



UNIVERSIDAD NACIONAL DEL ALTIPLANO
ESCUELA DE POSGRADO
DOCTORADO EN CIENCIAS DE LA COMPUTACIÓN



TESIS

**TECLADO VIRTUAL RAY PARA DISPOSITIVOS MÓVILES
OPTIMIZADO CON EL MODELO DE DISTRIBUCIÓN DE TECLAS CON
ALGORITMO DE INTELIGENCIA ARTIFICIAL**

PRESENTADA POR:

HUGO DAVID CALDERON VILCA

PARA OPTAR EL GRADO ACADÉMICO DE:

DOCTORIS SCIENTIAE EN CIENCIAS DE LA COMPUTACIÓN

PUNO, PERÚ

2021



DEDICATORIA

Dedico este trabajo de investigación a la Universidad Nacional Mayor de San Marcos, que tomé con mucho cariño ser parte de esta gran familia, como miembro titular del grupo de investigación “Inteligencia Artificial” me ha permitido aportar en varios proyectos de investigación entre ellos en multidisciplinarios, así mismo es una gran oportunidad de asesorar trabajos de tesis de postgrado y pregrado, valoro a los tesistas quienes realizan con mucho esmero su trabajo de investigación.

Dedico este trabajo a la Universidad Peruana de Ciencias Aplicadas, que me permite pertenecer a esta gran academia, hiper vinculada internacionalmente, nos da a los docentes una satisfacción diferente de compartir conocimiento con los estudiantes, donde valora a los investigadores docentes y alumnos.



AGRADECIMIENTOS

- Agradezco a mi país por motivar a los investigadores, los últimos años el Perú ha avanzado fuertemente en investigación, Concytec ha liderado en administrar y gestionar el conocimiento, sé que estamos avanzando, y me siento parte de ello que aportamos y me siento motivo en el camino de la investigación.
- A la Universidad Nacional del Altiplano mi alma mater por darnos esta oportunidad de estudiar doctorado en Ciencias de la Computación, lo tomé desde un inicio con mucho entusiasmo.



ÍNDICE GENERAL

	Pág.
DEDICATORIA	i
AGRADECIMIENTOS	ii
ÍNDICE GENERAL	iii
ÍNDICE DE TABLAS	vi
ÍNDICE DE FIGURAS	vii
ÍNDICE DE ANEXOS	viii
RESUMEN	ix
ABSTRACT	x
INTRODUCCIÓN	1

CAPÍTULO I

REVISIÓN DE LA LITERATURA

1.1. Marco teórico	2
1.1.1. Patente type writen machine	2
1.1.2. Máquina de escribir 1714	3
1.1.3. Máquina de escribir 1870	3
1.1.4. Máquina de escribir 1880	4
1.1.5. El dominante teclado Qwerty para máquinas de escribir	4
1.1.6. Teclado doble mayúscula/minúscula	5
1.1.7. Personal computer 1980 y su teclado	5
1.1.8. Teclado de Teléfonos móviles 1990 hasta lo presente	6
1.1.9. El dominante método de entrada de textos con teclado de 12 botones	7
1.1.10. Métodos alternativos de introducción de texto para teléfonos móviles	8
1.1.11. Teclado qwerty reducido para dispositivos móviles	9
1.1.12. Teclado en computadoras portátiles	10
1.1.13. Entradas de texto para PDAs	11
1.1.14. Mini-teclado físico	12
1.1.15. Teclado Qwerty keyboard (virtual Qwerty Keyboard)	12
1.1.16. Método de entrada de texto Wip-activated	13
1.1.17. Inteligencia Artificial	14
1.1.18. Algoritmos de optimización en inteligencia artificial	14
1.1.19. Heurísticas de los algoritmos de optimización	14



1.1.20.	Metaheurísticas	15
1.1.21.	Algoritmo Simulated Annealing	15
1.1.22.	Teclado QWERTY	16
1.1.23.	Teclado Virtual QWERTY (software)	16
1.2.	Antecedentes	17
1.2.1.	Modificaciones y propuestas al teclado físico	17
1.2.2.	Modificaciones sobre teclados virtuales (Virtual Keyboards)	23

CAPÍTULO II

PLANTEAMIENTO DEL PROBLEMA

2.1.	Identificación del problema	25
2.2.	Enunciados del problema	28
2.2.1.	Problema general	28
2.2.2.	Problemas específicos	28
2.3.	Justificación	29
2.4.	Objetivos	30
2.4.1.	Objetivo general	30
2.4.2.	Objetivos específicos	30
2.5.	Hipótesis	30
2.5.1.	Hipótesis general	30
2.5.2.	Hipótesis específicas	30

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1.	Lugar de estudio	31
3.2.	Población	31
3.3.	Muestra	32
3.4.	Método de investigación	32
3.5.	Descripción detallada de métodos por objetivos específicos	33
3.5.1.	OE1: Metodología de diseño del modelo de optimización de distribución de teclas del teclado virtual utilizando el algoritmo de Inteligencia Artificial Simulated Annealing	33
3.5.2.	OE3: Metodología de validación del modelo de optimización de distribución de teclas del teclado virtual utilizando miles de palabras	46



3.5.3. OE2: Metodología de diseño mockups de un nuevo teclado virtual denominado Ray-Keyboard para dispositivos móviles basado en la distribución de teclas optimizadas	47
---	----

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1. Datos de entrada para el modelo de optimización.	49
4.2. Análisis de resultados de optimización de distribución de teclado	51
4.3. Nueva distribución del teclado virtual	57
4.4. Diseño de los mockups del nuevo teclado virtual RAY Keyboard con distribución de teclas optimizada.	59
4.5. Discusión de resultados	64
CONCLUSIONES	66
RECOMENDACIONES	67
BIBLIOGRAFÍA	68
ANEXOS	72

Puno, 27 de enero de 2021.

Área: Ciencias de la Ingeniería
Tema: Optimización de posiciones del teclado
Línea: Sistemas, Computación e Informática



ÍNDICE DE TABLAS

	Pág.
1. Conjunto de teclas para la muestra	32
2. Posiciones de la distribución del teclado Qwerty	39
3. Distancias entre los pares de teclas ubicadas en el plano cartesiano	42
4. Estructura de salida del modelo (nueva distribución del teclado)	47
5. Textos múltiples como datos de entrada	50
6. Estado inicial del algoritmo Simulated Annealing	51
7. Resultados del intercambio del “Estado Actual” y el “Nuevo Estado”	53
8. Resultados de reemplazamiento del “Estado Actual” con el “Nuevo Estado”	54
9. Resultados de reemplazamiento al “Mejor Estado”	55
10. Salida final del modelo de optimización	57
11. Posiciones de las teclas de la nueva distribución	58

ÍNDICE DE FIGURAS

	Pág.
1. Tipo Writing Machine	3
2. Sholes – Glidden typewriter	4
3. Smith Premier - double keyboard typewriter	5
4. Key keypad on a mobile phone	8
5. Izquierda: teclado Fastap, derecha: teclado Qwerty reducido de BlackBerry	9
6. Conjunto de caracteres de Unistrokes (Goldberg y Richardson, 1993)	12
7. Teclado Qwerty del Nokia 9500	12
8. Diseño de teclado activado por limpieza de Montgomery	13
9. The Dvorak Simplified Keyboard (1936)	18
10. Teclado Komark	20
11. Teclado Keybowl	20
12. Teclado Datahand	21
13. The standard QWERTY layout, American National Standard Typewriter	25
14. Teclado español de las computadoras	31
15. Opción de Swype (deslizamiento) de los teclados	34
16. Modelo propuesto de optimización de distribución de las teclas	35
17. Sistema de recolección de textos con palabras	36
18. Teclado Qwerty como configuración inicial para el algoritmo de optimización	37
19. Teclado Qwerty en el plano cartesiano	38
20. Algoritmo Simulated Annealing	40
21. Medición de temperatura de Simulated Annealing	43
22. Teclado Qwerty como entrada para el modelo	43
23. Intercambio entre dos teclas	44
24. Función de aceptación	45
25. Configuración de temperatura y velocidad	45
26. Archivos .txt recolectados para datos de entrada	49
27. Teclado virtual con nueva distribución	59
28. Posición del teclado virtual en el dispositivo móvil	60
29. Teclado virtual con nueva distribución	61
30. Teclado numérico del nuevo teclado virtual	61
31. Opción de deslizamiento del nuevo teclado virtual	62
32. Teclado virtual a la izquierda para una sola mano	63
33. Teclado virtual a la derecha para una sola mano	63
34. Temas y estilos del teclado virtual	64



ÍNDICE DE ANEXOS

	Pág.
1. Código fuente del modelo de optimización	73

RESUMEN

El teclado de las computadoras QWERTY es el más usado internacionalmente en diferentes dispositivos, su origen data desde 1868, a pesar de que sufrió cambios el orden de posiciones de las teclas se ha impuesto como teclado oficial en las computadoras, también los dispositivos móviles adaptaron el teclado QWERTY como distribución del teclado virtual. Sin embargo, el teclado QWERTY no está optimizado su distribución de posiciones, otros investigadores también criticaron su distribución. En esta investigación proponemos un modelo de optimización de distribución del teclado utilizando el algoritmo de inteligencia artificial Simulated Annealing. En el diseño de la investigación se ha tomado textos que conforman miles de palabras, estas palabras se alimentaron al modelo como entrada, como estado inicial para la optimización se ha tomado el teclado Qwerty en posiciones en el plano cartesiano con la idea de calcular distancia entre teclas, en la optimización se ha buscado la menor distancia en sumatoria del trazo del conjunto de palabras con el teclado, en cada iteración se intercambia las posiciones de las teclas buscando la menor distancia total comparando con el anterior. Mostramos como resultado la distancia inicial 9,823,684.64 comparado con la distancia después de optimizar 6547352.95 logrando una diferencia de 3,276,331.69 (más de tres millones de unidades en distancia al optimizar). Con la nueva distribución de teclado optimizado diseñamos un nuevo teclado virtual denominado RAY Keyboard para dispositivos móviles, la cual permite escribir en menor tiempo un texto en los dispositivos móviles. Finalmente, discutimos nuestros resultados con otros estudios.

Palabras clave: Menor distancia, optimización, Simulated Annealing, teclado QWERTY, teclado virtual.



ABSTRACT

The QWERTY computer keyboard is the most used internationally in different devices, its origin dates from 1868, although the order of the positions of the keys underwent changes, it has been imposed as the official keyboard in computers. Mobile devices also adapted the QWERTY keyboard as a virtual keyboard layout. However, the QWERTY keyboard is not optimized for its position layout, other researchers also criticized its layout. In this research we propose a keyboard layout optimization model using the Simulated Annealing artificial intelligence algorithm. In the research design, texts that make up thousands of words have been taken, these words were fed to the model as input, as an initial state for optimization, the Qwerty keyboard has been taken in positions in the Cartesian plane with the idea of calculating distance between keys, in the optimization we have looked for the smallest distance in sum of the stroke of the set of words with the keyboard. In each iteration, the positions of the keys are exchanged looking for the smallest total distance comparing with the previous one. We show, as a result, the initial distance 9,823,684.64 compared to the distance after optimizing 6547352.95 achieving a difference of 3,276,331.69 (more than three million units in distance when optimizing). With the new optimized keyboard layout we designed a new virtual keyboard called RAY Keyboard for mobile devices, which allows writing a text in less time on mobile devices. Finally, we discuss our results with other studies.

Keywords: Keyboard QWERTY, Optimization, Simulated Annealing, Shorter distance, Virtual keyboard.

INTRODUCCIÓN

El teléfono móvil es parte del ser humano usado como herramienta principal en el que hacer de cada uno, entre tantas funcionalidades de estos dispositivos lo usamos para escribir todos los días una palabra, una frase, un mensaje o un documento, en ello invertimos tiempo necesario para interactuar mediante estos textos escritos.

Si miramos la historia del teclado que nos permite escribir los textos, este modelo de teclado proviene desde 1860 con la invención de tipos para generar algunas letras, al pasar los años se ha posicionado el modelo del teclado Qwerty, aunque ha sido criticado por los investigadores en el sentido que no hubo un fundamento suficiente para colocar en ese orden de posiciones del teclado, lo cierto que es que por costumbre lo usamos en ese modelo.

Hubieron varias propuestas en modificar las posiciones del teclado Qwerty, el más resaltante modificación fue realizado por Dvorak entre otros, en esta investigación proponemos un modelo de optimización de distribución del teclado virtual utilizando el algoritmo de inteligencia artificial Simulated Annealing.

Logramos obtener la mejor distribución del teclado virtual utilizando el algoritmo Simulated Annealing, alimentamos con miles de palabras y frases recolectados desde documentos formales como libros, cada palabra se considera una trayectoria en el plano cartesiano ubicado las letras, esto permite calcular la distancia de la trayectoria total con el conjunto de frases o palabras proporcionadas, en cada cambio de posición se evalúa la distancia de esta manera logramos reubicar las posiciones de las teclas, finalmente proponemos un nuevo teclado virtual desarrollado para dispositivos móviles Android.

El trabajo de investigación está organizado de la siguiente forma: Capítulo I, presentamos la Revisión de la literatura, donde describimos los modelos de teclados y sus cambios en la historia, también presentamos los antecedentes con los análisis de artículos científicos que aportan nuevos modelos de teclado. En Capítulo II presentamos el problema de la investigación describiendo su fundamento de la propuesta del nuevo modelo de optimización de teclado, así mismo planteamos los objetivos y justificación. En el Capítulo III describimos detalladamente la metodología de la propuesta del diseño del modelo de optimización para la distribución de las teclas. En el Capítulo IV presentamos resultados de la investigación y discutimos con otras propuestas.



CAPÍTULO I

REVISIÓN DE LA LITERATURA

1.1. Marco teórico

1.1.1. Patente type writen machine

El teclado que usamos internacionalmente conocido QWERTY fue inventado por Sholes *et al.* (1868) en el condado de Milwaekee y en el estado de Wisconsin, inventaron ciertas mejoras nuevas y útiles en las máquinas para escribir o imprimir correspondencia. En esta invención refieren a cierta clase de máquinas diseñadas para escribir con tipos en lugar de un bolígrafo; y la naturaleza y características principales de las mejoras consisten en un disco circular anular, provisto de ranuras y ranuras para sostener y guiar las barras tipo; una ranura preocupante alrededor de la periferia del disco, para sujetar, sostener y guiar los pivotes de las barras tipo; la combinación de un disco circular ranurado con barras de tipo, montadas y pivotadas en el mismo; la combinación de varillas, palancas y llaves para trabajar las barras tipo; un carro combinado y provisto de un trinquete pivotante, brazo y pasadores, y accesorios para mover el papel vertical y lateralmente, y la combinación de una varilla y abrazaderas para sujetar el flautista al carro.

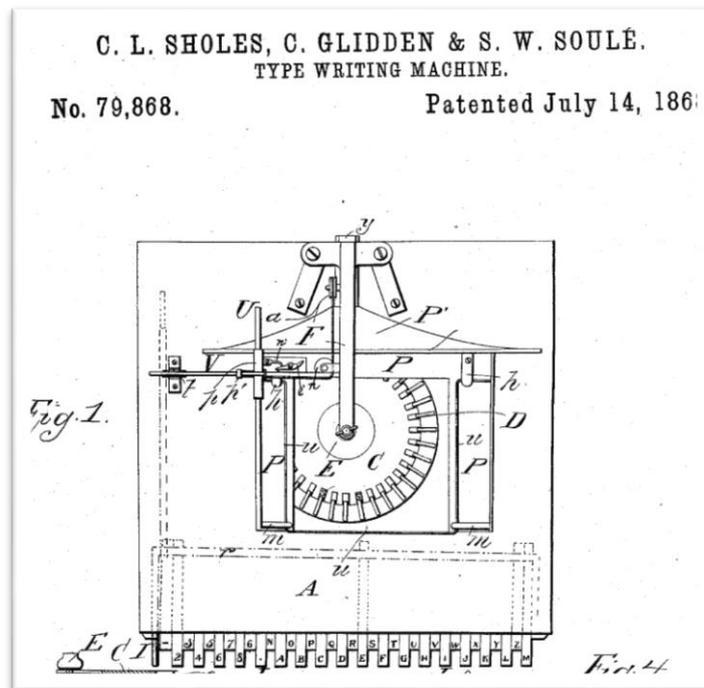


Figura 1. Tipo Writing Machine

1.1.2. Máquina de escribir 1714

La primera máquina de escribir con éxito comercial fue la “Escritura tipográfica de Sholes-Glidden”, en 1874 ha sido producida por la empresa Remington y Sons, entre las características de esta máquina de escribir es que su teclado tenía una similitud sorprendente con la distribución actual del teclado de la computadora.

La bola de escritura hecha por Malling Hansen había estado en el mercado desde 1870, aunque tuvo poco éxito. Se cree que en 1714 la primera máquina de escribir fue inventada por Henrik Mill, pero la mayoría de estos primeros inventos del teclado no se pusieron en práctica o, si lo fueron, no tuvieron éxito comercial (Yamada, 1980).

1.1.3. Máquina de escribir 1870

La primera máquina de escribir con éxito comercial fue la “Escritura tipográfica de Sholes-Glidden”, entre las características de esta máquina de escribir es que su teclado tenía una similitud sorprendente con la distribución actual del teclado de la computadora. Esta máquina de escribir fue desarrollada por Sholes *et al.* (1868). El inventor principal, Latham Sholes aunque no fue el primer inventor sino el 52°

inventor de máquina de escribir, fue posteriormente referido como el padre de la máquina de escribir, pero la máquina Sholes-Glidden no fue en realidad la primera máquina de escribir vendida comercialmente.

1.1.4. Máquina de escribir 1880

Una de las razones por las que la línea de máquinas Remington, a partir de la primera versión de Sholes-Glidden, fue un éxito fue que era el momento adecuado para un nuevo método de introducción de texto. El final del siglo XIX fue una época de fuerte industrialización. Las pequeñas empresas se estaban convirtiendo en grandes corporaciones, lo que creó la necesidad de mejorar la comunicación. También aparecieron otros inventos al mismo tiempo, por ejemplo, la luz eléctrica, el telégrafo y el teléfono. Estos inventos ayudaron a satisfacer la necesidad de una mayor comunicación. La máquina de escribir encajaba muy bien en la ecuación, y la sorprendente capacidad de crear documentos inequívocos con rapidez fue bien recibida en un entorno empresarial tan floreciente. A fines de la década de 1880, el monótono clic de la máquina de escribir se podía escuchar en casi todos los establecimientos comerciales bien considerados del país EE.UU (Yamada, 1980).



Figura 2. Sholes – Glidden typewriter

1.1.5. El dominante teclado Qwerty para máquinas de escribir

El nombre Qwerty proviene de las primeras seis letras de la fila superior de teclas alfabéticas. El diseño fue inventado por Sholes. Contrariamente a la creencia

común, el diseño en sí nunca fue patentado. Sholes y sus colegas tienen varias patentes, pero se encuentran principalmente en los mecanismos del teclado por ejemplo, (Sholes *et al.*, 1868). El diseño Qwerty fue posteriormente copiado por Remington para los sucesores del primer modelo Sholes-Glidden, así como por muchos otros fabricantes de máquinas de escribir.

Las ubicaciones de los teclados se eligieron principalmente para superar las limitaciones mecánicas de las primeras máquinas. Sholes dispuso las barras tipográficas de modo que algunos digramas frecuentes en el idioma inglés estuvieran ubicados lejos unos de otros en la disposición de barras tipográficas.

1.1.6. Teclado doble mayúscula/minúscula

En 1874 Sholes *et al.* (1868) original sólo podía ingresar letras mayúsculas, el desplazamiento de la platina se introdujo en la segunda generación: la máquina de escribir Remington Modelo 2 en 1978, sin embargo, los llamados teclados dobles proporcionaron una solución muy diferente, en estas máquinas había dos juegos completos de teclas alfabéticas, una para las minúsculas y otra para las mayúsculas.

Posteriormente se fabricaron máquinas de doble teclado, así como de Smith y Zhai (2001), el último fabricante de teclados dobles, presentó su primera máquina de tecla Mayúscula y los teclados dobles desaparecieron del mercado.



Figura 3. Smith Premier - double keyboard typewriter

1.1.7. Personal computer 1980 y su teclado

Las computadoras personales reemplazaron rápidamente las máquinas de escribir gracias a sus características más avanzadas.

Aunque las computadoras personales son claramente más avanzadas que las máquinas de escribir en muchos aspectos, el método de entrada de texto simplemente se copió de la máquina de escribir. Este fue un camino natural, ya que, las últimas máquinas de escribir electrónicas ya se habían convertido en computadoras simples. Dado que los teclados de computadora podían reemplazarse y cambiarse más fácilmente que los de las máquinas de escribir mecánicas, se podría esperar ver más variación en la distribución del teclado. Sin embargo, la situación es más bien contraria. La prevalencia de Qwerty ha sido asombrosa. La variación en los teclados se ha limitado a la adición de algunas teclas especiales (para, por ejemplo, navegar por Internet o juegos de computadora) o algunas optimizaciones ergonómicas, como el teclado Microsoft Natural. Quizás otra razón de la falta de variación, además de la paradoja de la producción mencionada anteriormente, podría ser una razón de infraestructura. Hoy en día, las computadoras se pueden encontrar en todas partes y, a menudo, son utilizadas por múltiples usuarios. Por lo tanto, es difícil cambiar el teclado en solo algunas de las computadoras, mientras que otras computadoras continuarían presentando diseños Qwerty. Esto conduciría a situaciones en las que un solo usuario tendría que usar y recordar varios diseños durante un día.

1.1.8. Teclado de Teléfonos móviles 1990 hasta lo presente

El teléfono se inventó a finales del siglo XIX. Al igual que con las máquinas de escribir, no hay consenso sobre quién fue el inventor original, pero el nombre de Alexander Graham Bell se menciona con mayor frecuencia en este contexto. Los primeros teléfonos usaban centralitas manuales y, más tarde, diales giratorios, pero el teclado numérico de 12 teclas se introdujo ya a mediados del siglo XX (véanse especialmente los estudios legendarios de R.L. (Deininger, 1960).

En la década de 1980 aparecieron los primeros teléfonos móviles (también llamados teléfonos móviles). Al principio, los teléfonos móviles se usaban de manera muy similar a los teléfonos fijos, es decir, principalmente para hablar. La interfaz de usuario también se copió del predecesor, y la mayoría de los teléfonos usan el teclado de 12 teclas incluso en la actualidad. Sin embargo, más recientemente, los teléfonos móviles han adquirido una gran cantidad de funciones nuevas. Los modelos más avanzados se están volviendo muy similares a las computadoras de

mano. Sin embargo, la mensajería de texto se ha convertido en otra piedra angular de estos dispositivos. Los teléfonos móviles entraron en la era de la mensajería de texto cuando surgió el servicio de mensajes cortos (SMS) en los teléfonos GSM a principios de la década de 1990.

Muy pronto, casi todos los teléfonos GSM incorporaron mensajería SMS y, en la actualidad, la mensajería instantánea, la mensajería multimedia e incluso el correo electrónico a gran escala también se han convertido en funciones utilizadas en todos los teléfonos móviles. La Asociación GSM 2006 estima que un total mundial de 1 billón de mensajes de texto SMS fueron enviados durante el año 2005.

1.1.9. El dominante método de entrada de textos con teclado de 12 botones

El teclado de 12 teclas presentó un desafío especial para la entrada de texto, ya que la mayoría de los idiomas tienen al menos 26 caracteres. Las dos técnicas dominantes utilizadas en los teléfonos móviles, las cuales pueden aparecer en el mismo dispositivo, son los métodos predictivos y de múltiples pulsaciones.

En el método de múltiples pulsaciones, cada tecla se presiona una o más veces para especificar qué carácter se desea. Este método aumenta el número de pulsaciones de teclas y, por tanto, es bastante lento. Son típicos valores de velocidad de escritura de alrededor de 10 palabras por minuto. En los métodos predictivos, generalmente se usa un diccionario en la memoria del teléfono para encontrar las palabras que coinciden con la secuencia de teclas. El método predictivo más conocido es probablemente el T9 de Tegic (www.t9.com), pero también existen otros métodos similares. En T9, el usuario pulsa una tecla para cada carácter, y la secuencia de teclas para toda la palabra coincide con el diccionario del teléfono. Si la predicción funciona bien, se puede ingresar texto con solo presionar una tecla por carácter. Sin embargo, la fluidez de la escritura depende en gran medida de la coincidencia entre el vocabulario del usuario y el diccionario del teléfono. Si la palabra deseada no está en el diccionario, el usuario debe hacer un esfuerzo adicional. Por lo tanto, no es fácil dar una estimación de velocidad única para métodos predictivos. En casos óptimos, cuando todas las palabras coinciden con el diccionario, se pueden obtener fácilmente velocidades de aproximadamente 20 palabras por minuto. En el peor de

los casos, la velocidad puede caer por debajo del nivel de velocidad de varias pulsaciones.



Figura 4. Key keypad on a mobile phone

1.1.10. Métodos alternativos de introducción de texto para teléfonos móviles

El teclado del teléfono móvil de 12 teclas es una función tan limitante para la entrada de texto en algunos teléfonos inteligentes avanzados los cuales están diseñados para admitir, por ejemplo, correo electrónico a gran escala, han abandonado por completo la entrada de texto con la tecla de 12 teclas.

Fastap, anteriormente conocido como OneTouch, tiene las 12 teclas numéricas estándar, pero las teclas de caracteres A – Z y algunas teclas de puntuación están entrelazadas entre ellas. Las teclas de caracteres están organizadas en orden alfabético. Todo el teclado es bastante pequeño, contiene más de 40 teclas en un espacio no mucho más grande que el teclado de un teléfono estándar. Por otro lado, todas las letras frecuentes en el idioma inglés están disponibles presionando una tecla. Esto hace que el diseño sea potencialmente bastante rápido (Levy, 2002).

Otro enfoque, utilizado en algunos dispositivos comerciales existentes (por ejemplo, Sony Ericsson M600 y BlackBerry 7100), es retener el diseño del número del teclado del teléfono y superponerlo con un teclado Qwerty reducido, en este caso, se refiere a los caracteres que se ordenan en una forma Qwerty típica, pero el número de teclas se ha reducido colocando dos o más caracteres en cada tecla. De manera similar a la entrada del teclado del teléfono, aquí también se pueden utilizar técnicas predictivas y de múltiples pulsaciones para superar la ambigüedad.

El dispositivo SonyEricsson utiliza una nueva estrategia, en la que cada tecla se puede inclinar presionándolas hacia abajo en el lado izquierdo o derecho de la tecla para seleccionar uno de los dos caracteres. Estos diseños son interesantes, pero todavía son relativamente nuevos, y hasta ahora no se han publicado estudios de usabilidad de ninguno de los diseños Qwerty reducidos.

1.1.11. Teclado qwerty reducido para dispositivos móviles

Tanto Fastap como los diseños Qwerty reducidos son bastante nuevos y no han sido ampliamente adoptados en dispositivos móviles. Quizás esto cambie en un futuro próximo. La situación recuerda la posición del diseño de Dvorak en el mundo de la informática dominado por Qwerty. Un nuevo diseño probablemente solo tendrá éxito si puede superar la paradoja de la producción. Es decir, los usuarios probablemente adoptarán un nuevo diseño solo si cumple dos necesidades. Primero, el diseño debe mostrar un beneficio claro para el usuario, por ejemplo, en términos de eficiencia mejorada o comodidad de uso. Y, en segundo lugar, el diseño debe ser bastante fácil de aprender. El primer punto puede ser difícil de estimar para el cliente individual; tal vez podría ser probado por la investigación, pero hasta ahora existen muy pocos estudios, y es casi seguro que no llegue a los clientes individuales. El segundo punto, la facilidad de aprendizaje, también puede ser difícil de estimar. Sin embargo, el diseño Qwerty reducido podría estar en una mejor posición a este respecto, ya que el diseño Qwerty familiar podría hacer que sea más fácil de aprender o al menos podría hacerlo parecer (Levy, 2002).



Figura 5. Izquierda: teclado Fastap, derecha: teclado Qwerty reducido de BlackBerry

1.1.12. Teclado en computadoras portátiles

La idea de dispositivos informáticos de mano potentes ha existido al menos desde 1968, cuando Alan Kay imaginó el concepto (Meyer, 1995). Era una idea de un dispositivo pequeño y liviano que podría usarse para tomar notas e interactuar de forma inalámbrica. Obviamente, a finales de la década de 1960, la tecnología no estaba lo suficientemente madura para realizar el Dynabook. Sin embargo, su idea alentó conceptos y prototipos posteriores. En 1987, Apple presentó un prototipo del concepto Knowledge Navigator, que a su vez evolucionó hasta convertirse en la primera computadora portátil comercial, la Newton Message Pad de Apple. El primer Newton fue lanzado en 1993, seguido de varios dispositivos avanzados durante los siguientes 5 años.

Los Newton eran dispositivos muy avanzados, con muchas características visionarias, y tenían una base de usuarios relativamente pequeña pero muy leal. En el lado negativo, los dispositivos eran bastante pesados, voluminosos y costosos, y algunas de las tecnologías (por ejemplo, el reconocimiento avanzado de escritura a mano) aún no estaban muy maduras.

En 1996, apareció un dispositivo operado con lápiz claramente más pequeño, más simple y asequible: el organizador Pilot. A pesar de su inferioridad técnica con respecto al Newton, el Pilot parecía satisfacer mejor las necesidades de los usuarios. Las ventas de Pilot aumentaron rápidamente, mientras que las de Newton disminuyeron. En febrero de 1998, Apple discontinuó oficialmente Newton y todos los productos relacionados.

Aparte de estos dispositivos operados con lápiz, había otra categoría de dispositivos que usaban un teclado Qwerty completo. Los organizadores de Psion en particular eran dispositivos populares y avanzados. Ya no se fabrican, pero se han convertido en dispositivos Symbian, como Nokia Communicators. Symbian también se usa en teléfonos inteligentes, que hoy en día cubren una gama tan amplia de funciones que también se pueden clasificar como computadoras de mano.

1.1.13. Entradas de texto para PDAs

Las entradas de texto para los PDAs como HP2000, BlackBerry, Motorola y Nokia, fueron diseñados con el teclado Qwerty en miniatura (Mackenzie y Soukoreff, 2011). En otro tipo de PDAs el dispositivo de entrada típicamente comprende un lápiz y una pantalla sensible al tacto.

En 1993, se vendió el primer dispositivo Apple Newton, Goldberg y Richardson de Xerox PARC. Mientras que Newton intentó proporcionar un método completamente natural de entrada de texto escrito a mano, y desafortunadamente fracasó, la técnica de Unistrokes se distanció completamente del uso natural. En lugar de permitir que el usuario escriba libremente, Unistrokes buscó optimizar la precisión del reconocimiento y la velocidad de entrada de texto. Goldberg y Richardson querían diseñar un conjunto de caracteres que se pudiera ingresar sin ojos. Para cumplir con estos requisitos, se diseñó un conjunto de caracteres completamente nuevo y altamente simplificado, en el que cada carácter podía dibujarse con un solo trazo (de ahí el nombre Unistrokes). El primer organizador piloto en 1996 introdujo Graffiti. Graffiti es similar a Unistrokes en que la mayoría de sus caracteres se pueden ingresar con un solo trazo, pero los caracteres de Graffiti son menos artificiales que los Unistrokes y mucho más cercanos a las letras impresas. Esta naturalidad se desarrolló aún más en versiones posteriores de Graffiti y otros sistemas similares, como el que se usa en las Pocket PC con Windows.

El graffiti recibió mucha atención a fines de la década de 1990 y todavía tiene sus defensores. Sin embargo, otros métodos de entrada de texto, como los teclados virtuales y los pequeños teclados mecánicos (Qwerty), se han vuelto cada vez más populares en las computadoras portátiles recientes.

Los reconocedores de escritura a mano también están disponibles para todos los principales sistemas operativos de PC, como la versión Tablet PC de Microsoft XP y el Inkwel para Mac OS X. Las PC que funcionan con lápiz todavía se utilizan principalmente, por ejemplo, por artistas gráficos y su uso depende mucho más de sus características de dibujo que de la entrada de texto.

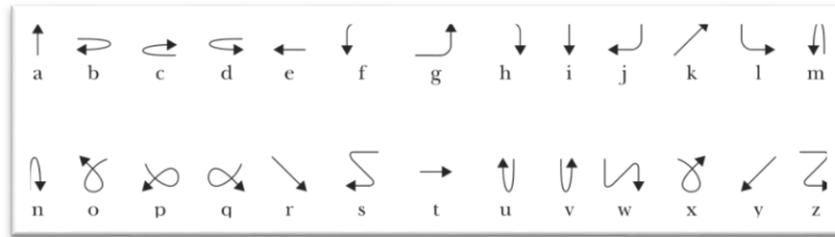


Figura 6. Caracteres de Unistrokes de Goldberg y Richardson en 1993

1.1.14. Mini-teclado físico

A diferencia del teclado virtual en pantalla, muchas computadoras portátiles y teléfonos inteligentes recientes también tienen un pequeño teclado mecánico. El diseño más utilizado es el Qwerty. Si bien el teclado de PC de tamaño completo es adecuado para el uso completo con 10 dedos, incluso para la escritura táctil, el uso de los mini teclados móviles se limita principalmente a la escritura con dos dedos.

La mayoría de los usuarios sostienen el dispositivo con las dos manos y usan los dos pulgares para escribir. Aunque tal escritura con el pulgar es obviamente más torpe que la mecanografía completa con 10 dedos.

Sin embargo, la desventaja de estos teclados, así como de cualquier teclado operado con los dedos, es que pueden crear lesiones por esfuerzo repetitivo con un uso extenso, especialmente porque los pulgares no son tan ágiles como, por ejemplo, los dedos índices. Por el momento, los mini-teclados y la extensa entrada de texto móvil son fenómenos relativamente nuevos, y hasta ahora no han surgido estudios sobre problemas ergonómicos.



Figura 7. Teclado Qwerty del Nokia 9500

1.1.15. Teclado Qwerty keyboard (virtual Qwerty Keyboard)

Con los teclados tradicionales, las teclas se presionan una a la vez para crear caracteres en secuencia. Los teclados de acordes requieren presionar varias teclas

simultáneamente para crear un carácter, como al tocar un acorde en un piano. Dado que cada tecla se puede utilizar en varias combinaciones para crear letras diferentes, el número de teclas en un teclado de acordes se puede reducir considerablemente. Por ejemplo, un teclado de acordes de cinco teclas puede producir hasta 31 caracteres. El principal interés en los teclados de acordes ha sido la clasificación de correo, tareas en la oficina de correos (Noyes, 1983).

1.1.16. Método de entrada de texto Wip-activated

Otro nuevo método de operación del teclado se basa en la tecnología de capacitancia. Debido a que no se trata de un viaje hacia abajo, el "teclado" se puede diseñar como una tableta plana sobre la que los dedos se deslizan o limpian para crear caracteres. El método alternativo de funcionamiento es utilizar un lápiz óptico para interactuar con el teclado. El Centro de Ciencias de la Salud de la Universidad de Texas en Dallas ha propuesto un teclado de este tipo. Para aprovechar el movimiento de limpieza, Montgomery también desarrolló un nuevo diseño de caracteres que se puede ver en la Figura 8, con esta disposición es posible crear muchas palabras con un solo movimiento de limpieza.

No se han informado pruebas (aparte de los análisis que comparan el número de toallitas con el número de pulsaciones de teclas necesarias en los teclados tradicionales) para determinar el rendimiento de los efectos de preferencia con este teclado. Dado que no tiene partes móviles, este dispositivo se puede producir en casi cualquier tamaño y puede ser utilizado fácilmente por personas discapacitadas.

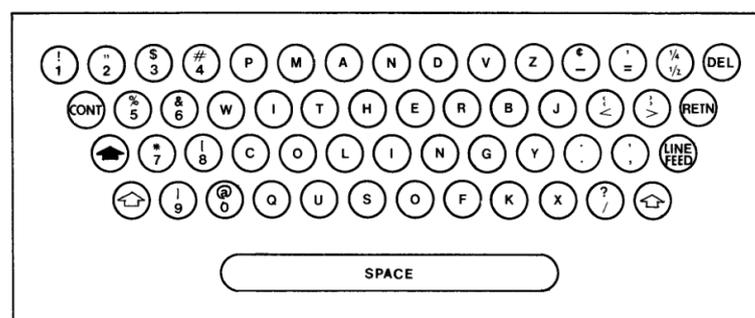


Figura 8. Diseño de teclado activado por limpieza de Montgomery

1.1.17. Inteligencia Artificial

La IA es la rama de la ciencia que se encarga del estudio de la inteligencia en elementos artificiales y, desde el punto de vista de la ingeniería, propone la creación de elementos que posean un comportamiento inteligente. Dicho de otra forma, la IA pretende construir sistemas y máquinas que presenten un comportamiento que si fuera llevado a cabo por una persona, se diría que es inteligente. El aprendizaje, la capacidad de adaptación a entornos cambiantes, la creatividad, etc., son facetas que usualmente se relacionan con el comportamiento inteligente.

Además, la IA es muy interdisciplinar, y en ella intervienen disciplinas tan variadas como la Neurociencia, la Psicología, las Tecnologías de la Información, la Ciencia Cognitiva, la Física, las Matemáticas (Russell y Norving, 2010).

1.1.18. Algoritmos de optimización en inteligencia artificial

Esta forma sistemática se alcanza manteniendo uno o más caminos en memoria y registrando qué alternativas se han explorado en cada punto a lo largo del camino y cuáles no. Cuando se encuentra un objetivo, el camino a ese objetivo también constituye una solución al problema. los algoritmos de búsqueda local son útiles para resolver problemas de optimización puros, en los cuales el objetivo es encontrar el mejor estado según una función objetivo (Russell y Norving, 2010).

1.1.19. Heurísticas de los algoritmos de optimización

La heurística un proceso que puede resolver un cierto problema, es una información adicional sobre el problema que apoya a buscar la mejor solución. Las heurísticas fueron un área predominante en los orígenes de la Inteligencia Artificial. Actualmente, el término suele usarse como un adjetivo, refiriéndose a cualquier técnica que mejore el desempeño en promedio de la solución de un problema, aunque no mejore necesariamente el desempeño en el peor caso (Russell y Norving, 2010).

1.1.20. Metaheurísticas

Una Metaheurística es un método que se aplica para resolver problemas genéricos. Combina funciones objetivo o heurísticas de una forma eficiente y abstracta que usualmente no dependen de la estructura del problema.

Los procedimientos Metaheurísticos son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria, en los que los heurísticos clásicos no son efectivos. Los Metaheurísticos proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos (Osman y Kelly, 1996).

1.1.21. Algoritmo Simulated Annealing

Un algoritmo de ascensión de colinas que nunca hace movimientos «cuesta abajo» hacia estados con un valor inferior (o coste más alto) garantiza ser incompleto, porque puede estancarse en un máximo local. En contraste, un camino puramente aleatorio, es decir moviéndose a un sucesor elegido uniformemente aleatorio de un conjunto de sucesores, es completo, pero sumamente ineficaz. Por lo tanto, parece razonable intentar combinar la ascensión de colinas con un camino aleatorio de algún modo que produzca tanta eficacia como completitud. El temple simulado es ese algoritmo. En metalurgia, el temple es el proceso utilizado para templar o endurecer metales y cristales calentándolos a una temperatura alta y luego gradualmente enfriarlos, así permite al material fundirse en un estado cristalino de energía baja. Para entender el temple simulado, cambiemos nuestro punto de vista de la ascensión de colinas al gradiente descendente (es decir, minimizando el coste) e imaginemos la tarea de colocar una pelota de ping-pong en la grieta más profunda en una superficie desigual. Si dejamos solamente rodar a la pelota, se parará en un mínimo local. Si sacudimos la superficie, podemos echar la pelota del mínimo local. El truco es sacudir con bastante fuerza para echar la pelota de mínimos locales, pero no lo bastante fuerte para desalojarlo del mínimo global. La solución del temple simulado debe comenzar sacudiendo con fuerza (es decir, a una temperatura alta) y luego gradualmente reducir la intensidad de la sacudida es igual a bajar la temperatura (Russell y Norving, 2010).

1.1.22. Teclado QWERTY

El originario del teclado QWERTY fue patentada por Sholes *et al.* (1868), teclado orientado para el idioma inglés, fue diseñado para “cazar y picotear” y no para teclear como ahora la tenemos, aunque el objetivo de los inventores era diseñar una máquina que pudiera imprimir palabras a una velocidad equivalente a la escritura a mano, la patente fue cedida a otros y fueron modificados. La primera patente que muestra el diseño del teclado QWERTY de (Beeching, 1894), en el mismo año la empresa Remington Arms Company comenzó a fabricar la máquina de escribir en cantidad.

Las críticas al teclado QWERTY, que el dedo anular y el meñique son forzados a estirar desde la fila de posición inicial a las tercera y cuarta fila, esto reduce la fuerza del golpe porque es conducido al borde de la punta del dedo en lugar del centro de la tecla para golpear adecuadamente; que la división de las teclas como filas para diferentes dedos forman líneas oblicuas paralelas de manera que para ambos manos izquierda y derecha son la misma forma esto es incongruente por que las dos manos son imágenes inversas entre sí (Biegel, 1934).

El teclado Qwerty ha sido revisado las teclas caracteres alfanuméricos y de puntuación, se verificó los méritos y la ocurrencia de digramas, trigramas, y diptongos en una investigación se concluyó que el teclado QWERTY no estaba coordinado para el idioma inglés y no estaba adaptado para las manos del mecanógrafo y fue lanzado un pronóstico que el teclado QWERTY fallaría (Gilbert, 1930). Posteriormente explicó cómo habría sido diseñado las posiciones de las teclas de QWERTY (Cocking, 1970).

1.1.23. Teclado Virtual QWERTY (software)

Un teclado virtual es un componente de software que permite la entrada de caracteres sin la necesidad de teclas físicas. La interacción con el teclado virtual ocurre principalmente a través de una interfaz de pantalla táctil, pero también puede tener lugar de forma diferente en realidad virtual o aumentada (Mackenzie y Soukoreff, 2011).

Los smartphones ya es bastante usado a nivel mundial, más aún actualmente muchos usan Smartwatches estas también tienen un Virtual Keyboard, pero que no parece ser tan funcional con su teclado, en una investigación proponen un nuevo teclado para smartwatches que permite ingresar alfabetos más frecuentes en un solo toque, otras teclas símbolos y números pueden ser ingresadas en doble toque, muestran resultados de promedio de palabras ingresadas por minuto (Rajkumar *et al.*, 2016).

En otra investigación analizaron que en los smartphones hay más frecuencia de movimientos entre el teclado virtual y las áreas de textos, por lo que diseñaron dos tipos de teclados virtuales el estático y dinámico; el estático muestra de carácter en carácter en cambio el dinámico muestra palabras de acuerdo a lo que se va escribiendo, en sus resultados muestra que el dinámica mejora en 11% (Huhn *et al.*, 2019).

Hubo propuestas de modificación de teclado QWERTY así como su reorganización en el teclado QWERTY agrupando las letras más utilizadas en el centro del teclado (Rowell, 1909). Otra modificación propuesta que presentó un teclado combinacional, afirmando que su teclado se basó en el resultado de siete millones de escrutinios de varios estilos de prosa inglesa y el resultado final fue un diseño basado en los 39 digramas más frecuentes (Nelson, 1920).

1.2. Antecedentes

1.2.1. Modificaciones y propuestas al teclado físico

Las revisiones de literatura científica del teclado desde sus inicios, así como los cambios, comparaciones y nuevas propuestas hasta 1983 acerca teclado fueron analizadas en el trabajo de Noyes (1983) “The QWERTY keyboard: a review” algunos de estos análisis se han tomado como referencia y antecedentes porque hay varias propuestas de modificación sobre el teclado QWERTY.

También Wolcott (1920) presentó un diseño del teclado se basó en el mismo principio de Benaji (1920) lo ubicó las letras menos utilizadas (J, Q, Z y X) debajo de los dedos índices.

La propuesta diferente que resalta bastante es de Dvorak y Dealey (1936) fue el más serio y prologando que rechazó al teclado QWERTY, presentó su diseño sobre investigación científica relacionados a la frecuencia de uso de diferentes teclas en secuencias de dos, tres, cuatro y cinco letras. El objetivo de Dvorak era colocar las teclas en cuatro filas y 42 teclas, superando a QWERTY de acuerdo con su investigación, y tenía los siguientes criterios:

- La mano derecha recibió más trabajos con 46% de carga
- La asignación de las teclas a los diferentes dedos fue realizada proporcional a su habilidad y fuerza de cada dedo, ejemplo la letra E fue asignado bajo los dedos fuertes.



Figura 9. The Dvorak Simplified Keyboard (1936)

- El 70% de las teclas están ubicadas en la fila de inicio con las letras (A, O, E, U, I, D, H, T, N, S), 22% de teclas fueron asignadas fila delantera y 8% de las teclas fueron asignadas en la fila trasera.
- Los alcances difíciles e incómodos desde la fila de inicio se redujeron al mínimo por que habían sido asignados las teclas menos utilizadas en estas posiciones.
- 35% de todas las palabras probadas con el idioma ingles fueron escritas exclusivamente en la fila de inicio, esto equivale a 3000 palabras.

El diseño del teclado Dvorak fue objeto de una intensa investigación por parte de la Marina de los Estados Unidos en 1944 y por la Oficina de Correos de Australia 1953, concluyeron el teclado Dvorak sí daría lugar a un aumento sustancial en la eficiencia de los mecanógrafos. En 1939, la Universidad de Chicago informó que el teclado Dvorak se ha empleó con éxito como parte de un programa de escritura en el teclado enseñando a los niños de la escuela, los maestros informaron que los

niños estaban aprendiendo a escribir dos veces más rápido en el teclado Dvorak y que podían superar las 50 palabras correctas por minuto.

La oficina Nacional de Estándares de los EE. UU. llevó a cabo una investigación del teclado Dvorak y examinó las evaluaciones realizadas en 1930 en adelante por las agencias y universidades de los Estados Unidos (Phillips, 1968). Aunque no se publicaron pruebas experimentales concretas, pero este informe concluyó que el D.S.K. era superior al teclado QWERTY.

Ya actualmente, la incorporación de la tecnología ha traído consigo el uso de los dispositivos móviles, cada vez más personas utilizan los teléfonos inteligentes para poder enviar mensaje y acceder al correo electrónico.

En otra investigación Zhai y Kristensson (2008) evaluaron varios diseños de teclado, incluyendo el teclado de orden alfabético optimizado (ATOMIK), QWERTY, y QWERTY entrelazado para la escritura por gestos. El autor realiza dos estudios, el primero consiste en estudios empíricos para la facilidad de búsqueda de letras de varios diseños y el segundo estudio consiste en realizar análisis matemático para maximizar la tolerancia a la imprecisión, diseñaron su investigación para dos casos:

Caso 1: Imprimieron tres diseños en papel (ALFABÉTICO, ATOMIK y QWERTY), se pide el apoyo de 12 voluntarios (todos usuarios diarios del teclado QWERTY), localizar visualmente 19 palabras preestablecidas durante un tiempo cronometrado. Se calcula la distancia entre dos trazos de la escritura por gestos utilizando el algoritmo de canal de localización no ponderado para cuatro diseños de teclado (QWERTY, iQWERTY, ATOMIK y ALFABETICO). Se obtuvo los siguientes resultados, ATOMIK=83s, ALFABETICO = 70s, QWERTY=47s. Se obtuvo los siguientes resultados, QWERTY tenía los trazos de palabras más idénticos e iQWERTY el menos.

Caso 2: Propusieron el teclado (iQWERTY), es una variante del teclado QWERTY manteniendo las posiciones relativas de las teclas originales. Posteriormente se realiza estudios de ergonomía, se imprimieron tres diseños en papel (ALFABETICO, QWERTY y iQWERTY), se pidieron el apoyo de 12 nuevos voluntarios (todos usuarios diarios del teclado QWERTY), localizar visualmente

19 palabras preestablecidas durante un tiempo cronometrado. Utilizaron el mismo método que en el primer estudio computacional, pero variando el umbral de 0 a 0.5. Obtuvieron los siguientes resultados, ALFABETICO=80.5s, iQWERTY=68.3s y QWERTY=60.6s. Concluyen que iQWERTY redujo los pares de palabras en conflicto en un 16% para el tamaño más grande de léxico.

En el trabajo de Ciobanu *et al.* (2016) compararon alternativas de teclados, diferentes alternativas fueron diseñados pensando en la ergonomía:

Kalq.- Es un teclado virtual dividida en dos, permite mayor velocidad sin calidad ergonómica, basado en disposición aleatoria.

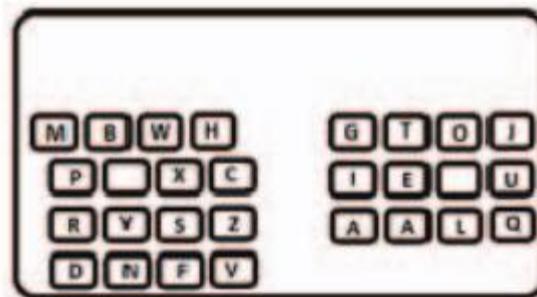


Figura 10. Teclado Komark

- Swype es un teclado para tables smartphones, es predictivo ingresando las palabras con arrastre en el touch.
- El teclado Keybowl utiliza dos botones para escribir letras y permite la formación de combinaciones de posiciones radiales de los botones. El teclado fue diseñado para eliminar el movimiento excesivo de los dedos, evitar el movimiento de la muñeca y mejorar la biomecánica de la mano.

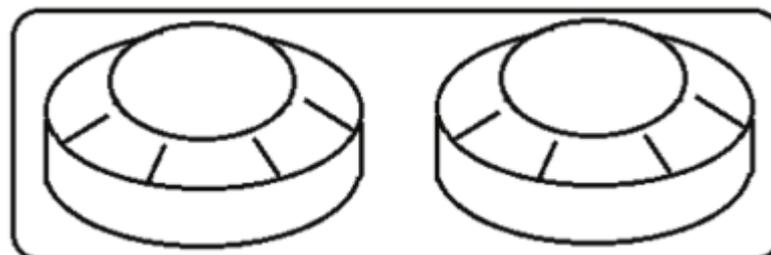


Figura 11. Teclado Keybowl

- Datahand keyboard.- fue introducido en 1995, separado en dos módulos, Cada dedo presiona cinco botones, cuatro en direcciones perpendiculares y uno central. El pulgar tiene cinco botones para presionar, uno adentro y dos afuera y un botón arriba y abajo. Los módulos son ajustables para adaptarse a los dedos del usuario y cada módulo se puede mover en cualquier dirección.

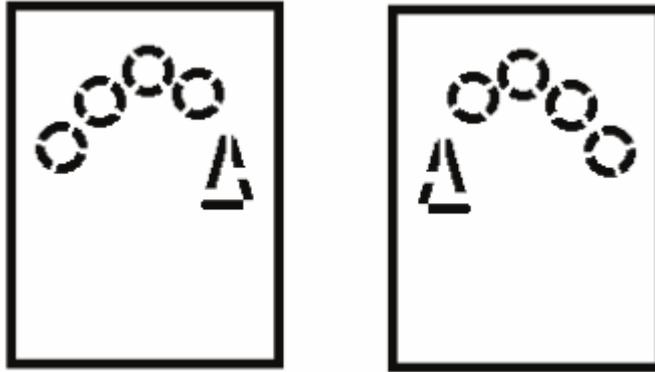


Figura 12. Teclado Datahand

De lo anterior llegaron a las siguientes conclusiones:

- Según varias investigaciones QWERTY son las menos ergonómicas y las teclas no están agrupadas lógicamente y su rendimiento puede ser mejorado.
- Hasta ahora se hicieron mejoras pero aun es necesario seguir investigando.
- Incluso los estándares Human Factors Design Standard como HF-STD-001 y otros, fallan en hacer recomendaciones ergonómicas porque están basadas en diseños QWERTY esto implica que todas las normas deben ser revisadas.
- Las teclas deben ser fáciles de encontrar y de aprender.
- El diseño debe provocar menos errores al escribir.
- El diseño debe permitir mantener la postura ergonómica de los dedos y articulares del cuerpo.

En la investigación Yang y Mali (2016) emplearon los diseños de 3x10, 4x7 y 5x6, para probar las distancias de recorrido tomaron más 233000 palabras de American National corpus (ANC), que es un corpus de código abierto en inglés y mediante la

técnica simulated annealing, buscaron mejorar los diseños de los teclados . La función objetiva de este procedimiento de mejora se calcula mediante la siguiente ecuación:

$$d = \sum_{i=1}^{26} \sum_{j=1}^{26} P_{ij} D_{ij}$$

D_{ij} es la distancia entre i e j en el diseño, P_{ij} es la probabilidad de aparición de digramas ij . D es la suma ponderada de las distancias de recorrido de los dedos para todos digramas. La distancia entre las teclas i y j es la distancia euclidiana entre las esquinas inferior izquierda de estas teclas. P_{ij} se encuentra al dividir el número total de apariciones de digramas ij en todas las palabras de la colección de palabras u oraciones elegidas por el número total de apariciones de palabras u oraciones.

Para esta investigación tomaron más 233000 palabras de American National corpus (ANC), que es un corpus de código abierto en inglés. Para el diseño inicial utilizaron 1916 anchos clave, luego colocaron en el centro del teclado la letra E la cual se repetía más veces. Según la posición de E procedieron a colocar las letras en digramas de alta frecuencia, incluido E, ER,EN,ES,EO,EA,ET,EH,y EI, colocaron las letras en estos digramas adyacentes a E, se repitió el proceso hasta colocar las 26 letras, sin embargo esto no resultó siendo un buen diseño. Teniendo este diseño como punto de partida se ejecutó el algoritmo 1 para 20000 iteraciones, de esta manera el algoritmo condujo a un mejor diseño donde el valor de $d=1.795$ anchos de tecla. Se fue probando el algoritmo con los demás diseños logrando así el mejor valor de d que es de 1.93 para el diseño 3x10, 1.83 para el diseño 4x7 y 1.78 para el diseño 5x6. Todos estos valores son al menos 40% más bajos que el QWERTY que tiene como 3.31 anchos clave.

Por resolver el problema posiciones de las teclas del teclado de la computadora, proponen una función de evaluación teniendo en cuenta criterios ergonómicos, utilizaron el algoritmo Ant Colony Algorithm, como resultado tuvieron un nuevo orden teclado para el idioma inglés (Eggers *et al.*, 2003).

1.2.2. Modificaciones sobre teclados virtuales (Virtual Keyboards)

Los primeros teléfonos móviles se fabricaron por 1980s llamados teléfonos celulares, principalmente estos teléfonos fueron diseñados para hablar, básicamente para ingresar números y textos de mensajes, su teclado de 12 teclas fue adaptado del teclado del teléfono de ese tiempo (Silfverberg, 2007), las entradas de texto se adaptaron en diferentes teléfonos celulares en diferentes idiomas y al menos tenían 26 caracteres.

Hubo varias propuestas de Virtual Keyboard así como de Scott y Zhang (1999), proponen un Virtual keyboard OPTI y comparan con QWERTY, con la participación de 5 usuarios en escribir durante 20 a 40 minutos, muestran las mejoras de su propuesta en este trabajo.

Enfocado en la velocidad de escritura en el smarthpones Dell'Amico *et al.* (2009), analizan desde óptica de optimización y proponen una nueva distribución de teclado para diferentes idiomas, para este nuevo teclado recomiendan escribir con un solo dedo o con lápiz óptico.

Para mejorar la velocidad de escritura con el teclado QWERTY Leung y Aarabi (2014) proponen un teclado circular para dispositivos móviles. Diseñaron dos teclados virtuales Wu *et al.* (2014) entrenaron participantes para observar que permite escribir más rápido que con el teclado QWERTY. También proponen un diseño de teclado chino Niu *et al.* (2014) con posiciones generados mediante algoritmo. También presentan un nuevo diseño de teclado virtual Pritom *et al.* (2016) con 10 botones funcionales y 5 botones principales para generar 27 letras de QWERTY.

En el Virtual Keyboard generalmente hay errores en el ingreso de textos, razones varias como hacinamiento entre teclas no permiten ingresar fácilmente textos, en ese marco los investigadores AbuHmed *et al.* (2015), presentan un teclado UOIT que está basado en escribir palabras como dibujando, solo pusieron 13 teclas grandes que reemplazan a 26 teclas normales, en su validación muestran un promedio de 11.3 palabras por minuto en la escritura, con error de escritura 3.8%.

Entre los críticos de QWERTY también está la investigación Lawrence y Aarabi (2014), dicen que QWERTY es rectangular y dificultoso en el ingreso de textos en los smarthphones, y en un tamaño menor no se puede acomodar toda las teclas, por eso proponen un teclado circular CAK, para operación de una sola mano usando el pulgar.

Continuando con los Smarwatches en su limitación de tamaño para usar un Virtual Keyboard en la investigación Cha *et al.* (2015), proponen un teclado aumentado (zoom) deslizante, el teclado sigue siendo QWERTY, probaron con 20 participantes con una combinación de 5 tamaños diferentes como resultado lograron escribir un promedio de 11.9 palabras por minuto, superando a otros citados en su investigación.

Una propuesta para los teclados en miniatura para dispositivos móviles, desarrollaron cuatro diseños de teclados distintos al teclado convencional Qwerty, evaluaron la velocidad de entrada, precisión, comodidad, simpatía y facilidad de aprendizaje, como resultado final el rendimiento de los teclados pequeños era peor que el teclado convencial (Hsiao *et al.*, 2014).

Los smarthphones ya es bastate usado a nivel mundial, más aún actualmente muchos usan Smartwatches estas también tienen un Virtual Keyboard, pero que no parece ser tan funcional con su teclado, motivado por ello investigadores Darbar *et al.* (2016), proponen un nuevo teclado para smartwatches que permite ingresar alfabetos más frecuentes en un solo toque, otras teclas símbolos y números pueden ser ingresadas en doble toque, muestran resultados de promedio de palabras ingresar por minuto.

En la investigación de Kim *et al.* (2019), analizaron que en los smarthphones hay más frecuencia de movimientos entre el teclado virtual y las áreas de textos, por lo que diseñaron dos tipos de teclados virtuales el estático y dinámico; el estático muestra de carácter en carácter en cambio el dinámico muestra palabras de acuerdo a lo que se va escribiendo, en sus resultados muestra que el dinámica mejora en 11%.

CAPÍTULO II

PLANTEAMIENTO DEL PROBLEMA

2.1. Identificación del problema

La máquina de escribir patentado en 1868 por Sholes *et al.* (1868) es el originario de diseño del teclado QWERTY, teclado orientado para el idioma inglés, fue diseñado para “cazar y picotear” y no para teclear como ahora la tenemos, aunque el objetivo de los inventores era diseñar una máquina que pudiera imprimir palabras a una velocidad equivalente a la escritura a mano, la patente fue cedida a otros y fueron modificados. La primera patente que muestra el diseño del teclado QWERTY fue presentado por (Beeching, 1894), en el mismo año la empresa Remington Arms Company comenzó a fabricar la máquina de escribir en cantidad.

El teclado QWERTY posteriormente se impuso, logrando que el diseño del teclado llegase a ser reconocido como un estándar internacional como se ve en la Figura.



Figura 13. The standard QWERTY layout, American National Standard Typewriter Keyboards)

Posteriormente Phillips (1968) explica cómo habría sido diseñado las posiciones de las teclas de QWERTY.

En los años 1930s el teclado estándar QWERTY fue fuertemente cuestionada. Dvorak y un equipo de ingenieros Dvorak y Dealey (1936) probaron 250 variaciones de teclados, concluyeron que el diseño QWERTY era uno de los peores para la escritura en cuanto a su distribución algunas pruebas que muestran en su investigación:

- Hay 57% de sobrecarga de trabajo para la mano izquierda la cual no es la más usada para la población en general.
- Sobrecarga de trabajo en ciertos dedos como los dedos pequeños que tienen que pulsar las teclas de cambio (Shift), la tecla mayúscula, la tecla de borrado (back space)
- QWERTY está diseñada en tres filas, la primera fila es las teclas que llevan el nombre QWERTY, la segunda fila inicia con las teclas ASD la tercera y la última fila inicia con ZXC, en ese sentido verificaron que las teclas de la primera se usan 32% al escribir, la segunda fila 16% y la última fila 52% no siendo equilibrado según fundamentaron.
- Se requieren saltos frecuentemente de la primera fila a la última fila como para las combinaciones “br, un, in” etc.

En otra investigación Ward (1936), demostró que muchas palabras comunes se escriben con la mano izquierda, hicieron una prueba con 3000 palabras de “Find and Wagnall’s Collegiate Dictionary” de los cuales sólo 300 palabras fueron escritas con la mano derecha y 2700 palabras con la mano izquierda.

También por su parte Ferguson y Duncan (1974) demostraron que la carga de las teclas de QWERTY se distribuye de forma aleatoria e inapropiada con excepciones de Z y X, recomiendan que la carga debería disminuir naturalmente de acuerdo a fuerza de los dedos, iniciando con el dedo índice hasta el dedo meñique de forma decreciente.

Por otro lado, han demostrado que los diagramas que requieren el uso de dos manos son más rápidos que realizar con los diferentes dedos de una mano.

Las características del teclado QWERTY su separación hace líneas verticales haciendo un diagonal de izquierda a derecha, en las primeras máquinas de escribir esto habría sido necesario espaciar las teclas y ayudar a los usuarios en la facilidad de ubicación, no obstante, esta forma de organización de las posiciones de las teclas fue estudiado por Martin (1972) verificó que provocaba que la letra siguiente de una tecla es obstruida por

la mano, como consecuencia verificó que se redujo la velocidad de pulsar la siguiente tecla.

Para ver el lado bueno de QWERTY analizaron los tiempos de digitación, en su investigación Kinhead (1975) cronometró y analizó 115,000 pulsaciones de 22 digitadores para determinar las diferencias en el tiempo de digitación entre manos, filas, columnas y teclas individuales, probando digitar a la máxima velocidad utilizando QWERTY descubrió que llegaría a pulsaciones más rápidas en 8% frente otro teclado imaginario con diseño óptimo. Sin embargo, en ese entonces los autores analizaron que la diferencia es infructuosa considerando que el teclado QWERTY se impone y es omnipresente dando respaldo a que dado el tiempo y motivación un individuo puede convertirse en experto con cualquier dispositivo diseñado de manera ineficiente dentro de un marco de la razón.

La imprecisión de teclado QWERTY en la escritura por gestos en los teléfonos móviles con pantalla táctil y tabletas con lápiz fue estudiada por Zhai y Kristensson (2008), analizaron que para dar forma a un texto escrito, el usuario forma patrones de las palabras deslizando un dedo o un lápiz sobre un teclado gráfico, el arrastre generado por el usuario es reconocido por el identificador de patrones que en muchos casos es impreciso. Con respecto a la escritura por gestos, el autor considera que para la facilidad de uso inicial de los usuarios, las letras deben ser fáciles de localizar visualmente, pero para el uso a largo plazo y óptimo el diseño debe maximizar la tolerancia a la imprecisión y la flexibilidad de escritura de todas las palabras. El uso a largo, el diseño debe maximizar la tolerancia a la impresión y la flexibilidad de escritura de todas las palabras.

Muchos dispositivos se diseñan para mejorar el confort y para incrementar la productividad, también esos dispositivos pueden ayudar a evitar el dolor y lesiones específicas Ciobanu *et al.* (2016), para diseñar se requiere expertis, gran parte de digitadores lectos buscan visualmente las teclas de forma aleatoria, los teclados QWERTY siguen siendo menos ergonómicos, en esta investigación se discute la existencia de estándares de ergonomía y diversos teclados comerciales, observando que hay recomendaciones de normas equivocadas de acuerdo al estándar de keyboard ISO 9241-410.

En la investigación Yang y Mali (2016) han analizado que lo menos se envían 110 mensajes de texto por día promedio. El uso del teclado QWERTY es ineficiente para el ingreso de texto en dispositivos móviles. Cuando se escribe un texto, el dedo pulgar debe viajar de un lado a otro varias veces para ingresar una palabra. La larga distancia del recorrido del dedo es perjudicial para la experiencia del ingreso de texto, originando una baja velocidad de entrada, además de como el usuario sostiene el teléfono con una mano y escribe el texto con el pulgar de la misma mano; el hecho de mover el pulgar hacia adelante y atrás entre los lados de un teclado es desagradable debido al rango limitado de movimiento de pulgar, además demasiado tiempo enviando mensajes de texto puede generar dolores en el cuello y hombro.

Otras propuestas en cuanto al teclado virtual, sabiendo que el tamaño del teclado es bastante limitado proponen teclados virtuales ambiguos, pero en la investigación Huang y Wu (2015), analizaron que estos teclados virtuales ambiguos toman más tiempo en búsqueda de las teclas necesarias para el usuario y no están enfocados a la necesidad del usuario. Por tanto, no sería recomendable resolver el problema de teclado virtual con teclados ambiguos.

2.2. Enunciados del problema

2.2.1. Problema general

¿El modelo de optimización basado en un algoritmo de inteligencia artificial distribuye las posiciones de las teclas del teclado virtual?

2.2.2. Problemas específicos

- ¿El modelo de optimización basado en un algoritmo de inteligencia artificial Simulated Annealing distribuye las posiciones de las teclas del teclado virtual?
- ¿Un conjunto de miles de palabras permite validar el modelo de optimización de distribución de teclas del teclado virtual?
- ¿La nueva distribución de las teclas del teclado virtual sirve para diseñar mockups de un nuevo teclado virtual para dispositivos móviles denominado Ray-Keyboard?

2.3. Justificación

Los diferentes modelos de teclados virtuales están basados en la distribución QWERTY, este modelo que no ha cambiado más de 132 años, al no adaptarse con los avances y modernidad de las tecnologías existentes es necesario innovar un nuevo modelo de distribución de las teclas que asigne óptimamente las ubicaciones en el teclado virtual.

En algunos modelos de teclado virtual de los dispositivos móviles es posible usar Swype (arrastré o deslizamiento) donde genera una palabra arrastrando y pasando sobre cada letra, la cual permite generar una medida de distancia de un punto a otro punto, en ese sentido es comparable la distancia que genera el teclado virtual Qwerty y el teclado virtual Ray, es importante esta investigación por que intentamos disminuir la distancia de deslizamiento en la escritura de palabras con la propuesta nuevo teclado virtual RAY.

Todos usamos los dispositivos móviles para mensajes en redes sociales o textos formales, por tanto, generamos o escribimos cientos de palabras o miles de palabras cada día, es así que llevamos tiempo en escribir las palabras durante el día, la propuesta de la investigación en tener un nuevo teclado virtual optimizado la distribución de sus teclas, permite ahorrar tiempo en la escritura aun sea bastante mínimo en una palabra, al generar miles de palabras en un día nos beneficia en usar menos tiempo en escribir para los usuarios de dispositivos móviles.

Un teclado virtual orientado a generar palabras basado en el deslizamiento, permite usar un enfoque de optimización en distancias, todas las teclas del teclado virtual son posibles generar basado en deslizamiento incluidos el teclado numérico donde se genera los números arrastrando por cada número, esto permite cambiar de enfoque para el usuario pensar todo en arrastrar o deslizar, es importante resaltar que los dispositivos móviles cada día van mejorando la captura de touch en ese sentido en futuro no será necesario picotear o presionar teclas específicas.

La ergonomía es importante, apoyar a que el usuario no aplique la fuerza en cada tecla sino un trabajo sencillo de deslizamiento para generar las palabras, aún más, un teclado virtual cuya distribución de teclas están optimizadas permite al usuario deslizar menos distancia para generar las palabras, de hecho, esta comodidad es parte de la ergonomía.

2.4. Objetivos

2.4.1. Objetivo general

Diseñar un modelo de optimización de distribución de teclas del teclado virtual utilizando algoritmos de inteligencia artificial.

2.4.2. Objetivos específicos

- Diseñar el modelo de optimización de distribución de teclas del teclado virtual utilizando el algoritmo de Inteligencia Artificial Simulated Annealing.
- Validar el modelo de optimización de distribución de teclas del teclado virtual utilizando miles de palabras.
- Diseñar mockups de un nuevo teclado virtual denominado Ray-Keyboard para dispositivos móviles basado en la distribución de teclas optimizadas.

2.5. Hipótesis

2.5.1. Hipótesis general

El modelo de optimización basado en un algoritmo de inteligencia artificial permite distribuir las posiciones de las teclas del teclado virtual.

2.5.2. Hipótesis específicas

- El modelo de optimización basado en un algoritmo de inteligencia artificial Simulated Annealing permite distribuir las posiciones de las teclas del teclado virtual
- El conjunto de miles de palabras permite validar el modelo de optimización de distribución de teclas del teclado virtual.
- La nueva distribución de las teclas del teclado virtual permite diseñar mockups de un nuevo teclado virtual para dispositivos móviles denominado Ray-Keyboard.

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1. Lugar de estudio

El presente estudio se realizó en la Ciudad de Lima Metropolitana, se ha analizado los diferentes teclados virtuales para móviles Android tales como: Gboard de Google, Teclado KK Emoji, Neon Dream Emoji Keyboard, SwiftKey entre otros.

La investigación se ha realizado en diferentes campos para lograr el objetivo:

- El mercado ofertante de teclados físicos para analizar los teclados más usados actualmente.
- El campo de programación de aplicaciones de teclados virtuales, para impulsar un nuevo teclado virtual.
- Literatura científica para analizar la historia de la trayectoria de cambios en el teclado físico y en el teclado virtual.

3.2. Población

La población está constituida por 27 teclas del teclado de las computadoras



Figura 14. Teclado español de las computadoras

3.3. Muestra

Para la muestra se ha considerado la misma cantidad de la población, considerando que el objetivo es distribuir las 27 teclas del teclado de las computadoras. En la muestra no incluye las teclas numéricas ni las otras teclas adicionales.

Tabla 1

Conjunto de teclas para la muestra

ID	Tecla
1	Q
2	W
3	E
4	R
5	T
6	Y
7	U
8	I
9	O
10	P
11	A
12	S
13	D
14	F
15	G
16	H
17	J
18	K
19	L
20	Ñ
21	Z
22	X
23	C
24	V
25	B
26	N
27	M

3.4. Método de investigación

La investigación es cuantitativa cuasi experimental, tomando las 27 teclas del teclado se ha sometido un conjunto grande de palabras para simular la escritura, midiendo la

distancia por cada intercambio entre dos teclas, la investigación se ha realizado por etapas:

- Recolección de textos diferentes las cuales se ha convertido en texto plano .txt, para ello hemos construido una herramienta que permite especificar URLs o archivos (html, pdf), los documentos son descargados y convertidos en texto .txt. Estos textos son entradas para el modelo de optimización expresadas como conjuntos de palabras.
- Se ha diseñado un modelo de optimización de distribución de teclas utilizando el algoritmo de inteligencia artificial Simulated Annealing.
- Diseño de estado inicial, se ha considerado el teclado QWERTY en el plano cartesiano como ubicaciones de puntos cada tecla, la cual es una solución del modelo, a partir de ello se optimiza con el algoritmo.
- Aplicación del algoritmo Simulated Annealing para intercambiar teclas y optimizar la distribución del mismo.
- Interpretación de resultados
- Diseño del nuevo teclado virtual denominado Ray Keyboard.

3.5. Descripción detallada de métodos por objetivos específicos

3.5.1. OE1: Metodología de diseño del modelo de optimización de distribución de teclas del teclado virtual utilizando el algoritmo de Inteligencia Artificial Simulated Annealing

Actualmente varios modelos de teclados virtuales QWERTY existentes para dispositivos móviles tienen opción de deslizamiento para escribir, esta forma de escribir es deslizar de una letra a otra letra continuados hasta formar una palabra, así como podemos ver en la siguiente figura.



Figura 15. Opción de Swype (deslizamiento) de los teclados

La forma de escribir por deslizamiento nos lleva pensar que mientras se desliza por diferentes letras es posible tener la distancia de deslizamiento, suponiendo que la distancia entre cada letra fuera 1 (uno), para una palabra ejemplo “Universidad” la distancia es aproximadamente 28 en el teclado virtual QWERTY, dependiendo si el deslizamiento fuera más o menos preciso pasando por las letras correspondientes.

Formada una frase o conjunto de palabras, sumamos las distancias generadas al pasar por cada letra, si escribimos un documento completo como: una tesis, un artículo, una historia en facebook, un libro etc, siempre se podrá sumar la distancia total de la escritura del documento, cada día, en cada instante usamos este recurso costo de la escritura utilizando el teclado móvil QWERTY. Por lo que presentamos en esta investigación un nuevo modelo de optimización de distribución de la ubicación de las teclas para el teclado virtual RAY.

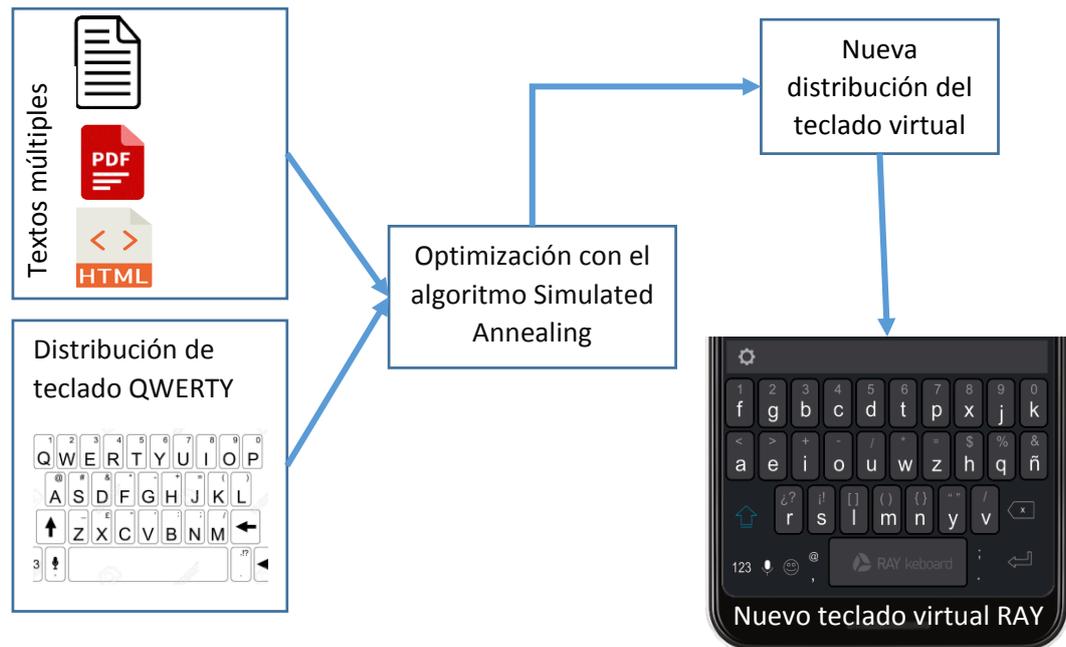


Figura 16. Modelo propuesto de optimización de distribución de las teclas

Para el modelo de optimización de distribución de las teclas del teclado virtual RAY se ha considerado dos entradas:

- Conjunto de textos múltiples (libros) para trazar la ruta de deslizamiento que es una sumatoria de distancias de miles de palabras.
- El estado inicial del algoritmo de optimización fue considerado la distribución de teclado Qwerty, adaptados y ubicados en el plano cartesiano

El algoritmo de optimización calcula la distancia inicial tal conforme con la ubicación inicial Qwerty, en su proceso utiliza intercambios calculando constantemente la distancia con el conjunto grande de palabras, si hay cambios convenientes que permiten disminuir la distancia se van registrando, como final es la nueva distribución de las teclas llamado RAY Keyboard.

También construimos el teclado virtual para dispositivos Android cuya distribución de las ubicaciones es la salida del algoritmo de optimización, este teclado llamado RAY Keyboard es un teclado basado en Swype basado en deslizamiento.

a) Recolección de datos para la prueba

Para el modelo de optimización de distribución de teclas del teclado virtual fue necesario un conjunto grande palabras, en este caso consideramos varios libros que tiene miles de palabras.

Para recolectar esos libros y otros textos fue necesario construir una herramienta GetTxt que consiste en un formulario de ingreso que permite considerar la categoría del texto, la url y el nombre del texto, de fondo utilizamos algoritmos de conversión de tipos hacia txt.

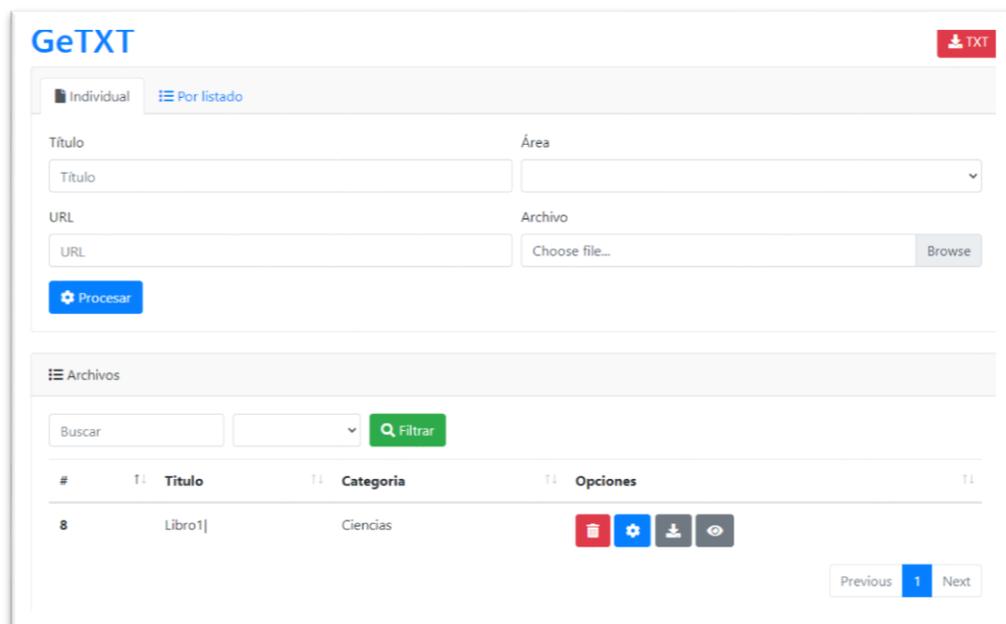


Figura 17. Sistema de recolección de textos con palabras

Los textos recolectados se consideran como entradas para el modelo de optimización, estos textos son documentos libros que pueden estar en PDF o en html, las cuales se convierten en formato .txt.

Título.- Es campo de texto para ingresar un titulo de texto a recolectar

Área.- Campo para especificar el área a la cual pertenece el texto a recolectar (Humanidades, Matemáticas, Ciencias)

Url.- Es la dirección de internet del documento a recolectar.

Archivo.- Para la posibilidad de alimentar un archivo .pdf o .html para convertir en un documento .txt.

Boton Procesar.- Convierte los documentos online a archivos .txt

Lista de documentos. - Es el conjunto de recolección de datos preparados para su conversión.

Botón TXT. - Permite descargar el texto convertido a .txt.

Adicionalmente se puede proporcionar una lista de documentos con las mismas características para para que sea convertido a .txt.

b) Distribución del teclado Qwerty como entrada inicial en el modelo de optimización.

Los algoritmos de optimización que utilizan heurísticas requieren de un estado inicial para realizar el trabajo de optimización, es un esquema de entrada de acuerdo al problema, naturalmente es una solución dentro de conjunto de soluciones del problema.

En esta investigación utilizamos el algoritmo Simulated Annealing como se fundamenta en los siguientes ítems, este algoritmo de optimización requiere de un estado inicial para optimizar la distribución del teclado, para ello consideramos como configuración inicial la distribución del teclado Qwerty.



Figura 18. Teclado Qwerty como configuración inicial para el algoritmo de optimización

Ubicando el teclado en el plano cartesiano para considerar como problema de ruta, conveniente para usar el algoritmo de optimización de hallar la mejor ruta considerando la distancia como heurística.



Figura 19. Teclado Qwerty en el plano cartesiano

Se levanta información de las posiciones de cada tecla ubicado en el plano cartesiano, el área de una tecla **t** cualquiera se considera como un punto en el plano cartesiano.

Considerando que la idea principal es calcular la distancia entre dos puntos del plano, en este caso cada punto representa una tecla (letra), de esta manera se considera un problema de ruta cuando pasamos por varios puntos (palabra), la suma de las distancias generadas al escribir varias palabras es la distancia total generada al escribir un documento.

Para colocar en el algoritmo se ha etiquetado cada tecla del plano cartesiano con ID de 1 has 27, agregando la tecla que corresponde, además las posiciones X y Y de cada tecla.

Tabla 2

Posiciones de la distribución del teclado Qwerty

ID	Tecla	Pos X	Pos Y
1	q	1	3
2	w	2	3
3	e	3	3
4	r	4	3
5	t	5	3
6	y	6	3
7	u	7	3
8	i	8	3
9	o	9	3
10	p	10	3
11	a	1	2
12	s	2	2
13	d	3	2
14	f	4	2
15	g	5	2
16	h	6	2
17	j	7	2
18	k	8	2
19	l	9	2
20	ñ	10	2
21	z	2	1
22	x	3	1
23	c	4	1
24	v	5	1
25	b	6	1
26	n	7	1
27	m	8	1

La información de entrada considerada como estado inicial del modelo de distribución como puntos de entrada (x,y).

c) Optimización de la distribución de teclas con el algoritmo de inteligencia artificial Simulated Annealing

En este estudio adaptamos las posiciones de cada tecla como un punto en las coordenadas en el espacio del teclado, luego aplicamos el algoritmo Simulated Annealing para optimizar las distancias, logramos reubicar sus posiciones de

las teclas buscando la distancia menor en la escritura con millones de letras y palabras.

El algoritmo Simulated Annealing es similar al algoritmo Hill Climbing pero estocástica, inspirado en el proceso físico de enfriamiento controlado de metales, si el enfriamiento es adecuado, se obtiene la estructura de menor energía (mínimo global o máximo global).

El algoritmo de optimización Simulated Annealing fue usado para resolver problemas de combinatoria para buscar una solución óptimo global, que recorre los espacios de estados con cambios aleatorios.

En esta investigación utilizamos este algoritmo Simulated Annealing para reubicar las posiciones de cada tecla del teclado virtual, buscando la menor distancia en el deslizamiento en la escritura de textos en el móvil.

```
Function SIMULATED-ANNEALING(Problem, Schedule)  
returns: a solution state.  
Inputs: Problem, Schedule.  
Local Variables: Current, Next, T.  
  
Current = MAKE-NODE(INITIAL-STATE[Problem])  
For t = 1 to  $\infty$  do  
    T = Schedule[t]  
    If T = 0 then return Current  
    Next = a randomly selected successor of Current  
     $\Delta E$  = VALUE[Next] – VALUE[Current]  
    if  $\Delta E > 0$  then Current = Next  
    else Current = Next only with probability  $e^{(-\Delta E/T)}$ 
```

Figura 20. Algoritmo Simulated Annealing

La función de aceptación es la heurística, utiliza básicamente temperatura y una función de probabilidad de Boltzmann, trabajando con dos estados.

$$P = e^{-\frac{\Delta E}{T}}$$

Donde:

ΔE = Es la diferencia entre dos estados, consideramos como un estado la distancia de recorrido entre las teclas dado un conjunto de palabras.

T = Temperatura, inicializamos con valor 10,000 que permite descender por cada iteración consiguiendo el enfriamiento de acuerdo al modelo.

1. Estado inicial

El estado inicial como punto de inicio de la optimización es la entrada necesaria para usar el algoritmo, para ello se ha considerado como estado inicial al conjunto de ubicaciones del teclado Qwerty diseñados en el plano cartesiano, mostramos los detalles de la ubicación en el punto 3.2.2.

2. Distancias entre las teclas

El algoritmo Simulated Annealing utiliza como función de evaluación la distancia de los estados, un estado es una ruta en este caso es la trayectoria de deslizamiento en las teclas para generar una palabra o conjunto de palabras, para ello se ha utilizado la distancia euclidiana tabla 3 entre un par de teclas ubicadas en el plano cartesiano.

Tabla 3

Distancias entre los pares de teclas ubicadas en el plano cartesiano

	q	w	e	R	T	y	u	i	o	p	a	s	d	f	g	h	j	k	l	ñ	z	x	c	v	b	n	m
q	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	1.0	1.4	2.2	3.2	4.1	5.1	6.1	7.1	8.1	9.1	2.2	2.8	3.6	4.5	5.4	6.3	7.3
w	1.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	1.4	1.0	1.4	2.2	3.2	4.1	5.1	6.1	7.1	8.1	2.0	2.2	2.8	3.6	4.5	5.4	6.3
e	2.0	1.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	2.2	1.4	1.0	1.4	2.2	3.2	4.1	5.1	6.1	7.1	2.2	2.0	2.2	2.8	3.6	4.5	5.4
r	3.0	2.0	1.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	3.2	2.2	1.4	1.0	1.4	2.2	3.2	4.1	5.1	6.1	2.8	2.2	2.0	2.2	2.8	3.6	4.5
t	4.0	3.0	2.0	1.0	0.0	1.0	2.0	3.0	4.0	5.0	4.1	3.2	2.2	1.4	1.0	1.4	2.2	3.2	4.1	5.1	3.6	2.8	2.2	2.0	2.2	2.8	3.6
y	5.0	4.0	3.0	2.0	1.0	0.0	1.0	2.0	3.0	4.0	5.1	4.1	3.2	2.2	1.4	1.0	1.4	2.2	3.2	4.1	4.5	3.6	2.8	2.2	2.0	2.2	2.8
u	6.0	5.0	4.0	3.0	2.0	1.0	0.0	1.0	2.0	3.0	6.1	5.1	4.1	3.2	2.2	1.4	1.0	1.4	2.2	3.2	5.4	4.5	3.6	2.8	2.2	#	2.2
i	7.0	6.0	5.0	4.0	3.0	2.0	1.0	0.0	1.0	2.0	7.1	6.1	5.1	4.1	3.2	2.2	1.4	1.0	1.4	2.2	6.3	5.4	4.5	3.6	2.8	2.2	2.0
o	8.0	7.0	6.0	5.0	4.0	3.0	2.0	1.0	0.0	1.0	8.1	7.1	6.1	5.1	4.1	3.2	2.2	1.4	1.0	1.4	7.3	6.3	5.4	4.5	3.6	2.8	2.2
p	9.0	8.0	7.0	6.0	5.0	4.0	3.0	2.0	1.0	0.0	9.1	8.1	7.1	6.1	5.1	4.1	3.2	2.2	1.4	1.0	8.2	7.3	6.3	5.4	4.5	3.6	2.8
a	1.0	1.4	2.2	3.2	4.1	5.1	6.1	7.1	8.1	#	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	1.4	2.2	3.2	4.1	5.1	6.1	7.1
s	1.4	1.0	1.4	2.2	3.2	4.1	5.1	6.1	7.1	8.1	1.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	1.0	1.4	2.2	3.2	4.1	5.1	6.1
d	2.2	1.4	1.0	1.4	2.2	3.2	4.1	5.1	6.1	7.1	2.0	1.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	1.4	1.0	1.4	2.2	3.2	4.1	5.1
f	3.2	2.2	1.4	1.0	1.4	2.2	3.2	4.1	5.1	6.1	3.0	2.0	1.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	2.2	1.4	1.0	1.4	2.2	3.2	4.1
g	4.1	3.2	2.2	1.4	1.0	1.4	2.2	3.2	4.1	5.1	4.0	3.0	2.0	1.0	0.0	1.0	2.0	3.0	4.0	5.0	3.2	2.2	1.4	1.0	1.4	2.2	3.2
h	5.1	4.1	3.2	2.2	1.4	1.0	1.4	2.2	3.2	4.1	5.0	4.0	3.0	2.0	1.0	0.0	1.0	2.0	3.0	4.0	4.1	3.2	2.2	1.4	1.0	1.4	2.2
j	6.1	5.1	4.1	3.2	2.2	1.4	1.0	1.4	2.2	3.2	6.0	5.0	4.0	3.0	2.0	1.0	0.0	1.0	2.0	3.0	5.1	4.1	3.2	2.2	1.4	1.0	1.4
k	7.1	6.1	5.1	4.1	3.2	2.2	1.4	1.0	1.4	2.2	7.0	6.0	5.0	4.0	3.0	2.0	1.0	0.0	1.0	2.0	6.1	5.1	4.1	3.2	2.2	1.4	1.0
l	8.1	7.1	6.1	5.1	4.1	3.2	2.2	1.4	1.0	1.4	8.0	7.0	6.0	5.0	4.0	3.0	2.0	1.0	0.0	1.0	7.1	6.1	5.1	4.1	3.2	2.2	1.4
ñ	9.1	8.1	7.1	6.1	5.1	4.1	3.2	2.2	1.4	1.0	9.0	8.0	7.0	6.0	5.0	4.0	3.0	2.0	1.0	0.0	8.1	7.1	6.1	5.1	4.1	3.2	2.2
z	2.2	2.0	2.2	2.8	3.6	4.5	5.4	6.3	7.3	8.2	1.4	1.0	1.4	2.2	3.2	4.1	5.1	6.1	7.1	8.1	0.0	1.0	2.0	3.0	4.0	5.0	6.0
x	2.8	2.2	2.0	2.2	2.8	3.6	4.5	5.4	6.3	7.3	2.2	1.4	1.0	1.4	2.2	3.2	4.1	5.1	6.1	7.1	1.0	0.0	1.0	2.0	3.0	4.0	5.0
c	3.6	2.8	2.2	2.0	2.2	2.8	3.6	4.5	5.4	6.3	3.2	2.2	1.4	1.0	1.4	2.2	3.2	4.1	5.1	6.1	2.0	1.0	0.0	1.0	2.0	3.0	4.0
v	4.5	3.6	2.8	2.2	2.0	2.2	2.8	3.6	4.5	5.4	4.1	3.2	2.2	1.4	1.0	1.4	2.2	3.2	4.1	5.1	3.0	2.0	1.0	0.0	1.0	2.0	3.0
b	5.4	4.5	3.6	2.8	2.2	2.0	2.2	2.8	3.6	4.5	5.1	4.1	3.2	2.2	1.4	1.0	1.4	2.2	3.2	4.1	4.0	3.0	2.0	1.0	0.0	1.0	2.0
n	6.3	5.4	4.5	3.6	2.8	2.2	2.0	2.2	2.8	3.6	#	5.1	4.1	3.2	2.2	1.4	1.0	1.4	2.2	3.2	5.0	4.0	3.0	2.0	1.0	0.0	1.0
m	7.3	6.3	5.4	4.5	3.6	2.8	2.2	2.0	2.2	2.8	7.1	6.1	5.1	4.1	3.2	2.2	1.4	1.0	1.4	2.2	6.0	5.0	4.0	3.0	2.0	1.0	0.0

Considerando las posiciones iniciales del teclado Qwerty una tabla de dos entradas es el cálculo de distancia desde el plano cartesiano.

3. Temperatura del algoritmo Simulated Annealing

Es el parámetro de control de enfriamiento, se determina usando una función de probabilidad, realiza saltos de amplitud de pasos largos al principio y luego cada vez más cortos, cuando la amplitud del paso aleatorio es suficientemente pequeña para no permitir bajar de la colina se puede decir que el resultado del algoritmo está templado.

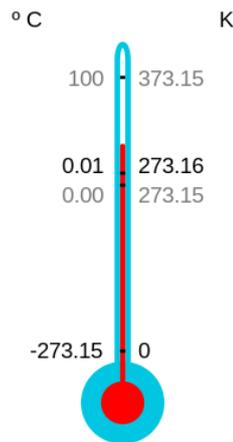


Figura 21. Medición de temperatura de Simulated Annealing

Inicializamos la temperatura:

$$T = 10000$$

Este valor va a ir descendiendo por cada intercambio de posiciones de dos teclas, a menor temperatura menor probabilidad de elegir sucesores peores.

En cada descenso de la temperatura (considerado como iteración) se evalúa si la distancia es menor en el nuevo estado se cambia la posición en caso contrario no se cambia, si hay cambio se actualiza el estado actual con el nuevo estado, de esta manera se llega a optimizar la distribución de las teclas.

4. Estado actual y estado nuevo del algoritmo Simulated Annealing

Un estado es el camino de deslizamiento al escribir una palabra o un documento entero, supongamos el caso de escribir la palabra “unap”, el estado es el camino trazado por el deslizamiento en el teclado “u+n+a+p”.

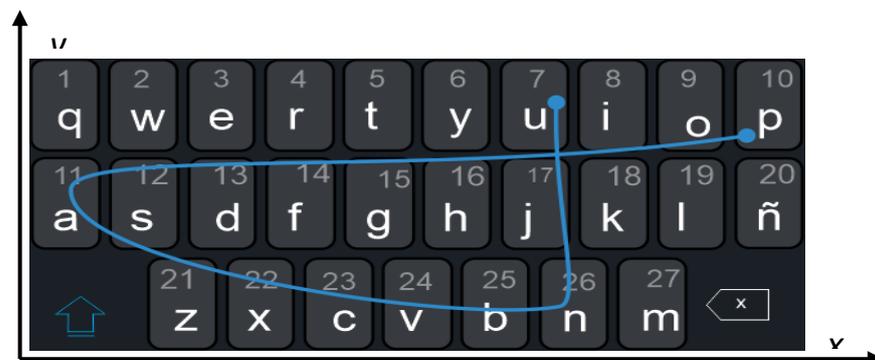


Figura 22. Teclado Qwerty como entrada para el modelo

Estado actual es el camino de las posiciones 7-26-11-10, corresponde al caso de la palabra “unap”. Su distancia es calculada sumando las distancias del camino recorrido, ver la tabla de distancias para sumar las distancias.

Estado actual (ruta) = 7-26-11-10

Distancia actual = 2.0+6.1+9.1= 17.138

El algoritmo Simulated Annealing, intercambia dos posiciones en forma aleatoria, si intercambiamos la posición de la letra “g” con la posición de la letra “n” el teclado habría quedado como en la siguiente figura.



Figura 23. Intercambio entre dos teclas

Estado nuevo es el camino de las posiciones 7-15-11-10, corresponde al caso de la palabra “unap” una vez intercambiada entre las teclas “g” y “n”, su distancia es calculada sumando las distancias del camino recorrido, ver la tabla de distancias para sumar las distancias.

Estado nuevo = 7-15-11-10

Distancia nueva = 2.2 + 4.0 + 9.1 = 15.291

5. Energía del algoritmo Simulated Annealing

La energía es la distancia calculada del camino tomado de un estado, permite el enfriamiento en el algoritmo Simulated Annealing, considerando los dos estados se tiene:

Energía Actual = 2.0+6.1+9.1= 17.138

Energía Nueva = 2.2 + 4.0 + 9.1 = 15.291

6. Función de aceptación del algoritmo Simulated Annealing

Es la heurística sobre la calidad de la solución $f'(n)$, que evalúa el reemplazamiento o no reemplazamiento del estado actual con el nuevo. La diferencia entre el estado actual y el estado nuevo es calcula para someter a esta función.

```
Si Energía Nueva < Energía Actual
  Prob = 1
sino
  Prob = exp( (Energía Actual – Energía Nueva)/Temperatura)
Si Prob > Random(0,1)
  Estado Actual = Estado Nuevo
```

Figura 24. Función de aceptación

Si la Energía Nueva es menor que la Energía Actual, la probabilidad es 1 se reemplaza el Estado Actual con el Estado Nuevo, en caso si no cumpliera la condición se somete a la probabilidad de Boltzmann, se puede aceptar la peor solución con una probabilidad de hallar la mejor.

Para el caso presentado, la Energía Nueva es menor que la Energía Actual, por tanto se tiene la probabilidad 1, se reemplaza el Estado Actual con el Estado Nuevo, la trayectoria guardada en este caso para ambos es:

Estado Actual = 7-15-11-10

Estado Nuevo = 7-15-11-10

También la distancia del Estado Actual 15.291 el mismo para el Estado Nuevo.

Seguidamente se actualiza la temperatura:

```
Temperatura = (1-VEnfriamiento) * Temperatura
Temperatura = (1-0.003) * 10000
Temperatura = 9700
```

Figura 25. Configuración de temperatura y velocidad

Por tanto, para el caso, en la primera iteración del algoritmo el teclado quedaría de la siguiente manera:

En este caso con el teclado Qwerty la distancia calculada para la palabra “unap” es 17.138, después de correr el algoritmo Simulated Annealing en una primera iteración se logra intercambiar las posiciones entre las teclas “g” y “n”, la distancia calculada es 15.291, podemos observar claramente que hay una diferencia de 1.847 con esto explicamos que la optimización en la distribución de las teclas permite obtener menos distancia en los millones de palabras que escribimos cada día.

Este proceso del Algoritmo Simulated Annealing, se repite hasta la que temperatura llegue a cero utilizando miles de palabras como entrada, se dice que está enfriado la temperatura gracias a la energía, las posiciones con las que se ha quedado el teclado serán las más óptimas para la escritura guardando la distancia menor al escribir un documento.

3.5.2. OE3: Metodología de validación del modelo de optimización de distribución de teclas del teclado virtual utilizando miles de palabras

Para la prueba se ha considerado datos de entrada para modelo de optimización:

- a) Se ha considerado 8 documentos .txt, recolectados en total 631,623 palabras y 3,606,564 caracteres, las cuales se han considerado como entrada para probar el modelo de optimización, con el objetivo de distribuir la ubicación de las teclas.
- b) También se ha considerado la distribución del teclado QWERTY la cual se ha puesto en un plano cartesiano para tomar las ubicaciones del mismo, estos datos fueron tomados como estado inicial para el modelo de optimización.

Prueba del modelo con el algoritmo Simulated Annealing hemos implementado en Python el proceso del algoritmo con las siguientes características.

- Temperatura: 10000
- Velocidad de enfriamiento: 0.001

- Cantidad de palabras: 631,623
- Cantidad de caracteres: 3,606,564
- Estado inicial: Teclado Qwerty
- Salida esperada: Nueva distribución del teclado

Una vez ejecutado, se obtiene los resultados, como resultado final la nueva distribución del teclado, la cual denominamos Ray Keyboard.

Interpretamos los resultados y comparando con otros propuesta en la sección de resultados.

También presentamos en los resultados mostramos la distancia inicial con el teclado QWERTY y la nueva distancia del teclado optimizado la cual es contrastable y discutido en los resultados.

3.5.3. OE2: Metodología de diseño mockups de un nuevo teclado virtual denominado Ray-Keyboard para dispositivos móviles basado en la distribución de teclas optimizadas

Para el diseño de los mockups se ha considerado la salida final del modelo de optimización, en la tabla 4 mostramos la estructura en la cual el modelo nos proporciona la nueva distribución optimizada de las posiciones del teclado.

Tabla 4

Estructura de salida del modelo (nueva distribución del teclado)

Id	Letra	X	Y
La Id es el identificador de la tecla	Es el nombre o etiqueta de la tecla del teclado.	Posiciones X en el plano cartesiano.	Posiciones Y en el plano cartesiano.

Considerando la nueva distribución del teclado diseñamos los siguientes mockups del nuevo teclado Ray Keyboard.



- Teclado virtual con nueva distribución
- Posición del teclado virtual en el dispositivo móvil
- Teclado virtual con nueva distribución
- Opción de deslizamiento del nuevo teclado virtual
- Teclado virtual a la izquierda para una sola mano
- Teclado virtual a la derecha para una sola mano
- Temas y estilo del teclado virtual

Los diseños de los mockups se presenta en la sección de resultados porque sus posiciones fueron considerados de acuerdo a los resultados finales de la optimización, la nueva distribución del teclado es la última iteración en la prueba de resultados.

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

El modelo de optimización ha sido validado con un conjunto de palabras como entrada para el algoritmo Simulated Annealing, también se ha considerado como estado inicial la distribución del teclado Qwerty y luego hemos analizado los resultados para evaluar si la optimización satisface a la nueva distribución del teclado virtual. En esta sección mostramos los detalles de los resultados y su interpretación.

4.1. Datos de entrada para el modelo de optimización.

Recolectamos 8 archivos diferentes las cuales se ha convertido en .txt con una herramienta propia diseñada en esta investigación la cual fundamentamos en la sección 3.2.1.

Nombre	Tamaño
1. LEYES DEL TRABAJO EN EQUIPO.txt	417 KB
2. LEYES DEL LIDERAZGO - JOHN C. MAXWELL.txt	369 KB
3. MANERAS DE GANARSE A LA GENTE - .txt	253 KB
4. ABE SHANA - LA ULTIMA SIRENA.txt	893 KB
5. ACTITUD DE VENCEDOR - JOHN C. MAXWELL.txt	288 KB
6. EL ORO Y LA CENIZA - ABECASSIS ELIETTE .txt	693 KB
7. SEAMOS PERSONAS DE INFLUENCIA - JOHN MAXWELL.txt	348 KB
8. VIVE TU SUEÑO - JOHN MAXWELL.txt	388 KB

Figura 26. Archivos .txt recolectados para datos de entrada

El contenido de cada archivo fue limpiado por la herramienta, como resultado se tiene archivos de texto.

Tabla 5

Textos múltiples como datos de entrada

N°	TEXTO	N° PAGINAS	PALABRAS	CARACTERES
1	LEYES DEL TRABAJO EN EQUIPO	211	70,999	413,558
2	LEYES DEL LIDERAZGO	180	63,130	366,245
3	MANERAS DE GANARSE A LA GENTE	147	44,695	252,231
4	ABE SHANA - LA ULTIMA SIRENA	513	158,472	881,673
5	ACTITUD DE VENCEDOR	168	48,616	283,861
6	EL ORO Y LA CENIZA	383	120,595	685,551
7	SEAMOS PERSONAS DE INFLUENCIA	189	58,611	344,414
8	VIVE TU SUEÑO	285	66,505	379,031
		TOTAL	631,623	3,606,564

Analizando cada texto se ha contabilizado la cantidad de páginas y la cantidad de palabras por cada documento, teniendo como resultado 3,606,564 caracteres en total, las cuales se ha alimentado al modelo de optimización que utiliza el algoritmo Simulated Annealing.

Por el otro lado, el algoritmo de optimización Simulated Annealing en su inicio del proceso requiere el estado inicial tabla 6 (solución inicial para optimizar), para ello proporcionamos los puntos (X,Y) de cada letra en el plano cartesiano, que corresponde la distribución del teclado convencional Qwerty.

Tabla 6

Estado inicial del algoritmo Simulated Annealing

ID	Tecla	Pos X	Pos Y
1	q	1	3
2	w	2	3
3	e	3	3
4	r	4	3
5	t	5	3
6	y	6	3
7	u	7	3
8	i	8	3
9	o	9	3
10	p	10	3
11	a	1	2
12	s	2	2
13	d	3	2
14	f	4	2
15	g	5	2
16	h	6	2
17	j	7	2
18	k	8	2
19	l	9	2
20	ñ	10	2
21	z	2	1
22	x	3	1
23	c	4	1
24	v	5	1
25	b	6	1
26	n	7	1
27	m	8	1

4.2. Análisis de resultados de optimización de distribución de teclado

Ponemos en marcha nuestro modelo de optimización de la distribución del teclado, proporcionamos los textos múltiples y el estado inicial al algoritmo Simulated Annealing, logramos ejecutar el algoritmo con los siguientes parámetros.

- Temperatura: 10000
- Velocidad de enfriamiento: 0.001
- Cantidad de palabras: 631,623



- Cantidad de caracteres: 3,606,564
- Estado inicial: Teclado Qwerty
- Salida esperada: Nueva distribución del teclado

Tabla 7

Resultados del intercambio del “Estado Actual” y el “Nuevo Estado”

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	Distancia
iteracion:1	cambio:(23,22)																											
Actual:	q,1,1	w,2,1	e,3,1	r,4,1	t,5,1	y,6,1	u,7,1	i,8,1	o,9,1	p,10,1	a,1,2	s,2,2	d,3,2	f,4,2	g,5,2	h,6,2	j,7,2	k,8,2	l,9,2	n,10,2	z,2,3	x,3,3	c,4,3	v,5,3	b,6,3	n,7,3	m,8,3	9,823,685
Nueva:	q,1,1	w,2,1	e,3,1	r,4,1	t,5,1	y,6,1	u,7,1	i,8,1	o,9,1	p,10,1	a,1,2	s,2,2	d,3,2	f,4,2	g,5,2	h,6,2	j,7,2	k,8,2	l,9,2	n,10,2	z,2,3	e,3,3	x,4,3	v,5,3	b,6,3	n,7,3	m,8,3	9,877,331
iteracion:2	cambio:(4,7)																											
Actual:	q,1,1	w,2,1	e,3,1	r,4,1	t,5,1	y,6,1	u,7,1	i,8,1	o,9,1	p,10,1	a,1,2	s,2,2	d,3,2	f,4,2	g,5,2	h,6,2	j,7,2	k,8,2	l,9,2	n,10,2	z,2,3	c,3,3	x,4,3	v,5,3	b,6,3	n,7,3	m,8,3	9,823,685
Nueva:	q,1,1	w,2,1	e,3,1	u,4,1	t,5,1	y,6,1	r,7,1	i,8,1	o,9,1	p,10,1	a,1,2	s,2,2	d,3,2	f,4,2	g,5,2	h,6,2	j,7,2	k,8,2	l,9,2	n,10,2	z,2,3	e,3,3	x,4,3	v,5,3	b,6,3	n,7,3	m,8,3	9,940,870
iteracion:3	cambio:(21,5)																											
Actual:	q,1,1	w,2,1	e,3,1	u,4,1	t,5,1	y,6,1	r,7,1	i,8,1	o,9,1	p,10,1	a,1,2	s,2,2	d,3,2	f,4,2	g,5,2	h,6,2	j,7,2	k,8,2	l,9,2	n,10,2	z,2,3	c,3,3	x,4,3	v,5,3	b,6,3	n,7,3	m,8,3	9,823,685
Nueva:	q,1,1	w,2,1	e,3,1	u,4,1	z,5,1	y,6,1	r,7,1	i,8,1	o,9,1	p,10,1	a,1,2	s,2,2	d,3,2	f,4,2	g,5,2	h,6,2	j,7,2	k,8,2	l,9,2	n,10,2	t,2,3	e,3,3	x,4,3	v,5,3	b,6,3	n,7,3	m,8,3	10,147,786
iteracion:4	cambio:(24,9)																											
...	...																											
...	...																											
iteracion:9204	cambio:(19,4)																											
Actual:	j,1,1	n,2,1	m,3,1	p,4,1	q,5,1	w,6,1	y,7,1	h,8,1	n,9,1	t,10,1	s,1,2	f,2,2	g,3,2	r,4,2	c,5,2	e,6,2	a,7,2	o,8,2	p,9,2	d,10,2	k,2,3	u,3,3	z,4,3	i,5,3	v,6,3	x,7,3	b,8,3	6,547,353
Nueva:	j,1,1	n,2,1	m,3,1	l,4,1	q,5,1	w,6,1	y,7,1	h,8,1	n,9,1	t,10,1	s,1,2	f,2,2	g,3,2	r,4,2	c,5,2	e,6,2	a,7,2	o,8,2	p,9,2	d,10,2	k,2,3	u,3,3	z,4,3	i,5,3	v,6,3	x,7,3	b,8,3	8,744,520
iteracion:9205	cambio:(2,3)																											
Actual:	j,1,1	n,2,1	m,3,1	l,4,1	q,5,1	w,6,1	y,7,1	h,8,1	n,9,1	t,10,1	s,1,2	f,2,2	g,3,2	r,4,2	c,5,2	e,6,2	a,7,2	o,8,2	p,9,2	d,10,2	k,2,3	u,3,3	z,4,3	i,5,3	v,6,3	x,7,3	b,8,3	6,547,353
Nueva:	j,1,1	m,2,1	n,3,1	l,4,1	q,5,1	w,6,1	y,7,1	h,8,1	n,9,1	t,10,1	s,1,2	f,2,2	g,3,2	r,4,2	c,5,2	e,6,2	a,7,2	o,8,2	p,9,2	d,10,2	k,2,3	u,3,3	z,4,3	i,5,3	v,6,3	x,7,3	b,8,3	8,863,910
iteracion:9206	cambio:(19,1)																											
Actual:	j,1,1	m,2,1	n,3,1	l,4,1	q,5,1	w,6,1	y,7,1	h,8,1	n,9,1	t,10,1	s,1,2	f,2,2	g,3,2	r,4,2	c,5,2	e,6,2	a,7,2	o,8,2	p,9,2	d,10,2	k,2,3	u,3,3	z,4,3	i,5,3	v,6,3	x,7,3	b,8,3	6,547,353
Nueva:	p,1,1	m,2,1	n,3,1	l,4,1	q,5,1	w,6,1	y,7,1	h,8,1	n,9,1	t,10,1	s,1,2	f,2,2	g,3,2	r,4,2	c,5,2	e,6,2	a,7,2	o,8,2	j,9,2	d,10,2	k,2,3	u,3,3	z,4,3	i,5,3	v,6,3	x,7,3	b,8,3	8,939,534

Tabla 8

Resultados de reemplazamiento del "Estado Actual" con el "Nuevo Estado"

N° Iteración	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	Distancia
1	q,1,1	w,2,1	e,3,1	r,4,1	t,5,1	y,6,1	u,7,1	i,8,1	o,9,1	p,10,1	a,1,2	s,2,2	d,3,2	f,4,2	g,5,2	h,6,2	j,7,2	k,8,2	l,9,2	ñ,10,2	z,2,3	x,3,3	c,4,3	v,5,3	b,6,3	n,7,3	m,8,3	9,823,685
2	q,1,1	w,2,1	e,3,1	u,4,1	z,5,1	y,6,1	r,7,1	i,8,1	v,9,1	p,10,1	a,1,2	s,2,2	d,3,2	f,4,2	g,5,2	h,6,2	j,7,2	k,8,2	l,9,2	ñ,10,2	t,2,3	c,3,3	x,4,3	o,5,3	b,6,3	n,7,3	m,8,3	9,553,614
3	n,1,1	s,2,1	e,3,1	u,4,1	z,5,1	x,6,1	r,7,1	p,8,1	w,9,1	i,10,1	a,1,2	t,2,2	f,3,2	d,4,2	g,5,2	h,6,2	j,7,2	k,8,2	l,9,2	ñ,10,2	v,2,3	c,3,3	y,4,3	o,5,3	b,6,3	q,7,3	m,8,3	9,452,460
4	n,1,1	s,2,1	e,3,1	u,4,1	z,5,1	x,6,1	r,7,1	p,8,1	w,9,1	i,10,1	a,1,2	t,2,2	f,3,2	d,4,2	l,5,2	h,6,2	j,7,2	k,8,2	g,9,2	ñ,10,2	v,2,3	c,3,3	y,4,3	o,5,3	b,6,3	q,7,3	m,8,3	8,814,607
5	n,1,1	s,2,1	e,3,1	u,4,1	z,5,1	x,6,1	r,7,1	p,8,1	w,9,1	i,10,1	u,1,2	t,2,2	f,3,2	d,4,2	l,5,2	h,6,2	j,7,2	k,8,2	g,9,2	c,10,2	v,2,3	ñ,3,3	y,4,3	o,5,3	b,6,3	q,7,3	m,8,3	8,752,360
6	o,1,1	c,2,1	e,3,1	a,4,1	s,5,1	x,6,1	r,7,1	p,8,1	v,9,1	i,10,1	u,1,2	n,2,2	s,3,2	d,4,2	l,5,2	h,6,2	k,7,2	j,8,2	f,9,2	g,10,2	w,2,3	ñ,3,3	y,4,3	t,5,3	b,6,3	q,7,3	m,8,3	8,612,746
7	m,1,1	f,2,1	d,3,1	i,4,1	s,5,1	r,6,1	t,7,1	x,8,1	o,9,1	h,10,1	ñ,1,2	n,2,2	g,3,2	q,4,2	w,5,2	c,6,2	k,7,2	a,8,2	p,9,2	z,10,2	l,2,3	u,3,3	e,4,3	v,5,3	b,6,3	y,7,3	j,8,3	8,577,319
8	m,1,1	f,2,1	d,3,1	i,4,1	s,5,1	r,6,1	t,7,1	x,8,1	o,9,1	h,10,1	ñ,1,2	n,2,2	g,3,2	q,4,2	a,5,2	c,6,2	k,7,2	w,8,2	p,9,2	z,10,2	l,2,3	u,3,3	e,4,3	v,5,3	b,6,3	y,7,3	j,8,3	7,653,701
9	m,1,1	f,2,1	d,3,1	i,4,1	v,5,1	r,6,1	t,7,1	x,8,1	o,9,1	h,10,1	ñ,1,2	n,2,2	g,3,2	q,4,2	a,5,2	c,6,2	k,7,2	w,8,2	p,9,2	z,10,2	l,2,3	u,3,3	e,4,3	s,5,3	b,6,3	y,7,3	j,8,3	7,625,191
10	b,1,1	f,2,1	d,3,1	i,4,1	v,5,1	r,6,1	t,7,1	x,8,1	o,9,1	h,10,1	ñ,1,2	n,2,2	q,3,2	j,4,2	a,5,2	k,6,2	c,7,2	w,8,2	p,9,2	z,10,2	l,2,3	u,3,3	e,4,3	s,5,3	m,6,3	y,7,3	g,8,3	7,623,009
11	s,1,1	m,2,1	w,3,1	u,4,1	n,5,1	p,6,1	ñ,7,1	d,8,1	j,9,1	f,10,1	h,1,2	x,2,2	l,3,2	o,4,2	a,5,2	i,6,2	r,7,2	z,8,2	y,9,2	v,10,2	b,2,3	e,3,3	c,4,3	k,5,3	g,6,3	t,7,3	q,8,3	7,548,841
12	l,1,1	v,2,1	w,3,1	u,4,1	n,5,1	k,6,1	p,7,1	d,8,1	z,9,1	f,10,1	h,1,2	x,2,2	s,3,2	o,4,2	a,5,2	e,6,2	r,7,2	j,8,2	i,9,2	ñ,10,2	b,2,3	y,3,3	c,4,3	m,5,3	g,6,3	t,7,3	q,8,3	7,096,273
13	v,1,1	x,2,1	u,3,1	o,4,1	m,5,1	g,6,1	r,7,1	f,8,1	w,9,1	k,10,1	b,1,2	i,2,2	a,3,2	p,4,2	s,5,2	z,6,2	t,7,2	j,8,2	y,9,2	ñ,10,2	d,2,3	e,3,3	n,4,3	l,5,3	c,6,3	q,7,3	h,8,3	6,728,046
14	4520	k,1,1	u,2,1	ñ,3,1	b,4,1	v,5,1	a,6,1	c,7,1	n,8,1	x,9,1	y,10,1	h,1,2	w,2,2	p,3,2	r,4,2	s,6,2	m,7,2	g,8,2	t,9,2	z,10,2	f,2,3	j,3,3	e,4,3	d,5,3	o,6,3	i,7,3	q,8,3	6,641,196
15	4522	k,1,1	u,2,1	p,3,1	b,4,1	v,5,1	a,6,1	c,7,1	n,8,1	x,9,1	h,10,1	y,1,2	w,2,2	ñ,3,2	r,4,2	s,6,2	m,7,2	g,8,2	t,9,2	z,10,2	f,2,3	j,3,3	e,4,3	d,5,3	o,6,3	i,7,3	q,8,3	6,636,108
16	4523	k,1,1	u,2,1	p,3,1	b,4,1	v,5,1	a,6,1	n,7,1	c,8,1	x,9,1	h,10,1	y,1,2	w,2,2	ñ,3,2	r,4,2	s,6,2	m,7,2	g,8,2	t,9,2	z,10,2	f,2,3	j,3,3	e,4,3	d,5,3	o,6,3	i,7,3	q,8,3	6,547,353

Tabla 9

Resultados de reemplazamiento al “Mejor Estado”

No Iteración	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	Distancia	
1	q,1,1	w,2,1	e,3,1	r,4,1	t,5,1	y,6,1	u,7,1	i,8,1	o,9,1	p,10,1	a,1,2	s,2,2	d,3,2	f,4,2	g,5,2	h,6,2	j,7,2	k,8,2	l,9,2	n,10,2	z,2,3	x,3,3	c,4,3	v,5,3	b,6,3	n,7,3	m,8,3	9,823,685	
2	q,1,1	w,2,1	e,3,1	u,4,1	z,5,1	y,6,1	r,7,1	i,8,1	v,9,1	p,10,1	a,1,2	s,2,2	d,3,2	f,4,2	g,5,2	h,6,2	j,7,2	k,8,2	l,9,2	n,10,2	t,2,3	c,3,3	x,4,3	o,5,3	b,6,3	n,7,3	m,8,3	9,553,614	
3	n,1,1	s,2,1	e,3,1	u,4,1	z,5,1	x,6,1	r,7,1	p,8,1	w,9,1	i,10,1	a,1,2	t,2,2	f,3,2	d,4,2	g,5,2	h,6,2	j,7,2	k,8,2	l,9,2	n,10,2	v,2,3	c,3,3	y,4,3	o,5,3	b,6,3	q,7,3	m,8,3	9,452,460	
4	n,1,1	s,2,1	e,3,1	u,4,1	z,5,1	x,6,1	r,7,1	p,8,1	w,9,1	i,10,1	a,1,2	t,2,2	f,3,2	d,4,2	l,5,2	h,6,2	j,7,2	k,8,2	g,9,2	n,10,2	v,2,3	c,3,3	y,4,3	o,5,3	b,6,3	q,7,3	m,8,3	8,814,607	
5	n,1,1	s,2,1	e,3,1	a,4,1	z,5,1	x,6,1	r,7,1	p,8,1	w,9,1	i,10,1	u,1,2	t,2,2	f,3,2	d,4,2	l,5,2	h,6,2	j,7,2	k,8,2	g,9,2	c,10,2	v,2,3	n,3,3	y,4,3	o,5,3	b,6,3	q,7,3	m,8,3	8,752,360	
6	o,1,1	c,2,1	e,3,1	a,4,1	z,5,1	x,6,1	r,7,1	p,8,1	v,9,1	i,10,1	u,1,2	n,2,2	s,3,2	d,4,2	l,5,2	h,6,2	k,7,2	j,8,2	f,9,2	g,10,2	w,2,3	n,3,3	y,4,3	t,5,3	b,6,3	q,7,3	m,8,3	8,612,746	
7	m,1,1	f,2,1	d,3,1	i,4,1	s,5,1	r,6,1	t,7,1	x,8,1	o,9,1	h,10,1	n,1,2	n,2,2	g,3,2	q,4,2	w,5,2	c,6,2	k,7,2	a,8,2	p,9,2	z,10,2	l,2,3	u,3,3	e,4,3	v,5,3	b,6,3	y,7,3	j,8,3	8,577,319	
8	m,1,1	f,2,1	d,3,1	i,4,1	s,5,1	r,6,1	t,7,1	x,8,1	o,9,1	h,10,1	n,1,2	n,2,2	g,3,2	q,4,2	a,5,2	c,6,2	k,7,2	w,8,2	p,9,2	z,10,2	l,2,3	u,3,3	e,4,3	v,5,3	b,6,3	y,7,3	j,8,3	7,653,701	
9	m,1,1	f,2,1	d,3,1	i,4,1	v,5,1	r,6,1	t,7,1	x,8,1	o,9,1	h,10,1	n,1,2	n,2,2	g,3,2	q,4,2	a,5,2	c,6,2	k,7,2	w,8,2	p,9,2	z,10,2	l,2,3	u,3,3	e,4,3	s,5,3	b,6,3	y,7,3	j,8,3	7,625,191	
10	b,1,1	f,2,1	d,3,1	i,4,1	v,5,1	r,6,1	t,7,1	x,8,1	o,9,1	h,10,1	n,1,2	n,2,2	q,3,2	j,4,2	a,5,2	k,6,2	c,7,2	w,8,2	p,9,2	z,10,2	l,2,3	u,3,3	e,4,3	s,5,3	m,6,3	y,7,3	g,8,3	7,623,009	
11	s,1,1	m,2,1	w,3,1	u,4,1	n,5,1	p,6,1	n,7,1	d,8,1	j,9,1	f,10,1	h,1,2	x,2,2	l,3,2	o,4,2	a,5,2	i,6,2	r,7,2	z,8,2	y,9,2	v,10,2	b,2,3	e,3,3	c,4,3	k,5,3	g,6,3	t,7,3	q,8,3	7,548,841	
12	l,1,1	v,2,1	w,3,1	u,4,1	n,5,1	k,6,1	p,7,1	d,8,1	z,9,1	f,10,1	h,1,2	x,2,2	s,3,2	o,4,2	a,5,2	e,6,2	r,7,2	j,8,2	i,9,2	n,10,2	b,2,3	y,3,3	c,4,3	m,5,3	g,6,3	t,7,3	q,8,3	7,096,273	
13	v,1,1	x,2,1	u,3,1	o,4,1	m,5,1	g,6,1	r,7,1	f,8,1	w,9,1	k,10,1	b,1,2	t,2,2	a,3,2	p,4,2	s,5,2	z,6,2	t,7,2	j,8,2	y,9,2	n,10,2	d,2,3	e,3,3	n,4,3	l,5,3	c,6,3	q,7,3	h,8,3	6,728,046	
14	4520	k,1,1	u,2,1	n,3,1	b,4,1	a,6,1	c,7,1	n,8,1	x,9,1	y,10,1	h,1,2	w,2,2	p,3,2	r,4,2	l,5,2	s,6,2	m,7,2	g,8,2	t,9,2	z,10,2	f,2,3	j,3,3	e,4,3	d,5,3	o,6,3	i,7,3	q,8,3	6,641,196	
15	4522	k,1,1	u,2,1	p,3,1	b,4,1	v,5,1	a,6,1	c,7,1	n,8,1	x,9,1	h,10,1	y,1,2	w,2,2	n,3,2	r,4,2	l,5,2	s,6,2	m,7,2	g,8,2	t,9,2	z,10,2	f,2,3	j,3,3	e,4,3	d,5,3	o,6,3	i,7,3	q,8,3	6,636,108
16	4523	k,1,1	u,2,1	p,3,1	b,4,1	v,5,1	a,6,1	n,7,1	c,8,1	x,9,1	h,10,1	y,1,2	w,2,2	n,3,2	r,4,2	l,5,2	s,6,2	m,7,2	g,8,2	t,9,2	z,10,2	f,2,3	j,3,3	e,4,3	d,5,3	o,6,3	i,7,3	q,8,3	6,547,353

Después de ejecutar el algoritmo en el modelo de optimización, podemos observar:

Iteraciones realizadas: 9,206

Reemplazos al estado actual: 16

Reemplazos el mejor estado: 16

Por la naturaleza del algoritmo no se definen en la ejecución la cantidad de iteraciones a realizar, sin embargo, de acuerdo a la temperatura y la velocidad de enfriamiento se ha ejecutado 9,206 iteraciones.

La primera vez, se tomó las posiciones del teclado Qwerty, realizando la trayectoria denominada “Estado Actual” con el conjunto de entrada 3,606,564 caracteres, se ha calculado la distancia en este caso es 9,823,684.64, ver la primera fila de la tabla 6. También se ha establecido como “Nuevo Estado” la misma trayectoria y distancia del “Estado Actual” de igual manera el “Mejor estado” se estableció la misma trayectoria y distancia del “Estado Actual”.

En su proceso, el algoritmo ha realizado 9,206 iteraciones como se puede observar en la tabla 7, en cada iteración eligió de forma aleatoria dos teclas de los 27 del teclado, ha intercambiado la posición entre esas dos teclas, este intercambio se ha modificado en su trayectoria al “Nuevo Estado”.

Además, en cada iteración se halló la probabilidad de Boltzman proporcionando la distancia del “Estado Actual” y la distancia del “Nuevo Estado”, si la probabilidad es mayor que un número aleatorio entre (0, 1) se reemplaza el “Estado Actual” por el “Nuevo Estado”, podemos observar la tabla 7 los detalles de los reemplazos, en este experimento hubieron 16 reemplazos durante 9,206 iteraciones ver tabla 8.

Finalmente, el algoritmo evalúa el “Estado Mejor” que almacena la mejor trayectoria en cada iteración, la evaluación consistió en comparar si la distancia del “Estado Actual” es menor que la distancia del “Estado Mejor” entonces se reemplaza el “Estado Mejor” por el “Estado Actual”, en este experimento hubieron 16 reemplazos durante 9,206 iteraciones, ver tabla 9.

En cada iteración se actualiza la temperatura que empezó con 10,000, desciende con su velocidad de enfriamiento considerado 0.001, en cada iteración se ha evaluado el “Estado

Actual” con su distancia, el “Nuevo Estado” con su distancia y el “Estado Mejor” con su distancia.

El algoritmo ha terminado su proceso cuando la temperatura ha descendido a 0.00 (cero), en este experimento se dice ya está estable, lo que indica que ya está optimizado las posiciones del teclado.

La última fila de la tabla 9 se muestra el último “Estado Mejor” cuyas posiciones del teclado fueron optimizados utilizando 3,606,564 caracteres, asimismo su distancia es 6547352.95, comprando con la distancia inicial 9,823,684.64, haciendo una diferencia el resultado es 3,276,331.69 (más de tres millones de unidades en distancia al optimizar).

Terminada el experimento presentamos la salida final en la tabla 10, la última evaluación al estado Mejor que corresponde a una distribución del teclado que guarda la menor distancia en el recorrido de todas las palabras proporcionada en la entrada.

Tabla 10

Salida final del modelo de optimización

1	2	3	4	5
ID1	ID2	ID3	ID4	ID5
k,1,1	u,2,1	p,3,1	b,4,1	v,5,1
ID6	ID7	ID8	ID9	ID10
a,6,1	n,7,1	c,8,1	x,9,1	h,10,1
ID11	ID12	ID13	ID14	ID15
y,1,2	w,2,2	ñ,3,2	r,4,2	l,5,2
ID16	ID17	ID18	ID19	ID20
s,6,2	m,7,2	g,8,2	t,9,2	z,10,2
ID21	ID22	ID23	ID24	ID25
f,2,3	j,3,3	e,4,3	d,5,3	o,6,3
ID26	ID27			
i,7,3	q,8,3			

La ID es el identificador de cada tecla de las 27 teclas del teclado español, en cada casillero mostramos la letra corresponde en esa posición del plano cartesiano X y Y.

4.3. Nueva distribución del teclado virtual

La salida final del modelo de optimización mostrada en la tabla 10, se registran, la ID es el identificador de la tecla, NOMBRE DE LA TECLA es la columna donde se guarda el

carácter o tecla, finalmente la posición X y Y de la tecla de acuerdo a las coordenadas como posición.

Tabla 11

Posiciones de las teclas de la nueva distribución

ID	NOMBRE TECLA	X	Y
1	K	1	1
2	U	2	1
3	P	3	1
4	B	4	1
5	V	5	1
6	A	6	1
7	N	7	1
8	C	8	1
9	X	9	1
10	H	10	1
11	Y	1	2
12	W	2	2
13	Ñ	3	2
14	R	4	2
15	L	5	2
16	S	6	2
17	M	7	2
18	G	8	2
19	T	9	2
20	Z	10	2
21	F	2	3
22	J	3	3
23	E	4	3
24	D	5	3
25	O	6	3
26	I	7	3
27	Q	8	3

Esta nueva distribución optimizada es considerada como nueva distribución del teclado virtual para dispositivos móviles.

Finalmente, la nueva distribución del teclado se implementa como aplicativo móvil para dispositivo Android.

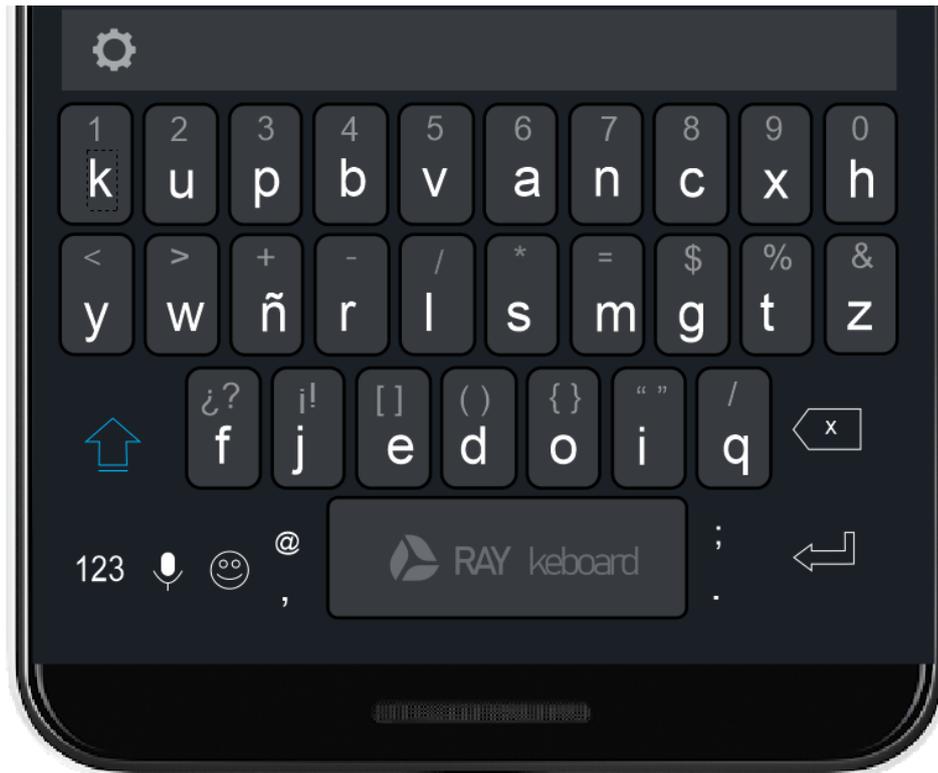


Figura 27. Teclado virtual con nueva distribución

4.4. Diseño de los mockups del nuevo teclado virtual RAY Keyboard con distribución de teclas optimizada.

El diseño del Rar Keyboard, está basado en la optimización de la ubicación de las teclas utilizando millones de letras y palabras.

a) Posición de teclado virtual en el dispositivo móvil.

Posición del teclado virtual en el dispositivo móvil, es clásico y convencional diseñamos el teclado virtual en la parte inferior del dispositivo en el tamaño de 1/3 de la pantalla de cualquier dispositivo móvil.

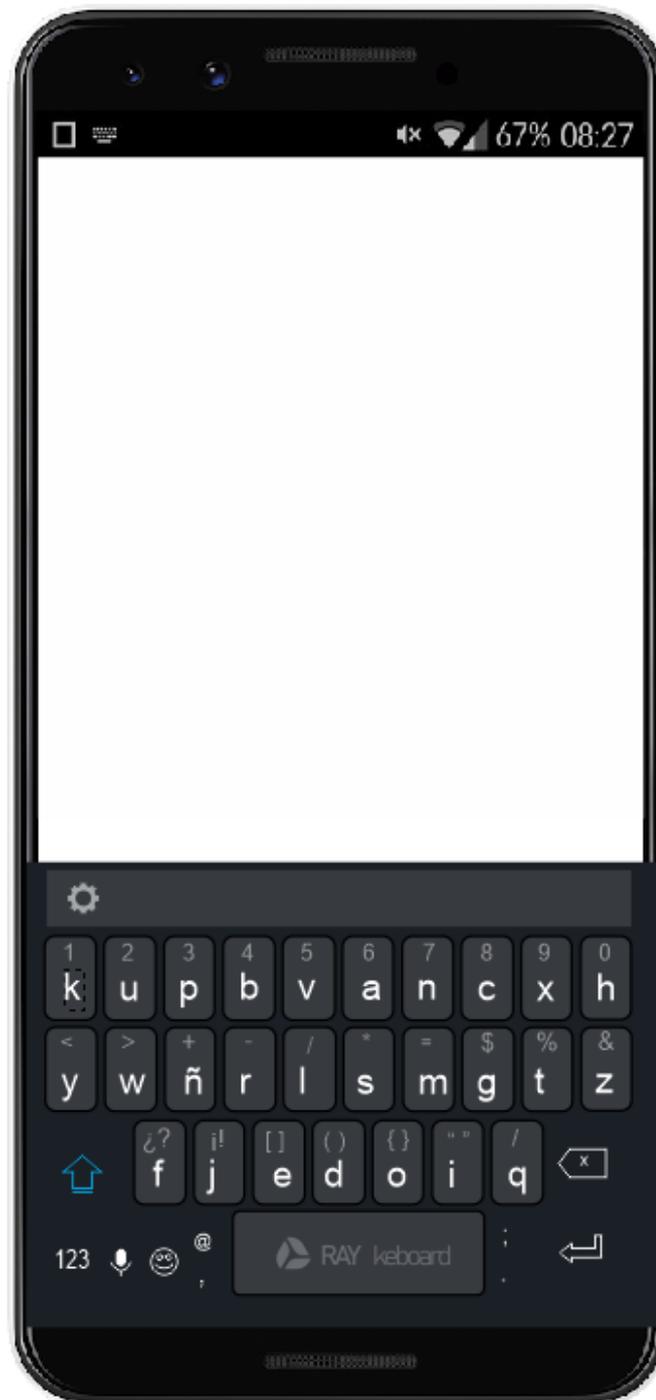


Figura 28. Posición del teclado virtual en el dispositivo móvil

b) Posiciones de las teclas en el nuevo teclado virtual

Las posiciones fueron adaptadas de acuerdo a la nueva distribución de las teclas, el teclado virtual consiste en 3 filas y 10 columnas, los números y otras teclas adicionales deben ser generadas al presionar más de un segundo en la tecla correspondiente.



Figura 29. Teclado virtual con nueva distribución

c) El teclado numérico

El teclado numérico, también se cambió las posiciones, para mayor facilidad en el deslizamiento, los números se presentan de forma ascendente iniciando de la parte inferior en posiciones, le acompañan los diferentes símbolos para escribir expresiones matemáticas para el usuario.

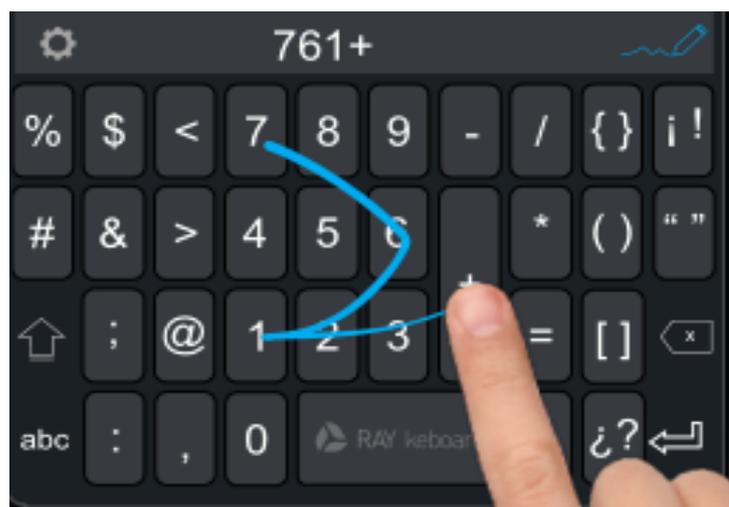


Figura 30. Teclado numérico del nuevo teclado virtual

d) Opción de deslizamiento

La opción de deslizamiento o arrastre (Swype) para generar palabras es importante, lo diseñamos el teclado virtual con esta opción, el usuario fácilmente puede generar palabras haciendo el paso de arrastre por las teclas necesarias, de esta forma logra escribir o generar sus palabras, la barra espaciadora es una tecla como cualquiera que permite el arrastre generando un espacio, talvez los usuarios puedan hacer deslizamiento constante cuando quiere escribir frases largas o documentos completos, en este caso el usuario debería realizar una larga trayectoria pasando por las diferentes letras y la borra espaciadora.

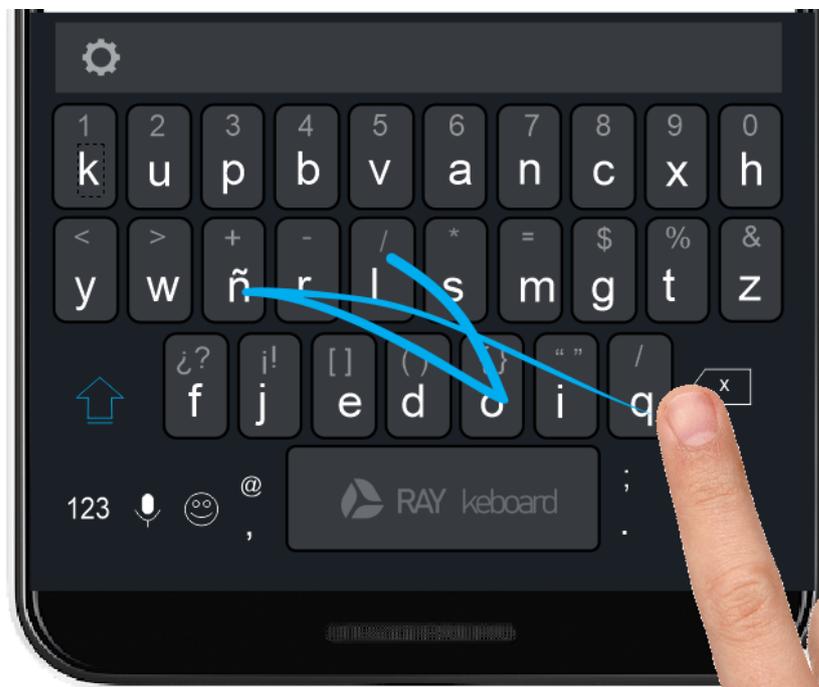


Figura 31. Opción de deslizamiento del nuevo teclado virtual

e) Configuración de la ubicación del teclado virtual recomendable para un solo mano

Para los usuarios exigentes que quieren escribir con una sola mano, diseñamos un modelo el costado sea a la izquierda o derecha configurable y adaptable a los dispositivos móviles, el botón “>” y “<”.

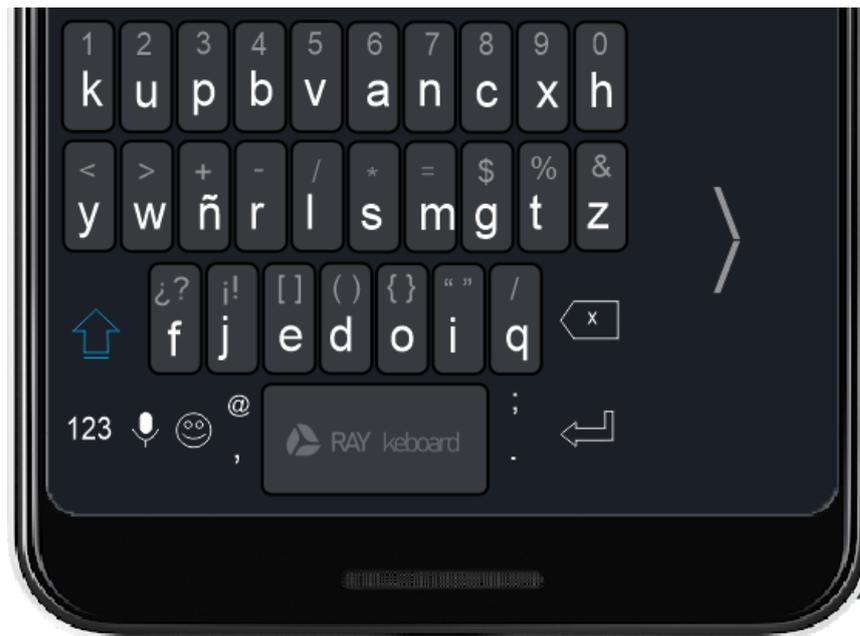


Figura 32. Teclado virtual a la izquierda para una sola mano

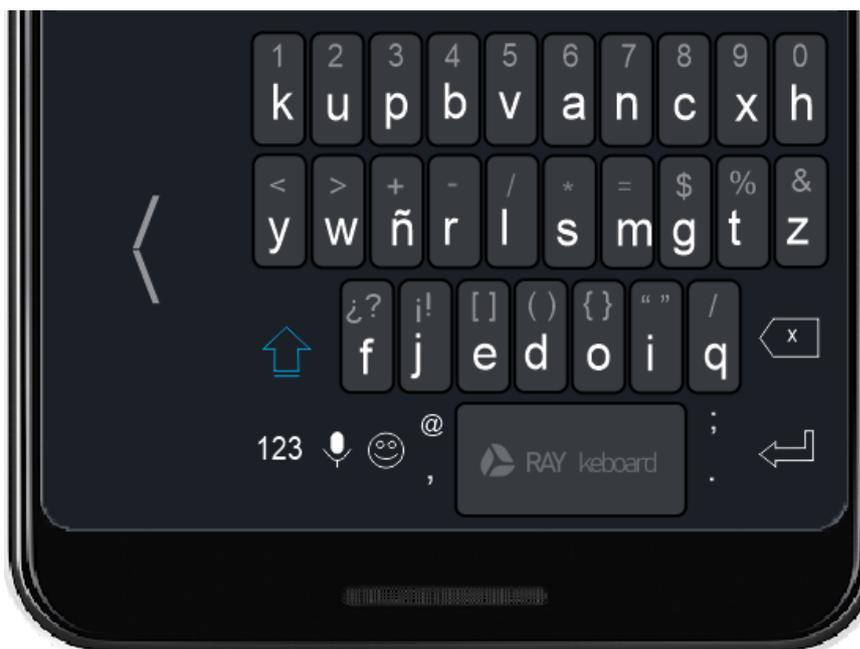


Figura 33. Teclado virtual a la derecha para una sola mano

f) Configuración de estilos y temas

Se ha diseñado cuatro temas para cambiar el estilo de formas y colores del teclado virtual, el usuario puede escoger de acuerdo con su gusto de colores. Esta opción se ofrece en la configuración del aplicativo.

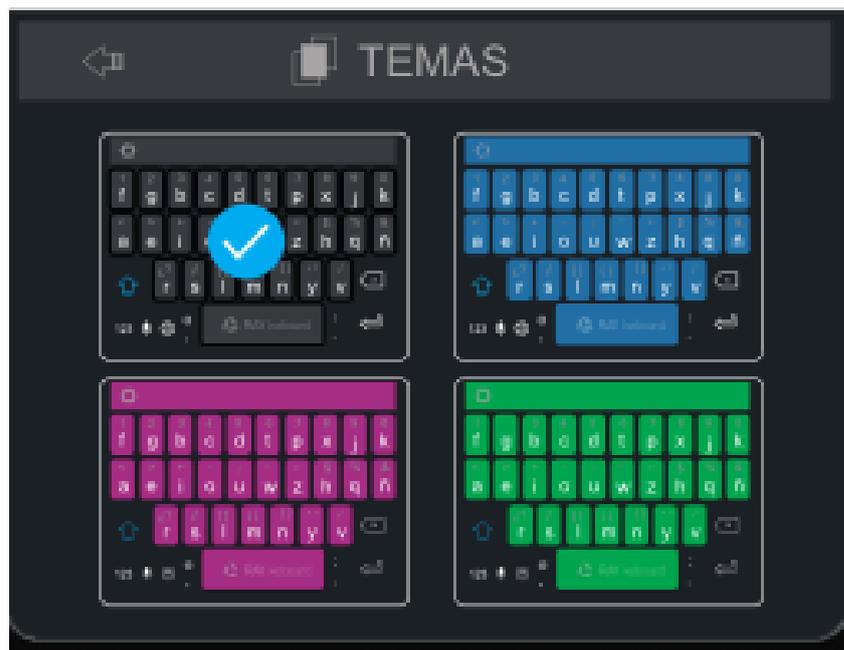


Figura 34. Temas y estilos del teclado virtual

4.5. Discusión de resultados

Las posiciones del teclado QWERTY ha sido modificado Wolcott (1920) ubicó las letras menos utilizadas (J,Q,Z y X) debajo de los dedos índices, por otro lado (Yang y Mali, 2016) diseñaron 3 tipos de teclados para optimizar las posiciones del teclado con el algoritmo Simulated Annealing utilizaron 233000 palabras de un corpus ingles American National Corpus. Comparando con esta investigación se propone un modelo de optimización de ubicación de las teclas del teclado utilizando el algoritmo de Inteligencia Artificial Simulated Annealing, modificamos la distribución del teclado QWERTY en su totalidad basado en los resultados de un experimento.

La propuesta de Dvorak y Dealey (1936) fue el más serio y prolongado que rechazó QWERTY, propone un diseño basado en investigación científica relacionados a la frecuencia de uso de las teclas, además el teclado Dvorak fue investigada intensamente por la Marina de Estados Unidos en 1944 y por la Oficina de Correos de Australia en 1953 concluyeron que el teclado Dvorak sí daría lugar a un aumento sustancial en la eficiencia de los mecanógrafos, en 1939 la Universidad de Chicago informó que el teclado Dvorak se empleó con éxito en un programa de escritura en el teclado enseñando a los niños de la escuela, lo niños lograron escribir más rápido que con el teclado QWERTY

lograron superar 50 palabras correctas por minuto. Comparando con nuestra investigación la distribución de las posiciones del teclado fue un intenso trabajo, para ello recolectamos 631,623 palabras con un total de 3,606,564 caracteres o letras en total que provienen de varios documentos textos formales como libros, sometidos como entrada las letras recolectadas al algoritmo de inteligencia artificial Simulated Annealing para optimizar la distribución del teclado, como salida final nuestro modelo obtiene una nueva distribución del teclado lo denominamos el nuevo teclado RayKeyboard, nuestra trabajo también fue un investigación científica basado en inteligencia artificial, mostramos resultados como recorridos donde la distancia inicial es 9,823,684.64, una vez optimizado la distancia lograda es 6547352.95, logramos una diferencia menor de más 3 millones, esto permite escribir documento o textos más rápido que el teclado Dvorak.

En cuanto a los teclados de los dispositivos móviles, desde 1980s presentaron el teclado de 12 teclas adaptado para los teléfonos celulares Silverberg (2007) podían generar al menos 26 caracteres AbuHmed *et al.* (2015) presenta un nuevo teclado virtual UOIT que está basado en escribir palabras como dibujando basado en 13 teclas grandes que reemplazan a 26 teclas normales, en su validación muestran un promedio de 11.3 palabras por minuto en la escritura; por otro lado Darbar *et al.* (2016) propone un nuevo teclado para smartwatches que permite ingresar alfabetos más frecuentes en un solo toque, muestran resultados de promedio de palabras de ingreso por minuto; en cuanto a los smartphones han adaptado el teclado QWERTY como teclado virtual. Comparando con nuestra investigación proponemos un nuevo teclado virtual Ray Keyboard basado en una nueva distribución de teclado, la misma que fue un resultado de investigación en cuanto a la optimización de la distribución de posiciones de los 27 caracteres del teclado español, este teclado Ray keyboard permite escribir más rápido cualquier texto o documento utilizando el método Swype (arrastre o deslizamiento sobre las letras en el teclado virtual).

CONCLUSIONES

- Para el conjunto de entrada del modelo de optimización de la distribución del teclado, se ha recolectado 631,623 palabras un total de 3,606,564 caracteres desde 8 textos con una herramienta desarrollada propiamente para esta investigación, además se ha considerado como estado inicial del modelo las posiciones del teclado Qwerty.
- Realizamos el experimento con el conjunto de entrada y el estado inicial utilizando el algoritmo de inteligencia artificial Simulated Annealing, mostramos 9,206 iteraciones de intercambio entre dos posiciones de teclas de forma aleatoria, evaluando en cada iteración el “Estado Actual”, “Nuevo Estado” y “Estado Mejor”, como final del modelo se tiene la nueva distribución del teclado basado en menor distancia con la trayectoria trazada con el conjunto de entrada. Mostramos en el experimento la distancia inicial 9,823,684.64 luego de optimizar el nuevo teclado obtiene 6547352.95 de distancia, realizando una diferencia se tiene 3,276,331.69 (más de tres millones de unidades en distancia al optimizar).
- La nueva distribución del teclado RayKeyboard de esta investigación compramos con el teclado Dvorak.
- Adicionalmente presentamos el nuevo teclado virtual para dispositivos móviles Android.

RECOMENDACIONES

- La nueva distribución del teclado virtual es cambiante por las iteraciones a realizarse en el algoritmo, incluso es cambiante de acuerdo a la cantidad de palabras presentadas como entrada, en ese sentido un usuario podría adaptar a la mejor distribución de acuerdo a lo que escribe un área determinada, sin embargo es necesario seguir investigación la distribución del teclado virtual para diferentes áreas que podría ser un usuario escritor.
- El modelo de optimización utilizando Simulated Annealing a arrojado buenos resultados en problemas de ruta, este trabajo de investigación también se muestra mejor distancia para la distribución del teclado virtual, sin embargo es necesario seguir investigando las nuevas distribuciones aplicando otras técnicas como los algoritmos genéticos.
- También se recomienda nuevas propuestas de construcción del teclado físico con nuevas distribuciones del teclado.

BIBLIOGRAFÍA

- AbuHmed, T., Lee, K., & Nyang, D. H. (2015). UOIT Keyboard: A Constructive Keyboard for Small Touchscreen Devices. *IEEE Transactions on Human-Machine Systems*, 45(6), 782–789. <https://doi.org/10.1109/THMS.2015.2449309>
- Beeching, W. (1894). Century of the typewriter. *British Typewriter Museum Publishing*.
- Biegel, R. (1934). An improved typewriter keyboard. *The Human Factor*, 8, 280-285.
- Cha, J. M., Choi, E., & Lim, J. (2015). Virtual Sliding QWERTY: A new text entry method for smartwatches using Tap-N-Drag. *Applied Ergonomics*, 51, 263–272. <https://doi.org/10.1016/j.apergo.2015.05.008>
- Ciobanu, O., Gavati, C., & Cozmei, R. (2016). The keyboard remains the least economically designed computer device. *E-Health and Bioengineering Conference EHB*, 0–3. <https://doi.org/10.1109/EHB.2015.7391585>
- Cocking, R. W. (1970). In place of QWERTY. *Daily Telegraph Supplement*, No. 301 (24 July).
- Darbar, R., Dash, P., & Samanta, D. (2016). ETAO Keyboard: Text Input Technique on Smartwatches. *Procedia Computer Science*, 84, 137–141. <https://doi.org/10.1016/j.procs.2016.04.078>
- Deiningner, R. L. (1960). Human Factors Engineering Studies of the Design and Use of Pushbutton Telephone Sets. *Bell System Technical Journal*, 39(4), 995–1012. <https://doi.org/10.1002/j.1538-7305.1960.tb04447.x>
- Dell'Amico, M., Díaz, J. C. D., Iori, M., & Montanari, R. (2009). The single-finger keyboard layout problem. *Computers and Operations Research*, 36(11), 3002–3012. <https://doi.org/10.1016/j.cor.2009.01.018>
- Dvorak, A., & Dealey, W. L. (1936). *Typewriter keyboard patent 2,040,248*.
- Eggers, J., Feillet, D., Kehl, S., Wagner, M. O., & Yannou, B. (2003). Optimization of the keyboard arrangement problem using an Ant Colony algorithm. *European Journal of Operational Research*, 148(3), 672–686. [https://doi.org/10.1016/S0377-2217\(02\)00489-7](https://doi.org/10.1016/S0377-2217(02)00489-7)

- Ferguson, D., & Duncan, J. (1974). Keyboard Design and Operating Posture. *Ergonomics*, 17(6), 731–744. <https://doi.org/10.1080/00140137408931420>
- Gilbert, W. A. (1930). *The Orthographic Keyboards*. Spokane, Washington: Shaw and Borden.
- Hsiao, H. C., Wu, F. G., & Chen, C. H. (2014). Design and evaluation of small, linear QWERTY keyboards. *Applied Ergonomics*, 45(3), 655–662. <https://doi.org/10.1016/j.apergo.2013.09.001>
- Huhn, K., Seungyoung, Y. & So-Yeon, Y., (2019) Exploring touch feedback display of virtual keyboards for reduced eye movements. *Displays*, Volume 56.
- Huang, Y. C., & Wu, F. G. (2015). Visual and manual loadings with QWERTY-like ambiguous keyboards: Relevance of letter-key assignments on mobile phones. *International Journal of Industrial Ergonomics*, 50, 143–150. <https://doi.org/10.1016/j.ergon.2015.08.002>
- Kim, H., Yi, S., & Yoon, S. Y. (2019). Exploring touch feedback display of virtual keyboards for reduced eye movements. *Displays*, 56(August 2018), 38–48. <https://doi.org/10.1016/j.displa.2018.11.004>
- Kinthead, R. (1975). Typing Speed, Keying Rates, and Optimal Keyboard Layouts. *Proceedings of the Human Factors Society Annual Meeting*, 19(2), 159–161. <https://journals.sagepub.com/doi/10.1177/154193127501900203>
- Leung, L., & Aarabi, P. (2014). Mobile circular keyboards. *Canadian Conference on Electrical and Computer Engineering*, 1–4. <https://doi.org/10.1109/CCECE.2014.6900941>
- Levy, D. (2002). The fastap keypad and pervasive computing. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2414, 58–68. https://doi.org/10.1007/3-540-45866-2_6
- Mackenzie, I. S., & Soukoreff, R. W. (2011). Text Entry for Mobile Computing : Models and Methods , Theory and Practice. *Human-Computer Interaction*, 17(December 2014), 37–41.

- Martin, A. (1972). A new keyboard layout. *Applied Ergonomics*, 3(1), 48–51.
[https://doi.org/10.1016/0003-6870\(72\)90011-7](https://doi.org/10.1016/0003-6870(72)90011-7)
- Meyer, A. (1995). Pen computing: a technology overview and a vision. *ACM SIGCHI Bulletin*, 27(3). <https://doi.org/10.1145/221296.221308>
- Nelson, W. (1920). The improvements in connection with keyboards for typewriters. *British Patent No. 155,446*.
- Niu, J., Liu, Y., Lin, J., Zhu, L., & Wang, K. (2014). Stroke ++ : A new Chinese input method for touch screen mobile phones. *Journal of Human Computer Studies*, 72(4), 440–450. <https://doi.org/10.1016/j.ijhcs.2014.01.001>
- Noyes, J. (1983). Chord keyboards. *Applied Ergonomics*, 14(1), 55–59.
[https://doi.org/10.1016/0003-6870\(83\)90221-1](https://doi.org/10.1016/0003-6870(83)90221-1)
- Osman, I., & Kelly, J. (1996). *Meta-Heuristics Theory and Applications*.
<https://doi.org/10.1007/978-1-4613-1361-8>
- Phillips, A. (1968). *No Title* (Office Her Majesty's Stationery (ed.)).
- Pritom, A. I., Mahmud, H., Ahmed, S., Hasan, M. K., & Khan, M. M. (2016). TYPEHEX keyboard: A virtual keyboard for faster typing in Smartphone. *2015 18th International Conference on Computer and Information Technology, ICCIT 2015*, 522–526. <https://doi.org/10.1109/ICCITechn.2015.7488126>
- Rajkumar, D., Punyashlok, D., & Debasis, S. (2016). ETAO Keyboard: Text Input Technique on Smartwatches, *Procedia Computer Science*, Volume 84.
- Rowell, S. W. (1909). Typewriter keyboard. *Specification of letters patent*, U.S. Patent No. 943,466
- Russell, S., & Norving, P. (2010). *Artificial Intelligence a Modern Approach* (Trird Edit). Pearson Education Inc. Recuperado de <https://www.iberdrola.com/te-interesa/tecnologia/que-es-inteligencia-artificial>
- Scott MacKenzie, I., & Zhang, S. X. (1999). The design and evaluation of a high-performance soft keyboard. *Conference on Human Factors in Computing Systems - Proceedings, May, 25–31*. <https://doi.org/10.1145/302979.302983>

- Sholes, C., Glidden, C., & Soulé, S. (1868). *Type writing machine patent 79,868*. 79.
- Silfverberg, M. (2007). Historical Overview of Consumer Text Entry Technologies. In *Text Entry Systems*. Elsevier Inc. <https://doi.org/10.1016/B978-012373591-1/50001-2>
- Smith, B. a., & Zhai, S. (2001). Optimised Virtual Keyboards with and without Alphabetical Ordering – A Novice User Study. In *Proceedings of INTERACT 2001*, 92–99.
- Ward, J. (1936). A Study of Typewriter Keyboards. *Educational Reasearch Bulletin*, 15(9), 243–246. Recuperado de <https://www.jstor.org/stable/1471353?seq=1>
- Wolcott, C. (1920). *Keyboard patent 1,342,244*.
- Wu, F. G., Huang, Y. C., & Wu, M. L. (2014). New chording text entry methods combining physical and virtual buttons on a mobile phone. *Applied Ergonomics*, 45(4), 825–832. <https://doi.org/10.1016/j.apergo.2013.10.011>
- Yamada, H. (1980). A historical study of typewriters and typing methods: from the position of planning Japanese parallels. *Journal of Information Processing*, 2(4), 175–202.
- Yang, N., & Mali, A. D. (2016). Modifying keyboard layout to reduce finger-Travel distance. *Proceedings - 2016 IEEE 28th International Conference on Tools with Artificial Intelligence, ICTAI 2016*, 165–168. <https://doi.org/10.1109/ICTAI.2016.31>
- Zhai, S., & Kristensson, P. O. (2008). Interlaced QWERTY - Accommodating ease of visual search and input flexibility in shape writing. *Conference on Human Factors in Computing Systems - Proceedings*, 593–596. <https://doi.org/10.1145/1357054.1357149>



ANEXOS

Anexo 1. Código fuente del modelo de optimización

PUNTOS.JAVA

```
package simulated;

public class Puntos {

    char Letra;

    int Id;

    int x;

    int y;

    public Puntos(){

        this.x = (int)(Math.random()*2000);

        this.y = (int)(Math.random()*2000);

        this.Letra = ' ';

        this.Id = 0;

    }

    public Puntos(int Id,char L,int x, int y){

        this.Id = Id;

        this.Letra = L;

        this.x = x;

        this.y = y;

    }

    public int getX(){

        return this.x;

    }

    public int getY(){

        return this.y;

    }

    public int getId(){

        return this.Id;

    }

}
```

```
}  
  
public char getLetra(){  
    return this.Letra;  
}  
  
public double distanceTo(Puntos city){  
    int xDistance = Math.abs(getX() - city.getX());  
    int yDistance = Math.abs(getY() - city.getY());  
    double distance = Math.sqrt( (xDistance*xDistance) + (yDistance*yDistance) );  
    return distance;  
}  
  
@Override  
public String toString(){  
    return ""+getLetra()+","+getX()+","+getY();  
}  
}
```

RECORRIDO.JAVA

```
package simulated;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.*;  
  
public class Recorrido {  
    private ArrayList tour = new ArrayList<Puntos>();  
    private double distance = 0;  
    public Recorrido(){  
        for (int i = 0; i < SeguimientoRecorrido.numberOfCities(); i++) {  
            tour.add(null);  
        }  
    }  
    public Recorrido(ArrayList tour){
```

```
        this.tour = (ArrayList) tour.clone();
    }
    public ArrayList getTour(){
        return tour;
    }
    public void generateIndividual() {
        for (int cityIndex = 0; cityIndex < SeguimientoRecorrido.numberOfCities(); cityIndex++) {
            setCity(cityIndex, SeguimientoRecorrido.getCity(cityIndex));
        }
        //Collections.shuffle(tour);
    }
    public Puntos getCity(int tourPosition) {
        return (Puntos)tour.get(tourPosition);
    }
    public void setCity(int tourPosition, Puntos city) {
        tour.set(tourPosition, city);
        this.distance = 0;
    }
    public void Cambio(int tourPosition, Puntos city,char L) {
//    System.out.println("Palabrast:"+tour.get(tourPosition));
        Puntos aux = new Puntos();
        aux = city;
        aux.Letra=L;
        tour.set(tourPosition, aux);

//    System.out.println("Palabrasw:"+tour.get(tourPosition));
        this.distance = 0;
    }
    public double getDistance(Vector Palabras){
        int T1 = Palabras.size();
//    System.out.println("T1:"+T1);
```

```
if (distance == 0) {
    double tourDistance = 0;
    for(int i=0; i<T1; i++){
//      System.out.println(i);
        String Palabra = Palabras.elementAt(i).toString();
        int T2 = Palabra.length();
//      System.out.println("T2:"+T2);
        int [] pos=new int[T2];
        int j=0;
        for(int k=0; k<T2; k++)
        {
//          System.out.println(k);
            for (int cityIndex=0; cityIndex < tourSize(); cityIndex++)
                if(getCity(cityIndex).Letra==Palabra.charAt(k)){
//                    System.out.println("ñ");
                        pos[j]=cityIndex;
                        j++;
                    }
                }

//          System.out.println("jjjj: "+j);
//          for(int k=0; k<j; k++)
//              System.out.println("Posicion: "+pos[k]);
//          System.out.println("----");
            for (int cityIndex=0; cityIndex < j-1; cityIndex++) {
                int Primero = pos[cityIndex];
//                System.out.println(Primero);

                int Segundo = pos[cityIndex+1];
//                System.out.println(Segundo);
                Puntos fromCity = getCity(Primero);
```



```
        Puntos destinationCity;
        if(cityIndex+1 < T2){
            destinationCity = getCity(Segundo);
        }
        else{
            destinationCity = getCity(0);
        }
        tourDistance += fromCity.distanceTo(destinationCity);
//        System.out.println("-yyy-");
//        System.out.println("Distancia:"+tourDistance);
    }

}
distance = tourDistance;
}
return distance;
}

//obtiene el numero de ciudades de nuestro tour
public int tourSize() {
    return tour.size();
}

@Override
public String toString() {
    String geneString = " ";
    for (int i = 0; i < tourSize(); i++) {
        geneString += getCity(i)+" ";
    }
    return geneString;
}
}
```

SEGUIMIENTORECORRIDO.JAVA

```
package simulated;

import java.util.ArrayList;

public class SeguimientoRecorrido {

    private static ArrayList destinationCities = new ArrayList<Puntos>();

    public static void addCity(Puntos city) {

        destinationCities.add(city);

    }

    public static Puntos getCity(int index){

        return (Puntos)destinationCities.get(index);

    }

    public static int numberOfCities(){

        return destinationCities.size();

    }

}
```

SIMULATEDANNEALING.JAVA

```
package simulated;

import java.util.ArrayList;

import java.util.Vector;

import java.util.*;

import java.io.*;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

public class SimulatedAnnealing {

    static Vector Palabras = new Vector();

    static int T1;
```

```
public static double ProbabilidadAceptacion(double Energia, double NuevaEnergia, double
Temperatura) {
    if (NuevaEnergia < Energia) {
        return 1.0;
    }
    return Math.exp((Energia - NuevaEnergia) / Temperatura);
}

public static void main(String[] args) throws FileNotFoundException, IOException {
    File file = null;
    FileReader fr = null;
    BufferedReader br = null;
    String line;
//    Palabras.addElement("amor parlamento");
    file = new File ("D:\\Dropbox\\TesisDoctoradoTeclado\\textos\\Texto1.txt");
    fr = new FileReader (file);
    br = new BufferedReader (fr);
    while ((line = br.readLine ())!= null)
        Palabras.addElement(line);

    file = new File ("D:\\Dropbox\\TesisDoctoradoTeclado\\textos\\Texto2.txt");
    fr = new FileReader (file);
    br = new BufferedReader (fr);
    while ((line = br.readLine ())!= null)
        Palabras.addElement(line);

    file = new File ("D:\\Dropbox\\TesisDoctoradoTeclado\\textos\\Texto3.txt");
    fr = new FileReader (file);
    br = new BufferedReader (fr);
    while ((line = br.readLine ())!= null)
        Palabras.addElement(line);

    file = new File ("D:\\Dropbox\\TesisDoctoradoTeclado\\textos\\Texto4.txt");
```



```
fr = new FileReader (file);  
  
br = new BufferedReader (fr);  
  
while ((line = br.readLine ())!= null)  
    Palabras.addElement(line);  
  
file = new File ("D:\\Dropbox\\TesisDoctoradoTeclado\\textos\\Texto5.txt");  
  
fr = new FileReader (file);  
  
br = new BufferedReader (fr);  
  
while ((line = br.readLine ())!= null)  
    Palabras.addElement(line);  
  
file = new File ("D:\\Dropbox\\TesisDoctoradoTeclado\\textos\\Texto6.txt");  
  
fr = new FileReader (file);  
  
br = new BufferedReader (fr);  
  
while ((line = br.readLine ())!= null)  
    Palabras.addElement(line);  
  
file = new File ("D:\\Dropbox\\TesisDoctoradoTeclado\\textos\\Texto7.txt");  
  
fr = new FileReader (file);  
  
br = new BufferedReader (fr);  
  
while ((line = br.readLine ())!= null)  
    Palabras.addElement(line);  
  
file = new File ("D:\\Dropbox\\TesisDoctoradoTeclado\\textos\\Texto8.txt");  
  
fr = new FileReader (file);  
  
br = new BufferedReader (fr);  
  
while ((line = br.readLine ())!= null)  
    Palabras.addElement(line);  
  
  
Puntos ciudad1 = new Puntos(1,'q',1, 1);    SeguimientoRecorrido.addCity(ciudad1);  
Puntos ciudad2 = new Puntos(2,'w',2, 1);    SeguimientoRecorrido.addCity(ciudad2);  
Puntos ciudad3 = new Puntos(3,'e',3,1);    SeguimientoRecorrido.addCity(ciudad3);  
Puntos ciudad4 = new Puntos(4,'r',4, 1);    SeguimientoRecorrido.addCity(ciudad4);  
Puntos ciudad5 = new Puntos(5,'t',5, 1);    SeguimientoRecorrido.addCity(ciudad5);  
Puntos ciudad6 = new Puntos(6,'y',6, 1);    SeguimientoRecorrido.addCity(ciudad6);
```

```
Puntos ciudad7 = new Puntos(7,'u',7, 1);    SeguimientoRecorrido.addCity(ciudad7);
Puntos ciudad8 = new Puntos(8,'i',8, 1);    SeguimientoRecorrido.addCity(ciudad8);
Puntos ciudad9 = new Puntos(9,'o',9, 1);    SeguimientoRecorrido.addCity(ciudad9);
Puntos ciudad10 = new Puntos(10,'p',10, 1);  SeguimientoRecorrido.addCity(ciudad10);

Puntos ciudad11 = new Puntos(11,'a',1, 2);   SeguimientoRecorrido.addCity(ciudad11);
Puntos ciudad12 = new Puntos(12,'s',2, 2);   SeguimientoRecorrido.addCity(ciudad12);
Puntos ciudad13 = new Puntos(13,'d',3, 2);   SeguimientoRecorrido.addCity(ciudad13);
Puntos ciudad14 = new Puntos(14,'f',4, 2);   SeguimientoRecorrido.addCity(ciudad14);
Puntos ciudad15 = new Puntos(15,'g',5, 2);   SeguimientoRecorrido.addCity(ciudad15);
Puntos ciudad16 = new Puntos(16,'h',6, 2);   SeguimientoRecorrido.addCity(ciudad16);
Puntos ciudad17 = new Puntos(17,'j',7,2);    SeguimientoRecorrido.addCity(ciudad17);
Puntos ciudad18 = new Puntos(18,'k',8, 2);   SeguimientoRecorrido.addCity(ciudad18);
Puntos ciudad19 = new Puntos(19,'l',9, 2);   SeguimientoRecorrido.addCity(ciudad19);
Puntos ciudad20 = new Puntos(20,'ñ',10, 2);  SeguimientoRecorrido.addCity(ciudad20);

Puntos ciudad21 = new Puntos(21,'z',2, 3);   SeguimientoRecorrido.addCity(ciudad21);
Puntos ciudad22 = new Puntos(22,'x',3, 3);   SeguimientoRecorrido.addCity(ciudad22);
Puntos ciudad23 = new Puntos(23,'c',4, 3);   SeguimientoRecorrido.addCity(ciudad23);
Puntos ciudad24 = new Puntos(24,'v',5, 3);   SeguimientoRecorrido.addCity(ciudad24);
Puntos ciudad25 = new Puntos(25,'b',6, 3);   SeguimientoRecorrido.addCity(ciudad25);
Puntos ciudad26 = new Puntos(26,'n',7, 3);   SeguimientoRecorrido.addCity(ciudad26);
Puntos ciudad27 = new Puntos(27,'m',8,3);    SeguimientoRecorrido.addCity(ciudad27);

double temp = 10000;
double VelocidadEnfriamiento = 0.001;
String ruta = "D:\\Dropbox\\TesisDoctoradoTeclado\\Simulated2 Sustentar\\salida.txt";
File archivo = new File(ruta);
BufferedWriter text;
text = new BufferedWriter(new FileWriter(archivo));
```

```
Recorrido SolucionActual = new Recorrido();

SolucionActual.generateIndividual();

double EnergiaActual = SolucionActual.getDistance(Palabras);

// System.out.print("A inici: " + SolucionActual);

// System.out.println("A dist: " + EnergiaActual);

Recorrido NuevaSolucion = new Recorrido(SolucionActual.getTour());

Recorrido ElMejor = new Recorrido(SolucionActual.getTour());

//text.write("Mejor: " + ElMejor+"\n");

double DMejor = EnergiaActual;

// System.out.println(" ----- antes de while");

int Iteration=1,cont1=1,cont2=1;

String Mejor=" "+cont2+" "+Iteration+" "+ElMejor.toString()+DMejor+"\n";

String Actual = " "+cont1+" "+Iteration+" "+SolucionActual.toString()+EnergiaActual+"\n";

String Teclado="";

//text.write("Mejor: " + ElMejor+"\n");

while (temp > 1) {

    int tourPos1 = (int) (NuevaSolucion.tourSize() * Math.random());

    int tourPos2 = (int) (NuevaSolucion.tourSize() * Math.random());

    while(tourPos2==tourPos1)

        tourPos2 = (int) (NuevaSolucion.tourSize() * Math.random());

    text.write("iteracion:"+Iteration+" cambio:("+tourPos1+1)+","+(tourPos2+1)+") \n");

    text.write("Actual: " + SolucionActual);

    text.write(" "+SolucionActual.getDistance(Palabras)+"\n");
```

```
//ElMejor = SolucionActual;
NuevaSolucion = new Recorrido(SolucionActual.getTour());

// System.out.print("it: " + Iteration+" ");
// System.out.print("Pos1: " + (tourPos1+1)+" ");
// System.out.println("Pos1: " + (tourPos2+1)+" ");

Puntos citySwap1 = NuevaSolucion.getCity(tourPos1);
Puntos citySwap2 = NuevaSolucion.getCity(tourPos2);
char x = citySwap1.Letra;
char y = citySwap2.Letra;
citySwap1.Letra =y;
citySwap2.Letra=x;

// NuevaSolucion.Cambio(tourPos1, citySwap1,citySwap2.Letra);
// NuevaSolucion.Cambio(tourPos2, citySwap2,citySwap1.Letra);

// NuevaSolucion.setCity(tourPos2, citySwap1);
// NuevaSolucion.setCity(tourPos1, citySwap2);
text.write("Nueva: " + NuevaSolucion);
text.write(""+NuevaSolucion.getDistance(Palabras)+"\n");
double EnergiaVecindad = NuevaSolucion.getDistance(Palabras);
// System.out.println("NS: " + NuevaSolucion);
// System.out.println("EA: " + EnergiaActual);
// System.out.println("EV: " + EnergiaVecindad);
Iteration++;
double Prob = ProbabilidadAceptacion(EnergiaActual, EnergiaVecindad, temp);
double Rand = Math.random();
if (Prob > Rand) {
    cont1++;
    SolucionActual = new Recorrido(NuevaSolucion.getTour());
}
```

```
        EnergiaActual = EnergiaVecindad;

        Actual = Actual+" "+cont1+" "+Iteration+" "+SolucionActual.toString()+"
"+EnergiaActual+"\n";

    }

    if (EnergiaActual < DMejor) {

        cont2++;

        ElMejor = new Recorrido(SolucionActual.getTour());

        DMejor = EnergiaActual;

        Mejor =Mejor+" "+cont2+" "+Iteration+" "+ElMejor.toString()+" "+DMejor+"\n";

    }

    //text.write("Mejor: " + ElMejor);

    //text.write(""+ElMejor.getDistance(Palabras)+"\n");

    temp *= 1-VelocidadEnfriamiento;

}

text.write("Reemplazos: \n" + Actual+"\n");
text.write("Tdo M: \n" + Mejor+"\n");
text.write("Iteracion: " + (Iteration-1)+"\n");
text.write("Reemplazos Solución actual: " + cont1+"\n");
text.write("Reemplazos El mejor: " + cont2+"\n");
text.close();

//    for (int i=0; i<3; i++)
//        for(int j=0; j<9; j++)
//            Mejor.
//
//    }
//
//String Palabra = Palabras.elementAt(i).toString();
```