

Configuración del Proyecto Angular: evitar caché en el cliente en cada despliegue

Requerimientos de Configuración del Proyecto

El proyecto de Angular debe tener configurado el despliegue continuo con Docker.

Primeros Pasos ===

1. Modificación del Dockerfile de la construcción del Proyecto Ubicar el archivo Dockerfile y agregar la opción `-output-hashing=all` en la misma línea donde se ejecuta la construcción del proyecto.

Nuestra línea de código debería quedar así:

```
RUN npm run build:${TYPE} --output-hashing=all
```

2. Creación del Script para Renombrar `index.html`: Crear un archivo de Node.js llamado `'rename_index_html.js'`, que se encargue de renombrar el `index.html` con un nombre basado en un timestamp.

Esto obligará al navegador a tomar los archivos del servidor en lugar de la caché, ya que el nombre del archivo cambiará en cada despliegue.

Código del archivo `rename_index_html.js` (guardar en la raíz del proyecto):

```
const fs = require('fs'); const path = require('path'); Ruta del archivo
generado en `dist/` const distPath = path.join(__dirname, 'dist',
'nombreProyectoAngular', 'browser'); Cambia 'nombreProyectoAngular' const
indexPath = path.join(distPath, 'index.html'); Generar un timestamp único
para el nombre const timestamp = Date.now(); const newIndexName =
`index.${timestamp}.html`; const newIndexPath = path.join(distPath,
newIndexName); Renombrar el archivo fs.renameSync(indexPath, newIndexPath);
console.log(`Renamed index.html to ${newIndexName}`);
```

3. Agregar la Ejecución del Script en el Dockerfile después de la línea que construye el proyecto en el archivo Dockerfile, agregar la ejecución del script para renombrar el `index.html`:

```
RUN node rename_index_html.js
```

4. Ajuste de Nginx en el Dockerfile En el mismo archivo Dockerfile, después de la línea:

```
COPY default.conf /etc/nginx/conf.d/default.conf
```

Agregar el siguiente código para que Nginx sirva el archivo `index.html` sin importar su nombre:

```
RUN HTML_FILE=$(find /usr/share/nginx/html -type f -name "index*.html" | head
-n 1 | xargs -I {} basename {}) && echo "Encontrado: $HTML_FILE" && \ sed -i
"s/index\.html/$HTML_FILE/g" /etc/nginx/conf.d/default.conf
```

Este comando encuentra el archivo `index.html` generado por Angular, captura su nombre dinámico y

lo reemplaza en la configuración de Nginx.

Estos son los cuatro pasos necesarios para evitar que el navegador del cliente almacene en caché el index.html y garantizar que siempre tome la versión más reciente del servidor en cada despliegue.

Por último, el archivo Dockerfile debería verse similar a la siguiente imagen:

```
# Usa una imagen base de Node.js
FROM node:20-alpine AS build-step

# Establecer el límite de memoria para Node.js
ENV NODE_OPTIONS=--max_old_space_size=4096

# Variable de entorno que define el ambiente
ARG TYPE

# Crea un directorio /app en la imagen
RUN mkdir -p /app

# Establece el directorio de trabajo
WORKDIR /app

# Copia los archivos de tu proyecto al directorio de trabajo
COPY package.json /app

# Instala las dependencias del proyecto
RUN npm install

# Copia el resto de los archivos de tu proyecto al directorio de trabajo
COPY . /app

# Construye la aplicación Angular
RUN npm run build:${TYPE} --output=hashing-all
#
RUN node rename_index.html.js
# Configura la imagen de producción de Nginx
FROM nginx:alpine

# Limpia directorio html
RUN rm -rf /usr/share/nginx/html/*

# Copia los archivos generados de la compilación de Angular a la carpeta de Nginx
COPY --from=build-step /app/dist/mfcajamenor/browser /usr/share/nginx/html

# Copia la configuración personalizada de Nginx
COPY default.conf /etc/nginx/conf.d/default.conf

RUN HTML_FILE=$(find /usr/share/nginx/html -type f -name "index.html" | head -n 1 | xargs -I {} basename {})
sed -i "s/index.html/$HTML_FILE/g" /etc/nginx/conf.d/default.conf

# Expone el puerto 80 para que se pueda acceder a la aplicación desde el navegador
EXPOSE 80

# Comando para iniciar Nginx cuando se ejecute el contenedor
CMD ["nginx", "-g", "daemon off;"]
```

From:
<http://wiki.adacsc.co/> - Wiki

Permanent link:
<http://wiki.adacsc.co/doku.php?id=ada:howto:sicoferp:factory:new-migracion-sicoferp:front:limpieza-cache-condicionado&rev=1738594073>

Last update: 2025/02/03 14:47

