|  |  |  |
| --- | --- | --- |
|  |  | Instructivo de  Gestión de Código: |
|  | **Área de Diseño y Desarrollo de**  **Software**  **ISO 9001:2015** |
|  |

Historial de Revisiones

| **Autor** | **Fecha** | **Revisión template** | **Versión documento** | **Principales Cambios** | **Aprobador** | **Fecha aprob. y publicación** |
| --- | --- | --- | --- | --- | --- | --- |
| Claudia Carretero | 29/11/2017 | Rev.00 | 00 | Ref. 1 | Gabriel Bulfón | 18/04/2018 |
| Claudia Carretero | 02/07/2018 | Rev. 01 | 01 | Ref. 2 | Gabriel Bulfon | 26/08/2018 |

|  |  |
| --- | --- |
| **Referencia** | **Descripción de cambios** |
| **Ref .1** | Creación del documento |
| **Ref. 2** | Se aplican los cambios del nuevo template de calidad “TMP - Documentación de Calidad”. |

**Índice**

[Instructivo de Diseño y Desarrollo - 4 -](#_Toc49321276)

[1 Objetivo. - 4 -](#_Toc49321277)

[2 Alcance. - 4 -](#_Toc49321278)

[3 Descripción. - 4 -](#_Toc49321279)

[3.1- Concepto – Sistema de Control de Versiones. - 4 -](#_Toc49321280)

[3.2- Versiones anteriores. - 5 -](#_Toc49321281)

[4 Metodología en doTech - 5 -](#_Toc49321282)

[4.1- Abrir proyecto. - 5 -](#_Toc49321283)

[4.2- Release.Txt - 5 -](#_Toc49321284)

[4.3- Deployar en QA. - 6 -](#_Toc49321285)

[4.4- Cierre de Proyecto. - 6 -](#_Toc49321286)

[4.5- Base de Datos - 7 -](#_Toc49321287)

[5 GLOSARIO DE TERMINOS Y ABREVIATURAS. - 7 -](#_Toc49321288)

# ****Instructivo de Diseño y Desarrollo****

# ****Objetivo.****

El objetivo de este instructivo es unificar criterios de forma de trabajo cuando se gestiona código fuente dentro del proceso de Diseño y Desarrollo de Software en doTech.

# ****Alcance.****

Está dirigido a los desarrolladores encargados codificar y deployar aplicaciones.

# ****Descripción.****

El control de código fuente (o el control de la versión) es la práctica de seguimiento y administración de los cambios en el código. Los sistemas de administración de código fuente (SCM) mantienen un historial del desarrollo de código, y ayudan a resolver conflictos a la hora de combinar las contribuciones de diferentes orígenes.

Este proceso pretende ordenar los pasos a seguir para gestionar exitosamente el código desarrollado y en desarrollo en el SCV.

# ****Concepto – Sistema de Control de Versiones.****

Para ello se van a utilizar los conceptos de trunk, branch y tag. Son tres de los conceptos principales a manejar a la hora de utilizar un sistema de control de versiones.

Trunk.

Es la línea principal de desarrollo (tronco) sobre la que se esta trabajando actualmente. La versión del trunk generalmente es la mas inestable de todas.

Branches.

Toda modificación de proyecto se hace en un branch (rama). Cada cambio que se hace luego se pasa a producción y el branch es promovido a trunk, Cada Branch es versiónado de acuerdo a galileo, así un ejemplo para el sistema Sistema Auro seria:

Branch-Auro-v1.0.0.0

Contendrá el Branch con la versión 1.0.0.0 del sistema Auro. El alcance de la versión puede conocerse en el documento de alcance de producto.

Los detalles de deploy pueden leerse en el documento Release.txt que contiene cada proyecto. La información de reléase esta duplicada en Galileo donde se puede ver la fecha de implementación sobre cada ambiente.

Tags.

Se utiliza para marcar una versión especifica o un estado especifico del proyecto que por alguna razón necesite ser identificado. Generalmente se utiliza en estadios intermedios.

# ****Versiones anteriores****

En versiones anteriores se guarda toda la codificación que no está vigente.

# ****Metodología en doTech****

# ****Abrir proyecto.****

Cuando comienza un proyecto o se modifica uno existente lo primero que se deberá solicitar es la versión del mismo al PM.

A partir de la implementación de este documento no se implementará más el trunk. Se abrirán los proyectos como Branch con su numero de Release a fin de mapear el alcance con la documentación.

# **Release.Txt**

Cada proyecto deberá contener un documento de Release con el detalle de los cambios realizados en el build.

Se deberá identificar con dos letras el responsable del cambio, se indicará también los archivos que se vieron modificados y que son necesarios subir/implementar cuando se pase la versión a un ambiente. Se deberá indicar también si hay necesidas de ejecutar scripts en la base. Se deberá indicar en este ultimo caso el path completo del script si la ruta no está organizada previamente.

Ejemplo:

Lucio Roca hizo un cambio actualizando el caso de uso 023-Modificar usuario cambiando la pagina y compilando, generando así un nuevo build. En el proceso agrego una columna en la tabla Persona y genero un script con el numero 9 con el cambio.

LR.Actualizacion del caso de uso 023-Modificar

Scripts

* 09-Agrega columna estado en Persona.sql

Deployar

\bin\\*.\*

\Persona\Persona.aspx

Esta información es útil para la persona que va a realizar la implementación entendiendo con ello quien hizo el cambio, que scripts hay que correr y que archivos se deben subir/copiar al servidor.

# ****Deployar en QA.****

Una vez finalizado el desarrollo en el branch se deberá probar integrado en el ambiente de QA. Cuando las pruebas se concluyen, se deploya el branch a producción

El deploy deberá registrarse en galileo haciendo una copia de los cambios que contiene el reléase.txt. Como buena práctica, se copia exactamente lo mismo desde el reléase.txt

Cuando el responsable de deployar integra todo y realiza el build, lo instala sobre el ambiente de QA. En ese momento se lee el build number y se copia en el archivo Release.txt a fin de saber exactamente cual es el build que se promueve al ambiente.

Se deberá copiar la URL del deploy de galileo en el archivo Release.txt a fin de facilitar la revisión futura.

# ****Cierre de Proyecto.****

Se deberá informar los indicadores de código: cantidad de código agregado y complejidad a PMO para el cierre del proyecto en galileo

# ****Base de Datos****

Se deberá tener la estructura de datos limpia, actualizada.

**4.6 - Ejemplo completo**

A continuación, se copia un ejemplo de entrada en el archivo Release.Txt del proyecto TRN-Web-Admin.

AA.TURNERA - VISOR: Permitir cargar la configuración realizada en el Alta y Modificación de la Sala virtual. Muestra mal el tipo de turnera configurado, configuré "Listado llamados" pero trae "Listado en espera". (Lic - Demartino - New 2)

AA. En sala virtual llamador: en firefox abria multiples ventanas el abrir la turnera de la sala.

AA. ALTA SALA DE ESPERA: si el logo no respeta las restricciones muestra object reference. Las especificaciones no son reales, permite subir un logo de mayor tamaño del indicado. También permite subir archivos txt.

AA. ALTA SALA DE ESPERA: si no se completan los campos nuevos, muestra Error en BD. Sigue fallando.

AA. Se agrego en nro de version y descripcion el logon y menu lateral.

Archivos.

Bin

Administracion/Turnera/turnera-viso.aspx

Administracion/SalaVirtual/registrar-sala-virtual.aspx

SalaVirtual/sala-virtual-llamador.aspx

Deploy

Version: 1.3.0.25549

QA:https://galileo.dotech.io/Deploys/detalle-deploys.aspx?deployId=7EA6DFE9A2E42598

# ****GLOSARIO DE TERMINOS Y ABREVIATURAS.****

|  |  |
| --- | --- |
| **Término o Abreviatura** | **Descripción** |
| PM | Oficina de gestión de proyectos |
| BUG | Error o fallo en un programa o sistema de software que desencadena un resultado indeseado. |
| QA | Aseguramiento de la calidad (del inglés Quality Assurance). En este documento hace referencia al despliegue del código en un servidor donde se correrán las pruebas. |
| SCV | Sistema de Control de Versiones |