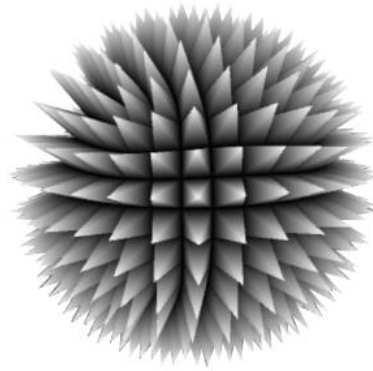


## 2. Responder los siguientes apartados (5pts):

a) Siguiendo el pipeline mostrado en la Figura 2 realizar una explicación (con los mayores detalles posibles) de como sería el transcurso de una serie de puntos (Figura 3 (a)) que atraviesan este pipeline para generar la imagen mostrada en la Figura 3 (b) (2pts).



(a) POCOS puntos iniciales de una esfera



(b) Geometría con diversas primitivas

Figura 3

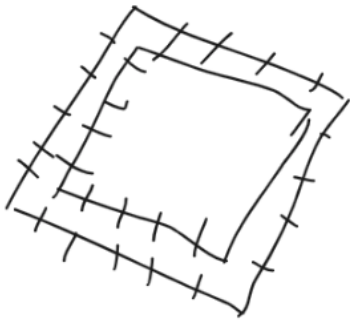
Enviamos los vertices y los indices del circulo y un PATH con 4 vertices desde el programa principal:



**Vertex Shader:** nada.

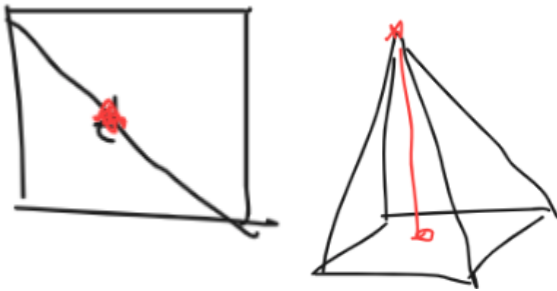
**Tessellation:**

- **Tessellation Control Shader** cada PATH estará formado por un cuadrado segun los indices y lo subdividimos en una malla de NxN puntos cada PATH para obtener mas triangulos.



- **Tessellation Evaluation Shader:** nada

**Geometry Shader:** Ingresan dos triangulos que del tesellator, se calcula el punto punto medio (color rojo) para formar un nuevo vertice, cual será la cuspide de la piramide con base formada por los dos triangulos. Calculamos las nuevas normales para iluminación, el cual será atributo de vertice de las caras creadas.



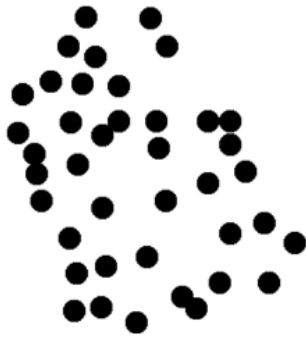
Salen 4 triangulos (12 vertices de salida)

### **Fragment Shader**

Se colorea el triangulo teniendo que la base de la piramide es color negro y la cuspide color blanco, con eso se tiene el grafico de la imagen (b).

**b) Realizar el proceso anterior teniendo en cuenta la imagen inicial: Figura 4 (a) y final:**

Figura 4 (b) (2pts).



(a) POCOS puntos iniciales de la superficie de un conejo



(b) Imagen final de un conejo "quebrantado"

Figura 4

Se observa una iluminacion.

**Vertex Shader:** nada.

**Tessellation:** Se subdivide

- **Tessellation Control Shader:**
- **Tessellation Evaluation Shader:**

**Geometry Shader:**

**Fragment Shader**

c) Tesellator también recibe el nombre de Tessellation Primitive Generator (TPG).