|  |
| --- |
|  |
| Prueba de Caja Blanca |
| ***“Reinas ESPE”*** |
|  |
|  |
| **Integrantes:**  Luca De Veintemilla  Dylan Hernández  Juan Reyes  Kevin Vargas  **Fecha 2024-25-02** |

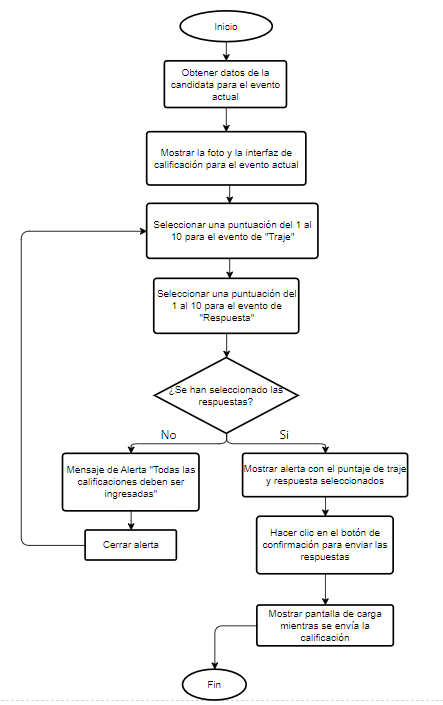
**BUSCAR PROBLEMAS**

**REQ005**

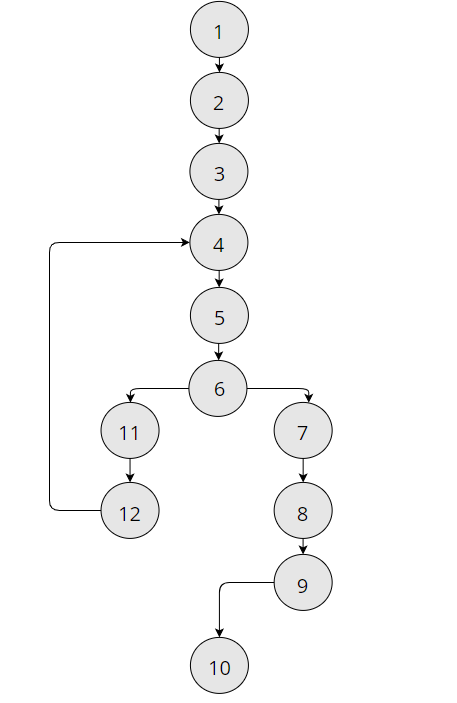
**CÓDIGO FUENTE**

|  |
| --- |
| <div class="caja\_botones">                          <div class="nota">                              <h2 class="titulo-puntuacion">Puntuación de TRAJE</h2>                              {puntuacion.map((i) => (                                  <>                                      <button                                          className={`btn-nota1 ${nota1 === i ? "click" : ""}`}                                          name="btn-nota1"                                          onClick={(e) => setNota1(i)}                                      >                                          {i}                                      </button>                                  </>                              ))}                          </div>                          <div class="nota">                              <h2 class="titulo-puntuacion">Puntuación de RESPUESTA</h2>                              {puntuacion.map((i) => (                                  <>                                      <button                                          className={`btn-nota2 ${nota2 === i ? "click" : ""}`}                                          name="btn-nota2"                                          onClick={(e) => setNota2(i)}                                      >                                          {i}                                      </button>                                  </>                              ))}                          </div>                      </div>                      <div id='enviarTG' className="enviar">                          <button type="button" className="btn-enviar" onClick={(e) => Enviar()}>                              ENVIAR                          </button>                          <Popup open={modalIsOpen} onClose={handleModalClose}>                              <div className="modal">                                  <h2 className="modal-title">¿Desea enviar las calificaciones?</h2>                                  <h2 className="modal-title">                                      Traje: {nota1} Respuesta: {nota2}                                  </h2>                                  <div className="botones-modal">                                      <button onClick={handleModalClose} className="btn-cancelar">                                          Cancelar                                      </button>                                      <button                                          onClick={() => {                                              handleModalClose();                                              handleClick1();                                          }}                                          className="btn-confirmar"                                      >                                          Aceptar                                      </button>                                  </div>                              </div>                          </Popup>                          <Popup open={popupAlerta} onClose={handlePopupAlertaClose}>                              <div className="modal">                                  <h2 className="modal-title">¡Alerta!</h2>                                  <h2 className="modal-title">Debe ingresar todas las calificaciones correspondientes antes de enviar.</h2>                                  <div className="botones-modal">                                      <button onClick={handlePopupAlertaClose} className="btn-cancelar">                                          Cerrar                                      </button>                                  </div>                              </div>                          </Popup>                      </div> |

**DIAGRAMA DE FLUJO**

****

**GRAFO**



**RUTAS**

**R1:** 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

**R2:** 1, 2, 3, 4, 5, 6, 11, 12

**COMPLEJIDAD CICLOMÁTICA**

Se puede calcular de las siguientes formas:

* V(G) = número de nodos predicados(decisiones)+1

V(G)= 1 + 1 = 1

* V(G) = A – N + 2

V(G)= 12 – 12 + 2 = 2

**DONDE:**

**P:** Número de nodos predicado

**A:** Número de aristas

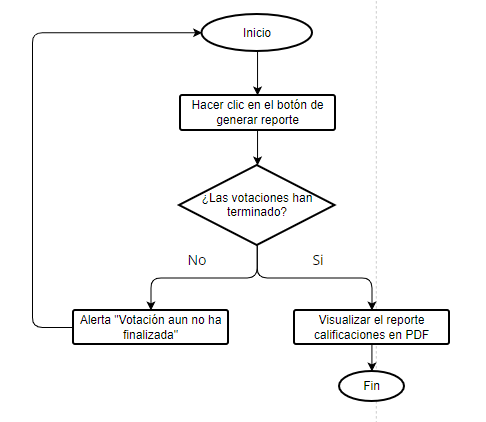
**N:** Número de nodos

**REQ006**

**CÓDIGO FUENTE**

|  |
| --- |
| <React.Fragment>      {listaReinas          .sort((a, b) => b.CAND\_NOTA\_FINAL - a.CAND\_NOTA\_FINAL)          .map((reina, index) => (              <View style={styles.contenedorColumna} key={reina.CANDIDATA\_ID}>                  <View style={styles.columna}>                      {index === 0 && (                          <Text style={styles.bold}>Reina Ganadora</Text>                      )}                      {index === 1 && (                          <Text style={styles.bold}>Candidata Confraternidad</Text>                      )}                      {index === 2 && (                          <Text style={styles.bold}>Candidata Simpatía</Text>                      )}                      {reina.ID\_ELECCION === 1 && (                          <Text style={styles.bold}>Candidata Amistad</Text>                      )}                      <Text style={styles.text}>Lugar {index + 1}: {reina.CAND\_NOMBRE1 + " " + reina.CAND\_APELLIDOPATERNO}</Text>                      <Text style={styles.text}>Puntuación Ponderada: </Text>                      <Text style={styles.text}>{(reina.CAND\_NOTA\_FINAL \* 10).toFixed(3)}/100</Text>                  </View>                  <View style={styles.columna}>                      {reina.CAND\_NOMBRE1 === "Stefany" && (                          <Image                              style={styles.imageFoto}                              src={"/reinas/" + reina.CAND\_NOMBRE2.substr(0, 1).toUpperCase() + reina.CAND\_APELLIDOPATERNO.substr(0, 1).toUpperCase() + "FX.jpg"}                          />                      )}                      {reina.CAND\_NOMBRE1 !== "Stefany" && (                          <Image                              style={styles.imageFoto}                              src={"/reinas/" + reina.CAND\_NOMBRE1.substr(0, 1).toUpperCase() + reina.CAND\_APELLIDOPATERNO.substr(0, 1).toUpperCase() + "FX.jpg"}                          />                      )}                  </View>              </View>          ))}  </React.Fragment> |

**DIAGRAMA DE FLUJO**

****

**GRAFO**

Diagrama

Descripción generada automáticamente

**RUTAS**

**R1:** 1, 2, 3, 4, 5

**R2:** 1, 2, 3, 6

**COMPLEJIDAD CICLOMÁTICA**

Se puede calcular de las siguientes formas:

* V(G) = número de nodos predicados(decisiones)+1

V(G)= 1 + 1 = 2

* V(G) = A – N + 2

V(G)= 6 – 6 + 2 = 2

**DONDE:**

**P:** Número de nodos predicado

**A:** Número de aristas

**N:** Número de nodos