

# Proyectos BinPar: Manual de Gestión en GitHub (v1.11)

#binpar/metodología

Versión 1.11

21 noviembre 2019

## Consideraciones básicas

- Las issues se deben cerrar por el equipo de BinPar una vez estén finalizadas y verificadas en el *entorno de desarrollo*.
  - Si en un issue no hay nada pendiente debe estar cerrada.
- Si una tarea ha sido cerrada y han surgido posibles colaterales o nuevas solicitudes en base a dicho desarrollo se debe crear un *nuevo issue* referenciando al original si nos aporta información valiosa para su seguimiento o resolución.
- En ocasiones surgen issues que no exigen ningún desarrollo y sirven para realizar una *consulta*, en estos casos deben ser etiquetadas con la etiqueta "question".
- Un número alto de issues en un repositorio no es sinónimo de mala *calidad*, recordamos que es un backlog.
  - Si se deben evitar las issues abiertas sin temas pendientes.
- Un issue solo puede estar *asignado* a una persona
- En algunos proyectos existen *plantillas* para la creación de issues que deberán ser cumplimentadas correctamente.
- Un issue no puede abordarse si no está toda la *información* necesaria disponible.
- No se podrá modificar el *alcance* de un issue una vez ha entrado en producción, es decir, se ha comenzado a desarrollar la versión donde está incluida.
  - Si el cambio de alcance es importante debemos sacar el issue de dicha versión y replantearla en otra versión.
- Si una issue se ha cerrado de forma errónea se debe reabrir. No si hay cambio de alcance, dudas u otros.

## Proyectos importantes

- En los proyectos de envergadura se dedicará el tiempo necesario a realizar su documentación, tutoriales y la formación al cliente. Este tipo de tareas tendrá su propia issue y estimación.
- En los nuevos presupuestos tendrá su propia partida.

## Gestión de Projects en GitHub

- En los proyectos con *versiones semanales o quincenales*, se creará un proyecto para cada versión.
- La versión en Beta se publicará los lunes y el miércoles si se ha dado el OK para su *publicación* se procederá a su despliegue en Producción.
- El viernes de cada semana debe quedar *definida y consensuada* con el responsable del cliente la siguiente versión.
  - Esta gestión del project de GitHub será responsabilidad del jefe de proyecto de BinPar.
- Las *columnas* necesarias para este tipo de proyectos serán: Inbox, To do, In progress, Done, Verified.
- En Inbox se incluirán todas las tareas *candidatas* a entrar en la próxima versión.
- Las que vayan a entrar en producción pasarán a la columna *"To do" priorizadas*.
- En la columna *"In Progress"* estarán las tareas que se están ejecutando en ese momento.
- Todas las tareas finalizadas pasarán a la columna *"Done"*
- La *verificación* de todas las funcionalidades de la versión puede ser realizada por el cliente o por BinPar. En cada proyecto habrá un responsable de esta tarea.
  - Existirá siempre una validación mínima en Beta de BinPar, principalmente orientada a asegurar que el cliente puede probar cada issue independiente.
  - Equipo de *Post-Venta (QA)*: Su participación será opcional dependiendo del proyecto.
    - Se responsabilizarían de un testing más exhaustivo tanto en Beta como en Producción, reduciendo las horas dedicadas del cliente al testing.
      - Serán tareas independientes en GitHub con su propia estimación.
      - En cuanto se revise una tarea y se considere correcta se asignará la etiqueta *"QA Verified"*
  - En última instancia, el cliente verificará las tareas aprobadas por QA de BinPar y si son correctas las moverá a la columna *"Verified"*. En caso contrario, reabrirá el issue que automáticamente pasará a la columna *"In Progress"*.
  - La *validación* final de la versión y por tanto, confirmación de su publicación en Producción será siempre del cliente.

### Tareas no planificadas

- En algunas ocasiones se da la circunstancia de que el cliente necesita incluir en un sprint en curso una tarea que no ha sido planificada en el sprint y tiene carácter de urgencia.
  - Si BinPar puede y tiene que realizar estas tareas en horas extra (fuera de horario de trabajo) el precio de tarifa será el doble del habitual.
  - Este tipo de issues se catalogarán con la etiqueta *"no-planificado"*.

### Catalogación issues

- La catalogación del issue a nivel de etiquetas se realizará por el equipo de BinPar.

- La excepción serían las tags de Prioridad y etiqueta con el nombre del producto (si existe) que deberán emplearlas el cliente en el momento de la creación.

## Gestión de bugs

- Un issue del tipo *bug* nunca debería recoger más de un problema
- Los bugs se catalogarán en función de su *severidad*, para lo cual se han definido 4 niveles:
  - *Severity 1*: Aplica cuando el sistema de producción no es operativo, afecta a todos los usuarios y no hay alternativa disponible. Por ejemplo: Se ha producido una caída del sistema.
  - *Severity 2*: Aplica cuando el sistema de producción está afectado de manera importante. Implica la imposibilidad de utilizar funcionalidades críticas. Por ejemplo: No se puede realizar una compra en un e-commerce.
  - *Severity 3*: Aplica cuando una funcionalidad relevante que antes ha funcionado correctamente ha dejado de funcionar. Por ejemplo: Problemas en la generación de eventos en el calendario de estudio.
  - *Severity 4*: Aplica cuando el sistema no funciona de acuerdo a lo establecido en los requisitos de la última versión publicada o a errores en funcionalidad poco crítica o defectos estéticos. Por ejemplo: Ha dejado de funcionar una casuística concreta del buscador o Errores que afectan al diseño del producto en su versión mobile
- Los bugs del tipo S1 o S2 se gestionarán como *Hotfix*.
  - Este tipo de bugs se acometerán en el momento de su recepción si el alta se produce en horario de oficina o a primera hora del día siguiente si se notifican por el 24x07.
- Los bugs de tipo S3 se resolverán el viernes de cada semana y se subirán a beta el lunes.
- Los bugs del tipo S4 se solventarán dentro de las versiones de evolución de los productos. Por tanto, serán considerados por los jefes de proyecto de BinPar y el cliente en la definición de cada versión de evolutivo.
- El cliente debe añadir siempre la etiqueta "bug" y la severidad a este tipo de issues para que entren en el scope de Post-Venta
- Este tipo de issues NO se catalogarán con etiquetas de prioridad.
- Para el alta de bugs se deberá cumplimentar una *plantilla* con toda la información necesaria para poder reproducirlo

## Sección Releases en GitHub

- En cada versión cerrada BinPar la enlazará con el project asociado.
- En cada entorno del proyecto se podrá ver la versión actual publicada.

## Gestión de Milestones en GitHub

- En el caso de tener hitos a futuro con una fecha concreta de entrega se podrán crear Milestones.
- Una milestone se podrá apoyar en uno o varios projects.

## Nomenclatura del versionado

vX.Y.Z

X: Se incrementará con cambios significativos: arquitectura, cambio general de interfaz de usuario, estructura de API,... . Tendrá un valor 0 hasta la primera publicación en producción, que será la v1.0.0

Y: Se incrementará con cada nueva versión (modelo incremental de mejoras o evolutivo).

Todas las piezas de esta nueva versión han de ser compatibles entre sí.

Z: Se incrementará con cada nuevo *hotfix*. Todas las piezas de esta nueva versión han de ser compatibles entre sí.

## Estimación de tareas

- Las tareas se deben *estimar* antes de su ejecución. Las etiquetas disponibles y su significado son las siguientes:
  - Zero Energy - 0,5 horas
  - Low Energy - 2 horas
  - Middle Energy - 5 horas
  - High Energy - 8 horas
- La estimación de la tarea no se puede cambiar una vez iniciado su desarrollo, ni tampoco el alcance de la misma.
  - En *casos excepcionales*, si al inicio del desarrollo de una tarea descubrimos que el tiempo necesario para ejecutar dicha tarea excede notablemente la estimación realizada debido a un impacto colateral desconocido y por tanto, puede ocasionar que su ejecución impida la incorporación de otras tareas comprometidas en el sprint, pausaremos la ejecución de la tarea, evaluaremos nuevamente su energy con la nueva información y solicitaremos al cliente su conformidad ante dicho cambio.
    - BinPar asumirá el tiempo dedicado previo a la nueva estimación y no acarreará coste alguno al cliente.
- Algunas tareas no se pueden estimar mediante energías, como por ejemplo: la gestión del proyecto, las reuniones,... Este tipo de tareas se facturaría a hora dedicada.
- Algunos proyectos vinculan este tipo de energías a la facturación.

## Historias de Usuario

- Si una tarea necesita más de un 8 horas para su ejecución deberá dividirse en subtareas. En este caso la tarea original se catalogará con la etiqueta "Historia Usuario" y se referenciará desde las subtareas.
- Dentro de cada historia de usuario y después de la definición del alcance aportada por el cliente (editando el comentario) se listarán las issues que la componen empleando el listado de checks.
  - Cuando QA verifique que cada issue es correcta deberá activar el check correspondiente.

NOTA: Esta documentación estará en continua evolución y se actualizará siempre que se definan nuevas mejoras.

---

Cristian Álvarez  
COO en BinPar  
[cristian@binpar.com](mailto:cristian@binpar.com)