

SimuPark: Simulando Un Parque De Diversiones

- Paula Catalina Hernández Ramírez 2180048 - Moderador
- Juan Pablo Claro Pérez - 2181707 - Relator
- Daniel David Delgado Cervantes - 2182066 - Expositor

Resumen

El presente documento examina una problemática que se presenta en cualquier lugar donde se encuentre una cantidad excesiva de personas y haya la necesidad de hacer filas. La teoría de colas es un estudio matemático que busca entender el comportamiento de las líneas de espera dentro de un sistema, donde el cliente presenta una solicitud para que se le presten un servicio y este a su vez dependerá de una variante, la demanda, ya que con esta se determina el tiempo de espera que debe tener la persona hasta obtener lo que requiere. El ecosistema a estudiar se presenta dentro de los parques de diversiones, en los cuales se presentan problemas a la hora de ingresar a diferentes atracciones, donde las personas pueden durar mucho mas tiempo realizando las colas que divirtiéndose en los juegos. Situación que además de desafortunada para el cliente, también lo será para el parque de diversiones, pues entre más cómoda sea la espera a las atracciones, mayores serán las probabilidades de que la persona quiera volver al parque. Es por ello que se plantea: ¿Existirá alguna forma para que las personas puedan divertirse más y así disminuir el tiempo de espera en las diferentes atracciones?

Introducción

La teoría de líneas de espera inicia con el trabajo de Agner Krarup Erlang matemático Danés, el cual en el estudio de líneas telefónicas obtuvo la fórmula para la distribución del número de líneas de espera, a partir de aquí la teoría de colas se aplica en un sinnúmero de estudios encargados del comportamiento de sistemas como el tráfico automovilístico, las filas de un banco o el flujo de mensaje.

El objetivo principal de la teoría de colas es modelar el sistema que presenta estas líneas de espera, con el fin de observar el comportamiento y posibles alternativas de solución para obtener el resultado más óptimo según el estudio que se realice. Teniendo esto presente, la teoría de colas será fundamental para el desarrollo del estudio, este se centrará en el sistema que conforma un parque de diversiones, donde se concentra una población grande de personas, las cuales desean probar cada atracción presente.

Este sistema a estudiar presenta múltiples variables ya que no se van a presentar los mismos escenarios, puede influir ya sea la hora del día, un fecha especial en el año, el clima, los precios de cada una de las atracciones, la fama de cada juego, entre otras. Las ferias encargadas de los parques de diversiones han aplicado una transformación al sistema donde se usa el Fast Pass, este es un método para agilizar las colas, pero ¿Es efectivo? ¿Las personas invierten menos tiempo en esperas y más en divertirse?

Formulación Del Problema

Uno de los principales problemas que se presentan en cualquier tipo de cola, o fila, está en el tiempo de espera que implica estar en estas. Este tiempo de espera depende de 2 variables: la tasa de servicio, o el tiempo en el cual sale gente de la fila; y la tasa de llegada, que se refiere a la velocidad con la que ingresan personas a la fila. Es a partir de esto que funcionan todas las colas.

Los parques de diversiones son centros de recreación que atraen a multitudes de personas que desean disfrutar de las distintas gamas de atracciones que se manejan. Siendo así, uno de los problemas a los que tienen que enfrentarse tanto los visitantes son las largas colas que se pueden presentar en muchas de las atracciones. Esto, en el peor de los casos, puede generar descontento en los visitantes debido a la gran cantidad de tiempo que perdieron esperando, lo que a la larga, podría afectar la reputación del parque y las ganancias del mismo.

Partiendo de esto, el problema que se busca solucionar son estas mismas colas, y las diferentes técnicas que se pueden emplear para poder reducir el tiempo de espera promedio de los visitantes. De esta manera, se busca mejorar la experiencia general del público asistente que, en respuesta, implicaría una mejora de los ingresos generados por el parque.

Objetivos

Objetivo General

- Analizar el comportamiento de las filas de las atracciones de un parque de diversiones a partir de simulaciones computacionales para determinar qué estrategias son más efectivas para la reducción del tiempo de espera promedio.

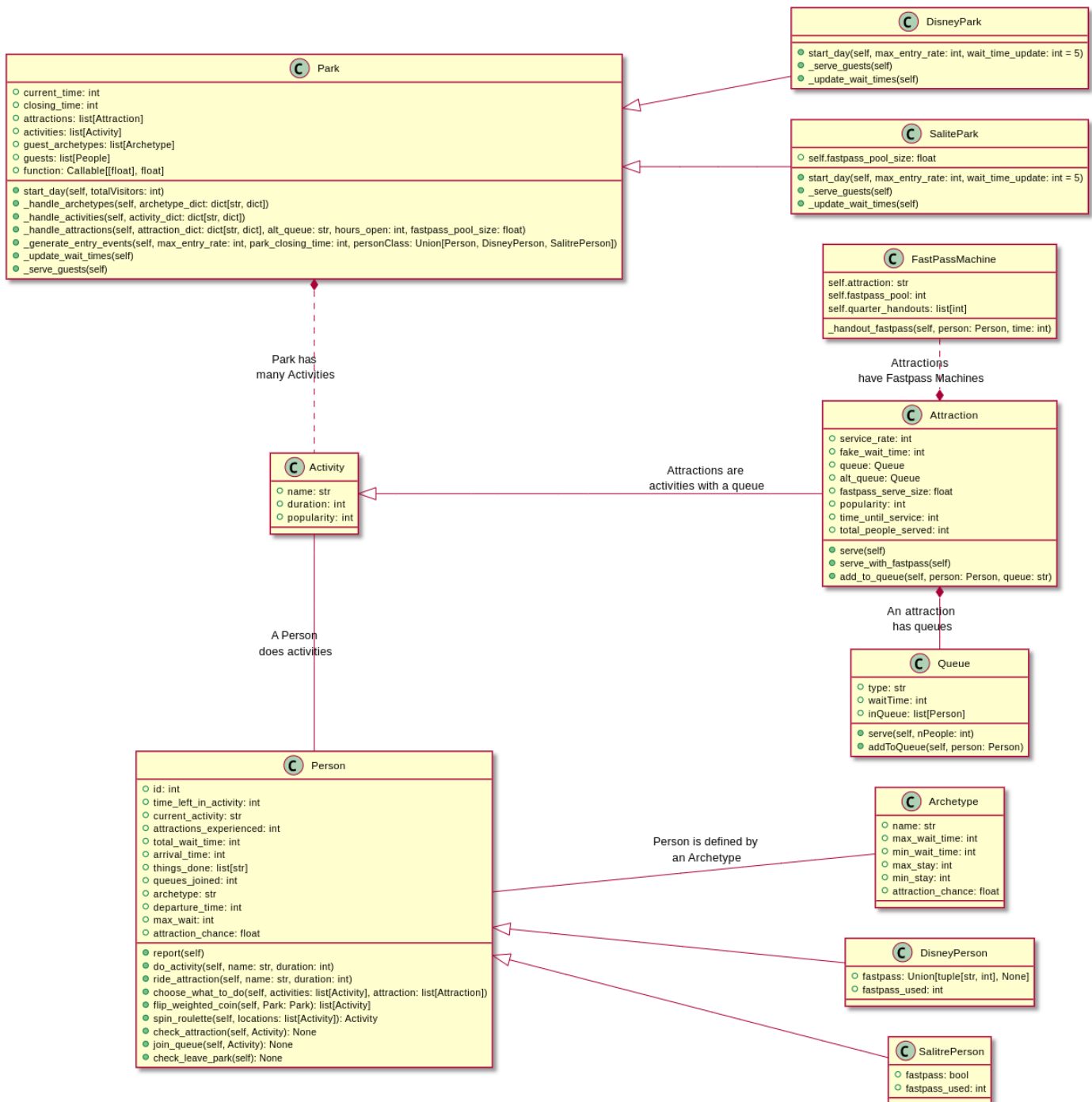
Objetivos Específicos

- Determinar qué estrategia conlleva la mayor reducción del tiempo de espera en las colas del parque.
- Identificar los diferentes factores que se relacionan con el tiempo de espera promedio de cada una de las atracciones del parque.
- Determinar las diferentes limitaciones que presenta el parque en cuanto al manejo de las colas de cada una de sus atracciones.
- Analizar qué tanto se puede incrementar el flujo de visitantes a un parque de diversiones y si la cantidad de atracciones en las que se subieron aumentó con un cambio de estrategia *FastPass*.

Conceptualización del modelo

El modelo empleado como punto de partida en el estudio de la simulación del comportamiento de las colas de espera presentadas en un parque de atracciones se puede ver simplificado en el siguiente diseño, el cual cuenta con ciertos componentes en los que se involucra la conducta o 'arquetipo' de cada persona a la hora de recurrir a las diferentes atracciones, las actividades que la persona podría elegir, las atracciones, y muchas más variables que influirán en la ejecución de la simulación:

SimuPark: Class Diagram



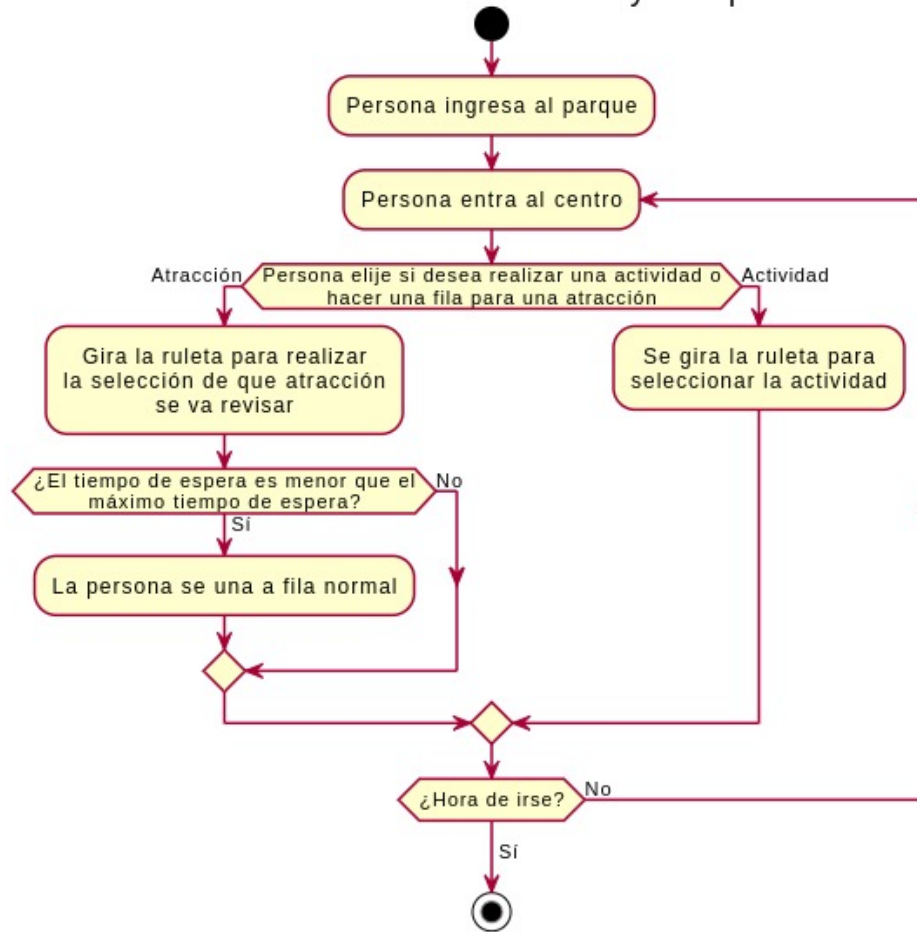
Se inicia con el estudio de las diferentes personas, sus gustos y arquetipos para obtener una lectura que permita crear un mapa global de cada una de las distintas opciones que puede elegir las personas, tomando el arquetipo como punto de partida para obtener los gustos que pueden presentar los visitantes en el modelo.

El parque de atracciones contará con la entidad de actividades que determina cada una de las opciones que pueden seleccionar las personas, teniendo en consideración que cada actividad presente puede ser una atracción distinta, la cual cuenta con tiempo de espera y un tiempo de servicio en la que cada persona puede participar.

El concepto estudiado en modelo parte desde un caso base presente en cada uno de los parque de atracciones en el cual no se emplean técnicas para mejorar la conducta de las colas sino empleando el sistema FIFO, este caso base se centra en un flujo donde la persona al ingresar al parque selecciona si desea realizar una actividad o si por el contrario decide emplear tiempo en la cola de espera de una de las atracciones para disfrutar de ella, luego de esto la persona decide si desea realizar otra

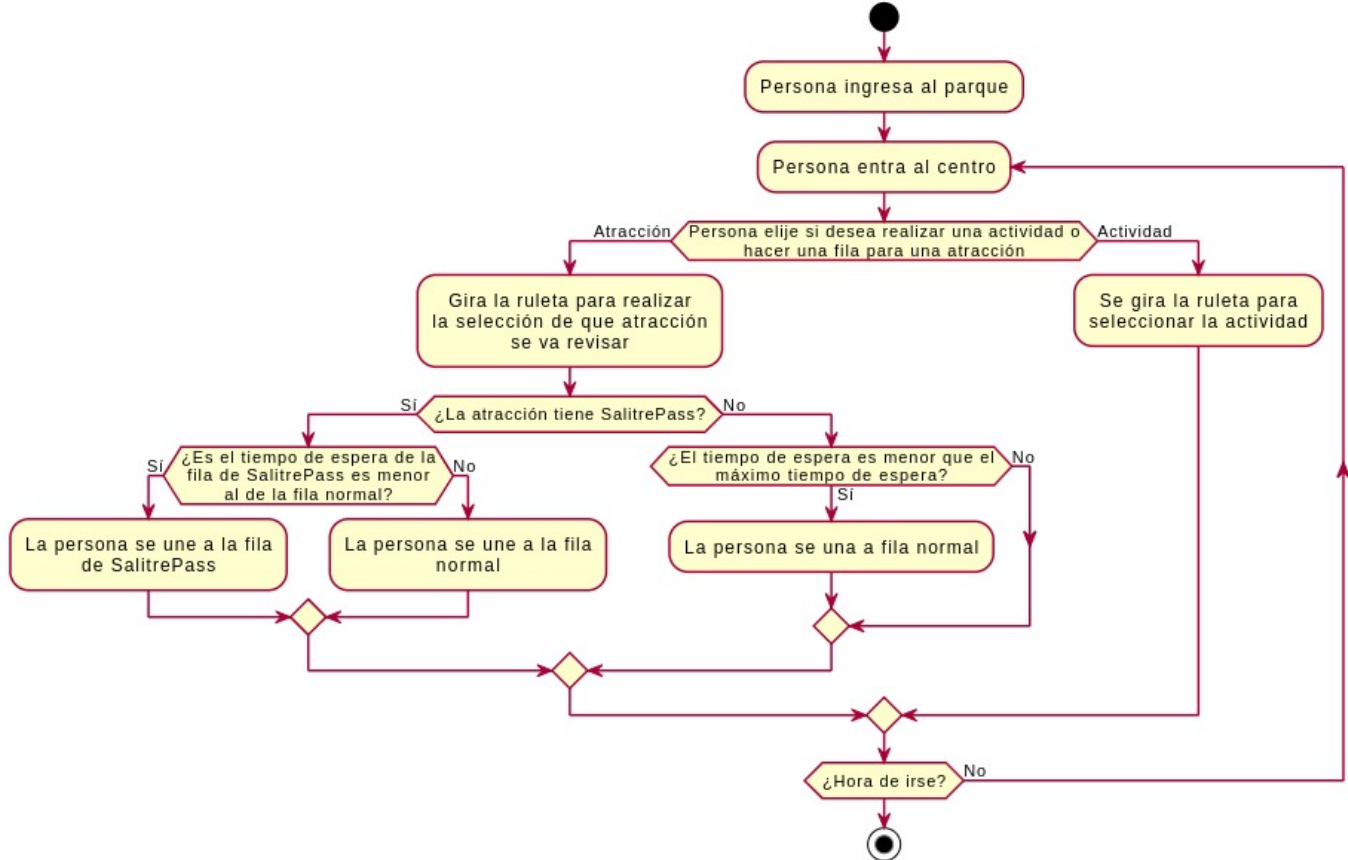
actividad o ingresar a otra atracción hasta que esta decida irse, teniendo como resultado el siguiente diagrama:

Diagrama de Flujo: Conducta de las Personas Caso Base y Tiempos Falsos



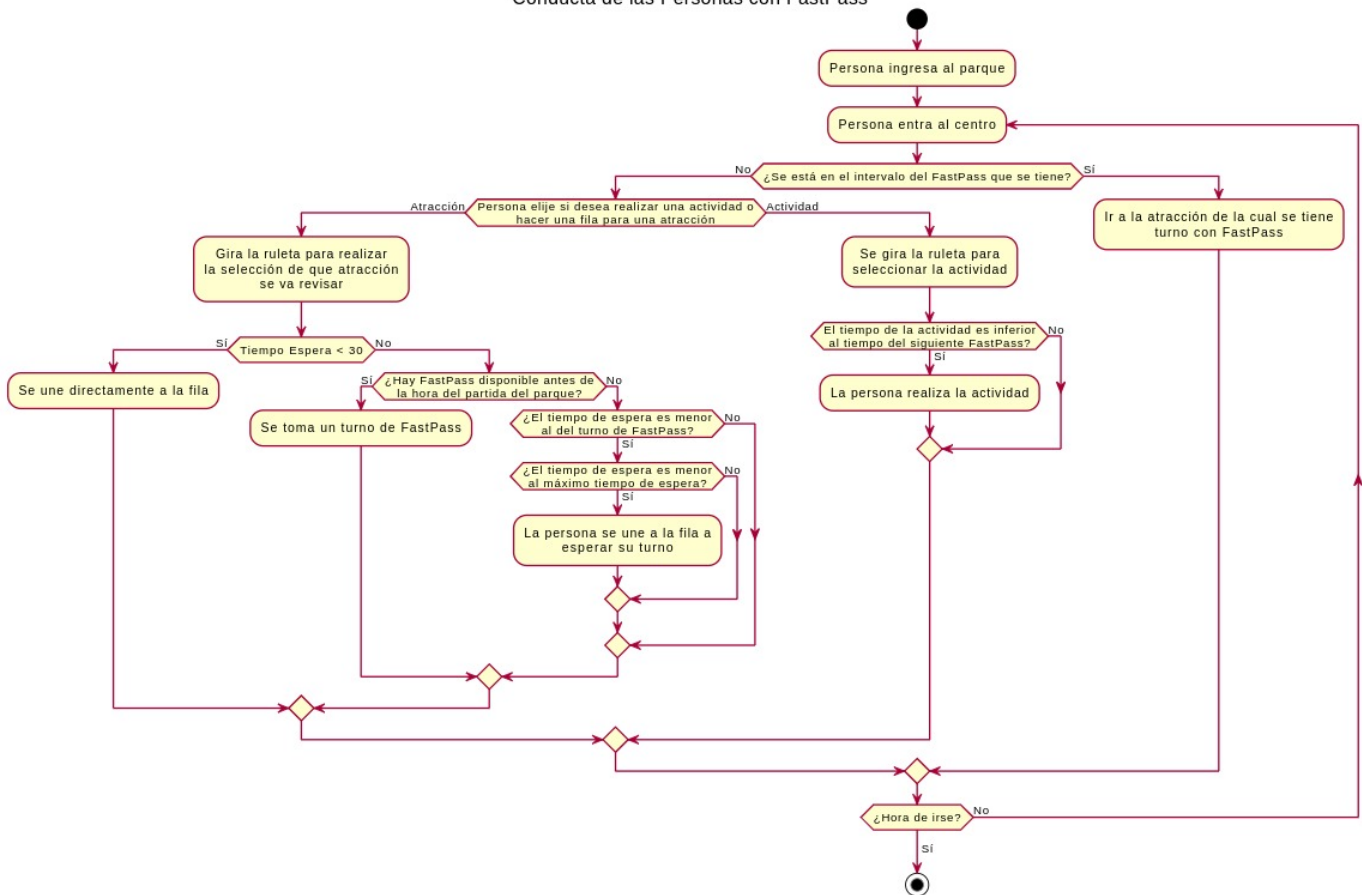
A su vez se desea comparar este modelo tradicional del funcionamiento de los parques de atracciones con otros modelos que emplean tácticas para que sus sistemas de espera faciliten la movilidad de los usuarios, teniendo presente esto, el modelo con un diseño como el del SalitrePass presenta una modificación en los tiempos de espera presente en cada una de las atracciones. La persona puede elegir si entrar a una fila normal o la fila perteneciente al SalitrePass, esto depende del tiempo de espera presentante en cada una y de la decisión de la persona.

Diagrama de Flujo: Conducta de las Personas con SalitrePass



Otro modelo de comparación es el empleado en los parques de diversiones de Disney donde presentan una facilidad llamada *FastPass* donde la persona que presenta el *FastPass* debe presentarse frente a la atracción a la hora especificada en el comprobante, esto con el fin de que la persona no realice filas sino que ingrese directamente a la atracción o actividad, de lo contrario decidirá si quiere unirse a la fila, comprar un *FastPass* para la atracción o realizar otra actividad, teniendo como resultado el siguiente flujo:

Diagrama de Flujo:
Conducta de las Personas con FastPass



Desarrollo

Recolectando Y Procesando Datos

Una de las partes más importantes en cuanto se refiere a la construcción de nuestro modelo de simulación, está en la recolección de los datos necesarios para poder estar lo más cerca a el comportamiento real. En este caso, esto no es la excepción.

Para poder realizar el modelado de las diferentes componentes del parque de diversiones de nuestra simulación, requeríamos conocer, o por lo menos aproximar, 3 parámetros los cuales afectan de gran medida los resultados finales. La función de llegada de los visitantes al parque, la cual determina la cantidad de personas que llegan a parque dependiendo la hora del día; los arquetipos de las personas, los cuales determinan el comportamiento de las personas ya en el parque; y las principales características de las atracciones, principalmente la tasa de servicio y la popularidad de las mismas. Siendo así, se llevaron acabo diferentes procedimientos para la determinación de estos parámetros.

Determinando La Función De Entrada Al Parque

La principal necesidad de determinar la función que describa el comportamiento de la entrada de las personas al parque, está en el como este afecta los tiempos de espera de manera general dentro del parque.

!!!! Expandir esto, me quedé sin ideas. Me estaba doliendo la cabeza !!!!

Lo primero a realizar es cargar nuestro dataset. Como tal, este dataset fue recolectado de diferentes páginas web las cuales indicaban una especie de nivel de "llenura". Aunque se han de reconocer las limitaciones de la información que verdaderamente provee, debido a la escasez de los datos que hay sobre las tasas de entrada en muchos de los parques, es lo más cercano que se tiene a un reporte de la manera en la que las personas visitan el parque.

```
entry_df <- read.table("https://gitlab.com/MrDahaniel/proyecto-simulacion/-/raw/main/datasets/crowd
LevelsMagicKindom.csv", header = T)
```

```
knitr::kable(head(entry_df))
```

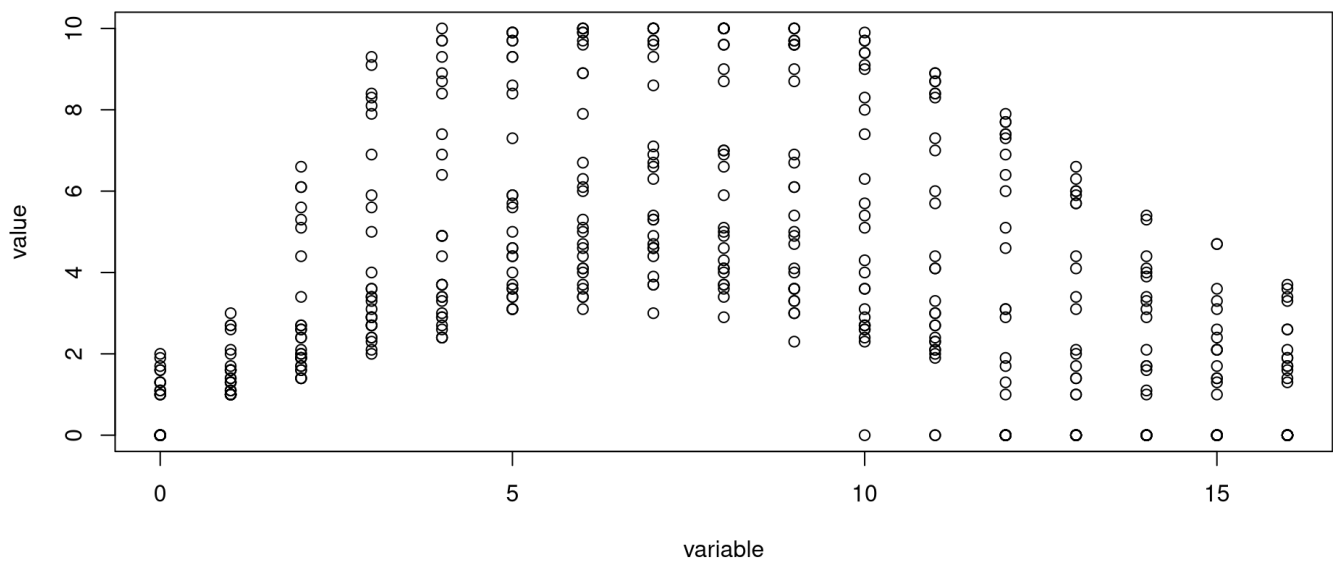
| Park | X8 | X9 | X10 | X11 | X12 | X13 | X14 | X15 | X16 | X17 | X18 | X19 | X20 | X21 | X22 | X23 | X24 |
|------|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| MK | 0 | 1.4 | 2.6 | 4.0 | 4.9 | 5.9 | 6.1 | 6.9 | 6.9 | 6.1 | 5.4 | 4.1 | 3.1 | 2.0 | 1.7 | 1.4 | 1.3 |
| MK | 0 | 1.0 | 1.7 | 2.4 | 2.9 | 3.4 | 3.6 | 3.7 | 3.7 | 3.3 | 2.4 | 2.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| MK | 0 | 1.1 | 2.0 | 2.9 | 3.4 | 4.4 | 4.6 | 4.7 | 4.1 | 3.6 | 2.6 | 1.9 | 1.3 | 1.0 | 1.0 | 0.0 | 0.0 |
| MK | 2 | 2.7 | 2.7 | 2.9 | 3.4 | 3.7 | 3.4 | 3.0 | 2.9 | 2.3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| MK | 0 | 1.0 | 2.4 | 3.6 | 3.7 | 4.4 | 5.0 | 5.3 | 5.0 | 4.7 | 3.6 | 3.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 |
| MK | 0 | 1.3 | 2.4 | 3.6 | 4.9 | 5.6 | 6.0 | 6.7 | 6.6 | 6.1 | 5.1 | 4.1 | 2.9 | 2.1 | 1.7 | 1.4 | 1.6 |

Eliminamos la fila del nombre del parque.

```
entry_df <- entry_df[, 2:18]
```

```
entry_df <- reshape2::melt(entry_df)
```

```
entry_df$variable <- as.numeric(sub("X", "", entry_df$variable)) - 8
```



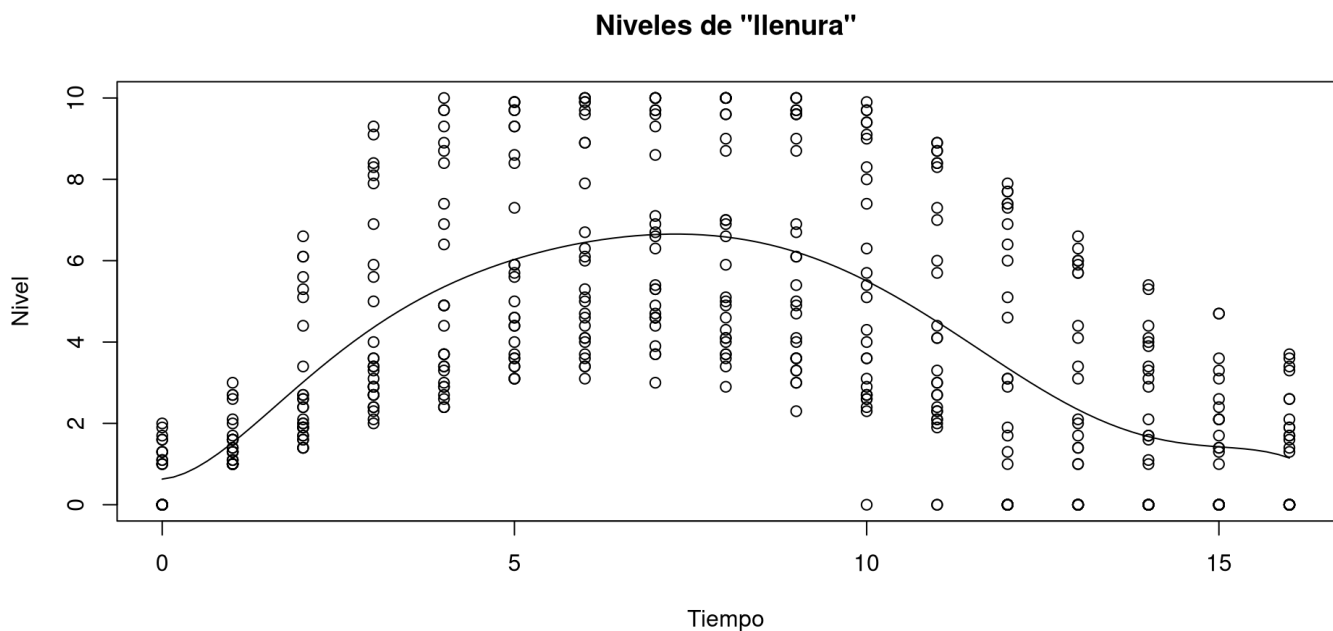
```
lm_model <- lm(formula = value ~ poly(variable, 7, raw = TRUE), data = entry_df)
```

```
##
## Call:
## lm(formula = value ~ poly(variable, 7, raw = TRUE), data = entry_df)
##
## Coefficients:
##              (Intercept)  poly(variable, 7, raw = TRUE)1
##              6.310e-01      1.416e-01
## poly(variable, 7, raw = TRUE)2  poly(variable, 7, raw = TRUE)3
##              1.083e+00      -3.869e-01
## poly(variable, 7, raw = TRUE)4  poly(variable, 7, raw = TRUE)5
##              6.456e-02      -5.801e-03
## poly(variable, 7, raw = TRUE)6  poly(variable, 7, raw = TRUE)7
##              2.638e-04      -4.724e-06
```

Ya a partir de estos coeficientes calculados por la función `lm`, podemos convertir el modelos a su forma de ecuación polinómica el modelo propuesto.

```
## [1] 0 + ( 0.631012633902875 ) * ( x ** 0 ) + ( 0.141641794969108 ) * ( x ** 1 ) + ( 1.0832056275
4808 ) * ( x ** 2 ) + ( -0.386896805430061 ) * ( x ** 3 ) + ( 0.0645561077512354 ) * ( x ** 4 ) + (
-0.00580069833046977 ) * ( x ** 5 ) + ( 0.000263776181717327 ) * ( x ** 6 ) + ( -4.72376212119374e-
06 ) * ( x ** 7 )
```

Y a partir de esto, podremos realizar la visualización del modelos planteado y ver el ajuste de este con respecto a los datos que se tienen.



Será a partir de este modelo por el cual se definirá la entrada de las personas al parque. Esto tiene una gran importancia pues es a partir de esto que se definirá tanto la cantidad de visitantes al parque como el estos van llegando.

Definiendo Los Arquetipos De Los Visitantes

Carguemos nuestro dataset relacionado con las respuestas recolectadas.

```
forms_df <- read.csv("https://gitlab.com/MrDahaniel/proyecto-simulacion/-/raw/main/datasets/formRes
ponses.csv")[2:10]
```

TomarFotos CompRecu InAtrac CamiPar ComAlim TiempoMinimoEstadia TiempoMaximoEstadia TiempoEspera S:

| TomarFotos | CompRecu | InAtrac | CamiPar | ComAlim | TiempoMinimoEstadia | TiempoMaximoEstadia | TiempoEspera | Si |
|------------|----------|---------|---------|---------|---------------------|---------------------|--------------|----|
| 1 | 1 | 3 | 1 | 2 | 1 a 2 horas | 3 a 4 horas | 5 a 10 min | \$ |
| 3 | 1 | 4 | 3 | 1 | 4 a 5 horas | 4 a 5 horas | 2 a 5 min | \$ |
| 3 | 3 | 4 | 2 | 3 | 3 a 4 horas | 4 a 5 horas | 10 a 20 min | \$ |
| 5 | 4 | 5 | 5 | 2 | 3 a 4 horas | 4 a 5 horas | 30 a 40 min | \$ |
| 3 | 4 | 3 | 3 | 4 | 1 a 2 horas | 4 a 5 horas | 10 a 20 min | \$ |
| 3 | 4 | 4 | 3 | 4 | 3 a 4 horas | 4 a 5 horas | 10 a 20 min | \$ |

Ya con estos datos disponibles, debemos determinar 3 componentes principales para poder definir cada uno de los arquetipos de los visitantes al parque. El primero de estos está en la preferencia de realizar una actividad sobre una atracción. En este sentido, tendremos que basarnos sobre las respuestas relacionadas con los gustos de las personas.

Probabilidad De Atracciones Sobre Actividades

```
preference_df <- forms_df[1:5]
```

| TomarFotos | CompRecu | InAtrac | CamiPar | ComAlim |
|------------|----------|---------|---------|---------|
| 1 | 1 | 3 | 1 | 2 |
| 3 | 1 | 4 | 3 | 1 |
| 3 | 3 | 4 | 2 | 3 |
| 5 | 4 | 5 | 5 | 2 |
| 3 | 4 | 3 | 3 | 4 |
| 3 | 4 | 4 | 3 | 4 |

Lo primero que vamos a hacer con el fin de definir los porcentajes de preferencia de las personas estará en promediar el valor de cada una de las actividades. De esta manera, tendremos el valor general de que estos realicen una actividad. Partiendo de esto, sólo queda sacar la fracción entre el valor de la atracción sobre la suma del puntaje promedio de las actividades con el puntaje de la atracción para tener un aproximado de las preferencias de las personas.

```
preference_df$PromAct <- rowMeans(preference_df[, c(1, 2, 4, 5)])
preference_df$ProbAttr <- (preference_df[, 3]) / (preference_df[, 3] + preference_df[, 6])
```

| TomarFotos | CompRecu | InAtrac | CamiPar | ComAlim | PromAct | ProbAttr |
|------------|----------|---------|---------|---------|---------|-----------|
| 1 | 1 | 3 | 1 | 2 | 1.25 | 0.7058824 |
| 3 | 1 | 4 | 3 | 1 | 2.00 | 0.6666667 |
| 3 | 3 | 4 | 2 | 3 | 2.75 | 0.5925926 |
| 5 | 4 | 5 | 5 | 2 | 4.00 | 0.5555556 |
| 3 | 4 | 3 | 3 | 4 | 3.50 | 0.4615385 |
| 3 | 4 | 4 | 3 | 4 | 3.50 | 0.5333333 |

Entonces, a partir de esto, podemos encontrar los cuantiles de nuestros promedios para poder tomar los valores base de las probabilidades de escoger una atracción sobre una actividad, De esta manera, podremos definir más adelante nuestros arquetipos para los visitantes del parque.

```
quantiles <- quantile(preference_df$ProbAttr)
```

```
##           0%          25%          50%          75%          100%  
## 0.2666667 0.5714286 0.5925926 0.6250000 0.7407407
```

A partir de estos 5 cuantiles, podemos definir los valores generales de la probabilidad de escoger una atracción sobre una actividad. Siendo así, y teniendo en cuenta que los cuantiles de 25%, 50% y 75 son relativamente similares, se trabajarán con un total de 3 probabilidades para el desarrollo de la simulación.

Tiempos De Espera Y Estadía

Ya teniendo definidas las diferentes probabilidades de escoger las atracciones, nos queda por definir los diferentes tiempos tanto de estadía el parque. Estos tiempos, nuevamente, serán definidos por las diferentes respuestas recibidas en la encuesta realizada.

Empecemos determinando los tiempos de espera máximos para cada uno de los arquetipos. Lo primero que haremos será determinar los porcentajes que representa cada uno de los intervalos que se han definido

```
wait_df <- forms_df[8]  
reps <- cbind(table(wait_df))  
reps <- unname(cbind(reps, row.names(reps)))  
reps <- data.frame(reps)  
  
colnames(reps) <- c("repeticiones", "tiempo_espera")  
  
t_espera <- c()  
total_respuestas <- nrow(wait_df)  
  
for (i in 1:nrow(reps)) {  
  t_espera <- c(t_espera, as.integer(reps[i, 1]) / total_respuestas)  
}  
  
reps$porcentaje <- cbind(t_espera)
```

| repeticiones | tiempo_espera | porcentaje |
|--------------|---------------|------------|
| 28 | 10 a 20 min | 0.34146341 |
| 6 | 2 a 5 min | 0.07317073 |
| 16 | 20 a 30 min | 0.19512195 |
| 8 | 30 a 40 min | 0.09756098 |
| 1 | 40 a 50 min | 0.01219512 |
| 19 | 5 a 10 min | 0.23170732 |
| 2 | 50 a 60 min | 0.02439024 |
| 2 | 90 min o más | 0.02439024 |

Apliquemos este mismo proceso para determinar los tiempos de estadía dentro del parque.

```
wait_df <- forms_df[6]
reps <- cbind(table(wait_df))
reps <- unname(cbind(reps, row.names(reps)))
reps <- data.frame(reps)

colnames(reps) <- c("repeticiones", "tiempo_estadia_min")

t_espera <- c()
total_respuestas <- nrow(wait_df)

for (i in 1:nrow(reps)) {
  t_espera <- c(t_espera, as.integer(reps[i, 1]) / total_respuestas)
}

reps$porcentaje <- cbind(t_espera)
```

| repeticiones | tiempo_estadia_min | porcentaje |
|--------------|--------------------|------------|
| 32 | 1 a 2 horas | 0.39024390 |
| 6 | 1 hora o menos | 0.07317073 |
| 10 | 2 a 3 horas | 0.12195122 |
| 22 | 3 a 4 horas | 0.26829268 |
| 8 | 4 a 5 horas | 0.09756098 |
| 3 | 5 a 6 horas | 0.03658537 |
| 1 | 8 horas o más | 0.01219512 |

```
wait_df <- forms_df[7]
reps <- cbind(table(wait_df))
reps <- unname(cbind(reps, row.names(reps)))
reps <- data.frame(reps)

colnames(reps) <- c("repeticiones", "tiempo_estadia_max")

t_espera <- c()
total_respuestas <- nrow(wait_df)

for (i in 1:nrow(reps)) {
  t_espera <- c(t_espera, as.integer(reps[i, 1]) / total_respuestas)
}

reps$porcentaje <- cbind(t_espera)
```

| repeticiones | tiempo_estadia_max | porcentaje |
|--------------|--------------------|------------|
| 2 | 1 a 2 horas | 0.02439024 |
| 1 | 1 hora o menos | 0.01219512 |
| 13 | 2 a 3 horas | 0.15853659 |
| 14 | 3 a 4 horas | 0.17073171 |
| 27 | 4 a 5 horas | 0.32926829 |
| 5 | 5 a 6 horas | 0.06097561 |

| repeticiones | tiempo_estadia_max | porcentaje |
|--------------|--------------------|------------|
| 6 | 6 a 7 horas | 0.07317073 |
| 7 | 7 a 8 horas | 0.08536585 |
| 7 | 8 horas o más | 0.08536585 |

A apartir de estos datos, se llegó a la determinación de lo siguientes arquetipos con los cuales se trabajará durante la simulación. Estos se encuentran en `constants.py` que es el que se encarga del manejo de esos datos.

```

ARCHETYPES = {
    "Tourist": {
        "AttractionPercentage": 0.6,
        "MinStayTime": 4 * 60,
        "MaxStayTime": 8 * 60,
        "WaitTime": 60,
    },
    "Average": {
        "AttractionPercentage": 0.5,
        "MinStayTime": 4 * 60,
        "MaxStayTime": 8 * 60,
        "WaitTime": 50,
    },
    "Activity Enjoyer": {
        "AttractionPercentage": 0.2666667,
        "MinStayTime": 1 * 60,
        "MaxStayTime": 5 * 60,
        "WaitTime": 10,
    },
    "Attraction Enjoyer": {
        "AttractionPercentage": 0.7407407,
        "MinStayTime": 4 * 60,
        "MaxStayTime": 9 * 60,
        "WaitTime": 70,
    },
    "All Day Guest": {
        "AttractionPercentage": 0.6,
        "MinStayTime": 8 * 60,
        "MaxStayTime": 12 * 60,
        "WaitTime": 40,
    },
    "Relaxed Visitor": {
        "AttractionPercentage": 0.4,
        "MinStayTime": 3 * 60,
        "MaxStayTime": 6 * 60,
        "WaitTime": 25,
    },
}

```

Compras Del *FastPass* de Salitre

Ahora, lo último que nos queda en término del tratamiento de datos respecto a los datos recolectados en la encuesta, está en determinar el porcentaje de personas que pagarían el costo extra del *FastPass* para el escenario de Salitre. En este sentido, se definirá el porcentaje de personas que estarían dispuestas a pagar un precio en especial.

Tomemos la cantidad de personas por cada uno de los precios que se definieron en la encuesta:

```
# Traemos la columna relacionada
fastpass_df <- forms_df[9]

no_compra <- length(fastpass_df[fastpass_df == "Ninguna de las anteriores"])
p1030 <- length(fastpass_df[fastpass_df == "$10.000-$30.000"])
p3050 <- length(fastpass_df[fastpass_df == "$30.000-$50.000"])
p50p <- length(fastpass_df[fastpass_df == "Mas de $50.000"])
```

| categoria | frecuencia | porcentaje | porcentaje_compra |
|---------------------------|------------|--------------------|--------------------|
| Ninguna de las anteriores | 17 | 0.217948717948718 | 1 |
| Entre \$10.000 y \$30.000 | 51 | 0.653846153846154 | 0.782051282051282 |
| Entre \$30.000 y \$50.000 | 8 | 0.102564102564103 | 0.128205128205128 |
| Mas de \$50.000 | 2 | 0.0256410256410256 | 0.0256410256410256 |

La Popularidad, Duración Y La Tasa De Servicio De Las Atracciones

Ya saliendo de las encuestas, pasamos a nuestro valores definidos ya como tal. Partamos por las tasas de servicio de las atracciones con las que estaremos trabajando. De manera oficial, no existen como tal registros de las tasas de servicio para ninguna de las atracciones dentro de los parques de Disney, por este motivo, se tuvo que recurrir a estimaciones realizadas por los visitantes del parque. Una de estas fuentes, wdwthemeparks.com (<https://wdwthemeparks.com/hourly-capacity-numbers/>), nos presenta con una gran mayoría de los tiempos de espera para los algunos de los parque de atracciones de Disney.

De manera similar, tenemos que definir el tiempo de duración de las atracciones. Estos tiempos definen tanto el periodo de inactividad de los visitantes como el de las atracciones. De igual manera, al no poder tomar datos directamente de los parques a trabajar, se emplearon reportes de diferentes visitantes al parque publicados en internet. En este caso, se están usando los valores reportados por touringplans.com (<https://touringplans.com/animal-kingdom/attractions/duration#>).

Para el desarrollo de este ejercicio, se tomará como referencia el parque de Animal Kingdom, esto principalmente a que es el parque con los datos más completos dentro todos los que se tienen disponibles. Siendo así, en la simulación se trabajarán con las siguientes atracciones:

| attractions | service_rates | duration |
|---------------------|---------------|----------|
| DINOSAUR | 1800 | 4 |
| Expedition Everest | 1800-2060 | 4 |
| Flight of Passage | 1280-1440 | 6 |
| Kali River Rapids | 1800-2000 | 10 |
| Kilimanjaro Safaris | 3000 | 20 |
| Na'vi River Journey | 1080-1440 | 5 |
| TriceraTop Spin | 800-900 | 2 |

Ahora, en cuanto a la popularidad de cada una de las atracciones, se calculó un valor aproximado de esta definido a partir de la multiplicación de 2 factores. El primero de estos, es la misma tasa de servicio definida en el punto anterior y el segundo de estos es el tiempo de espera mensual promedio reportado por algunos de los servicios como lo son thrill-data.com (<https://www.thrill-data.com/waits/park/wdw/animal-kingdom/>). A partir de esto tendremos un estimado del verdadero interés de una atracción en especial al tener en cuenta tanto que tanto hay que esperar en promedio para la atracción, lo que refleja el interés de los visitantes, y la tasa de servicio lo cual evita tiene en cuenta el como se relacionan estos tiempos de espera con la velocidad con la que se atienden a las personas.

Siendo así, para cada uno de las atracciones, estos serían los datos con los que estaríamos trabajando.

| attractions | service_rates | duration | reported_wait_times | popularity |
|---------------------|---------------|----------|---------------------|------------|
| DINOSAUR | 1800 | 4 | 33 | 59400 |
| Expedition Everest | 1800-2060 | 4 | 26 | 46800 |
| Flight of Passage | 1280-1440 | 6 | 98 | 125440 |
| Kali River Rapids | 1800-2000 | 10 | 29 | 52200 |
| Kilimanjaro Safaris | 3000 | 20 | 48 | 144000 |
| Na'vi River Journey | 1080-1440 | 5 | 70 | 75600 |
| TriceraTop Spin | 800-900 | 2 | 15 | 12000 |

En el momento de realizar la elección de que atracción será para la cual hará fila, se calcula el porcentaje de popularidad que representa el términos generales cada una de las atracciones.

Implementación De La Simulación

Para el desarrollo del prototipo para la simulación, se partió de lo que se definió inicialmente durante la fase de la conceptualización del modelo. Siendo así, tenemos los siguientes módulos:

El primero de estos es el módulo de `Park`, de manera general, este es nuestro componente principal pues engloba en si todos los demás componentes de la simulación. Hay que resaltar que, además de manejar la creación y manejo de los arquetipos, atracciones y actividades; este se encarga de como tal realizar la simulación del parque con el uso de la función `Park.start_day(self, max_entry_rate, wait_time_update)`.

```
class Park:
    def __init__(
        self,
        attraction_dict: dict[str, dict] = ATTRACTIONS,
        activities_dict: dict[str, dict] = ACTIVITIES,
        archetype_dict: dict[str, dict] = ARCHETYPES,
        fn: Callable[[float], float] = lambda x, k: k ** x * np.exp(-k) / gamma(x + 1),
        hours_open: int = 16,
        fastpass_pool_size: float = 0.3,
    ) -> None:
        self.current_time: int = 0
        self.closing_time: int = 60 * hours_open
        self.attractions: list[Attraction] = self._handle_attractions(
            attraction_dict=attraction_dict,
            alt_queue=None,
            hours_open=hours_open,
            fastpass_pool_size=fastpass_pool_size,
        )
        self.activities: list[Activity] = self._handle_activities(activities_dict)
        self.guest_archetypes: list[Archetype] = self._handle_archetypes(archetype_dict)
        self.guests: list[Union[Person, DisneyPerson]] = []
        self.function: Callable[[float], float] = fn
```

Así mismo, es a partir de la herencia de `Park` por el cual se realizan las diferentes implementaciones de nuestros diferentes casos alternos. En estos casos, los cambios más relevantes están principalmente en la función `start_day` al igual que algunas de las funciones auxiliares como `Park._update_fake_wait_time(self)` la cual varía en cada una de las implementaciones.

De manera un poco más interna, se implementaron los componenetes de `Activity` y `Attraction`. Estos, en especial `Attraction`, son los encargados del manejo de sus filas, al igual que el atender a las personas que las visitan.

```

class Activity:
    def __init__(self, name: str, popularity: int, duration: int) -> None:
        self.name: str = name
        self.duration: int = duration
        self.popularity: int = popularity

class Attraction(Activity):
    def __init__(
        self,
        name: str,
        popularity: int,
        duration: int,
        service_rate: int,
        alt_queue: str = None,
        hours_open: int = 16,
        fastpass_pool_size: float = 0.3,
        fastpass_serve_size: float = 0.6,
    ) -> None:
        super().__init__(name, popularity, duration)
        self.service_rate: int = service_rate
        self.fake_wait_time: int = 0
        self.queue: Queue = Queue()
        self.fastpass_serve_size: float = fastpass_serve_size

        self.time_until_service: int = duration

        self.total_people_served: int = 0

        if alt_queue in ["DFP", "SFP"]:
            self.alt_queue: Queue = Queue(type=alt_queue)
            self.queue: TotalQueue = TotalQueue()

            if alt_queue == "DFP":
                self.fast_pass_machine: FastPassMachine = FastPassMachine(
                    attraction_name=name,
                    service_rate=service_rate,
                    hours_open=hours_open,
                    fastpass_pool_size=fastpass_pool_size,
                )

```

La creación de instancias de estos módulos durante la simulación viene dada desde el parque. Las funciones más importantes dentro de `Attraction` es la función `Attraction.serve(self)` y `Attraction.serve_with_fastpass(self)` debido a que estas son las encargadas de llamar a las filas para que atiendan a las personas presentes en las filas.

En cuanto a las visitantes del parque, se implementó el módulo `Person`, el cual se encarga del manejo tanto de las características de las personas al igual que el comportamiento de las mismas en el momento de decir que harán en cuanto a atracciones y actividad y el cuanto están dispuestas a esperar en una fila.

```

class Person:
    def __init__(
        self,
        id: int,
        arrival_time: int,
        archetype: Archetype,
        park_closing_time: int = None,
    ) -> None:
        # General info
        self.id: str = id
        self.time_left_in_activity: int = 0
        self.current_activity: str = None

        # Used for statistics
        self.attractions_experienced: int = 0
        self.total_wait_time: int = 0
        self.arrival_time: int = arrival_time
        self.things_done: list[str] = []
        self.queues_joined: int = 0

        # Archetype defined
        self.archetype: str = archetype.name
        self.departure_time: int = (
            randint(archetype.min_stay, archetype.max_stay) + arrival_time
        )
        self.max_wait: int = randint(archetype.min_wait_time, archetype.max_wait_time)
        self.attraction_chance: float = archetype.attraction_chance

        # Handle time after park closes
        if park_closing_time is not None and park_closing_time < self.departure_time:
            self.departure_time = park_closing_time

```

Para cada una de las implementaciones alternas que se realizaron, se creó a partir de herencia un tipo de persona específico para cada uno de los diferentes escenarios. Esto se debe principalmente a que el comportamiento de las personas dentro del parque cambiaba dependiendo del escenario que se estaba trabajando. Siendo así, la función principal de `Person` sería `Person.check_attraction` pues es la que define el comportamiento de las personas dentro del parque en altera la manera de decidir sobre si espera o no en una fila.

Diseño Experimental

Durante el desarrollo de la simulación se identificaron una serie de tres alternativas, para ser comparadas con el caso base. Los cuatro escenarios a simular cuentan con estas características en común: la persona que ingrese al parque podrá escoger entre actividad (acción para la cual no necesita hacer ninguna fila), o atracción (acción para la que tendrá un tiempo de espera y por lo tanto tendrá que hacer fila). Y si la persona se decide por una atracción, será su arquetipo, es decir una serie de datos que determinarán su decisión, el que indicará si la persona escoge hacer fila por esa atracción o si mejor toma otra decisión (actividad o atracción).

Sintetizando, son cuatro escenarios en total los que se pondrán a prueba en la simulación (contando el caso base). Se describen a continuación:

Caso Base: En este escenario la persona ingresará al parque y tendrá la opción de escoger entre una actividad que podrá hacer inmediatamente, o una atracción para la que deberá esperar haciendo fila. Este sería el caso con el que cuenta un parque de atracciones sin implementar algún tipo de ticket especial o preferencial, que tendrá un mayor costo para el visitante y que en nuestro artículo se ha decidido nombrar como *'fast pass'*.

Caso *FastPass* Disney: En este escenario la persona ingresará al parque y tendrá la opción de realizar una actividad que al igual que en el caso base y en los demás, podrá hacer al instante, o podrá escoger una atracción. En caso de escoger la atracción pueden ocurrir dos situaciones, que la persona posea *FastPass* para cierta hora, es decir, a determinada hora del día pueda

pasar sin hacer fila, o podrá no tener *FastPass* y según si quiere esperar haciendo fila o no (dependiendo de su arquetipo), podrá subirse a la atracción que escogió. Para el caso *FastPass* se decidió que de las personas que llegaran al parque solo el 30% pudieran adquirir un *FastPass*.

Caso *FastPass* Salitre: En este escenario la persona ingresará al parque y tendrá que escoger entre actividades y atracciones. La diferencia radica en que el *FastPass* de Salitre es un ticket cuyo precio es mayor al normal, y le permite a la persona ingresar a una fila virtual para omitir la normal, y así poder disminuir su tiempo de espera en cola.

Caso Tiempos Falsos: En este escenario a simular las personas siguen el mismo comportamiento que tendrían en el caso base. La diferencia se encuentra en que los tiempos de espera mostrados a la persona que escoge cierta atracción, estarán aumentados según el rango de tiempo en el que se halle el tiempo de espera de tal atracción, con el fin de disuadir a las personas de hacer la fila y así reducir los tiempos de colas para las atracciones.

Resultados de la Simulación

Para tratar de tener varios días de cada parque, se simularon un total de 7 días para cada uno de los escenarios los cuales trabajaron. Esto se realizó con el fin de tratar de tener varios días con los que trabajar en cuanto al análisis de resultados. En total, esto nos dio un total de alrededor de 800000 registros en total, aproximadamente 200000 para cada una de los casos trabajados. El reportar estos datos en una tabla para poder sacar conclusiones, aunque posible, es realmente complicado pues no es la mejor manera de condensar los datos.

Siendo así, empecemos revisando los tiempos de espera promedio de cada una de los escenarios.

Tiempos De Espera

```
base_df <- read.csv("https://gitlab.com/MrDahaniel/proyecto-simulacion/-/raw/main/datasets/BASE.csv")
dfp_df <- read.csv("https://gitlab.com/MrDahaniel/proyecto-simulacion/-/raw/main/datasets/DFP.csv")
sfp_df <- read.csv("https://gitlab.com/MrDahaniel/proyecto-simulacion/-/raw/main/datasets/SFP.csv")
ftp_df <- read.csv("https://gitlab.com/MrDahaniel/proyecto-simulacion/-/raw/main/datasets/FTP.csv")
```

```
## [1] "Caso Base"
```

| X | Archetype | arrival_time | departure_time | time_in_park | max_wait | attractions_experienced | total_wait_time | average_ |
|---|---------------------|--------------|----------------|--------------|----------|-------------------------|-----------------|----------|
| 0 | Activity Enjoyer | 1 | 186 | 185 | 10 | 6 | 21 | |
| 1 | Relaxed Visitor | 1 | 315 | 314 | 25 | 6 | 49 | |
| 2 | Tourist | 1 | 504 | 503 | 60 | 7 | 99 | |
| 3 | Relaxed Visitor | 1 | 345 | 344 | 25 | 4 | 18 | |
| 4 | Tourist | 1 | 315 | 314 | 60 | 11 | 73 | |
| 5 | Relaxed Visitor | 1 | 365 | 364 | 25 | 7 | 44 | |

```
## [1] "Disney FastPass"
```

| X | Archetype | arrival_time | departure_time | time_in_park | max_wait | attractions_experienced | total_wait_time | average_ |
|---|-----------------|--------------|----------------|--------------|----------|-------------------------|-----------------|----------|
| 0 | Tourist | 1 | 412 | 411 | 60 | 10 | 166 | |
| 1 | All Day Park | 1 | 635 | 634 | 40 | 13 | 185 | |

| X | Archetype | arrival_time | departure_time | time_in_park | max_wait | attractions_experienced | total_wait_time | average_ |
|---|-----------------------|--------------|----------------|--------------|----------|-------------------------|-----------------|----------|
| 2 | Attraction Enjoyer | 1 | 415 | 414 | 70 | 10 | 199 | |
| 3 | Relaxed Visitor | 1 | 296 | 295 | 25 | 4 | 8 | |
| 4 | All Day Park | 1 | 676 | 675 | 40 | 12 | 229 | |
| 5 | Tourist | 1 | 371 | 370 | 60 | 6 | 72 | |

```
## [1] "Salitre FastPass"
```

| X | Archetype | arrival_time | departure_time | time_in_park | max_wait | attractions_experienced | total_wait_time | average_ |
|---|-----------------------|--------------|----------------|--------------|----------|-------------------------|-----------------|----------|
| 0 | All Day Park | 1 | 660 | 659 | 40 | 15 | 163 | |
| 1 | Relaxed Visitor | 1 | 258 | 257 | 25 | 3 | 25 | |
| 2 | Tourist | 1 | 315 | 314 | 60 | 8 | 83 | |
| 3 | Attraction Enjoyer | 1 | 343 | 342 | 70 | 5 | 114 | |
| 4 | Activity Enjoyer | 1 | 191 | 190 | 10 | 5 | 19 | |
| 5 | Tourist | 1 | 378 | 377 | 60 | 6 | 66 | |

```
## [1] "Tiempos Falsos"
```

| X | Archetype | arrival_time | departure_time | time_in_park | max_wait | attractions_experienced | total_wait_time | average_ |
|---|---------------------|--------------|----------------|--------------|----------|-------------------------|-----------------|----------|
| 0 | Activity Enjoyer | 1 | 143 | 142 | 10 | 2 | 9 | |
| 1 | Relaxed Visitor | 1 | 182 | 181 | 25 | 2 | 19 | |
| 2 | Tourist | 1 | 357 | 356 | 60 | 11 | 81 | |
| 3 | Activity Enjoyer | 1 | 71 | 70 | 10 | 0 | 0 | |
| 4 | All Day Park | 1 | 624 | 623 | 40 | 9 | 135 | |
| 5 | Average | 1 | 445 | 444 | 50 | 10 | 135 | |

Ya con los datos cargados, miremos los tiempos de espera promedio para aquellas personas que pudieron subirse aunque sea a una atracción. En el caso de no tener esta consideración, estaríamos añadiendo datos que no deberíamos considerar a los resultados.

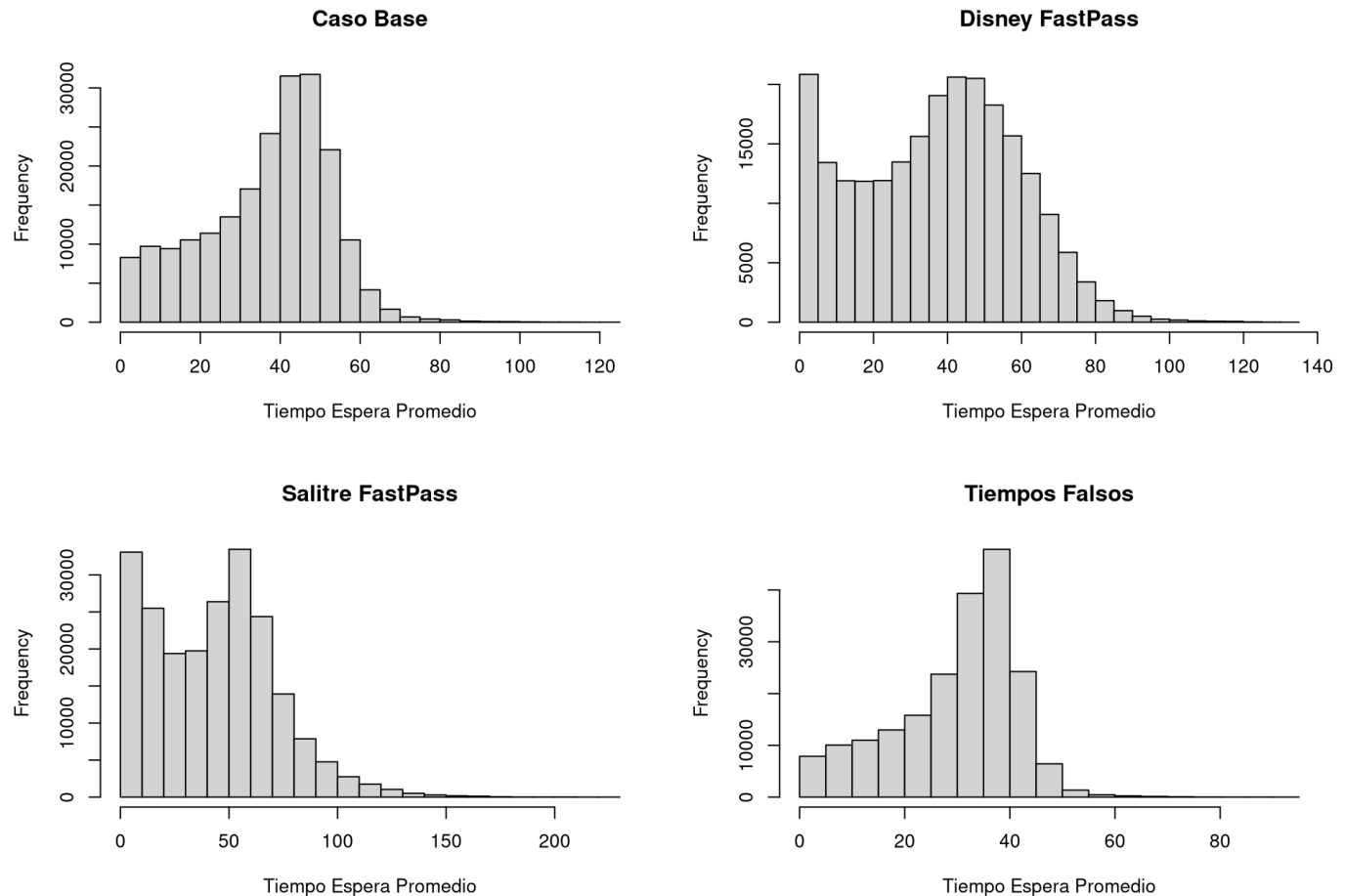
```
## [1] "Promedio tiempo de espera (Caso Base) 36.8056699419039"
```

```
## [1] "Promedio tiempo de espera (Disney FastPass) 37.4893496992982"
```

```
## [1] "Promedio tiempo de espera (Salite FastPass) 43.3208266770642"
```

```
## [1] "Promedio tiempo de espera (Tiempos Falsos) 29.9348825861302"
```

Veamos de manera gráfica la distribución de estos tiempos de espera.



De aquí podemos ver el como se comporta de manera general cada una de las implementaciones en cuanto el como altera las distribuciones de estos tiempos de espera.

Subidas A Atracciones

Repitamos el mismo proceso anterior pero para la cantidad promedio de veces que los visitantes del parque pudieron subirse a las atracciones.

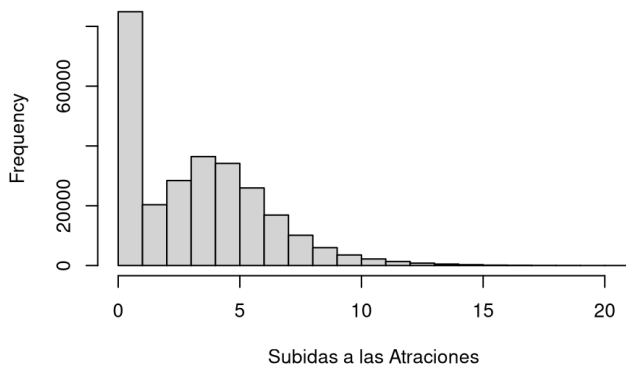
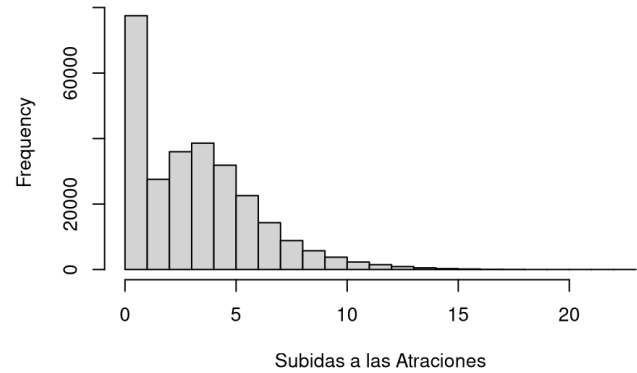
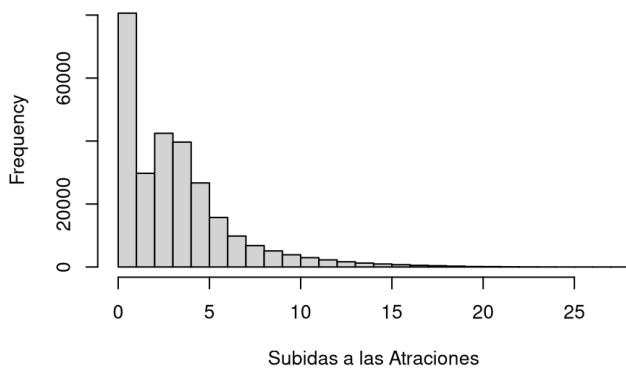
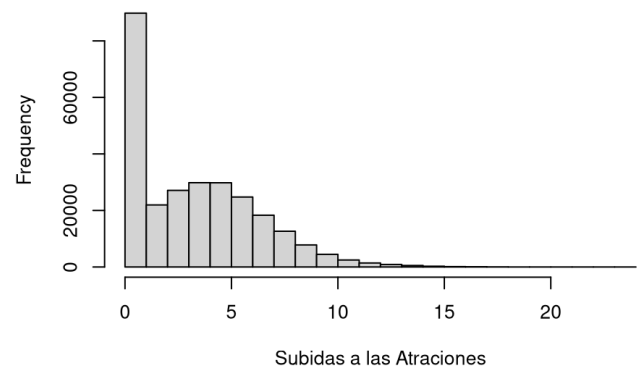
```
## [1] "Promedio atracciones subidas (Caso Base) 3.58002431043917"
```

```
## [1] "Promedio atracciones subidas (Disney FastPass) 3.58467256663144"
```

```
## [1] "Promedio atracciones subidas (Salite FastPass) 3.53372122513039"
```

```
## [1] "Promedio atracciones subidas (Tiempos Falsos) 3.59149306256353"
```

De manera gráfica, podemos apreciar la distribución de estos datos.

Caso Base**Disney FastPass****Salitre FastPass****Tiempos Falsos**

Veamos revisemos la cantidad de personas para cada uno de los escenarios que no pudieron subirse a ninguna atracción:

```
## [1] "Personas sin subirse (Caso Base) 9253.28571428571"
```

```
## [1] "Personas sin subirse (Disney FastPass) 6356.71428571429"
```

```
## [1] "Personas sin subirse (Salitre FastPass) 8115.28571428571"
```

```
## [1] "Personas sin subirse (Tiempos Falsos) 10120.1428571429"
```

Y la cantidad de personas que pudieron subirse a al menos una atracción:

```
## [1] "Personas que pudieron subirse (Caso Base) 29648.2857142857"
```

```
## [1] "Personas que pudieron (Disney FastPass) 32578.1428571429"
```

```
## [1] "Personas que pudieron (Salitre FastPass) 30751.7142857143"
```

```
## [1] "Personas que pudieron (Tiempos Falsos) 28809.4285714286"
```

Análisis De Los Resultados

Procesando los diferentes resultados arrojados por las simulaciones realizadas donde se emplearon diferentes técnicas para obtener lectura de reducción o mejora de los tiempos de espera, se observaron datos bastante cercanos entre cada una de las

lecturas.

Se realiza una comparación frente a cada caso evaluado teniendo en cuenta el estudio realizado a un sistema de parque de diversiones que no presente ninguna de las alternativas nombradas en secciones anteriores, ya que en ello se basará la observación de una mejora significativa al momento de emplear los distintos modelos. Se tendrá en consideración que cada simulación se realizará con un máximo de 95.000 / 960 personas es decir más o menos 100 personas por cada 5 minutos transcurridos dentro de la simulación, esto es decir entre 37.000 y 39.000 personas en el parque por un día.

Se iniciará con la comparación de la simulación del *FastPass* de Disney con el presente caso base, donde se observa un notorio cambio presente en las personas que no pudieron disfrutar de una atracción del parque, visualizando que entre el *FastPass* de Disney y el caso base de un parque se tiene la diferencia de 3007 personas que pudieron disfrutar de al menos una atracción en los parques de Disney. Teniendo esto en consideración se evidencia un claro aumento en el promedio de atracciones que fueron experimentadas por los distintos usuarios que visitan al parque. Esto puede visualizarse en las gráficas presentes en la sección de resultados de la simulación dedonde se puede observar con mayor detalle.

Los tiempos de espera entre cada una de las filas de las atracciones también se vieron afectados, ya que con la implementación del *FastPass* las personas podían disfrutar de la atracción el tiempo correspondido, haciendo que el número total de personas que no presentaron demoras en el ingreso a las distintas atracciones aumentará en comparación al modelo del parque tradicional.

Cabe resaltar que existe un ligero aumento entre los tiempos de espera de cada atracción ya que las personas que presentan la facilidad de *FastPass* únicamente pueden saltarse la fila de la atracción deseada en un tiempo específico haciendo que esta pueda elegir entre hacer una fila distinta para otra atracción mientras llega su turno o elegir una de las diferentes actividades a realizar en el parque.

Realizando la comparativa con la alternativa presentada por el SalitrePass contra los resultados obtenidos en el estudio del parque tradicional obtenemos una lectura con cambios perceptibles, teniendo en consideración el número de personas que no han logrado disfrutar de una atracción dentro del parque presenta una diferencia de 1259 personas, aumentando el número de personas que al menos disfrutaron de una atracción. Los tiempos de espera promedio también se vieron afectados ya que al tener el SalitrePass la persona puede saltarse la fila convencional e ingresar a una fila especial presentada por la misma alternativa, este presente caso puede presentar algunas averías dependiendo la cantidad de personas que presenten el SalitrePass ya que las filas pueden llegar a ser mucho más extensas que las convencionales.

Teniendo esto presente se observa que los tiempos de espera promedio han aumentado un aproximadamente un 15% pero se debe resaltar que las personas que no han presentado tiempo de espera son mucho mayores en el caso que presenta la alternativa del SalitrePass en comparación al parque tradicional. El tiempo de espera máximo por atracción se observa con ligeros cambios donde los tiempos de cada atracción han aumentado en el caso presente del SalitrePass.

Tomando en consideración los datos obtenidos para el análisis del caso de tiempos falsos contra los resultados presentes en la simulación de un parque tradicional se presenta que la cantidad de personas que no han ingresado a una atracción es mucho mayor que la presente en el caso base, partiendo de esto se encuentra una diferencia de 557 personas que no pudieron ingresar a una atracción más que en el caso tradicional.

Cabe resaltar que los tiempos de espera presentes en el caso de tiempos falsos es mucho menos que el presente en el caso base, esto se debe a que, al presentar un tiempo mayor al tiempo real establecido para el ingreso a la atracción, las personas evitan emplear el tiempo presente en esperar y desean realizar cualquier otra actividad, esto hace que los tiempos de espera entre cada una de las atracciones sea menor y por lo tanto la media de ellas será inferior.

Estableciendo una comparativa entre las diferentes alternativa presentes en el estudio de parques de atracciones se pueden obtener lecturas presente de una notoria diferencia entre ellas; La alternativa presentada por el SalitrePass un tiempo de espera más bajo que los presentados por el *FastPass* de Disney, esto debido a la capacidad que este presenta a la hora de ingresar a una atracción, pero cabe resaltar que al tener un tiempo de espera bajo la diferencia de promedio presente entre ambas es de 6.05 minutos aproximadamente, donde claramente el *FastPass* de Disney presenta las mejores lecturas.

Hablando sobre los resultados presente en el estudio de tiempos falsos se presenta que tiene lecturas que compiten con los tiempos presentes en el SalitrePass y el *FastPass* de Disney presentando una diferencia de 13.248 y 7.202 minutos respectivamente, pero se debe resaltar que la alternativa de tiempos falsos juega un papel sobre las personas que no presentan mayor tiempo de espera al presente en la atracción, por ello si la persona no realiza la cola los tiempos disminuyen pero un mayor número de personas no querrá ingresar a una atracción en dicho parque.

Conclusiones

- Observando los diferentes resultados obtenidos con cada uno de los casos estudiados en las simulaciones presentadas se puede concluir que las diferentes alternativas pueden ayudar a la reducción de los tiempos de espera presentes en cada parque de diversiones hasta el punto de poder combinar varias alternativas para así mejorar los tiempos de estas.
- La alternativa que con lleva a la mayor reducción del tiempo de espera presente en cada atracción es el caso de tiempos falsos, aunque presenta un inconveniente ya que dependiendo el nivel de espera que tiene la persona se puede obtener resultados variados en cada estudio.
- La mayor de las limitaciones presente con la alternativa de SalitrePass y tiempos falsos es que al presentar muchas personas que adquieran el SalitrePass o tengan la paciencia de esperar a los tiempos falsos se vuelven ineficientes al momento de emplearlas y pasarían a ser una cola similar a la tradicional.

Referencias

- Predecir los tiempos de espera de Disneyland a través de simulaciones de población. (s.f.). Ichi Pro. Disponible aquí (<https://ichi.pro/es/predecir-los-tiempos-de-espera-de-disneyland-a-traves-de-simulaciones-de-poblacion-36233626878710>).
- Datos sobre tiempos de espera en parque de diversiones. (s.f.). Minitab. Disponible aquí (<https://support.minitab.com/es-mx/datasets/quality-tools-data-sets/amusement-park-wait-times/>).
- Disney World Wait Times Available for Data Science and Machine Learning. (s.f.). touringplans. Disponible aquí (<https://touringplans.com/blog/disney-world-wait-times-available-for-data-science-and-machine-learning/>).
- Análisis De Líneas De Espera A Través De Teoría De Colas Y Simulación. (2010). Portilla, L. et al.
- Managing Capacity And Flow At Theme Parks. (1996). Reza H. Ahmadi.
- Duración de las atracciones en Animal Kingdom. (s.f.). Disponible aquí (<https://touringplans.com/animal-kingdom/attractions/duration#>).
- Tasa de Servicio de Animal Kingdom. Disponible aquí (<https://wdwthemeparks.com/hourly-capacity-numbers/>).
- Tiempos de Espera en Animal Kingdom. Disponible aquí (<https://www.thrill-data.com/waits/park/wdw/animal-kingdom/>).