# BLUE PRINTS - PARTE 1 (COMPONENTES Y CONTENEDORES)

**AUTORES:**

YORKS GOMEZ – CESAR VÁSQUEZ

# PROFESOR:

JAVIER IVAN TOQUICA

ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

ARQUITECTURAS DE SOTFWARE - GRUPO # 1

INGENIERÍA DE SISTEMAS

BOGOTÁ D.C.

2023

# INTRODUCCIÓN

# En este laboratorio entenderemos una forma de visualizar la organización por medio de los componentes de un sistema y las relaciones de dependencia entre ellos. Además, recordaremos el concepto de los contenedores los cuales serán usados con el fin de agrupar una aplicación con todos sus archivos necesarios en un entorno de ejecución. Realizaremos la inyección de dependencias por medio de springboot, configurando por medio de anotaciones spring, como por ejemplo services, autowired, components, qualifiers, entre otros. Profundizando en cada concepto, entendiendo cómo funciona y de qué sirve.

# DESARROLLO

# Taller

# 2. Revisamos el archivo de configuración de spring, el cual nos indica que Spring buscará automáticamente los Beans disponibles.

# Captura de pantalla de computadora Descripción generada automáticamente

# 3. Haciendo uso de la configuración de Spring basada en anotaciones marcamos con las anotaciones @Autowired y @Service las dependencias que deben inyectarse (EnglishSpellChecker y SpanishSpellChecker), y los 'beans' candidatos a ser inyectados (GrammarChecker).

# GrammarChecker será un bean, que tiene como dependencia algo de tipo 'SpellChecker'.

# EnglishSpellChecker y SpanishSpellChecker) serán las dependencias que se deben inyectar.

# Interfaz de usuario gráfica, Texto Descripción generada automáticamente

# 

# 4. Hacemos el programa de prueba, donde se crea una instancia de GrammarChecker mediante Spring, y se hace uso de esta:

# Captura de pantalla de computadora Descripción generada automáticamente

# Obtenemos de primeras el resultado exitoso de la inyección con EnglishSpellCheker.

# 

# 5. Modificamos la configuración con anotaciones para que el Bean ‘GrammarChecker‘ ahora haga uso de la clase SpanishSpellChecker para que a ‘GrammarChecker‘ se le inyecte SpanishSpellChecker en lugar de EnglishSpellChecker.

# Captura de pantalla de computadora Descripción generada automáticamente

# Obtenemos ahora el resultado exitoso de la inyección con SpanishSpellCheker

# DIAGRAMA DE COMPONENTES

# Sin Filtros

# Con Filtros

# CONCLUISIONES

# Logramos comprender el uso adecuado de las dependencias de spring con su respectiva configuración. Usando las anotaciones para hacer la inyección al momento de ser creado el vean.

# Entendimos y planteamos lo que a nuestro parecer es el diagrama de componentes correcto de la aplicación, realizando de forma que entendimos como estaba compuesto el sistema y como se relaciona.

# Fortalecimos aspectos y conceptos que no teníamos del todo claro antes de iniciar este laboratorio.