	MANUAL D'INTEGRACIÓ SIC	0192
	SIC Manual Integracio.docx	
	N. versió: 3.0.2	Pàg. 1 / 25

Servei 8.30 – Arquitectura de Desenvolupament

# Manual d'Integració

## Servei d'Integració Contínua


<b>Tema/Detall:</b> Manual d'integració d'aplicacions JEE en el Servei d'Integració Contínua.	
<b>Llista de Distribució:</b>	
<b>Nom del Fitxer:</b>	SIC Manual Integracio.docx

Versió	Data	Autor	Comentaris
3.0.0	01/06/2017	CS Canigó	SIC 2.0
3.0.1	04/07/2017	CS Canigó	Gestió binaris + Jobs desplegament CPD
3.0.2	04/12/2017	CS Canigó	Adaptacions de Jobs desplegament CPD


Aprovació	Data	Signatura

	Preparat	Revisat	Aprobat	Autoritzar
Nom				
Signatura				
Data				

<b>1 INTRODUCCIÓ.....</b>	<b>3</b>
1.1 OBJECTE.....	3
1.2 ABAST.....	3
1.3 DESCRIPCIÓ DEL SERVEI.....	3
1.4 DOCUMENTACIÓ RELACIONADA.....	3
<b>2 SIC: CUSTÒDIA DE CODI.....</b>	<b>4</b>
2.1 REQUISITS.....	4
2.1.1 Connectivitat.....	4
2.1.2 Lliurament de codi font.....	4
2.1.3 Estructura de projectes i repositoris a GITLAB.....	4

	MANUAL D'INTEGRACIÓ SIC	0192
	SIC Manual Integracio.docx	
	N. versió: 3.0.2	Pàg. 2 / 25

2.1.4 Estructura interna de projectes.....	5
2.2 PROCEDIMENTS.....	7
2.2.1 Procés d'alta d'aplicacions.....	7
2.2.2 Procediments operatius.....	8
<b>3 SIC: TASQUES D'ANÀLISI DE CODI FONT, CONSTRUCCIÓ D'ARTEFACTES I</b>	
<b>DESPLEGAMENTS AUTOMATITZATS.....</b>	<b>9</b>
3.1 REQUISITS D'INTEGRACIÓ.....	9
3.1.1 Requisits comuns.....	9
3.1.2 Requisits per a aplicacions JEE.....	9
3.1.3 Requisits per a aplicacions .NET.....	12
3.1.4 Requisits per a aplicacions Node JS.....	13
3.2 REQUISITS PER AL DESPLEGAMENT.....	13
3.2.1 Desplegaments en servidors físics (IAAS).....	13
3.2.2 Desplegaments al CLOUD.....	14
3.2.3 Execució d'scripts contra BBDD durant els desplegaments.....	15
3.3 PROCEDIMENTS.....	16
3.3.1 Procés d'alta.....	16
3.3.2 Procediments operatius.....	16
<b>4 CONSULTES I INCIDÈNCIES.....</b>	<b>20</b>
<b>5 ANNEX A.....</b>	<b>21</b>

	MANUAL D'INTEGRACIÓ SIC	0192
	SIC Manual Integracio.docx	
	N. versió: 3.0.2	Pàg. 3 / 25

# 1 Introducció

## 1.1 Objecte

Aquest document proporciona informació respecte a la metodologia d'integració d'aplicacions que vulguin utilitzar el Servei d'Integració Continua (SIC).

Important: per aplicació al SIC s'entén aquell component de software amb versionatge, procés de construcció i desplegament independent. Dins el mateix codi de diàleg en el nou model TIC poden haver-hi N aplicacions SIC.

## 1.2 Abast

Aquest document es centra en els requisits que han de complir les aplicacions per tal de poder utilitzar el SIC en concepte de :

- Custòdia de codi font
- Eina d'anàlisi de codi, construcció d'artefactes i automatització dels desplegaments.

També s'inclouen els procediments associats al servei.


## 1.3 Descripció del servei

El Servei d'Integració Continua (SIC) té com a objectiu donar suport al cicle de vida del desenvolupament d'aplicacions. Per fer-ho, ofereix els següents conjunts d'eines:

- Servei de custòdia de codi font (Repositori SVN)
- Tasques d'anàlisi de codi, construcció d'artefactes i desplegaments automatitzats (Jobs de Jenkins)

## 1.4 Documentació Relacionada

- Llibre Normatiu J2EE

	MANUAL D'INTEGRACIÓ SIC	0192
	SIC Manual Integracio.docx	
	N. versió: 3.0.2	Pàg. 4 / 25

## 2 SIC: Custòdia de codi

### 2.1 Requisits

#### 2.1.1 Connectivitat

Per poder accedir al repositori de custòdia de codi (repositori GitLab de SIC) via VPN cal assegurar que es disposa de connectivitat cap a **git.intranet.gencat.cat**, port 443.

Si no es disposa de visibilitat a la màquina on es troba el repositori de codi del SIC, haurà d'obrir una petició al SAU demanant l'obertura de Firewalls dels seus entorns a la màquina:

- 10.1.116.30, Port 443/TCP

L'accés al repositori de codi haurà d'efectuar-se via domini (git.intranet.gencat.cat), no via IP.

#### 2.1.2 Lliurament de codi font


Només es permet la pujada de codi font al repositori de codi del GitLab del SIC. No es poden incloure binaris de llibreries ni d'altres mòduls ni executables (JAR, WAR, EAR, SO, DLL, EXE...). A tal efecte, s'ha habilitat un sistema de Gestió de Binaris substitutori del SVN del SIC. El detall del funcionament d'aquest nou sistema resta explicat al Manual d'Usuari de SIC, sota l'apartat "4. Gestió de binaris".

#### 2.1.3 Estructura de projectes i repositoris a GITLAB

Al GitLab a cada repositori Git se'l denomina projecte. Un projecte albergarà un component de codi susceptible de ser versionat de forma independent. És a dir, pot consistir en una llibreria, un microservei, un mòdul o un programa que no té fragments independents.

GitLab també disposa de l'element anomenat Grup. Un grup pot albergar un conjunt de projectes relacionats amb alguna característica concreta. En el nostre cas, els grups de GitLab es correspondran als codis d'aplicació del CTTI i, per tant, tots els projectes estaran agrupats pel seu grup d'aplicació.

S'aconsella pujar el codi font de les aplicacions amb una determinada estructura. Sobretot en cas de tenir la intenció de fer servir també la integració amb les tasques d'anàlisi de codi font, construcció d'artefactes i desplegaments automatitzats que ofereix la Plataforma d'Integració Contínua (Jenkins) del SIC.

	MANUAL D'INTEGRACIÓ SIC	0192
	SIC Manual Integracio.docx	
	N. versió: 3.0.2	Pàg. 5 / 25

## 2.1.4 Estructura interna de projectes

Hi ha una sèrie de requeriments i de bones pràctiques definides per estructurar els projectes al SIC. Alguns d'aquests requeriments i d'aquestes bones pràctiques són d'aplicació per a tots els projectes i d'altres són aplicables en funció de la tecnologia.

### 2.1.4.1 Tots els projectes

Per a que el SIC pugui generar els jobs correctament, caldrà disposar de la carpeta `/sic/` al primer nivell de la carpeta de codi del projecte. Dins d'aquesta carpeta, cal crear l'arxiu `sic.yml` que albergarà la versió del programari per a que els jobs de Jenkins generin els tags corresponents.

Exemple de contingut de l'arxiu `sic.yml`:

```
version: 1.1.0
```


### 2.1.4.2 JEE

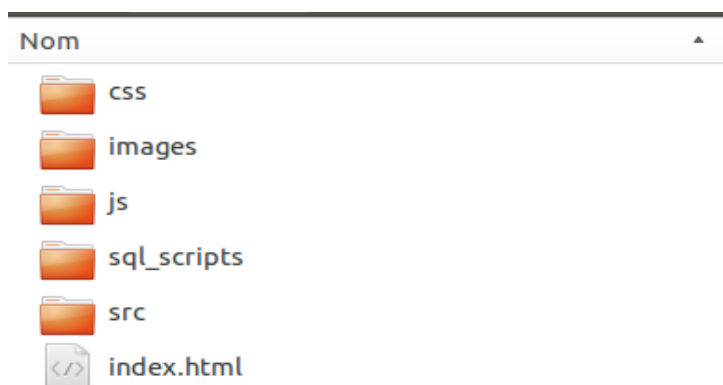
Per aplicacions JEE mavenitzades s'hauria de seguir la :

<code>src/main/java</code>	Application/Library sources
<code>src/main/resources</code>	Application/Library resources
<code>src/main/filters</code>	Resource filter files
<code>src/main/config</code>	Configuration files
<code>src/main/scripts</code>	Application/Library scripts
<code>src/main/webapp</code>	Web application sources
<code>src/test/java</code>	Test sources
<code>src/test/resources</code>	Test resources
<code>src/test/filters</code>	Test resource filter files
<code>src/it</code>	Integration Tests (primarily for plugins)
<code>src/assembly</code>	Assembly descriptors
<code>src/site</code>	Site
<code>LICENSE.txt</code>	Project's license
<code>NOTICE.txt</code>	Notices and attributions required by libraries that the project depends on
<code>README.txt</code>	Project's readme

### 2.1.4.3 PHP

Si bé no representa cap impediment perquè una aplicació PHP estigui correctament desenvolupada, és recomanable seguir un estàndard mínim d'estructuració de carpetes on desar els diferents tipus de components necessaris per a l'aplicació, tal i com es mostra a continuació:

	MANUAL D'INTEGRACIÓ SIC	0192
	SIC Manual Integracio.docx	
	N. versió: 3.0.2	Pàg. 6 / 25



### Il·lustració 3.1.3 1

El contingut que emmagatzema cada carpeta queda ben identificat pel nom de cada carpeta, segons el que es detalla a continuació.


- **css:** carpeta on es guardarà tots els fulls d'estils propis de l'aplicació.
- **images:** imatges que utilitza l'aplicació.
- **js:** guardarem tot els arxius amb codi JavaScript de l'aplicació, ja siguin propis o llibreries de tercers (jQuery, Ext.js, Angular, ...). En les llibreries de tercers que poguessin generar components de interfície d'usuari (UI), és possible també emmagatzemar fitxers d'estils i imatges que poguéss utilitzar aquestes llibreries.
- **sql\_scripts:** tots els fitxers necessaris per generar la base de dades que faci servir l'aplicació, en cas de fer-ho.
- **src:** tot el codi PHP de l'aplicació.
- **index.html** o altres: punt d'entrada de l'aplicació.

Com ja s'ha comentat aquesta és una proposta d'estructuració de carpetes, però això no exclou que l'aplicació pugui tenir una altra estructura, com podria donar-se el cas si aquesta aplicació està desenvolupada mitjançant un framework PHP (Symfony o similars).

En qualsevol cas, en el repositori SVN del SIC s'ha de fer entrega de tot el codi font necessari per poder desplegar l'aplicació als diferents entorns. Al SIC no està establert cap altre mètode d'obtenció de codi font, com podria ser Composer o CDN (Content Delivery Network).

#### 2.1.4.4 .NET

A .NET no s'especifica cap estructura tot i que es recomana una carpeta per a cada mòdul del projecte, una carpeta per les llibreries(lib), una carpeta per a les llibreries que s'han d'obtenir del

	MANUAL D'INTEGRACIÓ SIC	0192
	SIC Manual Integracio.docx	
	N. versió: 3.0.2	Pàg. 7 / 25

NuGet(packages) i el fitxer .sln a l'arrel.

Modul1/	Codi del mòdul
Modul2/	Codi del mòdul
ModulN/	Codi del mòdul
Lib/	Libreries específiques de l'aplicació
Packages/	Carpeta on es descarreguen les dependències del NuGet
Fitxer.sln	Fitxer per a realitzar la compilació

Com ja s'ha comentat aquesta és una proposta d'estructuració de carpetes.

### 2.1.4.5 Buildpacks

Depèn del tipus de buildpacks:

- En cas de ser Java seguir les indicacions del punt 2.1.4.2.
- En cas de ser PHP seguir les indicacions del punt 2.1.4.3
- Pels altres tipus de buildpacks no s'especifica cap estructura.

## 2.2 Procediments

### 2.2.1 Procés d'alta d'aplicacions


#### 2.2.1.1 Grups

Tots els codis d'aplicació que tenen aplicacions en servei susceptibles de repositar codi al SIC estan pre-creats com a Grups al GitLab.

Per als codis d'aplicació que únicament tinguin aplicacions en fase de projecte (codis d'aplicació nous), els grups encara no estaran creats a l'espai oficial del SIC. Els usuaris de lot d'aplicacions podran crear els seus propis grups al seu espai personal. Posteriorment, a la reunió de Fase 0, el SIC mourà els projectes i grups desitjats a l'espai oficial.

#### 2.2.1.2 Projectes

Els projectes poden ser creats per part del Release Managers (RML) de lot i àmbit. Els RMLs podran crear projectes als grups oficials corresponents als codis d'aplicació del seu lot i àmbit.

	MANUAL D'INTEGRACIÓ SIC	0192
	SIC Manual Integracio.docx	
	N. versió: 3.0.2	Pàg. 8 / 25

### 2.2.1.3 Assignació d'usuaris a projectes

#### 2.2.1.3.1 Usuaris de lot d'aplicacions

Hi ha dos tipus d'usuaris de lot d'aplicacions: els Release Managers (que tenen el rol de Màster al GitLab) i la resta d'usuaris del lot (que poden tenir el rol Developer).

Cada nit s'executa un procés que actualitza les assignacions dels Release Managers, afegint-los com a Masters de tots els grups als quals tenen accés segons el seu lot i àmbit. Com que disposen del rol de Master a Gitlab, poden assignar i treure permisos a la resta de companys assignant-los un dels següents rols:

- Developer: Si únicament han de tenir permís de lectura/escriptura
- Master: Si, a més de permís de lectura/escriptura, han de gestionar permisos.

El procés de regularització d'usuaris nocturn, promourà tots els Masters de projecte a Masters de grup i els afegirà com a Release Managers del lots i àmbits als quals corresponen els projectes als que estan assignats amb aquest rol. És a dir, assignar un company com a Màster de projecte, provocarà que al dia següent s'hagi promogut a Release Manager.

Es recomana màxima prudència en l'assignació de permisos, ja que -al tractar-se d'un SCM de versions lliurables i no de desenvolupament- convé que únicament el personal que gestioni les releases del programari hi tingui accés.

#### 2.2.1.3.2 Usuaris de CPD i Lloc de Treball

Els usuaris de CPD i de Lloc de Treball s'assignen cada nit als grups corresponents de forma automàtica.


## 2.2.2 Procediments operatius

### 2.2.2.1 Ús del repositori de codi font centralitzat

Aquest repositori **no és un entorn de desenvolupament**, el que vol dir que no s'ha de pujar codi que no s'hagi validat prèviament en un entorn local o de desenvolupament. Per tant, un projecte de desenvolupament tindrà configurats **dos repositoris** de codi font:

- el **propi de l'aplicació** o del proveïdor a on es guardi el **codi no definitiu** amb els canvis no consolidats i validats. És el que anomenem repositori local o d'ús diari.
- el **repositori del SIC** a on només es distribuirà el **codi font validat en entorns previs** que es vulgui distribuir als entorns dels serveis centrals.



	MANUAL D'INTEGRACIÓ SIC	0192
	SIC Manual Integracio.docx	
	N. versió: 3.0.2	Pàg. 9 / 25

La major part de l'equip de desenvolupament treballarà amb el repositori local. Dins de l'equip hi haurà una persona (o grup de persones) encarregat de consolidar el codi de tothom i lliurar-lo. És aquesta figura anomenada Release Manager.

### 3 SIC: Tasques d'anàlisi de codi font, construcció d'artefactes i desplegaments automatitzats

#### 3.1 Requisits d'integració

##### 3.1.1 Requisits comuns

###### 3.1.1.1 Connectivitat

Per poder accedir a les eines des d'on s'executen les tasques d'anàlisi de codi font, empaquetat d'artefactes i desplegaments automatitzats via VPN cal assegurar que es disposa de connectivitat a la màquina **hudson.intranet.gencat.cat** (10.48.143.21), port 443/TCP.

Si l'aplicació no té visibilitat a aquesta màquina, caldrà obrir una petició al SAU demanant l'obertura de Firewalls dels seus entorns a la màquina :

- **10.1.112.26**, Port 443

L'accés a la Plataforma d'Integració Contínua (Jenkins) haurà d'efectuar-se via domini (**hudson.intranet.gencat.cat**), no via IP.

##### 3.1.2 Requisits per a aplicacions JEE

###### 3.1.2.1 Versions JDK suportades


Les versions de JDK amb les que pot treballar actualment el SIC són:

- JDK 1.6
- JDK 1.7
- JDK 1.8

NOTA: JDK 1.5 ha deixat de ser suportada. Es dona suport estès a les aplicacions que ja estaven integrades al SIC amb aquesta versió, però no a noves.

###### 3.1.2.2 Versions de Maven suportades

Les versions de Maven amb les pot treballar actualment el SIC són:

	MANUAL D'INTEGRACIÓ SIC	0192
	SIC Manual Integracio.docx	
	N. versió: 3.0.2	Pàg. 10 / 25

- Maven 2.0.9
- Maven 2.0.10
- Maven 2.2.1
- Maven 3.2.2
- Maven 3.3.9

### 3.1.2.3 Construcció amb Maven 2/3

L'aplicació ha de fer servir Maven 2 o Maven 3 com a eina de construcció dels executables a partir del codi font. L'estructura de l'arbre de directoris del projecte ha de seguir preferentment les convencions de Maven i la indicada en el Llibre Normatiu JEE.

### 3.1.2.4 Proporcionar els processos de construcció

L'aplicació ha de proporcionar els processos necessaris (Maven goals) per a construir les parts estàtica i dinàmica de l'aplicació, segons està descrit en el Llibre Normatiu JEE.

Per tal de poder facilitar la generació d'aquests processos de construcció es proporcionarà un pom.xml d'exemple. Aquest pom.xml d'exemple està basat en el projecte d'exemple plantilla-canigo. Per més informació sobre els elements que integren el pom.xml necessari podeu consultar .

Cal tenir en compte que el codi que entra al sistema té per objectiu ser desplegat en els Serveis TIC Centrals, per tant cal que compleixi una sèrie de normes, per exemple les referents a l'estructura de directoris descrites en el punt "Directoris del Contingut Estàtic i Dinàmic". A l'hora de desplegar s'espera que:


- Les classes es trobin ubicades a la ruta `"/src/main/java"`
- Els recursos que hagin de ser inclosos en el classpath a `"/src/main/resources"`
- Tot el contingut del modul web `"/src/main/webapp"`

L'aplicació plantilla de Canigó es pot fer servir com a exemple i punt de partida també en aquest aspecte.

### 3.1.2.5 Construcció independent de la plataforma

Els processos de construcció de l'aplicació (Maven goals) han de ser independents de la màquina i plataforma on s'executen. Per exemple, tota referència a fitxers s'ha de fer amb paths relatius.

En particular, aquests processos han de poder executar-se sense canvi tant en els entorns de desenvolupament com en les màquines del SIC.

	MANUAL D'INTEGRACIÓ SIC	0192
	SIC Manual Integracio.docx	
	N. versió: 3.0.2	Pàg. 11 / 25

### 3.1.2.6 Execució de tests

No s'ha de barrejar el codi del programa amb el codi que serveix per provar, ni tampoc els recursos, per tant cal ubicar els tests i totes les classes que aquests necessitin en el directori “src/test/java” i els recursos utilitzats exclusivament per els tests s'ubicaran “src/test/resources”. Aquestes ubicacions no seran desplegades.

És possible que per tal d'executar els tests siguin necessaris una sèrie de dependències extra. És molt important que aquestes dependències siguin marcades com a tal, així evitem que siguin incloses en el desplegament. Per fer-ho caldrà indicar l'scope de la dependència Maven com a test.

Cal fer tests que siguin independents de les circumstàncies en que s'executin, en l'entorn on s'executaran no es tindrà accés a base de dades ni a la major part de recursos, però del que es tracta és de provar codi, no pas provar cap accés a recursos ni provar el gestor de base de dades ni la velocitat de la xarxa. Per tant és recomanable l'ús d'interfícies i Objectes Mock que les implementin automàticament en la capa DAO i de serveix i centrar-se en provar bé la capa de negoci o bé implementar objectes “dummy”.

### 3.1.2.7 Versionat

El número de versió inclòs en el POM del projecte ha de coincidir amb l'indicador de versió inclòs en l'arxiu /sic/sic.yml.


```
<version>1.0.1</version>
```

*figura 4.1.2.9.-1. Propietat de versió en el pom.xml*

### 3.1.2.8 Dependències localitzades en repositoris públics

Totes les dependències de l'aplicació han d'estar accessibles en els repositoris públics de Maven 2/3 configurats al Nexus del SIC. Es pot validar la existència o no de la dependència accedint a la següent URL:

Qualsevol llibreria o versió que no estigui en un repositori públic haurà de passar per un procés d'aprovació tant per part del projecte com del suport de SIC. En particular, no es poden fer referències a recursos del proveïdor.

	MANUAL D'INTEGRACIÓ SIC	0192
	SIC Manual Integracio.docx	
	N. versió: 3.0.2	Pàg. 12 / 25

### 3.1.2.9 No incloure llibreries que proporciona el servidor

Les llibreries tals com Drivers JDBC o implementacions de les APIs J2EE (Servlet, JSP,...) no han de quedar mai incloses dins del mòdul desplegable, ja que aquestes llibreries són proporcionades pel servidor d'aplicacions.

### 3.1.2.10 Funcionament de l'aplicació en format empaquetat

L'aplicació ha de ser capaç de funcionar correctament sobre el servidor d'aplicacions escollit en format empaquetat. Aquest fa referència als tipus més comuns: **WAR**, **EAR** o **JAR**.

Aquest requisit implica que el primer desplegament que es faci de l'aplicació en el servidor d'aplicacions escollit ha de ser en aquest mode, empaquetat. Si no es realitza d'aquesta forma, els re-desplegaments efectuats des del SIC sobre l'entorn d'Integració fallaran, ja que el SIC utilitza el mode de re-desplegament d'aplicacions en format empaquetat per efectuar els re-deploys.

### 3.1.2.11 L'accés a fitxers que estan dins del mòdul desplegable ha de fer-se via `getResource()`

Si una aplicació necessita obrir un fitxer dels continguts dins del mòdul desplegable, ho ha de fer com a recurs (`Class.getResourceAsStream`). No pot utilitzar les APIs Java d'accés a fitxers (`java.io.File`). Això és degut a que, en els entorns d'integració, els desplegaments automatitzats es fan com a arxius empaquetats (WAR, EAR) i no com a directoris desempaquetats. Els fitxers continguts dins d'un mòdul empaquetat només són accessibles com a recurs i en mode de només lectura.

## 3.1.3 Requisits per a aplicacions .NET

### 3.1.3.1 Versions MS Build suportades


Les versions de Ms Build amb les que pot treballar actualment el SIC són:

- MS Build 3.5
- MS Build 4.0
- MS Build 4.5.2

### 3.1.3.2 Versions de NuGet suportades

La versió de NuGet amb la que es pot treballar actualment al SIC és:

- 2.8.x

	MANUAL D'INTEGRACIÓ SIC	0192
	SIC Manual Integracio.docx	
	N. versió: 3.0.2	Pàg. 13 / 25

### 3.1.4 Requisits per a aplicacions Node JS

#### 3.1.4.1 Versions NodeJs suportades

Les versions de Ms Build amb les que pot treballar actualment el SIC són:

- NodeJS 0.12.13
- NodeJS 4.4.3
- NodeJS 5.10.1

## 3.2 Requisits per al desplegament

### 3.2.1 Desplegaments en servidors físics (IAAS)

#### 3.2.1.1 Desplegament de binaris no generats a Jenkins

Per poder demanar el desplegament d'artefactes binaris no desplegats a Jenkins es necessari haver-hi pujat prèviament, tant els artefactes com els manuals de desplegament, a l'espai de binaris del SIC (consultar el Manual d'Usuari per a més informació).

#### 3.2.1.2 JEE


Si es volen fer servir les tasques de desplegaments automatitzat des de SIC, cal que l'aplicació pugui ser desplegada sobre un dels següents servidors d'aplicacions:

- Weblogic (9.2, 10.3.x, 11g, 12c)
- Websphere 6.1, 8.5
- Tomcat (5.5, 6, 7, 8, 8.5)

Les tasques d'execució de desplegament automatitzat fan un re-desplegament de l'aplicació i no pas un desplegament. Per tant, cal que l'aplicació ja es trobi desplegada (en format empaquetat WAR/EAR). **La petició per fer aquest primer desplegament de l'aplicació corre a càrrec dels proveïdors de l'aplicació i en ella s'ha d'indicar a SAU de forma explícita que l'aplicació ha de desplegar-se en format empaquetat (WAR/EAR).**

També s'ha d'indicar que l'aplicació en qüestió estarà integrada en la plataforma d'Integració Continua del CTTI, el SIC. Aquest punt és molt important per que en carència d'aquesta especificació alhora de fer la petició, SAU ordenarà que l'aplicació es desplegui de forma explotada, fent que els successius intents de re-desplegament des del SIC fallin.

Destacar que en cas que els tècnics que administren els servidors d'aplicacions facin qualsevol canvi en la referència al artefacte a desplegar o canvi en el nom de l'aplicació dins el servidor d'aplicacions, ha de ser notificat a l'equip del SIC, ja que en cas contrari el job de desplegament deixarà de

	MANUAL D'INTEGRACIÓ SIC	0192
	SIC Manual Integracio.docx	
	N. versió: 3.0.2	Pàg. 14 / 25

funcionar.

### 3.2.1.3 .NET

El servei del SIC inclou la solució MS Web Deploy amb la que es pot desplegar en servidors IIS a partir d'una compilació NuGet / MSBuild. Dins de l'estructura de carpetes del projecte, es genera un empaquetat en format zip que pot ser utilitzat per aquesta eina per realitzar un desplegament de forma transparent.

La versió de MS Web Deploy amb la que pot treballar actualment el SIC és:

- MS Web Deploy 3.6

## 3.2.2 Desplegaments al CLOUD

### 3.2.2.1 Comuns

Si es vol desplegar la aplicació al Cloud, cal que l'aplicació proporcioni els fitxers **Dockerfile.ci** necessaris per crear les imatges que l'aplicació farà servir als contenidors que més endavant es desplegaran.

L'equip del SIC configurarà el job de BLD per la generació i publicació de les imatges definides.

### 3.2.2.2 Bluemix


Per desplegaments a Bluemix, la aplicació ha de proporcionar el fitxer **docker-compose-ci.yml** amb el guió de contenidors a generar.

El servei del SIC crearà un job de desplegament (un a PRE i altre a PRO), i el farà servir per crear l'ecosistema necessari pel funcionament de l'aplicació, és a dir, tot el necessari com BBDD, servidor d'aplicacions, etc .....

### 3.2.2.3 OpenShift


Per desplegaments a OpenShift, la aplicació ha de proporcionar el fitxer **docker-template-ci.yaml** amb el guió de contenidors a generar.

El servei del SIC crearà un job de desplegament (un a PRE i altre a PRO), i el farà servir per crear l'ecosistema necessari pel funcionament de l'aplicació.

	MANUAL D'INTEGRACIÓ SIC	0192
	SIC Manual Integracio.docx	
	N. versió: 3.0.2	Pàg. 15 / 25

### 3.2.3 Execució d'scripts contra BBDD durant els desplegaments

En cas que el desplegament de l'aplicació hagin d'estar lligats a canvis en BBDD de l'aplicació, els jobs Pipeline podran executar aquests scripts just abans realitzar el desplegament a cada entorn. Per fer-ho possible cal respectar les indicacions que es proporcionen al Manual d'Usuari SIC abans fer les pujades del codi al Gitlab.

	MANUAL D'INTEGRACIÓ SIC	0192
	SIC Manual Integracio.docx	
	N. versió: 3.0.2	Pàg. 16 / 25

### 3.3 Procediments

#### 3.3.1 Procés d'alta

El procés d'alta d'una aplicació al SIC per poder fer ús de les eines d'anàlisi de codi font, construcció d'artefactes i desplegaments automatitzats consta de les següents fases:

- **Alta de l'aplicació a les eines d'anàlisi de codi font, construcció d'artefactes i desplegaments automatitzats**

##### 3.3.1.1 Alta de l'aplicació a les eines d'anàlisi de codi font, construcció d'artefactes i desplegaments automatitzats.

Per a noves aplicacions no caldrà realitzar la petició d'alta de Jobs, doncs a partir de la Fase 0 de l'aplicació s'iniciarà tot el procés de recollida de dades i configuració de jobs. La informació complementària de l'aplicació que resti per recollir, es demanarà via bústia als proveïdors / CPD.

Per a integrar la resta d'aplicacions ja existents amb els jobs Jenkins de SIC, 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.


A partir d'aquesta petició, s'iniciarà el procés de recollida de la resta de dades necessàries per a la configuració dels jobs

#### 3.3.2 Procediments operatius

Tot tipus d'aplicació que es vulgui integrar amb la Plataforma d'Integració Contínua, comptarà amb un job de tipus pipeline. Aquest job comptarà amb una sèrie fases (stages) mínimes:

- Build: construcció o empaquetat d'artefactes.
- Desplegament: Desplegament a un entorn, pot ser automàtic o semiautomàtic.
  - Desplegament semiautomàtic
  - Desplegament automàtic



	MANUAL D'INTEGRACIÓ SIC	0192
	SIC Manual Integracio.docx	
	N. versió: 3.0.2	Pàg. 17 / 25

### 3.3.2.1 Construcció/empaquetat d'artefactes

#### 3.3.2.1.1 Aplicacions JEE

Per a les aplicacions JEE, la fase de Build realitzarà les següents accions:

- Anàlisi del codi font amb CAST i Fortify.
- Empaquetat dels artefactes segons la definició del pom.xml de l'aplicació.
- Arxivar els artefactes generats.

Adicionalment, si l'aplicació es desplega al Cloud, la fase de Build realitzarà les següents accions:

- Anàlisi del fitxer Dockerfile que es fa servir que la generació de la imatge.
- Generació de la imatge a partir del Dockerfile especificat.
- Anàlisi de les vulnerabilitats de la imatge generada mitjançant Clair.
- Publicació de la imatge generada al registre del cloud que pertorqui (Bluemix o OpenShift).

#### 3.3.2.1.2 Aplicacions PHP

Per a les aplicacions PHP, la fase de Build realitzarà les següents accions:

- Empaquetat del codi font a un zip. En cas que l'aplicació distingeixi entre codi estàtic i dinàmic, realitzarà l'empaquetat en 2 artefactes. El contingut de cada paquet ha de ser indicat al formulari d'alta (si no s'indica res, es donarà per fet que no tot s'empaquetarà junt i desplegarà a un únic Apache).
- Arxivar el codi empaquetat.

Adicionalment, si l'aplicació es desplega al Cloud, la fase de Build realitzarà les següents accions:

- Anàlisi del fitxer Dockerfile que es fa servir que la generació de la imatge.
- Generació de la imatge a partir del Dockerfile especificat.
- Anàlisi de les vulnerabilitats de la imatge generada mitjançant Clair.
- Publicació de la imatge generada al registre del cloud que pertorqui (Bluemix o OpenShift).


#### 3.3.2.1.3 Aplicacions .NET

Per a les aplicacions .NET, la fase de Build realitzarà les següents accions:

- Empaquetat dels artefactes segons la definició al fitxer SLN de l'aplicació.
- Arxivar els artefactes generats.

#### 3.3.2.1.4 Aplicacions Buildpacks

Per a les aplicacions Web estàtiques, la fase de Build realitzarà les següents accions:

	MANUAL D'INTEGRACIÓ SIC	0192
	SIC Manual Integracio.docx	
	N. versió: 3.0.2	Pàg. 18 / 25

- Generació manifest\_PRE.yml i manifest\_PRO.yml
- Empaquetat del codi en un Zip per arxivar.
- Arxivar els artefactes generats.

#### 3.3.2.1.5 Aplicacions Node JS

Per a les aplicacions Node JS, la fase de Build realitzarà les següents accions:

- Comprovació del codi de l'aplicació.
- Instal·lació de les llibreries necessàries

Adicionalment, si l'aplicació es desplega al Cloud, la fase de Build realitzarà les següents accions:

- Anàlisi del fitxer Dockerfile que es fa servir que la generació de la imatge.
- Generació de la imatge a partir del Dockerfile especificat.
- Anàlisi de les vulnerabilitats de la imatge generada mitjançant Clair.
- Publicació de la imatge generada al registre del cloud que pertoqui (Bluemix o OpenShift).

### 3.3.2.2 Desplegaments automatitzats

La plataforma d'Integració Contínua no s'ha de considerar com un entorn de proves com pot ser un entorn de desenvolupament. Això vol dir que no s'ha de fer un ús abusiu sobre les execucions de tasques de desplegament automatitzat.

#### 3.3.2.2.1 Aplicacions JEE

Les tasques de desplegament automatitzat fan re-desplegaments de l'aplicació basant-se en el fet de que aquesta aplicació es desplega en format empaquetat. Si no es complís aquesta premissa els re-desplegament sobre l'entorn d'Integració fallarien.


Les tasques de desplegament automatitzat necessiten que l'aplicació ja es trobi desplegada per tal de poder fer el re-desplegament, altrament l'execució de la tasca fallarà.

#### 3.3.2.2.2 Aplicacions PHP

A diferència del que s'estableix per a JEE, en el cas de PHP no aplica el procés de re-desplegament indicat anteriorment. La tasca de Jenkins s'encarrega d'enviar-la a la màquina/es que correspongui, esborrant prèviament si hagués hagut un desplegament anterior.

##### 3.3.2.2.2.1 Aplicacions Buildpacks

Les aplicacions web estàtiques despleguen automàticament a un cloud públic.

	MANUAL D'INTEGRACIÓ SIC	0192
	SIC Manual Integracio.docx	
	N. versió: 3.0.2	Pàg. 19 / 25

### 3.3.2.2.3 Desplegaments al CLOUD

Les aplicacions que es despleguen al Cloud, fan servir imatges generades i publicades a la fase de Build executada anteriorment.

Aquesta fase generarà al Cloud escollit, tot l'ecosistema de contenidors definit al fitxer descriptor especificat.

### 3.3.2.3 Desplegaments semiautomàtics

Aquest tipus de desplegaments generen una petició SAU-Remedy per tal que CPD realitzi el desplegament.

### 3.3.2.4 Projectes llibreria (només JEE, .NET, npm i bower)

El SIC ofereix la possibilitat de que una aplicació desenvolupi, construeixi i utilitzi les seves pròpies llibreries.

L'execució de la tasca creada per a aquest projecte instal·la la llibreria al repositori Nexus. Les aplicacions que vulguin utilitzar aquesta llibreria l'hauran d'importar amb el nom i versió correctes en els seus arxius de configuració corresponents (pom.xml, etc.).


### 3.3.2.5 Instal·lació de llibreria no disponible públicament

Per a la instal·lació al repositori local del Hudson de llibreries de tercers cal obrir una petició de suport funcional de l'aplicació 'FRAMEWORK SIC' a Remedy. Cal informar a la descripció de la petició les següents dades:

- Nom i versió de la llibreria
- URL on obtenir la llibreria (o adjuntar-la a la pròpia petició Remedy)
- Característiques de la llibreria
  - És una llibreria que no està disponible en els repositoris públics
  - Llibreria disponible públicament, però és una versió concreta que aporta una funcionalitat necessària que no es troba en les versions disponibles públicament
- Funcionalitat de la llibreria
- Raons per la utilització de la llibreria

### 3.3.2.6 Modificació de dades d'aplicació


Per a les actualitzacions de llibreries aplica la mateixa mecànica que per a primera la instal·lació, però

	MANUAL D'INTEGRACIÓ SIC	0192
	SIC Manual Integracio.docx	
	N. versió: 3.0.2	Pàg. 20 / 25

indicant que es tracta d'una actualització sobre una ja existent.

## 4 Consultes i incidències

Les consultes i incidències relacionades amb el Servei d'Integració Continua cal demanar-les/informar-les mitjançant les corresponents peticions a Remedy. Per a més informació veure el document que es troba penjat al .

	MANUAL D'INTEGRACIÓ SIC	0192
	SIC Manual Integracio.docx	
	N. versió: 3.0.2	Pàg. 21 / 25

## 5 Annex A


En aplicacions JEE mavenitzades el pom.xml de l'aplicació ha de disposar d'una sèrie de elements de configuració mínims per poder funcionar dintre del SIC. Aquests elements es descriuen a continuació.

```
<groupId>plantilla-canigo</groupId>
<artifactId>plantilla-canigo</artifactId>
<version>2.3.9</version>
<packaging>war</packaging>
<name>plantilla-canigo</name>
<description>Plantilla Canigó sense exemples</description>
```

figura A – 0. Configuració de propietats generals paràmetres groupId, artifactId, version, name, package i description

```
<sourceDirectory>${basedir}/src/main/java</sourceDirectory>
<testSourceDirectory>${basedir}/src/test/java</testSourceDirectory>
<outputDirectory>${basedir}/target/classes</outputDirectory>
<resources>
  <resource>
    <directory>${basedir}/src/main/resources</directory>
  </resource>
</resources>
<testResources>
  <testResource>
    <directory>${basedir}/src/test/resources</directory>
  </testResource>
  <testResource>
    <directory>${basedir}/src/main/resources</directory>
  </testResource>
  <testResource>
    <targetPath>webapp</targetPath>
    <directory>${basedir}/src/main/webapp</directory>
  </testResource>
</testResources>
```

figura A - 1. Configuració de la estructura interna del projecte

	MANUAL D'INTEGRACIÓ SIC	0192
	SIC Manual Integracio.docx	
	N. versió: 3.0.2	Pàg. 22 / 25

```

<plugin>
  <artifactId>maven-compiler-plugin</artifactId>
  <configuration>
    <source>1.5</source> <!-- versió de JDK -->
    <target>1.5</target>
  </configuration>
</plugin>

```

*figura A - 2. Configuració de la versió del Jdk utilitzat*

```

<plugin>
  <artifactId>maven-war-plugin</artifactId>
  <configuration>
    <warSourceDirectory>
      ${basedir}/src/main/webapp
    </warSourceDirectory>
  </configuration>
</plugin>

```

*figura A - 4. Configuració del directori on anirà el war generat*


```

<plugin>
  <artifactId>maven-assembly-plugin</artifactId>
  <configuration>
    <finalName>${artifactId}</finalName>
    <descriptors>
      <descriptor>src/main/resources/assemble/dynamic.xml</descriptor>
      <descriptor>src/main/resources/assemble/static.xml</descriptor>
      <descriptor>src/main/resources/assemble/audit.xml</descriptor>
    </descriptors>
  </configuration>
  <executions>
    <execution>
      <id>assembly-app</id>
      <phase>package</phase>
      <goals>
        <goal>assembly</goal>
      </goals>
    </execution>
  </executions>
</plugin>

```

*figura A - 5. Configuració de la construcció dels artefactes  
corresponents a la part dinàmica, part estàtica i fitxer d'auditoria de l'aplicació.*

Molt important és la configuració del maven-assembly-plugin expressada en la Figura A.5. Aquesta

	MANUAL D'INTEGRACIÓ SIC	0192
	SIC Manual Integracio.docx	
	N. versió: 3.0.2	Pàg. 23 / 25

configuració indicarà a aquest plugin de maven com generar els artefactes corresponents a la part dinàmica, part estàtica i fitxer d'auditoria de l'aplicació.


La configuració està dividida en tres arxius de configuració que s'hauran de declarar en el path src/main/resources/assemble/. Aquests tres arxius són dynamic.xml, static.xml i audit.xml

```

<assembly>
  <id>dynamic</id>
  <formats><format>war</format></formats>
  <includeBaseDirectory>false</includeBaseDirectory>
  <fileSets>
    <fileSet>
      <directory>target/${artifactId}-${version}</directory>
      <outputDirectory></outputDirectory>
      <includes>
        <include>WEB-INF/**/*.*</include>
        <include>**/*.jsp</include>
      </includes>
      <excludes>
        <exclude>**/*.html</exclude>
        <exclude>**/*.htm</exclude>
        <exclude>css/**/*.*</exclude>
        <exclude>images/**/*.*</exclude>
        <exclude>scripts/**/*.*</exclude>
        <exclude>config/**/*.*</exclude>
      </excludes>
    </fileSet>
  </fileSets>
</assembly>

```

figura A - 6. Arxiu exemple de configuració de la generació de la part dinàmica

	MANUAL D'INTEGRACIÓ SIC	0192
	SIC Manual Integracio.docx	
	N. versió: 3.0.2	Pàg. 24 / 25


```

<assembly>
  <id>static</id>
  <formats>
    <format>zip</format>
  </formats>
  <includeBaseDirectory>>false</includeBaseDirectory>
  <fileSets>
    <fileSet>
      <directory>target/${artifactId}-${version}</directory>
      <outputDirectory>.</outputDirectory>
      <includes>
        <include>/**/*.html</include>
        <include>/**/*.htm</include>
        <include>css/**/*.css</include>
        <include>images/**/*.png</include>
        <include>scripts/**/*.js</include>
        <include>config/**/*.xml</include>
      </includes>
    </fileSet>
  </fileSets>
</assembly>

```

*figura A-7. Arxiu exemple de configuració de la generació de la part estàtica*



	MANUAL D'INTEGRACIÓ SIC	0192
	SIC Manual Integracio.docx	
	N. versió: 3.0.2	Pàg. 25 / 25

```

<assembly debug="true">
  <id>audit</id>
  <formats>
    <format>zip</format>
  </formats>
  <includeBaseDirectory>>false</includeBaseDirectory>
  <fileSets>
    <fileSet>
      <directory>src</directory>
      <outputDirectory>/src</outputDirectory>
      <includes>
        <include>**/*.*</include>
      </includes>
    </fileSet>
  </fileSets>
  <dependencySets>
    <dependencySet>
      <outputDirectory>/lib</outputDirectory>
      <includes>
        <include>*.jar:</include>
      </includes>
    </dependencySet>
  </dependencySets>
</assembly>

```

*figura A-8. Arxiu exemple de configuració de la generació del fitxer d'auditoria*