

TECNOLÓGICO DE COSTA RICA



ESCUELA DE INGENIERÍA EN COMPUTACIÓN

IC8041 – Desarrollo de aplicaciones para dispositivos móviles

Proyecto 2 – Zombie Cataclysm

ELABORADO POR:

JUAN JOSÉ ARAYA CASTRO	2015138766
MAURO MENDEZ MORA	2015146966
FAURICIO NAVARRO FUENTES	2015183026

PROFESOR:
Andrei Fuentes Leiva

I SEMESTRE
CARTAGO, 2018.

Propósito de la Aplicación	3
Funcionalidades	4
Wireframes	5
Descripción de diseño de alto nivel	10
Descripción de los web services	11
Interacción con sistemas externos	12

Este documento tiene como función el externar el proyecto del juego llamado Zombie Cataclysm. Dicho trabajo se realizó como proyecto #2 del curso de Desarrollo de aplicaciones móviles en el Instituto Tecnológico de Costa Rica con el profesor Andrei Fuentes.

Propósito de la Aplicación

El proyecto Zombie Cataclysm nació a partir del deseo de que las personas que nos rodean tengan una herramienta/juego donde puedan realizar ejercicio y divertirse mientras lo hacen. Nuestro grupo de trabajo quiso desarrollar lo que viene hacer una solución a este problema, implementando una aplicación móvil de tipo juego en el cual el usuario final pueda motivarse a jugar y mientras juega, a mantenerse activo.

Para describir la aplicación, el proyecto consiste en un sistema el cual se compone de retos, estos retos están compuestos por puntos estratégicos. El usuario, tiene la opción de empezar uno de los retos e ir, con el mapa, buscando cada punto estratégico para reabastecerse de suministros. El juego está ambientado en un apocalipsis zombie, por lo tanto los retos y los premios obtenidos siguen la temática. Además mientras el usuario va caminando al siguiente punto estratégico, podrán aparecerle zombies que deberá matar o bien huir de ellos. Estas dos últimas acciones reducirán los suministros, más específicamente, balas y estamina. También, se le presenta música al usuario para que este pueda entretenerse mientras juega. Entre más retos complete, más puntos obtendrá e irá obteniendo logros.

Además del usuario anterior, la aplicación tendrá un segundo usuario el cual será el administrador del proyecto, el cual tiene el rol de mantener los datos dentro de la aplicación. Podrá agregar retos, puntos, usuarios, logros, entre otros; así mismo, podrá consultarlos, actualizarlos o eliminarlos. Para estas funcionalidades, la aplicación mostrará las pantallas correspondientes.

Los requerimientos a seguir en la elaboración del proyecto son los siguientes:

1. Desarrollo en Android nativo
2. Deben usar Material Design para el app de Android
3. Hacer un landing page de mercadeo para el app y publicarlo en Internet
4. Desarrollo de backend propio (no BaaS), que sea accesible por dominio público. Pueden usar cualquier framework para el Backend, excepto Python/Django.
5. El backend debe tener un backoffice web de administración, con su propia autenticación
6. Deben usar user login, tanto en backoffice como en aplicación
7. Deben manejar autenticación y autorización a nivel de API
8. Conexión con al menos dos APIs externos (sin contar el backend propio)
9. Usar una de las siguientes características del teléfono: GPS, acelerómetro, cámara, compás
10. El app debe tener al menos 12 activities
11. Incluir al menos:
 - a. 2 patrones de despliegue de datos
 - b. 1 patrón de búsqueda
 - c. 1 patrón de filtro
 - d. 1 patrón de ordenamiento

- e. 1 patrón de navegación
- 12. Deben manejar métricas de uso con Mixpanel y Fabric
- 13. Uso de AWS (EC2, S3 y Route53) para infraestructura (10% adicional en la nota de la progra)

Funcionalidades


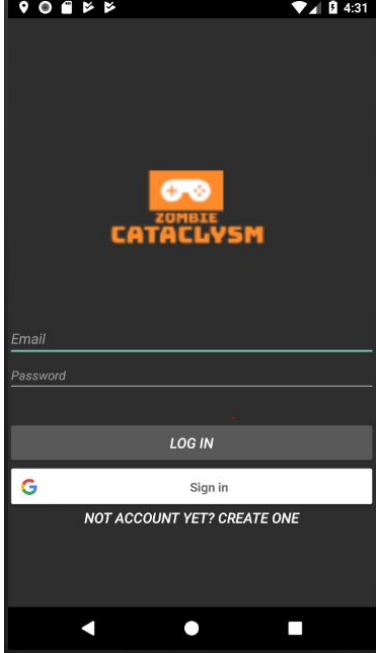
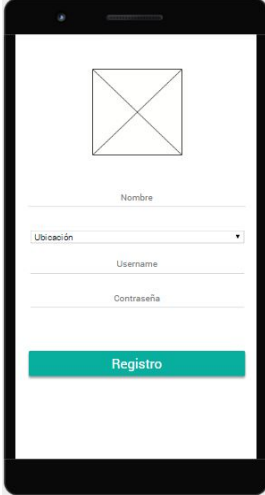
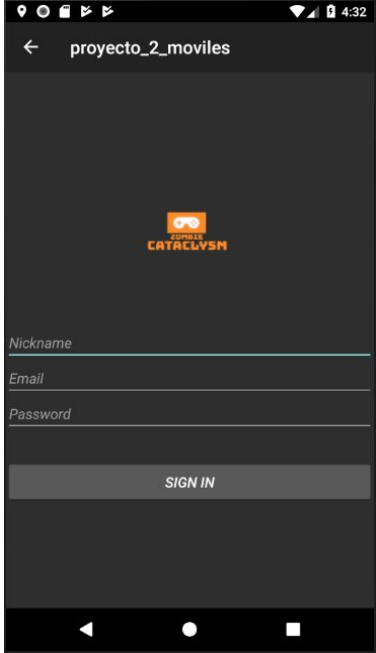
Como se mencionó en el apartado anterior, las funcionalidades de la aplicación dependerán del usuario que esté utilizando la aplicación. Sin embargo, al iniciar el proyecto todos los usuarios tendrán la función de **iniciar sesión**, ya sea con el sistema de logueo que implementamos o con una cuenta de Google.

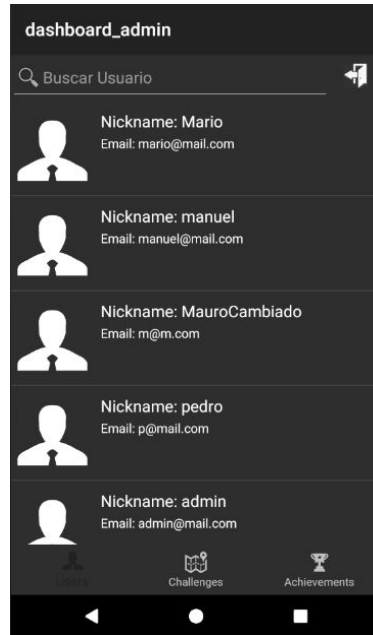
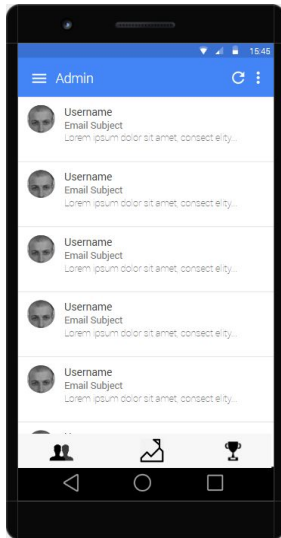
Seguidamente, el jugador podrá **visualizar los retos** que pueda empezar o jugar o que ya haya jugado y **visualizar los logros obtenidos**. Al ingresar en un reto podrá **empezar reto** esto desplegará un mapa donde muestra los puntos a los que tendrá que ir durante el reto. Podrá **buscar un reto en específico** así como todos los datos en la aplicación también tendrán esta funcionalidad. Y por último, podrá **cerrar sesión** para cerrar su cuenta. Otras funcionalidades que se presenta en el juego son: **cambiar configuración (música y vibración)**, a la hora de que aparezcan zombies **podrá disparar o huir, acumular puntos por cada punto al que llegue, ganar logros**.

Ahora bien, se encuentran una serie de funcionalidades para dar mantenimiento a los datos de la aplicación, para esto el usuario administrador, podrá ingresar a la aplicación haciendo una autenticación y podrá **crear, consultar, modificar y eliminar** los respectivos datos de **empresas de bus, rutas, paradas y choferes**.

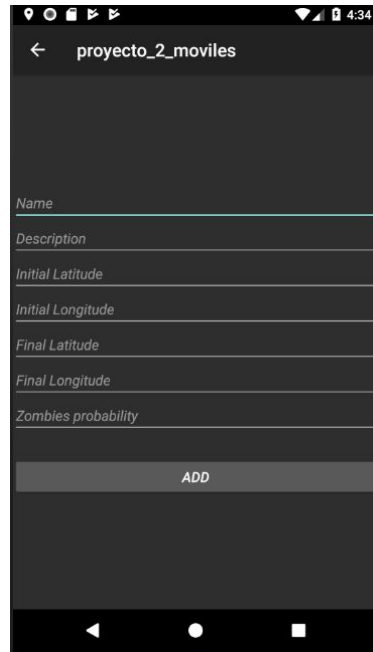
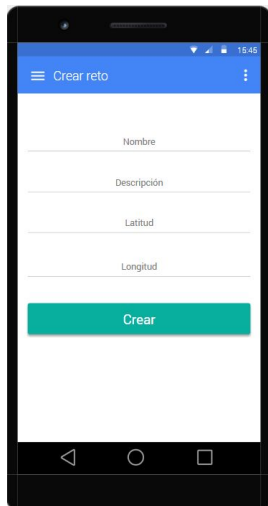
Wireframes

A continuación, se presentarán las interfaces que tendrá la aplicación. Se comparará los wireframes que se habían prediseñado con los implementados realmente.

Prediseñado	Real	Descripción
		<p>Pantalla de inicio de sesión: Se ingresa el nombre de usuario y contraseña para poder ingresar al app. La aplicación cuenta con 2 diferentes tipos de usuarios: Administrador y jugador. Además, la pantalla cuenta con la opción de crear una nueva cuenta y recuperar la contraseña. Se le agregó el logueo con Google.</p>
		<p>Pantalla que se muestra al dar click en la opción de crear una nueva cuenta. Para poder registrarse el usuario debe ingresar el nombre, ubicación, nombre de usuario y contraseña.</p>

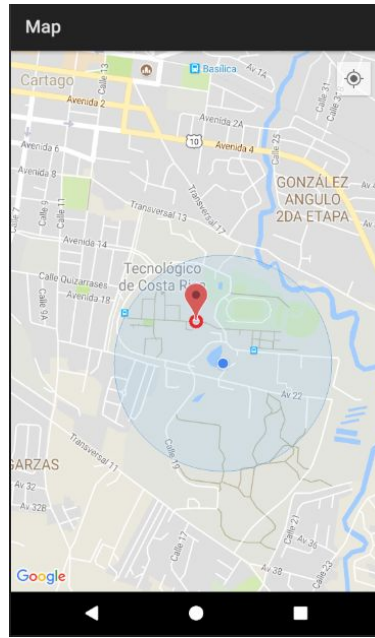
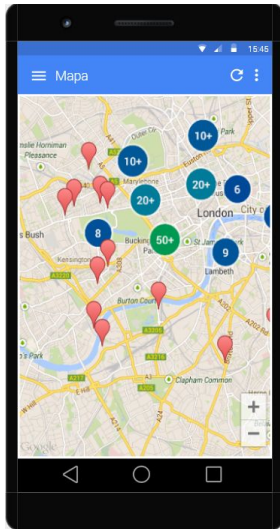


Esta pantalla sería la utilizada por el administrador la cual puede observar tanto usuarios, retos, puntos y logros que se despliega en la parte inferior de la pantalla con la posibilidad de agregar, editar o eliminar alguno de todas las posibles opciones mencionadas anteriormente.



Pantallas para agregar: Esta pantalla permite ingresar la información de un nuevo dato a la base de datos, existirá una pantalla para agregar usuario, reto, punto y logro. Ya que son pantallas con estructura similar simplemente con un cambio en los campos para ingresar información se omitieron.

		<p>Pantalla para visualizar, modificar y borrar: Esta pantalla permite modificar un dato ya existente en la base de datos, existirá una pantalla para realizar estas funcionalidades en retos, puntos, usuarios y logros. Ya que son pantallas con estructura similar simplemente con un cambio en los campos para la información existente se omitieron.</p>
		<p>Esta pantalla es visible para todos los usuarios del juego, en ella pueden ver el registro de cada uno de sus logros, además al dar click se despliega el detalle del logro obtenido.</p>



Mapa del jugador el cual muestra donde se encuentran los retos para que el usuario busque cumplirlos y así obtener premios que le serán útiles mas adelante en el juego



Pantalla de juego: Cada vez que una persona llegue a un punto en el mapa se le desplegará una pantalla de juego en la cual puede enfrentarse con los zombies o salir corriendo. Al comienzo del juego se le dará a los usuarios una cantidad de municiones o armas para poder enfrentarse a los zombies. Es importante enfrentarse y no salir corriendo, ya que cuando se acaban con todos se pueden recoger recursos y desbloquear logros.

Backoffice

Login

Login

123

Password

...

3/16

LOGIN

Login

Username:

admin

Password:

.....

Login

Name	Add
Challenges	
Points	
Users	

Site administration ▾

Tables

type_users +

type_goals +

achievements +

configuration +

users +

challenges +

goals +

challenges_and_goals_x_user +

achievements_x_user +

Name	E-mail	Edit	Delete
Nick Pettit	nick@example.com		
Andrew Chalkley	andrew@example.com		
Dave McFarland	dave@example.com		
Gull Hernandez	gull@example.com		

Select users to change. ▾

mail	password	name	challenges_completed	points	z
admin@mail.com	\$2a\$10\$Cgq4w5Yr8vvRtXvSaMy3n.qpm0oDocqV1Uswmc.d7qpVArQclwpG	admin	2	2	2
manuel@mail.com	\$2a\$08\$yRvDnbTZ6KU06H20AHJL7e.ITThD3szx6BtEMN00IUoL9vtnAP1hC	manuel	0	0	0
mario@mail.com	\$2a\$08\$MECb0ZjtTFOOXPYXgnb9JurrYmht8vIJMjrDpCd465N4w7zrnnwV	Mario	0	0	0

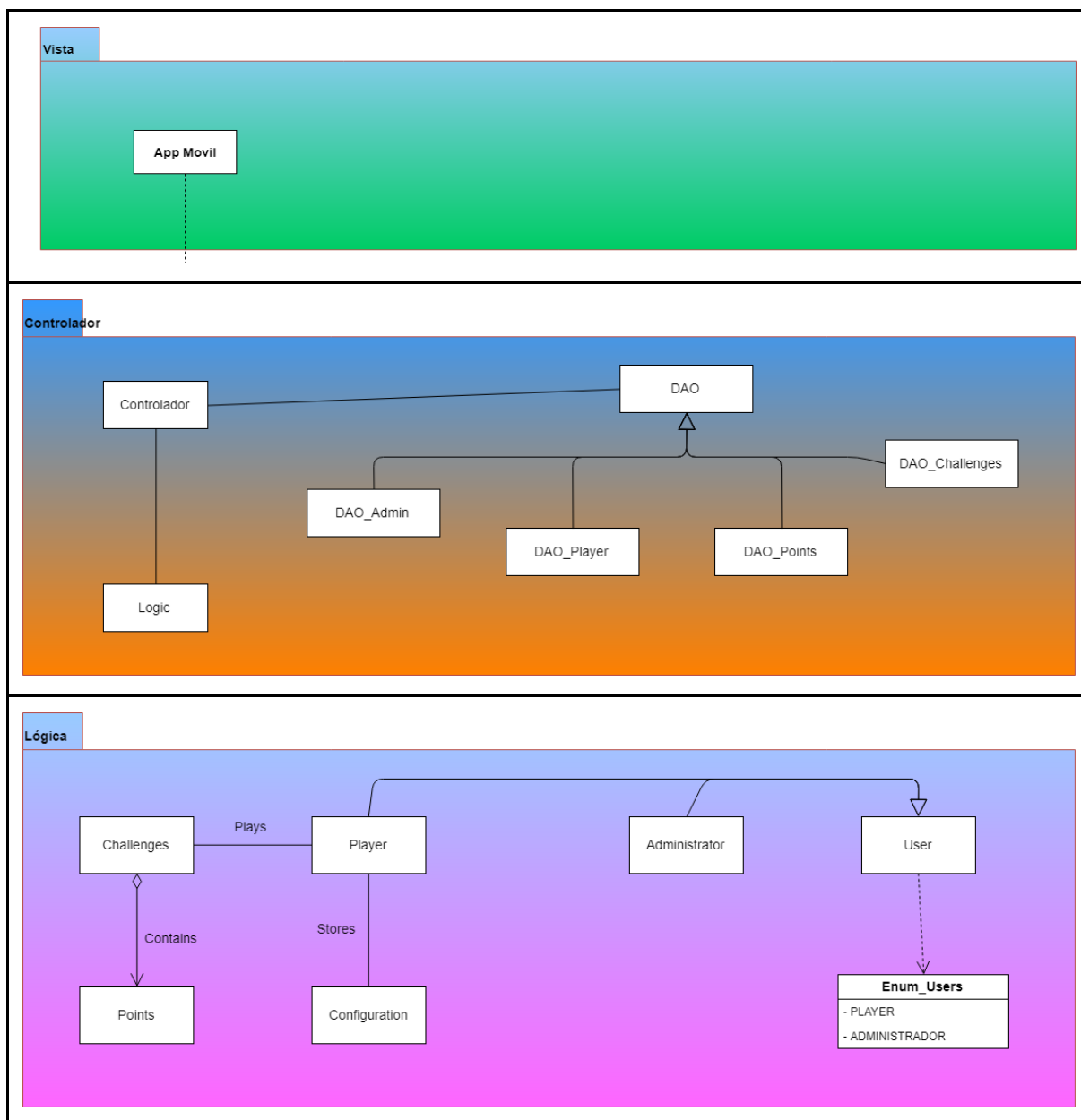
Nota Aclaratoria: Se omitieron pantallas (activities) que son esencialmente las mismas pero con diferentes datos. Por ejemplo: Visualizar logros y visualizar retos. En total se realizaron 22 pantallas.

Descripción de diseño de alto nivel

En la siguiente sección se presentará un diagrama a alto nivel que muestra cómo está organizada la aplicación internamente. El componente de Vista contiene todos los archivos correspondientes a Android, al no tener relación entre ellos, los representamos como una entidad que utiliza el backend.

El componente de Controlador contiene las entidades para representar la Lógica con la Vista, por lo tanto, utilizando los datos guardados, rellena la aplicación móvil visualmente. También está encargado de recibir las peticiones del usuario desde la Vista para mantener la base de datos actualizada.

El componente de Lógica contiene las identidades que representan los datos de manera persistente, este componente contiene las relaciones y atributos que se almacenan en la base de datos.



Descripción de los web services

Para el API se utilizó Node.js y Express como la plataforma de la aplicación debido a la facilidad que ofrece a la hora de realizar el Restful API y su popularidad en la actualidad. Nuestra aplicación utiliza distintas aplicaciones en Node.js y Express: El backend, donde se puede acceder a la base de datos por medio de requests, un landing page para dar información de la aplicación y un backoffice que es autogenerado con la herramienta Express-admin.

El backend ofrece varias funcionalidades:

- Crear: Se utilizará la ruta definida posteriormente, con un request POST, deberá tener entre sus headers el token del usuario y en el body los campos respectivos para la base de datos.
- Obtener Todos: Se utilizará la ruta definida posteriormente, con un request GET, deberá tener entre sus headers el token del usuario.
- Obtener Solo uno: Se utilizará la ruta definida posteriormente, con un request GET, añadiendo un "/" seguido del identificador único del dato a consultar. Deberá tener entre sus headers el token del usuario.
- Modificar: Se utilizará la ruta definida posteriormente, con un request PUT, añadiendo un "/" seguido del identificador único del dato a modificar. Deberá tener entre sus headers el token del usuario y en el body los campos respectivos para la base de datos.
- Eliminar: Se utilizará la ruta definida posteriormente, con un request DELETE, añadiendo un "/" seguido del identificador único del dato a eliminar. Deberá tener entre sus headers el token del usuario y en el body los campos respectivos para la base de datos.
- Ingresar: Se utilizará la ruta definida posteriormente, con un request POST, hará que el backend autorice al usuario y devuelva un token. Deberá tener en el body los campos respectivos para la base de datos.

Dentro de la aplicación se encuentran diferentes rutas:

- /api/users: esta ruta controlará el mantenimiento de los datos de usuarios.
- /api/challenges: esta ruta controlará el mantenimiento de los datos de retos.
- /api/achievements: esta ruta controlará el mantenimiento de los datos de logros.
- /api/type_users: esta ruta controlará el mantenimiento de los datos de los tipos de usuario.
- /api/type_goals: esta ruta controlará el mantenimiento de los datos de los tipos de puntos pertenecientes a los retos.
- /api/goals: esta ruta controlará el mantenimiento de los datos de puntos que pertenecen a los retos.
- /api/configurations: esta ruta controlará el mantenimiento de los datos de configuración de un usuario.
- /api/challengesGoalsUsers: esta ruta controlará el mantenimiento de los datos para saber cuáles retos han sido jugados por un jugador.
- /api/achievementsUsers: esta ruta controlará el mantenimiento de los datos para saber cuáles logros han sido desbloqueados por un jugador.

Un ejemplo de una ruta para un request es la siguiente:

<http://ec2-13-59-238-74.us-east-2.compute.amazonaws.com/api/users>

Interacción con sistemas externos

En esta sección se detallaron los sistemas en el entorno del proyecto. Dichos elementos son los siguiente:

- Google:
 - Google Maps: Se utiliza el API de Google para poder realizar el detalle del reto y los puntos a llegar para el jugador.
 - Google Login: Se utiliza el API de Google para loguearse en la aplicación sin la necesidad de crearse una cuenta utilizando la aplicación. Sin embargo, esta funcionalidad se permitirá solo para los jugadores, los administradores tendrán que usar el logueo propiciado por la aplicación.
- AWS: Se utilizó AWS para el alojamiento de la aplicación, de esta forma está en la nube y puede ser accesada desde cualquier lugar remotamente, utilizando la aplicación. Más específicamente se está usando EC2 para hostear el server y RDS para almacenar la base de datos, las dos con las máquinas que ofrece el free tier.
- Mixpanel: Se utilizó el sistema Mixpanel para tener métricas de uso en la aplicación, con este sistema podemos hacer rastreos de eventos en la aplicación y así saber estadísticas de la aplicación; por ejemplo: qué elementos visuales se usan más que otros.
- Fabric: Al igual que Mixpanel, Fabric nos permite tener métricas de uso de la aplicación y también un rastreo de los errores que se le presenten al usuario.