**UNIVERSIDAD TECNOLÓGICA DEL CENTRO DE VERACRUZ**

DESARROLLO WEB INTEGRAL

**SISTEMA COMPUTE PARTS**

ACATECATL BERNABE LUIS ENRRIQUE  
CABALLERO MORALES EDUARDO  
 MARTINEZ ALONSO EDUARDO  
VIVIANA VELASCO MONTERO

IDGS 9 “A”

FRANCISCO

# 1. Estructura de Ramas

La estructura de ramas sigue un modelo jerárquico donde se organiza el flujo de desarrollo desde versiones estables hasta características específicas. A continuación, se describe la jerarquía:

- production  
 - release  
 - v1.00  
 - v2.00  
 - integration  
 - develop  
 - WEB  
 - DEV\_WEB  
 - feat/...  
 - API  
 - DEV\_API  
 - feat/...  
 - DESIGN  
 - DEV\_DSG  
 - feat/...  
 - test

# 2. Nomenclatura

Cada colaborador tiene un prefijo definido para la creación de ramas y commits.

|  |  |  |
| --- | --- | --- |
| Nombre | Prefijo | Rol |
| Viviana Velasco Montero | VVMP0 / VVMPDBA | PO / DBA |
| Eduardo Caballero Morales | ECMSM / ECMINT | SM / INT |
| Eduardo Martínez Alonso | EMADI / EMATW | DI / TW |
| Luis Enrique Acatecatl Bernabe | LEABQA / LEABDEV | QA / DEV |

PO (Product Owner): Representa los intereses del cliente o usuario final. Es responsable de definir y priorizar los requisitos del producto en el backlog y de comunicar la visión del producto al equipo.

DBA (Database Administrator): Encargada de gestionar, mantener y asegurar el correcto funcionamiento de las bases de datos utilizadas por los sistemas.

SM (Scrum Master): Líder facilitador que guía al equipo en la metodología Scrum, asegurándose de que los procesos ágiles se ejecuten correctamente y eliminando obstáculos que afecten al avance del equipo.

INT (Integrator): Responsable de integrar las diferentes ramas del código y asegurar que el sistema funcione como un todo al combinar componentes desarrollados por distintos miembros del equipo.

DI (Designer): Diseñador encargado de la creación de interfaces, experiencia de usuario y elementos visuales para garantizar que el producto sea atractivo y funcional.

TW (Tech Writer): Redactor técnico que documenta procedimientos, guías de usuario y especificaciones técnicas para facilitar el entendimiento del producto por parte de los desarrolladores y usuarios.

QA (Quality Assurance): Encargado de garantizar la calidad del producto mediante pruebas sistemáticas y asegurándose de que cumpla con los estándares definidos antes de ser entregado.

DEV (Developer): Desarrollador responsable de escribir, probar y mantener el código necesario para implementar las funcionalidades del sistema.

## 1. Políticas de Combinación de Ramas

Las siguientes políticas están diseñadas para garantizar la calidad del código, la estabilidad del repositorio y la colaboración efectiva entre los distintos roles del equipo de desarrollo. Cada rama tiene reglas específicas de uso, responsables claros y requisitos técnicos definidos.

### 1.1 Rama production

La rama 'production' representa el estado final del producto en producción. Solo se permiten combinaciones desde ramas 'release/vX.XX'. Los responsables de aprobar cualquier fusión a esta rama son el Product Owner (VVMPO) y el responsable de QA (LEABQA). Antes de aprobar una fusión, se deben haber completado exitosamente las pruebas end-to-end (E2E), evaluaciones de seguridad y la documentación técnica debe estar actualizada por el responsable TW (EMATW). El tiempo máximo permitido para completar una revisión y aprobar un merge es de 48 horas.

### 1.2 Rama release/vX.XX

Estas ramas agrupan funcionalidades preparadas para ser liberadas en producción. Pueden originarse desde la rama 'integration' o directamente desde componentes como iwEB, API o DESIGN. El Product Owner y el Scrum Master (ECMSM) son responsables de su aprobación. Deben validarse todas las funcionalidades incluidas y documentarse adecuadamente antes de ser fusionadas a 'production'. El tiempo máximo de validación es de 24 horas.

### 1.3 Rama integration

La rama 'integration' sirve como punto de convergencia para los desarrollos provenientes de 'develop' y ramas de componentes. El integrador (ECMINT) es responsable de realizar el merge, siempre y cuando el build sea exitoso y se hayan superado las pruebas de smoke test. El tiempo máximo para este proceso es de 12 horas desde la solicitud de integración.

### 1.4 Rama develop

La rama 'develop' se utiliza como base de trabajo continua para los desarrolladores. Solo el Scrum Master (ECMSM) puede aprobar combinaciones en esta rama. Se exige que las funcionalidades cuenten con pruebas unitarias completas. Todas las integraciones deben resolver conflictos de manera ordenada y respetar los estándares de codificación. El merge debe realizarse dentro de un plazo de 12 horas.

### 1.5 Ramas de Componentes (iwEB, API, DESIGN)

Estas ramas permiten la colaboración específica por áreas de trabajo. Cada una tiene un responsable técnico:  
- iwEB: Desarrollador (LEABDEV)  
- API: Technical Writer (EMATW)  
- DESIGN: Diseñador (EMADI)  
  
Las fusiones deben pasar por revisiones de código o validaciones contractuales/diseño según corresponda. El tiempo máximo de integración es de 12 horas desde la solicitud del pull request.

### 1.6 Ramas de Trabajo (fret/\*)

Estas ramas se crean para desarrollar nuevas funcionalidades o resolver incidencias específicas. Su nomenclatura debe seguir el formato: 'fret/<Prefijo>-<ticket>-<descripción>' (ej. 'fret/LEABDEV-123-login-fix'). Toda fusión debe contar con al menos un Code Review obligatorio:  
- Si afecta funcionalidad: 1 DEV + 1 QA  
- Si afecta diseño o documentación: 1 DI + 1 TW  
  
El código solo puede integrarse si el build es exitoso (Build Verde). El merge debe realizarse dentro de las 24 horas siguientes a la solicitud del pull request.

### 1.7 Hotfixes (Excepciones)

En caso de errores críticos en producción, se permiten hotfixes directamente hacia ramas críticas. Deben contar con la aprobación de PO, QA y DBA (si afecta base de datos). Es obligatorio notificar al Scrum Master y registrar la modificación en el archivo '/hotfix-log.md'. El proceso completo debe realizarse en un lapso menor a 4 horas.

## 2. Sanciones por Incumplimiento

Para garantizar el cumplimiento de las políticas establecidas y mantener la integridad del repositorio, se aplicarán las siguientes sanciones ante cualquier incumplimiento detectado:

### 2.1 Merge sin Aprobación

Cualquier fusión realizada sin la debida aprobación por los responsables será revertida en un plazo máximo de 24 horas. Esta acción busca evitar la incorporación de código no validado que pueda comprometer la estabilidad del proyecto.

### 2.2 Dos Incumplimientos Reiterados

En caso de que un colaborador incurra en dos faltas consecutivas o acumuladas relacionadas con el incumplimiento de estas políticas, se procederá al bloqueo temporal de sus permisos de combinación (merge) por un periodo de 72 horas. Durante este periodo, el colaborador deberá asistir a una revisión de buenas prácticas con el equipo técnico.

### 2.3 Tres o Más Incumplimientos

Cuando un colaborador incurra en tres o más incumplimientos, el Product Owner (VVMPO) evaluará su acceso al repositorio. Podrá tomarse la decisión de restringir sus permisos, reasignar tareas o incluso suspender su participación temporal en el desarrollo activo. Esta decisión será documentada formalmente y comunicada al equipo.

# Políticas de Pull Request (PR)

* Nombres descriptivos: Cada Pull Request debe tener un título claro y específico que describa el propósito del cambio, facilitando así su identificación en el historial del proyecto.
* Vinculación a tareas o tickets: Los PR deben estar asociados a una tarea o ticket del sistema de gestión de proyectos. Esto asegura que los cambios realizados estén alineados con los objetivos del sprint o proyecto.
* Revisión de código: Antes de que un PR sea aprobado, debe pasar por una revisión exhaustiva por parte de al menos un integrante con rol de integrador (INT) o con mayor jerarquía. Esto garantiza la calidad y consistencia del código.
* Pruebas automáticas: Es obligatorio que todos los PR completen satisfactoriamente las pruebas automáticas configuradas en el proyecto. Estas pruebas verifican la funcionalidad y la estabilidad del código nuevo o modificado.
* Fusión en ramas principales: No se permite la fusión directa en ramas principales (como develop, release o production) sin una revisión previa aprobada. Esto asegura que los cambios sean confiables y estén completamente verificados.
* Uso de prefijos en nomenclatura: Los PR deben seguir los prefijos establecidos para identificar al autor y el tipo de cambio realizado. Ejemplos de prefijos incluyen "feat/", "fix/", "test/", "doc/", para facilitar la identificación rápida del propósito del PR.

## Sanciones en caso de incumplimiento

* Advertencia inicial: En caso de un incumplimiento menor, se emitirá una advertencia escrita al responsable, especificando el problema identificado y proporcionando recomendaciones para evitar futuros errores.
* Revisión obligatoria adicional: Si el incumplimiento se repite, el responsable deberá someter futuros PR a una revisión obligatoria por parte de dos integrantes con roles de integración o superior, hasta que se demuestre un cambio en las prácticas.
* Suspensión temporal: En casos graves o de incumplimientos repetitivos, el responsable puede ser suspendido de la creación de PR durante un período determinado. Durante este tiempo, sus propuestas deberán ser gestionadas por otro miembro del equipo.
* Impacto en evaluación del desempeño: Incumplimientos sistemáticos pueden ser considerados como factores negativos en las evaluaciones de desempeño del responsable, afectando su progreso profesional dentro del equipo.

Estas sanciones buscan fomentar el cumplimiento de las políticas, mantener altos estándares de calidad y promover la colaboración responsable entre los miembros del equipo.