```
module GeradorImagem (
    input wire clk25,
    input wire Hsync, Hactive,
    input wire Vsync, Vactive,
    input wire enable,

    input wire[8:0] CircleRow,
    input wire[9:0] CircleCol,

    output reg[7:0] R,G,B
);

    wire[8:0] Row;
    wire[9:0] Col;
    ContadorLinha contLinha(Hsync, Vsync, Vactive, Row);
    ContadorColuna contColuna(Hsync, clk25, Hactive, Col);

    wire [8:0] RowInferior, RowSuperior;
    wire [9:0] ColInferior, ColSuperior;

    reg[9:0] DistanceToEnd;


    assign RowInferior = CircleRow - 9'd20;
    assign RowSuperior = CircleRow + 9'd20;
    assign ColInferior = CircleCol - 10'd15;
    assign ColSuperior = CircleCol + 10'd15;




    //teste

    reg [20:0] CircleEquation;
    reg [20:0] RadiusSquared;
    reg [9:0] x_diff;
    reg [8:0] y_diff;


    localparam [9:0] RADIUS = 10'd30;

    always @(enable,CircleRow,CircleCol,Row,Col) begin
        if( enable == 1'b1) begin

            x_diff = (Col > CircleCol) ? (Col - CircleCol) : (CircleCol - Col);
            y_diff = (Row > CircleRow) ? (Row - CircleRow) : (CircleRow - Row);

            CircleEquation = (x_diff * x_diff) + (y_diff * y_diff);
            RadiusSquared = RADIUS * RADIUS;

            if (CircleEquation <= RadiusSquared) begin
                // If pixel is within the circle
                R <= 8'h00;
                G <= 8'h00;
                B <= 8'hff;
            end

            else begin
                R <= 8'hff;
                G <= 8'hff;
                B <= 8'hff;
            end
        end
        else begin
            R <= 8'h00;
            G <= 8'h00;
            B <= 8'h00;
        end
    end
endmodule
```