



Especificación de requerimientos de software

*Elaborado por el equipo de desarrollo HelpBits
para la edición 7 de Programathon*

Octubre del 2020

Versión 1.0

Tabla de contenidos

Tabla de contenidos	2
Introducción	3
Descripción	3
Perspectiva del producto	3
Características principales	3
Modelo de datos	4
Tipos de usuario	8
Tabla de roles	8
Tarjetas de rol	8
Funciones de la aplicación	9
Descripción de los módulos	9
Historias de usuario	9
Módulo de acceso	9
Módulo de gestión de puntos	13
Módulo de gestión de retos	18
Módulo de acceso externo	20
Anexos	22
Mockups	22

Introducción

Descripción

TuristeApp permite, a través de una aplicación móvil, crear diferentes tours a nivel local en los que las personas que practican turismo pueden consultar, visitar, calificar, y recomendar negocios y zonas turísticas, de una forma sencilla.

Esta es nuestra solución a la problemática económica que se ha generado en el sector turismo producto de la pandemia por COVID-19. Específicamente atacamos la poca disponibilidad de recursos que tienen algunos emprendimientos de este sector, poniendo a su disposición una plataforma gratuita para que accedan a tecnología de vanguardia y ganen visibilidad.

Perspectiva del producto

Características principales

Dentro de las características de nuestra aplicación destacan las siguientes:

1. Crear perfiles de usuario y conexión a redes sociales.
2. Consultar puntos turísticos en un mapa interactivo.
3. Agregar puntos turísticos en un mapa.
4. Compartir información sobre los puntos turísticos visitados.
5. Crear y participar de retos a partir de distintos puntos turísticos.

Todo lo anterior promoviendo la creación de contenido multimedia, específicamente el envío de fotos, de modo que la comunidad que utiliza la aplicación pueda calificar y considerar la experiencia de los demás usuarios.

Se espera que de manera libre se promueva la visita a zonas populares, pero también a los rincones más desconocidos y bellos que conforman el conjunto de atracciones turísticas con las que cuenta nuestro país, y se incentive al viajero nacional a visitarlos.

Modelo de datos

Se utilizará una base de datos NoSQL orientada a documentos. El siguiente diseño muestra la colecciones de documentos que se van a tener con sus distintas propiedades. A continuación se muestra el significado de cada color utilizado en las colecciones.

Las colecciones inician con la sigla MF(**M**odel **F**irebase)

Leyenda
Colección en Firebase (El usuario puede leer y escribir)
Modelo de Datos (no es collection, es un tipo de datos)
Colección en Firebase datos default (El usuario puede leer)
Propiedad generada por sistema
Es copia del objeto (No es referencia)

MFConfiguration: Colección que solo contendrá un único documento con las configuración por defecto. Permitiendo cambiar el estilo de la aplicación de una forma rápida.

MFConfiguration		
Name	Type	Description
id	String	
primaryColor	String	
secondaryColor	String	
mapDefaultZoom	Number	

MFSysytem: Colección que solo contendrá un documento las versión de la aplicación. Y campos a futuro de configuraciones del sistema.

MFSysytem		
Name	Type	Description
id	String	
version	String	

MFUser: Colección que almacena los documentos con la información de cada usuario. Se crea un documento por usuario. Este usuario tiene los ids de los retos visitados y completados, lo que permite filtrar visualizar de forma rápida los retos.

MFUser		
Name	Type	Description
id	String	
mail	String	

username	Number	
password	String	encrypted
birthdate	Date	
mailValidated	Bool	
role	MFRole	
visitedChallengePointIds	[String]	
completedChallengePointIds	[String]	

MFRole: Colección de documentos con los distintos roles que puede tener un usuario. En este caso a futuro se puede manejar los permisos y autorizaciones dinámicas.

MFRole		
Name	Type	Description
id	String	
name	String	
level	Number	lowest is highest
logEditorId	String	

MFChallenge: Colección de documentos que almacena la información relacionada con un reto.

MFChallenge		
Name	Type	Description
id	String	
points	[MFChallengePoint]	
name	String	
description	String	
popularity	Number	
photo	MFPhoto	
labels	[MFLabel]	
logEditorId	String	
creationDate	Date	

MFPhoto: Esto se maneja como tipo de datos, y no cómo una colección. Tiene la información asociada con una imagen. Almacena dos campos importantes el **storageURL** que almacena el link al almacenamiento de la imagen en firebase. Y **url** tiene la dirección para acceder directamente a la imagen.

MFPhoto		
Name	Type	Description

id	String	
name	String	
url	String	Link directo
storageURL	String	Link de storage
altText	String	
logEditorId	String	

MFChallengePoint: Colección de documentos que almacena la información de un punto de un reto. En esta colección se tiene la información de los retos que tienen este punto. Además se almacenan distintos post que realizan los usuarios.

MFChallengePoint		
Name	Type	Description
id	String	
latitude	String	
longitude	String	
name	String	
description	String	
popularity	Number	
photo	MFPhoto	
labels	[MFLabel]	
posts	[MFPost]	
checkins	[MFCheckin]	
logEditorId	String	
creationDate	Date	
includedInChallenges	[String]	

MFChallengePointMarker: Colección de documentos con una versión liviana de un punto de reto. Para mejorar el performance a la hora de mostrar los puntos en el mapa.

MFChallengePointMarker		
Name	Type	Description
id	String	
latitude	String	
longitude	String	
name	String	
description	String	

popularity	Number
challengePointId	String

MFLabel: Colección de documentos con información de labels para asociar los puntos de los mapas. Inicialmente serán datos por default, a futuro los podrían crear cada usuario.

MFLabel		
Name	Type	Description
id	String	
name	String	

MFCheckin: No es una colección de documentos, sino es un tipo de datos para manejar cada check-in en un punto de cada reto.

MFCheckin		
Name	Type	Description
id	String	
userId	String	
date	Date	

MFPost: Colección de documentos con los posts de cada usuario sobre un punto de un reto.

MFPost		
Name	Type	Description
id	String	
likedByUserIds	[String]	
review	String	reseña
photo	MFPhoto	
logEditorId	String	
creationDate	Date	

Tipos de usuario

Tabla de roles

Rol	¿Quién?
Admin user	La ONG o patrocinador final? required
Free user	Turistas locales que no están interesados en pagar por usar la App required
Premium user	Turistas locales que requieren funcionalidades extra y pagan por ellas nice to have
Gold Spot user	Los dueños de negocios turísticos que están interesados en pagar por datos nice to have

Tarjetas de rol

Rol: Admin user **(required)**

Descripción: Superusuario de la aplicación. Puede crear y borrar contenido del app, eliminar usuarios, ver registro de fallos y actividad, entre otras labores de mantenimiento.

Rol: Free user **(required)**

Descripción: Usuario mayoritario y común en la aplicación. Realiza viajes en Costa Rica de manera ocasional (1 o menos veces al mes), y utiliza la aplicación como fuente de información para decidir el destino turístico que visitará.

Rol: Premium user **nice to have**

Descripción: Usuario que realiza viajes en Costa Rica de manera frecuente (2 o más veces al mes), y utiliza la aplicación para planificar sus viajes y llevar registro de estos. Está muy interesado en compartir sus viajes a través de las redes sociales.

Rol: Gold Spot user **nice to have**

Descripción: Es el dueño de uno o más negocios turísticos. Sumamente interesado en poder ver datos que se generan en el App relativos a su mercado, zona geográfica, y negocio

particular (cuántas visitas, edad promedio de visitantes, comentarios generados, entre otras).
--

Funciones de la aplicación

Descripción de los módulos

- 1XX es la codificación para las historias del módulo de **acceso**: login y sign up.
- 2XX es la codificación para las historias del módulo de **gestión de puntos**: puntos en un mapa, renderizado, y funciones relacionadas.
- 3XX es la codificación para las historias del módulo de **gestión de retos**: agrupaciones de puntos dentro del mapa.
- 4XX es la codificación para las historias del módulo de **acceso externo**: compartir datos de la aplicación en otras.

donde XX es un consecutivo.

Historias de usuario

Para realizar las historias de usuario nos hemos basado en la consigna KISS, para lo cual usamos la plantilla que se muestra a continuación.

Card XXX	Como: <user role>	Quiero: <acción>
	Tal que: <descripción>	
Estimación: <1-5> 1 = Menor complejidad 5 = Mayor complejidad	Comentarios: <comentarios del equipo sobre esta carta>	
	Criterios de aceptación	

Módulo de acceso

Card 100	Como: free user	Quiero: registrarme en la aplicación (required)
	Tal que: pueda utilizar sus servicios ahora y en el futuro.	

3	<p>Comentarios: se requiere mínimo tener el sign up propio. <i>Nice to have</i> tener acceso con otras redes sociales (Google, Microsoft o Facebook, esto podría no incluirse en este sprint).</p> <p>Validaciones de datos: Todos los campos son requeridos Campo email con formato de email Contraseñas coinciden Fecha correcta</p> <p>Escenario: Dado un usuario que no se ha registrado en la aplicación, cuando inicia la aplicación debe mostrar una pantalla con dos opciones, crear cuenta o loguearse. Selecciona crear cuenta.</p> <p>Usuario ingresa datos en formatos correctos: El usuario digita datos correctos, se habilita el botón de crear usuario, el usuario presiona el botón. La aplicación deberá mostrar un mensaje que muestra que ha creado la cuenta con éxito.</p> <p>Usuario ingresa datos incorrectos: Se debe mostrar mensaje en cada campo con el error y deshabilitar el botón de crear usuario.</p> <p>Usuario ingresa correo existente: El usuario digita datos correctos pero un correo existente, la aplicación debe mostrar un mensaje de error con el mensaje "Correo ya se encuentra registrado."</p> <p>Escenario nice to have: Dado un usuario que no se ha registrado en la aplicación, cuando inicia la aplicación debe mostrar una pantalla con dos opciones, crear cuenta o loguearse. Selecciona loguearse con google.</p> <p>Usuario ingresa correo gmail con credenciales correctas: Redirige a pantalla con login de google al digitar datos correctamente el usuario se loguea en el sistema. Y se muestra pantalla para agregar fecha y nombre de usuario.</p> <p>Usuario ingresa correo gmail con credenciales incorrectas: Redirige a pantalla con login de google al digitar datos incorrecto el usuario no se debe loguear en el sistema.</p>
---	---

Tareas
<ol style="list-style-type: none"> 1. Crear componente de registro 2. Crear validaciones de datos y mensajes de error 3. Integrar con BE

Card 101	Como: free user	Quiero: confirmar mi correo electrónico nice to have
	Tal que: pueda validar que mi cuenta existe y es la correcta, en el momento que me registre por primera vez en la aplicación.	
3	Comentarios: si un usuario se registra usando un correo propio, entonces debe validarse para confirmar la cuenta. La idea es evitar el registro de varias cuentas con el mismo correo, o usando cuentas fantasma.	

Tareas
<ol style="list-style-type: none"> 1. Configurar servidor de correos (SendGrid probablemente) 2. Envío de correo una vez que se registra el usuario en el sistema 3. Validación de enlace enviado al correo 4. Validación de usuario registrado al momento del login

Card 102	Como: free user	Quiero: iniciar sesión en la aplicación (required)
	Tal que: pueda utilizar sus servicios ahora.	
4	<p>Comentarios: iniciar sesión usando las credenciales creadas. Si estas son incorrectas no se puede usar el app.</p> <p>Validaciones de datos: Todos los campos son requeridos Campo email con formato de email</p> <p>Escenario: Dado un usuario que se ha registrado en la aplicación, cuando inicia la aplicación debe mostrar una pantalla con dos opciones, crear cuenta o loguearse. Selecciona crear cuenta.</p> <p>Usuario ingresa correo y contraseña correctos: El usuario digita datos correctos, se habilita el botón de iniciar sesión, y es presionado por el usuario. La aplicación deberá redirigirlo a la pantalla principal de la aplicación.</p> <p>Usuario ingresa correo y contraseña incorrectos: El usuario digita datos incorrectos, se habilita el botón de iniciar sesión, y es presionado por el usuario. La aplicación deberá mostrar un mensaje de error con el mensaje "Contraseña o correo incorrectos".</p>	

	<p>Escenario: Dado un usuario que ha iniciado sesión anteriormente.</p> <p>Usuario ingresa a aplicación: Los datos deben estar almacenados y redirigir a la pantalla principal sin tener que iniciar sesión de nuevo.</p> <p>Escenario nice to have: Dado un usuario que se ha registrado en la aplicación mediante el inicio de sesión con Google, cuando inicia la aplicación debe mostrar una pantalla con dos opciones, crear cuenta o loguearse. Selecciona loguearse con google.</p> <p>Usuario ingresa correo gmail con credenciales correctas: Redirige a la pantalla principal.</p> <p>Usuario ingresa correo gmail con credenciales incorrectas: Muestra la pantalla de inicio de sesión.</p>
--	--

Tareas
<ol style="list-style-type: none"> 1. Crear screen de login 2. Validación de input 3. API call que valida credenciales 4. Guardar token en el local storage 5. Configuración de manejo del token de sesión

Card 103	Como: free user	Quiero: recuperar mi contraseña nice to have
	Tal que: acceda al app en caso de que olvide, o la hayan robado.	
3	Comentarios: Probablemente no se implemente para esta primera iteración.	
	<p>Validaciones de datos: El correo tiene formato de correo</p> <p>Escenario: Usuario olvidó contraseña, presiona botón de olvidé contraseña</p> <p>Usuario presiona el botón de "Olvidé contraseña": Se muestra pantalla con campo para digitar correo, cuando este tiene el formato correcto se habilita el botón para recuperar contraseña.</p> <p>Usuario presiona el botón de "Recuperar contraseña": Se muestra un mensaje informativo con el texto "Se ha enviado un mensaje de recuperación a su correo.". Este mensaje siempre se muestra en caso de existir o no el correo.</p>	

Tareas
<ol style="list-style-type: none"> 1. Screen donde se solicita el correo 2. Validar el correo en un api cal 3. Enviar un correo con el enlace de restablecimiento de contraseña 4. Validar enlace para restablecer contraseña 5. Modificar el usuario mediante un llamado al API

Módulo de gestión de puntos

Card 200	Como: free user	Quiero: ver un resumen de puntos turísticos creados en un mapa de Costa Rica (required)
	Tal que: pueda visitar el o los puntos de mi interés, accediendo a la información más básica del punto turístico elegido.	
5	Comentarios: el mapa debe ser interactivo. Podemos traernos todos los puntos al mapa (coordenadas solamente), y destacar un top de puntos por ubicación del user. Por defecto mostrar un filtro con los puntos más cercanos según la ubicación actual del usuario. Pantalla principal	
	Validaciones de datos: No aplica	
	Escenario: Usuario en pantalla de mapa	
	Usuario ingresa: Se carga mapa centrado en la ubicación del usuario y se muestran puntos cercanos. Usuario hace presiona sobre punto: Se muestra en la parte inferior información del punto(nombre, descripción, popularidad) Usuario busca en el mapa: Se muestran datos de puntos en todo el mapa. Sin embargo, no se muestran todos los datos que existen. Se muestran datos limitados por cantidad de datos default(50)	

Tareas
<ol style="list-style-type: none"> 1. Obtener ubicación del usuario 2. Configurar api de mapas 3. Crear screen donde se muestre el mapa 4. Crear componente que va a representar un punto 5. API call para obtener todos los puntos

6. Consulta de puntos cercanos
7. Crear Card que muestra información de un punto
8. Crear botones flotantes en la vista del mapa (acceder a configuraciones y demás)

Card 200.1	Como: free user	Quiero: ver poder centrar el mapa en mi ubicación (required)
	Tal que: pueda encontrar mi ubicación.	
2	Comentarios:	
	Criterios de aceptación:	
	Validaciones de datos: No aplica	
	Escenario: Usuario en pantalla de mapa	
	Usuario hace presiona botón centrar: Se enfoca mapa en posición que da gps del dispositivo.	

Tareas
<ol style="list-style-type: none"> 1. Obtener ubicación del usuario 2. Enfocar mapa

Card 201	Como: free user	Quiero: crear puntos turísticos nuevos en el mapa. (required)
	Tal que: cualquier punto turístico pueda ser visitado por otros usuarios en el futuro, añadiendo la información requerida nombre, descripción, foto y ubicación.	
4	Comentarios: discutimos que esta opción puede ser incluida en alguna parte de la misma pantalla del mapa, ya que es una acción que le afecta, pero pensemos en que a mediano plazo el mapa va a ser más leído que escrito, por lo que se puede colocar en otra parte de la pantalla. Picker va incluido acá.	
	Validaciones de datos: Todos los campos requeridos Al menos un label Campo de descripción con un máximo de 128 caracteres	

	<p>Escenario: Usuario hace presiona botón de agregar punto</p> <p>Usuario presiona botón agregar punto: Se muestra picker, cuando se posiciona y se presiona en continuar se muestra pantalla de agregar punto con los siguientes datos (nombre, descripción, foto)</p> <p>Usuario presiona botón agregar punto y existen puntos cercanos: Se muestra picker, cuando se posiciona y se presiona en continuar se muestran puntos cercanos a la ubicación seleccionada con un botón de continuar o regresar. Si presiona continuar se muestra pantalla de agregar punto. En caso contrario se regresa a la pantalla anterior.</p>
--	--

Tareas
<ol style="list-style-type: none"> 1. Pantalla con form para creación de puntos 2. Validación de datos 3. API call con los datos 4. Creación de componente que permite seleccionar un punto en un mapa

Card 202	Como: free user	Quiero: ver el feed de fotos de un punto turístico específico de manera ordenada (required)
	Tal que: pueda decidir si visito el punto turístico o no, con base en la percepción de las fotos del lugar subidas por otros usuarios.	
4	<p>Comentarios:</p> <p>Criterios de aceptación: Validaciones de datos: No aplica</p> <p>Escenario: Usuario se encuentra en mapa que contiene puntos y presiona algún punto y arrastra la pantalla con información hacia arriba.</p> <p>Usuario presiona punto: Se muestra datos completos del lugar(nombre, imagen, descripción, check in) y todos los post asociados con este. En caso que el usuario haya visitado el lugar al menos una vez antes, se mostrará el botón de check in activado. Además se hace focus al mapa en ese punto seleccionado, es decir, el punto se ve en el centro de la pantalla.</p> <p>Usuario presiona botón cerrar: Al presionar botón la pantalla se esconde y solo se muestra la información minimizada del punto.</p>	

Tareas
<ol style="list-style-type: none"> 1. Creación de componente 2. API call to get data 3. Manejo de Navegación *

Card 203	Como: free user	Quiero: darle like un post en el feed de un punto turístico (required)
	Tal que: pueda emitir una opinión sobre los posts que más me gustan.	
2	Comentarios:	
	Criterios de aceptación: Validaciones de datos: No aplica Escenario: Usuario en pantalla de información de punto del mapa. Usuario presiona botón Like: Cuando se presiona el botón se verá en un color distinto, y en caso que se vuelva a presionar se tiene que regresar al estilo inicial del botón. Usuario ve post que anteriormente había presionado me gusta: Al ingresar a puntos de un reto en el que anteriormente se le haya dado like a un post, se debe mostrar el botón de like seleccionado.	

Tareas
<ol style="list-style-type: none"> 1. Crear componente 2. API call 3. Manejo de contador y de relación al usuario que le dio like a dicho post

Card 204	Como: free user	Quiero: confirmar la visita a un punto cuando esté cerca de él. (required)
	Tal que: pueda registrar en la aplicación la visita realizada.	
3	Comentarios: Check in. Podemos hacer que ese check-in se muestre como un nuevo entry en el feed de el punto (optional)	
	Criterios de aceptación: Validaciones de datos:	

	<p>No aplica</p> <p>Escenario: Usuario en pantalla de información de punto del mapa.</p> <p>Usuario presiona presiona botón check in: Cuando se presiona el botón este se verá en un color distinto, y en caso que se vuelva a presionar se tiene que regresar al estilo inicial del botón.</p>
--	---

Tareas
<ol style="list-style-type: none"> 1. Crear componente 2. Manejo del check-in, api call y registro en la BD

Card 205	<p>Como: free user</p> <p>Quiero: subir fotos en el feed de un punto turístico que ya visité. + reseña. (required)</p>
	<p>Tal que: pueda compartir mi experiencia en el punto turístico con otros usuarios de la aplicación.</p> <p>Esto es básicamente crear un post en el feed de un punto con información</p>
3	<p>Comentarios:</p> <p>Criterios de aceptación: Validaciones de datos: Todos los campos son requeridos</p> <p>Escenario: Usuario en pantalla de información de punto del mapa.</p> <p>Usuario presiona presiona botón agregar reseña: Cuando se presiona el botón se muestran los campos de imagen y reseña.</p> <p>Usuario presiona presiona botón agregar reseña con datos correctos: Cuando se presiona el botón se muestran los campos de imagen y reseña. Cuando los datos se completan se activa botón de publicar post. Y en caso de que el botón de check in está desactivado, este se activa y no se puede desactivar mientras haya al menos un post.</p> <p>Usuario presiona presiona botón agregar reseña con datos incorrectos: Cuando se presiona el botón se muestran los campos de imagen y reseña. Mientras no se hayan completado los datos se muestra deshabilitado el botón de agregar reseña.</p>

Tareas

1. Creación de componente con form y posibilidad de añadir foto
2. Validación de datos

Card 206	Como: free user	Quiero: saber cuántas veces he visitado un lugar. (required)
	Tal que:	
2	Comentarios:	
	Criterios de aceptación:	
	Validaciones de datos: No aplica	
	Escenario: Usuario en pantalla de información de punto del mapa.	
	Usuario en pantalla de visitas: Se muestra campo en pantalla con cantidad de visitas al lugar.	

Tareas
<ol style="list-style-type: none"> 1. Query que me dice esto. 2. Crear componente que se muestre en algún lugar de la info del punto

Módulo de gestión de retos

Card 300	Como: free user	Quiero: crear retos nice to have
	Tal que: pueda relacionar varios puntos turísticos para que sean visitados por mí o más usuarios.	
4	Comentarios: Que quede libre en este primer sprint. A futuro puede restringirse por cantidad de check ins. Para este sprint se insertarán algunos de manera manual para efectos de mostrar la funcionalidad	
	Validaciones de datos: Todos los campos requeridos Al menos 5 puntos en el reto.	
	Escenario: Usuario quiere agregar un nuevo reto	

	<p>Usuario presiona botón agregar nuevo reto: Se muestra pantalla de creación de retos.</p> <p>Usuario presiona botón de asociar punto: Se muestra una lista de retos con un buscador, cuando se hace clic sobre un punto se agrega a la pantalla.</p>
--	--

Tareas
<ol style="list-style-type: none"> 1. Crear componente con form 2. Validación de datos 3. Crear componente de búsqueda para seleccionar puntos 4. API calls (queries de puntos, creación del reto)

Card 301	Como: free user	Quiero: ver retos por estado (required)
	Tal que: pueda elegir un tour atractivo para mí, y decida visitar cada uno de los puntos turísticos que lo conforman.	
3	<p>Comentarios: disponibles, en progreso y completados</p> <p>Criterios de aceptación: Validaciones de datos: Se muestran los siguientes filtros: En progreso, Disponibles, Completados.</p> <p>Escenario: Usuario en pantalla retos.</p> <p>Usuario selecciona filtro en progreso: Se muestran los retos que se están realizando aún.</p> <p>Usuario selecciona filtro disponibles: Se muestran los retos que están completados.</p> <p>Usuario selecciona filtro disponibles: Se muestran los retos que aún no se han iniciado.</p>	

Tareas
<ol style="list-style-type: none"> 1. Componente que liste retos 2. Creación de filtro por estado (vista y API call) 3. Componente reto

Card 302	Como: free user	Quiero: completar retos (required)
-----------------	------------------------	--

	Tal que: a medida que visite los puntos turísticos que lo conforman se registre mi progreso, y me notifique cuando ya he acabado con todos los puntos.
4	<p>Comentarios: Si un usuario hace check in en un lugar, se debe verificar los retos en los que está inscrito. Si uno de esos retos contienen ese lugar, marcarlo como completado (esa parte del reto)</p> <p>Validaciones de datos: No aplica</p> <p>Escenario: Usuario hace check in en puntos</p> <p>Usuario hace check in en último punto del mapa: Se muestra pantalla pop up donde se muestra que ha completado el reto.</p>

Tareas
<ol style="list-style-type: none"> 1. Información del reto 2. Posibilidad de inscribirse en el reto 3. Manejo de progreso 4. Mostrar progreso de reto

Módulo de acceso externo

Card 400	Como: free user	Quiero: compartir fotos que subí al feed de un punto turístico en redes sociales (required)
	Tal que: pueda compartir mi experiencia en el punto turístico con usuarios de otras aplicaciones.	
3	<p>Comentarios: al abrir el enlace, ¿a dónde me lleva? Por ahora puede ser solo al app, solo si el usuario que lo presiona tiene el app instalada. Tendríamos que tener una aplicación web para hacerlo de manera diferente y que está luego lleve al app. Por ahora puede ser solo generar el enlace y que el usuario tenga que pegarlo donde desee compartirlo</p> <p>Validaciones de datos: No aplica</p> <p>Escenario: Usuario presiona botón de compartir</p> <p>Usuario presiona botón compartir: Se muestran las opciones por defectos para compartir el link del post.</p>	

	<p>Usuario presiona botón compartir nice to have: Se muestran las opciones por defectos para compartir el post en red social(instagram).</p> <p>Usuario presiona link: Redirecciona a post correcto</p>
--	---

Card 401	Como: free user	Quiero: compartir un reto completado en redes sociales (required)
	Tal que: pueda compartir mi experiencia en el tour con usuarios de otras aplicaciones.	
3	<p>Comentarios: al abrir el enlace, ¿a dónde me lleva? Por ahora puede ser solo al app, solo si el usuario que lo presiona tiene el app instalada. Tendríamos que tener una aplicación web para hacerlo de manera diferente y que está luego lleve al app. Por ahora puede ser solo generar el enlace y que el usuario tenga que pegarlo donde desee compartirlo</p>	
	<p>Validaciones de datos: No aplica</p> <p>Escenario: Usuario presiona botón de compartir</p> <p>Usuario presiona botón compartir: Se muestra el diálogo nativo para compartir un mensaje con el tour completado y la imagen del tour.</p> <p>Usuario presiona link: Redirecciona al post correcto dentro de la aplicación.</p>	

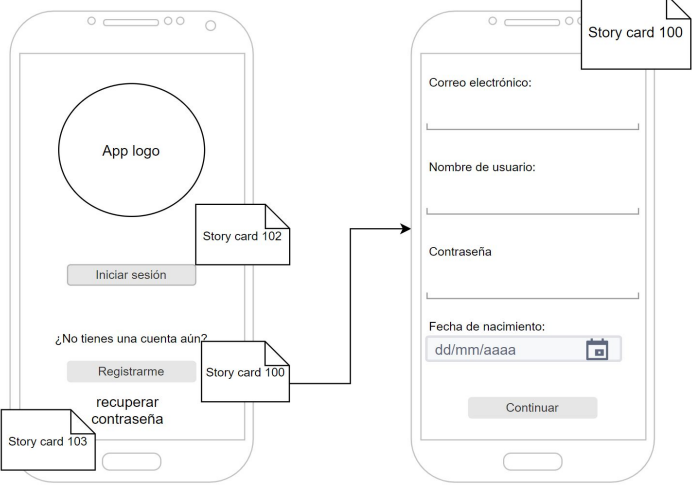
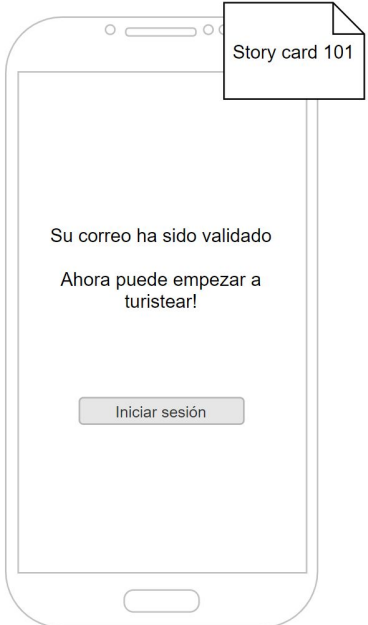
Tareas
<ol style="list-style-type: none"> 1. Acción de compartir en la app 2. Crear componente que lo maneja 3. Generación de link para compartir

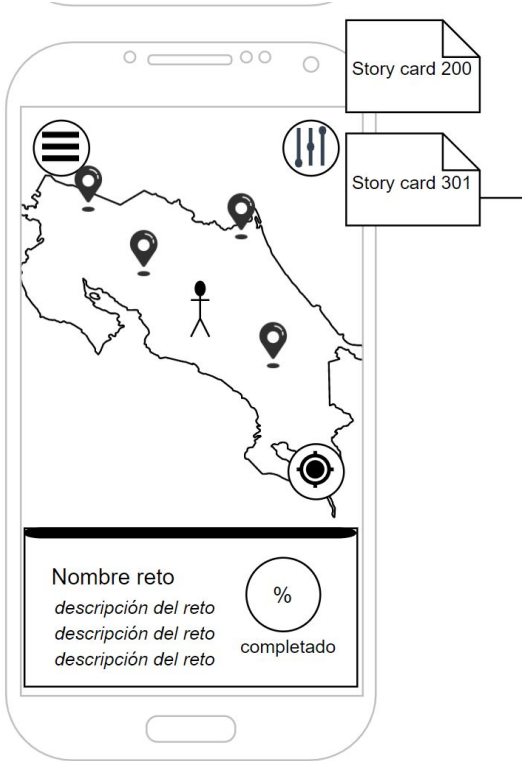
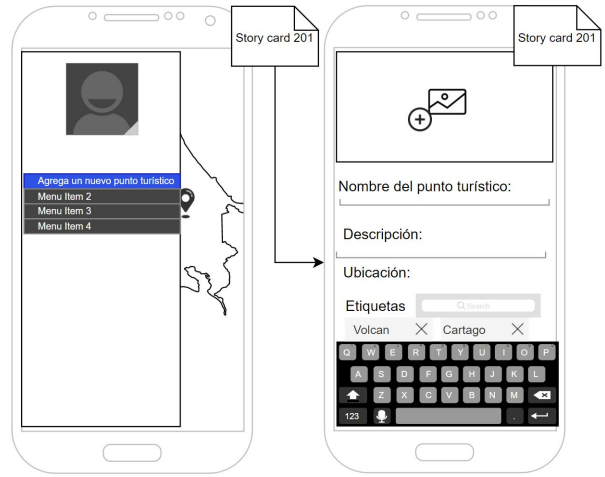
Otras tareas necesarias:

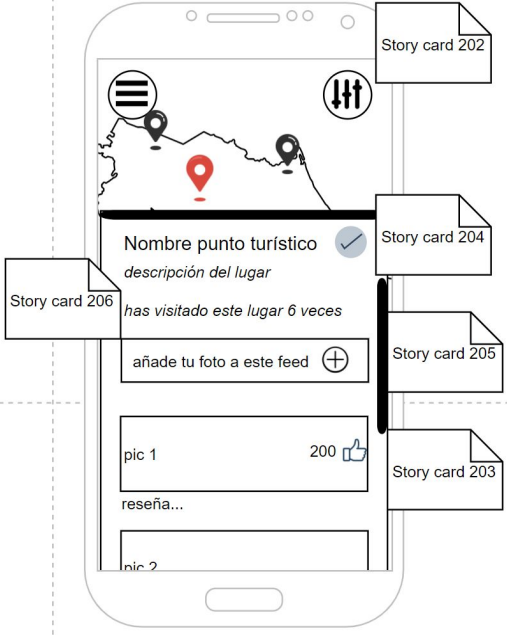
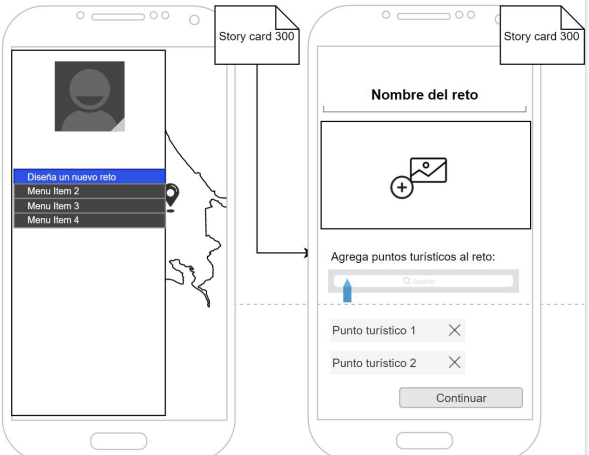
Tareas
<ol style="list-style-type: none"> 1. Configuración inicial 2. Configuración de navegación dentro de la app 3. Configuración de almacenamiento de imágenes



Anexos

Mockups

	<ul style="list-style-type: none">• Pantalla de SignUp y Login.• Opción de recuperación de contraseña.• Logo y marca.
	<ul style="list-style-type: none">• Pantalla de verificación de correo.• Se muestra una vez confirmado el email, por medio de un url.

 <p>The diagram shows a mobile app interface for the main view. It features a map of a region with several location pins. A hamburger menu icon is in the top left, and a compass icon is in the bottom right. Two story cards, labeled 'Story card 200' and 'Story card 301', are positioned above the map. At the bottom, a panel displays challenge information: 'Nombre reto', a description, a progress indicator with a percentage sign and the word 'completado', and a list of challenge descriptions.</p>	<ul style="list-style-type: none">• Vista principal del app, muestra puntos en el mapa.• Contiene un panel inferior que es arrastrable, estilo Uber.• Botones para centrar la ubicación, filtrado de retos y un menú de hamburguesa.
 <p>The diagram illustrates the process of adding a new point of interest. It shows two mobile app screens. The first screen displays a menu with the option 'Agrega un nuevo punto turístico' and a list of menu items. An arrow points from this menu item to the second screen, which is a form for adding a new point of interest. The form includes fields for 'Nombre del punto turístico:', 'Descripción:', and 'Ubicación:'. It also has a section for 'Etiquetas' (tags) with a search bar and a list of tags including 'Volcan' and 'Cartago'. A keyboard is shown at the bottom of the second screen.</p>	<ul style="list-style-type: none">• Los puntos se agregan desde el menú de hamburguesa.• Se solicita la información en un formulario y se crea.

	<ul style="list-style-type: none"> • Cuando se selecciona un punto, al hacer scroll up del panel inferior, se muestra el feed del punto elegido. • Opción de checkin, agregar imágenes, dar like a otros posts.
	<ul style="list-style-type: none"> • Los retos se crean desde el menú de hamburguesa. • Se agregan los datos necesarios a través del formulario.

 <p>The screenshot shows a mobile app interface. At the top, there are three story cards labeled 'Story card 301', 'Story card 303', and 'Story card 304'. Below them is a filter menu titled 'Estado' with a dropdown arrow. The menu is open, showing three options: 'Ver retos' (highlighted), 'En progreso', and 'Disponibles'. Below the menu, there are two challenge cards. Each card has a placeholder for a picture ('pic 1', 'pic 2'), a description field ('descripción'), and a 'Ver en mapa' button. An arrow points to the first challenge card.</p>	<ul style="list-style-type: none">• Es posible ver el filtrado de los retos por estado. Se accede desde la opción de filtro en la pantalla principal.
 <p>The screenshot shows a mobile app interface with a large '¡Felicidades!' (Congratulations!) message. Below it, it says 'Ha completado el reto X.' (You have completed challenge X.). At the bottom, there is a button labeled 'compartir en redes sociales' (Share on social media). An arrow points to the right of the button.</p>	<ul style="list-style-type: none">• Se notifica cuando se completa un reto y se genera url para compartir externamente.