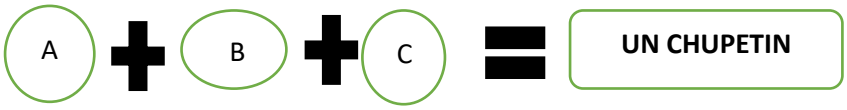
TRABAJO ENCARGADO

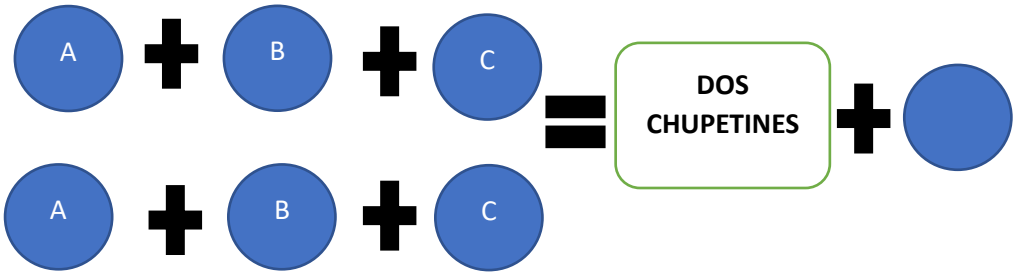
Aplicación de Métodos Iterativos [Ejercicio Supervivencia]

Descripción:

En el siguiente trabajo se aplicará métodos iterativos del juego que se realizó en la sesión de clases que es “supervivencia”, donde se dividió dos grupos de 9 integrantes a cada grupo, de las cuales a cada integrante del grupo se les dio dos tipos de dulces.

El juego consistía en que debían de arman grupos de tres dulces de tipo a+ b + c donde te daban un chupetín a cambio, también podías armar dos grupos de dulces y te daban dos chupetines más un dulce de cualquier tipo ya se a, b o c ,y cada grupo para ganar ,cada uno de los integrantes tenían que contar con un chupetín para ser salvados ,donde tenían que tener o realizar estrategias para ganar y llegar al objetivo, las restringiones fueron las siguientes:





TRABAJO ENCARGADO:

Realizar un código para saber cuantas iteraciones se tiene para llegar al objetivo en el lenguaje R :

CODIGO:

```
set.seed(123)

# Función de simulación
simular_paletas <- function() {
  # Cada estudiante tiene conteo de A, B, C y Paletas (P)
  estudiantes <- vector("list", 9)
  for (i in 1:9) estudiantes[[i]] <- c(A=0, B=0, C=0, P=0)

  iteraciones <- 0

  while (TRUE) {
    iteraciones <- iteraciones + 1

    # Cada estudiante recibe 2 caramelos aleatorios
    for (i in 1:9) {
      nuevos <- sample(c("A", "B", "C"), 2, replace = TRUE)
      for (candy in nuevos) estudiantes[[i]][candy] <- estudiantes[[i]][candy] + 1
    }

    # Evaluar canjes por paletas y posibles devoluciones
    for (i in 1:9) {
      # Número de tríos A,B,C disponibles
      sets <- min(estudiantes[[i]][c("A", "B", "C")])

      if (sets >= 2) {
        # Dos sets: 1 paleta + 1 caramelo aleatorio
        estudiantes[[i]]["P"] <- estudiantes[[i]]["P"] + 1
        estudiantes[[i]][c("A","B","C")] <- estudiantes[[i]][c("A","B","C")] - 2
        bonus <- sample(c("A", "B", "C"), 1)
```

```
      estudiantes[[i]][bonus] <- estudiantes[[i]][bonus] + 1
    } else if (sets == 1) {
      # Un set: 1 paleta
      estudiantes[[i]]["P"] <- estudiantes[[i]]["P"] + 1
      estudiantes[[i]][c("A","B","C")] <- estudiantes[[i]][c("A","B","C")] - 1
    }

    # Si devuelve una paleta, recibe 3 caramelos aleatorios
    if (estudiantes[[i]]["P"] >= 1 && runif(1) < 0.3) {  # 30% de probabilidad
      estudiantes[[i]]["P"] <- estudiantes[[i]]["P"] - 1
      devolucion <- sample(c("A","B","C"), 3, replace = TRUE)
      for (candy in devolucion) estudiantes[[i]][candy] <- estudiantes[[i]][candy] + 1
    }
  }

  # Verificar si los 9 tienen al menos una paleta
  paletas_totales <- sapply(estudiantes, function(e) e["P"])
  if (all(paletas_totales >= 1)) break
 }

 return(iteraciones)
}

# Ejecutar simulaciones
n_rep <- 300
resultados <- replicate(n_rep, simular_paletas())

# Resultados
cat("Promedio de iteraciones necesarias:", mean(resultados), "\n")

# Visualización
```

```
hist(resultados,
    main="Distribución de iteraciones necesarias",
    xlab="Iteraciones",
    col="lightgreen", border="white")
```