

# eosDAC Reporte del equipo de desarrollo

## Período 2020/07

La intención de los reportes del equipo de desarrollo ha sido ayudar a traducir lenguaje técnico a una narrativa descriptiva para ayudar a inversionistas, miembros del DAC, y público en general, a entender dónde estamos, el avance realizado, y los pendientes.

El principal avance del mes anterior fue la creación del ambiente de pruebas. Gracias a ello se logró invitar a diferentes interesados para crear DACs y poder reportar casos de problemas u observaciones. En varios casos se tuvo problemas y se necesitó asistencia manual para “desbloquear” el proceso de la creación del DAC. La prioridad para este mes fue ayudar a estos usuarios a poder crear los DACs de pruebas, e identificar cuales eran los casos que realmente detuvieran la liberación del DAC Factory, y cuales casos podrían tratarse en una siguiente etapa. De esta forma se redujo considerablemente la cantidad de tareas pendientes.

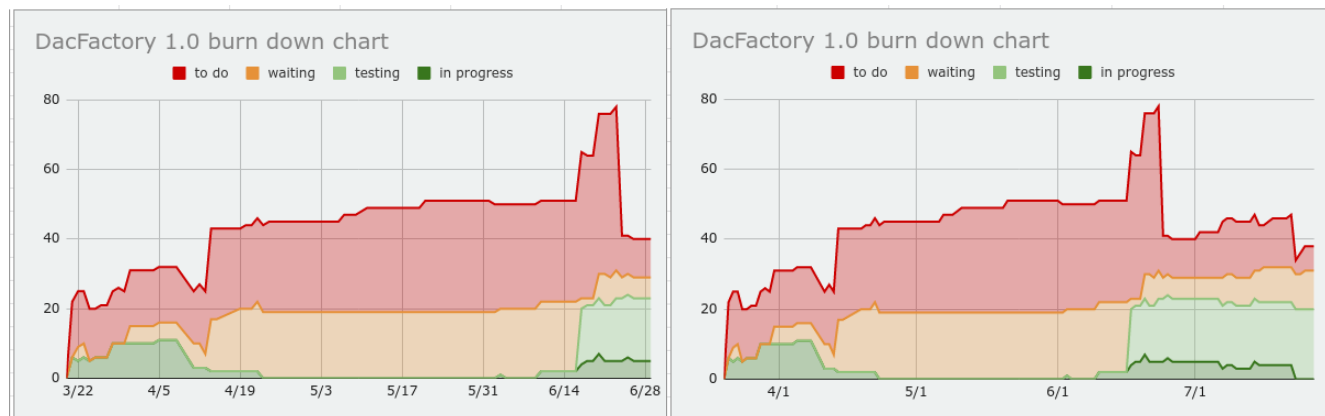
## ¿Qué se hizo en este mes?

Durante el mes se lograron cerrar los siguientes casos:

- **Caso #42 cobro del CPU al DAC y no al usuario.**  
Se concluyó que no era necesario gracias a la integración con Greymass/Anchor. No se desarrolló nada.
- **Caso #24 certificados HTTPS para subdominios del dacfactory.**  
Se se configuró un certificado multi dominios para los DACs creados. No se desarrolló nada.
- **Caso #92 redireccionar páginas HTTP a HTTPS**  
Ahora que ya tenemos HTTPS, se redireccionan las URLs. No se desarrolló nada.
- **Caso #101 error con DACs de 2 decimales.**  
Gracias a las pruebas realizadas se han detectado múltiples errores, el más impactante era que el sistema no permitía la creación de DACs con menos de 4 decimales. Dallas desarrolló la solución junto con pruebas automatizadas para probar casos extremos (sin decimales, o hasta 8 decimales). La solución ya está aplicada en Jungle 3.

## ¿Qué queda pendiente?

Gracias a la reducción de las tareas pendientes, actualmente tenemos únicamente 6 tareas por completar. De ellas, 3 son problemas visuales, 2 de manejo de errores, y solamente uno de ellos es problema funcional. En la gráfica vemos el área roja representando los pendientes.



*Comparativa de las gráficas de junio contra julio.*

En esta gráfica podemos ver que el equipo de desarrollo no tiene tareas en curso (área verde oscuro), y tiene pocas tareas por hacer (área en rojo). La gráfica muestra que el área más grande se encuentra en la validación y aprobación de las tareas, o control de calidad (área verde claro y naranja). Los casos por ser probados y aprobados suman un total de 31 tareas contra solo 6 por iniciar. El enfoque principal durante este mes ha sido realizar pruebas, creando DACs de pruebas, pero no se ha tenido una documentación del proceso.

Para formalizar las pruebas que sirvan como documentación de respaldo de aprobación para la liberación del proyecto se creó el [caso #105 “Create flowchart for testing worker proposals”](#). Dallas está trabajando en documentar el módulo de propuestas de trabajo para ser la base para las pruebas.

Las pruebas son la prioridad #1 para la primera semana de agosto.

## Avances no técnicos

Se agregó más información a la documentación en el wiki en <https://github.com/eosdac/documentation>:

- Se asigna una página específica centralizando la documentación sobre custodios. Se agregó una tabla de las grabaciones de las llamadas de los custodios para su fácil acceso.
- Se inició la creación de documentación por módulo del eosDAC, con el primer módulo sobre el nuevo módulo del sistema de propuestas de trabajo.

## Conclusiones

En el reporte de junio se identificó como principal debilidad, la falta de definiciones de responsabilidades específicas y horario específico del personal del equipo de desarrollo. Todo el peso sigue cayendo sobre Michael Yeates quien es responsable de los servidores, producción de bloques, actualización de la página, creación de contenido, creación del proyecto AlienWorlds, mantenimiento de las librerías de autenticación con EOS y con WAX, desarrollo, pruebas, y autorización de cambios del código del DAC client y DAC factory. Para liberar este cuello de botella se necesitan a uno o dos personas para tener avances en paralelo.

Vemos en la gráfica que la mayor cantidad de trabajo no es de desarrollo, sino de control de calidad, y la mayoría de desarrollo pendiente puede ser realizado por un programador frontend, sin experiencia en blockchain. El problema es que seguimos sin conocer los estados financieros para llenar esas plazas.

¿El mismo programador que desarrolló el sistema podría realizar el control de calidad? Esta es una pregunta típica en empresas empíricas de desarrollo de sistemas o en empresas en etapas de inicio, sin experiencia en el área de desarrollo de sistemas. Una empresa no quiere gastar más recursos en contratar alguien nuevo para validar el trabajo ya pagado. Simplemente espera que funcione.

La verdad es que cuando un programador auto valida su propio código, inconscientemente vuelve a caminar sobre sus mismas pisadas en la arena. Nos damos cuenta de esto con el ejemplo del primer DAC creado por alguien que no hizo el código del sistema. El DAC no pudo ser creado por usar una configuración distinta a la que se ha usado siempre.

El costo de no validar el trabajo por alguien más, en sistemas financieros como blockchain es muy alto. Hoy no podemos saber si podemos contratar a alguien para esa tarea pues no tenemos estados financieros formales. Es mi opinión personal que para poder avanzar, necesitamos estados financieros para poder saber si podemos contratar a alguien de control de calidad.



Stuardo Rodríguez