Tabla 1. Líneas de código efectivas por proyecto y lenguaje de programación.

Mi idea principal para realizar esta tabla era utilizar la API de GitHub y obtener los lenguajes con los que trabaja el proyecto como tal. Pero después encontré un herramienta llamada cloc. Cloc cuenta líneas en blanco, líneas de comentarios y líneas de código en diferentes lenguajes de programación.
La instalación es muy fácil https://github.com/AlDanial/cloc y haciendo una prueba eso fue un resultado que me arrojo con el proyecto de provisioning

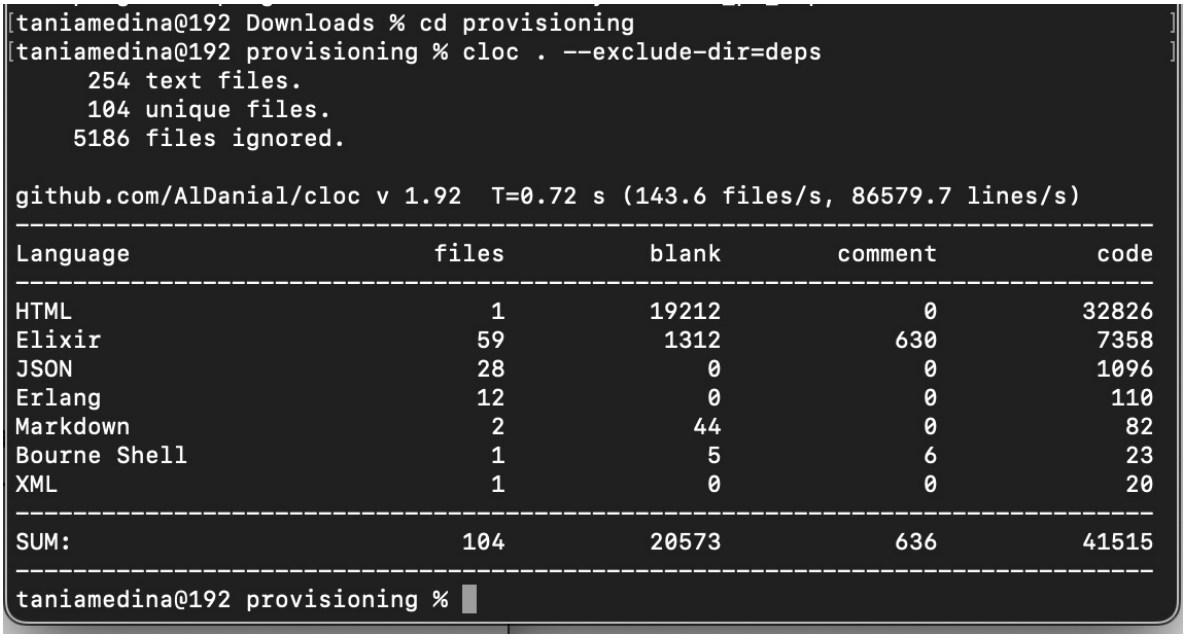

Figura 1. Grafo de dependencias entre los distintos repositorios de MachineCorpz.

Para obtener un diagrama con las dependencias del proyecto, creo que es buena idea utilizar mix deps.tree. Que nos arrojará algo parecido a la imagen de abajo. Obviamente esto se tendría que realizar con todo el proyecto como tal. Ya que en este caso solo se está realizando con provisioning.

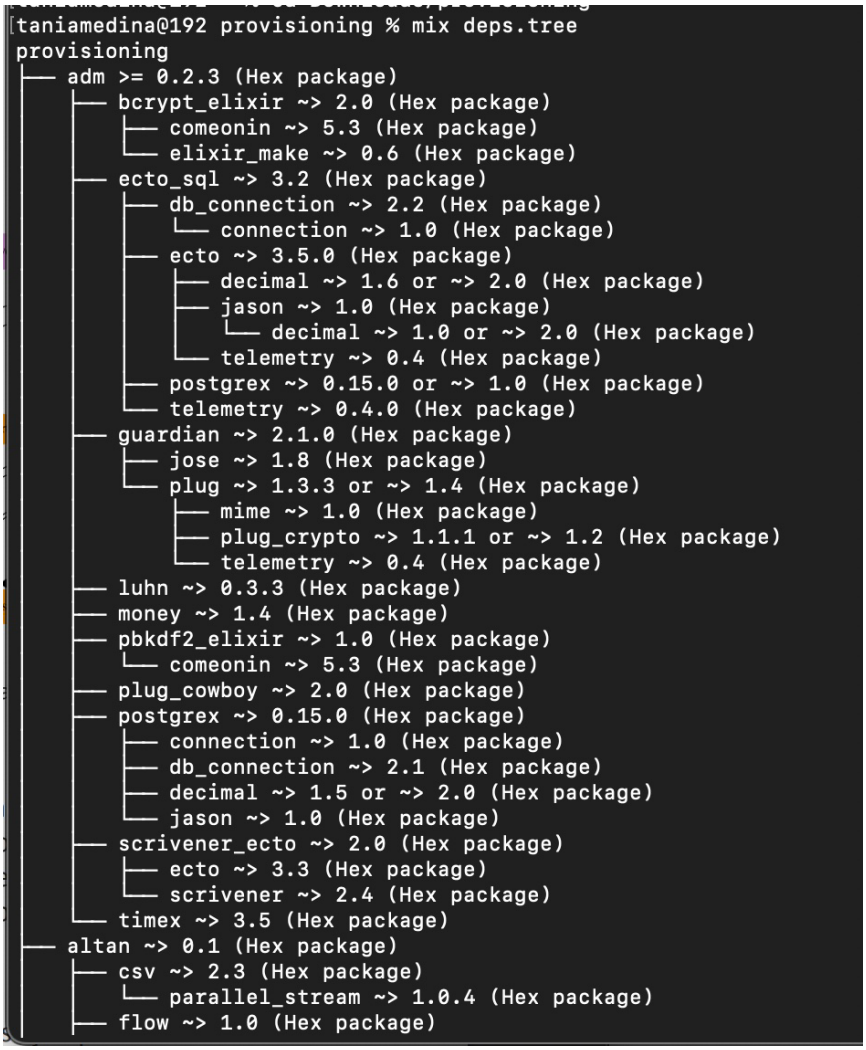

Tabla 2. Métricas de pruebas por proyecto.
Con el comando mix test se pueden obtener el total de pruebas y cuantas de esas pruebas son exitosas o cuantas fallan. Para obtener el % de cobertura se puede hacer uso de Excoveralls y tan solo con hacer mix coveralls se puede obtener esa información.

Tabla 3. Warnings de compilación por proyecto.
Con el comando mix compile se puede apreciar si tu código no tiene algún error o si presenta warnings en el código. Es muy bueno ya que te indica el porqué del error o del warning y es más fácil de corregir.

Tabla 4. Conto de warnings clasificados por tipo. Los tipos que están en rojo son potencialmente bugs

Yo creo que existe algún comando donde le puedes indicar que te muestre la cantidad de warnings dependiendo el tipo de warning que quieras consultar. De momento no lo pude

encontrar. Solo se me ocurre categorizar a mano cada warning y contarlos para llenar la tabla

Tabla 5. Hallazgos de calidad encontrados con herramientas de análisis.
Esta tabla se puede llenar con el uso del comando mix credo -a que muestra cuantos warnings encontró, así como los refactoring opportunities. Esto se tendría que hacer para cada proyecto.

Tabla 6. Líneas de acción propuestas y sus respectivos estimados.
Esta tabla en particular creo que se llena a partir de los hallazgos que se encontraron del proyecto y de las tablas anteriores asi como de la capacidad que tienen los desarrolladores de Bunsan para estimar un tiempo especifico para completar cada tarea y asignarle dicha prioridad a cada tarea.