea de comandos ( `yo hipster` ). El CLI le permite especificar las opciones utilizadas durante la creación del código de la aplicación. Instrucciones aquí .

C) En 2017, se creó un sitio web en la nube de JHipster para proporcionar un formulario para seleccionar las opciones predeterminadas que utiliza el sitio web para crear el código de la aplicación. Los pasos para eso:

- 1). Instala un cliente Git (si aún no lo has hecho).
- 2). Vaya a <https://github.com/login> e inicie sesión.
- 3). Haga clic en "+" para crear un nuevo repositorio público. Lo llamo "JHipster-app1".
- 4). En un navegador web, vaya a JHipster Online en <https://start.jhipster.tech>

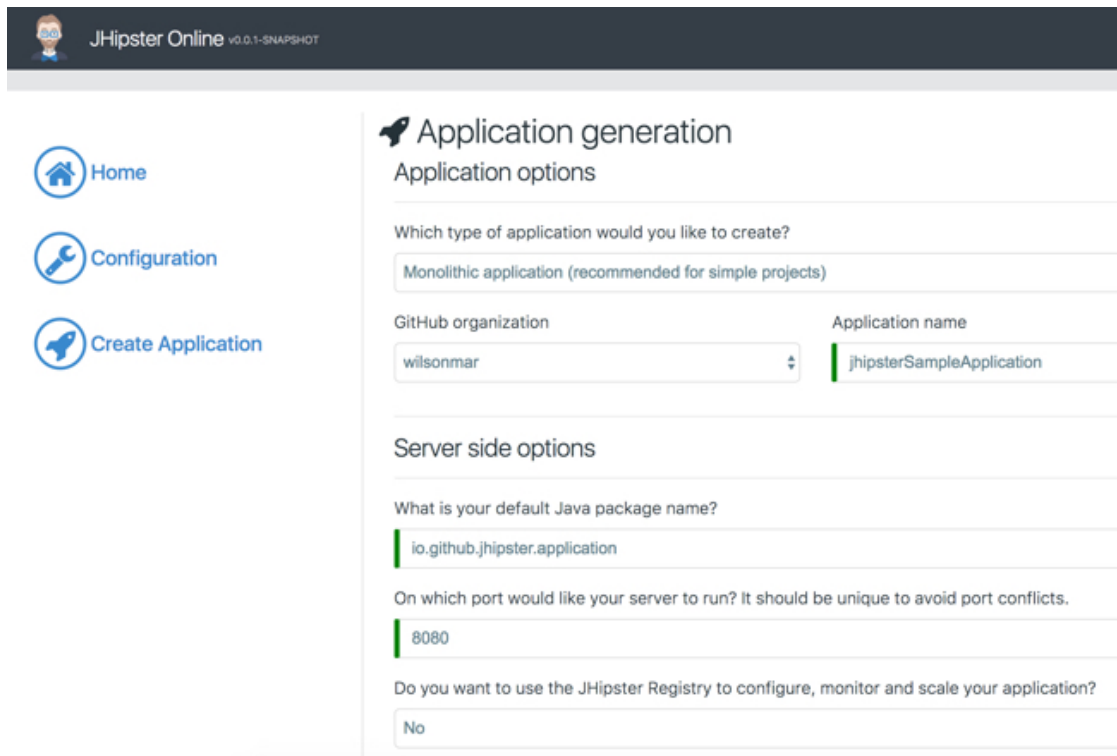
y regístrese.

5). En el correo electrónico, haga clic en el enlace para activar. Inicia sesión en JHipster Online.

6). Autorice a "JHipster by jhipster" para acceder a su cuenta de GitHub.

NOTA: revoque el acceso a la aplicación en <https://github.com/settings/applications>

3. Crear / Generar aplicación en <https://start.jhipster.tech/#/generate-application>



The screenshot shows the JHipster Online v0.0.1-SNAPSHOT interface. On the left is a sidebar with three icons: a home icon labeled 'Home', a wrench icon labeled 'Configuration', and a rocket icon labeled 'Create Application'. The main content area is titled 'Application generation' and 'Application options'. It contains several form fields: 'Which type of application would you like to create?' with a dropdown menu showing 'Monolithic application (recommended for simple projects)'; 'GitHub organization' with a dropdown menu showing 'wilsonmar'; 'Application name' with a text input field showing 'jhipsterSampleApplication'; 'Server side options' section with 'What is your default Java package name?' showing 'io.github.jhipster.application'; 'On which port would like your server to run? It should be unique to avoid port conflicts.' showing '8080'; and 'Do you want to use the JHipster Registry to configure, monitor and scale your application?' with a dropdown menu showing 'No'.

4. La opción predeterminada "jhipsterSampleApplication" no tiene que cambiar, pero tenga en cuenta que este es el nombre de su repositorio que JHipster genera. Así que no hay espacios en el nombre. Esto es también lo que muestra JHipster en cada página de UI.

5. Desplácese hacia abajo más allá de todas las selecciones predeterminadas.

- Tipo de aplicación monolítica
- puerto 8080 para el servidor
- No usar JHipster Registry para configurar, monitorear y escalar su aplicación
- Autenticación JWT (sin estado, con un token)
- MySQL para base de datos de producción
- Base de datos en memoria H2 (con persistencia basada en disco) para el trabajo de desarrollo
- Utilice el caché de segundo nivel de Hibernate, con encache (encache local, para un solo nodo)

- Maven (no Gradle) para construir el backend
- No hay soporte de internacionalización.
- Jugo, Karma, Gatling, Pepino, Transportador

NOTA: Estas selecciones se almacenan en un archivo `.yo-rc.json` , como el de la gran aplicación de ejemplo .

- Haga clic en "Generar en GitHub".
- Haga clic con el botón secundario en el enlace, como <https://github.com/wilsonmar/jhipsterSampleApplication>
- Haga clic en Cerrar.

## Descargar desde GitHub

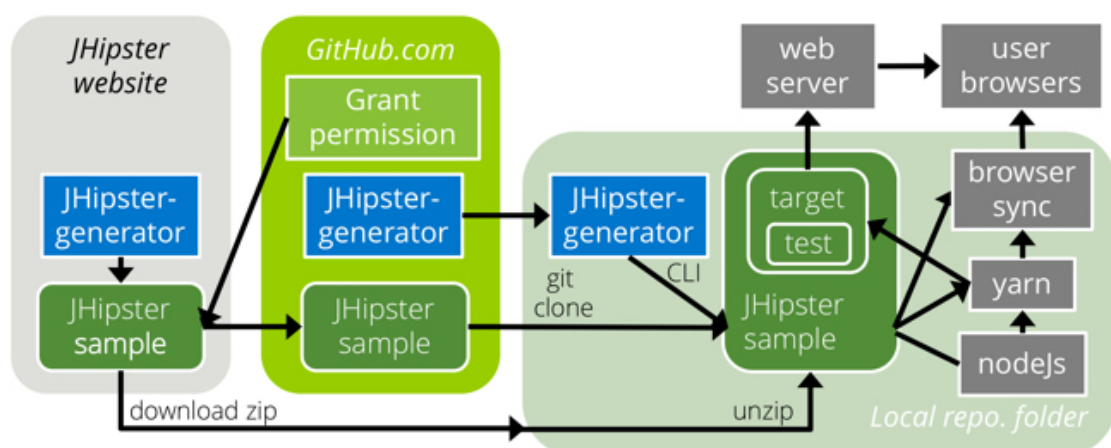
- En una Terminal, cree o navegue a una carpeta donde este comando cree una carpeta, como una carpeta que contenga el nombre de su cuenta.

```
git clone
https://github.com/wilsonmar/jhipsterSampleApplication.git
```

NOTA: Al momento de escribir, la descarga consta de 1,128,887 bytes (2.1 MB en disco) para 398 artículos.

## Construir código de aplicación

Ahora que el código personalizado de la aplicación está disponible (siguiendo los pasos anteriores) ...



## Instala NodeJs

- Utilice estas instrucciones para Windows y Mac / Linux.

## Instala Ruby y Yeoman



## BLOG

Utilice estas instrucciones para Windows:

11. Instala Ruby en Windows desde <http://rubyinstaller.org/>.
12. Instalar usando la utilidad gema de Ruby:

```
gema instalar bundler
gema instalar brújula
```

13. Instale NodeJs desde <http://nodejs.org/>. Asegúrese de marcar "agregar ruta a las variables de entorno" durante la instalación.
14. Abra el símbolo del sistema. Instalar globalmente Bower, Grunt, Yeoman (comando yo):

```
npm install -g bower grunt-cli yo
```

## Instalar el gestor de paquetes Yarn utilizado por JHipster

JHipster trata de usar el "hippest" (el último "objeto brillante"). Con esto quiero decir que permite el uso de herramientas alternativas porque integra la alternativa automáticamente cuando genera una aplicación y un código DevOps.

Para empezar, en lugar del NPM (Node Package Manager) que viene con Node, JHipster usa su hilo alternativo más rápido y "confiable", que fue lanzado en octubre de 2016 por Facebook (en colaboración con Exponent, Google y Tilde). Su <https://registry.yarnpkg.com> va a la <https://www.cloudflare.com/> CDN.

15. Aunque el hilo se puede instalar usando NPM, no se recomienda.

En Windows, [haga clic aquí para descargar el último archivo del instalador msi](#).

En MacOS o Unix, use un comando curl:

```
curl -o- -L https://yarnpkg.com/install.sh | golpetazo
```

## Instalación de hilados (gen. Lockfile)

16. Ejecute yarn para crear una carpeta **node\_modules** y **rellénela** según las especificaciones en el archivo **package.json** del proyecto :

```
instalación de hilo
```

En la respuesta están:

```
info No se ha encontrado ningún archivo de bloqueo.
...
éxito guardado archivo de bloqueo.
```

NOTA: Yarn crea automáticamente un archivo **yarn.lock** guardado con otros activos en el repositorio. El archivo garantiza que la misma versión de cada archivo y la misma estructura de archivos exacta del directorio `node_modules` sea la misma en todas las máquinas. Por el contrario, NPM

`npm-shrinkwrap.json` es generado manualmente por el `npm shrinkwrap` comando.

Luego, note que la carpeta ha crecido a 284,269,020 bytes (409.9 MB en disco) para 45,877 artículos.

## .gitignore

17. El archivo **.gitignore** enumera las carpetas que el hilo acaba de crear para que no se carguen en GitHub.

`/target/www/**` Contiene los activos utilizados por el servidor web.

## JHipster info

18. Navega a la carpeta que contiene la aplicación generada por JHipster

NOTA : recopile información sobre su instalación usando este comando:

**info jhipster**

Una respuesta de muestra (buscando archivo `.yo-rc.json`):

```
Usando la versión de JHipster instalada localmente en los node_modules del pro
Ejecutando jhipster: info
Opciones:
Bienvenido al Sub-Generador de Información JHipster

\ # \ # \ # \ # \ * \ * Versión (es) de JHipster \ * \ *
```



```
jhipster-sample-application@0.0.0 / Users / mac / gits / wilsonmar /
jhipsterSampleApplication └─ generator-jhipster@4.7.0 ##### ** Configuración
de JHipster, un archivo `yo-rc.json` generado en la carpeta raíz ** & LT; detalles> &
LT; resumen> .yo-rc.json file & LT; / summary> & LT; pre> {“generator-jhipster”:
{“applicationType”: “monolith”, “gitHubOrganization”: “wilsonmar”, “baseName
```

```

": "jhipsterSampleApplication", "packageName": "io.github.jhipster.application",
"serverPort": 8080, "serviceDiscoveryType": false, "authenticationType": "jwt",
"uaaBaseName": "../uaa", "hibernateCache": "ehcache", "clusteredHttpSession": false,
"websocket": false, "databaseType": "sql", "devDatabaseType": "h2Disk", "prodDatabaseType": "mysql",
"SearchEngine": false, "messageBroker": false, "BuildTool": "maven", "enableSocialSignIn": false,
"useSass": false, "clientPackageManager": "yarn", "testFrameworks": [], "enableTranslation": false,
"nativeLanguage": "en", "languages": ["en"], "clientFramework": "angularX", "jhipsterVersion": "4.7.0",
"packageFolder": "io/github/jhipster/application", "jwtSecretKey": "reemplazado por jhipster-info",
"jhiPrefix": "jhi"} & LT; / pre> & LT; / details> ##### ** JDL para la (s) configuración (es) de la entidad "Io / github / jhipster / application",
"jwtSecretKey": "reemplazado por jhipster-info", "jhiPrefix": "jhi"} & LT; / pre> & LT; / detalles> ##### ** JDL para la (s) configuración (es) de la entidad "Io / github / jhipster / application",
"jwtSecretKey": "reemplazado por jhipster-info", "jhiPrefix": "jhi"} & LT; / pre> & LT; / detalles> ##### ** JDL para la (s) configuración (es) de la entidad entityName.json archivos generados en el directorio .jhipster` ** & LT; detalles> & LT; resumen> Definiciones de entidades JDL & LT; / resumen> & LT; pre> & LT; / pre> & LT; / detalles> ##### ** Entorno y herramientas ** Versión Java "1.8.0_102" Java (TM) SE Runtime Environment (compilación 1.8.0_102-b14) Java HotSpot (TM) VM de servidor de 64 bits (compilación 25.102-b14, modo mixto) git versión 2.13.0 nodo: v8.3.0 npm: 5.3.0 bower: 1.7.9 trago: [06:26:48] versión CLI 3.9.1 yeoman: 2.0.0 hilado: 0.27.5 versión Docker 17.06.1-ce, compilación 874a737 docker-compose versión 1.14.0, compilación c7bdf9e Ejecución completa </pre>

```

## ## Ejecutar el servidor web localmente

Hay varios tipos de carreras:

1) Ejecuciones de prueba sin cabeza 2) Ejecuciones de prueba de IU 3) Ejecuciones de invocación del servidor web 4) Ejecuciones de prueba de rendimiento

## ### Pruebas de hilo para cobertura

1. `/target/test-results` se vuelve a crear cuando las pruebas se ejecutan utilizando el navegador "sin cabeza" PhantomJS.

### prueba de hilo

Una muestra de respuesta de la cobertura:

```

===== Resumen de la cobertura =====
Declaraciones: 65,47% (622/950)
Sucursales: 32,43% (72/222)
Funciones: 44% (99/225)
Líneas: 63.28% (536/847)

```

Hecho en 20.62s.

Una pantalla de muestra de la GUI generada es:

File	Statements	Branches	Functions	Lines
account/activate/	85.29%	29/34	100%	24/29
account/password-reset/finish/	76.74%	33/43	25%	27/37
account/password-reset/init/	96.97%	32/33	100%	27/28
account/password/	84.34%	70/83	73.08%	61/74
account/register/	93.62%	44/47	100%	39/42
account/settings/	90.91%	20/22	100%	18/20
admin/audits/	79.17%	57/72	12.5%	52/66
admin/health/	78.79%	104/132	69.81%	97/122
blocks/config/	100%	13/13	100%	10/10
shared/	100%	47/47	100%	38/38
shared/alert/	28.13%	18/64	0%	14/57
shared/auth/	46.07%	88/191	0%	73/168
shared/constants/	100%	1/1	100%	1/1
shared/login/	41.38%	36/87	0%	30/81
shared/model/	31.58%	6/19	0%	5/18
shared/user/	38.71%	24/62	0%	20/56

[Haga clic para ver una imagen más grande.](#)

TODO: Discutir la cobertura.

## 2. Ejecutar las pruebas de la aplicación:

**./mvnw prueba limpia**

La respuesta concluye con algo como esto:

Pruebas ejecutadas: 118, Fallos: 0, Errores: 0, Saltados: 0

```
[INFO]
[INFO] --- jacoco-maven-plugin: 0.7.9: informe (post-unit-test) @ jhipster-sar
[INFO] Cargando el archivo de datos de ejecución /Users/mac/gits/wilsonmar/jhi
[INFO] Paquete analizado 'Aplicación de muestra Jhipster' con 64 clases
[INFO] -----
[INFO] CONSTRUIR ÉXITO
[INFO] -----
[INFO] Tiempo total: 02:13 min
[INFO] Terminado a las: 2017-09-07T16: 41: 08-06: 00
[INFO] Memoria final: 48M / 447M
[INFO] -----
```

Una pantalla de muestra de la GUI generada es:

file:///Users/mac/gits/wilsonmar/jhipsterSampleApplication/target/test-results/coverage/jacoco/index.html

Jhipster Sample Application

### Jhipster Sample Application

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
io.github.jhipster.application.aop.logging		0%		0%	10	10	24	24	5	5	1	1
io.github.jhipster.application.config		91%		70%	23	83	29	321	12	63	2	17
io.github.jhipster.application		20%		20%	8	11	18	26	3	6	1	2
io.github.jhipster.application.web.rest.util		76%		100%	5	13	7	32	5	10	0	2
io.github.jhipster.application.web.rest.vm		65%		n/a	3	24	4	39	3	24	0	4
io.github.jhipster.application.security		79%		66%	11	26	12	54	5	17	2	5
io.github.jhipster.application.service		96%		62%	5	39	4	166	2	35	0	3
io.github.jhipster.application.service.mapper		78%		66%	4	11	9	35	2	8	0	1
io.github.jhipster.application.web.rest.errors		91%		71%	6	32	8	69	2	25	1	6
io.github.jhipster.application.domain		95%		100%	5	73	6	101	5	60	4	8
io.github.jhipster.application.web.rest		98%		97%	3	63	2	168	2	45	0	8
io.github.jhipster.application.config.audit		97%		77%	4	14	2	31	0	5	0	1
io.github.jhipster.application.service.util		75%		n/a	1	4	2	5	1	4	0	1
io.github.jhipster.application.security.jwt		100%		90%	1	15	0	73	0	10	0	3
io.github.jhipster.application.service.dto		100%		n/a	0	20	0	44	0	20	0	1
io.github.jhipster.application.repository		100%		90%	1	10	0	26	0	5	0	1
Total	632 of 5,147	87%	52 of 212	75%	90	448	127	1,214	47	342	11	64

[Haga clic para ver una imagen más grande.](#)

El informe cubre las carpetas creadas por el [complemento Jocooco Maven](#) :

```
| - clases
| - fuentes generadas
| - fuentes de prueba generadas
| - Estado de Maven
| - Informes de fuego seguro
| - clases de prueba
|- resultados de la prueba / cobertura / jacoco / index.html
```

Jocooco proporciona números de línea en los informes de cobertura y el resaltado del código fuente de los archivos de clase del objetivo de prueba compilado con información de depuración.

TODO: Marco de prueba Surefire mantenido por la fundación Apache.  
<https://maven.apache.org/components/surefire/index.html>

## Inicio maven

3. Inicie el servidor de back-end de Java:

```
./mvnw
```

Esto hace referencia a las especificaciones en el `application-dev.yml` archivo.

Alternativamente, la implementación mediante la configuración de "producción" implica la especificación de un perfil:

```
./mvnw -prod.
```

Esto hace referencia a las especificaciones en el `application-prod.yml` archivo.

Alternativamente, el comando Gradle se refiere al perfil de muestra:

```
./gradlew -prod.
```

Para generar un archivo WAR, consulte <http://www.jhipster.tech/production/>

## Procesos de inicio de hilo.

4. Abre otra terminal.
5. **PROTIP:** Abra un editor de texto para el archivo `package.json` que se ejecuta por hilo. Desplácese hasta la sección de "scripts" cerca de la parte inferior.  
`"start": "yarn run webpack:dev"`.
6. Navegue a la misma carpeta para ejecutar `webpack:dev` la interfaz del servidor:

comienzo del hilado

Una respuesta de muestra en la consola de línea de comandos:

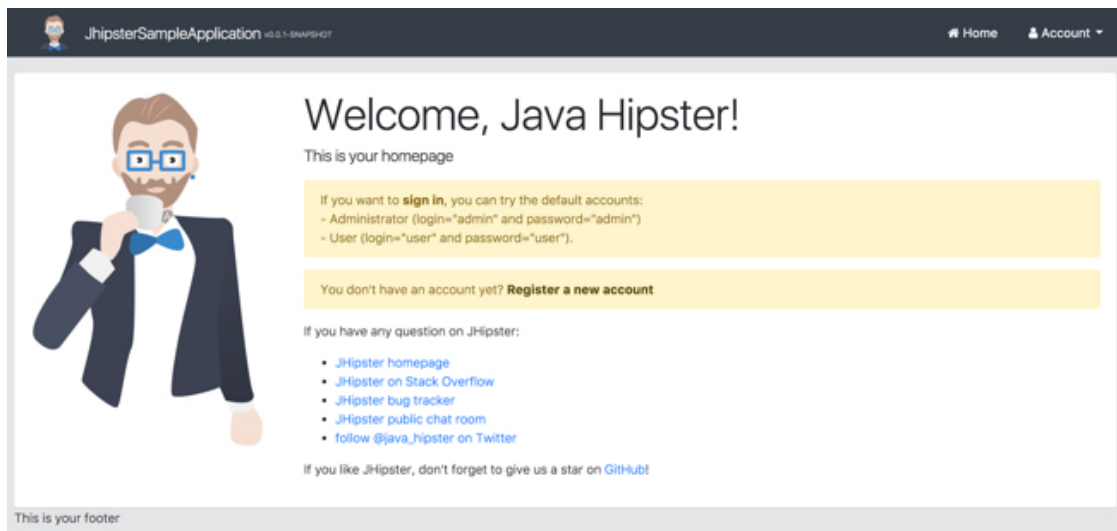
```
Fábrica: 162ms de construcción: 1051ms = 1213ms
Paquete web: Compilado con éxito.
[Browsersync] Proxying: http: // localhost: 9060
[Browsersync] URL de acceso:
-----
Local: http: // localhost: 9000
Externo: http://10.0.0.5:9000
-----
IU: http: // localhost: 3001
UI externo: http://10.0.0.5:3001
```

NOTA: la dirección IP `http://127.0.0.1` es equivalente a `http://localhost`

NOTA: Asegúrese de haber iniciado `./mvnw` si ve un error como este:

[HPM] Ocurrió un error al intentar hacer un proxy de solicitud / api / profile

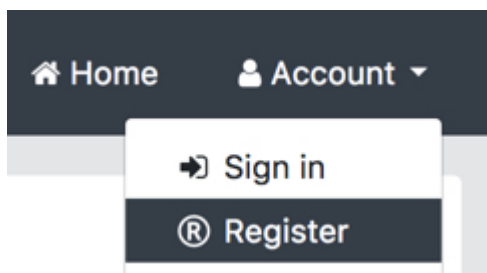
Esto hace que aparezca una nueva ventana en el navegador web predeterminado `http://localhost:9000` con esta **página de destino** :



NOTA: las aplicaciones creadas basadas en JHipster generalmente usan otro puerto, como 9010.

## Inicio de sesión de administrador

7. En la esquina superior derecha, haga clic para expandir **Cuenta** para seleccionar Iniciar sesión.



8. Inicie sesión con el `admin` usuario y la `admin` contraseña incorporados para las opciones del menú Admin:

## User settings for [admin]

First Name

Julien

Last Name

Dubois

E-mail

Your e-mail

Your e-mail is required.

Language










English

Save

([haga clic en la imagen para verla en pantalla completa](#)).

9. Marque "Recordarme" antes de hacer clic en "Iniciar sesión".

### Administration ▾

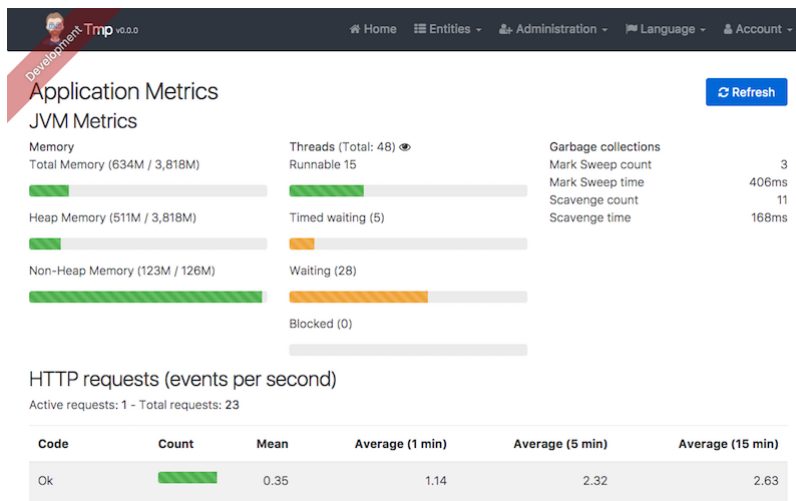
-  User management
-  User tracker
-  Metrics
-  Health
-  Configuration
-  Audits
-  Logs
-  API
-  Database

PROTIP: la maravilla del generador JHipster es que genera un conjunto bastante completo de funcionalidades de servidor "back-end".

## Métricas Admin (Monitoreo)

10. Haga clic en Administrador para seleccionar **Métricas** .





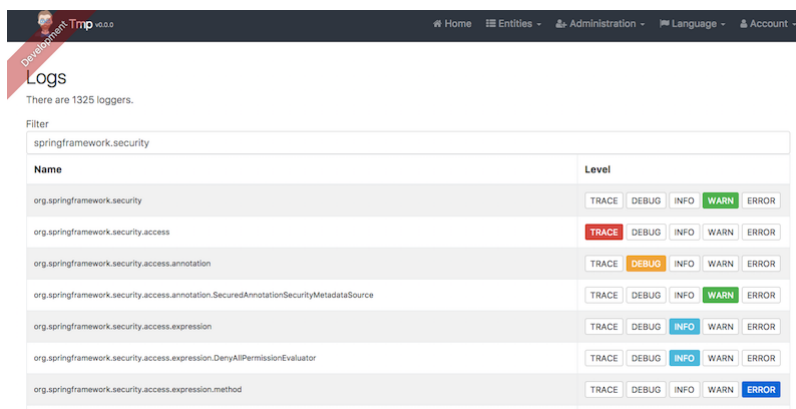
(haga clic en la imagen para verla en pantalla completa).

Esto es posible porque los agentes de supervisión de JVM se instalaron automáticamente en los servidores como parte de la generación JHipster.

Consulte <http://www.jhipster.tech/monitoring/> acerca del uso de Kibana en una pila de ELK para mostrar la consola JHipster. Presenta la visualización del rastreo distribuido de un servidor Zipkin que trabaja con ID de correlación para rastrear lapsos.

## Registros (Logging)

- Haga clic en Administrador para seleccionar **Registros**.



(haga clic en la imagen para verla en pantalla completa).

- Haga clic en otro nivel de registro, como "TRACE", que proporciona el registro más detallado.

## Hacer un pequeño cambio

PROTIP: La filosofía de desarrollo "Agile" es que se agregan pequeños cambios (incrementales) a lo largo del tiempo para exponer los problemas a lo largo de la cadena de herramientas hacia la producción.

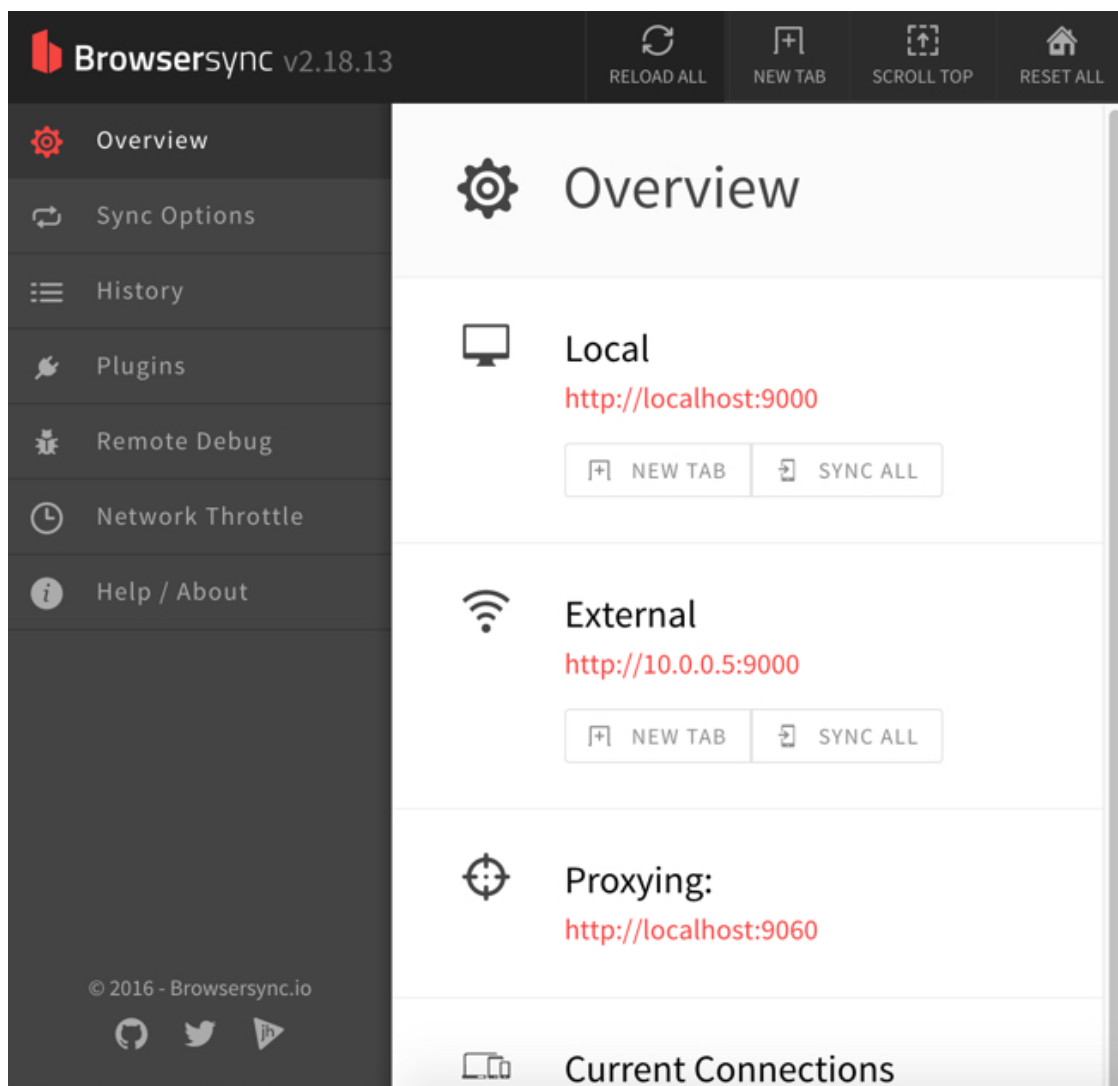
Si las opciones de componentes subyacentes no se cambian, no necesitamos ejecutar JHipster Generator nuevamente.

13. Para cambiar el texto "Este es su pie de página" en la esquina inferior izquierda de la pantalla de inicio predeterminada de JHipster, abra Atom para la carpeta superior y busque el archivo `footer.component.html` dentro de la carpeta

```
src/main/webapp/app/layouts/footer
```

## BrowserSync

Después del inicio y luego del inicio del hilo , el "puerto ui 3001" es la interfaz de usuario para el servidor BrowserSync, que realiza automáticamente una recarga de página (como presionar Ctrl + R) cuando detecta un cambio en el código. Para su información, la detección se produce al configurarse como un proxy entre el cliente y el servidor del navegador. Luego inyecta un archivo javascript en cada página para utilizar WebSockets para comunicarse entre el servidor y el cliente para ver los cambios en su código o en la acción del navegador. Ver [browsersync.io](http://browsersync.io).



## Reconstruir la aplicación

14. Dentro de la carpeta `src/main/resources` se `banner.txt` ejecuta el archivo en la Consola de línea de comandos cuando aparece la aplicación.
15. Vaya a la Terminal y si hay un proceso en ejecución, deténgalo presionando control + C en MacOS.
16. La ejecución `./mvnw` y el hilado se inician nuevamente para reconstruir y reiniciar el servidor.
17. Inicia sesión de nuevo como administrador.

## Cambiar imagenes

Estos archivos están referenciados para mostrarse en un archivo

`home.css` dentro de la carpeta

`src/main/webapp/app/home/`

18. Cree un archivo de imagen sin comprimir de **694x994** píxeles llamado `hipster2x.png` dentro de la carpeta `src/main/webapp/content/images`.
19. Cree un archivo de imagen comprimido de **347x497** píxeles nombrado `hipster.png` dentro de la carpeta `src/main/webapp/content/images`.
20. Cree un archivo de imagen comprimido de **94x119** píxeles llamado `logo-jhipster.png` dentro de la carpeta `src/main/webapp/content/images` para sustituir el logotipo que se muestra en la esquina superior derecha de cada página.

NOTA: la utilidad Webpack mueve este archivo dentro de la carpeta

`src/main/Webpack` para mostrarlo en los navegadores.

21. Cree un archivo de imagen comprimido con los mismos **94x119** píxeles nombrados `logo-jhipster.png` dentro de la carpeta `Webpack` para sustituir el logotipo que se muestra en la esquina superior derecha de cada página.

Este archivo está referenciado para mostrarse en un archivo

`navbar.css` dentro de la carpeta

`src/main/webapp/app/layouts/navbar/`

NOTA: la utilidad Webpack mueve este archivo dentro de la carpeta

`src/main/Webpack` para mostrarlo en los navegadores.

BLAH: Las imágenes y otros archivos descargados en clínicas de todo el mundo deben estar en una CDN en varias ubicaciones para que los clientes puedan recuperarlos más rápidamente.

## Cambiar el nombre de la aplicación

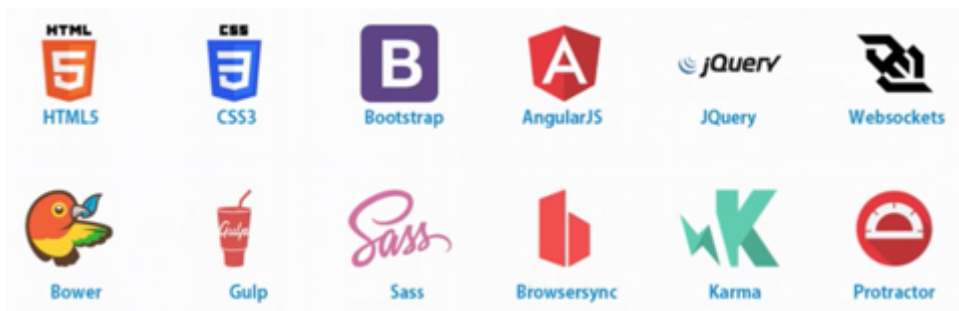
PROTIP: debido a la generación de código, html no se puede cambiar como si fuera una página estática.

22. Use un editor de texto para cambiar el nombre que se muestra en la esquina superior izquierda editando el archivo `application.yml` dentro de la carpeta `src/main/resources/config`

```
primavera:
  solicitud:
    nombre: jhipsterSampleApplication
```

## Configuración de interfaz de usuario inmersión profunda

Ahora echemos un vistazo a cómo funciona la interfaz de usuario. JHipster hace uso de varias tecnologías líderes:



## Configuración de administrador

23. Haga clic en Administración para seleccionar **Configuración**. BLAH: Resalta todo el texto en la página.

Prefix	Properties
<code>application</code>	
<code>endpoints</code>	<div>enabled <span>true</span></div> <div>sensitive <span>null</span></div>
<code>endpoints.auditevents</code>	<div>enabled <span>true</span></div>

([haga clic en la imagen para verla en pantalla completa](#)).

PREGUNTA: ¿por qué se ocultan las propiedades usando una fuente blanca?

Las propiedades bajo la `systemProperties` etiqueta, como PID, se obtienen del sistema operativo. Las propiedades bajo la `systemEnvironment` etiqueta, como la variable de entorno PATH, se obtienen del servidor. applicationConfig: [classpath: /config/application-dev.yml] applicationConfig: [classpath: /config/application.yml] actualización

## Ver código fuente

24. Haga clic con el botón derecho en una parte "blanca" de la página web y seleccione **Ver origen de página**.

En la nueva pestaña del navegador, observe que no hay mucho HTML.

PROTIP : Esto significa que el software de prueba funcional tradicional como Selenium requiere que Protractor examine HTML y CSS generados dinámicamente en el cliente por Angular.

PROTIP : Esto significa que los scripts de prueba funcional deben esperar a que se procesen las páginas. Selenium WebDriver es una herramienta / biblioteca que permite la automatización, mientras que Protractor es un Run Runner / Test Framework que define el diseño / estructura de la automatización. SerenityBDD?

25. Use un editor de texto para ver el index.html en

```
/src/main/webapp/index.html
```

PROTIP: Descomentar al trabajador de servicio para PWA.

```
<script>
  if ('serviceWorker' en el navegador) {
    navigator.serviceWorker
      .register ('./ sw.js')
      .then (function () {console.log ('Service Worker Registered')}
    )
  }
</script>
```

PROTIP: Descomente Google Analytics después de reemplazar 'UA-XXXXX-X' con su propio código de cuenta de Google.

```
<script>
  (función (b, o, i, l, e, r) {b.GoogleAnalyticsObject = l; b [l] || (b [l]
  función () {(b [l] .q = b [l] .q || []). push (argumentos)}}; b [l] .l =
  e = o.createElement (i); r = o.getElementsByTagName (i) [0];
  e.src = '// www.google-analytics.com/analytics.js';
  r.parentNode.insertBefore (e, r)} (ventana, documento, 'script', 'ga'));
  ga ('crear', 'UA-XXXXX-X'); ga ('enviar', 'vista de página');
</script>
```

# JavaScript

Los archivos JavaScript justo antes de la `</body>` etiqueta ("debajo del pliegue") son:

- `app / manifest.bundle.js` - para PWA
- `app / polyfills.bundle.js` - para compatibilidad con versiones anteriores
- `aplicación / vendor.bundle.js`
- `app / main.bundle.js`
- `app / global.bundle.js`

PROTIP: La aplicación hace uso de muchas bibliotecas de JavaScript, pero cada una de ellas no se descarga por separado porque se ejecutó **Webpack** para consolidarlas en las pocas descargas de clientes anteriores.

Estos archivos `.js` procesados por Webpack se generan a partir de archivos `.ts` , que denotan los archivos de **TypeScript** que desarrollan el código.

## Requiere navegador moderno

`[if lt IE 9]` significa que aquellos que usan Internet Explorer 8 o una versión anterior recibirán el mensaje:

Usted está utilizando un navegador **obsoleto** . Por favor, actualice su navegador para mejorar su experiencia.

Ver [browsehappy.com](https://browsehappy.com) .

## Polyfill hacia atrás

El `shim.js` hace cumplir todos los métodos estándar de ECMAScript. Los `index.js polyfills` no son métodos estándar. <https://github.com/zloirock/core-js> proporciona polyfill ES6. Ver <https://gitter.im/zloirock/core-js>

## Depuración angular

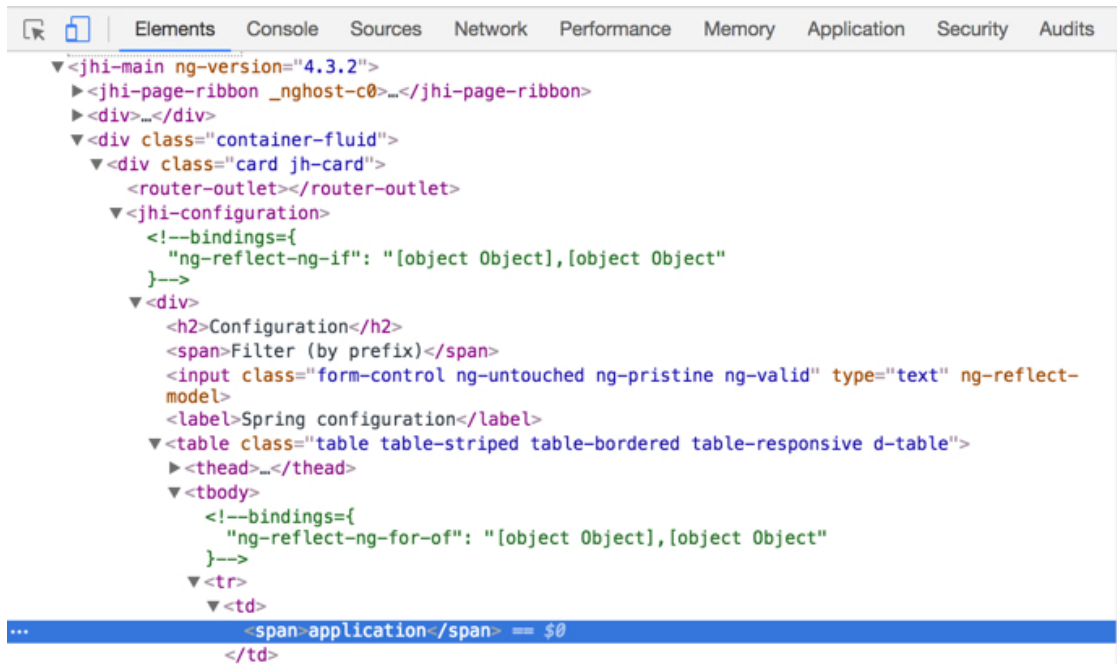
26. Haga clic con el botón derecho en una parte "blanca" de la página web y seleccione **Inspeccionar** para usar el depurador del navegador.

`<jhi-main ng-version="4.3.2">` cerca de la parte superior significa que se está utilizando **Angular 4** .

Por cierto, "AngularJs" se refiere al marco Angular1.x en [AngularJS.org](https://angularjs.org) . "Angular" se refiere a una versión actual (en [angular.org](https://angular.org)):

- Angular 2 inicialmente no era compatible con versiones anteriores hasta Typescript.
- Angular 3 se omite como un número de versión.
- Angular 4 en 2017-03 (compatible con versiones anteriores de Angular 2)
- Angular 5 en 2017-09
- Angular 6 en 2018-03
- Angular 7 en 2018-09

27. Haga clic en las flechas en la ventana Inspeccionar para ver cada nivel de HTML:



PROTIP: aquí se usan tablas tradicionales, no CSS.

## CSS adaptable a Bootstrap

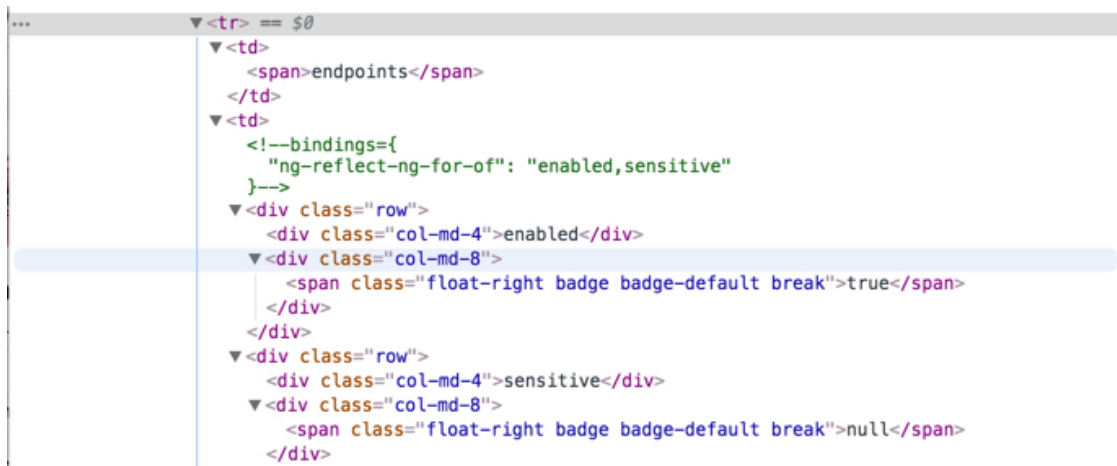
`<div class="col-md-8">` es un marcador utilizado por la biblioteca **Bootstrap** para CSS "sensible" que muestra diferentes diseños dependiendo del tamaño de la pantalla detectada en el navegador del cliente utilizado.

La biblioteca de iconos de Bootstrap y Font-awesome SVG se introduce en el cliente mediante `@import` declaraciones en

```
src/main/webapp/content/css/vendor.css
```

```
/ * después de cambiar este archivo, ejecute 'yarn run webpack: build' * /
@import '~ bootstrap / dist / css / bootstrap.min.css';
@import '~ font-awesome / css / font-awesome.css';
```

28. Haga clic en las filas para ver dónde se define "nulo":



```
<span class="float-right badge badge-default break">null</span>
```

Esto está en el archivo `configuration.component.html` dentro de la carpeta

```
src/main/webapp/app/admin/configuration/
```

Las anulaciones de JHipster de Bootstrap CSS se especifican en

```
src/main/webapp/content/css/global.css
```

Hay una regla de color de fondo para `.badge-default` en bootstrap v4.0.0-alpha.6

PREGUNTA: ¿Dónde está eso?

## Cambiar CSS

TODO: ¿El preprocesador de hojas de estilo LibSass para tu CSS?

## Administración de usuarios admin

29. Haga clic en el menú debajo de "Administrador" para seleccionar "Administración de usuarios".

Antes de que un nuevo usuario pueda registrarse por completo, se necesita cierta configuración del servidor.

PREGUNTA : La opción del menú de administración "Seguimiento de usuarios" no está en el menú.

### Configurar servidor de correo electrónico

Consulte

[https://jhipster.github.io/tips/011\\_tip\\_configuring\\_email\\_in\\_jhipster.html](https://jhipster.github.io/tips/011_tip_configuring_email_in_jhipster.html)



30. Use un editor de texto para editar el archivo `application-dev.yml` dentro de la carpeta `src/main/resources/config` para cambiar:

```
correo:
  anfitrión: localhost
  puerto: 25
  nombre de usuario:
  contraseña:
```

**PRECAUCIÓN:** este enfoque expone las credenciales al servidor de correo electrónico. **PREGUNTA:** ¿Puede usar Hashicorp Vault para administrar secretos?

31. Use un editor de texto para editar el archivo `messages.properties` dentro de la carpeta `src/main/resources/i18n/` para cambiar el texto que aparece en los correos electrónicos:

```
# Correo de activación
email.activation.title = jhipsterSampleApplication activación de cuenta
email.activation.greeting = Estimado {0}
email.activation.text1 = Se ha creado su cuenta jhipsterSampleApplication, ha
email.activation.text2 = Saludos,
email.signature = jhipsterSampleApplication Team.
```

32. Use un editor de texto para editar archivos html dentro de la carpeta `src/main/resources/emails/` para cambiar el texto que aparece en los correos electrónicos:

```
# Correo de activación
email.activation.title = jhipsterSampleApplication activación de cuenta
email.activation.greeting = Estimado {0}
email.activation.text1 = Se ha creado su cuenta jhipsterSampleApplication, ha
email.activation.text2 = Saludos,
email.signature = jhipsterSampleApplication Team.
```

33. Reconstruir la aplicación de nuevo.

### Registrar usuario

34. Si inició sesión como administrador (administrador), haga clic en el texto del menú de la cuenta para cerrar sesión.
35. Haga clic en el enlace de **Cuenta** en la esquina superior derecha para **Registrarse**.
36. Rellene el formulario de registro como con otros a través de internet:

# Registration

## Login

wilsonmar

## E-mail

W

Your e-mail is invalid.

Your e-mail is required to be at least 5 characters.

## New password

.....

Your password cannot be longer than 50 characters.

Password strength: 

## New password confirmation

Confirm the new password

REGISTER

([haga clic en la imagen para verla en pantalla completa](#)).

PROTIP: el generador JHipster crea un formulario de registro de usuario con todas las funciones, con indicador de seguridad de contraseña, restablecimiento de contraseña, etc.

PREGUNTA: ¿Cuáles son las reglas de contraseña permitidas? Helloworld2

37. Después de hacer clic en "Registrarse", no permita que el navegador guarde la contraseña.

El mensaje de respuesta:

Registro guardado! Por favor revise su correo electrónico para la confirmación

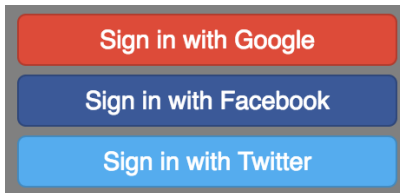


PROTIP: Defina en una hoja de cálculo varios usuarios falsos para realizar pruebas, luego haga uso de los datos en una secuencia de comandos de prueba automatizada que llena a los usuarios al completar el formulario de registro.

## Inicio de sesión del usuario

38. Haga clic en el enlace **Cuenta** en la esquina superior derecha para **Iniciar sesión** .

Alternativamente, seleccione un inicio de sesión de usuario a través de Google, Facebook, Twitter (un servicio de terceros) si esa opción fue seleccionada durante la generación :

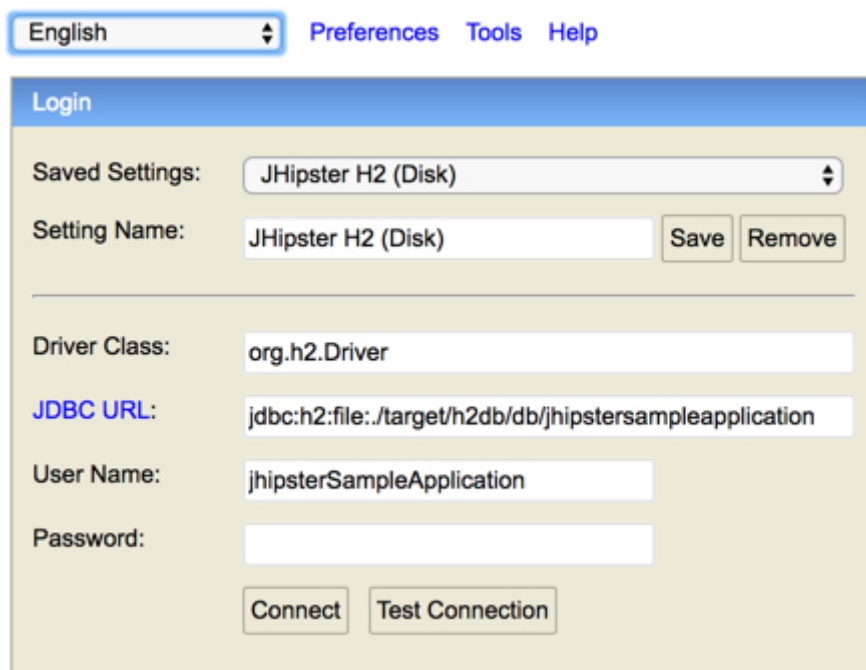


## Back-end

Acerca de las tecnologías de back-end:

## Base de datos de administración

39. En el menú superior, haga clic en "Administrador" para seleccionar "Base de datos":



NOTA: Esto es para la base de datos H2 predeterminada elegida durante las selecciones predeterminadas de JHipster Generator .

## Liquibase db schema tracking

Dentro de la carpeta se `src/main/resources/config/liquibase` encuentran los archivos de **registro de cambios** en formato XML legible pero rastreable por humanos para que los cambios en los esquemas de base de datos puedan verificarse en el control de origen, que permite la refactorización. Los otros archivos en la carpeta definen permisos.

NOTA : Liquibase (en [liquibase.org](http://liquibase.org) y con licencia [datical](http://datical.com) para las empresas) proporciona comandos como actualización, rollback, y diff (diferencia) a los cambios de esquema en las bases de datos JDBC.

## Procesos del servidor local

40. Abra una nueva ventana de Terminal para enumerar los procesos en ejecución.

En una máquina Mac / Linux, amplíe la ventana de Terminal, luego:

**PD**

Una respuesta de muestra en la consola de línea de comandos:

```
PID TTY TIEMPO CMD
37896 ttys000 0: 00.56 -bash
42984 ttys001 0: 00.44 -bash
59726 ttys001 0: 34.20 /Library/Java/JavaVirtualMachines/jdk1.8.0_102.jdk/Cont
59776 ttys001 1: 47.99 /Library/Java/JavaVirtualMachines/jdk1.8.0_102.jdk/Cont
60307 ttys002 0: 00.43 -bash
61076 ttys002 0: 00.94 nodo /usr/local/Cellar/yarn/0.27.5_1/libexec/bin/yarn.
61077 ttys002 0: 00.93 nodo /usr/local/Cellar/yarn/0.27.5_1/libexec/bin/yarn.
61078 ttys002 0: 00.94 nodo /usr/local/Cellar/yarn/0.27.5_1/libexec/bin/yarn.
61079 ttys002 0: 31.52 nodo --max_old_space_size = 4096 node_modules / webpac
61655 ttys002 0: 00.03 abierto -W http: // localhost: 9000
```

En Windows, haga clic con el botón derecho en la hora y seleccione "Procesos".

## Entidades administrativas

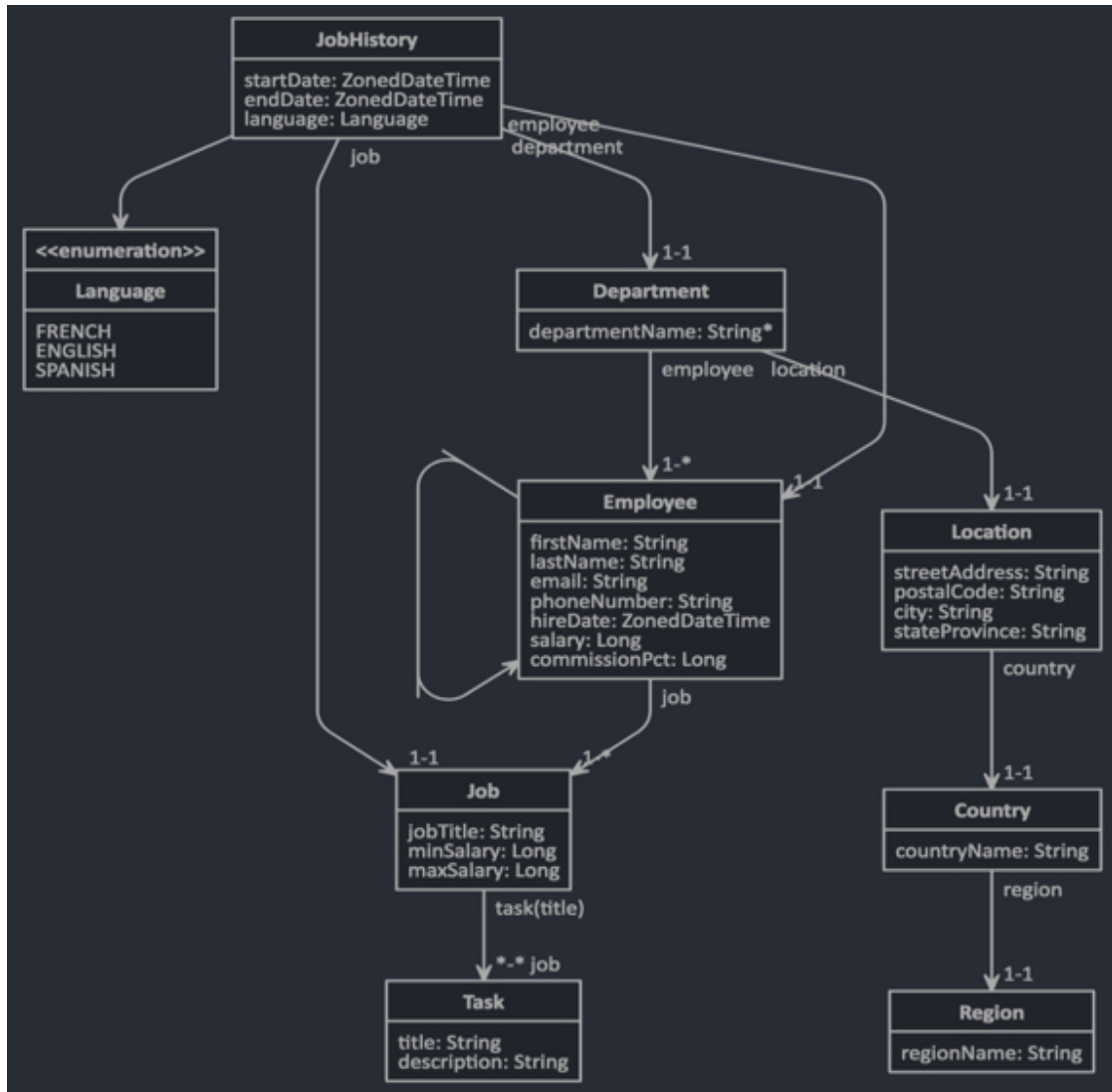
41. Seleccione Administración para seleccionar el texto del menú **Entidades**.

Tenga en cuenta que no hay entidades definidas aún en la aplicación predeterminada generada.

## Añadir Entidades

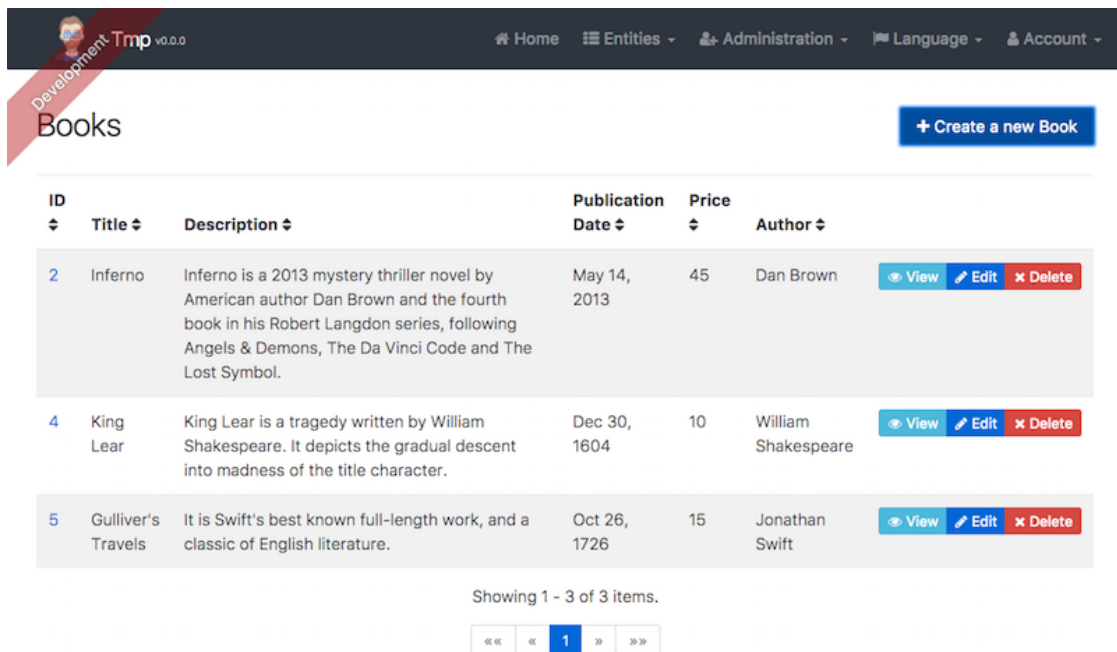
### VÍDEO

JHipster genera código a partir de un "modelo de datos de entidad" que habla de objetos de nivel empresarial, no de entidades técnicas. Un ejemplo:



Esta presentación visual es de la gente de [JDL Studio](https://www.jhipster.io/) JHipster creada para construir y administrar modelos. ( [Ver video](#) )

??? Agregue otra entidad y JHipster crea una pantalla de formulario para administrar los datos:



ID	Title	Description	Publication Date	Price	Author	
2	Inferno	Inferno is a 2013 mystery thriller novel by American author Dan Brown and the fourth book in his Robert Langdon series, following Angels & Demons, The Da Vinci Code and The Lost Symbol.	May 14, 2013	45	Dan Brown	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
4	King Lear	King Lear is a tragedy written by William Shakespeare. It depicts the gradual descent into madness of the title character.	Dec 30, 1604	10	William Shakespeare	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
5	Gulliver's Travels	It is Swift's best known full-length work, and a classic of English literature.	Oct 26, 1726	15	Jonathan Swift	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>

Showing 1 - 3 of 3 items.

« « 1 » » » »

(haga clic en la imagen para verla en pantalla completa).

El texto para las configuraciones de entidades se genera a partir del diagrama.

## Aplicaciones personalizadas construidas sobre JHipster

Las aplicaciones de muestra creadas con JHipster se mencionan en estos sitios:

<https://jhipster.github.io/showcase>

enumera las aplicaciones creadas por esta tecnología.

<https://github.com/pascalgrimaud/qualitoast>

enumera las aplicaciones creadas por esta tecnología.

<https://github.com/kbastani/spring-cloud-microservice-example> por Kenny Gastani, defensor de desarrolladores de Spring en Pivotal

<https://github.com/joshlong/microservices-dzrc>

Monitoreo por Josh Long en Spring Pivotal

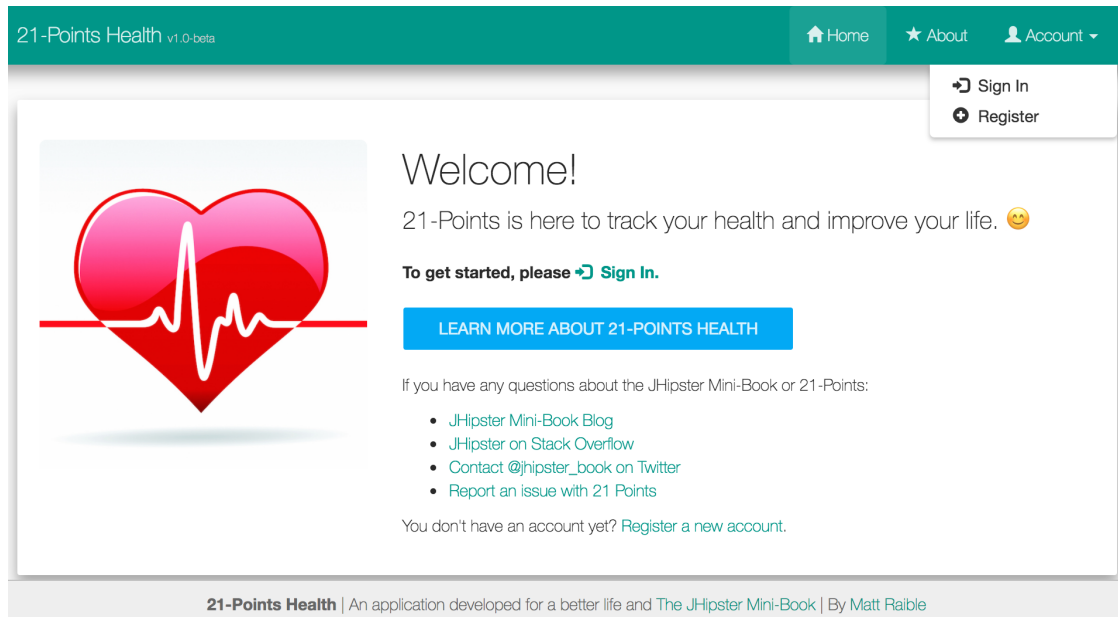
### JHipster libro para construir

- Ver el [mini-libro JHipster v2](#) de Matt Raible (@jhipster\_book) publicado por InfoQ, describe para desarrolladores web la combinación de AngularJS, Bootstrap y Spring Boot tal como lo usa JHipster. Se basa en Node.js v4.5.0 y JDK 8.

El libro pdf describe, paso a paso, las 1.157 líneas de personalización, con JHipster generando el 94.5% de todo el código.

## 21-points.com

43. <https://21-points.com> es la URL de la aplicación de muestra en línea del libro JHipster.



La velocidad de la página de Google clasifica el sitio en 88 para escritorio y 78 para móvil. El sitio fue modificado por imágenes no optimizadas, almacenamiento en caché, bloqueo de procesamiento de JavaScript, etc.

44. Si ve una alerta sobre un certificado de seguridad no válido, Agregar excepción.

El sitio está alojado físicamente en <http://health-by-points.herokuapp.com> construido como se describe en el libro de [Matt Raible](#) .

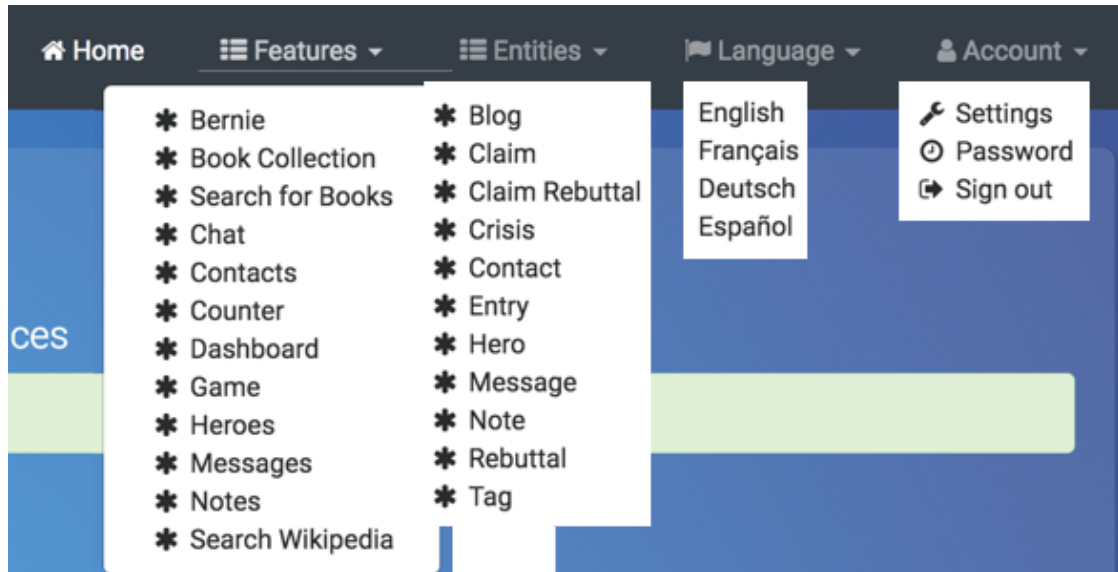
45. Despliegue Cuentas para seleccionar Registrar para un formulario con todas las funciones que presenta errores cerca de los campos e indicador de seguridad de la contraseña.
46. Confirme el correo electrónico e inicie sesión para agregar puntos, lecturas de presión arterial y peso. Establecer preferencias. Ver el calendario histórico.

### Gran gran ejemplo

47. Vea la aplicación "Great Big Example" ejecutándose todo el tiempo en la nube de Heroku:

<https://great-big-example-application.herokuapp.com/#/>

Dan Cancro creó la aplicación basada en JHipster, y luego agregó Características basadas en Entidades en la base de datos:



48. Seleccione las opciones del menú Característica para mostrar datos de bases de datos como formularios.
49. Seleccione las opciones del menú Entidades para mostrar las columnas de la base de datos como una hoja de cálculo.
50. Cambiar el idioma cambia inmediatamente los títulos a ese nuevo idioma elegido.
51. Haga clic en los terminales en ejecución y presione control + C para cerrar cada proceso.
52. Para instalar en tu propio servidor:

`git clone https://github.com/dancancro/great-big-example-application`

Al momento de escribir, esto toma 72.3 MB de espacio en disco de GitHub.

53. Cree la carpeta `node_modules` y rellénela según el archivo **package.json** (895.6 MB):

**instalación de hilo**

54. Cree la carpeta de destino (54.1 MB):

`./mvnw`



```
-----
La aplicación 'GreatBigExampleApplication' se está ejecutando! URL de acceso:
Local: http: // localhost: 8090
Externo: http://192.168.1.136:8090
Perfil (s): [swagger, dev]
-----
```

## 55. Correr:

### comienzo del hilado

Esto abre un navegador a la URL anotada en el registro:

`http://localhost:8090/#/`

## 56. Ver carpetas desde GitHub:

```
.git
.jhipster
.mvn
.vscode
docs
e2e
src
webpack
```

## 57. En una Terminal viendo la carpeta:

`tree -L 2` muestra varias bibliotecas y archivos para un sitio con todas las funciones:

```
| - clases
| | - banner.txt
| | - config
| | - i18n
| | - logback-spring.xml
| | - correos
| | - org
| `-- plantillas
| - elasticsearch
| `-- datos
| - fuentes generadas
| `-- anotaciones
| - fuentes de prueba generadas
| `-- anotaciones de prueba
| - Estado de Maven
| `-- maven-compiler-plugin
| - clases de prueba
| | - config
| | - i18n
| | - logback.xml
| | - correos
| `-- org
```

```
`- www
  | - aplicación
  | - contenido
  | - core-js-shim.min.js
  | - favicon.ico
  | - i18n
  | - index.html
  | - manifest.webapp
  | - robots.txt
  | - swagger-ui
  | - vendor.dll.js
  `- vendor.json
```

## Fuente principal de Java

58. Vamos a sumergirnos en los principales directorios de aplicaciones de Java:

```
cd src / main /
```

Esto contiene la codificación del contenedor **docker** .

La carpeta de la aplicación **web** contiene muchos de los archivos estáticos en la carpeta de destino **www**.

59. Inferior...

```
cd src / main / java / org / exampleapps / greatbig
```

`GreatBigExampleApplicationApp.java` Es el código que proporciona el punto de entrada.

`ApplicationWebXml.java` se utiliza para el registro, la carpeta de referencia `aop` (Programación Orientada a Aspectos).

`security` contiene el código jwt y social de inicio de sesión.

`config` Posee subclases de auditoría, sociales, etc.

`domain` Realiza clases correspondientes a las Entidades en la UI.

`repository` contiene clases que definen la capa persistente (base de datos).

`service` contiene el código de utilidad utilizado por los servicios.

`web` contiene el código de recursos de red web y websocket.

Estos utilizan un DTO (objeto de transferencia de datos), que reduce el número de llamadas de ida y vuelta al agregar varias llamadas en una sola llamada. Los DTO no tienen comportamiento, excepto el almacenamiento remoto y la recuperación de sus propios datos (mutadores y accesorios), por lo que no requieren pruebas.

## Fuente de prueba de Java

60. Vamos a sumergirnos en los directorios de código de prueba de Java:

```
cd src / prueba
```

Esto contiene código de emulación de cliente de rendimiento / carga de **gatling**.

61. Inferior...

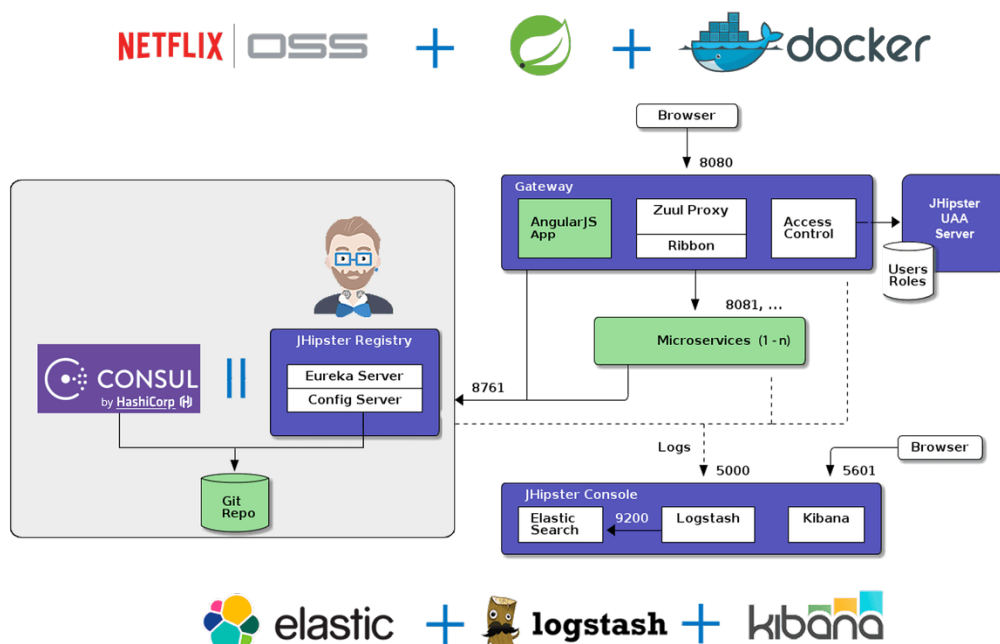
```
cd src / test / java / org / exampleapps / greatbig
```

Esto tiene las pruebas unitarias estructuradas de la misma manera que los archivos fuente principales.

Echemos un vistazo a las diversas tecnologías de DevSecOps de back-end:

## DevOps Back-end technologies

El diagrama de arquitectura: (Haga clic en el diagrama para hacerlo en una ventana de tamaño completo)\_



Spring Boot "microservice architecture" que ejecuta el OSS (software de código abierto) de Netflix con contenedores Docker.

i18n (internacionalización)

En un servidor de Gateway, el software Zuul Proxy distribuye el tráfico a los servidores de aplicaciones AngularJS.

Para aplicaciones de control de acceso, haga referencia a un servidor JHipster UAA (Autenticación y autorización de usuario) que contiene información de rol de usuario.

Las aplicaciones de referencia Ribbon maneja las llamadas de WebSocket a los servidores que manejan varios microservicios.

Cónsul de Hashicorp

JHipster Registry contiene un servidor Eureka que ... y un servidor Spring Cloud Config. Estos servidores son proporcionados por Netflix. Acceden a los repositorios de git.

La consola JHipster consiste en una base de datos de Elasticsearch alimentada por registros ingeridos por Logstash y mostrados por Kibana. Se accede a la administración de esta pila ELK mediante una instancia de navegador independiente en el puerto 5601 de forma predeterminada. Los registros gestionados hacen uso del puerto predeterminado 5000.

Más sobre microservicios:

- <https://martinfowler.com/microservices/>
- <https://martinfowler.com/articles/microservices.html>

## Netflix OSS

Netflix OSS ( [netflix.github.io](https://netflix.github.io) ) es un conjunto de marcos y bibliotecas que Netflix escribió para resolver algunos problemas interesantes de sistemas distribuidos a escala.

NOTA : Netflix OSS se puede usar sin Spring Cloud.

NOTA : Además de la nube de AWS, Netflix OSS puede ejecutarse en Docker con Kubernetes, que viene con muchas "características de sistemas distribuidos" listas para usar. Las imágenes de "Zero to Docker " en <http://search.maven.org/#search%7Cga%7C1%7Cnetflix> permiten que cualquier persona con un host Docker (en su computadora portátil, en una VM

en la nube, etc.) pueda, con un comando único, ejecute un solo nodo de cualquier tecnología NetflixOSS.

---

## Instalación de generador local

Ver <https://jhipster.github.io/installation/>

1. Después de instalar Java 8, Git, NodeJs, Yarn / MPM, Yeoman (yo)

```
npm instalar -g generador-jhipster
```

Alternativamente:

```
hilo global agregar generador-jhipster
```

## Construir el entorno de Devbox

Ver <https://github.com/jhipster/jhipster-devbox>

1. Instala Vagrant.
2. `vagrant up`

```
==> Proveedor 'virtualbox' no encontrado. Lo instalaremos automáticamente ahora.
El proceso de instalación comenzará a continuación. La interacción humana puede
requerido en algunos puntos. Si estás incómodo con automáticamente
Al instalar este proveedor, puede presionar Ctrl-C este proceso e instalarlo
manualmente
==> Descargando VirtualBox 5.0.10 ...
Puede que esta no sea la última versión de VirtualBox, pero es una versión
Eso se sabe que funciona bien. Con el tiempo, actualizaremos la versión que
esta instalado.
==> Instalando VirtualBox. Esto tomará unos pocos minutos...
Es posible que se le solicite su contraseña de administrador durante este tiempo.
Si no se siente cómodo al ingresar su contraseña aquí, instálelo
VirtualBox manual.
Contraseña:
==> ¡VirtualBox ha sido instalado exitosamente!
Se requiere un entorno Vagrant o una máquina de destino para ejecutar este
mando. Ejecute `vagrant init` para crear un nuevo entorno Vagrant. O,
obtener un ID de una máquina de destino de `vagrant global-status` para ejecutar
este comando en. Una última opción es cambiar a un directorio con un
Vagrantfile y para volver a intentarlo.
```

El uso de una imagen errante no requeriría estas acciones para configurar un cuadro de desarrollo:

- Instalar el sistema operativo Ubuntu 16.04.

- Goggle Chrome y los navegadores Firefox
- Instalar OpenJDK (Java Development Kit)
- Instalar Ubuntu Make
- Utilice Make para instalar IntelliJ IDEA o Eclipse
- Install Gulp task runner (herramienta de compilación) se ejecuta en Node / NPM.
- Instale Bower para administrar las dependencias de CSS y JavaScript utilizadas en esta aplicación.
- pgAdmin 3
- ¿Sistema de mensajería Apache Kafka? (No RabbitMQ como lo mantiene Pivotal)
- MySQL Workbench
- Cliente MongoDB
- Línea de comando de Cassandra DB (comando cqlsh)
- Línea de comando de Cloud Foundry (comando cf)
- Línea de comando de Heroku toolbelt

3. Navegue o cree una carpeta para guardar:

4. Obtener el repo:

```
git clone https://github.com/jhipster/jhipster-devbox --depth = 1
vagabundo
```

La respuesta:

```
Clonando en 'jhipster-devbox' ...
remoto: contando objetos: 17, hecho.
remoto: Compresión de objetos: 100% (15/15), hecho.
remoto: Total 17 (delta 0), reutilizado 13 (delta 0), paquete reutilizado 0
Desembalaje de objetos: 100% (17/17), hecho.
```

5. Init Vagrant de un repo:

```
vagabundo
```

La respuesta:

```
Llevando la máquina 'por defecto' al proveedor 'virtualbox' ...
==> predeterminado: no se pudo encontrar la casilla 'bento / ubuntu-16.04'. Ir
por defecto: Box Provider: virtualbox
por defecto: Versión de la caja:> = 0
==> predeterminado: Cargando metadatos para el cuadro 'bento / ubuntu-16.04'
predeterminado: URL: https://atlas.hashicorp.com/bento/ubuntu-16.04
==> predeterminado: Agregar el cuadro 'bento / ubuntu-16.04' (v2.3.7) para el
predeterminado: descarga: https://app.vagrantup.com/bento/boxes/ubuntu-16.04/
==> predeterminado: Cuadro añadido con éxito 'bento / ubuntu-16.04' (v2.3.7)
==> predeterminado: importando el cuadro base 'bento / ubuntu-16.04' ...
==> predeterminado: coincidencia de la dirección MAC para la red NAT ...
==> predeterminado: verificando si la casilla 'bento / ubuntu-16.04' está actu
```

```

==> predeterminado: configurando el nombre de la VM: jhipster-devbox_default_1
==> predeterminado: se ha corregido la colisión del puerto para 22 => 2222. Ah
==> predeterminado: borrando cualquier interfaz de red establecida previamente
==> predeterminado: preparando interfaces de red basadas en la configuración .
    por defecto: Adaptador 1: nat
==> predeterminado: Reenviando puertos ...
    predeterminado: 8080 (invitado) => 8080 (host) (adaptador 1)
    predeterminado: 9000 (invitado) => 9000 (host) (adaptador 1)
    predeterminado: 22 (invitado) => 2200 (host) (adaptador 1)
==> predeterminado: ejecutando personalizaciones de VM 'pre-arranque' ...
==> por defecto: arrancando VM ...
==> predeterminado: esperando que la máquina arranque. Esto puede tomar unos p
    predeterminado: dirección SSH: 127.0.0.1:2200
    predeterminado: nombre de usuario SSH: vagrant
    predeterminado: SSH auth método: clave privada

```



Alternativamente:

Instalar el cuadro de Vagrant creado en Windows:

```

vagrant init jhipster / jhipster-devbox
vagrant up --provider virtualbox

```

En Mac o Linux:

```

vagrant init jhipster / jhipster-devbox; vagrant up --provider virtualbox

```

La respuesta:

```

Se ha colocado un `vagrantfile` en este directorio. Ahora estas
Listo para 'vagrant up' tu primer entorno virtual! Por favor lee
Los comentarios en el Vagrantfile, así como la documentación en
`vagrantup.com` para obtener más información sobre el uso de Vagrant.
Llevando la máquina 'por defecto' al proveedor 'virtualbox' ...
==> predeterminado: no se pudo encontrar la casilla 'jhipster / jhipster-devbox'
    por defecto: Box Provider: virtualbox
    por defecto: Versión de la caja:> = 0
==> predeterminado: Cargando metadatos para el cuadro 'jhipster / jhipster-devbox'
    predeterminado: URL: https://atlas.hashicorp.com/jhipster/jhipster-devbox
==> predeterminado: Agregar la casilla 'jhipster / jhipster-devbox' (v4.6.1) p
    predeterminado: Descarga: https://app.vagrantup.com/jhipster/boxes/jhipster-c

==> predeterminado: Cuadro añadido con éxito 'jhipster / jhipster-devbox' (v4.
==> predeterminado: importando el cuadro base 'jhipster / jhipster-devbox' ...
==> predeterminado: coincidencia de la dirección MAC para la red NAT ...
==> predeterminado: verificando si la casilla 'jhipster / jhipster-devbox' est
==> predeterminado: configuración del nombre de la máquina virtual: jhipster-s
==> predeterminado: borrando cualquier interfaz de red establecida previamente
==> predeterminado: preparando interfaces de red basadas en la configuración .
    por defecto: Adaptador 1: nat
==> predeterminado: Reenviando puertos ...
    predeterminado: 22 (invitado) => 2222 (host) (adaptador 1)
==> por defecto: arrancando VM ...
==> predeterminado: esperando que la máquina arranque. Esto puede tomar unos p

```

```
predeterminado: dirección SSH: 127.0.0.1:2222
predeterminado: nombre de usuario SSH: vagrant
predeterminado: SSH auth método: clave privada
```

## 6. Abra un navegador (Chrome) y vaya al servidor local:

127.0.0.1:2222

### Haz esto rápido o verás:

Tiempo de espera mientras se espera que la máquina arranque. Esto significa que Vagrant no pudo comunicarse con la máquina invitada dentro de el período de tiempo configurado (valor "config.vm.boot\_timeout").

Si mira arriba, debería poder ver los errores que Vagrant tenía al intentar conectarse a la máquina. Estos errores Por lo general, son buenos consejos sobre lo que puede estar mal.

Si está utilizando un cuadro personalizado, asegúrese de que las redes estén trabajando y puedes conectarte a la máquina. Es un comun Problema de que la red no está configurada correctamente en estos cuadros. Verifique que las configuraciones de autenticación también estén configuradas también.

Si el cuadro parece estar arrancando correctamente, es posible que desee aumer el valor del tiempo de espera ("config.vm.boot\_timeout").

## 7. Instale Node Node.js para ejecutar un servidor web de desarrollo y compilar el proyecto.

## 8. Instale Yarn (en lugar de NPM) para administrar las dependencias de los nodos, de modo que la recarga en vivo del código del lado del cliente (HTML, CSS, TypeScript) funcione (gracias a Webpack y Browsersync).

```
hilo global add yo
```

### La respuesta que obtuve:

```
hilo global v0.27.5
[1/4] Resolviendo paquetes ...
[2/4] Recuperando paquetes ...
[3/4] Vinculando dependencias ...
[4/4] Construyendo paquetes frescos ...
éxito instalado "yo@2.0.0" con binarios:
  - yo
  - yo-completo
Hecho en 7.63s.
```

## 9. Para usar las herramientas globalmente, asegúrese de tener en su camino:



```
$ HOME / .config / yarn / global / node_modules / .bin
```

10. En Mac o Linux: `export PATH = "$ PATH yarn global bin :: $ HOME / .config / yarn / global / node_modules / .bin"`

## Instalar JHipster usando hilo

11. Instalar JHipster

```
hilo global agregar generador-jhipster
```

La respuesta:

```
hilo global v0.27.5
warning package.json: Sin campo de licencia
advertencia No hay campo de licencia
[1/4] Resolviendo paquetes ...
[2/4] Recuperando paquetes ...
error jhipster-core@1.3.4: el "nodo" del motor es incompatible con este módulo
error encontrado módulo incompatible
info Visite https://yarnpkg.com/en/docs/cli/global para obtener documentación
```

12. Edita cualquier archivo y el navegador se actualiza automáticamente.

```
yo
```

La primera vez que aparece esto:

```
? =====
;Estamos constantemente buscando formas de mejorarte!
;Podemos informar de manera anónima las estadísticas de uso para mejorar la he
Más información: https://github.com/yeoman/insight & http://yeoman.io
===== (S
```

13. Escriba "y" para esta respuesta:

```
? 'Allo mac! Que te gustaría hacer?
Instalar un generador
? Búsqueda de npm para generadores:

_  _
| | | ¡Adiós de nosotros! |
| - (o) - | | Charla pronto. |
`-----' team Equipo Yeoman |
(_'U`_) | http://yeoman.io
/ ____ A ____ \ / _____
| ~ |
_.'_.'_
/ \ | ° / Y`
```

14. Presiona control + C para salir.

PROTIP: El "flujo de trabajo de Yeoman" se compone de tres tipos de herramientas para mejorar su productividad y satisfacción al crear una aplicación web:

- herramienta de andamios (yo),
- herramienta de compilación (Grunt, Gulp, etc.), y
- gestor de paquetes (Bower, npm, etc.)

## Instalar JHipster

### 15. Instalar JHipster

```
hilo global añadir generar-jhipster
```

La respuesta cuando lo hice:

```
hilo global v0.27.5
[1/4] Resolviendo paquetes ...
[2/4] Recuperando paquetes ...
error jhipster-core@1.3.4: el "nodo" del motor es incompatible con este módulo
error encontrado módulo incompatible
info Visite https://yarnpkg.com/en/docs/cli/global para obtener documentación
```

## Instalar JHipster Design Language Studio

### 16. Instale JHipster Studio dentro de la carpeta generators-jhipster / generators:

```
yo jhipster: import-jdl jhipster-jdl.jh
```

En futuras versiones use sub-generador Yeoman:

```
jhipster import-jdl
```

La respuesta cuando lo hice:

Parece que no tiene un generador con el nombre "jhipster: import-jdl" instalado. Pero la ayuda esta en camino:

Puede ver los generadores disponibles en npm search yeoman-generator o en http://yeoman.io/generators/. Instálalas con npm install generator-jhipster: import-jdl.

Para ver todos los generadores instalados, ejecute yo sin ningún argumento. Así:

Si no puede encontrar el generador, ejecute yo doctor para solucionar los problemas.

### 17. Iniciar el ???

```
comienzo del hilado
```

```

inicio de hilo v0.27.5
error No se pudo encontrar un archivo package.json en "/ Users / mac / gits /

```

## 18. Generador de aplicaciones front-end Yeoman:

### Tirón de Maven

Utilice el envoltorio Maven incrustado para ejecutar:

```

Jhipster
./mvnw

```

Las instalaciones anteriores se basan en el **pom.xml** :

19. Sass = "hojas de estilo sintácticamente impresionantes"
20. Websockets
21. Browsersync
22. Spring Data JPA genera repositorios JPA automáticamente
23. Liquibase de <http://www.datical.com/>: el ORM (Object Relation Model) que funciona con Spring y JPA (Java Persistence) para manejar la actualización de esquemas de base de datos
24. Elasticsearch para informes JMX o Graphite
25. <https://webpack.github.io/> para comprimir solo las funciones de JavaScript que solo se usan. Compila, minimiza, optimiza el front-end para recargar en caliente.
26. Leaflet.js
27. Motor de plantillas Thymeleaf para generar páginas web del lado del servidor. (reemplaza JSPs y JSTLs, que están desactualizados y en desuso). Ejemplo:

```
<h1 th: text = "# {title}"> Hola, mundo </h1>
```

28. <http://www.dropwizard.io/1.1.4/docs/> reúne bibliotecas estables y maduras del ecosistema de Java en un paquete simple y liviano para crear servicios web RESTful listos para producción.
29. Netflix OSS proporciona componentes de registro JHipster:
  - Eureka - Registro de servicio para balanceo de carga y failover.
  - Zuul - Proxy para enrutamiento dinámico, monitoreo y seguridad
  - Ribbon - software de balanceo de carga para servicios
30. Almacenamiento en caché

- Ehcache es la solución más utilizada.
- HazelCast tiene soporte de agrupamiento
- Hibernate caché de segundo nivel o abstracción Spring Caching

## Eclipse

## VÍDEO

31. Edite el `.editorconfig` archivo para especificar `indent_size`, etc.

## Ejecutar generador

PROTIP: Una de las grandes ventajas de usar un generador de código es tener diferentes opciones para hacer algo similar.

32. Crea o navega a la carpeta de tu aplicación.

A diferencia de otros generadores de aplicaciones Java, Yeoman espera estar en el directorio donde crea su proyecto en lugar de hacerlo por usted.

33. Generación iniciada

```
yo jhipster
```

Por cierto, este mismo comando se utiliza para actualizar también.

El generador hace una serie de preguntas (más o menos 15).

PREGUNTA: ¿Hay una muestra de automatización para responder preguntas?

? (1/16) ¿Qué *tipo* de aplicación te gustaría crear?

- Monolítico
- Servicios ???

## Subgeneradores

El sub-generador "import-jdl" genera un modelo de entidad complejo utilizando JDL Studio

El subgenerador "entidad" genera una entidad JPA completa, desde la base de datos hasta Angular

El subgenerador "servicio" genera un servicio comercial de Spring.

El subgenerador de "idiomas" agrega idiomas para i18n (español)

## Archivos de configuración

34. Ejecute el subgenerador ci-cd para generar archivos de configuración para varios sistemas de integración continua.

```
jhipster ci-cd
```

Consulte la página Configuración de integración continua para obtener más información.

Consulte <https://www.npmjs.com/package/generator-jhipster>

```
npm generator-jhipster
```

35. Zipkin para rastrear solicitudes HTTP a través de sus servicios
36. Elastalert para recibir una alerta cuando se produce un problema

## Examinar código de aplicación de muestra

<https://github.com/jhipster/generator-jhipster>

1. La aplicación en línea se construye desde un repositorio en GitHub:

<https://github.com/jhipster/jhipster-sample-app-ng2>

<https://github.com/jhipster/jhipster-sample-app>

## Prueba

2. Instalar karma

<http://karma-runner.github.io/>

Escriba pruebas unitarias para controladores y directivas angulares usando

<http://jasmine.github.io/2.5/introduction.html>

escriba pruebas de integración con el transportador:

<http://www.protractortest.org/#/>

3. Para iniciar las pruebas de su aplicación, ejecute:

```
./mvnw prueba limpia
```

4. Instala Scala para ejecutar Gatling.
5. Instale Gatling para ejecutar pruebas de rendimiento.

## Dockerize

```
DESDE pbesson / jhipster: onbuild
DESDE pbesson / jhipster: onrun
```

## Generador para kubernetes

El generador de Kubernetes genera descriptores de implementación para implementar en Kubernetes con el fin de escalar horizontalmente con facilidad y adaptarse a los escenarios de falla.

## Ejecutar en modo de producción localmente

```
./mvnw -prodel paquete
docker-compose -f src / main / docker / mysql.yml up -d
java -jar target / *. war
```

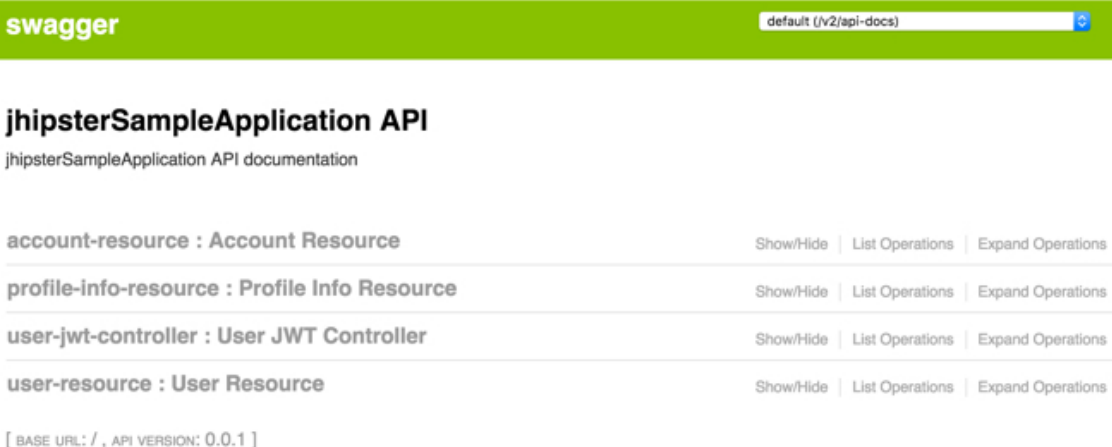
---

## disponibilidad de API

??? Microservicios

### Admin Swagger

1. En el menú superior, haga clic en "Administrador" para seleccionar "Swagger":



account-resource : Account Resource	Show/Hide	List Operations	Expand Operations
profile-info-resource : Profile Info Resource	Show/Hide	List Operations	Expand Operations
user-jwt-controller : User JWT Controller	Show/Hide	List Operations	Expand Operations
user-resource : User Resource	Show/Hide	List Operations	Expand Operations

[ BASE URL: / , API VERSION: 0.0.1 ]

El index.html para esta página está en la carpeta.

```
/src/main/webapp/swagger-ui/
```

---

## Modificaciones

### ### PWA

PWA (Progressive Web Apps) para que el contenido se cargue para el uso fuera de línea en Mobile + desktop

Hecho por líneas sin comentarios en webpack.common.js y en index.html.

### ### Diseño de materiales

```
npm install -g generator-jhipster-bootstrap-material-design;
yo jhipster-bootstrap-material-design

? ¿Quieres instalar el diseño del material de Bootstrap? Sí
Componiendo la configuración de JHipster con el módulo bootstrap-material-design
Leyendo la configuración del proyecto JHipster para tu módulo.
conflicto bower.json
? Sobrescribir bower.json? exagerar
fuerza bower.json
crear src / main / webapp / app / blocks / config / bootstrap-material.config.js
conflicto src / main / webapp / scss / main.scss
? ¿Sobrescribir src / main / webapp / scss / main.scss? exagerar
fuerza src / main / webapp / scss / main.scss
conflicto src / main / webapp / content / css / main.css
? ¿Sobrescribir src / main / webapp / content / css / main.css? exagerar
force src / main / webapp / content / css / main.css
```

**PROTIP:** vendor.css está antes de todas las hojas de estilo de diseño de materiales en su index.html

### ### Añadir entidad

**entidad jhipster Foo**

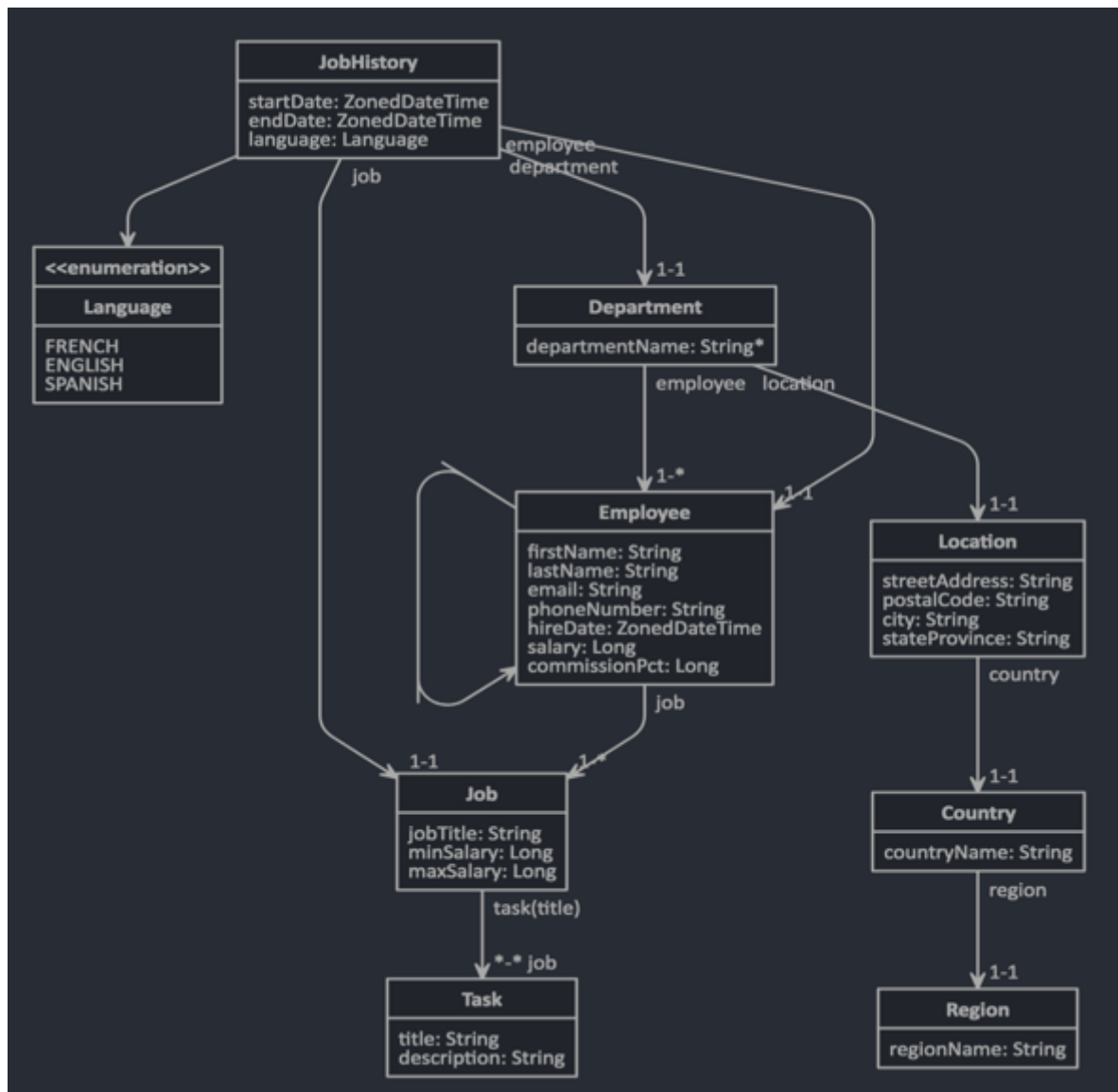
La codificación para manejar las entidades se genera a partir del código JDL.

JDL es el lenguaje de dominio JHipster procesado dentro de JDL Studio.

<https://github.com/jhipster/jdl-samples>

JHipster generará una relación de "uno a muchos" como una pantalla de lista, que crea todas las entidades JPA, DTO, controladores de descanso de primavera hasta el lado del cliente con Angular.

Entre los vídeos de YouTube de Matt está presente sobre [github.com/mraible/jhipster4-demo](https://github.com/mraible/jhipster4-demo) Los modelos de diagramas entidad muestra una base de datos que contiene el historial de tareas con referencia a la Lengua y Departamentos con una ubicación que consiste en País, Región.



### Infinite scroll módulo de terceros

[https://jhipster.github.io/tips/019\\_tip\\_infinite\\_scroll\\_with\\_slice.html](https://jhipster.github.io/tips/019_tip_infinite_scroll_with_slice.html)

### Sprint conectores sociales

[https://jhipster.github.io/tips/012\\_tip\\_add\\_new\\_spring\\_social\\_connector.html](https://jhipster.github.io/tips/012_tip_add_new_spring_social_connector.html)

### Gráficos

<https://www.primefaces.org/primeng/#/> PrimeNG UI library for Angular

npm instalar primeng --save

### WebSockets

<http://rick-hightower.blogspot.com/2015/12/websocket-microservice-vs-rest.html>

## Módulos adicionales



Un mercado de módulos de terceros (que actúan como subgeneradores JHipster estándar) puede acceder a las variables y funciones de JHipster a través de mecanismos de enganche antes y después de la generación de aplicaciones Yeoman :

- Usa VueJS
- Google analitico
- mapas de Google

## Aún queda por hacer

Aquí hay algunas capacidades para la lista de tareas pendientes.

1. 2FA (autenticación de dos factores) o, mejor aún, el uso de la aplicación móvil Authy en teléfonos inteligentes.
2. Páginas estáticas para carga rápida, como con Jekyll?
3. Google Maps de datos de geolocalización.
4. Carro de compras con Hazelcart + MySQL.
5. Visualizaciones de datos de aplicaciones (usando D3.js o angular-nvd3 ).
6. Ajuste automático de los parámetros de configuración.
7. Comparación de “costo total por transacción de aplicación” que varía los diversos componentes.
8. Integración de Twilio e IFTTT para enviar SMS, hacer llamadas, etc.
9. Motor de reglas de negocios
10. Ideas de otros

## Implementación en la nube

JHipster se envía con soporte para el despliegue en varias nubes.

Las invocaciones en la nube normalmente requieren la especificación de una región geográfica.

NOTA: Amazon EC2 requiere que las instancias estén en la región de EE. UU. Este para ser empaquetadas en imágenes.

## Implementar a la nube Heroku

1. A la nube de Heroku:

```
yo jhipster: heroku
```

Alternativamente: Otros cloudfoundry, AWS, y Boxfuse.

gobernadores

2. Ver la aplicación en Heroku:

<https://jhipster-4-demo.herokuapp.com>

BLAH: Esto regresó con "Error de aplicación".

TODO: Construir e implementar en AWS, Azure, Google clouds.

## Construir e implementar en CloudFoundry

Ver <http://www.jhipster.tech/cloudfoundry/>

```
jhipster cloudfoundry
```

Esto utiliza [Spring Cloud Connectors](#).

Ver <http://www.jhipster.tech/production/>

---

## Medios de comunicación social

### Desarrolladores de JHipster

Los desarrolladores hablan entre ellos [en el foro de Google](#) .

Julien Dubois (@juliendubois)

- <https://www.youtube.com/watch?v=d1MEM8PdAzQ> tutorial de 20 minutos.
- <https://www.youtube.com/watch?v=dzdjP3CPOCs> Sea productivo con JHipster por Julien Dubois y Deepu K Sasidharan 8 de noviembre de 2016 en Devxxx

EE. UU.

- <https://www.slideshare.net/julien.dubois/devoxx-being-productive-with-jhipster>
- <https://www.youtube.com/watch?v=dOO-8cAKDu8> Novedades de JHipster en 2016 por Julien Dubois @ Spring I / O 2016

Deepu K Sasidharan ( [@deepu105](#) ) Defensor de desarrolladores en XebiaLabs

- <https://github.com/deepu105>
- <https://deepu.js.org/jh-slides-v3/#/intro>
- <https://www.youtube.com/watch?v=YmEKCITBkUI> Creación de aplicaciones de arranque + Angular4 de Spring en minutos
- <https://www.youtube.com/watch?v=nKRpVkba4Ck> 2016-Mar en Tata
- <https://www.youtube.com/watch?v=YmEKCITBkUI> Creación de aplicaciones de arranque + Angular4 de Spring en minutos con JHipster @ Spring I / O 2017

Matt Raible Matt Raible - ( [@mraible](#) ) <https://raibledesigns.com/>  
<https://www.linkedin.com/in/mraible>

- Su blog en junio de 2017 muestra cómo construir una arquitectura de microservicios con JHipster 4.5.4. Genere una puerta de enlace (con tecnología de Netflix Zuul y JHipster Gateway), un microservicio (que habla con MongoDB), y use Docker Compose para asegurarse de que todo se ejecute localmente. Luego desplázalo a Minikube y Google Cloud usando Kubernetes.
- <http://sudheerjonna.com/my-books/> PrimeFaces temas y PrimeNG.
- [https://www.youtube.com/watch?v=kkHN2G\\_nXV0](https://www.youtube.com/watch?v=kkHN2G_nXV0) Obtén Hip con JHipster: Spring Boot + AngularJS + Bootstrap (Matt Raible) Devoxx FR
- <https://www.youtube.com/watch?v=ZGF4gEM4FuA> JHipster 2.7 Blog Demo [20:30]

Ray Tsang - ( [@saturnism](#) ) Arquitecto / Defensor de la tecnología en Google

Yakov Fain - ( [@yfain](#) ) [yakovfain.com](http://yakovfain.com)

Alfredo Rueda [@aruedaunsain](#)

David L. Whitehurst

Cómo cifrar valores sensibles

## Márketing

- [] Pegatinas para cubiertas de portátiles - [] Camisetas, sudaderas con capucha - [] Tazas de café

- [] Clases prácticas, boot-camp: freecodecamp.com? - [] Charlas de conferencia - [] Webinars para gerentes de desarrollo

- [] Mercado de temas - [] Ideas específicas para contribución - [] Trabajos para desarrolladores

<https://stackoverflow.com/questions/tagged/jhipster>

<http://scholarcoder.com/1072/making-full-stack-java-development-easier-jhipster>

## Videos

[Sé productivo con JHipster por Julien Dubois y Deepu K Sasidharan 8 de noviembre de 2016 Devovx 24K vistas](#)

[Desarrolle aplicaciones Spring Boot + Angular4 en minutos con JHipster Spring I / O 2017 - 18 -19 de mayo, Barcelona por Deepu K Sasidharan \(@ deepu105, deepu.js.org\) trabajando en Xebia Labs.](#)

[https://www.youtube.com/watch?v=DcWx\\_BOugIM](https://www.youtube.com/watch?v=DcWx_BOugIM) Jhipster 4 + Angular 2 | Aplicación web Java. MySQL. por la venganza de codificación

<https://www.youtube.com/watch?v=XRREt1KB4Y8&t=162s> Comenzar con JHipster 4 Matt Raible 66K visitas

<https://www.youtube.com/watch?v=ZGF4gEM4FuA&t=259s> Demostración del blog de JHipster 2.x Matt Raible

<https://www.youtube.com/watch?v=5CWu4NoMsSk> JHipster, la mejor manera de crear una nueva aplicación web de Jfokus

<https://www.youtube.com/watch?v=R3jm2qmqctI> Seminario web: JHipster para Spring Boot SpringDeveloper

<https://www.youtube.com/watch?v=dOO-8cAKDu8&t=601s> Novedades de JHipster en 2016 por Julien Dubois @ Spring I / O 2016 Spring I / O

<https://www.youtube.com/watch?v=NdcnpD-GpZ4> Aplicación de pila completa desde cero - JHipster - 03 vistas de la armería electrónica 11K

<https://www.youtube.com/watch?v=nKRpVgba4Ck> JHipster 3.0 night! - Singapore Spring User Group Engineers.SG 6.1K vistas

[https://www.youtube.com/watch?v=kkHN2G\\_nXV0](https://www.youtube.com/watch?v=kkHN2G_nXV0) Get Hip with JHipster: Spring Boot + AngularJS + Bootstrap (Matt Raible) Devovx FR 7.4K views

<https://www.youtube.com/watch?v=baVOGuFle9M> Get Hip with JHipster: Spring Boot + AngularJS + Bootstrap de Matt Raible Devovx 32K vistas

[https://www.youtube.com/watch?v=g-3tfSh\\_4g8](https://www.youtube.com/watch?v=g-3tfSh_4g8) JHipster Demo Ippon Technologies 13K vistas

<https://www.youtube.com/watch?v=72K7cMCv7eM&t=6s> Microservicios para las masas con Spring Boot, JHipster y JWT de Matt Raible Devovx 4.7K vistas

<https://www.youtube.com/watch?v=d1MEM8PdAzQ> JHipster 20 minutos tutorial 13 de marzo de 2015

Estos videos contienen material que es demasiado general para ser útil. Repiten lo que otros muestran (y son más fáciles de entender):

<https://www.youtube.com/watch?v=zHtd8qav8kk> Jhipster - Introduction, build simple application (Part 1) Jan 1, 2017 JHipster learning with Yappy Toy

<https://www.youtube.com/watch?v=-4gUDncO11U> Jhipster - Introduction, build simple application - Create database by using JDL(Part 2) JHipster learning with Yappy Toy 2.5K views

<https://dzone.com/articles/building-microservices-with-netflix-oss-apache-kaf>

Michael Hoffman (@mhl\_inc, [michaelhoffmaninc.com](http://michaelhoffmaninc.com)) released on Dec 11, 2017 a 2h 31m video course <https://app.pluralsight.com/library/courses/scaffolding-spring-boot-and-angular-with-jhipster/discussion> Scaffolding Spring Boot and Angular Web Apps with JHipster <https://github.com/michaelhoffmantech/jhipster-reference-material/wiki/JHipster-Installation>

## More on DevOps

This is one of a series on DevOps:

1. [DevOps\\_2.0](#)
2. [ci-cd \(Continuous Integration and Continuous Delivery\)](#).
3. [User Stories for DevOps](#)

4. [Git and GitHub vs File Archival](#)
5. [Git Commands and Statuses](#)
6. [Git Commit, Tag, Push](#)
7. [Git Utilities](#)
8. [Data Security GitHub](#)
9. [GitHub API](#)
10. [TFS vs. GitHub](#)
  
11. [Choices for DevOps Technologies](#)
12. [Java DevOps Workflow](#)
13. [AWS DevOps \(CodeCommit, CodePipeline, CodeDeploy\)](#)
14. [AWS server deployment options](#)
  
15. [Cloud regions](#)
16. [AWS Virtual Private Cloud](#)
17. [Azure Cloud Onramp](#)
18. [Azure Cloud](#)
19. [Azure Cloud Powershell](#)
20. [Bash Windows using Microsoft's WSL \(Windows Subsystem for Linux\)](#)
  
21. [Digital Ocean](#)
22. [Cloud Foundry](#)
  
23. [Packer automation to build Vagrant images](#)
24. [Terraform multi-cloud provisioning automation](#)
  
25. [Powershell Ecosystem](#)
26. [Powershell on MacOS](#)
27. [Powershell Desired System Configuration](#)
  
28. [Jenkins Server Setup](#)
29. [Jenkins Plug-ins](#)
30. [Jenkins Freestyle jobs](#)
31. [Jenkins2 Pipeline jobs using Groovy code in Jenkinsfile](#)
  
32. [Dockerize apps](#)
33. [Configuración de Docker](#)
34. [Docker Build](#)
  
35. [Maven en MacOSX](#)
  
36. [Ansible](#)
  
37. [Configuración de MySQL](#)

38. [Exploración de código estático de SonarQube](#)

39. [API de administración de Microsoft](#)

40. [API de administración de Amazon](#)

41. [Escenarios para carga](#)

La aplicación JHipster y el generador DevOps fueron publicados en 12 de septiembre de 2017.

0 Comments

wilsonmar.github.io

 **Javier Martín Alon...** ▾

 Recommend 1

 Tweet

 Share

Sort by Best ▾



Start the discussion...

Be the first to comment.

ALSO ON WILSONMAR.GITHUB.IO

### Git HEAD commit-ish references

1 comment • 2 years ago

**tonterias** — Really good one, Thanks for the video. Small and to the point.

### Gulp Task Runner

1 comment • 3 years ago

**Susan Forman** — I need help. I have this after running the first two commands in your post. I used

### Data visualization using Tableau

1 comment • 2 years ago

**om vijay** — Great post dear. It definitely has increased my knowledge on Tableau. Please keep sharing similar

### Hygieia DevOps Dashboard

2 comments • 2 years ago

**Ritesh Shah** — Another question: has anyone seen unsupported media type error on dashboards? Any ideas why

 [Subscribe](#)  [Add Disqus to your site](#) [Add Disqus](#)

© 2019 Wilson Mar. Desarrollado por Jekyll utilizando el tema Minimal Mistakes .