

```

INFO:hf-to-gguf:Set model quantization version
INFO:gguf.gguf_writer:Writing the following files:
INFO:gguf.gguf_writer:modelo_futbol_gguf\temp.gguf: n_tensors = 255, total_size = 6.4G
Writing: 100% | 6.43G/6.43G [00:09<00:00, 702Mbyte/s]
INFO:hf-to-gguf:Model successfully exported to modelo_futbol_gguf\temp.gguf

```

En la primera imagen se muestra la instalación y configuración del entorno de trabajo. Se observan comandos en la terminal para instalar librerías necesarias para trabajar con modelos de lenguaje. Esta parte del proceso consiste en preparar el sistema con las dependencias requeridas para poder entrenar y ejecutar el modelo correctamente.

```

>> [
>>   {"instruction": "¿Quién ganó el último Mundial de fútbol?", "input": "", "output": "Argentina ganó el Mundial de Qatar 2022, derrotando a Francia en la final por penaltis tras un empate 3-3. Fue el tercer título mundial para la selección argentina."}
>>   {"instruction": "¿Cuántos Balones de Oro tiene Messi?", "input": "", "output": "Lionel Messi ha ganado 8 Balones de Oro (2009, 2010, 2011, 2012, 2015, 2019, 2021 y 2023), siendo el jugador con más galardones en la historia del premio."}
>>   {"instruction": "Explica el fuera de juego.", "input": "", "output": "El fuera de juego (offside) se produce cuando un jugador está más cerca de la línea de gol rival que el balón y el penúltimo defensor en el momento de recibir el pase. No es falta si está en su propio campo o hay al menos dos jugadores rivales entre él y la portería."}
>>   {"instruction": "¿Qué es el equipo con más Champions League?", "input": "", "output": "El Real Madrid es el equipo con más títulos de Champions League (o Copa de Europa), con 15 trofeos. Le siguen el AC Milan con 7 y el Liverpool con 6."}
>>   {"instruction": "Posición del portero en un córner defensivo.", "input": "", "output": "En un córner defensivo, el portero debe posicionarse en el centro de la portería, ligeramente adelantado para reducir el ángulo de lanzamiento, listo para salir a por el balón si es necesario o quedarse en la línea para una parada."}
>>   {"instruction": "¿Qué es el pressing alto?", "input": "", "output": "El pressing alto es una táctica defensiva donde el equipo presiona al rival en su propio campo, cerca de la zona de salida de balón, buscando recuperar la posesión."}
>>   {"instruction": "Regla del fuera de juego modificada en 2023.", "input": "", "output": "En 2023, la IFAB modificó la regla del fuera de juego: ahora se considera posición adelantada si cualquier parte del cuerpo que puede marcar gol está por delante del balón y el penúltimo defensor. Anteriormente se contaba cualquier parte del cuerpo."}
>>   {"instruction": "¿Cuántos jugadores pueden estar en el campo en un partido oficial?", "input": "", "output": "Un equipo de fútbol tiene 11 jugadores en el campo durante un partido oficial: 1 portero y 10 jugadores de campo (defensas, centrocampistas y delanteros). Los equipos pueden hacer hasta 5 sustituciones según las reglas actuales de la FIFA."}
>>   {"instruction": "Duración de un partido de fútbol profesional.", "input": "", "output": "Un partido de fútbol profesional consta de dos tiempos de 45 minutos cada uno, con un descanso de 15 minutos entre ambos. El árbitro puede añadir tiempo de descuento al final de cada parte por interrupciones."}
>>   {"instruction": "¿Qué es el VAR (Video Assistant Referee)?", "input": "", "output": "El VAR (Video Assistant Referee) es un sistema de asistencia arbitral por video que ayuda al árbitro principal a revisar decisiones sobre goles, penaltis, tarjetas rojas directas y errores de identidad de jugadores sancionados."}
>>   {"instruction": "¿Quién es el máximo goleador de la historia del Real Madrid?", "input": "", "output": "Cristiano Ronaldo es el máximo goleador de la historia del Real Madrid con 451 goles en 438 partidos oficiales entre 2009 y 2018. Le sigue Karim Benzema con 354 goles."}
>>   {"instruction": "¿Cuándo se fundó el FC Barcelona?", "input": "", "output": "El FC Barcelona fue fundado el 29 de noviembre de 1899 por un grupo de jóvenes dirigidos por el suizo Joan Gamper. El club tiene su sede en Barcelona y es propiedad de sus socios."}
>>   {"instruction": "¿Qué significa FIFA?", "input": "", "output": "FIFA significa Fédération Internationale de Football Association (Federación Internacional de Fútbol Asociación). Es el organismo rector mundial del fútbol, fundado en París en 1904."}
>>   {"instruction": "¿Qué es el fair play?", "input": "", "output": "El fair play (juego limpio) es un código de conducta deportiva que promueve el respeto, la integridad y la deportividad. La FIFA otorga un premio anual al equipo o jugador que mejor representa estos valores."}
>>   {"@ | Out-File -FilePath "futbol_data.jsonl" -Encoding utf8"

```

En la segunda imagen se visualiza la carga de un modelo de lenguaje preentrenado. El sistema descarga o abre los archivos del modelo base desde una fuente específica. Este paso es necesario porque el fine-tuning no se realiza desde cero, sino que se parte de un modelo previamente entrenado.

```

Unsloth 2026.2.1 patched 28 layers with 28 QKV layers, 28 O layers and 28 MLP layers.
↳ Fusionando modelo base + adapters...
Found HuggingFace hub cache directory: C:\Users\harka\.cache\huggingface\hub
model.safetensors.index.json: 20.9kB [00:00, 39.7MB/s]
Checking cache directory for required files...
Cache check failed: model-00001-of-00002.safetensors not found in local cache.
Not all required files found in cache. Will proceed with downloading.
Checking cache directory for required files...
Cache check failed: tokenizer.model not found in local cache.
Not all required files found in cache. Will proceed with downloading.
model-00001-of-00002.safetensors: 100% | 4.97G/4.97G [00:43<00:00, 114MB/s]
model-00002-of-00002.safetensors: 100% | 1.46G/1.46G [00:13<00:00, 109MB/s]
Unsloth: Preparing safetensors model files: 100% | 2/2 [00:58<00:00, 29.05s/it]
Note: tokenizer.model not found (this is OK for non-SentencePiece models)
Unsloth: Merging weights into 16bit: 100% | 2/2 [00:08<00:00, 4.02s/it]
Unsloth: Merge process complete. Saved to `C:\LLM-Training\modelo_futbol_merged` 
☑ Modelo fusionado guardado en 'modelo_futbol_merged'

```

En la tercera captura se observa la carga y preparación de los datos que se utilizarán para entrenar el modelo. Aquí se organizan los textos en el formato adecuado para que el modelo pueda procesarlos. Esta fase es importante porque los datos determinan el conocimiento específico que aprenderá el modelo.

```

Unslot: Fast downloading is enabled - ignore downloading bars which are red colored!
Unslot 2026.2.1 patched 28 layers with 28 QKV layers, 28 O layers and 28 MLP layers.
Unslot: Tokenizing ["text": 100%] | 14/14 [00:00<00:00, 540.50 examples/s]
⌚ Entrenando modelo de fútbol...
==((==))= Unslot - 2x faster free finetuning | Num GPUs used = 1
\\ /| Num examples = 14 | Num Epochs = 50 | Total steps = 100
0^0/ \_ \ Batch size per device = 2 | Gradient accumulation steps = 4
\ \ / Data Parallel GPUs = 1 | Total batch size (2 x 4 x 1) = 8
"__" Trainable parameters = 24,313,856 of 3,237,063,680 (0.75% trained)
{'loss': 2.2616, 'grad_norm': 0.5982801914215088, 'learning_rate': 0.0, 'epoch': 0.57}
{'loss': 2.4315, 'grad_norm': 0.7219143509864807, 'learning_rate': 4e-05, 'epoch': 1.0}
{'loss': 2.2975, 'grad_norm': 0.6334890127182007, 'learning_rate': 8e-05, 'epoch': 1.57}
{'loss': 2.352, 'grad_norm': 0.7189835906028748, 'learning_rate': 0.00012, 'epoch': 2.0}
{'loss': 2.237, 'grad_norm': 0.8489921689033508, 'learning_rate': 0.00016, 'epoch': 2.57}
{'loss': 2.1664, 'grad_norm': 1.0354658365249634, 'learning_rate': 0.0002, 'epoch': 3.0}
{'loss': 2.1251, 'grad_norm': 1.3314117193222046, 'learning_rate': 0.00019789473684210526, 'epoch': 3.57}
{'loss': 1.5159, 'grad_norm': 1.272682547569275, 'learning_rate': 0.00019578947368421054, 'epoch': 4.0}
{'loss': 1.3606, 'grad_norm': 1.048068881034851, 'learning_rate': 0.0001936842105263158, 'epoch': 4.57}
{'loss': 1.5695, 'grad_norm': 1.4900039434432983, 'learning_rate': 0.00019157894736842104, 'epoch': 5.0}
{'loss': 1.1714, 'grad_norm': 1.267288040542603, 'learning_rate': 0.00019847368421052632, 'epoch': 5.57}
{'loss': 1.114, 'grad_norm': 1.026964545249939, 'learning_rate': 0.0001873684210526316, 'epoch': 6.0}
{'loss': 0.8624, 'grad_norm': 0.826546311378479, 'learning_rate': 0.00018526315789473685, 'epoch': 6.57}
{'loss': 0.9787, 'grad_norm': 1.2298682928085327, 'learning_rate': 0.0001831578947368421, 'epoch': 7.0}
{'loss': 0.7218, 'grad_norm': 0.9294831156730652, 'learning_rate': 0.00018105263157894739, 'epoch': 7.57}
{'loss': 0.6864, 'grad_norm': 1.1458503007888794, 'learning_rate': 0.00017894736842105264, 'epoch': 8.0}
{'loss': 0.5076, 'grad_norm': 0.9509326815605164, 'learning_rate': 0.0001768421052631579, 'epoch': 8.57}
{'loss': 0.537, 'grad_norm': 1.1099213361740112, 'learning_rate': 0.00017473684210526317, 'epoch': 9.0}
{'loss': 0.4301, 'grad_norm': 0.9897706508636475, 'learning_rate': 0.00017263157894736842, 'epoch': 9.57}
{'loss': 0.4248, 'grad_norm': 1.783308506011963, 'learning_rate': 0.0001705263157894737, 'epoch': 10.0}
{'loss': 0.3757, 'grad_norm': 1.398154377937317, 'learning_rate': 0.00016842105263157895, 'epoch': 10.57}
{'loss': 0.3412, 'grad_norm': 1.18347099644317627, 'learning_rate': 0.00016631578947368423, 'epoch': 11.0}
{'loss': 0.3446, 'grad_norm': 1.12060546875, 'learning_rate': 0.00016421052631578948, 'epoch': 11.57}
{'loss': 0.3144, 'grad_norm': 1.492128849029541, 'learning_rate': 0.00016210526315789473, 'epoch': 12.0}
{'loss': 0.3167, 'grad_norm': 1.3325821161270142, 'learning_rate': 0.00016, 'epoch': 12.57}

```

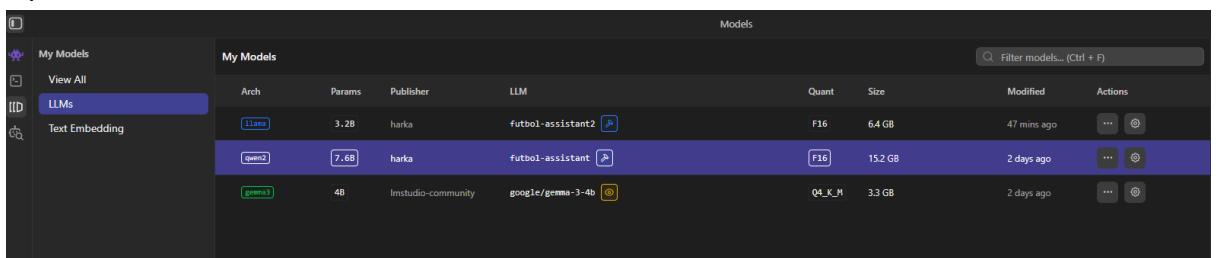
En la cuarta imagen se muestran los parámetros de entrenamiento, como el número de épocas (epochs), el tamaño del batch y la tasa de aprendizaje (learning rate). En este paso se definen las condiciones bajo las cuales el modelo ajustará sus pesos para adaptarse al nuevo dominio.

```

{'loss': 0.0019, 'grad_norm': 0.25631463527079443, 'learning_rate': 3.578947368421053e-05, 'epoch': 42.0}
{'loss': 0.0593, 'grad_norm': 1.8485994637012482, 'learning_rate': 3.368421052631579e-05, 'epoch': 42.57}
{'loss': 0.0588, 'grad_norm': 0.258191853761673, 'learning_rate': 3.157894736842105e-05, 'epoch': 43.0}
{'loss': 0.0577, 'grad_norm': 0.16142846643924713, 'learning_rate': 2.9473684210526314e-05, 'epoch': 43.57}
{'loss': 0.0599, 'grad_norm': 0.2578180134296417, 'learning_rate': 2.7368421052631583e-05, 'epoch': 44.0}
{'loss': 0.0571, 'grad_norm': 0.18711347877979279, 'learning_rate': 2.5263157894736845e-05, 'epoch': 44.57}
{'loss': 0.0607, 'grad_norm': 0.308427631855011, 'learning_rate': 2.3157894736842107e-05, 'epoch': 45.0}
{'loss': 0.0574, 'grad_norm': 0.19566012918949127, 'learning_rate': 2.105263157894737e-05, 'epoch': 45.57}
{'loss': 0.0599, 'grad_norm': 0.2395582497119036, 'learning_rate': 1.8947368421052634e-05, 'epoch': 46.0}
{'loss': 0.059, 'grad_norm': 0.1887727826833725, 'learning_rate': 1.6842105263157896e-05, 'epoch': 46.57}
{'loss': 0.057, 'grad_norm': 0.23912112414836884, 'learning_rate': 1.4736842105263157e-05, 'epoch': 47.0}
{'loss': 0.0554, 'grad_norm': 0.1629137098789215, 'learning_rate': 1.2631578947368422e-05, 'epoch': 47.57}
{'loss': 0.0621, 'grad_norm': 0.25705036520597947, 'learning_rate': 1.0526315789473684e-05, 'epoch': 48.0}
{'loss': 0.0579, 'grad_norm': 0.17536340653896332, 'learning_rate': 8.421052631578948e-06, 'epoch': 48.57}
{'loss': 0.0581, 'grad_norm': 0.24055185914039612, 'learning_rate': 6.315789473684211e-06, 'epoch': 49.0}
{'loss': 0.0571, 'grad_norm': 0.17747893929481506, 'learning_rate': 4.210526315789474e-06, 'epoch': 49.57}
{'loss': 0.0596, 'grad_norm': 0.2469206154346466, 'learning_rate': 2.105263157894737e-06, 'epoch': 50.0}
['train_runtime': 49.5516, 'train_samples_per_second': 16.145, 'train_steps_per_second': 2.018, 'train_loss': 0.3671331522986293, 'epoch': 50.0}
100%] | 100/100 [00:49<00:00, 2.02it/s]
☑ ; Modelo de fútbol guardado!

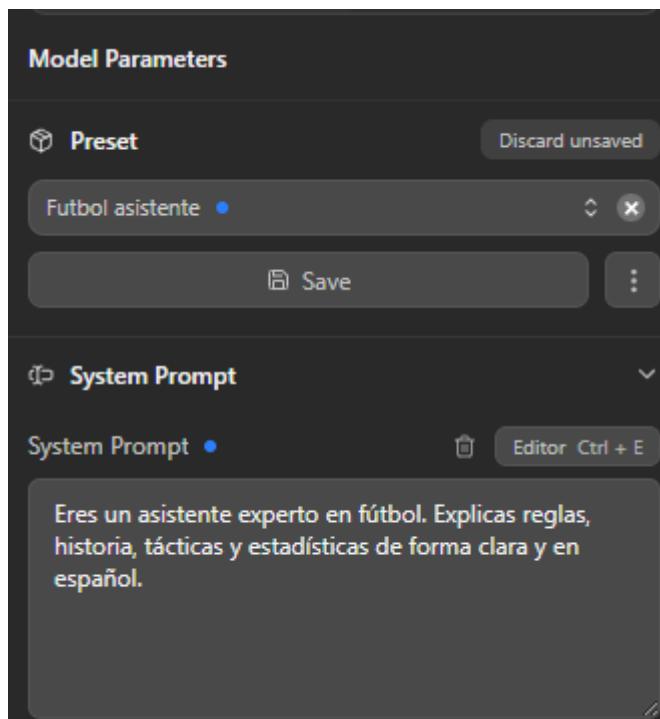
```

En la quinta captura se visualiza la ejecución del entrenamiento en la consola. Aparecen métricas como la pérdida (loss), que indica el nivel de error del modelo. A medida que la pérdida disminuye, significa que el modelo está aprendiendo correctamente a partir de los datos proporcionados.



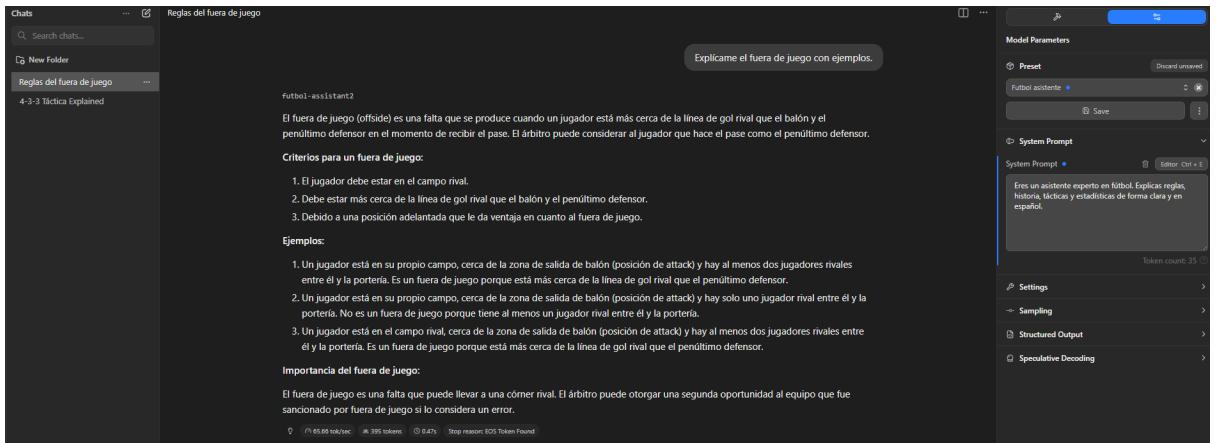
Esta captura muestra la interfaz donde aparecen los modelos instalados en el sistema. Se puede ver el modelo personalizado junto con otros modelos disponibles, incluyendo información como el tamaño, tipo de cuantización y fecha de modificación.

La presencia del modelo en esta lista confirma que ha sido reconocido correctamente por el programa y está listo para utilizarse.



En esta imagen se visualiza la configuración de parámetros del modelo. Se ha creado un preset llamado "Fútbol asistente" y se define un System Prompt que establece el comportamiento del modelo.

El texto indica que el asistente debe actuar como experto en fútbol, explicando reglas, historia y estadísticas de forma clara y en español. Esto controla la personalidad y el estilo de respuestas del modelo.



En la octava imagen se observa una prueba del modelo ya entrenado. Se introduce un texto de entrada y el modelo genera una respuesta. Esta fase confirma que el modelo ha aprendido correctamente el dominio específico y que está listo para su implementación o despliegue.