

1. OBJETIVO

Definir los lineamientos y políticas de administración de código fuente en el desarrollo, colaboración y/o revisión de proyectos de desarrollos.

2. ALCANCE

Aplica para toda la plataforma de tecnología en todos sus ambientes tales como; productivos, contingentes, desarrollo y pruebas

3. DEFINICIONES

Ambiente de Pruebas: Infraestructura de hardware y software necesaria para realizar la ejecución de pruebas de integración, sistema y funcionales para determinar el comportamiento de la aplicación en escenarios reales.

Ambiente de desarrollo: Es un conjunto de procedimientos y herramientas que se utilizan para desarrollar un código fuente o programa. Este ambiente es utilizado para escribir, generar, probar y depurar un programa.

Ambiente de contingencia: Es una Infraestructura con procedimientos y planes en las cuales están integrados todos los servicios tecnológicos críticos de la compañía.

Ambiente de Producción: El ambiente de Producción es donde están disponibles las funcionalidades necesarias tecnológicas y de procesos para la ejecución de los procedimientos.

Control de código fuente: 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.

4. CONDICIONES GENERALES

Herramientas

Github

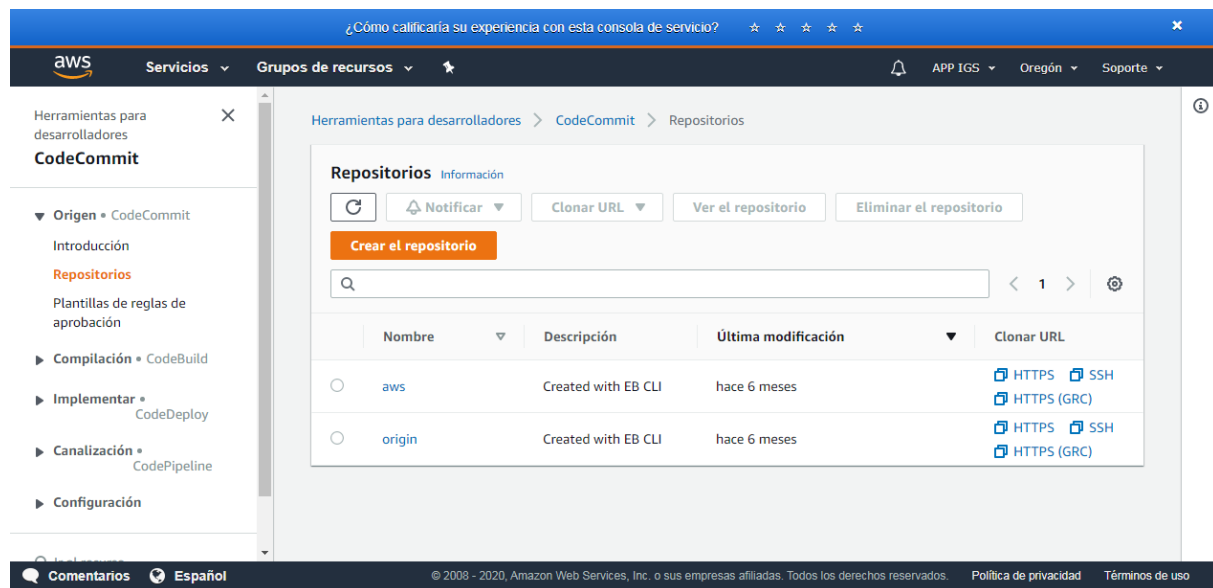
Es un sistema de administración de código fuente distribuido de código abierto. Git permite crear una copia de su repositorio, conocida como rama. Con esta rama, podemos en su código con independencia de la versión estable de la base de código. Cuando los cambios estén listos, podemos almacenarlos como un conjunto de diferencias, conocido como confirmación.

Podemos insertar confirmaciones de otros colaboradores en el repositorio, enviar confirmaciones a otras personas y combinar sus confirmaciones de nuevo en la versión principal del repositorio.

Comparación gráfica en paralelo

12	<% if((String)session.getAttribute("cuit")!=null){ %>	12	<% if((String)session.getAttribute("cuit")!=null){ %>
13	- Datos De Asignacion Asistencias	13	+ <div id="divimg" style="height:10%; background-color: gray;">
		14	+
		15	+ <out.println("Cuit: " + (String)session.getAttribute("cuit") + ""); %>
		16	+ Salir
		17	+ </div>
14	<div id="divAsignacion" class="jrapid_listing_table" style="	18	+ Datos De Asignacion Asistencias
15	height: 280.484px;	19	<div id="divAsignacion" class="jrapid_listing_table" style="
16	max-height: 100%;	20	height: 280.484px;
		21	max-height: 100%;
23	padding: 15px 1% 5px 1%;	28	padding: 15px 1% 5px 1%;
24	display: block;	29	display: block;
25	width: 98%;	30	width: 98%;
26	- height:80%;	31	+ height:70%;
27	>	32	>
28	<table border="1" class="jrapid_table " style="	33	<table border="1" class="jrapid_table " style="
29	border-spacing: 0px;	34	border-spacing: 0px;
102	Usted No Tiene Permiso Para Acceder A Esta Plataforma	107	Usted No Tiene Permiso Para Acceder A Esta Plataforma
103		108	
104		109	
105	- <% %>	110	+ <% %>
106	</body>	111	</body>

Interfaz de la herramienta



Proceso y gestión de código fuente.

IGS al escribir una aplicación sencilla o un gran proyecto o colaborar como parte de un equipo, el control de código fuente es un componente esencial del proceso de desarrollo.

Los sistemas de administración de código fuente permiten supervisar los cambios en el código, consultar el historial de revisiones del código y volver a versiones anteriores de un proyecto cuando sea necesario.

Con los sistemas de administración de código fuente, puede trabajar en equipo en el código, aislar el trabajo hasta que esté listo e identificar quién realizó los cambios y qué cambios se realizaron para poder solucionar rápidamente los problemas.

Los sistemas de administración de código de fuente nos permiten optimizar el proceso de desarrollo y proporcionan un origen centralizado para todo su código.

Seguir los siguientes lineamientos.

- Ingresar a la aplicación GitHub con su usuario / contraseña
- Crear el repositorio

- Realizar siempre antes de terminar un commit y un push
- Inspeccionar los cambios antes de hacer un commit
- Realizar un Fetch/Pull antes de hacer un commit
- No efectuar push directamente al master
- Usar y comprender la herramienta de mezcla y comparación
- Versionar bases de datos y otras cosas.
- Excluir archivos personales auxiliares del control de código, usar el archivo .gitignore para esta labor.
- Se bajan los cambios y se revisa el código. (*)
- Posteriormente a la revisión entra en a pruebas.
- Posteriormente entra al procedimiento de “gestión de cambios” para ir a producción.

Revisión de código fuente.

En este apartado se revisa lo siguiente:

- Que el código este limpio, excluir nombres de terceros, personales, ni loops, etc.
- Que se cumplan los estándares de desarrollo definidos en la Política de IGS.
- Cumplimiento del formato estándar utf-8.
- Vulnerabilidades y librerías desactualizadas.
- Seguimiento del Manual de procesos,
- Nombres de variables y clases coherentes.
- Control de errores y manejo de clases.

5. DOCUMENTOS DE REFERENCIA

- Política Seguridad de la Información.

6. HISTORIAL DE CAMBIOS

FECHA	VERSION	NATURALEZA DEL CAMBIO
30-08-2019	1.0	➤ Creación del documento.
22-02-2020	2.0	➤ Adición en Ítem 4.0 "Proceso y gestión de código Fuente y Revisión de Código Fuente.

DE USO INTERNO