

Manual de Roles

Descripción de roles en metodologías ágiles utilizados en proyectos finales.

Líder Técnico

El líder técnico es el responsable último de la integridad del código. Esto requiere que, por un lado, sea el administrador de las versiones, decidiendo qué versiones se integran a las ramas principales del proyecto y cuáles no cumplen los requisitos de entrega, ya sea porque no llegan a los estándares de calidad o porque la tarea está incompleta. Por otro lado será el nexo de comunicación con el Project Manager quien elevará los pedidos hacia otros roles como el Product Owner o Tech Manager. El líder técnico es el representante dentro del equipo de desarrolladores, por lo cual es a quien sus compañeros acuden por ayuda en sus bloqueos de tipo técnico, a quién se le presentan propuestas de mejoras de código y quien media en sesiones de planificación del proyecto en donde se toman decisiones estructurales. Como líder debe velar por la participación de cada integrante del equipo, utilizando el potencial de cada uno de manera que pueda hacer un aporte de valor, sin intervención en la tarea que ya ha sido asignada salvo en las instancias de evaluación y feedback (ante un pull request) o cuando se le solicita ayuda. Por último no deberá descuidar sus tareas como desarrollador, aunque puede recibir menor carga de tareas por tener mayor responsabilidad en esta área, pero de ninguna manera deberá tomar más tareas que sus compañeros de equipo.

Lo que un LT **sí** es:

- Un desarrollador full-stack, o al menos con mayor habilidad en una de las áreas back/front, aunque idealmente en este caso el equipo deba contar con un referente de la área en la que el LT no es competente.
- Como desarrollador es preferible que tome un rol fijo en back o front, pero pueda ser flexible a la demanda de ayuda de cualquier miembro del equipo.
- Líder máximo dentro del equipo de desarrolladores.
- Representante del equipo y quien se reúna con los PM, PO y TM.
- Nexo entre el equipo y el PM.
- Mediador en las sesiones de toma de decisiones estructurales del proyecto.



Proyecto Final Integrador

- Responsable de definir un estándar de calidad de formato en los proyectos (indentación, máximo de largo de línea, orden de los métodos, nombres de las carpetas y las clases, formas de nombrar los métodos) que favorezca, la claridad y limpieza del código. Estos deberán configurarse al comenzar los proyectos en todos los IDEs, siendo esta responsabilidad no del LT sino de los desarrolladores.
- Puede definir estándares de código limpio que se utilizarán en el proyecto. Podría por ejemplo sugerir mejoras de código, utilización de herencia cuando ve código repetido, implementación de interfaces, sobrecarga, etc.
- Responsable de exigir a los desarrolladores que cumplan con dichos estándares.
- Administrador de las versiones de git en ramas principales:
 - Hará las configuraciones iniciales de seguridad en GitHub para proteger las ramas principales de merges directos.
 - Aceptará/rechazará/evaluará pull requests.
 - Se asegurará de que la tarea del desarrollador esté completa y sea de alta calidad antes de aceptarla.
 - Tendrá la potestad de hacer modificaciones pertinentes a una rama desarrollador y efectuar un merge, cuando ante repetidas solicitudes de cambios, el desarrollador fallara en efectuarlos efectivamente y si la tarea es bloqueante para otro desarrollador o es requerida para el cierre de Sprint. Nivel de tolerancia recomendado: 2 solicitudes de cambios (aunque cada líder podrá definir con su equipo la mejor estrategia, mientras sea transparente y se le de feedback a su desarrollador).
- Mantener los canales de comunicación abiertos con el resto del equipo. El LT debe saber a través de las dailies el estado de las tareas de sus compañeros desarrolladores y debe mostrarse disponible para que sus compañeros de equipo puedan acudir por ayuda, información o cuando deban comunicarle algo. Es ideal tener un canal propio de equipo.
- Informar a su equipo de decisiones tomadas sobre el código en general o a un compañero de decisiones de cambio sobre el código en el que el compañero ha estado trabajando. Esto da, por un lado, una oportunidad de crecimiento para el compañero que puede aprender de sus errores y, por otro, muestra respeto por el tiempo que el compañero invirtió en dicho código. Dar oportunidad de corrección al mismo desarrollador que inició la tarea siempre que sea posible.



Proyecto Final Integrador

Lo que un LT **no** es:

- No es el dueño del proyecto. El proyecto es público y pertenece a cada uno de los integrantes del equipo que podrían aún después de terminado el proyecto decidir continuar en equipo o por su cuenta creando un fork del proyecto terminado.
- No es necesariamente quien brindó la idea del proyecto. En el caso de estar trabajando con un proyecto propio, no es necesario que quien haya dado con la idea sea el responsable máximo del código, ya que para esto existen otros roles que podría ocupar (product owner, arquitecto). Aun así puede serlo.
- No es quien manda. El LT es un desarrollador más con rol de liderazgo pero las decisiones se toman en equipo. El LT es elegido por su idoneidad en cuanto a código pero también por sus dotes de líder. Al tener mayor conocimiento en cuanto a código puede ser quien proponga muchos de los lineamientos de trabajo, pero estos deben estar claros y ser aceptados por sus compañeros. Estos estándares de trabajo se harán idealmente al inicio del proyecto, pero pueden ir puliéndose o agregándose nuevos de sprint a sprint, siempre y cuando se notifiquen y no sea una sorpresa o imposición y que sea beneficioso para el proyecto y conserve la buena convivencia. El objetivo es siempre generar consensos.
- No es quien hace la tarea de otros. Cada desarrollador es responsable de su propio código y de seguir las pautas preestablecidas. Si el desarrollador falla, el equipo falla, no el LT. Así que cualquier desarrollador puede brindar ayuda a otro sin necesidad de que sea siempre el LT, ya que los objetivos, logros y fallos son de todo el equipo y no de su líder.
- No es el más sobrecargado de trabajo y quien brinda su tiempo extra. La repartición de tareas debe ser equilibrada y contemplando el tiempo y dificultades que presenta tener un rol extra con otras responsabilidades. (velará por esto el Agile Delivery Manager)
- No es quien deba responder a la demanda de ayuda de sus compañeros siempre, pero es quien deba responder **ANTES** de llamar al Project Manager y al Tech Manager. Si todo se complica el último en recurrir es el LT y de no poder resolverlo se pedirá una audiencia con el TM a través del PM.





Proyecto Final Integrador

Desarrollador

Es el rol común a todos los miembros del proyecto. Es recomendable que luego de recibir la asignación a los grupos, cada desarrollador elija, según los requerimientos del proyecto, su preferencia y la cantidad de personas, un sub-rol *frontend* o *backend*. Definamos un poco qué se espera de este rol.

- En la repartición de tareas diarias cada desarrollador elige tareas a realizar (que no estén bloqueadas por otra tarea) y manifiesta qué tiempo le llevará realizar dicha tarea.
- Mantiene al equipo informado de su avance a través de la ceremonia *daily*.
- Debe estar presente en todas las ceremonias.
- Debe mantener informado a su LT de cualquier inconveniente para cumplir con sus obligaciones, ya sea con ceremonias o sus tareas. No es necesario esperar a una *daily* para informar algo que con antelación sabe que no podrá cumplir.
- Debe mantener su código limpio, buenas prácticas y sobre todo asegurarse que las siguientes condiciones se den antes de solicitar un *pull request* de su rama:
 - Proyecto funcionando, sin excepciones de las que tenga conocimiento. Todas las validaciones correspondientes a su tarea deben estar también funcionando.
 - No dejar comentarios que no documenten algo necesario en el código.
 - Dar formato antes de hacer un commit. Respetar las pautas de estilo que se han propuesto al inicio del proyecto.
 - Que efectivamente los cambios que intenta subir sean desde su propia rama y no de otra.
- Debe formar parte de la toma de decisiones que atañen al grupo.
- Debe respetar los acuerdos generados en las *retros* y los lineamientos técnicos propuestos por el LT o generados a partir de acuerdos grupales.
- Debe modificar su código a pedido del LT.
- Puede proponer mejoras de código, pero debe respetar las decisiones generales o de estilo del líder técnico.
- Puede proponer acuerdos y manifestar su visión. Es recomendable hacerlo durante la ceremonia *retro* para no interrumpir el flujo de trabajo, pero si el Scrum Master lo considera necesario, puede llamar a una reunión extra para tratar casos extraordinarios.

Un desarrollador **NO debe:**





Proyecto Final Integrador

- Hacer el trabajo de otro compañero.
- Intervenir en la tarea de otro compañero a menos que tenga un pedido explícito de ayuda. Sí puede ofrecerse cuando el compañero manifiesta un bloqueo.
- Tomar tareas sin solicitarlas al Agile Delivery Manager.
- Hacer *merge* con ramas principales.

Agile Delivery Manager

Es el responsable de la organización de las tareas. Por ende, es quien administra el **gestor de tareas**. En Proyectos Finales recomendamos el uso de Trello por su estructura simple y la rapidez en la que se puede aprender lo básico de la herramienta y con la que se puede empezar a utilizar. Al ser un rol compartido con desarrollador, este rol debe tomar menor cantidad de tareas de código para que el reparto sea justo. Para profundizar en lo que implica desempeñarse en este rol, podemos decir un ADM es:

- El encargado de dividir el trabajo en tareas pequeñas. Puede solicitar ayuda de sus compañeros para definir cuál es el mínimo en el que se pueden dividir dichas tareas. *Por ejemplo, una tarea que implica hacer la entidad Persona, su repositorio, servicio con CRUD y controladores es demasiado grande. Se pueden identificar por lo menos 4 o más tareas pequeñas.*
- El encargado de hacer seguimiento del *backlog*. Es recomendable ver a diario las actualizaciones de los compañeros en cuanto a sus tareas.
- El encargado de ver qué tareas pueden ser bloqueantes de otras al asignar las tareas para que ningún desarrollador se quede con “tiempo muerto” esperando a otro para continuar. Si llegase a pasar se le puede asignar otra tarea que no esté bloqueada o sugerir ayudar al compañero con la tarea bloqueante de la propia.
- Debe asignar tareas adicionales. El desarrollador debe acudir al ADM antes de tomar otra tarea si ya ha finalizado con las asignadas.
- Debe estar en comunicación directa con el LT, acudiendo a este cuando una situación le llame la atención, ya que al tener la visión global de lo que hay que hacer y lo que falta, puede pasarle información vital que ayuden a llegar con las tareas de cada Sprint o prevenir inconvenientes.





Proyecto Final Integrador

- Es el **responsable de asegurarse que se repartan las tareas prioritarias y correspondientes al MVP** (mínimo producto viable) que hayan acordado al principio del proyecto **antes que las tareas relacionadas a mejoras**.

Un ADM **NO** es:

- Quien necesariamente debe definir el tiempo que debe tomar cada tarea. Sino que el tiempo que le toma a cada desarrollador realizar su tarea es personal y debe manifestarlo al tomar la tarea.
- Quien debe definir completamente el tamaño de cada tarea. Puede recibir ayuda si no se cuenta con los conocimientos técnicos necesarios para hacerlo.
- El responsable que el desarrollador cumpla con la tarea que asumió.

Scrum Master

La persona que ejerza este rol es la responsable de dirigir las ceremonias y de entregar los productos de las mismas, como reportes de *dailies* y limpieza del *jamboard*. También es quien se ocupará de que haya una buena comunicación y dinámica de trabajo en general. La principal herramienta de cambio de este rol es la **reunión retrospectiva** y su técnica correspondiente, dónde se ordenará el discurso de los miembros del equipo sobre distintos aspectos del trabajo como por ejemplo: dificultades, oportunidades, preocupaciones, expectativas, etc. Y se traducirán en una serie de **acuerdos** grupales para mejorar el trabajo y hacerlo más eficiente.

El Scrum Master (SM), entonces es:

- Quien vela por que las ceremonias internas del equipo se lleven a cabo en **tiempo y forma**.
- Quien dirige las *dailies*, es decir, quien pregunta a cada miembro de equipo “¿qué hiciste ayer?”, “¿qué vas a hacer hoy?”, “¿tenés algún bloqueo?”
- El que hace el reporte de cada *daily* y lo envía al PM, PO y TM.
- La persona que dirige la retrospectiva.
- Tomando en cuenta lo trabajado en la retrospectiva, quien se encarga de proponer **acuerdos** o fomentarlos para mejorar el trabajo en equipo y su eficiencia.
- Quien recuerda al resto del equipo los acuerdos hechos en las retrospectivas y estimula su cumplimiento.
- Quien potencia la comunicación del equipo y posibilita que la palabra circule entre todos los miembros.





Proyecto Final Integrador

“ Llegado el caso de que se presente una situación de conflicto emocional/conductual entre pares, el SM podrá proponer acuerdos donde se mantenga la buena convivencia y sobre todo donde el foco se dirija hacia el cumplimiento de las tareas y los objetivos prácticos. Recordemos que estamos en un espacio de trabajo y las diferencias personales no deberían ser un impedimento para el trato cordial, la responsabilidad y el cumplimiento de los objetivos grupales por encima de los individuales. “

Un SM **NO** es:

- Alguien que solo se encarga de “la parte blanda” del proceso, es un desarrollador más, con una responsabilidad especial.
- La persona que decide quién tiene razón y quién no en una discusión o conflicto.
- La persona responsable de que todos “se lleven bien”, sino de proponer acuerdos para que el trabajo fluya mejor.