

---

# *Avalanche Escape*

El (*casi*) juego del año

---

Javier Alejandro Lougedo Lorente [javier.lougedo@estudiante.uam.es](mailto:javier.lougedo@estudiante.uam.es)

Salvador Martín Barcia [salvador.amaya@estudiante.uam.es](mailto:salvador.amaya@estudiante.uam.es)



Proyecto de Sistemas Informáticos Grupo 1361 - Pareja 2  
Ingeniería Informática - Universidad Autónoma de Madrid  
Madrid, España  
13 de diciembre de 2019

## Índice

<b>Introducción</b>	<b>2</b>
<b>Guía de uso</b>	<b>2</b>
<b>En un mundo ideal...</b>	<b>3</b>
<b>Conclusiones</b>	<b>4</b>

## Introducción

BIENVENIDOS, AMIGOS MIOS! En esta pequeña guía de funcionamiento podréis ver como utilizar nuestro maravilloso juego, Avalanche Escape.

¿Recordáis aquel juego del ratón y el gato? ¡Pues es lo mismo, pero con una temática algo más navideña para celebrar con algo de alegría las festividades futuras! Además, esta vez, vuestro objetivo como cazadores (o como avalancha) no será otro que... ¡Cazar a un alpinista! Nada de encerrarle, ¡ahora hay que arrollar con el! ¡En Avalanche Escape y en la vida, arrasa con lo que veas y generoso no seas!

Hechas las presentaciones, pasemos a explicar como funciona nuestra página.

## Guía de uso

En primer lugar, vamos a comentarte una cosa: el uso ideal de nuestra página es en perspectiva apaisada y pantalla competa. Así podrás vivir una experiencia hermosa y llena de calidad. Sin ser completamente *responsive* y tener una perspectiva “bonita” para móviles, tenemos una representación ahí también, y la perspectiva es adaptativa.

Bien, cuando accedes por primera vez te encontrarás con un navideño menú. Es posible que, si tienes algún tipo de *adblocker* activado, no visualices la inmersiva nieve adecuadamente. Si es así, te recomendamos desactivarlo para poder ver correctamente la página: prometemos no llenar de publicidad tus ojos.

En el menú inicial podrás observar arriba a la izquierda un botón con forma de casa que te lleva inicio, el nombre de la web. Junto a este, están los nombres de sus creadores, y links a sus currículums. Arriba a la derecha, observarás, si estas registrado, tu nombre (ten cuidado, **BAJO NINGÚN CONCEPTO** pulses en tu nombre, no nos hacemos responsables del resultado), junto a un botón de cerrar sesión. Si por el contrario no hemos iniciado sesión, lo que nos encontraremos será un botón para iniciar sesión y un pequeño biberón para crear una nueva cuenta.

Abajo, en el menú inicial, a la derecha, nos encontramos con el contador de errores, tanto en sesión como global. Por último, la joya de la corona, nuestro menú desplegable en avalancha. Si clickamos en el mismo, invadirá nuestra pantalla. Tras la carga del mismo, podremos cerrarlo pulsando el botón inferior izquierdo con una cruz.

En el menú podremos acceder, según el orden de tamaño de cada una de las franjas del menú, a lo siguiente:

- Ver juego: en esta pantalla podremos escoger el juego pasado que queramos seleccionar. Además, tenemos una amigable perspectiva que nos muestra si vencimos o fuimos derrotados en dicha batalla.
- Escoger: en esta esta pestaña se nos permite navegar entre los distintos juegos que tenemos activos actualmente, y a los que pertenecemos, para poder jugarlos luego en *Ir al juego seleccionado*.
- Continuar viendo: nos dará la opción de continuar viendo la última partida cargada en sesión.

- Jugar como alpinista: nos ofrecerá una lista de partidas disponibles a las que unirnos como presa de la montaña.
- Jugar como avalancha: iniciaremos una nueva avalancha en busca de un abandonado y solitario alpinista.
- Ir al juego seleccionado: cargará la última partida que tengamos en sesión. Si no tenemos ninguna, nos devolverá un pequeño error (hecho en parte para poder observar con facilidad la pestaña de error), recordándonos que debemos seleccionar un juego antes de poder jugar.

Eso serían cada una de nuestras características. Más en detalle, vamos a observar algunas de las más interesantes cosas de nuestra web:

- A.** En primer lugar, nuestra nieve, programada en JavaScript, CSS y HTML, con ayuda de los canvas que no habíamos utilizado hasta ahora y de código libre en CodePen.
- B.** En segundo lugar, nuestro *drag & drop* en el juego, que nos permite jugar y comprobar los movimientos, “capando” a un usuario que no deba hacer nada y permitiendo únicamente las consultas “básicas” a fin de reducir el número de consultas al servidor, dando un poco más de trabajo al cliente.
- C.** En tercer lugar, nuestro espectacular menú que habrás podido observar, además de nuestra gracia y estilo para realizar ciertos detalles de la web.
- D.** Y por último, y no menos importante, todos esas pequeñas cosas, como los botones, selección de colores y demás que hemos tratado de hacer lo mejor posible, para que **TU** experiencia sea **ESPECTACULAR**.

Vistas las características de la página, de las cuales creemos que no nos olvidamos de nada, damos paso a nuestra bajada de pantalones, que es de lo más adecuada, correcta y requerida, y comentamos todo aquello que no hemos hecho o que no funciona como queríamos, además de las ideas que teníamos para hacerlo mejor.

## En un mundo ideal...

Bien, empecemos con lo que querríamos haber hecho:

- En primer lugar, nos habría gustado introducir como extra que los listados funcionasen con **AJAX**, de manera que actualizasen únicamente el listado como tal. No lo hemos hecho por dos razones: falta de tiempo y de dominio en AJAX, con lo que no hemos querido liarnos.
- Nos habría gustado también que nuestro tablero funcionase adecuadamente y no lo mal que funciona, con algo más de estilo. Además, actualmente tienes que actualizar la ventana aunque actualice el movimiento, debido a que, pese a actualizar la vista, no actualiza el JQuery de los dragables y demás, quedando la pestaña algo “boba” e inútil. De ahí que hayamos metido la pestaña de espera, donde al usuario le da la sensación de actualizar y aseguramos que vaya todo algo más fluido, disimulando ese pequeño error sin necesidad de recargar continuamente toda la página. Lo ideal habría sido que funcionasen como un emisor y un receptor (el servidor avisa

al usuario que espera cuando esta listo y hace una recarga), pero no supimos implementarlo adecuadamente, y el traer un solo marco nos dio pequeños errores.

Comentado eso, pasamos a ver cosas que no hemos realizado de la práctica:

- No filtramos los resultados, debido a que nos pareció algo complejo y no teníamos claro como hacerlo, por lo que optamos por plantear algo como veremos a continuación, pero no logramos llevarlo a cabo. Nuestra idea era hacer una función en *views.py* que recibiese un listado y una serie de filtros que aplicar como parte del método POST o GET, y posteriormente devolviese ese mismo listado organizado con dichos filtros.
- No hemos realizado tests adicionales, de nuevo, pero por una buena razón, y es que la poca nueva funcionalidad implementada es fácilmente comprobable en una partida: que el alpinista sea cazado (o se suicide) frente a una bola de nieve, o que el mismo llegue a la cima de la montaña.
- No actualizamos correctamente el juego, como ya hemos comentado anteriormente.
- No pasamos algunos tests de models, debido a que hemos modificado models.py, y ahora mismo tenemos 2 estados de finalizado por comodidad. Tampoco pasamos algunos tests de testservice-p4, debido a que nuestra condición de victoria es distinta. En lugar de comprobar que el alpinista haya sido encerrado, lo que hacemos es comprobar si este y una bola de nieve comparten casilla. Si es así, gana la avalancha. Si quisiésemos “encerrar” al alpinista, sería tan sencillo como quitar la condición de occupied (que teníamos en la anterior práctica y comprobaba que una casilla no estaba ocupada antes de moverse) y posteriormente deberíamos hacer la comprobación pertinente. Sin embargo, no quisimos hacerlo así, ya que el juego perdía gracia si no te permitía suicidarte, aunque lo hace todavía más difícil para el alpinista.
- No hemos pasado Flake8, porque las principales directivas de corrección a la hora de programar las cumplimos y no queríamos comernos la cabeza con cosas que se pasasen de la longitud de línea o detalles que quitasen cierta legibilidad a nuestro código.
- Por último, no hemos juntado todos los listados en una única pestaña ni en una única función de *views.py*, debido a que queríamos tener un poco de fragmentación en ambos lugares, a fin de tener la información más organizada. Además, modificamos levemente el estilo entre listados para que el usuario tenga una idea intuitiva de donde se encuentra en este momento.

Dicho esto, podemos pasar ya a las conclusiones, mencionando que hemos cumplimentado el requisito de drag & drop, aunque con la falta de que se actualice apropiadamente, cosa que sabemos que es debida a que no termina de cargar los objetos del JQuery correctamente. Cabe mencionar también que nuestro admin funciona adecuadamente, y tiene un perfil precargado para alumnodb y alumnodb, por si se quiere observar.

## Conclusiones

Creo que estas serían todas las consideraciones a tener en cuenta. Seguramente hay alguna cosa que se nos olvide, pero en cualquier momento se nos puede contactar al número +34 686 17 92 12, vía What'sApp o llamada, y responderemos cualquier duda de funcionamiento encantados!

Sobre la práctica, comentar que nos esforzamos excesivamente en un CSS que al final ha acabado resultando algo pesado, pero más que satisfactorio y con el que sentimos haber aprendido muchísimo del proceso para llevarlo a cabo y que nos satisfaga, cogiendo ideas e implementandolas.

Por otro lado, nos llevamos bastante bien con AJAX y JavaScript, no han acabado siendo un problema y han facilitado nuestras misiones. Lo que más nos ha costado entonces ha sido el hecho de estar vinculados con el servidor y el “linkar” las cosas entre Python y HTML, sitio donde más errores hemos tenido.

Lo que sí está claro es que lo hemos pasado de 10, hemos aprendido muchísimo y hemos visto que somos más de FrontEnd que de BackEnd, lo que dado nuestro campo de estudio resulta algo “triste”.

¡Eso ha sido todo! Un cordial saludo: Salvador y Javier, los credores de Avalanche Escape.