

Prueba de Caja Blanca

*“Sistema Automatizado de Gestión de Check-in/out y
Limpieza para el Hotel Luxur Dream”*

Integrantes:

- Cano Intriago Juan Pablo
- Villegas Solange Alexandra
- Menéndez Chalco Tamara Gissel

Fecha 2025-07-23

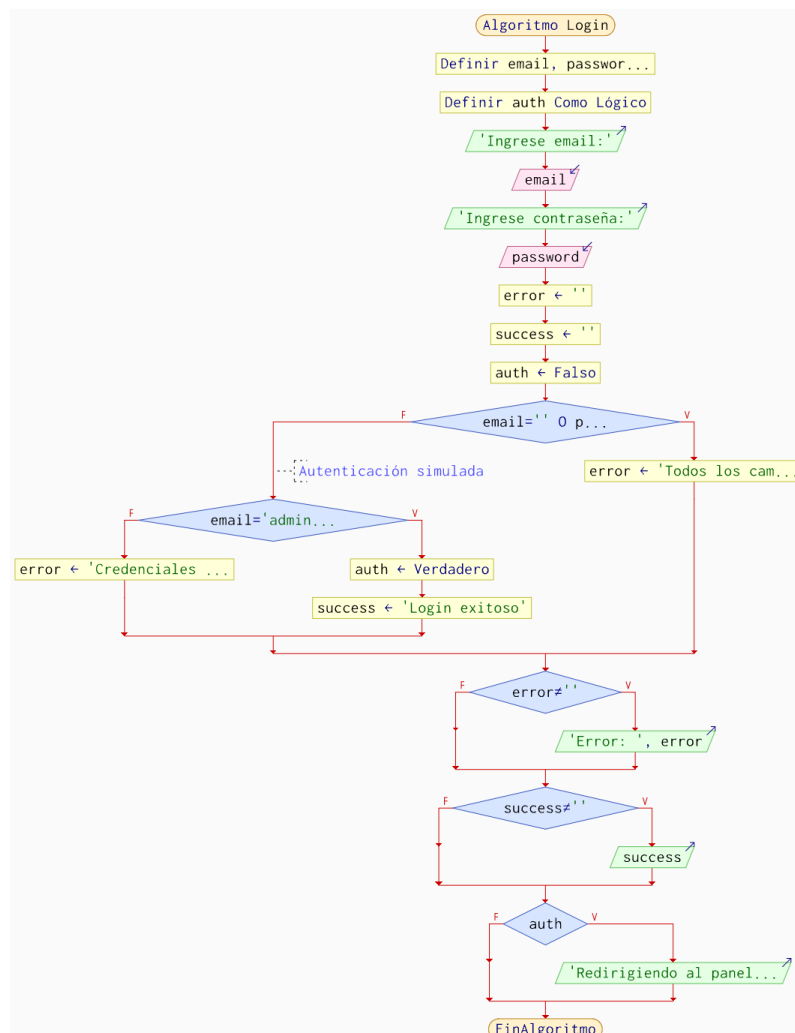
Prueba caja blanca del:

REQUISITO 001: Mantener los datos seguros y privados.

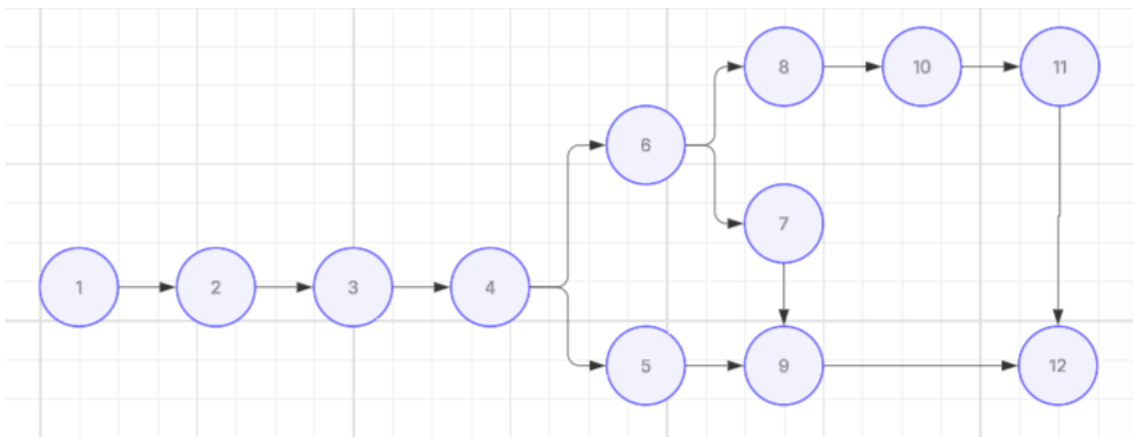
1. CÓDIGO FUENTE

```
14 const handleSubmit = async (e) => {
15   e.preventDefault();
16   setError("");
17   setSuccess("");
18   if (!email || !password) {
19     setError("Todos los campos son obligatorios.");
20     return;
21   }
22   // Aquí puedes agregar más validaciones si es necesario
23
24   try {
25     await signInWithEmailAndPassword(auth, email, password);
26     navigate("/");
27   } catch (err) {
28     setError(traducirErrorFirebase(err.message));
29   }
30 };
31
32 function traducirErrorFirebase(error) {
33   if (!error) return "";
34   if (error.includes("auth/invalid-credential") || error.includes("auth/wrong-password")) {
35     return "Correo o contraseña incorrectos.";
36   }
37   if (error.includes("auth/user-not-found")) {
38     return "Usuario no encontrado.";
39   }
40   if (error.includes("auth/too-many-requests")) {
41     return "Demasiados intentos. Intenta más tarde.";
42   }
43   // Puedes agregar más casos según tus necesidades
44   return "Ocurrió un error. Intenta de nuevo.";
45 }
```

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

RUTAS

R1: 1, 2, 3, 4, 5, 9, 12

R2: 1, 2, 3, 4, 6, 7, 9, 12

R3: 1, 2, 3, 4, 6, 8, 10, 11, 12

5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = 2 + 1$
 $V(G) = 3$
- $V(G) = 13 - 12 + 2$
 $V(G) = 3$

DONDE:

P: 02 Número de nodos predado

A: 13 Número de aristas

N: 12 Número de nodos

REQ002: Registro manual de huéspedes.

1. CÓDIGO FUENTE

```

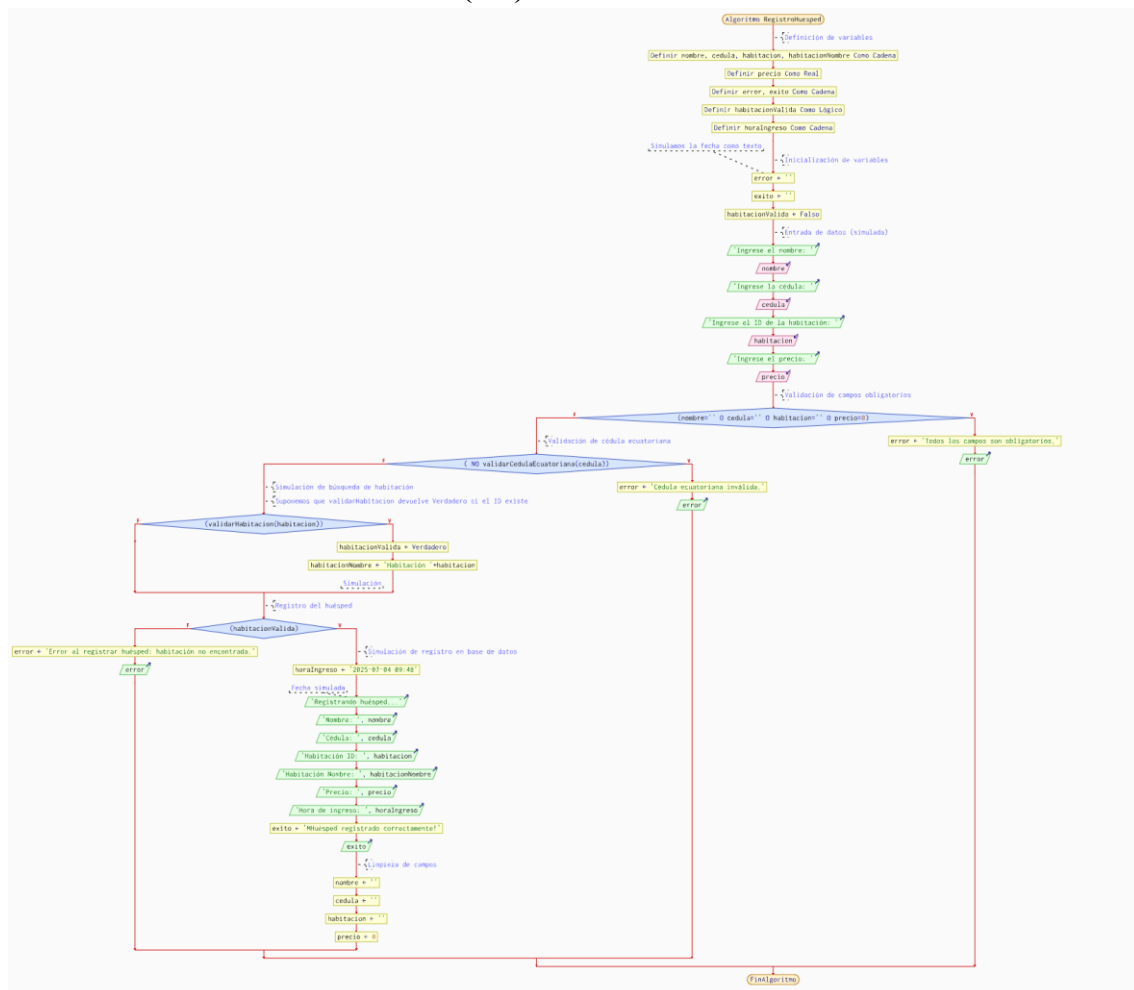
if (!nombre || !cedula || !habitacion || !precio) {
  setError("Todos los campos son obligatorios.");
  return;
}

if (!validarCedulaEcuatoriana(cedula)) {
  setError("Cédula ecuatoriana inválida.");
  return;
}

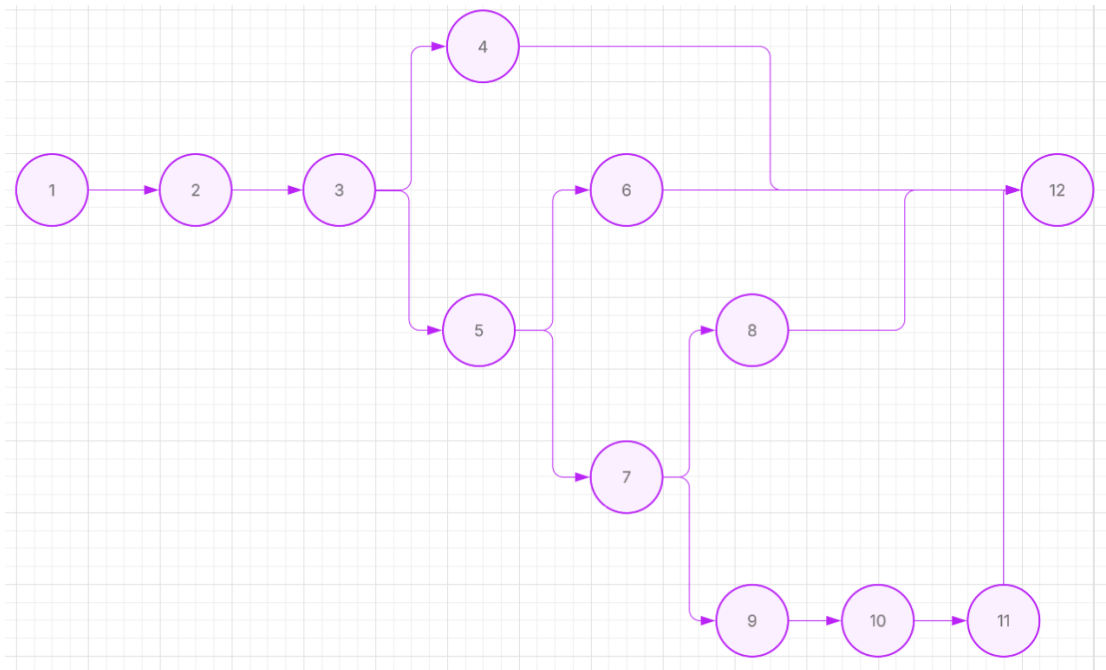
try {
  const habitacionSeleccionada = habitaciones.find(h => h.id === habitacion);
  await addDoc(collection(db, "huespedes"), {
    nombre,
    cedula,
    habitacionId: habitacion,
    habitacionNombre: habitacionSeleccionada?.nombre || "",
    precio: Number(precio),
    horaIngreso: new Date()
  });
  setNombre("");
  setCedula("");
  setHabitacion("");
  setPrecio("");
  setSuccess("¡Huésped registrado correctamente!");
} catch (err) {
  setError("Error al registrar huésped");
}

```

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

RUTAS

R1: 1, 2, 3, 4, 12

R2: 1, 2, 3, 5, 6, 12

R3: 1, 2, 3, 5, 7, 8, 12

R4: 1, 2, 3, 5, 7, 9, 10, 11, 12

5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = 3 + 1$
 $V(G) = 4$
- $V(G) = 14 - 12 + 2$
 $V(G) = 4$

DONDE:

P: 03 Número de nodos predicho

A: 14 Número de aristas

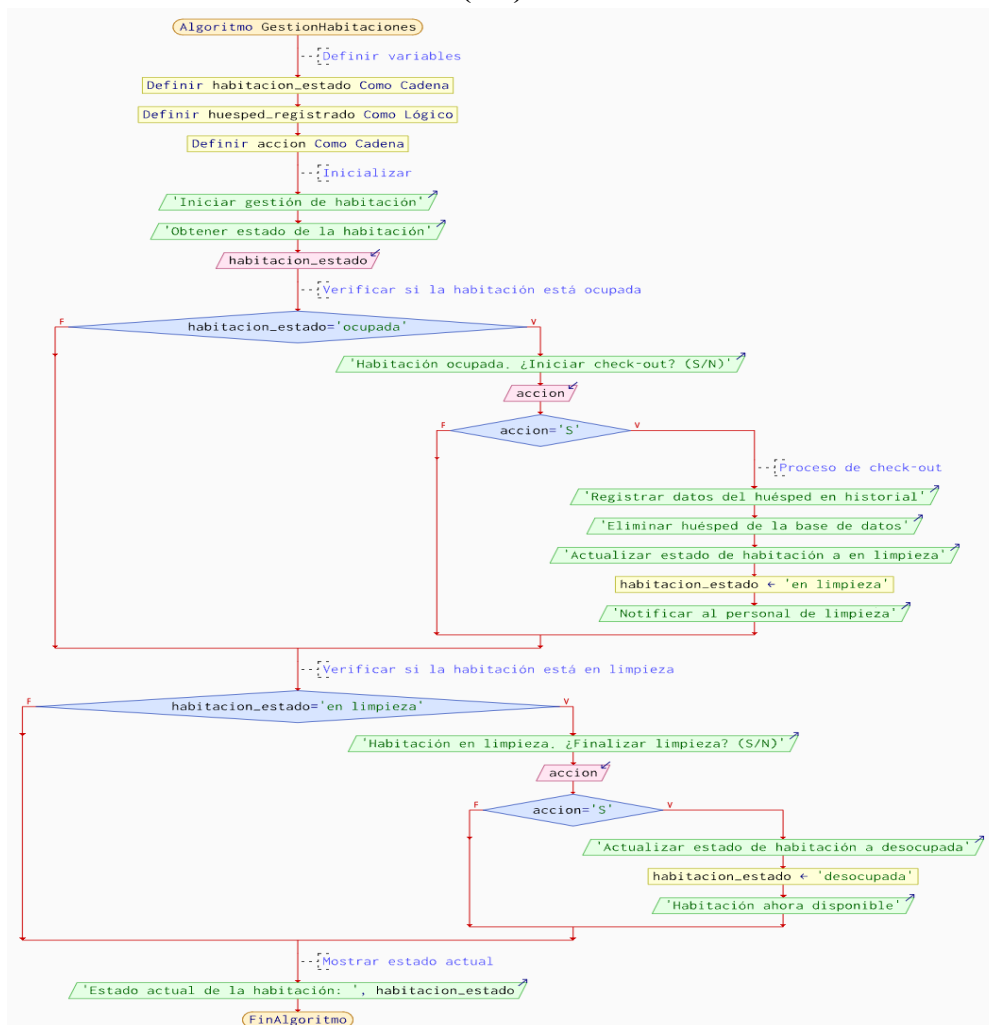
N: 12 Número de nodos

REQ003: Gestión de Habitaciones

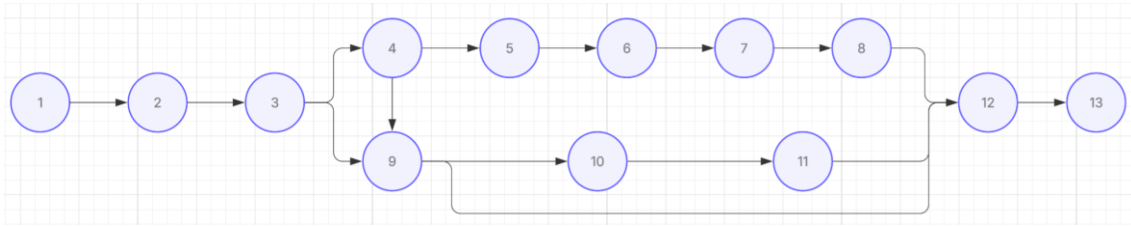
1. CÓDIGO FUENTE

```
22 const estaOcupada = (habitacionId) =>
23   hspedeses.some(h => h.habitacionId === habitacionId);
24
25 const handleCheckout = async (habitacion, hspedes) => {
26   try {
27     await addDoc(collection(db, "historial"), {
28       ...hspedes,
29       fechaSalida: new Date(),
30     });
31     await deleteDoc(doc(db, "hspedeses", hspedes.id));
32     await updateDoc(doc(db, "habitaciones", habitacion.id), {
33       estado: "en limpieza",
34     });
35     await fetchData(); // <-- AGREGA ESTO
36   } catch (err) {
37     alert("Error en check-out: " + err.message);
38   }
39 };
40
41 const handleFinLimpieza = async (habitacion) => {
42   try {
43     await updateDoc(doc(db, "habitaciones", habitacion.id), {
44       estado: "desocupada",
45     });
46     await fetchData(); // <-- AGREGA ESTO
47   } catch (err) {
48     alert("Error al finalizar limpieza: " + err.message);
49   }
50 };
51
```

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

RUTAS

R1: 1, 2, 3, 9, 12, 13

R2: 1, 2, 3, 4, 9, 12, 13

R3: 1, 2, 3, 4, 5, 6, 7, 8, 12, 13

R4: 1, 2, 3, 9, 10, 11, 12, 13

5. COMPLEJIDAD CICLOMÁTICA

- $V(G) = 3 + 1$
 $V(G) = 4$
- $V(G) = 15 - 13 + 2$
 $V(G) = 4$

DONDE:

P: 03 Número de nodos predicado

A: 15 Número de aristas

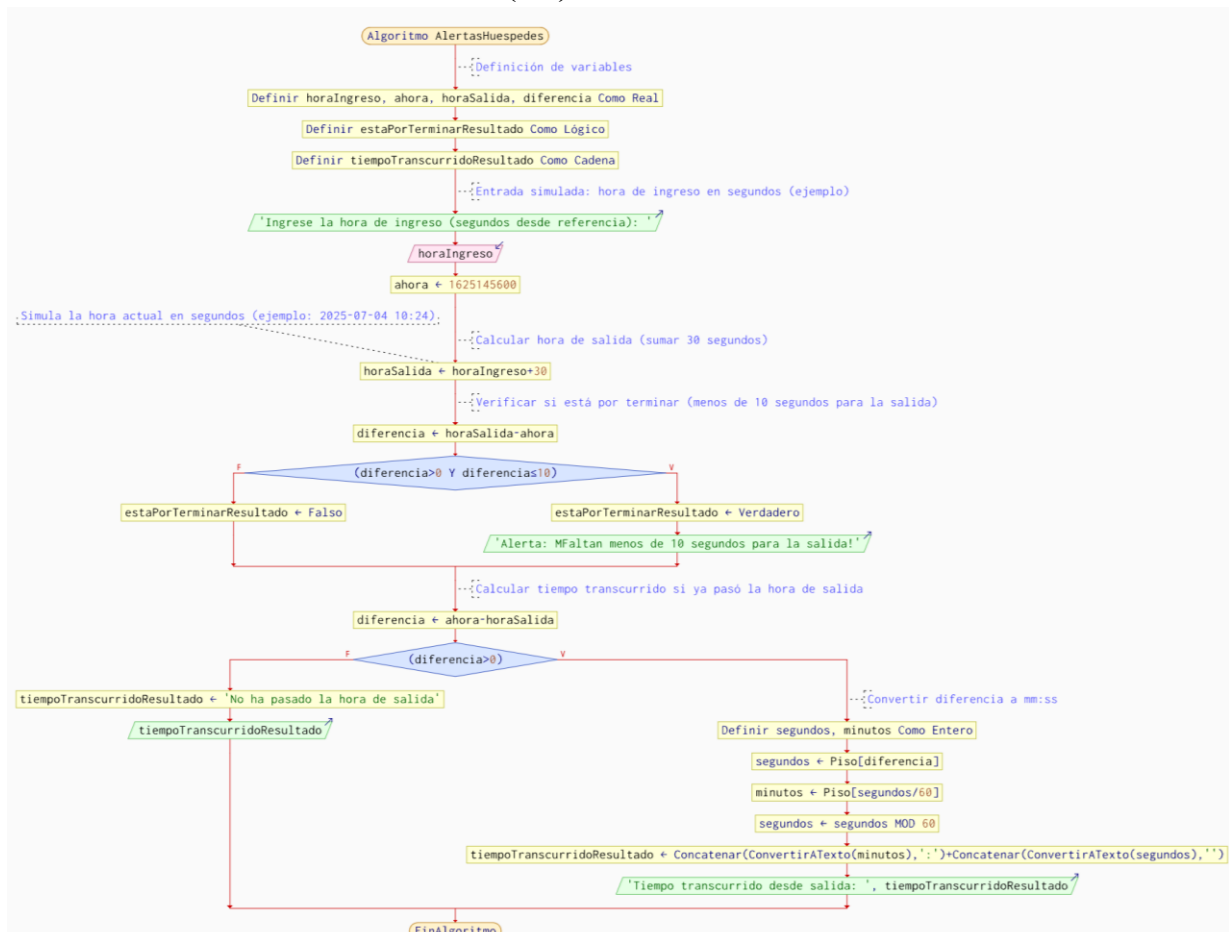
N: 13 Número de nodos

REQ004: Alertas para huéspedes que exceden la estadía.

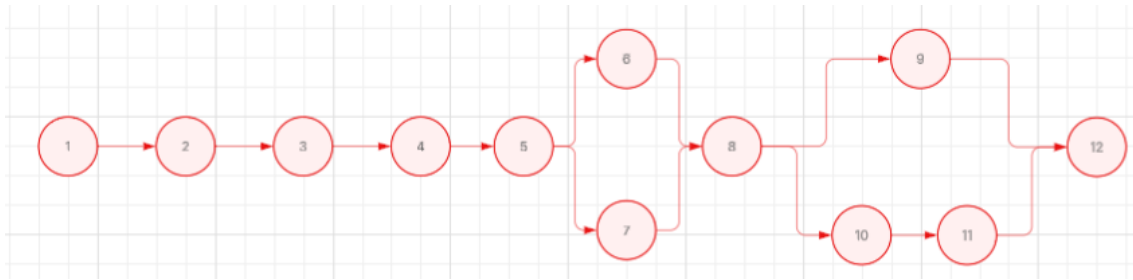
1. CÓDIGO FUENTE

```
37 const calcularHoraSalida = (horaIngreso) => {
38   if (!horaIngreso) return "";
39   let fechaIngreso;
40   if (horaIngreso.seconds) {
41     fechaIngreso = new Date(horaIngreso.seconds * 1000);
42   } else {
43     fechaIngreso = new Date(horaIngreso);
44   }
45   const fechaSalida = new Date(fechaIngreso.getTime() + 30 * 1000); // suma 30 segundos
46   return fechaSalida;
47 };
48
49 // Devuelve true si faltan menos de 10 segundos para la salida
50 const estaPorTerminar = (horaIngreso) => {
51   const salida = calcularHoraSalida(horaIngreso);
52   const ahora = new Date();
53   const diff = salida - ahora;
54   return diff > 0 && diff <= 10 * 1000; // alarma si faltan menos de 10 segundos
55 };
56
57 const tiempoTranscurrido = (horaIngreso) => {
58   const salida = calcularHoraSalida(horaIngreso);
59   const ahora = new Date();
60   const diff = ahora - salida;
61   if (diff <= 0) return null; // Solo mostrar si ya pasó la hora de salida
62
63   // Formatea a mm:ss
64   const segundos = Math.floor(diff / 1000);
65   const min = Math.floor(segundos / 60);
66   const seg = segundos % 60;
67   return `${min.toString().padStart(2, "0")}:${seg.toString().padStart(2, "0")}`;
68 };
69
```

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

RUTAS

R1: 1, 2, 3, 4, 5, 7, 8, 10, 11, 12

R2: 1, 2, 3, 4, 5, 7, 8, 9, 12

R3: 1, 2, 3, 4, 5, 6, 8, 10, 11, 12

5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = 2 + 1$
 $V(G) = 3$
- $V(G) = 13 - 12 + 2$
 $V(G) = 3$

DONDE:

P: 02 Número de nodos predicado

A: 13 Número de aristas

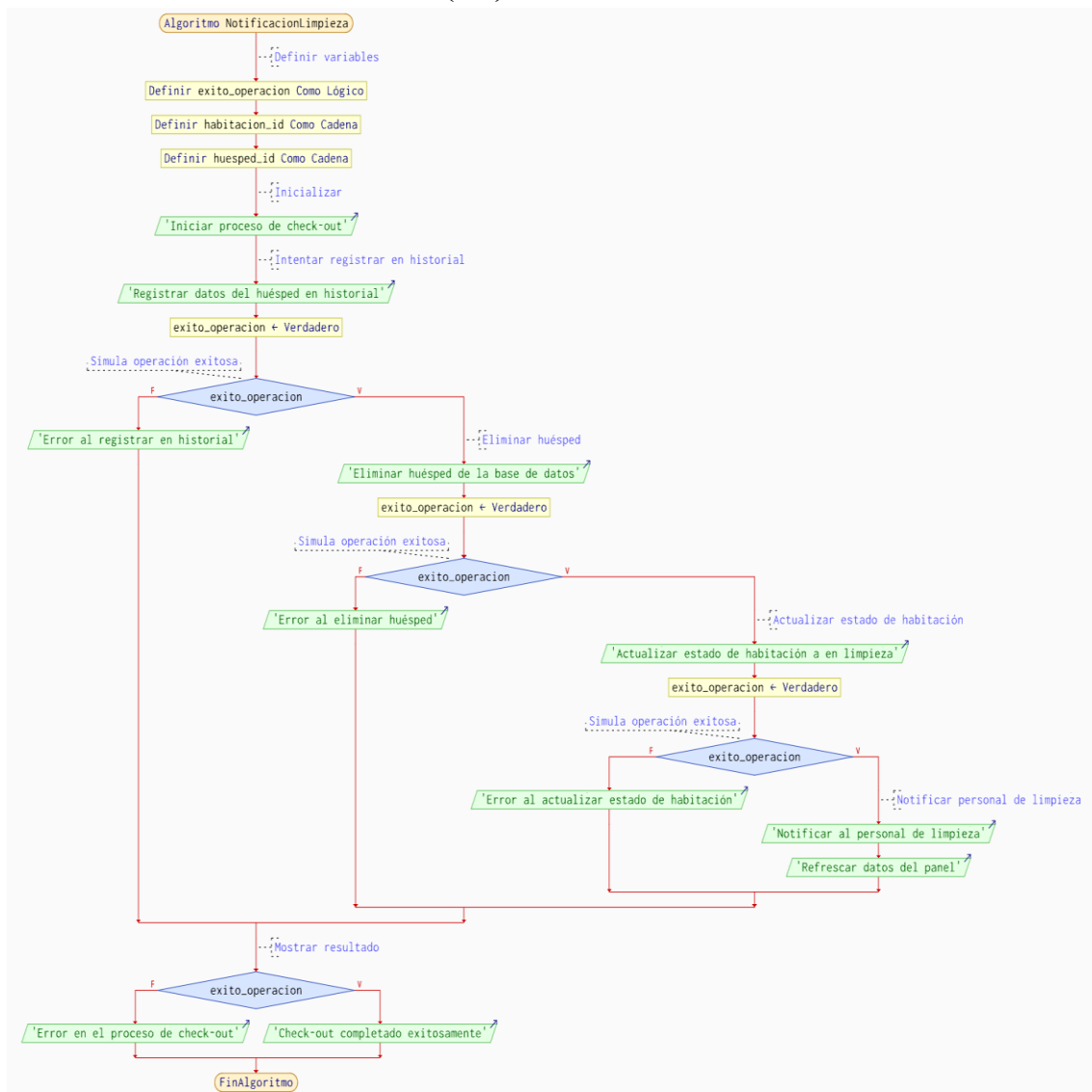
N: 12 Número de nodos

REQ005: Notificación de Limpieza

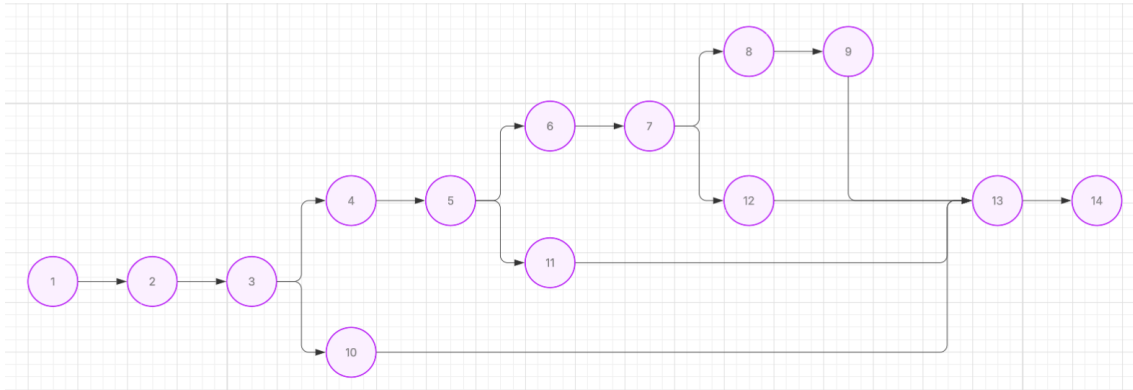
1. CÓDIGO FUENTE

```
25   const handleCheckout = async (habitacion, huesped) => {
26     try {
27       await addDoc(collection(db, "historial"), {
28         ...huesped,
29         fechaSalida: new Date(),
30       });
31       await deleteDoc(doc(db, "huespedes", huesped.id));
32       await updateDoc(doc(db, "habitaciones", habitacion.id), {
33         estado: "en limpieza",
34       });
35       await fetchData(); // <-- AGREGA ESTO
36     } catch (err) {
37       alert("Error en check-out: " + err.message);
38     }
39   };
```

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

RUTAS

R1: 1, 2, 3, 10, 13, 14

R2: 1, 2, 3, 4, 5, 11, 13, 14

R3: 1, 2, 3, 4, 5, 6, 7, 12, 13, 14

R4: 1, 2, 3, 4, 5, 6, 7, 8, 9, 13, 14

5. COMPLEJIDAD CICLOMÁTICA

- $V(G) = 3 + 1$
 $V(G) = 4$
- $V(G) = 16 - 14 + 2$
 $V(G) = 4$

DONDE:

P: 03 Número de nodos predicado

A: 16 Número de aristas

N: 14 Número de nodos

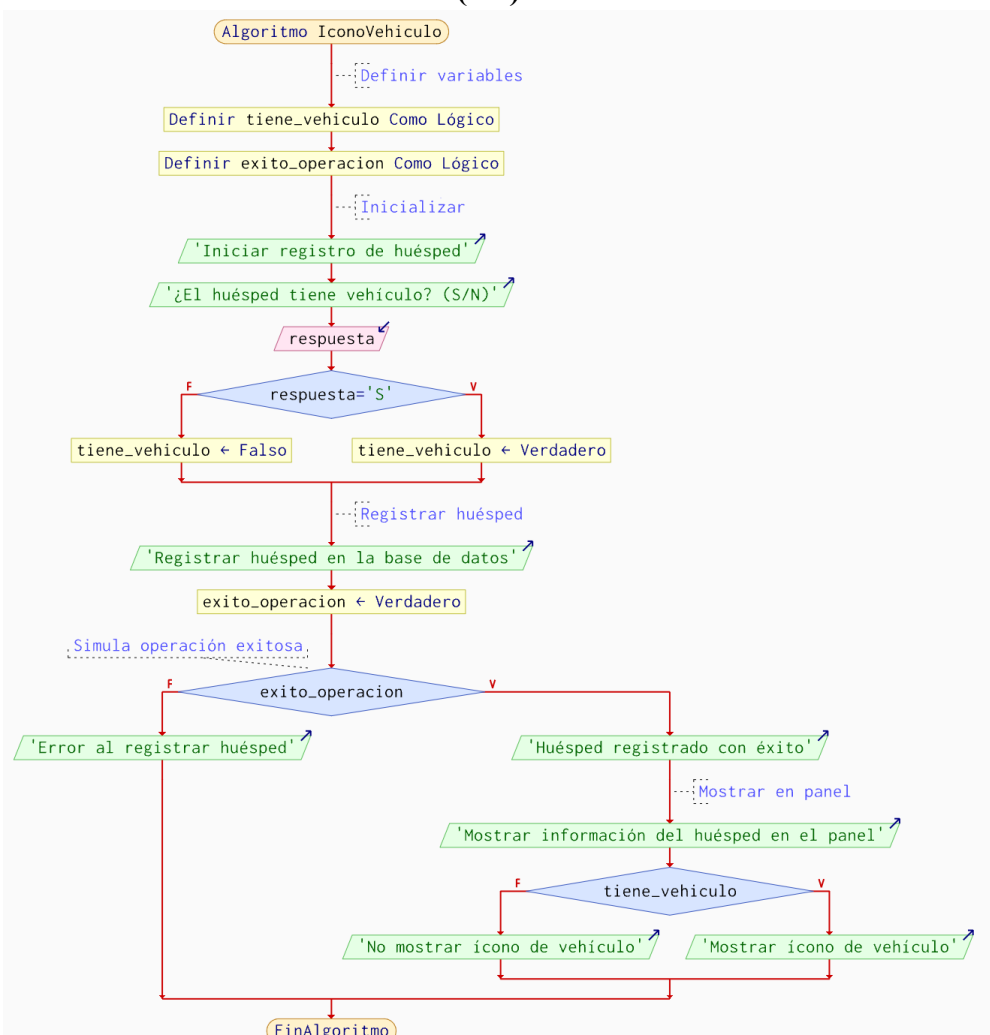
REQ006: Registro de vehículos

1. CÓDIGO FUENTE

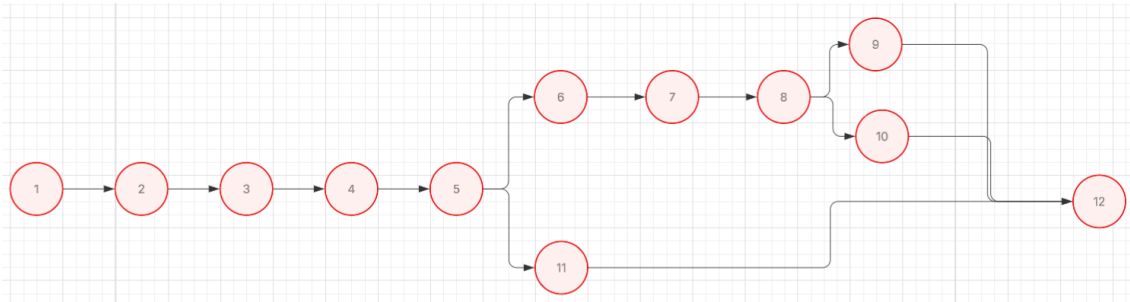
```
16   const [tieneVehiculo, setTieneVehiculo] = useState(false);
153   <label style={{ marginBottom: "1rem" }}>
154     <input
155       type="checkbox"
156       checked={tieneVehiculo}
157       onChange={e => setTieneVehiculo(e.target.checked)}
158       style={{ marginRight: "8px" }}
159     />
160     ¿El huésped tiene vehículo?
161   </label>
56   const handleSubmit = async (e) => {
68     try {
85       await addDoc(collection(db, "huespedes"), {
86         nombre,
87         cedula,
88         habitacionId: habitacion,
89         habitacionNombre: habitacionSeleccionada?.nombre || "",
90         precio: Number(precio),
91         horaIngreso: new Date(),
92         tieneVehiculo
93       });
133       {huespedOcupando.tieneVehiculo && (
134         <span title="Tiene vehículo" style={{ color: "#22c55e", fontSize: "1.3em" }}>
135           
136         </span>
137       )}

```

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

RUTAS

R1: 1, 2, 3, 4, 5, 11, 12

R2: 1, 2, 3, 4, 5, 6, 7, 8, 9, 12

R3: 1, 2, 3, 4, 5, 6, 7, 8, 10, 12

5. COMPLEJIDAD CICLOMÁTICA

- $V(G) = 2 + 1$
 $V(G) = 3$
- $V(G) = 13 - 12 + 2$
 $V(G) = 3$

DONDE:

P: 02 Número de nodos prediado

A: 13 Número de aristas

N: 12 Número de nodos