# **ESTRUCTURA DE DATOS PARA LA CLASIFICACIÓN Y OPTIMIZACIÓN APLICADA A LA GANADERÍA DE PRECISIÓN**

|  |  |  |  |
| --- | --- | --- | --- |
| María Antonia Velásquez  Universidad EAFIT  Colombia  mavelasqur@eafit.edu.co | Diego Alexander Múnera  Universidad EAFIT  Colombia  damunerat@eafit.edu.co | Simón Marín Universidad Eafit Colombia smaring1@eafit.edu.co | Mauricio Toro  Universidad Eafit  Colombia  [mtorobe@eafit.edu.co](mailto:mtorobe@eafit.edu.co) |

# **RESUMEN**

## Desarrollar un método óptimo mediante el cual pueda llevarse a cabo el control de calidad y análisis del proceso ganadero de manera precisa y automatizada realizando un estudio de la apariencia del ganado que convertido en fragmentos de texto más detallados, es decir, sometido a descompresión permita la clasificación de este con el más mínimo margen de error

## Esta alternativa permitirá un estudio más acertado a la hora de evaluar el proceso ganadero en las granjas para conocer el ganado en buen y mal estado lo cual es indispensable tratándose del consumo humano de este y de crear tal alternativa tomando en cuenta el detalle más mínimo puede acertar en el proceso más que el ojo humano para además optimizar tiempo, materia prima y errores

## **Palabras clave**

|  |
| --- |
| Algoritmos de compresión, aprendizaje de máquina,  aprendizaje profundo, ganadería de precisión, salud animal. |

# **1. INTRODUCCIÓN**

# La actividad ganadera es una de las más básicas e importantes en el desarrollo de la vida diaria pues de allí provienen productos como leche y carne para el consumo humano.

# Debido a esto, es de vital importancia cuidar el estado del ganado durante su desarrollo y proceso de producción que terminará en el mercado.

# Y es por esto que, la implementación de tecnologías inteligentes, la inteligencia artificial y el lenguaje de máquina nos brindaría un estudio minucioso del estado del ganado durante todo su proceso de producción permitiendo además que este proceso dure menos tiempo y tenga menos fallas en su análisis.

# **Problema**

Comprimir las imágenes de ganado para clasificar  
la salud animal en el contexto de la ganadería de  
precisión permitiría que el ganado en mal estado pueda  
quedar fuera de consideración para la producción y distribución del producto final con un mayor índice de acertación y optimización del tiempo. Con esto a su vez, se evitarían enfermedades en los humanos debido a la carne en mal estado que producirían vacas enfermas que no fueron clasificadas correctamente.

**1.2 Solución**

En este trabajo, utilizamos una red neuronal convolucional para clasificar la salud animal, en el ganado vacuno, en el contexto de la ganadería de precisión (GdP). Un problema común en la GdP es que la infraestructura de la red es muy limitada, por lo que se requiere la compresión de los datos.

*Expliquen*, brevemente, su solución al problema *(En este semestre, la solución es una implementación de algoritmos de compresión. ¿Qué algoritmos han elegido? ¿Por qué?)*

**1.3 Estructura del artículo**

En lo que sigue, en la Sección 2, presentamos trabajos relacional con el problema. Más adelante, en la Ssección 3, presentamos los conjuntos de datos y los métodos utilizados en esta investigación. En la Sección 4, presentamos el diseño del algoritmo. Después, en la Sección 5, presentamos los resultados. Finalmente, en la Sección 6, discutimos los resultados y proponemos algunas direcciones de trabajo futuras.

**2. TRABAJOS RELACIONADOS**

## En lo que sigue, explicamos cuatro trabajos relacionados. en el dominio de la clasificación de la salud animal y la compresión de datos. en el contexto del PLF.

## Explique cuatro (4) artículos relacionados con el problema descrito en la sección 1.1. Puede encontrar los problemas relacionados en las revistas científicas, en lo posible, en inglés. Considere Google Scholar para su búsqueda. *(En este semestre, el trabajo relacionado es la investigación sobre la clasificación de la salud animal y la compresión de datos, en el contexto de la GdP).*

**3.1An Animal Welfare Platform for Extensive Livestok Production Systems.**

En este artículo se plantea una solución para hacer seguimiento del cambio en el comportamiento de los animales de la ganadería ya que así

es posible conocer el estado de bienestar del animal lo cual satisfacerá la demanda hecha por los consumidores de que sus alimentos

se produzcan y procesen con mayor conciencia y respeto por el bienestar de los animales.

Se implementó un prototipo de dispositivo personalizado, que sería llevado por

el animal en todo momento. El dispositivo se diseñó para colocarse en el cuello de un animal rumiante, como un collar normal, como

esto es fácil de manejar y no introduciría restricciones desconocidas para el animal. El dispositivo debe

ser capaz de recopilar datos de un acelerómetro y un giroscopio para el reconocimiento de actividad, manteniendo también la capacidad

para geo localizar al animal en tiempo real.

El estudio presentó una solución para el seguimiento de la actividad y el comportamiento de los animales en el ganado

obteniendo indicadores que sustentan el bienestar animal. La solución explota un solo tipo de sensor inalámbrico

(dispositivo de collar) para registrar la actividad de los animales (es decir, movimiento, velocidad, información de geolocalización) con baja implementación

y costo, dispositivos de computación de borde con capacidades computacionales, capaces de realizar datos en tiempo real y sin conexión

para el reconocimiento de patrones a través de algoritmos de redes neuronales profundas, computación en la nube para ambos datos

y almacenamiento de modelos de aprendizaje profundo, y visualizaciones utilizables y efectivas en dispositivos móviles que proporcionan a los

usuarios finales (agricultores) información valiosa.

**3.2 Visual Localisation and Individual Identification**

**of Holstein Friesian Cattle via Deep Learning**

En este artículo se propone que la identificación individual del ganado Holstein Friesian puede

ocurrir de forma automática y no intrusiva para detectar/localizar esta especie, ayudar en la eficiencia

dentro de la granja y contribuir al bienestar animal.

Se llevó a cabo utilizando visión artificial.

Tuberías alimentadas por arquitecturas estándar que utilizan

redes neuronales profundas y canalización de monitoreo de video.

La detección Friesian y la identificación individual se pueden

abordar a través de canalizaciones de aprendizaje profundo estándar.

Para identificación en particular, se mostró que las arquitecturas basadas en convoluciones

están bien adaptadas para aprender y distinguir las propiedades del patrón y la estructura dorsal

exhibidos por la especie individualmente. Es importante destacar que este proceso

puede tener lugar de forma no intrusiva en prácticas relevantes, en contraste con la mayora de la

identificación existente.

**3.3 Cloud services integration for farm animals’**

**behavior studies based on smartphones as activity sensors**

En este artículo se propone una arquitectura de nube lambda, una plataforma utilizada para archivar, almacenar y procesar datos

de alta frecuencia para integrar desarrollos futuros de Internet de las Cosas aplicadas al seguimiento de animales domésticos.

El comportamiento del ganado se estudia mediante sensores colocadosen el animal.

Los datos medidos por los sensores se pueden utilizar para calcular otros

parámetros derivados que se envían mediante el protocolo UDP a través de Wi-Fi

a la plataforma Cloud. los datos se guardan localmente en un archivo CSV, que luego se transmite

y es procesado en una puerta de enlace antes de ser transmitido a la plataforma en la nube.

Esta arquitectura lambda fue capaz de recopilar datos con alta frecuencia y fue adaptable

fácilmente a muchos casos.

Esta arquitectura también fue capaz de integrar datos complementarios como imágenes de UAV

y datos externos de otra nube como plataformas de salud y producción.

**3.4 A systematic literature review on the use of machine learning in precision livestock farming**

Este artículo presenta una revisión sistemática de la literatura de trabajos recientes sobre el uso del aprendizaje automático (ML) en

ganadería de precisión (PLF), centrándose en dos áreas de interés: pastoreo y sanidad animal.

Entre los métodos aplicados en este estudio del uso de ML están: la clasificación del comportamiento animal,

una primera categoría describe los documentos que se centran en cómo clasificar el comportamiento del ganado en general,

Una segunda categoría presenta documentos específicamente sobre la clasificación del consumo de hierba y actividades de rumia.

Siguiente a esto, una sección se centra en documentos de monitoreo animal.

Trayendo juntos el pastoreo y la salud animal también permitirán desarrollar modelos de

diagnóstico y prescripción como herramientas para la prevención y control de

enfermedades animales. Estas herramientas supondrán considerables beneficios para el ganado

producción, seguridad alimentaria, salud pública, salud animal y acceso a mercados.

## **3. MATERIALES Y MÉTODOS**

En esta sección, explicamos cómo se recogieron y procesaron los datos y, después, diferentes alternativas de algoritmos de compresión de imágenes para mejorar la clasificación de la salud animal.

## **3.1 Recopilación y procesamiento de datos**

Recogimos datos de *Google Images* y *Bing Images* divididos en dos grupos: ganado sano y ganado enfermo. Para el ganado sano, la cadena de búsqueda era "cow". Para el ganado enfermo, la cadena de búsqueda era "cow + sick".

En el siguiente paso, ambos grupos de imágenes fueron transformadas a escala de grises usando Python OpenCV y fueron transformadas en archivos de valores separados por comas (en inglés, CSV). Los conjuntos de datos estaban equilibrados.

El conjunto de datos se dividió en un 70% para entrenamiento y un 30% para pruebas. Los conjuntos de datos están disponibles en https://github.com/mauriciotoro/ST0245-Eafit/tree/master/proyecto/datasets .

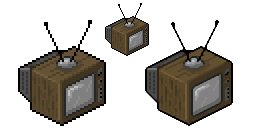
Por último, utilizando el conjunto de datos de entrenamiento, entrenamos una red neuronal convolucional para la clasificación binaria de imágenes utilizando *Teachable Machine* de Google disponible en https://teachablemachine.withgoogle.com/train/image.

## **3.2 Alternativas de compresión de imágenes con pérdida**

## A continuación, serán presentados diferentes algoritmos existentes usados para comprimir imágenes con perdida de algunos parametros para reducir la memoria utilizada.

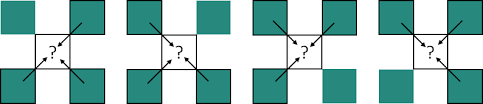
**3.2.1 Nearest-neighbor interpolation**

Este algoritmo está enfocado en reemplazar cada pixel con el que este más cercano en la salida para aumentar la escala de la imagen. A partir de esto, se introducen irregularidades en las imagenes que fuesen suaves antes de la codificación. La complejidad del algoritmo está enfocada en que el vecino mas cercano no se refiere como tal a este matematicamente sino, un redondeo, haciendo más facil de calcular el tamaño.



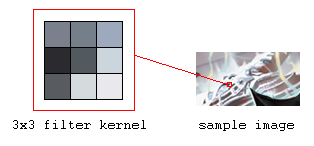
**3.2.2 Bilinear and bicubic algorithm**

Este algoritmo parte de su funcionalidad interpolando los valores de color en cada pixel, introduciendo una transición continua en la salida. Este algoritmo logra reducir los bordes nitidos de la imagen. La complejidad de este algoritmo surge a partir de sus dos formas de ejecutarse, bilineal y bicubicamente; una genera la imagen con bordes pixelados y la otra con bordes lisos pero con un costo computacional alto.



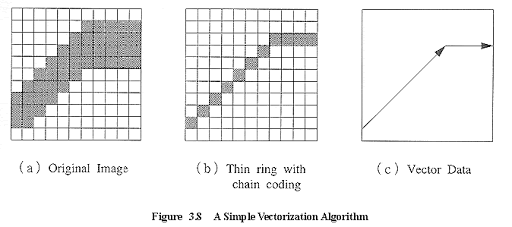
**3.2.3 Box sampling**

Este algoritmo consiste en tomar el pixel destino como una ‘caja’ en la imagen original y luego mapear todos los pixeles dentro de la caja. Esto con el fin de que al convertir de nuevo la imagen, todos los pixels contribuyan a la salida. La complejidad de ese algoritmo radicó en eliminar los errores o las debilidades de algoritmos como ‘Sinc and Lanczos sampling’ con su baja optimización de pixeles.



**3.2.4 Vectorization**

Este algoritmo extrae las representaciones visuals de la resolucion del grafico a comprimir como vectores independientes y luego, esta version se representa como una imagen rasterizada con una resolución optima.

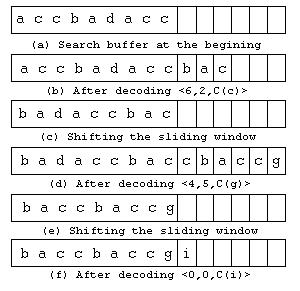


## **3.3 Alternativas de compresión de imágenes sin pérdida**

## En lo que sigue, presentamos diferentes algoritmos usados para comprimir imágenes sin pérdida.

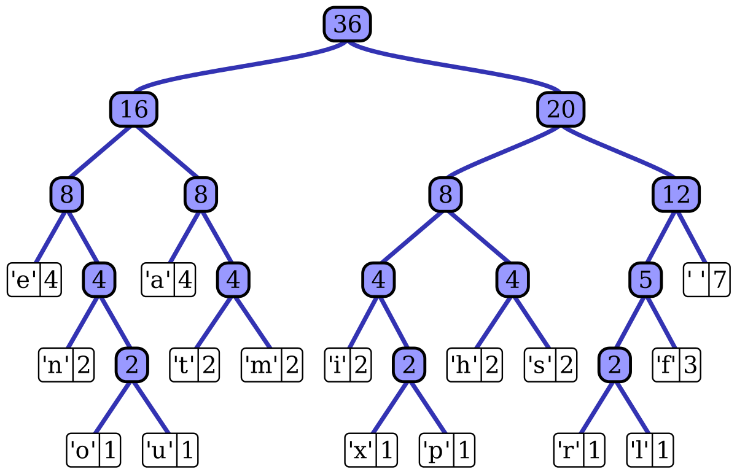
**3.3.1 LZ77\_and\_LZ78**

Este algoritmo logra la compresion de imagenes reemplazando las ocurrencias repetidas de datos con referencias en memoria a una sola copia de esos datos existentes anteriormente en el flujo de datos sin comprimir. La complejidad de este algoritmo radica en que para encontrar las coincidencias se codifican dos números que, están llamados longitude-distancia y estos denominan el desplazamiento en el buffer del flujo de datos sin comprimir.



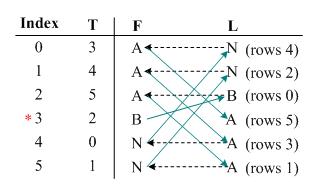
**3.3.2 Huffman Compression Algorithm**

Este algoritmo funciona creando un arbol binario de nodos en los que se puede almacenar un vector regular, cuyo tamaño depende del numero de símbolos n. Cada nodo puede ser un nodo hoja o un nodo interno. En cada uno de estos aparece inicialmente su simbolo y su frecuencia de aparición. Así que, las letras se ordenan aumentando la frecuencia y las menos frecuentes en cada paso se combinan y reinsertan construyendo un arbol parcial.



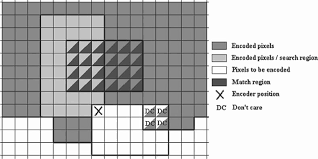
**3.3.3 Burrows Wheeler transform**

Este algoritmo trabaja con el concepto de compresion por clasificacion de bloques, éste reorganiza una cadena de caracteres en series de estos mismos similares. Este tipo de algoritmo es util y en esto a su vez radica su complejidad; en la codificación de cadenas con longitudes repetidas y su reversibilidad.



**3.3.4 GS-2D-LZ**

Este algoritmo de compresión esta enfocado en codificar por orden de procesamiento un bloque por pixel en cada paso. Luego un valor aproximado se busca en la imagen previa a la compresión y el bloque es señalado como un puntero de la localización de su igual, sus dimensiones y su información residuo. Mediante este método se asegura que la compression no tendra perdidas.



## **4. DISEÑO E IMPLEMENTACIÓN DE LOS ALGORITMOS**

## En lo que sigue, explicamos las estructuras de datos y los algoritmos utilizados en este trabajo. Las implementaciones de las estructuras de datos y los algoritmos están disponibles en Github[[1]](#footnote-1).

## **4.1 Estructuras de datos**

## *Explique* la estructura de datos utilizada para hacer la compresión de las imágenes y haga una figura que la explique. No utilice figuras de Internet. *(En este semestre, ejemplo de las estructuras de datos son los árboles y las tablas hash)*

**Figura 1:** Árbolde Huffman generado a partir de las frecuencias exactas del texto "this". (Por favor, no dude en cambiar esta figura si utilizan una estructura de datos diferente).

**4.2 Algoritmos**

En este trabajo, proponemos un algoritmo de compresión que es una combinación de un algoritmo de compresión de imágenes con pérdidas y un algoritmo de compresión de imágenes sin pérdidas. También explicamos cómo funciona la descompresión para el algoritmo propuesto.

Expliquen el diseño de los algoritmos para resolver el problema y hagan una figura. No uses figuras de Internet, haz las tuyas propias. *(En este semestre, un algoritmo debe ser un algoritmo de compresión de imágenes con pérdidas, como el escalado de imágenes, el tallado de costuras o la compresión con ondeletas, y el segundo algoritmo debe ser un algoritmo de compresión de imágenes sin pérdidas, como la codificación Huffman, LZS o LZ77).*

**4.2.1 Algoritmo de compresión de imágenes con pérdida**

Explique, brevemente, cómo se aplicó un algoritmo de compresión de imágenes con pérdidas, como, por ejemplo, el tallado de costuras o el escalado de imágenes. Explique también la descompresión.

**Figura 2:** Escaladode la imagen mediante interpolación bilineal. (Por favor, siéntase libre de cambiar esta figura si utiliza una estructura de datos diferente).

**4.2.2 Algoritmo de compresión de imágenes sin pérdida**

Explique brevemente cómo aplicó un algoritmo de compresión de imágenes sin pérdidas como la codificación Huffman, LZS o LZ77. Explique también la descompresión.

**4.3 Análisis de la complejidad de los algoritmos**

Explique, con sus propias palabras, el análisis del peor caso usando la notación O. ¿Cómo calculó tales complejidades. Por favor, explique brevemente.

|  |  |
| --- | --- |
| **Algoritmo** | **La complejidad del tiempo** |
| Compresión | O(N2\*M2) |
| Descompresión | O(N3\*M\*2N) |

***Tabla* 2:** Complejidad temporal de los algoritmos de compresión y descompresión de imágenes. *(Por favor, explique qué significan N y M en este problema).*

|  |  |
| --- | --- |
| **Algoritmo** | **Complejidad de la memoria** |
| Compresión | O(N\*M\*2N ) |
| Descompresión | O(2M\*2N) |

***Tabla* 3:** Complejidad de memoria de los algoritmos de compresión y descompresión de imágenes. *(Por favor, explique qué significan N y M en este problema).*

**4.4 Criterios de diseño del algoritmo**

Explica por qué el algoritmo fue diseñado de esa manera. Use un criterio objetivo. Los criterios objetivos se basan en la eficiencia, que se mide en términos de tiempo y consumo de memoria. Ejemplos de criterios no objetivos son: "Estaba enfermo", "fue la primera estructura de datos que encontré en Internet", "lo hice el último día antes del plazo", etc. Recuerde: Este es el 40% de la calificación del proyecto.

**5. RESULTADOS**

**5.1 Evaluación del modelo**

En esta sección, presentamos algunas métricas para evaluar el modelo. La exactitud es la relación entre el número de predicciones correctas y el número total de muestras de entrada. La precisión es la proporción de estudiantes exitosos identificados correctamente por el modelo a estudiantes exitosos identificados por el modelo. Por último, sensibilidad es la proporción de estudiantes exitosos identificados correctamente por el modelo a estudiantes exitosos en el conjunto de datos.

**5.1.1 Evaluación del conjunto de datos de entrenamiento**

A continuación presentamos las métricas de evaluación del conjunto de datos de entrenamiento en la Tabla 3.

|  |  |
| --- | --- |
|  | ***Conjunto de datos de entrenamiento*** |
| *Precisión* | 0.02 |
| *Precisión* | 0.03 |
| *Recordar* | 0.01 |

## **Tabla 3.** Evaluación del modelo de clasificación de imágenes con el conjunto de datos de entrenamiento.

**5.1.2 Evaluación del conjuntos de datos de prueba**

A continuación presentamos las métricas de evaluación del conjunto de datos de prueba, en la Tabla 4, sin compresión y, en la Tabla 5, con compresión.

|  |  |
| --- | --- |
|  | ***Conjunto de datos de prueba*** |
| *Exactitud* | 0.01 |
| *Precisión* | 0.012 |
| *Sensibilidad* | 0.013 |

## **Tabla 4.** Evaluación del modelo de clasificación de imágenes, con el conjunto de datos de prueba, sin compresión.

|  |  |
| --- | --- |
|  | ***Conjunto de datos de prueba*** |
| *Exactitud* | 0.001 |
| *Precisión* | 0.0012 |
| *Sensibilidad* | 0.0013 |

## **Tabla 5.** Evaluación del modelo de clasificación de imágenes, con el conjunto de datos de prueba, con compresión.

**5.2 Tiempos de ejecución**

En lo que sigue explicamos la relación entre el tiempo promedio de ejecución y el tamaño promedio de las imágenes del conjunto de datos completo, en la Tabla 6.

Calcular el tiempo de ejecución de cada imagen en Github. Informar del tiempo medio de ejecución vs. el tamaño medio del archivo.

## 

|  |  |  |
| --- | --- | --- |
|  | ***Tiempo promedio de ejecución (s)*** | ***Tamaño promedio del archivo (MB)*** |
| *Compresión* | 100.2 s | 12.4 MB |
| *Descompresión* | 800.1 s | 12.4 MB |

## **Tabla 6:** Tiempo deejecución de los algoritmos *(Por favor, escriba el nombre de los algoritmos, por ejemplo, tallado de costuras y LZ77)* para diferentes imágenes en el conjunto de datos.

## **5.3 Consumo de memoria**

Presentamos el consumo de memoria de los algoritmos de compresión y descompresión en la Tabla 7.

|  |  |  |
| --- | --- | --- |
|  | ***Consumo promedio de memoria (MB)*** | ***Tamaño promedio del archivo (MB)*** |
| Compresión | 634 MB | 3.12 MB |
| Descompresión | 9 MB | 878.12 MB |

## **Tabla 7:** Consumopromedio de memoria de todas las imágenes del conjunto de datos, tanto para la compresión como para la descompresión.

## Para medir el consumo de memoria, deberían usar un generador de perfiles. Uno muy bueno para Java es VisualVM, desarrollado por Oracle, http://docs.oracle.com/javase/7/docs/technotes/guides/visualvm/profiler.html. Para Python, usa el *C profiler*.

## **5.3 Tasa de compresión**

Presentamos los resultados de la tasa de compresión del algoritmo en la Tabla 8.

|  |  |  |
| --- | --- | --- |
|  | ***Ganado sano*** | ***Ganado enfermo*** |
| Tasa de compresión promedio | 1:23 | 1:34 |

## **Tabla 8:** Promedio redondeado de la tasa de compresión de todas las imágenes de ganado sano y ganado enfermo.

## **6. DISCUSIÓN DE LOS RESULTADOS**

Explique los resultados obtenidos. ¿Son la exactitud, la precisión y la sensibilidad apropiadas para este problema? ¿El modelo está sobreajustado? ¿Es apropiado el consumo de memoria y el consumo de tiempo? ¿Es la relación de compresión apropiada? ¿Cambia la compresión significativamente la exactitud con el conjunto de datos de la prueba? *(En este semestre, según los resultados, ¿puede la compresión mejorar la clasificación de la salud animal en el contexto del PLF? )*

**6.1 Trabajos futuros**

Responda ¿qué le gustaría mejorar en el futuro? ¿Cómo le gustaría mejorar su algoritmo y su implementación? ¿Qué tal usar la transformación de coseno discreto o la compresión con ondeletas a futuro?

# **RECONOCIMIENTOS**

Identifique el tipo de reconocimiento que quiere escribir: para una persona o para una institución. Considere las siguientes pautas: 1. El nombre del profesor no se menciona porque es un autor. 2. No debe mencionar sitios web de autores de artículos que no ha contactado. 3. Debe mencionar estudiantes, profesores de otros cursos que le hayan ayudado.

Como ejemplo: Esta investigación fue apoyada/parcialmente apoyada por [Nombre de la Fundación, Donante, Beca].

Agradecemos la asistencia con [técnica particular, metodología] a [Nombre Apellido, cargo, nombre de la institución] por los comentarios que mejoraron enormemente el manuscrito o la codificación del algoritmo.

# **REFERENCIAS**

La referencia se obtiene usando el formato de referencia ACM. Lea las directrices de ACM en <http://bit.ly/2pZnE5g>

A modo de ejemplo, consideremos estas dos referencias:

1.Adobe Acrobat Reader 7, Asegúrate de que el texto de las secciones de referencia es *Ragged Right*, Not Justified. http://www.adobe.com/products/acrobat/.

2. Fischer, G. y Nakakoji, K. Amplificando la creatividad de los diseñadores con entornos de diseño orientados al dominio. en Dartnall, T. ed. Artificial Intelligence and Creativity: An Interdisciplinary Approach, Kluwer Academic Publishers, Dordrecht, 1994, 343-364.

Por favor, quite las referencias de arriba, son sólo un ejemplo.

1. http://www.github.com/ ????????? /proyecto/ [↑](#footnote-ref-1)