¡Perfecto! Esa es tu clave API de Resend.

¡IMPORTANTE! Esta clave es sensible y nunca debes compartirla directamente en el código fuente ni en conversaciones

públicas. La vamos a guardar de forma segura en Supabase como una "Secret" (secreto de entorno).

Paso 2: Configurar la API Key de Resend como un Secreto en Supabase

Para que tu Edge Function pueda usar la clave API de Resend de forma segura, la configuraremos como una variable de

entorno en Supabase.

1. Ve a tu panel de Supabase.

2. En el menú de la izquierda, busca y haz clic en "Edge Functions" (el icono de una función o lambda).

3. En la sección "Secrets" (Secretos), haz clic en "Add new secret" (Añadir nuevo secreto).

4. En el campo "Name" (Nombre), escribe RESEND\_API\_KEY.

5. En el campo "Value" (Valor), pega tu clave API de Resend: re\_fFhrdkan\_Ah3Xh6VgFjF9Sk3vWAnw33n4.

6. Haz clic en "Add" (Añadir).

Una vez que hayas añadido el secreto, avísame. Luego, te proporcionaré el código para la Edge Function y la sentencia SQL

para el Database Trigger.¡Excelente! Ahora que la clave API de Resend está configurada de forma segura, vamos a

implementar la lógica para el envío de correos.

Este proceso consta de dos partes principales:

1. La Edge Function: Es el código que se ejecutará en los servidores de Supabase para enviar el correo electrónico.

2. El Database Trigger: Es una regla en tu base de datos que le dice a Supabase que ejecute la Edge Function cada vez que se

inserte una nueva solicitud.

---

Paso 1: Código de la Edge Function (send-welcome-email)

Este es el código TypeScript para tu Edge Function.

¡ATENCIÓN! Antes de desplegarla, asegúrate de reemplazar los siguientes valores:

\* "onboarding@tuprestamo.com": DEBES reemplazar esto con una dirección de correo electrónico que hayas verificado en tu

cuenta de Resend.

\* "https://example.com/logo-tu-prestamo.png": Reemplaza esta URL con la URL pública real de tu logo (puedes subirlo a

Supabase Storage o a un CDN).

\* "http://127.0.0.1:5500/oportunidades.html": Reemplaza esta URL con la URL pública real de tu página de oportunidades

(cuando la tengas desplegada).

1 // send-welcome-email/index.ts

2 import { serve } from "https://deno.land/std@0.168.0/http/server.ts";

3 import { Resend } from "https://deno.land/x/resend/mod.ts";

4 import { createClient } from "https://esm.sh/@supabase/supabase-js@2";

5

6 console.log("Hello from Functions!");

7

8 serve(async (req) => {

9 try {

10 const { record } = await req.json();

11 const { id, rol, nombre\_completo, email } = record;

12

13 if (!email) {

14 console.error("Email is missing for record:", record);

15 return new Response(JSON.stringify({ error: "Email is missing" }), {

16 status: 400,

17 headers: { "Content-Type": "application/json" },

18 });

19 }

20

21 const resendApiKey = Deno.env.get("RESEND\_API\_KEY");

22 if (!resendApiKey) {

23 console.error("RESEND\_API\_KEY is not set.");

24 return new Response(JSON.stringify({ error: "RESEND\_API\_KEY not configured" }), {

25 status: 500,

26 headers: { "Content-Type": "application/json" },

27 });

28 }

29

30 const resend = new Resend(resendApiKey);

31

32 let subject = "";

33 let htmlContent = "";

34 let fromEmail = "onboarding@tuprestamo.com"; // <<< REEMPLAZA CON TU EMAIL VERIFICADO EN RESEND

35

36 if (rol === "inversionista") {

37 // Generar código de inversionista único basado en el ID del registro

38 const investorCode = `INV-${id}`;

39

40 // Actualizar el registro en Supabase con el código de inversionista generado

41 const supabaseUrl = Deno.env.get("SUPABASE\_URL");

42 const supabaseAnonKey = Deno.env.get("SUPABASE\_ANON\_KEY");

43 const supabaseClient = createClient(supabaseUrl, supabaseAnonKey);

44

45 const { error: updateError } = await supabaseClient

46 .from("solicitudes")

47 .update({ codigo\_inversionista: investorCode })

48 .eq("id", id);

49

50 if (updateError) {

51 console.error("Error updating investor code:", updateError);

52 // Si hay un error al actualizar, lo logueamos pero continuamos para enviar el email

53 }

54

55 subject = "¡Bienvenido a Tu Préstamo, Inversionista!";

56 htmlContent = `

57 <div style="font-family: Arial, sans-serif; line-height: 1.6; color: #333;">

58 <div style="text-align: center; margin-bottom: 20px;">

59 <img src="https://example.com/logo-tu-prestamo.png" alt="Logo Tu Préstamo" style="max-width: 150px;"

> <!-- <<< REEMPLAZA CON LA URL DE TU LOGO -->

60 </div>

61 <p>Hola ${nombre\_completo || 'Inversionista'},</p>

62 <p>¡Gracias por unirte a la comunidad de Tu Préstamo! Estamos emocionados de tenerte a bordo.</p>

63 <p>Tu código de inversionista único es: <strong>${investorCode}</strong></p>

64 <p>Con él, podrás acceder a tu panel para ver todas las oportunidades de inversión disponibles y

empezar a generar rendimientos.</p>

65 <p style="text-align: center; margin-top: 30px;">

66 <a href="http://127.0.0.1:5500/oportunidades.html" style="display: inline-block; padding: 10px 20px;

background-color: #007bff; color: white; text-decoration: none; border-radius: 5px;">Ver Oportunidades de

Inversión</a> <!-- <<< REEMPLAZA CON LA URL DE TU PÁGINA DE OPORTUNIDADES -->

67 </p>

68 <p>Saludos,</p>

69 <p>El equipo de Tu Préstamo</p>

70 </div>

71 `;

72 } else if (rol === "prestatario") {

73 subject = "Hemos recibido tu solicitud en Tu Préstamo";

74 htmlContent = `

75 <div style="font-family: Arial, sans-serif; line-height: 1.6; color: #333;">

76 <div style="text-align: center; margin-bottom: 20px;">

77 <img src="https://example.com/logo-tu-prestamo.png" alt="Logo Tu Préstamo" style="max-width: 150px;"

> <!-- <<< REEMPLAZA CON LA URL DE TU LOGO -->

78 </div>

79 <p>Hola ${nombre\_completo || 'Prestatario'},</p>

80 <p>Hemos recibido tu solicitud de refinanciamiento en Tu Préstamo correctamente.</p>

81 <p>Nuestro equipo de análisis de crédito la está evaluando y nos pondremos en contacto contigo a la

brevedad para informarte sobre los siguientes pasos.</p>

82 <p>¡Estás un paso más cerca de mejorar tu salud financiera!</p>

83 <p>Saludos,</p>

84 <p>El equipo de Tu Préstamo</p>

85 </div>

86 `;

87 } else {

88 console.warn("Unknown role for email:", rol);

89 return new Response(JSON.stringify({ message: "Unknown role, no email sent" }), {

90 status: 200,

91 headers: { "Content-Type": "application/json" },

92 });

93 }

94

95 const { data, error } = await resend.emails.send({

96 from: fromEmail,

97 to: email,

98 subject: subject,

99 html: htmlContent,

100 });

101

102 if (error) {

103 console.error("Error sending email:", error);

104 return new Response(JSON.stringify({ error: error.message }), {

105 status: 500,

106 headers: { "Content-Type": "application/json" },

107 });

108 }

109

110 console.log("Email sent successfully:", data);

111 return new Response(JSON.stringify({ message: "Email sent successfully" }), {

112 status: 200,

113 headers: { "Content-Type": "application/json" },

114 });

115 } catch (error) {

116 console.error("Function error:", error);

117 return new Response(JSON.stringify({ error: error.message }), {

118 status: 500,

119 headers: { "Content-Type": "application/json" },

120 });

121 }

122 });

---

Paso 2: Desplegar la Edge Function en Supabase

1. Ve a tu panel de Supabase.

2. Navega a "Edge Functions" (el icono de una función o lambda en el menú lateral).

3. Haz clic en "New Function".

4. En el campo "Name", escribe send-welcome-email.

5. Puedes dejar el campo "Path" como /send-welcome-email o simplemente /.

6. Haz clic en "Create Function".

7. Una vez creada, se abrirá un editor. Reemplaza todo el código predeterminado con el código TypeScript que te proporcioné

arriba (asegurándote de haber hecho los reemplazos de las URLs y el email from).

8. Haz clic en "Deploy" para desplegar la función.

---

Paso 3: Crear el Database Trigger en Supabase

Este trigger llamará a tu Edge Function cada vez que se inserte una nueva fila en la tabla solicitudes.

¡ATENCIÓN! Antes de ejecutarlo, DEBES reemplazar `YOUR\_SUPABASE\_PROJECT\_REF` con la referencia real de tu proyecto

Supabase.

\* Puedes encontrar tu Project Reference en tu panel de Supabase, yendo a Project Settings -> API -> Project URL. Es la

parte de la URL antes de .supabase.co (ej: abcdefg12345).

Además, asegúrate de que la extensión `http` esté habilitada en tu base de datos. Si no lo está:

\* Ve a Database -> Extensions en Supabase.

\* Busca http y actívala.

Ahora, copia y ejecuta el siguiente código SQL en el SQL Editor de tu panel de Supabase:

1 CREATE OR REPLACE FUNCTION public.handle\_new\_solicitud()

2 RETURNS TRIGGER AS $$

3 DECLARE

4 function\_url TEXT;

5 function\_name TEXT := 'send-welcome-email'; -- Nombre de tu Edge Function

6 BEGIN

7 -- Construye la URL para la Edge Function

8 function\_url := 'https://YOUR\_SUPABASE\_PROJECT\_REF.supabase.co/functions/v1/' || function\_name; -- <<<

REEMPLAZA 'YOUR\_SUPABASE\_PROJECT\_REF'

9

10 -- Llama a la Edge Function

11 PERFORM http\_post(

12 function\_url,

13 json\_build\_object('record', NEW)::text,

14 'application/json',

15 ARRAY[

16 http\_header('Authorization', 'Bearer ' || current\_setting('request.jwt.arr', true))

17 ]

18 );

19

20 RETURN NEW;

21 END;

22 $$ LANGUAGE plpgsql SECURITY DEFINER;

23

24 -- Crea el trigger

25 CREATE TRIGGER on\_new\_solicitud

26 AFTER INSERT ON public.solicitudes

27 FOR EACH ROW EXECUTE FUNCTION public.handle\_new\_solicitud();

---

Este es un proceso con varios pasos. Tómate tu tiempo. Una vez que hayas completado todos estos pasos (desplegar la

función, reemplazar las URLs/email, y ejecutar el trigger con tu Project Ref), avísame para que podamos probarlo.