

Distometria

Escrito por Manuel Diego Valdearenas Martin

Domingo, 11 de Junio de 2017 20:14 - Actualizado Miércoles, 26 de Agosto de 2020 10:37

DISTOMETRIA

La distometría trata el cálculo de la potencia de una lente según su distancia a la cara corneal anterior o distancia al vértice.

Con esta función escrita en Delphi se puede calcular la potencia de una lente esferocilíndica colocada a una distancia determinada.

```
////////////////////////////////////
//////// 1      DISTOMETRIA //////////////////////////////////////
//      necesita (distancia vertice en mm o metros, el valor de la
esfera, el valor del cilindro, si DV es en mm (0) o metros (1)
(SINGLE)
//      devuelve en result[0] la esfera y en result[1] el cilindro
(SINGLE)
//      necesita una declaracion en var del procedimiento que llama:
"datos: array of single";
////////////////////////////////////
FUNCTION
FN_REFRACCION_Distometria(VAR_Distancia_Vertice,VAR_Esfera_Inicial,VAR
_Cilindro_Inicial,DV_Formato_Unidad: single):
T_FN_REFRACCION_Distometria_ARRAY;
var
    Cilindro_Calculo:      single;
begin
    SetLength(result, 2);
    if DV_Formato_Unidad = 0 then // DV en milímetros
    begin
        if VAR_Esfera_Inicial <> 0 then
        begin
            result[0] := ((1000) /
(((1000)/(VAR_Esfera_Inicial + 0.001)) - VAR_Distancia_Vertice));
        end;
        if VAR_Cilindro_Inicial <> 0 then
        begin
            result[1] := ((1000) /
(((1000) / ((VAR_Esfera_Inicial + 0.001) + (VAR_Cilindro_Inicial
+0.001))) - VAR_Distancia_Vertice)) -
((1000)/(((1000)/(VAR_Esfera_Inicial + 0.001)) -
VAR_Distancia_Vertice));
        end;
    end;
    if DV_Formato_Unidad = 1 then // DV en metros
    begin
        if VAR_Esfera_Inicial <> 0 then
        begin
```

Distometria

Escrito por Manuel Diego Valdearenas Martin

Domingo, 11 de Junio de 2017 20:14 - Actualizado Miércoles, 26 de Agosto de 2020 10:37

```
// dv metros
result[0] :=
(VAR_Esfera_Inicial) / (1 - (VAR_Distancia_Vertice *
VAR_Esfera_Inicial));
end;
if VAR_Cilindro_Inicial <> 0 then
begin
// dv metros
Cilindro_Calculo := (VAR_Esfera_Inicial) +
(VAR_Cilindro_Inicial);
result[1] := (Cilindro_Calculo)
/ (1 - (VAR_Distancia_Vertice * Cilindro_Calculo));
end;
end;
end;
```