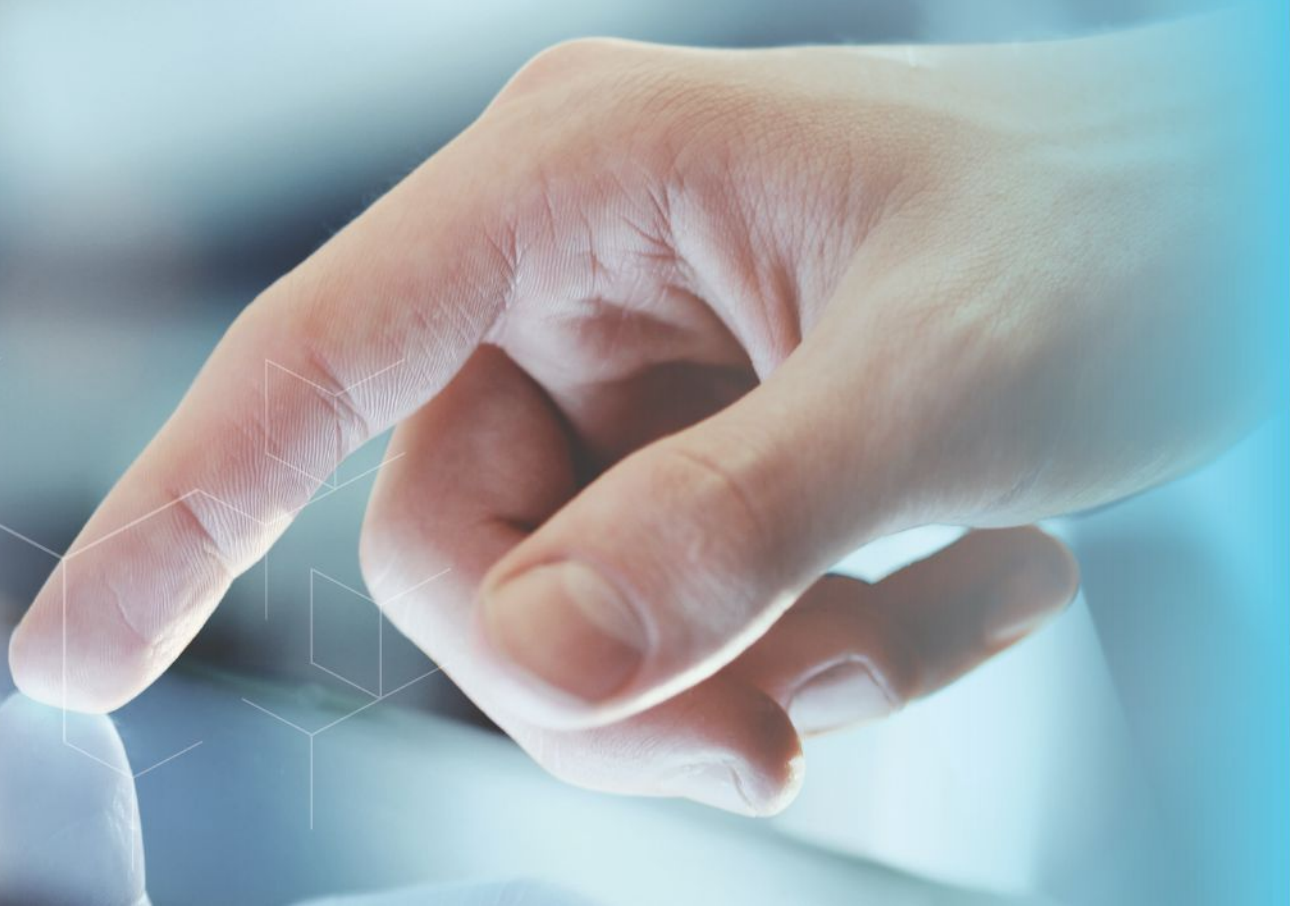




CITI

Value in real time





ESTÁNDARES

- ◆ Administración de versionado
- ◆ Calidad del código
- ◆ Performance
- ◆ Pruebas unitarias
- ◆ Estándares generales de proyecto
- ◆ Estandarización Logs
- ◆ Liberaciones
- ◆ Plataforma JIRA





BRANCHES BASE

UAT

*Este branch contiene estrictamente el código fuente de aplicativos y aplicaciones que se encuentran en el ambiente **UAT**, a este branch la escritura está **restringida** a únicamente el **coordinador del proyecto** ó **administrador de la configuración**.*

**(Según aplique, caso SVA)*





BRANCHES BASE

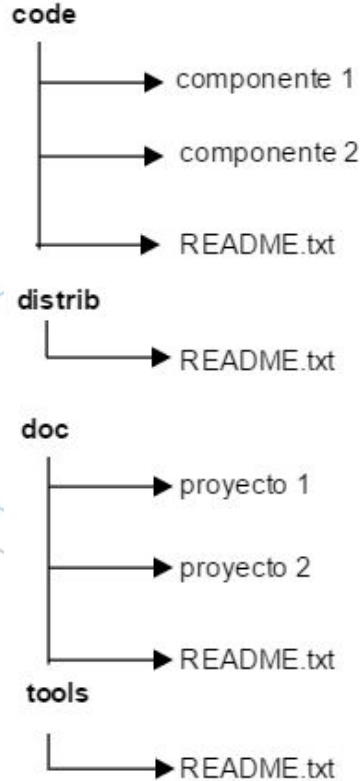
PRODUCCIÓN (master)

*Este branch (master) contiene estrictamente el código fuente de aplicativos y aplicaciones que se encuentran en el ambiente de **PRODUCCIÓN**, a este branch la escritura esta **restringida** a únicamente el **coordinador del proyecto** ó **administrador de la configuración**.*





ESTRUCTURA DEL GIT



README.MD

*Para cada repositorio se implementa el archivo **README.MD** con el fin de controlar las versiones, y las relevancias entre ellas, así como crear una guía para montar el ambiente de cada proyecto.*

1. *Nombre del proyecto*
2. *Versión*
3. *Descripción*
4. *Creación del ambiente*
5. *Componentes*
6. *Desarrolladores*
7. *Seguridad*
8. *Tecnologías*

<http://200.34.113.186:7990/projects/TLC/repos/portalsaqueteamigo>



NOMENCLATURA BRANCHES

Los branches creados deben ser nombrados en base a los siguientes:

1.- Seleccionar el tipo de proyecto o desarrollo

- **Bugfix** Identifica un branch donde se resuelve un bug durante un release
- **Feature** Identifica un branch donde se desarrolla una mejora para el proyecto
- **Release** Identifica un branch donde se desarrolla una nueva versión del proyecto
- **Custom** Identifica un branch que no cabe en las características anteriores

2.- Nombrar el branch con en base a los siguientes datos

- **Clave del proyecto**
- **Nombre corto del proyecto**

Ej: **Release/CIT-O-001-ConsumoDeInternet**



COMMIT MESSAGE

El **mensaje** de los commit realizados hacia los repositorios deberán contar con la siguiente estructura.

DEVOPS-99 bugfix: Se resuelve incidencia en core

Prefijos:

- **Feat:** Nueva característica
- **Fix:** Soluciona un bug
- **Docs:** Se realizan cambios en documentación
- **Style:** Se aplicaron cambios en formatos visuales
- **Test:** Se añadieron pruebas
- **Chore:** Actualización de tareas build, conf, etc.





MERGE A BRANCH BASE

*El encargado de realizar el merge a un branch base (Producción ó UAT) es el encargado de registrar la versión y las relevancias de versión en el archivo **README.MD***

- ***Incrementar el número de versión del componente(s) afectados***
- ***Incrementar el número de versión del proyecto***





CONVENCIONES DE CODIFICACIÓN

*El código fuente de los desarrollos es sometido a pruebas de calidad de código mediante la herramienta **SonarQube** la cual hace la verificación de cumplimiento de los estándares **CWE**.*

*Los desarrolladores deberán instalar en su IDE el plugin **SonarLint** el cual muestra en tiempo real las incidencias de convenciones de programación, reduciendo las incidencias que podrían encontrarse durante la revisión con la herramienta.*



PERFORMANCE TEST

JMeter

*Los desarrollos Web deberán contener pruebas de estrés, las cuales deberán estar en formato **JMX** y ubicadas bajo la siguiente ruta en el proyecto:*

`$PROJECT_HOME/src/test/jmeter/*.jmx`

El POM del proyecto se debe configurar de tal manera que al ejecutar el comando

`maven verify`





UNIT TEST

JMeter

Los desarrollos deberán contener pruebas de unitarias y de integración, las cuales deberán estar desarrolladas bajo las especificaciones de Junit y ubicadas bajo la siguiente ruta en el proyecto:

\$PROJECT_HOME/src/test/java

*El **POM.xml** del proyecto se debe configurar de tal manera que al ejecutar el comando*

maven test

Estas pruebas sean ejecutadas.



NUEVOS DESARROLLOS

Para nuevos desarrollos, se deben crear en base a las siguientes especificaciones:

- *Utilizar la herramienta Maven (En caso java), o equivalente para otros lenguajes*
- *Utilizar perfiles Maven para el empaquetado en diversos ambientes*
- *Parametrización de datos sensibles (Endpoints, puertos, usuarios, etc)*
- *Escritura de logs en base a los estándares aplicables para el proyecto*
- *Utilizar como repositorio Bitbucket (Git)*
- *Utilizar la siguiente nomenclatura para el empaquetamiento:*

NombreDelComponente-1.00.00.war



ESTANDARIZACIÓN DE LOGS

Con la finalidad de obtener estadísticas de las transacciones realizadas en los componentes desarrollados por CITI y de terceros, se estandariza la escritura de logs, de un estándar genérico, y desprendiéndose de ahí modificaciones específicas para cada proyecto, dependiendo de las necesidades particulares.

Archivos *.log a generar:

- **Trans.log:** Archivo donde se debe de imprimir solo 1 línea por transacción, es para dar un seguimiento general de cuál fue el resultado de la transacción.
- **Trace.log:** Archivo donde se pone el detalle de la transacción.
- **Legacy-trans.log:** Archivo donde se registran las llamadas a plataformas externas de la aplicación. En este log el registro de los tiempos de respuesta es lo más importante.



NOMENCLATURA DE LOGS

Los archivos **Log** generados, deberán almacenarse bajo la siguiente ruta:

\$PROJECT_HOME/logs/*.log

Deberán contar con las siguientes características:

- *Nombrado en base al siguiente patrón: **yyyy-MM-dd-HH.log***
- *Rotación por hora.*
- *Tiempo en milisegundos de respuesta.*



LEGACY TRANS

\$PROJECT_HOME/logs/legacy-trans.log

*Es requerido que en un nuevo archivo log llamado **legacy-trans.log**, en él se deberán de tener el registro de cada interacción con componentes externos, escribiendo él tiempo de respuesta en **milisegundos**.*

**Consultar estándares por proyecto*





TRANS LOG

`$PROJECT_HOME/logs/trans.log`

*Es requerido que el archivo **trans.log** cuente con al menos la siguiente información escrita en solo una línea por cada operación, separada por un carácter especial:*

- Fecha
- Hora Inicio
- Hora Final
- Tiempo De Duración
- Identificador de Transacción
- Nombre Del Servicio
- Resultado De La Operación
- Código De Respuesta
- Descripción Del Error

**Consultar estándares por proyecto*





TRACE LOG

`$PROJECT_HOME/logs/trace.log`

*Es requerido que el archivo **trace.log** detalle a nivel **INFO** el cuerpo de las transacciones, así como las rutas que sigue a través de los diversos componentes. En este log se requiere identificar con un **T1** para **request** y un **T2** para **response**.*

Ejemplo Request:

INFO - T1 Sending request to OpenApi - XXXXX 8115017970 getCustomerData

Ejemplo Response:

INFO - T2 200 ms Received response from OpenApi - XXXXX 8115017970 getCustomerData

**Consultar estándares por proyecto*



ERROR LOG

`$PROJECT_HOME/logs/error.log`

*Es requerido que en un nuevo archivo log llamado **error.log**, a nivel **ERROR** en él se deberán de tener el registro de los errores ocurridos en los diversos componentes del desarrollo. Mostrando la siguiente información:*

- **Fecha y Hora**
- **Nombre del error**
- **Descripción del error**
- **Método invocado**

**Consultar estándares por proyecto*





JIRA DE LIBERACIÓN

El jira de liberación debe contener las siguientes secciones

- ***Título***
- ***Descripción***
- ***Glosario de Términos***
- ***Instrucciones***
- ***Pruebas***

Las cuales dan al personal de ingeniería contexto del ambiente en el que se está instalando el aplicativo. Así como instrucciones de cómo proceder con la instalación y la validación de la misma.

JIRA Ejemplo



DESCRIPCIÓN

Descripción detallada de lo que consiste o corrige la liberación, incluir lo siguiente de manera obligatoria

Sección	Contenido
<i>Descripción</i>	Descripción general de las funcionalidades que contiene la instalación.
<i>Dependencias</i>	Especifica los jiras que se requiere instalar previamente para el correcto funcionamiento del aplicativo.
<i>Impactos</i>	Funcionalidades que son agregas y/o modificadas, si es que ya existe una instalación anterior.
<i>Funcionalidades Afectadas</i>	Descripción de las posibles afectaciones a funcionalidades que se puedan desprender de la instalación de este JIRA.



EJEMPLO DE DESCRIPCIÓN

Descripción

Se libera la funcionalidad que consiste en enviar notificaciones SMTP para cada uno de los mensajes enviados a las apps. Para mas detalle se adjunta la Especificación de Requerimiento de Software adjunta en el Jira.

Dependencias

Es necesaria la instalación [PMC-548](#) para que se pueda habilitar esta funcionalidad.

Impactos

Con la liberación se agrega la nuevas funcionalidad de envío de notificaciones SMTP

Riesgos

Este cambio puede afectar el envío general de correos electrónicos



GLOSARIO DE TÉRMINOS

La descripción del JIRA de liberación debe contener un glosario con la descripción de palabras clave, o variables de entorno necesarias durante la instalación, por ejemplo:

Glosario de Terminos

<code>\$MULE_HOME</code>	Se refiere a la ruta donde esta instalado mule; ejemplo: <code>/app/cautivo/mule-standalone-3.2.1</code>
<code>\$CORE_HOME</code>	Se refiere a la ruta donde esta instalado el captiveCore; ejemplo: <code>/app/cautivo/mule-standalone-3.2.1/apps/captiveCore</code>
<code>\$WS_INTERFACE_HOME</code>	Se refiere a la ruta donde esta instalado el wsInterface; ejemplo <code>/app/cautivo/tomcat/webapps/ws-interface</code>



INSTRUCCIONES

La descripción del JIRA de liberación debe contener las instrucciones de instalación lo más detallado posible, con la finalidad de que independientemente quien lo lea, este sea comprensible.

Estas instrucciones pueden ser apoyadas con el glosario creado anteriormente, se deben enlistar las siguientes instrucciones, si es que aplican:

- ➔ ***Sustitución de archivos***
- ➔ ***Borrado de archivos***
- ➔ ***Modificación de archivos (Mostrar antes y después)***
- ➔ ***Ejecución de scripts (SQL, Bash, Etc)***
- ➔ ***Reinicio de servicios si es necesario***

****Consultar archivo de liberaciones en JIRA***

MONTERREY

+52 (81) 1001-0460

DF

+52 (55) 5985-7926

COLOMBIA

+571 6467-848

BRASIL

+55 11 99369 7435



CITI

Value in real time

www.citi.com.mx



/citi_vrt

/citivrt