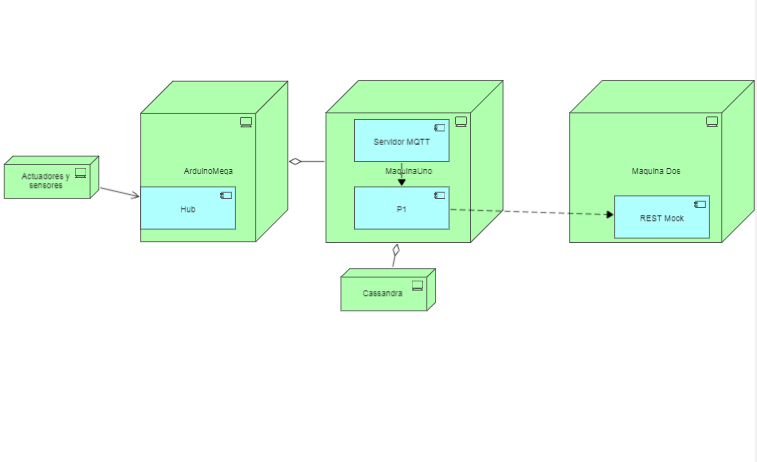
Escenario

# Detalles escenarios de Prueba

* Conjuntos: La Moneda(id=95), Nogales(id=25), Altos Buenavista(id=75)
* Aptos: id 1-20 en La Moneda, 21-75 en Nogales, 76 a 175 en Altos Buenavista
* Alarmas: Se inicia sin ellas originalmente, para que luego se llenen en pruebas.

# Diagrama de despliegue



* Arduino Mega: El habitual Arduino Mega
* Máquinas 1 y 2: CPU = 4, Memoria = 16384, Almacenamiento = 86 GB, Windows 7(x64), interno a red de universidad, máquina virtual.
* Cassandra: Donde se tiene la base de datos, en otra máquina igual a 1 y 2.

# Comparación resultados obtenidos contra esperados

Con 200 datos funciona sin el menor problema pero, predeciblemente, falla en estar con 300000. Por ende, se requieren mejoras ya que se presentan errores mayores al 50% y tiempo de respuesta de hasta 10 veces lo esperado en el caso. Se requieren cambios para lograr los escenarios de calidad propuestos.

# Reflexión diseño actual

El diseño actual es bueno para no tener aún un balanceador de carga. Pero eso no significa que sea útil para un proyecto del tamaño indicado. Por esto, y para conveniencia, se planea luego de esta entrega crear un balanceador de carga de tipo Least\_conn y agregar múltiples máquinas a la arquitectura. Estas incluyen una con base de datos adicional y redundancia fría junto a mínimo otras 2 máquinas para la capa speed de la arquitectura, dando mayor disponibilidad y disminuyendo tiempos de respuesta para intentar así cumplir los escenarios de calidad. Además de esto, se espera separar el servidor MQTT de las máquinas que trabajan el programa “P”.