

<b>CSCanigó</b>	MANUAL D'USUARI SIC	CODI
	SIC Manual Usuari v3.0.0.docx	
	N. versió: 3.0.0	Pàg. 1 / 22

Servei 8.30 – Arquitectura de Desenvolupament

# Manual d'Usuari


## Servei d'Integració Continua

### Tema/Detall:

Manual d'usuari del Servei d'Integració Continua.


Versió	Data	Autor	Comentaris
3.0.0	05/05/2017	CS Canigó	Autoservei de repositoris i usuaris
Aprovació		Data	Signatura

	Preparat	Revisat	Aprovat	Autoritzar
Nom				
Signatura				
Data				


	MANUAL D'USUARI SIC	CODI
	SIC Manual Usuari v3.0.0.docx	
	N. versió: 3.0.0	Pàg. 2 / 22

## Índex

<b>1INTRODUCCIÓ.....</b>	<b>5</b>
1.1Objecte.....	5
1.2A qui va dirigit.....	5
1.3Abast.....	5
<b>2GLOSSARI DE TERMES.....</b>	<b>6</b>
<b>3PUBLICACIÓ DEL CODI FONT AL GITLAB.....</b>	<b>8</b>
3.1Actualització del codi font al repositori local.....	8
3.2Publicació del codi font al repositori de GitLab.....	8
3.3Normativa d'ús de GitLab.....	8
3.3.1Estructuració de repositoris.....	8
3.3.2Limitacions.....	8
3.3.3Carpeta especial /sic.....	9
<b>4PIPELINE JENKINS.....</b>	<b>10</b>
4.1Stage CHECKOUT.....	10
4.2Stage BUILD.....	10
4.3Stage UNIT TEST.....	10
4.4Stage SONARQUBE.....	10
4.5Stage COMMIT TEST.....	10
4.6Stage INT.....	11
4.7Stage SMOKE TEST.....	11
4.8Stage PRE.....	11
4.9Stage SMOKE TEST.....	11
4.10Stage ACCEPTANCY TEST.....	11
4.11Stage EXPLORATORY TEST.....	11
4.12Stage GENERACIÓ TAG DEFINITU.....	12
4.13Stage PRO.....	12
4.14Stage SMOKE TEST.....	12
4.15Obtenció de resultats.....	12

	MANUAL D'USUARI SIC	CODI
	SIC Manual Usuari v3.0.0.docx	
	N. versió: 3.0.0	Pàg. 3 / 22

<b>5ACCÉS AL JENKINS.....</b>	<b>13</b>
5.1Autenticació sobre la plataforma.....	13
5.2Accés al Pipeline.....	13
5.3Visualització de Resultats.....	14
5.4Peticions d'integració d'aplicacions al Jenkins.....	17
5.4.1Sol·licitud de creació de jobs de construcció i desplegament.....	17
5.4.2Sol·licitud de modificació de jobs de construcció i desplegament.....	17
5.5Normativa d'integració d'aplicacions al Jenkins.....	18
5.5.3Instal·lació de llibreries JEE, Microsoft i mòduls npm i bower.....	21
5.5.4Desplegament de binaris no generats per Jenkins.....	22

	MANUAL D'USUARI SIC	CODI
	SIC Manual Usuari v3.0.0.docx	
	N. versió: 3.0.0	Pàg. 4 / 22

## 1 INTRODUCCIÓ

### 1.1 Objecte

L'objectiu d'aquest document és oferir als futurs usuaris del SIC un manual de funcionament de la plataforma. En aquest manual es descriuen tots els flux d'execució possibles així com la interacció que tenen amb els usuaris.

Aquest manual agafa com a punt de partida el fet que l'usuari estigui donat d'alta en la plataforma i que l'aplicació o aplicacions amb les que vulguin interaccionar també estiguin donades d'alta.

L'estructura d'aquest document està organitzada en torn a les tasques que els usuaris poden realitzar dintre de la plataforma. S'han previst els següents cassos d'ús:


- Instal·lació i ús del client de GIT per accedir al repositori de codi.
- Autenticació sobre la plataforma.
- Accés als diferents Jobs.
- Visualització genèrica de resultats.

### 1.2 A qui va dirigit

Aquest document va dirigit a tots els usuaris del Servei d'Integració Contínua.

### 1.3 Abast

L'abast d'aquest document inclou els processos propis del SIC des de la perspectiva dels *release managers* i responsables de projecte. Existeixen una sèrie de requisits pel que fa al software, queda però fora de l'abast d'aquest document detallar-los. Es troben ben definits en la "Norma J2EE dels serveis TIC Centrals" i en la "Guia de Desenvolupament en J2EE" per aplicacions JAVA, i els documents "Nomenclatura i esquema de base de dades" (norma SC-NOR11-01) i Desenvolupament WEB (NOR27). Cal que els desenvolupadors els coneguin be per assegurar que codi entregat pugui ser desplegat als Serveis TIC Centrals.

	MANUAL D'USUARI SIC	CODI
	SIC Manual Usuari v3.0.0.docx	
	N. versió: 3.0.0	Pàg. 5 / 22

## 2 GLOSSARI DE TERMES

- **Integració Continua (CI)**

La integració contínua (o CI, per Continuous Integration) és una pràctica de desenvolupament de programari en la que els membres d'un equip integren la seva feina freqüentment, en general una vegada com a mínim al dia – generant múltiples integracions al dia. Cada integració és verificada per una construcció automàtica (incloent les proves) per detectar errors d'integració tan ràpid com sigui possible.

- **Servei d'Integració Continua (SIC)**

El Servei del CTTI que dona suport a la integració continua, així com a d'altres aspectes del cicle de vida del programari

- **Cicle de vida del programari**

El cicle de vida del programari és el concepte que engloba tots els aspectes lligats al desenvolupament d'una aplicació, des de la presa de requeriments fins al desplegament en un entorn productiu. També conegut com a ALM (per Application Lifecycle Management).

- **Jenkins**

Jenkins és una eina d'integració continua 100% Java que s'executa en un servidor d'aplicacions tipus Tomcat o WebLogic. És un projecte de programari lliure que darrerament ha obtingut molta popularitat i diversos premis. El SIC està basat en Jenkins. Jenkins necessita tenir per sota una eina de gestió de la construcció i un repositori de codi.

- **Sistema de Gestió de la Construcció**


Un Sistema de Gestió de la Construcció (o Build Management, o Build Automation System) permet automatitzar en un script les tasques pròpies del desenvolupament diari com: generació d'executables a partir del codi font, execució de proves unitàries, creació de documentació (javadoc), etc.

- **Maven**

Maven és una eina de programari lliure que permet l'automatització de la construcció. A partir del codi font i un fitxer descriptor del projecte és capaç de generar els executables a desplegar. Una de les seves grans virtuts és la gestió de les dependències del projecte (de quines llibreries depèn, i de quina versió d'elles). Té una arquitectura basada en plug-ins que li permeten fer moltes més coses.

- **Repositori de codi / Sistema de Control de Versions**

Un sistema de control de versions manté diferents versions de molts tipus de documents i permet marcar-los (**tags**) i seguir múltiples camins d'evolució (**branches**) a partir d'una

	MANUAL D'USUARI SIC	CODI
	SIC Manual Usuari v3.0.0.docx	
	N. versió: 3.0.0	Pàg. 6 / 22

branca per defecte (**master**). El lloc on físicament s'emmagatzemen totes les versions és el repositori. El procés de copiar una nova versió d'un fitxer en el sistema de control de versions local s'anomena **commit** o **check-in**. La publicació d'aquesta nova versió al repositori central del SIC s'anomena **push**. En el cas d'un entorn de desenvolupament, això permet disposar d'un conjunt coherent de fitxers de codi font a partir dels quals generar els executables, així com mantenir múltiples bases de codi per separat (per exemple, la versió 1.x i la versió 2.x que evolucionin en paral·lel i per separat).

- **Git**

És un sistema de control de versions pensat especialment per aplicacions amb molt arxius de codi font i gestionades per molta gent.

Característiques principals:

- Rapidesa en la gestió de branques i barrejat de versions: molt potent per desenvolupaments no lineals.
- Gestió distribuïda: cada desenvolupador té una còpia local del codi sencer.

- **GitLab**

És un aplicatiu que publica una interfície web per facilitar l'administració i l'ús del sistema de control de versions GIT.

- **Jenkins Pipeline**

Una Pipeline de Jenkins és un nou tipus de job de Jenkins disponible en el core del producte des de la versió 2.0. Aquest nou tipus de job permet tenir la configuració del mateix en un arxiu anomenat jenkinsfile. El jenkinsfile és un arxiu groovy que recull les tasques que s'han d'executar i permet dividir-les en *stages* (fases).


- **Release Manager / Gestor de Lliuraments**

Persona de l'equip de desenvolupament que s'encarrega gestionar les versions d'una aplicació. És qui efectua els commits i posa els tags de versió en el repositori de codi.

- **Desplegament Automatitzat**

L'automatització del desplegament consisteix en escriure un script que contingui les tasques relacionades amb el desplegament d'una versió d'una aplicació, fent servir les interfícies proporcionades pels sistemes on s'ha de desplegar. Això permet reduir els punts de fallida ja que s'automatitzen tasques repetitives. Els servidors d'aplicacions com Weblogic acostumen a donar una API que pot ser invocada des d'una eina de construcció com Maven. El SIC fa ús d'aquestes funcionalitats per a assolir desplegaments automàtics als entorns d'integració.

En el cas de les tecnologies Microsoft, els servidors IIS poden permetre la connexió, la instal·lació y l'actualització d'aplicacions des d'un equip client utilitzant MS Web Deploy.

	MANUAL D'USUARI SIC	CODI
	SIC Manual Usuari v3.0.0.docx	
	N. versió: 3.0.0	Pàg. 7 / 22

### 3 Publicació del codi font al GitLab

#### 3.1 Actualització del codi font al repositori local

Els desenvolupadors hauran de tenir instal·lat/configurat el SCM Git al seu equip.

Git proporciona a cada desenvolupador una còpia local del codi font de l'aplicació (repositori local).

El desenvolupador anirà actualitzant aquest "repositori local" amb els seus canvis executant COMMIT.

#### 3.2 Publicació del codi font al repositori de GitLab

Una vegada s'hagi validat internament el codi de l'aplicació, es publicarà al repositori GitLab executant PUSH.

En aquest moment, al fer el PUSH al GitLab, automàticament es llençarà al Jenkins la execució del Pipeline associat a l'aplicació, en el cas que l'aplicació tingui un job creat a tal efecte.

Les múltiples tasques que fa aquest pipeline, es descriuen a l'apartat 4.

#### 3.3 Normativa d'ús de GitLab

##### 3.3.1 Estructuració de repositoris

Al GitLab del SIC, hi són precreats una sèrie de grups que es corresponen -un a un- amb tots els codis d'aplicació. D'aquesta manera, cada grup disposarà de tot el codi font corresponent al codi d'aplicació que ve representat pel seu nom.


Dins de cadascun d'aquests grups, s'albergaran tots els projectes del codi d'aplicació. Hi haurà codis d'aplicació amb només un sol projecte i n'hi haurà que tindran més d'un. El criteri general per decidir què és un projecte i que no és preguntar-se si aquest conjunt de codi font és susceptible de ser versionat de forma independent al de la resta de projectes del codi d'aplicació.

Per exemple, una aplicació basada en microserveis, requerirà un projecte per a cada microservei.

##### 3.3.2 Limitacions

Per a propiciar el bon ús del repositori i les bones pràctiques en la gestió del cicle de vida de les aplicacions, s'han establert les següents limitacions:

1. En quant a la mida:
  1. La mida màxima dels arxius serà de 20 MB. Qualsevol arxiu multimèdia hauria més

	MANUAL D'USUARI SIC	CODI
	SIC Manual Usuari v3.0.0.docx	
	N. versió: 3.0.0	Pàg. 8 / 22

gran que aquesta mida màxima hauria de proporcionar-se mitjançant una altra via.

2. En quant al nom:

1. No es permet l'existència de carpetes amb nom «node\_modules». Aquesta carpeta es sol utilitzar per a la descàrrega de dependències en aplicacions Node.js en el procés de construcció. Per tant, no cal afegir-la al repositori, ja que el propi Jenkins l'obtindrà en el procés de construcció.

3. En quant a l'extensió:

1. No es permet l'existència d'arxius amb les següents extensions:

- JAR
- WAR
- EAR
- DLL
- EXE

Aquestes extensions són pròpies d'arxius binaris. Aquests arxius, si són generats per proveïdor d'aplicacions, també s'haurien de generar a través de Jenkins. Si, en canvi, són generats per tercers, s'haurien de depositar al repositori d'artefactes Nexus.


Tota violació de les limitacions aquí exposades invalidaran el push al servidor oficial. Durant el procés de push, es mostrarà a l'usuari un missatge d'error amb els incumpliments detectats.

### 3.3.3 Carpeta especial /sic

Es requereix l'existència de la carpeta especial **/sic**. Aquesta carpeta albergarà el següent contingut:

- L'arxiu **versio.txt** (sense accent), que serà un arxiu de text amb una sola línia i el seu contingut consistirà amb la versió que s'ha pujat al repositori (per exemple: **1.0.0**).



	MANUAL D'USUARI SIC	CODI
	SIC Manual Usuari v3.0.0.docx	
	N. versió: 3.0.0	Pàg. 9 / 22

## 4 Pipeline Jenkins

Una vegada que el Release Manager d'una aplicació valida el codi font, executa un PUSH per tal de publicar-lo al repositori GitLab. Això fa que automàticament s'executi al Jenkins el Pipeline que l'equip del SIC ha creat i associat per l'aplicació.

Aquest Pipeline realitza multitud de tasques organitzades en STAGES, que es descriuen a continuació.

En cas de produir-se algun error a qualsevol etapa, l'execució del Pipeline es cancel·larà i s'enviarà un correu electrònic al responsable de l'aplicació informant del que ha passat.

### 4.1 Stage CHECKOUT

A aquesta etapa el Jenkins es connecta al repositori GitLab i es descarrega el codi font de l'aplicació, al seu workspace.

### 4.2 Stage BUILD

A aquesta etapa el Jenkins construeix l'aplicació i els artefactes pertinents a partir del codi descarregat a l'etapa anterior.

La construcció es fa d'acord a la naturalesa de l'aplicació, és a dir, per a aplicacions Java es fa mitjançant Maven, per a aplicacions .NET mitjançant MS Build, etc.

A més a més, els artefactes resultants s'arxivaran al Jenkins de forma que estiguin disponibles per consultar per l'usuari.

### 4.3 Stage UNIT TEST

A aquesta etapa s'executaran els tests unitaris, si s'escau. Pot ser un requeriment haver-ne de tenir i anirà en funció de l'aplicació.


Actualment, només es dona suport a test unitaris JAVA a través de MAVEN.

### 4.4 Stage SONARQUBE

Aquesta etapa permetrà executar l'anàlisi de codi estàtic a través de Sonarqube. Actualment no està disponible i es fa un bypass a la següent etapa.

### 4.5 Stage COMMIT TEST

Aquesta etapa executarà els tests de commit, si s'escau. D'igual manera que a l'etapa de UNIT

	MANUAL D'USUARI SIC	CODI
	SIC Manual Usuari v3.0.0.docx	
	N. versió: 3.0.0	Pàg. 10 / 22

TEST, pot ser un requeriment haver-los de passar amb èxit.

Si els tests de commit s'executen amb èxit, es generarà un TAG, anomenat “versió construïble”, al repositori de l'aplicació.

La nomenclatura dels TAG's serà:

<versio>.B001

<versio>.B002

...

<versio>.B00N

On <versio> és la versió indicada al fitxer versio.txt que proporciona l'aplicació dins de la carpeta /sic.

Actualment, el SIC no executa tests de commit i, per tant, en aquesta etapa es genera el TAG descrit anteriorment.

## 4.6 Stage INT

Etapa que fa el desplegament automàtic de l'aplicació a l'entorn d'Integració.

## 4.7 Stage SMOKE TEST

Aquesta etapa realitza una validació bàsica per detectar que l'aplicació s'ha publicat correctament (com per exemple accedir a una URL de l'aplicació i veure que respon).

## 4.8 Stage PRE

Etapa que fa el desplegament de l'aplicació a l'entorn de Preproducció.

Aquest desplegament es farà automàticament o mitjançant petició al SAU o obertura ticket Remedy depenent de la configuració de l'aplicació.

## 4.9 Stage SMOKE TEST


Aquesta etapa realitza una validació bàsica per detectar que l'aplicació s'ha publicat correctament (com per exemple accedir a una URL de l'aplicació i veure que respon).

## 4.10 Stage ACCEPTANCY TEST

Execució dels tests automàtics d'acceptació.

## 4.11 Stage EXPLORATORY TEST

Execució dels tests manuals d'acceptació.

	MANUAL D'USUARI SIC	CODI
	SIC Manual Usuari v3.0.0.docx	
	N. versió: 3.0.0	Pàg. 11 / 22

## 4.12 Stage GENERACIÓ TAG DEFINITIU

Aquesta etapa, com el seu nom indica, genera un TAG, anomenat “versió desplegable”, al GitLab. Representa una versió que s'ha compilat, muntat i desplegat correctament, a més d'haver passat tests unitaris, de commit, d'acceptació i exploratoris. És a dir, es tracta d'una versió que ha passat tots els filtres per poder ser desplegada a l'entorn productiu.

La nomenclatura del TAG serà la versió indicada al fitxer versió.txt que proporciona l'aplicació.

## 4.13 Stage PRO

Etapa que fa el desplegament de l'aplicació a l'entorn de Producció.


Aquest desplegament es farà automàticament o mitjançant petició al SAU o obertura ticket Remedy depenent de la configuració de l'aplicació.

## 4.14 Stage SMOKE TEST

Aquesta etapa realitza una validació bàsica per detectar que l'aplicació s'ha publicat correctament (com per exemple accedir a una URL de l'aplicació i veure que respon).

## 4.15 Obtenció de resultats

En finalitzar cada execució d'un Job, es poden veure una sèrie d'informes i resultats. Aquestes dades seran documentades en l'explicació de cada Job.

	MANUAL D'USUARI SIC	CODI
	SIC Manual Usuari v3.0.0.docx	
	N. versió: 3.0.0	Pàg. 12 / 22

## 5 Accés al Jenkins

### 5.1 Autenticació sobre la plataforma

Per poder efectuar aquesta tasca l'usuari ha d'accedir a la plataforma mitjançant el formulari d'autenticació de Jenkins. La url d'accés és .

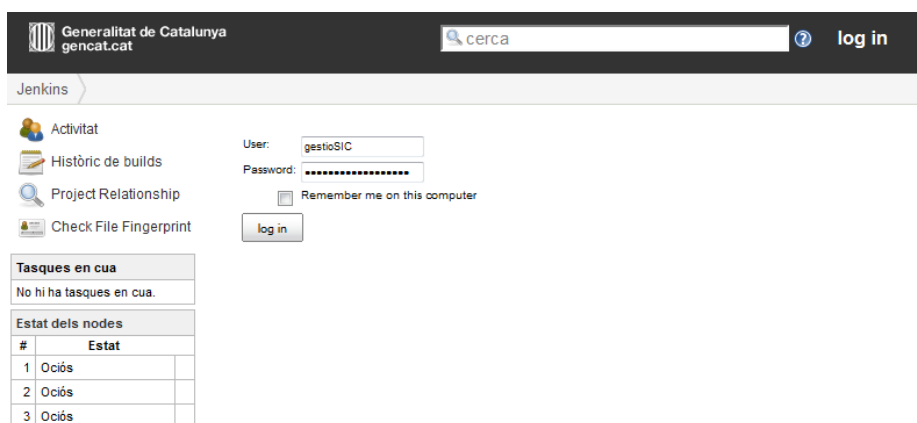


Figura 3.2.2 – 1





Una vegada fet el login, s'accedeix a la pantalla de tasques disponibles per l'usuari.

### 5.2 Accés al Pipeline


Una vegada fet el login, s'accedeix a la llista de tasques disponibles per l'usuari al menú de l'esquerra i a la vista central apareixeran les pipelines que té disponibles l'usuari.

Com es pot observar apareixen diversos camps:






El primer indica l'estat general del Job/Pipeline, **S** (Status),

Estat	Descripció
	El projecte encara no ha estat construït mai.
	L'última execució ha anat correctament.
	L'última execució ha anat correctament però és inestable.
	L'última execució ha fallat.

Taula 3.2.3 – 2


	MANUAL D'USUARI SIC	CODI
	SIC Manual Usuari v3.0.0.docx	
	N. versió: 3.0.0	Pàg. 13 / 22

El segon correspon a la salut general del Job **W** (Weather). Es calcula la salut general del projecte basant-se en una sèrie d'indicadors. En el nostre cas es basaran en l'estabilitat, cobertura i tests.

Estat	Descripció
	Indica una salut d'entre 80-100%.
	Indica una salut d'entre 60-79%.
	Indica una salut d'entre 40-59%.
	Indica una salut d'entre 20-39%.
	Indica una salut d'entre 0-19%.

Taula 3.2.3 – 3

La resta de columnes venen descrites en la taula següent;

Columna	Descripció
Name	Nom del Job o Pipeline.
Darrer muntatge correcte	Temps des de l'última execució amb èxit del Job.
Darrer muntatge fallit	Temps des de l'última execució sense èxit del Job.
Darrera durada	Durada de l'última execució del Job
Last success Version 	Icona d'execució ràpida del Job. Llença una execució del Job (build).

Taula 3.2.3 – 4

### 5.3 Visualització de Resultats

De forma genèrica es pot consultar l'estat de finalització d'un Pipeline. Aquesta informació es pot visualitzar en la pàgina principal de cada Pipeline. Per poder accedir-hi a aquesta pàgina només s'ha de fer clic en el nom del Pipeline (figura 3.2.3 – 1).

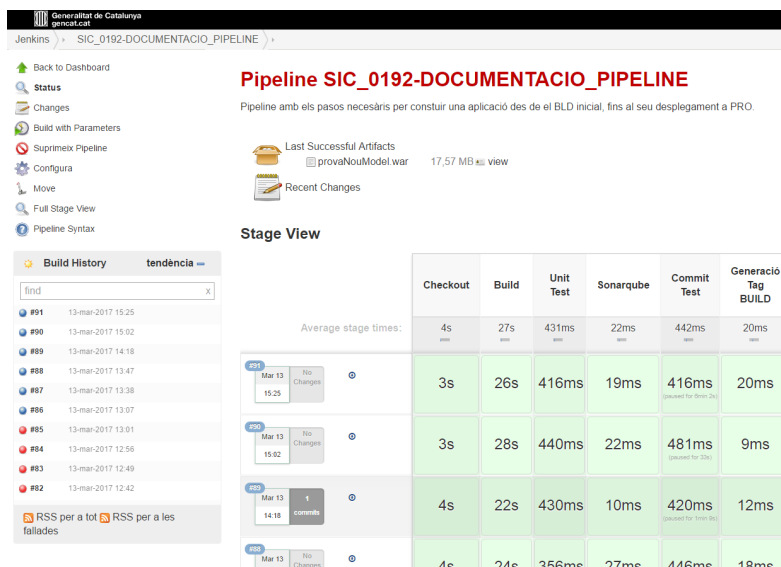


Figura 3.2.4 – 1

En el menú lateral esquerre es pot veure un quadre anomenat “Build History”. Aquest quadre mostra amb una icona l'estat de salut general del projecte. El significat és consultable en la taula 3.2.3 – 3.

Llistat a continuació apareixen els últims builds d'aquest Job. L'estat de cada un d'ells ve representat per la icona que els precedeix (veure taula 3.2.3 – 2). Es disposa també d'unes estadístiques del Job que poden veure's fent clic sobre l'enllaç “tendència”.

### Build Time Trend

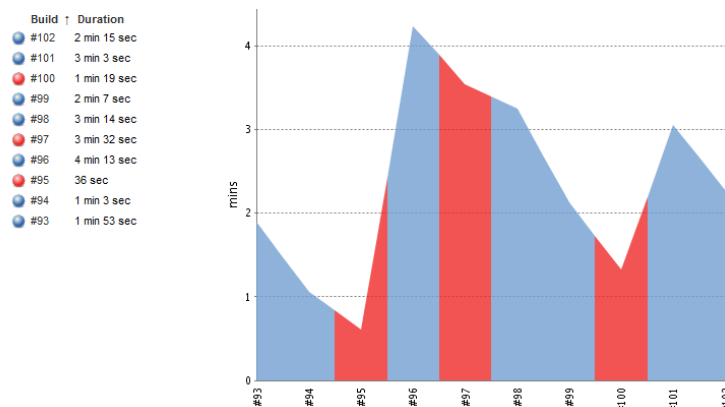









Figura 3.2.4 – 2

La gràfica mostra un històric dels builds executats sobre aquest Pipeline i el temps que han trigat cadascun així com el seu estat de finalització.

A la zona central de la pantalla es mostra una gràfica amb les darreres execucions del Pipeline, i

el resultat a cadascuna de les etapes.

	Checkout	Build	Unit Test	Sonarqube	Commit Test	Generació Tag BUILD	INT	Smoke Test	Generació Tag DEFINITIU	PRE	Smoke Test	Acceptancy Test	Exploratory Test	PRO	Smoke Test
Average stage times:	4s	27s	431ms	22ms	442ms	20ms	31s	140ms	518ms	15s	NaNy NaNd	18ms	17ms	16s	NaNy NaNd
 Mar 13 15:25 No Changes	3s	26s	416ms	19ms	416ms	20ms	32s	24ms	491ms	18s	14ms	21ms	17ms	15s	404ms
 Mar 13 15:52 No Changes	3s	28s	440ms	22ms	481ms	9ms	29s	29ms	382ms	17s	12ms	15ms	13ms	15s	427ms
 Mar 13 14:18 1 commits	4s	22s	430ms	10ms	420ms	12ms	30s	23ms	575ms	18s	15ms	12ms	11ms	15s	444ms
 Mar 13 13:47 No Changes	4s	24s	356ms	27ms	446ms	18ms	33s	14ms	421ms	18s	23ms	23ms	21ms	16s	521ms
 Mar 13 13:38 No Changes	3s	23s	418ms	18ms	395ms	10ms	30s	9ms	553ms	18s	20ms	19ms	22ms	16s	450ms
 Mar 13 13:57 No Changes	3s	26s	492ms	33ms	402ms	22ms	33s	19ms	527ms	18s	33ms	20ms	19ms	18s	454ms
 Mar 13 13:01 No Changes	4s	25s	481ms	20ms	566ms	28ms	31s	20ms	880ms	9s					

Si es vol més detall de l'estat d'un build d'un Job es pot fer clic sobre el build (figura 3.2.4 – 1). En fer això s'accedeix al detall de l'execució d'un build. La informació que es mostra depèn del tipus de Job que sigui. Tot i així sempre apareixerà la opció “Console Output”. En fer clic sobre ella es podrà accedir al log de la tasca.

Al final d'aquest log es pot veure la paraula **SUCCESS** o **FAILED** que indica si el build va anar bé o malament.

També pot donar-se el resultat **ABORTED**, el qual indicaria la cancel·lació del job per part de l'usuari.


Un detall important d'aquest build es troba en la part central de la pàgina principal del Job. En l'apartat “Last Successful Artifacts” es mostren tots els artefactes generats pel Build, mantenint-se un històric de 3 construccions (builds) per a aquest job.

## JEE

En aplicacions JEE es generen típicament dos artefactes, un WAR/EAR que representarà la part dinàmica de l'aplicació i un ZIP que representarà la part estàtica.

## PHP

En aplicacions PHP es genera un o dos ZIP's en funció de si es necessita contemplar separació entre codi estàtic i dinàmic (.php's) per ser desplegat a diferents Apache's.

	MANUAL D'USUARI SIC	CODI
	SIC Manual Usuari v3.0.0.docx	
	N. versió: 3.0.0	Pàg. 16 / 22

## .NET

En aplicacions .NET es genera un ZIP per ser desplegat al servidor.

## Buildpacks

Depén del tipus de buildpacks. Les aplicacions Java es genera l'artefacte que s'ha de desplegar. Els altres tipus es desplega el codi sense necessitat de compilar.

En finalitzar el Job correctament s'enviarà una notificació via mail al responsable del projecte. En cas de finalització incorrecta s'enviarà tant al responsable com a la Oficina tècnica de Canigó.

## 5.4 Peticions d'integració d'aplicacions al Jenkins

### 5.4.1 Sol·licitud de creació de jobs de construcció i desplegament

Per a la integració d'aplicacions al Jenkins, cal realitzar una petició de suport funcional a Remedy, tot indicant:

1. **El servei:** Framework SIC
2. **Descripció breu del suport:** "SIC – Alta Jobs aplicació [CODI DIÀLEG] [ÀMBIT]"
3. **Descripció detallada del suport:** Incloure el detall que considereu oportú.
4. Cal adjuntar el DA de l'aplicació que es desitja integrar.

Un cop la petició sigui gestionada per l'equip del SIC, es tancarà i s'obrirà un tiquet al Centre de Suport JIRA del SIC on es realitzarà el seguiment de tot el tràmit.

Per a més informació, podeu consultar l'apartat de peticions al SIC del web d'arquitectura:

### 5.4.2 Sol·licitud de modificació de jobs de construcció i desplegament


Per a la modificació de jobs d'aplicacions al Jenkins, cal realitzar una petició de suport funcional a Remedy, tot indicant:

1. **El servei:** Framework SIC
2. **Descripció breu del suport:** "SIC – Modificació Jobs aplicació [CODI DIÀLEG] [ÀMBIT]"
3. **Descripció detallada del suport:** Incloure el detall que considereu oportú.
4. Cal adjuntar el DA de l'aplicació que es desitja integrar.

Un cop la petició sigui gestionada per l'equip del SIC, es tancarà i s'obrirà un tiquet al Centre de Suport JIRA del SIC on es realitzarà el seguiment de tot el tràmit.

Per a més informació, podeu consultar l'apartat de peticions al SIC del web d'arquitectura:



	MANUAL D'USUARI SIC	CODI
	SIC Manual Usuari v3.0.0.docx	
	N. versió: 3.0.0	Pàg. 17 / 22

### 5.4.3 Execució del job pipeline de l'aplicació

Els jobs pipeline no es podran invocar directament al portal Jenkins ni es podrà sol·licitar la seva execució mitjançant una petició Remedy. Els jobs s'executaran quan es produeixi un push al projecte Git per part del lot d'aplicacions.

Es recomana ser curosos, ja que qualsevol push efectuat contra el repositori disparà el job. Per tant, és important seguir els següents punts:

1. Limitar la quantitat d'usuaris que utilitzin el servei Git del SIC. D'aquesta manera es té controlat més fàcilment quan i qui fa el push. Desde el SIC, sempre s'ha demanat que hi hagi la figura del Release Manager (gestor de revisions), que és l'encarregat entre d'altres tasques de pujar la versió final de l'entorn de desenvolupament al SIC.
2. Fer un únic push amb èxit per versió. És a dir, si el job falla en algun punt (construcció, desplegament, etc.) es pot tornar a fer un nou push amb les correccions pertinents al codi. Però un cop s'ha generat el TAG definitiu, no es permetrà fer el push de nou sense incloure una nova versió a l'arxiu del projecte **/sic/versio.txt**.

## 5.5 Normativa d'integració d'aplicacions al Jenkins

### 5.5.1 Estructuració del repositori del projecte

Seguir les indicacions especificades a l'apartat 3.3, Normativa d'ús de GitLab.

### 5.5.2 Execució de scripts de base de dades en el desplegament a Integració.

En el cas que es vulgui executar scripts de base de dades en el moment del desplegament a Integració d'aplicacions, l'usuari haurà de pujar prèviament tant el fitxer de plans com els scripts a un directori independent. A continuació, es mostra un exemple on l'hem col·locat en la carpeta "sql\_scripts":

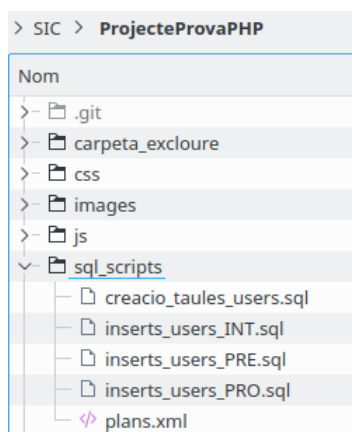



Figura 3.2.6.3 – 1

	MANUAL D'USUARI SIC	CODI
	SIC Manual Usuari v3.0.0.docx	
	N. versió: 3.0.0	Pàg. 18 / 22

El format del contingut del fitxer de plans haurà de ser el següent:

```
<llista-scripts>
<script entorn="[ INT | PRE | PRO | FOR ]" failure="[stop/continue]" idBBDD="[identificador de la base de dades]" file="[fitxer1.sql]"/>
...
<script entorn="[ INT | PRE | PRO | FOR ]" failure="[stop/continue]" idBBDD="[identificador de la base de dades]" file="[fitxer2.sql]"/>
</llista-scripts>
```


Els camps a omplir al fitxer de plans seran els següents:

- **entorn:** Determina per a quin entorn cal utilitzar l'script. Té quatre valors possibles: “*INT*” (Integració), “*PRE*” (Preproducció), “*PRO*” (Producció) i “*FOR*” (Formació). Es pot fer servir un script per a més d'un entorn. La informació facilitada en aquest atribut s'utilitza tant en els desplegaments automàtics com en els manuals. En els desplegaments automàtics, el servidor Jenkins despleguen els scripts de l'entorn que pertoca en cada etapa de desplegament. En els desplegaments manuals, empaquetarà els scripts de l'entorn concret per adjuntar-los a la petició de desplegament.
- **failure:** Podrà tenir dos valors, “*stop*” o “*continue*”. El valor “*stop*” indicarà que en cas que es produeixi algun error en el processament del script, s'aturarà el procés d'execució de scripts i no s'executarà cap més. El valor “*continue*” indicarà que tot i que es produeixin errors en el processament del script, el procés d'execució de scripts no s'aturarà.
- **idBBDD:** Identificador de la BBDD on s'executa aquest script. Aquest identificador és un nom arbitrari decidit pel proveïdor d'aplicacions. Pot ser un nom indicatiu de la base de dades a la qual s'ha de connectar l'aplicació. Pot haver-hi tants identificadors com bases de dades utilitzi l'aplicació. Exemples: (sqlserver\_dwh\_int, sqlserver\_trans\_pro, oracle\_corp1\_pre, oracle\_ens\_int, etc.).
- **file:** A part d'indicar el número del script, el departament, l'aplicació i una breu descripció del que fa el script (per exemple: insert, update, delete, create, etc), caldrà informar el fitxer com a “*.sql*” en el cas que es tracti un script SQL i com a “*.pl*” en el cas que es tracti d'un fitxer amb scripts PL/SQL. Serà molt important indicar la correcta extensió del fitxer segons el contingut, ja que sinó el processament serà erroni.

Un exemple del format seria el següent:

- **Fitxer de plans:** *CTTI\_test\_plans\_INT.xml*
- **Contingut:**

```
<llista-scripts>
<script entorn="INT" failure="stop" idBBDD="oracle int" file="creacio taules users.sql"/>
```

	MANUAL D'USUARI SIC	CODI
	SIC Manual Usuari v3.0.0.docx	
	N. versió: 3.0.0	Pàg. 19 / 22

```
<script entorn="INT" failure="continue" idBBDD="oracle_int" file="inserts_users_INT.sql"/>
<script entorn="PRE" failure="stop" idBBDD="oracle_pre" file="creacio_taulas_users.sql"/>
<script entorn="PRE" failure="continue" idBBDD="oracle_pre" file="inserts_users_PRE.sql"/>
<script entorn="PRO" failure="stop" idBBDD="oracle_pro" file="creacio_taulas_users.sql"/>
<script entorn="PRO" failure="stop" idBBDD="oracle_pro" file="inserts_users_PRO.sql"/>
</llista-scripts>
```

Pel que fa al contingut dels scripts, caldrà seguir una lògica segons quin tipus es faci servir (SQL o PL/SQL) i el nom del fitxer de scripts.

En el cas dels fitxers amb scripts PL/SQL serà imprescindible per la seva correcta execució, que es finalitzi el script correctament informant al final del script el següent:


```
/
EXIT;
```

Abans d'executar el script es comprovarà això i, en cas que no estigui indicat aquest final de fitxer, aquest no és processarà.

Un exemple de com quedaria el script PL/SQL seria el següent:

```
DECLARE
variable1 CHAR(50) := 'Test1';
variable2 CHAR(50) := Test2;
variable3 CHAR(50) := 'Valor1';
BEGIN
UPDATE taula_test
SET valor = variable3
WHERE prova1 = variable1
AND prova2 = variable2;
COMMIT;
END;
/
EXIT;
```

Durant la integració de l'aplicació al SIC, es sol·licitarà al proveïdor d'aplicacions les dades necessàries corresponents a cada identificador de Base de Dades (cadena de connexió, servidor, port, etc.).

	MANUAL D'USUARI SIC	CODI
	SIC Manual Usuari v3.0.0.docx	
	N. versió: 3.0.0	Pàg. 20 / 22

#### 5.5.2.1 Esborrament dels fitxers temporals al servidor d'aplicacions Weblogic en el desplegament a Integració d'aplicacions JEE.

Per fer redesplegaments de les aplicacions amb l'esborrament dels fitxers temporals que es creen al servidor d'aplicacions quan aquesta es desplega, les aplicacions ho podran demanar mitjançant el formulari d'alta d'aplicació en el cas que l'aplicació desplegui en un servidor d'aplicacions Weblogic.

Caldrà omplir la informació de l'apartat *“Redespiegament d'aplicacions amb esborrat de fitxers temporals al servidor d'aplicacions”*, on s'haurà d'indicar la ruta absoluta de l'artefacte dinàmic que es desplega a la màquina servidor d'aplicacions (si es coneix).

### 5.5.3 Instal·lació de llibreries JEE, Microsoft i mòduls npm i bower


#### 5.5.3.1 Introducció

Aquesta tasca ve representada pel Job amb sufix “LIB”. Aquest Job compila, construeix e instal·la una llibreria en el repositori local del SIC amb l'objectiu que sigui utilitzada per altres aplicacions en el SIC.

El nom del job aconsegueix la següent sintaxi: [nomAmbit/Departament]\_[codiDiàleg]-[nomAplicacio].


Els projectes de llibreries/mòduls han de ser projectes independents, ja que són susceptibles de ser versionats independentment de l'aplicació que els utilitza. Per tant, s'hauran de crear en un repositori exclusiu per a cada llibreria/mòdul.

#### 5.5.3.2 Execució del job

El procediment per l'execució d'aquest build és anàleg a la resta de Jobs. Hi ha dues formes de fer-ho, des de la llista de builds (figura 3.2.3 – 1) amb el botó  o bé amb el link “Build now” des de la pàgina principal del build (figura 3.2.4 – 1).

En accedir al formulari d'execució es demanarà la versió de la llibreria a compilar. Aquest valor a de coincidir amb el nom del tag on estigui el codi de la llibreria que es vulgui compilar.

L'execució satisfactòria d'aquest job farà que es compili i s'instal·li la llibreria en el repositori local del Hudson de forma que sigui utilitzable per altres aplicacions que estiguin al SIC.

	MANUAL D'USUARI SIC	CODI
	SIC Manual Usuari v3.0.0.docx	
	N. versió: 3.0.0	Pàg. 21 / 22

## 5.5.4 Desplegament de binaris no generats per Jenkins


### 5.5.4.1 Introducció

Aquest Job realitza una pujada d'artefactes i manuals a un espai compartit amb CPD per tal que aquest pugui recollir-los i poder-los desplegar adequadament.

Els artefactes a desplegar i els manuals amb les instruccions, seran especificats a l'execució del job.

El nom del job és fix per totes les aplicacions: `Pujada_Binaris`.

### 5.5.4.2 Execució del job

El procediment per poder executar el build es pot realitzar de dues formes: des de la llista de builds amb el botó  o bé amb el link "Build now" des de la pàgina principal del build.

A continuació es descriuen els camps a omplir:

- CODI\_APLICACIO.- codi de diàleg de l'aplicació
- NOM\_APLICACIO.- nom de l'aplicació
- VERSIO.- versió del producte
- FITXER\_ARTEFACTES.- fitxer dels artefactes a desplegar
- FITXER\_DOCUMENTACIO.- fitxer dels manuals amb la informació pel desplegament

El job enviarà un correu electrònic a l'usuari executor del job amb dos enllaços de descàrrega. Un per al fitxer de documentació i un altre per al fitxer de binaris. Aquests enllaços poden ser indicats a les peticions de desplegament corresponents per a que CPD pugui descarregar-los sense problemes.

### 5.5.4.3 Visualització del contingut pujat

Una vegada executat el build es poden veure els resultats. Es podrà accedir al sistema de fitxers en mode lectura a través d'una interfície web. Aquesta interfície serà accessible a través del URL <http://bin.sic.intranet.gencat.cat/binaris/> i mostrarà una estructura de carpetes organitzada de la següent manera:

Codi aplicació → Aplicació → Versió → [ bin/doc ]

Els permisos d'accés es controlaran al primer nivell (codi d'aplicació), podent-ne accedir

<b>CS</b> Canigó	MANUAL D'USUARI SIC		CODI
	SIC Manual Usuari v3.0.0.docx		
	N. versió: 3.0.0		Pàg. 22 / 22

únicament els *Release Managers* i els administradors del CPD/LLT corresponent.

En cas de finalització incorrecta s'enviarà una notificació tant al responsable del projecte com a la Oficina tècnica de Canigó (oficina-tecnica.canigo.ctti@gencat.cat).