

```

Log.info(StringUtil.creaString("Iniciase a comprobacion do usuario de Google co token: ", idTokenString));

HttpTransport httpTransport = GoogleNetHttpTransport.newTrustedTransport();
JsonFactory jsonFactory = JacksonFactory.getDefaultInstance();

GoogleIdTokenVerifier verifier = new GoogleIdTokenVerifier.Builder(httpTransport, jsonFactory)
    //Id do cliente da aplicacion para acceder a Google
    .setAudience(Collections.singletonList("937334461735-5ld5hochfmdj5hour3tdh0cbigtobtjo.apps.googleusercontent.com"))
    .build();

GoogleIdToken idToken = verifier.verify(idTokenString);
if (idToken != null) {
    Payload payload = idToken.getPayload();
    boolean emailVerified = Boolean.valueOf(payload.getEmailVerified());
    if (emailVerified) {
        String userId = payload.getSubject();
        Log.info(StringUtil.creaString("User ID: ", userId));

        // Get profile information from payload
        String email = payload.getEmail();

        //Recupera a conta de usuario a traves dun email. Se non existe a conta, creaa
        Usuario usuario = this.museoManager.getUsuario(email);
        resposta.setIdUsuario(usuario.getIdUsuario());
        resposta.setContaUsuario(email);

        List<UsuarioEdificio> listaUsuarioEdificio = this.museoManager.getListUsuarioEdificioPorIdUsuario(usuario.getIdUsuario());
        for (UsuarioEdificio ue : listaUsuarioEdificio) {
            if (ue.getAdministrador()) {
                resposta.getListIdEdificioAdministrador().add(ue.getIdEdificio());
            }
        }
    }
    else {
        Log.info("O email non foi verificado");
    }
}
else {
    Log.error("Token invalido");
}

```