



# **Máster Universitario de Investigación en Ingeniería De Software Y Sistemas Informáticos**

---

**31105151 - Trabajo Fin de Máster**

**Autor**

César Hugo Bárzano Cruz

**Tutor**

Rubén Heradio

Generative Cloud Manager: code-runner

—  
2019/2020



# **Máster Universitario de Investigación en Ingeniería De Software Y Sistemas Informáticos**

---

**31105151 - Trabajo Fin de Máster Generative Cloud Manager:  
code-runner**

Consultar el tipo de trabajo: Trabajo Propuesto por Alumno

**Autor**

César Hugo Bárzano Cruz

**Tutor**

Rubén Heradio



—  
2019/2020

## Resumen

Este proyecto muestra como el uso de la programación generativa y las tecnologías de la información mejoran el proceso aprendizaje y desarrollo software en el ámbito de las soluciones orientadas a la web 2.0.

Se pretende proporcionar al desarrollador software de un sistema web orientado a la generación, construcción y despliegue de aplicaciones web o servicios cloud de diversa naturaleza partiendo de una especificación de alto nivel. Normalmente, las etapas de desarrollo software están segmentadas en el desarrollo del código, la compilación de este código junto con las dependencias de terceros necesarias para producir un ejecutable, binario o artefacto para su posterior ejecución, la configuración necesaria para poder ejecutar el artefacto resultante así como la infraestructura virtual necesaria para desplegar y soportar la actividad de negocio de dicho software, siendo este accesible desde Internet.

Con este proyecto no se pretende desprestigiar a los procesos de desarrollo software tradicionales pero si mostrar como la unificación de los conceptos comunes entre aplicaciones web de diversa naturaleza así como la automatización genérica de las etapas básicas de producción software pueden dar lugar a soluciones funcionales ahorrando al desarrollador numerosas horas de trabajo. Otro enfoque igual de valido es el de dotar al desarrollador inexperto de un punto de partida en su aprendizaje.

Este proyecto pone de manifiesto que independientemente del tipo de aplicación web o servicio cloud que el desarrollador quiera producir existen unas etapas básicas, comunes y transversal por las que la gran mayoría aplicaciones han de pasar comenzando por la implementación del código fuente hasta alcanzar la madurez necesaria para ser accesibles a través de Internet. En este proyecto dichas etapas se sintetizan de la siguiente manera:

1. **Code Generation** Etapa inicial donde se sintetiza el proceso de desarrollo software, es decir se produce el código necesario para la solución especificada por el desarrollador.
2. **Artefact Generation** El objetivo de esta etapa es aprovechar el código fuente generado en la etapa Code Generation para producir un artefacto auto-contenido junto con la configuración necesaria para su ejecución en un entorno virtual.
3. **Deploy Generation** En esta etapa se aprovecha el producto de

la etapa Artefact Generation para disponibilizar la solución en Internet.

En los siguientes capítulos se profundiza en estas ideas, detallando cuales son los objetivos de alto nivel o casos de uso que se esperan del sistema así como se entrará en detalle de cada una de estas etapas.



# Índice general

<b>1. Introducción</b>	<b>9</b>
1.1. Motivación . . . . .	9
1.2. Objetivos . . . . .	9
<b>2. Descripción del Problema</b>	<b>11</b>
2.1. El Problema . . . . .	11
2.2. Generadores de aplicaciones . . . . .	11
2.3. Despliegue de código . . . . .	11
<b>3. Descripción de la Solución</b>	<b>13</b>
3.1. Análisis . . . . .	14
3.1.1. Requisitos de Información . . . . .	14
3.1.2. Requisitos Funcionales . . . . .	15
3.1.3. Requisitos No Funcionales . . . . .	15
3.1.4. Casos de Uso . . . . .	15
<b>4. Medición y Evaluación</b>	<b>17</b>
<b>5. Conclusiones</b>	<b>19</b>
<b>6. Anexo</b>	<b>23</b>
6.1. Plantilla Corrección Tipo Test . . . . .	23





# **Índice de figuras**



# Capítulo 1

## Introducción

### 1.1. Motivación

La motivación de este proyecto es la de mejorar el proceso de desarrollo de aplicaciones web y servicios cloud, ayudando al desarrollador a comprender y adquirir conocimientos relativos a las tecnologías de la información. En el proceso de desarrollo de software informático no existe un proceso universal que indique las pautas a seguir para la correcta producción software. Comúnmente, este proceso ha de amoldarse a la naturaleza del software que se desea desarrollar, a la tecnología con la que ha de desarrollarse y como no a los recursos humanos o desarrolladores que van a construir dicho software. La motivación de este proyecto es la de dotar a los desarrolladores de un sistema que los abstraiga de estas etapas, premiando la rápida producción y disponibilidad del software deseado.

### 1.2. Objetivos

El objetivo de este proyecto es el de conseguir un sistema generativo y automatizado de aplicaciones que asista a los desarrolladores en el proceso de producción software, minimizando los tiempos y costes en el desarrollo y optimizando el software resultante mediante el uso de buenas prácticas. El principal objetivo que cubre este sistema es el lema "From appSpec to Cloud.<sup>es</sup> decir, dada una especificación de aplicación a alto nivel, materializar dicha especificación en código fuente junto con lo necesario para que sea accesible por Internet.

**OBJ-1.**

**OBJ-2.**



## **Capítulo 2**

# **Descripción del Problema**

### **2.1. El Problema**

El auge de las tecnologías de la información a nivel empresarial, social, ético...blacbla

La cloud esta formada por un conjunto de tecnologías que habilitan al consumidor una serie de servicios de información para ser consumidos vía Internet. Cada vez son mas y mas las empresas y entidades confían en este tipo de servicios para realizar sus actividades económicas. La diversidad de soluciones demandadas ha creado un ecosistema en el que interactúan diversos sistemas y diversas tecnologías. Debido a esto la demanda de personal cualificado o desarrolladores también se ha incrementado.

Hoy en día la sociedad esta rodeada de

Diversidad de aplicaciones cloud. Nuevas tecnologías entorno variado y cambiante. Demanda de nuevas tecnologías. Capacitación para desarrolladores. Diversidad de aplicaciones, mismo tratamiento, simplificar proceso de aprendizaje y desarrollo software. FullStack.

### **2.2. Generadores de aplicaciones**

### **2.3. Despliegue de código**



## Capítulo 3

# Descripción de la Solución

Con el objetivo de generalizar y definir aplicación software se han decidido usar las partes:

Repositorio Readme Makefile Licencia src código fuente config configuración

1. **Code Generation** Etapa inicial donde se sintetiza el proceso de desarrollo software, es decir se produce el código necesario para la solución especificada por el desarrollador.
2. **Artefact Generation** El objetivo de esta etapa es aprovechar el código fuente generado en la etapa Code Generation para producir un artefacto auto-contenido junto con la configuración necesaria para su ejecución en un entorno virtual.
3. **Deploy Generation** En esta etapa se aprovecha el producto de la etapa Artefact Generation para disponibilizar la solución en Internet.

Programación generativa: generar, automatizar y despliegue accesible a Internet.

La aplicación software es el elemento de negocio de este sistema. Para materializar esto, el sistema crea un repositorio de código por cada aplicación creada para albergar la producción de código especificado por el desarrollador.

Generar: Aplicaciones completas, independientemente de la tecnología, utilidad o fin. Las aplicaciones generadas por el sistema son auto-contenidas y automatizadas, es decir mantienen una semántica común

build test run pull push

Lo que permite al desarrollador integrar fácilmente nuevos cambios.

Definir el período producción dentro del marco de este sistema, consiste en la siguiente serie de etapas:

## 3.1. Análisis

Como se ha visto, el termino producción en el termino de este sistema considera

### 3.1.1. Requisitos de Información

Los requisitos de información se caracterizan por reunir la información relevante para la solución, que debe gestionar y almacenar el sistema software.

**RI-1. Workspace:** Representación del espacio de trabajo del usuario. Contiene el conjunto de las aplicaciones que el usuario ha creado usando el sistema. Concepto similar al del Escritorio o "Desktop" de un sistema operativo. Contenido: nombre de usuario o propietario del workspace, referencia al conjunto de aplicaciones generadas.

**RI-2. App:** Representación de cada uno de las aplicaciones generadas por el sistema. Contenido:

```
1 {
2   "_id" : "appName",
3   "repo" : "https://github.com/user/appName.git",
4   "spec" : {
5     "dockerId" : "appNameDockerId",
6     "port" : "4343",
7     "nature" : "staticApp"
8   },
9   "des" : "",
10  "url" : "http://deploy.domain:4343",
11  "owner" : "user>s",
12  "status" : "running"
13 }
```

**RI-3. Información de Usuario:** Información sensible del usuario o desarrollador que va a usar el sistema. Credenciales al sistema de gestión de repositorios para el código fuente, en este caso relativa al perfil de github. Contenido: nombre de usuario, correo electrónico, organización.



### 3.1.2. Requisitos Funcionales

Como se define en la ingeniería de requisitos, los requisitos funcionales establecen el comportamiento del sistema.

**RF-1. Gestión de aplicaciones:** El sistema ha de gestionar el ciclo de vida de las aplicaciones creadas.

RF-1.1. El sistema permitirá crear aplicaciones.

RF-1.2. El sistema permitirá eliminar aplicaciones.

RF-1.3. El sistema permitirá ejecutar aplicaciones.

RF-1.4. El sistema permitirá detener aplicaciones.

RF-1.5. El sistema permitirá visualizar información de las aplicaciones.

**RF-2. Gestión de credenciales:** El sistema proporcionará las credenciales necesarias para trabajar con las aplicaciones .

RF-2.1. El sistema permitirá

### 3.1.3. Requisitos No Funcionales

Los requisitos no funcionales, se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar por el sistema, sino características de funcionamiento.

**RNF-1** Se necesitará acceso a Internet para utilizar las funcionalidades del sistema.

**RNF-2** Se necesitará una cuenta en github para utilizar las funcionalidades del sistema.

### 3.1.4. Casos de Uso

Los diagramas de casos de uso, son diagramas UML que representan gráficamente a todos los elementos que forman parte del modelo de casos de uso junto con la frontera del sistema. Delimitan el sistema a diseñar. Determinan el contexto del uso del sistema. Describen el punto de vista de los usuarios en el sistema.



## **Capítulo 4**

# **Medición y Evaluación**



## **Capítulo 5**

# **Conclusiones**



# Bibliografía

- [1] Tema 1, pag 8. *Visión por computador: imágenes digitales y aplicaciones*. Gonzalo Pajares Martinsanz y Jesús Manuel de la Cruz García





## **Capítulo 6**

### **Anexo**

#### **6.1. Plantilla Corrección Tipo Test**