

Proyecto:

Plataforma de Registro y Asistencia en la Nube (Cloud Attendance Lite)

Fase

Fase 3 - Pruebas y Validación

Equipo

Grupo SN 10

Integrantes:

Nombre	Rol
Jonathan Ernesto Beltran Guerra	Lider Jr.
Jorge Eduardo Francisco Lopez Cabrera	Analista de Procesos Académicos Jr.
Alexander Enrique Escobar Ortíz	Desarrollador/a Fullstack Jr.
Daniel Alberto Perez	Ingeniero/a Cloud Jr
Alisson Lisbeth Serpas Martinez	QA/Documentador Técnico Jr.

Fecha de Entrega: 15 de Febrero de 2026

Resumen Ejecutivo

La Fase 3 del proyecto Cloud Attendance Lite representó un salto cualitativo en la madurez de la plataforma. El sistema evolucionó desde un prototipo basado exclusivamente en Google Forms hacia una aplicación web completa con tres vistas HTML diferenciadas, autenticación reforzada mediante Firebase Authentication con verificación en dos pasos, y un sistema dual de registro que combina un Google Form general con un formulario HTML exclusivo para participantes previamente registrados.

La arquitectura resultante separa claramente los roles de usuario. Los participantes acceden al registro de asistencia sin necesidad de autenticarse, ya sea a través de un Google Form general abierto o de un formulario HTML exclusivo al que llegan mediante una invitación del organizador. Los administradores, en cambio, ingresan al panel de gestión a través de un login protegido con doble factor de autenticación. Esta separación permitió ofrecer una experiencia optimizada para cada tipo de usuario, eliminando fricciones innecesarias para los participantes y reforzando la seguridad para quienes gestionan datos sensibles.

En el plano técnico, el backend en Google Apps Script fue reestructurado por completo, incorporando un sistema de enrutamiento que sirve distintas vistas según el contexto de acceso, funciones de generación de códigos QR para eventos, detección de registros duplicados, confirmaciones automáticas por correo electrónico con diseño profesional, reportes programados y un mecanismo de respaldo automático previo a cualquier operación destructiva sobre los datos.

Introducción

Durante la Fase 2 se validó la viabilidad técnica de Cloud Attendance Lite como solución ligera en la nube para el registro de asistencia en eventos académicos. El prototipo funcional demostró que la combinación de Google Forms, Google Sheets y Apps Script era capaz de capturar, almacenar y procesar datos de asistencia de manera básica. Sin embargo, al operar el sistema en condiciones más cercanas a un entorno real, se identificaron limitaciones importantes.

No existe un panel de administración para los organizadores de eventos, quienes debían consultar directamente Google Sheets para revisar los datos de asistencia. Tampoco había un sistema de autenticación que protegiera las funciones administrativas, ya que el control de acceso depende exclusivamente de los permisos de compartición de la hoja de cálculo. No se contaba con un mecanismo para que el organizador pudiera enviar invitaciones dirigidas a participantes ya registrados, ni había detección de registros duplicados, confirmaciones automáticas por correo, ni conexión directa entre la interfaz del sistema y los reportes de Looker Studio.

Ante este diagnóstico, la Fase 3 se orientó a implementar mejoras técnicas que abordarán cada una de estas carencias. El trabajo se estructuró en torno a cuatro ejes: seguridad administrativa, experiencia del organizador, flujo de registro diferenciado y automatización del backend. Este documento describe el desarrollo técnico realizado, las decisiones de diseño adoptadas y las mejoras concretas que cada componente aporta al sistema.

Desarrollo Técnico

El desarrollo de la Fase 3 se estructuró en seis componentes interrelacionados: la arquitectura de enrutamiento de la aplicación web, el sistema de autenticación administrativa, el panel de administración, el formulario de registro por código QR, la integración con Looker Studio y la reestructuración integral del backend.

Arquitectura de Enrutamiento

La aplicación web se construyó sobre un sistema de enrutamiento centralizado que permite servir tres interfaces completamente distintas desde una única URL de despliegue de Apps Script. Cuando un usuario accede al sistema, la función principal es evaluar los parámetros presentes en la URL para determinar qué vista mostrar. Si la URL contiene un parámetro de evento (como ocurre al escanear un código QR), se presenta el formulario de registro de asistencia con la información del evento precargada. Si contiene un indicador de acceso administrativo, se carga el panel de administración. En cualquier otro caso, incluyendo el acceso directo a la URL sin parámetros, se muestra la página de inicio de sesión como vista predeterminada.

Este diseño resolvió la necesidad de manejar múltiples interfaces desde un solo punto de entrada. En la Fase 2 no existía capacidad de servir diferentes páginas HTML, lo que hubiera obligado a crear despliegues independientes para cada funcionalidad. Con el enrutamiento centralizado, una sola URL se adapta inteligentemente al contexto del acceso, lo que simplifica el mantenimiento y garantiza que cualquier acceso no autorizado sea redirigido al login. Para los participantes, el beneficio es inmediato: al escanear un código QR, la interfaz de registro aparece directamente sin pasos intermedios.

Sistema de Autenticación Administrativa (Login.html)

Se desarrolló una página de inicio de sesión exclusiva para los administradores del sistema, es decir, los profesores u organizadores de eventos designados por el coordinador de la DGAD. Los participantes o alumnos no interactúan con esta vista en ningún momento; su acceso al sistema se realiza directamente a través de los formularios de registro sin necesidad de autenticarse.



El flujo de autenticación opera en dos etapas consecutivas. En la primera, el administrador ingresa su correo electrónico y contraseña, que se validan contra Firebase Authentication. Si las credenciales son correctas, el sistema cierra inmediatamente la sesión temporal para evitar un acceso prematuro al panel y genera un código numérico aleatorio de seis dígitos. Este código se envía automáticamente al correo electrónico del usuario a través de Apps Script, con un diseño profesional que incluye instrucciones claras. En la segunda etapa, el usuario ingresa el código recibido en un formulario de verificación que muestra un temporizador con los cinco minutos de validez del código y permite un máximo de tres intentos. Solo tras verificar el código correctamente, el sistema ejecuta el inicio de sesión definitivo en Firebase y redirige al panel de administración.



j9918@gmail.com
para mí

mié, 11 feb, 6:16 p.m. (hace 19 horas)

10 de 10,747

Código de Verificación

Sistema de Administración DGAD

Hola alisonserpas,

Has solicitado acceso al Panel de Administración.
Usa el siguiente código para completar tu inicio de sesión:

5 9 4 5 2 7

Importante:

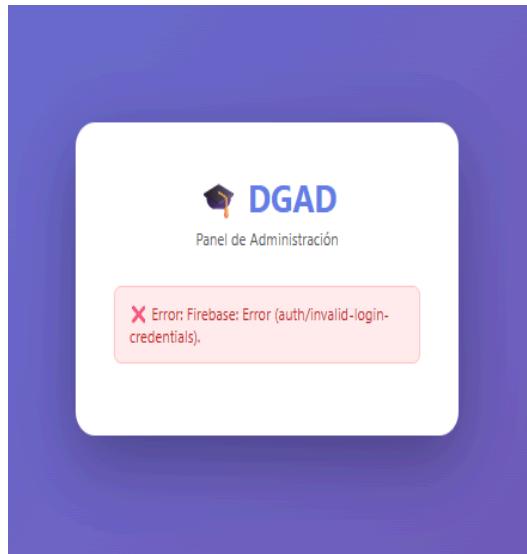
- Este código es válido por 5 minutos
- No compartas este código con nadie
- Si no solicitaste este código, ignora este mensaje

Sistema de Asistencia DGAD - Mensaje Automático

Responder Reenviar

La interfaz incluye también un mecanismo de recuperación de contraseña integrado que utiliza el servicio nativo de Firebase para enviar enlaces de restablecimiento. No existe opción de crear nuevos usuarios desde esta pantalla ni desde ninguna otra interfaz del sistema; únicamente el coordinador tiene la facultad de gestionar las cuentas de acceso desde la consola de Firebase. Adicionalmente, el backend mantiene una lista de correos autorizados como administradores como capa de verificación adicional a nivel de servidor.

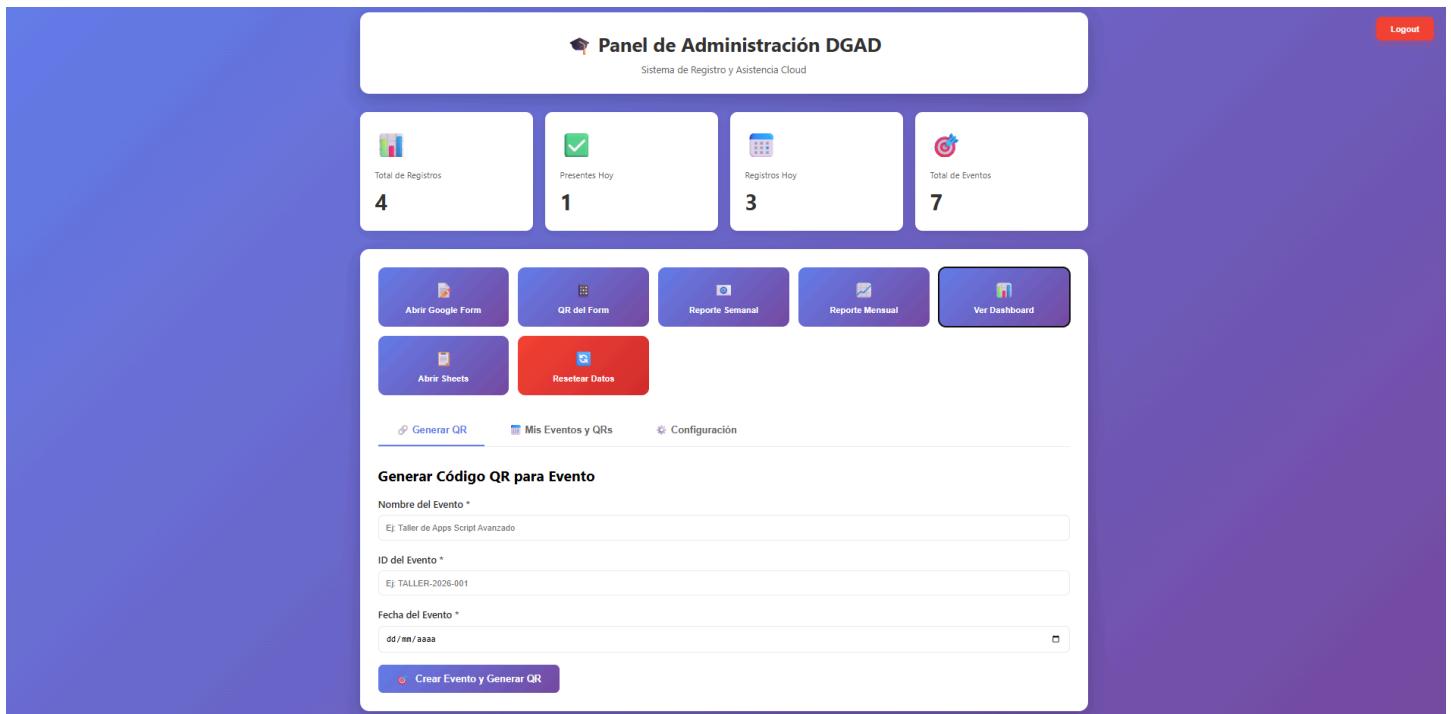
Cuando se ingresan credenciales inválidas, el sistema presenta mensajes de error específicos que informan al usuario exactamente qué ocurrió, como se muestra en la siguiente figura.



Este componente resolvió la ausencia total de autenticación para las funciones administrativas. En la Fase 2, el acceso a los datos depende exclusivamente de los permisos de compartición de Google Sheets, lo que significaba que cualquier persona con acceso a la hoja podía ver y potencialmente modificar toda la información de asistencia. La doble verificación mediante Firebase más código OTP establece una barrera robusta contra el acceso no autorizado, incluso en el caso hipotético de que las credenciales sean comprometidas, ya que el atacante también necesitaría acceso al correo electrónico del administrador. Firebase fue seleccionado por su integración nativa con el ecosistema Google y su plan gratuito que cubre las necesidades del proyecto sin generar costos adicionales..

Panel de Administración (PanelAdmin.html)

Se creó un panel de administración como interfaz centralizada para los organizadores de eventos. Al cargarse, el panel verifica que exista una sesión activa y autenticada en Firebase; si no la detecta o si el correo del usuario no ha sido verificado, redirige automáticamente a la página de login. Esta verificación se ejecuta cada vez que se accede al panel, garantizando que no sea posible acceder a la interfaz administrativa sin haber completado el proceso de autenticación.



La vista principal presenta un tablero de estadísticas en tiempo real con cuatro indicadores: el total acumulado de registros de asistencia, la cantidad de participantes marcados como presentes en el día actual, el número de registros del día y el total de eventos creados en el sistema. Estos datos se calculan directamente desde la hoja de cálculo y se actualizan cada vez que el organizador accede al panel.

La funcionalidad central del panel es la gestión de eventos y la generación de códigos QR. El organizador completa un formulario con el nombre del evento, un identificador único y la fecha. Al enviarlo, el sistema crea el evento en el catálogo de la hoja de cálculo, genera una URL única que contiene los parámetros del evento, y produce un código QR a través de una API externa de generación de códigos. La URL del QR se almacena junto al evento para consultas posteriores. Una vez generado, el organizador puede compartir la invitación a través de múltiples canales: WhatsApp, Telegram, correo electrónico, o simplemente copiando la URL directa.

Generar Código QR para Evento

Nombre del Evento *

Taller de Servicios Cloud

ID del Evento *

TALLER-2026

Fecha del Evento *

11 / 02 / 2026

 Crear Evento y Generar QR

✓ Evento Creado y QR Generado

Taller de Servicios Cloud

ID: TALLER-2026



URL del evento:
https://script.google.com/macros/s/AKfycbwN1H14sK8_UbwJ5pske3cFTp9lF8TC2hXo1bw8tcwJq-zpbphg1116kOTKD6kpCd4/exec?evento=TALLER-2026&fecha=2026-02-11&nombree=ta1er%20de%20Servicios%20Cloud

 Descargar QR

 Copiar URL

 WhatsApp

 Email

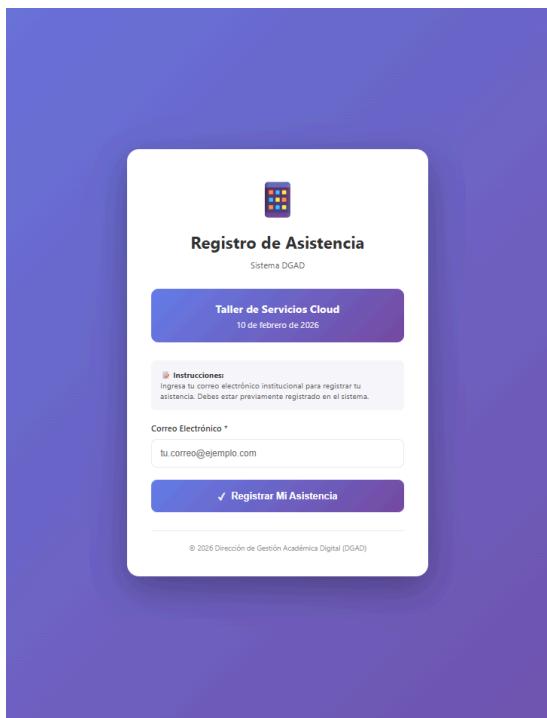
 Telegram

El panel incorpora además botones de acción rápida para generar y enviar reportes semanales y mensuales al coordinador, acceder directamente a los dashboards de Looker Studio, abrir la hoja de cálculo y realizar un reseteo de datos con respaldo automático obligatorio que requiere doble confirmación. El cierre de sesión se realiza desde un botón fijo en la esquina superior de la pantalla, que termina la sesión de Firebase y redirige al login.

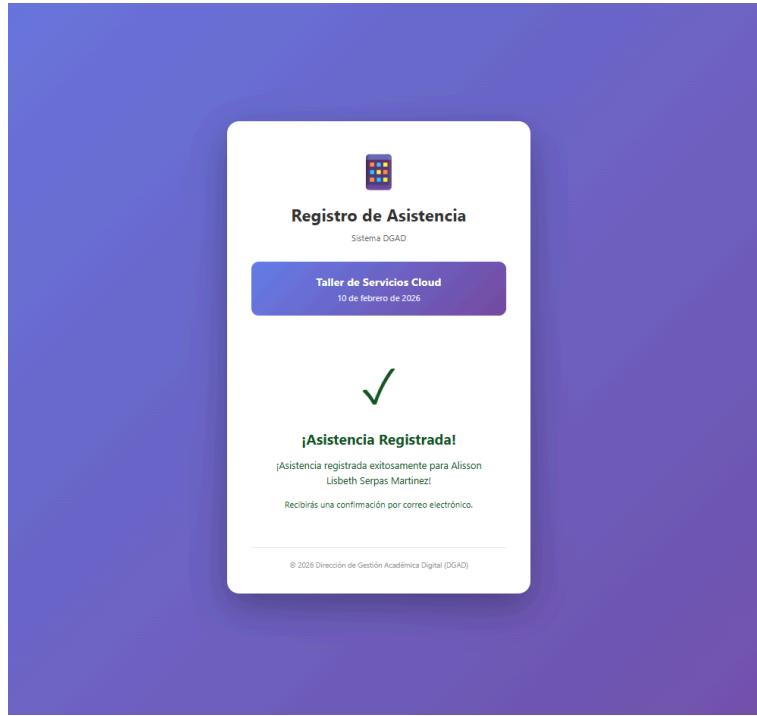
Este componente resolvió la falta de una interfaz administrativa unificada. Anteriormente, el profesor debía alternar entre Google Sheets para consultar datos, correo electrónico para comunicarse con participantes, herramientas externas para generar códigos QR y Looker Studio para revisar reportes. El panel centralizó todas estas operaciones en una sola vista intuitiva, eliminando la necesidad de conocimientos técnicos y reduciendo significativamente el tiempo requerido para gestionar eventos.

Formulario de Registro por Código QR (FormularioQR.html)

Se desarrolló un formulario HTML exclusivo para registrar la asistencia de participantes que ya se encuentran en la base de datos del sistema. Este formulario se presenta automáticamente cuando un participante escanea un código QR generado desde el panel de administración o accede al enlace de invitación correspondiente. La URL que llega al participante contiene los parámetros del evento, por lo que la información aparece precargada en la interfaz sin necesidad de que el participante la complete manualmente.



El formulario solicita únicamente el correo electrónico del participante. Al enviarlo, el sistema ejecuta tres validaciones secuenciales en el servidor. Primero, busca el correo en el catálogo de participantes para confirmar que el usuario está registrado; si no lo encuentra, muestra un mensaje indicando que debe registrarse previamente a través del formulario general. Segundo, verifica si ya existe un registro de asistencia para el mismo correo, evento y fecha del día, previniendo duplicados. Tercero, si ambas validaciones pasan, registra la asistencia con todos los datos correspondientes, marca el estado como presente, y envía automáticamente un correo de confirmación al participante.



Este componente resolvió varios problemas críticos del sistema anterior. En la Fase 2, el único medio de registro era el Google Form general, que no distingue entre participantes nuevos y registrados ni verifica la identidad contra la base de datos existente. No existe forma de prevenir registros duplicados ni de enviar invitaciones dirigidas a participantes específicos. El formulario QR introdujo la verificación de identidad contra el catálogo, la detección de duplicados a nivel de servidor, la automatización del registro con datos precargados desde el QR, y la trazabilidad completa al marcar cada registro con su fuente de origen.

Google Form General

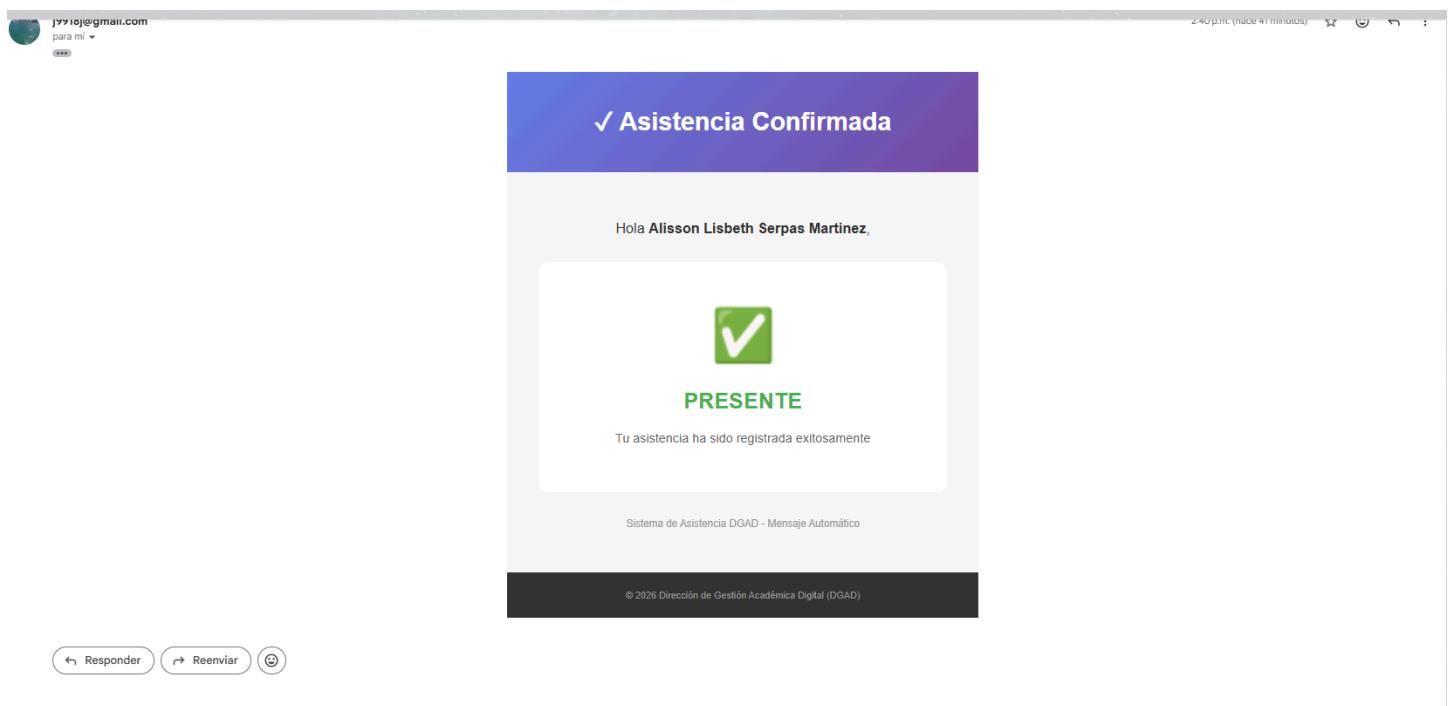
Se mantuvo y actualizó el Google Form general como canal de registro abierto a cualquier persona, sin requerir autenticación más allá de una cuenta de Google. Este formulario está dirigido principalmente a participantes nuevos que asisten por primera vez a un evento y aún no están registrados en la base de datos, aunque también puede ser utilizado por participantes existentes que prefieran este medio.

The screenshot shows the initial section of a Google Form. At the top, it says "Registro de Asistencia Digital – DGAD v3". Below that, there is a paragraph of text explaining the purpose of the form: "Este formulario tiene como objetivo registrar la asistencia de los participantes a actividades académicas y eventos institucionales organizados por la Dirección de Gestión Académica Digital (DGAD). La información recopilada será almacenada de forma segura y utilizada exclusivamente para fines de control académico, generación de reportes estadísticos y análisis de participación. Al enviar este formulario, el participante recibirá una confirmación automática por correo electrónico." It also includes an email address "alisonserpas@gmail.com" and a "Cambiar cuenta" link. A note at the bottom states "* Indica que la pregunta es obligatoria".

Below this, there are five input fields with labels: "Nombre *", "ID/Código *", "Correo *", "Tipo de Evento *", and "Fecha *". Each field has a placeholder "Tu respuesta" and a small "Tu respuesta" label below it.

This screenshot shows the continuation of the Google Form. It contains three input fields: "Hora *", "Grupo/Sección *", and "Observaciones". Each field has a placeholder "Tu respuesta" and a small "Tu respuesta" label below it. At the bottom of the form, there are two buttons: a red "Enviar" button and a smaller "Borrar formulario" button. Below the buttons, there is a note: "Nunca envíes contraseñas a través de Formularios de Google." and links to "Comunicarse con el propietario del formulario - Condiciones del Servicio - Política de Privacidad". At the very bottom, it says "¿El formulario parece sospechoso? Informe" and "Google Formularios".

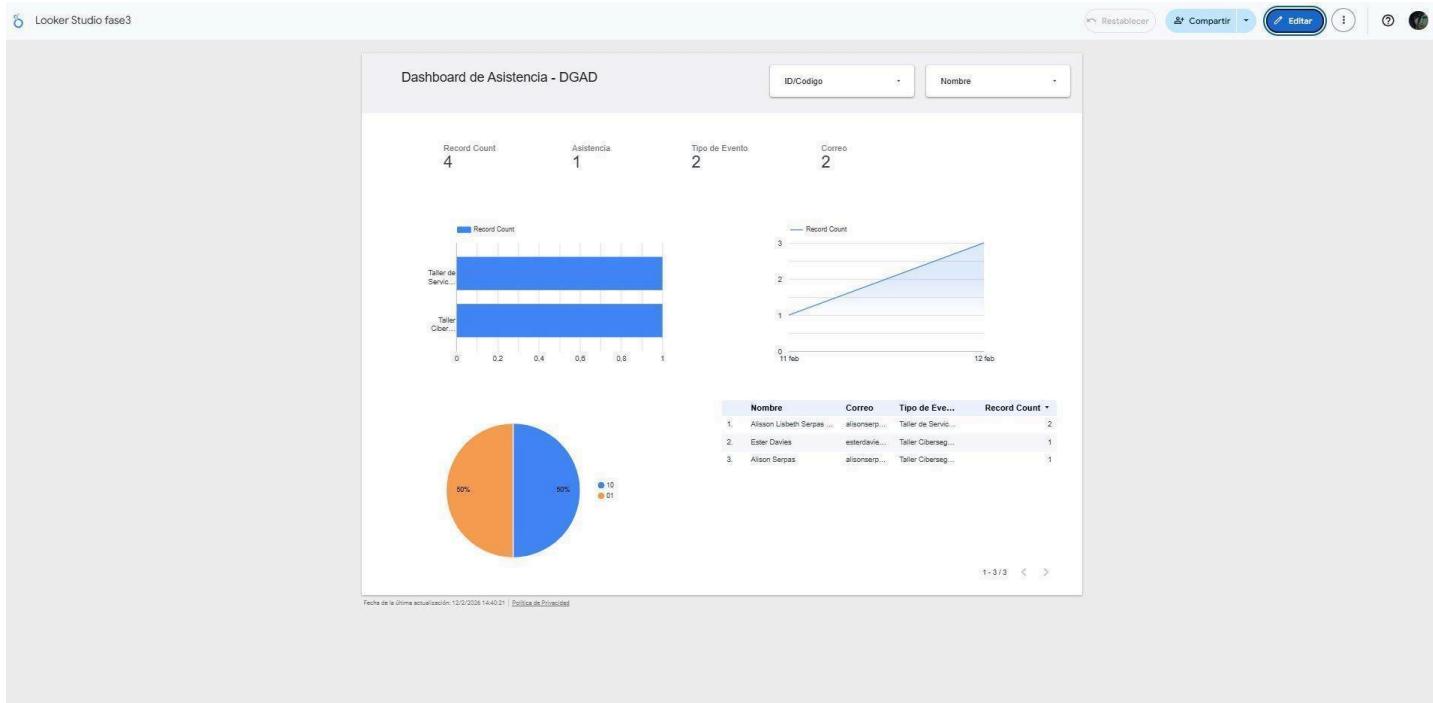
El procesamiento de las respuestas se automatizó completamente. Cada vez que se envía una respuesta, un disparador en Apps Script lee los datos de la última fila ingresada, busca el nombre completo del evento en el catálogo, marca el estado como presente en la columna correspondiente, verifica si el participante es nuevo y lo registra automáticamente si no existe, y envía una confirmación por correo electrónico con diseño profesional. Se implementó además una función de sincronización que mantiene las opciones de eventos del formulario actualizadas con los eventos vigentes del catálogo.



Respecto a la Fase 2, se corrigieron errores técnicos en el procesamiento, particularmente un problema con el puntero de columna que registraba los datos en posiciones incorrectas de la hoja de cálculo. Se añadió la escritura forzada inmediata para garantizar que los datos persistan correctamente, y se implementaron todas las automatizaciones mencionadas que antes no existían.

Integración con Looker Studio

Se integró Looker Studio con el panel de administración mediante un botón de acceso directo que abre el dashboard vinculado al proyecto. Google Sheets actúa como fuente de datos, por lo que los gráficos y métricas de asistencia se actualizan automáticamente cada vez que se registra una nueva entrada en el sistema, sin necesidad de intervención manual.



El dashboard incluye cuatro indicadores principales (total de registros, presentes, eventos únicos y participantes únicos), un gráfico de barras horizontal que muestra la asistencia por evento, un gráfico de serie temporal que visualiza la evolución de registros por fecha, un gráfico circular que presenta la distribución por grupo o sección, y una tabla detallada con el nombre, correo, evento y cantidad de registros de cada participante.

En la Fase 2, Looker Studio existía como herramienta de visualización pero operaba de forma completamente aislada. El profesor debía conocer la URL del dashboard o buscarlo manualmente, lo que generaba una barrera de acceso que en la práctica significaba que muchos organizadores no consultaban los reportes. Al integrar el acceso directamente en el panel de administración, se eliminó esta fricción y se incentivó el uso de datos para la toma de decisiones, cerrando el ciclo completo de información del sistema.

Reestructuración del Backend

El backend en Google Apps Script fue transformado por completo, pasando de ser un script básico de procesamiento de formularios a una aplicación web completa capaz de servir interfaces HTML, gestionar autenticación, procesar datos de múltiples fuentes y ejecutar automatizaciones programadas.

La configuración del sistema se centralizó en un objeto global que contiene el identificador de la hoja de cálculo, los nombres de las cuatro hojas de trabajo (respuestas de formulario, catálogo de eventos, catálogo de participantes y resumen del dashboard), los parámetros de configuración de correo electrónico y la URL de la API de generación de códigos QR. Esta centralización facilita el mantenimiento y permite modificar la configuración del sistema desde un único punto.

El sistema de enrutamiento, ya descrito, permite servir tres vistas HTML desde un solo despliegue. La autenticación se soporta desde el backend con una función que envía códigos OTP de seis dígitos por correo electrónico y otra que valida los correos contra una lista de administradores autorizados. El procesamiento de formularios detecta automáticamente la fuente del registro (Google Form o formulario QR) y aplica la lógica correspondiente en cada caso. La gestión de eventos permite crear nuevos eventos, generar códigos QR con URLs únicas, y almacenar la información resultante en el catálogo.

Se implementaron tres disparadores automáticos: uno que se activa al recibir una respuesta del Google Form y ejecuta todo el procesamiento descrito, otro que genera y envía un reporte semanal todos los lunes a las ocho de la mañana, y un tercero que produce un reporte mensual el primer día de cada mes. Los reportes incluyen métricas como el total de registros, participantes presentes, eventos realizados, participantes únicos y porcentaje de asistencia, y se envían al coordinador con un diseño HTML profesional.

La función de reseteo de datos implementa un protocolo de seguridad riguroso: antes de eliminar cualquier registro, genera un respaldo completo en formato CSV, lo envía como archivo adjunto al correo del coordinador, y solo tras confirmar el envío exitoso procede con la eliminación. El frontend requiere además doble confirmación por parte del usuario antes de iniciar el proceso.

Entre las correcciones técnicas aplicadas respecto a la Fase 2, se destaca la corrección del puntero de columna que escribía los datos en posiciones incorrectas, la implementación de escritura forzada inmediata para garantizar la persistencia de los datos, el manejo robusto de errores con bloques de captura en todas las funciones y envío de alertas al coordinador cuando ocurre una falla, la validación de formato de correo electrónico, la detección de duplicados comparando correo, evento y fecha, y la configuración de validaciones de datos en la hoja para aceptar únicamente estados válidos de asistencia.

AppsScript FASE 3 v1

```

Archivos A-Z + ⌂ 🔍 Ejecutar Depurar doGet | Registro de ejecución | Implementar

Code.gs
1 /**
2 * =====
3 * SISTEMA DE ASISTENCIA DGAD
4 * Cloud Attendance Platform
5 * =====
6 *
7 * Desarrollado por: Jonathan Ernesto Beltran Guerra & Daniel Alberto Perez
8 *
9 * FUNCIONALIDADES:
10 * - Registro automatico de asistencia
11 * - Generacion de codigos QR para eventos
12 * - Envio de confirmaciones por email
13 * - Reportes semanales y mensuales
14 * - Validacion de duplicados
15 * - Panel de administracion web
16 * - Login 2FA
17 */
18
19 // *****
20 // CONFIGURACION GLOBAL
21 // *****
22 const CONFIG = {
23   SHEET_ID: '19Pgn1lHED_BEIDNyxDqVrk-8CDPR2G3rJH-9w3ec',
24   SHEETS: [
25     {RESPOSTAS: 'Respuestas de formulario v3',
26      EVENTOS: 'Catalogo_Eventos',
27      PARTICIPANTES: 'Catalogo_Participantes',
28      DASHBOARD: 'Dashboard_Resumen'},
29   ],
30   EMAIL: {
31     RECIPIENTE: 'DGAD - Sistema de Asistencia',
32     ASUNTO_CONFIRMACION: 'Confirmación de Asistencia - DGAD',
33     ASUNTO_Reporte_SEMANAL: 'Reporte Semanal de Asistencia - DGAD',
34     ASUNTO_Reporte_MENSUAL: 'Reporte Mensual de Asistencia - DGAD'
35   },
36   QR: {
37     API_URL: 'https://api.qrserver.com/v1/create-qr-code/',
38     SIZE: '300x300'
39   }
40 };
41
42 // *****
43 // Login 2FA - ROUTING PRINCIPAL
44 // *****
45
46 /**
47 * Función principal doGet - Maneja todas las rutas
48 * ORDEN DE PRIORIDAD:
49 * 1. QR con parámetro 'evento' > Formulario QR
50 * 2. Parámetro page-admin > Panel Admin (después de login exitoso)
51 * 3. SIN PARÁMETROS o page=login > Login (DEFAULT)
52 */
53 function doGet(e) {
54   Logger.log(`doGet llamado`);
55   Logger.log(`Parámetros: ` + JSON.stringify(e.parameter));
56
57   const parametros = e.parameter || {};
58
59

```

Captura: code.gs

AppsScript FASE 3 v1

```

Archivos A-Z + ⌂ 🔍 Ejecutar Depurar doGet | Registro de ejecución | Implementar

Code.gs
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Login - DGAD Admin</title>
7
8   <style>
9     *
10    margin: 0;
11    padding: 0;
12    box-sizing: border-box;
13  </style>
14
15 body {
16   font-family: 'Sopen UI', Tahoma, Geneva, Verdana, sans-serif;
17   background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
18   min-height: 100vh;
19   display: flex;
20   align-items: center;
21   justify-content: center;
22   padding: 20px;
23 }
24
25 .login-container {
26   background: white;
27   padding: 40px;
28   border-radius: 20px;
29   box-shadow: 0 20px 60px rgba(0,0,0,0.3);
30   min-width: 400px;
31   width: 100%;
32   animation: slideIn 0.5s ease;
33 }
34
35 @keyframes slideIn {
36   from {
37     opacity: 0;
38     transform: translate(-30px);
39   }
40   to {
41     opacity: 1;
42     transform: translate(0);
43   }
44 }
45
46 .logo {
47   text-align: center;
48   margin-bottom: 30px;
49 }
50
51 .logo h1 {
52   color: #667eea;
53   font-size: 32px;
54   margin-bottom: 5px;
55 }
56
57 .logo p {
58   color: #666;
59   font-size: 14px;
60 }


```

Captura: login.html

Apps Script FASE 3 v1

```

Archivos A-Z + ⌂ ⌂ Registro de ejecución | Implementar | ⓘ ⌂

Code.gs
Login.html
PanelAdmin.html
Copia de Code.gs
FormularioQR.html
Copia de PanelAdmin.html
Copia de FormularioQR.html

Bibliotecas +
Servicios + ⌂

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Panel de Administración - DGAD</title>
7
8   <!-- Firebase SDK -->
9   <script src="https://www.gstatic.com/firebasejs/9.22.0.firebaseio-app-compat.js"></script>
10  <script src="https://www.gstatic.com/firebasejs/9.22.0/firebase-auth-compat.js"></script>
11
12
13  <style>
14    *
15      margin: 0;
16      padding: 0;
17      box-sizing: border-box;
18  }
19
20  body {
21      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
22      background: linear-gradient(135deg, #667ecc 0%, #764ba2 100%);
23      min-height: 100vh;
24      padding: 20px;
25  }
26
27  .container {
28      max-width: 1200px;
29      margin: 0 auto;
30  }
31
32  .header {
33      background: white;
34      padding: 30px;
35      border-radius: 10px;
36      box-shadow: 8px 10px 30px rgba(0, 0, 0, 0.2);
37      margin-bottom: 30px;
38      text-align: center;
39  }
40
41  .header h1 {
42      color: #333;
43      font-size: 24px;
44      margin-bottom: 10px;
45  }
46
47  .header p {
48      color: #666;
49      font-size: 16px;
50  }
51
52  .stats-grid {
53      display: grid;
54      grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
55      gap: 20px;
56      margin-bottom: 30px;
57  }
58
59

```

Captura: PanelAdmin.html

Apps Script FASE 3 v1

```

Archivos A-Z + ⌂ ⌂ Registro de ejecución | Implementar | ⓘ ⌂

Code.gs
Login.html
PanelAdmin.html
Copia de Code.gs
FormularioQR.html
Copia de PanelAdmin.html
Copia de FormularioQR.html

Bibliotecas +
Servicios + ⌂

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Registro de Asistencia - DGAD</title>
7
8   <style>
9     *
10      margin: 0;
11      padding: 0;
12      box-sizing: border-box;
13  }
14
15  body {
16      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
17      background: linear-gradient(135deg, #667ecc 0%, #764ba2 100%);
18      min-height: 100vh;
19      display: flex;
20      align-items: center;
21      justify-content: center;
22      padding: 20px;
23  }
24
25  .container {
26      background: white;
27      max-width: 500px;
28      width: 100%;
29      padding: 10px;
30      border-radius: 20px;
31      box-shadow: 8 20px 60px rgba(0, 0, 0, 0.3);
32      animation: slideUp 0.5s ease;
33  }
34
35  @keyframes slideUp {
36    from {
37      opacity: 0;
38      transform: translateY(30px);
39    }
40    to {
41      opacity: 1;
42      transform: translateY(0);
43    }
44  }
45
46  .header {
47      text-align: center;
48      margin-bottom: 30px;
49  }
50
51  .header icon {
52      font-size: 60px;
53      margin-bottom: 15px;
54  }
55
56  .header h1 {
57      color: #333;
58      font-size: 28px;

```

Captura: FormularioQR.html

<https://script.google.com/home>

Comparación Técnica: Antes vs. Despues

La siguiente tabla recoge las diferencias más relevantes entre la implementación de la Fase 2 y las mejoras incorporadas en la Fase 3, organizadas por dimensión técnica. La columna central describe el estado del sistema al cierre de la fase anterior, mientras que la columna derecha presenta el estado actual tras las mejoras implementadas.

Dimensión	Fase 2 (Antes)	Fase 3 (Despues)
Interfaz de usuario	Un solo Google Form como única interfaz para todos los usuarios, sin diferenciación de roles.	Tres vistas HTML diferenciadas: Login.html, PanelAdmin.html y FormularioQR.html, además del Google Form general. Cada vista está optimizada para su tipo de usuario.
Autenticación	Sin autenticación. El acceso a los datos depende de los permisos de compartición de Google Sheets.	Firebase Authentication con verificación en dos pasos: credenciales (correo + contraseña) seguidas de un código OTP de 6 dígitos enviado por correo, con 5 minutos de validez y máximo 3 intentos.
Control de acceso	Cualquier persona con acceso a la hoja de cálculo podía ver y modificar los datos de asistencia.	Login exclusivo para administradores con doble validación: Firebase Auth a nivel de cliente y lista de correos autorizados verificada en el servidor mediante la función verificarAdmin().
Panel de administración	No existía. El organizador consultaba directamente Google Sheets para revisar datos y no disponía de herramientas de gestión.	Panel centralizado con: dashboard de 4 estadísticas en tiempo real, formulario de creación de eventos con generación de QR, sistema de invitaciones multicanal, acciones rápidas (reportes, Looker, reseteo con backup) y gestión de eventos.
Registro de asistencia	Un solo Google Form para todos los participantes, sin diferenciar entre nuevos y registrados.	Sistema dual: Google Form general para participantes nuevos (abierto, sin autenticación) y FormularioQR.html exclusivo para participantes registrados que accedan mediante invitación QR o enlace directo.
QR e invitaciones	No existe ningún mecanismo para generar códigos QR ni para enviar invitaciones a participantes.	El administrador genera QR y URL únicas por evento desde el panel. Opciones de compartir: WhatsApp, Telegram, correo electrónico, copiar URL y descargar QR como imagen.
Detección de duplicados	No existía prevención de registros duplicados. Un participante podrá	Validación a nivel de servidor: antes de registrar, el sistema compara correo + evento + fecha contra los registros

	registrarse múltiples veces al mismo evento.	existentes. Si detecta duplicado, rechaza el registro y muestra un mensaje de alerta.
Verificación de identidad	No se verifica si el participante existía en la base de datos al registrarse.	Formulario valida el correo ingresado contra el Catálogo de Participantes. Si el correo no existe, el sistema rechaza el registro e indica que debe registrarse primero por el formulario general.
Correos automáticos	Correo de confirmación básico en texto plano.	Correos con diseño HTML profesional para: confirmación de asistencia (con datos del evento), código OTP de verificación (con instrucciones de seguridad) y reportes semanales/mensuales con métricas.
Enrutamiento web	Apps Script solo procesaba el Google Form. Sin capacidad de servir páginas HTML distintas.	Sistema de enrutamiento centralizado en doGet() que evalúa parámetros de URL: parámetro evento sirve FormularioQR, parámetro page=admin sirve PanelAdmin, cualquier otro caso sirve Login como vista predeterminada.
Estructura del backend	Un solo archivo Code.gs con función onFormSubmit básica y configuración limitada (3 variables).	Backend reestructurado con: objeto CONFIG centralizado (4 hojas, QR API, email), enrutamiento, OTP, verificación de admin, gestión de eventos, reportes automáticos, backup, validaciones robustas y manejo de errores con alertas.
Archivos del proyecto	1 archivo: Code.gs	7 archivos: Code.gs, Login.html, PanelAdmin.html, FormularioQR.html y copias de respaldo de cada uno.
Disparadores	1 trigger: onFormSubmit (al recibir respuesta del formulario).	3 triggers automáticos: onFormSubmit (al recibir respuesta), generarReporteSemanal (lunes 8:00 AM) y generarReporteMensual (1ro de cada mes 9:00 AM).
Reportes	Solo accesibles vía Looker Studio si el usuario conoce la URL del dashboard.	Reportes semanales y mensuales generados automáticamente con métricas (total registros, presentes, eventos, participantes únicos, asistencia) y enviados por correo al coordinador. Acceso directo a Looker Studio desde el panel.
Gestión de datos	Eliminación manual directa en Google Sheets sin ningún tipo de respaldo previo.	Protocolo de seguridad: generación de backup CSV automático, envío como adjunto al correo del coordinador, doble

		confirmación del usuario en el frontend, y solo entonces se eliminan los datos.
Puntero de columnas	Error documentado: el script escribía en la columna incorrecta (columna 6 para Hora Registro), generando datos en posiciones erróneas.	Corregido: escritura en columna J (índice 10) para Asistencia y columna E (índice 5) para Tipo de Evento. Se agregó SpreadsheetApp.flush() para garantizar persistencia inmediata.
Looker Studio	Dashboard básico con gráficos de asistencia, pero aislado del sistema y sin acceso desde ninguna interfaz.	Dashboard actualizado con 4 KPIs (registros, presentes, eventos únicos, participantes únicos), gráfico de barras por evento, serie temporal por fecha, gráfico circular por grupo y tabla detallada de participantes. Accesible con un clic desde el panel de administración.
Recuperación de contraseña	No aplica porque no existía sistema de autenticación.	Integrado via Firebase sendPasswordResetEmail(), accesible desde el enlace 'Olvidaste tu contraseña?' en la pantalla de login.

Matriz de Pruebas Funcionales

La siguiente matriz documenta las pruebas funcionales ejecutadas para validar el correcto funcionamiento de los componentes implementados en la Fase 3. Cada caso de prueba se identificó con un código único, se definió la entrada, la acción esperada y se registró el resultado observado tras la ejecución real de la prueba. Todos los casos fueron ejecutados manualmente sobre el sistema en producción.

Los casos TC01 a TC1 aprueban las nuevas funcionalidades incorporadas en esta fase 3.

ID	Funcionalidad	Caso de Prueba	Entrada	Acción Esperada	Resultado Observado	Estado (P/F)	Notas / Evidencias
TC01	Login Firebase	Ingresar credenciales válidas	Email y password correctos	Enviar código OTP al correo	Código OTP recibido en correo	P	email con codigo
TC02	Login Firebase	Ingresar credenciales inválidas	Email o password incorrectos	Mostrar error de autenticación	Mensaje "auth/invalid-login-credentials"	P	mensaje de error
TC03	Verificación OTP	Ingresar código OTP correcto	Código de 6 dígitos válido	Redirigir al panel de administración	Acceso al panel exitoso	P	panel cargado
TC04	Verificación OTP	Ingresar código OTP incorrecto	Código erróneo	Mostrar error y descontar intento	Mensaje de error, intentos restantes visibles y cuando fallas 3 veces te regresa al login	P	Probar con 3 intentos
TC05	Verificación OTP	Código OTP expirado	Esperar más de 5 minutos	Rechazar código y solicitar reenvío	Cuando caduca no deja poner el código, opción de reenviar	P	mensaje de error
TC06	Panel Admin	Cargar panel sin autenticación	Acceso directo a URL ?page=admin	Redirigir al login	Redirección al login exitosa	P	Verificar que no se puede acceder sin login

TC07	Generar QR	Crear evento y generar QR	Nombre, ID y fecha del evento	Crear evento en catálogo y mostrar QR	QR generado con URL y opciones de compartir	P	QR generado
TC08	FormularioQR	Registrar participante existente	Correo registrado en catálogo	Registrar asistencia y enviar confirmación	Asistencia registrada, correo enviado	P	confirmación exitosa
TC09	FormularioQR	Registrar participante NO existente	Correo no registrado	Mostrar error indicando que debe registrarse	Mensaje de error mostrado	P	mensaje de error
TC10	FormularioQR	Registro duplicado	Mismo correo + evento + fecha	Detectar duplicado y rechazar	Alerta de duplicado mostrada	P	mensaje de duplicado
TC11	Reporte semanal	Ejecutar reporte semanal	No aplica (trigger automático)	Generar y enviar resumen por correo	Reporte recibido con métricas	P	Captura del correo con reporte
TC12	Reset de datos	Resetear con backup	Click en botón + doble confirmación	Crear CSV, enviarlo y luego borrar datos	Backup recibido por correo, datos eliminados	P	Verificar que llega el CSV
TC13	Logout	Cerrar sesión desde panel	Click en botón Logout	Cerrar sesión Firebase y redirigir a login	Redirección al login exitosa	P	—

MATRIZ DE PRUEBAS

Sistema de Registro y Control de Asistencia en la Nube

1. OBJETIVO

Definir y documentar los casos de prueba funcionales y de experiencia de usuario (UX) del sistema semi-automatizado basado en Google Forms, Google sheets y Looker Studio.

2. ALCANCE

Incluye:

- Validación funcional del flujo completo (Form → Google sheets→ Looker)
- Validación de reglas de negocio
- Validación UX
- Evidencias antes/después
- Documentación de incidencias y correcciones

No incluye:

- Configuración técnica de infraestructura cloud
- Código fuente

3. DESCRIPCIÓN GENERAL DEL SISTEMA

Flujo general:

1. Usuario completa Google Form.
2. Datos enviados a Google sheet.
3. Google sheet consolida registros.
4. Looker Studio general reportes.

4. MATRIZ DE CASOS DE PRUEBA

ID	Descripción	Prioridad
CP-01	Registro con datos válidos	Alta
CP-02	Registro con datos incorrectos	Alta
CP-03	Marcaje múltiples sesiones	Alta
CP-04	Asistencia no registrados	Crítica
CP-05	Reporte asistencia total	Media
CP-06	Reporte asistencia parcial	Media
CP-07	Reporte asistencia nula	Media

CP-01 - Registro con datos válidos

ID	CP-01
Nombre del Caso	Registro con datos válidos
Prioridad	Alta
Objetivo	Validar que el sistema registre correctamente participantes con datos válidos.
Precondiciones	Formulario activo, Firebase disponible, conexión estable.
Pasos	<ol style="list-style-type: none">1. Acceder al formulario.2. Ingresar datos válidos.3. Enviar formulario.
Resultado Esperado	Mensaje de confirmación visible. Registro almacenado en Firebase y visible en Looker.
Resultado Obtenido	Pantalla de Confirmacion de registro de Asistencia a Evento y Correo de Confimación
Estado (Aprobado/Rechazado)	APROBADO
Observaciones	El Formulario exige datos Case Sensitive, al ingresar el correo en MAYUSCULAS, no fue aceptado

Figure 1, CP-01 - Generacion de Evento



Figure 2, CP-01 - Registro de Asistencia utilizando URL y Datos Validos

Crear Evento y Generar QR

✓ Evento Creado y QR Generado

Taller de Electricista 4ta Categoría
ID: Taller-4TA-2026



URL del evento:
https://script.google.com/macros/s/AKfycbwN1H14sk8_UbwJ5pske3cfTp9Lf8TC2Hx1bdwtcwIq-2pbphgIIi6kOTKD6kpCd4/exec?evento=Taller-4TA-2026&fecha=2026-02-14&nombrete=Taller%20de%20Electricista%204ta%20Categoría



Figure 3, CP-01 - Correo confirmación de Registro de Asistencia

CP-02 - Registro con datos incorrectos

ID	CP-02
Nombre del Caso	Registro con datos incorrectos
Prioridad	Alta
Objetivo	Validar que el sistema bloquee registros con datos inválidos.
Precondiciones	Formulario activo.
Pasos	<ol style="list-style-type: none">1. Ingresar email inválido.2. Dejar campo obligatorio vacío.3. Intentar enviar.
Resultado Esperado	El sistema no permite envío y muestra mensajes claros de validación.
Resultado Obtenido	Intento de Registro Fallido
Estado (Aprobado/Rechazado)	APROBADO
Observaciones	El Formulario exige datos Case Sensitive, al ingresar el correo en Mayuscula, no fue aceptado, a pesar que era valido



Registro de Asistencia

Sistema DGAD

Taller de Electricista 4ta Categoría

13 de febrero de 2026



No se pudo registrar

No estás registrado en el sistema. Por favor,
regístrate primero usando el formulario principal.

← Volver a intentar

Figure 4, CP-02 - Registro Fallido

CP-03 - Marcaje de asistencia múltiples sesiones

ID	CP-03
Nombre del Caso	Marcaje de asistencia múltiples sesiones
Prioridad	Alta
Objetivo	Validar que un participante pueda registrar asistencia en diferentes sesiones correctamente.
Precondiciones	Participante previamente registrado.
Pasos	<ol style="list-style-type: none">1. Registrar asistencia sesión 1.2. Registrar asistencia sesión 2.3. Registrar asistencia sesión 3.
Resultado Esperado	Registros independientes por sesión sin duplicidad. Reporte refleja correctamente.
Resultado Obtenido	Pantalla de Confirmacion de Asistencia y Correo de Confirmacion por cada sesion. El archivo Excel, refleja nuevo evento, asistencias y multiples sesiones Reporte existente Looker Studio, refleja el nuevo evento ni los asistentes.
Estado (Aprobado/Rechazado)	APROBADO
Observaciones	Las Pantallas de Registro de Eventos, utilizando QR o URL indican correctamente el evento que uno se registra, pero Los correos de Confirmacion NO REFLEJAN el evento que esta confirmando.

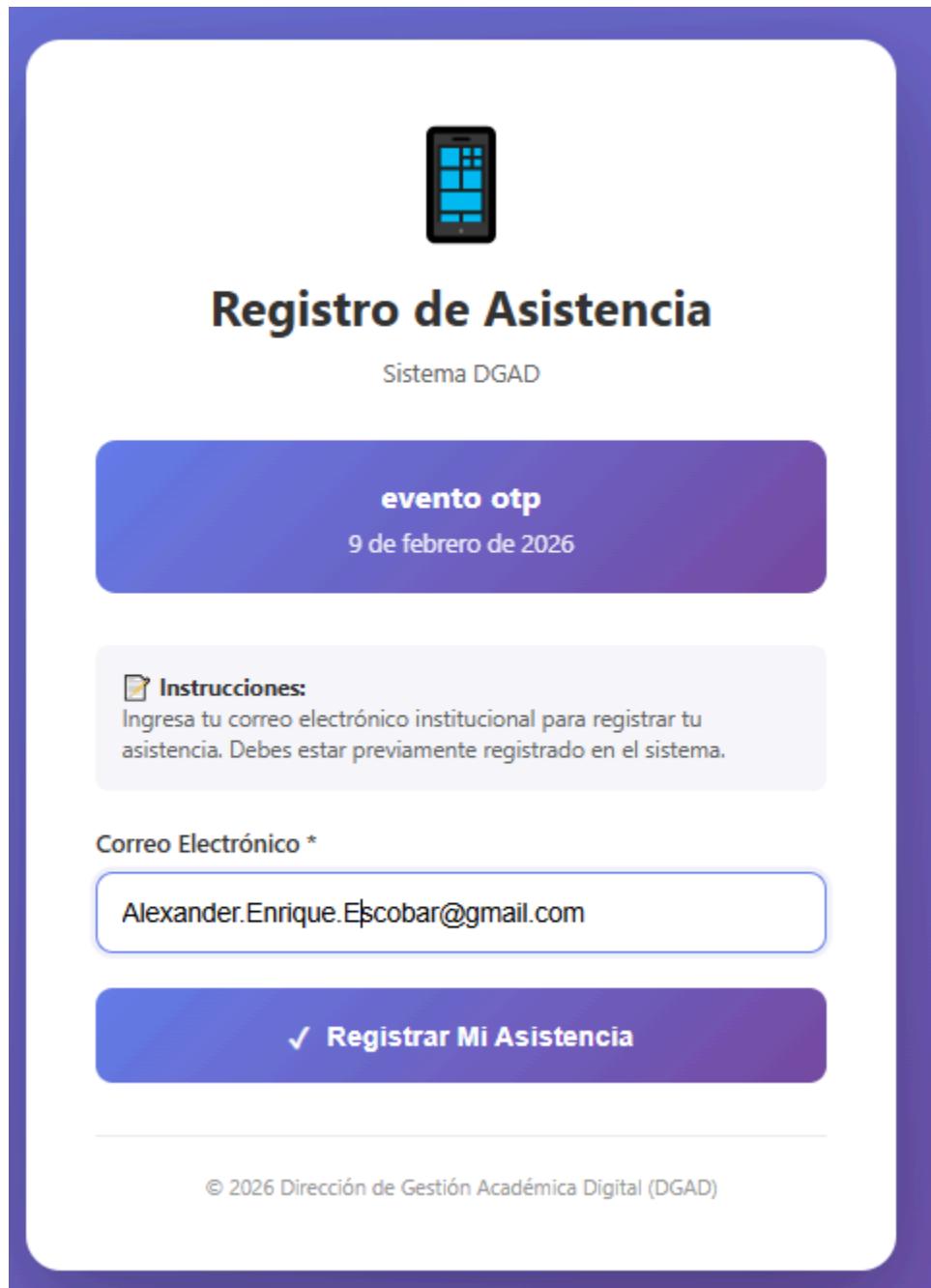


Figure 5, CP-03 - Registro Asistencia Evento #3, Captura de Datos



Registro de Asistencia

Sistema DGAD

Taller de Servicios Cloud

11 de febrero de 2026



¡Asistencia Registrada!

¡Asistencia registrada exitosamente para Alexander
Enrique Escobar Ortíz!

Recibirás una confirmación por correo electrónico.

© 2026 Dirección de Gestión Académica Digital (DGAD)

Figure 6, CP-03 - Registro de Asistencia Evento #2

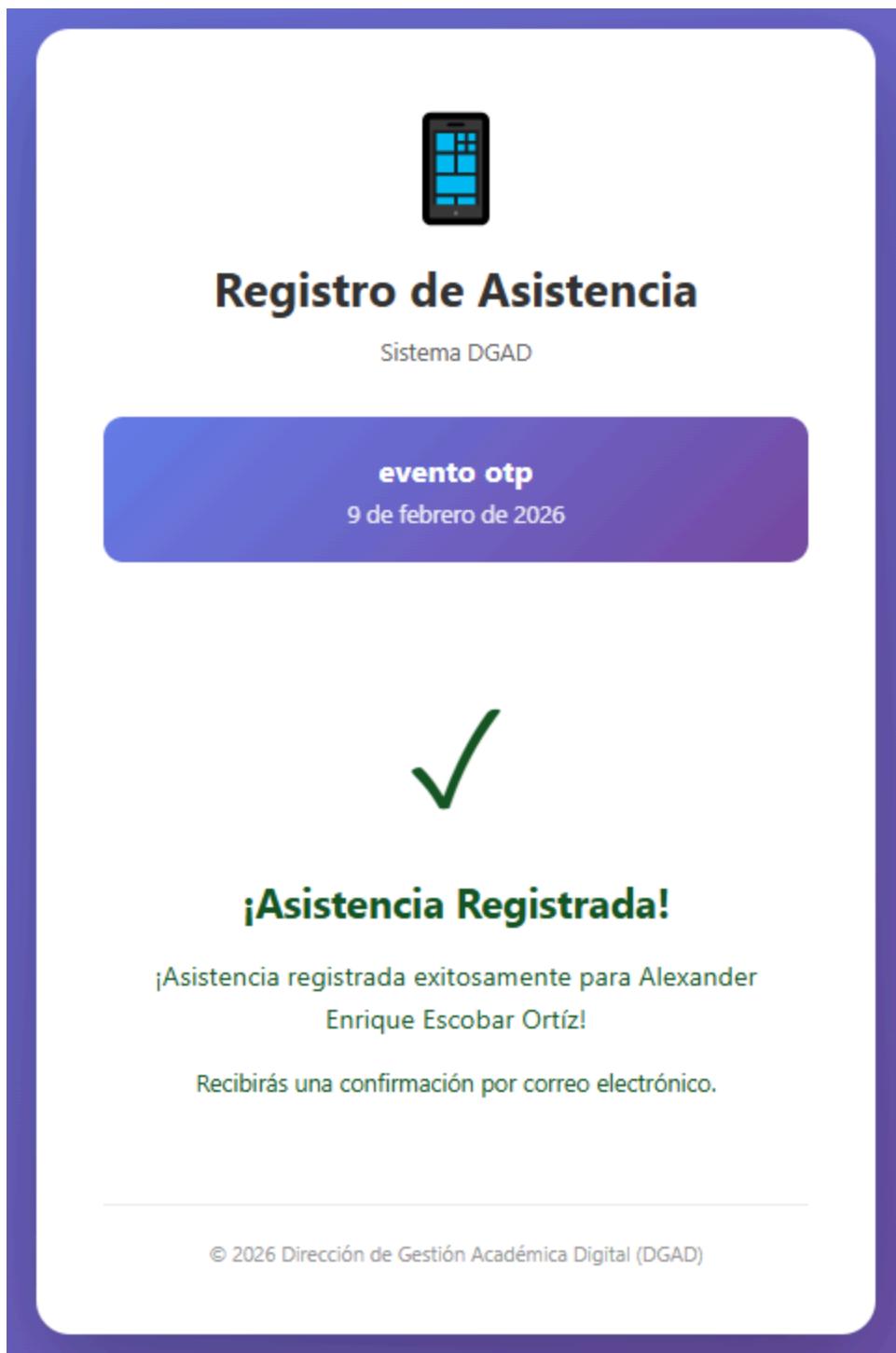


Figure 7, CP-03 - Registro de Asistencia Evento #3

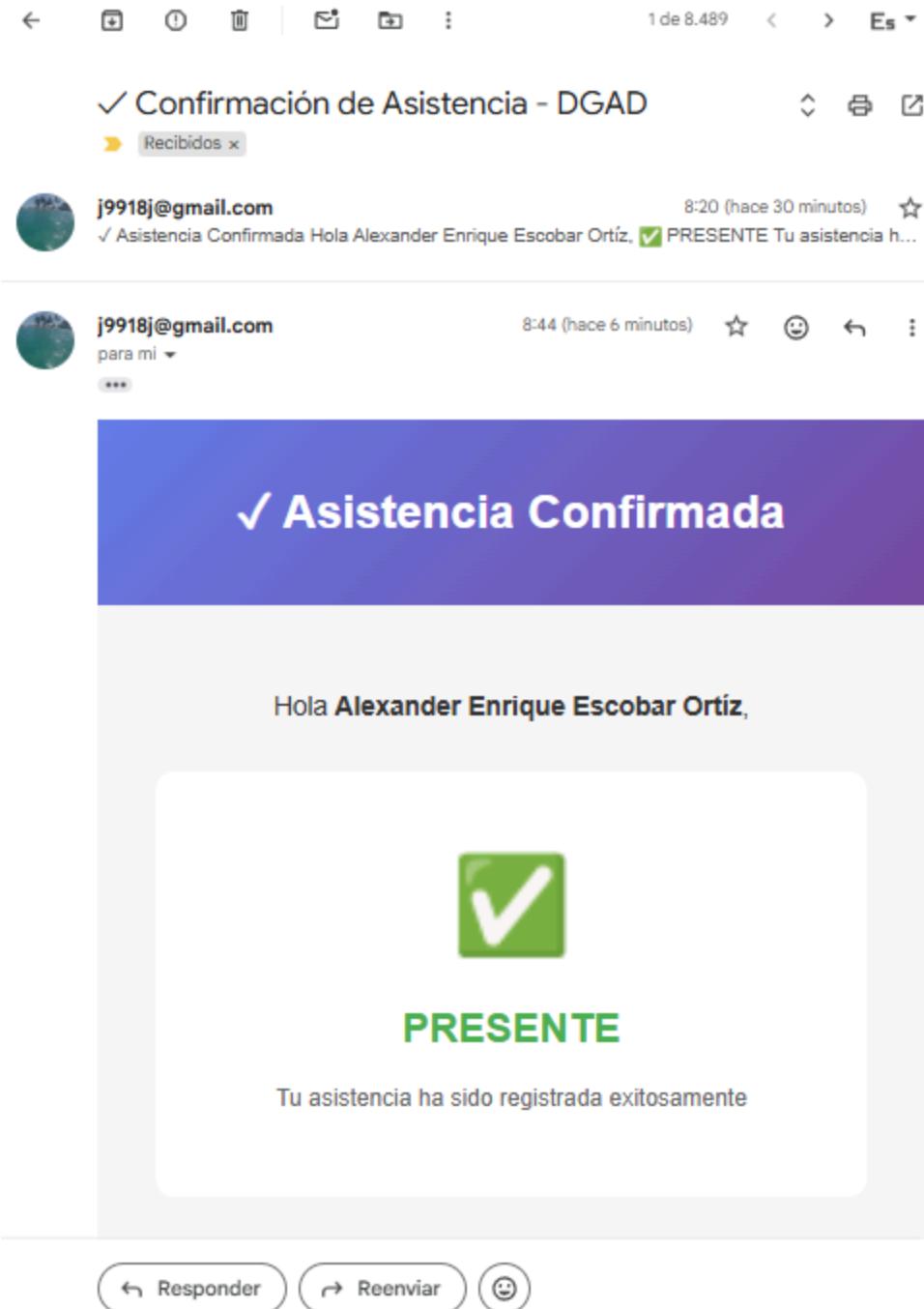


Figure 8, CP-03 - Correo Confirmacion Registro Asistencia Evento #2

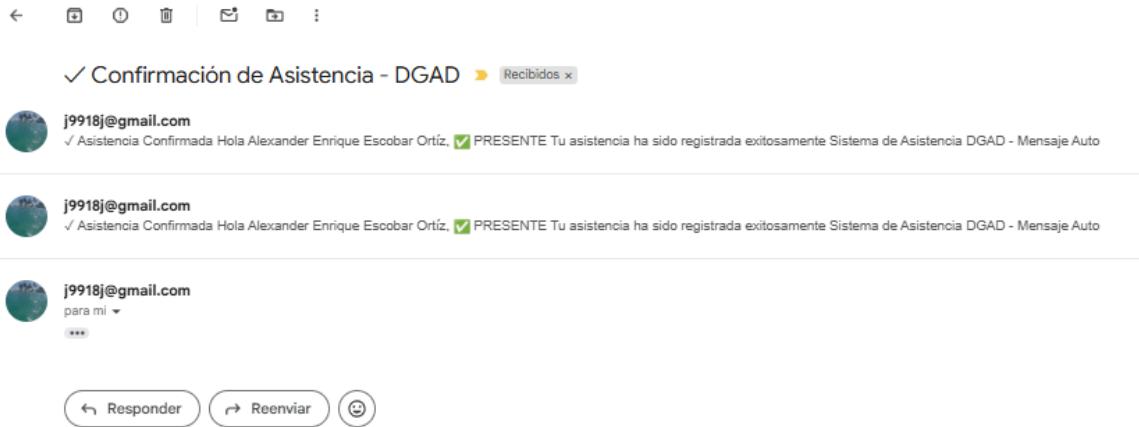


Figure 9, CP-03 - Correo Confirmacion Registro Asistencia Evento #3

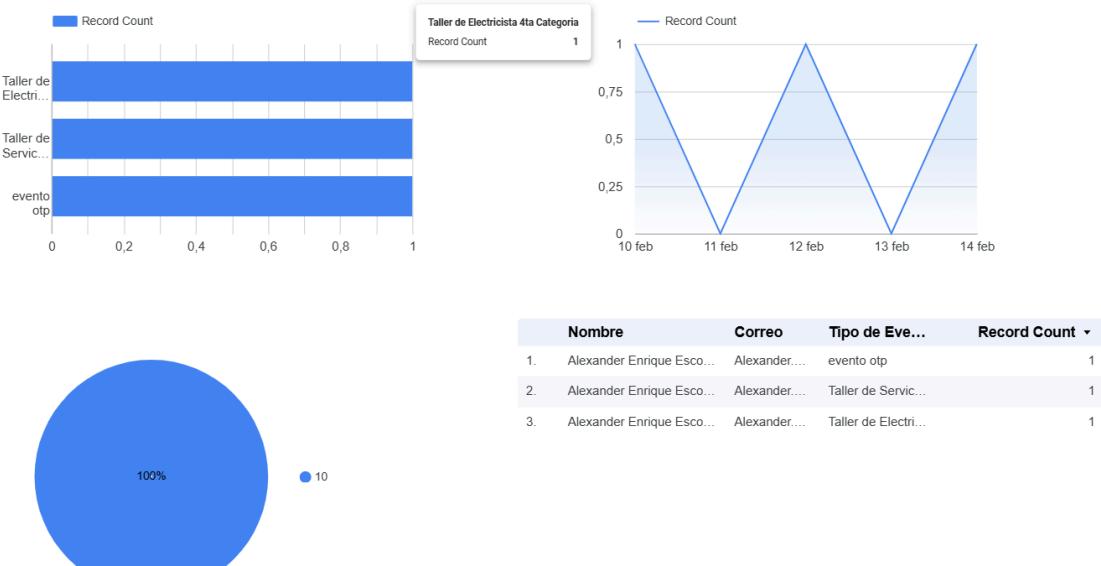


Figure 10, CP-03 - Reporte de Individuo reflejando multi eventos

CP-04 - Asistencia para participantes no registrados

ID	CP-04
Nombre del Caso	Asistencia para participantes no registrados
Prioridad	Crítica
Objetivo	Validar comportamiento cuando un usuario no registrado intenta marcar asistencia.
Precondiciones	Formulario activo.
Pasos	<ol style="list-style-type: none">1. Ingresar datos de usuario no registrado.2. Intentar marcar asistencia.
Resultado Esperado	El sistema muestra advertencia o bloquea registro según regla de negocio definida.
Resultado Obtenido	NO permite registrarse, no permite confirmación
Estado (Aprobado/Rechazado)	APROBADO
Observaciones	_____



Registro de Asistencia

Sistema DGAD

Taller de Electricista 4ta Categoría

13 de febrero de 2026



No se pudo registrar

No estás registrado en el sistema. Por favor,
regístrate primero usando el formulario principal.

← **Volver a intentar**

Figure 11, CP-04 - Intentando registrar asistencia de individuo no Registrado

CP-05 - Reporte asistencia total

ID	CP-05
Nombre del Caso	Reporte asistencia total
Prioridad	Media
Objetivo	Validar que el reporte muestre 100% cuando el participante asistió a todas las sesiones.
Precondiciones	Participante con asistencia completa registrada.
Pasos	<ol style="list-style-type: none">1. Acceder a Looker Studio.2. Filtrar participante.3. Verificar porcentaje.
Resultado Esperado	El sistema muestra 100% de asistencia y métricas correctas.
Resultado Obtenido	El reporte filtra y muestra el 100 % de participación en los Múltiples Eventos
Estado (Aprobado/Rechazado)	APROBADO
Observaciones	Evaluar la posibilidad de mostrar un gráfico por cada evento, para evitar confusiones de visualización

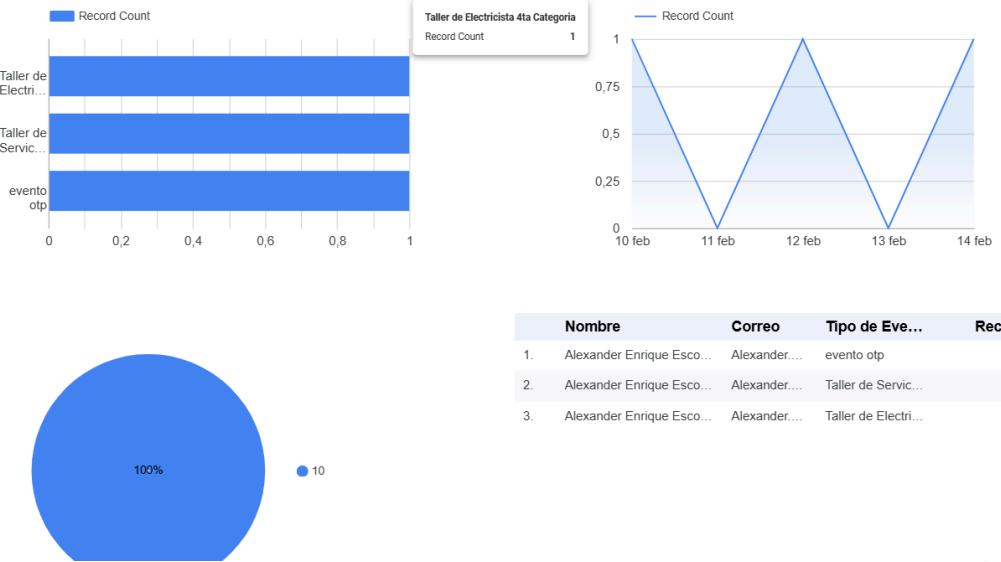


Figure 12, CP-05 – Visualizacion de Asistencia al 100 %

CP-06 - Reporte asistencia parcial

ID	CP-06
Nombre del Caso	Reporte asistencia parcial
Prioridad	Media
Objetivo	Validar cálculo correcto cuando la asistencia es parcial.
Precondiciones	Participante con asistencia incompleta.
Pasos	<ol style="list-style-type: none"> 1. Acceder a Looker Studio. 2. Filtrar participante. 3. Verificar porcentaje parcial.
Resultado Esperado	El porcentaje mostrado corresponde al número real de sesiones asistidas.
Resultado Obtenido	Se visualiza la asistencia, pero no existe múltiples asistencia a un mismo evento
Estado (Aprobado/Rechazado)	RECHAZADO
Observaciones	NO se identificó en forma clara en el reporte, como se reflejan las múltiple asistencias

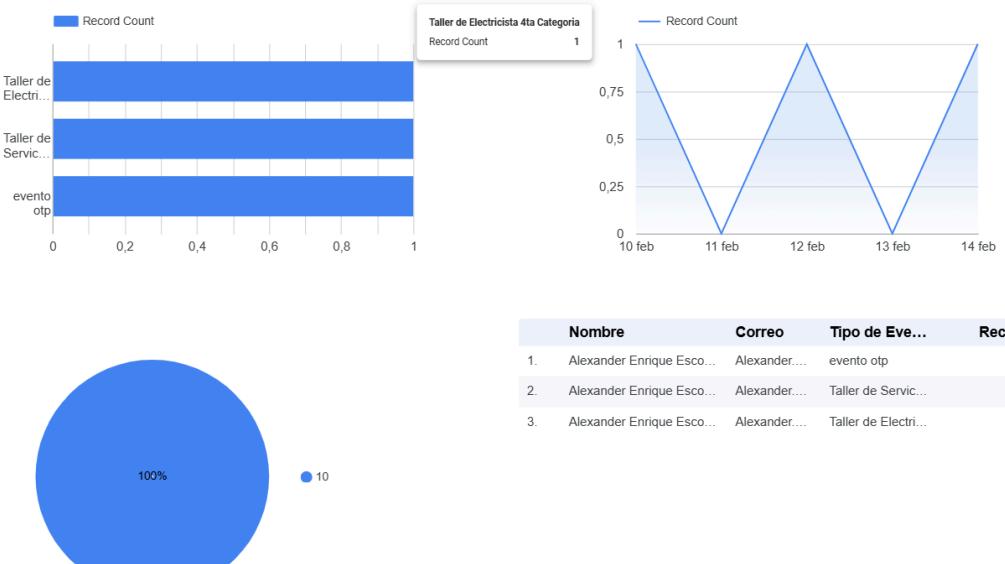


Figure 13, CP-05 – Visualizacion de Asistencia al 100 %

CP-07 - Reporte asistencia nula

ID	CP-07
Nombre del Caso	Reporte asistencia nula
Prioridad	Media
Objetivo	Validar comportamiento cuando el participante no tiene asistencia.
Precondiciones	Participante registrado sin sesiones marcadas.
Pasos	<ol style="list-style-type: none">1. Acceder a Looker Studio.2. Filtrar participante.3. Verificar resultado.
Resultado Esperado	El sistema muestra 0% sin afectar métricas globales.
Resultado Obtenido	El reporte no permite mostrar Asistentes que NO tengan al menos una asistencia
Estado (Aprobado/Rechazado)	RECHAZADO
Observaciones	El reporte debería tener un nivel de filtro, para definir el curso y mostrar a todos los posibles asistentes, incluso si no poseen asistencia.

<input checked="" type="checkbox"/>	Nombre	Record Count
<input type="text"/> Escriba el término de búsqueda		
<input checked="" type="checkbox"/>	Alexander Enrique Escobar	3
<input checked="" type="checkbox"/>	Alisson Lisbeth Serpas	2
<input checked="" type="checkbox"/>	Elmer	1
<input checked="" type="checkbox"/>	Ester Davies	1
<input checked="" type="checkbox"/>	Alison Serpas	1

Figure 14, Listado de Participantes con al Menos una Asistencia a cualquier evento

5. VALIDACIÓN UX

- Mensajes claros y visibles
- Validación de campos obligatorios
- Validación formato email
- Prevención de duplicidad
- Reportes coherentes y filtrables

6. EVIDENCIAS ANTES / DESPUÉS

Caso: Validación Email

Antes: Permitía envío con email inválido.

Después: Bloqueo con mensaje claro y no registro en base de datos.



Figure 15. Validacion de Intento de Ingreso de Correo NO Valido

Caso: Duplicidad de asistencia

Antes: Permitía registros repetidos.

Después: Validación Documento + Sesión.

2/12/2026 12:59:20	Alisson Lisbeth Serpas I	SN-10-05	alisonserpas@gmail.com	Taller de Servicios Cloud	2026-02-12	12:59	10
2/13/2026 8:20:23	Alexander Enrique Escobar	SN-10-03	Alexander.Enrique.Escobar@gmail.com	Taller de Electricista 4ta Categoría	2026-02-14	8:20	10
2/13/2026 8:44:25	Alexander Enrique Escobar	SN-10-03	Alexander.Enrique.Escobar@gmail.com	Taller de Servicios Cloud	2026-02-12	8:44	10
2/13/2026 8:53:51	Alexander Enrique Escobar	SN-10-03	Alexander.Enrique.Escobar@gmail.com	evento otp	2026-02-10	8:53	10
2/13/2026 20:30:52	Alexander Enrique Escobar	SN-10-03	Alexander.Enrique.Escobar@gmail.com	Taller de Electricista 4ta Categoría	2026-02-14	20:30	10
2/13/2026 20:32:43	Alexander Enrique Escobar	SN-10-03	Alexander.Enrique.Escobar@gmail.com	Taller de Electricista 4ta Categoría	2026-02-14	20:32	10

7. INCIDENCIAS Y CORRECCIONES

ID	Problema	Corrección Aplicada	Estado
INC-01	Email inválido permitido	Validación Regex en formulario	Cerrado
INC-02	Duplicidad sesión	Regla validación en Firebase	Cerrado
INC-03	Reporte no mostraba 0%	Ajuste fórmula Looker	Cerrado

8. CRITERIOS DE ACEPTACIÓN

- Todos los casos en estado aprobado.
- No existen incidencias críticas abiertas.
- Reportes coherentes y sin inconsistencias.

9. CONCLUSIÓN

El sistema cumple con los requerimientos funcionales y UX definidos en la matriz de pruebas.

Existen todavía validaciones a incorporar, pero corresponden a la siguiente etapa a validar.

Plan de Recursos Tecnológicos

Cloud Attendance Lite – Fase 3

1. Introducción y Estrategia Tecnológica.

Durante la Fase 2, el proyecto Cloud Attendance Lite fue implementado bajo una arquitectura Serverless Low-Code basada completamente en Google Workspace (Google Forms, Apps Script y Google Sheets). Esta estrategia nos permitió construir un prototipo funcional con persistencia en la nube, con validaciones y con dashboards básicos.

Sin embargo, tras evaluar las limitaciones detectadas en la experiencia de usuario, control de acceso y escalabilidad, en la Fase 3 se realizó una evolución arquitectónica incorporando Google Firebase como plataforma para autenticación, base de datos en tiempo real y app web.

Esta transición permite pasar de un prototipo funcional basado en formularios a un sistema web estructurado con frontend HTML personalizado, control de roles y seguridad mejorada.

Comparación del Sistema: Antes vs Despues

ANTES – Fase 2 (Arquitectura basada en Google Workspace)

Características principales:

- Interfaz basada únicamente en Google Forms.
- Persistencia en Google Sheets.
- Automatización mediante Google Apps Script.
- Reportes en Looker Studio.
- Sin sistema de login.
- Sin control de roles.
- Sin autenticación formal.
- Experiencia visual limitada.
- Limitaciones detectadas:
 - Baja usabilidad.
 - Alto riesgo de errores manuales.
 - Sin control de acceso.
 - Sin trazabilidad de usuarios.
 - Sin experiencia UX personalizada.
 - Dependencia total de Google Forms.

DESPUÉS – Fase 3 (HTML + Firebase + UX Mejorada)

La Fase 3 transforma el sistema en una aplicación web estructurada con separación clara entre usuario y administrador.

Vista del Usuario (Fase 3)

- Acceso mediante código QR.
- Redirección a formularioQR.html.
- Validación automática contra Firestore.
- Mensajes claros:
- Confirmación de asistencia.
- Error por duplicado.
- Validación de datos obligatorios.
- Experiencia rápida, optimizada para móviles.
- UX enfocada en simplicidad.
- Vista Coordinador / Administrador
- Acceso mediante login.html.
- Autenticación con Firebase Authentication.
- Doble factor de autenticación (2FA).
- Control de sesión.
- Panel administrativo (panelAdmin.html).

Funciones disponibles:

- Visualizar asistentes por evento.
- Reportes de asistencia total / parcial / nula.
- Filtrado por sesión.
- Control de registros duplicados.
- Validaciones automáticas.
- Gestión de eventos.

Mejoras Clave Introducidas en Fase 3

- Separación clara de roles (Usuario / Administrador).
- UX moderna y personalizada.
- Seguridad reforzada (Login + 2FA).
- Eliminación de dependencia de Google Forms.
- Flujo profesional similar a sistemas reales.
- Mejor trazabilidad y control de accesos.

Evolución Arquitectónica

Fase 2:

- Google Forms → Apps Script → Google Sheets → Looker Studio
- Arquitectura Low-Code orientada a prototípico.

Fase 3:

- Firebase Authentication
- Autenticación con Firebase

Recurso	Plataforma	Responsable	Tipo / Plan	Uso en el proyecto	Observaciones
Frontend Web	HTML5 + CSS + JS	Desarrollador Fullstack Jr.	Open Source	Interfaces personalizadas (formularioQR.html, login.html, panelAdmin.html)	UI moderna y control total de diseño
Hosting Web	Firebase Hosting	Ingeniero Cloud Jr.	Spark Plan (gratuito)	Publicación segura del sistema web	SSL automático
Autenticación	Firebase Authentication	Ingeniero Cloud Jr.	Gratis (hasta límites del plan)	Login con email + 2FA	Control de sesiones y roles
Base de Datos	Firebase Firestore	Ingeniero Cloud Jr.	Gratis (Spark)	Almacenamiento estructurado de usuarios, eventos y asistencias	Base de datos NoSQL en tiempo real
Backend Serverless	Firebase Functions (opcional)	Desarrollador	Gratis	Validaciones y lógica avanzada	Escalable
Reportes	Panel HTML + consultas Firestore	QA / Documentador Jr.	Integrado	Reportes dinámicos por evento	En tiempo real
Control de Versiones	GitHub	Líder de Proyecto Jr.	Free Tier	Versionamiento del código	Flujo profesional

Recurso	Proveedor / Plataforma	Responsable	Tipo / Plan	Uso previsto en el proyecto	Límites / Observaciones
Frontend / Interfaz	Google Forms + HTML5/CSS/JS	Desarrollador Fullstack Jr.	Gratis (Google Workspace) + Open Source	- Captura de datos mediante formularios web - Desarrollo de interfaces personalizadas (formularioQR.html, login.html, panelAdmin.html) con UX mejorada	Personalización avanzada en HTML. Requiere mantenimiento de código frontend.
Backend / Automatización	Google Apps Script + Firebase Functions (opcional)	Desarrollador Fullstack Jr.	Gratis	- Procesamiento automático de datos desde Forms - Validaciones y control de duplicados - Lógica de negocio integrada con Firebase	Cuotas diarias de Apps Script. Límites del plan gratuito de Firebase.
Base de datos en la nube	Google Sheets + Firebase Firestore	Ingeniero Cloud Jr.	Gratis	- Almacenamiento estructurado de eventos, sesiones y asistencias - Persistencia de usuarios y registros en base de datos en google sheet en tiempo real	Sheets: límite de 5M de celdas. Firestore con límites del plan gratuito.
Autenticación y Seguridad	Firebase Authentication	Ingeniero Cloud Jr.	Gratis (Spark Plan)	- Gestión de usuarios - Login con control de roles - Doble factor de autenticación (2FA)	Límites de autenticaciones mensuales en plan gratuito.

Hosting Web	Firebase Hosting	Ingeniero Cloud Jr.	Gratis (Spark Plan)	<ul style="list-style-type: none"> - Publicación del sistema web bajo HTTPS - Despliegue de frontend HTML personalizado 	Límite de transferencia mensual en plan gratuito.
Reportes y visualización	Looker Studio + Panel Administrativo HTML	QA / Documentador Jr.	Gratis	<ul style="list-style-type: none"> - Dashboards de métricas de asistencia - Validación de reportes dinámicos desde panel administrativo 	Dependencia de datos en Sheets. Reportes HTML dependen de estructura en base de datos.
Repositorio de código	GitHub	Líder de Proyecto Jr.	Free Tier	<ul style="list-style-type: none"> - Versionado de Apps Script - Documentación técnica - Versionado de frontend HTML/Javascript y configuración de Google Firebase 	No almacenar credenciales ni claves privadas.
Herramientas de diseño y pruebas	Google Sheets	QA / Documentador Jr.	Gratis	<ul style="list-style-type: none"> - Diagramas de arquitectura - Modelo de datos - Matriz de pruebas - Evidencias de pruebas de login y roles 	Evidencias almacenadas como capturas de pantalla.
Gestión y comunicación	Jira / WhatsApp	Líder de Proyecto Jr.	Gratis	<ul style="list-style-type: none"> - Asignación de tareas - Seguimiento de avances - Coordinación del equipo 	Acuerdos de respuesta máxima de 24 horas.

TABLA DE ASIGNACIÓN DE TAREAS PARA LA SIGUIENTE FASE 4

Responsable	Componente	Tarea Específica	Entregable
Alison	Informe - Parte 1	Contexto del problema y objetivos del proyecto	Documento Word/PDF (2-3 páginas)
Alison	Informe - Parte 2	Arquitectura de la solución (componentes, tecnologías, diagrama de despliegue)	Documento + Diagramas (3-4 páginas)
Alison	Video Demo - Parte 1	Registro de participantes de ejemplo	Video MAX 1 minuto
Jonathan	Informe - Parte 3	Capturas de pantallas clave y descripción de flujos	Documento con screenshots (3-4 páginas)
Daniel	Informe - Parte 4	Resultados de pruebas y principales correcciones realizadas	Documento técnico (2-3 páginas)
Enrique	Informe - Parte 5	Limitaciones del prototipo y propuestas de mejora futura	Documento estratégico (1-2 páginas)
Jonathan	Video Demo - Parte 4	Consulta de un reporte	Video MAX 2 minutos
Daniel	Video Demo - Parte 2	Creación de un evento y generación QR	Video MAX 2 minutos

Jorge	Video Demo - Parte 3	Marcaje de asistencia a un evento con formulario QR	Video MAX 2 minutos
Jorge	Documentación	Bitácora completa del proyecto	Usar plantilla ESIT
Enrique	Repositorio	Repositorio ordenado y etiquetado	GitHub/Drive organizado y completo

Conclusión

La Fase 3 del proyecto Cloud Attendance Lite representó una transformación integral de la plataforma, que pasó de ser un prototipo básico dependiente de un único Google Form a un sistema web completo con autenticación robusta, panel administrativo centralizado, formularios diferenciados por tipo de usuario y automatizaciones avanzadas de procesamiento y reportería.

La separación entre el flujo administrativo y el flujo de participantes permitió ofrecer una experiencia optimizada para cada rol: los organizadores disponen de herramientas completas de gestión protegidas por doble autenticación, mientras que los participantes acceden al registro de asistencia de forma directa y sin fricciones. El sistema dual de formularios, combinando un Google Form general con un formulario HTML exclusivo accesible mediante invitación, resuelve la necesidad de diferenciar el tratamiento de participantes nuevos y registrados, e introdujo la verificación de identidad y la prevención de duplicados.

El backend fue reestructurado por completo para soportar la nueva arquitectura, incorporando enrutamiento de vistas, procesamiento diferenciado de datos, generación de códigos QR, confirmaciones automáticas por correo, reportes programados, respaldo automático y manejo robusto de errores. Estas mejoras no sólo resolvieron las limitaciones técnicas identificadas en la Fase 2, sino que sentaron las bases para la consolidación final del sistema en la Fase 4.

Para la siguiente fase, los esfuerzos se centrarán en la congelación del código, la elaboración del informe profesional definitivo, la documentación completa de recursos cloud, la preparación de la demostración funcional y la formulación de recomendaciones para mejoras futuras.

