Acuerdos Técnicos

Actualización del GRPUy Administrativo, Financiero y Presupuestal

TABLA DE CONTENIDO

HISTORIA DE CAMBIOS 3

1. OBJETIVO 4

2. ACUERDOS 4

Historia de cambios

Historia de revisiones – Redacción			
Versión	**Fecha**	**Autor**	**Motivo del Cambio**
1.0	16/1/2024	Equipo Tec. PFGP	Versión inicial
1.1	06/3/2024	Equipo Tec. PFGP	Se agrega reporte de incidentes de MR en redmine.

Historia de revisiones – Revisión y aprobación				
Versión	**Fecha**	**Nombre**	**Revisión**	**Aprobación**
1.0	16/1/2024	Equipo Tec. PFGP		
	07/02/2024	Equipo Tec. PFGP		
	26/02/2024	Equipo Tec. PFGP		

La fecha de aprobación indica la fecha que se aprobó la institucionalización del mismo.

1. Objetivo

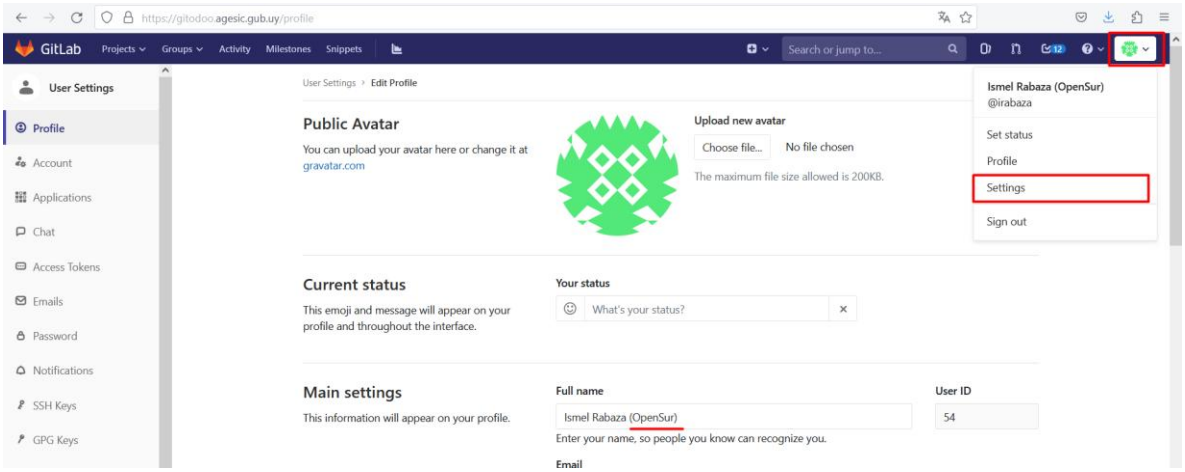
El presente documento tiene por finalidad describir los estándares o acuerdos a nivel técnico que se utilizaran en el proceso de desarrollo durante el proyecto.

2. Acuerdos

- Versión de Odoo a utilizar: 17.
- Commit a utilizar al inicio: 20240123 (https://github.com/odoo/odoo/tree/140e58c5a68db674fc1d240147e4d669b29f2cad)
  La versión de Odoo 17 se va a actualizar en el transcurso del proyecto, con el objetivo de tomar una versión más completa y con menos bugs.
- Herramienta para integración
  - Integration Brocker
    Se utilizará la herramienta presentada para todas las integraciones que sean mantenidas y actualizadas en el GRPUy.
- Documentación al desarrollar
  - Cada desarrollador en su primer commit, debe al menos inicializar todos sus módulos, es decir los módulos que están bajo su responsabilidad.
- Estándar de desarrollo
  - Se utilizarán recomendaciones de la comunidad
    Guías de desarrollo
    - https://www.odoo.com/documentation/17.0/contributing/development/coding_guidelines.html
    - https://www.odoo.com/documentation/17.0/contributing/development/git_guidelines.html
    - Estándares de desarrollo de Python 3
  - Los nombres se mantendrían tal cual están en los campos y variables.
  - Si hay una variable nueva, se seguirá la codificación de Odoo pero en español cuando la terminología es asociada a GRPUy.

- Procedimiento de trabajo Git

PFGP necesita identificar en el GitLab la empresa a la que pertenece cada usuario autor del commit sin necesidad de entrar al perfil. Para ello cada usuario debe establecer el nombre de la empresa en la configuración de su perfil. Ver imagen siguiente.



A su vez, configurar la copia de trabajo local del repositorio del proyecto para establecer igual nombre en el Autor. Esta configuración se puede hacer desde el cliente Git que se use o editando el fichero de configuración local o global del Git, en la sección **user**:

```
[user]
    name = Ismel Rabaza (OpenSur)
    email = irabaza@opensur.com
```

Se plantea tener un procedimiento de trabajo organizado y a la vez sencillo, que no sea necesario crear nuevos repositorios. El procedimiento sería mediante la creación de ramas (branches) para las funcionalidades o incidencias y luego la solicitud de Merge Requets a la rama correspondiente.

1. Clonar rama master del repositorio Git: https://gitodoo.agesic.gub.uy/odoo/grpuy4.git
2. Crear una rama a partir de la master para la funcionalidad/hotfix que se está desarrollando, la rama debe llamarse “features/*proveedor_funcionalidad*” o “hotfix/*proveedor_incidencia*”.

Ejemplo: “features/opensur_mantenimiento_v17”

3. Push de la rama. Trabajo de la funcionalidad o incidencia e ir subiendo los cambios.
4. Solicitar Merge Request hacia la rama master desde el repositorio en https://gitodoo.agesic.gub.uy/odoo/grpuy4 donde la rama origen debe ser la rama con los cambios subidos y la rama destino la master. En “Asignado a” establecer a Liber Matos. Tener en cuenta desmarcar el checkbox “Eliminar rama origen cuando Merge Request sea aceptado”, de forma que el proveedor pueda volver a usar esta rama en caso que se detecte algún problema o incidencia asociada al feature/hotfix.
5. Si el Merge Request no es aceptado se debe corregir o ajustar según los comentarios del integrador en la misma rama origen, así como en el ticket que se creará en redmine.

Se reportará el incidente al proveedor que lo pidió, con los siguientes datos:

- URL: Peticiones - CORE Vertical Estado Uruguay - Odoo (mef.gub.uy)
- Proyecto: Core vertical estado uruguayo
- Tipo: error MR
- Estado: Nueva
- Versión detectada: versión 4.0.0 (no cambia porque las nuevas subversiones se generan LUEGO de un Merge Request exitoso, por lo que no sería posible anticipar la subversión)
- Proveedor: quien solicita el Merge Request

- Versionado: El número de versión en el fichero __manifest__.py del módulo debe ser la versión principal de Odoo (por ejemplo, 17.0) seguida de los números de versión del módulo **x.y.z**.

Por ejemplo: se espera **17.0.1.0.0** para la primera versión entregada del módulo
Los números de versión **x.y.z** siguen la semántica break.feature.fix:

x: aumenta cuando el modelo de datos o las vistas tienen cambios significativos. Es posible que sea necesaria la migración de datos o que los módulos dependientes se vean afectados. Si corresponde, se espera que los cambios importantes incluyan instrucciones o scripts para realizar la migración en instalaciones actuales.
y: aumenta cuando se agregan nuevas funciones o ajustes continuos al módulo. Probablemente será necesaria una actualización de módulos.
z: aumenta cuando se corrigen errores o ajuste pequeño. Generalmente solo es necesario un reinicio del servidor.

- Seguridad por Roles: Los roles de seguridad deberán definirse e incluir su configuración en el código (fichero XML) en cada módulo. Cada proveedor es responsable de agregar los roles definidos funcionalmente según el negocio.
- Modularidad: Las validaciones correspondientes a ciertas clases deben realizarse en los módulos que las definen.
- Se solicita revisar los modelos con relaciones a Unidades Organizativas, Unidades Operativas y otras entidades, y cuando tenga dependencia no permitir su eliminación. Técnicamente se debe usar ondelete="restrict" en la definición del campo relacional para no permitir eliminar el registro si está referenciado en otros. Aquellos casos que generen duda se deben consultar a PFGP para definir si se permite eliminar el registro o se restringe.

- Se establece utilizar el idioma Español Uruguay (Spanish (UY) / Español (UY)). En caso de que no se esté trabajando con ese idioma configurarlo desde el sistema. Para hacerlo ir a Ajustes / Traducciones / Idiomas, buscar el registro de nombre Spanish (UY) / Español (UY) y clicar en el botón Activar de la vista Lista y luego en el

botón Agregar del asistente que se abre; finalmente clicar en Cambiar a Español UY y Cerrar.

3. Buenas prácticas de diseño y desarrollo

Se deberán contemplar las recomendaciones definidas en el documento compartidos en el siguiente link:

PGAFH_GRPUy40_20240201_PrácticasOdoo_Calidad_Código